EPFL

Thèse n° 8036

Encoder-Decoder Models for Human Segmentation and Motion Analysis

Présentée le 10 mars 2022

Faculté informatique et communications
Laboratoire de vision par ordinateur
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Isinsu KATIRCIOGLU

Acceptée sur proposition du jury

Dr M. Rajman, président du jury
Prof. P. Fua, Dr M. Salzmann, directeurs de thèse
Prof. L. Agapito, rapporteuse
Prof. P. Favaro, rapporteur
Prof. A. Zamir, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Acknowledgements

There are many people without whom this work would not have been possible. First of all, I would like to thank my supervisors, Professor Pascal Fua and Dr. Mathieu Salzmann for their guidance, the inspiring discussions we had and the opportunity to join the Computer Vision Laboratory at EPFL to pursue research with exceptional people.

I owe special thanks to Prof. Helge Rhodin with whom I worked over three years. His continuous support and the ideas he brought to this work constructed the basis of my work on self-supervised segmentation. He was a great source of motivation during my PhD and I learned a lot from him regarding systematic way of conducting research. I would also like to thank Prof. Vincent Lepetit with whom I exchanged ideas and had fruitful discussions on 3D human body pose estimation at the beginning of my PhD. I would like to thank the members of my thesis committee, Prof. Lourdes Agapito, Prof. Paolo Favaro, Prof. Amir Zamir and Dr. Martin Rajman for accepting to evaluate this thesis and sharing their insightful comments to strengthen my work.

I would like to thank all the present and former members of the CVLab who made the time at EPFL unforgettable. In particular, I would like to thank my dearest friend Sena who has been always there for me during both challenging and joyful times. It's not easy to find a companion like her in life who knows you thoroughly and can truly empathize with you over the setbacks. Therefore, I'm grateful to have such an understanding, compassionate and courageous friend by my side. I enjoyed every moment of our yoga, hiking, skiing, climbing and coffee sessions. The long discussions we had on humankind and research broadened my perspective on many different aspects of life. I would also like to thank Carlos for cultivating such a warm and supportive friendship over the years. The philosophical debates that he initiated during coffee breaks and his enthusiasm for machine learning have inspired me a lot. I also appreciate the time and effort Sena and Carlos put in proof reading this work. I would like to thank Róger, Agata, Pablo and Radhakrishna for the brain stimulating coffee breaks and intellectual conversations during happy hours. I would like to thank Jan, Vidit, Fayez, Frederike, Martin, Nicolas and Eric for the board game nights, dinner parties and the encouragement they provided throughout my PhD. I was fortunate to share my office with Weizhe and Shuxuan with whom I had uncountable laughs and food. I would also like to thank Victor for his contributions to this work. To all my colleagues, Erhan, Anne, Mateusz, Artem, Ksenia, Eduard, Andrii, Kaicheng, Udaranga, Semih, David,

Acknowledgements

Doruk, Leonardo, Joachim, Shaifali, Krzysztof, Andrey, Benoît, Edoardo, Louis, Benoît, thanks for making this experience a great and memorable one. I would also like to thank Ariane for her excellent administrative support and kindness.

I want to thank my community in Lausanne for making this beautiful city home for me. I would like to thank Daniela and Agata for bringing a touch of art to my life and Rafael for sharing many interesting and humorous stories as well as his love for robotics with me. The evenings and trips we organized with this small gang made my time outside research more enjoyable. I'm thankful to Erhan and Ceren for their immense support, the apéros by the lake and the fun discussions we had on life. I'm grateful to have the continuous support and delicious food of Firat, Ayca, Nergiz and Cem who have always cheered me up along the way. I'm very lucky to have crossed paths with Selin who has inspired me a lot with her energetic and confident attitude towards life. The company of Selin and Cagri made my time in Switzerland much more pleasant.

I would also like to express my gratitude towards my childhood friends, Pinar, Ekin and Cosan and my university friends Gokce, Duygu, Ege and Emre who have constantly encouraged me despite the physical distance between us.

I thank the cosmos for introducing Amaury to my life. He has greatly inspired me with his love for learning and enthusiasm for philosophy, psychology, sports and 3D computer vision. Our adventures during hikes, camping, acroyoga, and traveling gave me the most remarkable memories of my life. I learned a lot from our mutual experiences on becoming a more grounded person and accepting the things that are not under our control.

Last, and most importantly, I want to give special thanks to my family for their unconditional love and support. I would like to express my deepest gratitude towards my mother Şenay and my father Bayram who have patiently and dedicatedly offered me the education and opportunities that have made me who I am. They always encouraged me to explore new directions in life and supported my decisions wholeheartedly. I am grateful to my sister Deniz and her husband Onur for being there as true friends and role models as I was growing up. This journey would not have been possible if not for them, and I dedicate this milestone to my family.

Lausanne, January 7, 2022 I. K.

Abstract

Detecting people from 2D images and analyzing their motion in 3D have been long standing computer vision problems central to numerous applications such as autonomous driving and athletic training. Recently, with the availability of large amounts of training data and the advent of deep learning, the performance in human segmentation, 3D human pose prediction has improved significantly. However, these problems remain challenging due to several factors. In this thesis, we decompose the human motion analysis into three sub-tasks; 2D human segmentation, 3D human body pose estimation and 3D human motion forecasting. Our goal is to alleviate the challenges in these problems using various encoder-decoder models.

While supervised detection and segmentation methods achieve impressive accuracy, they generalize poorly to images whose appearance significantly differs from the data they have been trained on. To remedy this, they require overly large amounts of annotated data in domain-specific applications. Therefore, self-supervised detection and segmentation of foreground objects in complex scenes is gaining attention. However, existing self-supervised approaches predominantly rely on restrictive assumptions of appearance and motion. This precludes their use in scenes depicting highly dynamic activities or involving camera motion. To tackle this, we introduce a self-supervised detection and segmentation approach that can work with single images captured by a potentially moving camera. At the heart of our approach lies the observation that object segmentation and background reconstruction are linked tasks. For structured scenes, background regions can be re-synthesized from their surroundings, whereas regions depicting the moving object cannot. We encode this intuition into a self-supervised loss function that we exploit to train a proposal-based encoder-decoder segmentation network. To account for the discrete nature of the proposals, we develop a Monte Carlo-based training strategy. This allows the algorithm to explore the large space of object proposals. We apply our method to human detection and segmentation in images that visually depart from those of standard benchmarks and outperform existing self-supervised methods.

Second, we extend our work on self-supervised detection and segmentation of human in scenarios with dynamic activities and camera motion. We propose a multi-camera framework in which geometric constraints are embedded in the form of multi-view consistency during training. This is achieved via coarse 3D localization in a voxel grid and fine-grained offset regression. In this

manner, we learn a joint distribution of proposals over multiple views. At inference time, our method operates on single RGB images. We outperform the previous techniques both on images depicting unusual human activities and on those of the classical Human3.6m dataset.

In 3D human pose estimation, predicting 3D pose from a 2D image is an inherently ill-posed problem due to the loss of depth information during projection from 3D to 2D. A potential solution to reduce the ambiguities caused by this is to exploit the dependencies between human joints. This enables learning the structure of an articulated body more reliably, which has largely been overlooked by earlier work. To this end, we introduce a deep learning based regression method for structured prediction of 3D human pose from monocular images or 2D joint location heatmaps. Our 3D pose recovery model relies on a traditional CNN to extract image features and an autoencoder to learn a high-dimensional latent pose representation that accounts for the human body joint dependencies. We further propose a Long Short-Term Memory (LSTM) network to enforce temporal consistency on the 3D pose predictions. We demonstrate that our method outperforms earlier approaches both in terms of structure preservation and prediction accuracy on standard 3D human pose estimation benchmarks.

In 3D motion forecasting, the existing work has mostly focused on predicting the future motion from the past sequence of poses for single humans in isolation. However, when there are multiple people engaged in strong interactions, the current state-of-the-art approaches remain suboptimal. Differently from the earlier work, we jointly reason about the collective behavior of the subjects in the scene. This allows us to preserve the long-term motion dynamics in a more realistic way and predict the unusual and faced-paced poses, such as the ones in a dance scenario. To address this problem, we introduce a pairwise attention mechanism that explicitly takes into account the mutual dependencies in the motion history of the subjects. When combined with the self-attention mechanism integrated into an encoder-decoder network, our approach can outperform the state-of-the-art single person baselines. We evaluate the proposed method on our newly introduced dance dataset, Lindyhop600k, that comprises of strong dyadic interactions.

Keywords: Computer vision, self-supervised detection and segmentation, multi-view consistency, 3D human pose estimation, 3D motion forecasting, attention mechanism, deep learning.

Résumé

La détection de personnes à partir d'images 2D et l'analyse de leur mouvement en 3D sont des problèmes de vision par ordinateur de longue date qui sont au cœur de nombreuses applications telles que la conduite autonome et l'entraînement sportif. Récemment, avec la disponibilité de grandes quantités de données d'entraînement et l'avènement de l'apprentissage profond, les performances de la segmentation humaine et de la prédiction de la pose humaine en 3D se sont considérablement améliorées. Cependant, ces problèmes restent difficiles en raison de plusieurs facteurs. Dans cette thèse, nous décomposons l'analyse du mouvement humain en trois sous-tâches : segmentation humaine en 2D, estimation de la pose du corps humain en 3D et prévision du mouvement humain en 3D. Notre objectif est d'alléger les défis de ces problèmes en utilisant différents modèles d'encodeurs-décodeurs.

Bien que les méthodes de détection et de segmentation supervisées atteignent une précision impressionnante, elles se généralisent mal aux images dont l'apparence diffère significativement des données sur lesquelles elles ont été entraînées. Pour y remédier, elles nécessitent des quantités trop importantes de données annotées dans des applications spécifiques au domaine. C'est pourquoi la détection et la segmentation auto-supervisées d'objets de premier plan dans des scènes complexes suscitent un intérêt croissant. Cependant, les approches auto-supervisées existantes reposent principalement sur des hypothèses restrictives d'apparence et de mouvement. Cela exclut leur utilisation dans des scènes représentant des activités hautement dynamiques ou impliquant des mouvements de caméra. Pour résoudre ce problème, nous présentons une approche auto-supervisée de détection et de segmentation qui peut fonctionner avec des images uniques capturées par une caméra potentiellement en mouvement. Au cœur de notre approche se trouve l'observation que la segmentation d'objets et la reconstruction du fond sont des tâches liées. Pour les scènes structurées, les régions de l'arrière-plan peuvent être resynthétisées à partir de leur environnement, alors que les régions représentant l'objet en mouvement ne le peuvent pas. Nous encodons cette intuition dans une fonction de perte auto-supervisée que nous exploitons pour entraîner un réseau de segmentation encodeur-décodeur basé sur des propositions. Pour tenir compte de la nature discrète des propositions, nous développons une stratégie d'entraînement basée sur la méthode de Monte Carlo. Cela permet à l'algorithme d'explorer le vaste espace des propositions d'objets. Nous appliquons notre méthode à la détection et à la segmentation d'humains dans des images qui s'écartent visuellement de celles des références standard et nous

surpassons les méthodes auto-supervisées existantes.

Deuxièmement, nous étendons notre travail sur la détection et la segmentation auto-supervisées de l'homme dans des scénarios avec des activités dynamiques et des mouvements de caméra. Nous proposons un cadre multi-caméras dans lequel les contraintes géométriques sont intégrées sous la forme d'une cohérence multi-vues pendant l'apprentissage. Ceci est réalisé via une localisation 3D grossière dans une grille de voxels et une régression de décalage à grain fin. De cette manière, nous apprenons une distribution conjointe de propositions sur plusieurs vues. Au moment de l'inférence, notre méthode fonctionne sur des images RVB uniques. Nous obtenons de meilleurs résultats que les techniques de pointe, tant sur les images représentant des activités humaines inhabituelles que sur celles de la base de données classique Human3.6m.

Dans l'estimation de la pose humaine en 3D, la prédiction de la pose 3D à partir d'une image 2D est un problème intrinsèquement mal posé en raison de la perte d'informations sur la profondeur pendant la projection de la 3D vers la 2D. Une solution potentielle pour réduire les ambiguïtés causées par ce problème consiste à exploiter les dépendances entre les articulations humaines. Cela permet d'apprendre la structure d'un corps articulé de manière plus fiable, ce qui a été largement négligé par les travaux antérieurs. À cette fin, nous présentons une méthode de régression basée sur l'apprentissage profond pour la prédiction structurée de la pose humaine 3D à partir d'images monoculaires ou de cartes thermiques de localisation des articulations 2D. Notre modèle s'appuie sur un CNN traditionnel pour extraire les caractéristiques de l'image et sur un autoencodeur pour apprendre une représentation latente de la pose en haute dimension qui tient compte des dépendances des articulations du corps humain. Nous proposons également un réseau de mémoire à long terme et à court terme (LSTM) pour renforcer la cohérence temporelle des prédictions de pose 3D. Nous démontrons que notre méthode est plus performante que les approches précédentes en termes de préservation de la structure et de précision de prédiction sur des repères standard d'estimation de pose humaine en 3D.

Dans le domaine de la prévision de mouvement 3D, les travaux existants se sont principalement concentrés sur la prédiction du mouvement futur à partir de la séquence passée de poses pour des humains isolés. Cependant, lorsque plusieurs personnes sont engagées dans de fortes interactions, les approches actuelles de l'état de l'art restent sous-optimales. À la différence des travaux précédents, nous raisonnons conjointement sur le comportement collectif des sujets dans la scène. Cela nous permet de préserver la dynamique du mouvement à long terme d'une manière plus réaliste et de prédire les poses inhabituelles et face-à-face, comme celles d'un scénario de danse. Pour résoudre ce problème, nous introduisons un mécanisme d'attention par paire qui prend explicitement en compte les dépendances mutuelles dans l'historique des mouvements des sujets. Lorsqu'elle est combinée au mécanisme d'auto-attention intégré à un réseau d'encodeurs-décodeurs, notre approche peut surpasser les références de l'état de l'art pour une seule personne. Nous évaluons la méthode proposée sur notre nouvel ensemble de données de danse, Lindyhop600k, qui comprend de fortes interactions dyadiques.
Mots-clés : Vision par ordinateur, détection et segmentation auto-supervisées, cohérence multi-

vues, estimation de la pose humaine en 3D, prévision du mouvement en 3D, mécanisme d'attention, apprentissage profond.

Contents

Acknowledgements i

Abstract (English/Français) iii

List of Figures xiii

List of Tables xv

1 Introduction 1
- 1.1 Motivation and Applications . . . 4
- 1.2 Challenges . . . 6
- 1.3 Problem Definition . . . 7
- 1.4 Contributions . . . 9
- 1.5 Outline . . . 10

2 Related Work 11
- 2.1 Self-supervised Human Segmentation From Single Images . . . 11
  - 2.1.1 Weakly-supervised methods. . . . 11
  - 2.1.2 Motion-based methods. . . . 12
  - 2.1.3 Self-supervised methods. . . . 13
- 2.2 Multi-view Human Segmentation . . . 13
  - 2.2.1 Multi-View Self-Supervised Approaches. . . . 13
  - 2.2.2 Multi-View Self-Supervised Training for Single View Inference. . . . 14
- 2.3 3D Human Pose Estimation . . . 14
  - 2.3.1 Single-Image Methods . . . 15
  - 2.3.2 Multi-View Images. . . . 16
  - 2.3.3 Depth-based Methods . . . 17
  - 2.3.4 Temporal Methods . . . 18
  - 2.3.5 Body Mesh Reconstruction . . . 18
- 2.4 3D Human Motion Forecasting . . . 19
  - 2.4.1 Motion prediction using RNN, GAN and GCN. . . . 19
  - 2.4.2 Attention-based human motion prediction. . . . 20
  - 2.4.3 Social interactions in motion prediction. . . . 20

3 Self-supervised Human Detection and Segmentation via Background Inpainting 23
3.1 Approach . . . 24
3.1.1 Outline . . . 24
3.1.2 Network Architecture . . . 25
3.1.3 Training Losses . . . 27
3.1.4 Monte Carlo and Importance Sampling . . . 29
3.1.5 Exploiting Optical Flow for Training Purposes . . . 30
3.1.6 Implementation Details . . . 31
3.2 Experiments . . . 33
3.2.1 Unusual Activity Filmed Using PTZ-Cameras . . . 33
3.2.2 Activities Captured Using Moving Cameras . . . 36
3.2.3 Comparison to Supervised Models . . . 38
3.2.4 Ablation Study . . . 38
3.2.5 Discussion . . . 43
3.3 Conclusion . . . 46

4 Human Detection and Segmentation via Multi-view Consensus 47
4.1 Approach . . . 48
4.1.1 Multi-View Self-Supervised Training . . . 49
4.1.2 Single-View Inference . . . 54
4.2 Experiments . . . 54
4.2.1 Images and Metrics . . . 54
4.2.2 Comparative Results with Moving Cameras . . . 55
4.2.3 Comparative Results with Static Cameras . . . 58
4.2.4 Ablation Study . . . 59
4.3 Conclusion . . . 60

5 Learning Latent Representations of 3D Human Pose with Deep Neural Networks 61
5.1 Approach . . . 64
5.1.1 Structured Latent Representations via Autoencoders . . . 65
5.1.2 Regression in Latent Space . . . 66
5.1.3 Fine-Tuning the Whole Network . . . 67
5.2 Modeling Temporal Consistency . . . 67
5.2.1 LSTMs . . . 67
5.2.2 Recurrent Pose Estimation . . . 68
5.3 Experiments . . . 69
5.3.1 Datasets . . . 69
5.3.2 Implementation Details . . . 70
5.3.3 Evaluation Protocol . . . 71
5.3.4 Evaluation . . . 72
5.3.5 Comparison Between KDE and Autoencoders . . . 81
5.3.6 Parameter Choices . . . 82
5.4 Conclusion . . . 83

6 Dyadic Human Motion Prediction **85**
6.1 Approach . . . 86
6.1.1 Single Person Baseline . . . 86
6.1.2 Pairwise Attention for Dyadic Interactions . . . 88
6.1.3 Training . . . 89
6.1.4 Implementation Details . . . 90
6.2 Experiments . . . 90
6.2.1 LindyHop600K . . . 90
6.2.2 Data Pre-processing . . . 91
6.2.3 Results . . . 91
6.2.4 Ablation Study . . . 93
6.2.5 Limitations . . . 99
6.3 Conclusion . . . 99

7 Concluding Remarks **101**
7.1 Summary . . . 101
7.2 Limitations and Future Directions . . . 102

A Appendix for Chapter 3 **105**
A.1 Qualitative Results . . . 105
A.2 Importance Sampling Theory . . . 105

B Appendix for Chapter 4 **111**
B.1 Implementation Details . . . 111
B.1.1 Multi-view Consistency . . . 111
B.1.2 Architectures . . . 113
B.1.3 Training Details . . . 114
B.2 Qualitative Results . . . 116

Bibliography **121**

Curriculum Vitae **141**

List of Figures

1.1 Human segmentation and motion analysis 2
1.2 Challenges in human segmentation and motion analysis 6

3.1 Domain specific detection and segmentation 23
3.2 Our self-supervised detection and segmentation architecture 25
3.3 Optical flow image generation on Ski-PTZ and Handheld190k 31
3.4 Off-the-shelf inpainting results on Ski-PTZ 33
3.5 Soft segmentation masks generated by our method and P-GAN 35
3.6 Single-view segmentation qualitative results on the Ski-PTZ 35
3.7 Single-view segmentation qualitative results on the Handheld190k 37
3.8 Single-view segmentation qualitative results on the FS-Singles 37
3.9 Qualitative segmentation results of MaskRCNN 39
3.10 Impact of the segmentation mask regularizer 41
3.11 Ablation study on Human3.6m 42
3.12 Optical flow failure 44
3.13 Multi-person detection and segmentation results 44
3.14 Examples of qualitative results on DAVIS2016 45

4.1 Leveraging multi-view consistency at training time 48
4.2 3D proposal grid 49
4.3 Overview of the underlying single-view self-supervised segmentation pipeline. 50
4.4 Finding bounding boxes that are view consistent 52
4.5 Multi-view consistency qualitative results on the Ski-PTZ 56
4.6 Multi-view consistency qualitative results on the Handheld190k dataset 57
4.7 Multi-view consistency qualitative results on Human3.6m 58

5.1 Overview of our structured prediction approach for 3D pose estimation 62
5.2 Our architecture for the structured prediction of the 3D human pose 65
5.3 Our (B)LSTM networks to enforce temporal consistency 68
5.4 Qualitative 3D pose estimation results on Human3.6m 73
5.5 Analysis on structure preservation ability of our 3D pose estimation approach . 76
5.6 Visualization of the learned latent pose space 77
5.7 3D pose estimation results on HumanEva-I 78
5.8 3D pose estimation results on KTH Multiview Football II 79

5.9 3D pose estimation results on LSP 79
5.10 3D pose estimation results with LSTMs on Human3.6m 81

6.1 Single person motion forecasting baseline 87
6.2 Overview of our 3D motion forecasting model based on self- and pairwise attention 88
6.3 Optimizing 3D poses in the Lindyhop600k dataset 92
6.4 Qualitative 3D motion prediction results on the Lindyhop600k test subject with the follower role 94
6.5 Qualitative 3D motion prediction results on the Lindyhop600k test subject with the leader role 95
6.6 Qualitative 3D motion prediction results on the Lindyhop600k test subjects . . 96
6.7 Example failure cases on the Lindyhop600k test subjects 97

A.1 Capture setup of our in-house Handheld190k posing dataset. 106
A.2 Additional single-view detection and segmentation results on the test subjects of Ski-PTZ 108
A.3 Additional single-view detection and segmentation results on the test subjects of Handheld190k 109
A.4 Additional single-view detection and segmentation results on Human3.6m . . . 110

B.1 Additional multi-view consistency results on the Ski-PTZ dataset 117
B.2 Additional multi-view consistency results on the Human3.6m dataset 118
B.3 Additional multi-view consistency results on the Handheld190k dataset 119
B.4 Multi-person detection and segmentation at test time 120

List of Tables

3.1 Single-view segmentation results on the Ski-PTZ, Handheld190k and FS-Singles datasets . . . 34
3.2 MaskRCNN segmentation results on the Ski-PTZ, Handheld190k and FS-Singles datasets . . . 38
3.3 Single-view segmentation analysis of the mask prior effect and ImageNet pre-training on the Ski-PTZ dataset . . . 39
3.4 Single-view segmentation hyper-parameter study on the Ski-PTZ dataset . . . 40
3.5 Single-view detection results on the Human3.6m and Ski-PTZ datasets . . . 43

4.1 Multi-view consistency segmentation results on the Ski-PTZ . . . 56
4.2 Multi-view consistency comparative results on Human3.6m . . . 58
4.3 Multi-view consistency segmentation ablation study on the Ski-PTZ . . . 59
4.4 Multi-view consistency detection ablation study on Human3.6m . . . 59
4.5 Influence of voxel resolution . . . 60

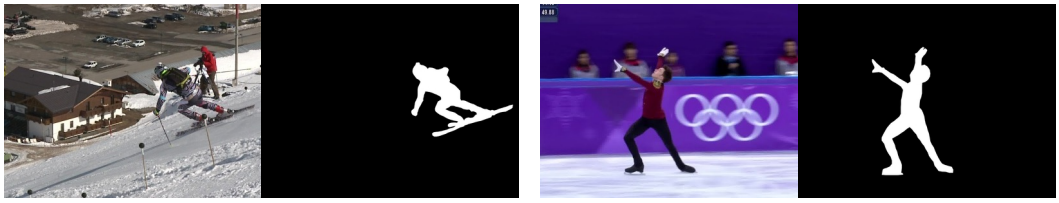
5.1 Comparison of our structured prediction approach with earlier work on Human3.6m 72
5.2 Comparison of our structured prediction approach with earlier work after Procrustes transformation on Human3.6m . . . 73
5.3 Ablation studies for our structured prediction approach . . . 74
5.4 Evaluation of our structured prediction approach with very deep network architectures . . . 75
5.5 Quantitative results of our structured prediction approach on Walking sequences of the HumanEva-I dataset . . . 78
5.6 3D pose estimation results on KTH Multiview Football II . . . 79
5.7 Analysis of our different (B)LSTM architectures . . . 80
5.8 Comparison of our (B)LSTM-based architectures to the earlier work . . . 80
5.9 Comparison of our structured prediction approach to KDE . . . 82
5.10 Analysis on the hyperparameter choices for our structured prediction approach 82

6.1 Lindyhop600k dataset structure . . . 91
6.2 Comparison of our dyadic motion prediction approach with the state-of-the-art methods on the Lindyhop600k dataset . . . 93
6.3 Ablation study for incorporating interactions . . . 98

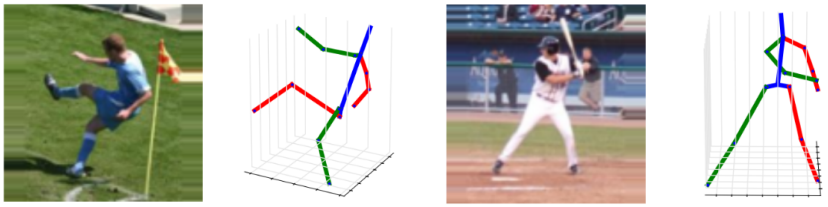
1 Introduction

The process of training computers to perceive and interpret visual information from the real world focuses mainly on objects that stand out in everyday life. Among them, humans have been of great importance since people are constantly exchanging information with the surrounding through theirs actions. Therefore, the human body is key to understanding the environment around us. It has a highly articulated structure giving it the ability to move in different ways creating large shape and appearance variety. Identifying the body parts and modeling their movement in diverse settings have been of particular interest to computer vision since it facilitates many applications ranging from video surveillance to autonomous driving [68]. However, considering that humans often interact with each other or other objects, analyzing each person in isolation is not sufficient. The dynamics that result from complex human articulation and interactions cannot be integrated to an automated system in a straightforward manner. Therefore, we need robust algorithms and discriminative representations that can identify human body in various motion and model the dependencies inherent in their actions to have a better understanding of the entire scene. In this thesis, we analyze human motion by addressing 2D human segmentation, 3D human pose estimation and motion forecasting.

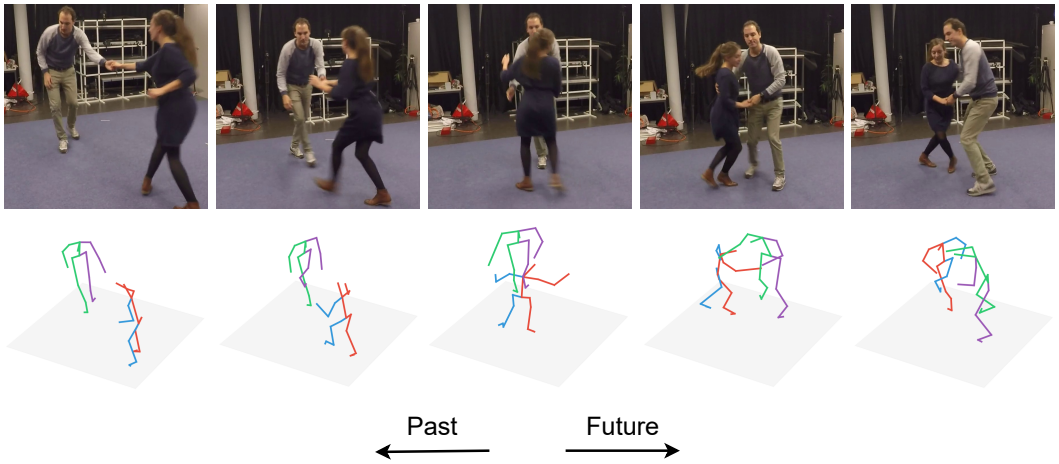
The goal of object detection and segmentation is to produce a bounding box for each object of interest in the scene along with their binary masks. This problem applies to a wide range of video understanding applications, such as video surveillance, unmanned vehicle navigation, action recognition and motion prediction. Our focus is primary object segmentation which is the task to segment a single salient object from the background as depicted in Fig. 1.1(a). We consider moving objects in a moving camera setting. In general, the object to segment is expected to appear and move differently than the background and repeat frequently in a sequence of images [132]. Earlier approaches focused on background modeling in static camera scenarios [18]. Recently, this focus has shifted from static cameras to freely moving cameras [201, 267, 159, 305]. Due to the complex video content and the dynamic nature of the background, object segmentation remains a challenging problem. The existing work tackles this problem by using appearance and motion cues. Appearance based models focus on learning the color and shape distribution of the foreground object based on RGB values without assuming any prior knowledge on its size and



(a) Human segmentation



(b) 3D human pose estimation



(c) 3D motion forecasting

Figure 1.1 – **Human segmentation and motion analysis.** Humans and their motion have been the main interest of many computer vision applications. We focus on (a) detecting and segmenting humans from their background, particularly in challenging scenarios such as skiing and figure skating, (b) recovering their 3D poses from images and (c) predicting their future poses given the past ones.

location [201]. In case of video data, optical flow based methods [268, 267, 108] have emerged to account for the movement of the foreground object. To achieve increasing consistency of masks across frames, models enforcing temporal coherence between neighboring frames have been developed [148, 62, 132]. However, these methods might fail to handle difficult conditions such as motion blur, background clutter, occlusions and unusual human motion. To remedy this, we introduce a proposal-based method that relies on encoding and decoding the content of a scene to learn to decompose it into a foreground and a background. To better handle occlusions and ambiguities, we leverage multi-view consistency and investigate how this additional source of information can constrain the learning process. In this thesis, we focus on scenarios including

human motion. Therefore, we limit the applicability of our method to images or videos capturing humans as the salient object.

Once the person of interest is identified in an image using the appearance and motion cues in 2D, the next step is to have a more detailed analysis of the human pose. 3D pose estimation, as shown in Fig. 1.1(b), involves recovering the articulated 3D joint locations of a human body from an image or video. It has a great variety of potential applications including autonomous driving, human–computer interaction, video surveillance, sports performance analysis and virtual try-on. Nonetheless, predicting 3D joint coordinates from 2D images remains highly challenging [68] since it is an inherently ambiguous problem and existing datasets are not diverse enough to cover the full range of human pose space. Earlier approaches tackle this problem by relying on body silhouettes [3], deformable template matching [192, 193], 2D joint locations [221, 328], temporal consistency [277, 10], multiple cameras [28, 124, 20, 207] and depth images [247, 311]. When estimating 3D human poses from multiple views, the main challenges include the cost of having calibrated and synchronized cameras, larger state space and cross-view ambiguities. In case of using depth information, existing approaches might suffer from the errors in data acquisition caused by the ambient background light, noise characteristics of depth cameras, multi-device interference and dynamic scenery [241]. With the introduction of deep learning approaches, some of the constraints requiring additional sources of information have been relaxed. Consequently, the interest in estimating 3D human pose from a single monocular image has increased in recent years. In this thesis, we propose a deep learning based monocular solution to 3D human pose estimation that can learn discriminative latent pose representations on a wide range of datasets. To account for the complex mapping between the image and the corresponding 3D human pose, we train an autoencoder that disentangles the body joint dependencies in high dimensional latent space and enforces structural constraints on the output.

Recovering the current 3D pose of the people from videos is fundamental to human motion analysis, yet it remains limited for understanding the complex dynamics of motion. What is central to our interaction with the outside world is predicting the upcoming motion of people and it is done inherently as a part of our daily lives. This task becomes prominent in many sports activities. Knowing in advance what the opponent is going to do is a skill in itself that our brains can learn naturally while practicing. Humans are also capable of fluidly walking or driving in a crowded environment by anticipating the movement of others. Although this type of process is easy for us, it is extremely complex for a machine to accomplish a similar behavior. As illustrated in Fig. 1.1(c), human pose forecasting aims to predict the future poses from the observed poses in the past [65]. While this problem has recently received increasing attention, existing solutions focus mainly on estimating the motion of a single person in isolation [285, 263, 65, 107, 185, 181]. In real world scenarios, people often interact with each other and the motion of one person depends highly on the social context [5, 238, 169]. We observe that taking such motion dependencies explicitly into account enables us to predict long-term future more accurately [11]. In this work, we focus on human motion forecasting for dyadic motions with strong human-human interactions. One such motion can be observed in a lindy hop dance sequence which is comprised of energetic moves ranging from frenzied kicks to smooth and sophisticated body movements. The dancers

synchronize their fast-paced steps with one another and the music. They often improvise flips and twirls which makes it hard to predict the steps that follow without observing the moves of the partner. Therefore, there is a great interest in analyzing the motion of lindy hop dancers. To estimate the future poses of each individual more reliably, we exploit an encoder-decoder model. This model learns a spatio-temporal contextual representation from the mutual information between the interacting people.

In this thesis, we propose various deep neural network based models to address the previously introduced tasks. With the advent of deep learning, the field has taken great leaps and has been able to surpass humans in some tasks such as object detecting and classification. One of the driving factors behind this is the massive amount of data generated every day. In the paradigm of supervised learning, this data has to be carefully labeled to train a model that performs decently and an increased amount of annotated data further boosts the performance. However, labeled data is costly to prepare and can be biased as well. In the context of object detection and segmentation, data labeling requires providing the bounding box location and the binary mask of the object in pixel level for every frame. In human pose estimation and motion forecasting, the 2D and 3D location of every human body joint should be acquired, typically through MoCap systems with multi-camera setup and special hardware. However, to capture outdoor scenes with moving objects, the footage, camera calibration and human pose annotation require more effort. An example to this is a ski footage that aims to film a fast moving skier often occluded by snow on a wide slope. Despite the availability of large annotated datasets for usual activities, the models pre-trained on these datasets usually do not generalize well to domain-specific images [133] capturing less common activities such as skiing for which large training databases are not available [226]. Hence, significant amounts of data should be collected and annotated for any new domain to obtain a desired level of accuracy in a supervised setting. To overcome the tedious work of annotation, a very recent trend aims to revisit the self-supervised learning [197, 204, 319, 203, 70, 110, 198, 112]. We note here that the term self-supervised learning replaced the previously used term unsupervised learning since unsupervised learning is a misleading term that suggests that the learning uses no supervision at all. Self-supervised learning obtains supervisory signals from the data itself, often leveraging the underlying structure in the data. Thus, it does not rely on manual annotation. We investigate these different learning paradigms via encoder-decoder based models throughout this thesis.

In the remainder of this chapter, we first discuss a few practical applications and present several key challenges related to human segmentation, 3D human pose estimation and motion forecasting. Then, we define the aforementioned tasks in detail. Finally, we summarize our main contributions and give an outline of the thesis.

1.1 Motivation and Applications

Vision based human segmentation and motion recognition have fascinated many researchers. This is partly due to the increasing popularity of wearable kinematic sensors and multi-view cameras

that enabled a large amount of human mocap data to be recorded. Combined with the computing power which has become more affordable and easily accessible, analyzing this tremendous amount of data revolving around human activities has facilitated a fine-grained understanding of the physical world. The knowledge we acquire by investigating the human presence and motion in videos can be integrated into vision based automated systems. Today, we have a wide spectrum of applications [309, 287], ranging from augmented reality to surveillance, that focus both on the static and dynamic aspects of human body. These applications can reduce the need for human labor and the human errors, particularly in healthcare and athletic training. In the following, we briefly discuss potential applications that exploit human segmentation and motion analysis.

Human-Computer Interaction. The interfaces connecting human users to automated systems are ranging from conventional devices such as screens and keyboards to smartphones, head-mounted displays [95] and haptic technologies. To decrease the complexity of human-computer interaction, we need well-designed interfaces that can interpret human gesture and pose. To this end, vision based solutions rely on robust detection of the users and analysis of their body motion in the 3D world.

Augmented Reality. Computer vision based augmented reality involves integrating virtual elements such as images or audio over what we see in the physical world. It enables consumers to try on clothes, visualize furniture in their homes or overlay masks on their faces. One such application is introducing digital characters into a scene. It is important that virtual elements and real world are combined in a seamless way and this requires accurate localization and pose estimation of the human in the scene [252].

Autonomous Driving. For self-driving cars, it is important to detect surrounding vehicles, pedestrians and objects to maintain security in traffic. To this end, such autonomous systems need to understand the scene and interpret the intentions of the surrounding humans. Thus, localizing and predicting the pose of pedestrians have great significance [141, 66].

Healthcare. Accurate tracking of human motion can be useful in clinical environments when giving feedback on the health status of patients [36]. It can assist the medical staff in monitoring anomalies in intensive care units. In addition to that, it can help correct the sitting posture, and gait or sleep-related motion disorders [231]. Identifying the movement patterns during epileptic seizures or Parkinson disease can serve as clinical decision tool for physicians. In some situations, infants are born with muscles or joints that are not working properly and early diagnosis of such diseases via computer vision can save many lives [92].

Athletic Training. Automated estimation of 3D poses from videos can serve as a virtual coach to provide the athletes with detailed feedback on how they can improve their moves or educate novices in sports such as skiing, swimming and yoga [286]. Since most of the sports competitions leave a lot to interpretation, such systems can also help referees make decisions more objectively and evaluate the performance of athletes in ambiguous cases. In addition to that, motion forecasting frameworks can be used in combat or team sports to train athletes against

(a) Occlusions (b) Cluttered background (c) Motion blur

(d) Changing background (e) Changes in appearance (f) Changes in viewpoint

Figure 1.2 – **Challenges in human segmentation and motion analysis.** (a) Self- and person-to-person occlusions leading to missing image cues for certain body parts. Images are taken from the Lindyhop600k and boxing datasets introduced in this thesis. (b) Cluttered background with other people and objects from 3D Poses in the Wild dataset [280] (c) Motion blur caused by fast-paced movements. Images are taken from [120]. (d) Dynamically changing background with a different looking scenery from one frame to another. Images are taken from 3D Poses in the Wild dataset [280]. (e) Different clothing and appearance among people exhibiting the same pose in the Handheld190k dataset [120]. (f) Same pose captured from different viewing angles in the Ski-PTZ dataset [226].

their opponents by predicting a combination of possible movements from observed motion.

Video Surveillance. Accurately interpreting what is happening in a footage has the potential to enhance public security. When powered by computer vision, such surveillance systems can provide life-saving information that can go unnoticed by a security staff. Today, such frameworks focus on localizing and tracking the people in crowded areas [315, 54]. Thus, human detection and pose estimation have a key role in this group of applications [246, 45].

1.2 Challenges

We discuss the main challenges in human segmentation and motion analysis below and in the next section we provide our solutions to address them.

Occlusion and Clutter. When the human body is partially or fully occluded, as illustrated in Fig. 1.2(a,b), segmentation and motion prediction become less robust or infeasible. A common practice to make the estimation resilient to such cases is to augment the training data with synthetic occlusions by hiding the content of randomly chosen regions in object segmentation or feeding noisy poses with missing joints as input in motion prediction.

Fast Motion. When an object moves fast, it leads to unreliable optical flow estimation and motion blur causing the object to have fuzzy boundaries in the image, as shown in Fig. 1.2(c). The blur

induced at a particular pixel on a moving object cannot be resolved simply as it results from the combined effects of camera motion, the object's own independent motion during exposure and its relative depth in the scene.

Dynamic Background. Having background regions such as the one in Fig. 1.2(d) that move or deform poses a challenge for foreground object segmentation. Flowing water, leaves moved by the wind, snow splashed by the skier are examples of such dynamic background that can be detected as foreground objects, i.e. false positives.

Appearance and Viewpoint Changes. The shape, size and appearance of people vary significantly and even for the same person, the appearance can change drastically from one frame to another due to viewpoint or illumination changes. An example of this is depicted in Fig. 1.2(e,f). Therefore, it is necessary that the model learns appearance invariant features to generalize to unseen people at test time.

Lack of Annotated Data. Learning based methods, in particular deep neural networks, demand collecting large labeled datasets for training. In human detection and segmentation, annotating images requires pixel level localization of the object, whereas in pose estimation and motion forecasting, the 2D and 3D location of each body joint should be provided. Such ground truth can be obtained either from synthetic data or from annotations on real-world data. The former introduces inevitable domain gap between the data used for training and the real-world test scene. The latter is a costly, labor-intensive, and error-prone process. To mitigate the annotation burden, learning discriminative pose and scene representations is a key solution. A practical and scalable way to achieve this is to learn from the underlying structure of the data rather than the annotations.

1.3 Problem Definition

We decompose the human motion analysis into three sub-tasks; human segmentation, 3D pose estimation and 3D motion forecasting.

We consider human segmentation as a primary object segmentation task which aims at segmenting a single salient object from the background across all frames [309]. This involves computing dense pixel level masks for the foreground object, and placing a bounding box surrounding this object without having any prior knowledge on the size and location. It is closely related to video object segmentation problem which can be roughly categorized into unsupervised and semi-supervised protocols [108, 158, 159, 307, 309]. Unsupervised methods attempt to extract the salient object without using any manual annotations at test time. On the other hand, semi-supervised approaches segment the foreground object given the mask annotation for the first frame in a video clip. Our goal is to achieve an entirely self-supervised algorithm, which differs from the standard unsupervised strategies requiring domain-specific annotations during training [158, 159, 307, 175]. Without any prior information on the primary object, it is difficult to model the foreground in the scene. Therefore, we build our method on the following

assumptions [150]: (1) Salient objects are outliers in the global scene, having distinct appearance, smaller size and different movement patterns than their larger background. (2) The pixels that belong to the foreground object should display coherent appearance and motion cues in space and time. We integrate these principles into an encoder-decoder structure that decomposes an RGB image into a foreground and a background according a segmentation mask of the foreground object. To exploit the motion cues, we extend our method to use optical flow as intermediate supervision. Furthermore, we build upon our single-view approach and develop a self-supervised object detection and segmentation method that explicitly encodes multi-view geometry during training. At test time, our method operates on single RGB images and yields a bounding box and binary segmentation mask of the foreground object. Although our method is generic and does not depend on any category based knowledge, we apply it in domain-specific human scenarios such as skiing where the general purpose detectors tend to fail.

3D human pose estimation can be formulated either as a discriminative or a generative method [20, 287]. The former directly learns a mapping from the input to the human pose space, whereas the latter models the underlying structure of the human body. We employ a discriminative method to recover the 3D joint coordinates of the human body from a single RGB image. Based on the type of representation, there are three commonly used human body models: skeleton-based, contour-based and volume-based model [287]. The skeleton-based model is a kinematic model that represents the body as a set of joint locations or relative limb orientations. In the contour-based model, the human body is represented as the boundaries of a person's silhouette. Finally, volume-based models correspond to the geometric shapes or meshes. We adopt the skeleton-based model and use a skeleton with J joints represented by a vector of dimension $3J$ in the Cartesian space. We predict the 3D joint locations in the camera coordinate system relative to a root joint, e.g. hip.

