3D Hand Pose Estimation from Single RGB Images with Auxiliary Information

Dissertation

zur

Erlangung des Doktorgrades (*Dr. rer. nat.*)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich–Wilhelms–Universität Bonn

vorgelegt von

Linlin YANG

aus

V.R. China

Bonn 2022

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachterin und Betreuerin: Prof. Dr. Angela Yao
2. Gutachter: Prof. Dr. Reinhard Klein
Tag der Promotion: 23.11.2022
Erscheinungsjahr: 2022

Abstract

by Linlin Yang

for the degree of

Doctor rerum naturalium

3D hand pose estimation from monocular RGB inputs is critical for augmented and virtual reality applications, and has achieved remarkable progress due to the revolution of deep learning. Existing deep-learning-based hand pose estimation systems target learning good representations for hand poses, requiring a large amount of accurate ground truth labels, which are difficult to obtain. We turn to explore different auxiliary information to aid representation learning and reduce the reliance on data annotation. This dissertation explores different auxiliary information, *i.e.* , image factors, multi-modal data, and synthetic data, for 3D hand pose estimation.

Motivated by the image rendering that requires a number of image factors of variation, we propose to learn disentangled representations to better analyze these factors of variation. The disentangled representations enable explicit control over different factors of variation for synthesizing hand images and training with hand factors as weak labels for hand pose estimation. Besides labelled or shared hand factors, different modalities (*e.g.* , RGB images and depth maps) of the same hand should have shared information. Therefore, we present multi-modalities as auxiliary information for RGB inputs. Specifically, we explore multi-modal alignment in three aspects: latent space alignment based on variational autoencoder and product of Gaussian expert, pixel-level alignment via attention fusion, and low-dimensional subspace alignment via contrastive learning. Besides multi-modal alignment, the auxiliary modalities can also serve as weak labels for hand pose estimation.

To further remove the requirements of image factors or different modalities, we emphasize the importance of synthetic data. Synthetic data is flexible, infinite, and easy to achieve. With synthetic data as auxiliary information, we can significantly reduce the number of labelled real-world data. Therefore, we introduce a challenging scenario that learns only from labelled synthetic data and fully unlabelled real-world data. To address this challenging scenario, we present a semi-supervised framework with pseudo-labelling and consistency training, and try to address noisy pseudo-labels using modules like label correction and self-distillation.

This dissertation advances the state-of-the-art 3D hand pose estimation, explores representation learning, weakly- and semi-supervised learning for pose estimation, and paves a path forward for learning pose estimation with diverse auxiliary information.

Keywords: 3D Hand Pose Estimation, Weakly-Supervised Learning, Semi-Supervised Learning, Multi-Modal Learning.

Zusammenfassung

von Linlin Yang

zur Erlangung des Doktorgrades

Doctor rerum naturalium

Die 3D-Handhaltungsschätzung, basierend auf den monokularen RGB-Eingaben, ist für die Anwendungen der Augmented Reality (AR) und Virtual Reality (VR) von großer Bedeutung. Aufgrund der Revolution des Deep Learnings wurden bemerkenswerte Fortschritte in dem Bereich der 3D-Handhaltungsschätzung erzielt. Die bestehenden Deep-Learning-basierten Systeme zur Handhaltungsschätzung zielen darauf ab, gute Darstellungen für Handhaltungen zu lernen, was eine große Menge an genauen Ground-Truth-Etiketten erfordert, die schwer zu erhalten sind. Daher untersuchen wir diverse Hilfsinformationen, um das Repräsentationslernen zu unterstützen und die Abhängigkeit von Datenannotationen zu verringern. In der Dissertation werden diverse Hilfsinformationen für die 3D-Handhaltungsschätzung erforscht, d.h. Bildfaktoren, multimodale Daten und synthetische Daten.

Die Bildwiedergabe erfordert eine Reihe von Bildvariationsfaktoren. Wir werden dadurch inspiriert und schlagen vor, die entwirrten Darstellungen zu lernen, um diese Variationsfaktoren besser zu analysieren. Die entwirrten Darstellungen ermöglichen eine explizite Kontrolle über verschiedene Variationsfaktoren zum Synthetisieren von Handbildern und ein Training mit Handfaktoren als schwache Etiketten für die Schätzung der Handhaltung. Neben beschrifteten oder gemeinsam genutzten Handfaktoren sollten verschiedene Modalitäten (z.B. RGB-Bilder, Tiefenkarten) von derselben Hand gemeinsame Informationen haben. Daher stellen wir Multimodalitäten als Hilfsinformation für RGB-Eingaben vor. Insbesondere untersuchen wir die multimodale Ausrichtung aus drei Aspekten: Die auf Variations-Autoencoder und dem Produkt des Gaußschen Experten basierende Ausrichtung des latenten Raums, die Ausrichtung auf die Pixelebene durch Aufmerksamkeitsfusion und die Ausrichtung im niedrigdimensionalen Subraum durch kontrastives Lernen. Neben der multimodalen Ausrichtung können die Hilfsmodalitäten auch als schwache Etiketten für die Schätzung der Handhaltung dienen.

Um die Anforderungen bestimmter Faktoren oder verschiedener Modalitäten weiter zu beseitigen, wird die Bedeutung synthetischer Daten hervorgehoben. Synthetische Daten sind flexibel, unendlich und einfach zu erreichen. Mit synthetischen Daten als Hilfsmittel können wir die Anzahl gekennzeichneter realer Daten erheblich reduzieren. Daher führen wir ein Herausforderungsszenario ein, bei dem nur aus gekennzeichneten synthetischen Daten und vollständig nicht gekennzeichneten Daten aus der realen Welt gelernt wird. Um dieses Herausforderungsszenario anzugehen, stellen wir ein semiüberwachtes Framework mit Pseudo-Label und Konsistenztraining vor. Damit versuchen wir, laute Pseudo-Labels durch Module wie Korrektur und Selbstdestillation des Labels zu beheben.

Diese Dissertation bringt den Stand der Technik in der 3D-Handhaltungsschätzung voran, erforscht das Repräsentationslernen, schwach und semi-überwachtes Lernen für die Haltungsschätzung und ebnet einen Weg für das Lernen der Haltungsschätzung mit verschiedenen Hilfsinformationen.

Schlagwörter: Schätzung der Handhaltung, schwach überwachtes Lernen, semi-überwachtes Lernen, multimodales Lernen..

Publications

The following first/co-first author publications are included in this dissertation:

- Linlin Yang and Angela Yao. "Disentangling Latent Hands for Image Synthesis and Pose Estimation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019. DOI:10.1109/CVPR.2019.01011
- Linlin Yang*, Shile Li*, Dongheui Lee and Angela Yao. "Aligning Latent Spaces for 3D Hand Pose Estimation." *International Conference on Computer Vision (ICCV)*. 2019. * equal contribution. DOI:10.1109/ICCV.2019.00242
- Linlin Yang, Shicheng Chen and Angela Yao. "SemiHand: Semi-supervised Hand Pose Estimation with Consistency." *International Conference on Computer Vision (ICCV)*. 2021. DOI:10.1109/ICCV48922.2021.01117
- Qiuxia Lin*, Linlin Yang* and Angela Yao. "Dual-Modality Network for Semi-Supervised Hand Pose Estimation." *In Submission*. 2022. * equal contribution.

Additionally, I contributed to the following publications related to hand pose estimation, which are not included in this dissertation:

- Kerui Gu, Linlin Yang and Angela Yao. "Removing the Bias of Integral Pose Regression." *International Conference on Computer Vision (ICCV)*. 2021. DOI:10.1109/ICCV48922.2021.01088
- Ziwei Yu, Linlin Yang, Shicheng Chen, Angela Yao. "Local and Global Point Cloud Reconstruction for 3D Hand Pose Estimation." *British Machine Vision Conference (BMVC)*. 2021. URL: https://www.bmvc2021-virtualconference.com/assets/papers/0817.pdf
- Kerui Gu, Linlin Yang, Angela Yao. "Dive Deeper Into Integral Pose Regression." *International Conference on Learning Representations (ICLR)*. 2022. URL: https://openreview.net/forum?id=vHVcB-ak3Si

Acknowledgements

I would like to thank the following people, without whom I would not have been able to complete this dissertation.

First of all, I would like to thank my supervisor, Prof. Dr. Angela Yao. Thank you for giving me the opportunity to pursue my dreams in Germany and Singapore. Also, thank you for giving me the freedom and courage to pursue my ideas and challenge myself. You taught me how to write scientific papers, have “big picture” thinking, and be a super nice person. Without your encouragement, enthusiasm, tireless guidance, and support, it is impossible for me to finish my research and write this dissertation. It is a pleasure and honor to study under your supervision.

Moreover, thanks to my co-supervisor Prof. Dr. Reinhard Klein, for solving my problems and giving me help in my life in Bonn.

I would also like to thank my colleagues, Chengde, Fadime, Soumajit and Moritz. I enjoyed the daily chat and discussion we had at the office. Also, thank you very much for providing valuable experience to help me take less detours. Thanks to Shile and Prof. Dr. Dongheui Lee from TUM, for being my collaborators and for so many discussions. This is my first collaboration, broadening my horizons and encouraging me to explore more relevant fields. Thanks to all the members of the Computer Graphics Group in Bonn. Special thanks to Michael, Peng, Ruotong, Sebastian, and Simone for giving me much help in my life and study in Bonn. Thanks to all the Computer Vision and Machine Learning Group (CVML) members in Singapore. The cooperation and discussion helped us learn from each other and think from different perspectives. Special thanks to Kerui, Qiuxia, Qiyuan, Rongyu, Shicheng, and Ziwei, for our collaborative effort in the submissions.

Finally, thanks to my family for supporting and encouraging me during all my years of study. Special thanks to Wanjun; love grows in Beijing and Bonn; we are fortunate to accompany each other.

Contents

List of Figures **xii**

List of Tables **xiii**

Nomenclature **xvi**

1 Introduction **1**

1.1 What is 3D Hand Pose Estimation? 1

1.2 Motivation 3

1.3 Contribution of the Dissertation 4

1.3.1 Learning with Image Factors 4

1.3.2 Learning with Multi-Modalities 5

1.3.3 Learning with Synthetic Data 7

1.4 Organization 8

2 Preliminaries **11**

2.1 Datasets 11

2.2 Architectures 13

2.3 Coordinate Representation 14

2.3.1 Heatmap-based Representation 15

2.3.2 Numerical Coordinate-based Representation. 16

2.4 Hand Surface Model 17

2.4.1 3D Approximation Model 17

2.4.2 3D Mesh Model 18

2.5 Evaluation Metric 19

3 Related Works **21**

3.1 Multiple Tasks 22

3.1.1 Modality-based Multi-Task Learning 22

3.1.2 Structure-based Multi-Task Learning 25

3.2 Domain Adaptation 26

3.2.1 Transfer from Different Modalities 26

3.2.2 Transfer from Synthetic Data 28

3.2.3 Others 28

3.3 Hand Models 29

3.3.1 Model-driven Paradigm 29

3.3.2 Data-driven Paradigm 30

3.3.3 Hybrid Paradigm 31

3.4 Representation Learning 32

3.4.1 Disentangled Representation 32
3.4.2 Contrastive Learning 34
3.5 Less Supervision 35
3.5.1 Weakly-Supervised learning 36
3.5.2 Semi-Supervised learning 37
3.5.3 Hand Pose Correction 39

4 Disentangling Latent Hands for Image Synthesis and Pose Estimation 41
4.1 Motivation 43
4.2 Related Works 45
4.2.1 Hand Pose Estimation 45
4.2.2 Disentangled Representations 45
4.3 Methodology 46
4.3.1 Cross Modal VAE 46
4.3.2 Disentangled VAE 46
4.3.3 Applications 49
4.4 Experimentation 51
4.4.1 Implementation Details 52
4.4.2 Datasets & Evaluation 52
4.4.3 Synthesizing Images 52
4.4.4 3D Hand Pose Estimation 53
4.5 Conclusion 55

5 Aligning Latent Spaces for 3D Hand Pose Estimation 57
5.1 Motivation 59
5.2 Related Works 60
5.3 Methodology 61
5.3.1 Cross Modal VAE and Its Extension 61
5.3.2 Latent Space Alignment 62
5.3.3 Application Towards Hand Pose Estimation 65
5.4 Implementation Details 65
5.4.1 Data Pre-Processing and Augmentation 65
5.4.2 Encoder and Decoder Modules 66
5.5 Experimentation 67
5.5.1 Datasets and Evaluation Metrics 68
5.5.2 Qualitative Results 69
5.5.3 RGB 3D Hand Pose Estimation 69
5.6 Conclusion 72

6 SemiHand: Semi-supervised Hand Pose Estimation with Consistency 73
6.1 Motivation . . . 75
6.2 Related Works . . . 77
6.2.1 3D Hand Pose Estimation . . . 77
6.2.2 Semi-Supervised Learning . . . 77
6.3 Methodology . . . 78
6.3.1 Supervised Pose Estimation . . . 79
6.3.2 Pseudo-labels for Pose Estimation . . . 79
6.3.3 Self Consistency for Pose Estimation . . . 80
6.3.4 Data Augmentation . . . 83
6.4 Experiments . . . 83
6.4.1 Implementation Details . . . 83
6.4.2 Datasets and Evaluation Metrics . . . 83
6.4.3 Ablation Study . . . 84
6.4.4 Comparison to State-of-the-Art . . . 85
6.4.5 Comparison to Weakly-Supervised Methods . . . 86
6.5 Conclusions . . . 88

7 Dual-Modality Network for Semi-Supervised Hand Pose Estimation 89
7.1 Motivation . . . 91
7.2 Related Work . . . 92
7.2.1 Contrastive Learning . . . 92
7.2.2 Semi-Supervised Learning . . . 93
7.3 Method . . . 93
7.3.1 Problem Definition . . . 93
7.3.2 Multi-modal Supervision . . . 94
7.3.3 Model Architecture . . . 94
7.3.4 Pre-training with Multi-modal Alignment . . . 96
7.3.5 Fine-tuning with Noisy Pseudo-Labels . . . 98
7.4 Experiments . . . 99
7.4.1 Datasets & Evaluation . . . 99
7.4.2 Implementation Details . . . 100
7.4.3 Augmentation . . . 100
7.4.4 2D Keypoint Detection . . . 101
7.4.5 3D Keypoint Estimation . . . 101
7.4.6 Ablation Study . . . 102
7.4.7 Qualitative Results . . . 103
7.5 Conclusion . . . 104

8 Conclusions **105**
8.1 Summary 105
8.2 Discussion and Future Work 107
8.2.1 Multi-Modal Data 107
8.2.2 Disentangled Representation 108
8.2.3 Synthetic Data 108
8.2.4 Interpretation Technique 109
8.2.5 Real-world Application 109

Bibliography **111**

List of Figures

1.1 The illustration of hand joint locations, hand root and reference bone. “Wrist”, “MCP”, “PIP”, “DIP”, “TIP”, denote wrist, MetaCarpoPhalangeal joints, Proximal InterPhalangeal joints, Distal InterPhalangeal joints, fingertips respectively. “T”, “I”, “M”, “R”, “P” denote Thumb, Index, Middle, Ring, Pinky fingers respectively. The hand root and reference bone are based on [178]. . . 2

1.2 Application of hand pose estimation. Images are from [1, 45, 101, 141] 3

1.3 Illustration of depth ambiguities associated with monocular RGB images. We can see that different 3D poses may project into the same 2D locations in the image coordinate. 4

1.4 Illustration of two annotation equipment and their labelled examples. Images from [165, 33, 179] 4

1.5 Pipeline of hand image rendering. Based on hand image factors of variation like 3D hand models, backgrounds, lights and camera viewpoints, we can generate hand images by means of a renderer like Blender. 5

1.6 Examples of synthetic data. Images from [153]. 7

2.1 Illustration of three common network architectures for pose estimation, (a) Hourglass [84], (b) Cascaded Pyramid Network [22], and (c) PoseResNet [149]. Images are from [149]. 13

2.2 Comparison of 2D numerical coordinate-based representations (top) and 2D heatmap-based representations (bottom) of fingertips from one hand. Red double headed arrows indicate their correspondences. 15

2.3 3D hand surface models. (a) Sum-of-Gaussians model, (b) Primitives approximation, (c) Sphere-Meshes can be thought of as a generalization of the previous models, (d) Loop Subdivision Surface of a triangular control mesh , (e) hand MANO model. Images reproduced from [100, 132]. 17

3.1 Illustration of latent 2.5D heatmap regression. Given a cropped RGB hand image, the network produces latent heatmaps including K latent 2D heatmaps H^{*2D} and K latent depth maps $H^{*\hat{z}^r}$ for K joints. To approximate 2D heatmaps with an argmax operation but be differentiable, the latent 2D heatmaps are decoded into the numerical coordinates by applying a softmax normalization and an expectation operation. To estimate the normalized depth values, we use the summation of $H^{\hat{z}^r}$, which obtained by multiplying $H^{*\hat{z}^r}$ with H^{2D}, *i.e.* , the softmax normalized H^{*2D}. The image is from [50]. 23

3.2 A cross-modal prediction pipeline with shared latent space. The RGB images $\mathbf{x}_{RGB}$ and 3D poses $\mathbf{x}_{3D}$ are encoded into the same latent space via a RGB encoder $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and a 3D pose encoder $q_2(\mathbf{z}|\mathbf{x}_{3D})$. With a latent variable $\mathbf{z}$, we can decode $\mathbf{z}$ into $\mathbf{x}_{RGB}$ or $\mathbf{x}_{3D}$ via a RGB decoder $p_1(\mathbf{x}_{RGB}|\mathbf{z})$ and a 3D pose decoder $p_2(\mathbf{x}_{3D}|\mathbf{z})$, respectively. We formulate an image reconstruction task with $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and $p_1(\mathbf{x}_{RGB}|\mathbf{z})$, and a hand pose estimation task with $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and $p_2(\mathbf{x}_{3D}|\mathbf{z})$. The image is from [111]. 24

3.3 Illustration of two commonly used hand joints grouping. Different colours indicate different groups. Based on hand palm and fingers, works [106, 29] divide hand joints into palm joints and finger joints. Based on the function of different fingers, the work [29] divides hand joints into thumb finger joints, index finger joints and others. 26

3.4 Exploration of depth maps for enhancing RGB-based 3D hand pose estimation. There are two branches, a depth branch (top) and a RGB branch (bottom). First, two branches are pre-trained independently to minimize 3D pose loss to exploit external large-scale depth maps. Then, based on paired RGBD data, two branches are fine-tuned based on their 3D pose losses and an intermediate feature loss which encourages the features from RGB branch to mimic the responses of the corresponding features from depth branch. This feature imitation transfers common visual cues from the depth branch to the RGB branch. The image is from [163]. 27

3.5 Comparison of the data-driven paradigm (a) and the model-driven paradigm (b). We can see the data-driven paradigm requires training data during training and get one-shot estimation during inference. In contrast, the model-driven paradigm without training needs several optimization steps to model-fit to the references. Images reproduced from [132]. 30

3.6 Illustration of a classic hybrid pipeline. For the data-driven part, given a depth image as input, the correspondence regression network (CoRN) provides segmentation masks and correspondence maps as references. For the model-driven part, an energy minimization optimization is adopted to fit the parameters of a parametric model (MANO) to best explain the references from data-driven part. Note that even this framework is designed for a depth map input, it is also applicable for an RGB-based framework directly. The image is from [82]. 31

3.7 Illustration of a disentangled representation-based framework for unsupervised domain adaptation. The source data $\mathbf{x}^s$ and the target data $\mathbf{x}^t$ are passed to a task-relevant encoder E_c and a domain-relevant encoder E_d, with output features $\mathbf{z}_c$ and $\mathbf{z}_d$, respectively. To encourage $\mathbf{z}_c$ and $\mathbf{z}_d$ to be disentangled, task-relevant loss (content loss), domain loss and mutual information minimization loss are adopted. C is the content decoder to get task-relevant predictions, and D is the domain discriminator. The mutual information between $\mathbf{z}_c$ and $\mathbf{z}_d$ is minimized. The image is from [23]. 33

3.8 Illustration of a simple framework for contrastive learning. A positive pair $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ is generated based on the same data $\mathbf{x}$ but with different augmentations. The positive pair is then passed to a base encoder network $f(\cdot)$ and a projection head $g(\cdot)$, with representations $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, and output $\mathbf{z}_i$ and $\mathbf{z}_j$. $f(\cdot)$ and $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training, the representations h can be used for downstream tasks. The image is from [17]. . 34

3.9 Illustration of two properties of contrastive representations, alignment and uniformity of feature distributions on the output unit hypersphere. Contrastive learning makes similar sample pairs stay close together while dissimilar samples are further apart. Also contrastive learning encourages features uniformly distributed throughout the hyper-sphere feature space. Images are from [138]. 35

3.10 Illustration of a pre-training and fine-tuning pipeline for 3D hand pose estimation. During pre-training, a contrastive learning strategy for a large unlabelled hand image dataset is adopt to yield useful image embeddings that can be used for the downstream hand pose estimation task. During fine-tuning, the encoder is fine-tuned based on a smaller labelled dataset. using a smaller learning rate. The image is from [177]. 36

3.11 Illustration of a weakly-supervised pipeline based on MANO and a weak perspective camera model. The pipeline takes as input a hand image and optionally 2D joint heatmaps. The encoder produces shape parameters and pose parameters for MANO to generate 3D poses, and view parameters for camera model. Using orthographic re-projection, the pipeline exploits 2D poses as weak labels for 3D hand pose estimation.The image is from [9]. 37

3.12 Illustration of a semi-/self-supervised pipeline. It adopts synthetic data to warm up the pose estimation network and uses consistencies within the data and the hand model to serve as a label for learning. The image is from [132]. 38

4.1 Illustration of dVAE. The red lines denote variational approximations while the black lines denote the generative model. With the help of labelled factors of variations (*e.g.* pose, viewpoint and image content), we learn a disentangled and specifiable representation for RGB hand images in a VAE framework. . . 44

4.2 Graphical models of disentangled VAEs. The shaded nodes represent observed variables while un-shaded nodes are latent. The red and black solid lines denote variational approximations q_ϕ or encoders, and the generative models p_θ or decoders respectively. The dashed lines denote deterministically constructed variables. Figure best viewed in colour. 47

4.3 Inference models for the tasks of image synthesis (left and middle) and pose estimation (right). 50

4.4 Latent space walk. The images in the red boxes are provided inputs. The first two rows show synthesized images when interpolating on the latent 3DPose space; the third row shows skeletons of the reconstructed 3DPose. The fourth row shows synthesized images when the pose is fixed (to the fourth column) when interpolating in the content latent space. 51

4.5 Latent space walk, interpolating $\mathbf{z_u}$ representing image background content. The images along with groundtruth 3DPose (red) in the red box are the input points; the first row shows generated images and the second row corresponding reconstructed 3DPose (blue). Note that because we are interpolating only on the background content, the pose stays well-fixed. 53

4.6 Pose transfer. The first column corresponds to images from which we extract the 3DPose (ground truth pose in second column); the first row corresponds to tag images columns we extract the latent content; the 2-3 rows, 3-5 columns are pose transferred images. 53

4.7 Quantitative evaluation. 3D PCK on RHD (left) and STB (middle). Mean EPE (mm) on RHD and STB (right). 54

4.8 CPose and 3DPose estimation on RHD and STB. For each quintet, the left most column corresponds to the input images, the second and the third columns correspond to CPose groundtruth (red) and our prediction (blue), the right most two columns correspond to 3DPose groundtruth (red) and our prediction (blue). 54

4.9 Mean EPE of our model on the semi-supervised setting and the weakly-supervised setting. 55

5.2 Graphical models. (a) Cross modal; (b) Extended cross modal; (c) Latent alignment with a KL divergence loss; (d) Latent alignment with the product of Gaussian experts. The shaded nodes represent observed variables while un-shaded nodes are latent. The red and black solid lines denote variational approximations q_ϕ or encoders, and the generative models p_θ or decoders respectively. The dashed lines denote the operation that embedding cross-modal latent spaces into a joint shared latent space; it is a KL divergence optimization for (c) and product of Gaussian experts for (d). Figure best viewed in colour. 63

6.1 Pseudo-labelling of SemiHand. Our pseudo-label with confidence is generated based on the prediction from original (blue pose), the prediction from perturbation (green pose) and the corrected prediction (red pose). 76

6.2 Overview of SemiHand. The model is pre-trained on labelled synthetic data. Consistency training (orange double headed arrow, see Sec. 6.3.3) on unlabelled real-world data with perturbation augmentations (see Sec. 6.3.4) and label correction and sample selection (blue dash-dotted arrow, See Fig. 6.1 and Sec. 6.3.2) together with augmentation of differing difficulties. (see Sec. 6.3.4). 78

6.3 Overview of view consistency loss. 81

6.4 Hand model and pose registration. Left: the ground-truth hand mask; Middle: Our rendered hand mask based on ground-truth 2D pose (blue points); Right: pose registration of the template hand (black) to observed joints (grey) to result in a registered hand (orange). Figure best viewed in colour. 82

6.5 Overview of cross-modal consistency loss. (uv, d) are 2.5D hand outputs; **w** denotes the hand mask. 82

6.6 Comparison of baseline, with only consistency training, with only pseudo-labeling and our proposed SemiHand. Our proposed two modules both improve the performance with respect to the baselines, and their combination further leads to a higher accuracy. 85

6.7 AUC: Comparison to state-of-the-art on STB. Our SemiHand improves the baseline's AUC and achieves comparable performance to other supervised learning methods. 86

6.8 AUC: Comparison to state-of-the-art on DO. Our SemiHand improves the baseline's AUC and outperforms some supervised learning methods using the mix-and-train strategy. 87

6.9 Gradual convergence from the prediction of pre-trained model to our final prediction. The arrows indicate the direction and distance of prediction movement during fine-tuning. For 10$^{\text{th}}$iteration, the optimization converges because the length of arrows become almost zeros. We highlight the differences between our stable predictions and the ground-truth poses with red boxes. Figure best viewed in colour. 87

6.10 Mean EPE on STB testing data with fine-tuning on different percentage of STB training data. As the amount of training data increases, SemiHand achieves a similar trend as the weakly-supervised methods, i.e., the mean EPE decreases correspondingly. 88

7.1 Visualisation of one prediction from FreiHAND before and after fine-tuning. From left to right: multi-modal predictions (depth maps, segmentation masks, poses), attention weights from depth maps and attention weights on RGB image. Our model takes in multi-modal predictions based on the pretrained model and monocular RGB and output corrected multi-modal predictions. . . 92

7.2 (a) Overview of our proposed dual-modality, *i.e.* , the RGB modality and the depth map (DM) modality, and the three branches framework, *i.e.* , the RGB branch (blue arrows), DM branch (green arrows) and fusion branch (red arrows); (b) illustration of the shared decoder and the multi-modal output; (c) design of the attention module. Note that we only use the RGB branch during inference. 94

7.3 The illustration of (a) the RGB branch, (b) the DM branch, (c) the fusion branch and (d) the overall losses during pre-training. The yellow and blue arrows show the forward of a positive pair. The superscript $^{+}$ denotes the positive paired sample. 96

7.4 (a) The architecture and (b) overall losses during fine-tuning. The yellow and blue arrows show the forward of a weakly-strongly augmented image pair. . . 98

7.5 2D pose visualization on STB, H3D and HSD. We compare our method with four state-of-the-art methods and highlight the differences between the predictions and the ground truth poses with red boxes. Figure best viewed in colour. 103

7.6 Visualisation of two examples before fine-tuning (first row) and after fine-tuning (second row). From left to right: RGB images, multi-modal predictions (depth maps, segmentation masks, poses), attention weights from depth maps and attention weights on RGB image. Figure best viewed in colour. 104

8.1 Our goal is to utilize more auxiliary information and less annotation information for model training. The exploration is in two directions. First, with auxiliary information, we aim to achieve better representations by means of representation learning. Second, we would like to exploit auxiliary information for weakly-/semi- supervise learning and hence reduce the reliance on annotation. However, until now, even with diverse auxiliary information, the accuracy still has not been close to that in supervised learning. 105

List of Tables

2.1 Comparison of existing RGB-based hand pose benchmarks. We first present two synthetic datasets: RHD and ObMan. We then list real-world datasets: STB, DO, YT3D, H3D and FreiHAND. Here, "Modality" corresponds the available input source of dataset, "Resolution" corresponds the resolution of input source, "Subjects" corresponds the number of hands, "Views" corresponds the number of viewpoints, "Frames" corresponds the number of images, "Syn/Real" indicates the dataset is either synthetic data or real-world data, "Joints" shows the number of keypoint and either in 3D space or in 2D space, "MANO" shows if MANO parameters are provided, "Annotation" shows the type of annotation method. 12

5.1 Comparison of different training strategies on the RHD dataset. The mean EPE values are obtained from monocular RGB images. (R: RGB, C: point cloud, P: pose, H: heatmap). Poses estimated from monocular RGB images can be improved by increasing number of different encoders and decoders during training. 69

5.2 Comparison to state-of-the-art on the RHD and STB with mean EPE [mm]. Ours refers to S4 in Table 1 (RC2CHP). 70

6.1 Ablation study with mean EPE [mm]. w/ and w/o indicates with and without labels for training. 84

7.1 (a) The performance comparisons for 2D keypoint detection; (b) the performance comparisons for 3D keypoint estimation. The relative performance boost between 1st and 2nd best methods can be seen in *Improvement*. Our approach brings consistent improvements over previous state-of-the-art methods. **Bold** numbers indicate the best performance. 101

7.2 (a) Ablation study for the components of our approach. Adding (+) the components incrementally improves performance. (b) Multi-modal contrastive learning with different positive pairs. RGB only: training on $\mathcal{L}_c^{RGB}$; RGB & DM: training on $\mathcal{L}_c^{RGB}$ and $\mathcal{L}_c^{DM}$; RGB & DM*: training on $\mathcal{L}_c^{RGB}$, $\mathcal{L}_c^{DM}$ and $\eta(\boldsymbol{z}_{RGB}, \boldsymbol{z}_{DM})$; w/ and w/out SG: denotes training on $\mathcal{L}_c$ with and without the stop-gradient operation respectively. Our full model ("w/ SG") correctly leverages the depth map modality and generates better representations for the cross-domain dataset. **Bold** indicates the best performance. 102

Nomenclature

Abbreviations

An alphabetically sorted list of abbreviations used in the dissertations:

3DPose	3D Hand Pose
AR	Augmented Reality
AUC	Area under the ROC Curve
CNN	Convolutional Neural Network
CPN	Cascaded Pyramid Network
CPose	Canonical 3D Hand Pose
DIP	Distal InterPhalangeal
DM	Depth Map
DOF	Degree of Freedom
dVAE	disentangled VAE
ELBO	Evidence Lower Bound
EMD	Earth Mover's Distance
EPE	End Point Error
FCN	Fully Convolutional Network
GAN	Generative Adversarial Network
HCI	Human and Computer Interaction
ICP	Iterated Closest Point
KL	Kullback Leibler
LBS	Linear Blend Skinning
LM	Levenberg Marquard
MANO	hand Model with Articulated and Non-rigid defOrmations
MCP	MetaCarpoPhalange
MLP	Multilayer Perceptron
MMD	Maximum Mean Discrepancy
MoCap	Motion Capture
PCK	Percentage of Correct Keypoints
PEL	Permutation Equivariant Layer
PIP	Proximal InterPhalangea
PoE	Product of Experts
PSO	Particle Swarm Optimization
RDF	Random Decision Forest
RDT	Random Decision Tree
ROC	Receiver operating characteristic curve
SoG	Sum of Gaussians
VAE	Variational Autoencoder
VR	Virtual Reality

Chapter 1

Introduction

Contents

1.1 What is 3D Hand Pose Estimation? 1

1.2 Motivation 3

1.3 Contribution of the Dissertation 4

1.3.1 Learning with Image Factors 4

1.3.2 Learning with Multi-Modalities 5

1.3.3 Learning with Synthetic Data 7

1.4 Organization 8

1.1 What is 3D Hand Pose Estimation?

The hand is the most frequently used body part in our daily activities, performing everyday tasks and interacting with everyday objects. It is the most valuable tool we have no matter we are at work, at home, or at play. With realizing the importance of our hands, we begin by introducing the research of hands "3D hand pose estimation" throughout this dissertation.

The goal of 3D hand pose estimation is to predict hand joint locations in 3D world space. Based on the kinematic configuration of hand, existing works model hand using 21 joints [178]. With the 21 joints skeleton hand model, we can recover our hand poses accurately. Specifically, the 21 joints include a hand wrist (Wrist) and Distal InterPhalangeal joints (DIP), Proximal InterPhalangeal joints (PIP), MetaCarpoPhalangeal joints (MCP), hand fingertips (TIP) for each finger as shown in Fig. 1.1.

Being able to accurately estimate articulated hand enables many applications. As shown in Fig. 1.2 (a), with pose estimation, we can achieve the re-targeting of human hand for robotic hand and hence remote control the robot for many tasks like object grasping and object handling. Besides that, pose estimation is also a prerequisite of different tasks. For example, with the help of given pose sequences, we can analyze the activities (Fig. 1.2 (b)) and recognize the sign language (Fig. 1.2 (c)). Furthermore, hand is the most natural, comfortable and immersive interaction with objects in virtual reality (VR) and augmented reality (AR) environments. This makes hand pose estimation the key to the next generation of human and computer interaction (HCI) as shown in Fig. 1.2 (d).

Thanks to the development of deep learning and depth sensors during the past decade, early works for hand pose estimation revolve around using depth maps as input and have

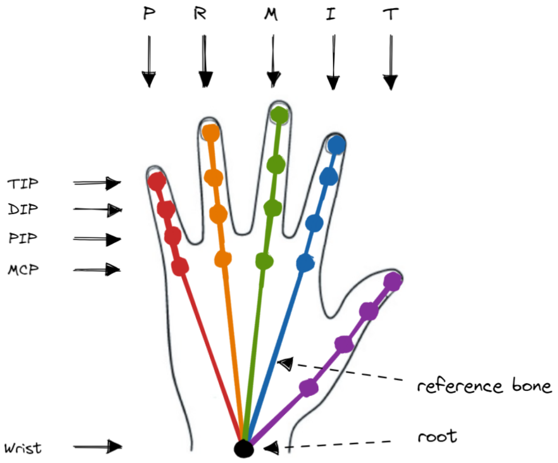
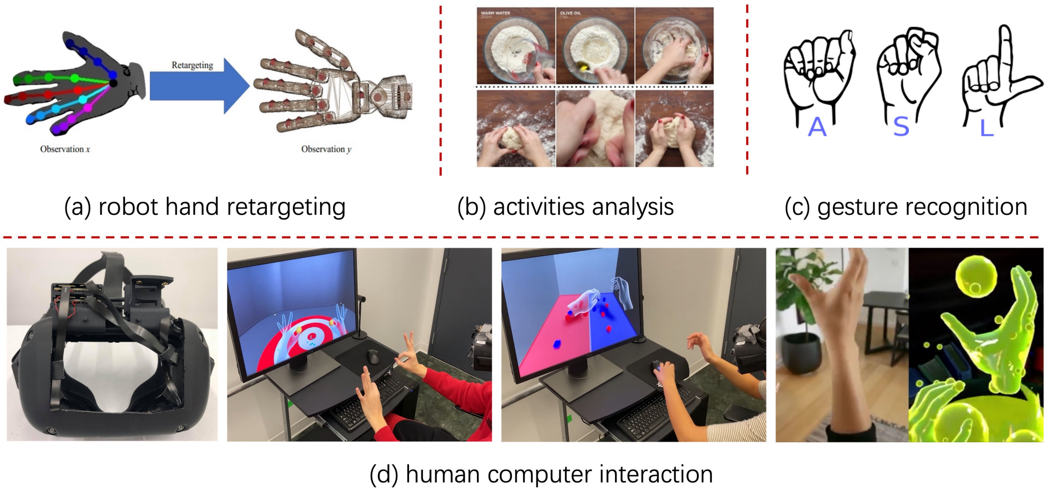


Figure 1.1: The illustration of hand joint locations, hand root and reference bone. “Wrist”, “MCP”, “PIP”, “DIP”, “TIP”, denote wrist, MetaCarpoPhalangeal joints, Proximal InterPhalangeal joints, Distal InterPhalangeal joints, fingertips respectively. “T”, “I”, “M”, “R”, “P” denote Thumb, Index, Middle, Ring, Pinky fingers respectively. The hand root and reference bone are based on [178].

achieved a high degree of accuracy. However, consumer depth sensors may suffer from limited spatial resolution, depth measurement noise and restricted operating ranges, and hence their usage scenario is still limited to indoor environments. In this case, recent works start to explore RGB cameras, as the existing large amount of RGB cameras like phone cameras, as well as existing RGB footage are still far more ubiquitous than depth cameras and depth data. This dissertation emphasizes the high need for accurate RGB-based 3D hand pose estimation methods and focuses on 3D hand pose estimation from monocular RGB images.

Estimating 3D poses from a monocular hand image is an ill-posed problem due to scale and depth ambiguities. As shown in Fig. 1.3, we can see that different 3D keypoints may project into the same 2D locations in the image coordinate. To get the accurate 3D hand joint locations, we need either a given hand skeleton model with known bone lengths or provided metric depth in the z axis (*i.e.* , optical axis). Without sufficient information, we can only estimate relative normalized 3D joint locations. In this case, to recover the original 3D poses, we follow the most common problem setting to disambiguate by providing one hand joint as root and one bone length as reference besides the RGB input. As shown in Fig. 1.1, the work [178] provides wrist as root and the first bone length of the index finger as the reference bone length. Note the definition of hand root and reference bone length is not fixed and different works may use different definition sets.



(a) robot hand retargeting (b) activities analysis (c) gesture recognition

(d) human computer interaction

Figure 1.2: Application of hand pose estimation. Images are from [1, 45, 101, 141]

1.2 Motivation

Unlike depth maps as 2.5D images, monocular RGB images only with 2D information suffer from depth ambiguities for 3D pose estimation. Moreover, RGB images often exhibit a large discrepancy between factors of variation ranging from image background content to lighting. Deep models trained in limited scenes are prone to over-fitting to specific artifacts and the performance of models can deteriorate significantly when applied to different scenes. To tackle the ambiguities and the diverse appearance associated with monocular RGB inputs, most works require sufficient and diverse training data. Unfortunately, gains from purely increasing dataset size tend to saturate, because it is difficult to get accurate and diverse real-world data. Labelling real-world 3D hand pose labels can be laborious and time-consuming, and even the quality of labels is hard to be guaranteed. We show current annotation methods for RGB hand poses, *i.e.* , 6DoF sensors-based methods [165, 33] and semi-automatic annotation methods [179] in Fig. 1.4. They all have non-negligible drawbacks. For 6DoF sensors-based methods, the RGB data suffer from RGB image degradation as the sensors are visible. For semi-automatic annotation methods, human-annotated labels are noisy and biased because of the human annotation error and the multi-view camera rigs are in-applicability in unconstrained environments. The drawbacks make it non-trivial to gather high-quality "in-the-wild" data.

Instead of seeking more accurate labelled data, can we use other auxiliary information to aid the representation learning and relieve the burden of annotation for 3D hand pose estimation? In the following, we investigate three sources of auxiliary information: (1) image factors such as background and viewpoint (2) multiple modalities such as RGB images and depth maps and (3) synthetically rendered images, and provide insights to utilize auxiliary information for deep models.

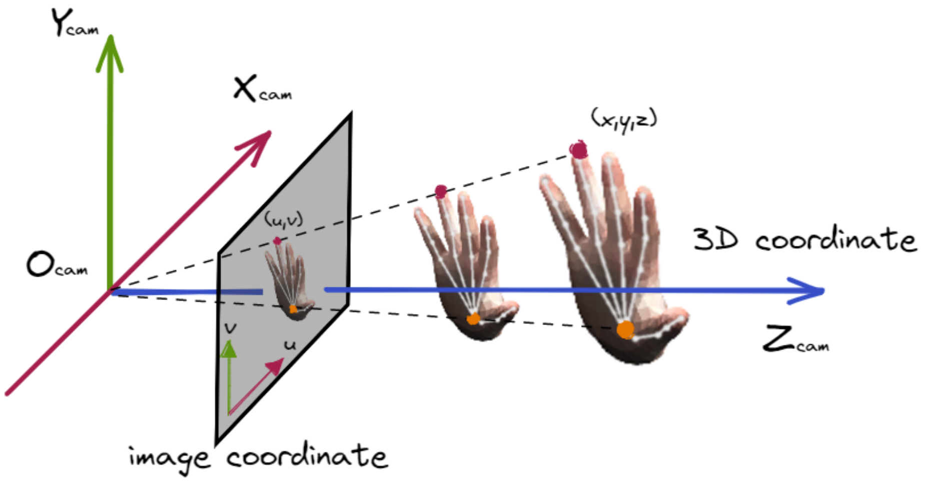
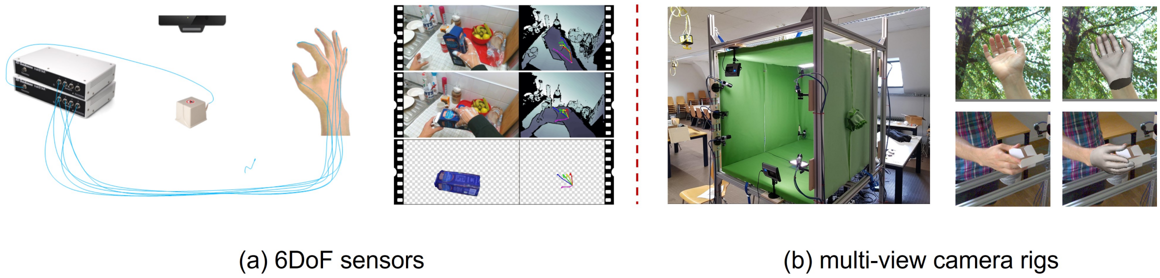


Figure 1.3: Illustration of depth ambiguities associated with monocular RGB images. We can see that different 3D poses may project into the same 2D locations in the image coordinate.


(a) 6DoF sensors

(b) multi-view camera rigs

Figure 1.4: Illustration of two annotation equipment and their labelled examples. Images from [165, 33, 179]

1.3 Contribution of the Dissertation

1.3.1 Learning with Image Factors

From the aspects of image rendering, an image can be understood in terms of a number of image factors of variation like background, viewpoint, texture, lighting as shown in Fig. 1.5. We explore the image factors of variation as auxiliary information to get better representations or server as weak-labels for hand poses. Below, we summarize two challenges.

Black-box Latent Representations. RGB hand images have a large discrepancy between factors of variation ranging from image background content to camera viewpoint. Those factors make both hand image synthesis and pose estimation from RGB images highly challenging. Existing works tend to learn only black-box latent representations and offer little control for conditioning upon image factors. Thus, there is a demand for disentangled repre-

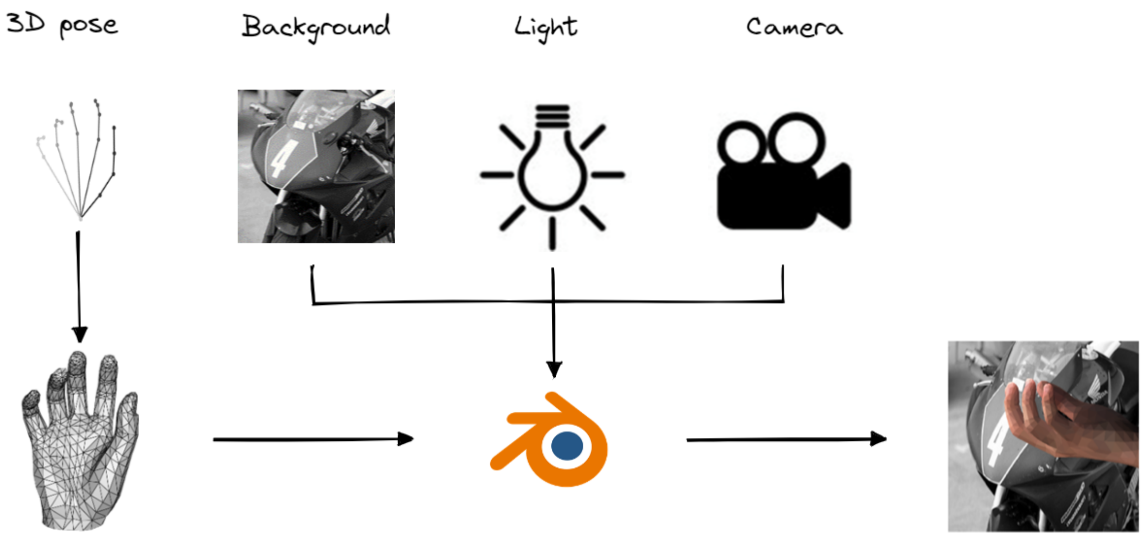


Figure 1.5: Pipeline of hand image rendering. Based on hand image factors of variation like 3D hand models, backgrounds, lights and camera viewpoints, we can generate hand images by means of a renderer like Blender.

sentations to better analyze these factors of variation.

Data with Labelled Factors. It is easy to get data with labelled or shared factors of variation. Taking video sequences for example, we can obtain data with the same background easily when the camera position is fixed [167]. Moreover, we can get hand images with the same canonical hand poses by keeping our hand pose fixed but moving our wrist [43]. However, it is difficult to use the labelled factors as they are implicit and hence no existing works try to explore frameworks that work with this kind of auxiliary information.

To address those challenges, in Chapter 4, based on variational autoencoder (VAE), we propose a novel disentangled variational autoencoder (dVAE) model; this model is the first VAE-based model that uses independent factors of variations to learn disentangled representations. With dVAE, we decouple the learning of disentangling factors and the embedding of image content. With disentangled representations, we enable explicit control over different factors of variation and introduce the first model with multiple degrees of freedom for synthesizing hand images. Based on the proposed framework and factor-labelled data, we also explore semi- and weakly-supervised settings for hand pose estimation.

1.3.2 Learning with Multi-Modalities

Real-world hand data usually comes with multiple modalities. For example, commercial RGBD cameras can provide RGB images and depth maps; With gloves [7] or color paints [108], we can further provide hand segmentation masks. Moreover, based on the connection among the modalities, most modalities can be inter-conversion. Take a hand voxel for example, we can convert it to a hand mesh with marching cube algorithms [74]. As hand modalities are representations of hands in different aspects, different modalities could be auxiliary information for the learning of each other. Therefore, we aim to explore the usage of hand modalities as auxiliary information. Here we list some common hand modalities below:

- Hand RGB image
- Hand depth map
- Hand point cloud
- Hand segmentation mask
- 2D hand heatmap
- Hand mesh
- 3D joint location
- 2D joint location
- Hand voxel

With aforementioned multi-modalities, we have the following challenges, the usage of multi-modal data and the design of a flexible multi-modal framework.

The Usage of Multi-Modal Data. Auxiliary modalities share common visual cues, such as the underlying geometry, or semantics, and hence be beneficial for the training of target modalities. However, it is still unclear how to exploit multi-modal data. Prior works either adopt a multi-task framework to encourage the models to reconstruct different modalities or use modalities as weak labels by using a cross-modal reconstruction network [11]. However, they limit their focus on reconstruction, regardless of the role of multi-modal data for representation learning in the feature space.

The Design of Multi-Modal Framework. Although there are various modalities of hand, only depth maps have been explored to enhance RGB-based hand pose estimation [163]. Different modalities are with different representations and hence require ad-hoc network structures. For example, we usually adopt PointNet [38] for hand point clouds while convolutional neural network (CNN) for RGB images. It is non-trivial to introduce a unified framework to handle arbitrary modalities with ad-hoc networks . Moreover, it is favorable for a framework to work with arbitrary data pairs. Assuming that we have some RGBD data as the supplement of RGB data, to exploit the RGBD data, it would be better if our framework can take both RGB and RGBD as input during training.

To handle the challenges, we introduce multi-modal data as auxiliary information for RGB images in Chapter 5 and Chapter 7 with the technique of multi-modal alignment.

In Chapter 5, to get a flexible framework for arbitrary modalities, we formulate RGB-based hand pose estimation as a multi-modal learning, cross-modal inference problem. Based on this, we propose a VAE-based framework learning from different hand inputs of various modalities. All the operations are processed in a shared latent space and we only need to encode different modalities into latent variables. This makes the framework flexible and easy to incorporate arbitrary modalities. Also, regarding hand modalities, we explore non-conventional inputs such as point clouds and heatmaps for learning the latent hand space and try to align modalities in the shared latent space via the product of Gaussian experts.

In Chapter 7, beyond simple latent space alignment, we explore more different alignments for multi-modal representation learning. Especially, we design a dual-modality network to take two modalities (*i.e.* , RGB images and depth maps) as input and align their features in pixel-level via an attention-based multi-modal training. The fusion enables the model for RGB images to better capture features relating to the common visual cues in the depth maps. Moreover, we subsequently design multi-modal contrastive learning that allows us to

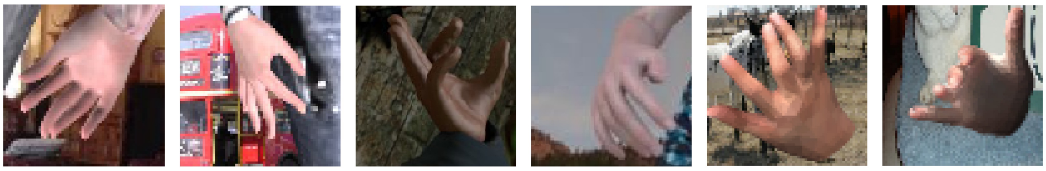


Figure 1.6: Examples of synthetic data. Images from [153].

construct a well-structured low-dimensional subspace that aligns similar poses across different modalities. The different levels of alignment make our framework easy to achieve better representations and thus get better performance.

Overall, we exploit multi-modal data as auxiliary information for RGB images in multiple levels, *i.e.* , latent space alignment (see Chapter 5), pixel-level alignment (see Chapter 7) and low-dimensional subspace alignment (see Chapter 7). Meanwhile, we propose a flexible framework for arbitrary modalities via a shared latent space (see Chapter 5).

1.3.3 Learning with Synthetic Data

As data driven methods, deep models require sufficient and diverse labelled data. Unfortunately, labelling accurate real-world labels is non-trivial. As an alternative, synthetic data can be used as auxiliary information for real-world data. The benefits of synthetic data are threefold. First, synthesizing samples is an easy way to get accurate labels, thanks to the improvements in image renderers. Second, synthetic data can be generated based on specific needs or conditions. We can adjust the statistical properties, have fine-grained control over the factors of variation, and cover rare cases in the real world. Lastly, synthetic data is the key in avoiding the risk of privacy. With the growing concerns that real-world data may lead to privacy risk, synthetic data preserve the important properties of real-world data and relieve the dependence on real-world data, making it favorable to reduce the privacy risk.

To synthesize hand data, we need 3D hand models and image renderers. For 3D hand models, we can use "hand Model with Articulated and Non-rigid defOrmations" (MANO) [100], which is a parametric model based on around 1000 high-resolution 3D scans of real-world hands. Also, we can get hand models from human character animation services like Mixamo, Maya and Blender. Once we have a 3D hand model, we may use the renderer software (*e.g.* Maya, Blender) or open-source renderer library (*e.g.* Neural Renderer [54], OpenDR) to render images. We show three different synthetic hand data in Fig. 1.6. We can see that different synthesis strategies have their own characteristics like hand appearances and lighting schemes for rendering. We summarize the challenges of learning with synthetic data as below.

Domain Gap. Based on the data generation process, synthetic data may have particular blending artifacts and are still far from "realistic". Therefore, there still exists a significant domain gap between synthetic and real-world data. Deep models trained on synthetic data easily may over-fit to blending artifacts and the performance of models can deteriorate significantly when applied to real-world data.

Semi-Supervision. From the view of training settings, training with labelled synthetic data and unlabelled real-world data is in a semi-supervised setting. Once we have the pre-trained model from synthetic data and the unlabelled data from real world, a common practice for the semi-supervised setting is to fine-tune the model on unlabelled data with pseudo-labels. However, naively generated pseudo-labels are inevitably noisy and deteriorate model performance. To overcome this, existing classification works propose to correct pseudo-labels using operations like argmax [65], sharpening [6] or thresholding [107]. However, extending such concepts for a regression task and in the context of 3D pose estimation is non-trivial.

To overcome those challenges, in Chapter 6 and Chapter 7, by leveraging the readily available synthetic data as auxiliary information, our proposed framework learns with labelled synthetic data and unlabelled real-world data.

In Chapter 6, we propose the first semi-supervised framework that combines pseudo-labeling with consistency training for RGB-based hand pose. For pseudo-labeling, taking the feasibility of hand poses into account, we investigate the pose registration, which corrects pose based on the limits of bone lengths and hence reduces the noise of pseudo-label. Also, we estimate the confidence of pseudo-labels according to the consistency and the feasibility, and then train with only high-confidence pseudo-labels. For consistency training, based on the spatial information, we introduce two consistency losses for 3D pose estimation to encourage the predictions to be consistent with perturbations and auxiliary modalities. Last, we also introduce training with data augmentation of differing difficulties, which could improve the stability of our framework.

In Chapter 7, we shift our focus to reducing the negative influence of pseudo-label noise. Based on our new design dual-modality network and synthetic multi-modal data, we construct a self-distillation structure for pseudo-labelling to gradually improve pseudo-labels instead of replacing them dramatically. Moreover, we further improve the pose registration in Chapter 6. Specifically, beyond our prior work, which focuses only on bone lengths, we also take joint angles into account and hence guarantee biomechanical feasibility of the corrected hand pose concerning both the bone lengths and the joint angles.

In summary, we develop different modules (pose registration, self-distillation) and different strategies (consistency training, pseudo-labeling and self-paced training) for cross-domain semi-supervised hand pose estimation to reduce the gap between synthetic data and real-world data.

1.4 Organization

In this dissertation, we explore the usage of three auxiliary information for RGB-based 3D hand pose estimation. The dissertation is organized as below:

- In Chapter 2, we introduce the preliminaries of RGB-based hand pose estimation.
- In Chapter 3, we present a comprehensive summary of existing 3D hand pose estimation methods related to my topics.

- In Chapter 4, we present our **Disentangling Latent Hands for Image Synthesis and Pose Estimation** to use image factors of variations as auxiliary information [156].

- In Chapter 5, we introduce **Aligning Latent Spaces for 3D Hand Pose Estimation** to use multi-modal data as auxiliary information [155]. Specially, we propose a flexible framework for arbitrary modalities alignment via a shared latent space.

- In Chapter 6, we target the challenging scenario of learning models from labelled synthetic data and unlabelled real-world data and propose **SemiHand: Semi-supervised Hand Pose Estimation with Consistency** to use synthetic data as auxiliary information [153]. We propose the first cross-domain semi-supervised framework for 3D hand pose estimation, including consistency training, pseudo-labelling and pose correction.

- In Chapter 7, we target the same scenario but shift our focus to multi-modal representation learning and pseudo-labelling. We propose **Dual-Modality Network for Semi-Supervised Hand Pose Estimation** to use both multi-modal data and synthetic data as auxiliary information [70]. Specifically, we propose two feature alignments (*i.e.* , multi-modal pixel-level alignment and multi-modal low-dimensional subspace alignment), and reduce the negative influence of noisy pseudo-labels via self-distillation and pose correction.

- In Chapter 8, we summarize the addressed and remaining challenges, and conclude the dissertation.

Chapter 2

Preliminaries

Contents

2.1 Datasets 11

2.2 Architectures 13

2.3 Coordinate Representation 14

2.3.1 Heatmap-based Representation 15

2.3.2 Numerical Coordinate-based Representation. 16

2.4 Hand Surface Model 17

2.4.1 3D Approximation Model 17

2.4.2 3D Mesh Model 18

2.5 Evaluation Metric 19

In this chapter, we provide a detailed overview of datasets in Sec. 2.1, architectures in Sec. 2.2, coordinate representations in Sec. 2.3, hand surface models in Sec. 2.4 and evaluation metrics in Sec. 2.5 for RGB-based hand pose estimation.

2.1 Datasets

In this section, we compare the statistics of existing RGB hand pose benchmarks including RHD [178], ObMan [46], DO [113], FreiHAND [179], H3D [173], STB [167] and YT3D [63] in Tab. 2.1 and introduce them briefly as below.

Rendered Handpose Dataset (RHD) is a synthesized dataset rendered by Blender and 3D models from Mixamo. It is composed of 41k training and 2.7k testing images of 320×320 resolution from 20 animated characters. Each rendered RGB image comes with corresponding multi-modal like depth map, segmentation mask, 2D joint location and 3D joint location. As a synthetic dataset, the hand images of RHD have very different appearances compared to those from the real world, which makes the model trained from RHD generalize poorly on real-world data.

ObMan is a large-scale synthetic image dataset of hands grasping objects. It includes 8 object categories of everyday objects from ShapeNet and uses MANO as a hand model. In order to generate plausible hand grasps for those objects, the dataset uses the automatic robotic grasping software GraspIt. With 3D hand and object model, 150K RGB images are generated by Blender, along with 2D/3D hand keypoints, object and hand segmentation

Dataset	Modality	Resolution	Subjects	Views	Frames	Syn/Real	Joints	MANO	Annotation
RHD	RGBD	320×320	20	1	44K	Syn	21 3D	No	syn
ObMan	RGBD	256×256	-	1	150K	Syn	21 3D	Yes	syn
DO	RGBD	640×480	2	1	3K	Real	5 3D	No	manual
FreiHAND	RGB	240×240	32	1	36K	Real	21 3D	Yes	semi-auto
H3D	RGB	3840×2160	10	15	22K	Real	21 3D	No	semi-auto
STB	RGBD	640×480	1	2	36K	Real	21 3D	No	manual
YT3D	RGB	mixed	-	1	51K	Real	21 2D	Yes	semi-auto

Table 2.1: Comparison of existing RGB-based hand pose benchmarks. We first present two synthetic datasets: RHD and ObMan. We then list real-world datasets: STB, DO, YT3D, H3D and FreiHAND. Here, "Modality" corresponds the available input source of dataset, "Resolution" corresponds the resolution of input source, "Subjects" corresponds the number of hands, "Views" corresponds the number of viewpoints, "Frames" corresponds the number of images, "Syn/Real" indicates the dataset is either synthetic data or real-world data, "Joints" shows the number of keypoint and either in 3D space or in 2D space, "MANO" shows if MANO parameters are provided, "Annotation" shows the type of annotation method.

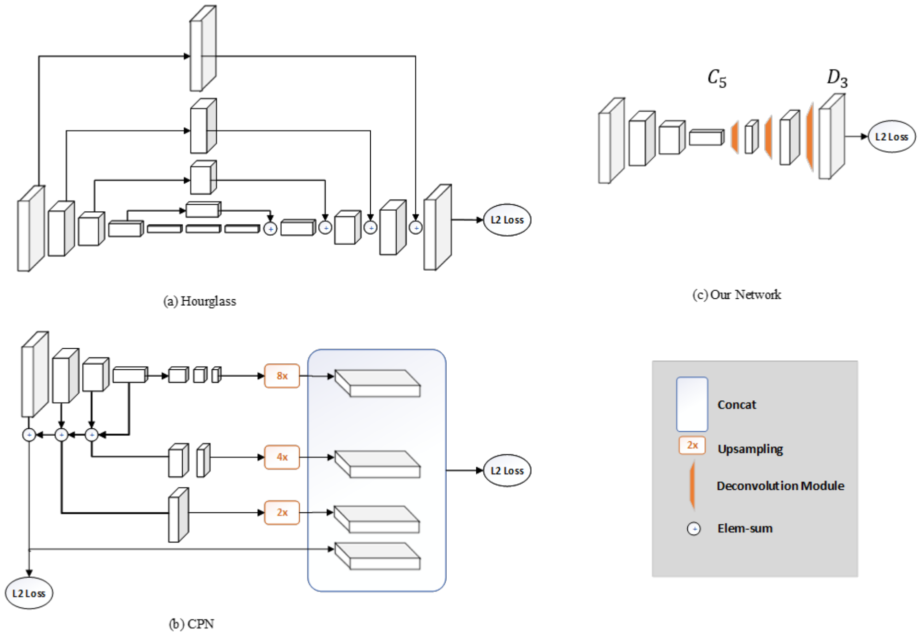
masks, and depth maps. Since this is also a synthetic dataset, it suffers from the same issue as RHD.

Dexter+Object (DO) is an evaluation dataset for fingertip estimation. It consists of 6 sequences with 2 actors and varying interactions with a simple object. The dataset provides 3K frames of color images, depth maps, and manually annotated fingertips positions. Due to the incomplete hand annotation and relatively small size, this dataset is only used to study the cross-dataset performance.

FreiHAND is a large-scale, multi-view hand dataset. To annotate in the real world, it adopts a recording setup and a MANO fitting method. The recording setup is with 8 calibrated and temporally synchronized RGB cameras located at the corners of a cube rig. With multi-view RGBD data, an iterative, semi-automated approach is used for labelling. It contains 130,240 training samples and 3,960 test samples. Each training sample contains a single view RGB image, annotations of MANO-based 3D hand joints and mesh, as well as camera pose parameters. The training data is recorded with a green screen and then augmented by leveraging the green screen for background subtraction and creating composite images using new backgrounds. In contrast, the testing set is directly recorded in real-world scenario.

Hand 3D Studio dataset (H3D) is a multi-view dataset captured by a customized acquisition system with 15 high-quality DSLR cameras. The dataset collects 50 one-handed gestures and 27 hand-object interaction gestures in daily life from 10 persons. It consists of high-resolution multi-view RGB images with corresponding point cloud data, 2D and 3D hand joints location and the fitted 3D hand mesh models. It is an indoor dataset with fixed backgrounds. This limits it to apply in a real-world scenario.

Stereo Tracking Benchmark Dataset (STB) contains 12 sequences of a single person's left hand in front of 6 backgrounds with two different poses. The first poses is counting poses with slowly moving fingers. The second poses is random poses with self-occlusions and



(a) Hourglass

(c) Our Network

(b) CPN

Figure 2.1: Illustration of three common network architectures for pose estimation, (a) Hourglass [84], (b) Cascaded Pyramid Network [22], and (c) PoseResNet [149]. Images are from [149].

global rotations. It provides RGB images, corresponding depth maps as well as annotated 3D poses. The hand poses in this dataset are relatively simple and therefore the performance of state-of-the-art methods on it has been saturated.

YouTube 3D Hands (YT3D) is curated from YouTube videos, most of which are sign language conversations performed by people of a wide variety of nationalities and ethnicity. The training set has 47k images from 102 videos while the test set has 1.5k images from 7 videos. The dataset is semi-automatically annotated with OpenPose and MANO. The annotation performance is bounded by the performance of OpenPose. Therefore, it can be used only for self-comparison.

2.2 Architectures

Early works for RGB-based hand pose estimation adopt VGG-like plain deep CNN [105], which stacked convolutional layers with activation functions and pooling operations. However, those CNNs suffer from the vanishing gradient problem when going deeper. Skip connection introduced by Residual Network (ResNet) [47] makes it possible for neural networks to go deeper and perform better. With skip connection, two kinds of network architectures have been the dominant architectures for pose estimation. The first one is the cascaded fully convolutional network (FCN) with a coarse-to-fine design paradigm, like Hourglass [84] and Cascaded Pyramid Network (CPN) [22]. The second one is ResNet and its variants like

PoseResNet [149]. Based on the target representations, additional layers like convolutional layers, deconvolution layers or Multilayer Perceptron (MLP) are added to produce a final set of predictions. We take Hourglass, CPN and PoseResNet for examples in the following. Those networks are shown in Fig. 2.1.

Stacked Hourglass Network is a symmetric network with downsampling layers, upsampling layers, and skipping connections as shown in Fig 2.1 (a). The overall architecture is like an encoder-decoder architecture to get a very low-resolution representations via convolutional and max pooling layers, and then gradually recover the high-resolution representations using convolutional and simple nearest neighbor upsampling. The main difference from prior designs is its symmetric topology and the skipping connection to concatenate feature maps of downsampling layers to their symmetric feature maps of upsampling layers for subsequent upsampling. This enables the network to capture and consolidate information across all resolutions.

Cascaded Pyramid Network involves two modules, which serve for feature extraction and feature refinement respectively, motivated by the effective stacking operation in stacked hourglass networks. The feature extraction module using a feature pyramid structure enables sufficient context information from different spatial resolutions while the feature refinement module transmits and integrates the information of different resolutions by means of upsampling and concatenating.

PoseResNet provides a quite simple yet surprisingly effective architecture for pose estimation, aiming to encourage researchers to shift their focus from the design of network structures to the algorithms. From the aspects of an autoencoder, it uses a ResNet backbone as the encoder and the deconvolutional layers as the decoder. ResNet is the most common encoder for feature extraction while the deconvolutional layers combining the upsampling and convolutional parameters is the most simple decoder to generate heatmaps. Note that there are no skip layer connections between the encoder and the decoder.

Based on the above basic architectures, more variants are proposed. For example, the work [154] introduces a multi-task bisected hourglass, which modifies the hourglass network by adding one more decoder and allows the networks to encapsulate homogeneous information. Spurr *et al.* [111] proposes to use ResNet with MLP layers as backbone to directly predict 3D hand poses.

2.3 Coordinate Representation

For a cropped hand image $\mathbf{x} \in \mathbb{R}^{m \times n}$, the hand pose $\mathbf{J} \in \mathbb{R}^{n_J \times d_J}$ with n_J keypoints in d_J-dimensional space can be recovered by first encoding the image with $\mathbf{h} = \mathrm{En}(\mathbf{x})$ and then decoding the representation $\mathbf{h}$ into joint coordinates $\mathbf{J} = \mathrm{De}(\mathbf{h})$. Existing pose estimation architectures use an image encoding network, but differ in the way of decoding the representation $\mathbf{h}$ to coordinate representations. In Fig. 2.2, we show one example of 2D numerical coordinate-based representations and 2D heatmap-based representations. We can see heatmap-based representations are pixel-wise representations and have pixel-wise cor-

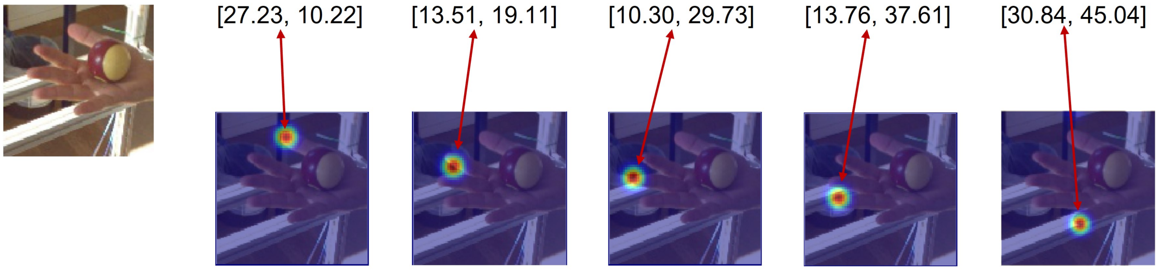


Figure 2.2: Comparison of 2D numerical coordinate-based representations (top) and 2D heatmap-based representations (bottom) of fingertips from one hand. Red double headed arrows indicate their correspondences.

respondence to original inputs while numerical coordinate-based representations are simple numerical values. Existing 3D hand pose works either directly use lifting networks to learn a mapping from 2D pose to 3D pose [11] or adjust existing 2D representations to 3D representations by ad-hoc design for the z dimension [149, 50]. In the following, we focus more on how to work with two different coordinate representations for 3D hand pose estimation.

2.3.1 Heatmap-based Representation

Heatmap-based methods, which locate the joints by estimating a likelihood heatmap, are dominant for pose estimation. The rationale is that working with heatmaps allows the architecture to remain fully convolutional, thereby retaining spatial structures throughout the encoding and decoding process. As labels, heatmap methods use a Gaussian centered at the ground truth joint coordinate. This formulation converts pose estimation into a detection problem; the network is tasked with predicting, at each pixel, the probability of that pixel being a joint pixel. For 2D pose estimation, the output heatmap $\mathbf{h}$ with element $h_{\mathbf{p}}$ at location $\mathbf{p}$ is supervised directly by the ground truth heatmaps $\mathbf{h}^{\mathrm{gt}}$ with corresponding element $h_{\mathbf{p}}^{\mathrm{gt}}$ generated by the ground truth $\mathbf{J}^{\mathrm{gt}}$, with the Gaussian variance σ_h:

$$L = \sum_{\mathbf{p}} (h_{\mathbf{p}} - h_{\mathbf{p}}^{\mathrm{gt}})^2, \quad h_{\mathbf{p}}^{\mathrm{gt}} = \exp(-\frac{1}{2\sigma_h^2}(\mathbf{p} - \mathbf{J}^{\mathrm{gt}})^T(\mathbf{p} - \mathbf{J}^{\mathrm{gt}})). \tag{2.1}$$

One intuitive way to get 3D poses is to extend the 2D heatmaps to 3D heatmaps [49]. However, getting poses from heatmaps is not differentiable. As such, the post-processing (*i.e.* , argmax) is required. As an alternative, other heatmap-based approaches [178, 11, 175] prefer an additional network named the pose lifting network to lift 2D heatmaps to 3D poses by a neural network. In other words, they keep end-to-end training by using 2D heatmaps supervision as intermediate supervision and lifting the 2D heatmaps to 3D poses with lifting networks. Note that the lifting networks also have been explored to work with 3D surface model. Recent works start to lift 2D heatmaps to 3D voxel space [49] or parametric 3D hand models [9, 4, 169].

2.3.2 Numerical Coordinate-based Representation.

It is intuitive to solve pose estimation from RGB images by a straightforward numerical coordinate regression, *i.e.* , directly predicting the joint coordinates in 2D/3D space. For 2D pose estimation, the joint outputs $\mathbf{J}$ is supervised directly by an L2 loss with respect to the ground truth coordinates $\mathbf{J}_{\text{gt}}$:

$$L = \|\mathbf{J} - \mathbf{J}^{\text{gt}}\|_2^2. \tag{2.2}$$

However, compared with heatmap-based methods, direct numerical coordinate regression is hard to capture spatial distributions around ground truth coordinates and therefore worsens the performance [160]. Existing numerical coordinate regression methods explore different directions to capture more spatial attention or improve the generalization. Based on VAE and cross-training, the work [111] uses a shared latent space for cross modalities estimation to regularize pose estimation from RGB images. Li *et al.* [68] introduces the network to exploit bio-structure of hand, avoiding the negative transfer among less related joints. Among all the numerical coordinate-based works, two directions have drawn much attention, integral pose regression and 3D model regression.

Integral Pose Regression. Integral pose regression [117, 50] benefits from combining heatmap-based and numerical coordinate-based representations by introducing softargmax and the latent heatmaps to approximate the operation of argmax as well as the explicit heatmaps. It can train end-to-end directly and capture spatial distributions around ground truth coordinates while retaining the benefits of fully convolutional architectures. For 2D pose estimation, the decoder is same as heatmap-based methods but the supervision comes directly from the ground truth $\mathbf{J}^{\text{gt}}$. Like [117, 50], the joint outputs $\mathbf{J}$ is calculated based on the latent heatmaps $\mathbf{h}$ with element $h_{\mathbf{p}}$ at location $\mathbf{p}$ and supervised directly by an L2 loss with respect to the ground truth coordinates $\mathbf{J}^{\text{gt}}$ as below

$$L = \|\mathbf{J} - \mathbf{J}^{\text{gt}}\|_2^2, \;\; \mathbf{J} = \sum_{\mathbf{p}} \tilde{h}_{\mathbf{p}} \cdot \mathbf{p}, \;\; \tilde{h}_{\mathbf{p}} = \frac{\exp(h_{\mathbf{p}})}{\sum_{\mathbf{p}' \in \Omega} \exp(h_{\mathbf{p}'})}. \tag{2.3}$$

To address 3D pose estimation, Sun *et al.* [117] constructs 3D space with grid cells like voxel and directly extends 2D heatmaps to 3D space. In contrast, the work [50] proposes to estimate the 2.5D pose which equals to 3D pose when given the camera intrinsic. A 2.5D pose consists of 2D coordinates of the hand keypoints in the image space (*i.e.* , 2D pose), and scale normalized metric depth for each keypoint relative to the root. The scale normalized metric depth value is obtained as the summation of the Hadamard product of 2D latent heatmaps and latent metric depth value maps.

3D Model Regression. A recent trend for 3D hand pose estimation is to formulate it as a sub-problem of image-3D surface model alignment. Numerical coordinate regression is easy to work with 3D hand models when the 3D hand models are rigged. Benefiting from the sophisticated modeling of 3D hand, numerical coordinate regression with a 3D hand model has been developing rapidly. Based on the parametric hand model MANO, works [9, 46] take hand images as input and predict the hand meshes as well as 3D poses. Besides parametric models, other works propose to train non-parametric 3D hand mesh networks [63, 37] from

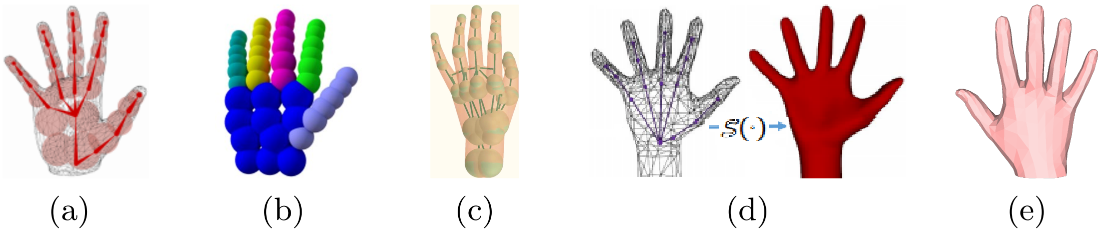


Figure 2.3: 3D hand surface models. (a) Sum-of-Gaussians model, (b) Primitives approximation, (c) Sphere-Meshes can be thought of as a generalization of the previous models, (d) Loop Subdivision Surface of a triangular control mesh , (e) Hand MANO model. Images reproduced from [100, 132].

large hand mesh datasets and then get the underlying joints from hand meshes.

Compared to other numerical coordinate-based methods, the main difference is that the decoder De(·) of 3D model regression methods is either an articulated mesh deformation model represented with a differentiable function for parametric models or an ad-hoc surface decoder for non-parametric models. For 2D pose estimation, the joint outputs $\mathbf{J}$ is supervised directly by an L2 loss with respect to the ground truth coordinates $\mathbf{J}_{\mathrm{gt}}$:

$$L = \|\mathbf{J} - \mathbf{J}^{\mathrm{gt}}\|_2^2 + R(\mathbf{h}), \;\; \mathbf{J} = \Pi(M(\mathrm{De}(\mathbf{h}))), \tag{2.4}$$

where $M(\cdot)$ is a predefined function to get 3D poses from hand surfaces and $\Pi(\cdot)$ is the camera projection function to get predicted 2D poses in the image space. $R(\cdot)$ is a regularization term for the representation $\mathbf{h}$.

2.4 Hand Surface Model

3D hand pose estimation can be formulated as a sub-problem of image-3D hand surface alignment, thanks to the development of skeletal animation. With the help of model fitting, 3D models explicitly build a dense pixel-level connection with RGB images. If models are rigged ones, the underlying skeletons then can be calculated easily. Hand surface models have shown promising potential in numerical coordinate-based works for hand pose estimation. Based on the type of 3D models, we introduce hand approximation models and hand mesh models, which are the mainstream of 3D hand models.

2.4.1 3D Approximation Model

Approximating hand surfaces using a set of primitives enables balancing the accuracy and the efficiency of model fitting. Existing methods tend to use spheres to approximate 3D hand as shown in Fig 2.3 (a) - (c). Works [114, 82] model hand using a implicit sphere model, *i.e.* , the Sum of Gaussians (SoG) representation (See Fig 2.3 (a)). The mean and the variance of a

Gaussian can be treated as the center and the radius for a sphere. SoGs are mathematically smooth, approachable for constructing energy functions and hence enabling fast optimization. Differently, works [96, 132] model a hand explicitly using a number of spheres (See Fig 2.3 (b)). Therefore, the hand surface can be approximated via the surfaces of spheres. To further improve the sphere models to be more accurate to fit hand surfaces, the work [126] introduces the use of sphere-meshes as a novel geometric representation (See Fig 2.3 (c)).

2.4.2 3D Mesh Model

3D mesh models have proven to be effective, because they are easy to render, visualize and be derived. For hand, 3D hand mesh models also show favorable benefits as they can be deformed and animated naturally. Specifically, there are two kind of hand mesh models, parametric hand mesh models and non-parametric hand mesh models.

Parametric Hand Mesh Model. Most works prefer parametric hand models. The benefits of parametric hand models are twofold: first, the parametric model do not require much training data and can work well even with only weak labels. Second, the prediction, *i.e.* , the parameters, is interpretable and flexible to add constraints if needed. Early works like [124] exploit subdivision surfaces to define a smooth continuous hand model. The follow-ups [55, 100] then articulate the models with standard Linear Blend Skinning (LBS) (See Fig 2.3 (d) - (e)). Among them, MANO [100] is most commonly used and we introduce MANO in detail.

MANO is an articulated mesh deformation model, which learned from around 1000 3D scans of hands. The hand triangulated meshes from MANO consist $N = 778$ vertices on the hand surfaces and $K = 16$ joints. As a parametric model, MANO parameterizes the mesh using a differentiable function $M(\cdot)$ with shape parameters $\beta \in \mathbb{R}^{10}$ which represents coefficients of PCA components that sculpt the identity subject and pose parameters $\theta \in \mathbb{R}^{K\times 3}$, which means the relative 3D rotation of K joints. Specifically, $M(\cdot)$ is defined by a LBS function $W(\cdot)$ as below:

$$M(\beta, \theta) = W(T(\beta, \theta), J(\beta), \theta, \mathcal{W}). \tag{2.5}$$

Here, $J(\cdot)$ is a linear regressor to calculate the joint locations from mesh vertices given β, and $\mathcal{W}$ is the blend weights. $T(\cdot)$ aims to get hand template by deforming a mean mesh $\bar{T}$ with both shape and pose deformations, $B_S(\cdot)$ and $B_P(\cdot)$, respectively.

$$T(\beta, \theta) = \bar{T} + B_S(\beta) + B_P(\theta). \tag{2.6}$$

The mean mesh $\bar{T}$, the blend weights $\mathcal{W}$ and the functions $B_S(\cdot)$, $B_P(\cdot)$, $J(\cdot)$ are predefined and learned using the registration of 3D hand scans. Especially, pose-dependent corrective offsets $B_P(\cdot)$ are learned to address the loss of volume caused by the skinning method, which makes MANO more accurate than other parametric hand models. However, for parametric models, the parameters are still highly constrained to specified forms, which simplifies the learning process, but limits the learning ability. In practice, naive parameter estimation methods with a parametric hand model are unlikely to achieve satisfactory fitting performance.

Non-parametric Hand Mesh Model. The non-parametric hand models are more powerful to fit the underlying function and result in higher performance for prediction. Recent works [78, 37, 64, 63] introduce various frameworks to predict hand mesh from RGB input. However, compared to the parametric models, those non-parametric models require more labelled training data, more sophisticated network architectures and training strategies to improve the fitting capability of deep models. This makes the models have the ability to address more detailed hand surfaces but also more risk of over-fit to the training data.

2.5 Evaluation Metric

For quantitative evaluation and comparison with other works on hand pose estimation, there are two common metrics, mean end-point-error (EPE) and the area under the curve (AUC) on the percentage of correct keypoints (PCK) score. Mean EPE is defined as the average euclidean distance between predicted and ground truth keypoints; PCK is the percentage of predicted keypoints that fall within some given distance with respect to the ground truth. Assuming that we have N predicted poses $\hat{\mathbf{J}}$ and their corresponding ground truth poses $\mathbf{J}$, and each pose contains K joints. $\hat{J}_n^k$ and J_n^k are the k^{th} joint of n^{th} pose from $\hat{\mathbf{J}}$ and $\mathbf{J}$. The mean EPE is computed as

$$EPE(\hat{\mathbf{J}}, \mathbf{J}) = \frac{1}{N \times K} \sum_{n=1}^{N} \sum_{k=1}^{K} ||\hat{J}_n^k - J_n^k||_2, \tag{2.7}$$

and PCK is defined as

$$PCK(\hat{\mathbf{J}}, \mathbf{J}) = \frac{1}{N \times K} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{1}(||\hat{J}_n^k - J_n^k||_2 < \tau), \tag{2.8}$$

where $\mathbb{1}(\cdot)$ is the indicator function and τ is a given distance as a threshold.

Chapter 3

Related Works

Contents

3.1 Multiple Tasks 22
3.1.1 Modality-based Multi-Task Learning 22
3.1.2 Structure-based Multi-Task Learning 25
3.2 Domain Adaptation 26
3.2.1 Transfer from Different Modalities 26
3.2.2 Transfer from Synthetic Data 28
3.2.3 Others 28
3.3 Hand Models 29
3.3.1 Model-driven Paradigm 29
3.3.2 Data-driven Paradigm 30
3.3.3 Hybrid Paradigm 31
3.4 Representation Learning 32
3.4.1 Disentangled Representation 32
3.4.2 Contrastive Learning 34
3.5 Less Supervision 35
3.5.1 Weakly-Supervised learning 36
3.5.2 Semi-Supervised learning 37
3.5.3 Hand Pose Correction 39

Significant progress and remarkable performance have already been made for RGB-based hand pose estimation. As deep-learning-based methods have dominated as the best performers in recent years, we mostly focus on deep-learning-based methods for RGB-based 3D hand pose estimation. Note that we also introduce some works from related topics like depth map-based hand pose estimation and 2D pose estimation to make the introduction more comprehensively.

In this chapter, we introduce the training of RGB-based 3D hand pose estimation with multiple tasks in Sec. 3.1, with domain adaptation in Sec. 3.2, with hand models in Sec. 3.3, with representation learning technique in Sec. 3.3 and with less supervision in Sec. 3.5.

3.1 Multiple Tasks

3D hand pose estimation with multi-task learning strategies aims to improve the generalization of deep models by using the shared information contained in related tasks or sub-tasks. Existing hand pose frameworks explore multiple tasks in view of modalities or hand structures.

3.1.1 Modality-based Multi-Task Learning

Different hand modalities are representations of hands in different aspects. They are all highly related; hence, different modality estimation tasks can be used as auxiliary tasks to aid the training. Based on the type of shared network architecture, we introduce two strategies for multi-task learning, shared backbone and shared latent space separately.

Shared Backbone. The typical multi-task strategy is to learn tasks with a shared backbone and different heads. Most multi-task strategy for 3D hand pose estimation also fall in this line. 3D hand pose related tasks like 2D pose estimation, segmentation and depth estimation, have achieved significant performance improvements when using as pixel-level prediction tasks with FCNs like Hourglass and CPN. Therefore, FCNs with multiple heads have been the most common shared backbone.

Early works emphasize to embed 2D heatmap estimation into the framework due to its pixel-level representation and connection to 3D pose estimation. Based on the shared backbone, works [178, 175] adopt 2D heatmap estimation as an intermediate task and introduce a 2D-to-3D lifting network to get 3D poses. Moreover, Zhou *et al.* [175] introduces one additional head to predict bone direction heatmaps, which are constructed by tiling the coordinates of bone direction to the size of the heatmaps, as an auxiliary task.

2D heatmaps for pose estimation allow to accurately localize the keypoint via pixel-wise prediction, however, suffer from quantisation error and require post-processing to get 2D poses. Beyond simply using a lifting network to get 3D poses from 2D heatmaps, recent works tend to explicitly build the connection between 2D poses and 3D poses by introducing latent heatmaps [50] or offset maps [134].

Iqbal *et al.* [50] introduces latent heatmaps for 3D hand pose estimation and decomposes 3D pose estimation into two pixel-wise prediction sub-problems for K joints, the prediction of 2D poses $\{(x_i, y_i)\}_{i\in K}$ in the image coordinate and the prediction of metric depth values $\{\hat{Z}_i^r\}_{i\in K}$. They define $\{(x_i, y_i, \hat{Z}_i^r)\}_{i\in K}$ as a 2.5D representation. As shown in Fig. 3.1, given a cropped hand image with size $h \times w$, to estimate K hand keypoints, the network produces K latent 2D heatmaps H^{*2D} and K latent depth maps $H^{*\hat{Z}^r}$. For i^{th} joint, its 2.5D components (x_i, y_i) and $\hat{Z}_i^r$ are estimated based on the corresponding latent heatmap H_i^{*2D} and latent depth map $H_i^{*\hat{Z}^r}$ as below:

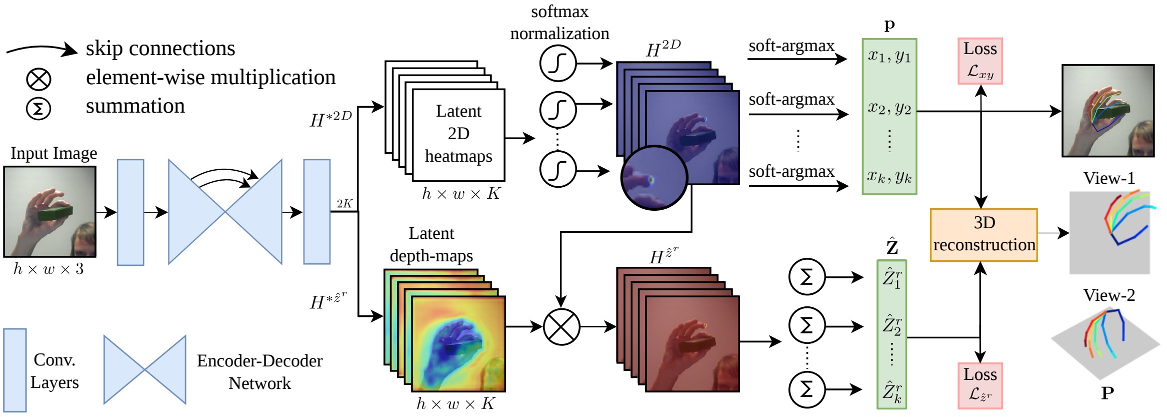


Figure 3.1: Illustration of latent 2.5D heatmap regression. Given a cropped RGB hand image, the network produces latent heatmaps including K latent 2D heatmaps H^{*2D} and K latent depth maps $H^{*\hat{z}^r}$ for K joints. To approximate 2D heatmaps with an argmax operation but be differentiable, the latent 2D heatmaps are decoded into the numerical coordinates by applying a softmax normalization and an expectation operation. To estimate the normalized depth values, we use the summation of $H^{\hat{z}^r}$, which obtained by multiplying $H^{*\hat{z}^r}$ with H^{2D}, *i.e.* , the softmax normalized H^{*2D}. The image is from [50].

$$\begin{aligned}(x_i, y_i) &= \sum_{g\in\Omega} g \otimes \text{softmax}(\beta H_i^{*2D})(g),\\ \hat{Z}_i^r &= \sum_{g\in\Omega} H_i^{*\hat{Z}^r}(g) \otimes \text{softmax}(\beta H_i^{*2D})(g),\end{aligned} \tag{3.1}$$

where Ω is the set of all pixel locations, $\otimes$ is the element-wise product, β is the learnable parameter and the function softmax$(\cdot)$ serves as normalization. With the predicted 2.5D result $\{(x_i, y_i, \hat{Z}_i^r)\}_{i\in K}$ and its corresponding ground truth $(\{(x_i^{gt}, y_i^{gt}, \hat{Z}_i^{gt})\}_{i\in K})$, the final loss with hyper-parameters λs can be written as follows:

$$\mathcal{L} = \frac{1}{K}\sum_{i=1}^{K}(||x_i - x_i^{gt}||_2 + \lambda_1||y_i - y_i^{gt}||_2 + \lambda_2||\hat{Z}_i^r - \hat{Z}_i^{gt}||_2). \tag{3.2}$$

Besides latent heatmaps, the offset map is another alternative to improve 2D heatmaps. The offset map is a vector field composed of vectors pointing to the joint location from individual pixels/voxels. The magnitude of each vector is the distance between the pixels/voxels and the joint. Note that offset maps are more commonly used for 3D pose estimation from depth maps as the pixel-wise hand surface locations from depth maps are important to support the learning of offset maps. Wan *et al.* [134] introduces to use offset maps and heatmaps by decomposing the joint locations into three per-pixel estimations, *i.e.* , 2D heatmaps, 3D heatmaps and unit 3D directional vector fields. The pixel-wise estimations can be directly translated into a vote casting scheme.