In human motion prediction, the standard protocol is to predict the future poses of a single person given the past ones. An input pose sequence consists of consecutive pose vectors of length T_p, each of dimension $3J$, and the output is also a sequence of consecutive poses of length T_f, with $T_f <= T_p$ [181]. In contrast to prior work, we investigate this problem from a novel perspective that involves humans performing collaborative tasks and engaging in close interactions. We focus on scenarios that include a primary subject and the interactee (second person) such as dancing couples [11]. Our goal is to facilitate the recovery of the future poses of the primary actor by paying attention to the interactions between the two. Therefore, the input to our pipeline is the history of the coupled motion and we infer the future motion of the primary person. Existing single person based solutions [183, 181, 145, 167] use ground truth poses as the history of motion. In practice, we do not have access to these ground truth poses corresponding to the observed part of a video clip. A more realistic solution is to first extract the pose vectors from a sequence of images via a 3D pose estimation pipeline and feed them to the motion forecasting framework. To this end, we design two different models; one that learns from ground truth past motion and the other that operates on a sequence of consecutive images. The latter yields noisy predictions for the motion history. This enables the model to be resilient to erroneous past motion when predicting the future. The performance is validated on dance sequences with short-term ($< 500ms$) and

long-term ($< 1000ms$) forecast times.

1.4 Contributions

In this thesis, we consider three tasks related to human motion analysis, namely self-supervised human segmentation, 3D human pose estimation and 3D human motion forecasting. Our goal is to develop algorithms that learn robust latent representations via different encoder-decoder models. We explore various input modalities such as single RGB images in studio environment for 3D pose recovery, single and multi-view outdoor image sequences captured by hand-held moving cameras and optical flow data for human segmentation. In future motion prediction, we use human body pose vectors. We show that our contributions apply to a wide range of datasets and the proposed methods outperform the prior work in the corresponding fields. We describe below the main contributions of this thesis.

Self-supervised Human Detection and Segmentation via Background Inpainting. We present a self-supervised method for object detection and segmentation from single RGB and optical flow images that outperform general purpose detectors in domain-specific applications [120]. Our core contributions are the Monte Carlo-based optimization of proposal-based detection, new foreground and background objectives, and their joint training on unlabeled videos captured by static, rotating and handheld cameras. We introduce a new dataset captured by moving cameras in an outdoor environment depicting daily human activities such as the ones in Human3.6m [102] and release it for public use.

Human Detection and Segmentation via Multi-view Consensus. We propose a self-supervised end-to-end trainable object detection and segmentation approach that explicitly leverages 3D multi-view geometry during training [121]. In contrast to most recent works that relies on single view or multi-view images acquired using static cameras, our approach can handle moving background while enforcing consistency across views. To this end, our model comprises a 3D object proposal framework that enables an efficient multi-view voting scheme without having to introduce additional loss terms.

Learning Latent Representations of 3D Human Pose with Deep Neural Networks. We introduce one of the first deep learning frameworks for 3D pose estimation that takes into account the human body joint dependencies [122]. While prior work tackled this problem by directly regressing the 3D coordinates of joints from single RGB images, we propose to map the input image to a high dimensional latent representation learned by training an overcomplete autoencoder. We show that the embedding we obtain from the pose data itself can encode and preserve the implicit structure of human body pose more accurately than a standard CNN. Furthermore, to enforce temporal consistency, we propose the first LSTM based pose recovery model that takes as input the initial 3D pose predictions and refines them.

Exploiting Interactions in Human Motion Prediction. We propose an approach that addresses

the challenges of human motion forecasting for dyadic motions with strong human-human interactions. We propose the first attention based 3D motion forecasting model that exploits the motion dependencies among socially interacting people. Furthermore, we introduce a new dance dataset, Lindyhop600k, which consists of videos and 3D human body poses of dancers performing diverse swing motions. We release it for research use.

1.5 Outline

The remainder of this thesis is organized as follows. In Chapter 2, we discuss the related work on object segmentation, 3D human pose estimation and motion prediction. In Chapter 3, we introduce a self-supervised proposal-based human detection and segmentation method. We develop an inpainting based model to tackle moving background and a Monte-Carlo based sampling strategy to handle the discrete nature of proposals. We show the effectiveness of our method on domain-specific human activities. In Chapter 4, we provide the multi-view extension of the method presented in Chapter 3. To this end, we introduce a self-supervised object detection and segmentation approach that explicitly leverages 3D multi-view geometry during training to enforce consistency across the views. In Chapter 5, we present our work on supervised structured prediction of 3D human pose from single images. We demonstrate that a high dimensional latent representation learned via an autoencoder based model can account for the joint dependencies more reliably than directly regressing the joint locations from an RGB image. In Chapter 6, we propose a 3D motion forecasting model that exploits person-to-person interactions to recover the long-term motion dynamics more reliably. We build a new dance dataset Lindyhop600k that comprises of strong interactions and evaluate our method on this dance scenario. Finally, in Chapter 7, we conclude this thesis with a summary of our findings and ideas for future work.

2 Related Work

In this thesis, we different computer vision problems ranging from human segmentation in 2D to 3D human motion analysis. We start this chapter by reviewing the previous methods in 2D video object segmentation and discuss in more detail the self-supervised techniques that are the most relevant to our single-view human segmentation approach. We then give a brief overview of methods using multiple cameras to detect and segment objects. In 3D human pose estimation, the literature is very diverse and therefore, we limit our discussion mostly to monocular approaches. Although we focus on the work prior to ours, we also include the recent state-of-the-art approaches to give a general idea about how the field has evolved. Finally, we discuss the related work in 3D motion forecasting.

2.1 Self-supervised Human Segmentation From Single Images

Most salient object detection and segmentation algorithms are fully-supervised [41, 88, 222, 253, 22, 162, 174, 243, 244, 308] and require large annotated datasets with paired images and labels. Our goal in Chapter 3 and 4 is to train a purely self-supervised method without either segmentation or object bounding box annotations. Note that this differs from the so-called *unsupervised object segmentation* methods [213, 98, 108, 158, 159, 175, 284, 307, 291, 318, 322], that require domain-specific annotations during training but not at test time, or the label of the first frame at inference time [292]. We focus our discussion on self- and weakly-supervised methods with regard to the type of training data used and refer to [132] for a complete discussion of methods using hand-crafted optimization.

2.1.1 Weakly-supervised methods.

An early weakly-supervised method is the Hough Matching algorithm [43]. It uses an object classification dataset and identifies foreground as the image regions that have re-occurring Hough features within images of the same class. Similar principles have been followed to train deep networks for object detection [108, 296], optical flow estimation [267, 268], and object

saliency [158]. These methods make the implicit assumption that the background varies across the examples and can therefore be excluded as noise. This assumption is violated when training on domain-specific images, where foreground and background are similar across the examples.

2.1.2 Motion-based methods.

Conventional methods [148, 201, 62, 125, 290, 83, 132, 254, 305] explore the motion information mainly by resorting to hand-crafted features. [290] proposes a spatial-temporal energy function applied to optical flow field to obtain spatiotemporally consistent saliency maps that are further improved by using global appearance and location models. Similarly, [201] computes the optical flow to detect motion boundaries and refines them through ray-casting strategy. An alternative temporal solution [132] relies on the recurrence property of the primary object in a video. It finds the recurring candidate regions in the entire sequence by extracting color and motion cues through ultrametric contour maps. Identifying the matching segment tracks in different frames is done by minimizing a chi-square distance temporally in the feature space. Given video sequences, the temporal information can be exploited by assuming that the background changes slowly [18] or linearly [254]. However, even a static scene induces non-homogeneous deformations under camera translation, and it can be difficult to handle all types of camera motion within a single video, and to distinguish articulated human motion from background motion [235]. Some of the resulting errors can be corrected by iteratively refining the crude background subtraction results of [254] with an ensemble of student and teacher networks [50]. This, however, induces a strong dependence on the teacher used for bootstrapping. Recently, [176] showed that leveraging the temporal information at different granularities through forward-backward patch tracking and cross-frame semantic matching can be used to learn video object patterns from unlabeled videos. Note that these methods can only operate on video streams and exploit a strong temporal dependency, which our model does not require.

Our self-supervised detection and segmentation approach is conceptually related to VideoPCA [254], which models the background as the part of the scene that can be explained by a low-dimensional linear basis. This implicitly assumes that the foreground is harder to model than the background and can therefore be separated as the non-linear residual. In Chapter 3, in addition to using motion cues, we propose to rely on the predictability of image patches from their spatial neighborhood using deep neural networks. This gives us an advantage over VideoPCA, which only works with videos and comparably little background motion and complexity. Another closely related work [305] employs a similar inpainting network to ours on flow fields. It relies on an adversarial model that tries to hallucinate the optical flow from its surrounding while generating the mask of a supposedly moving object in the region where the inpainting network yields poor reconstruction. [305] is based on the PWC network [257] that is trained with supervision on a large object database to predict flow with clear object boundaries. In that sense, as the methods based on deep optical flow, it is not strictly self-supervised and can suffer from degenerate cases when applied to still images with no or little movement. We will nonetheless show that our single-view human segmentation approach can also benefit from such optical flow prediction if available,

outperforming the other methods that use this information.

2.1.3 Self-supervised methods.

Most similar to our single-view human detection and segmentation approach are the self-supervised ones to object detection [24, 49, 61, 224] that complement autoencoder networks by an attention mechanism. They first detect one or several bounding boxes, whose content is extracted using a spatial transformer [105]. This content is then passed through an autoencoder and re-composited with a background. In [224], the background is assumed to be static and in [49, 61] even single colored, a severe restriction in practice. [49] uses a proposal-based network similar to ours, but resorts to approximating the proposal distribution with a continuous one to make the model differentiable. In Chapter 3, we demonstrate that much simpler importance sampling is sufficient. In [203] a noisy segmentation masks is predicted by an unsupervised version [62] used as a pseudo label to train a ConvNet to segment moving objects from single images. [24] uses a generative model relying on the assumption that the image region strictly covering the salient object can be subject to random shifts without affecting the realism of the scene. Similarly, the method of [37] relies on an adversarial network whose generator extracts the object mask and redraws the object by assigning different color or texture features to that region. This is very different from our human detection and segmentation approach that aims to reconstruct the scene from its background. Along similar lines, the algorithm of [13] searches for the foreground object by compositing it into another image so that the discriminator fails to classify the resulting image as fake. These methods can be easily deceived by other background objects whose random displacement or texture change can still yield realistic images. In contrast to these GAN-based techniques, our self-supervised single-view approach works with images acquired using a moving camera and with an arbitrary background.

In addition to object detection, the algorithm of [224] also returns instance segmentation masks by reasoning about the extent and depth ordering of multiple people in a multi-camera scene. However, this requires multiple static cameras and a static background at training time, as does the approach of [17] that performs instance segmentation in crowded scenes.

2.2 Multi-view Human Segmentation

In this section, we discuss the multi-view object detection and segmentation approaches relevant to our self-supervised human segmentation technique that leverages multi-view consistency in Chapter 4.

2.2.1 Multi-View Self-Supervised Approaches.

Earlier works include the generative unsupervised multi-person detection and tracking methods proposed in [64, 17]. The former localizes and matches persons across several cameras with

overlapping fields of view using a grid of candidate positions on the ground plane. The latter uses a joint CNN-CRF architecture and Mean-Field inference to produce a Probabilistic Occupancy Map (POM) as in [64] but leverages discriminative features extracted by a CNN. Both require background subtraction images as input and can therefore only work with static cameras. Furthermore, they exploit multiple views at inference time, whereas we aim to perform monocular person segmentation.

2.2.2 Multi-View Self-Supervised Training for Single View Inference.

Our work using multi-view information for 2D detection and segmentation during training is closely related to [225, 224] in that we do not use any segmentation annotation to learn the foreground region. In [225, 224], novel view synthesis is used in conjunction with multi-view synchronized videos of human motions captured by calibrated cameras to learn a geometry-aware embedding. In contrast to our approach, it requires a known background to decompose the scene into foreground and background regions. Hence, it cannot handle scenes filmed by moving cameras. In Chapter 4, we introduce a method that works with a changing background. To this end, we do not rely on novel view synthesis but instead exploit multi-view consistency by relating the 2D detections of the multiple views to a common 3D capture volume.

2.3 3D Human Pose Estimation

3D human body pose estimation can be roughly classified into two categories; traditional marker-based motion capture systems and image-based 3D human pose estimation techniques. In this thesis, we focus on the latter and discuss different input modalities along with various pose representations used in this problem.

Pose Representations. The most common human body models are either based on the skeleton or the shape parametrization. In the skeleton-based model, the body is represented as 3D keypoints connected through a tree structure. The 3D location of a keypoint can be defined as the 3D position relative to the camera center or a root joint such as the pelvis. Many existing approaches estimate the 3D Cartesian joint coordinates $\mathbf{y} = [\mathbf{p}_1^T, \mathbf{p}_2^T, ..., \mathbf{p}_N^T]^T$ where $\mathbf{p}_i$ is the 3D location of the i-th joint in a skeleton with N joints. An alternative is to use 3D-joint rotations which are then integrated via forward kinematics. The 3D rotations can be defined using 3D or 4D representations such as exponential map, Euler angles or quaternions. In case of using exponential map, the pose is represented as $\mathbf{y} = [\mathbf{p}_g^T, \boldsymbol{\theta}_g^T, \boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, ..., \boldsymbol{\theta}_{N-1}^T]$ where $\mathbf{p}_g^T$ is the global translation and $\boldsymbol{\theta}_g^T$ is the global orientation of the root joint. For other joints, the joint angles $\boldsymbol{\theta}_i^T$ are relative to their parent joint.

In the shape-based model, recent works use the skinned multi-person linear (SMPL) model [173, 27, 144, 208, 118, 129] represented as a triangulated mesh. The methods employing this representation estimate the shape parameters that are used to model the body proportions and the pose

parameters that determine how the body is deformed.

2.3.1 Single-Image Methods

3D human body pose can be recovered from monocular images either through direct mapping [156, 160, 122, 327, 206, 258, 177] or lifting the corresponding 2D pose [202, 186, 265, 271, 191, 188, 259, 205, 82].

Methods Preceding Our Work in 3D Human Pose Estimation. The first group of methods relies on an end-to-end network that predicts the 3D coordinates of joints in a straightforward manner. [156] designs a deep neural network within a multi-task learning framework that jointly learns to detect the 2D joints and regress their 3D locations. [160] proposes a structured learning within a deep neural network framework and encodes the joint dependencies by extending the structured SVM model for 3D human pose estimation. It learns a similarity score between feature embeddings of the input image and the 3D pose. This process, however, comes at a high computational cost at test time, since, given an input image, the algorithm needs to search for the highest-scoring pose. Our structured prediction of the 3D human pose [122] fits in that line research, which involves combining autoencoders with CNNs that account for the dependencies between body articulations. Earlier approaches [101, 102] achieve this through kernel dependency estimation (KDE) and encode complex dependencies between human joints in a lower dimensional space. However, given the highly complex structure of human body, we show that dimensionality reduction is not the most effective way of achieving this. The key to our approach is mapping 3D human pose to a higher dimensional latent representation via an autoencoder to disentangle the inherent dependencies. In Chapter 5, we explain the methods that are the most relevant to our approach in more detail. In the remainder of this section, we discuss the more recent single-image methods and other deep neural network based strategies that are employed following our work.

Methods Following Our Work in 3D Human Pose Estimation. In contrast to the previous regression methods, [177] combines the 2D and 3D pose estimation tasks with human action recognition. To preserve the geometric structure of human body [327] introduces a kinematic object model consisting of bones that have a fixed length and [258] uses bone representation to enforce geometric constraints. [206] proposes a volumetric approach and discretizes the 3D space around the subject to train a network for predicting per voxel likelihoods for each joint.

To alleviate the ambiguities in 3D pose estimation in-the-wild, many recent works have leveraged joint heatmaps or 2D pose results. [202] combines 2D pose estimation results with image features to recover the 3D position relative to multiple joints. [186] proposes a simple baseline pioneering the research on lifting 2D poses to 3D. [265] fuses 2D joint heatmaps and 3D image cues in a trainable scheme. [271] employs a multi-stage pipeline to jointly estimate 2D and 3D body poses in an iterative manner by predicting belief maps for the location of the 2D landmarks and lifting them to 3D. [196] predicts the depth of human joints based on 2D human joint locations through a

LSTM network. Similarly, [82, 323] use both the 2D joint heatmaps and depth features to estimate 3D pose. Differently from the previous methods, [191] formulates this problem as a regression between matrices encoding 2D and 3D joint distances, preserving the structure of the human body more accurately than the standard Cartesian representation. To improve the in-the-wild performance, [188] explores transfer learning from features learned for 2D pose estimation. [259] unifies the heatmap and regression based approaches by transforming the heatmaps into joint location coordinates in a differentiable way. For predicting egocentric 3D human pose, [270, 269] first trains a model to extract the 2D heatmaps of the body joints and then regresses the 3D pose via an autoencoder with a dual branch decoder. Due to the cost of annotating 3D pose, weakly supervised methods have emerged to limit the requirement for labeled data. [325] adapts transfer learning that mixes 2D and 3D labels from different datasets in a unified deep neutral network. To reduce the need for accurate 3D ground truth, [205] predicts only the depths of the human joints and augments 2D keypoint annotations with the ordinal relations to predict the 3D pose coordinates. A recent self-supervised approach [289] predicts geometrically coherent 3D human poses from monocular images without needing additional 3D pose annotations. As an intermediate supervision, [289] relies on 2D-to-3D pose transformation and 3D-to-2D pose projection. To cope with the limited training data, [157, 74] propose to augment the existing datasets with novel valid 3D skeletons. With the success of graph convolutional neural networks (GCNs), recent methods revisit structured 3D pose estimation. [321, 44] encode the patterns in the spatial configuration of the human joints through a GCN. However, the standard way of defining the graph according to the human skeleton can be suboptimal since the motion patterns might not follow the natural connections of body joints. To this end, [330] uses weight modulation to enable the GCN to learn diverse relational patterns between different body joints. To produce more realistic 3D human poses, generative adversarial networks (GANs) are often used. [276, 304, 281] integrate 2D pose information and camera parameters into an adversarial training scheme to discriminate whether a 3D pose generated by the network is plausible. Inspired by normalizing flows, [294] generates a diverse set of feasible 3D pose hypotheses by utilizing the known 3D to 2D projection during training.

2.3.2 Multi-View Images.

Early work [9, 28, 20] use 2D pose estimations obtained from calibrated cameras to produce 3D pose by triangulation or pictorial structures model. Recently, given a multi-view camera setup, [207] achieves this by combining 2D joint heatmap predictions from each view through 3D pictorial structure. [226, 225] integrates a loss function that adds view-consistency terms to a standard supervised loss evaluated on a small amount of labeled data. In contrast to the earlier methods, [131] investigates the epipolar geometry to recover the 3D pose from the 2D poses in a multi-view setting without requiring 3D pose annotations. [272] uses a multi-stage multi-view approach in which the 2D predictions from all views are used to reconstruct a single 3D pose, consistent with all camera views. However, in [272] there is no gradient flow from the 3D predictions to 2D heatmaps to correct the prediction in 3D. By contrast, [104] learnable triangulation methods that combine 3D information from multiple 2D views in a 3D

grid. To demonstrate that the expense of using a 3D grid is not required [223] learns a unified view-independent representation of the 3D pose disentangled from camera view-points. [220] introduces a cross-view fusion scheme to incorporate multi-view geometric priors. In case of wearable sensors, [320] employs a fuses heatmap predictions across views with the help of the orientations of IMUs. [275] relies on a proposal based architecture to aggregate features in all camera views in the 3D voxel space. An alternative way to enforce geometry consistency is novel view synthesis. In [39, 224] a view synthesis framework is proposed to learn the shared 3D representation between different views by generating the human pose from one viewpoint to another. Similarly, [143] learns disentangled representations from image pairs in wild videos without labels and yields pose and part segmentations in a novel image synthesis scheme. An alternative strategy is devised by [111] that casts this problem as a self-supervised learning task classifying whether two images depict two views of the same scene up to a rigid transformation. To address the challenges in acquisition of labeled data, [103] adopts an end-to-end approach using unlabeled multi-view data along with an independent collection of images with 2D pose annotations. However, [103] tends to overfit to a specific dataset and uses a loss term computed from the ground truth 3D poses of the Human3.6m. To overcome these limitations, [30] benefits from a self-supervised approach to estimate the 3D pose from a single image by training on unlabeled multi-view images and mixing poses across views.

2.3.3 Depth-based Methods

The availability of high-speed depth sensors and the launch of the Microsoft Kinect camera has paved the way for pose and shape estimation for articulated objects using one single depth camera, with the goal of removing ambiguities in 3D pose estimation. The work in this field either follows a generative [77, 311, 67, 312] or a discriminative model [217, 248, 218, 116, 84, 190]. [311] uses motion exemplars and matches the observed point cloud to them. [67] employs a MAP inference in a probabilistic temporal model and extends the iterative closest points (ICP) objective by modeling the constraints on movement of the subject. [312] combines the articulated deformation model with the probabilistic framework and uses a Gaussian Mixture Model to explain the observed point cloud. [248] demonstrates that Random Forest based approach can accurately predict the 3D locations of body joints along with the body parts from single depth images. While [248] assumes a uni-modal Gaussian for pixel-to-joint distribution, [218] uses kernel density estimation (KDE) and introduces Metric Space Information Gain (MSIG), a new decision forest training objective. [116] adopts offset regression and estimates the precise locations of the joints by regressing K closest joints from every pixel with the use of a random tree. More recently, [84] employs a CNN and RNN to predict partial poses in the presence of noise and occlusion. [190] encodes a single depth map in a 3D voxelized grid and estimates the per-voxel likelihood for each keypoint.

2.3.4 Temporal Methods

One way to mitigate ambiguities in pose estimation is to use monocular video as input to enforce consistency across frames. [58] relies on a dual-stream network and height maps to first predict 2D poses and then recover 3D poses based on body length, projection and continuity constraints. [266] directly regresses the 3D pose from a spatio-temporal volume of bounding boxes centered on the target frame. [328] represents a 3D pose as a linear combination of predefined basis poses and first predicts the 2D joint heatmaps that are fed into an Expectation-Maximization framework to recover the 3D pose sequence. [189] introduces the first real-time method based on model-based kinematic skeleton fitting against the 2D/3D pose predictions to produce temporally stable joint angles. Following our approach, [164, 48, 146, 97] propose to use LSTMs for exploiting temporal correlations. Differently from the other LSTM based methods [146] connects several LSTMs to extract depth information from 2D pose predictions. [97] designs a sequence-to-sequence model with LSTM units to estimate a sequence of 3D poses from 2D joint locations. [53] integrates weak supervision into a temporal framework by jointly learning from large-scale in-the-wild 2D and synthetic 3D data. [209] is the first work that shows using dilated temporal convolutions on 2D keypoint sequences can be more efficient than RNN-based models to estimate 3D poses in a video. [168] combines temporal convolutional network with an attention mechanism to capture long-range temporal relationships across frames.

2.3.5 Body Mesh Reconstruction

A closely related line of research that provides a richer information on human pose is 3D mesh reconstruction of a human body. It uses a mesh representation that is parameterized by shape and 3D joint angles. Early work [12] uses SCAPE body model for partial view completion and 3D animation of a moving person from marker motion capture data. Recent work [173] uses SMPL body model based on vertex-based skinning approach and learns the body parameters by minimizing vertex reconstruction error on large amounts of data. [27] estimates the 2D joint locations via CNN, and then fits a 3D SMPL body model to these joints while [144, 208] improve upon this work by matching the image silhouette and the silhouette projected from the SMPL model. In contrast to the previous methods, [119] employs an adversarial training in addition to minimizing the reprojection error. [200] first predicts a semantic body part segmentation and learns to map them to SMPL body model parameters. To address the weaknesses of previously introduced optimization and regression methods, [135] proposes an iterative optimization routine to fit the body model to 2D joints within the training loop, and the optimized body parameters are used to supervise the regression network. [86] formulates the problem differently and exploits the human-world interaction constraints to better estimate the human pose from monocular images. [301] is the first work using pixel-to-surface correspondence maps [79] to regress parametric pose and shape. To generate more realistic and temporally consistent body shapes, [129] relies on a temporal encoder and body parameter regressor as well as a motion discriminator. [130] extracts 2D and 3D body part features and fuses them via a part attention module to regress the SMPL shape and pose parameters.

2.4 3D Human Motion Forecasting

2.4.1 Motion prediction using RNN, GAN and GCN.

In human motion prediction, deep learning approaches have outperformed conventional methods based on Hidden Markov Models [91], Gaussian Processes for time-series analysis [285], conditional restricted Boltzmann machine [263] and dynamic random forest [149]. The inherent similarity of motion forecasting and sequence-to-sequence prediction tasks [260, 261, 16, 59, 14] have driven the research in this field towards encoder-decoder models. Pioneered by the Encoder-Recurrent-Decoder (ERD) model [65], RNNs have become the standard in sequential human motion analysis. Following [65], [107] introduces Structural-RNN (S-RNN) based on a spatio-temporal graph that encapsulates the dependencies among body joints over time. As an alternative to this, [69] leverages de-noising autoencoders to learn the spatial structure of the human skeleton by randomly removing information about joints during training. [185] designs a residual architecture that predicts velocities instead of poses. Similarly, [75] integrates motion derivatives and [326] feeds the network output back into itself to model long-term motion trajectories. In contrast to [185], [42] uses velocities as both inputs and outputs to encode different hierarchies in human dynamics at various timescales. Nevertheless, the predictions produced by RNNs still suffer from discontinuities at the transition between the last observed and the first predicted poses. Another drawback is that they often predict the mean ground truth pose in the long term.

To address these limitations, [78] integrates adversarial training to enforce frame-wise geometric plausibility and sequence-wise coherence for the pose predictions. [234] presents a GAN with several discriminators that operates on input sequences with masked joints and learns to inpaint the missing information. Similarly, [52] introduces a GAN based on multiple discriminators and spectral normalization to apply temporal attention. By contrast, [282] formulates this problem as reinforcement learning and generative adversarial imitation learning to focus on shorter sequences by breaking long ones into smaller chunks. Stochastic motion prediction methods [19, 303, 142, 314, 8, 7, 182] rely on generative models, such as VAEs and GANs, to predict multiple diverse motion sequences in the future from a single input sequence. [8, 7] achieve this through conditional variational autoencoder (CVAE) while [182] generates the motion of different body parts sequentially. [314] proposes a novel sampling strategy to produce diverse samples from a pretrained generative model. [32, 288] generate multiple trajectory and pose predictions conditioned on the scene context.

Recently, graph convolutional network (GCN) has been widely used to learn dynamic relations among pose joints. [183] encodes the spatio-temporal relationship among joints via a GCN that adaptively learns the body connectivity unlike S-RNN [107] with a fixed structure. [145] also relies on a GCN and processes the past sequence at different lengths. Recently, [167] proposes to predict the poses first at a coarse level, and then at finer levels using a multi-scale residual GCN. Similarly, [155] employs a multi-scale GCN that jointly learns action categories and motion dynamics at different granularities.

2.4.2 Attention-based human motion prediction.

Attention mechanism has been proven to be effective in machine translation and image caption generation [16, 300, 14] but has not reached its full potential in motion prediction. [262] proposes an attention mechanism to focus on the moving joints of the human body for motion forecasting. A similar idea is employed in [249] to learn the spatial coherence and temporal evolution of joints via a co-attention mechanism. Built on [183], [181] combines GCN with an attention module to learn the repetitive motion patterns in the past. Instead of modeling attention on the full body alone, as in [181], [184] fuses the predictions from three attention modules that process motion at different levels; full body, body parts, and individual joints. Differently from the previous approaches, [178] trains a computationally less intensive Transformer [14] to infer the future poses in parallel. This is achieved by a non-autoregressive strategy that comes at a cost of performance degradation.

2.4.3 Social interactions in motion prediction.

Modeling human-to-human interactions is a long studied problem focusing on social dynamics in a group of people [90] to learn human navigation. Early approaches use hand-crafted features such as Social Affinity Map (SAM) obtained by a Gaussian Mixture Model. Alternatively, [6] predicts the trajectories of pedestrians via a Markov-chain model. Based on the social forces introduced in [90], [187] models the motion of pedestrians using optical flow and [210, 302, 229] formulate the task an energy minimization problem.

Recently, the attention shifted from hand-crafted energy potentials to learning human-to-human and human-to-object interactions in a data-driven way using RNNs. [5, 239, 55, 255, 256, 1] propose social pooling layers to reason about the collective behavior of people or trajectories of vehicles. [81, 236, 138] introduce a GAN based model to predict multiple plausible future trajectories of all people in the scene simultaneously. In contrast to the previous approaches, [170] uses a contrastive method for learning socially-aware motion representations. [147] formulates this problem as a driving scenario with multiple agents and solves it via an inverse reinforcement learning strategy. [99, 35, 317] capture the dynamics in the scene via graph neural network (GNN). Recently, attention mechanisms have been adopted to learn the spatial-temporal dependencies of elements in the scene [153, 154, 171, 245].

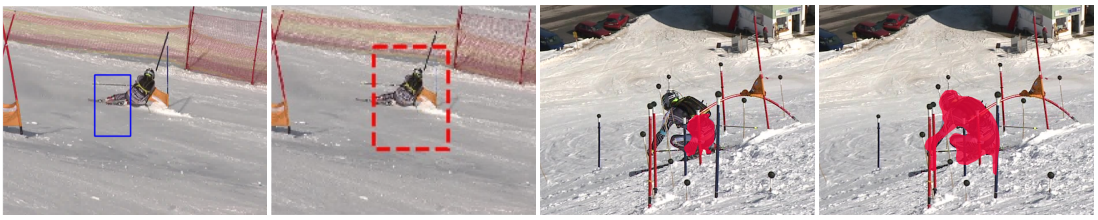
3D multi-person pose estimation has greatly benefited from contextual information. [316, 86, 113] integrate scene constraints through loss terms, such as penalizing inter-penetration of bodies, that jointly optimize the 3D pose of multiple people. [46, 80] encode social context information by passing the hidden representation of all individual poses through an attention layer. [283] encodes interaction information in a hierarchical way as instance, part and joint levels. [195] takes on a different path and casts the human interaction modeling as a 3D ego-pose prediction. It predicts the body pose of a camera wearer using the pose of the observed second person.

Existing work focus on either motion forecasting for single human in isolation or multiple weakly

interacting people in the scene. However, for modeling strong motion dependencies in a small group, these methods remain suboptimal. In this work, we address the novel task of predicting the future motion of people engaged in dyadic interactions.

3 Self-supervised Human Detection and Segmentation via Background Inpainting

Robust detection and segmentation of moving objects can now be achieved reliably in scenarios for which large amounts of annotated data are available [88]. However, for less common activities, such as skiing, it remains challenging, because the required training databases do not exist, as shown in Fig. 3.1. Self-supervised approaches [61, 132, 24, 37, 139, 49, 50, 224, 305, 166, 21, 176] promise to address this problem. However, some can only operate on video streams as opposed to single images [132, 50, 305, 176] while most others depend on strong constraints being satisfied, such as the target objects being seen against a static background.



(a) YOLOv3 (b) Ours (c) MaskRCNN (d) Ours

Figure 3.1 – **Domain specific detection and segmentation.** Our self-supervised method detects the skier well, while YOLO trained on a general dataset does not generalize to this challenging domain. Similarly, MaskRCNN trained on a general dataset sometimes misses body parts such as the upper body of the skier in (c).

To develop a more generic approach, we start from the observation that in most images the background forms a consistent, natural scene. Therefore, the appearance of any background patch can be predicted from its surroundings. By contrast, a moving person's appearance is unpredictable from the neighboring scene content and can be expected to be very different from what an inpainting algorithm would produce. We incorporate this insight into a proposal-generating deep network whose architecture is inspired by those of YOLO [222] and MaskRCNN [88] but does not require explicit supervision.

Specifically, for each proposal, we synthesize a background image by masking out the corresponding region and inpainting it from the rest of the image. The loss function we minimize favors the largest possible distance between this reconstructed background and the input image. This encourages the network to select regions that cannot be explained from their surrounding and are therefore salient. To handle the discrete nature of the proposals, we develop a Monte Carlo-based strategy to train our network. It operates on a discrete distribution, is unbiased, exhibits low variance, and is end-to-end trainable.

Our approach [120] overcomes limitations in existing self-supervised human pose estimation methods requiring static cameras [224] or monochromatic background [139, 49]. We propose a self-supervised method that operates on single images and demonstrate its effectiveness on several human motion datasets captured with cameras that are static, pan-tilt-zoom, or hand-held. We can handle large camera motions and do not require *any* manual annotation. We focus on images acquired in realistic conditions such as Ski-PTZ dataset of [226], daily human motion Handheld190k in outdoor scene and figure skating FS-Singles as well as those of the standard Human3.6m benchmark [102]. Fig. 3.1 depicts such a scenario in which our approach outperforms a state-of-the-art detection and instance segmentation method [88] trained on large annotated dataset [165]. It also outperforms existing self-supervised segmentation techniques [254, 132, 37, 50, 305]. Following standard practice in the self-supervision literature [132, 305, 224], we start from pre-trained network weights, which we fine-tune without any additional supervision in our target domain. However, we can also train from scratch with only a small performance loss. Finally, even though we focus on people, we show that our approach also applies to other kinds of target objects.

3.1 Approach

Our goal is to learn a salient person detector and segmentor from unlabeled videos acquired in as practical a setup as possible. We therefore only use raw videos or images as input and do not constrain the frame-to-frame camera motion.

3.1.1 Outline

Our basic intuition is that when people move with respect to the background, the area they occupy often looks quite different from the background. More specifically, we operate under the following two assumptions.

- **A1:** The foreground and background are distinguishable by color or texture as explained in detail by [150]. As discussed in Section 3.1.5, this can be relaxed by using optical flow.
- **A2:** Every part of the background must be uncovered more often than covered. This assumption is almost always valid in long videos depicting moving people, unlike the assumptions made in related approaches [267, 268, 132, 98, 305] that require people to

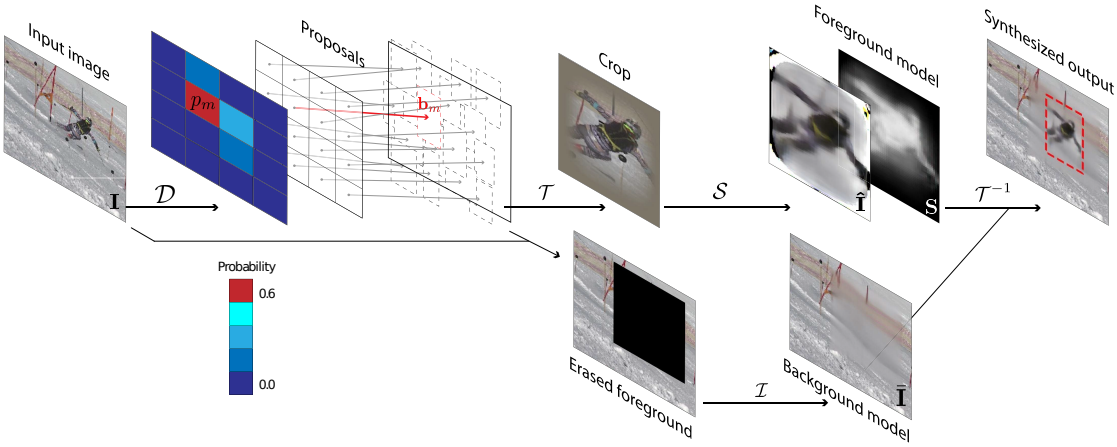


Figure 3.2 – **Our self-supervised detection and segmentation architecture.** Our model $\mathcal{F}$ passes the input image $\mathbf{I}$ to a detector $\mathcal{D}$ that proposes potential bounding boxes. One of them is passed to a spatial transformer $\mathcal{T}$ that crops $\mathbf{I}$ and the result is fed to a segmentation network $\mathcal{S}$ that outputs a segmentation mask $\mathbf{S}$ and the corresponding foreground image $\hat{\mathbf{I}}$. In a separate branch, an inpainting network $\mathcal{I}$ fills the content of the bounding box to generate a background image $\bar{\mathbf{I}}$. Finally, the inverse transformer $\mathcal{T}^{-1}$ is used to combine $\hat{\mathbf{I}}$, masked by $\mathbf{S}$, and $\bar{\mathbf{I}}$ into an image that should be similar to the original one.

move in every frame.

Hence, we cast the foreground segmentation task as one of finding an area that, when inpainted using information from the background, yields an image that is as different as possible from the true one. This makes sense under assumption A1 that people look different from the background. Assumption A2 is required to be able to train the inpainting network in a self-supervised manner. In the remainder of this section, we first present the architecture of the network we use for this purpose and then explain how we train it.

3.1.2 Network Architecture

We use the model $\mathcal{F}$ depicted by Fig. 3.2. It takes a single image $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ as input. It then resynthesizes it by sampling a candidate bounding box, cropping the corresponding image patch, and, in parallel, predicting a foreground image $\hat{\mathbf{I}} \in \mathbb{R}^{128 \times 128 \times 3}$ and a segmentation mask $\mathbf{S} \in \mathbb{R}^{128 \times 128}$ from the crop, while inpainting the cropped region to generate a background image $\bar{\mathbf{I}} \in \mathbb{R}^{W \times H \times 3}$. Finally, the foreground crop and the background image are re-composed according to the segmentation mask. Formally, this can be written as

$$\mathcal{F}(\mathbf{I}) = \mathcal{T}^{-1}(\hat{\mathbf{I}} \circ \mathbf{S}) + \bar{\mathbf{I}} \circ (1 - \mathcal{T}^{-1}(\mathbf{S})), \tag{3.1}$$

where $\mathcal{T}$ is the spatial transformer corresponding to the selected bounding box, and $\circ$ is the element-wise multiplication.

To generate the segmentation mask $\mathbf{S}$, $\mathcal{F}$ relies on a detection network $\mathcal{D}$ inspired by the YOLO architecture [222]. It divides the image into a grid and computes for each cell c a probability p_c of a detection expressed in terms of a bounding box $\mathbf{b}_c \in \mathbb{R}^4$ that defines a center and offset from the grid center. Hence, it outputs a set of C candidate bounding boxes $\{\mathbf{b}_c\}_{c=1}^C$ and corresponding probabilities $\{p_c\}_{c=1}^C$ out of which one bounding box $\mathbf{b}_c$ is sampled according to its probability p_c. A segmentation network $\mathcal{S}$ then encodes and decodes the content of $\mathbf{b}_c$ into a segmentation mask $\mathbf{S}$ and the corresponding foreground image $\hat{\mathbf{I}}$.

In a separate branch, an inpainting network $\mathcal{I}$ generates the background image $\bar{\mathbf{I}}$. Since off-the-shelf inpainting networks [204, 313] trained on large and generic datasets tend to hallucinate objects, we rely instead on a U-Net architecture [232] to implement $\mathcal{I}$, which we pre-train without using any labels and for each dataset, as discussed below. When the background $\mathbf{B}$ is known *a priori*, for example because we use a static-camera, we can simplify our architecture by removing the inpainting branch and replacing $\bar{\mathbf{I}}$ by $\mathbf{B}$. This specific case has been addressed in [49, 224] but we will show that our approach yields better results.

Algorithm 1: Our training and test procedures

input : $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$
output : $\mathbf{I}^{'} \in \mathbb{R}^{W \times H \times 3}$ // Resynthesized image
for $\mathbf{I}_n$ *in* $\{\mathbf{I}_n\}_1^N$ **do**
  $\{(\mathbf{b}_c), (p_c)\}_{c=1}^C \leftarrow \mathcal{D}(\mathbf{I}_n)$ // Bounding box prediction
  if *training* **then**
    $m \leftarrow$ *sample_2D_cell*(p_c);
  else
    $m \leftarrow \text{argmax}_c\, p_c$;
  end
  if *exists(***B***)* **then**
    $\bar{\mathbf{I}}_n \leftarrow \mathbf{B}$
  else
    $\mathbf{I}_n^{bg} \leftarrow \mathbf{I}_n$
    $\mathbf{I}_n^{bg}[\mathbf{b}_m] \leftarrow 0$
    $\bar{\mathbf{I}}_n \leftarrow \mathcal{I}(\mathbf{I}_n^{bg})$ // Background inpainting
  end
  $\mathbf{I}_n^{crop} \leftarrow \mathcal{T}(\mathbf{I}_n, \mathbf{b}_m)$
  $\hat{\mathbf{I}}_n, \mathbf{S}_n \leftarrow \mathcal{S}(\mathbf{I}_n^{crop})$
  $\mathbf{I}_n^{'} \leftarrow \mathcal{T}^{-1}(\hat{\mathbf{I}}_n \circ \mathbf{S}_n) + \bar{\mathbf{I}}_n \circ (1 - \mathcal{T}^{-1}(\mathbf{S}_n))$ // Resynthesis
end

At inference time, we simply run the trained model on the test image and pick the 2D grid cell with the highest occupancy probability $c^* = \text{argmax}_c\, p_c$. Its bounding box parameter estimates are fed into the spatial transformer $\mathcal{T}$ to crop the region of interest, which is then segmented by the segmentation network $\mathcal{S}$, as described above. The corresponding pseudo-code is given in Algorithm 1. Re-composing the image and background inpainting are only essential to train our model. They can be omitted at inference time for bounding box and segmentation mask

prediction.

3.1.3 Training Losses

Given a set of unlabeled training images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$, we first train $\mathcal{I}$ and then $\mathcal{F}$, and therefore $\mathcal{D}$ and $\mathcal{S}$, in a self-supervised manner.

To train $\mathcal{I}$, we randomly remove image regions from the training images and inpaint them from their immediate surrounding. We compare the result to the original image using an L_2 pixel-wise loss augmented by a perceptual loss, which we minimize. This works well as long as assumption A2 introduced in Section 3.1.1 holds.