Same to [134], Wu *et al.* [145] combines 2D heatmaps supervision with two dense guidance

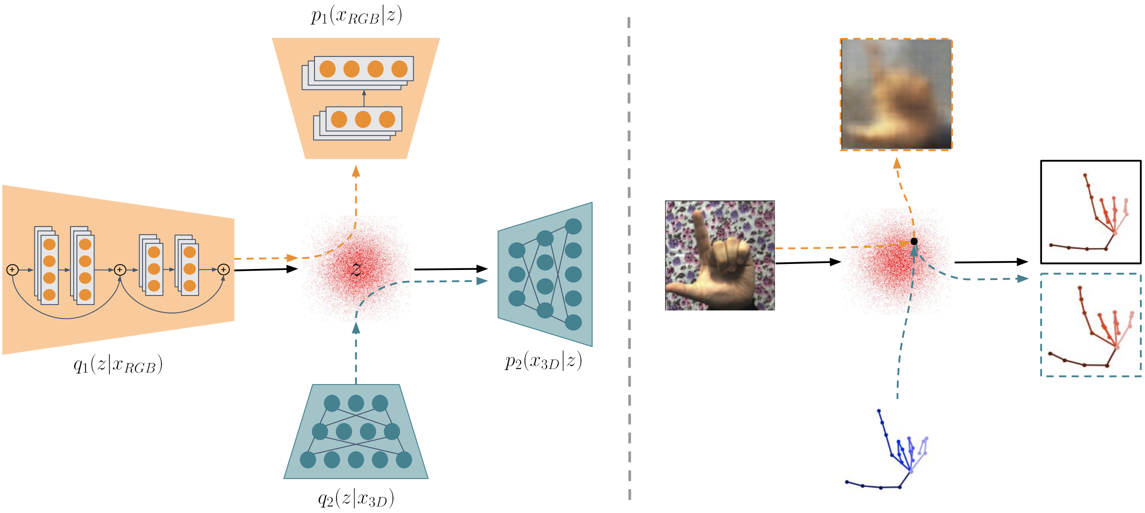


Figure 3.2: A cross-modal prediction pipeline with shared latent space. The RGB images $\mathbf{x}_{RGB}$ and 3D poses $\mathbf{x}_{3D}$ are encoded into the same latent space via a RGB encoder $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and a 3D pose encoder $q_2(\mathbf{z}|\mathbf{x}_{3D})$. With a latent variable $\mathbf{z}$, we can decode $\mathbf{z}$ into $\mathbf{x}_{RGB}$ or $\mathbf{x}_{3D}$ via a RGB decoder $p_1(\mathbf{x}_{RGB}|\mathbf{z})$ and a 3D pose decoder $p_2(\mathbf{x}_{3D}|\mathbf{z})$, respectively. We formulate an image reconstruction task with $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and $p_1(\mathbf{x}_{RGB}|\mathbf{z})$, and a hand pose estimation task with $q_1(\mathbf{z}|\mathbf{x}_{RGB})$ and $p_2(\mathbf{x}_{3D}|\mathbf{z})$. The image is from [111].

maps supervision, *i.e.* , the distance map supervision and the vector field supervision. Those dense maps enable composing vectors pointing to the joint location from individual pixels, which builds the connection between joints and each pixel. Notice that not all points or pixels contribute equally for a certain joint. Based on this observation, Fang *et al.* [30] predicts voting weights from Graph CNNs for predicted dense pixel-wise offset and Xiong *et al.* [152] proposes to get final joint locations based on the selected informative anchor points and their corresponding offsets. Those anchor points will capture the global-local spatial context information in an ensemble way and improve performance.

For 3D pose estimation, besides 2D pose estimation, other hand modality estimation tasks like hand segmentation and hand depth estimation are also easily extended into the framework by simply adding more heads of the shared backbone [168, 153]. Instead of predicting different modalities jointly, Yang *et al.* [154] introduces to predict different modalities based on the information of modalities. Using two-stacked network architecture, they predict 2D heatmaps and silhouettes in the first stage and 3D heatmaps and depth maps in the second stage, to make the training process more effective.

Shared Latent Space. Besides shared backbone, shared latent space is also promising for multi-task learning due to its flexibility. Compared to shared backbone, shared latent space enables us to train with different modality pairs. Spurr *et al.* [111] formulates hand pose estimation as a cross-modal prediction problem. Given two corresponding modalities $\mathbf{x}_i$ and $\mathbf{x}_t$, they aim to estimate $\mathbf{x}_t$ using $\mathbf{x}_i$ as input via maximizing the log probability of $p(\mathbf{x}_t)$.

$$
\begin{aligned}
\log p(\mathbf{x}_t) &= \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}_i) \log \frac{q_\phi(\mathbf{z}|\mathbf{x}_i)}{p_\theta(\mathbf{z}|\mathbf{x}_t)} d\mathbf{z} + \int_{\mathbf{z}} q_\theta(\mathbf{z}|\mathbf{x}_i) \log \frac{p(\mathbf{x}_t) p_\theta(\mathbf{z}|\mathbf{x}_t)}{q_\phi(\mathbf{z}|\mathbf{x}_i)} d\mathbf{z} \\
&= D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z}|\mathbf{x}_t)) + E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}_i)} \log p_\theta(\mathbf{x}_t|\mathbf{z}) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})).
\end{aligned}
\tag{3.3}
$$

Here, $D_{KL}(\cdot)$ is the Kullback-Leibler divergence. The variational approximation $q_\phi(\mathbf{z}|\mathbf{x}_i)$ can be thought of as an encoder from $\mathbf{x}_i$ to $\mathbf{z}$, while $p_\theta(\mathbf{x}_t|\mathbf{z})$ can be thought of as a decoder from $\mathbf{z}$ to $\mathbf{x}_t$. $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Gaussian prior on the latent space. As the term $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z}|\mathbf{x}_t))$ is intractable and greater than zero, we maximize the evidence lower bound instead via a latent variable $\mathbf{z}$ as below:

$$
\log p(x_t) \geq ELBO(\mathbf{x}_i, \mathbf{x}_t) = E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}_i)} \log p_\theta(\mathbf{x}_t|\mathbf{z}) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})), \tag{3.4}
$$

where ELBO$(\cdot)$ is the evidence lower bound function, we can get the final loss for cross modal prediction as follows:

$$
\mathcal{L} = \mathrm{ELBO}(\mathbf{x}_i, \mathbf{x}_t). \tag{3.5}
$$

The cross modal framework is flexible and can be used for both reconstruction and cross modal prediction. Therefore, for hand pose estimating, Spurr *et al.* [111] proposes to construct paired modalities to exploit complementary information from auxiliary paired data. Benefiting from cross training, different hand modalities are embedded into a shared latent space and therefore regularize each other. Specifically, as shown in Fig. 3.2, they have two modalities, RGB images $\mathbf{x}_{RGB}$ and 3D poses $\mathbf{x}_{3D}$, and construct four data pairs for cross training. The final loss is

$$
\mathcal{L} = \mathrm{ELBO}(\mathbf{x}_{RGB}, \mathbf{x}_{RGB}) + \mathrm{ELBO}(\mathbf{x}_{RGB}, \mathbf{x}_{3D}) + \mathrm{ELBO}(\mathbf{x}_{3D}, \mathbf{x}_{3D}) + \mathrm{ELBO}(\mathbf{x}_{3D}, \mathbf{x}_{RGB}). \tag{3.6}
$$

However, projecting multi-modal data into a shared latent space may be difficult because the modality-specific features usually interfere the learning of the optimal latent space. To circumvent this, the follow-up work [41] disentangles the latent features into modality-specific features and others, and then aligns those two accordingly.

3.1.2 Structure-based Multi-Task Learning

Rather than dealing with all hand joints as a whole, existing works also divide hand pose estimation into sub-tasks based on the differences in the functional importance of hand structure [106, 176, 29] as shown in Fig. 3.3.

As the articulation complexity of palm and fingers is different, works like [106, 29] divide hand estimation into palm estimation and finger estimation (See Fig. 3.3 left). Sinha *et al.* [106] uses CNN to hierarchically regress the hand joints from palm to fingers. With the observation that palm estimation is an easier task than that of the fingers, they propose a conditioned search to use predicted palm as a condition and then fine-tune the predicted

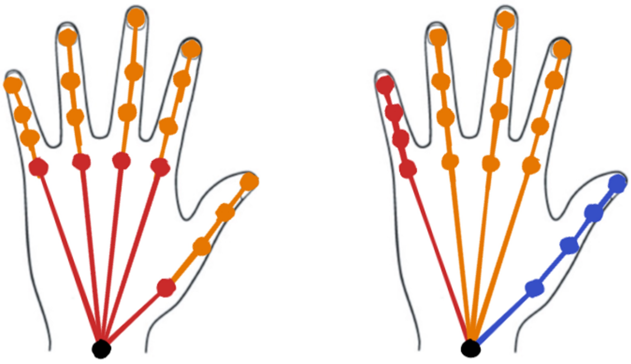


Figure 3.3: Illustration of two commonly used hand joints grouping. Different colours indicate different groups. Based on hand palm and fingers, works [106, 29] divide hand joints into palm joints and finger joints. Based on the function of different fingers, the work [29] divides hand joints into thumb finger joints, index finger joints and others.

fingers, which will lead to more discriminative features and achieve better accuracy. Differently, Du *et al.* [29] emphasizes the information interaction between two sub-tasks and designs a two-branch cross-connection structure to share the beneficial complementary information between palm and fingers to improve the performance.

To highlight the different functions of fingers, Zhou *et al.* [176] divides hand pose estimation into three finger estimations. Specifically, they partition hand into three parts: thumb, index finger and others (See Fig. 3.3 right). Based on the three parts, they design a three-branch CNN, where each branch corresponds to one hand part. After the three-branch CNN, the features are fused with a feature ensemble layer and then decoded to hand joint locations.

3.2 Domain Adaptation

While acquiring annotations for real-world RGB data is a difficult task, works circumvent this problem by utilizing the features from other domains like different modalities or synthetic RGB data.

3.2.1 Transfer from Different Modalities

Unlike RGB images are difficult to get accurate and diverse 3D annotations, we can gather high-quality labelled depth maps via 6DoF sensors-based methods. Therefore, it is possible to utilize labelled depth map as auxiliary information. Benefiting from external large-scale depth map datasets and existing labelled RGBD data, the work [163] proposes to align the features from real-world RGB images to the features from real-world depth maps.

As shown in Fig. 3.4, there are two branches (depth branch and RGB branch), and each for one input modality. For the purpose to transfer information, both branches adopt same

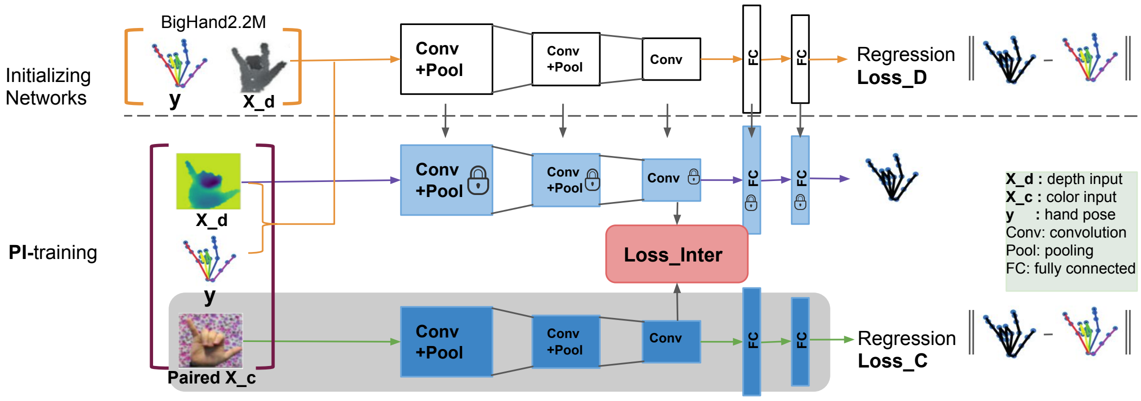


Figure 3.4: Exploration of depth maps for enhancing RGB-based 3D hand pose estimation. There are two branches, a depth branch (top) and a RGB branch (bottom). First, two branches are pre-trained independently to minimize 3D pose loss to exploit external large-scale depth maps. Then, based on paired RGBD data, two branches are fine-tuned based on their 3D pose losses and an intermediate feature loss which encourages the features from RGB branch to mimic the responses of the corresponding features from depth branch. This feature imitation transfers common visual cues from the depth branch to the RGB branch. The image is from [163].

backbone. Especially, CNNs are preferred as they are widely used in hand pose estimation and have shown powerful ability to transfer knowledge across networks.

To exploit the external large-scale depth map datasets, they first initialize the depth branch based on a numerical coordinate regression. With predicted joints $\mathbf{J}_{dm}$ from depth map and their corresponding ground truth $\mathbf{J}^{gt}_{dm}$, the pre-training joint regression loss is

$$\mathcal{L}_D = ||\mathbf{J}_{dm} - \mathbf{J}^{gt}_{dm}||_2. \tag{3.7}$$

Similarly, they also initialize the RGB branch based on the small size RGB data. With predicted joints $\mathbf{J}_{rgb}$ from RGB images and their corresponding ground truth $\mathbf{J}^{gt}_{rgb}$, the joint regression loss is

$$\mathcal{L}_C = ||\mathbf{J}_{rgb} - \mathbf{J}^{gt}_{rgb}||_2. \tag{3.8}$$

With RGBD paired labelled data and a pre-trained deep model for depth maps, they introduce the intermediate features from depth maps as supervision to aid the training of RGB images. For the k^{th} layer feature maps from depth branch and RGB branch, f^k_{dm} and f^k_{rgb}

$$\mathcal{L}_{inter} = ||\text{stop}(f^k_{dm}) - f^k_{rgb}||_2. \tag{3.9}$$

Here, k is chosen as a hyper-parameter, stop$(\cdot)$ is a stop-gradient operation to fix depth branch as the depth branch is more accurate. This loss encourages the features from RGB branch to mimic the responses of the corresponding features from depth branch. The final loss to fine-tune the RGB branch is

$$\mathcal{L} = \mathcal{L}_{inter} + \lambda \mathcal{L}_{C}, \tag{3.10}$$

where λ is used to balance the two losses.

Instead of mimicking the intermediate features from other modalities, based on knowledge distillation, Zhao *et al.* [172] introduces a teacher-student framework to distill knowledge learned from other modalities to real-world RGB images using paired examples.

3.2.2 Transfer from Synthetic Data

Synthesizing samples is an easy way to get accurate labels. More and more synthetic dataset [178, 46, 37] and synthetic data generator [4, 9] are introduced. Existing works also prefer to incorporate synthetic data to enrich the training data. However, synthetic data may have particular blending artifacts and are still far from “realistic”. In this case, deep models trained on synthetic data easily over-fit to specific blending artifacts and the performance of models can deteriorate significantly when applied to real-world data.

Given the pre-trained deep models for synthetic RGB, Rad *et al.* [97] aims to learn a mapping from the features of real-world RGB to the intermediate representations of synthetic RGB. They first construct paired real/synthetic data by rendering 3D model under the same ground truth of real-world RGB to obtain the corresponding synthetic RGB. After that, the mapping network is trained by minimizing the distance between the features extracted from the real-world RGB and that from the corresponding synthetic RGB. The mapping mostly removes the large difference between real-world and synthetic RGB and therefore reduces the domain gap.

Other works [178, 76] emphasize the importance of data augmentation for synthetic data and empirically demonstrate both color and geometry augmentations offer complementary benefits. Beyond data augmentation, more recent works turn to enhancing the appearance of synthetic data. Inspired by Cycle Generative Adversarial Network (GAN), Mueller *et al.* [81] takes unpaired real/synthetic images as input during training and translates synthetic images to “real” images using an adversarial loss and a cycle-consistency loss. Since hand pose is sensitive to the geometric perturbation, a geometric information, *i.e.* , hand masks, is introduced as auxiliary supervision. To take both high/low frequency into account, Chen *et al.* [16] explicitly decomposes RGB hand into color part (blurred RGB) and shape part (hand edge), and propose conditional GAN with those two parts as conditions for image generation. Therefore, it can generate realistic hand images and keep the color and shape from synthetic data.

3.2.3 Others

Domain adaptation techniques for 3D hand pose estimation are still in the infancy. Related research fields like 2D hand pose estimation [53] have shown some potential directions. With the observation that the failure predictions in the target domain usually fall on the wrong joint locations instead of background, Jiang *et al.* [53] proposes a sparse output space for 2D pose estimation based on the potential joint locations and use this sparse output space to guide the adversarial training to minimax of target disparity [171].

3.3 Hand Models

In this section, we introduce three paradigms (*i.e.* , model-driven, data-driven and hybrid), to exploit hand models for 3D hand pose estimation.

3.3.1 Model-driven Paradigm

Considering that the articulated hand model (*e.g.* , a hand surface model) is given or hypothesized, the target of hand model-based approaches is to fit the model to the available references (*e.g.* , depth maps, 2D poses and segmentation masks) (See Fig. 3.5 (b)). This can be formulated as an optimization problem whose objective function measures the discrepancy between the references and their corresponding pixels/points based on the articulated model. Direct optimization may not require training and are more easily extendable. Note that existing model-driven methods are all using depth maps as input due to the fact that depth maps can serve as references while RGB images can not. In this case, for model-driven paradigms, we only introduce 3D hand pose estimation from depth maps.

LM. A straightforward optimization for the articulated hand model is using LM algorithm. Based on a smooth user-specific hand mesh model and subdivision surface, Taylor *et al.* [124] introduces to optimize the model parameters with an LM-based optimizer; they minimize fitting energy with as-rigid-as-possible regularizers to deform the hand model to fit a target point cloud. Beyond a user-specific model, Khamis *et al.* [55] builds a skeleton-driven morphable mesh model of hands with aspects of pose and shape. This mesh is articulated using standard LBS and based on a smooth subdivision surface like [124]. They parameterize the model with latent parameters, that can be optimized using the LM optimizer. Further, Taylor *et al.* [123] accelerates the optimization by introducing a better initialization and reformulating the energy function as a weighted sum of several terms for model fitting, which is amenable for the proposed optimizer.

ICP. LM optimizer as a local optimizer needs multiply iterations, even under the situation that model is rigid non-articulated and correspondences are known. In contrast, in that situation, vanilla ICP would recover the registration of non-articulated bodies in a single iteration. However, the vanilla ICP is non-applicable for articulated models. To extend ICP for articulated structures, the work [92] proposes articulated ICP with structure constraints for point clouds; they divide the articulated structure into parts, which can be aligned rigidly in the way of the vanilla ICP, and keep the articulated structure feasible using additional constraints. To improve the efficiency of ICP, Stoll *et al.* [116] uses a sparse subset instead of the whole point cloud as reference. To further avoid ICP suffering from local minima, Ganapathi *et al.* [32] introduces to combine ICP with ray-casting likelihood function using depth

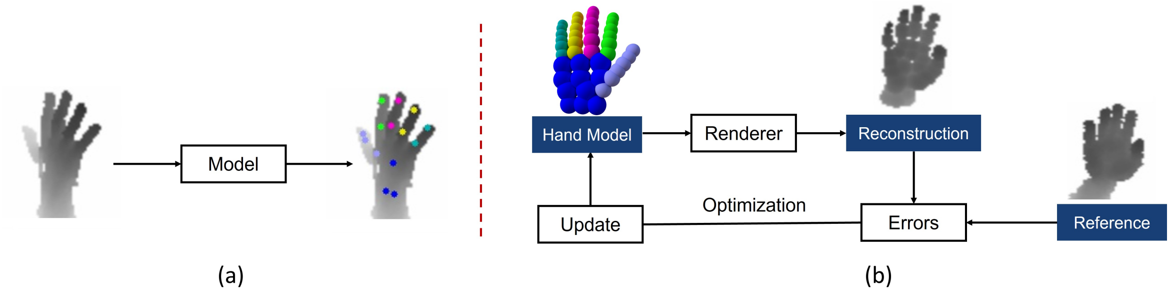


Figure 3.5: Comparison of the data-driven paradigm (a) and the model-driven paradigm (b). We can see the data-driven paradigm requires training data during training and get one-shot estimation during inference. In contrast, the model-driven paradigm without training needs several optimization steps to model-fit to the references. Images reproduced from [132].

maps as references. ICP and ray-casting likelihood function are complementary, balancing the fitting accuracy and efficiency.

Others. Recently, more and more optimization strategies have been explored to efficiently search the high-dimensional parameter space of hand configurations. For example, the work [87] proposes Particle Swarm Optimization (PSO) and another work [88] presents an evolutionary algorithm. Interestingly, decomposing the high-dimensional parameter space of hand configurations into smaller ones has also proven to be effective and efficient. Specifically, Tang *et al.* [121] decomposes pose parameters into subsets based on the tree structure of hand and proposes hierarchical sampling optimization to optimize the parameter space accordingly.

3.3.2 Data-driven Paradigm

Model-driven methods, as online optimization methods, are sensitive to the time complexity of the optimization during inference. In contrast, data-driven methods with one-shot prediction can quickly deliver a solution (See Fig. 3.5 (a)). Here, we introduce deep-learning-based methods for RGB-based 3D pose estimation because they have dominated as the best performers of data-driven methods. Specifically, we introduce deep-learning-based methods based on the type of hand models, parametric or non-parametric models.

Parametric Models. To embed the existing predefined articulated hand models into the deep learning frameworks, works [4, 169, 9, 46] develop a parametric model, MANO, as a differentiable layer. Based on this differentiable hand model layer, works like [4, 169, 9, 46] propose neural networks to predict the parameters of MANO to fit the hand models to RGB images using different modalities (*e.g.* segmentation masks, 2D poses, 3D poses) as supervisions. As the MANO model provides explicit control over the shapes and the poses of 3D hand meshes, works [4, 9] also render RGB images based on MANO and neural renderers [54] to enrich the training data. However, in practice, model fitting with a parametric model may suffer from poor fitting performance due to the mediocre image-model alignment.

Non-Parametric Models. Unlike hand parametric models are predefined based on LBS,

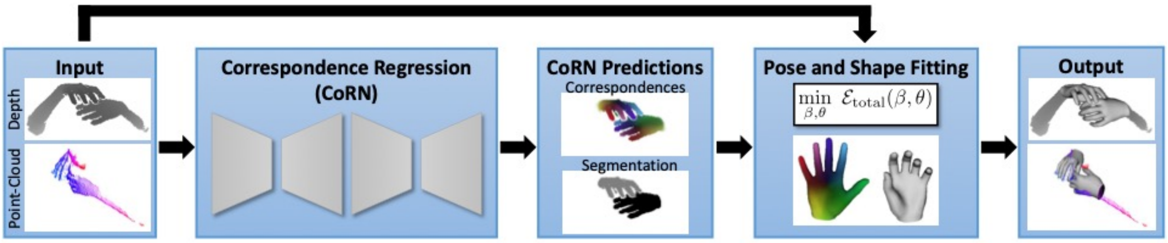

Input
Depth
Point-Cloud
Correspondence Regression (CoRN)
CoRN Predictions
Correspondences
Segmentation
Pose and Shape Fitting
$\min_{\beta,\theta} \mathcal{E}_{\text{total}}(\beta,\theta)$
Output

Figure 3.6: Illustration of a classic hybrid pipeline. For the data-driven part, given a depth image as input, the correspondence regression network (CoRN) provides segmentation masks and correspondence maps as references. For the model-driven part, an energy minimization optimization is adopted to fit the parameters of a parametric model (MANO) to best explain the references from data-driven part. Note that even this framework is designed for a depth map input, it is also applicable for an RGB-based framework directly. The image is from [82].

non-parametric hand models rely on training data and ad-hoc network architectures based on the hand structure. Non-parametric models as "implicit" hand models, can be trained with other part of the network simultaneously. Recently, as more and more large-scale 3D hand surface data have been synthesised or collected [37, 63], non-parametric models start to draw more and more attention for hand. Based on Chebyshev spectral graph CNN [26], works like [37, 64] encode RGB into a latent representation of meshes and then recover the full 3D hand surface mesh. The difference between those two methods is, besides mesh losses, Ge *et al.* [37] uses both 3D poses and depth maps as auxiliary supervisions while Kulon *et al.* [64] only introduces the readily available 2D poses as auxiliary supervision. Furthermore, instead of Chebyshev spectral mesh decoder, the work [63] presents a spatial convolutional mesh decoder to directly reconstructs meshes in image coordinates, which also shows superior performance on mesh reconstruction and pose estimation. Based on the voxel representations, Moon *et al.* [78] models the location of a mesh vertex using a 3D heatmap representation and formulates hand mesh estimation as vertex heatmap estimation. However, directly predicting 3D heatmaps is computationally infeasible for large-scale vertices. As an alternative, the work [78] proposes to predict one-dimensional heatmaps for each mesh vertex coordinates. Considering the 3D heatmaps as joint distributions of all coordinates, each one-dimensional heatmap represents the marginal distributions of the coordinates like [160].

3.3.3 Hybrid Paradigm

Optimization from scratch needs more iterations to converge and is easy to fall into local minima. In contrast, optimization with good initialization is more efficient. Data-driven methods can quickly deliver a solution but often suffer from lower accuracy or missing anatomical validity; using parametric models, the models have limited fitting ability as the parameters of models are highly constrained to specified forms; using non-parametric models, the models have the risk of over-fit to the training data and generate infeasible predictions.

As deep models and optimization are somehow complimentary by nature, hybrid methods with both elements are appealing to inherit the advantages of both paradigms. Overall, hybrid methods initial poses and references with the deep-learning-based part, and then refine the poses based on the predicted references and the model-driven part. We show a classic hybrid pipeline in Fig. 3.6. We can see the deep-learning-based part predicts segmentation masks and correspondence maps as references based on a large dataset for the model-driven part. Using MANO as hand model, the model-driven part takes the feasibility and the references into account to optimize the parameters of MANO.

To start with a good initialization, hybrid works explore to use tight hand regions [61, 102], the fingertips [114, 96, 140] or selected joints [93, 159, 132] from the detection modules. As for the hand sequences, initialization from the previous frame is also preferable [82]. References or observations are essential to energy minimization. Most methods [61, 132] use the depth maps or hand segmentation masks as references. Note that with auxiliary hand annotations for data-driven methods, other annotations like 2D heatmaps [128], hand part labels [112] or UV mappings from 3D mesh [135] can be predicted and serve as references to speed up the optimization of energy minimization.

3.4 Representation Learning

In this section, we will introduce two common representation learning techniques, *i.e.* , disentangled representation learning and contrastive learning, for hand pose estimation.

3.4.1 Disentangled Representation

Disentangled representations disentangle the features based on salient factors of variation and encode them as separate dimensions. Existing disentangled representations can be learned in an unsupervised setting or by exploiting “cheap” weak labels such as grouping information [8, 62, 39] and pairwise similarities [52].

One potential use of disentangled representations is we can precisely localize task-relevant features and omit the task-irrelevant ones. To learn disentangled representations, one effective solution is to minimize the mutual information [23]. To learn disentangled task-relevant features for unsupervised domain adaptation tasks, an information-theoretical framework as shown in Fig. 3.7 is commonly used. In an unsupervised domain adaptation setting, we have images $\mathbf{x}^s$ from the source domain $\mathcal{X}^s$ with labels $\mathbf{y}^s$ and images $\mathbf{x}^t$ from the target domain $\mathcal{X}^t$ without labels. Our target is to get satisfying predictions in both source and target domain without an obvious domain gap. To achieve this, two encoders, the content encoder E_c and the domain encoder E_d, are introduced. We train the encoders to disentangle the features into task-relevant features $\mathbf{z}_c = E_c(\mathbf{x})$ and task-irrelevant features (*i.e.* , domain features) $\mathbf{z}_d = E_d(\mathbf{x})$. Noticed that besides the labels $\mathbf{y}^s$ for images $\mathbf{x}^s$, we also have the domain labels.

The content embedding $\mathbf{z}_c^s$ from the source domain is further used as an input to a content decoder $C(\cdot)$ to get the predictions, with a content loss $\mathcal{L}_c$ defined as based on the specific task. For 3D hand pose numerical coordinate regression, the content loss could be

$$\mathcal{L}_c = ||\mathbf{y}^s - C(\mathbf{z}_c^s)||_2. \tag{3.11}$$

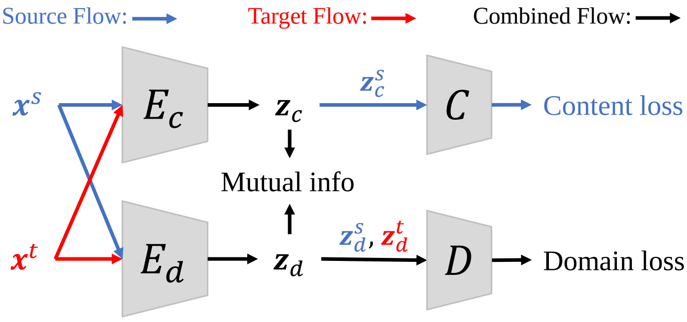


Figure 3.7: Illustration of a disentangled representation-based framework for unsupervised domain adaptation. The source data $\mathbf{x}^s$ and the target data $\mathbf{x}^t$ are passed to a task-relevant encoder E_c and a domain-relevant encoder E_d, with output features $\mathbf{z}_c$ and $\mathbf{z}_d$, respectively. To encourage $\mathbf{z}_c$ and $\mathbf{z}_d$ to be disentangled, task-relevant loss (content loss), domain loss and mutual information minimization loss are adopted. C is the content decoder to get task-relevant predictions, and D is the domain discriminator. The mutual information between $\mathbf{z}_c$ and $\mathbf{z}_d$ is minimized. The image is from [23].

The domain embedding $\mathbf{z}_d$ (including $\mathbf{z}_d^s$ and $\mathbf{z}_d^t$) is input to a domain discriminator $D(\cdot)$ to predict whether the observation comes from the source domain or target domain, with a domain loss defined as

$$\mathcal{L}_d = \mathbb{E}_{\mathbf{x}\in\mathcal{X}^s}[\log D(\mathbf{z}_d)] + \mathbb{E}_{\mathbf{x}\in\mathcal{X}^t}[\log(1 - D(\mathbf{z}_d))]. \tag{3.12}$$

Since the content information and the domain information should be independent, we minimize the mutual information between the content embedding $\mathbf{z}_c$ and domain embedding $\mathbf{z}_d$. The mutual information estimator MI$(\cdot)$ could be L1Out [95] or CLUB [23]. The final objective with λs as hyper-parameters is:

$$\min_{E_c,E_d,C,D} \mathrm{MI}(\mathbf{z}_c, \mathbf{z}_d) + \lambda_c\mathcal{L}_c + \lambda_d\mathcal{L}_d. \tag{3.13}$$

On the other hand, disentangled representations can be learned for image or video generation or translation. Such representations have been applied successfully to image editing [8, 25, 62, 83, 103, 120], video generation [129] and image-to-image translation [52]. To learn a disentangled and controllable latent representation, existing works [27, 39] resort to constructing paired images where their difference is only one factor of variation has changed. With those paired images, they encourage unchanged part of features extracted from paired images as close as possible via contrastive learning [67, 27] or cycle consistency [52]. However, it is nontrivial to get paired images. One easy solution is to utilize synthetic images even there is an added challenge of domain adaptation to the real-world images [27]. Based on the synthetic data, We can train the image generation network to imitate the image rendering process and use adversarial training to reduce the domain gap between synthetic images and real-world images.

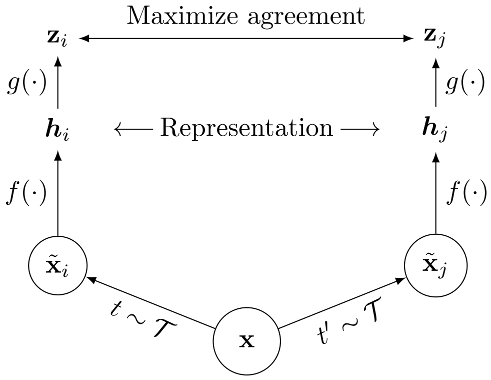


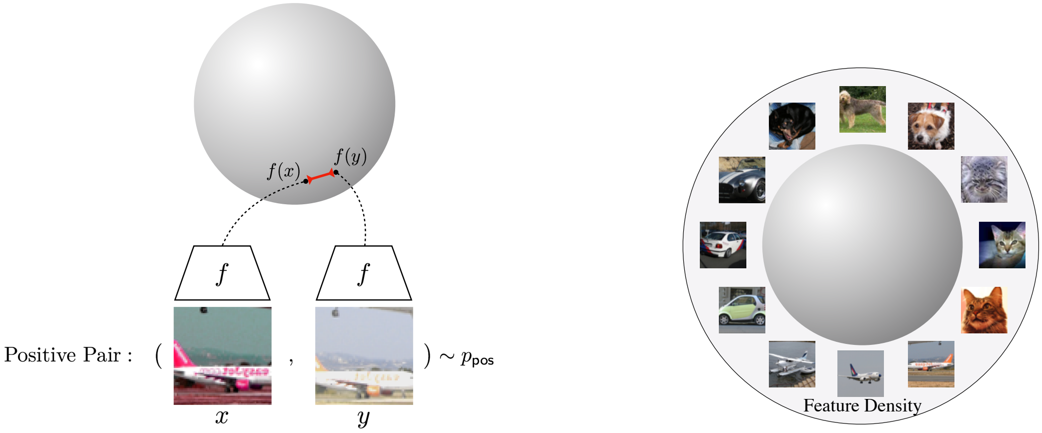
Figure 3.8: Illustration of a simple framework for contrastive learning. A positive pair $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ is generated based on the same data $\mathbf{x}$ but with different augmentations. The positive pair is then passed to a base encoder network $f(\cdot)$ and a projection head $g(\cdot)$, with representations $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, and output $\mathbf{z}_i$ and $\mathbf{z}_j$. $f(\cdot)$ and $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training, the representations h can be used for downstream tasks. The image is from [17].

3.4.2 Contrastive Learning

Contrastive learning is a powerful representation learning technique used to learn the general features by encouraging the model to learn a low-dimensional space for data in which similar sample pairs (positive pairs) stay close together while dissimilar samples (negative pairs) are further apart. It has been successfully applied in the unsupervised setting [125, 18, 17]. Creating beneficial positive-negative pairs forms the basis of contrastive learning. Existing works [125, 18, 17, 56] prefer to create positive pairs based on data augmentation. As shown in Fig. 3.8, two data augmentation operators are sampled from the same family of augmentations $\mathcal{T}$ and applied to data $\mathbf{x}$ to obtain two correlated views, $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$. After that, a base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. The projection head $g(\cdot)$ usually is two fully-connected layers to obtain 128-dimension normalized latent features. Using a minibatch of N examples as input, $2N$ data points are obtained after data augmentation. We treat $\mathbf{z}_i$ and $\mathbf{z}_j$ from same sample as a postive pair and the other $2(N-1)$ augmented examples within a minibatch as negative examples. We have the loss function for a positive pair of examples (i, j) is defined as below:

$$\ell_{i,j} = -\log \frac{\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]}(\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau\right))}, \tag{3.14}$$

where τ is a temperature parameter and usually set to 0.5, $\text{sim}(\cdot, \cdot)$ is the cosine similarity, and $\mathbb{1}$ is the indicator function. It also can be applied in a supervised setting [56] by simply combining supervised losses with the contrastive loss. Based on the explanation from [138], the features of contrastive representation learning will obtain the benefit of two properties, the alignment of features from positive pairs (See Fig. 3.9 (a)), and the uniformity of the induced distribution of the (normalized) features on the hyper-sphere (See Fig. 3.9 (b)). This



(a) **Alignment:** Similar samples have similar features.

(b) **Uniformity:** Preserve maximal information.

Figure 3.9: Illustration of two properties of contrastive representations, alignment and uniformity of feature distributions on the output unit hypersphere. Contrastive learning makes similar sample pairs stay close together while dissimilar samples are further apart. Also contrastive learning encourages features uniformly distributed throughout the hypersphere feature space. Images are from [138].

explanation provides the effectiveness of contrastive learning as a representation learning technique and also highlights the necessity of using normalized latent features on the unit hyper-sphere for contrastive learning.

Existing pose estimation works [177, 109] propose to contrastive pre-train on the unlabelled data and then fine-tune on the labelled data as shown in Fig. 3.10. Unlike classification, contrastive learning in 3D pose estimation needs to account for semantic information. To construct positive pairs for hand poses, Zimmermann *et al.* [177] simply augments the images by performing randomized: crops of the image, color jitter, grayscale, and conversion and Gaussian blur. Also, based on their collected multi-view green-screen background hand sequences, they introduce more positive pairs via background randomization and temporal/multi-view sampling. After contrastive pre-training, they adopt supervised training like other existing supervised training pipelines [179] but using a smaller learning rate to fine-tune the models.

3.5 Less Supervision

Even existing fully supervised methods have improved the performance significantly on public datasets, they usually fail to handle cross-dataset evaluation. Moreover, obtaining diverse and sufficient training data is non-trivial and supervised approaches are still data-hungry. Therefore, increasing attention is paid to 3D hand pose estimation with less supervision.

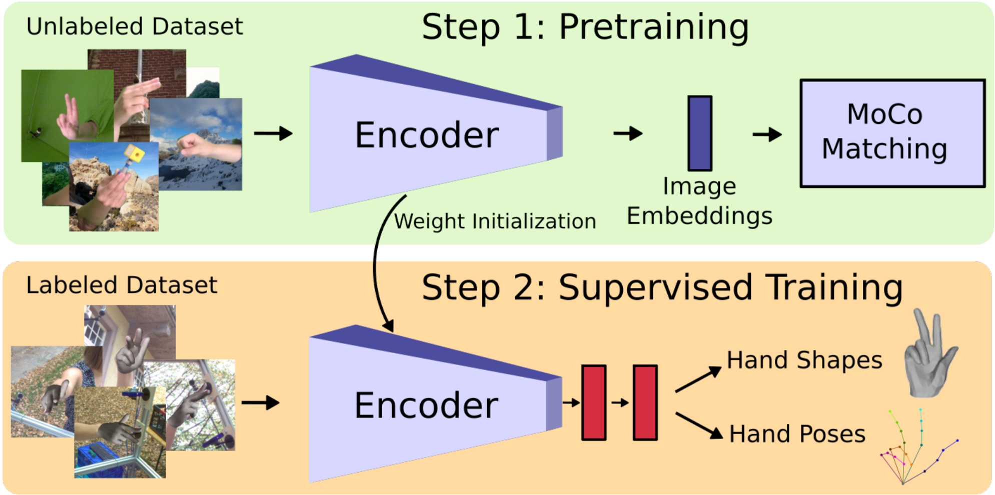


Figure 3.10: Illustration of a pre-training and fine-tuning pipeline for 3D hand pose estimation. During pre-training, a contrastive learning strategy for a large unlabelled hand image dataset is adopt to yield useful image embeddings that can be used for the downstream hand pose estimation task. During fine-tuning, the encoder is fine-tuned based on a smaller labelled dataset. using a smaller learning rate. The image is from [177].

3.5.1 Weakly-Supervised learning

Hand modalities are representations of hand in different aspects. While obtaining paired real-world RGB images and 3D annotation is expensive, existing methods resort to exploring the use of RGB images with other corresponding modalities, *e.g.* 2D keypoints, segmentation masks or depth maps. Real-world RGB images with corresponding easy-to-achieve modalities as weak labels is an effective way to improve the cross-dataset performance.

To alleviate the burden of 3D annotations in real world, works like [28, 11] first propose to fine-tune the pre-trained model by leveraging the depth maps as weak labels. They both need renderers to get depth maps from 3D poses. Their difference is mainly the design of the renderers. Dibra *et al.* [28] obtains depth maps from a parametric hand model while Cai *et al.* [11] prefers to estimate depth maps from a pre-trained renderer network. The follow-up works [137, 9, 63] either improve the pre-trained renderer networks or explore 3D hand surface models. To relief the concern of insufficient real-world depth maps, the work [14] proposes a conditional GAN model to generate realistic depth maps conditioned on the input RGB image. Therefore, the synthesized depth maps can be used to regularize the pre-trained model.

While 2D annotations are much easier to obtain than 3D annotations, 2D annotations are one of the most efficient weak labels. Boukhayma *et al.* [9] exploits 2D annotations as weak labels by embedding a differentiable parametric hand model (*i.e.* , MANO) and an orthographic camera model into the framework. As shown in Fig. 3.11, with predicted pose parameters θ and shape parameters β, 3D poses can be obtained by $J(\beta, \theta)$ where $J(\cdot)$ is a predefined function in MANO to get 3D poses from hand parameters. For the camera

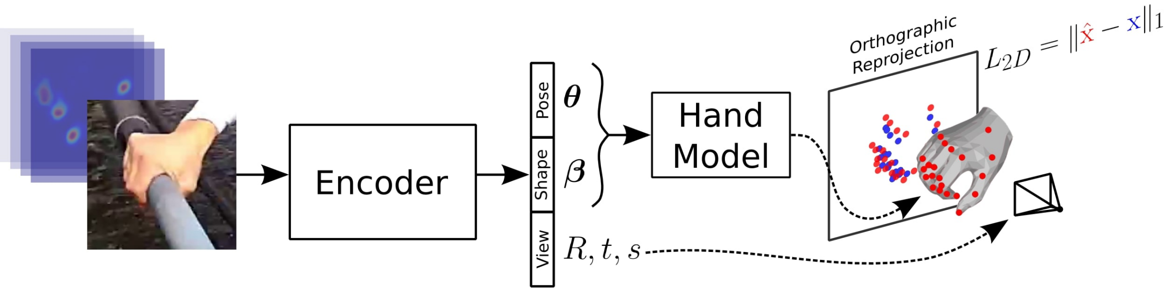


Figure 3.11: Illustration of a weakly-supervised pipeline based on MANO and a weak perspective camera model. The pipeline takes as input a hand image and optionally 2D joint heatmaps. The encoder produces shape parameters and pose parameters for MANO to generate 3D poses, and view parameters for camera model. Using orthographic re-projection, the pipeline exploits 2D poses as weak labels for 3D hand pose estimation.The image is from [9].

model, a global rotation matrix R, a translation t and a scaling s are estimated. Based on orthographic projection function $\Pi(\cdot)$, 2D poses $\hat{\mathbf{y}}$ are obtained

$$\hat{\mathbf{y}} = s\Pi(RJ(\beta, \theta)) + t. \tag{3.15}$$

Therefore, the weak supervision is a simple ℓ_1 loss between predicted 2D poses $\hat{\mathbf{y}}$ and ground truth 2D poses $\mathbf{y}$:

$$\mathcal{L}_{2D} = ||\hat{\mathbf{y}} - \mathbf{y}||_1. \tag{3.16}$$

Even existing 2D annotation-based weakly-supervised frameworks are effective, the key difficulty stems from the fact that direct application of additional 2D supervision mostly benefits the 2D proxy objective but does little to alleviate the depth and scale ambiguities [110]. To embrace this challenge, the work [110] introduces biomechanical losses that constrain the predicted bone lengths and joint angles to lie within the valid ranges based on the biomechanical feasibility of 3D hand configurations. Those biomechanical losses further improve the performance when using 2D poses as weak labels.

3.5.2 Semi-Supervised learning

Semi-supervised learning for RGB-based 3D hand pose estimation is still in its infancy. The only work [71] limits the research scenario on large-scale unlabelled videos and proposes spatial-temporal consistency constraints to encourage the consistency between predicted 3D poses and predicted 2D poses, and the consistency of hand shape (skeleton) and 2D pose over time. As a complement, we introduce semi-supervised learning in two related research fields, 3D hand pose estimation from depth maps and 2D pose estimation.