Foreground vs Background

To learn the weights of $\mathcal{F}$, we minimize a weighted sum of a foreground loss L_{fg} and a background loss L_{bg}. Given the probabilistic nature of the detections generated by the detector network $\mathcal{D}$, we take them to be expected values. We write

$$L_{\text{fg}}(\mathbf{I}) = \sum_{c=1}^{C} p_c L_2\left(\mathcal{F}_c(\mathbf{I}), \mathbf{I}\right) , \tag{3.2}$$

$$L_{\text{bg}}(\mathbf{I}) = -\sum_{c=1}^{C} p_c \frac{L_2(\bar{\mathbf{I}}_c, \mathbf{I})}{\texttt{area}(\mathbf{b}_c)} \tag{3.3}$$

where L_2 is the pixel-wise mean square loss and p_c is the probability associated to bounding box $\mathbf{b}_c$ by the detector network. $\mathcal{F}_c(\mathbf{I})$ indicates the resynthesized image and $\bar{\mathbf{I}}_c$ is the background image generated by inpainting based on the sampled cell c, as discussed in Section 3.1.2. Minimizing L_{fg} encourages $\mathcal{F}_c(\mathbf{I})$ to be as similar as possible to $\mathbf{I}$, for all training images, but does not preclude the generation of bounding boxes on background objects. That is the role of L_{bg}. Because of the minus sign in front of the summation, minimizing it favors bounding boxes for which the inpainting generates an image that is different from the original one, which denotes an image location that cannot be reliably reconstructed from surrounding pixels by inpainting. Note that we normalize by dividing by $\texttt{area}(\mathbf{b}_c)$, which is the maximum number of pixels that may be different in $\bar{\mathbf{I}}_c$ and $\mathbf{I}$. This makes L_{bg} insensitive to the size of the bounding box. Without this division, L_{bg} would favor large regions, whether they contain an object or not. Nevertheless, minimizing L_{bg} by itself can favor bounding boxes with high-error density, whether or not they cover the whole person, as we will demonstrate in the ablation study of the results section.

Disentangled Training Strategy

In short, minimizing L_{bg} does not guarantee bounding boxes that fit to the person completely or precisely. By contrast, minimizing L_{fg} favors a tight fit of the segmentation mask $\mathbf{S}$ when the bounding box $\mathbf{b}_c$ is correctly located because the rest of $\mathcal{F}(\mathbf{I})$ is resynthesized using only background information, which is not relevant to the person's appearance. However, it can also yield meaningless solutions in which $\mathbf{b}_c$ is located in the background. To get the best of both world, we must therefore minimize L_{fg} and L_{bg} jointly.

Unfortunately, finding a balance between these two competing objectives by relative weighting alone has proved difficult, if not impossible. Instead, we designed a *disentangled training strategy* in which we isolate their conflicting influence on the individual network components to stabilize the training when their contributions are weighted.

Specifically, the probabilities p_c are only optimized according to L_{bg} so that L_{fg} cannot bias them towards the background regions, where it has a trivial solution. Conversely, $\mathbf{b}_c$ is optimized only according to L_{fg} to favor a tight fit without the opposite bias from L_{bg} towards high error density $\mathbf{b}_c$ with only partial coverage of the person. Similarly, $\mathcal{S}$ is optimized solely according to L_{fg} to yield the best possible reconstruction, instead of the largest distance to the background as induced by L_{bg}. This can all be computed in a single forward-backward pass by treating the excluded variables as constants in the respective objectives, that is, by cutting their gradient flow.

Full Training Loss

To speed up the convergence and to make the segmentation crisper, we introduce a perceptual loss and regularization terms in addition to L_{fg} and L_{bg}.

Perceptual Loss. We take it to be

$$L_{\phi} = \sum_{c=1}^{C} p_c L_2\left(\phi(\mathcal{F}_c(\mathbf{I})), \phi(\mathbf{I})\right), \tag{3.4}$$

where $\phi(.)$ denotes the low level features obtained by passing its input to a pre-trained ResNet18 network.

Probability Regularizer. We take it to be the L_1 loss

$$L_p = \sum_{c=1}^{C} |p_c| \tag{3.5}$$

that promotes sparsity of the non-zero probabilities.

Segmentation Mask Regularizer. We take it to be a v-shaped prior that operates on $\mathbf{S}$ and

stabilizes the early training iterations by encouraging the average value of the segmentation mask to be larger than a threshold value λ yet sparse and less noisy when exceeding this threshold. We write

$$L_v = \left| \left(\frac{1}{WH} \sum_{x}^{W} \sum_{y}^{H} \mathcal{T}^{-1}(\mathbf{S})_{xy} \right) - \lambda \right| + \lambda \,, \tag{3.6}$$

where W and H are the image width and height, respectively, and λ is set to 0.005. Note that this threshold does not control the size of the segmentation. The small value is exceeded quickly and makes L_v an L_1 prior for subsequent training iterations.

Joint Loss. In practice, we use a weighted combination of these losses, given by

$$L_{\text{joint}} = \alpha L_{\text{bg}} + \beta L_{\text{fg}} + \gamma L_{\phi} + \eta L_v + \zeta L_p \tag{3.7}$$

applied to N unlabeled images within a batch, where $\alpha = 0.1, \beta = 1, \gamma = 2, \eta = 0.25$ and $\zeta = 0.1$.

3.1.4 Monte Carlo and Importance Sampling

Computing the losses of Eqs. 3.2, 3.3, and 3.4 involves summing over the C bounding boxes proposed by the detection network $\mathcal{D}$ and their corresponding probabilities. In practice, we use $C = 64$ and back-propagating through all 64 possibilities at each training iteration makes the computation expensive. Hence, for practical purposes, it has proved necessary to reduce this cost.

Since all three losses are of the form $L = \sum_{c=1}^{C} p_c f(\mathbf{I}, \mathbf{b}_c)$, where f is a differentiable function, the simplest way to speed up the computation would be to randomly sample a small subset of the C bounding boxes and write

$$L \approx \mathbf{E}_c \left[f(\mathbf{I}, \mathbf{b}_c) \right] \text{ with } c \sim p \,, \tag{3.8}$$

where $\mathbf{E}_c$ denotes the expectation over c drawn from the categorical proposal distribution $p = \{p_1, \ldots, p_C\}$ output by the network $\mathcal{D}$. Unfortunately, the resulting loss estimate would then not be differentiable with respect to the network weights, thus precluding end-to-end gradient-based optimization.

Instead, we use Monte Carlo sampling to evaluate all three losses and introduce an auxiliary distribution q to rewrite Eq. 3.8 as

$$L \approx \mathbf{E}_c \left[\frac{p_c}{q_c} f(\mathbf{I}, \mathbf{b}_c) \right] \text{ with } c \sim q \,. \tag{3.9}$$

This approximation holds for any two probability distributions and drawing the samples according to q instead of p does not depend on the network weights, thus provides differentiability [233]. However, this Monte Carlo sampling comes at the cost of a potentially high approximation error when using only a few samples. For instance, by choosing q to be the uniform sampling

distribution $\mathcal{U}$, most of the uniformly drawn samples will have a low probability p and, therefore, negligible influence. To reduce this error, we rely on importance sampling [117, 134] to provide a low-variance unbiased estimator by taking the sampling distribution q to be similar to p. Then $p_c / q_c \approx 1$ and the fraction does not influence the result much. However, the derivatives can still be computed because q_c is a constant and the gradient of Eq. 3.9 is the same as in the likelihood ratio method [73] used in the REINFORCE algorithm [298]. We provide the details of the change of distribution and importance sampling variance in Appendix A.2.

In practice, to prevent division by very small values that could lead to numerical instability, we take the q probabilities to be

$$q_c = p_c(1 - C\epsilon) + \epsilon \,. \tag{3.10}$$

As a side effect, ϵ controls the probability that an unlikely case is chosen, which induces a form of exploration that is helpful in the early training stages of the network.

When approximating the expectation with a single sample, we can rewrite the losses introduced in Sections 3.1.3 and 3.1.3 as

$$L_{\text{bg}}(\mathbf{I}) = -\frac{p_c}{q_c} \frac{L_2(\bar{\mathbf{I}}_c, \mathbf{I})}{\texttt{area}(\mathbf{b}_c)} \,, \tag{3.11}$$

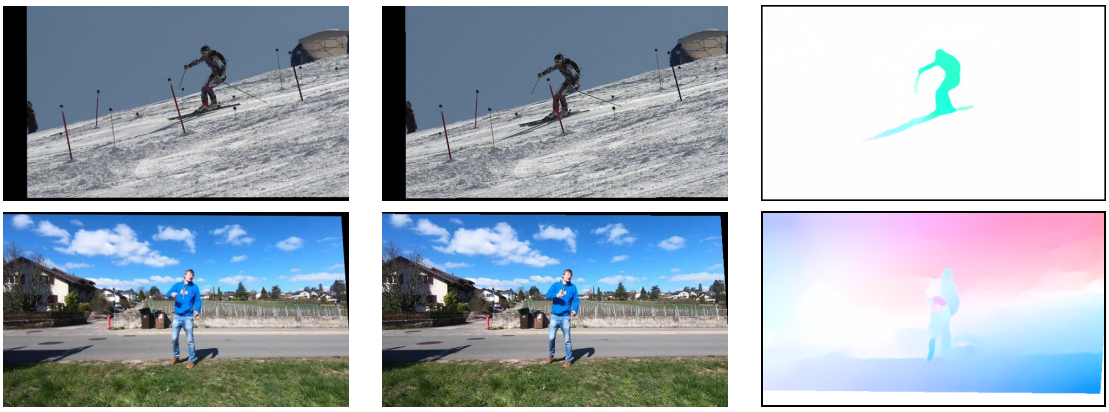
$$L_{\text{fg}}(\mathbf{I}) = \frac{p_c}{q_c} L_2\left(\mathcal{F}_c(\mathbf{I}), \mathbf{I}\right) \,, \tag{3.12}$$

$$L_{\phi}(\mathbf{I}) = \frac{p_c}{q_c} L_2\left(\phi(\mathcal{F}_c(\mathbf{I})), \phi(\mathbf{I})\right) \,, \tag{3.13}$$

with $c \sim q$ and inject these new definitions into that of the joint loss L_{joint} of Eq. 3.7.

3.1.5 Exploiting Optical Flow for Training Purposes

When video sequences are available at training time, we can exploit optical flow to help detect the foreground subject. To this end, we use optical flow images obtained by running FlowNet 2.0 [100] on pairs of consecutive frames stabilized by computing a homography using SIFT keypoints to warp one onto the other. We use the resulting optical flow image $\mathbf{I}_f$ as an intermediate supervision to our model. To this end, we train a second inpainting network $\mathcal{I}_f(\mathbf{I}_f, \mathbf{b}_c)$ to reconstruct flow images instead of regular ones. We then introduce an additional flow background objective $L_{\text{bg}}(\mathbf{I}_f)$, with the same weight as $L_{\text{bg}}(\mathbf{I})$, into L_{joint} of Eq. 3.7 that favors the $\mathbf{b}_c$s with higher inpainting loss on the flow images. This objective regulates bounding box detection by assigning higher confidence to foreground regions where the motion is clearly different from that of the background. As shown in Fig. 3.3, this lets us ignore the background motion due to a moving camera. Because we only use flow images for intermediate supervision, our model still operates at test time with single images. FlowNet is pretrained on the synthetic MPI Sintel Flow Dataset [29] and, when included, makes our approach superior to other approaches using this



(a) (b) (c)

Figure 3.3 – **Optical flow image generation on Ski-PTZ and Handheld190k.** We use a homography based on SIFT keypoints to compute rectified images that are provided as input to FlowNet 2.0. (a) Source image warped to the target scene; (b) Target image; (c) Optical flow image highlighting the moving foreground region between the source image and the target image after the background motion is eliminated. In Ski-PTZ, the optical flow images provide strong cues about the foreground object as the scene was captured by rotating cameras, making homography estimation effective. In Handheld190k, because the camera undergoes translations, the homography and optical flow estimates are less accurate, but can nonetheless improve our segmentation performance.

level of supervision.

3.1.6 Implementation Details

Overall Training. All training stages are performed on a single NVIDIA V100 32GB GPU using Adam with a learning rate of 1e-3 and batch size 16. First, the inpainting network is optimized for 200k iterations and subsequently the complete network for an additional 100k iterations. The decoding part of the synthesis network $\mathcal{S}$ uses a reduced learning rate of 1e-4, to prevent occasional diverging behavior. We use an input image resolution of 640px×360px for the Ski-PTZ, Handheld190k and FS-Singles datasets, and 500px×500px for Human3.6m.

We typically use ImageNet-trained weights to initialize our encoder components but can also train them from scratch. We rely on the Focal Spatial Transformers (FST) of [224] to speed up convergence, and expand the erased region in $\mathcal{I}$ in both dimensions by 15% of the size of that predicted by $\mathcal{D}$ to increase the chances of covering the object. Moreover, we discard location offsets outside the image and limit the offset to 1.5 times the bounding box width, as larger ones are already fully covered by the neighboring bounding boxes. We performed a grid search on the relative weights of the loss terms, the offset limits, and λ.

Detection Network. We predict one candidate bounding box relative to each grid cell in a regular

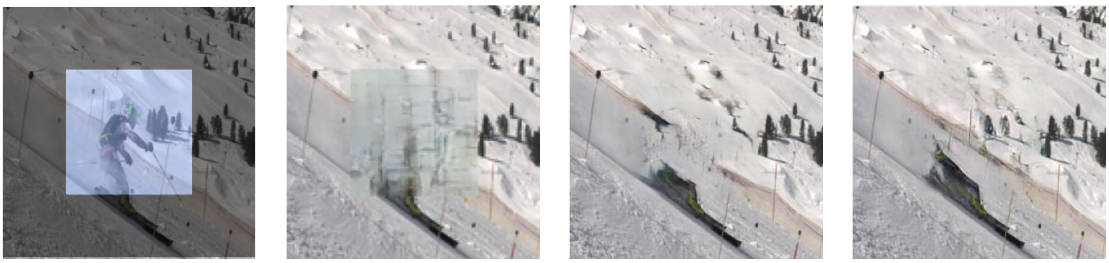
grid using a fully-convolutional architecture similar to that of YOLO [222]. We use a ResNet-18 backbone [89], which reduces the input dimensionality by a factor 16, from 128×128 to 8×8. The feature size is set to five, two for the bounding box location offset, two for scale, and one for the probability. Each feature output represents the bounding box parameters predicted by one grid cell and the offset is relative to the cell center, as shown in Fig. 3.2. The estimated probabilities p_c are forced to be positive and to sum to one by using a soft-max activation unit. To prevent this network from constantly predicting bounding boxes at the borders of the image, where the inpainting error would be high, we zero out the outer cell probabilities.

Synthesis Network. $\mathcal{S}$ is a bottleneck autoencoder based on the publicly available implementation of [225]. The encoding part is a 50-layer residual network, and the weights are initialized with ones trained on ImageNet classification. The hidden layer is 856 dimensional, split into a 600 dimensional space and a 256 dimensional space that is replicated spatially to a $512 \times 8 \times 8$ feature map to encode spatially invariant features. The decoding is done with the second half of a U-Net [232] architecture with 64, 128, 256, 512 feature channels in each stage, respectively. The final network layer outputs four feature maps, three to predict the color image $\hat{\mathbf{I}}$ and one for the segmentation mask $\mathbf{S}$.

Inpainting Network. In principle, any off-the-shelf inpainting network trained on large and generic background datasets could be used. For instance, those of [204, 313] can produce very plausible results. However, in domain-specific images, they tend to hallucinate objects, as shown in Fig. 3.4, and are therefore ill-suited for our purpose. Instead, we train $\mathcal{I}$ from scratch, by reconstructing randomly removed rectangular image regions. Note that it is acceptable for $\mathcal{I}$ not to generalize well to new scenes as it is not needed at test time. We implement it using a 6 layer U-Net model [232] with 8, 16, 32, 64, 128, 256 feature channels in each stage. It takes as input an image from which a selected bounding box region is removed and outputs the entire image with the initially removed patch re-synthesized. It is trained independently from the rest of the pipeline and separately for each dataset by feeding images with randomly occluded regions of varying sizes. In our full pipeline, the weights of the inpainting network are frozen and to remove the image evidence corresponding to the foreground person, the hidden patch in the input image to the inpainting network is selected to be the predicted bounding box expanded by 15% in both dimensions.

Importance sampling. For the importance sampling function q, we use $\epsilon = 0.001$, which makes the method numerically stable while the probability of choosing a random bounding box stays low, i.e., 6.4% for 64 cells.

Following common practice in the self-supervised segmentation literature [158, 159, 305], the final segmentation masks are generated by a CRF [140] post-processing step that uses both unary and pairwise bilateral potential terms. This CRF post-processing does not involve any training; the unary potentials are taken to be the thresholded segmentation masks predicted by our method, and we use the default values of [140] for the pairwise potentials.



(a) (b) (c) (d)

Figure 3.4 – **Off-the-shelf inpainting results on Ski-PTZ**. (a) Input image with the middle part hidden. We show the inpainting results of (b) [204], (c) [313] trained on ImageNet and (d) [313] on Places2.

3.2 Experiments

In this section, we first demonstrate the effectiveness of our approach at dealing with unusual motions acquired with PTZ cameras using the Ski-PTZ dataset of [226]. We then introduce a novel Handheld190k dataset depicting people performing 14 everyday activities and a figure skating FS-Singles dataset with different step, spin and jump combinations to demonstrate that our method can handle general moving cameras. For evaluation purposes, we provide ground truth segmentations for both. Finally, we present the experiments with different loss functions and hyper-parameter study on the Ski-PTZ dataset and analyze the influence of different aspects of our approach on the well-known Human3.6m dataset [102]. Altogether our results show that our approach outperforms the existing self-supervised segmentation techniques, including the ones that exploit temporal cues at inference time [132, 305], approaches the accuracy of supervised methods on objects they have been trained for but seen in different conditions, and outperforms them on previously-unseen objects.

3.2.1 Unusual Activity Filmed Using PTZ-Cameras

Let us first consider the Ski-PTZ dataset of [226] featuring six skiers on a slalom course. We split the videos of six skiers as four/one/one to form training, validation, and test sets, with, respectively, 7800, 1818 and 1908 frames. The intrinsic and extrinsic parameters of the pan-tilt-zoom cameras are constantly adjusted to follow the skier. As a result, nothing is static in the images, the background changes quickly, and there are additional people standing as part of the background. We use the full image as input, evaluate detection accuracy using the available 2D pose annotations and segmentation accuracy by manually segmenting 16 frames from each of the six cameras, which add up to 192 frames in two test sequences. To determine the hyperparameter values, we use 3 manually segmented frames from each of the six cameras, for a total 36 frames in two validation sequences.

In Table 3.1(left), we compare our approach to several self-supervised segmentation baselines in terms of the J- and F-measures of [212]. The former is defined as the intersection-over-union

		Ski-PTZ		Handheld190k		FS-Singles	
Method	Uses Optical Flow	J Measure	F Measure	J measure	F measure	J measure	F measure
ReDO [37]	✗	0.43	0.49	0.33	0.38	0.68	0.77
VideoPCA [254]	✓	0.54	0.61	0.47	0.49	0.55	0.69
ARP [132]	✓	0.72	0.82	0.60	0.68	0.56	0.69
Unsup-DilateU-Net [50]	✓	0.63	0.73	0.67	0.75	0.53	0.53
Unsup-Mov-Obj w/o CRF [305]	✓	0.61	0.71	0.60	0.68	0.53	0.73
Unsup-Mov-Obj [305]	✓	0.66	0.76	0.75	0.83	0.68	0.85
Ours-SV w/o optical flow	✗	0.62	0.69	0.75	0.87	0.66	0.72
Ours-SV w/ optical flow	✓	0.70	0.77	0.70	0.79	0.69	0.80
Ours-SV w/ optical flow + CRF	✓	**0.73**	**0.83**	**0.76**	**0.85**	**0.71**	**0.86**

Table 3.1 – **Segmentation results on the Ski-PTZ, Handheld190k and FS-Singles datasets**. Our method with optical flow consistently outperforms the other self-supervised methods, and ours without flow exceeds or is on par with the other baselines on all three datasets. The best results in each column are shown in bold.

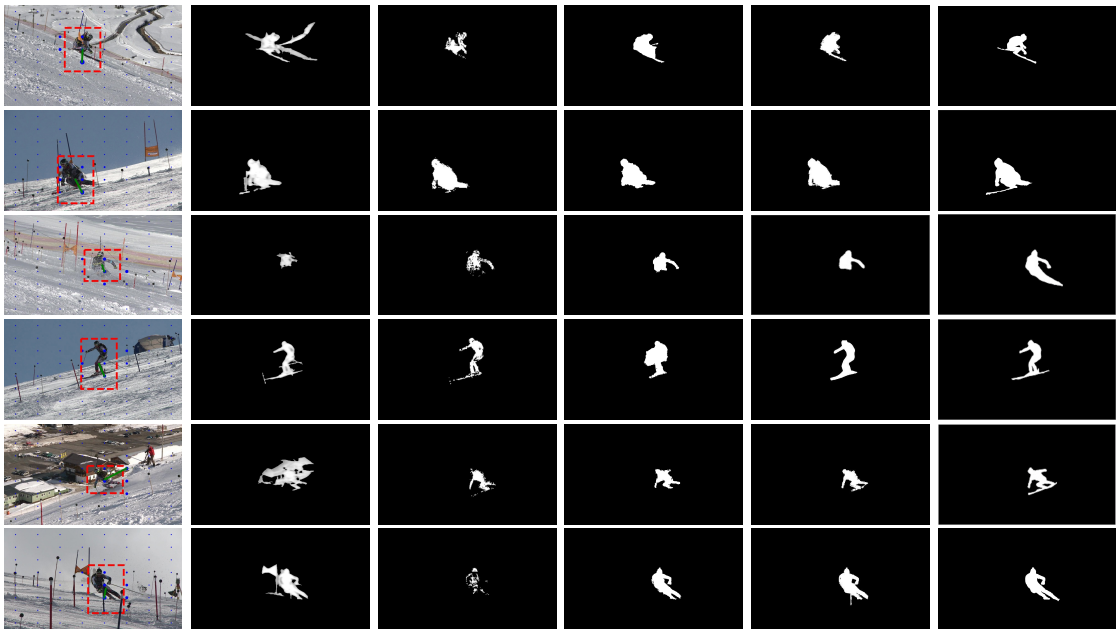
between the ground truth segmentation mask and the prediction, while the latter is the harmonic average between the precision and the recall at the mask boundaries. To be fair, we compensate for different segmentation masks quantification levels by a grid search (at 0.05 intervals) to select the best J-measure threshold for each method. Our approach with optical flow outperforms all the baselines in terms of both J- and F-measure. When not using optical flow for training purposes, our approach remains on par with other self-supervised methods despite their use of explicit temporal dependencies. In particular, the comparison to [305] without CRF post-processing shows that our method can achieve the same performance against an optical flow based method without needing a flow-based intermediate supervision. Note that all the baselines are trained on our datasets from scratch using same amount of data, except for [50] that additionally uses a segmentation mask discriminator trained on the combination of the ImageNet VID and YouTube Objects datasets. In other words, while this method is trained in a self-supervised fashion, it relies on a significantly larger amount of data than ours.

In Fig. 3.5, we compare our method qualitatively to a recent self-supervised method [24]. Note that their generative model fails to segment the foreground object alone and instead segments background objects and sometimes even the ground. Therefore, we couldn't obtain any reasonable quantitative results for [24]. This method relies on the property that foreground regions can undergo random perturbations without altering the realism of the scene. However, in the Ski-PTZ dataset, some background objects, such as poles, also satisfy this property, and the generator can choose to keep these regions. We also trained [13], another recent self-supervised method that discovers object masks by copying the selected region of the image onto another image with the goal of obtaining a realistic scene, on the Ski-PTZ dataset and obtained implausible masks for the same reason. Since these methods performed poorly on the training samples, we do not provide their quantitative results on the test data.

We provide qualitative results in Fig. 3.6. The probability distribution, visualized as blue dots whose magnitude reflect the predicted likelihood, shows clear peaks on the persons. The limitations include occasional false positives, such as the gates on the slope in close proximity to

(a) Input (b) Ours-SV (c) P-GAN [24]

Figure 3.5 – **Soft segmentation masks generated by our method and PerturbedGAN (P-GAN) [24] on training examples.** Top row: P-GAN mask generated on the Ski-PTZ dataset, the poles and snow patches are segmented as foreground. Bottom row: P-GAN mask generated on the Handheld190k dataset contains the foreground subject together with the ground they are standing on.



(a) Inp/Ours detection (b) [50] (c) [305] (d) [132] (e) Ours-SV w/ flow (f) Ground truth

Figure 3.6 – **Qualitative results on the Ski-PTZ.** Example results on the test images. (a) The detection results show the predicted bounding box with red dashed lines, the relative confidence of the grid cells with blue dots and the bounding box center offset with green lines (better viewed on screen). (b) Segmentation mask prediction of [50]. (c)Segmentation mask prediction of [305]. (d) Segmentation mask prediction of [132]. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Note that in the third row even though the skier is mostly occluded by snow, our method can detect and segment the visible part of the body. Our method is more accurate than [50] in terms of background removal and outperforms [305] in terms of correctness of the object boundary. Note that in contrast to our method, [132] uses explicit temporal cues at inference time.

the skier, reducing precision. Additional qualitative results can be found in Appendix A.1.

3.2.2 Activities Captured Using Moving Cameras

To demonstrate the effectiveness of our approach in the presence of general moving cameras, we introduce a new Handheld190k dataset captured by hand-held cameras. It features three training, one validation and one test sequences, comprising 120 855, 23 076 and 46 326 images, respectively, with a single actor performing actions mimicking those in Human3.6m. We manually annotated 112 frames in the validation and 240 frames in the test sequence to provide ground truth segmentation masks, which we believe will be useful for evaluating other self- and weakly-supervised methods. The camera operators moved laterally, to test robustness to camera translation and hand-held rotation. We provide examples of our detection and segmentation results in Fig. 3.7. Our method is robust to the undirected camera motion and to dynamic background motion, such as branches swinging in the wind and clouds moving, and to salient textures in the background, such as that of the house facade. Additional qualitative results can be found in Appendix A.1.

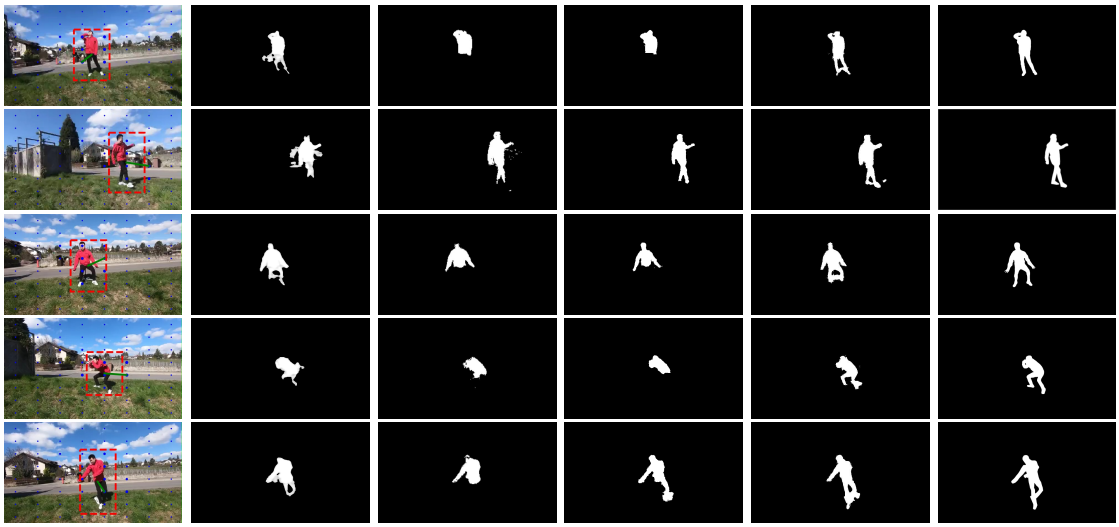
To perform a quantitative comparison, we use the 240 manually-segmented test images taken from different motion classes with the subject in many different poses. In Table 3.1(middle), we compare the results of our approach with those of the same methods as for the ski dataset. Our approach, both with and without optical flow, outperforms all the self-supervised baselines. This is even true for [50] despite its use of a much larger dataset to train a discriminator in an unsupervised fashion and also for [132] that exploits strong temporal dependencies.

We also evaluate our method on a new FS-Singles dataset composed of single men's figure skating videos collected from YouTube. The videos are captured by general moving cameras and these cameras are usually adjusted fast enough to follow the movements of the skater to keep the subject in the footage. The FS-Singles dataset contains 18 training, 2 validation and 3 test sequences with 10 613, 684 and 1656 frames and 6, 2 and 1 skaters, respectively.

The quantitative experiments on this dataset are conducted using 50 manually-segmented test images including diverse and extreme figure skating motions such as axel jump, sit spin and camel spin. In Table 3.1(right), we compare our approach to the self-supervised baselines. Our approach with optical flow outperforms all of them. The overall lower scores of the self-supervised methods on this dataset are due to the motion blur caused by the fast movements of the skaters, the low contrast between the ice and certain body parts and the audience in the background. In Fig. 3.8, we compare the segmentation results of our method to those of the second, third and fourth best-performing methods. Note that our method can accurately detect the skater, even when the scene is cluttered with the audience in the background. The failure cases of our method are mainly due to the low contrast between the ice and the hands and feet of the skater, particularly in extreme spinning poses. Furthermore, the appearance of the skater occasionally matches that of the background people, making it difficult to detect the foreground subject precisely.

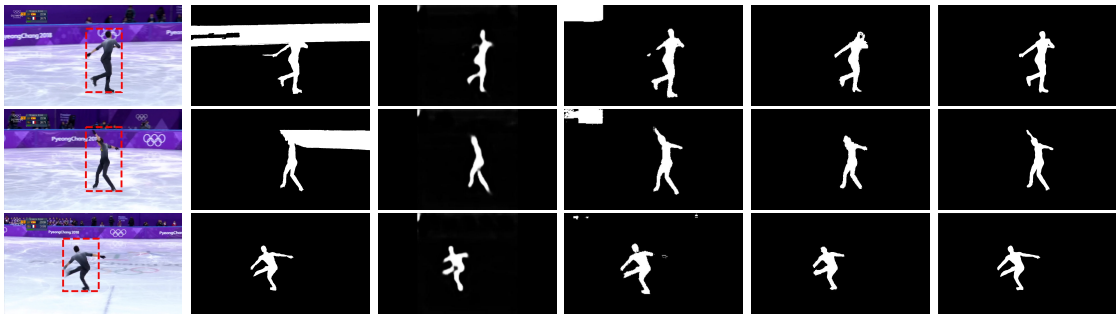
Overall, our method that relies on a single image at test time consistently yields the highest scores

on all three datasets against other self-supervised methods that operate on single images [13, 24, 37, 50] as well as the ones that require video and use temporal cues at inference time [254, 132, 305].



(a) Inp/Ours detection (b) [50] (c) [305] (d) [132] (e) Ours-SV w/ flow (f) Ground truth

Figure 3.7 – **Qualitative results on the Handheld190k.** (a) Our detection result. The blue dots coincide with the grid cell centers and their size indicates the confidence of the bounding box proposals. The selected bounding box is illustrated with a red dashed line and the center of the grid cell yielding this proposal is connected to the center of the red box through the green line. (b) Segmentation mask prediction of [50]. (c)Segmentation mask prediction of [305]. (d)Segmentation mask prediction of [132]. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Our method can segment the full body of the actor more accurately than [50, 305, 132] despite the other moving objects in the scene such as the clouds and occasionally appearing cars and pedestrians. In some frames, the shadow is also segmented since it moves with the primary object.



(a) Inp/Ours detection (b) [132] (c) [37] (d) [305] (e) Ours-SV w/ flow (f) Ground truth

Figure 3.8 – **Qualitative results on the FS-Singles.** (a) Our detection result. (b) Segmentation mask prediction of [132]. (c) Segmentation mask prediction of [37]. (d) Segmentation mask prediction of [305]. (e) Our segmentation result. (f) Ground truth segmentation mask. Our method is more accurate than [132] and [305] in terms of removing the background regions.

	Ski-PTZ		Handheld190k		FS-Singles	
Method	J Measure	F Measure	J measure	F measure	J measure	F measure
MaskRCNN [88]	**0.73**	0.77	**0.83**	**0.95**	**0.87**	**0.96**
ARP [132]	0.72	0.82	0.60	0.68	0.56	0.69
Unsup-Mov-Obj [305]	0.66	0.76	0.75	0.83	0.68	0.85
Ours-SV w/ flow + CRF	**0.73**	**0.83**	0.76	0.85	0.71	0.86

Table 3.2 – **MaskRCNN segmentation results on the Ski-PTZ, Handheld190k and FS-Singles datasets**. The direct application of off-the-shelf MaskRCNN on Handheld190k and FS-Singles datasets outperforms the self-supervised methods in Table 4.1 whereas on Ski-PTZ dataset with unusual motions, our method reaches the maximum F score and is on par with MaskRCNN in J score. This outcome is expected since MaskRCNN is trained on MS-COCO dataset that includes person class as one of the training categories.

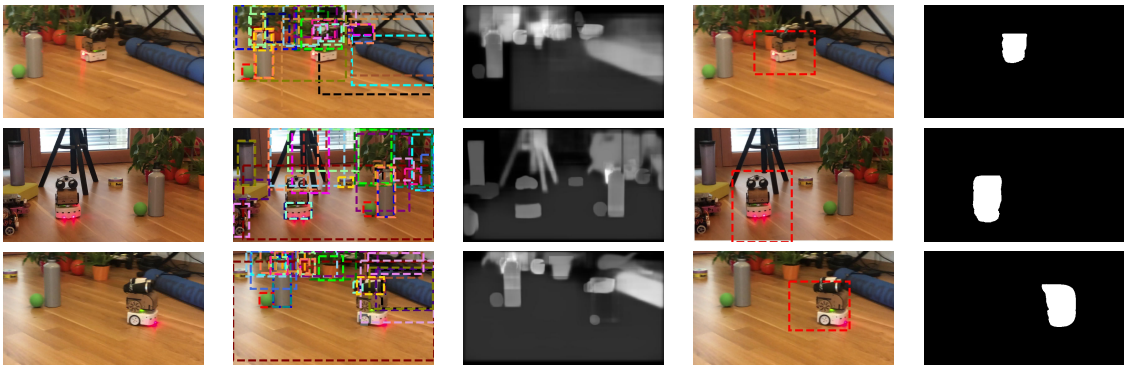
3.2.3 Comparison to Supervised Models

In this section we compare our method to MaskRCNN applied in an off-the-shelf manner. Table 3.2 reports the results of MaskRCNN trained on the MS-COCO dataset [165], which contains the person class in various sports and daily life scenarios, including skiing and skating. On the Ski-PTZ dataset, our method outperforms MaskRCNN. This demonstrates the benefits of self-supervised learning to handle unusual scenarios, where the data differs significantly from that in the publicly-available datasets. On the Handheld190k and FS-Singles datasets, MaskRCNN yields the highest scores, which is not surprising as the test sequences look similar to those in the MS-COCO training set. However, many other object categories are not present in the MS-COCO dataset. In those cases, simply exploiting MaskRCNN becomes non-trivial, because it provides class-specific segmentations, and thus cannot directly handle unknown objects.

To nonetheless evaluate the performance of MaskRCNN in this challenging scenario, we captured an indoor scene featuring many static objects and a moving robot that we aim to segment with a hand-held camera. Fig. 3.9 compares the detections and segmentation masks output by MaskRCNN for all MS-COCO classes with those obtained with our method. Because the custom robot cannot be associated with any existing MS-COCO category, MaskRCNN tends to split it into multiple objects. Obtaining a consistent mask of the robot would then require parsing these multiple detections. By contrast, our self-supervised approach naturally generalizes to such a previously-unseen object.

3.2.4 Ablation Study

In Table 3.3, we investigate the effectiveness of different mask priors introduced in Section 3.1.3 and ImageNet pre-training on the validation part of the Ski-PTZ dataset. Although L_1 yields better segmentation masks than L_2, it tends to suppress the mask values too strictly, which causes convergence problems. This is mitigated by our L_v prior, which achieves the highest scores in



(a)Input (b)MaskRCNN (c)MaskRCNN (d)Ours-SV (e)Ours-SV

Figure 3.9 – **Qualitative results of MaskRCNN on a moving robot sequence captured with a handheld camera.** MaskRCNN generally fails to detect the moving robot as a single object and does not yield a segmentation mask with high confidence.

	Ski-PTZ	
Setting	J Measure	F Measure
Ours-SV w/o optical flow w/o prior	0.51	0.53
Ours-SV w/o optical flow w/ L_2 prior	0.61	0.69
Ours-SV w/o optical flow w/ L_1 prior	0.62	0.69
Ours-SV w/o optical flow w/ L_v prior	**0.67**	**0.73**
No ImageNet pre-training, L_v prior	0.60	0.63
Unsupervised pre-training [299], L_v prior	0.62	0.68

Table 3.3 – **Analysis of the mask prior effect and ImageNet pre-training on the Ski-PTZ validation sequences**. We demonstrate the influence of using mask priors to supress the noise surrounding the foreground object and have clear-cut masks. At the bottom part of the table we show the results of using random weights and features from [299] instead of using weights from ImageNet pre-training.

all measures, with consistently reliable results. This demonstrates that imposing regularization on the segmentation masks allows us to obtain sharper masks, removing the noise around the foreground object. We repeated the Ski-PTZ experiment without optical flow extension four times with the best-performing configuration and computed the mean and std on the validation sequences; the J- and F-measure are consistent, respectively, $0.67 \pm 0.004, 0.73 \pm 0.006$.

Table 3.3 also shows the comparison of using ImageNet or self-supervised weights for network initialization, with only a small performance drop for the latter.

Furthermore, Table 3.4 compares the performance of our method for different values of hyper-parameters, where the subscript of **b** corresponds to the minimum and maximum size of the bounding box and λ used in our L_v prior is the percentage of the pixels that should be activated in the segmentation mask.

	Ski-PTZ	
Setting	J Measure	F Measure
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.0005}$	0.55	0.55
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.001}$	0.57	0.62
$\mathbf{b}_{[0.1,0.5]}$, $L_{v,\lambda=0.005}$	0.54	0.56
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.0005}$	0.61	0.70
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.001}$	0.60	0.65
$\mathbf{b}_{[0.20,0.5]}$, $L_{v,\lambda=0.005}$	0.61	0.67
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.0005}$	0.57	0.64
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.001}$	0.57	0.64
$\mathbf{b}_{[0.30,0.5]}$, $L_{v,\lambda=0.005}$	0.57	0.63
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.0005}$	0.61	0.69
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.001}$	0.61	0.68
$\mathbf{b}_{[0.20,0.60]}$, $L_{v,\lambda=0.005}$	0.62	0.68
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.0005}$	0.62	0.67
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.001}$	0.59	0.66
$\mathbf{b}_{[0.20,0.70]}$, $L_{v,\lambda=0.005}$	0.62	0.65
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.0005}$	0.61	0.68
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.001}$	**0.67**	**0.73**
$\mathbf{b}_{[0.20,0.80]}$, $L_{v,\lambda=0.005}$	0.61	0.66

Table 3.4 – **Hyper-parameter study on the Ski-PTZ validation sequences**. In this table we analyze the effectiveness of our hyper-parameter choice for the minimum and maximum bounding box sizes (given in square brackets as $\mathbf{b}_{[scale_{min},scale_{max}]}$) as well as the threshold λ for the L_v loss. We conduct these experiments using our approach without optical flow.

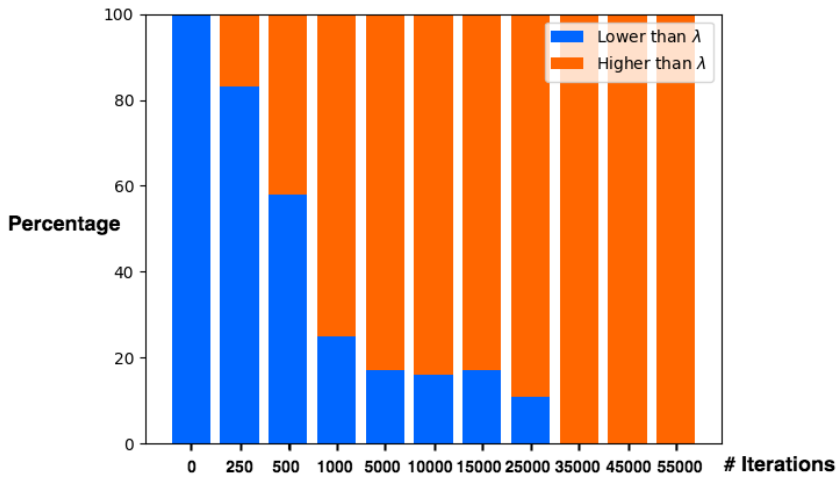


Figure 3.10 – **Impact of the segmentation mask regularizer of Eq. 3.6.** Early in the training, a high percentage of masks have a mean value lower than λ. When the model converges, all masks have a mean value above this threshold.

Fig. 3.10 depicts the influence of the segmentation mask regularizer of Eq. 3.6. It shows the percentage of segmentation masks that have lower and higher mean values than λ at different training stages. At convergence, the mean segmentation mask value is always higher than λ. Without this regularizer, the mean value of the segmentation mask would grow even larger, causing the mask to incorporate noise and fuzzy regions around the person. Since λ doesn't have to match the exact size of the object, setting it to a small value suffices to trigger the generation of masks early on. We use the same value $\lambda = 0.005$ in all our experiments.

People in a Controlled Environment. We evaluate different aspects of our approach using the Human3.6m dataset [102] that comprises 3.6 million frames and 15 motion classes. It features 5 subjects for training and 2 for validation, seen from different viewpoints against a static background and with good illumination.