3D Hand Pose Estimation from Depth Maps. Unlike RGB images, a depth map contains 2.5D information regardless of texture or lighting information, which makes it easy to build connection to 3D surface models. Therefore, reconstruction of depth maps or their

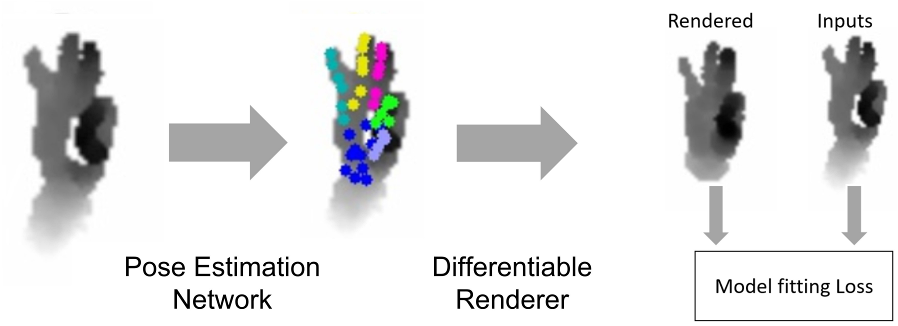


Figure 3.12: Illustration of a semi-/self-supervised pipeline. It adopts synthetic data to warm up the pose estimation network and uses consistencies within the data and the hand model to serve as a label for learning. The image is from [132].

equivalent representations (such as point clouds) is used to improve the representation learning in a self-/semi-supervised setting.

Works like [20, 133] aim at making use of the unlabelled depth maps by projecting both hand poses and depth maps into a shared latent space. The rationale is that a large number of unlabelled depth maps will build a well-structured depth map space, and limited paired hand poses and depth maps will encourage to align pose space to depth map space. As a follow-up, the work [94] proposes to extend the corresponding depth maps from single-view to multi-view for hand poses.

Those data-driven strategies are effective, and the shared latent space implicitly models the hand. However, it is still hard to guarantee the feasibility of a hand. Also, the data synthesizers and data-driven discriminators are isolated. To overcome this, recent works like [132, 135] propose a hybrid framework to integrate discriminative models with model fitting modules. Specifically, they use synthetic data to warm up the discriminative models and then adopt the prediction from discriminative models as references to drive the model fitting module. The final objective is the self reconstruction losses based on the hand models. Note that an explicit hand model exists in the model fitting module to render depth maps. The design of the hand model directly determines the model fitting and generalization ability of the entire hybrid framework. We show one classic self-supervised hybrid pipeline in Fig. 3.12. The rationale of this self-supervised hybrid pipeline is to use the consistency between inputs and pre-defined articulated hand models to serve as labels for learning.

2D Pose Estimation. Works like [66, 80, 99] attempt to improve the stability of predictions on unlabelled data by exploiting various consistency constraints, pseudo-labelling strategies and training strategies. To discover the consistency between input and output, different augmentation strategies have been explored. If the augmentation does not change factors associated with 2D pose estimation, the predicted pose is expected to be the same. In contrast, if the augmentation causes geometric transformations in 2D images, the prediction should be changed accordingly. The rationale here is that the predictions of a well-behaved model should be robust to noise or perturbation. To explore the consistency over time, a smoothness consistency is proposed to regularize the predicted poses. It is reasonable to assume that the poses from neighboring frames can not change dramatically in a real-world video. As for pseudo-labelling, the key problem is how to find pseudo-labels with high confidence. The work [66] proposes to use the maximum value of a 2D heatmap to select pseudo-labels because a 2D heatmap itself models the probability of each pixel being a joint pixel. Differently, Mu *et al.* [80] proposes to determine the confidence of samples based on their consistencies, with the assumption that high-confident samples should satisfy these consistencies. Without explicitly estimating the confidence of a pseudo-label, the work [99] takes an ensembling strategy to determine the pseudo-label of an unlabelled image by aggregating the predictions from a single model applied to multiple transformations of this unlabeled image. Besides consistency constraints and pseudo-labelling strategies, the training strategies are also important. works [80, 66] borrow the ideas from curriculum learning to gradually increase the number of training samples and learn models in an iterative fashion, which will improve the stability of the framework during training. Moreover, the work [80] demonstrates a multi-task learning strategy can further boost the performance.

3.5.3 Hand Pose Correction

Even existing methods adopt complex networks and a large amount of data to fit the 3D poses, the biomechanical feasibility of a hand pose is not guaranteed. This issue will become increasingly serious when training with less supervision. We highlight hand pose correction because the feasible solution space of inherent articulated hand model is the key to distinguish 3D pose estimation from other localization problems. The biomechanical feasibility can be predefined and used as auxiliary information for 3D hand pose estimation. To avoid invalid poses during testing, different correction strategies are explored.

Most pose correction methods [90, 170] are based on explicit physical constraints of hand. Panteleris *et al.* [90] introduces joint limits by post-processing that fits the prediction to a 3D hand model with only plausible solution space of hand articulations. Zhang *et al.* [170] proposes a knowledge distillation framework to transfer the physical constraints of hand from a teacher network to a student network. Specifically, the teacher network corrects the angle-invalid pose predictions from the student network and provides comprehensive supervision based on the corrected poses. Especially, the work [110] proposes to build a local coordinate system for hand and introduces valid ranges of bone lengths and joint angles based on the statistics of feasible hand poses.

The data-driven-based pose correction methods are also preferred as a large amount of motion capture (MoCap) data have been collected. Pre-trained on Mocap data, the VAE-based prior [132, 91] is a simple yet efficient method to penalize infeasible hand poses while admitting valid ones. Zhou *et al.* [175] formulates 3D poses to angle-based representations as an inverse kinematics problem and proposes to regress the angles from 3D poses with a neural network. The network is pre-trained on MoCap data. It is robust to noise, and can correct the noisy 3D predictions.

Chapter 4

Disentangling Latent Hands for Image Synthesis and Pose Estimation

Contents

4.1 Motivation **43**

4.2 Related Works **45**

4.2.1 Hand Pose Estimation 45

4.2.2 Disentangled Representations 45

4.3 Methodology **46**

4.3.1 Cross Modal VAE 46

4.3.2 Disentangled VAE 46

4.3.3 Applications 49

4.4 Experimentation **51**

4.4.1 Implementation Details 52

4.4.2 Datasets & Evaluation 52

4.4.3 Synthesizing Images 52

4.4.4 3D Hand Pose Estimation 53

4.5 Conclusion **55**

This chapter presents a VAE-based framework with the image factors of variation as auxiliary information to get better representations for hand poses. Hand image synthesis and pose estimation from RGB images are both highly challenging tasks due to the large discrepancy between factors of variation ranging from image background content to camera viewpoint. Inspired by the procedure of image rendering, we make a strong assumption that the latent features extracted from RGB images can be deterministically decomposed into independent factors, which are directly associated with observed variables. Taking RGB hand images as examples, they contain independent image factors like image background contents and hand poses. Based on this assumption and the labels of factors, we aim to get task-relevant features or controllable features for image synthesis. As such, this chapter presents a VAE-based framework to learn disentangled representations for RGB hand images. Based on VAE, we introduce two steps, the disentangling step and the embedding step, to get the disentangled representations. In the disentangling step, we consider the joint distribution between images and factors of variation, and aim to generate images based on the latent variables from image factors. With the disentangling step, we learn a disentangled latent space based on the given image factors. In the embedding step, we aim to learn a mapping from images to the aforementioned disentangled latent space. Combining the disentangling step and the embedding step, we can get disentangled representations from RGB images and also generate images based on disentangled representations. In other words, the proposed disentangled variational autoencoder (dVAE) allows for specific sampling and inference of given factors. We also provide analysis for the objective of two steps based on evidence lower bound. The derived objective from the variational lower bound as well as the proposed training strategy is highly flexible, allowing us to handle cross-modal encoders and decoders as well as semi-supervised learning scenarios.

We verify our framework via multiple tasks, including RGB hand image synthesis and 3D pose estimation from RGB images on RHD dataset and STB dataset. For synthesizing hand images, we conduct the experiments based on two factors, *i.e.* , the image content and the hand pose. First, we show latent space walks from one image to another image. Specifically, we show the synthesized images when we interpolate the pose while keeping the image content fixed and when we interpolate image content while keeping the pose fixed. In both latent space walks, the reconstructed poses as well as the synthesized images demonstrate a smoothness and consistency of the latent space. Also, we show examples of pose transfer. We take poses from one image, content from other images and recombine them. Therefore, we can generate images with our selected image content and hand poses. For 3D hand pose estimation, we conduct the experiments based on two factors, the hand viewpoint and the canonical hand pose. We estimate viewpoints, canonical hand poses and 3D hand poses simultaneously, and achieve comparable performance compared to other state-of-the-art pose estimation works. Experiments show that our dVAE can synthesize highly realistic images of the hand specifiable by both pose and image background content and also estimate 3D hand poses from RGB images with accuracy competitive with state-of-the-art on two public benchmarks. As for future work, the assumption that the factors of variation here should be labelled and independent is too strict and limits its application in the real world. We will consider relaxing the need of labelled and independent between factors. Also, we will introduce contrastive learning and mutual information minimization to further improve the disentangled representations. The publication, contributors and author contributions in this chapter are listed below:

Publication:

- Linlin Yang and Angela Yao. "Disentangling Latent Hands for Image Synthesis and Pose Estimation." *IEEE Conference on Computer Vision and Pattern Recognition(CVPR).* 2019.

Other Contributors:

- Angela Yao (Thesis Supervisor)

Contributions:

- Linlin Yang proposed the framework, wrote the code and conducted the experiments. Linlin Yang and Prof. Dr. Angela Yao developed the idea, analyzed the results and wrote the main body of the article.

4.1 Motivation

Vision-based hand pose estimation has progressed very rapidly in the past years [118, 162], driven in part by its potential for use in human-computer interaction applications. Advancements are largely due to the widespread availability of commodity depth sensors as well as the strong learning capabilities of deep neural networks. As a result, the majority of state-of-the-art methods apply deep learning methods to depth images [34, 35, 36, 42, 75, 85, 86, 133, 134]. Estimating 3D hand pose from single RGB images, however, is a less-studied and more difficult problem which has only recently gained some attention [11, 81, 90, 111, 178].

Unlike depth, which is a 2.5D source of information, RGB inputs have significantly more ambiguities. These ambiguities arise from the 3D to 2D projection and diverse backgrounds which are otherwise less pronounced in depth images. As such, methods which tackle the problem of monocular RGB hand pose estimation rely on learning from large datasets [178]. However, given the difficulties of accurately labelling hand poses in 3D, large-scale RGB datasets collected to date are synthesized [81, 178]. Real recorded datasets are much smaller, with only tens of sequences [130, 167]. This presents significant challenges when it comes to learning and motivates the need for strong kinematic and or image priors.

Even though straight-forward discriminative approaches have shown great success in accurately estimating hand poses, there has also been growing interest in the use of deep generative models such as adversarial networks (GANs) [81, 133] and variational autoencoders (VAEs) [111]. Generative models can approximate and sample from the underlying distribution of hand poses as well as the associated images, and depending on the model formulation, may enable semi-supervised learning. This is particularly appealing for hand pose estimation, for which data with accurate ground truth can be difficult to obtain. One caveat, however, is that in their standard formulation, GANs and VAEs learn only black-box latent representations. Such representations offer little control for conditioning upon human-interpretable factors. Of the deep generative works presented to date [81, 111, 133], the latent representations are specifiable only by hand pose. Consequently it is possible to sample only a single (average) image per pose.

A recent work combining VAEs and GANs [25] introduced a conditional dependency structure to learn image backgrounds and demonstrated the possibility of transferring body poses onto different images. Inspired by this work, we would like to learn a similar latent representation that can disentangle the different factors that influence how hands may appear visually, *i.e.* normalized hand pose, camera viewpoint, scene context and background, *etc.* At the same time, we want to ensure that the disentangled representation remains sufficiently discriminative to make highly accurate estimates of 3D hand pose.

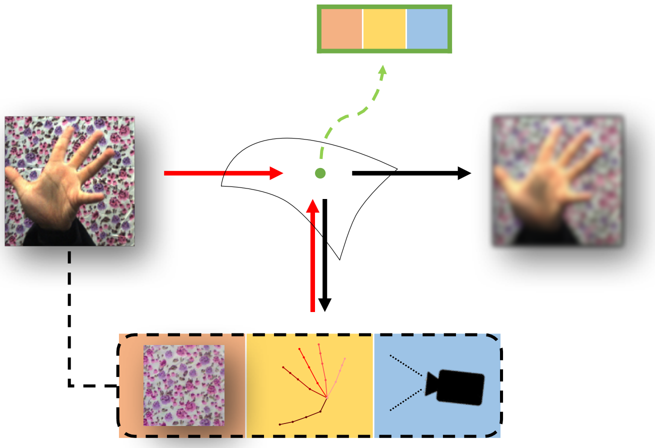


Figure 4.1: Illustration of dVAE. The red lines denote variational approximations while the black lines denote the generative model. With the help of labelled factors of variations (*e.g.* pose, viewpoint and image content), we learn a disentangled and specifiable representation for RGB hand images in a VAE framework.

We present a disentangled variational autoencoder (dVAE) – a novel framework for learning disentangled representations of hand poses and hand images. As the factors that we would like to disentangle belong to different modalities, we begin with a cross-modal VAE [89, 111] as the baseline upon which we define our dVAE. By construction, our latent space is a disentangled one, composed of sub-spaces calculated by factors and a training strategy to fuse different latent space into one disentangled latent space. We show how these disentangled factors can be learned from both independent and confounding label inputs. To the best of our knowledge, our proposed model is the first disentangled representation that is able to both synthesize hand images and estimate hand poses with explicit control over the latent space. A schematic illustration of our dVAE and the disentangled factors is shown in Fig. 4.1. We summarize our contributions below:

- We propose a novel disentangled VAE model crossing different modalities; this model is the first VAE-based model that uses independent factors of variations to learn disentangled representations.
- Our dVAE model is highly flexible and handles multiple tasks including RGB hand image synthesis, pose transfer and 3D pose estimation from RGB images.
- We enable explicit control over different factors of variation and introduce the first model with multiple degrees of freedom for synthesizing hand images.
- We decouple the learning of disentangling factors and the embedding of image content and introduce two variants of learning algorithms for both independent and confounding labels.

4.2 Related Works

4.2.1 Hand Pose Estimation

Much of the progress made in hand pose estimation have focused on using depth image inputs [34, 35, 36, 42, 50, 75, 79, 85, 86, 133, 134, 142]. State-of-the-art methods use a convolutional neural network (CNN) architecture, with the majority of works treating the depth input as 2D pixels, though a few more recent approaches treat depth inputs as a set of 3D points and or voxels [36, 34, 79].

Estimating hand poses from monocular RGB inputs is more challenging. Early methods could recognize only a restricted set of poses [3, 146] or used simplified hand representations instead of full 3D skeletons [115, 147]. In more recent approaches, the use of deep learning and CNNs has become common-place [11, 90, 178]. In [81, 111], deep generative models such as variational auto-encoders (VAE) [111] and generative adversarial networks (GANs) [81] are applied, which makes feasible not only to estimate pose, but also generate RGB images from given hand poses.

Two hand pose estimation approaches [133, 111] stand out for being similar to ours in spirit. They also use shared latent spaces, even though the nature of these spaces are very different. Wan *et al.* [133] learns two separate latent spaces, one for hand poses and one for depth images, and uses a one-to-one mapping function to connect the two. Spurr *et al.* [111] learns a latent space that cross multiple hand modalities, such as RGB to pose and depth to pose. To force the cross-modality pairings onto a single latent space, separate VAEs are learned in an alternating fashion, with one input modality contributing to the loss per iteration. Such a learning strategy is non-ideal, as it tends to result in fluctuations in the latent space and has no guarantees for convergence. Additionally, by assuming all crossing modalities as one-to-one mappings, only one image can be synthesized per pose.

Different from [133] and [111], our dVAE learns a single latent space by design. We learn the latent space with the different modalities jointly, as opposed to alternating framework of [111]. We find that our joint learning is more stable and has better convergence properties. And because we explicitly model and disentangle image factors, we can handle one-to-many mappings, *i.e.* synthesize multiple images of the same hand pose.

4.2.2 Disentangled Representations

Disentangled representations separate data according to salient factors of variation and have recently been learned with deep generative models such as VAEs and GANs. Such representations have been applied successfully to image editing [8, 25, 62, 83, 103, 120], video generation [129] and image-to-image translation [52]. Several of these works [103, 120, 129, 136], however, require specially designed layers and loss functions, making the architectures difficult to work with and extend beyond their intended task.

Previous works learning disentangled representations with VAEs [8, 52, 62] typically require additional weak labels such as grouping information [8, 62] and pairwise similarities [52]. Such labels can be difficult to obtain and are often not defined for continuous variables such as hand pose and viewpoint. In [25, 83], a conditional dependency structure is proposed to train disentangled representations for a semi-supervised learning. The work of [25] resembles ours in the sense that they also disentangle pose from appearance; however, their conditional dependency structure is sensitive to the number of factors. As the number of factors grows, the complexity of the network structure increases exponentially. In comparison to existing VAE approaches, we are able to learn interpretable and disentangled representations by the shared latent space produced by image and its corresponding factors without additional weak labels.

4.3 Methodology

4.3.1 Cross Modal VAE

Before we present how a disentangled latent space can be incorporated into a VAE framework across different modalities, we first describe the original cross modal VAE [89, 111]. As the name suggests, the cross modal VAE aims to learn a VAE model across two different modalities $\mathbf{x}$ and $\mathbf{y}$. We begin by defining the log probability of the joint distribution $p(\mathbf{x}, \mathbf{y})$. Since working with this distribution is intractable, one maximizes the evidence lower bound (ELBO) instead via a latent variable $\mathbf{z}$. Note that $\mathbf{x}$ and $\mathbf{y}$ are assumed to be conditionally independent given the latent $\mathbf{z}$, *i.e.* $(\mathbf{x} \perp \mathbf{y} \,|\, \mathbf{z})$.

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{y}) &\geq \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}, \mathbf{y}, \theta_{\mathbf{x}}, \theta_{\mathbf{y}}, \phi) \\ &= E_{\mathbf{z}\sim q_\phi} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) + E_{\mathbf{z}\sim q_\phi} \log p_{\theta_{\mathbf{y}}}(\mathbf{y}|\mathbf{z}) \\ &- D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})).\end{aligned} \tag{4.1}$$

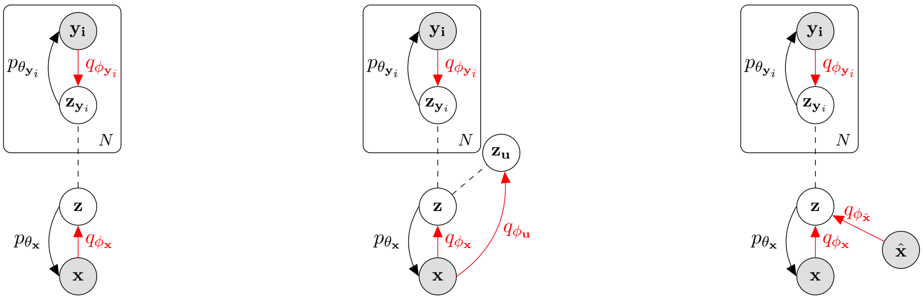
Here, $D_{KL}(\cdot)$ is the Kullback-Leibler divergence. The variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ can be thought of as an encoder from $\mathbf{x}$ to $\mathbf{z}$, while $p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z})$ and $p_{\theta_{\mathbf{y}}}(\mathbf{y}|\mathbf{z})$ can be thought of as decoders from $\mathbf{z}$ to $\mathbf{x}$ and $\mathbf{z}$ to $\mathbf{y}$ respectively. $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Gaussian prior on the latent space.

In the context of hand pose estimation, $\mathbf{x}$ would represent the RGB or depth image modality and $\mathbf{y}$ the hand skeleton modality. One can then estimate hand poses from images by encoding the image $\mathbf{x}$ into the latent space and decoding the corresponding 3D hand pose $\mathbf{y}$. A variant of this model was applied in [111] and shown to successfully estimate hand poses from RGB images or depth images.

4.3.2 Disentangled VAE

In our disentangled VAE, we define a latent variable $\mathbf{z}$ which can be deterministically decomposed into $N+1$ independent factors $\{\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{y}_2}, ..., \mathbf{z}_{\mathbf{y}_N}, \mathbf{z}_{\mathbf{u}}\}$. Of these factors, $\{\mathbf{z}_{\mathbf{y}_i}\}_{i=1...N}$ are directly associated with observed variables $\{\mathbf{y}_i\}_{i=1...N}$. $\mathbf{z}_{\mathbf{u}}$ is an extra latent factor which is not independently associated with any observed variables; it may or may not be included (compare Fig. 4.2a versus Fig. 4.2b).

Fully specified latent $\mathbf{z}$: We begin first by considering the simplified case in which $\mathbf{z}$ can be fully specified by $\mathbf{z}_{\mathbf{y}_i}$ without $\mathbf{z}_{\mathbf{u}}$, *i.e.* all latent factors can be associated with some


$\mathbf{y_i}$
$p_{\theta_{\mathbf{y}_i}}$
$q_{\phi_{\mathbf{y}_i}}$
$\mathbf{z}_{\mathbf{y}_i}$
N
$\mathbf{z}$
$p_{\theta_{\mathbf{x}}}$
$q_{\phi_{\mathbf{x}}}$
$\mathbf{x}$
$\mathbf{z_u}$
$q_{\phi_{\mathbf{u}}}$
$q_{\phi_{\hat{\mathbf{x}}}}$
$\hat{\mathbf{x}}$

(a) dVAE (b) dVAE with $\mathbf{z_u}$ (c) dVAE with $\hat{\mathbf{x}}$

Figure 4.2: Graphical models of disentangled VAEs. The shaded nodes represent observed variables while un-shaded nodes are latent. The red and black solid lines denote variational approximations q_ϕ or encoders, and the generative models p_θ or decoders respectively. The dashed lines denote deterministically constructed variables. Figure best viewed in colour.

observed $\mathbf{y}_i$. For clarity, we limit our explanation to $N=2$, though the theory generalizes to higher N as well. Our derivation can be separated into a disentangling step and an embedding step. In the ***disentangling step***, we first consider the joint distribution between $\mathbf{x}$, $\mathbf{y}_1$ and $\mathbf{y}_2$. The evidence lower bound of this distribution can be defined as:

$$\begin{aligned}\log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2) &\geq \text{ELBO}_{\text{dis}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2},\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2},\theta_{\mathbf{x}})\\ &= \lambda_{\mathbf{x}} E_{\mathbf{z}\sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z})\\ &+ \lambda_{\mathbf{y}_1} E_{\mathbf{z_{y_1}}\sim q_{\phi_{\mathbf{y}_1}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z_{y_1}})\\ &+ \lambda_{\mathbf{y}_2} E_{\mathbf{z_{y_2}}\sim q_{\phi_{\mathbf{y}_2}}} \log p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z_{y_2}})\\ &- \beta D_{KL}\left(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1,\mathbf{y}_2)||p(\mathbf{z})\right),\end{aligned} \tag{4.2}$$

where the λs and β are additional hyperparameters added to trade off between latent space capacity and reconstruction accuracy, as recommended by the β trick [48].

The ELBO in Eq. 4.2 allows us to define a disentangled $\mathbf{z}=[\mathbf{z}_{\mathbf{y}_1},\mathbf{z}_{\mathbf{y}_2}]$ based on $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{x}$. In this step, one can learn the encoding and decoding of $\mathbf{y}_i$ to and from $\mathbf{z}_{\mathbf{y}_i}$, as well as the decoding of $\mathbf{z}$ to $\mathbf{x}$. However, the mapping from $\mathbf{x}$ to $\mathbf{z}$ is still missing so we need an additional ***embedding step*** [131] to learn the encoder $q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$. Keeping all decoders fixed, $q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$ can be learned by maximizing:

$$\begin{aligned}\mathcal{L}(\phi_{\mathbf{x}}|\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2},\theta_{\mathbf{x}}) &= -D_{KL}\left(q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)\right)\\ &= \text{ELBO}_{\text{emb}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{x}}) - \log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2).\end{aligned} \tag{4.3}$$

Since the second term is constant with respect to $\phi_{\mathbf{x}}$ and the θ's, the objective simplifies to the following evidence lower bound with λ' and β' as hyperparameters:

$$\begin{aligned}\mathrm{ELBO}_{\mathrm{emb}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{x}}) = {} & \lambda'_{\mathbf{x}} E_{\mathbf{z}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) \\ & + \lambda'_{\mathbf{y}_1} E_{\mathbf{z}_{\mathbf{y}_1}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}) \\ & + \lambda'_{\mathbf{y}_2} E_{\mathbf{z}_{\mathbf{y}_2}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2}) \\ & - \beta' D_{KL}(q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).\end{aligned} \tag{4.4}$$

Combining the disentangling and embedding evidence lower bounds, we get the following joint objective:

$$\begin{aligned}\mathcal{L}(\phi_{\mathbf{x}},\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2},\theta_{\mathbf{x}},\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2}) = {} & \\ & \mathrm{ELBO}_{\mathrm{dis}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2},\theta_{\mathbf{x}},\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2}) \\ & + \mathrm{ELBO}_{\mathrm{emb}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{x}}).\end{aligned} \tag{4.5}$$

The above derivation shows that the encoding of modality $\mathbf{x}$ can be decoupled from $\mathbf{y}_1$ and $\mathbf{y}_2$ via a disentangled latent space. We detail the training strategy for the fully specified version of the dVAE in Alg. 4.1.

Additional $\mathbf{z_u}$: When learning a latent variable model, many latent factors may be very difficult to associate independently with an observation (label), *e.g.* the style of handwritten digits, or the background content in an RGB image [25, 62, 8]. Nevertheless, we may still want to disentangle such factors from those which can be associated independently. We model these factors in aggregate form via a single latent variable $\mathbf{z_u}$ and show how $\mathbf{z_u}$ can be disentangled from the other $\mathbf{z}_{\mathbf{y}_i}$ which are associated with direct observations $\mathbf{y}_i$. For clarity of discussion, we limit $N = 1$, such that $\mathbf{z} = [\mathbf{z}_{\mathbf{y}_1}, \mathbf{z_u}]$. To disentangle $\mathbf{z_u}$ from $\mathbf{z}$, both of which are specified by a confounding $\mathbf{x}$, we aim to make $\mathbf{z_u}$ and $\mathbf{y}_1$ conditionally independent

Algorithm 4.1 dVAE learning for fully specified $\mathbf{z}$.

```
Require: $\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \lambda_{\mathbf{x}}, \lambda_{\mathbf{y}_1}, \lambda_{\mathbf{y}_2}, \beta, T_1, T_2$
Ensure: $\phi_{\mathbf{x}}, \phi_{\mathbf{y}_1}, \phi_{\mathbf{y}_2}, \theta_{\mathbf{x}}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{y}_2}$
Initialize $\phi_{\mathbf{x}}, \phi_{\mathbf{y}_1}, \phi_{\mathbf{y}_2}, \theta_{\mathbf{x}}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{y}_2}$
for $t_1 = 1, \ldots, T_1$ epochs do
    Encode $\mathbf{y}_1, \mathbf{y}_2$ to $q_{\phi_{\mathbf{y}_1}}(\mathbf{z}_{\mathbf{y}_1}|\mathbf{y}_1), q_{\phi_{\mathbf{y}_2}}(\mathbf{z}_{\mathbf{y}_2}|\mathbf{y}_2)$
    Construct $\mathbf{z} \leftarrow [\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{y}_2}]$
    Decode $\mathbf{z}$ to $p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}), p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}), p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2})$
    Update $\phi_{\mathbf{y}_1}, \phi_{\mathbf{y}_2}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{y}_2}, \theta_{\mathbf{x}}$ via gradient ascent of Eq. 4.2
end for
for $t_2 = 1, \ldots, T_2$ epochs do
    Encode $\mathbf{x}$ to $q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$
    Construct $[\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{y}_2}] \leftarrow \mathbf{z}$
    Decode $\mathbf{z}$ to $p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}), p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}), p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2})$
    Update $\phi_{\mathbf{x}}$ via gradient ascent of Eq. 4.4
end for
```

given $\mathbf{z}_{\mathbf{y}_1}$ To achieve this, we try to make $p(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{u}})$ approximately equal to $p(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1})$ and update the encoder and the decoder of $\mathbf{y}_1$ by random sampling of $\mathbf{z}_{\mathbf{u}}$ and minimizing the distance between $p(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{u}})$ and $p(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1})$. The training strategy for this is detailed in Alg. 4.2. In this case, the joint distribution of $\mathbf{x}$ and $\mathbf{y}_1$ has the following evidence lower bound in the ***disentangling step*** with hyperparameters λ'' and β'':

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{y}_1) \geq & \, \text{ELBO}^{\mathbf{u}}_{\text{dis}}(\mathbf{x}, \mathbf{y}_1, \phi_{\mathbf{y}_1}, \phi_{\mathbf{u}}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{x}}) \\ =& \lambda''_{\mathbf{x}} E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1}, \phi_{\mathbf{u}}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) \\ +& \lambda''_{\mathbf{y}_1} E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1}, \phi_{\mathbf{u}}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}) \\ -& \beta'' D_{KL}(q_{\phi_{\mathbf{y}_1}, \phi_{\mathbf{u}}}(\mathbf{z}|\mathbf{y}_1, \mathbf{x})||p(\mathbf{z})).\end{aligned} \tag{4.6}$$

Note that in the above ELBO, $\mathbf{z}_{\mathbf{u}}$ is encoded from $\mathbf{x}$ by $q_{\phi_{\mathbf{u}}}$ instead of being specified by some observed label $\mathbf{u}$, as was done previously in [62, 8, 25]. After this modified disentangling step, we can apply the same embedding step in Eq. 4.3 to learn $q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$.

Multiple x modalities: The situation may arise in which we have multiple input modalities which fully specify and share the latent space of $\mathbf{z}$, *i.e.* not only an $\mathbf{x}$ but also an additional $\hat{\mathbf{x}}$ (see Fig. 4.2c). Here, it is possible to first consider the joint distribution between $\mathbf{x}$, $\mathbf{y}_1$ and $\mathbf{y}_2$, and maximize the ELBO in Eq. 4.2 for the disentangling step. To link the two modalities of $\mathbf{x}$ and $\hat{\mathbf{x}}$ into the same disentangled latent space and embed $\hat{\mathbf{x}}$, we can use the following:

$$\begin{aligned}\mathcal{L}(\phi_{\hat{\mathbf{x}}}|\theta_{\mathbf{x}}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{y}_2}) &= -D_{KL}(q_{\phi_{\hat{\mathbf{x}}}}(\mathbf{z}|\hat{\mathbf{x}})||p_{\theta}(\mathbf{z}|\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)) \\ &= \text{ELBO}'_{\text{emb}}(\hat{\mathbf{x}}, \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \phi_{\hat{\mathbf{x}}}) - \log p(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2).\end{aligned} \tag{4.7}$$

Similar to Eq. 4.4, we get the following evidence lower bound with λ''' and β''' as hyperparameters:

$$\begin{aligned}\text{ELBO}'_{\text{emb}}(\hat{\mathbf{x}}, \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \phi_{\hat{\mathbf{x}}}) =& \lambda'''_{\mathbf{x}} E_{\mathbf{z} \sim q_{\phi_{\hat{\mathbf{x}}}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) \\ &+ \lambda'''_{\mathbf{y}_1} E_{\mathbf{z}_{\mathbf{y}_1} \sim q_{\phi_{\hat{\mathbf{x}}}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}) \\ &+ \lambda'''_{\mathbf{y}_2} E_{\mathbf{z}_{\mathbf{y}_2} \sim q_{\phi_{\hat{\mathbf{x}}}}} \log p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2}) \\ &- \beta''' D_{KL}(q_{\phi_{\hat{\mathbf{x}}}}(\mathbf{z}|\hat{\mathbf{x}})||p(\mathbf{z})).\end{aligned} \tag{4.8}$$

For learning, one simply encodes $\hat{\mathbf{x}}$ with $q_{\phi_{\hat{\mathbf{x}}}}(\mathbf{z}|\hat{\mathbf{x}})$ to $\mathbf{z}$ instead of $p_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$ as shown currently in line 9 of Alg. 4.1.

4.3.3 Applications

Based on the theory proposed above, we develop two applications: image synthesis and pose estimation from RGB images. Like [178], we distinguish between an absolute 3D hand pose (3DPose), a canonical hand pose (CPose), and a viewpoint. The canonical pose is a normalized version of the 3D pose within the canonical frame, while viewpoint is the rotation matrix that rotates CPose to 3DPose.

In **image synthesis**, we would like to sample values of $\mathbf{z}$ and decode this into an image $\mathbf{x}$ via the generative model $p_{\theta_{\mathbf{x}}}$. To control the images being sampled, we want to have a latent

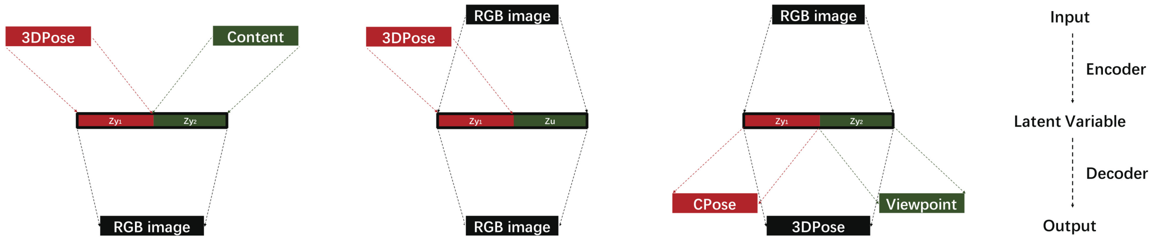


Figure 4.3: Inference models for the tasks of image synthesis (left and middle) and pose estimation (right).

$\mathbf{z}$ which is disentangled with respect to the 3DPose, and image (background) content, *i.e.* all aspects of the RGB image not specifically related to the hand pose itself. A schematic of the image synthesis is shown in the left panel of Fig. 4.3; in this case, we follow the model in Fig. 4.2a and use Alg. 4.1. Here, $\mathbf{y}_1$ would represent 3DPose and $\mathbf{y}_2$ would represent the image content; similar to [129], this content is specified by a representative tag image. By changing the inputs $\mathbf{y}_1$ and $\mathbf{y}_2$, *i.e.* by varying the 3DPose and content through the encoders $q_{\phi_{\mathbf{y}_1}}$ and $q_{\phi_{\mathbf{y}_2}}$, we synthesize new images with specified poses and background content. Furthermore, we can also evaluate the pose error of the synthesized image via the pose decoder $p_{\theta_{\mathbf{y}_1}}$.

Tag images for specifying background content are easy to obtain if one has video sequences

Algorithm 4.2 dVAE learning for additional $\mathbf{z}_\mathbf{u}$.

```
Require: $\mathbf{x}, \mathbf{y}_1, \lambda_\mathbf{x}, \lambda_{\mathbf{y}_1}, \beta, T_1, T_2, T_3$
Ensure: $\phi_\mathbf{x}, \phi_{\mathbf{y}_1}, \phi_\mathbf{u}, \theta_\mathbf{x}, \theta_{\mathbf{y}_1}$
Initialize $\phi_\mathbf{x}, \phi_{\mathbf{y}_1}, \phi_\mathbf{u}, \theta_\mathbf{x}, \theta_{\mathbf{y}_1}$
for $t_1 = 1, \dots, T_1$ epochs do
    Encode $\mathbf{x}, \mathbf{y}_1$ to $q_{\phi_{\mathbf{y}_1}}(\mathbf{z}_{\mathbf{y}_1}|\mathbf{y}_1), q_{\phi_\mathbf{u}}(\mathbf{z}_\mathbf{u}|\mathbf{x})$
    Construct $\mathbf{z} \leftarrow [\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_\mathbf{u}]$, $[\mu, \sigma] \leftarrow q_{\phi_\mathbf{u}}(\mathbf{z}_\mathbf{u}|\mathbf{x})$
    Decode $\mathbf{z}$ to $p_{\theta_\mathbf{x}}(\mathbf{x}|\mathbf{z}), p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z})$
    Update $\phi_{\mathbf{y}_1}, \phi_\mathbf{u}, \theta_{\mathbf{y}_1}, \theta_\mathbf{x}$
    for $t_2 = 1, \dots, T_2$ epochs do
        Encode $\mathbf{y}_1$ to $q_{\phi_{\mathbf{y}_1}}(\mathbf{z}_{\mathbf{y}_1}|\mathbf{y}_1)$
        Construct $\mathbf{z}_{noise} \leftarrow \mathcal{N}(\mu, \sigma)$, $\mathbf{z} \leftarrow [\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{noise}]$
        Decode $\mathbf{z}$ to $p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z})$
        Update $\phi_{\mathbf{y}_1}, \theta_{\mathbf{y}_1}$
    end for
end for
for $t_3 = 1, \dots, T_3$ epochs do
    Encode $\mathbf{x}$ to $q_{\phi_\mathbf{x}}(\mathbf{z}|\mathbf{x})$
    Construct $[\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_\mathbf{u}] \leftarrow \mathbf{z}$
    Decode $\mathbf{z}$ to $p_{\theta_\mathbf{x}}(\mathbf{x}|\mathbf{z}), p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z})$
    Update $\phi_\mathbf{x}$
end for
```

from which to extract RGB frames. However, for some scenarios, this may not be the case, *i.e.* if each RGB image in the training set contains different background content. This is what necessitates the model in Fig. 4.2b and the learning algorithm in Alg. 4.2. In such a scenario, $\mathbf{y}_1$ again represents the 3DPose, while the image content is modelled indirectly through $\mathbf{x}$. For testing purposes, however, there is no distinction between the two variants, as input is still given in the form of a desired 3DPose and an RGB image specifying the content.

For **hand pose estimation**, we aim to predict 3DPose $\mathbf{x}$, CPose $\mathbf{y}_1$ and viewpoint $\mathbf{y}_2$ from RGB image $\hat{\mathbf{x}}$ according to the model in Fig. 4.2c by disentangling $\mathbf{z}$ into the CPose $\mathbf{z}_{\mathbf{y}_1}$ and viewpoint $\mathbf{z}_{\mathbf{y}_2}$. In this case, we embed $\mathbf{x}$ and $\hat{\mathbf{x}}$ into a shared latent space. We apply inference as shown by the right panel in Fig. 4.3 and learn the model with Alg. 4.1. Moreover, because annotated training data is sparse in real world applications, we can further leverage unlabelled or weakly labelled. Our proposed method consists of multiple VAEs, which can be trained respectively for semi- and weakly-supervised setting. For semi-supervised setting, we use both labelled and unlabelled CPose, viewpoint and 3DPose data to train the encoders $q_{\phi_{\mathbf{y}_1}}$,$q_{\phi_{\mathbf{y}_2}}$ and all decoders in the disentangled step. For weakly-supervised setting, we exploit images and their weak labels like viewpoint $\mathbf{y}_2$ by training the VAE with $q_{\phi_{\hat{\mathbf{x}}}}$ and $p_{\theta_{\mathbf{y}_2}}$ in the embedding step.

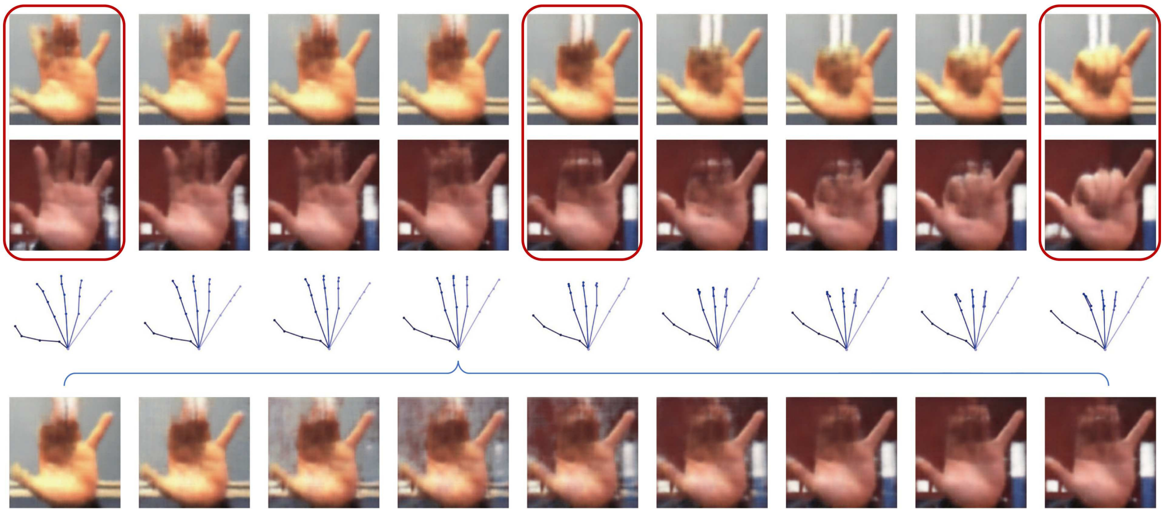


Figure 4.4: Latent space walk. The images in the red boxes are provided inputs. The first two rows show synthesized images when interpolating on the latent 3DPose space; the third row shows skeletons of the reconstructed 3DPose. The fourth row shows synthesized images when the pose is fixed (to the fourth column) when interpolating in the content latent space.

4.4 Experimentation

A good disentangled representation should show good performance on both discriminative tasks such as hand pose estimation as well as generative tasks. We transfer attributes between images and infer 3D hand poses from monocular hand RGB images via disentangled representations. More precisely, for image synthesis, we transfer image content with fixed 3DPose, while for 3D hand pose estimation, we predict viewpoint, CPose and 3DPose.

4.4.1 Implementation Details

Our architecture consists of multiple encoders and decoders. For encoding images, we use Resnet-18 [47]; for decoding images, we follow the decoder architecture DCGAN [98]. For encoding and decoding hand poses, we use six fully connected layers with 512 hidden units.

For learning, we use the ADAM optimizer with a learning rate of 10^{-4}, a batch size of 32. We fix the dimensionality of d of $\mathbf{z}$ to 64 and set the dimensionality of sub-latent variable $\mathbf{z}_{\mathbf{y}_1}$ and $\mathbf{z}_{\mathbf{y}_2}$ to 32 and 32. For all applications, the λ's are fixed ($\lambda_{\mathbf{x}} = 1, \lambda_{\mathbf{y}_1} = \lambda_{\mathbf{y}_2} = 0.01$) while we must adjust β ($\beta = 100$ for image synthesis, $\beta''' = 0.01$ for pose estimation).

4.4.2 Datasets & Evaluation

We evaluate our proposed method on two publicly available datasets: Stereo Hand Pose Tracking Benchmark (STB) [167] and Rendered Hand Pose Dataset (RHD) [178].

The **STB dataset** features videos of a single person's left hand in front of 6 real-world indoor backgrounds. It provides the 3D positions of palm and finger joints for approximately 18k stereo pairs with 640 × 480 resolution. Image synthesis is relatively easy for this dataset due to the small number of backgrounds. To evaluate our model's pose estimation accuracy, we use the 15k / 3k training/test split as given by [178]. For evaluating our dVAE's generative modelling capabilities, we disentangle $\mathbf{z}$ into two content and 3DPose according to the model in Fig. 4.2a synthesize images with fixed poses as per the left-most model in Fig. 4.3.

RHD is a synthesized dataset of rendered hand images with 320×320 resolution from 20 characters performing 39 actions with various hand sizes, viewpoints and backgrounds. The dataset is highly challenging due to the diverse visual scenery, illumination and noise. It is composed of 42k images for training and 2.7k images for testing.

For quantitative evaluation and comparison with other works on 3D hand pose estimation, we use the common metrics, mean end-point-error (EPE) and the area under the curve (AUC) on the percentage of correct keypoints (PCK) score. Mean EPE is defined as the average euclidean distance between predicted and groundtruth keypoints; PCK is the percentage of predicted keypoints that fall within some given distance with respect to the ground truth.

4.4.3 Synthesizing Images

We evaluate the ability of our model to synthesize images by sampling from latent space walks and by transferring pose from one image to another.

For the **fully specified latent z** model we show the synthesized images (see Fig. 4.4) when we interpolate the 3DPose while keeping the image content fixed (rows 1-3) and when we interpolate image content while keeping the pose fixed. In both latent space walks, the reconstructed poses as well as the synthesized images demonstrate a smoothness and consistency of the latent space.