On this dataset, we first study the importance of our model choices for training and probabilistic inference. As shown in Fig. 3.11(a), using uniform sampling instead of importance sampling does not converge. Fig. 3.11(b) illustrates that joint training of $\mathcal{D}$ with L_{fg} and L_{bg}, instead of our disentangled one, produces bounding boxes that are too large. Fig. 3.11(c) shows that using only the background objective leads to small detections that miss the subject and (d) that direct regression without multiple candidates diverges. These failure cases are representative of the behavior on the whole dataset. To explore an alternative strategy to Monte Carlo-based sampling, we replaced the importance sampling in our method with the categorical reparameterization used in [49]. Since both strategies approximate the same objective, they had similar outcomes with a difference in the convergence speed and detection performance. To this end, we tried

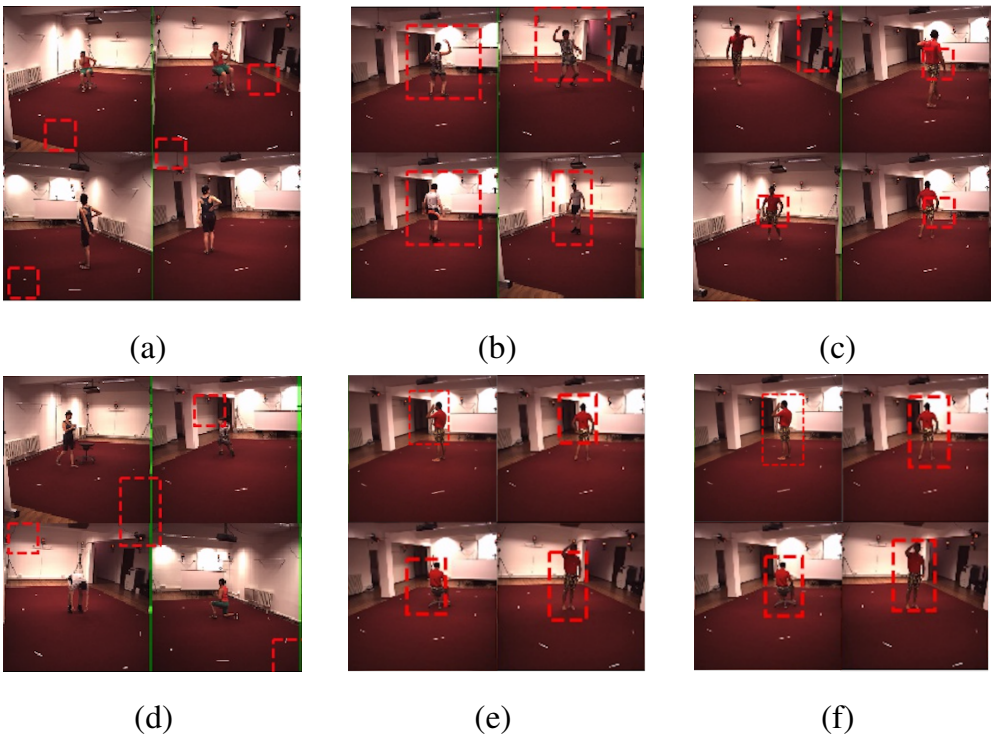


Figure 3.11 – **Ablation study on Human3.6m.** (a) Uniform sampling does not converge. (b) Joint training of L_{fg} and L_{bg} (c) only L_{bg} (d) direct regression of a single bounding box using L_{fg} and L_{bg} (e) Gumbel-Softmax (f) Ours-SV.

Gumbel-Softmax distribution [109]. We found out that setting the temperature to 0.1 yielded the best results. Increasing this value has a similar effect as increasing the ϵ in Eq. 3.10 and approaches uniform sampling. Our experiments show that Gumbel-Softmax based categorical reparameterization did not lead to faster convergence and in fact degraded the detection performance as shown in Fig. 3.11(e). Our method delivers a $\text{mAP}_{0.5}$ score of 0.58 which is significantly higher than the $\text{mAP}_{0.5}$ score of 0.30 obtained by using Gumbel-Softmax as our sampling strategy. Furthermore, our importance sampling approach is simpler than [49] and is an unbiased estimator. It does not need custom layers that behave differently in the forward and backwards passes during optimization, which is the case for the Gumbel-Softmax categorical reparameterization. Please note that direct comparison to [49] is not possible since it requires monochromatic backgrounds. Therefore, it does not apply to the Ski-PTZ, Handheld190k and FS-Singles datasets and was demonstrated only on simple synthetic cases, such as MNIST and Atari games, with multiple objects that go beyond the scope of our approach. Finally, Fig. 3.11(f) demonstrates that our full model using the disentangled training strategy and importance sampling can accurately detect the person and estimate tighter bounding boxes.

In Table 3.5, we evaluate detection accuracy on Human3.6m and Ski-PTZ. Note that our method delivers an $\text{mAP}_{0.5}$ score that is significantly better than that of the general YOLO [222] detector trained on MS-COCO dataset. On the left side of Table 3.5, we compare our detection accuracy

to that of a very recent self-supervised deep learning method [224]. Our slightly lower accuracy stems from not explicitly assuming a static background, which [224] does. While valid in a lab, this assumption results in total failure in outdoor scenes with moving backgrounds. Notably, our method is robust to undirected camera motion and to dynamic background motion, and works equally well for the very different domains of skiing and every-day activities.

Human3.6m dataset	
Method	$mAP_{0.5}$
NSD [224]	**0.710**
Ours-SV	0.580

Ski-PTZ	
Method	$mAP_{0.5}$
YOLOv3 [222]	0.155
Ours-SV	**0.520**

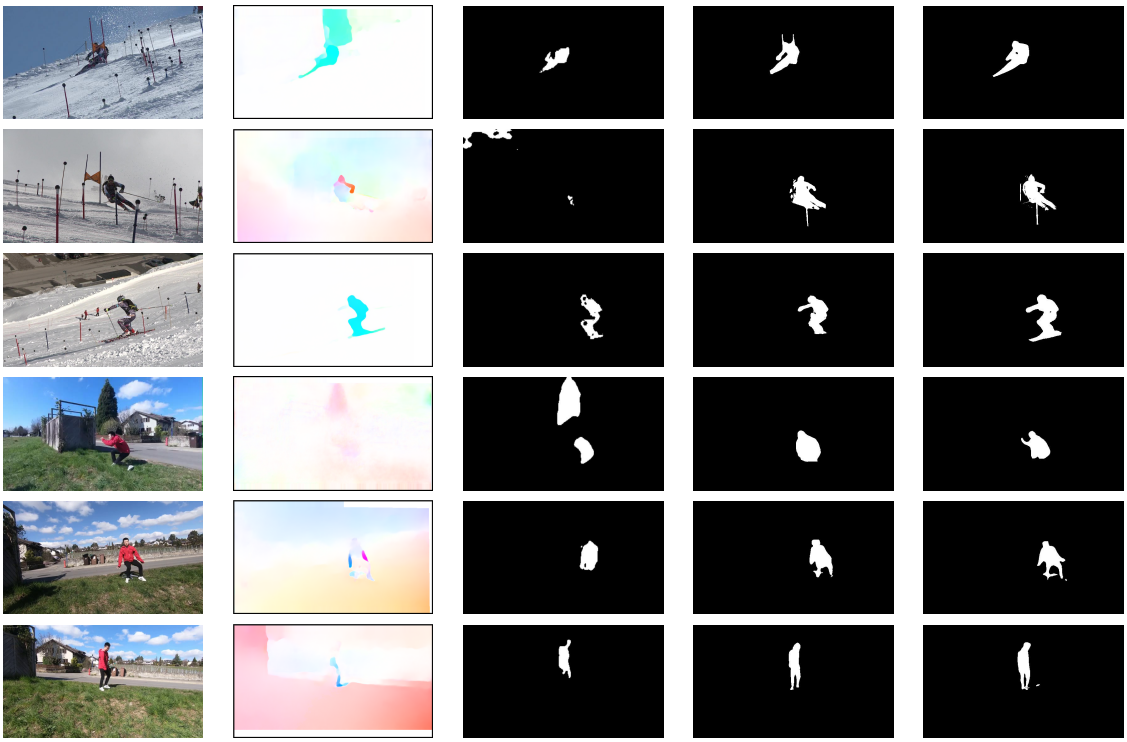
Table 3.5 – **Detection results on the Human3.6m and Ski-PTZ datasets.**

3.2.5 Discussion

Optical flow. As noted in [305], the motion-based segmentation methods that require computing the optical flow between consecutive images can be error-prone due to the irregular or insufficient movement of the object. This gives us leverage against approaches that rely only on optical flow since our method can reliably detect the foreground object from single RGB images and uses optical flow only as an extension during training time. In Fig. 3.12, we present possible failure cases that can occur when the optical flow partially covers the object due to its static parts. Since the inpainting module in [305] tries to reconstruct the masked optical flow, it is prone to errors whenever the optical flow image is unreliable. It can be seen that our method can accurately segment the object in this case. Hence, based on our experimental evidence in Table 3.1, optical flow should always be used if available and in combination with the RGB image.

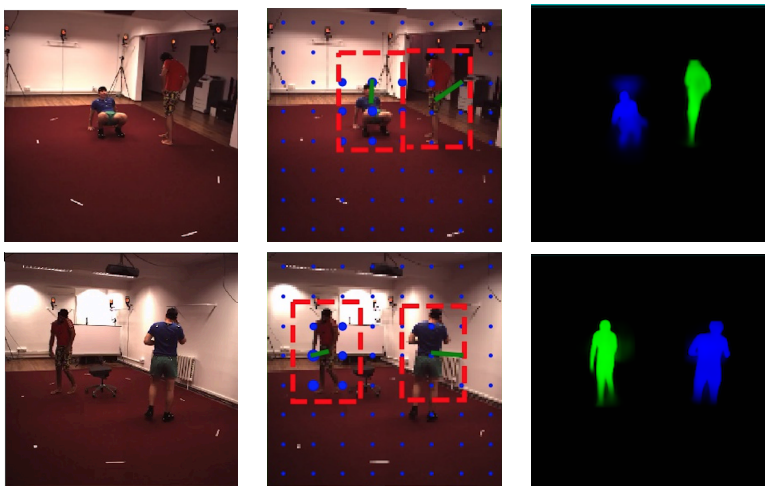
Multiple people. Although our focus is on handling single objects or persons, our probabilistic framework can handle several at test time by sampling more than once. Fig. 3.13 shows the predicted cell probability as blue dots whose size is proportional to the probability. The fully-convolutional architecture operates locally and thereby predicts a high person probability close to both subjects. As a result, both the detection and segmentation results remain accurate as long as the individuals are sufficiently separated. Note that the model used for this experiment was still trained on single subjects. In future work, we will attempt self-supervised training of multiple interacting people, which has so far only been established in controlled environments.

Other object categories. In this section, we investigate the applicability of our method to standard benchmarks with other object categories. The existing object detection datasets SegTrackV2 and FBMS59 comprise multiple objects, which we do not support. Therefore, we demonstrate the qualitative performance of our method on the standard DAVIS2016 [212] benchmark that consists of various object categories such as car, cow and goat. DAVIS2016 contains 30 training and 20 testing sequences, which are very short compared to other benchmarks suitable for deep-learning based methods. We follow the standard procedure and use the validation sequences for evaluation.



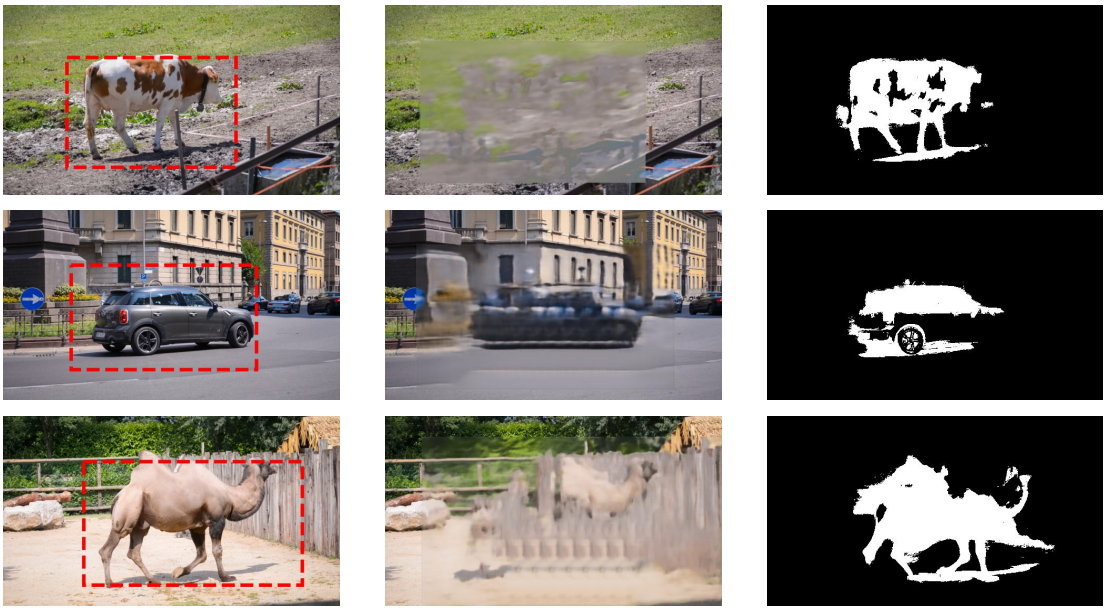
(a)Input (b)Optical flow (c) [305] (d)Ours-SV (e)Ours-SV w/ flow

Figure 3.12 – **Optical flow failure.** When the optical flow image cannot be used to find the complete outline of the subject, for example because some part of their body is static, our method can still segment the moving object from a single RGB image, whereas [305] tends to yield poor results. To highlight the effect of using optical flow, we present the raw segmentation predictions of [305] and ours, before the post-processing step.



(a) Input (b) Detection (c) Segmentation

Figure 3.13 – **Multi-person detection and segmentation results**, generated by sampling our model multiple times. As the model is trained on single persons this only works for non-intersecting cases.



(a)Input/Ours (b)Inpainting result (c)Segmentation

Figure 3.14 – **Examples of qualitative results on DAVIS2016 [212] validation sequences.** Top row: successful segmentation of the moving object. Middle row: partially successful case in which the background scene carries information about the moving object's location. Bottom row: poor segmentation result that occurs when our inpainting network can resonstruct certain parts of the moving object due to its slow motion.

Since our method does not require any annotations, we train and test on the validation sequences with an average of 70 frames per video. So far, we have evaluated our method on datasets with human subjects. Therefore we pick non-human object categories in the DAVIS2016 validation dataset to show that our method is not specific to a particular object type. As shown in Fig. 3.14, our performance on DAVIS2016 varies, depending on the length and footage of the sequence. However, we do not expect our method to compete with approaches tuned for short video snippets. Many of these short sequences include objects that move slowly, remaining mostly in the same image region. This makes them easy to inpaint, thus violating our assumptions A1 and A2 (Section 3.1.1). Fig. 3.14(top) shows a successful segmentation result on a longer sequence with a moving object. Fig. 3.14(middle) illustrates a partially successful case that occurs when the location of the object changes with the background elements and the content of the scene in a short video clip provides significant clue about the reconstruction of the foreground object. In Fig. 3.14(bottom), we present a failure case that occurs when our method is applied to very short videos with negligible object displacement. In this case, our inpainting network can reconstruct the foreground object together with the background region, which causes holes in the regions of the segmentation mask that are already reconstructed by the inpainting network.

In short, DAVIS2016 features only few videos per category, with each video being short, making them ill-suited to deep-learning based self-supervised approaches that exploit large unlabeled

video collections. By contrast, we contribute new benchmarks with manual annotations for quantitative evaluation and three very different settings with significantly more and longer training videos that can be used to evaluate future self-supervised deep learning-based segmentation methods.

3.3 Conclusion

We have proposed a self-supervised method for object detection and segmentation that lends itself for application in domains where general purpose detectors fail. Our core contributions are the Monte Carlo-based optimization of proposal-based detection, new foreground and background objectives, and their joint training on unlabeled videos captured by static, rotating and handheld cameras. Our experiments demonstrate that, even if trained only on single persons, our approach generalizes to multi-person detection, as long as the persons are sufficiently separated. In contrast to many existing solutions [18, 235, 254, 132], our approach does not exploit temporal cues at test time. In the future, we will integrate temporal dependencies explicitly, which will facilitate addressing the scenario where multiple people interact closely, by incorporating physics-inspired constraints enforcing plausible motion.

4 Human Detection and Segmentation via Multi-view Consensus

Robust detection and segmentation of moving people can now be achieved reliably in scenarios for which large amounts of annotated data are available. However, for less common activities, such as skiing, it remains challenging, because the required training databases do not exist. Self-supervised approaches [61, 132, 24, 37, 49, 50, 224, 305, 166, 21, 176] promise to address this problem. However, most of them depend on strong constraints, such as the target objects being seen against a static background, or rely on object localization and object-boundary detection networks pre-trained with supervision, which limits their applicability.

In this chapter, we propose to remove these limitations by using a multi-camera setup for training purposes and explicitly encoding the 3D geometry of the scene. At inference time, our trained network can then handle single images and outperforms earlier techniques, as shown in Fig. 4.1. Our algorithm [121] can be applied to any object as long as the two assumptions from [150] hold: foreground and background are distinguishable by color or texture; every part of the background must be visible more often than not.

Using several cameras complicates data acquisition but only in a limited way because both synchronization and calibration are well understood tasks for which off-the-shelf solutions exist. In practice, for static cameras, this has to be dealt with only once before a filming session using well-known techniques [85, 63] and requires far less effort than manually annotating images. For moving cameras, SLAM methods are now robust enough to perform the calibration automatically and fast in the wild [329, 293]. Hence, there are many applications in which training with multiple cameras makes perfect sense, especially those with unusual activities for which large training databases are not available.

To leverage multi-view training data as weak supervision, we introduce the object proposal strategy depicted by Fig. 4.2. Candidate 2D bounding boxes are produced by a network that can be trained in an unsupervised fashion. They are used to vote into a 3D proposal grid, and multi-view geometry constraints are then imposed to align proposals from different views in a differentiable manner. To train the resulting network, we sample a 3D proposal, deconstruct and reconstruct the image in each view using the corresponding 2D bounding box, and compare the

Figure 4.1 – **Leveraging multi-view consistency at training time** to segment the salient object from *single* images at inference time and to outperform baselines exploiting temporal consistency [132], optical flow [305, 120] and novel view synthesis [224].

resulting resynthesized images to the original ones.

While our self-supervised learning strategy leverages multiple views during training, the resulting model can be used for detection and segmentation in monocular images acquired by moving cameras and featuring unknown backgrounds. Our contributions can be summarized as follows.

- We introduce a self-supervised end-to-end trainable object detection and segmentation approach that explicitly leverages 3D multi-view geometry as weak supervision during training.
- It comprises a 3D object proposal framework that enables to enforce prediction consistency across views without having to introduce additional loss terms.

To show that our approach can handle unusual activities and fast motion, we demonstrate it on the skiing dataset depicted by Fig. 4.1, captured by moving cameras, on a small dataset acquired using hand-held cameras, as well as on the more standard Human3.6m dataset [102] acquired using fixed cameras. Note that our multi-view supervision differs from weak supervision in video object segmentation as it does not require any segmentation annotation. Hence, our method relates to self-supervised approaches. We show that the proposed multi-view training increases single-image accuracy performance at inference time, which allows us to outperform the previous single-view [132, 305, 50, 176, 120] and multi-view [224] approaches.

4.1 Approach

Our goal is to develop a self-supervised algorithm that generates a bounding box and the corresponding segmentation mask from a single image. However, whereas earlier methods use videos from a single camera for training purposes, we want to demonstrate that using calibrated and

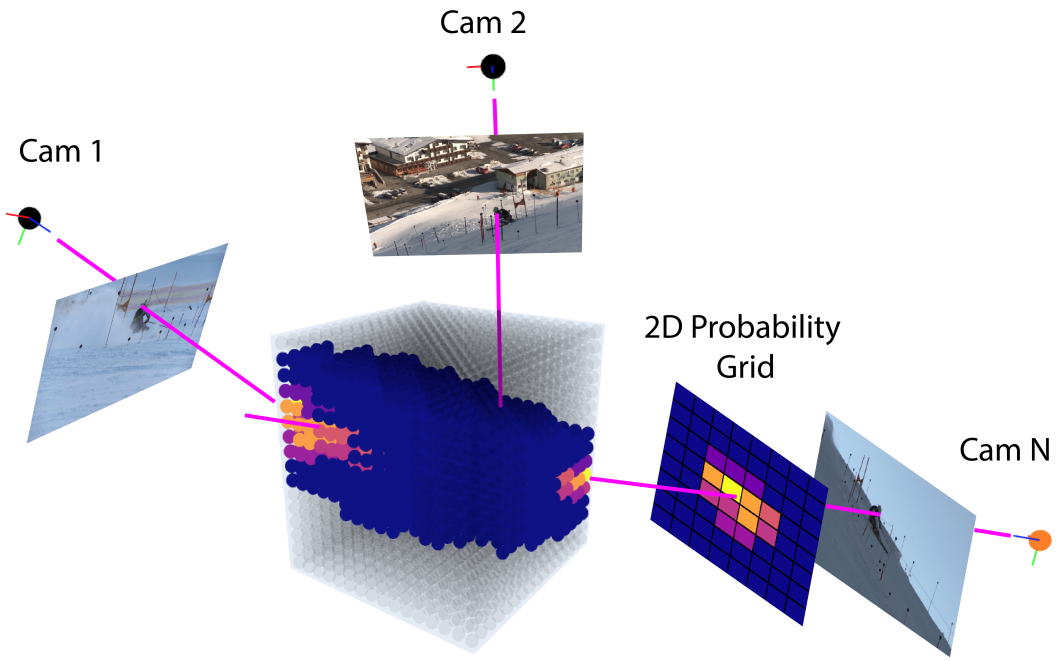


Figure 4.2 – **3D Proposal Grid.** The consensus between individual views is found on a 3D voxel grid (black) as a combination of 2D probabilities projected on the voxels (rainbow colors). Once a coarse grid location is found, a fine offset is found via offset prediction and 3D triangulation (purple lines).

synchronized cameras for training purposes increases performance. Therefore, let us assume that we have videos acquired by $Z > 1$ calibrated and synchronized projective cameras. For each z between 1 and Z, camera z captures image $\mathbf{I}_z$ and its behavior is modeled by a 3×4 projection matrix $\mathbf{P}_z$.

4.1.1 Multi-View Self-Supervised Training

Let us now turn to the task of exploiting such multi-view data to train our detection and segmentation network. Because we ultimately aim to perform single-view 2D detection and segmentation, our approach produces bounding boxes and segmentation masks for each individual view. Nevertheless, we exploit multi-view geometry to better constrain the training process and enforce consistency across the views. Furthermore, we do this without requiring additional loss terms that would make the process more complex and force us to carefully weigh these additional terms against the original ones. To this end, our training algorithm goes through the following steps

1. We use a network $\mathcal{F}$ to compute a probability map for 2D bounding boxes over an image grid for each view c. These probability maps are used to vote in a 3D grid for potential 3D locations of these bounding boxes.

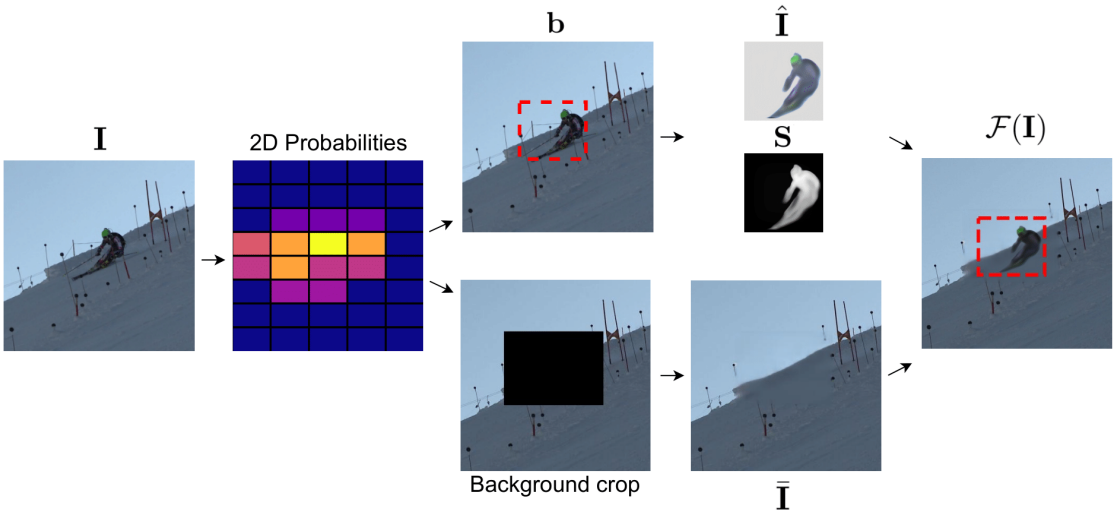


Figure 4.3 – **Overview of the underlying single-view self-supervised segmentation pipeline.** This figure summarizes our starting point, the single view approach. It predicts 2D occupancy probabilities, an associated bounding box, and a foreground mask within this window. It is trained to reconstruct the input image by pasting the foreground region underneath the mask on a background image obtained by inpainting the predicted bounding box.

2. We sample individual 3D voxels in that 3D grid according to the resulting probability density. This corresponds to one 2D bounding box for each view.
3. We compute the 3D center and object height that best agree with these 2D bounding boxes in a least-square sense.
4. We project the resulting 3D center and height in each view to define new 2D bounding boxes, keeping the original width of the sampled boxes.
5. These boxes are then used to evaluate the loss function associated to $\mathcal{F}$ in each image.

Multi-view consistency is achieved both by sampling the 3D proposal grid and adjusting the 2D bounding boxes. Hence, we do not require additional losses to enforce consistency. This is a central element of our approach because, as observed in [226], such loss terms tend to favor degenerate solutions that are consistent but wrong. This is something our ablation study confirms. In the remainder of this section, we describe these steps in more detail.

Bounding Boxes in Individual Views

Let us consider the network $\mathcal{F}$ of [120], which we use as the backbone of our approach. It takes an image $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ as input and resynthesizes it. In the process, it produces a probability map over a grid, encoding for each cell i the probability p_i that a bounding box $\mathbf{b}_i$ at this location contains a person. As depicted by Fig. 4.3, resynthesis is achieved by sampling a candidate bounding box, cropping the corresponding image patch, and, in parallel, predicting a foreground image $\hat{\mathbf{I}} \in \mathbb{R}^{128 \times 128 \times 3}$ and a segmentation mask $\mathbf{S} \in \mathbb{R}^{128 \times 128}$ from the crop, while inpainting the

cropped region to generate a background image $\bar{\mathbf{I}} \in \mathbb{R}^{W \times H \times 3}$. We then re-compose the foreground crop and the background image according to the segmentation mask. Formally, this can be written as

$$\mathcal{F}(\mathbf{I}) = \mathcal{T}^{-1}(\hat{\mathbf{I}} \circ \mathbf{S}) + \bar{\mathbf{I}} \circ (1 - \mathcal{T}^{-1}(\mathbf{S})), \tag{4.1}$$

where $\mathcal{T}$ is the spatial transformer corresponding to the selected bounding box, and $\circ$ is the element-wise multiplication. This allows one to train $\mathcal{F}$ in a self-supervised fashion, by comparing the reconstructed image to the input one.

Consistent Sampling using a 3D Proposal Grid

To link 2D detections across views, we construct a 3D proposal grid with V voxels centered at the point nearest to the optical axes of all cameras in the 3D world coordinate system, as shown in Fig. 4.2. For each voxel j of that grid, we compute its center $\mathbf{v}_j \in \mathbb{R}^3$, together with a probability of occupancy q_j, discussed below.

Since we know the camera matrix $\mathbf{P}_z$ for each image $\mathbf{I}_z$, we can project the center $\mathbf{v}_j$ of each 3D voxel into it. The projected center will fall into image grid cell $i^z(j)$ to which $\mathcal{F}$ has associated a probability $p^z_{i^z(j)}$, as discussed at the beginning of Section 4.1.1. We repeat this operation over all images and all voxels and sum the resulting log probabilities for each voxel. We then normalize the resulting probability density over the 3D grid so that it integrates to one. Formally, this can be written as

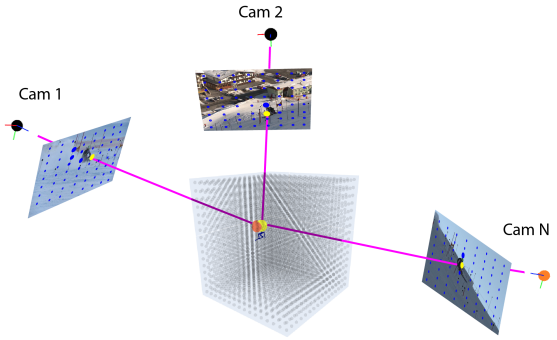
$$q_j = \frac{1}{G} \exp\left(\sum_z \log(p^z_{i^z(j)})\right), \tag{4.2}$$

where G is a normalization constant easily computed on a discrete grid of finite dimensions.

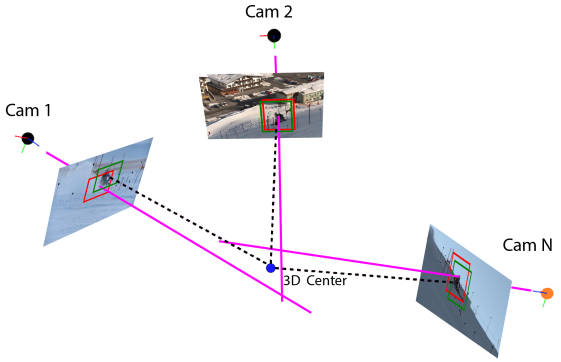
To train our network in a self-supervised fashion, we then sample one voxel location j according to the distribution in Eq. 4.2. The sampled voxel then corresponds to one bounding box candidate in each view, inherently encouraging consistency across the views as illustrated in Fig. 4.4(a). This consistency, however, is only a partial one because each view still predicts the precise location and dimensions of its own bounding box. Hence, the final bounding boxes may still disagree. To prevent this, we explicitly enforce geometric consistency as discussed below.

Enforcing Geometric Bounding Box Consistency

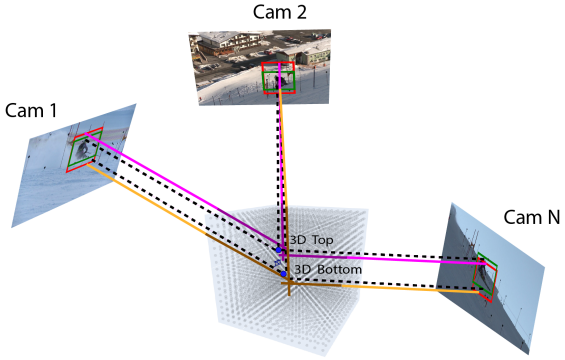
To enforce geometric consistency between the bounding boxes from different views, we want to ensure that their 2D centers all match the same point in 3D and that their 2D heights correspond to the same 3D size. In other words, we want to modify the bounding box locations so that the new ones have consistent 2D centers and heights and we want to achieve this with as little displacement as possible. Since the cameras are often set in a rough circle pointing at the subject, enforcing height consistency makes sense because the camera up directions are aligned.



(a) Multi-view voting.



(b) Making the bounding box centers consistent.



(c) Making the bounding box heights consistent.

Figure 4.4 – **Finding bounding boxes that are view consistent.** (a) The blue dots overlaid on each view represent the initial 2D probabilities and vote in the 3D grid along their respective lines of sight. As a result, the yellow 3D voxel becomes very likely to be sampled. (b) The red bounding box drawn in each view is the initial prediction and the purple line of sight is going through the bounding box center. The 3D center is the point closest to all these lines and its re-projection in the images becomes the center of the new bounding boxes, shown in green. (c) The red bounding boxes represent the initial prediction and the purple and orange lines indicate the line of sight going through the bounding box top and bottom points. The 3D top and bottom locations are taken to be the point closest to purple and orange lines respectively. Their re-projection in the images become the top and bottom middle points of the new bounding boxes, shown in green.

Only when the camera angle varies, as in drone footage taken from arbitrary angles, should the height constraint be replaced. We do not constrain the bounding-box width because the left-right direction of cameras is not aligned unless the cameras are parallel. This makes the width view dependent, as in Fig. 4.1 where the skier's projection is wider in some views.

In essence, we seek to *project* the bounding boxes to new ones that satisfy the center and height constraints and that will be used by the network to evaluate its objective function during its forward pass. It is therefore essential that this projection be differentiable such that the backward pass can be carried out during training.

Adjusting Bounding Box Centers.

As shown in Fig. 4.4(b), we use the lines of sights defined by the 2D centers of the bounding boxes, find the 3D point closest to all of them, and use its re-projection into the images as the modified center for the bounding boxes. For each view z, the line of sight $\mathbf{l}_z$ in image $\mathbf{I}_z$ can be expressed as

$$\mathbf{l}_z = \mathbf{M}_z^{-1} [u_z, v_z, 1]^T , \tag{4.3}$$

where $\mathbf{M}_z$ is the 3×3 matrix formed by the first 3 columns of $\mathbf{P}_z$ and u_z, v_z are the 2D pixel coordinates of the bounding box center in $\mathbf{I}_z$. Hence, finding the point closest to all the $\mathbf{l}_z$ amounts to solving a least-squares problem, which can itself be achieved by solving a linear system of equations and is therefore differentiable. In practice, we use a differentiable least-squares implementation for this purpose and provide its details in Appendix B.1.

Adjusting Bounding Box Heights.

As shown in Fig. 4.4(c), we similarly use the midpoints of the top and bottom parts of the bounding boxes in each view to predict two new intersection points, one for the top and one for the bottom of the bounding box in 3D. We then take the distance between the re-projections of these points into the image to be the new height of the bounding boxes. As before, this is a differentiable operation.

Training

Because our 2D bounding boxes are made to be consistent, we can train our network by minimizing the same loss as in the single view approach of [120], except for the fact that we jointly compute it over several images, and do not require to introduce an additional loss to enforce consistency.

More specifically, we minimize the weighted sum of two loss functions $L_{\text{bg}}(\mathbf{I}_1, \ldots, \mathbf{I}_Z)$ and $L_{\text{fg}}(\mathbf{I}_1, \ldots, \mathbf{I}_Z)$. L_{bg} accounts for the fact that a region containing a moving foreground object is unlikely to be well re-synthesized by the inpainter and is critical to train the network to place the bounding box at the right location in each image. L_{fg} gauges how well $\mathcal{F}$ resynthesizes the complete original images and is minimized when the segmentation mask fits the salient object as

well as possible within the sampled bounding box. In practice, they are taken to be

$$L_{\text{bg}}(\mathbf{I}_1,\ldots,\mathbf{I}_Z) = -\sum_{z=1}^{Z} r_j \frac{\|\bar{\mathbf{I}}_z - \mathbf{I}_z\|^2}{area(\mathbf{b}^z_{i^z(j)})}\,, \tag{4.4}$$

$$L_{\text{fg}}(\mathbf{I}_1,\ldots,\mathbf{I}_Z) = \sum_{z=1}^{Z} r_j \|\mathcal{F}(\mathbf{I}_z) - \mathbf{I}_z\|^2\,, \tag{4.5}$$

where $area(\mathbf{b}^z_{i^z(j)}) \in \mathbb{N}_0$ is the area of the bounding box obtained by sampling voxel j and enforcing geometric consistency. As in [120], the sampled voxel is obtained by importance sampling, and r_j is the ratio of the probability q_j, from Eq. 4.2, by its importance sampling probability. In addition to these loss terms, as [120], we use an L_1 prior on $\mathbf{S}$ to favor a crisp segmentation, and compute Eq. 4.5 not only on pixel color but also on learned features. Additional details on the sampling, hyper-parameters, training and network architectures are provided in Appendix B.1.

4.1.2 Single-View Inference

Once trained using multiple views, our model can detect and segment the salient object from single RGB images at inference time without any further changes. We run our network on the image and simply choose the 2D grid cell with the highest occupancy probability. Its bounding box parameter estimations are fed into the spatial transformer $\mathcal{T}$ to crop the region of interest, which is encoded into the corresponding segmentation mask and foreground, and decoded into the reconstructed image as illustrated in Fig. 4.3.

4.2 Experiments

Unlike that of [224], our self-supervised approach is designed to work using multiple-cameras that can move. In this section, we show that it does, yet outperforms [224] even when the background is static. Furthermore, we show that using multiple cameras for training purposes delivers the hoped-for performance boost over the previous monocular approaches [132, 305, 50, 176, 120].

4.2.1 Images and Metrics

We first describe the image datasets we work with and then the metrics we use for comparison purposes.

Images acquired using moving cameras.

The Ski-PTZ dataset of [226] features six skiers on a slalom course. We use the official training/validation/test sets that split the 12 videos of six skiers as four/one/one, with, respectively, 7800, 1818 and 1908 frames. The pan-tilt-zoom cameras constantly adjust to follow the skier.

Nothing remains static, the background changes quickly, and there are additional people standing in the background. The cameras were calibrated using static scene markers *without* any markers or keypoints on the skier's body. We use the full image as input and evaluate detection accuracy using the available 2D pose annotations and segmentation accuracy of the 300 labeled frames in the test sequences. To pick the hyperparameters, we use 36 labeled validation frames (3 frames each from six cameras and two sequences). Due to the large distance between cameras and subject, the 3D proposal grid has 16^3 voxels with cuboid side length of 8 meters.

To demonstrate the applicability of our method to scenes without an initial camera calibration, we use the Handheld190k dataset [120] captured by three hand-held cameras that translate and rotate in an unscripted fashion. It comprises three training, one validation, and one test sequences. They all feature one person performing actions mimicking the human motions in an outdoor environment with a changing background. We used OpenSFM [1] to calibrate 4200 frames from the training set using and tested on the same images as [120]. The 3D proposal grid has 16^3 voxels with cuboid side length of 12 meters.

Images acquired using Static Cameras.

To compare against algorithms requiring a static background, we evaluate our approach in the more controlled environment of the Human3.6m dataset [102]. It was acquired using four static cameras and comprises 3.6 million frames and 15 motion classes. It features 5 subjects for training and 2 for validation, seen from different viewpoints against a static background and with good illumination. The 3D proposal grid consists of 10^3 voxels, with cuboid side length of 4 meters.

Metrics. We report our segmentation scores in J- and F-measure as defined in [212]. The former is defined as the intersection-over-union (IoU) between the ground truth segmentation mask and the prediction, while the latter is the harmonic average between the precision and the recall at the mask boundaries. The detection scores are calculated in terms of $mAP_{0.5}$, the mean probability of having an IoU of more than 50%. Different segmentation algorithms set the foreground-background threshold differently. Hence, to allow a fair comparison, we perform a line search from 0 to 1 with a step-size of 0.05, selecting the optimal value for all baselines and variants for each individual dataset.

4.2.2 Comparative Results with Moving Cameras

Fig. 4.5 depicts qualitative results on the Ski-PTZ dataset and we report the corresponding quantitative results using 4 cameras in Table 4.1, in which we use the scores reported in [120] for the baselines. [2]

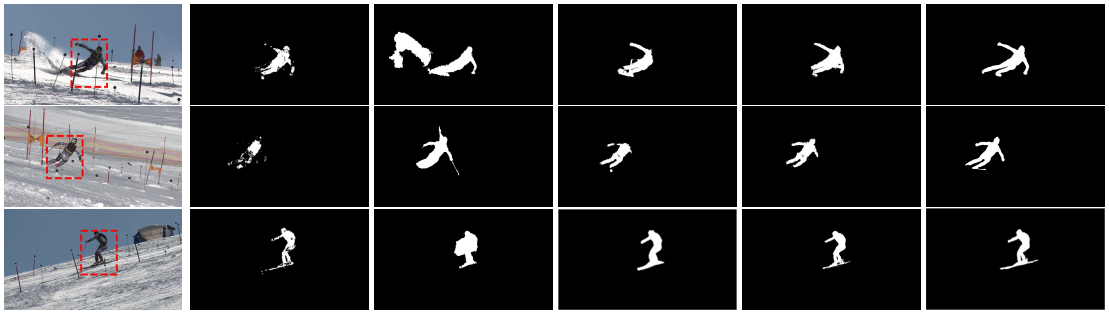
We outperform all existing single-view self-supervised segmentation approaches [132, 305, 50,

[1] https://www.opensfm.org/

[2] The implementation of [176] was provided by the authors.

	Ski-PTZ		
Method	J Score	F Score	Run-time (sec)
Chen et al. [37]	0.37	0.42	0.11
Stretcu et al. [254]	0.51	0.56	0.02
Lu et al. [176]	0.51	0.60	0.60
Katircioglu et al. [120]	0.61	0.67	0.24
Rhodin et al. [224] + [120]	0.61	0.70	0.23
Croitoru et al. [50]	0.62	0.72	0.15
Yang et al. [305] w/o CRF	0.61	0.71	0.32
Yang et al. [305]	0.67	0.77	1.12
Katircioglu et al. [120] w/ flow	0.69	0.79	0.24
Koh et al. [132]	0.70	0.80	107.4
Ours-MVC	**0.71**	**0.83**	0.17

Table 4.1 – **Multi-view consistency segmentation results on the Ski-PTZ.** We compare against the single-view approaches and a modified version of the multi-view approach of [224].

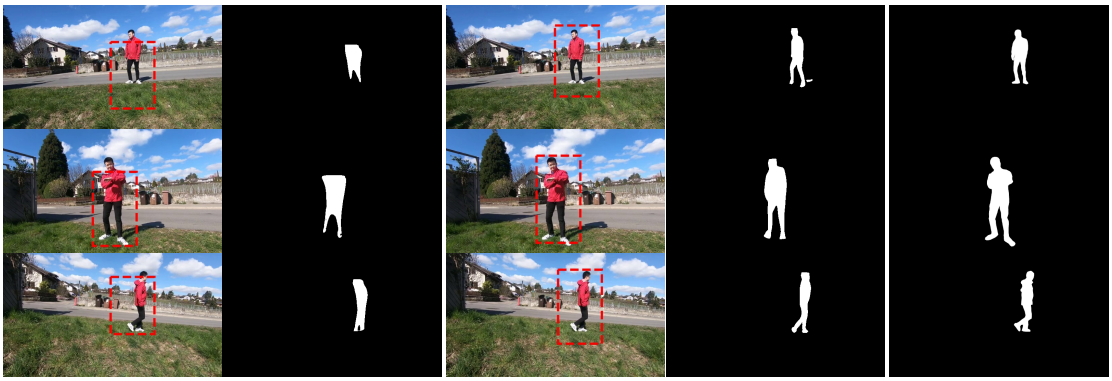


(a)Inp/Ours detection (b) [305] (c) [132] (d) [120] (e) Ours-MVC (f) GT

Figure 4.5 – **Multi-view consistency qualitative results on the Ski-PTZ dataset.** (a) Input images with our predicted bounding box overlaid in red. (b,c,d) Segmentation masks predicted by three of our baselines. (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Note the quality of our predicted masks even though, unlike the methods of [132] and [305], we do not use explicit temporal cues at inference time.

120, 254, 37, 176] while being comparatively fast. For completeness, we also report results for [305] without CRF post-processing. This shows that a great deal of the method's performance comes from such post-processing, which we do not require. Note that, in contrast to [120] with flow and [132], our approach does not require computing optical flow. Unlike DAVIS2016 [212], our datasets feature large camera motions with quick background changes, which causes methods such as [176] to often merge portions of the background and the human.

The only other self-supervised multi-view approach for which a public implementation is available is that of [224]. Unfortunately, it requires background images as an input, which are not given in this case and are not trivial to create because the cameras rotate and zoom. To do so anyway, we use the single-view approach of [120] to produce background images that we can feed to



(a) Katircioglu et al. [120] (b) Ours-MVC (c) GT

Figure 4.6 – **Multi-view consistency qualitative results on the Handheld190k dataset.** (a) The detection and segmentation mask results of [120] trained and tested on single images. (b) The predictions of our model trained using 3-camera multi-view consistency and tested on single images. (c) Ground truth. Our results are generally more accurate, which justifies the effort invested in calibrating the cameras.

the network of [224] for multi-view training. As can be seen in Table 4.1, this modified version of [224] does slightly better than [120] in F score terms but remains far behind our method. The method that comes closest to ours is that of [132], which operates on the whole sequence and is therefore prohibitively slow as discussed below. By contrast our approach operates on a single image and does not require motion information.

The inference times for each method are shown in the last column of Table 4.1 and computed using code that is either publicly available or that the authors made available to us privately. All except those of [254, 132] were obtained using a single NVIDIA TITAN X Pascal GPU. Since [254, 132] are designed to run on CPU, the inference for them is computed on Intel(R) Xeon(R) Gold 6240 CPUs. The tailored optimization approach of [132] that comes closest to our results is three orders of magnitude slower than our approach because it tracks several patches over time. Unlike [305], our method does not require optical flow computation or CRF post-processing which brings a five-fold speedup. Our computational complexity is similar to that of [120, 224] since the triangulation time is negligible. The training time of our model on the Ski-PTZ is approximately 8 hours whereas that of [224] and [120] are 14 and 7.5 hours, respectively.

We also evaluate our method on Handheld190k trained using 4200 images from multiple views and compare against the network of [120] trained using the same 4200 images. We obtain a J-score of 0.66 instead of 0.64 and an F-score of 0.77 instead of 0.71, again showing the importance of multi-view consistency. Our method benefits from multi-view information obtained in an automated off-the-shelf manner, particularly in tightly fitting to the subject, as shown in Fig. 4.6. In short, the improvement demonstrated here highlights the previously untapped potential of multi-view constraints for self-supervised segmentation.

		Human3.6m	
Method	Training Type	Background Assumption	mAP
Katircioglu et al. [120]	single-view	dynamic	0.57
Rhodin et al. [224]	multi-view	static	0.71
Ours-MVC	multi-view	dynamic	**0.85**

Table 4.2 – **Multi-view consistency comparative results on the Human3.6m dataset.** Our detection accuracy improves in terms of $mAP_{0.5}$.

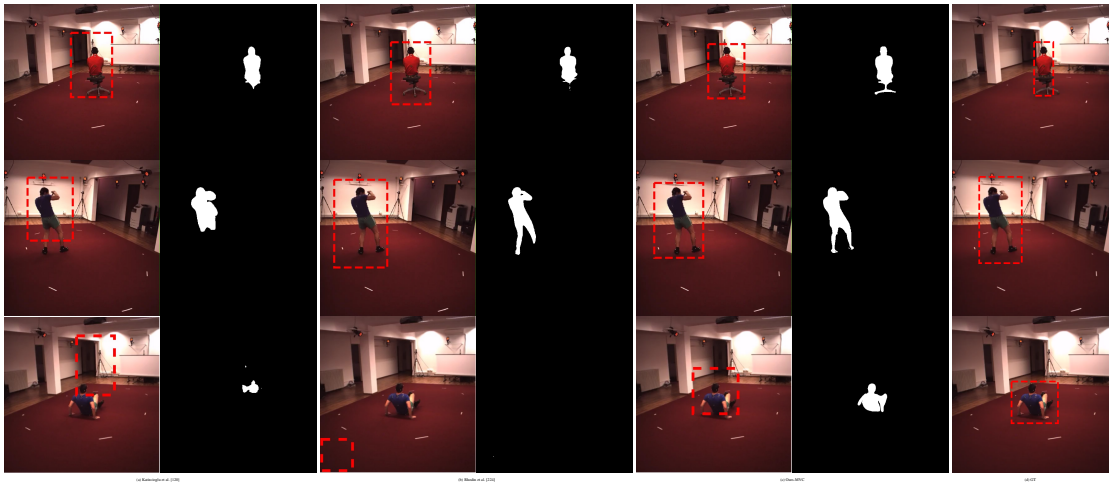


Figure 4.7 – **Multi-view consistency qualitative results on the Human3.6m dataset.** (a) The detection and segmentation results of [120] trained and tested on single images. (b) The results of [224] trained with a pair of camera views and tested on single images. (c) Our predictions obtained from the model trained with the 4-cam multi-view consistency and tested on single images. (d) Ground truth. Our method consistently detects the person whereas [120, 224] occasionally produce inconsistent results, such as the failed detections in the last row.

4.2.3 Comparative Results with Static Cameras

In the previous example, we had to modify the multi-view self-supervised algorithm of [224] to make it work on images with a moving background. To evaluate the original version instead, we compare on the Human3.6m dataset and report the results using again 4 cameras in Table 4.2. As in the Ski-PTZ case, we outperform it and, this time, the difference cannot be caused by any background modification we made. This is somewhat surprising because the method of [224] assumes a constant static background, which is the case here, whereas ours is learned without any such constraint. We attribute this result to the explicit consistency of bounding box positions in 3D and the background inpainting constraint. The latter triggers when part of the subject is outside the bounding box leading to correctly segmented legs while the method of [224] has trouble distinguishing the skin and floor color when in shadow, as depicted in Fig. 4.7. See additional qualitative results in Appendix B.2.

In Table 4.2, we also report the result of [120], that is, our backbone network run on single views.

	# Cam	*Ours w/o VC*	*Ours w/o HC*	*Ours w/ TC*	*Ours w/ WC*	*Ours-MVC*
J Score	2	0.66	0.67	0.66	0.61	0.66
	3	0.68	0.70	0.68	0.68	**0.71**
	4	0.68	0.70	0.67	0.68	**0.71**
	5	0.67	0.67	0.67	0.68	0.69
	6	0.66	0.70	0.67	0.67	0.68
F Score	2	0.73	0.73	0.73	0.65	0.75
	3	0.75	0.77	0.75	0.74	0.81
	4	0.75	0.79	0.75	0.77	**0.83**
	5	0.74	0.74	0.74	0.75	0.78
	6	0.73	0.78	0.73	0.74	0.76

Table 4.3 – **Multi-view consistency ablation study on the Ski-PTZ**. We test variants of our approach while using varying numbers of cameras.

	# Cam	*Ours w/o VC*	*Ours w/o HC*	*Ours w/ TC*	*Ours w/ WC*	*Ours-MVC*
mAP	2	0.73	0.74	0.74	0.73	0.75
	3	0.78	0.80	0.79	0.79	0.82
	4	0.79	0.83	0.82	0.84	**0.85**

Table 4.4 – **Multi-view consistency ablation study on the Human3.6m dataset.** We test variants of our approach while using varying numbers of cameras.

The performance drops, which once again highlights the usefulness to exploit multiple views for training when they are available.

4.2.4 Ablation Study

We compare the following variants of the multi-view constraints of Section 4.1.1: *Ours-MVC* denotes the full model that employs all the steps shown in Fig. 4.4. *Ours w/o HC* excludes the bounding box height consistency depicted by Fig. 4.4 (c). *Ours w/o VC* leaves out both the center and height adjustment of Fig. 4.4 (b,c) and enforces only consistent sampling. *Ours w/ WC* imposes a bounding box width consistency in addition to the full model. Finally, *Ours w/ TC* is a baseline that replaces the view consistency with a triangulation loss minimizing the distance between the lines joining the centers of the camera and predicted 2D bounding box.

In Table 4.3 and Table 4.4 we report results as a function of the number of cameras we used. We can use only 2 cameras but the best results are obtained for 3 or 4. Beyond that, additional cameras add little new information while taking more space in the training batches, resulting in less diverse batches and lower performance. The numbers for the different variants in Fig. 4.4 show that all the elements we have incorporated into our approach contribute positively and that the one we have purposely ignored—constraining the width—would degrade performance. Crucially *Ours w/ TC* also performs worse, hence substantiating our claim that imposing consistency constraints using the projection mechanism of Section 4.1.1 is crucial to our success.

3D Grid Size	**Ski-PTZ** J-Score
[10 × 10 × 10]	0.64
[16 × 16 × 16]	**0.68**
[24 × 24 × 24]	0.66

3D Grid Size	Human3.6m $mAP_{0.5}$
[6 × 6 × 6]	0.76
[10 × 10 × 10]	**0.79**
[16 × 16 × 16]	0.76

Table 4.5 – **Influence of voxel resolution.** The numbers in square brackets indicate the number of voxels in the 3D proposal grid and we use 4 cameras.

We also analyzed the influence of the voxel resolution on the reconstruction accuracy. Table 4.5 shows that a 10^3 cube is more accurate than a 6^3 cube while going to a 16^3 does not bring further improvements in Human3.6m dataset. The 0.01 lower mAP may indicate that learning a discrete distribution on the 3D grid may be less efficient on larger spaces. However, as the ski footage covers a wider area, a 16^3 cube yields the best performance on the Ski-PTZ.

4.3 Conclusion

We have presented a self-supervised detection and segmentation technique that exploits multi-view geometry during training to accurately separate foreground from background in single RGB images at inference time. It outperforms the earlier work on the challenging Ski-PTZ, depicting unusual activities captured with moving cameras, and on Human3.6m, acquired with static cameras. We have focused on scenes with a single salient object. However, our method has the potential to handle multiple objects by sampling more than one proposal as long as they are not overlapping. Our future work will be in this direction.

5 Learning Latent Representations of 3D Human Pose with Deep Neural Networks

In spite of much recent progress, estimating 3D human pose from a single ordinary image remains challenging because of the many ambiguities inherent to monocular 3D reconstruction. They include occlusions, complex backgrounds, and, more generally, the loss of depth information resulting from the projection from 3D to 2D.

Recent regression-based methods can directly and efficiently predict the 3D pose given the input image [156] or images [266] but often ignore the underlying body structure and resulting joint dependencies, which makes them vulnerable to ambiguities. Several methods have recently been proposed to account for these dependencies [237, 102, 160]. In particular, by leveraging the power of Deep Learning, the method of [160] achieves high accuracy. However, it involves a computationally expensive search procedure to estimate the 3D pose.

Since pose estimation is much better-posed in 2D than in 3D, an alternative way to handle ambiguities is to use discriminative 2D pose regressors [34, 40, 57, 71, 106, 194, 214, 216, 274, 295, 306] to extract the 2D pose and then infer a 3D one from it [27, 60, 310, 328]. This however also involves fitting a 3D model in a separate optimization step, and is thus more expensive than direct regression.

In this chapter, we demonstrate that we can account for the human pose structure within a deep learning regression framework. To this end, we propose to first train an overcomplete autoencoder that projects body joint positions to a high dimensional space represented by its middle layer, as depicted by Fig. 5.1(a). We then learn a CNN-based mapping from the image to this high-dimensional pose representation as shown in Fig. 5.1(b). Finally, as illustrated in Fig. 5.1(c), we connect the decoding layers of the autoencoder to the CNN, and fine-tune the whole model for pose estimation. This procedure is inspired by Kernel Dependency Estimation (KDE) in that it can be understood as replacing the high-dimensional feature maps in kernel space by autoencoder layers that represent the pose in a high-dimensional space encoding complex dependencies between the different body parts. However, our approach has the advantage over

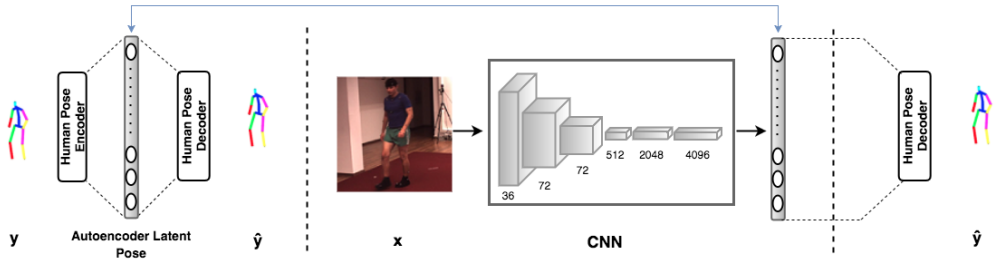


Figure 5.1 – **Overview of our approach.** (a) An autoencoder whose hidden layers have a larger dimension than both its input and output layers is pretrained. In practice we use either this one or more sophisticated versions that are described in more detail in Section 5.1.1 (b) A CNN maps either a monocular image or a 2D joint location heatmap to the latent representation learned by the autoencoder. (c) The latent representation is mapped back to the original pose space using the decoder.

KDE of directly providing us with a mapping back to the pose space, thus avoiding the need for a computationally expensive optimization at test time. Altogether, and as will be demonstrated by our experiments, our framework [122] enforces implicit constraints on the human pose, preserves the human body statistics, and improves prediction accuracy.

With the growing availability of large training datasets, 2D pose estimation algorithms have achieved tremendous success [194, 216, 295] by relying on Deep Learning. They exploit the fact that finding 2D joint locations in a color image is easier than direct 3D pose prediction, which is fraught with depth ambiguities. To leverage the well-posedness of the 2D localization problem, we therefore use the reliable 2D joint location heatmaps produced by [194] as input to our autoencoder-based regression architecture. We show that this improves 3D pose accuracy upon direct regression from an RGB image. We further show that our autoencoder-based regression approach scales to very deep architectures and achieves competitive performance when used with ResNet architecture [89].

Because we can perform 3D pose-estimation using a single CNN, our approach can easily be extended to handling sequences of images instead of single ones. To this end, we introduce two LSTM-based architectures: one that acts on the pose predictions in consecutive images, and one that models temporal information directly at the feature level. Our experiments evidence the additional benefits of modeling this temporal information over our single-frame approach.

In short, our contribution is to show that combining traditional CNNs for supervised learning with autoencoders for structured learning preserves the power of CNNs while also accounting for dependencies, resulting in increased performance. In the remainder of the chapter, we first briefly discuss earlier approaches. We then present our structured prediction framework in more detail, introduce our LSTM-based architectures and finally demonstrate that our approach achieves competitive performance with the earlier methods on standard 3D human pose estimation benchmarks.

Previous work. Following recent trends in Computer Vision, human pose estimation is now usually formulated within a Deep Learning framework. The switch away from earlier representations started with 2D pose estimation by learning a regressor from an input image either directly to pose vectors [274] or to heatmaps encoding 2D joint locations [106, 214, 273]. This has been exploited very effectively to infer 3D poses by fitting a 3D model to the 2D predictions [27, 60, 310, 328]. These approaches involve a separate, typically expensive model-fitting stage, outside of the Deep Learning framework.

In parallel, there has been a trend towards performing direct 3D pose estimation [102, 156], formulated as a regression problem. In other words, the algorithms output continuous 3D joint locations, because discretizing the 3D space is more challenging than the 2D one.

Our work fits in that line research, which involves dealing with the ambiguities inherent to inferring a 3D pose from a 2D input. To resolve them, recent algorithms have sought to encode the dependencies between the different joints within Deep Learning approaches, thus effectively achieving structured prediction. In particular, [96] uses autoencoders to learn a shared representation for 2D silhouettes and 3D poses. This approach, however, relies on accurate foreground masks and exploits handcrafted features, which mitigates the benefits of Deep Learning. In the context of hand pose estimation, [199] introduces a bottleneck, low dimensional layer that aims at accounting for joint dependencies. This layer, however, is obtained directly via PCA, which limits the range of dependencies it can model.

The work of [160] constitutes an effective approach to encoding dependencies within a Deep Learning framework for 3D human pose estimation. This approach extends the structured SVM model to the Deep Learning setting by learning a similarity score between feature embeddings of the input image and the 3D pose. This process, however, comes at a high computational cost at test time, since, given an input image, the algorithm needs to search for the highest-scoring pose. Furthermore, the final results are obtained by averaging over multiple high-scoring ground truth training poses, which might not generalize well to unseen data since the prediction can thus only be in the convex hull of the ground truth training poses.

To achieve a similar result effectively, we drew our inspiration from earlier KDE-based approaches [101, 102], which map both image and 3D pose to high-dimensional Hilbert spaces and learn a mapping between these spaces. In this chapter, we show how to do this in a Deep Learning context by combining CNNs and autoencoders. Not only does this allow us to leverage the power of learned features, which have proven more effective than hand-designed ones such as HOG [2] and 3D-HOG [297], but it yields a direct and efficient regression between the two spaces. Furthermore, it also allows us to learn the mapping from high-dimensional space to pose space, thus avoiding the need of KDE-based methods to solve an optimization problem at test time.

Using autoencoders for unsupervised feature learning has proven effective in several recognition tasks [137, 128, 279]. In particular, denoising autoencoders [278] that aim at reconstructing

the perfect data from a corrupted version of it have demonstrated good generalization ability. Similarly, contractive autoencoders have been shown to produce intermediate representations that are robust to small variations of the input data [228]. All these methods, however, rely on autoencoders to learn features for recognition tasks. By contrast, here, we exploit them to model the output structure for regression purposes.

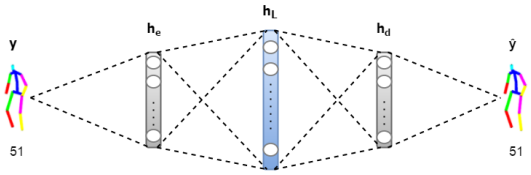
In this chapter, we further investigate the use of Recurrent Neural Networks (RNNs), and in particular LSTMs, to model temporal information. RNNs have recently been used in many Natural Language Processing [136, 260] and Computer Vision [163, 215] tasks, and, at the intersection of these fields, for image captioning and video description [56, 114]. More closely related to our work, in [65, 107], RNNs have been employed to model human dynamics. Nevertheless, these methods do not tackle human pose estimation, but motion capture generation, video pose labeling and forecasting for [65], and human-object interaction prediction for [107]. To the best of our knowledge, prior to our work, [161] is the only method that exploits RNNs for 3D human pose estimation from images. However, this approach operates on single images and makes use of RNNs to iteratively refine the pose predictions of [160]. By contrast we leverage the power of RNNs at modeling long term temporal dependencies across image sequences.

5.1 Approach

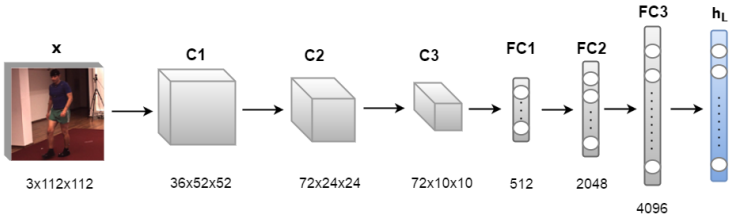
In this work, we aim at directly regressing from an input image or heatmap x to a 3D human pose. As in [26, 102, 156], we represent the human pose in terms of the 3D locations $y \in \mathbb{R}^{3J}$ of J body joints relative to a root joint. An alternative would have been to predict the joint angles and limb lengths. However, this is a less homogeneous representation and is therefore rarely used for regression purposes.

As discussed above, a straightforward approach to creating a regressor is to train a conventional CNN such as the one used in [156]. However, this fails to encode dependencies between joint locations. In [160], this limitation was overcome by introducing a substantially more complex, deep architecture for maximum-margin structured learning. Here, we encode dependencies in a simpler, more efficient, and, as evidenced by our experiments, more accurate way by learning a mapping between the output of a CNN and a latent representation obtained using an overcomplete autoencoder, as illustrated in Fig. 5.2. The autoencoder is pre-trained on human poses and comprises a hidden layer of *higher dimension* than its input and output. In effect, this hidden layer and the CNN-based representation of the image play the same role as the kernel embeddings in KDE-based approaches [47, 101, 102], thus allowing us to account for structure within a direct regression framework. Once the mapping between these two high-dimensional embeddings is learned, we further fine-tune the whole network for the final pose estimation task, as depicted at the bottom of Fig. 5.2.

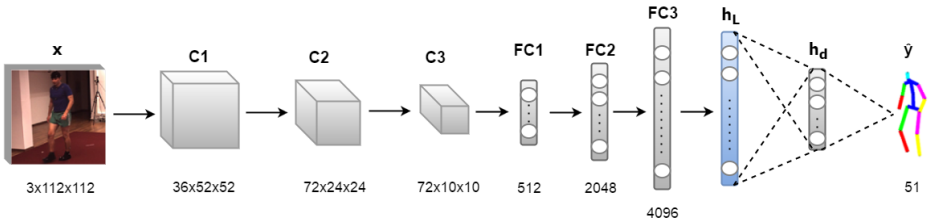
In the remainder of this section, we describe the different stages of our single-frame approach. We then extend this framework to modeling temporal consistency in Section 5.2.



(a) Autoencoder training



(b) Regression in latent space



(c) Fine-tuning

Figure 5.2 – **Our architecture for the structured prediction of the 3D human pose.** (a) We train a stacked denoising autoencoder that learns the structural information and enforces implicit constraints about human body in its latent middle layer h_L. (b) Our CNN architecture maps the raw image or the 2D joint location heatmap predicted from the input image to the latent representation h_L learned by the autoencoder. (c) We stack the decoding layers of the autoencoder on top of the CNN for reprojection from the latent space to the original pose space and fine-tune the entire network by updating the parameters of all layers.

5.1.1 Structured Latent Representations via Autoencoders

We encode the dependencies between human joints by learning a mapping of 3D human pose to a high-dimensional latent space. To this end, we use a denoising autoencoder that can have one or more hidden layers.

Following standard practice [279], given a training set of pose vectors $\{y_i\}$, we add isotropic Gaussian noise to create noisy versions $\{\tilde{y}_i\}$ of these vectors. We then train our autoencoder to take as input a noisy $\tilde{y}_i$ and return a denoised y_i. The behavior of the autoencoder is controlled by the set $\theta_{ae} = (W_{enc,j}, b_{enc,j}, W_{dec,j}, b_{dec,j})_{j=1}^{L}$ of weights and biases for L encoding and decoding layers.

We take the middle layer to be our latent pose representation and denote it by $h_L = g(\tilde{y}, \theta_{ae})$, where $g(\cdot)$ represents the encoding function. For example, with a single layer, the latent representation can be expressed as

$$h_L = g(\tilde{y}, W_{enc}, b_{enc}) = r(W_{enc}\tilde{y} + b_{enc}) \,, \tag{5.1}$$

where $r(\cdot)$ is the activation function. In practice, we use ReLU as the activation function of the encoding layers. This favors a sparse hidden representation [72], which has been shown to be effective at modeling a wide range of human poses [4, 221]. For the decoding part of the autoencoder, we use a linear activation function to be able to predict both negative and positive joint coordinates. To keep the number of parameters small and reduce overfitting, we use tied weights for the encoder and the decoder, that is, $W_{dec,j} = W_{enc,j}^T$.

To learn the parameters θ_{ae}, we rely on the square loss between the reconstruction, $\hat{y}$, and the true, noise-free pose, y, over the N training examples. To increase robustness to small pose changes, we regularize the cost function by add-ing the squared Frobenius norm of the Jacobian of the hidden mapping $g(\cdot)$, that is, $J(\tilde{y}) = \frac{\partial g}{\partial \tilde{y}}(\tilde{y})$. Training can thus be expressed as finding

$$\theta_{ae}^* = \underset{\theta_{ae}}{\operatorname{argmin}} \sum_{i=1}^{N} ||y_i - f(\tilde{y}_i, \theta_{ae})||_2^2 + \lambda \|J(\tilde{y}_i)\|_F^2 , \tag{5.2}$$

where $f(\cdot)$ represents the complete autoencoder function, and λ is the regularization weight. Unlike when using KDE, we do not need to solve a complex problem to go from the latent pose representation to the pose itself. This mapping, which corresponds to the decoding part of our autoencoder, is learned directly from data.

5.1.2 Regression in Latent Space

Once the autoencoder is trained, we aim to learn a mapping from the input image or heatmap to the latent representation of the human pose. To this end, and as shown in Fig. 5.2(b), we use a CNN to regress the image to a high-dimensional representation, which is itself mapped to the latent pose representation.

More specifically, let θ_{cnn} be the parameters of the CNN, including the mapping to the latent pose representation. Given an input image or heatmap x, we consider the square loss between the representation predicted by the CNN, $f_{cnn}(x, \theta_{cnn})$, and the one that was previously learned by the autoencoder, h_L. Given our N training samples, learning amounts to finding

$$\theta_{cnn}^* = \underset{\theta_{cnn}}{\operatorname{argmin}} \sum_{i=1}^{N} ||f_{cnn}(x_i, \theta_{cnn}) - h_{L,i}||_2^2 . \tag{5.3}$$

In practice, we either rely on a standard CNN architecture shown in Fig. 5.2(b), similar to the one of [156, 274] or a very deep network architecture, e.g. ResNet-50 [89]. In our implementation, the input volume is a three channel image of size 128×128 or a 16 channel heatmap of size 128×128. The last fully-connected layer of the base network is mapped linearly to the latent pose embedding. Except for this last linear layer, each layer uses a ReLU activation function. When we use images as input, we initialize the convolutional layers of our CNN from those of a network trained for the detection of body joints in 2D as in [156, 188].

In the case of 3D pose prediction from 2D joint location heatmaps, we rely on the stacked hourglass network design [194], which assigns high confidence values to most likely joint positions in the image. In practice, we have observed a huge performance improvement in overall 3D pose estimation accuracy when using reliable 2D joint location heatmaps produced by stacked hourglass networks compared to directly using RGB images as input to our standard CNN architecture in Fig. 5.2(b).

5.1.3 Fine-Tuning the Whole Network

Finally, as shown in Fig. 5.2(c), we append the decoding layers of the autoencoder to the CNN discussed above, which maps the latent pose estimates to the original pose space. We then fine-tune the resulting complete network for the task of human pose estimation. We take the cost function to be the squared difference between the predicted and ground truth 3D poses, which yields the optimization problem

$$\theta^*_{ft} = \underset{\theta_{ft}}{\operatorname{argmin}} \sum_{i}^{N} ||f_{ft}(x_i, \theta_{ft}) - y_i||_2^2 \,, \tag{5.4}$$

where θ_{ft} are the model parameters, including θ_{cnn} and the decoding weights and biases $(W_{dec,j}, b_{dec,j})_{j=1}^{L}$, and f_{ft} is the mapping function.

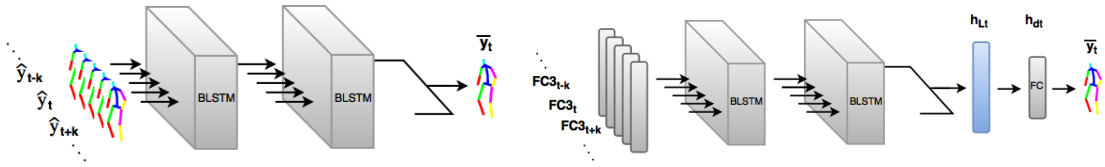
At test time, a new input image or heatmap is then simply passed forward through this fine-tuned network, which predicts the 3D pose via the learned latent representation.

5.2 Modeling Temporal Consistency

We have so far focused on predicting 3D poses from single images or heatmaps. However, it is well known that accounting for temporal consistency increases robustness. In this section, we show that our approach naturally allows us to use Long Short-Term Memory Units (LSTMs) to this end. Below, we first briefly review LSTMs and then introduce two different ways to exploit them to encode temporal information in our framework.

5.2.1 LSTMs

Recurrent Neural Networks (RNNs) have become increasingly popular to model temporal dynamics. In their simplest form, they map a sequence of inputs to a sequence of hidden states, each connected to its temporal neighbors, which are in turn mapped to a sequence of outputs. In theory, simple memory units and backpropagation through time (BPTT) allow RNNs to capture the temporal correlations between distant data points. However, in practice, longer sequences often cause the gradients to either vanish or explode, thus making optimization impossible. LSTMs [94] were introduced as a solution to this problem. Although they have four times as many parameters



(a) (b)

Figure 5.3 – **Our (B)LSTM networks to enforce temporal consistency.** (a) The (B)LSTM-Pose approach involves refining 3D human pose predictions by feeding those obtained as described in Fig. 5.2(c) into a (B)LSTM network, which yields the final 3D poses. (b) The (B)LSTM-Feature approach maps the features obtained from the last fully-connected layer of a CNN trained to directly regress 3D pose from monocular images to the latent representation h_L of Fig. 5.2(a) via a (B)LSTM network. The final pose is recovered by the decoder part of the autoencoder.

as traditional RNNs, they can be trained efficiently thanks to their sharing of parameters across time slices. An LSTM unit is defined by the recurrence equations

$$\begin{aligned} i_t &= \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ h_t &= o_t \odot \sigma_h(c_t)\,, \end{aligned} \tag{5.5}$$

where x_t, c_t and h_t are the input, hidden/cell state and output at time t, respectively, and i_t, f_t and o_t represent gate vectors to forget/select information. $\sigma_.(\cdot)$ are sigmoids and $\odot$ denotes the Hadamard or element-wise product.

In practice, we use either LSTMs or Bidirectional LSTMs (BLSTMs). A BLSTM comprises two LSTMs with information traveling in opposite temporal directions [76]. They have been shown to boost performance when the quantity to be predicted depends on contextual information coming from both forward and backward in time [76]. This is typically the case for human pose estimation, where the estimate at time t is correlated to those at time $t-1$ and $t+1$.

5.2.2 Recurrent Pose Estimation

We tested two different ways to incorporate (B)LSTMs into our framework.

Constraining the Final Poses

The first is to refine the pose estimates by imposing temporal consistency on the output of the network introduced in the previous section, as shown in Fig. 5.3(a).

More specifically, let $S_t = [\hat{y}_{t-\frac{T}{2}+1}, \ldots, \hat{y}_t, \ldots, \hat{y}_{t+\frac{T}{2}}]$ be the input sequence of T predicted poses

centered at time t. The network prediction can be expressed as

$$\bar{y}_t = f_p(S_t, \theta_p)\,, \tag{5.6}$$

where θ_p includes all the parameters of the network. During training, these parameters are taken to be

$$\theta_p^* = \underset{\theta_p}{\operatorname{argmin}} \sum_{t=T/2}^{N-T/2} ||f_p(S_t, \theta_p) - y_t||_2^2\,. \tag{5.7}$$

We refer to this method as *(B)LSTM-Pose*.

Constraining the Features

An alternative would be to enforce temporal consistency not on the poses, but earlier in the network on the features extracted from a direct CNN regressor. To this end, we made use of the features of the penultimate layer of our base network. This, for example, corresponds to FC3 features for the network shown in Fig. 5.2(b). These features act as input to the model depicted in Fig. 5.3(b), which stacks two BLSTM layers and maps the features to the latent representation learned by the autoencoder of Section 5.1.1. This is followed by the decoder to finally predict 3D poses.

Let $F_t = [\mathrm{FC}_{t-T/2+1}, \ldots, \mathrm{FC}_t, \ldots, \mathrm{FC}_{t+T/2}]$ be the sequence of such features. Then, training this network can be achieved by solving the problem

$$\theta_f^* = \underset{\theta_f}{\operatorname{argmin}} \sum_{t=T/2}^{N-T/2} ||f_f(F_t, \theta_f) - y_t||_2^2\,, \tag{5.8}$$

where $f_f(F_t, \theta_f)$ represents the complete network mapping, with parameters θ_f. We refer to this method as *(B)LSTM-Feature*.

5.3 Experiments

In this section, we first describe the datasets we tested our approach on. We then give implementation details and describe the evaluation protocol. Finally, we compare our results against those of the previous methods.

5.3.1 Datasets

We evaluate our method on the Human3.6m [102], HumanEva [250], KTH Multiview Football II [28] and Leeds Sports Pose (LSP) [115] datasets.

Human3.6m comprises 3.6 million image frames with their corresponding 2D and 3D poses. The

subjects perform complex motion scenarios based on typical human activities such as discussion, eating, greeting and walking. The videos were captured from 4 different camera viewpoints. Following the standard procedure of [156], we collect the input images by extracting a square region around the subject using the bounding box present in the dataset and the output pose is a vector of 17 3D joint coordinates.

HumanEva-I comprises synchronized images and motion capture data and is a standard benchmark for 3D human pose estimation. The output pose is a vector of 15 3D joint coordinates.

KTH Multiview Football II is a recent benchmark to evaluate the performance of pose estimation algorithms in unconstrained outdoor settings. The camera follows a soccer player moving around the field. The videos are captured from 3 different camera viewpoints and the output pose is a vector of 14 3D joint coordinates.

LSP is a standard benchmark for 2D human pose estimation and does not contain any ground truth 3D pose data. The images are captured in unconstrained outdoor settings. 2D pose is represented in terms of a vector of 14 joint coordinates. We report qualitative 3D pose estimation results on this dataset.

5.3.2 Implementation Details

We trained our autoencoder using a greedy layer-wise training scheme followed by fine-tuning as in [93, 279]. We set the regularization weight of Eq. 5.2 to $\lambda = 0.1$. We experimented with single-layer autoencoders, as well as with 2-layer ones. The size of the layers were set to 2000 and 300-300 for the 1-layer and 2-layer cases, respectively. We corrupted the input pose with zero-mean Gaussian noise with standard deviation of 40 for 1-layer and 40-20 for 2-layer autoencoders. In all cases, we used the ADAM optimization procedure [127] with a learning rate of 0.001 and a batch size of 128.

The number and individual sizes of the layers of our base architecture are given in Fig. 5.2. The filter sizes for the convolutional layers are consecutively 9×9, 5×5 and 5×5. Each convolutional layer is followed by a 2×2 max-pooling layer. The activation function is the ReLU in all the layers except for the last one that uses linear activation. As for the autoencoders, we used ADAM [127] with a learning rate of 0.001 and a batch size of 128. To prevent overfitting, we applied dropout with a probability of 0.5 after each fully-connected layer and augmented the data by randomly cropping 112×112 patches from the 128×128 image. When using 2D heatmaps as input, the 64×64 outputs of stacked hourglass network of [194] were upscaled to 128×128 before processing.

To demonstrate that our approach scales to very deep architectures, we also use ResNet-50 [89] as baseline CNN architecture. More specifically, we use it up to level 5, with the first three levels initialized on a 2D pose estimation task as in [188] and then kept constant throughout the 3D pose prediction process. We then use two additional convolutional layers of size 512 and 128 and

a linear layer to regress the 3D pose from the convolutional features of level 4.

To train *Ours-LSTM-Feature* and *Ours-BLSTM-Feature*, we relied on the features extracted from the penultimate layer of a CNN trained to directly predict 3D pose, referred to later as *CNN-Direct*. We did not backpropagate the loss of our LSTM-based models through this network, but rather kept its weights fixed. By contrast, *Ours-LSTM-Pose* and *Ours-BLSTM-Pose* take as input the 3D pose predictions obtained using the network in Fig. 5.2(c). In all cases, we cascaded two (B)LSTM layers of size 512, whose output sequence was merged into a single fully-connected layer of size 51. The activation function was $tanh$ for the recurrent layers and linear for the fully-connected layer at the end. In all architectures, we used a temporal window of length $T = 5$ with a stride of 5 covering 0.5 seconds for 50 fps Human3.6m videos. The first $T/2-1$ and the last $T/2$ frames were excluded from the evaluation. We optimized the recurrent networks using the ADAM optimization procedure [127] with a learning rate of 0.001 and a batch size of 128.

5.3.3 Evaluation Protocol

On Human3.6m, for the comparison to be fair, we used the same data partition protocol as in earlier work [156, 160] to obtain the training and test splits. The data from 5 subjects (S1,S5,S6,S7,S8) was used for training and the data from 2 different subjects (S9,S11) was used for testing. We trained a single model for all actions. We evaluate the accuracy of 3D human pose estimation in terms of average Euclidean distance between the predicted and ground truth 3D joint positions as in [156, 160]. To compare against [27, 240], we further evaluate the pose estimation accuracy after Procrustes transformation. The accuracy numbers are reported in milimeters for all actions. Training and testing were carried out monocularly in all camera views for each separate action.

On HumanEva-I, we trained our model on the Walking sequences of subjects S1, S2 and S3 as in [251, 328] and evaluate on the validation sequences of all subjects. We pretrained our network on the Walking sequences of Human3.6m and used only the first camera view for further training and validation.

On KTH Multiview Football II, we trained our model on the first half of the sequence containing Player 2 and test on the second half, as in [28]. We report accuracy using the percentage of correctly estimated parts (PCP) score with a threshold of 0.5 for a fair comparison. Since the training set is quite small, we pretrained our CNN model on the synthetic dataset introduced in [38], which contains images of sports players with their corresponding 3D poses.

On LSP, in order to generalize to the unconstrained outdoor settings, we trained our regressor on the recently released synthetic dataset of [38] and tested on the actual data from the LSP dataset.

Method	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting	Sitting Down
Ionescu et al. [102]	132.71	183.55	132.37	164.39	162.12	150.61	171.31	151.57	243.03
Li & Chan [156]	-	148.79	104.01	127.17	-	-	-	-	-
Li et al. [160]	-	134.13	97.37	122.33	-	-	-	-	-
Li et al. [161]	-	133.51	97.60	120.41	-	-	-	-	-
Zhou et al. [328]	-	-	-	-	-	-	-	-	-
Rogez & Schmid [230]	-	-	-	-	-	-	-	-	-
Tekin et al. [264]	-	129.06	91.43	121.68	-	-	-	-	-
Park et al. [202]	100.34	116.19	89.96	116.49	115.34	117.57	106.94	137.21	190.82
Zhou et al. [327]	91.83	102.41	96.95	98.75	113.35	90.04	93.84	132.16	158.97
Tome et al. [271]	64.98	73.47	76.82	86.43	86.28	68.93	74.79	110.19	173.91
Pavlakos et al. [206]	67.38	71.95	66.70	69.07	71.95	65.03	68.30	83.66	**96.51**
OURS (ShallowNet-Autoencoder)	94.98	129.06	91.43	121.68	133.54	115.13	133.76	140.78	214.52
OURS (ShallowNet-Hm-Autoencoder)	69.64	93.79	69.02	96.47	103.42	83.36	85.22	116.62	147.57
OURS (ResNet-Autoencoder)	**57.84**	**64.62**	**59.41**	**62.83**	**71.52**	**57.50**	**60.38**	**80.22**	104.14

Method:	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Avg. (6 Actions)	Avg. (All)
Ionescu et al. [102]	162.14	205.94	170.69	96.60	177.13	127.88	159.99	162.14
Li & Chan [156]	-	189.08	-	77.60	146.59	-	132.20	-
Li et al. [160]	-	166.15	-	68.51	132.51	-	120.17	-
Li et al. [161]	-	163.33	-	73.66	135.15	-	121.55	-
Zhou et al. [328]	-	-	-	-	-	-	-	120.99
Rogez & Schmid [230]	-	-	-	-	-	-	-	121.20
Tekin et al. [264]	-	162.17	-	65.75	130.53	-	116.77	-
Park et al. [202]	105.78	149.55	125.12	62.64	131.90	96.18	111.12	117.34
Zhou et al. [327]	106.91	125.22	94.41	79.02	126.04	98.96	104.73	107.26
Tome et al. [271]	84.95	110.67	85.78	71.36	86.26	73.14	84.17	88.39
Pavlakos et al. [206]	71.74	**76.97**	65.83	59.11	74.89	63.24	69.78	71.90
OURS (ShallowNet-Autoencoder)	121.26	162.17	138.2	65.75	130.53	113.34	116.77	127.07
OURS (ShallowNet-Hm-Autoencoder)	87.17	120.50	95.31	55.87	85.69	64.66	86.89	91.62
OURS (ResNet-Autoencoder)	**66.31**	80.50	**61.20**	**52.55**	**69.97**	**60.08**	**61.20**	**67.27**

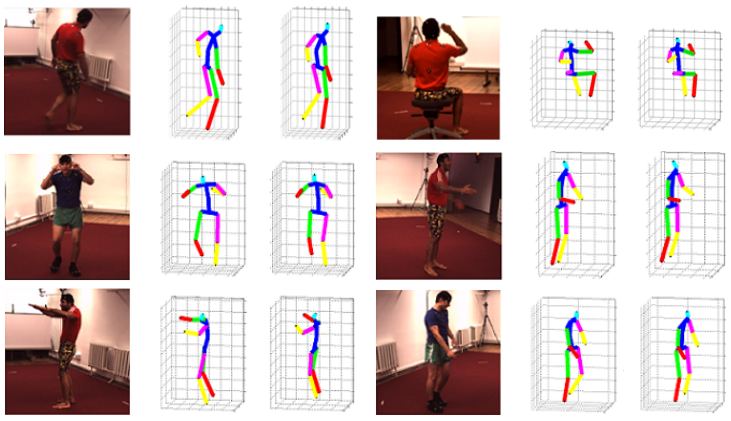
Table 5.1 – **Comparison of our structured prediction approach with earlier work on Human3.6m.** We report 3D joint position errors in mm, computed as the average Euclidean distance between the ground truth and predicted joint positions. '-' indicates that the results were not reported for the respective action class in the original paper. Note that our method achieves the best overall accuracy.

5.3.4 Evaluation

We first discuss our results on predicting 3D pose from a single image, and then turn to the case where we use multiple consecutive frames as input.

Human Pose from a Single Image

Fig. 5.4 depicts selected pose estimation results on Human3.6m. In Table 5.1, we report our single-image autoencoder-based results on this dataset along with those of the following single image-based methods: KDE regression from HOG features to 3D poses [102], jointly training a 2D body part detector and a 3D pose regressor [156, 202], the maximum-margin structured learning framework of [160, 161], the deep structured prediction approach of [264], pose regression with kinematic constraints [327], pose estimation with mocap guided data augmentation [230], volumetric pose prediction approach of [206] and lifting 2D heatmap predictions to 3D human pose [271]. *ShallowNet-Autoencoder* refers to our autoencoder-based regression approach using



(a) Image (b) Prediction (c) GT (d) Image (e) Prediction (f) GT

Figure 5.4 – **Pose estimation results on Human3.6m.** (a,d) Input image. (b,e) Recovered pose. (c,f) Ground truth. Examples are from the *Walking, Eating, Taking Photo, Greeting, Discussion* and *Walking Dog* actions of the Human3.6m database. In each scenario, our structured prediction approach can reliably recover the 3D pose of the subject. Best viewed in color.

Model	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting
Bogo et al.[27]	62.0	60.2	67.8	76.5	92.1	73.0	75.3	100.3
Sanzari et al.[240]	48.82	56.31	95.98	84.78	96.47	66.30	107.41	116.89
OURS (ResNet-Autoencoder)	**43.89**	**48.54**	**46.57**	**49.95**	**53.94**	**43.77**	**43.94**	**60.20**

Model	Sitting Down	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Average
Bogo et al.[27]	137.3	83.4	77.0	77.3	86.8	79.7	81.7	82.3
Sanzari et al.[240]	129.63	97.84	105.58	65.94	92.58	130.46	102.21	93.15
OURS (ResNet-Autoencoder)	**73.64**	**51.15**	**59.29**	**46.30**	**39.81**	**52.25**	**47.18**	**50.69**

Table 5.2 – **Comparison of our structured prediction approach with earlier work after Procrustes transformation on Human3.6m.** We report average Euclidean distance (in mm) between the ground truth 3D joint locations and those predicted by competing methods [58, 27, 240] as wll as ours after Procrustes transformation.

the base architecture depicted in Fig. 5.2, and *ResNet-Autoencoder* to the one using the ResNet-50 architecture. For the shallow network architecture, we also evaluate the pose estimation accuracy using the 2D joint location heatmaps of [194] as input. This is referred to as *ShallowNet-Hm-Autoencoder*.