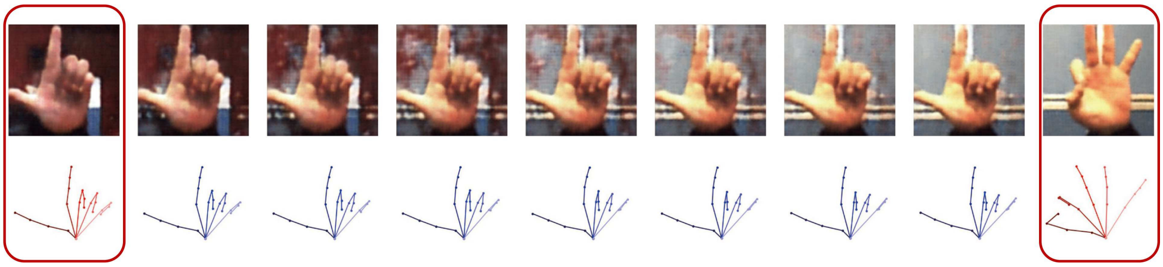


Figure 4.5: Latent space walk, interpolating $\mathbf{z_u}$ representing image background content. The images along with groundtruth 3DPose (red) in the red box are the input points; the first row shows generated images and the second row corresponding reconstructed 3DPose (blue). Note that because we are interpolating only on the background content, the pose stays well-fixed.

We can also extract disentangled latent factors from different hand images and then recombine them to transfer poses from one image to another. Fig. 4.6 shows the results when we take poses from one image (leftmost column), content from other images (top row) and recombine them (rows 2-3, columns 3-5). We are able to accurately transfer the hand poses while faithfully maintaining the tag content.

With additional $\mathbf{z_u}$ we also show interpolated results from a latent space walk on $\mathbf{z_u}$ in Fig. 4.5. In this case, the 3DPose stays well-fixed, while the content changes smoothly between the two input images, demonstrating our model's ability to disentangle the image background content even with out specific tag images for training.

4.4.4 3D Hand Pose Estimation

We evaluate the ability of our dVAE to estimate 3D hand poses from RGB images based on the model variant described in Section 4.3.3 and compare against state-of-the-art methods [11, 111, 178, 81, 90] on both the RHD and STB datasets. In [178], a two-stream architecture is

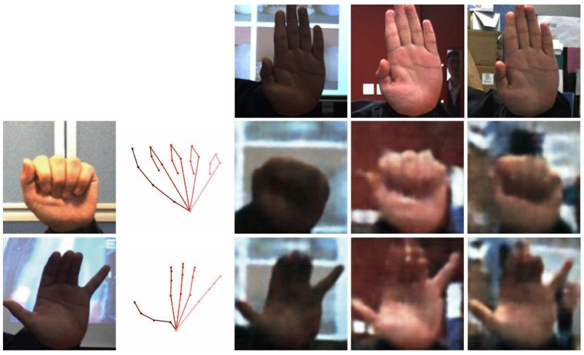
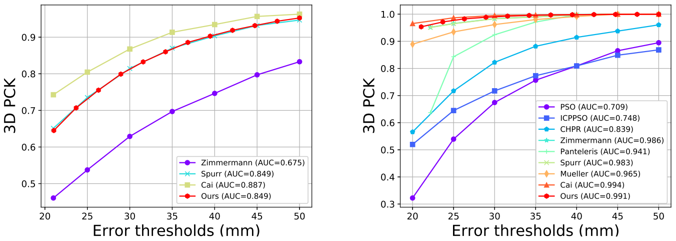


Figure 4.6: Pose transfer. The first column corresponds to images from which we extract the 3DPose (ground truth pose in second column); the first row corresponds to tag images columns we extract the latent content; the 2-3 rows, 3-5 columns are pose transferred images.



Method	RHD		STB	
	CPose	3DPose	CPose	3DPose
[178]	16.37	30.42	6.07	8.68
[111]	\	19.73	\	8.56
Ours	13.93	19.95	6.09	8.66

Figure 4.7: Quantitative evaluation. 3D PCK on RHD (left) and STB (middle). Mean EPE (mm) on RHD and STB (right).

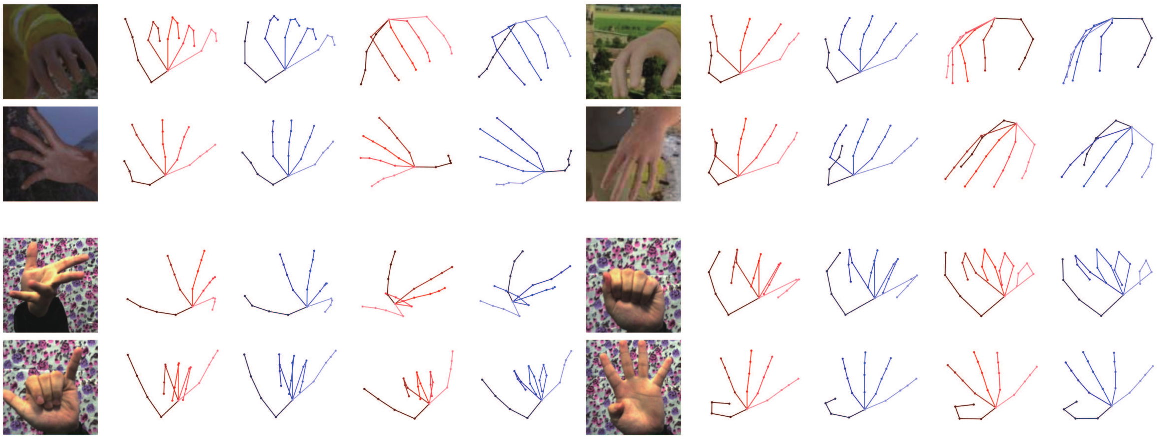


Figure 4.8: CPose and 3DPose estimation on RHD and STB. For each quintet, the left most column corresponds to the input images, the second and the third columns correspond to CPose groundtruth (red) and our prediction (blue), the right most two columns correspond to 3DPose groundtruth (red) and our prediction (blue).

applied to estimate viewpoint and CPose; these two are then combined to predict 3DPose. To be directly comparable, we disentangle the latent $\mathbf{z}$ into a viewpoint factor and a CPose factor, as shown in Fig. 4.3 right. Note that due to the decompositional nature of our latent space, we can predict viewpoint, CPose and 3DPose through one latent space.

We follow the experimental setting in [178, 111] that left vs right handedness and scale are given at test time. We augment the training data by rotating the images in the range of $[-180°, 180°]$ and making random flips along the y-axis while applying the same transformations to the ground truth labels. We compare the mean EPE in Fig. 4.7 right. We outperform [178] on both CPose and 3DPose. These results highlight the strong capabilities of our dVAE model for accurate hand pose estimation. Our mean EPE is very close to that of [111], while our 3D PCK is slightly better. As such, we conclude that the pose estimation capabilities of our model is comparable to that of [111], though our model is able to obtain a disentangled representation and make full use of weak labels. We compare the PCK curves with state-of-the-art methods [11, 111, 178, 81, 90] on both datasets in Fig. 4.7. Our method is comparable or better than most existing methods except [11], which has a higher AUC of

0.038 on RHD and 0.03 on STB for the PCK. However, these results are not directly comparable, as [11] incorporate depth images as an additional source of training data. Fig. 4.8 shows some our estimated hand poses from both RHD and STB datasets.

Semi-, weakly-supervised learning: To evaluate our method in semi- and weakly-supervised settings, we sample the first $m\%$ images as labelled data and the rest as unlabelled data by discarding the labels of 3DPose, CPose and viewpoint. We also consider using only viewpoints as a weak label while discarding 3DPose and CPose. For the RHD dataset, we vary $m\%$ from 5% to 100% and compare the mean EPE against the fully supervised setting. We can see that our model makes full use of additional information. With CPose, viewpoint and 3DPose labels, we improve the mean EPE up to 3.5%. With additional images and viewpoint labels, the improvement is up to 7.5%.

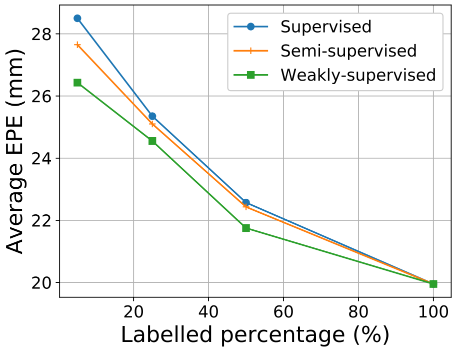


Figure 4.9: Mean EPE of our model on the semi-supervised setting and the weakly-supervised setting.

4.5 Conclusion

We presented a VAE-based method for learning disentangled representations of hand poses and hand images. We find that our model allows us to synthesize highly realistic looking RGB images of hands with full control over factors of variation such as image background content and hand pose.

As for the future work, the assumption that the factors of variation here should be labelled and independent is too strict and limits its application in the real world. We will consider to relax the need of labelled and independent between factors. Overall, we highlight the exploration of weak labels like domain labels, and dependent labels like group labels of shared factors. First, we highlight the exploration of weak labels like domain labels, and dependent labels like group labels of shared factors. Specifically, we will investigate disentangled representations with multimodal learning and contrastive learning. Multimodal data provide shared factors. In this case, multimodal data with shared image factors are more common and easy to provide ”positive image factors”. With the help of contrastive learning, we may encourage the features of ”positive image factors” as close as possible. Second, we target developing a common task-relevant representation learning framework based on disentangled representations for downstream tasks. Third, it is important to construct data pairs with shared factors of variation. We will exploit synthetic data to aid the training of disentangled representations. Based on the synthetic data, We will warm-up/pre-train the disentangled representation-based image generation network to imitate the image rendering process. Moreover, we will construct data pairs based on some ad-hoc record settings to get real-world images with shared factors of variation.

Chapter 5

Aligning Latent Spaces for 3D Hand Pose Estimation

Contents

5.1 Motivation . . . 59

5.2 Related Works . . . 60

5.3 Methodology . . . 61

5.3.1 Cross Modal VAE and Its Extension . . . 61

5.3.2 Latent Space Alignment . . . 62

5.3.3 Application Towards Hand Pose Estimation . . . 65

5.4 Implementation Details . . . 65

5.4.1 Data Pre-Processing and Augmentation . . . 65

5.4.2 Encoder and Decoder Modules . . . 66

5.5 Experimentation . . . 67

5.5.1 Datasets and Evaluation Metrics . . . 68

5.5.2 Qualitative Results . . . 69

5.5.3 RGB 3D Hand Pose Estimation . . . 69

5.6 Conclusion . . . 72

This chapter presents a VAE-based framework with multi-modal data as auxiliary information for hand pose estimation. Hand pose estimation from monocular RGB inputs is a highly challenging task. Many previous works for monocular settings only used RGB information for training despite the availability of corresponding data in other modalities such as depth maps. Note that real-world hand data usually comes with multiple modalities and hand modalities are representations of hands in different aspects. Therefore, we aim to use different modalities as auxiliary information for RGB input. With this target, we first formulate RGB-based hand pose estimation as a multi-modal learning, cross-modal inference problem, and then propose to align the latent spaces between RGB images and auxiliary modalities (*e.g.* , point clouds) to improve the representations of RGB hand. Specifically, based on VAE, we propose to learn a joint latent representation that leverages other modalities as prior knowledge during training to improve RGB-based hand pose estimation. We treat the training of joint latent representations as a distribution alignment. By design, our architecture is highly flexible in embedding various diverse modalities such as heat maps, depth maps and point clouds. We start from naive cross-modal learning, and further propose cross-modal learning with multiple decoders, distribution alignment with a KL divergence loss and distribution alignment with the product of Gaussian experts. We find distribution alignment with the product of Gaussian experts is flexible and can achieve the best performance. Besides the distribution alignment, we highlight that encoding and decoding the point cloud of the hand surface can improve the quality of the joint latent representations. Also, we introduce a technique, *i.e.* view correction, for point clouds and 3D poses, to alleviate the one-to-many mapping problems of 3D points for RGB input. Moreover, our framework can also be used in the weakly-supervised setting. For example, we can use surface point clouds as weak labels for unlabelled data to aid the training process and further improve the performance.

We conduct experiments on two public benchmarks, RHD and STB. First, we verify the performance of hand pose estimation with point clouds as prior knowledge. Experiments show that with the aid of other modalities during training, our proposed method boosts the accuracy of RGB-based hand pose estimation systems and significantly outperforms state-of-the-art methods. Second, we verify the proposed view correction. Experiments show the view correction boosts the accuracy of RGB-based hand pose estimation systems stably and reliably. Last, we evaluate the ability of our model to synthesize hand poses and point clouds. From two RGB images of the hand, we estimate the corresponding latent variables and then sample points by linearly interpolating between 3D hand pose and point cloud reconstructions of the interpolated points via our learned decoders. We observe that the learned latent space reconstructs a smooth and realistic transition between different poses, with changes in both global rotations and local finger configurations. This also verifies the alignment of the distribution. In the future, we may explore more alignment in pixel-level features by encouraging the corresponding features between different modalities to be close so that getting better representations. The publication, contributors and author contributions in this chapter are listed below:

Publication:

- Linlin Yang*, Shile Li*, Dongheui Lee and Angela Yao. "Aligning Latent Spaces for 3D Hand Pose Estimation." *International Conference on Computer Vision(ICCV).* 2019. * equal contribution.

Other Contributors:

- Shile Li (PhD student)
- Dongheui Lee (Supervisor of Shile Li)
- Angela Yao (Thesis Supervisor)

Contributions:

- Linlin Yang and Dr. Shile Li developed the method and wrote the code jointly, where Linlin Yang is more responsible for the algorithm and Dr. Shile Li is more responsible for the implementation. Linlin Yang and Dr. Shile Li wrote the main body of the article. Linlin Yang, Dr. Shile Li, Prof. Dr. Dongheui Lee and Prof. Dr. Angela Yao analyzed the results and revised the article.

5.1 Motivation

Hand pose estimation plays an important role in areas such as human activity analysis, human computer interaction, and robotics. Depth-based 3D hand pose estimation methods are now highly accurate [133, 69, 157] largely due to advancements from deep learning. Despite commodity depth sensors being more commonplace, high-quality depth maps can still only be captured indoors, thereby limiting the environments in which depth-based methods can be deployed. Furthermore, simple RGB cameras, as well as existing RGB footage are still far more ubiquitous than depth cameras and depth data. As such, there is still a need for accurate RGB-based 3D hand pose estimation methods, especially from monocular viewpoints.

To tackle the ambiguities associated with monocular RGB inputs, previous works have relied on large amounts of training data [178, 81]. Gains from purely increasing dataset size tend to saturate, because it is very difficult to obtain accurate ground truth labels, *i.e.* 3D hand poses. Annotating 3D hand joint positions accurately is a difficult task and there is often little consensus between human annotators [118]. While several methods have been developed to generate RGB images [81], there still exists a large domain gap between synthesized and real-world data, limiting the utility of synthetic data.

Even though accurate ground truth for RGB data is hard to collect, there exists plenty of unlabelled RGB-D hand data which can be leveraged together with labelled depth maps. Cai *et al.* [11] first proposed the use of labelled depth maps as regularizers to boost RGB-based methods. Yang *et al.* [156] introduced a disentangled representation so that viewpoint can be used as a weak label. Inspired by these works, we aim to leverage multiple modalities as weak labels for enhancing RGB-based hand pose estimation.

Here, we consider different modalities of hand data (*e.g.* RGB images, depth maps, point clouds, 3D poses, heatmaps and segmentation masks) and formulate RGB-based hand pose estimation as a cross-modal inference problem. In particular, we propose the use of a multi-modal variational autoencoder (VAE). VAEs are an attractive class of deep generative models which can be learned on large-scale, high-dimensional datasets. They have been shown to capture highly complex relationships across multiple modalities [119, 131, 144] and have also been applied to RGB-based pose estimation in the past [111, 156]. However, both [111] and [156] learn a single shared latent space and as a result must compromise on pose reconstruction accuracy.

In this work, we propose to align latent space from individual modalities. More specifically, we derive different objectives for three diverse modalities, namely 3D poses, point clouds, and heatmaps, and show two different ways to aligning their associated hand latent spaces. While such a solution may appear less elegant than learning one shared latent space directly, it is has several practical advantages. First and foremost, it is much faster to converge and results in a well-structured latent space; in comparison, the multimodal shared latent space of [111] tends to fluctuate as one draws data from the multiple modalities. Additionally, the learning scheme through alignment offers more flexibility in working with non-corresponding data and also weak supervision. The resulting latent representation allows for estimating highly accurate hand poses and synthesizing realistic-looking point clouds of the hand surface, all from monocular RGB images (See Fig. 5.1).

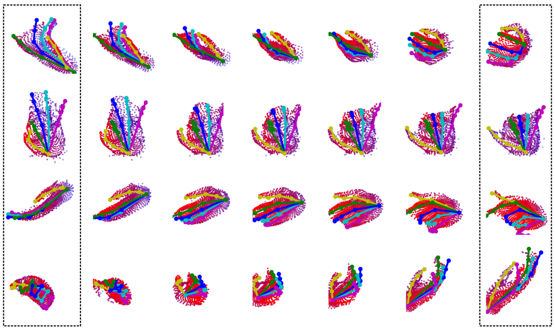


Figure 5.1: Latent space interpolation. The far left and far right columns (dashed boxes) are generated poses and point clouds from monocular RGB images sampled from the training data. Other columns are generated from linear interpolations on the latent space. The smoothness and consistency imply that different cross-modal latent spaces can be embedded and aligned into one shared latent space.

The main contributions are as follows:

- We formulate RGB-based hand pose estimation as a multi-modal learning, cross-modal inference problem and propose three strategies for learning from different hand inputs of various modalities.
- We explore non-conventional inputs such as point clouds and heatmaps for learning the latent hand space and show how they can be leveraged for improving the accuracy of an RGB-based hand pose estimation system. A side product of our framework is that we can synthesize realistic-looking point clouds of the hand from RGB images.
- By evaluating on two publicly available benchmarks, we show that our proposed framework makes full use of auxiliary modalities during training and boosts the accuracy of RGB pose estimates. Our estimated poses surpass state-of-the-art methods on monocular RGB-based hand pose estimation, including a whopping 19% improvement on the challenging RHD dataset [178]

5.2 Related Works

One way to categorize hand pose estimation approaches is according to either generative or discriminative methods. Generative methods employ a hand model and use optimization to fit the hand model to the observations [96, 87, 127]. They usually require a good initialization; otherwise they are susceptible to getting stuck in local minima. Discriminative methods learn a direct mapping from visual observations to hand poses [128, 156, 69, 86, 178, 11]. Thanks to large-scale annotated datasets [178, 164, 128], deep learning-based discriminative methods have shown very strong performance in the hand pose estimation task.

In particular, works using depth or 3D data as input are the most accurate. Oberweger *et al.* [86] use 2D CNNs to regress the hand pose from depth images, using a bottleneck layer to regularize the pose prediction to a certain prior distribution. Moon *et al.* [79] use 3D voxels as input and regress the hand pose with a 3D CNN. More recent works [69, 34] apply 3D point clouds as input and can estimate very accurate hand poses.

3D data is not always available either at training or at testing. Some recent works have started to explore the use of monocular RGB data. For example, Zimmermann *et al.* [178] regress heatmaps for each hand keypoint from RGB images and then regress the 3D hand pose from these heatmaps with fully-connected layers. Mueller *et al.* [81] follow a similar approach, but obtain the final 3D hand pose by using a kinematic skeleton model to fit the probability distribution of predicted heatmaps.

More recent monocular RGB-based methods leverage depth information for training [11, 111], even though testing is done exclusively with RGB images. Our proposed method also falls into this line of work. Cai *et al.* [11] propose an additional decoder to render depth maps from corresponding poses to regularize the learning of an RGB-based pose estimation system. This architecture is essentially two independent networks with a shared hand pose layer. This shared layer however cannot leverage data without pose annotations. Spurr *et al.* [111] propose a VAE-based method that learns a shared latent space for hand poses from both RGB and depth images.

However, its alternating training strategy from the different modalities ignores the availability of corresponding data and leads to a slow convergence speed.

5.3 Methodology

The aim of cross-modal methods is to capture relationships between different modalities so that it is possible to obtain information of target modalities given observations of some other modalities. In this section, we first present the cross modal VAE (CrossVAE) [89, 111] and our extensions to handle inputs and outputs from multiple modalities (Sec. 5.3.1). We then introduce two latent space alignment operators strategies (Sec. 5.3.2) and how they can be applied for RGB-based hand pose estimation (Sec. 5.3.3).

5.3.1 Cross Modal VAE and Its Extension

Given data sample $\mathbf{x}$ from some input modality, the cross modal VAE aims to estimate its corresponding target value $\mathbf{y}$ in a target modality by maximizing the evidence lower bound (ELBO) via a latent variable $\mathbf{z}$.

$$\begin{aligned}\log p(\mathbf{y}) &\geq \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x};\mathbf{y};\theta,\phi)\\ &= E_{\mathbf{z}\sim q_\phi}\log p_\theta(\mathbf{y}|\mathbf{z}) - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).\end{aligned} \tag{5.1}$$

Here, $D_{KL}(\cdot)$ is the Kullback-Leibler divergence. β is a hyperparameter introduced by [48] to balance latent space capacity and reconstruction accuracy. $p(\mathbf{z}) = \mathcal{N}(\mathbf{0},\mathbf{I})$ is a Gaussian prior on the latent variable $\mathbf{z}$. The variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ is an encoder from $\mathbf{x}$ to $\mathbf{z}$, and $p_\theta(\mathbf{y}|\mathbf{z})$ is a decoder or inference network from $\mathbf{z}$ to $\mathbf{y}$.

In addition to $\mathbf{x}$ and $\mathbf{y}$, we assume that there are corresponding data from N other modalities $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ and that these modalities are conditionally independent given latent representation $\mathbf{z}$. For clarity, we limit our derivation below to $N = 1$, though the theory generalizes to higher N as well. To encode these additional modalities, we can extend the ELBO from Eq. 5.1 as follow:

$$\begin{aligned} \log\ p(\mathbf{y}, \mathbf{w}_1) \geq\ & \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}, \mathbf{w}_1; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{x},\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}) \\ =& E_{\mathbf{z}\sim\phi_{\mathbf{x},\mathbf{w}_1}} \log p_{\theta_{\mathbf{y}}}(\mathbf{y}|\mathbf{z}) + \lambda_{\mathbf{w}_1} E_{\mathbf{z}\sim\phi_{\mathbf{x},\mathbf{w}_1}} \log p_{\theta_{\mathbf{w}_1}}(\mathbf{w}_1|\mathbf{z}) \\ -& \beta D_{KL}\left(q_{\phi_{\mathbf{x},\mathbf{w}_1}}(\mathbf{z}|\mathbf{x}, \mathbf{w}_1)||p(\mathbf{z})\right), \end{aligned} \tag{5.2}$$

where $\lambda_{\mathbf{w}_1}$ is a hyperparameter that regulates the reconstruction accuracy between $\mathbf{w}_1$ and $\mathbf{y}$. Graphical models of the original cross modal VAE and its extension to more modalities are shown in Fig 5.2a and Fig 5.2b.

We expect the $\mathbf{z}$ sampled from the variational approximation $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w}_1)$ in Eq. 5.2 to be more informative than the one sampled from $q_\phi(\mathbf{z}|\mathbf{x})$ in Eq. 5.1, since it is conditioned on both $\mathbf{z}$ and $\mathbf{w}_1$. Furthermore, the expectation term for the decoder $p_{\theta_{\mathbf{w}_1}}$ can be regarded as a regularizer that prevents the latent space from over-fitting to $\mathbf{y}$'s modality. From here onwards, ,we define $\mathbf{z}_{\text{joint}}$ as $\mathbf{z}$ from Eq. 5.2.

Note that Eq. 5.2 assumes that corresponding data from modalities $\mathbf{x}$, $\mathbf{w}_1$ are always available. While this is a reasonable assumption for training, *i.e.* having corresponding data samples from multiple modalities, this severely limits the applicability.

One possibility is to simplify the encoder to take only inputs from $\mathbf{x}$, so that Eq. 5.2 simplifies to $\mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1})$. The associated algorithm is shown in Alg. 5.1. Note that this reduces the richness of the latent space and thereby the decoding capabilities.

5.3.2 Latent Space Alignment

An alternative solution is to learn $q_{\phi_{\mathbf{x},\mathbf{w}_1}(\mathbf{z}|\mathbf{x},\mathbf{w}_1)}$ and $q_{\phi_{\mathbf{x}}(\mathbf{z}|\mathbf{x})}$ jointly and ensure that they correspond, *i.e.* are equivalent, by aligning the two distributions together. Note that equivalence between the two distributions follows naturally from our originally assumption that $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{w}_i$ are all conditionally independent given $\mathbf{z}$. Inspired by multimodal learning work of [5], we propose joint training objectives to align the latent spaces learned from single modalities to

Algorithm 5.1 Extended cross modal with one encoder.

```
Require: $\mathbf{x}, \mathbf{y}, \mathbf{w}_1, T$
Ensure: $\phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$
Initialize $\phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$
for $t = 1, \ldots, T$ epochs do
    Encode $\mathbf{x}$ to $q_{\phi_{\mathbf{x}}}(\mathbf{z_x}|\mathbf{x})$
    Decode $\mathbf{z_x}$ to $p_{\theta_{\mathbf{x}}}(\mathbf{y}|\mathbf{z_x}), p_{\theta_{\mathbf{w}_1}}(\mathbf{w}_1|\mathbf{z_x})$
    Update $\phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$ via gradient ascent of $\mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1})$
end for
```

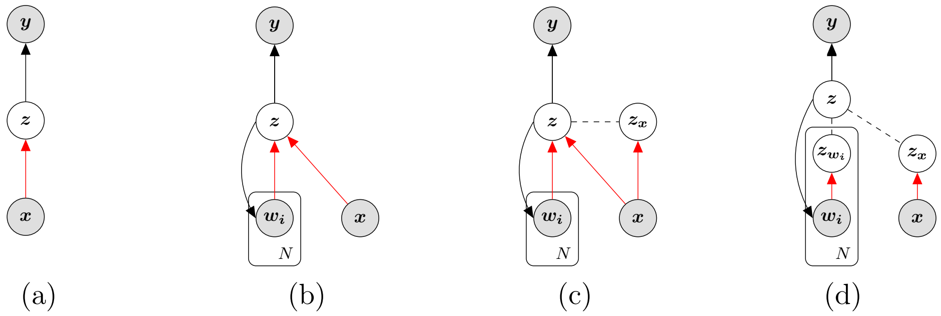


Figure 5.2: Graphical models. (a) Cross modal; (b) Extended cross modal; (c) Latent alignment with a KL divergence loss; (d) Latent alignment with the product of Gaussian experts. The shaded nodes represent observed variables while un-shaded nodes are latent. The red and black solid lines denote variational approximations q_ϕ or encoders, and the generative models p_θ or decoders respectively. The dashed lines denote the operation that embedding cross-modal latent spaces into a joint shared latent space; it is a KL divergence optimization for (c) and product of Gaussian experts for (d). Figure best viewed in colour.

the one learned with joint modalities to improve inference capabilities. More specifically, we would like to align $\mathbf{z}_\mathbf{x}$ (the latent representation learned only from $\mathbf{x}$), with the joint latent representation $\mathbf{z}_{\text{joint}}$ learned from both $\mathbf{x}$ and $\mathbf{w}$ so as to leverage the modalities of $\mathbf{w}$. One can also regard this as bringing together $q_{\phi_{\mathbf{x},\mathbf{w}_1}(\mathbf{z}|\mathbf{x},\mathbf{w}_1)}$ and $q_{\phi_\mathbf{x}(\mathbf{z}|\mathbf{x})}$ as close as possible.

KL divergence Loss. An intuitive way of aligning one latent space with another is to incorporate an additional loss term to reduce the divergence between $q_{\phi_{\mathbf{x},\mathbf{w}_1}(\mathbf{z}|\mathbf{x},\mathbf{w}_1)}$ and $q_{\phi_\mathbf{x}(\mathbf{z}|\mathbf{x})}$. This was first proposed by [119] for handling missing data from input modalities in multimodal setting. While we have no missing data in our cross-modal setting, we introduce a similar KL-divergence term D_{KL} with hyper-parameter β' to align the latent spaces.

$$\begin{aligned}\mathcal{L}(\phi_{\mathbf{x},\mathbf{w}_1},&\phi_\mathbf{x},\theta_\mathbf{y},\theta_{\mathbf{w}_1})\\ &= \text{ELBO}_{\text{cVAE}}(\mathbf{x},\mathbf{w}_1;\mathbf{y},\mathbf{w}_1;\phi_{\mathbf{x},\mathbf{w}_1},\theta_\mathbf{y},\theta_{\mathbf{w}_1})\\ &+ \text{ELBO}_{\text{cVAE}}(\mathbf{x};\mathbf{y},\mathbf{w}_1;\phi_\mathbf{x},\theta_\mathbf{y},\theta_{\mathbf{w}_1})\\ &- \beta' D_{KL}\left(q_{\phi_{\mathbf{x},\mathbf{w}_1}}(\mathbf{z}_{\text{joint}}|\mathbf{x},\mathbf{w}_1)||q_{\phi_\mathbf{x}}(\mathbf{z}_\mathbf{x}|\mathbf{x}\right)).\end{aligned} \tag{5.3}$$

Note that the decoders $\theta_\mathbf{y},\theta_{\mathbf{w}_1}$ are shared in the above ELBOs in Eq. 5.3. This implicitly forces $\mathbf{z}_{\text{joint}}$ and $\mathbf{z}_\mathbf{x}$ to be embedded to the same space (see Fig. 5.2c and Alg. 5.2).

The above formulation suffers from two major drawbacks on the encoding side. Firstly, as the number of modalities or N increases, the joint encoder $q_{\phi_{\mathbf{x},\mathbf{w}_1}}$ becomes difficult to learn. Secondly, with only the two encoders $q_{\phi_\mathbf{x}}$ and $q_{\phi_{\mathbf{x},\mathbf{w}_1}}$, we are not able to leverage data pairs ($\mathbf{w}_1$,$\mathbf{y}$). To overcome these weaknesses, we introduce the product of experts (PoE) as an alternative form of alignment.

Product of Gaussian Experts. It was proven in [144] that the joint posterior is proportional to the product of individual posteriors, *i.e.* $q(\mathbf{z}|\mathbf{x},\mathbf{w}_1) \propto p(\mathbf{z})q(\mathbf{z}|\mathbf{x})q(\mathbf{z}|\mathbf{w}_1)$. To that end, we can estimate the joint latent representation from unimodal latent representations.

Recall that in the formulation of the VAE, both $p(\mathbf{z})$ and $q(\mathbf{z}|\cdot)$ are Gaussian; as such, we arrive at $q(\mathbf{z}|\mathbf{x}, \mathbf{w}_1)$ through a simple product of Gaussian experts, $q(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{w}_1)$ [13, 144] (see model in Fig. 5.2d). With the help of shared decoders, we arrive at a joint latent representation through the following objective:

$$\begin{aligned}
&\mathcal{L}(\phi_{\mathbf{x}}, \phi_{\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}) = \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{x}}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}) \\
&+ \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{w}_1; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}) \\
&+ \mathrm{ELBO}_{\mathrm{cVAE}}(\mathbf{x}, \mathbf{w}_1; \mathbf{y}, \mathbf{w}_1; \phi_{\mathbf{x}}, \phi_{\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}) \\
&= E_{\mathbf{z_x} \sim q_{\phi_{\mathbf{x}}}} \log p_\theta(\mathbf{y}, \mathbf{w}_1 | \mathbf{z_x}) + E_{\mathbf{z}_{\mathbf{w}_1} \sim q_{\phi_{\mathbf{w}_1}}} \log p_\theta(\mathbf{y}, \mathbf{w}_1 | \mathbf{z}_{\mathbf{w}_1}) \\
&+ E_{\mathbf{z}_{\mathrm{joint}} \sim \mathrm{GProd}(\mathbf{z_x}, \mathbf{z}_{\mathbf{w}_1})} \log p_\theta(\mathbf{y}, \mathbf{w}_1 | \mathbf{z}_{\mathrm{joint}}) \\
&- \beta(D_{KL}\left(q_\phi(\mathbf{z_x}|\mathbf{x}) || p(\mathbf{z})\right) + D_{KL}\left(q_\phi(\mathbf{z}_{\mathbf{w}_1}|\mathbf{w}_1) || p(\mathbf{z})\right)),
\end{aligned} \tag{5.4}$$

where the GProd$(\cdot)$ is the product of Gaussian experts. Note in this formulation, we do not need a joint encoder $\phi_{\mathbf{x},\mathbf{w}_1}$ for $\mathbf{x}$ and $\mathbf{w}_1$ as was the case for alignment with KL divergence in Eq. 5.3. Instead, we use $q(\mathbf{z}|\mathbf{x})$ and $q(\mathbf{z}|\mathbf{w}_1)$ as two Gaussian experts. Suppose that $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_1, \Sigma_1)$ and $q(\mathbf{z}|\mathbf{w}_1) = \mathcal{N}(\mu_2, \Sigma_2)$. The product of two Gaussian experts is also Gaussian with mean μ and covariance Σ, where

$$\begin{aligned}
&\mu = (\mu_1 T_1 + \mu_2 T_2)/(T_1 + T_2), \qquad \text{and} \\
&\sigma = 1/(T_1 + T_2), \quad \text{where } T_1 = 1/\Sigma_1, T_2 = 1/\Sigma_2.
\end{aligned} \tag{5.5}$$

All operations in the product of Gaussian experts are element-wise. In this way, we can build a connection between $\mathbf{z}_{\mathrm{joint}}$ and $\mathbf{z_x}, \mathbf{z}_{\mathbf{w}_1}$, forcing them all into one shared latent space. This alignment strategy is more flexible than Alg. 5.2, because the encoders of different modalities can be trained individually, even from different datasets, while for Alg. 5.2, the joint encoder must be trained on the complete $\mathbf{x}, \mathbf{w_1}$ pairs. The learning algorithm can be found in Alg. 5.3.

Algorithm 5.2 Latent alignment with Eq. 5.3.

```
Require: $\mathbf{x}, \mathbf{y}, \mathbf{w}_1, T$
Ensure: $\phi_{\mathbf{x}}, \phi_{\mathbf{x},\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$
Initialize $\phi_{\mathbf{x}}, \phi_{\mathbf{x},\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$
for $t = 1, \ldots, T$ epochs do
    Encode $\mathbf{x}$ to $q_{\phi_{\mathbf{x}}}(\mathbf{z_x}|\mathbf{x})$
    Encode $\mathbf{x}, \mathbf{w}_1$ to $q_{\phi_{\mathbf{x},\mathbf{w}_1}}(\mathbf{z}_{\mathrm{joint}}|\mathbf{x}, \mathbf{w}_1)$
    Decode $\mathbf{z_x}$ to $p_{\theta_{\mathbf{x}}}(\mathbf{y}|\mathbf{z_x}), p_{\theta_{\mathbf{w}_1}}(\mathbf{w}_1|\mathbf{z_x})$
    Decode $\mathbf{z}_{\mathrm{joint}}$ to $p_{\theta_{\mathbf{x}}}(\mathbf{y}|\mathbf{z}_{\mathrm{joint}}), p_{\theta_{\mathbf{w}_1}}(\mathbf{w}_1|\mathbf{z}_{\mathrm{joint}})$
    Construct $D_{KL}(q_{\phi_{\mathbf{x},\mathbf{w}_1}}(\mathbf{z}_{\mathrm{joint}}|\mathbf{x}, \mathbf{w}_1)||q_{\phi_{\mathbf{x}}}(\mathbf{z_x}|\mathbf{x}))$
    Update $\phi_{\mathbf{x}}, \phi_{\mathbf{x},\mathbf{w}_1}, \theta_{\mathbf{y}}, \theta_{\mathbf{w}_1}$ via gradient ascent of Eq. 5.3
end for
```

5.3.3 Application Towards Hand Pose Estimation

In the context of RGB-based hand pose estimation, $\mathbf{x}$ represents RGB images and $\mathbf{y}$ 3D hand poses. Other modalities like heatmaps, depth maps, point clouds and segmentation masks can be used as $\mathbf{w}$ during training to improve the learning of the latent space and thereby leading to more accurate hand pose estimates from RGB inputs. Here, we use point clouds (C) and heatmaps (H) as additional modalities $\mathbf{w}$ to improve the cross modal inference of RGB (R) to 3D poses (P). In the rest, we use the format "A2B" to represent the estimation of target modality "B" from input modality "A" during training. For example, R2CHP represents the estimation of point clouds, heatmaps and 3D poses from RGB input. Note that unless indicated otherwise, the test settings use RGB images as the source modality or input and 3D hand poses as the target modality or output.

5.4 Implementation Details

5.4.1 Data Pre-Processing and Augmentation

From the RGB image, the region containing hand is cropped from ground truth masks and resized to 256×256. The corresponding region in the depth image is converted to point clouds using the provided camera intrinsic parameters. For each training step, a different set of 256 points are randomly sampled as training input.

Viewpoint correction. After cropping the hand from the RGB image, the center of the hand in the image moves from some arbitrary coordinates to the center of the image. As such, the 3D hand pose and associated point cloud must be rotated such that the viewing angle towards the hand aligns with the optical axis. As indicated in [69], this correction is necessary to remove the many-to-one observation-pose pairings. We follow the approach given in [69].

Data augmentation was performed online during training. The images are scaled randomly between $[1, 1.2]$, translated $[-20, 20]$ pixels and rotated $[-\pi, \pi]$ around the camera view axis. Furthermore, the hue of the image is randomly adjusted by [-0.1, 0.1]. The point clouds are rotated randomly around the camera view axis and the 3D pose labels are also

Algorithm 5.3 Latent alignment with Eq. 5.4.

```
Require: $\mathbf{x}, \mathbf{y}, \mathbf{w}_1, T$
Ensure: $\phi_\mathbf{x}, \phi_{\mathbf{w}_1}, \theta_\mathbf{y}, \theta_{\mathbf{w}_1}$
Initialize $\phi_\mathbf{x}, \phi_{\mathbf{w}_1}, \theta_\mathbf{y}, \theta_{\mathbf{w}_1}$
for $t = 1, \ldots, T$ epochs do
    Encode $\mathbf{x}$ to $q_{\phi_\mathbf{x}}(\mathbf{z}_\mathbf{x}|\mathbf{x})$
    Encode $\mathbf{w}_1$ to $q_{\phi_{\mathbf{w}_1}}(\mathbf{z}_{\mathbf{w}_1}|\mathbf{w}_1)$
    Construct $\mathbf{z}_\text{joint} = \text{GProd}(\mathbf{z}_\mathbf{x}, \mathbf{z}_{\mathbf{w}_1})$
    Decode $\mathbf{z}_\mathbf{x}, \mathbf{z}_{\mathbf{w}_1}, \mathbf{z}_\text{joint}$ to $p_{\theta_\mathbf{x}}(\mathbf{y}|\cdot), p_{\theta_{\mathbf{w}_1}}(\mathbf{w}_1|\cdot)$ respectively
    Update $\phi_\mathbf{x}, \phi_{\mathbf{w}_1}, \theta_\mathbf{y}, \theta_{\mathbf{w}_1}$ via gradient ascent of Eq. 5.4
end for
```

rotated accordingly.

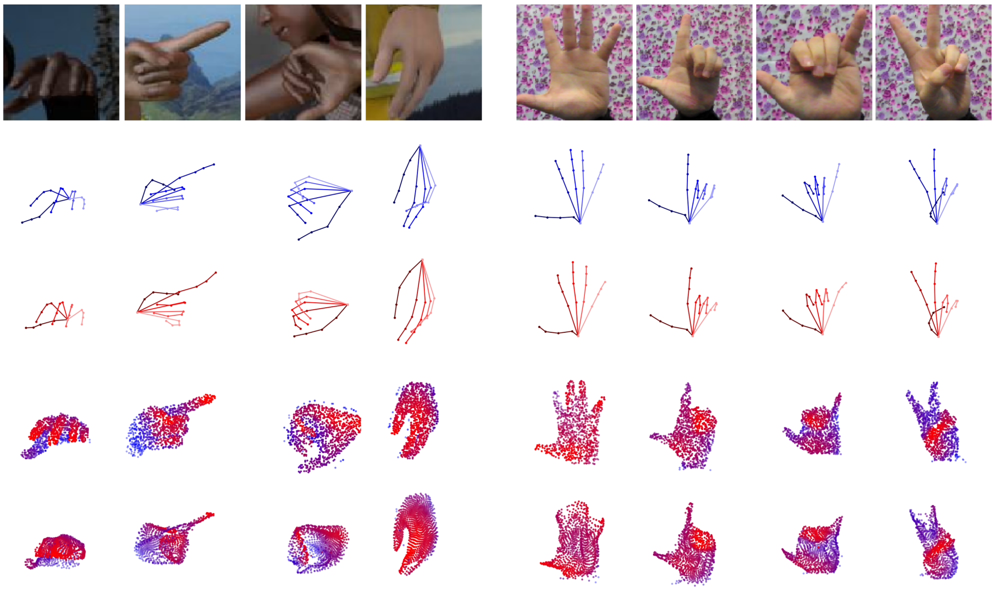


Figure 5.3: 3D pose estimation and point cloud reconstruction for RHD (left) and STB (right) dataset. From top to bottom: RGB images, ground-truth poses in blue, estimated poses from $\mathbf{z}_{\text{rgb}}$ in red, ground-truth point clouds, reconstructed point clouds from $\mathbf{z}_{\text{rgb}}$. The color for point clouds decodes the depth information, closer points are more red and further points are more blue. Note that the ground-truth point clouds are not used for inference, it is shown here only for comparison purpose.

5.4.2 Encoder and Decoder Modules

Our proposed method is highly flexible and can integrate many different modalities to construct a common latent space. In the current work, we learn encoders for RGB images and point clouds and decoders for 3D hand poses, point clouds and heatmaps of the 2D hand key points on the RGB image. We choose to convert the 2.5D depth information as 3D point clouds instead of standard depth maps, due to its superior performance in hand pose estimation, as shown in previous works [69, 19, 38]. Heatmaps are chosen as a third modality for decoding to encourage convergence of the RGB encoder, since the heatmaps are closely related to activation areas on the RGB images.

For encoding RGB images, we use Resnet-18 from [47] and two additional fully connected layers to predict the mean and variance vector of the latent variable. For encoding point clouds, we employ the ResPEL network [69], which is an learning architecture that takes unordered point cloud as input. While we use same number of PEL layers as in [69], the number of hidden units are reduced by half to ease the computational load.

To decode the heatmaps, we follow the decoder architecture of the DC-GAN [98]. The loss function used for the heatmaps is the L2 loss function of pixel-wise difference between prediction and ground-truth:

$$\mathcal{L}_{\text{heat}} = \sum_{j=1}^{J} ||\hat{H}_j - H_j||, \tag{5.6}$$

whereas H_j is the ground-truth heatmap for the j-th hand keypoint and $\hat{H}_j$ is the prediction. For decoding point clouds, we follow the FoldingNet architecture [157] and try to reconstruct a point cloud representing the visible surface of the hand. To learn the decoder, we use two different loss terms based on the Chamfer distance and Earth Mover's distance (EMD). The Chamfer distance is the sum of the Euclidean distance between points from one set and its closest point in the other set and vice versa:

$$\mathcal{L}_{\text{Chamfer}} = \frac{1}{|P|} \sum_{p \in P} \min_{\hat{p} \in \hat{P}} ||\hat{p} - p|| + \frac{1}{|\hat{P}|} \sum_{\hat{p} \in \hat{P}} \min_{p \in P} ||\hat{p} - p||. \tag{5.7}$$

For the Earth Mover's distance, one-to-one bijective correspondences are established between two point clouds, and the Euclidean distances between them are summed:

$$\mathcal{L}_{\text{EMD}} = \min_{\phi: P \to \hat{P}} \frac{1}{|P|} \sum_{p \in P} ||p - \phi(p)||, \tag{5.8}$$

In both Eq. 5.7 and 5.8, $\hat{P}, P \in \mathbb{R}^3$ represent the predicted point clouds and the ground truth point clouds respectively and the number of points in both clouds are 256.

The decoder for 3D pose consists of 4 fully-connected layers with 128 hidden units for each layer. To learn the pose decoder, we use an L2 loss:

$$\mathcal{L}_{\text{pose}} = ||\hat{y} - y||, \tag{5.9}$$

where $\hat{y}, y$ are the predicted and the ground truth hand poses describing the 3D locations of 21 keypoints.