The shallow network architecture provides satisfactory pose estimation accuracy with a fast computational runtime of 6 ms/frame, which corresponds to 166 fps real-time performance, whereas *ResNet-Autoencoder* comes at the cost of a three times slower runtime. Our autoencoder-based regression approach using ResNet-50 as base network outperforms all the baselines.

In [27], the reconstruction error was evaluated by first aligning the estimated skeleton to the ground truth one by Procrustes transformation, and we confirmed through personal communication that the same protocol was used in [240]. To compare our results to those of the earlier work, we therefore also report in Table 5.2 the joint error after Procrustes transformation. Altogether,

Model	Discussion	Eating	Greeting	Taking Photo	Walking	Walking Dog
CNN-Direct	135.36	105.98	133.35	177.62	77.73	153.02
OURS-Autoencoder, 1 layer no FT	134.02	96.01	127.58	158.73	68.55	146.28
OURS-Autoencoder, 2 layer no FT	129.67	98.57	124.80	162.69	73.47	146.46
OURS-Autoencoder, 1 layer with FT	130.07	94.08	121.96	**158.51**	65.83	135.35
OURS-Autoencoder, 2 layer with FT	**129.06**	**91.43**	**121.68**	162.17	**65.75**	**130.53**

(a)

Model	Joint error
CNN-Direct	177.62
CNN-ExtraFC[2000]	179.29
CNN-PCA[30]	170.74
CNN-PCA[40]	167.62
CNN-PCA[51]	182.64
OURS-Autoencoder[40]	165.11
OURS-Autoencoder[2000]	**158.51**

(b)

Table 5.3 – **Ablation studies for our structured prediction approach.** We report the average Euclidean distance (in mm) between the groundtruth 3D joint locations and those computed (a) using either no autoencoder at all (CNN) or 1-layer and 2-layer encoders (OURS-Autoencoder), with or without fine-tuning (FT), (b) by replacing the autoencoder by either an additional fully-connected layer (*CNN-ExtraFC*) or a PCA layer (*CNN-PCA*) on the sequences of *Taking Photo* action class. The bracketed numbers denote the various dimensions of the additional layer we tested. Our approach again yields the most accurate predictions.

by leveraging the power of deep neural networks and accounting for the dependencies between body parts, *ResNet-Autoencoder* significantly outperforms the previous methods.

We further evaluated our approach on the official test set of Human3.6m for two different actions. We obtained a pose reconstruction error of 64.38 and 63.86 mm for the *Directions* and *Discussion* actions, respectively. Our method currently ranks second in the leaderboard for these two actions. Note that the first ranking method [219] relies on the knowledge of body part segmentations whereas we do not use this additional piece of ground truth information.

To validate our design choices, we report in Table 5.3, the pose estimation accuracies obtained with various autoencoder configurations using the shallow network depicted in Fig. 5.2. The results reported in Tables 5.1 and 5.2 were obtained using a two layer autoencoder. However, as discussed in Section 5.1.1 our formalism applies to autoencoders of any depth. Therefore, in Table 5.3(a), we also report results obtained using a single layer one obtained by turning off the final fine-tuning of Section 5.1.3. For completeness, we also report results obtained by using a CNN similar to the one of Fig. 5.2(b) to regress directly to a 51-dimensional 3D pose vector *without* using an autoencoder at all. We will refer to it as *CNN-Direct*. We found that both kinds of autoencoders perform similarly and better than *CNN-Direct*, especially for actions such as *Taking Photo* and *Walking Dog* that involve interactions with the environment and are

Model	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting
ResNet	56.77	64.73	60.94	63.49	74.98	57.65	61.08	81.29
ResNet-Autoencoder w/o ExtraMocap	57.84	64.62	59.41	62.83	**71.52**	57.50	60.38	**80.22**
ResNet-Autoencoder w/ ExtraMoCap	**55.87**	**63.65**	**59.08**	**62.64**	72.08	**56.15**	**58.88**	80.53

Model	Sitting Down	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Average
ResNet	102.45	66.65	80.96	60.87	53.26	70.27	60.95	68.29
ResNet-Autoencoder w/o ExtraMoCap	104.14	66.31	80.50	61.20	52.55	69.97	60.08	67.27
ResNet-Autoencoder w/ ExtraMoCap	**102.30**	**65.68**	**78.25**	**59.05**	**51.81**	**68.44**	**58.19**	**66.17**

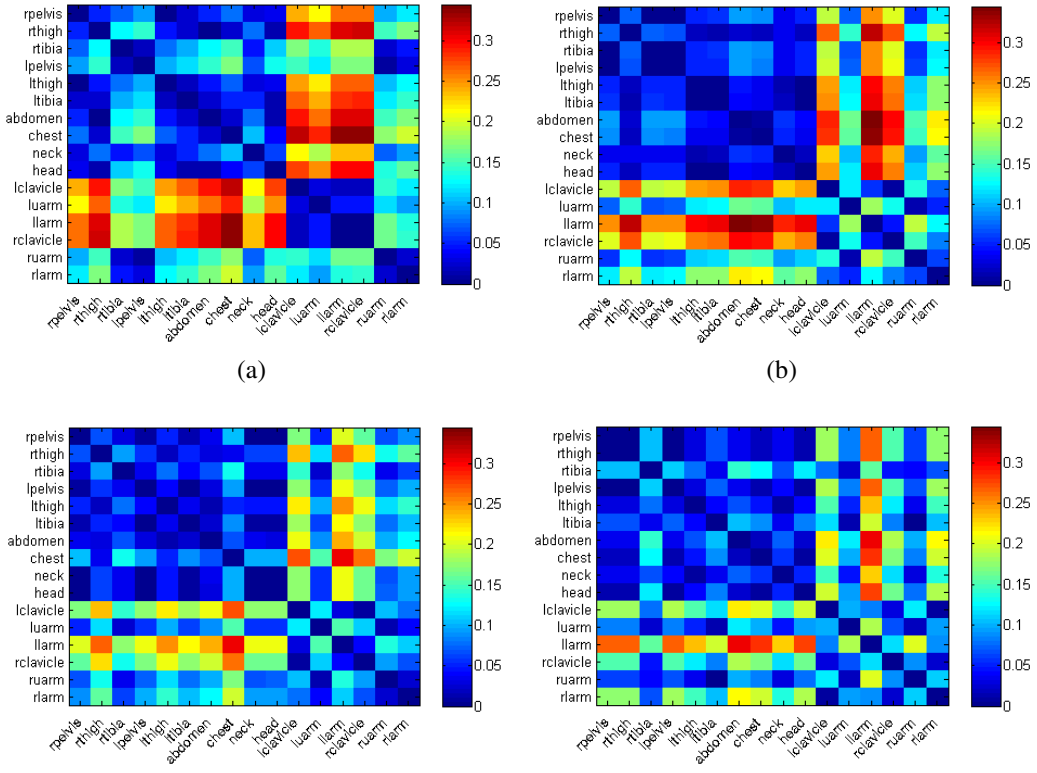
Table 5.4 – **Evaluation of our approach with deep network architectures.** We report the average Euclidean distance (in mm) between the groundtruth and predicted 3D joint locations. The predictions are obtained using a direct ResNet regressor, ResNet-Autoencoder trained with only motion capture data from Human3.6m and ResNet-Autoencoder trained with motion capture data from Human3.6m and MPI-INF-3DHP.

thus physically more constrained. This confirms that the power of our method comes from autoencoding. Furthermore, as expected, fine-tuning consistently improves the results.

During fine-tuning, our complete network has more fully-connected layers than *CNN-Direct*. One could therefore argue that the additional layers are the reason why our approach outperforms it. To disprove this, we evaluated the baseline, *CNN-ExtraFC*, in which we simply add one more fully-connected layer. We also evaluated another baseline, *CNN-PCA*, in which we replace our autoencoder latent representation by a PCA-based one. In Table 5.3(b), we show that our approach significantly outperforms these two baselines on the *Taking Photo* action. This suggests that our overcomplete autoencoder yields a representation that is more discriminative than other latent ones. Among the different PCA configurations, the one with 40 dimensions performs the best. However, training an autoencoder with 40 dimensions outperforms it.

To learn a more powerful latent pose space, we exploit additional motion capture data from the MPI-INF-3DHP dataset [188] for training the autoencoder. In Table 5.4, we report results with and without this additional data. We achieve better pose estimation accuracy when we train on a wider range of poses. As Human3.6m already includes a large variety of poses and the marker placements between the two datasets do not exactly match each other, we only observe a slight improvement. However, our results suggest that training an autoencoder on a larger pose space without any dataset bias would result in an even more representative latent pose space and, eventually, a higher pose estimation accuracy. We further compare our autoencoder-based regression approach to a direct regression baseline. The relative contribution of the autoencoder on very deep neural networks is smaller than that on a shallower network. However, we still increase the accuracy by applying our autoencoder training on top of the ResNet architecture.

Following [101], we show in Fig. 5.5 the differences between the ground truth limb ratios and the limb ratios obtained from predictions based on KDE, *CNN-Direct* and our autoencoder-based approach. These results demonstrate that our predictions better preserve these limb ratios, and thus better model the dependencies between joints.



(a) (b)

(c) (d)

Model	Lower Body	Upper Body	Full Body
KDE [102]	1.02	7.18	16.43
CNN	0.57	6.86	14.97
OURS-Autoencoder no FT	0.62	5.30	11.99
OURS-Autoencoder with FT	0.77	5.43	11.90

(e)

Figure 5.5 – **Analysis on structure preservation ability of our 3D pose estimation approach.** We visualize the matrix of differences between estimated log of limb length ratios and those computed from ground truth poses. The rows and columns correspond to individual limbs. For each cell, the ratios are computed by dividing the limb length in the horizontal axis by the one in the vertical axis as in [101] for **(a)** KDE [102], **(b)** CNN-Direct as in Table 5.3, and **(c,d)** our method without and with fine-tuning. An ideal result would be one in which all cells are blue, meaning the limb length ratios are perfectly preserved. Best viewed in color. **(e)** Sum of the log of limb length ratio errors for different parts of the human body. All methods perform well on the lower body. However, ours outperforms the others on the upper body and when considering all ratios in the full body.

In Fig. 5.6, we visualize the latent space learned by the autoencoder after embedding it in 2D using the t-SNE algorithm [180]. It can be seen that the upper left corner spans the downward-facing body poses, the diagonal includes mostly the upright body poses and the lower right corner

Figure 5.6 – **Visualization of the learned latent pose space.** t-SNE embedding [180] for the latent representation of the poses from the *Sitting Down* category in Human3.6m.

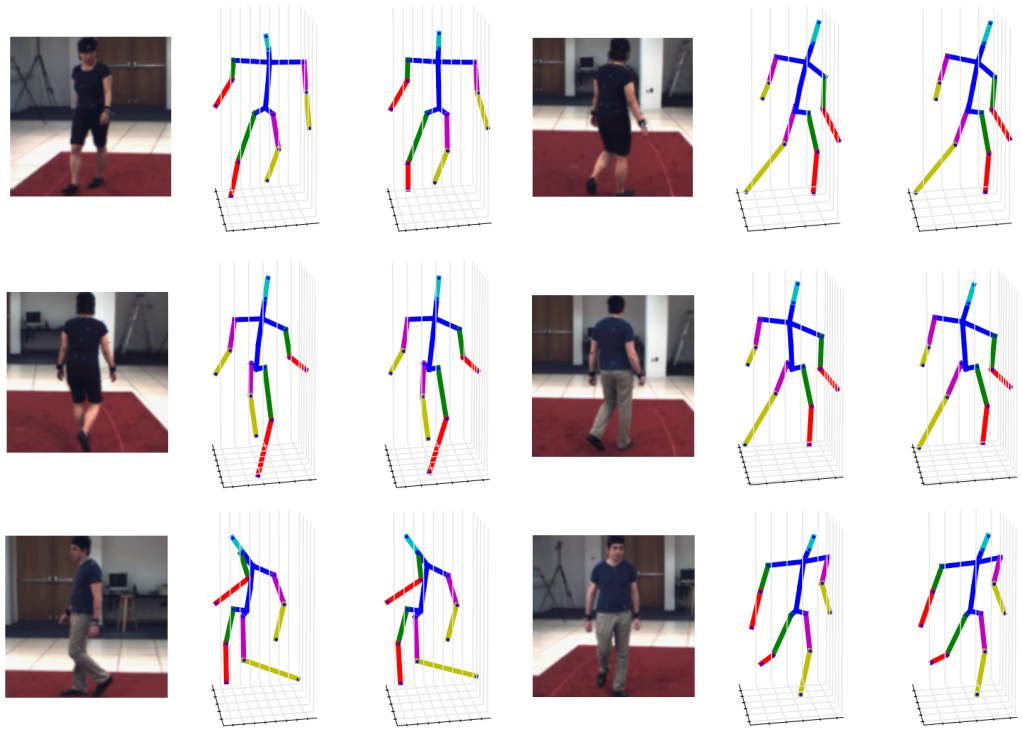
clusters the forward-facing body poses sitting on the ground. Note that our latent representation covers the entire low-dimensional space, thus making it well-suited to discriminate between poses with small variations.

We further report single-image 3D pose estimation accuracy on the HumanEva-I dataset in Table 5.5 and show qualitative pose estimation results in Fig. 5.7. We follow the protocol adopted in the previous methods to 3D inference from 2D body part detections [251] and to 3D model-fitting [27, 328]. Following these methods, we measure 3D pose error after aligning the prediction to the ground truth by a rigid transformation. Note that [328] uses video instead of a single frame for prediction. Our method outperforms the previous methods on this standard benchmark.

On the KTH Multiview Football II dataset, we compare our autoencoder-based approach against [28], which is the only monocular single-image 3D pose estimation method publishing

Method	S1	S2	S3	Average
Simo-Serra et al. [251]	65.1	48.6	73.5	62.4
Bogo et al. [27]	73.3	59.0	99.4	77.2
Zhou et al. [328]	34.2	30.9	**49.1**	38.07
OURS-Autoencoder	**29.32**	**17.94**	59.51	**35.59**

Table 5.5 – **Quantitative results of our approach on Walking sequences of the HumanEva-I dataset [250].** S1, S2 and S3 correspond to Subject 1, 2, and 3, respectively. The accuracy is reported in terms of average Euclidean distance (in mm) between the predicted and ground truth 3D joint positions.



(a) Image (b) Prediction (c) GT (d) Image (e) Prediction (f) GT

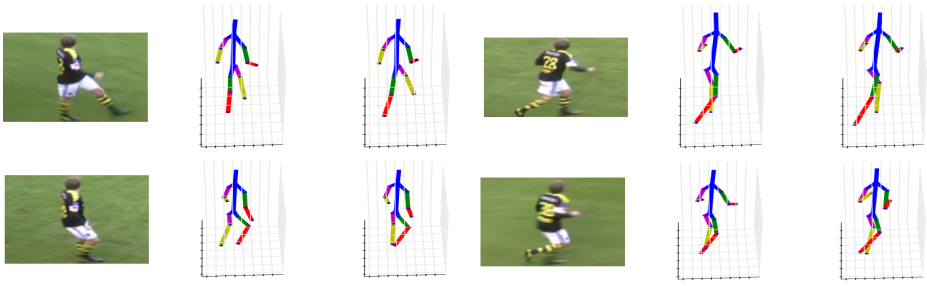
Figure 5.7 – **Pose estimation results on HumanEva-I.** (a,d) Input image. (b,e) Recovered pose. (c,f) Ground truth. Best viewed in color.

results on this dataset so far. As can be seen in Table 5.6, we outperform the PCP accuracy of this baseline significantly on all body parts except for the pelvis. Fig. 5.8 depicts example pose estimation results on this dataset.

In Fig. 5.9, we provide additional qualitative results on the LSP dataset, which features challenging poses. Our autoen-coder-based regression approach nevertheless delivers accurate 3D predictions.

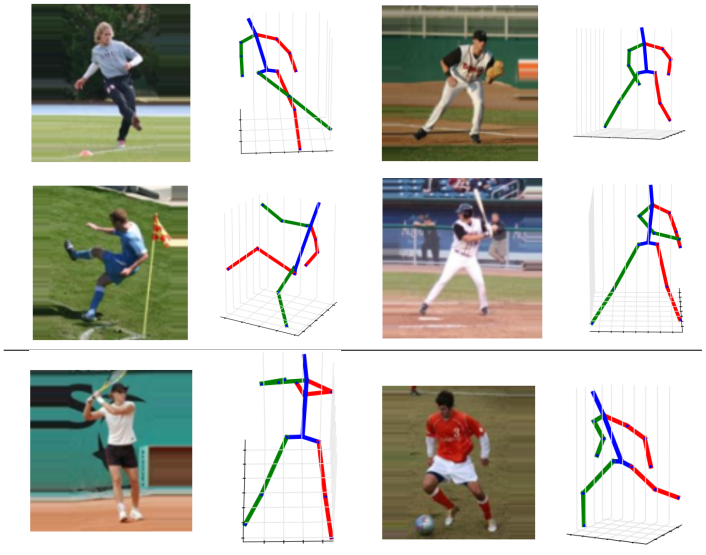
Method:	Pelvis	Torso	Upper Arms	Lower arms	Upper Legs	Lower Legs	All parts
[28]	**97**	87	14	6	63	41	43
OURS-Autoencoder	66	**100**	**66.5**	**16.5**	**83**	**66.5**	**63.1**

Table 5.6 – **Evaluation of our approach on KTH Multiview Football II.** On this dataset, we compare our method that uses a single image to that of [28]. We rely on the percentage of correctly estimated parts (PCP) score to evaluate the performance as in [28]. Higher PCP score corresponds to better 3D pose estimation accuracy.



(a) Image (b) Prediction (c) GT (d) Image (e) Prediction (f) GT

Figure 5.8 – **Pose estimation results on KTH Multiview Football II.** (a, d) Input images. (b, e) Recovered pose. (c, f) Ground truth. Best viewed in color.



(a) Image (b) Prediction (c) Image (d) Prediction

Figure 5.9 – **Pose estimation results on LSP.** (a,c) Input images. (b,d) Recovered pose. We trained our network on the recently released synthetic dataset of [38] and tested it on the LSP dataset. The quality of the 3D pose predictions demonstrates the generalization of our method. In the last row, we show failure cases in the 3D pose prediction of lower legs due to foreshortening (left) and orientation ambiguities (right)

.

Model	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting
OURS (ResNet-Autoencoder)	57.84	64.62	59.41	62.83	71.52	57.50	60.38	80.22
OURS-LSTM-Pose	55.63	64.55	57.56	62.20	70.71	56.52	57.37	78.93
OURS-BLSTM-Pose	54.93	63.26	57.26	62.30	70.28	56.66	57.08	78.98
OURS-LSTM-Feature	71.34	68.88	67.12	75.87	79.36	66.19	61.49	83.28
OURS-BLSTM-Feature	70.01	68.74	64.64	75.90	78.99	64.21	60.50	83.10

Model	Sitting Down	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Average
OURS (ResNet-Autoencoder)	104.14	66.31	80.50	61.20	52.55	69.97	60.08	67.27
OURS-LSTM-Pose	98.47	64.43	77.18	62.32	50.12	67.50	66.77	66.02
OURS-BLSTM-Pose	97.13	64.29	77.40	61.94	49.76	67.11	62.26	65.37
OURS-LSTM-Feature	97.66	71.51	83.93	78.67	63.69	73.23	69.03	74.08
OURS-BLSTM-Feature	96.44	70.29	83.51	77.83	62.02	71.11	69.55	73.52

Table 5.7 – **Analysis of our different (B)LSTM architectures.** We report the average Euclidean distance (in mm) between the ground truth 3D joint locations and the predictions obtained by our ResNet-Autoencoder approach evaluated using different LSTM architectures on the video data.

Model	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting
Du et al. [58]	85.07	112.68	104.90	122.05	139.08	105.93	166.16	117.49
Tekin et al. [266]	102.41	147.72	88.83	125.28	118.02	112.3	129.17	138.89
Zhou et al. [328]	87.36	109.31	87.05	103.16	116.18	106.88	99.78	124.52
OURS (ResNet-Autoencoder)	57.84	64.62	59.41	62.83	71.52	57.50	60.38	80.22
OURS-BLSTM-Pose	**54.93**	**63.26**	**57.26**	**62.30**	**70.28**	**56.66**	**57.08**	**78.98**

Model	Sitting Down	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Average
Du et al. [58]	226.04	120.02	135.91	117.65	99.26	137.36	106.54	126.47
Tekin et al. [266]	224.90	118.42	182.73	138.75	55.07	126.29	65.76	124.97
Zhou et al. [328]	199.23	107.42	143.32	118.09	79.39	114.23	97.70	113.01
OURS (ResNet-Autoencoder)	104.14	66.31	80.50	**61.20**	52.55	69.97	**60.08**	67.27
OURS-BLSTM-Pose	**97.13**	**64.29**	**77.40**	61.94	**49.76**	**67.11**	62.26	**65.37**

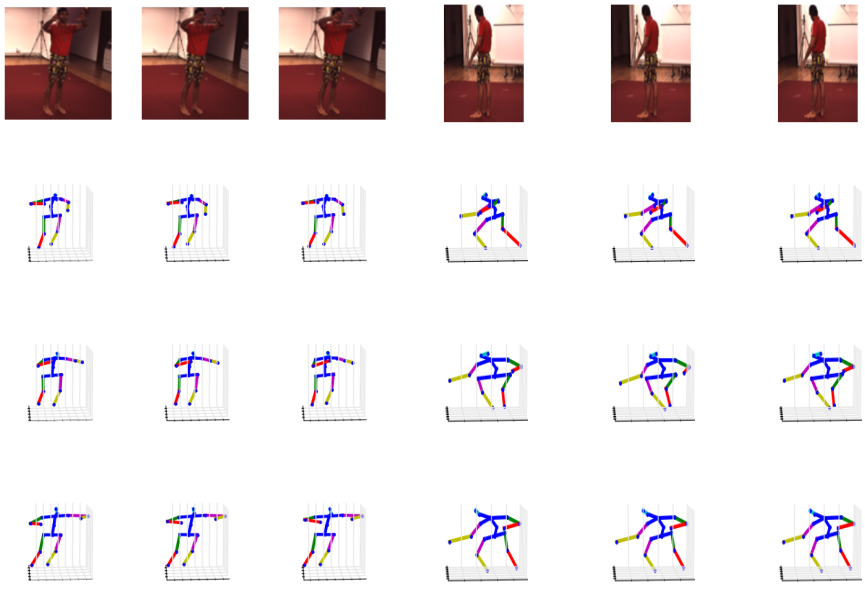
Table 5.8 – **Comparison of our (B)LSTM-based architectures to the earlier work.** We report the average Euclidean distance in mm between the ground truth 3D joint locations and the predictions obtained by our ResNet-Autoencoder approach with and without BLSTM regularization on the output poses, compared to [58, 266, 328]

.

Human Pose from Video

In Table 5.7, we demonstrate the effectiveness of imposing temporal consistency using LSTMs on Human3.6m, as described in Section 5.2. We compare our results with and without LSTMs against those of [58, 266, 328], which also rely on video sequences. On average, our LSTM-based approaches applied to the 3D pose predictions of *ResNet-Autoencoder* bring an improvement over single-image results, with the one of Section 5.2.2 that enforces temporal consistency at pose level being significantly better than the other. Using standard LSTMs instead of BLSTMs degrades the accuracy but eliminates the latency involved in working on image-batches, which can be a worthwhile trade-off if real-time performance is required.

As shown in Table 5.8, our LSTM units improves the pose estimation accuracy on average by approximately 3% and our ResNet-based results are significantly more accurate than the other methods, with an average pose estimation accuracy of 65.37 mm vs 124.97 mm for [266], 113.01



(a) $t-k$ (b) t (c) $t+k$ (d) $t-k$ (e) t (f) $t+k$

Figure 5.10 – **Pose estimation results with LSTMs on Human3.6m.** (a,d) $t-k^{th}$ frame. (b,e) t^{th} frame. (c,g) $t+k^{th}$ frame. k denotes the stride between consecutive frames. Top row: Input image, Second row: Our pose estimate from the single image, Third row: Our BLSTM pose estimate, Last row: Ground truth. Our BLSTM network can correct for the errors made by the autoencoder by accounting for the temporal consistency. Best viewed in color.

mm for [328] and 126.47 mm for [58]. Fig. 5.10 depicts example pose estimation results of our BLSTM approach compared to our autoencoder-based approach based on a single image.

We further compare our *OURS-BLSTM-Pose* model with a network where the BLSTM was replaced by two fully-connected layers, thus giving it a similar capacity as the BLSTM one, but not explicitly modeling temporal consistency. This model gives an average pose estimation accuracy on all Human3.6m actions of 77.96 mm, whereas our BLSTM-based model achieves 65.37 mm. Our method significantly outperforms this baseline, thus showing that the better performance of our LSTM-based networks does not just come from their larger number of parameters, but truly from their ability to model temporal information.

5.3.5 Comparison Between KDE and Autoencoders

In Table 5.9, we compare two structured 3D human pose estimation methods: Our autoencoder-based deep network approach and kernel dependency estimation (KDE) [101, 102]. In the earlier works of [101] and [102], KDE is applied to handcrafted HOG features, whereas in our approach we rely on deep features. In order to compare the structured regression performance of KDE to our autoencoder-based approach, we also applied KDE to the deep features extracted from a CNN.

Model	Directions	Discussion	Eating	Greeting	Phone Talk	Posing	Buying	Sitting
HOG + KDE [102]	132.71	183.55	132.37	164.39	162.12	150.61	171.31	151.57
Conv3 Feat. + KDE	99.13	160.84	112.10	137.32	137.97	118.16	137.13	153.79
FC3 Feat. + KDE	99.06	160.39	104.53	132.01	132.35	118.13	144.36	149.80
CNN-Direct	106.23	161.54	108.42	136.15	136.21	123.37	148.68	157.15
OURS-Autoencoder	**94.98**	**129.06**	**91.43**	**121.68**	**133.54**	**115.13**	**133.76**	**140.78**

Model	Sitting Down	Smoking	Taking Photo	Waiting	Walking	Walking Dog	Walking Pair	Average
HOG + KDE [102]	243.03	162.14	205.94	170.69	96.60	177.13	127.88	162.14
Conv3 Feat. + KDE	190.48	137.06	181.77	151.15	93.97	149.81	120.46	138.74
FC3 Feat. + KDE	206.35	133.91	169.31	150.76	86.44	144.83	113.20	136.36
CNN-Direct	217.88	136.59	169.42	157.71	88.75	149.58	115.02	140.85
OURS-Autoencoder	**214.52**	**121.26**	**162.17**	**138.2**	**65.75**	**130.53**	**113.34**	**127.07**

Table 5.9 – **Comparison of our structured prediction approach to KDE [102].** We report the average Euclidean distance (in mm) between the groundtruth 3D joint locations and those predicted by competing methods [102] as well as ours.

Layer Configuration	Greeting
[40]	129.49
[500]	123.95
[1000]	121.96
[2000]	**121.96**
[3000]	123.49
[250-250]	125.61
[300-300]	**121.68**
[250-500]	128.98
[500-1000]	126.52
[200-200-200]	126.78
[500-500-500]	127.73

Table 5.10 – **Analysis on the hyperparameter choices for our structured prediction approach.** We report average Euclidean distance (in mm) between the ground truth 3D joint locations and the ones predicted by our approach. We train our model using autoencoders with different number of layers and different number of channels per layer as indicated by the bracketed numbers. This validation was performed on the *Greeting* action and the optimal values were used for all other actions.

We extract either the features from the last convolutional layer (Conv3) or the last fully-connected layer (FC3) of the network depicted in Fig. 5.2(b). As can be seen in Table 5.9, we consistently outperform all the baselines, which demonstrates the power of autoencoding.

5.3.6 Parameter Choices

In Table 5.10, we compare the results of different autoencoder configurations in terms of number of layers and channels per layer on the *Greeting* action. Similarly to what we did in Table 5.3(b), the bracketed numbers denote the dimension of the autoencoder's hidden layers. We obtained the best result for 1 layer with 2000 channels or 2 layers with 300-300 channels. These values are

those we used for all the experiments described above. They were chosen for a single action and used unchanged for all others, thus demonstrating the versatility of our approach.

5.4 Conclusion

We have introduced a novel Deep Learning regression architecture for structured prediction of 3D human pose from a monocular image or a 2D joint location heatmap. We have shown that our approach to combining autoencoders with CNNs accounts for the dependencies between the human body parts efficiently and significantly improves accuracy. We have also shown that accounting for the temporal information with LSTMs further increases the accuracy of our pose estimates. Since our framework is generic, in future work, we intend to apply it to other structured prediction problems, such as deformable surface reconstruction.

6 Dyadic Human Motion Prediction

Forecasting future motion from observed past 3D poses has primarily been studied in a single-person setting [151, 183, 181, 145, 167]. A naive way to extend these approaches to the multi-person case is to simply treat each subject independently. However, this fails to account for interactions that condition future behavior. Only in [1] is there an attempt to capture them via the use of social cues obtained by pooling the learned features for each individual. While effective in the presence of weak social interactions, this approach is ill-suited to modeling the stronger dependencies that arise from two closely-interacting individuals whose movements are highly correlated.

In this paper, we therefore introduce an approach to dyadic, or pairwise, human motion prediction that more strongly models interactions. To this end, we develop an encoder-decoder architecture with both self- and pairwise attention modules. While self-attention captures the similarities between someone's present and past motions, pairwise attention models capture the dependencies between the pose histories of both subjects. Then, for each subject, the decoder takes as input the subject's own self-attention features and the pairwise attention ones, and outputs the future 3D pose sequence.

As there is no dyadic motion prediction benchmark with closely-interacting people, we build the Lindyhop600k dataset. It features Lindy Hop dancers performing energetic moves, ranging from frenzied kicks to smooth and sophisticated body motions. The dancers synchronize their fast-paced steps with one another and the music. The standard footwork can be followed by infrequent twirls, which make the upcoming pose prediction hard without observing the highly correlated moves of the partner. The motion of one person gives significant clues about infrequent or subtle motion patterns of the other that cannot be easily inferred from the isolated individual motion.

Hence, our contributions are twofold.

- We propose the first 3D motion prediction method that models the dyadic motion dependencies between two subjects.

- We introduce a new dance dataset, Lindyhop600k, which consists of videos and 3D human body poses of dancers performing diverse swing motions.

Our experiments on the Lindyhop600k dataset clearly demonstrate the benefits of our method. It outperforms both the state-of-the-art single person baselines and the use of weaker social cues [1]. Our results are especially promising in terms of long-term prediction. The proposed method models the motion dynamics much more reliably than the baselines.

6.1 Approach

Let us now introduce dyadic human motion prediction method for closely-interacting people. To this end, we first review the single person motion prediction formalism at the heart of our method, and then present our approach to modeling pairwise interactions to predict the future poses of two people.

6.1.1 Single Person Baseline

Our work builds on “History Repeats Itself (HRI)" [181], which relies on an attention mechanism and a GCN to predict the future poses of a single person based on their observed sequence of historical poses. Intuitively, the attention mechanism aims to focus the prediction on the most relevant parts of the motion history and the GCN decodes the resulting representation into the future pose predictions while encoding the dependencies across the different joints.

Formally, given a sequence of T_p past 3D poses of an individual, $\mathbf{X}_{1:T_p} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{T_p}]^T$, single-person human motion prediction aims to estimate the T_f future 3D poses $\mathbf{X}_{T_p+1:T_p+T_f} = [\mathbf{x}_{T_p+1}, \mathbf{x}_{T_p+2}, ..., \mathbf{x}_{T_p+T_f}]^T$. Each pose $\mathbf{x}_t \in \mathbb{R}^K$, where $K = J \times 3$, comprises J joints forming a skeleton. In HRI, the similarity between past motions and the last observed motion context is captured by dividing the motion history $\mathbf{X}_{1:T_p}$ into $T_p - T_l - T_f + 1$ sub-sequences $\{\mathbf{X}_{t:t+T_l+T_f-1}\}_{t=1}^{T_p-T_l-T_f+1}$, each containing $T_l + T_f$ consecutive poses. The attention mechanism is then built by treating the first T_l poses of every sub-sequence as key and the entire sub-sequence $\{\mathbf{X}_{t:t+T_l+T_f-1}\}$ as value. In practice, the values are in fact represented in trajectory space as the Discrete Cosine Transform (DCT) coefficients of the corresponding poses. That is, the value of each subsequence is taken as $\{\mathbf{V}_t\}_{t=1}^{T_p-T_l-T_f+1}$, where $\mathbf{V}_t \in \mathbb{R}^{K \times (T_l+T_f)}$ encodes the DCT coefficients. Finally, the query corresponds to the last observed sub-sequence $\mathbf{X}_{T_p-T_l+1:T_p}$ with T_l poses.

The query and keys are computed as the output of two neural networks f_q and f_k, respectively. These functions map the poses to latent vectors of dimension d, that is,

$$\mathbf{q} = f_q(\mathbf{X}_{T_p-T_l+1:T_p}), \tag{6.1}$$

$$\mathbf{k}_t = f_k(\mathbf{X}_{t:t+T_l-1}), \tag{6.2}$$

where $\mathbf{q}, \mathbf{k}_t \in \mathbb{R}^d$ and $1 \leq t \leq T_p - T_l - T_f + 1$. A similarity score a_t is then computed for each

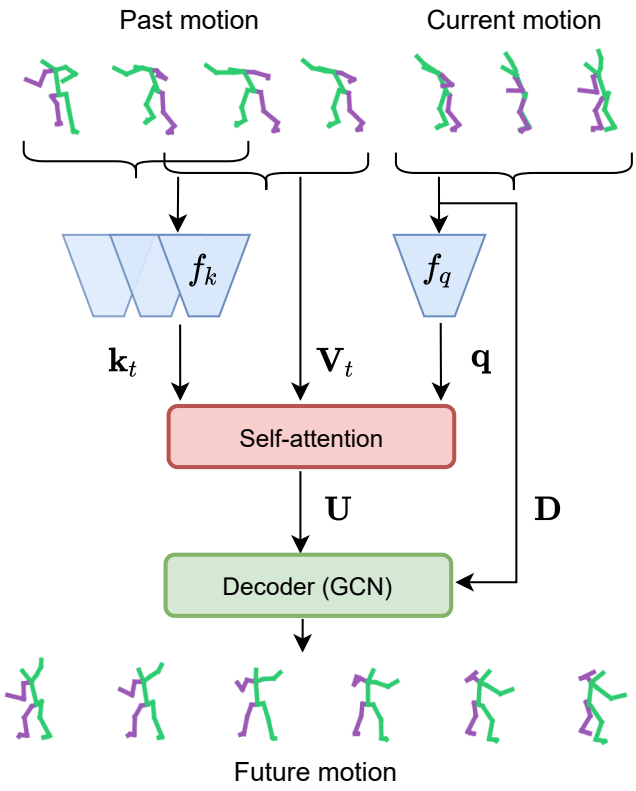


Figure 6.1 – **Single person motion forecasting baseline [181].** The baseline model aggregates information from the history of poses (keys) by comparing them to the last observed sequence of poses (query) through an attention mechanism. f_k and f_q are modeled with convolutional layers. The weighted sum of the values are concatenated with the DCT coefficients of the last observed poses and fed into the GCN that outputs the future pose predictions.

key-query pair, and these scores are employed to obtain a weighted combination of the values. This is expressed as

$$a_t = \frac{\mathbf{q}\mathbf{k}_t^T}{\sum_{j=1}^{T_p-T_l-T_f+1} \mathbf{q}\mathbf{k}_j^T}, \quad \mathbf{U} = \sum_{t=1}^{T_p-T_l-T_f+1} a_t \mathbf{V}_t, \tag{6.3}$$

where $\mathbf{U} \in \mathbb{R}^{K\times(T_l+T_f)}$. Then, the last observed sub-sequence is extended to a sequence of length $T_l + T_f$ by replicating the last pose and passed to the DCT module yielding $\mathbf{D} \in \mathbb{R}^{K\times(T_l+T_f)}$. Finally, $\mathbf{U}$ and $\mathbf{D}$ are fed into the decoder GCN module, which outputs the future pose predictions $\hat{\mathbf{X}}_{T_p+1:T_p+T_f}$. The attention module explained in this section is depicted in Fig. 6.1, and will be referred to as self-attention in the rest of this paper, as it computes the attention of a single person on themselves.

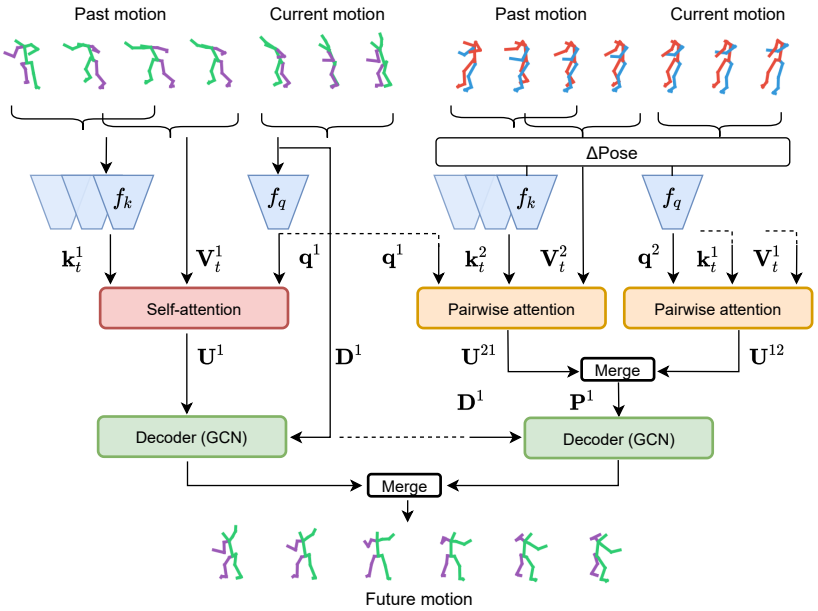


Figure 6.2 – **Overview of our 3D motion forecasting model based on self- and pairwise attention.** Our model takes as input the past poses of the primary (skeleton models depicted using green-purple) and past poses of the auxiliary (red-blue) subject relative to the primary one depicted by ΔPose operation. The superscript 1 is used for the primary subject whereas 2 represents the interactee. As proposed by [181], the self-attention module takes as input the key, query and value vectors of the primary subject. We build on top of this approach by integrating a pairwise module that takes as input the query from one subject and key-value pair from the other subject. This module learns to put higher attention on the sub-sequences in the motion history of the primary subject that are more relevant to the current motion of the interactee. The merge block applies concatenation followed by a convolutional layer. The embeddings from self- and pairwise attention are fed into two separate GCNs with shared weights. The outputs of GCNs are projected to the future pose predictions of the primary subject via the merge block.

6.1.2 Pairwise Attention for Dyadic Interactions

Our goal is to perform motion predictions for multiple people. Formally, given the history of poses $\{\mathbf{X}^s_{1:T_p}\}_{s=1}^S$ for S subjects, our model predicts the future poses $\{\mathbf{X}^s_{T_p+1:T_p+T_f}\}_{s=1}^S$. In particular, we focus on the case where $S=2$ and aim to model the strong dependencies arising from the close interaction of the two subjects. As shown in Fig. 6.2, our approach combines self- and pairwise attention modules, and we refer to one person as the *primary* subject and to the other as the *auxiliary* one, denoted by the superscripts 1 and 2, respectively. Our goal then is to predict the future poses of the primary subject given the observed motions of both. Note that, to predict the future poses of the second subject, we simply inverse the roles.

To combine self- and pairwise attention, we first compute keys $\mathbf{k}_t^1$ and query $\mathbf{q}^1$ vectors for the primary subject as in Eqs. 6.1, 6.2. The values $\mathbf{V}_t^1$ together with $\mathbf{k}_t^1$ and $\mathbf{q}^1$ are then fed into the self-attention module, which yields $\mathbf{U}^1$ as in Eq. 6.3. We then design a pairwise attention module that computes the similarity scores between the keys of the primary subject and the query of the

auxiliary one, and vice-versa, to detect how relevant the coupled motion is at a given time in the past. A straightforward way of incorporating pairwise attention would consist of computing the auxiliary keys and query vectors directly from the observed motion of the auxiliary subject. However, as we show in the experiments, using the relative motion between the primary and auxiliary subject facilitates the modeling of interactions. Therefore, we compute the query, keys and values for the auxiliary subject as

$$\mathbf{q}^2 = f_q(\mathbf{X}^1_{T_p-T_l+1:T_p} - \mathbf{X}^2_{T_p-T_l+1:T_p}) \,, \tag{6.4}$$

$$\mathbf{k}^2_t = f_k(\mathbf{X}^1_{t:t+T_l-1} - \mathbf{X}^2_{t:t+T_l-1}) \,, \tag{6.5}$$

$$\mathbf{V}^2_t = DCT(\mathbf{X}^1_{t:t+T_l+T_f-1} - \mathbf{X}^2_{t:t+T_l+T_f-1}) \,. \tag{6.6}$$

We then define pairwise attention scores between the past motion of the primary subject and the relative motion with respect to the auxiliary one as

$$c^{12}_t = \frac{\mathbf{q}^2 {\mathbf{k}^1}^T_t}{\sum_{j=1}^{T_p-T_l-T_f+1} \mathbf{q}^2 {\mathbf{k}^1}^T_j} \,. \tag{6.7}$$

This lets us compute a weighted sum of primary subject values as

$$\mathbf{U}^{12} = \sum_{t=1}^{T_p-T_l-T_f+1} c^{12}_t \mathbf{V}^1_t \,. \tag{6.8}$$

We also compute $\mathbf{U}^{21}$ using $\mathbf{V}^2_t$ and the pairwise scores c^{21}_t of $\mathbf{q}^1$ and $\mathbf{k}^2_t$. In the final stage of the encoder, we concatenate the pairwise embeddings $\mathbf{U}^{12}$ and $\mathbf{U}^{21}$ and feed them to a convolutional layer corresponding to the merge block in Fig. 6.2. The output is denoted as $\mathbf{P}^1$.

As for single-person prediction, the last observed sub-sequence of the primary subject is extended by repeating its last observed pose and transformed into DCT coefficients denoted by $\mathbf{D}^1$. Our decoder then has two GCNs with shared parameters. One takes as input the concatenated matrices $\mathbf{D}^1$ and $\mathbf{U}^1$ and the other $\mathbf{D}^1$ and $\mathbf{P}^1$. Finally, the GCNs' outputs are projected via a convolutional layer to the future pose predictions of the primary subject. The same strategy is applied when exchanging the roles to obtain the future poses of the second subject.

6.1.3 Training

The entire network is trained by minimizing the Mean Per Joint Position Error (MPJPE). The loss for one training sequence is thus written as

$$L = \frac{1}{J(T_l+T_f)} \sum_{t=T_p-T_l+1}^{T_p+T_f} \sum_{j=1}^{J} ||\hat{\mathbf{x}}_{t,j} - \mathbf{x}_{t,j}||^2 \,, \tag{6.9}$$

where $\hat{\mathbf{x}}_t \in \mathbb{R}^{3\times J}$ encodes the estimated 3D pose for time t, $\mathbf{x}_t$ represents the corresponding ground-truth pose, and $\mathbf{x}_{t,j}$ denotes the 3D position of the j-th joint.

6.1.4 Implementation Details

Training Details. We train our network using the ADAM [127] optimizer with a learning rate of 0.0005 and a batch size of 32. We use $T_p = 60$ poses, corresponding to 2 seconds, as motion history and predict $T_f = 30$ poses, corresponding to 1 second in the future. Our models are trained for 500 epochs, and we report the results of the model with the highest validation score.

Network Structure. The networks f_q and f_k in the self- and pairwise attention modules consist of two 1D convolutional layers with kernel sizes 6 and 5, respectively, each followed by a ReLU. The hidden dimension of the query and key vectors in Eq. 6.1 and Eq. 6.2 is 256. We use a GCN with 12 residual blocks as in [181]. The human skeleton has $J = 19$ joints and our model has approximately $3.27M$ parameters similar to [181] that has $3.26M$ parameters.

6.2 Experiments

In this section, we demonstrate the effectiveness of our approach at exploiting dyadic interactions. To this end, we first introduce our Lindyhop600k dataset depicting couples that perform lindy hop dance movements.

6.2.1 LindyHop600K

Lindy hop is a type of swing dance with fast-paced steps synchronized with the music. It constitutes a good example of motions with strong mutual dependencies between the subjects, who are engaged in close interactions. To build this dataset, we filmed three men and four women dancers paired up in different combinations. Overall, Lindyhop600k contains nine dance sequences, each two to three minutes long, with a maximum of eight cameras at 60 fps. We use the shortest two sequences as validation and test sets. Table 6.1 shows the details of the dataset organization. Our dataset displays standard lindy hop dancer positions and steps, such as the so-called open, closed, side and behind positions. In the open and closed positions, the dancers are facing each other with a varying distance between them. In the side position, both are facing the same direction, and in the behind position, the leader stands directly behind the follower, both facing the same direction. In each position, the dancers communicate through hand and shoulder grips. To the best of our knowledge, Lindyhop600k is the first large dance dataset involving the videos and 3D ground-truth poses of dancers.

To obtain the 3D poses of the dancers, we first extract 2D pixel locations of the visible joints using OpenPose [33]. Because our dataset was captured with multiple cameras, this lets us obtain the 3D joint coordinates by performing a bundle adjustment using the 2D joint locations in all the views. However, this process comes with several problems because it requires annotating the poses of both subjects together. The major issues encompass body part confusions, missing 2D annotations and tracking errors in the OpenPose predictions, which occur when two people are very close to each other or wear similar garments. An example of this is shown in Fig. 6.3.

Sequence	Couple	Frames	Cameras	Split
1	A1	10152	5	Train
2	B2	8819	8	Train
3	C3	6519	8	Validation
4	A4	7687	8	Test
5	B1	9977	8	Train
6	C2	9636	8	Train
7	A3	8930	7	Train
8	B4	9027	8	Train
9	C1	9635	8	Train

Table 6.1 – **Lindyhop600k dataset structure.**

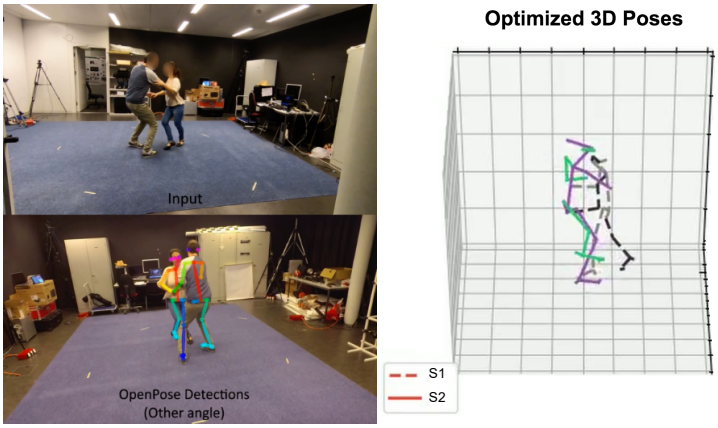
To remedy this, we adopt a solution based on temporal smoothness. Specifically, we assign manually the 2D joint locations to each dancer in the first frame of each sequence. For the subsequent frames, the low confidence joint detections are replaced with ones interpolated using the high confidence joints from the neighboring frames. Despite these 2D joint corrections, the 3D locations extracted from the bundle adjustment procedure can still be very noisy. Thus, we employ a third degree spline interpolation across 30 frames coupled with an optimization scheme to generate the final 3D poses. Since the spline interpolation is done separately for each dimension of each joint, the length of each limb varies from one frame to another. To tackle this problem, we implement an optimization scheme which minimizes the squared difference between the length of a limb c in the current frame and the average length of limb c. We combine this loss function with additional regularizers penalizing feet from sliding on the floor, constraining the shape of the hips and shoulders, and preventing the optimization to the initial 3D pose estimates. For more detail, we refer the reader to the supplementary material.

6.2.2 Data Pre-processing

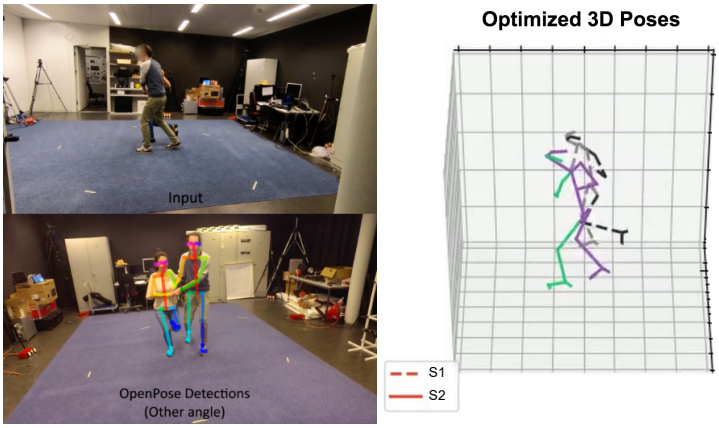
Each video sequence is first downsampled to 30 fps. The human body skeleton in the Lindyhop600k dataset originally comprises of 25 body joints. We remove some of the facial, hand and foot joints and train our models with a skeleton of 19 joints. The 3D joint locations are represented in the world coordinates. Since the position and orientation of the dancers change from one frame to another, we apply a rigid transformation to the poses. We first subtract the global position of the hip center joint from every joint coordinate in every frame. Then, for each sequence, we take the first pose as reference and rotate it such that the unit vector from the left to right shoulder is aligned with the x-axis and the unit vector from the center hip joint to the neck is aligned with the z-axis. We apply the same rotation to all the other poses in the sequence.

6.2.3 Results

In this section, we evaluate our approach depicted by Fig. 6.2 on our new Lindyhop600k dataset. We compare our method with the state-of-the-art single person approaches. They include HRI [181], which relies on an attention mechanism and a GCN decoder [183] to predict the



(a) OpenPose 2D detection failure and the optimized 3D poses



(b) Correct OpenPose detections and the optimized 3D poses

Figure 6.3 – **Optimizing 3D poses in the Lindyhop600k dataset.** (a) Example of OpenPose 2D detection failure. The left leg of the woman is mapped to the left leg of the man. Our multi-view footage and refinement strategy allow us to obtain accurate 3D poses of the dancers despite the mismatch in the 2D detections. (b) Example of correct OpenPose detections and the optimized 3D ground truth poses.

milliseconds	100	200	300	400	500	600	700	800	900	1000	Average
TIM [145]	6.06	12.39	19.83	29.35	41.80	56.91	73.17	89.23	104.31	118.20	51.13
MSR-GCN [167]	9.02	17.02	24.79	33.26	43.69	56.34	70.49	85.00	98.37	109.73	51.11
HRI-Itr [181]	2.21	4.94	9.51	17.71	30.93	49.66	72.95	98.39	122.93	144.24	50.41
HRI [181]	5.34	9.95	15.08	22.19	32.45	45.82	61.29	77.40	92.47	105.15	43.17
Ours	**1.31**	**4.31**	**9.49**	**17.33**	**27.42**	**39.85**	**54.22**	**70.20**	**86.23**	**100.09**	**37.57**

Table 6.2 – **Comparison of our dyadic motion prediction approach with the state-of-the-art single person methods on the Lindyhop600k dataset.** We present the MPJPE for short-term (< 500ms) and long-term (> 500ms) motion prediction in mm. Despite the fast-paced and nonrepetitive nature of the dance moves, our method outperforms all the baselines for both short-term and long-term prediction. The best results in each column are shown in bold.

future poses of the individuals in isolation; HRI-Itr, which uses the output of the predictor as input and predicts the future motion recursively; TIM [145], which extends [183] by combining it with a temporal inception layer to process the input at different subsequence lengths; and MSR-GCN [167], the most recent method, which extracts features from the human body at different scales by grouping the joints in close proximity. All the baselines rely on a GCN architecture that is trained and tested according to the data split shown in Table 6.1. They take as input a sequence of 60 poses as past motion. Except for HRI-Itr that recursively predicts 10 poses at a time, all the baselines predict 30 poses in the future.

In Table 6.2, we report the MPJPE for short-term (< 500ms) and long-term (> 500ms) motion prediction in mm. Our method outperforms the baselines by a large margin. Fig. 6.4, 6.5, 6.6 depict qualitative results of our approach and the best performing three baselines for the Lindyhop600k test subjects with the corresponding follower and leader roles in the top two and bottom two portions, respectively. In contrast to the baselines, our method accurately predicts moves that are hard to anticipate in the long term, such as fast changing feet movements and less frequent arm openings. Although the observed motion of the primary subject does not include sufficient clues for such moves, the second person provides a useful prior so that our model can learn to predict the motion complementary or symmetric to that of the auxiliary subject. Therefore, we attribute this performance to our modeling of the motion dependencies via our pairwise attention mechanism. Failure cases for our approach can be seen in Fig. 6.7. In some rare cases, as in other baselines, our method fails to predict the correct rotation of the body. However, even in such cases, our predictions are plausible in the long-term.

6.2.4 Ablation Study

We evaluate the effect of modeling interactions via different strategies:
HRI-Concat concatenates the motion history of the primary and auxiliary subject to treat them as one person.
Ours-SumPooling, *Ours-AvgPooling* and *Ours-MaxPooling* discard the pairwise attention module, apply self-attention on the sequences of both subjects independently and combines the individual

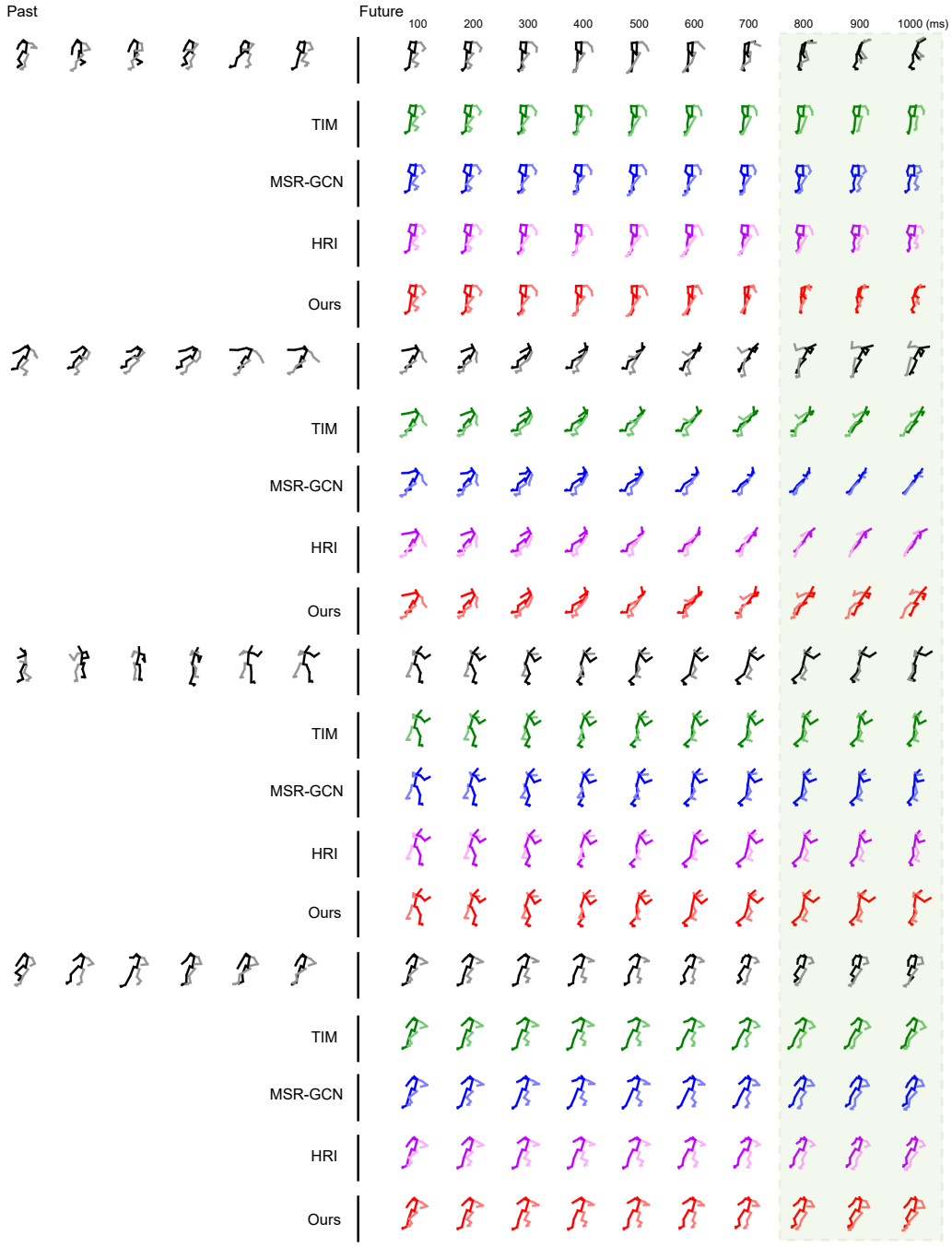


Figure 6.4 – **Qualitative evaluation of our results on the LindyHop600K test subjects compared to the state-of-the-art methods.** Black: Ground truth, green: TIM [145], blue: MSR-GCN [167], violet: HRI [181], red: Ours-Dyadic. The examples show the predictions for dancer with the follower role. The left side of the vertical bar in the black row depicts the sampled input to our model and the right side shows the ground truth future poses. The colored rows correspond to the predictions of the state-of-the-art single person approaches. The red row depicts the output of our model. The numbers at the top indicate the timestamp in milliseconds and the green region highlights the long-term predictions.

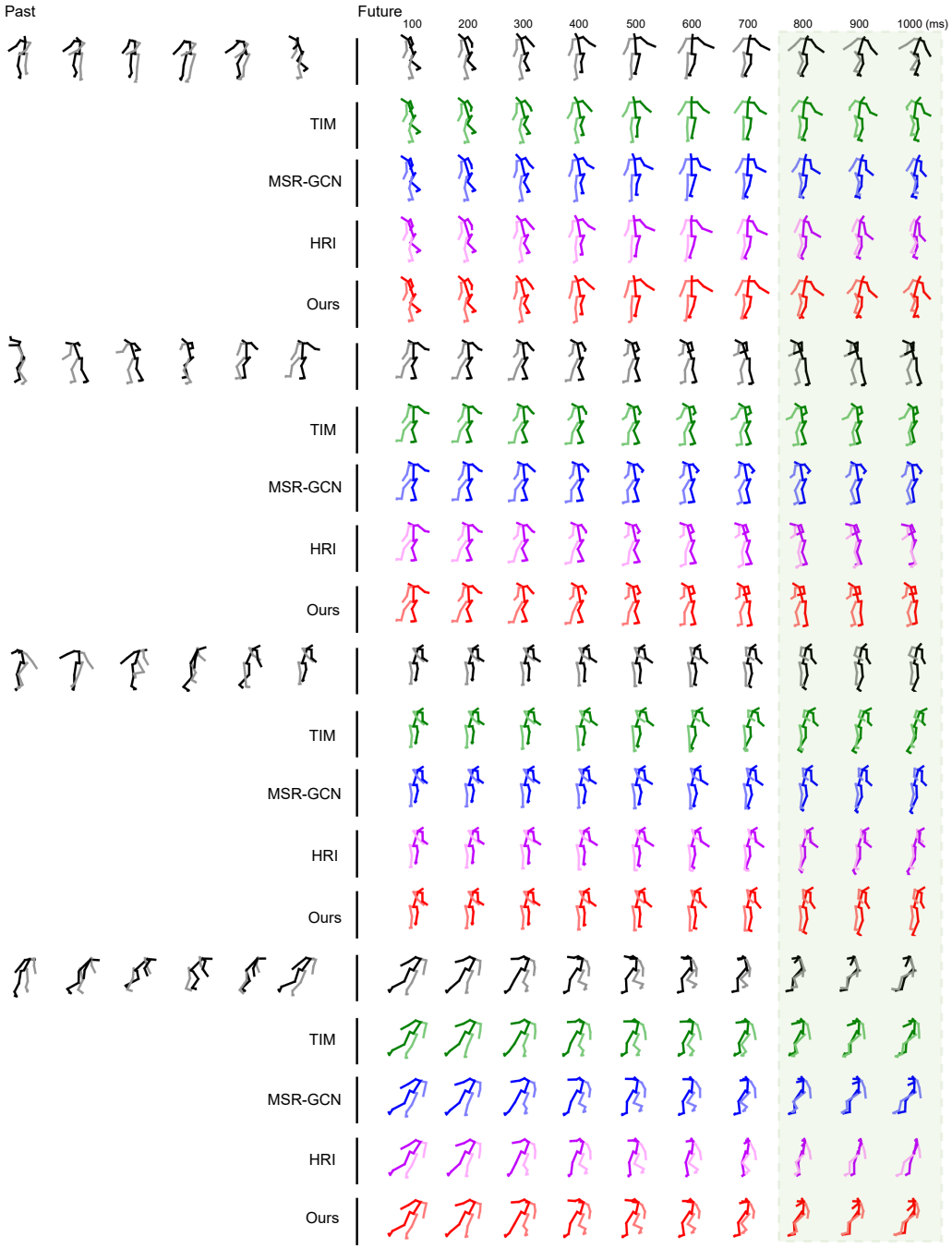


Figure 6.5 – **Qualitative evaluation of our results on the LindyHop600K test subjects compared to the state-of-the-art methods.** Black: Ground truth, green: TIM [145], blue: MSR-GCN [167], violet: HRI [181], red: Ours-Dyadic. The examples show the predictions for dancer with the leader role. The left side of the vertical bar in the black row depicts the sampled input to our model and the right side shows the ground truth future poses. The colored rows correspond to the predictions of the state-of-the-art single person approaches. The red row depicts the output of our model. The numbers at the top indicate the timestamp in milliseconds and the green region highlights the long-term predictions.

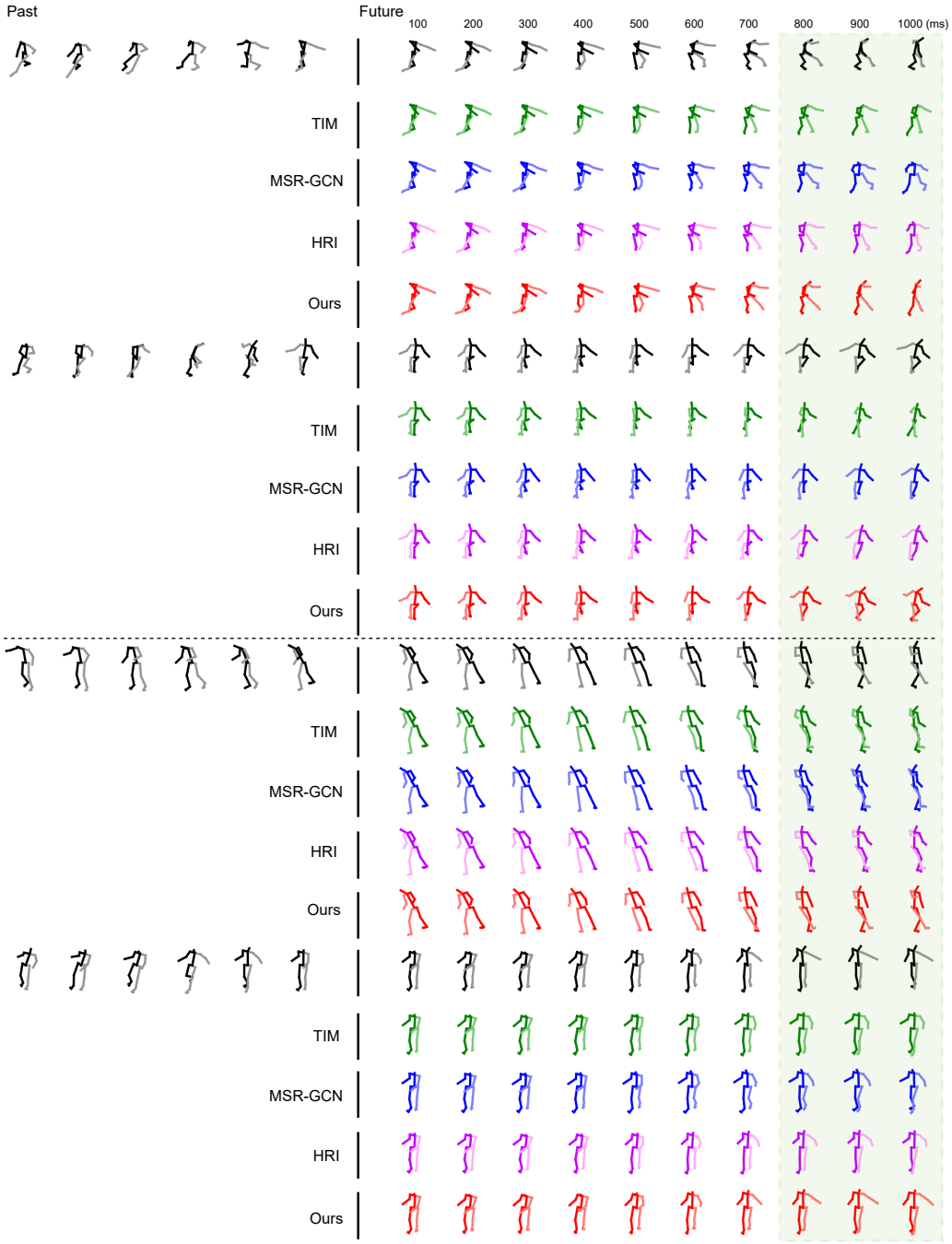


Figure 6.6 – **Qualitative evaluation of our results on the LindyHop600K test subjects compared to the state-of-the-art methods.** Black: Ground truth, green: TIM [145], blue: MSR-GCN [167], violet: HRI [181], red: Ours-Dyadic. Top two portions show the predictions for dancer with the follower role. Bottom two portions show the predictions for the dancer with the leader role. The left side of the vertical bar in the black row depicts the sampled input to our model and the right side shows the ground truth future poses. The colored rows correspond to the predictions of the state-of-the-art single person approaches. The red row depicts the output of our model shown in Fig. 6.2. The numbers at the top indicate the timestamp in milliseconds and the green region highlights the long-term predictions.

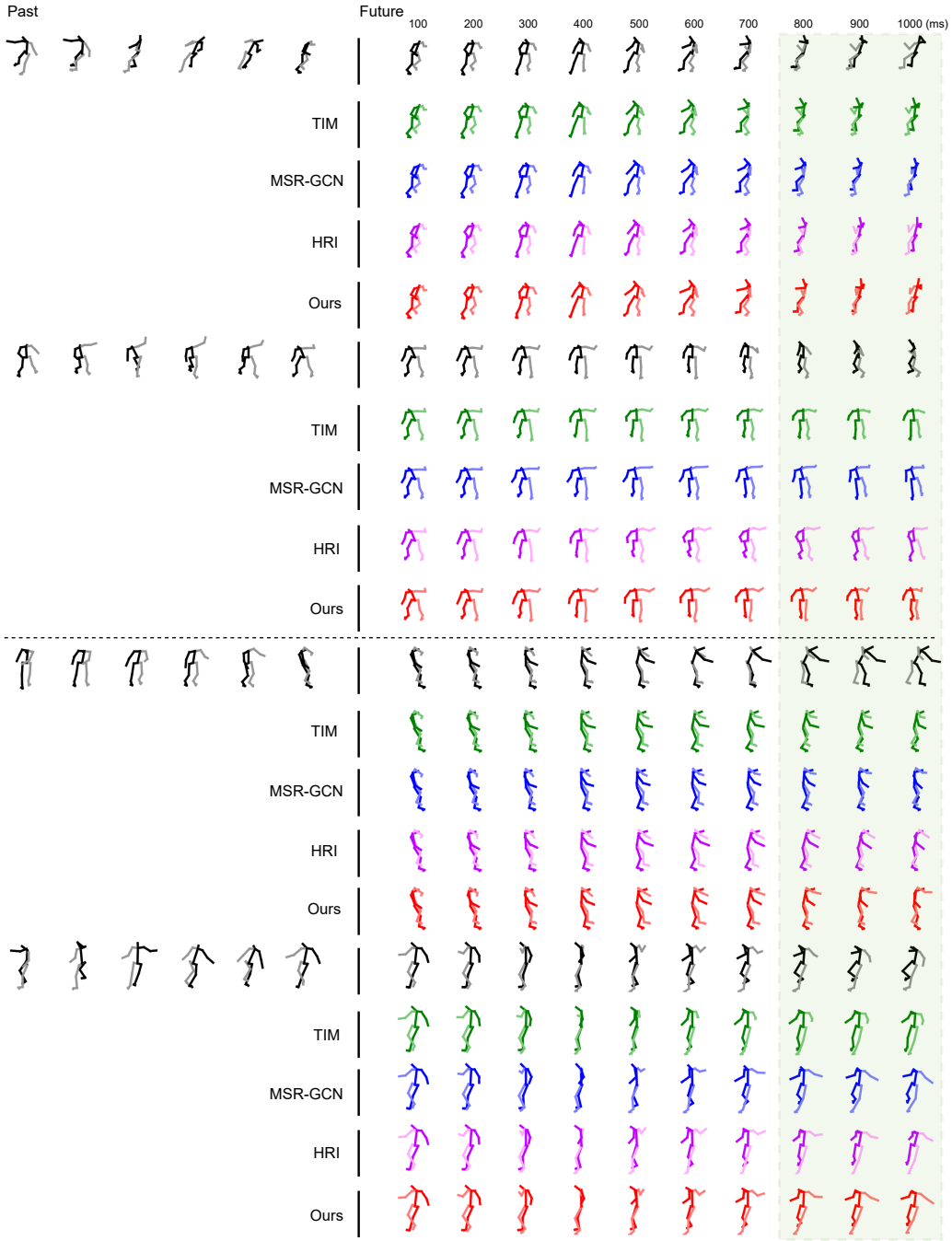


Figure 6.7 – **Example failure cases on the LindyHop600K test subjects.** Black: Ground truth, green: TIM [145], blue: MSR-GCN [167], violet: HRI [181], red: Ours-Dyadic. Top two portions show the predictions for dancer with the follower role. Bottom two portions show the predictions for the dancer with the leader role. The left side of the vertical bar in the black row depicts the sampled input to our model and the right side shows the ground truth future poses. The colored rows correspond to the predictions of the state-of-the-art single person approaches. The red row depicts the output of our model. The numbers at the top indicate the timestamp in milliseconds and the green region highlights the long-term predictions.

milliseconds	100	200	300	400	500	600	700	800	900	1000	Average
HRI-Concat	17.13	33.99	51.32	69.89	90.67	113.41	136.00	156.10	172.06	183.40	96.34
Ours-SumPooling	5.77	10.78	16.07	22.86	32.41	45.17	60.63	77.40	93.45	106.94	43.54
Ours-AvgPooling	5.66	10.47	15.90	23.53	34.46	48.68	65.13	82.19	97.99	111.02	45.77
Ours-MaxPooling	5.07	9.50	14.57	21.65	31.79	44.89	60.13	76.26	91.61	104.72	42.48
Ours-w/oPairwiseAtt	3.60	11.48	25.08	43.00	62.22	81.41	100.25	118.70	135.48	149.39	68.04
Ours-w/oΔPose	3.28	8.36	16.84	23.87	36.77	52.22	68.67	85.02	100.02	112.07	46.33
Ours-EarlyMerge	4.25	8.11	12.78	19.25	28.45	40.84	56.05	73.11	90.27	105.40	40.27
Ours-w/SelfAttAux	1.30	5.04	10.47	18.12	28.95	42.41	57.89	74.52	90.47	104.09	39.76
Ours-PairwiseAtt$\mathbf{U}^{12}$	**1.17**	4.48	9.74	17.82	28.35	41.27	56.25	72.32	88.09	101.77	38.66
Ours	1.31	**4.31**	**9.49**	**17.33**	**27.42**	**39.85**	**54.22**	**70.20**	**86.23**	**100.09**	**37.57**

Table 6.3 – **Ablation study for incorporating interactions.** We present the MPJPE for short-term (< 500ms) and long-term (> 500ms) motion prediction in mm. Here, we analyze different ways of incorporating interactions. HRI-Concat concatenates the motion history of the primary and auxiliary subject to treat them as one person. Ours-SumPooling, Ours-AvgPooling and Ours-MaxPooling use the social pooling layers from [1]. The remaining baselines show the benefits of the different components in our approach. Ours, depicted in Fig. 6.2, outperforms all other baselines and poses an effective way of handling coupled motion. The best results in each column are shown in bold.

embeddings using the different pooling strategies proposed by [1]. The resulting vector is fed to the GCN decoder to predict the future poses of the primary subject.
Ours-w/oPairwiseAtt excludes the pairwise attention module, applies self-attention and the GCN decoder on the sequences of both subjects independently and merges the GCN outputs from the two people to predict the future poses of the primary subject.
Ours-w/oΔPose is our model which takes as input the past motion of the auxiliary subject directly instead of their relative motion to the primary subject.
Ours-EarlyMerge merges the pairwise embeddings $\mathbf{U}^{12}$ and $\mathbf{U}^{21}$ with the self-attention embedding of the primary subject $\mathbf{U}^{1}$ before feeding them to the GCN module.
Ours-w/SelfAttAux applies self-attention also on the sequence of the auxiliary subject and merges the result with the pairwise embeddings $\mathbf{U}^{12}$ and $\mathbf{U}^{21}$.
Ours-PairwiseAtt$\mathbf{U}^{12}$ excludes the pairwise attention that takes the keys and values from the auxiliary and the query from the primary subject.

As can be seen in Table 6.3, our method achieves the highest MPJPE in all timestamps. The comparison with *HRI-Concat* shows that the naive way of combining the motion of the subjects is not an effective strategy to model their dependencies. The results of *Ours-SumPooling*, *Ours-AvgPooling* and *Ours-MaxPooling* show that the social pooling layers proposed by [1] are suboptimal in the presence of strong interactions. The comparison to the remaining baselines evidence the benefits of the different components in our approach, which all contribute to the final results.

6.2.5 Limitations

In Fig. 6.7 and in the additional qualitative results, we observe that the lower arms and feet joints are usually difficult to predict and deviate the most from the ground-truth positions. Although Lindy Hop is a structured dance with highly correlated coupled motion, the dancers have their own styles. Therefore, predicting a single future is likely not to accurately match the body extremities which undergo the largest motion. This, however, can be overcome performing multiple diverse motion prediction, following a similar strategy to that used in [314, 7, 182] for single-person motion prediction.

Another limitation of our model and many other motion prediction works in general is its use of complete sequences of ground-truth 3D poses as input. This may make our model sensitive to missing or faulty observations. To remedy this, as future work, we aim to incorporate the 3D poses obtained from the input images into our forecasting network and handle incomplete or noisy sequences to predict realistic future 3D poses for the interacting people.

6.3 Conclusion

In this chapter, we have devised a novel strategy for exploiting dyadic interactions in 3D human motion prediction. Contrary to the previous work that takes into account the motion history of each person independently, we propose to jointly reason about the observed poses of the subjects engaged in a coupled motion. To this end, we design an encoder-decoder model that leverages self- and pairwise attention mechanisms to learn the mutual dependencies in the collective motion. We introduce a new dataset, Lindyhop600k, to showcase the effectiveness of our model. To the best of our knowledge, this dataset is the first large dance dataset that provides the videos and 3D body pose annotations of couples performing a swing dance. We outperform the current state-of-the-art single person baselines on this dataset and demonstrate that incorporating the interlinked motion of an interactee yields more accurate long term predictions for the primary subject. Our future work will focus on incorporating visual context in motion forecasting, and study not only interactions between two humans but interactions among objects as well.

7 Concluding Remarks

In this thesis, we have presented solutions to self-supervised human detection and segmentation, 3D pose estimation and 3D human motion forecasting. All of the proposed methods benefit from different encoder-decoder models in various ways. Below, we first summarize the contributions of the individual chapters. Then, we discuss the remaining limitations in this field and possible extensions for future research.

7.1 Summary

In Chapter 3, we introduce a self-supervised object detection and segmentation approach that can work effectively on domain-specific images capturing humans. We define the foreground object as an image region that cannot be easily reconstructed from the neighboring scene content via an inpainting network. We integrate this intuition into a proposal-based encoder-decoder model. To tackle the discrete nature of region proposals, we introduce an importance sampling based strategy. Our method can handle large camera motions without requiring any manual annotations.

In Chapter 4, we introduce a self-supervised end-to-end trainable object detection and segmentation approach that explicitly leverages 3D multi-view geometry during training. We construct a 3D object proposal framework that enforces prediction consistency across views without having to introduce additional loss terms. To impose geometric consistency between the bounding boxes from different views, we want to ensure that their 2D centers all match the same point in 3D and that their 2D heights correspond to the same 3D size. This is achieved by solving a least-squares problem using the known camera matrices.

In Chapter 5, we propose to enforce implicit structural constraints on 3D human pose prediction within a deep learning regression framework. We combine traditional CNNs for supervised learning with autoencoders for unsupervised feature learning. The autoencoder is pre-trained on 3D human poses and comprises of a hidden layer of higher dimension than its input and output. The latent representation learned by the autoencoder accounts for the joint dependencies. We refine the pose predictions by imposing temporal consistency on the output of the network

through a LSTM-based architecture.

In Chapter 6, we devise a novel strategy for 3D human motion forecasting with dyadic interactions. We present a pairwise attention mechanism that explicitly takes into account the mutual dependencies in the motion history of the subjects. When combined with the self-attention mechanism and integrated into an encoder-decoder network, our approach can predict long-term future poses more reliably. To showcase the results of our method, we build a new dance dataset, Lindyhop600k, that involves strong human-to-human interactions.

7.2 Limitations and Future Directions

In this section we discuss possible improvements to the proposed methods and potential future directions.

Self-supervised multiple salient object segmentation. In this thesis, we have covered single and multi-view self-supervised strategies for human detection and segmentation in videos with large camera motion. However, in real-world scenarios, it is more common to encounter cluttered scenes with humans interacting with objects. One possible future direction is to devise a multi-object segmentation algorithm that does not require annotations. The current challenge [31] in this field is to identify multiple salient objects that would capture human attention and consistently appear throughout the video sequence. Given object proposals from an instance discrimination network, [324] attempts to design a target-aware tracking network for associating these proposals of the same identities over each image sequence. However, such unsupervised methods require annotations during training and there is still room for improvement for the self-supervised counterparts. Our proposed segmentation method is generic and can be applied to any object category, therefore it would be quite a natural extension.

3D human shape and pose estimation from unconstrained images in the wild. In Chapter 5, we have formulated the 3D human pose estimation as recovering the 3D joint locations of a person via learning pose priors. Following our work, data-driven priors have been widely used in the human motion domain. A similar idea is now achieved through a hierarchical motion variational autoencoder [152]. Instead of learning a single latent space, [152] exploits global and local latent spaces to capture the holistic and refined motion. Our work on 3D human pose estimation does not involve learning the 3D body shape. However, predicting the parameters of a 3D body model provides a useful prior over human body shape and is gaining much attention since it has potential applications in augmented, virtual and mixed reality. This issue can be addressed by learning a latent code on the vertices of the SMPL body model. A neural representation-based method [211] employs novel view synthesis of dynamic humans by mapping a set of latent codes encoding geometry and appearance into density and color fields. To this end, [211] discretizes the 3D bounding box of the human where each voxel has a latent code. In case of having calibrated cameras, the latent code volume can be constructed as explained in Chapter 4 and our multi-view sampling strategy can be used to obtain the latent codes of SMPL vertices.

Other challenges in this task involve estimating the 3D body shape and pose from a group of images of the same human subject without any constraints on the subject's pose, camera viewpoint and surrounding environment. Based on this, [242] probabilistically combines predicted body shape distributions from each image to obtain a final multi-image shape prediction. Recently, [25] takes on a novel direction and predicts a set of plausible 3D meshes corresponding to a single ambiguous input image of a human.

The potential extension of our work to implicit representations learned from latent human shape codes aligns well with current objectives in the virtual reality technology. A crucial part of this technology is based on the virtual avatars of users and requires to combine the physical and virtual worlds in a seamless way. This can be accomplished by building animatable human body and face models [15, 172, 23, 227, 179]. The underlying models rely on a set of latent code encoding geometry and facial expressions. Apart from generating photorealistic avatars, another fundamental element in a virtual environment is the social interactions. It comes with a major challenge to generate plausible behavior for interacting avatars and this task has not been exploited to its full potential yet. One way to tackle it is to combine our work on attention based modeling of dyadic interactions in Chapter 6 with the existing models generating full-body avatars.

Scene-aware 3D human shape and pose synthesis. The standard practice in computer vision is to estimate human pose in isolation from the 3D scene. Realistic placement of 3D people in 3D scenes while accounting for semantic interactions paves the way for new applications in augmented reality. Recently, [87] learns how humans interact with scenes based on conditional variational autoencoder and exploits this to enable virtual characters to do the same. Handling the penetration between the body and scene is currently a major challenge in this task. Similarly, [288] takes into account the interaction between the scene and the human motion. To this end, it devices a GAN-based learning approach to enforce the compatibility between the synthesized human motions and the surrounding scene context. An extension to this could be learning the dynamics of a group of people conditioned on the scene.

3D human motion forecasting based on noisy or incomplete observations. The existing work in 3D motion prediction is highly sensitive to noisy or missing observations in the motion history. In real-world scenarios, such cases are inevitable due to the mutual occlusions of joints or the obstacles in the scene. Even with professional MoCap devices, erroneous measurements can appear in the raw data. Because the current state-of-the-art approaches in human motion prediction use ground truth data as the past sequence of poses, they are limited to complete observations. To generate more accurate and realistic poses in unconstrained settings, a possible future direction is to employ motion infilling conditioned on corrupted past observations and visual context. Recently, [234, 123, 51] have attempted to repair the missing information in motion sequences.

Very long-term motion prediction over 1 second time horizon. The current state-of-the-art 3D motion forecasting methods are limited to predicting the future poses up to 1 second. They tend to

collapse to static predictions for longer timespans. However, for safety-critical applications such as human-robot interaction and autonomous driving, generating plausible and realistic predictions up to several seconds is required. Given a single scene image and 2D pose histories, [32] first samples multiple possible future 2D destinations, and then predicts 3D human path towards each destination within 2-3 second time span. Recently, [126] proposes to predict a few future keyposes and approximate intermediate ones by linearly interpolating the keyposes. This enables to predict diverse futures for long-term durations of 5 seconds.

A Appendix for Chapter 3

In this section, we demonstrate some more qualitative results, and present a mathematical justification for the importance sampling transformations.

A.1 Qualitative Results

The capture setup of our Handheld190k dataset is depicted in Fig. A.1. As shown in Fig. A.1, it is an outside recording. The dataset is composed of five actors performing the same actions as those available in the Human3.6m dataset [102], namely *directions, discussion, eating, greeting, phone talk, posing, buying, sitting, sitting down, smoking, taking photo, waiting, walking, walking dog* and *walking in pair*. We excluded *lying on the floor* actions, not to make our actors lie in the dirt. The data from three actors compose our training set and the other two form the test set. The data was obtained using 3 GoPro6 cameras recording FullHD videos at 30 FPS in linear lens mode. For the entire duration, the cameras were subject to lateral movement and varying hand-held rotation. The motion stablization of the GoPros was deactivated during the recording.

We provide additional examples of our detection and segmentation results on the Ski-PTZ, Handheld190k and Human3.6m test datasets in Fig. A.2, Fig. A.3 and Fig. A.4 respectively.

A.2 Importance Sampling Theory

In the following, we give an explanation for the change of distribution in Section 3.1.4 when computing an expectation. Let p, q be two discrete probability distributions and $f(c)$ an arbitrary function of c. Reasoning about the limit towards infinitely many samples c_k drawn according to

Figure A.1 – **Capture setup of our in-house Handheld190k posing dataset.** The subject is recorded by three persons with handheld GoPro action cameras.

q, we derive

$$\begin{aligned}
\mathbf{E}_q\left[\frac{p_c}{q_c}f(c)\right] &= \lim_{K\to\infty}\frac{1}{K}\sum_{k=1}^{K}\left(\frac{p_{c_k}}{q_{c_k}}f(c_k)\right) \\
&= \lim_{K\to\infty}\frac{1}{K}\sum_{c=1}^{C}\sum_{1}^{\sum_{k=1}^{K}1_{c_k=c}}\left(\frac{p_c}{q_c}f(c)\right) \\
&= \frac{1}{K}\sum_{c=1}^{C}Kq_c\left(\frac{p_c}{q_c}f(c)\right) \\
&= \sum_{c=1}^{C}\Big(p(c)f(c)\Big) \\
&= \mathbf{E}_p\left[f(c)\right], \qquad \text{(A.1)}
\end{aligned}$$

where K is the number of samples drawn and $1_{c_k=c}$ is one if c_k equals c . In the second line, we exploit that we have a finite number of classes C. Each sample must fall into one of them and the probability of coming from cell c is q_c.