Combining all the losses in Eq. 5.6-5.9, we obtain the following reconstruction loss function:

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{pose}} + \lambda_{\text{heat}} \mathcal{L}_{\text{heat}} + \lambda_{\text{cloud}} (\mathcal{L}_{\text{Chamfer}} + \mathcal{L}_{\text{EMD}}). \tag{5.10}$$

The overall loss for training is the sum of reconstruction loss and its corresponding D_{KL} loss based on Eq. 5.2-5.4.

5.5 Experimentation

In the experiments, we set the dimensionality of latent variable $\mathbf{z}$ to 64, λ_{heat} to 0.01, λ_{cloud} to 1 for all cases and β' to 1 for Eq. 5.3 . Our method is implemented with Tensorflow. For learning, we use an Adam optimizer with an initial learning rate of 10^{-4} and a batch size of 32. We lower the learning rate by a factor of 10 two times after convergence. The value of β is annealed from 10^{-5} to 10^{-3}.

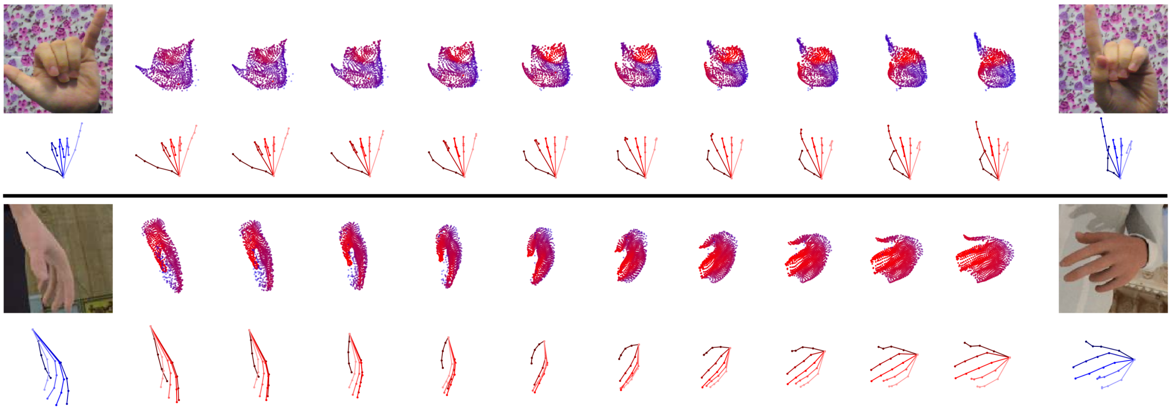


Figure 5.4: Latent space interpolation. Two examples of reconstructing point clouds and hand poses from the latent space. The most left and most right column are RGB images and their corresponding ground-truth poses. Other columns are generated point clouds and poses when interpolating linearly on the latent space.

5.5.1 Datasets and Evaluation Metrics

Our method is evaluated on two publicly available datasets: the Rendered Hand Pose Dataset (RHD) [178] and the Stereo Hand Pose Tracking Benchmark (STB) [167].

RHD is a synthesized dataset of rendered hand images with 320×320 resolution from 20 characters performing 39 actions. It is composed of 41238 samples for training and 2728 samples for testing. For each RGB image, a corresponding depth map, segmentation mask, and 3D hand pose are provided. The dataset is highly challenging because of the diverse visual scenery, illumination, and noise.

STB contains videos of a single person's left hand in front of six different real-world backgrounds. The dataset provides stereo images, color-depth pairs with 640×480 resolution and 3D hand pose annotations. Each of the 12 sequences in the dataset contains 1500 frames. To make the 3D pose annotations consistent for RHD, we follow [178, 11] and modify the palm joint in STB to the wrist point. Similar to [178, 11, 111, 156], we use 10 sequences for training and the other 2 for testing.

To evaluate the accuracy of the estimated hand poses, we use the common metrics mean end-point-error (EPE) and area under the curve (AUC) on the percentage of correct keypoints (PCK) curve. EPE is measured as the average Euclidean distance between predicted and ground-truth hand joints, whereas AUC represents the percentage of predicted keypoints that fall within certain error thresholds compared with ground-truth poses. To compare with the state-of-the-art methods in a fair way, we follow the similar condition used in [111, 50, 11, 156] to assume that the global hand scale and the hand root position are known in the experimental evaluations, where we set the middle finger's base position as the root of the hand.

Strategy	Encoder	Decoder	Mean EPE [mm]
S1 (Eq. 5.1)	R	P	16.61
S2 (Alg. 5.1)	R	H+P	16.10
	R	C+P	15.91
	R	C+H+P	15.49
S3 (Alg. 5.2)	R+C	C+H+P	14.93
S4 (Alg. 5.3)	R+C	C+H+P	**13.14**

Table 5.1: Comparison of different training strategies on the RHD dataset. The mean EPE values are obtained from monocular RGB images. (R: RGB, C: point cloud, P: pose, H: heatmap). Poses estimated from monocular RGB images can be improved by increasing number of different encoders and decoders during training.

5.5.2 Qualitative Results

Using the flexible design of our method, we train the networks exploiting all the available modalities and test using only limited modalities. In Fig. 5.3, we show some qualitative examples of poses and point clouds decoded from the $\mathbf{z}_{\text{rgb}}$. The 3D poses and point clouds can be successfully reconstructed from the same latent variable $\mathbf{z}$. The reconstructed point clouds' surfaces are smoother than the original inputs, since the inputs are sub-sampled from raw sensor data, while the reconstructed point clouds hold some structured properties from the FoldingNet decoder.

We also evaluate the ability of our model to synthesize hand poses and point clouds. From two RGB images of the hand, we estimate the corresponding latent variables $\mathbf{z}_{1,2}$ and then sample points by linearly interpolating between the two. 3D hand pose and point cloud reconstructions of the interpolated points via our learned decoders are shown in Fig. 5.4. We observe that the learned latent space reconstructs a smooth and realistic transition between different poses, with changes in both global rotations and local finger configurations.

5.5.3 RGB 3D Hand Pose Estimation

Note that even though our network is trained with multiple modalities, the results provided here are based only in monocular RGB inputs.

Training Strategy. We first compare different training strategies (S) in Table 5.1: S1. Baseline method to only use RGB-pose pairs for training. S2. Training with extended decoders, where the latent variables $\mathbf{z}_{\text{rgb}}$ reconstruct more modalities (heatmaps and point clouds) besides poses. S3. Training with an additional encoder for point clouds, where the different latent variables are aligned as per Alg. 5.2. S4. The alignment method in S3 is changed to the product of Gaussian experts (Alg. 5.3). More comparison results with AUC metric are shown in Fig. 5.5

Comparing S1 to the other strategies, we observe that the baseline performance can be improved by training with increasing number of additional encoders or decoders. Comparing

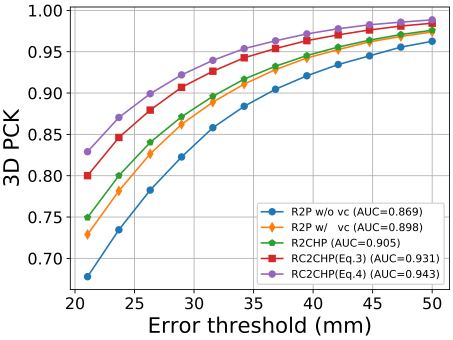


Figure 5.5: Comparisons of 3D PCK results of our different strategies on RHD dataset. The abbreviations can be found in Sec. 5.3.3 and "vc" stands for "view correction"

S4 to S3, the alignment with the Gaussian product outperforms the intuitive KL-divergence alignment method by capturing a better joint posterior of different input modalities.

Furthermore, we emphasize the necessity of viewpoint correction (Sec. 5.4.1). We applied both view corrected and uncorrected data for training the baseline strategy "R2P" (S1). The difference can be seen from Fig. 5.5, where the view corrected data clearly improves the AUC metric.

	Method	RHD	STB
VAE-based	Spurr *et al.* [111]	19.73	8.56
	Yang *et al.* [156]	19.95	8.66
	Ours	**13.14**	**7.05**
Others	Z&B [178]	30.42	8.68
	Iqbal *et al.* [50]	13.41	\

Table 5.2: Comparison to state-of-the-art on the RHD and STB with mean EPE [mm]. Ours refers to S4 in Table 1 (RC2CHP).

Comparison to state-of-the-art. In Table 5.2, we compare the EPE of our method with VAE-based methods [111, 156] which are most related to our method as well as other state-of-the-art [178, 50]. On both datasets, our proposed method achieves the best results, including an impressive 1.61mm or 19% improvement on the STB dataset.

We also compare the PCK curve of our approach with other state-of-the-art methods [111, 156, 178, 50, 81, 90] in Fig. 5.6 and Fig. 5.7. For both datasets, our method achieves the highest AUC value on the 3D PCK. We marginally outperform the state-of-the-art [50, 11] on the STB dataset, whereas on the RHD dataset, we surpass all reported methods to date [178, 156, 11, 111] with a significant margin. We note, however, that the STB dataset contains much less variation in hand poses and backgrounds than the RHD dataset and that performance by state-of-the-art methods on STB has become saturated. As such, there is little room for improvement on STB, whereas the benefits of our method is more visible on

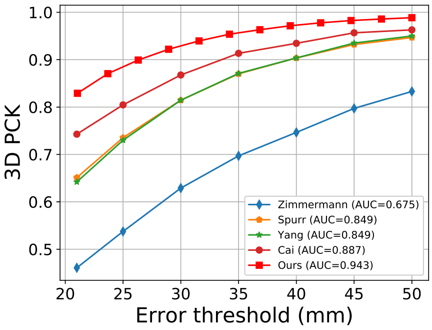


Figure 5.6: AUC: Comparison to state-of-the-art methods on the RHD dataset. Ours refers to S4 in Table 1 (RC2CHP).

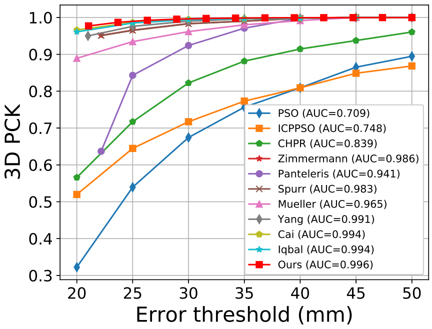


Figure 5.7: AUC: Comparison to state-of-the-art methods on the STB dataset. Ours refers to S4 in Table 1 (RC2CHP).

the RHD dataset.

Weakly-supervised learning. Thanks to flexibility of the proposed method, (surface) point clouds can be also used as “weak” labels for unlabelled data to aid the training process. We tested our method under a weakly-supervised setting on the RHD dataset, where we sample the first m% samples as labelled data (including RGB, point clouds and 3D poses) and the rest as unlabelled data (including RGB, point clouds) by discarding 3D pose labels. We compare the supervised setting with the weakly-supervised setting for the “RC2CHP” networks (S4 in Table 5.1). In the supervised training setting, we train the networks with only m% samples, In the weakly-supervised setting, besides fully supervised training on m% data, we also train the “RC2C” sub-parts with the rest (100-m)% samples simultaneously. The percentage of labelled data is varied from 5% to 100% to compare the mean EPE between supervised and weakly-supervised settings. From Fig. 5.8 we can see that our method makes full usage of additional unlabelled information, where the improvement is up to 6%.

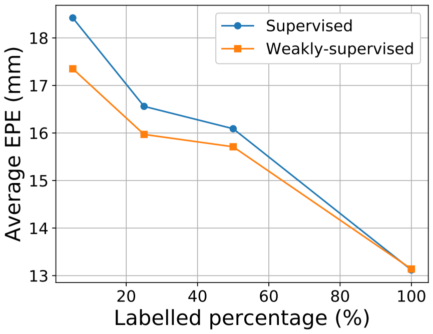


Figure 5.8: Mean EPE of our model on the weakly-supervised setting. Our method makes full use of unlabelled data, as the weakly-supervised setting performs almost as well as the supervised one.

5.6 Conclusion

We formulate RGB-based hand pose estimation as a multimodal learning and cross-modal inference problem. We derive different objectives for three hand modalities, and show different ways of aligning their associated latent spaces with a joint one. We highlight the flexibility of this framework as it can take arbitrary data pair for training. We will continue to explore the network that learns from multimodal data but is applicable during inference to one specific inputs. This is an efficient and effective way to use multimodal data. However, the alignment in this work is in the latent space, which neglects the spatial information in pixel-level features. We may explore more alignment in pixel-level features by encouraging the corresponding features between different modalities be close so that getting better representations. Note that this is applicable for 2D and 3D modalities, as we can connect 3D points with 2D pixels using the camera projection. Moreover, there is a strong connection between our proposed alignment strategy and knowledge distillation because both strategies require pre-training and imitation. We will encourage the two strategies to learn from each other and propose a unified framework for the alignment.

Chapter 6

SemiHand: Semi-supervised Hand Pose Estimation with Consistency

Contents

6.1 Motivation **75**

6.2 Related Works **77**

6.2.1 3D Hand Pose Estimation 77

6.2.2 Semi-Supervised Learning 77

6.3 Methodology **78**

6.3.1 Supervised Pose Estimation 79

6.3.2 Pseudo-labels for Pose Estimation 79

6.3.3 Self Consistency for Pose Estimation 80

6.3.4 Data Augmentation 83

6.4 Experiments **83**

6.4.1 Implementation Details 83

6.4.2 Datasets and Evaluation Metrics 83

6.4.3 Ablation Study 84

6.4.4 Comparison to State-of-the-Art 85

6.4.5 Comparison to Weakly-Supervised Methods 86

6.5 Conclusions **88**

With synthetic data as auxiliary information, we aim to reduce the burden of annotation for real-world data. This chapter targets a new hand pose estimation setting, *i.e.* , the cross-domain pose estimation, learning from labelled synthetic data and unlabelled real-world RGB data, for application on real-world RGB data. Due to the large domain gap between synthetic data and real-world data, training models with synthetic data generalize poorly to real-world settings. To address the cross-domain pose estimation problem, we introduce SemiHand, a framework that considers pseudo-labelling and consistency training for semi-supervised hand pose estimation. Pseudo-labelling and consistency training are already established in semi-supervised classification. However, extending such concepts for a regression task and in the context of 3D pose estimation is non-trivial. We show their difference in the following. First, with perturbations like noise, rotation, translation and flip, the labels of classification remain the same. In contrast, for 3D hand pose estimation, labels change accordingly based on perturbation. For example, when rotating the RGB images, the keypoint location in the image coordinate also changes. Second, classification is a single modality task while 3D pose estimation may consider different modalities and adopt multi-task frameworks. Different modalities should be consistent. Third, classification takes one-hot labels as ground-truth labels while 3D hand pose estimation aims to predict 3D poses. 3D poses as continuous labels should be biomechanically feasible. Last, for label correction, threshold or sharpening for classification are used to revise the one-hot labels. For 3D poses, we should correct poses based on their biomechanical feasibility. Inspired by classification, we highlight their difference and transfer pseudo-labelling and consistency training to 3D hand pose estimation accordingly.

Specifically, for perturbation and different modalities, we propose view consistency and cross-modal consistency for 3D pose estimation to encourage the predictions to be consistent with perturbations and auxiliary modalities. For label correction, we propose a template-based label correction module to refine the pseudo-labels for real-world data and encourage the model to be robust to noisy label outputs. For pseudo-labelling, we use consistency loss as the confidence to select confident pseudo-labels and then revise the infeasible pseudo-labels, to avoid confirmation bias. Moreover, to improve the stability of the fine-tuning, we propose a self-paced strategy and gradually take the refined predictions from weakly augmented input to supervise the predictions from strongly augmented input. Here, weak augmentations refer to small rotations and translations, while strong augmentation includes larger rotations and translations as well as image scaling. To evaluate our method, we pre-train the model on one synthetic dataset, RHD and then fine-tune it with only the training data of a (single) real-world dataset's training partition. The test data is withheld completely and we use the labels of these test data only for evaluation purposes. We evaluate our method on four real-world datasets. With our pseudo-labelling, consistency training and self-pace training strategy, we achieve significant improvement on real-world data. Moreover, we compare our method with existing weakly-supervised methods. Without any labels, our SemiHand achieves a similar improvement, demonstrating the effectiveness of our method, compared to existing weakly-supervised methods. In the future, we aim to remove even the requirement of unlabelled data. We would like to explore the techniques like domain randomization and hand image renderer, and make the model purely trained on synthetic data achieve a satisfactory result in the real world. The publication, contributors and author contributions in this chapter are listed below:

Publication:

- Linlin Yang, Shicheng Chen and Angela Yao. "SemiHand: Semi-supervised Hand Pose Estimation with Consistency." *International Conference on Computer Vision(ICCV).* 2021.

Other Contributors:

- Shicheng Chen (Master Student)
- Angela Yao (Thesis Supervisor)

Contributions:

- Linlin Yang proposed the framework, wrote the code and conducted the experiments. Linlin Yang and Shicheng Chen were responsible for the label correction module, including the implementation and experiments to verify the effectiveness of label correction. Linlin Yang and Prof. Dr. Angela Yao developed the idea, analyzed the results and wrote the main body of the article.

6.1 Motivation

A key challenge of monocular 3D hand pose estimation is getting sufficient high-quality ground-truth poses. Labelling real-world data to an accurate enough degree often requires dedicated interfaces and or multi-view camera rigs. This makes it non-trivial to gather "in-the-wild" data that is much sought-after for actual application deployment.

Synthesizing training data is considered an easy alternative to get accurate labels and has been incorporated into many learning-based frameworks. Yet there exists a significant domain gap between synthetic and real-world images so the performance of models trained on synthetic data deteriorates significantly when applied to real-world data. The favoured approach for reducing the domain gap is a mix-and-train strategy [50], *i.e.* mixing multiple real-world datasets together with synthetic data for training. Such a strategy depends largely however on the quantity and quality of the labelled samples in the combined datasets.

What if we tried to learn only from labelled synthetic data and fully unlabelled real-world data? We target exactly this scenario and present the first framework for domain-separated semi-supervised learning for 3D hand pose estimation. A classic approach in semi-supervised learning is to generate pseudo-labels [65] for the unlabelled data, usually via a classifier learned from the labelled portion of the data [65, 107]. The utility of pseudo-labels is highly variable. Used naively, these labels are even detrimental to learning because of confirmation bias [2], *i.e.* , the classifier over-fits to the pseudo-labels which tend to be noisy and or inaccurate, so additional corrections are necessary [2, 44, 174, 158]. Additionally, consistency training with unlabelled data [107, 2, 150] can increase the reliability of pseudo-labels.

We integrate these concepts and introduce SemiHand, a framework that considers spatial consistency and biomechanical feasibility for semi-supervised hand pose estimation. We propose two consistency losses to encourage the predictions to be consistent with perturbations and other modalities. As our labelled and unlabelled data come from different domains, *i.e.* synthetic vs real RGB images, there is the added challenge of domain adaptation to the unlabelled data. To bridge the domain gap, we propose a cross-modal consistency and leverage semantic predictions [73] from an auxiliary task to provide guidance for the predicted poses. Meanwhile, we regard predictions on real-world data as noisy labels; further training

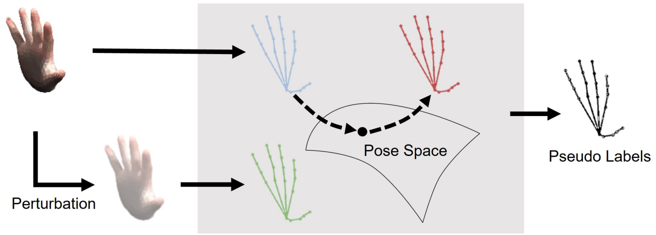


Figure 6.1: Pseudo-labelling of SemiHand. Our pseudo-label with confidence is generated based on the prediction from original (blue pose), the prediction from perturbation (green pose) and the corrected prediction (red pose).

the network from these predictions directly may actually be detrimental due to their inaccuracy. To mitigate the impact of this confirmation bias, we introduce label correction and sample selection based on the feasibility so that we train with only corrected pseudo-labels with high-confidence. We show our pseudo-labelling strategy in Fig. 6.1.

Pseudo-labelling and consistency training are already established in semi-supervised classification [65, 107, 2]. However, extending such concepts for a regression task and in the context of 3D pose estimation is non-trivial and we are the first to present a unified framework to do so. For example, existing methods [44, 139] primarily learn a noise transition matrix to correct pseudo-labels; such an approach is not applicable for regression and we instead focus on the confidence and feasibility of poses as a selection and correction criteria. Similarly, consistency training in classification simply keeps the predicted categories unchanged under perturbation. Consistency in 3D pose estimation however needs to account for the change in label, *i.e.* the pose after perturbation. We summarize our contributions below:

- We propose a novel RGB-based hand pose estimation framework using labelled synthetic data and unlabelled real-world data; it is the first semi-supervised framework that combines pseudo-labeling with consistency training for RGB-based hand pose.

- Based on the feasibility of hand poses, we propose a method for pose registration and sample selection to correct noisy label outputs and select pseudo-labels of high confidence for training.

- We propose two consistency losses for 3D pose estimation to encourage the predictions to be consistent with perturbations and auxiliary modalities.

- Using a pre-trained synthetic model, we are able to adapt our model to challenging real-world datasets without any labels. Our results are compelling when compared to fully supervised frameworks and outperform previous works on synthetic image enhancement.

6.2 Related Works

6.2.1 3D Hand Pose Estimation

Most recent methods apply deep learning and propose dedicated network architectures and or training strategies, *e.g.* voxel-to-voxel predictions [79], point-to-point regression [38, 69], and pixel-wise estimations [30, 50]. Other works like [29] propose a tree-like network structure to capture the hand's topology. As for training strategies, existing works are diverse and have explored multitask learning [11, 9, 169], multi-view constraints for self-supervision [132, 135], and biomechanical constraints [110] as regularization. In RGB-based hand pose estimation, datasets are still relatively small and highly variable from each other. As such, most approaches cannot generalize to other datasets or in-the-wild scenarios. To improve the cross-dataset generalization, existing works like [50] adopt a mix-and-train strategy, *i.e.* , mix multiple real-world datasets together with synthetic data for training. Following this approach, most RGB-based works tend to synthesize more training samples using a GAN [81] or a generation model [63].

For 2D pose, semi-supervised learning methods like [99] treat each 2D keypoint independently and select 'labels' based on heatmap peaks. For 3D pose, weakly- and semi-supervised learning explore using weak labels or simply unlabelled data to improve cross dataset performance. Works like [9, 4] use 2D pose or the hand mask as weak labels while projecting the points in 3D to image coordinates.

Self-supervised learning for 3D pose removes even the requirement of weak labels, The most related works are for depth-based inputs [20, 132, 135] and human pose estimation [51]. Depth-based works like [20] use point cloud reconstruction as an auxiliary task to improve the performance of 3D hand pose estimation. Beyond that, Wan *et al.* [132, 135] introduce model-fitting with differentiable renderers for depth map reconstruction to utilize unlabelled data. RGB images however are affected by illumination and complex backgrounds, which prevent direct application of reconstruction or rendering approaches to RGB. As for the RGB-based human pose estimation, existing work [51] focuses on unlabelled multi-view images, which is still a highly limited scenario.

6.2.2 Semi-Supervised Learning

Consistency training and pseudo-labeling has recently shown much promise for semi-supervised classification [107, 150, 44, 6, 2, 122] and segmentation [174, 31]. Recent semi-supervised works have achieved comparable performance to supervised methods with only a fraction of the labels. For consistency training, works like [150, 31] have explored various augmentations. The mean teacher strategy [122] accelerates consistency training by averaging model weights instead of label predictions. For pseudo-labeling, operations such as argmax [65], sharpening [6] or thresholding [107] have been introduced to modify predictions as labels. Others [2, 44, 174, 158] treat predictions as noisy labels and introduce label correction to generate pseudo-labels.

Our work is the first to explore pseudo-labelling and consistency learning for hand pose estimation. Several distinctions separate pose estimation from the previous application of these techniques for image classification and segmentation. Formulation-wise, it is a regression problem that critically depends on spatial information. Secondly, there is a clear separation between biomechanical feasible versus infeasible poses. Therefore, we design a novel pipeline for semi-supervised hand pose estimation with corrected pseudo-labels and spatial consistency.

6.3 Methodology

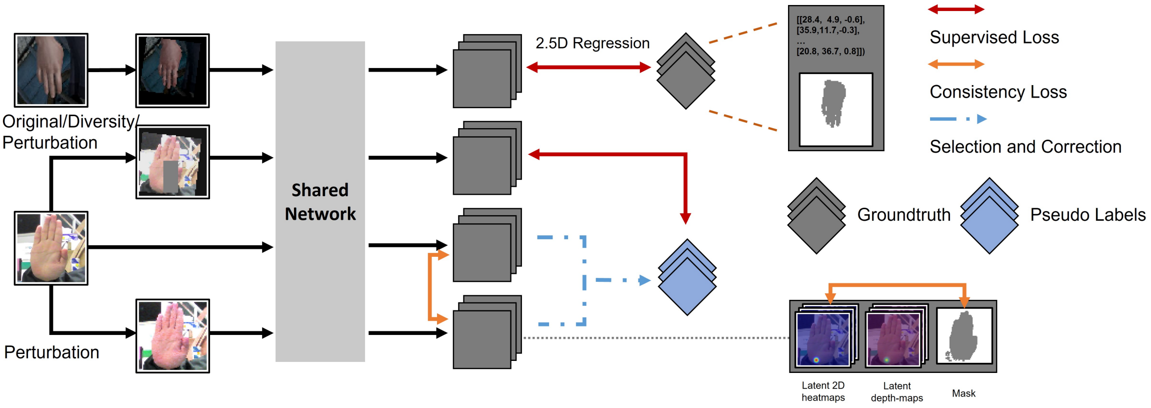


Figure 6.2: Overview of SemiHand. The model is pre-trained on labelled synthetic data. Consistency training (orange double headed arrow, see Sec. 6.3.3) on unlabelled real-world data with perturbation augmentations (see Sec. 6.3.4) and label correction and sample selection (blue dash-dotted arrow, See Fig. 6.1 and Sec. 6.3.2) together with augmentation of differing difficulties. (see Sec. 6.3.4).

We present an overview of our framework in Fig. 6.2. For pose estimation, let $X_L = \{(\mathbf{x}_i^l, \mathbf{p}_i, \mathbf{w}_i) : i \in (1, \cdots, N)\}$ be N labelled examples, where $\mathbf{x}_i^l$ is a labelled synthetic RGB image of a hand, $\mathbf{p}_i = (\mathbf{uv}_i, \mathbf{d}_i)$ is its target 2.5D hand pose, where $\mathbf{uv}$ is the the image pixel coordinates and $\mathbf{d}$ is its metric depth relative to the root keypoint, and $\mathbf{w}_i$ is a binary mask outlining the overall hand shape. Let $X_U = \{(\mathbf{x}_j^u) : j \in (1, \cdots, M)\}$ be M unlabelled examples, where $\mathbf{x}_i^u$ is an unlabelled real-world RGB image of a hand. We aim to estimate the 2.5D hand pose and its associated hand mask by learning a mapping f in the form of a neural network parameterized by θ, such that $(\mathbf{p}, \mathbf{w}) = f(\mathbf{p}, \mathbf{w}|\theta; X_L, X_U)$. In practice, the hand mask $\mathbf{w}$ is obtained by our shared fully convolutional network though our formulation is sufficiently general that it can also be learned by a separate network. We optimize a mixed objective of

$$\mathcal{L} = \mathcal{L}_{\text{sup}}(X_L) + L_{\text{unsup}}(X_U) + \lambda_c \mathcal{L}_{\text{cons}}(X_L, X_U), \tag{6.1}$$

where $\mathcal{L}_{\text{sup}}$ is the supervised loss, $\mathcal{L}_{\text{unsup}}(X_U)$ is an unsupervised loss with pseudo-labels and $\mathcal{L}_{\text{cons}}(X_L, X_U)$ is a consistency loss. λ_c is a hyperparameter. In the following, we introduce the details of the three losses.

6.3.1 Supervised Pose Estimation

A standard approach for 3D hand pose estimation is 2.5D pose regression [50] followed by a lifting into full 3D if camera intrinsics are known. The main benefit of regressing pose in 2.5D is the pixel-wise representation. This adds flexibility for multitask learning and can easily be extended to predict other pixel-wise outputs such as segmentations or depth maps with fully convolutional networks. The multitasking strategy achieves improvement for hand pose estimation [154]. In our work, besides 2.5D pose $\mathbf{p}$, we also predict hand mask $\mathbf{w}$. Here, we first define the distance ℓ between two 2.5D poses $\mathbf{p}_1 = (\mathbf{uv}_1, \mathbf{d}_1)$ and $\mathbf{p}_2 = (\mathbf{uv}_2, \mathbf{d}_2)$ as

$$\ell(\mathbf{p}_1, \mathbf{p}_2) = ||\mathbf{uv}_1 - \mathbf{uv}_2||_2^2 + \lambda_{\mathbf{d}}||\mathbf{d}_1 - \mathbf{d}_2||_2^2, \tag{6.2}$$

where $\lambda_{\mathbf{d}}$ is a hyperparameter with a value of 50. Given a ground-truth $\mathbf{p}_{gt}$, $\mathbf{w}_{gt}$ and corresponding predictions $\mathbf{p}$, $\mathbf{w}$, the supervised loss is defined as:

$$\mathcal{L}_{\text{sup}}(X_L) = \ell(\mathbf{p}, \mathbf{p}_{gt}) + \lambda_{\mathbf{w}}||\mathbf{w} - \mathbf{w}_{gt}||_1, \tag{6.3}$$

where $\lambda_{\mathbf{w}}$ is a hyperparameter. Here, we adopt the two-stacked hourglass with 2.5D regression as our backbone to estimate 2.5D representation and hand mask.

6.3.2 Pseudo-labels for Pose Estimation

For now, assume we have some initial network $f(\theta)$ from pre-training. We initialize pseudo-labels $\hat{\mathbf{p}} = (\hat{\mathbf{uv}}, \hat{\mathbf{d}})$ of X_U using the prediction of $f(\theta)$ and fine-tune the model with corrected pseudo-labels $\mathbf{r}$. With the prediction $\mathbf{p}$ from $f(\mathbf{p}|\theta; X_U)$, the objective $\mathcal{L}_{\text{unsup}}(X_U)$ can be formulated as:

$$\mathcal{L}_{\text{unsup}}(X_U) = \mathbb{1}(\mathcal{C}(\hat{\mathbf{p}}) \leq \tau)\ell(\mathbf{p}, \hat{\mathbf{p}}), \quad \text{where } \hat{\mathbf{p}} \backsim \mathcal{M}. \tag{6.4}$$

Here, $\mathbb{1}(\cdot)$ is the indicator function, $\mathcal{C}(\cdot)$ is a function to estimate the confidence of given pseudo-labels, and τ is a confidence threshold. Pseudo-labels are often noisy and may require corrections [72, 44]. In this objective, we constrain the pseudo-pose $\hat{\mathbf{p}}$ to be drawn from $\mathcal{M}$, a pose space whose points are biomechanical feasible poses in which bone lengths are consistent with the given hand model. Based on Eq. 6.4, we introduce a pose registration function $P(\cdot)$ to project the pseudo-labels $\hat{\mathbf{p}}$ to corrected poses $\mathbf{r}$ and add a loss to minimize the distance between the prediction $\mathbf{p}$ and $\mathbf{r}$. To prevent degenerate labels $\mathbf{r}$, we add a regularizer to encourage $\mathbf{r}$ to remain close to $\hat{\mathbf{p}}$. Adding these terms, we get

$$\mathcal{L}_{\text{unsup}}(X_U) = \mathbb{1}(\mathcal{C}(\hat{\mathbf{p}}) \leq \tau)\ell(\mathbf{p}, \hat{\mathbf{p}}) + \ell(\mathbf{r}, \mathbf{p}) + \ell(\mathbf{r}, \hat{\mathbf{p}}), \tag{6.5}$$

with $\mathbf{r} = P(\hat{\mathbf{p}})$. For learning the network θ and the pseudo-labels $\hat{\mathbf{p}}$, We solve the objective iteratively. First, we update the parameter of the network θ by

$$\mathcal{L}_{\text{unsup}}(X_U) = \mathbb{1}(\mathcal{C}(\hat{\mathbf{p}}) \leq \tau)\ell(\mathbf{p}, \hat{\mathbf{p}}) + \ell(\mathbf{r}, \mathbf{p}), \tag{6.6}$$

which can be solved by gradient descent. We then estimate the pseudo-labels $\hat{\mathbf{p}}$ and its

Algorithm 6.1 Semi-supervised hand pose estimation.

```
Require: Pre-trained model $\theta_0$ based on $\mathcal{L}_{sup}$, threshold $\tau$, epoch number $K$, $X_L$ and $X_U$
Ensure: Final model $\theta$ and pseudo labels $\hat{\mathbf{p}}$
Initialize the pseudo-labels $\hat{\mathbf{p}}$ for $X_U$
Initialize the corrected pseudo-labels $\mathbf{r}$ for $X_U$
for $t = 1, \ldots, K$ epochs do
    Calculate $\mathcal{C}(\hat{\mathbf{y}})$
    Update $\theta$ via gradient ascent of Eq. 6.6 with $\mathcal{L}_{sup}(X_L)$ and $\mathcal{L}_{cons}(X_L, X_U)$
    Update $\hat{\mathbf{p}}$ and $\mathbf{r}$ based on Eq. 6.7
end for
```

correction $\mathbf{r}$ based on the previous prediction $\mathbf{p}'$ and the previous correction $\mathbf{r}'$,

$$\begin{aligned} \hat{\mathbf{p}} &= \arg\min_{\hat{\mathbf{p}}} \ell(\mathbf{p}', \hat{\mathbf{p}}) + \ell(\mathbf{r}', \hat{\mathbf{p}}), \\ \mathbf{r} &= P(\hat{\mathbf{p}}). \end{aligned} \tag{6.7}$$

Label Correction. Estimating the joint locations independently is not effective to ensure the biomechanical feasibility of the hand. Inspired by the similarity transformation of [135], we propose a pose registration function P. More specifically, we estimate the transformation T with a greedy approximation based on the hand's kinematic chain. As shown in Fig. 6.4 right, given a template (black) and a prediction (gray), we first align the root by translation, and then calculate the bone direction (dotted gray line) using the parent node of registered pose and the child node of estimation. With calculating T of each bone along with the chain of a hand, we get the registered pose (orange). The proposed greedy approximation avoids the accumulation of end point errors and ensure the feasibility of bone lengths without any training.

Sample Selection. We design the confidence function $\mathcal{C}$ for samples based on the plausibility and stability of the pseudo-labels $\hat{\mathbf{p}}$ for the unlabelled data $\mathbf{x}^u$ as below:

$$\mathcal{C}(\hat{\mathbf{p}}) = \ell(\mathcal{T}(\hat{\mathbf{p}}), f(\mathbf{p}|\theta; \mathcal{T}(\mathbf{x}^u))) + \ell(\hat{\mathbf{p}}, P(\hat{\mathbf{p}})), \tag{6.8}$$

where $\mathcal{T}$ is a random perturbation augmentation. The proposed confidence is a sum of the distance between the prediction of perturbed image and its corresponding pseudo-label, and the distance between the pseudo-label and its corrected pseudo-label.

6.3.3 Self Consistency for Pose Estimation

For both X_L and X_U, we introduce a view consistency term $\mathcal{L}_{\text{vc}}$ and a cross-modal consistency term $\mathcal{L}_{\text{cc}}$ to improve generalization. The consistency loss $\mathcal{L}_{\text{cons}}(\mathcal{X}_L, \mathcal{X}_U)$ is simply the sum of the two:

$$\mathcal{L}_{\text{cons}} = \mathcal{L}_{vc} + \mathcal{L}_{cc}. \tag{6.9}$$

View Consistency. As shown in Fig. 6.3, we augment the training samples by rotating or translating the samples, as depicted in Sec. 6.3.4, and encourage transformed 2.5D predictions to be consistent with predictions of the transformed samples like existing 2D works [99]. The proposed loss function, with random perturbation $\mathcal{T}$ is:

$$\begin{aligned}\mathcal{L}_{vc} &= \ell(f(\mathbf{p}|\theta;\mathcal{T}(\mathbf{x})),(\mathcal{T}(f(\mathbf{p}|\theta;\mathbf{x})))) \\ &+ ||f(\mathbf{w}|\theta;\mathcal{T}(\mathbf{x})) - (\mathcal{T}(f(\mathbf{w}|\theta;\mathbf{x}))||_1.\end{aligned} \tag{6.10}$$

This loss encourages more robust and stable predictions for unlabelled data X_U.

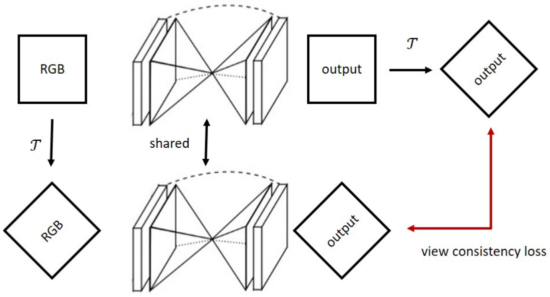


Figure 6.3: Overview of view consistency loss.

Cross-modal Consistency. Zamir *et al.* [166] observed that learning with cross-modal consistency improves prediction accuracy. In that regard, different modality representations *e.g.* RGB image, depth map, of the same hand should be 'consistent' in their pose. But how can we enforce this consistency across these modalities without actual pose labels? In this case, we incorporate multi-task learning and estimate multi-modal outputs *i.e.* pose and mask, and add a model-fitting energy term. The proposed energy function encourages consistency between the 2D pose and the hand mask, which we find improves pose and overall generalization. Additionally, we adopt a stop-gradient operation stop($\cdot$) to the mask as shown in Fig. 6.5 to prevent inaccurate poses from degenerating the masks.

Specifically, we approximate the hand mask with 55 circles: 9 for each finger and 10 for the palm. The circle hand model is parameterized as $\mathbf{m} = \{m^0, \cdots, m^{54}\}$, where $m^i = (c^i, r^i)$ is the i^{th} circle centered at c^i with radius r^i. The circle centers are manually defined based on the 2D pose, while radii are pre-trained from synthetic data. Fig. 6.4 middle shows an example of the approximated hand mask and the circles for the little finger.

The cross-modality consistency loss $\mathcal{L}_{\text{cc}}$ is the sum of two standard model-fitting energy terms:

$$\mathcal{L}_{\text{cc}}(\mathbf{uv},\mathbf{w}) = \mathcal{L}_{\text{m2d}}(\mathbf{uv},\text{stop}(\mathbf{w})) + \mathcal{L}_{\text{d2m}}(\mathbf{uv},\text{stop}(\mathbf{w})). \tag{6.11}$$

The model-to-data term $\mathcal{L}_{\text{m2d}}$ is an L1 distance encouraging the circle-approximated mask to be as similar as possible to the estimated mask:

$$\mathcal{L}_{m2d}(\mathbf{uv},\mathbf{w}) = ||R(G(\mathbf{uv})) - \mathbf{w}||_1, \tag{6.12}$$

where $G(\cdot)$ estimates the centers and radius based on the 2D hand pose and $R(\cdot)$ renders the circles to a hand mask like [132]. Note that this term has no gradients on the background of the rendered mask. Hence, we add a data-to-model term $\mathcal{L}_{d2m}$ to measure the registration error between the estimated hand model and hand mask:

$$\mathcal{L}_{d2m}(\mathbf{uv}, \mathbf{w}) = \sum_{g \in \Omega} d(\mathbf{w}(g), G(\mathbf{uv})), \tag{6.13}$$

where Ω is the set of all pixel locations and the distance function $d(\cdot)$ is defined as:

$$\begin{aligned} & d(\mathbf{w}(g), \mathbf{m}) \\ = & \begin{cases} \max(\min\limits_{i \in [0,54]}(||g - c^i||_2 - r^i), 0) & \text{if } \mathbf{w}(g) = 1, \\ \max(\max\limits_{i \in [0,54]}(r^i - ||g - c^i||_2), 0) & \text{otherwise.} \end{cases} \end{aligned} \tag{6.14}$$

Specifically, the distance estimates pixel g's distance to the nearest circle m^i with radius r^i centered at c^i. If the predicted mask value at g is correct, the distance is set to 0.

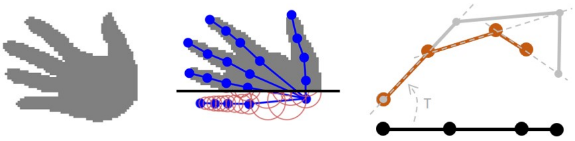


Figure 6.4: Hand model and pose registration. Left: the ground-truth hand mask; Middle: Our rendered hand mask based on ground-truth 2D pose (blue points); Right: pose registration of the template hand (black) to observed joints (grey) to result in a registered hand (orange). Figure best viewed in colour.

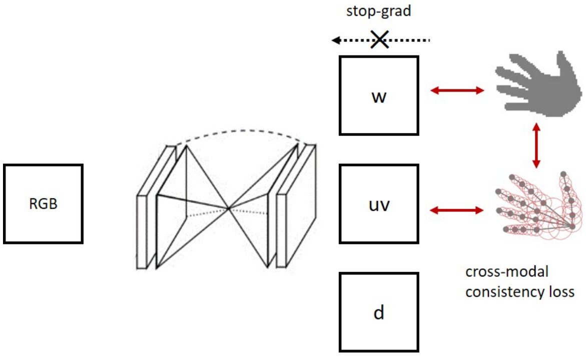


Figure 6.5: Overview of cross-modal consistency loss. (uv, d) are 2.5D hand outputs; **w** denotes the hand mask.

6.3.4 Data Augmentation

Initially, we found that adding view-point consistency to be non-convergent. We speculated the cause to be mode collapse, *i.e.* all the 2D pose predictions gradually move to the center of the image. A similar phenomenon was observed in FixMatch [107]; they found that data augmentation of differing difficulties could improve training stability. As such, we also adopt two types of data augmentation like [107], as shown in Fig. 6.2. Specifically, we introduce diversity augmentation for the labelled and high-confidence pseudo-labelled data and perturbation augmentation for unlabelled data respectively, which we found to mitigate the problem of mode collapse.

In all of our experiments, diversity augmentation is similar to augmentations used in existing supervised learning methods [50, 11, 155]. It includes color jitter, translation, rotation, scale, gray-scale and random erasure. Differently, for unlabelled data, we simply perturb with translations of [-5,5] pixels or rotations of either $[-2°, 2°]$ or 90°,180° and 270°.

6.4 Experiments

6.4.1 Implementation Details

In the experiments, we adopt the two-stacked hourglass as our backbone. The input and output resolution are both 64×64. We set the hyperparameters from Eqs. 6.1 to 6.4 empirically, with $\lambda_c = 0.1$, $\lambda_\mathbf{d} = 50$, $\lambda_\mathbf{w} = 100$ and $\tau = 1.5$. For pre-training on the synthetic data, we use an Adam optimizer with an initial learning rate of 10^{-3} and a batch size of 32. We train the model for 100 epochs, lowering the learning rate by a factor of 10 at the 60^{th} and 90^{th} epoch. For fine-tuning, we use the learning rate 10^{-4} and a batch size of 128. We set K to 10. At 5^{th} iteration, we lower the learning rate to 10^{-5}. The associated algorithm is shown in Alg. 6.1.

6.4.2 Datasets and Evaluation Metrics

Our method is trained on one synthetic dataset, the Rendered Hand Pose Dataset (RHD) [178] and evaluated on four real-world datasets, Stereo Hand Pose Tracking Benchmark (STB) [167], Dexter+Object Dataset (DO) [113], Hand-3D-Studio (H3D) [173] and YouTube 3D Hands (YT3D) [63].

To further verify the effectiveness of our proposed method, we also introduce and evaluate on a new real-world hand sequence dataset (HSD) [161]. HSD is a video dataset with 3D poses annotated in a semi-automated fashion like [179]. It consists of 4 sequences. Each sequence is performed by one actor and contains 20K frames. We use the first two sequences for training and others for testing.

To evaluate the accuracy of estimated poses, we use two common metrics: (1) mean end-point-error (EPE), measuring the average Euclidean distance between predicted and ground-truth joints, and (2) area under the curve (AUC) on the percentage of correct keypoints (PCK) curve based on certain error thresholds. For a fair comparison with state-of-the-art, we follow [111, 155], assuming that the global hand scale and the hand root position are

Method	training set	testing set	
		STB train	STB test
baseline	RHD train(w/)	23.41	23.83
baseline	STB train(w/)	5.27	18.04
baseline	RHD train(w/) STB train(w/)	5.25	7.32
with vc	RHD train(w/) STB train(w/o)	19.98	21.03
with cc		20.59	20.92
with vc+cc		19.18	19.93
with pseudo-labeling	RHD train(w/) STB train(w/o)	15.68	16.31
our proposed	RHD train(w/) STB train(w/o)	13.82	14.60
our proposed	RHD train(w/) STB test(w/o)	15.83	14.51
our proposed	RHD train(w/) STB train+test(w/o)	13.78	13.95

Table 6.1: Ablation study with mean EPE [mm]. w/ and w/o indicates with and without labels for training.

known, and set the middle finger's base position as the hand root. For convenience, we also assume that hand template is given. For H3D and YT3D, we use 40 mm from STB as reference bone length defined by [179]. Our default setting is fine-tuning with only the training data of a (single) real-world dataset's training partition. Following the convention of [132], the test data is withheld completely. Additionally, we use the labels of these real-world datasets only for evaluation purposes.

6.4.3 Ablation Study

Baseline. To start with, we first investigate the domain gap that exists between the synthetic RHD versus the real-world STB. The pre-trained network, trained and tested on RHD achieves good performance with a mean EPE 12.08 mm. However, the same network's errors almost double to a mean EPE of 23.41 mm and 23.83 mm on the STB training and testing datasets respectively (see 'baseline' method in Tab. 6.1). If we train the network only on STB, it is prone to over-fitting due to the small size of the dataset, so it leads to a large error on testing data (18.04 mm). If one merges the training datasets of RHD and STB in a mix-and-train strategy, we can lower this error to 7.32 mm and this serves as the upper bound in performance for semi-supervised methods.

Impact of our components. We next analyse the performance of our method's individual components to isolate the impact of consistency training and pseudo-labeling. We fine-tune the pre-trained model with only view consistency loss (with vc), only cross-model consistency loss (with cc), both consistency losses (with vc+cc) and with pseudo-labelling in Tab. 6.1. Each component improves the performance; adding pseudo-labelling achieves

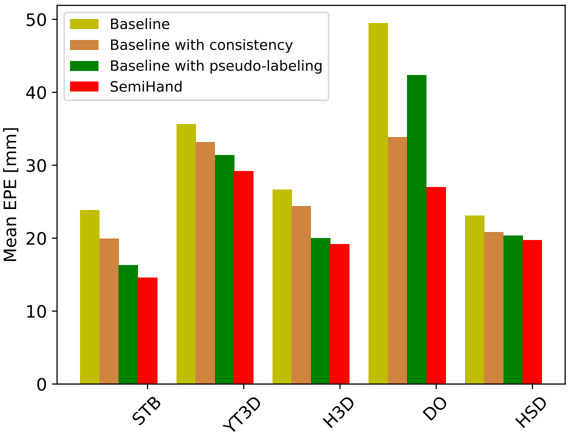


Figure 6.6: Comparison of baseline, with only consistency training, with only pseudo-labeling and our proposed SemiHand. Our proposed two modules both improve the performance with respect to the baselines, and their combination further leads to a higher accuracy.

an impressive 7.52 mm improvement on the STB testing set. Combining these components further decreases the error. With both consistency training and pseudo-labeling, we achieve a 9.23mm improvement on the STB testing set with fine-tuning on the unlabelled STB training set.

For further verification, we compare the following: (1) baseline, (2) baseline with consistency training, (3) baseline with pseudo-labels and (4) our proposed method on all real-world datasets (see results in Fig. 6.6). We can see that both consistency training and pseudo-labeling can improve the performance with respect to the baselines. Furthermore, the combination of our two modules leads to a higher accuracy. With our semi-supervised fine-tuning, we achieve a decrease in mean EPE of up to 9.2 mm on STB, 22.4 mm on DO, 6.4 mm on YT3D, 7.46 mm on H3D and 3.3 mm on HSD as shown in Fig. 6.6. The full model is comparable to existing supervised methods.

Impact of training data. In Tab. 6.1 under 'our proposed', we fine-tune the network on different STB sets, *i.e.* , STB train set only, STB test set only and both. We find that fine-tuning on the testing image directly achieve lower mean EPE (13.82 mm/13.78 mm versus 15.83 mm for STB train and 14.51 mm/13.95 mm versus 14.60 mm for STB test). Moreover, as the amount of unlabelled training data increases, the mean EPE decreases correspondingly. As shown in Tab. 6.1, fine-tuning with both STB train and test sets outperforms fine-tuning independently. We also verify this by fine-tuning with different percentages of STB training data in Fig. 6.10. We decrease the mean EPE of STB test set from 17.31mm to 14.60 mm by increasing the percentage of unlabelled STB training data during training.

6.4.4 Comparison to State-of-the-Art

We compare our hand pose estimation results with state-of-the-art methods [4, 81, 50, 148, 156, 155, 9, 111, 90], on STB and DO as shown in Fig. 6.7 and 6.8. We can see that

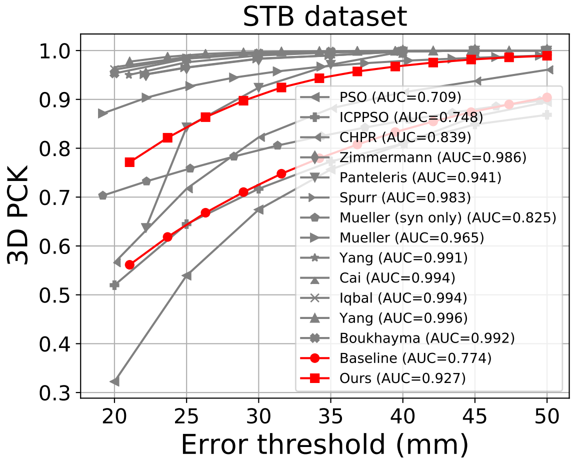


Figure 6.7: AUC: Comparison to state-of-the-art on STB. Our SemiHand improves the baseline's AUC and achieves comparable performance to other supervised learning methods.

after fine-tuning, our SemiHand improves the baseline's AUC significantly (0.774 to 0.927 for STB, 0.546 to 0.747 for DO). For STB, our semi-supervised method achieves comparable performance to other supervised learning methods, even without any labels of STB. The work [81] also reports its performance training on synthetic data only. As shown in Fig. 6.7, ours outperforms [81] by a large margin (0.927 vs. 0.825).

Many existing methods use DO to evaluate cross-dataset performance. Our proposed semi-supervised method outperforms most existing supervised methods, even though they mix-and-train RHD with other synthetic data [4, 81], STB [167], MPII+NZSL [104] or MVBS [104]. This confirms our original motivation of exploiting unlabelled RGB images and improving the accuracy of pose estimation. Note that [148] does report better performance but they incorporate a large-scale (111K) labelled real-world dataset for training.

With our proposed semi-supervised method, the predictions of unlabelled data will gradually converge. We show two qualitative examples of the gradual convergence from the predictions of pre-trained model to our stable predictions in Fig. 6.9. Interestingly, we also find cases like the example shown in Fig. 6.9, where our predictions seem more accurate than the manually annotated ground-truth, *i.e.* predicted keypoints are centered on the finger, while labelled keypoints lie at the edge of the fingers. Given the saturated results of state-of-the-art methods on STB, it is likely that many networks are over-fitting to manual annotation biases or noise.

6.4.5 Comparison to Weakly-Supervised Methods

As our SemiHand is the first semi-supervision framework for 3D hand pose estimation from monocular images, there are no direct comparable methods. We compare instead to a weakly-supervised method [9]. We fine-tune the pre-trained model on m% STB training data, either without any labels (ours, SemiHand), with ground-truth (strong supervision) and with weak

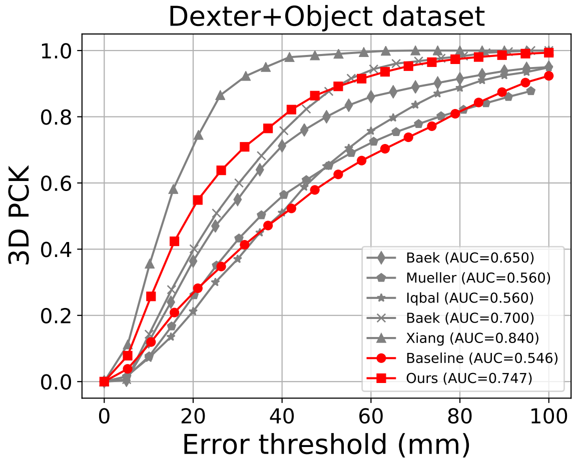


Figure 6.8: AUC: Comparison to state-of-the-art on DO. Our SemiHand improves the baseline's AUC and outperforms some supervised learning methods using the mix-and-train strategy.

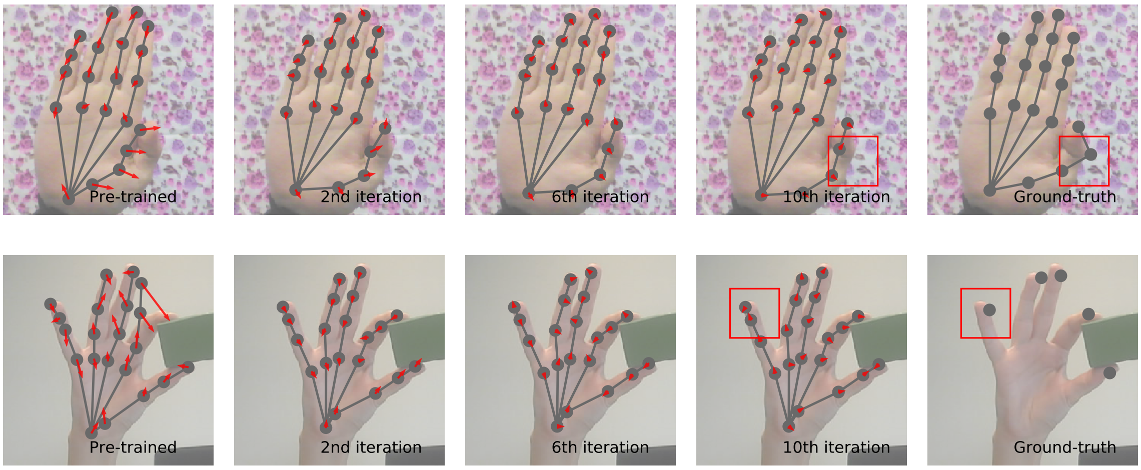


Figure 6.9: Gradual convergence from the prediction of pre-trained model to our final prediction. The arrows indicate the direction and distance of prediction movement during fine-tuning. For 10$^{\text{th}}$iteration, the optimization converges because the length of arrows become almost zeros. We highlight the differences between our stable predictions and the ground-truth poses with red boxes. Figure best viewed in colour.

labels of either 2D poses or masks. The percentage of STB training set is varied from 5% to 100% to compare the mean EPE on STB testing set. As shown in Fig. 6.10, when fine-tuning with masks or 2D poses as weak labels, the weakly-supervised method [9] achieves 4.0 mm and 7.1 mm improvement on STB testing set respectively. This indicates that 2D pose provides stronger supervision than simply a mask. Meanwhile, without any labels, our SemiHand achieves a 9.2 mm improvement, demonstrating the effectiveness of our method compared

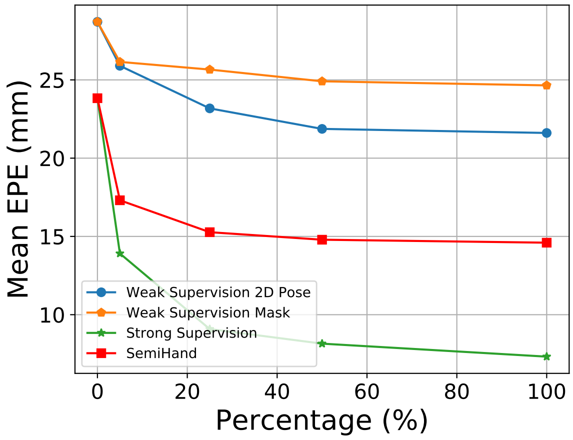


Figure 6.10: Mean EPE on STB testing data with fine-tuning on different percentage of STB training data. As the amount of training data increases, SemiHand achieves a similar trend as the weakly-supervised methods, i.e., the mean EPE decreases correspondingly.

to [9]. Note that we discuss only the relative improvement as we use a different backbone than [9]. Given that adding even a small amount of labels (as per the fully supervised method) is still better, this encourages us to further explore the use of unlabelled images.

6.5 Conclusions

We aim to develop a semi-supervised 3D pose estimation framework, using labelled synthetic and unlabelled real-world data. Directly applying the existing semi-supervised method is nontrivial because pose estimation is a regression problem that critically depends on spatial information. We therefore designed a new framework based the pose feasibility and spatial consistency, with pseudo-labels and consistency training. Experiments on different datasets demonstrate that our approach successfully leverages real-world RGB images without any labels, paving a path forwards for learning pose estimation systems with only synthetic labels. In the future, we would like to explore two research directions. First, We aim to remove even the requirement of unlabelled data. We would like to explore the techniques like domain randomization and hand image renderer, and make the model purely trained on synthetic data may achieve satisfactory result in the real world. Second, we will explore few-shot learning setting to make the pre-trained model migrate to a specific application scenario easily. In other words, we aim to transfer a general model to a personalized model.

Chapter 7

Dual-Modality Network for Semi-Supervised Hand Pose Estimation

Contents

7.1 Motivation **91**

7.2 Related Work **92**

7.2.1 Contrastive Learning 92

7.2.2 Semi-Supervised Learning 93

7.3 Method **93**

7.3.1 Problem Definition 93

7.3.2 Multi-modal Supervision 94

7.3.3 Model Architecture 94

7.3.4 Pre-training with Multi-modal Alignment 96

7.3.5 Fine-tuning with Noisy Pseudo-Labels 98

7.4 Experiments **99**

7.4.1 Datasets & Evaluation 99

7.4.2 Implementation Details 100

7.4.3 Augmentation 100

7.4.4 2D Keypoint Detection 101

7.4.5 3D Keypoint Estimation 101

7.4.6 Ablation Study 102

7.4.7 Qualitative Results 103

7.5 Conclusion **104**

As the rendering process for generating synthetic data makes it easy to synthesize multiple data modalities, we further propose to use synthetic multi-modal data as auxiliary information to aid the training of real-world data. The advantages of synthetic multi-modal data can be derived from combining multi-modal data with synthetic data. We reduce the reliance on real-world data by using synthetic data and further improve the performance of our proposed system by using multi-modal representation learning. Specifically, in this chapter, we still focus on the challenging scenario of learning models from labelled multi-modal synthetic data and unlabelled real-world data like Chapter 6. We propose a novel dual-modality network that exploits multiple data modalities of synthetic data.

First, we explore synthetic data to pre-train the model with multi-modal alignment. During pre-training, we introduce to align modalities in a low-dimensional latent space and in a feature space, thus facilitating better representation. To align modalities in a low-dimensional latent space, we adopt multi-modal contrastive learning. The features extracted from the different modalities of similar poses should be close in the feature space. Intuitively, we can treat each RGB and depth map pair as positive samples and cross-pair combinations as negatives. Due to the large visual difference between RGB images and depth maps, pushing these features far away has no effect on the uniformity purpose, *i.e.* , preserving maximal information. Therefore, instead of creating positive pairs based on different modalities, we consider a way of regulating their feature distances in a closer subspace by using their fusion. To align modalities in a feature space, inspired by RGB image-depth map pairs that have pixel-level correspondences, we propose to align their feature maps with an attention-based fusion and a shared encoder. The alignment makes it easier for the RGB encoder to perform cross-modal learning and capture shared visual cues.

Second, we aim to reduce the noise of pseudo-labels during fine-tuning. The pseudo-labels from pre-trained model are inevitably noisy and used naively, are even detrimental to learning as the model will over-fit to the noise. Therefore, we integrate the proposed method with pose correction and self-distillation during fine-tuning. Benefiting from the design of our dual-modality network, we take the predicted depth map from the RGB branch as an input to the fusion branch and construct a self-distillation structure. With self-distillation, we encourage the refined prediction to be consistent with its past prediction, distill the knowledge to obtain a softer prediction, and generate a pseudo-label accordingly.

For the model training, we use a synthetic dataset, and four real-world hand datasets. Experiments show that our framework beats existing state-of-the-art methods by a large margin. We visualize the multi-modal predictions and the intermediate attention, and find that after fine-tuning, the predicted hand depth maps and hand segmentation masks will be more complete and the attention region will focus more on the hand region. In the future, we intend to dive deeper into contrastive learning and self-distillation for semi-supervised hand pose estimation. Also, we will try to make the model lightweight by redesigning the depth map branch. The publication, contributors and author contributions in this chapter are listed below:

Publication:

- Qiuxia Lin*, Linlin Yang* and Angela Yao. "Dual-Modality Network for Semi-Supervised Hand Pose Estimation." *In Submission.* 2022. * equal contribution.

Other Contributors:

- Qiuxia Lin (PhD student)
- Angela Yao (Thesis Supervisor)

Contributions:

- Qiuxia Lin and Linlin Yang developed the method jointly where Qiuxia Lin was more responsible for the algorithm and the implementation, and Linlin Yang was more responsible for the algorithm and the analysis. Qiuxia Lin and Linlin Yang wrote the main body of the article. Qiuxia Lin and Linlin Yang and Prof. Dr. Angela Yao developed the idea, analyzed the results and revised the article.

7.1 Motivation

Hand pose estimation supports a wide range of applications, including sign language recognition [60, 58] and gesture-based human-computer interaction systems [10]. However, training deep-learning-based hand pose estimation systems requires a large amount of accurate ground truth labels, which are difficult to obtain. Training models with synthetic data [178] can bypass this label scarcity, but such models generalize poorly to real-world settings due to the domain gap between synthetic data and real data. While the careful modelling of synthetic data can narrow this gap, the performance drop is still noticeable [81].

Here, we address the cross-domain pose estimation problem, focusing on a semi-supervised setting. We target learning from labelled synthetic data and unlabelled real-world data, for application on real-world data. It is now commonplace for methods to pre-train with synthetic data and then subsequently fine-tune with real data [53, 153]. Such approaches, however, leverage only the RGB modality of synthetic datasets. Yet the rendering process for generating synthetic data makes it easy to synthesize multiple data modalities. For example, the RHD dataset [178] makes both photo-realistic RGB images and depth maps available.

We believe that there are common visual cues shared by the different modalities, such as the underlying geometry, or semantics. Leveraging these common cues can enhance RGB-based hand pose estimation and limit a model's sensitivity to non-informative cues and shortcuts, such as background or texture appearances [40, 155]. To that end, we propose a dual-modality network for RGB images and depth maps that aligns their features via an attention-based multi-modal pre-training. Based on an RGB image and depth map encoder, we design a fusion branch with an attention module that fuses local and global relationships learned from depth maps to RGB features. The fusion enables the RGB branch to better capture features relating to the common visual cues in the depth map. We subsequently design a multi-modal contrastive learning applied to all three branches that allows us to construct a well-structured feature space that aligns similar poses across different modalities.

After obtaining a pre-trained model, a common practice is to utilize pseudo-labelling [66, 12, 80] to incorporate unlabelled data for fine-tuning. However, naïvely generated pseudo-labels are inevitably noisy and deteriorates model performance. To handle noisy pseudo-labels, we propose a correction of hand poses based on the feasibility of bone lengths and joint angles. Moreover, based on architectural dependencies between the RGB and fusion branches, we construct a self-distillation structure to obtain softer predictions for pseudo-label generation. This will encourage the model to gradually improve pseudo-labels instead

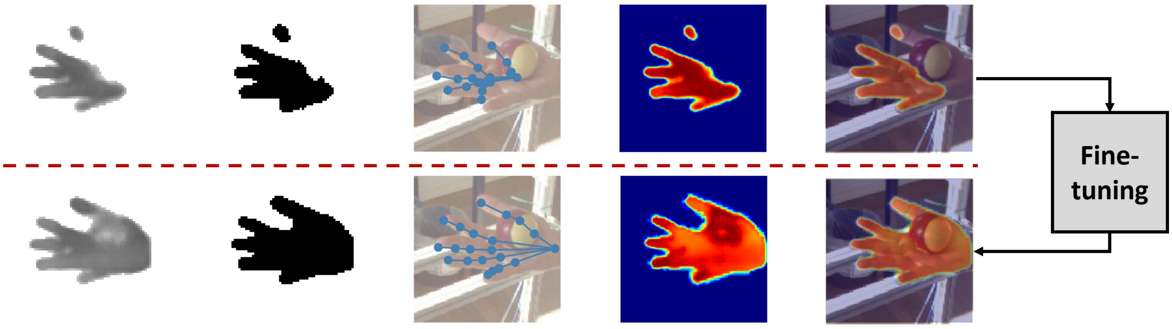


Figure 7.1: Visualisation of one prediction from FreiHAND before and after fine-tuning. From left to right: multi-modal predictions (depth maps, segmentation masks, poses), attention weights from depth maps and attention weights on RGB image. Our model takes in multi-modal predictions based on the pretrained model and monocular RGB and output corrected multi-modal predictions.

of replacing them dramatically [66, 57]. As shown in Fig. 7.1, we can see that multi-modal predictions are much more accurate after applying pose correction and self-distillation.

In summary, we make the following contributions:

1. We propose a dual-modality network that learns from RGB images and depth maps during pre-training but is applicable during fine-tuning and inference to only monocular RGB inputs. The network features a specially designed cross-modal attention module that enables the RGB branch to better capture common visual cues in the depth map.

2. We propose a multi-modal contrastive learning for synthetic data on a supervised setting. By creating positive pairs based on the fused features, our proposed contrastive loss avoids the large discrepancy of using different modalities and facilitates better representation.

3. To exploit the noisy pseudo-labels during fine-tuning, we adopt pose correction for hand to guarantee the biomechanical feasibility of hand poses and introduce self-distillation based on our dual-modality network.

4. Our extensive experimentation shows that the proposed method significantly improves the state-of-the-art by up to 16.0% and 19.2% for 2D keypoint detection and 3D keypoint estimation tasks, respectively.

7.2 Related Work

7.2.1 Contrastive Learning

Contrastive learning encourages the model to learn a low-dimensional space for data in which similar sample pairs (positive pairs) stay close together while dissimilar samples (negative pairs) are further apart. It has been successfully applied in both unsupervised [125, 18, 17] and supervised [56] settings. Creating beneficial positive-negative pairs forms the basis of contrastive learning. Existing works [125, 18, 17, 56] prefer to create positive pairs based on data augmentation. Interestingly, a recent work [125] introduced the use of different modalities of one instance as a positive pair, showing great potential. However, the large discrepancy between the different modalities may still limit the performance.

Previous pose estimation works [177, 109] explored contrastive learning with unlabelled RGB images during pre-training. In contrast, we explore the use of labelled multi-modal synthetic data. Specifically, we create positive pairs for RGB features by fusing RGB images and depth maps. This approach avoids the large discrepancy of using different modalities during pre-training and helps the model to align modalities in low-dimensional space and facilitates better representation.

7.2.2 Semi-Supervised Learning

As acquiring 3D annotations for real-world images is difficult, pose estimation works often study how to learn with limited annotations. To exploit unlabelled data, various consistency constraints or pseudo-labelling strategies have been explored, including temporal consistency [15], temporal pseudo-labels [71] and multiview consistency [132] for video sequences and multiview images. To further remove the temporal or multiview requirement for unlabelled data, template-corrected pseudo-labels [153] and photometric consistency [21] based on model-fitting have been introduced.

A special case of semi-supervised learning is to learn from only labelled synthetic data and unlabelled real-world data. However, this new setting also introduces an additional domain gap that makes it more challenging. In addition to simple training with consistency or pseudo-labels [80, 153], recent works [53, 66] have started to introduce other strategies, such as domain adaptation, as auxiliary supervisions to bridge the gap. As pseudo-labels tend to be noisy, we emphasize how to exploit these noisy pseudo-labels during fine-tuning. Specifically, we explore pose correction for hand to guarantee the biomechanical feasibility of hand poses and introduce self-distillation to alleviate the negative effect of noisy pseudo-labels.

7.3 Method

7.3.1 Problem Definition

We aims to estimate the 2D and 3D keypoints of the hand from a monocular RGB image in a cross-domain setting. The problem can be formulated as follows. Given the set of synthetic data $\mathcal{D}^s = \{(\mathbf{x}_i^s, \mathbf{y}_i)\}_{i=1}^{N_s}$, we have for each synthesized RGB image $\mathbf{x}_i^s \in \mathbb{R}^{3\times H\times W}$ the multi-modal labels $\mathbf{y}_i = (\mathbf{p}_i, \mathbf{d}_i, \mathbf{m}_i)$ in the form of a 2.5D pose $\mathbf{p}_i \in \mathbb{R}^{J\times 3}$, a depth map $\mathbf{d}_i \in \mathbb{R}^{1\times H\times W}$, and a binary segmentation mask $\mathbf{m}_i \in \mathbb{R}^{1\times H\times W}$. Note that the 2.5D pose $\mathbf{p}$ is expressed as a triplet of the 2D pose and the metric depth relative to the root. For real-world data $\mathcal{D}^r = \{(\mathbf{x}_j^r)\}_{j=1}^{N_r}$, we have real RGB image $\mathbf{x}_j^r \in \mathbb{R}^{3\times H\times W}$ without access to any of the multi-modal labels.

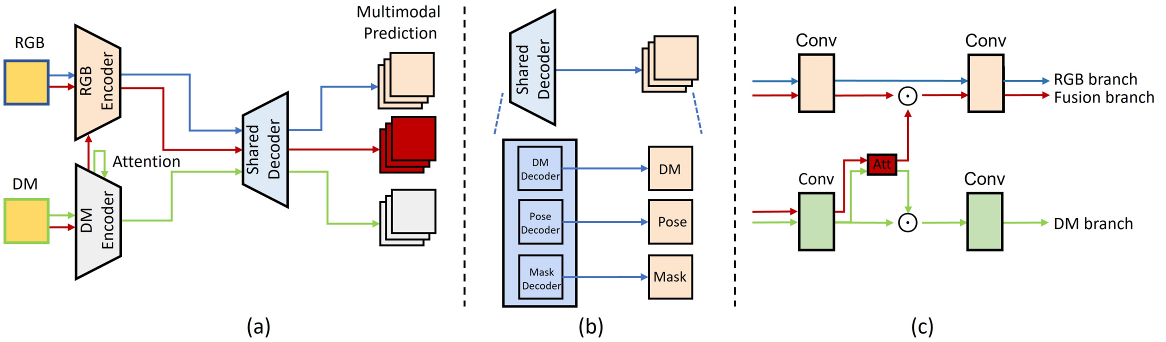


Figure 7.2: (a) Overview of our proposed dual-modality, *i.e.* , the RGB modality and the depth map (DM) modality, and the three branches framework, *i.e.* , the RGB branch (blue arrows), DM branch (green arrows) and fusion branch (red arrows); (b) illustration of the shared decoder and the multi-modal output; (c) design of the attention module. Note that we only use the RGB branch during inference.

A straightforward approach in cross-domain hand pose estimation is training the model on synthetic data $\mathcal{D}^s$ and then applying it to real data $\mathcal{D}^r$. We denote the training on $\mathcal{D}^s$ as only a baseline (Sec. 7.3.2). Unsurprisingly, applying only multi-modal supervision to synthetic data does not yield good performance for real data due to the inherent discrepancy in distributions across the domains. To alleviate this discrepancy, we design a dual-modality network with three branches (Sec. 7.3.3) and introduce a pre-training (Sec. 7.3.4) and fine-tuning (Sec. 7.3.5) strategy.

7.3.2 Multi-modal Supervision

We follow a multi-modal pose estimation pipeline for $\mathcal{D}^s$, similar to [50, 153], and simultaneously predict the 2.5D representations $\mathbf{p}$, segmentation masks $\mathbf{m}$ and depth maps $\mathbf{d}$ from a given RGB input. Specifically, we use as below a multi-modal supervised loss with ground truth $\mathbf{y}_{gt} = (\mathbf{p}_{gt}, \mathbf{m}_{gt}, \mathbf{d}_{gt})$ and the corresponding predictions $\mathbf{y} = (\mathbf{p}, \mathbf{m}, \mathbf{d})$:

$$M(\mathbf{y}_{gt}, \mathbf{y}) = \ell(\mathbf{p}, \mathbf{p}_{gt}) + \lambda_{\mathbf{m}}||\mathbf{m} - \mathbf{m}_{gt}||_1 + \lambda_{\mathbf{d}}||\mathbf{d} - \mathbf{d}_{gt}||_1, \tag{7.1}$$

where $\lambda_{\mathbf{m}}$ and $\lambda_{\mathbf{d}}$ are trade-off hyperparameters to balance the weights of the different modalities. ℓ is the 2.5D pose distance, which is the sum of the weighted Euclidean distance between two 2D poses and that between two metric depths relative to the root keypoint, defined in [153].

7.3.3 Model Architecture

Here, we introduce our dual-modality network for the RGB and depth map modalities. The two inputs have their own input-encoding branches. Beyond the independent inputs, we define a fusion branch that applies an attention module to enable features from the RGB images to be activated in more relevant regions. Put together, our framework comprises three branches: the RGB branch, the depth map (DM) branch and the fusion branch. All three branches project to a common latent space, and are illustrated in Fig. 7.2 (a)-(c) respectively. The encoders of the RGB and depth map modalities have the same architecture, *i.e.* , ResNet101. All three branches share a single decoder *i.e.* , 3 deconvolution layers with BN and ReLU, to produce a final set of feature maps, which are then decoded into 2.5D pose, mask and depth map outputs as shown in Fig. 7.2 (a)-(c).

Training of our network is split into a pre-training and fine-tuning stage. During pre-training, the input of the network are RGB images with the accompanying depth maps. The depth maps can be be either the ground truth depth map from $\mathcal{D}^s$ or a depth map predicted from an RGB input encoded and decoded through the RGB branch. During fine-tuning, only the RGB modality is given, so we use it together with the predicted depth map from RGB branch as inputs for the network. During inference, we use the RGB branch with RGB images only for prediction.

7.3.3.1 Attention Module.

To reveal the relationships between the local and global responses in the feature maps, we design an attention module $\mathrm{Att}(\cdot)$ to estimate the local attention weights. Specifically, given a feature map $\boldsymbol{f} \in \mathbb{R}^{c\times h\times w}$, $\boldsymbol{f}_{ij} \in \mathbb{R}^{c\times 1\times 1}$ is the feature vector at position $[i, j]$, and $\bar{\boldsymbol{f}} = pool(\boldsymbol{f}) \in \mathbb{R}^{c\times 1\times 1}$ is the average 2D spatial values on $\boldsymbol{f}$. The $pool(\cdot)$ function is a channel-wise average pooling operation. Note that $\bar{\boldsymbol{f}}$ and $\boldsymbol{f}_{ij}$ have the same shape. The attention weight $\boldsymbol{w}$ is defined as the inner product $\langle\cdot\rangle$ of $\bar{\boldsymbol{f}}$ and $\boldsymbol{f}_{ij}$.

$$\boldsymbol{w}_{ij} = \frac{\langle \bar{\boldsymbol{f}}, \boldsymbol{f}_{ij}\rangle}{\sum_{i=1}^{h}\sum_{j=1}^{w}\langle \bar{\boldsymbol{f}}, \boldsymbol{f}_{ij}\rangle} \times (h \times w). \tag{7.2}$$

The attention activated feature is defined as $\mathrm{Att}(\boldsymbol{f}) \odot \boldsymbol{f}$, where $\odot$ is a channel-wise multiplication operation. Our proposed attention mechanism helps the model focus more on the relevant region of the input than on irrelevant parts.

The proposed attention module is embedded into the two ResNet encoders after the conv1 and conv2_x to conv5_x layers. Suppose $\boldsymbol{f}^{RGB}$ and $\boldsymbol{f}^{DM}$ are a pair of corresponding intermediate feature maps, with the same shape, derived from the RGB branch and DM branch, respectively. We utilize $\boldsymbol{f}^{DM}$ to apply attention to both $\boldsymbol{f}^{RGB}$ and $\boldsymbol{f}^{DM}$ to obtain an attention-fused branch $\mathrm{Att}(\boldsymbol{f}^{DM}) \odot \boldsymbol{f}^{RGB}$ and a self-attended DM branch $\mathrm{Att}(\boldsymbol{f}^{DM}) \odot \boldsymbol{f}^{DM}$, as shown in Fig. 7.2 (c). Here, we use $\mathrm{Att}(\boldsymbol{f}^{DM})$ instead of $\mathrm{Att}(\boldsymbol{f}^{RGB})$ for the RGB branch as monocular RGB images suffer from depth ambiguities and the attention from the 2.5D information will make it easier for the RGB encoder to capture the 2.5D information and semantic meanings.

We set the fusion branch to share the same encoder as the original RGB branch, but set its own unique Batch Normalization (BN) layers due to the different statistics. Regarding the depth map encoder, the fusion branch directly uses the BN layers of the depth map branch.

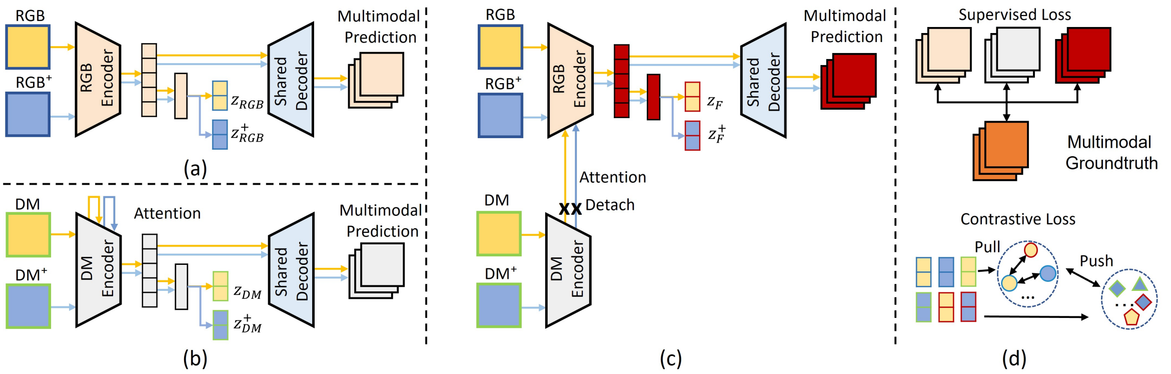


Figure 7.3: The illustration of (a) the RGB branch, (b) the DM branch, (c) the fusion branch and (d) the overall losses during pre-training. The yellow and blue arrows show the forward of a positive pair. The superscript $^+$ denotes the positive paired sample.

7.3.4 Pre-training with Multi-modal Alignment

Fig. 7.3 (a)-(c) shows the overall architecture during pre-training. We pre-train the model based on the synthetic data $\mathcal{D}^s$. Since the ground-truth depth map provides 2.5D information, we consider aligning the features from different branches to help the RGB modality capture such information. To this end, we propose a multi-modal contrastive scheme with an attention-based alignment.

7.3.4.1 Multi-modal Contrastive Learning.

The features extracted from the different modalities of similar poses should be close in the feature space. Intuitively, we can treat each RGB and depth map pair as positive samples and cross-pair combinations as negatives. However, due to the large visual difference between RGB images and depth maps, pushing these features far away has no effect on the uniformity purpose, *i.e.* , preserving maximal information [138]. Therefore, instead of creating positive pairs based on different modalities, we consider a way of regulating their feature distances in a closer subspace by using their fusion. Our attention-based fusion creates positive pairs for the RGB image by activating the RGB feature map based on the attention from the depth map, thereby avoiding a large discrepancy with other modalities.

Specifically, for each modality, we create positive pairs and negative pairs based on the augmentations. We define $\mathrm{T}_{\mathrm{RGB}}(\cdot)$, $\mathrm{G}_{\mathrm{RGB}}(\cdot)$, $\mathrm{T}_{\mathrm{DM}}(\cdot)$ and $\mathrm{G}_{\mathrm{DM}}(\cdot)$ as the texture and geometric augmentations of the RGB image and depth map, respectively. Texture augmentations do not affect the labels, *i.e.* , the hand pose, while geometric augmentations require the labels or hand poses to be adjusted accordingly. Based on these augmentations, we obtain the positive pairs $(\mathbf{x}, \mathrm{T}_{\mathrm{RGB}}(\mathbf{x}))$ or $(\mathrm{G}_{\mathrm{RGB}}(\mathbf{x}), \mathrm{T}_{\mathrm{RGB}}(\mathrm{G}_{\mathrm{RGB}}(\mathbf{x})))$ for the RGB image $\mathbf{x}$. Similarly, we create positive pairs for the depth map.

As shown in Fig. 7.3(a)-(c), we augment the data for the RGB branch, DM branch and fusion branch. We then use two fully-connected layers after the encoders to obtain 128-

dimension normalized latent features $\boldsymbol{z}$, $\boldsymbol{z}^+$ for contrastive learning. After that, we adopt the normalized temperature-scaled cross-entropy (NT-Xent) loss as below:

$$\eta(\boldsymbol{z}, \boldsymbol{z}^+) = -\sum_{i=1}^{B} \log \frac{\exp\left(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_i^+)/\tau\right)}{\sum_{k=1}^{B} \mathbb{1}_{[k \neq i]} (\exp\left(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau\right) + \exp\left(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k^+)/\tau\right))}, \tag{7.3}$$

where B is the batch size and $\boldsymbol{z}$ and $\boldsymbol{z}^+$ comprise a positive pair of the same sample. The temperature is set to τ=0.5, $\mathrm{sim}(\cdot,\cdot)$ is the cosine similarity, and $\mathbb{1}$ is the indicator function. Based on NT-Xent loss, we define the contrastive loss for RGB, depth map and fusion as:

$$\begin{aligned} \mathcal{L}_c^{\text{RGB}} &= \eta(\boldsymbol{z}_{RGB}, \boldsymbol{z}_{RGB}^+) \\ \mathcal{L}_c^{\text{DM}} &= \eta(\boldsymbol{z}_{DM}, \boldsymbol{z}_{DM}^+) \\ \mathcal{L}_c^{\text{Fusion}} &= \eta(\boldsymbol{z}_{RGB}, \boldsymbol{z}_F) + \eta(\boldsymbol{z}_{RGB}^+, \boldsymbol{z}_F^+). \end{aligned} \tag{7.4}$$

As shown in Fig. 7.3(d), the final multi-modal contrastive loss is

$$\mathcal{L}_c = \mathcal{L}_c^{\text{RGB}} + \mathcal{L}_c^{\text{DM}} + \mathcal{L}_c^{\text{Fusion}}. \tag{7.5}$$

7.3.4.2 Alignment with Attention.

Using only monocular RGB images as input increases the learning difficulty, since RGB images have only 2D information and suffer from depth ambiguities. As RGB image-depth map pairs have pixel-level correspondences, we propose aligning their feature maps with an attention-based fusion and a shared encoder. The alignment makes it easier for the RGB encoder to perform cross-modal learning and capture shared visual cues. As shown in Fig. 7.2 (c), the attention derived from the depth map propagates the dependency relationship learned in the depth branch and provides accurate attention guidance for the features of the RGB image. Specifically, we conduct multi-modal supervision for RGB images, depth maps and their fusion simultaneously, as shown in Fig. 7.3 (d). The final multi-modal supervised loss is

$$\mathcal{L}_s = M(\mathbf{y}_{RGB}, \mathbf{y}_{gt}) + M(\mathbf{y}_{DM}, \mathbf{y}_{gt}) + M(\mathbf{y}_F, \mathbf{y}_{gt}), \tag{7.6}$$

where $\mathbf{y}_{RGB}, \mathbf{y}_{DM}, \mathbf{y}_F$ are the predictions of the RGB, depth map and fusion branches, respectively and M is the multi-modal supervised loss defined in Eq. 7.1.

Note that a stop-gradient operation is added to stop the back propagation from the fusion branch in Fig. 7.3(c) to prevent inaccurate RGB features from degenerating the attention. This is because the fusion alone is insufficient to fully discard distractor information that may be present in the features. Overall, we pre-train the model on synthetic data with the following objective function and a hyper-parameter λ_c:

$$\mathcal{L}_{pretrain} = \mathcal{L}_s + \lambda_c \mathcal{L}_c. \tag{7.7}$$

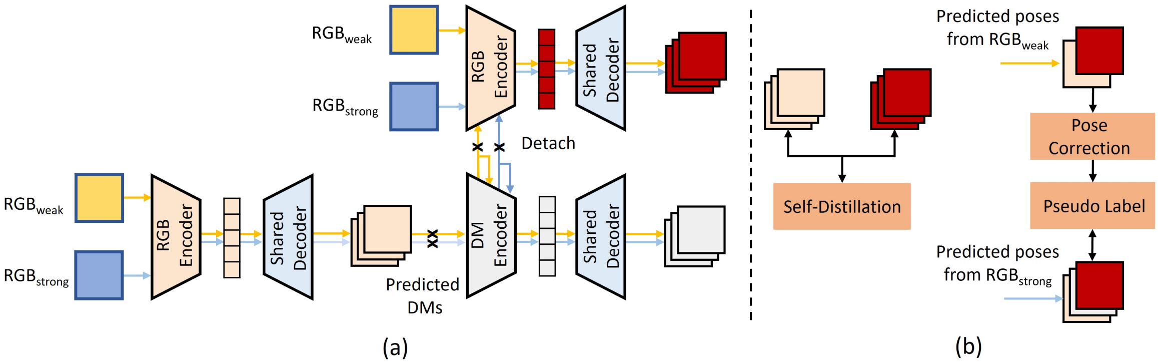


Figure 7.4: (a) The architecture and (b) overall losses during fine-tuning. The yellow and blue arrows show the forward of a weakly-strongly augmented image pair.

7.3.5 Fine-tuning with Noisy Pseudo-Labels

Assuming that we have the pre-trained network based on Sec. 7.3.4, we go on to explore fine-tuning so that the model can generalize to real-world data $\mathcal{D}^r$. As we are fine-tuning on unlabelled real-world data, we do not have multi-modal labels. As such, we rely on pseudo-labels generated by the pre-trained model, specifically depth maps and poses. However, the pseudo-labels are inevitably noisy and used naively, are even detrimental to learning as the model will over-fit to the noise. We therefore integrate pose correction and self-distillation into the proposed dual-modality network to exploit noisy labels. The architecture and overall losses during fine-tuning are shown in Fig. 7.4(a)-(b).

7.3.5.1 Pseudo-Labelling.

To learn discriminative features conditioned on real data, a straightforward approach is to employ a pseudo-label with a high confidence. Therefore, we adopt the pseudo-labelling strategy and enforce a modality agreement on the label assignment to realize consistency. Directly training with (over-confident) pseudo-labels tends to deteriorate the model [2]. We mitigate this with a pose correction step, *i.e.* , we correct 3D poses with rectifications on bone lengths and joint angles to guarantee the biomechanical feasibility of the hand poses. Following [110], we build a local coordinate system for the given poses and use a greedy approximation to correct the given poses based on the hand's kinematic chain. Our correction is inspired by [153], but beyond their work which focuses only on bone lengths, we also take joint angles into account. During the approximation, joint angles that exceed a valid interval are rectified. Our pose correction guarantees biomechanical feasibility of the corrected hand pose with respect to both the bone lengths and the joint angles.

Moreover, to improve the stability of the fine-tuning, we follow a self-paced strategy [151, 153] and gradually take the refined predictions from weakly augmented input (RGB_{weak}) to supervise strongly augmented input (RGB_{strong}). Here, weak augmentations refer to small rotations and translations, while strong augmentation includes larger rotations and translations as well as image scaling. The usage of the pose correction and self-paced strategy is shown in Fig. 7.4 (b).

Based on the observation that the RGB and fusion branches perform better on real data than the depth branch, we generate pseudo-labels $\boldsymbol{r}$ by averaging the poses from the weakly augmented images before and after correction in the RGB and fusion branches. The confidence of a pseudo-label is determined by the variance. Then, we use the pseudo-labels to supervise the poses from the strongly augmented images $\mathbf{p}_{strong}$ in all branches, as below

$$\mathcal{L}_l = \mathbb{1}(\mathcal{C}(\boldsymbol{r}) \leq \varepsilon)\ell(\mathbf{p}_{strong}, \boldsymbol{r}), \tag{7.8}$$

where $\mathcal{C}(\cdot)$ provides the confidence of the pseudo-labels. We select only samples with a confidence larger than ε for further training.

7.3.5.2 Self-Distillation.

Pseudo-labels are inevitably noisy and may change dramatically during fine-tuning, all of which hurts model performance [66]. To address noisy pseudo-labels, our goal is to gradually improve pseudo-labels instead of replacing them dramatically. As our fusion branch of the dual-modality network enable features of the RGB branch to be activated in more relevant regions, it can be considered a denoised RGB branch with the possibility to output more refined predictions. Therefore, we can by design take the predicted depth maps from the RGB branch as an input to the fusion branch and construct a self-distillation structure [66, 57] as shown in Fig.7.4 (a). By encouraging the refined prediction to be consistent with its past prediction, we distill the knowledge to obtain a softer prediction, and generate a pseudo-label accordingly. This achieves our purpose of improving pseudo-labels gradually. Concretely, we apply consistency to the outputs by using multi-modal supervised loss, as below

$$\mathcal{L}_f = M(\mathbf{y}_{RGB}, \mathbf{y}_F), \tag{7.9}$$

where $\mathbf{y}_{RGB}$ and $\mathbf{y}_F$ are the predictions of the RGB branch and the fusion branch, respectively, during fine-tuning.

Overall, we fine-tune the pre-trained model using RGB images of real data based on self-distillation $\mathcal{L}_f$ and pseudo-labelling $\mathcal{L}_l$, together with supervision from synthetic data $\mathcal{L}_s$. The overall objective of this stage with hyper-parameters λ_f and λ_l is as follows:

$$\mathcal{L}_{finetune} = \mathcal{L}_s + \lambda_f \mathcal{L}_f + \lambda_l \mathcal{L}_l. \tag{7.10}$$

7.4 Experiments

7.4.1 Datasets & Evaluation

For the model training, we use the synthetic RHD [178], and four real-world hand datasets: STB [167], FreiHAND [179], H3D [173], and HSD [161]. **RHD** is a large-scale synthetic hand dataset containing 20 characters performing 39 actions. **STB** has 12 video sequences recording finger counting or random poses against 6 different backgrounds, with a total of

15k frames for training and 3k frames for testing. **FreiHAND** is a challenging hand dataset with 130k training images and 4k testing images. The training set comprises 32,560 distinct poses, each with 4 backgrounds, including green screen. Unlike STB and RHD, this dataset features severe object occlusions. In **H3D**, we consider the subset of one-handed gestures for our experiments, which comprises 11k training data and 2k testing data. **HSD** is a newly released hand dataset features 4 actors from 4 camera views. It has 42k images for training and 42k images for testing.

We evaluate 2D keypoint detection with the percentage of correct keypoints (PCK) and regard an estimation as correct when its distance to the ground truth is within 0.05 of the output size. In the 3D keypoint estimation task, the mean end-point-error (EPE) is used to evaluate pose accuracy.

7.4.2 Implementation Details

We adapt ResNet-101 initialized from ImageNet as our backbone network. The input data has a fixed resolution of 256×256, while the resolution for the output is 64×64. In our experiments, the model is first trained with synthetic data using the Adam optimizer with momentum of (0.9, 0.99). The initial learning rate is set to 2.5e-4, and is decreased by a factor of 0.1 after 40 epochs and 50 epochs. We then use synthetic and real data to jointly train the model with a decayed learning rate of 2.5e-5 for 6 epochs. We set the batch size to 140 for pre-training and 20 for fine-tuning. We set the hyper-parameters of Eqs. 7.1 and 7.7-7.10 empirically, with $\lambda_{\mathbf{m}} = 100$, $\lambda_{\mathbf{d}} = 50$, $\lambda_c = 0.1$, $\lambda_f = 0.2$, $\lambda_l = 1$ and $\varepsilon = 1.5$.

In our experiments, RHD provides the RGB hand images, depth maps, segmentation masks and 3D annotations; the real-world datasets provides the RGB hand images and the 3D hand templates for training, as in [153]. Our default setting is pre-training on RHD and fine-tuning with a single real-world dataset's training data and report results on the evaluation data. For fine-tuning on FreiHAND and HSD, we select only a subset of the training to match the size of the STB training set for convenience.

The empirical results of the compared methods are taken directly from the corresponding papers if available; otherwise, they are generated based on officially released code.

7.4.3 Augmentation

Here, we introduce the details of different augmentation strategies for contrastive learning and self-paced learning.

Contrastive Learning. We define $\mathrm{T}_{\mathrm{RGB}}(\cdot)$, $\mathrm{G}_{\mathrm{RGB}}(\cdot)$, $\mathrm{T}_{\mathrm{DM}}(\cdot)$ and $\mathrm{G}_{\mathrm{DM}}(\cdot)$ as the texture and geometric augmentations of the RGB image and depth map, respectively. Texture augmentations do not affect the labels, *i.e.* , the hand poses, while geometric augmentations require the labels or hand poses to be adjusted accordingly. We list the details as below:

- $\mathrm{T}_{\mathrm{RGB}}(\cdot)$ consists of colour jitter, grey-scale and random erasure.
- $\mathrm{G}_{\mathrm{RGB}}(\cdot)$ consists of a rotation of [-180°,180°], scale of [0.8,1] and translation of [-20,20] pixels.

PCK@0.05	STB	FreiHAND	H3D	HSD
Baseline	0.547	0.511	0.555	0.515
RegDA [53]	0.613	0.622	0.720	0.601
CC-SSL [80]	0.655	0.631	0.717	0.602
AnimalDA [66]	0.631	0.629	0.676	0.640
SemiHand [153]	0.668	0.564	0.672	0.563
Ours	**0.775**	**0.658**	**0.749**	**0.689**
Improvement	↑16.0%	↑4.3%	↑4.0%	↑7.7%

(a) 2D Keypoint SOTA comparison

EPE(mm)	STB	FreiHAND	H3D	HSD
Baseline	19.66	21.56	27.77	21.21
SemiHand [153]	14.60	19.33	19.19	19.75
Ours	**11.99**	**15.61**	**17.08**	**16.45**
Improvement	↑17.9%	↑19.2%	↑11.0%	↑16.7%

(b) 3D Keypoint SOTA comparison

Table 7.1: (a) The performance comparisons for 2D keypoint detection; (b) the performance comparisons for 3D keypoint estimation. The relative performance boost between 1st and 2nd best methods can be seen in *Improvement*. Our approach brings consistent improvements over previous state-of-the-art methods. **Bold** numbers indicate the best performance.

- $T_{\text{DM}}(\cdot)$ consists of random erasure, salt and pepper noise.
- $G_{\text{DM}}(\cdot)$ consists of a rotation of [-180°,180°], scale of [0.8,1] and translation of [-20,20] pixels.

Self-Paced Learning. We define weak augmentation and strong augmentation for RGB images. Strong augmentation is similar to augmentations used in other supervised learning methods. It consists of colour jitter, grey-scale, random erasure, rotation of [-180°,180°], scale of [0.8,1] and translation of [-20,20] pixels. In contrast, weak augmentation refers to less intensity of rotation and translation for augmentation. It includes translation of [-8,8] pixels and rotation of either [-2°,2°] or 90°, 180° and 270°.

7.4.4 2D Keypoint Detection

As shown in Table 7.1 (a), we compare our approach with state-of-the-art methods [53, 80, 66, 153] for 2D keypoint detection tasks using the metric PCK@0.05. Notably, our method consistently achieves the best results for the four benchmarks, surpassing the second-best method by a large margin, which can be seen in *Improvement*. All compared methods perform better than the baseline that is merely trained with RHD supervision. In particular, we observe that RegDA works better on H3D than the others, demonstrating that it can more easily detect wrong predictions for the dataset with a simple background. Meanwhile, compared with SemiHand, which is the most related to our work, we significantly improve the performance for FreiHAND and HSD, at **16.6%** and **22.4%**, respectively. This further verifies the effectiveness of our proposed method.

7.4.5 3D Keypoint Estimation

SemiHand [153] is the only published work on semi-supervised cross-domain 3D keypoint estimation. Table 7.1 (b) shows that both SemiHand and our approach outperform the baseline by a large margin. However, compared with SemiHand, our method further decreases EPE from 2.1 to 3.7mm. The decrease is larger for the datasets with complex backgrounds

Method	STB	FreiHAND	H3D	HSD
Baseline	19.66	21.56	27.77	21.21
+ $\mathcal{L}_c$	16.37	19.19	25.94	19.62
+ $\mathcal{L}_l$	12.59	16.27	17.45	16.91
+ $\mathcal{L}_f$	**11.99**	**15.61**	**17.08**	**16.45**

(a) Component Ablations

Positive pairs	STB	FreiHAND	H3D	HSD
None	19.66	21.56	27.77	21.21
RGB only	17.14	19.76	26.83	20.01
RGB & DM	17.32	19.95	26.33	20.40
RGB & DM*	18.01	19.49	26.61	19.71
w/out SG	17.22	19.55	26.01	20.17
w/ SG	**16.37**	**19.19**	**25.94**	**19.62**

(b) Positive Pair Ablations

Table 7.2: (a) Ablation study for the components of our approach. Adding (+) the components incrementally improves performance. (b) Multi-modal contrastive learning with different positive pairs. RGB only: training on $\mathcal{L}_c^{RGB}$; RGB & DM: training on $\mathcal{L}_c^{RGB}$ and $\mathcal{L}_c^{DM}$; RGB & DM*: training on $\mathcal{L}_c^{RGB}$, $\mathcal{L}_c^{DM}$ and $\eta(\boldsymbol{z}_{RGB}, \boldsymbol{z}_{DM})$; w/ and w/out SG: denotes training on $\mathcal{L}_c$ with and without the stop-gradient operation respectively. Our full model ("w/ SG") correctly leverages the depth map modality and generates better representations for the cross-domain dataset. **Bold** indicates the best performance.

and more camera views (*i.e.* , FreiHAND and HSD) confirming our aim of forcing the model to focus on semantically meaningful areas.

7.4.6 Ablation Study

Model Components. Table 7.2 (a) outlines the contributions of multi-modal contrastive learning (+$\mathcal{L}_c$), pseudo-labelling (+$\mathcal{L}_l$) and self-distillation (+$\mathcal{L}_f$) as they are incrementally added to the model. Each component successively decreases the mean EPE for all four datasets.

Multi-modal Contrastive Learning. Eq. 7.5 has three loss terms, representing the contrastive learning under the RGB, depth map, and fusion branches, respectively. Each branch is trained with specific positive pairs. Table 7.2 (b) shows the contrastive learning with different combinations of positive pairs. "RGB only", in which we train with only the RGB branch with RGB contrastive pairs improves the mean EPE over "none", in which we train the model without contrastive learning. Yet if we also apply a contrastive loss to the depth map, *i.e.* , "RGB & DM" there is a slight performance drop on some datasets. To investigate the drop, we are inspired by [125] to apply a contrastive loss to each RGB and depth map pair, *i.e.* , $\eta(\boldsymbol{z}_{RGB}, \boldsymbol{z}_{DM})$, denoted by "RGB & DM*". Yet this still deteriorates on STB compared to "RGB only". We conjecture that the large visual differences across the RGB and depth map already make the latent features of the negative pair between RGB and depth map distant so any uniformity effects from the negative samples are invalid.

However, this problem can be repaired, and even an extra boost can be seen, when training together with the designed multi-modal contrastive learning ("w/ SG" setting), especially on STB, whereby the accuracy can reach 16.37 mm which is a 16.7% increase over the baseline. As such, we demonstrate that the proposed attention-based feature fusion makes multi-modal contrastive learning not only feasible but also brings about improvements, despite the large discrepancy between the RGB image and depth map. Note that this does require stopping the back propagation of the gradient ("w/ SG") to the attention module when training the

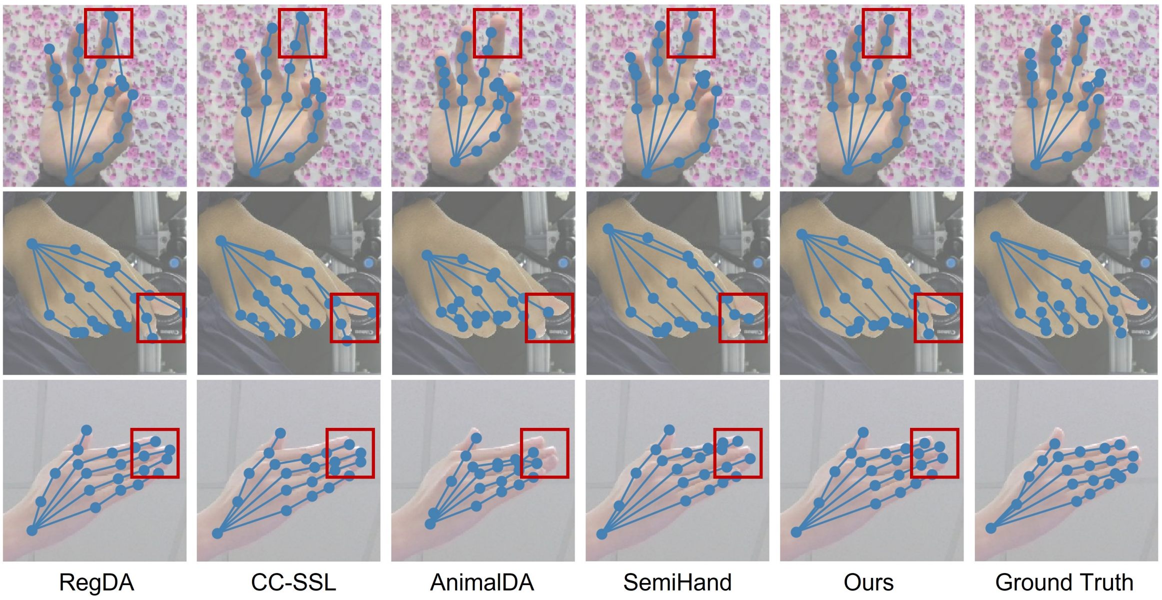


Figure 7.5: 2D pose visualization on STB, H3D and HSD. We compare our method with four state-of-the-art methods and highlight the differences between the predictions and the ground truth poses with red boxes. Figure best viewed in colour.

fusion branch. This can be verified by allowing the gradient to propagate, *i.e.* , "w/out SG", versus "w/ SG". This confirms our hypothesis that the attention from the DM encoder makes it easier for the RGB encoder to capture the 2.5D information and improve cross-dataset performance. On the other hand, the abundant non-informative features captured in the RGB modality impedes the learning of the attention module.

7.4.7 Qualitative Results

As illustrated in Fig. 7.5, we show three qualitative examples of the 2D pose detection generated by the compared methods and our method on STB, H3D and HSD. We can see that our predictions are most similar to the ground truth, while the other methods show poor performance with wrong predictions or scale errors, especially in the finger tips.

We also visualize how the predicted depth maps and the attention weights from conv1 layers have changed after conducting the model fine-tuning, as shown in Figs. 7.1 and 7.6. The results show analysis experiments based on two FreiHAND samples with object occlusion and one STB sample with extreme lighting. In those cases, the pre-trained model can only estimate the depth values for few areas, *e.g.* , some fingers. However, after model fine-tuning, we obtain better depth map predictions for complete hands. Correspondingly, we can see that there are some larger attention weights, which were not activated before fine-tuning.

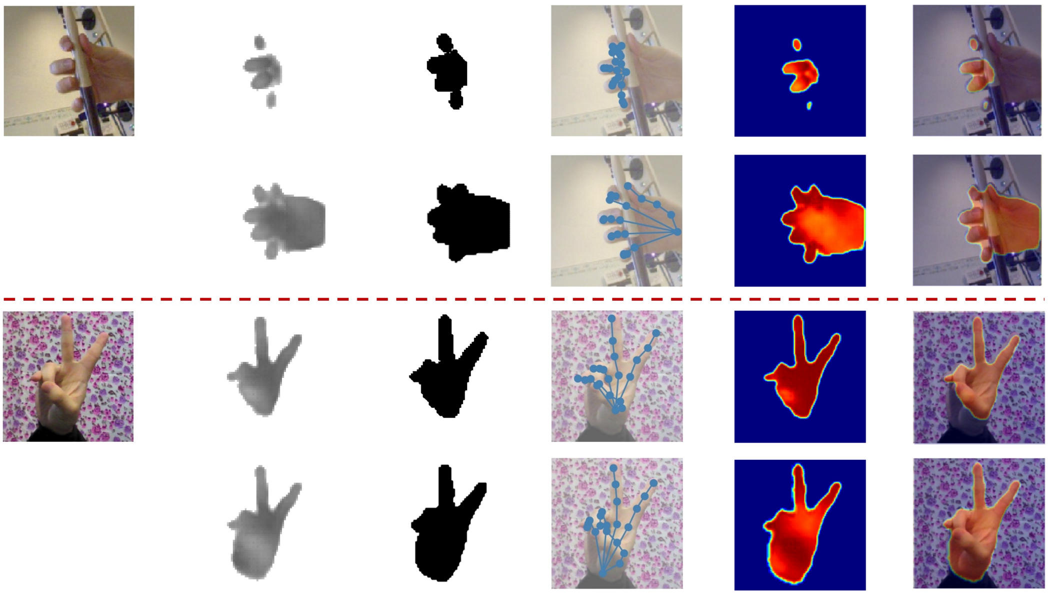


Figure 7.6: Visualisation of two examples before fine-tuning (first row) and after fine-tuning (second row). From left to right: RGB images, multi-modal predictions (depth maps, segmentation masks, poses), attention weights from depth maps and attention weights on RGB image. Figure best viewed in colour.

7.5 Conclusion

We propose a dual-modality network to address the cross-domain pose estimation problem in a semi-supervised setting. By leveraging the multiple data modalities of synthetic data, we explore multi-modal learning during pre-training, including multi-modal contrastive learning and feature alignment. This enables the RGB encoder to create a well-structured dimensional space and better capture the most related features regarding the 2.5D information and semantic meanings. During fine-tuning, we explore pose correction and self-distillation based on our proposed dual-modality network and provide a unified fine-tuning scheme for real data with noisy pseudo-labels. Our experiments show that our approach significantly outperforms state-of-the-art methods on four datasets. In the future, we intend to dive deeper into contrastive learning and self-distillation for semi-supervised hand pose estimation. Also, we will try to make the model lightweight by redesigning the depth map branch.

Chapter 8

Conclusions

Contents

8.1 Summary 105

8.2 Discussion and Future Work 107

8.2.1 Multi-Modal Data 107

8.2.2 Disentangled Representation 108

8.2.3 Synthetic Data 108

8.2.4 Interpretation Technique 109

8.2.5 Real-world Application 109

8.1 Summary

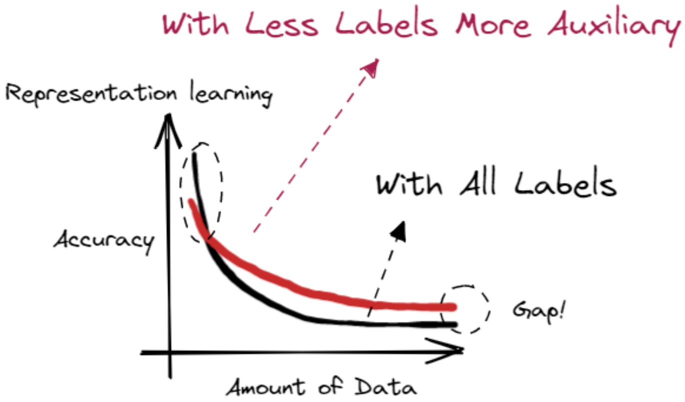


Figure 8.1: Our goal is to utilize more auxiliary information and less annotation information for model training. The exploration is in two directions. First, with auxiliary information, we aim to achieve better representations by means of representation learning. Second, we would like to exploit auxiliary information for weakly-/semi- supervise learning and hence reduce the reliance on annotation. However, until now, even with diverse auxiliary information, the accuracy still has not been close to that in supervised learning.

The overall goal of this dissertation is to explore auxiliary information to aid representation learning and relieve the burden of annotation for 3D hand pose estimation from single

RGB images. As shown in Fig. 8.1, with respect to the modal accuracy and the amount of training data, we can use auxiliary information for representation learning to get better representations and hence improve the performance. Moreover, we would like to train with less labelled data but more data with accessible auxiliary information to approach the performance of training with all labelled data. Towards this goal, we explore three auxiliary information with different strategies in Chapters 4-7. In the following, we summarize our contributions.

In Chapter 4, we explore the image factors of variation among images as auxiliary information. Inspired by image render processing, we present a VAE-based method for learning disentangled representations. With the disentangled representations, we can synthesize RGB images or generate hand poses with full control over factors of variation. This also provides the potential to better analyze these factors of variation in both latent and appearance spaces. Interestingly, our proposed model enables image factors as weak labels and evaluation in a weakly-supervised setting.

In Chapter 5, we introduce different modalities as auxiliary information. We first formulate RGB-based hand pose estimation as a multi-modal learning, cross-modal inference problem. As such, we have a flexible framework to incorporate different modalities. Especially, we explore a non-conventional modality, point clouds, for learning the latent hand space and show its superiority as auxiliary information for RGB-based hand pose estimation. Moreover, we present the product of Gaussian expert operation to align the latent space. The product of Gaussian expert alignment is flexible and enables the model to use different data pairs as input. Due to the flexibility of our proposed multi-modal framework, different modalities especially point clouds can be used as weak labels to support the training process.

In Chapter 6, we explore synthetic data as auxiliary information to aid the training of unlabelled real-world data. We propose the first cross-domain semi-supervised framework for 3D hand pose estimation. Directly applying the existing semi-supervised method is non-trivial because pose estimation is a regression problem that critically depends on spatial information. By design, we propose a template-based hand pose correction module to refine the predictions. Moreover, to stabilize the training, we propose data augmentation of differing difficulties.

In Chapter 7, we further combine both synthetic data and multi-modal data as auxiliary information. We introduce a novel dual-modality network that learns from two modalities (RGB images and depth maps) during pre-training but is applicable to only monocular RGB inputs during fine-tuning and inference. During pre-training, we explore multi-modal learning, including multi-modal contrastive learning and multi-modal feature alignment. During fine-tuning, we explore pose correction and self-distillation based on our proposed dual-modality network, and provide an effective fine-tuning scheme for real-world data with noisy pseudo-labels.

8.2 Discussion and Future Work

Even we investigate different auxiliary information, the gap in Fig. 8.1 still exists. This encourages us to further explore the use of auxiliary information. In the following, we introduce challenges we still faced in our research direction and also emphasize the interpretation techniques and real-world application for hand pose estimation.

8.2.1 Multi-Modal Data

There is still a need to explore different modalities and emphasize their own characteristic for specific application scenarios. Recently, 3D surface representations of hand have drawn much attention. However, so far, most works only focus on predicting hand mesh from single RGB input. Existing mesh works either directly [37, 135, 78], or indirectly [9, 179, 59] through the use of parametric models like MANO [100] for the hand. I would like to break this limit by introducing more hand representations. Here, we highlight the use of UV maps and implicit functions for 3D surfaces and 3D poses.

- Different modalities have their own advantages and hence it is worthy of exploring modalities for specific application scenarios based on their advantages. Taking UV maps for example, a UV coordinate map can be easily augmented with their built-in correspondences through additional channels of information, such as surface texture and regions of object contact. This creates a natural connection between the 3D hand shape, its appearance, and its interactions with objects in a seamless representation space. As such, we emphasize the potential application of UV maps for pose estimation and 3D reconstruction under the scenario of hand objection interaction.

- Besides the type of input modalities, how to utilize 3D representations for hand pose estimation is also essential. This starts to be difficult as recent 3D representations become more effective yet more complex. This leads us to think about - how shall we build the skeleton or even the muscles for arbitrary 3D representations in one network? Implicit functions like implicit neural representations [24] and neural radiance fields [77] have shown great promise as they produce continuous reconstructions. It would be interesting to get a "rigged" implicit function for 3D surfaces and 3D poses.

- Leveraging common cues from multi-modal data can enhance RGB-based hand pose estimation. We propose to achieve this by explicitly aligning the latent spaces [155], the attention and the low-dimensional subspace [70]. However, it is still a promising direction to explore more different "alignment" for multi-modal data for representation learning. We may explore more alignment in pixel-level features by encouraging the corresponding features between different modalities to be close so that getting better representations. Note that this is applicable for 2D and 3D modalities, as we can connect 3D points with 2D pixels using the camera projection.

8.2.2 Disentangled Representation

Disentangled representation learning can learn the task-relevant features without labels or with weak labels. We continue exploring disentangled representation learning for label-efficient pose estimation and improve the generalization of deep models.

- To learn a disentangled and controllable latent representation, it is important to construct data pairs with shared factors of variation. One easy solution is to utilize synthetic image like [27]. Also, we can control data record like fixing one of the factors of variation. For a practical application in the real world, it is interesting to develop an indoor environment to record such data. With both real-world and synthetic data, we may have disentangled representations and generate realistic images.

- Disentangled representation learning has not been a mainstream framework to extract task-relevant features for downstream tasks like contrastive learning. It would be interesting if we can integrate disentangled representation with few-shot learning to improve the generalization of deep models. Also, we should explore more cheap weak labels like domain labels for disentangled representation learning to make it more practical in the real world applications.

- Contrastive self-supervised representation learning with unlabelled real-world data has achieved great success. However, constructing positive pairs seems still hard as simple augmentations at most provide texture or geometric specific features. In contrast, data with shared image factors are more common and easy to provide "positive image factors" even not "positive images". In this case, we would like to combine contrastive learning with disentangled representation learning to address the data with shared image factors. The combination will provide a novel representation learning framework.

8.2.3 Synthetic Data

Training models with synthetic data can bypass label scarcity, but such models generalize poorly to real-world settings due to the domain gap between synthetic data and real-world data. It is still promising to "transfer" the information from synthetic data to real-world data effectively. In this case, we continue exploring to reduce the gap and improve the representations based on synthetic data.

- The quality of synthetic hand data is still far from enough and this will high limit its development. We should borrow the tools and ideas from related research fields like face [143] to synthesize better hand images.

- Due to the flexibility of synthetic data, we highlight the potential research direction that combining synthetic data with other auxiliary information like data with shared image factors or multi-modal data, to get better representations. We will continue exploring the "synthetic data-guided" representation learning.

- During cross-domain semi-supervised learning, not all synthetic samples or real-world samples are equally informative during training. As such, it is potential to explore strategies like curriculum training or self-paced training, which finds the informative samples in different training stages, to accelerate the training speed and get better performance. Also, this sample selection procedure may be a key to stabilize the training.
- Pseudo-labelling are effective way to bridge the domain gap. However inevitably, the pseudo-labels are noisy. To reduce the influence of noise, we should correct the pseudo-labels. Even we propose pose correction in Chapters 6 and 7, we still try to find more efficient (*e.g.* one-shot) correction methods based on more hand representations like hand surfaces or hand heatmaps.

8.2.4 Interpretation Technique

By means of different auxiliary information for hand pose estimation, we improve the representations of models and exploit those information in different supervised settings. However, the usage of those auxiliary information is still hard to interpret and limited work try to reveal the difference apart from the accuracy. Therefore, there is a need of interpretation techniques for hand pose estimation to understand the behaviour of models and interpret the rationale or details behind pipelines.

- There exists large amount of methods for hand pose estimation. We should systematically analyze different pose estimation pipelines. This will further provide insights to understand the influence of different components and frameworks and assist diagnosing models.
- Pose estimation frameworks exhibit flexible target modalities, *i.e.* , heatmap or coordinate in 2D or 3D spaces. Therefore, We should compare and analyze different target modalities in the same metric space.
- Pose estimation including diverse topics like human pose estimation, hand pose estimation, animal pose estimation and head pose estimation. Each of them has own characteristics. It is favorable to analyze their similarities and differences.

8.2.5 Real-world Application

Hand pose estimation plays an important role for robotics, action recognition and immersive interaction, and has broad real-world application prospects. Instead of single hand pose estimation, most existing works turn to more challenging scenarios like hand object interaction or hand hand interaction. However, they still limit their focus on the performance of public benchmarks and overlook its application in real-world scenes. Differently, I highlight two potential research directions for real-world hand pose estimation.

- To apply hand pose estimation on edge devices like AR/VR or mobile devices, we aim to explore lightweight network architectures and network compression. This will give the customer a more comfortable and convenient immersive interactive experience.

- Given a pre-trained model, we target transferring the model to work better for new scenes or people quickly. This will ensure the accuracy of pre-trained models in different scenarios.

Bibliography

[1] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *ECCVW*, pages 0–0, 2018.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*, pages 1–8. IEEE, 2020.

[3] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *CVPR*. IEEE, 2003.

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *CVPR*, pages 1067–1076, 2019.

[5] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *TPAMI*, 41(2):423–443, 2019.

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[7] Abhishake Kumar Bojja, Franziska Mueller, Sri Raghu Malireddi, Markus Oberweger, Vincent Lepetit, Christian Theobalt, Kwang Moo Yi, and Andrea Tagliasacchi. Handseg: An automatically labeled dataset for hand segmentation from depth images. In *CRV*, pages 151–158. IEEE, 2019.

[8] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI*, 2018.

[9] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, pages 10843–10852, 2019.

[10] Giuseppe Caggianese, Nicola Capece, Ugo Erra, Luigi Gallo, and Michele Rinaldi. Freehand-steering locomotion techniques for immersive virtual environments: A comparative evaluation. *IJHCI*, 36(18):1734–1755, 2020.

[11] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*, pages 666–682, 2018.

[12] Jinkun Cao, Hongyang Tang, Hao-Shu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai. Cross-domain adaptation for animal pose estimation. In *ICCV*, pages 9498–9507, 2019.

[13] Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

[14] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, Wei Fan, and Xiaohui Xie. Dggan: Depth-image guided generative adversarial networks for disentangling rgb and depth images in 3d hand pose estimation. In *WACV*, pages 411–419, 2020.

[15] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In *WACV*, pages 1050–1059, 2021.

[16] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Hui Tang, Yufan Xue, Yen-Yu Lin, Xiaohui Xie, and Wei Fan. Tagan: Tonality aligned generative adversarial networks for realistic hand pose synthesis. In *BMVC*, 2019.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[18] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[19] Xinghao Chen, Guijin Wang, Cairong Zhang, Tae-Kyun Kim, and Xiangyang Ji. Shpr-net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6:43425–43439, 2018.

[20] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *ICCV*, pages 6961–6970, 2019.

[21] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *CVPR*, pages 10451–10460, 2021.

[22] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018.

[23] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *ICML*, pages 1779–1788. PMLR, 2020.

[24] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*, pages 6970–6981, 2020.

[25] Rodrigo de Bem, Arnab Ghosh, Thalaiyasingam Ajanthan, Ondrej Miksik, N Siddharth, and Philip HS Torr. A semi-supervised deep generative model for human body analysis. In *ECCVW*, 2018.

[26] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *NeurIPS*, 29, 2016.

[27] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, pages 5154–5163, 2020.

[28] Endri Dibra, Silvan Melchior, Ali Balkis, Thomas Wolf, Cengiz Oztireli, and Markus Gross. Monocular rgb hand pose inference from unsupervised refinable nets. In *CVPRW*, pages 1075–1085, 2018.

[29] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Crossinfonet: Multi-task information sharing based hand pose estimation. In *CVPR*, pages 9896–9905, 2019.

[30] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. *arXiv preprint arXiv:2007.04646*, 2020.

[31] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020.

[32] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *ECCV*, pages 738–751. Springer, 2012.

[33] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR*, pages 409–419, 2018.

[34] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, pages 8417–8426, 2018.

[35] 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 *CVPR*, pages 3593–3601, 2016.

[36] 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 *CVPR*, pages 1991–2000, 2017.

[37] 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 *CVPR*, pages 10833–10842, 2019.

[38] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *ECCV*, pages 475–491, 2018.

[39] Yunhao Ge, Sami Abu-El-Haija, Gan Xin, and Laurent Itti. Zero-shot synthesis with group-supervised learning. *arXiv preprint arXiv:2009.06586*, 2020.

[40] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[41] Jiajun Gu, Zhiyong Wang, Wanli Ouyang, Jiafeng Li, Li Zhuo, et al. 3d hand pose estimation with disentangled cross-modal latent space. In *WACV*, pages 391–400, 2020.

[42] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *ICIP*, pages 4512–4516. IEEE, 2017.

[43] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, pages 3196–3206, 2020.

[44] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, pages 5138–5147, 2019.

[45] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ToG*, 39(4):87, 2020.

[46] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[48] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual

concepts with a constrained variational framework. In *ICLR*, 2017.

[49] Fuyang Huang, Ailing Zeng, Minhao Liu, Jing Qin, and Qiang Xu. Structure-aware 3d hourglass network for hand pose estimation from single depth image. *arXiv preprint arXiv:1812.10320*, 2018.

[50] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV*, pages 118–134, 2018.

[51] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *CVPR*, pages 5243–5252, 2020.

[52] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *ECCV*, 2018.

[53] Junguang Jiang, Yifei Ji, Ximei Wang, Yufeng Liu, Jianmin Wang, and Mingsheng Long. Regressive domain adaptation for unsupervised keypoint detection. In *CVPR*, 2021.

[54] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, pages 3907–3916, 2018.

[55] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *CVPR*, pages 2540–2548, 2015.

[56] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, volume 33, pages 18661–18673, 2020.

[57] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *ICCV*, pages 6567–6576, 2021.

[58] Oscar Koller, Necati Cihan Camgoz, Hermann Ney, and Richard Bowden. Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *TPAMI*, 42(9):2306–2320, 2019.

[59] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, pages 2252–2261, 2019.

[60] Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. A deep learning approach for analyzing video and skeletal features in sign language recognition. In *IST*, pages 1–6. IEEE, 2018.

[61] Philip Krejov, Andrew Gilbert, and Richard Bowden. Combining discriminative and model based approaches for hand pose estimation. In *FG*, volume 1, pages 1–7. IEEE, 2015.

[62] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *NeurIPS*, 2015.

[63] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, pages 4990–5000, 2020.

[64] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. In *BMVC*, 2019.

[65] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICMLW*, 2013.

[66] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *CVPR*, 2021.

[67] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. In *NeurIPS*, volume 34, 2021.

[68] Moran Li, Yuan Gao, and Nong Sang. Exploiting learnable joint groups for hand pose estimation. *AAAI*, 2021.

[69] Shile Li and Dongheui Lee. Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In *CVPR*, pages 11927–11936, 2019.

[70] Qiuxia Lin, Linlin Yang, and Angela Yao. Dual-modality network for semi-supervised hand pose estimation. In *Submission*, 2022.

[71] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *CVPR*, pages 14687–14697, 2021.

[72] Zhiwei Liu, Xiangyu Zhu, Guosheng Hu, Haiyun Guo, Ming Tang, Zhen Lei, Neil M Robertson, and Jinqiao Wang. Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection. In *CVPR*, pages 3467–3476, 2019.

[73] Adrian Lopez-Rodriguez and Krystian Mikolajczyk. Desc: Domain adaptation for depth estimation via semantic consistency. *arXiv preprint arXiv:2009.01579*, 2020.

[74] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[75] Meysam Madadi, Sergio Escalera, Alex Carruesco, Carlos Andujar, Xavier Baró, and Jordi Gonzàlez. Occlusion aware hand pose recovery from sequences of depth images. In *FG*, 2017.

[76] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *IJCV*, 126(9):942–960, 2018.

[77] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020.

[78] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, pages 752–768. Springer, 2020.

[79] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR*, pages 5079–5088, 2018.

[80] Jiteng Mu, Weichao Qiu, Gregory D Hager, and Alan L Yuille. Learning from synthetic animals. In *CVPR*, pages 12386–12395, 2020.

[81] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, pages 49–59, 2018.

[82] 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. *ToG*, 38(4):1–13, 2019.

[83] Siddharth Narayanaswamy, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *NeurIPS*, 2017.

[84] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.

[85] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3D hand pose estimation. In *ICCVW*, 2017.

[86] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. In *WACV*, 2015.

[87] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, 2011.

[88] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. Evolutionary quasi-random search for hand articulations tracking. In *CVPR*, pages 3422–3429, 2014.

[89] Gaurav Pandey and Ambedkar Dukkipati. Variational methods for conditional multimodal deep learning. In *IJCNN*, 2017.

[90] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*, 2018.

[91] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019.

[92] Stefano Pellegrini, Konrad Schindler, and Daniele Nardi. A generalisation of the icp algorithm for articulated bodies. In *BMVC*, volume 3, page 4, 2008.

[93] Georg Poier, Konstantinos Roditakis, Samuel Schulter, Damien Michel, Horst Bischof, and Antonis A Argyros. Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. In *BMVC*, 2015.

[94] Georg Poier, David Schinagl, and Horst Bischof. Learning pose specific representations by predicting different views. In *CVPR*, pages 60–69, 2018.

[95] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, pages 5171–5180. PMLR, 2019.

[96] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *CVPR*, pages 1106–1113, 2014.

[97] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *CVPR*, pages 4663–4672, 2018.

[98] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv preprint arXiv:1511.06434*, 2015.

[99] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, pages 4119–4128, 2018.

[100] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ToG*, 36(6):1–17, 2017.

[101] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *ICCV*, 2019.

[102] 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 *CHI*, pages 3633–3642, 2015.

[103] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[104] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, pages 1145–1153, 2017.

[105] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[106] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *CVPR*, pages 4150–4158, 2016.

[107] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[108] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. Fingerinput: Capturing expressive single-hand thumb-to-finger microgestures. In *ISS*, pages 177–187, 2018.

[109] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *ICCV*, pages 11230–11239, 2021.

[110] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *ECCV*, 2020.

[111] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, pages 89–98, 2018.

[112] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, pages 3213–3221, 2015.

[113] 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 *ECCV*, 2016.

[114] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*, pages 2456–2463, 2013.

[115] Bjoern Stenger, Paulo RS Mendonça, and Roberto Cipolla. Model-based 3d tracking of an articulated hand. In *CVPR*. IEEE, 2001.

[116] Carsten Stoll, Zachi Karni, Christian Rössl, Hitoshi Yamauchi, and Hans-Peter Seidel. Template deformation for point cloud fitting. In *SIGGRAPH*, pages 27–35, 2006.

[117] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 529–545, 2018.

[118] James S Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan.

Depth-based hand pose estimation: data, methods, and challenges. In *ICCV*, pages 1868–1876, 2015.

[119] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

[120] Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. Challenges in disentangling independent factors of variation. In *arXiv preprint arXiv:1711.02245*, 2017.

[121] 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 *ICCV*, pages 3325–3333, 2015.

[122] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[123] 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. *ToG*, 35(4):1–12, 2016.

[124] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *CVPR*, pages 644–651, 2014.

[125] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794. Springer, 2020.

[126] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ToG*, 35(6):1–11, 2016.

[127] Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. Online generative model personalization for hand tracking. *ToG*, 36(6):243, 2017.

[128] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ToG*, 33(5):169, 2014.

[129] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018.

[130] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*, 118, 2016.

[131] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. In *ICLR*, 2018.

[132] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In *CVPR*, pages 10853–10862, 2019.

[133] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *CVPR*, pages 680–689, 2017.

[134] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *CVPR*, pages 5147–5156, 2018.

[135] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dual grid net: hand

mesh vertex regression from single depth maps. In *ECCV*, pages 442–459. Springer, 2020.

[136] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao. Tag disentangled generative adversarial networks for object image re-rendering. In *IJCAI*, 2017.

[137] Kewen Wang and Xilin Chen. Pmd-net: Privileged modality distillation network for 3d hand pose estimation from a single rgb image. In *BMVC*, 2020.

[138] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939. PMLR, 2020.

[139] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020.

[140] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *BMVC*, 2015.

[141] Wikipedia. American sign language. https://en.wikipedia.org/wiki/American_Sign_Language.

[142] Jan Wöhlke, Shile Li, and Dongheui Lee. Model-based hand pose estimation for generalized hand shape with appearance normalization. In *arXiv preprint arXiv:1807.00898*, 2018.

[143] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone. In *ICCV*, pages 3681–3691, 2021.

[144] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. In *NeurIPS*, 2018.

[145] Xiaokun Wu, Daniel Finnegan, Eamonn O'Neill, and Yong-Liang Yang. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *ECCV*, pages 237–253, 2018.

[146] Ying Wu and Thomas S Huang. View-independent recognition of hand postures. In *CVPR*. IEEE, 2000.

[147] Ying Wu, John Y Lin, and Thomas S Huang. Capturing natural hand articulation. In *ICCV*, page 426. IEEE, 2001.

[148] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, pages 10965–10974, 2019.

[149] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pages 466–481, 2018.

[150] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[151] Rongchang Xie, Chunyu Wang, Wenjun Zeng, and Yizhou Wang. An empirical study of the collapsing problem in semi-supervised 2d human pose estimation. In *ICCV*, pages 11240–11249, 2021.

[152] 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 *ICCV*, pages 793–802, 2019.

[153] Linlin Yang, Shicheng Chen, and Angela Yao. Semihand: Semi-supervised hand pose estimation with consistency. In *ICCV*, 2021.

[154] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. Bihand: Recovering hand mesh with multi-stage bisected hourglass networks. *arXiv preprint arXiv:2008.05079*, 2020.

[155] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. Aligning latent spaces for 3d hand pose estimation. In *ICCV*, pages 2335–2343, 2019.

[156] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *CVPR*, pages 9877–9886, 2019.

[157] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, pages 206–215, 2018.

[158] Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W Tsang, and Jun Sun. Safeguarded dynamic label regression for noisy supervision. In *AAAI*, pages 9103–9110, 2019.

[159] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, pages 346–361. Springer, 2016.

[160] Shi Yin, Shangfei Wang, Xiaoping Chen, Enhong Chen, and Cong Liang. Attentive one-dimensional heatmap regression for facial landmark detection and tracking. In *ACMMM*, pages 538–546, 2020.

[161] Ziwei Yu, Linlin Yang, Shicheng Chen, and Angela Yao. Local and global point cloud reconstruction for 3d hand pose estimation. In *BMVC*, 2021.

[162] Shanxin Yuan, Guillermo Garcia-Hernando, Bjorn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. 3d hand pose estimation: From current achievements to future goals. In *CVPR*, 2018.

[163] Shanxin Yuan, Bjorn Stenger, and Tae-Kyun Kim. Rgb-based 3d hand pose estimation via privileged learning with depth images. In *ICCVW*, 2018.

[164] Shanxin Yuan, Qi Ye, Guillermo Garcia-Hernando, and Tae-Kyun Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv:1707.02237*, 2017.

[165] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017.

[166] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, pages 11197–11206, 2020.

[167] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, pages 982–986. IEEE, 2017.

[168] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *ICCV*, pages 11281–11292, 2021.

[169] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *ICCV*, pages 2354–2364, 2019.

[170] Yumeng Zhang, Li Chen, Yufeng Liu, Wen Zheng, and Jun-Hai Yong. Explicit knowledge distillation for 3d hand pose estimation from monocular rgb. In *BMVC*, 2020.

[171] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, pages 7404–7413. PMLR, 2019.

[172] Long Zhao, Xi Peng, Yuxiao Chen, Mubbasir Kapadia, and Dimitris N Metaxas. Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. In *CVPR*, pages 6528–6537, 2020.

[173] Zhengyi Zhao, Tianyao Wang, Siyu Xia, and Yangang Wang. Hand-3d-studio: A new multi-view system for 3d hand reconstruction. In *ICASSP*, pages 2478–2482. IEEE, 2020.

[174] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, pages 1–15, 2020.

[175] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, pages 5346–5355, 2020.

[176] 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 *ECCV*, pages 501–516, 2018.

[177] Christian Zimmermann, Max Argus, and Thomas Brox. Contrastive representation learning for hand shape estimation. In *GCPR*, pages 250–264. Springer, 2021.

[178] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, pages 4903–4911, 2017.

[179] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, pages 813–822, 2019.