This relation provides us with a tool to change the sample distribution for expectations. Next, we analyze the variance of such an estimator depending on the chosen sampling distribution.

Importance Sampling Variance

The variance of an estimator gives us a measure of the expected accuracy with a limited number of samples. The variance of estimating the objective L_{fg} in Eq. 3.2 with a Monte Carlo sum over

$c_1, \ldots, c_K$ samples drawn independently from p is

$$\begin{aligned}\mathrm{Var}\left[L_{\mathrm{fg}}(\mathbf{I})\right] &\approx \mathrm{Var}\left[\frac{1}{K}\sum_{k=1}^{K} L\left(\mathcal{F}_{c_k}(\mathbf{I}), \mathbf{I}\right)\right] \\ &= \frac{1}{K^2}\sum_{k=1}^{K} \mathrm{Var}\left[L\left(\mathcal{F}_{c_k}(\mathbf{I}), \mathbf{I}\right)\right] \\ &= \frac{1}{K}\mathrm{Var}\left[L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right] . \end{aligned} \tag{A.2}$$

Here, $\mathrm{Var}\left[L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right]$ is the variance over the random variable c, and we utilized the identity $\mathrm{Var}[ax] = a^2\,\mathrm{Var}[x]$ and the independence of samples. The variance reduces linearly with the number of samples and is proportional to that of $L(\cdot)$.

Using uniform sampling $q = U_c$, yields a quadratic variance growth with the number of cells, i.e.,

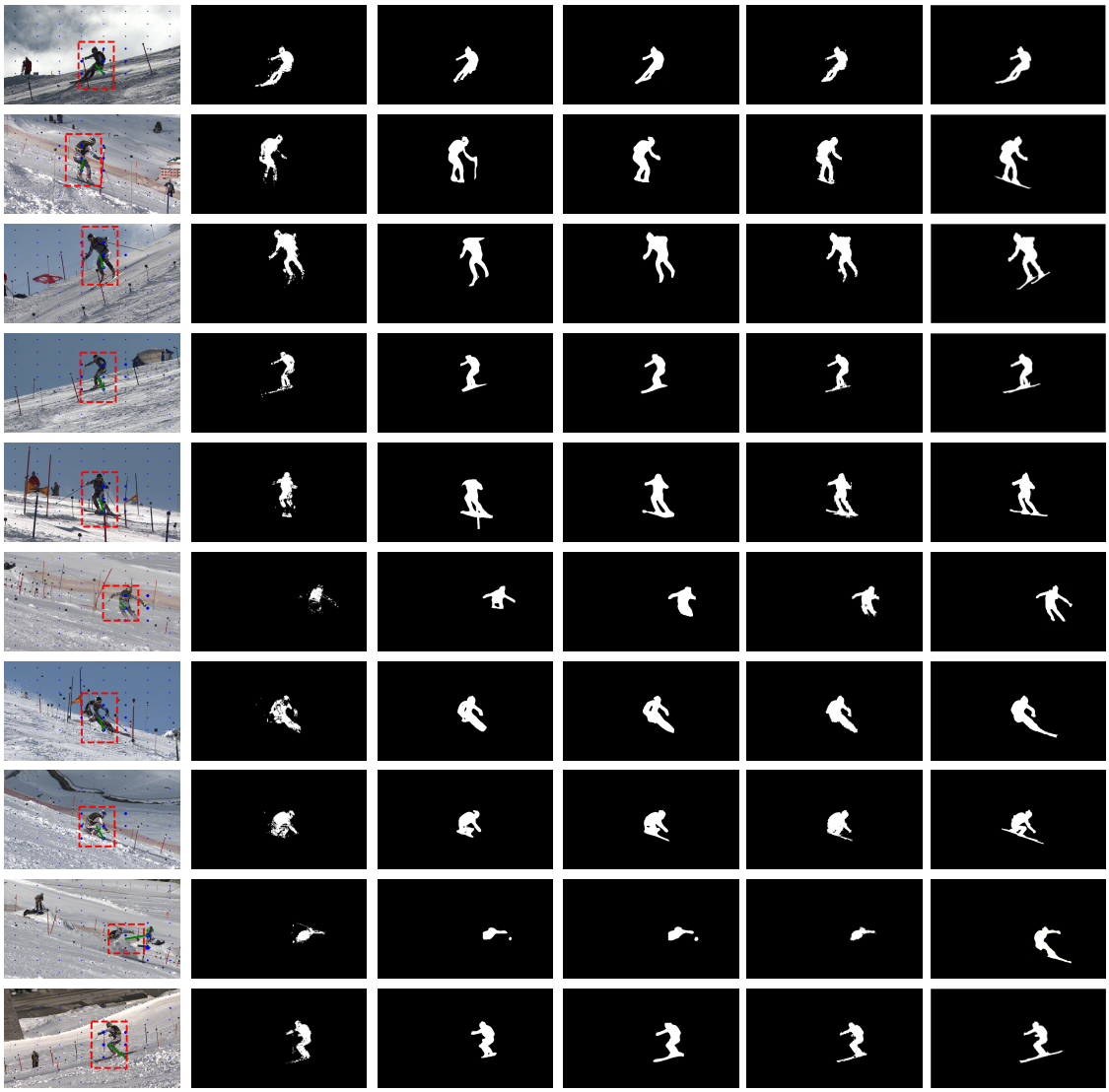
$$\begin{aligned}\mathrm{Var}\left[L_{\mathrm{fg}}(\mathbf{I})\right] &\approx \frac{1}{K}\mathrm{Var}\left[\frac{p_c}{\mathcal{U}_C(c)} L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right] \\ &= \frac{1}{K}\mathrm{Var}\left[C p_c L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right] \\ &= \frac{C^2}{K}\mathrm{Var}\left[p_c L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right] . \end{aligned} \tag{A.3}$$

This lets us conclude that uniform sampling leads to a higher variance, since the remaining $\mathrm{Var}[p(\cdot)L(\cdot)]$ term is not expected to improve on $\mathrm{Var}[L(\cdot)]$.

With importance sampling according to q, the variance is

$$\mathrm{Var}\left[L_{\mathrm{fg}}(\mathbf{I})\right] \approx \frac{1}{K}\mathrm{Var}\left[\frac{p_c}{q_c} L\left(\mathcal{F}_{c}(\mathbf{I}), \mathbf{I}\right)\right] . \tag{A.4}$$

If $q \approx p$, it is equivalent to the one of Eq. A.2. In general, q should be constructed to minimize Eq. A.4. In our case, $\mathcal{F}$ depends on each individual image and cell c. As such, it is difficult to impose assumptions to reduce the variance of L without evaluating it for each c. Therefore, setting $q \approx p$ is a good choice from the perspective of variance reduction.



(a) Inp/Ours detection (b) [305] (c) Ours w/o flow (d) Ours w/ flow (e) Ours w/ flow+CRF (f) Ground truth

Figure A.2 – **Additional detection and segmentation results on the test subjects of Ski-PTZ.** (a) The detection results show the predicted bounding box with red dashed lines, the relative confidence of the grid cells with blue dots and the bounding box center offset with green lines (better viewed on screen). (b) Segmentation mask prediction of [305]. (c) Our segmentation mask prediction obtained by training our method without optical flow. (d) Our segmentation mask prediction obtained by training our method with the proposed optical flow strategy. (e) CRF post-processing applied to our result in (d). (f) Ground truth segmentation mask.

(a) Inp/Ours detection (b) [305] (c) Ours w/o flow (d) Ours w/ flow (e) Ours w/ flow+CRF (f) Ground truth

Figure A.3 – **Additional detection and segmentation results on the test subjects of Handheld190k.** (a) The detection results show the predicted bounding box with red dashed lines, the relative confidence of the grid cells with blue dots and the bounding box center offset with green lines (better viewed on screen). (b) Segmentation mask prediction of [305]. (c) Our segmentation mask prediction obtained by training our method without optical flow. (d) Our segmentation mask prediction obtained by training our method with the proposed optical flow strategy. (e) CRF post-processing applied to our result in (d). (f) Ground truth segmentation mask.

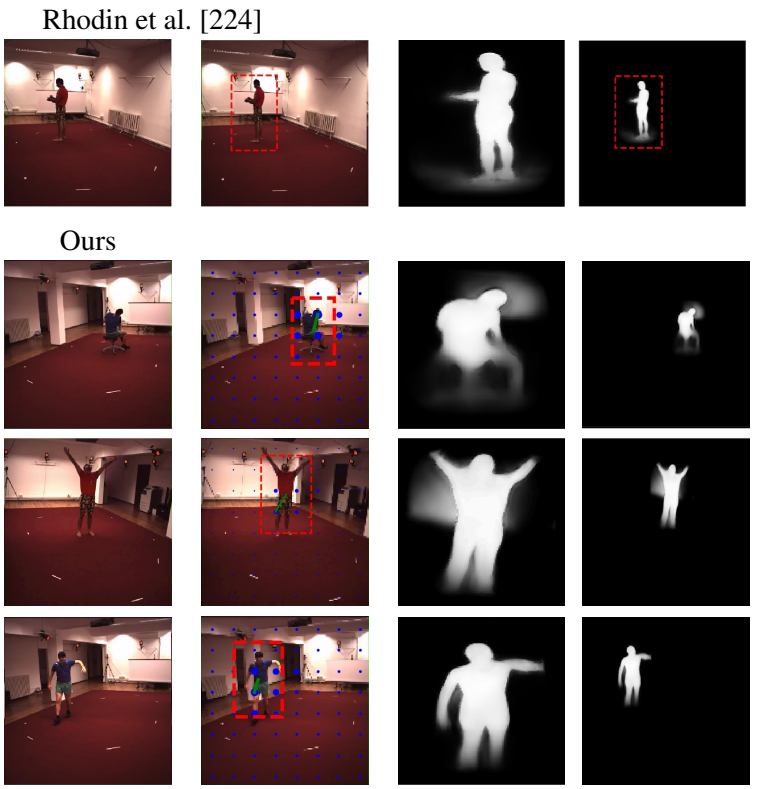


Figure A.4 – **Detection and segmentation results on Human3.6m.** Results match in quality with those from [224], with a slight bleeding due to not having a perfect background prediction oracle.

B Appendix for Chapter 4

In this section, we present the details of our differentiable multi-view consistency formulation, architecture design, and training strategies. Furthermore, we show additional qualitative results on the Ski-PTZ, Human3.6m and Handheld190k datasets. We provide a video which can be accessed on https://youtu.be/bg3AYjTa1NY to explain the multi-view consistency setup and demonstrate the Ski-PTZ and Handheld190k video results including the intermediate outputs of our method along with the detection and segmentation predictions on consecutive frames for all cameras and test subjects.

B.1 Implementation Details

B.1.1 Multi-view Consistency

Adjusting Bounding Box Centers. The candidate object location proposed by the sampled grid cell in camera z is defined as $\mathbf{b}_z = [\delta x, \delta y, s_x, s_y]$ where $\delta x, \delta y \in [0,1]$ are the offsets from the grid center and $s_x, s_y \in [0,1]$ are the width and height of the bounding box respectively. The center location of the proposal in pixel coordinates can be written as

$$\begin{aligned} u_z &= W * \delta x + g_x \,, \\ v_z &= H * \delta y + g_y \,, \end{aligned} \tag{B.1}$$

where $u_z \in [0, W]$, $v_z \in [0, H]$ and $g_x \in [0, W]$, $g_y \in [0, H]$ denote the grid center in pixel coordinates. To reach a multi-view consensus on the center of a 3D bounding box, namely $\bar{\mathbf{u}} \in \mathbb{R}^{3\times 1}$, we take into account the lines emerging from camera positions $\mathbf{o}_z \in \mathbb{R}^{3\times 1}$ for each camera. The line of sight for the proposal center is calculated as

$$\mathbf{l}_z = \mathbf{M}_z^{-1} \begin{bmatrix} u_z \\ v_z \\ 1 \end{bmatrix} , \tag{B.2}$$

where $\mathbf{l}_z$ represents all the points corresponding to the center of the sampled box in world coordinates relative to the camera center and $\mathbf{M}_z$ is the 3×3 matrix formed by the first 3 columns of the projection matrix $\mathbf{P}_z$. Note that we use bold symbols ($\mathbf{l}_z$) for vectors in 3D world space and normal letters (u and v) for coordinates in the 2D image plane. The unit direction vector for each of these lines is

$$\mathbf{n}_z = \frac{\mathbf{l}_z}{\|\mathbf{l}_z\|}\,, \tag{B.3}$$

where $\mathbf{n}_z \in \mathbb{R}^{3\times 1}$. To find the nearest point $\bar{\mathbf{u}}$ to a set of lines, we calculate the point with minimum distance to them. Given that each line is defined by its origin $\mathbf{o}_z$ and the unit direction vector $\mathbf{n}_z$, the squared perpendicular distance from the point $\bar{\mathbf{u}}$ to one of these lines is given by

$$\mathbf{d}_z = (\mathbf{o}_z - \bar{\mathbf{u}})^T(\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})(\mathbf{o_z} - \bar{\mathbf{u}})\,, \tag{B.4}$$

where the matrix $(\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})$ serves as the projector of the line vectors into the space orthogonal to $\mathbf{n}_z$. By minimizing the sum of squared distances, we can obtain the nearest point in the least squares sense for Z cameras. The objective we want to minimize is

$$\sum_{z=1}^{Z} \mathbf{d}_z = \sum_{z=1}^{Z} (\mathbf{o}_z - \bar{\mathbf{u}})^T(\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})(\mathbf{o_z} - \bar{\mathbf{u}})\,. \tag{B.5}$$

The derivative with respect to $\bar{\mathbf{u}}$ gives

$$\sum_{z=1}^{Z} -2(\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})(\mathbf{o_z} - \bar{\mathbf{u}}) = \mathbf{0}\,, \tag{B.6}$$

where $\mathbf{I}$ is the 3×3 identity matrix. Re-arranging this, we obtain a system of linear equations

$$\begin{aligned} \mathbf{A}\bar{\mathbf{u}} &= \mathbf{m}\,, \\ \mathbf{A} &= \sum_{z=1}^{Z} (\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})\,, \\ \mathbf{m} &= \sum_{z=1}^{Z} (\mathbf{I} - \mathbf{n_z}(\mathbf{n_z})^\mathbf{T})\mathbf{o_z}\,, \end{aligned} \tag{B.7}$$

with $\mathbf{A} \in \mathbb{R}^{\mathbf{3\times 3}}$ and $\mathbf{m} \in \mathbb{R}^{\mathbf{3\times 1}}$. The optimum is achieved at the least squares solution. Therefore, $\bar{\mathbf{u}} = lstsq(\mathbf{A}, \mathbf{m})$ and we use a differentiable implementation of *lstsq* function to solve it.

The new center computed through multi-view consistency is projected onto each view to update the value of the 2D bounding box centers. Thus,

$$\begin{bmatrix} \bar{u}_z \\ \bar{v}_z \\ 1 \end{bmatrix} = \mathbf{M}_z\bar{\mathbf{u}}\,, \tag{B.8}$$

where $\bar{\mathbf{u}}$ represents the coordinates of the new center in 3D and $\{\bar{u}_z, \bar{v}_z\}$ are the updated 2D

bounding box centers in each view.

Adjusting Bounding Box Heights. Similarly, the top and bottom points of the 2D bounding boxes can be subject to the multi-view consistency. The top and bottom locations, $\{u_{t,z}, v_{t,z}\}$ and $\{u_{b,z}, v_{b,z}\}$ respectively, of the bounding box in camera view z are computed as

$$\begin{aligned} u_{t,z} &= W * \delta x + g_x \,, \\ v_{t,z} &= H * \delta y + g_y - (H * s_x)/2 \,, \\ u_{b,z} &= W * \delta x + g_x \,, \\ v_{b,z} &= H * \delta y + g_y + (H * s_x)/2 \,. \end{aligned} \tag{B.9}$$

To find the consensus top and bottom locations in 3D, we apply the least squares solution explained in the previous section separately to the top and bottom points. For the top point, we consider the set of lines originating from camera positions $\mathbf{o}_z \in \mathbb{R}^{3\times 1}$ for each camera. The line of sight for the top and bottom points of the bounding box in camera view z are given as

$$\begin{aligned} \mathbf{l}_{t,z} &= \mathbf{M}_z^{-1} \begin{bmatrix} u_{t,z} \\ v_{t,z} \\ 1 \end{bmatrix} \,, \\ \mathbf{l}_{b,z} &= \mathbf{M}_z^{-1} \begin{bmatrix} u_{b,z} \\ v_{b,z} \\ 1 \end{bmatrix} \,. \end{aligned} \tag{B.10}$$

To find the nearest point $\bar{\mathbf{u}}_t$ to these lines, we apply Eq. B.3, B.4, B.5, B.6 and B.7. Finally, we obtain the updated pixel location for the top point of the bounding box as follows

$$\begin{bmatrix} \bar{u}_{t,z} \\ \bar{v}_{t,z} \\ 1 \end{bmatrix} = \mathbf{M}_z \bar{\mathbf{u}}_t \,. \tag{B.11}$$

We update the 2D bottom location using the same multi-view least-squares strategy.

B.1.2 Architectures

Our main network $\mathcal{F}$ consists of a detection and a synthesis network that reconstruct the input scene against the background image generated by the inpainting network.

Detection network. We predict one candidate bounding box relative to each 2D grid cell in a regular 8×8 grid using a fully-convolutional architecture similar to YOLO [222]. We use a ResNet-18 backbone [89] without pre-training, that reduces the input dimensionality by a factor 16, forming a low resolution grid of features, e.g., to spatial resolution 8×8 from 128×128.

The feature size is set to 5; two for bounding box location offset $\delta x, \delta y \in [0,1]$, two for scale $s_x, s_y \in [0,1]$, and one for the probability p. Each feature output represents the bounding box parameters predicted by one grid cell, and the offset is relative to the cell center $\{g_x, g_y\}$. The output p is forced to be positive and form a proper distribution, with $\sum_{c=1}^{C} p_c = 1$ where $C = 64$, by a soft-max activation unit. To prevent this network from constantly predicting bounding boxes at the borders of the image, we zero out the outer cell probabilities.

Synthesis network. This network takes as input the cropped image region corresponding to the sampled bounding box and has the form of a bottle-neck autoencoder, based on the publicly available implementation of [225]. The encoding part is a 50-layer residual network, and the weights are initialized with ones trained on ImageNet classification. The hidden layer is 856 dimensional, split into a 600 dimensional space and a 256 dimensional space that is replicated spatially to a $512 \times 8 \times 8$ feature map to encode spatially invariant features. The decoding is done with the second half of a U-Net architecture with 64, 128, 256, 512 feature channels in each stage. The final network layer outputs four feature maps, three to predict the color image $\hat{\mathbf{I}} \in \mathbb{R}^{128\times128\times3}$ and one for the segmentation mask $\mathbf{S} \in \mathbb{R}^{128\times128}$.

Inpainting network. The inpainting network is trained separately for each dataset, from scratch and on the training split, without requiring any annotation. It is a 6 layer U-Net model with 8, 16, 32, 64, 128, 256 feature channels in each stage. It is trained independently from the rest of the pipeline by feeding images with randomly occluded regions of varying sizes. To compare the reconstructed image $\mathbf{I}'$ to the original one $\mathbf{I}$, we use the L_2 pixel reconstruction and perceptual losses

$$L_{reconst} = ||\mathbf{I} - \mathbf{I}'||^2 , \tag{B.12}$$

$$L_{perc} = ||\phi(\mathbf{I}) - \phi(\mathbf{I}')||^2 , \tag{B.13}$$

where $\phi(.)$ indicates the low level features obtained by passing its input to a pre-trained ResNet18 network. The pixel reconstruction and perceptual losses are weighted 1:2.

We integrate the inpainting network to our full pipeline and use it in an off-the-shelf manner. The input to the inpainter is an image where the selected bounding box region is hidden and the output is the entire image with the initially hidden patch being reconstructed. In our full pipeline, the weights of the inpainting network are frozen and to remove the image evidence corresponding to the foreground person, the hidden patch in the input image to the inpainting network is selected to be the bounding box region expanded by 15% in both dimensions.

B.1.3 Training Details

Overall training. We train our model with L_2 pixel reconstruction and perceptual losses on the reconstructed image $\mathcal{F}(\mathbf{I}_z)$ and the L_2 pixel reconstruction loss on the inpainted background image $\bar{\mathbf{I}}_z$ in view z . We rely on the same prior terms as in [120] to regularize the predicted segmentation

masks and probability values for the voxels. We use an L_{seg} prior, which encourages the mean value of a segmentation mask to be larger than a threshold lambda but small in general,

$$L_{seg} = \left| \left(\frac{1}{WH} \sum_{x}^{W} \sum_{y}^{H} \mathcal{T}^{-1}(\mathbf{S})_{xy} \right) - \lambda \right| + \lambda \,, \tag{B.14}$$

where λ is set to 0.005. It encourages a non-zero segmentation mask at the beginning of the training, when the decoder still produces non-perfect foreground, which improves and stabilizes convergence. The voxel probabilities q_j are regularized with

$$L_q = \sum_{j}^{V} |q_j| \tag{B.15}$$

that favors only few voxels to have non-zero values. The total training loss we minimize can be written as

$$\begin{aligned} L_{total} = & -\alpha \sum_{z=1}^{Z} r_j \frac{\|\bar{\mathbf{I}}_z - \mathbf{I}_z\|^2}{area(\mathbf{b}^z_{i^z(j)})} \\ & + \beta \sum_{z=1}^{Z} r_j \|\mathcal{F}(\mathbf{I}_z) - \mathbf{I}_z\|^2 \\ & + \gamma \sum_{z=1}^{Z} r_j \|\phi(\mathcal{F}(\mathbf{I}_z)) - \phi(\mathbf{I}_z)\|^2 \\ & + \eta \sum_{z=1}^{Z} L^z_{seg} + \zeta L_q \end{aligned} \tag{B.16}$$

where $\alpha = 0.1, \beta = 1, \gamma = 2, \eta = 0.25, \zeta = 0.1$ and $\phi(.)$ indicates the low level features obtained by passing its input to a pre-trained ResNet18 network. The first three terms of L_{total} correspond to $L_{\text{bg}}(\mathbf{I}_1, \ldots, \mathbf{I}_Z)$, $L_{\text{fg}}(\mathbf{I}_1, \ldots, \mathbf{I}_Z)$ and the perceptual version of $L_{\text{fg}}(\mathbf{I}_1, \ldots, \mathbf{I}_Z)$.

As a baseline (*Ours w/ TC*), we report the results of using a L_2 loss term to minimize the distance between lines passing through the initial bounding box centers and camera optical centers in each view.

All training stages are performed on a single NVIDIA TITAN X Pascal GPU with Adam and a learning rate of 1e-3. First, the inpainting network is optimized for 100k iterations and subsequently the complete network for an additional 50k iterations. The decoding part of the synthesis network uses a reduced learning rate of 1e-4, to prevent occasional diverging behavior. We use a batch size of 48 and an input image resolution of 640px×360px for the Ski-PTZ and Handheld190k and 500px×500px for Human3.6m.

Importance sampling. Sampling from a discrete distribution is not differentiable with respect to its parameters. Therefore, we integrate importance sampling as in [120]. However, instead of sampling from a 2D grid of cells, we sample a voxel from a 3D grid of proposals. Importance sampling allows us to introduce an auxiliary distribution k that is used as the importance sampling distribution while maintaining the differentiability and optimizing the voxel probability

distribution q. The relationship between k and q can be expressed as

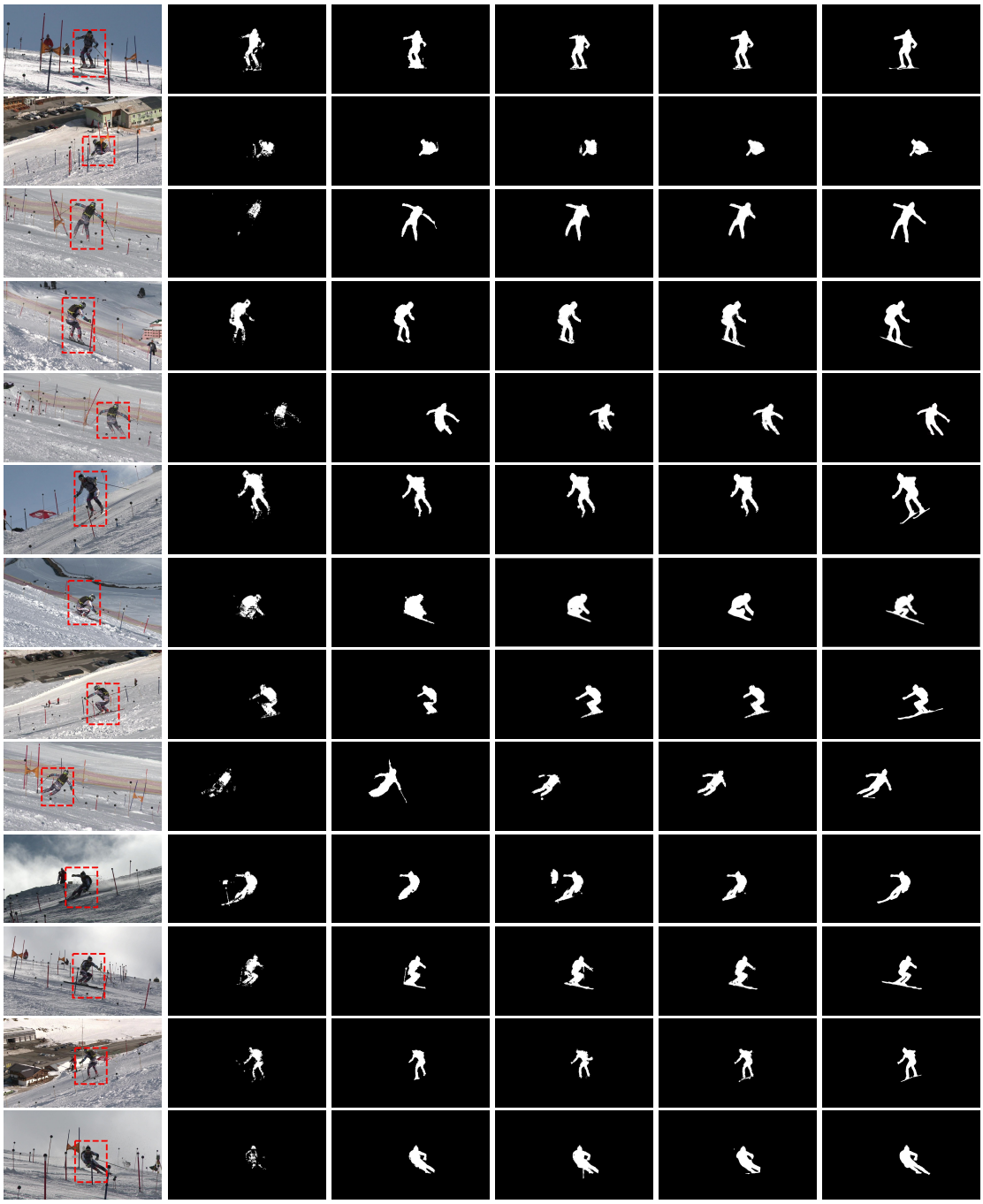
$$k_j = q_j(1 - V\epsilon) + \epsilon \tag{B.17}$$

for a voxel j, where $(1 - V\epsilon)$ determines the probability of choosing a random voxel. In the multi-view setting, the number of voxels that can be seen by all the cameras change from one frame to another. Therefore the importance sampling related hyper-parameters must be adjusted accordingly. As in [120], we take $(1 - V\epsilon) = 0.064$ and to satisfy this equality, we use an adaptive $\epsilon \approx 0.0002$, which makes the method numerically stable while the probability of choosing a random bounding box stays low, i.e., 6.4% for on average $V = 300$ voxels that participate the multi-view consensus voting. In Section 4.1.1, r_j is the ratio of the probability q_j by its importance sampling probability k_j.

Consistency. We demonstrate that the proposed training strategy is stable and produces consistent results when repeated using the same configuration. To this end, we train the best-performing model on the Ski-PTZ and Human3.6m datasets three times from scratch and provide the mean and std of the scores on the test sequences. The J- and F- measures on the Ski-PTZ dataset are consistent, respectively, 0.71 ± 0.006, 0.83 ± 0.002 and the $\text{mAP}_{0.5}$ score on Human3.6m dataset is 0.85 ± 0.004.

B.2 Qualitative Results

We present additional qualitative results on Ski-PTZ, Human3.6m and Handheld190k in Fig. B.1, Fig. B.2 and Fig. B.3 respectively. On Ski-PTZ, our method reliably detects the skier even when there are other people in the scene and our segmentation predictions cover the entire body and skis more accurately than [305], relying on multi-view consistency

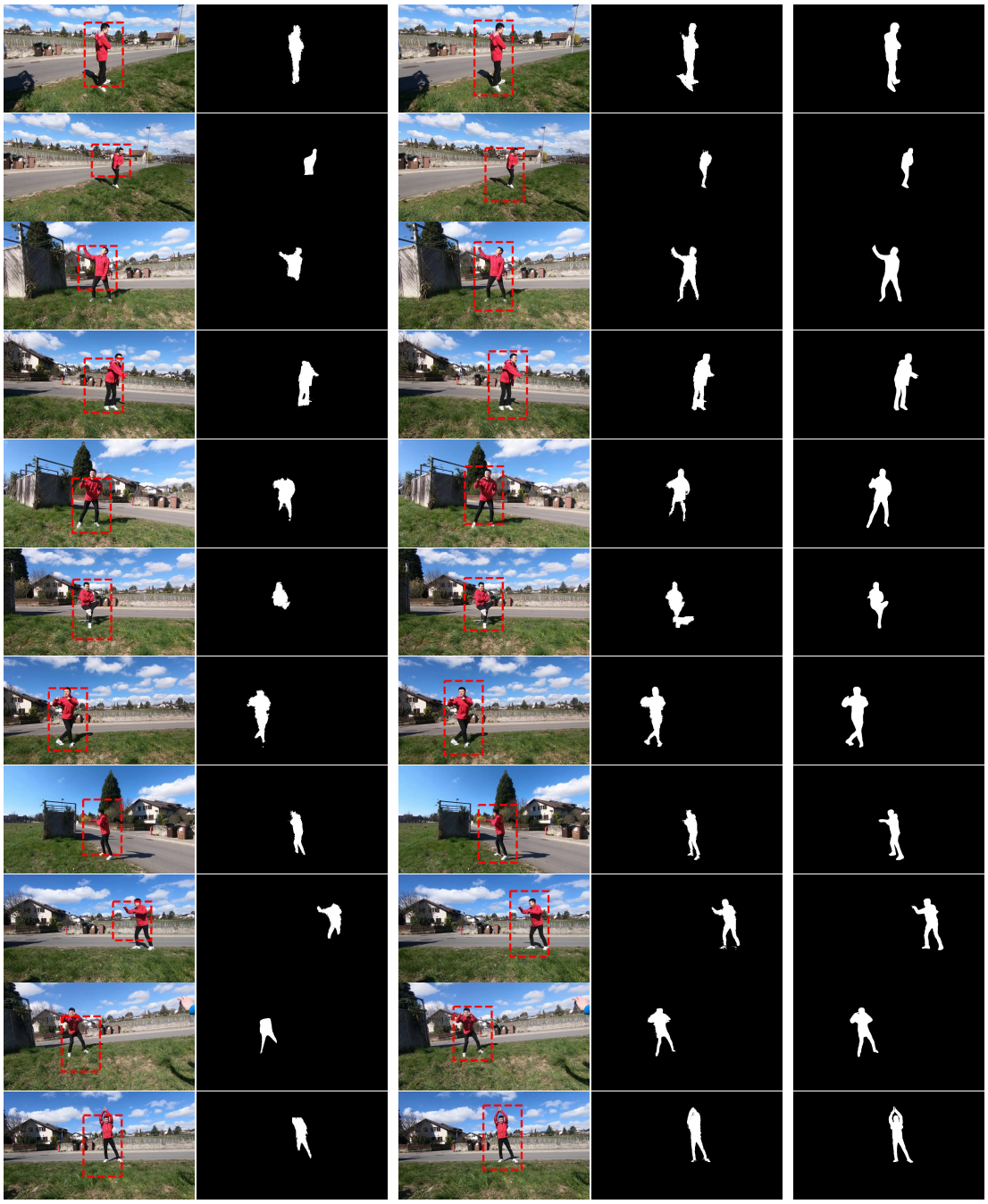


(a) Inp/Ours detection (b) [305] (c) [132] (d) [120] (e) Ours (f) GT

Figure B.1 – **Additional multi-view consistency results on the Ski-PTZ.** (a) Input images with our predicted bounding box overlaid in red. (b) Segmentation mask prediction of [305]. (c) Segmentation mask prediction of [132]. (d) Segmentation mask prediction of [120] (e) Our segmentation mask prediction. (f) Ground truth segmentation mask. Note that, unlike our method, [132] and [305] use explicit temporal cues at inference time.

(a) [120] (b) [224] (c) Ours (d) GT

Figure B.2 – **Additional multi-view consistency results on the Human3.6m dataset.** (a) The detection and segmentation mask results of Katircioglu et al. [120] trained and tested on single images. (b) The results of [224] trained with a pair of camera views and tested on single images. (c) Our predictions obtained from the model trained with the 4-cam multi-view consistency and tested on single images. (d) Ground truth.



(a) [120] (b) Ours (c) GT

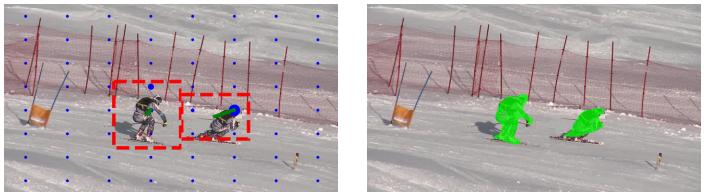
Figure B.3 – **Additional multi-view consistency results on the Handheld190k dataset.** (a) The detection and segmentation mask results of [120] trained and tested on single images. (b) The predictions of our model trained using 3-camera multi-view consistency and tested on single images. (c) Ground truth. Our results are generally more accurate, which justifies the effort invested in calibrating the cameras.

during training, whereas [132] uses strong temporal cues both during training and test time and [120] leverages optical flow images during training. Due to the background objective, our approach favors tight bounding boxes around the subject and this causes the auxiliary object moving with the primary one to be partially included in the detection. Therefore, compared to the ground truth masks, our predictions do not contain the skis entirely. However, compared to other baselines, our method can segment out the skis more precisely.

On Human3.6m dataset, our method has more accurate hand detections and lower legs are more precisely segmented compared to [120] employing a single view approach with optical flow images during training and [224] using multi-view images during training for novel view synthesis. Even in the rare cases of performing an action on the floor, our method can still reliably detect the person. The failure cases include the detections that miss the head and feet when the chair is in close proximity to the subject. This is expected since the chair is also hard to be reconstructed from its neighboring regions and can be treated as a foreground object.

To demonstrate that our method can be applied to in-the-wild scenes without initial camera calibration, we used the OpenSFM software to calibrate 4200 frames out of 120000 training images in the Handheld190k dataset. We ran OpenSFM with HaHOG (the combination of Hessian Affine feature point detector and HOG descriptor) features and the calibration took approximately 7 hours. We did not provide masks for the moving objects. Nonetheless, we managed to obtain accurate camera poses. Our results in Fig. B.3 show that when trained with a small calibrated part of the training set, our multi-view approach can detect and segment the person more accurately than [120] which often fails to detect the moving object precisely.

Although we target the detection of a single object or person, our probabilistic framework can handle several of them at test time by sampling more than once. In Fig. B.4, we show an example of this on Ski-PTZ, by synthetically creating an image with two skiers. Our method trained on single person images can accurately detect and segment two skiers as long as they are sufficiently separated.



(a) Detection (b) Segmentation

Figure B.4 – **Multi-person detection and segmentation at test time.**

Bibliography

[1] V. Adeli, E. Adeli, I. Reid, J.C. Niebles, and H. Rezatofighi. Socially and Contextually Aware Human Motion and Pose Forecasting. In *International Conference on Intelligent Robots and Systems*, 2020.

[2] A. Agarwal and B. Triggs. 3D Human Pose from Silhouettes by Relevance Vector Regression. In *Conference on Computer Vision and Pattern Recognition*, 2004.

[3] A. Agarwal and B. Triggs. Recovering 3D Human Pose from Monocular Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[4] I. Akhter and M. J. Black. Pose-Conditioned Joint Angle Limits for 3D Human Pose Reconstruction. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[5] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[6] A. Alahi, V. Ramanathan, and L. Fei-Fei. Socially-Aware Large-Scale Crowd Forecasting. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[7] S. Aliakbarian, F. S. Saleh, L. Petersson, S. Gould, and M. Salzmann. Contextually Plausible and Diverse 3D Human Motion Prediction. In *International Conference on Computer Vision*, 2021.

[8] S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould. A Stochastic Conditioning Scheme for Diverse Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[9] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele. Multi-View Pictorial Structures for 3D Human Pose Estimation. In *British Machine Vision Conference*, 2013.

[10] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D Pose Estimation and Tracking by Detection. In *Conference on Computer Vision and Pattern Recognition*, 2010.

[11] M. Andriluka and L. Sigal. Human Context: Modeling Human-Human Interactions for Monocular 3D Pose Estimation. In *International Conference on Articulated Motion and Deformable Objects*, pages 260–272, 2012.

[12] D. Anguelov, S. P., D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape Completion and Animation of People. *ACM Transactions on Graphics*, 24:408–416, 2005.

[13] R. Arandjelovic and A. Zisserman. Object Discovery with a Copy-Pasting GAN. In *arXiv Preprint*, 2019.

[14] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing*

Systems, 2017.

[15] T. Bagautdinov, C. Wu, T. Simon, F. Prada, T. Shiratori, S.-E. Wei, W. Xu, Y. Sheikh, and J. Saragih. Driving-Signal Aware Full-Body Avatars. *ACM Transactions on Graphics*, 40(4):1–17, 2021.

[16] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, 2015.

[17] P. Baqué, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-Target Detection. In *International Conference on Computer Vision*, 2017.

[18] O. Barnich and M. Van Droogenbroeck. Vibe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE Transactions on Image processing*, 20(6):1709–1724, 2011.

[19] E. Barsoum, J. Kender, and Z. Liu. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[20] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3D Pictorial Structures for Multiple Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[21] Y. Benny and L. Wolf. OneGAN: Simultaneous Unsupervised Learning of Conditional Image Generation, Foreground Segmentation, and Fine-Grained Clustering. In *European Conference on Computer Vision*, 2020.

[22] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. V. Gool, and R. Timofte. Learning What to Learn for Video Object Segmentation. In *European Conference on Computer Vision*, 2020.

[23] S. Bi, S. Lombardi, S. Saito, T. Simon, S.-E. Wei, K. Mcphail, R. Ramamoorthi, Y. Sheikh, and J. Saragih. Deep Relightable Appearance Models for Animatable Faces. *ACM Transactions on Graphics*, 4:1–15, 2021.

[24] A. Bielski and P. Favaro. Emergence of Object Segmentation in Perturbed Generative Models. In *Advances in Neural Information Processing Systems*, 2019.

[25] B. Biggs, S. Erhardt, H. Joo, B. Graham, A. Vedaldi, and D. Novotny. 3D Multi-bodies: Fitting Sets of Plausible 3D Human Models to Ambiguous Image Data. In *Advances in Neural Information Processing Systems*, 2020.

[26] L. Bo and C. Sminchisescu. Twin Gaussian Processes for Structured Prediction. *International Journal of Computer Vision*, 2010.

[27] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In *European Conference on Computer Vision*, 2016.

[28] M. Burenius, J. Sullivan, and S. Carlsson. 3D Pictorial Structures for Multiple View Articulated Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[29] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In *European Conference on Computer Vision*, 2012.

[30] B.Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn. CanonPose: Self-Supervised Monocular 3D Human Pose Estimation in the Wild. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[31] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K.Maninis, and L. V. Gool. The 2019 DAVIS Challenge on VOS: Unsupervised Multi-Object Segmentation. In *CVPR 2019 Workshop*, 2019.

[32] Z. Cao, H. Gao, K. Mangalam, Q. Cai, M. Vo, and J. Malik. Long-Term Human Motion Prediction with Scene Context. In *European Conference on Computer Vision*, 2020.

[33] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Conference on Computer Vision and Pattern Recognition*, pages 1302–1310, 2017.

[34] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human Pose Estimation with Iterative Error Feedback. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[35] S. Casas, C. Gulino, R. Liao, and R. Urtasun. SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data. In *International Conference on Robotics and Automation*, 2020.

[36] K. Chen, P. Gabriel, A. Alasfour, C. Gong, W. K. Doyle, O. Devinsky, D. Friedman, P. Dugan, L. Melloni, T. Thesen, D. Gonda, S. Sattar, S. Wang, and V. Gilja. Patient-Specific Pose Estimation in Clinical Environments. *IEEE Journal of Translational Engineering in Health and Medicine*, 6:1–11, 2018.

[37] M. Chen, T. Artieres, and L. Denoyer. Unsupervised Object Segmentation by Redrawing. In *Advances in Neural Information Processing Systems*, 2019.

[38] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing Training Images for Boosting Human 3D Pose Estimation. In *International Conference on 3D Vision*, 2016.

[39] X. Chen, K.-Y. Lin, W. Liu, C. Qian, and L. Lin. Weakly-Supervised Discovery of Geometry-Aware Representation for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[40] X. Chen and A. L. Yuille. Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations. In *Advances in Neural Information Processing Systems*, 2014.

[41] J. Cheng, Y. H. Tsai, S. Wang, and M. H. Yang. Segflow: Joint Learning for Video Object Segmentation and Optical Flow. In *International Conference on Computer Vision*, 2017.

[42] H.-K. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles. Action-Agnostic Human Pose Forecasting. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[43] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised Object Discovery and Localization in the Wild: Part-Based Matching with Bottom-Up Region Proposals. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[44] H. Ci, C. Wang, X. Ma, and Y. Wang. Optimizing Network Structure for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2019.

[45] M. Cormier, F. Ropke, T. Golda, and J. Beyerer. Interactive Labeling for Human Pose Estimation in Surveillance Videos. In *International Conference on Computer Vision Workshops*, 2021.

[46] E. Corona, A. Pumarola, G. Alenyà, and F. Moreno-Noguer. Context-Aware Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[47] C. Cortes, M. Mohri, and J. Weston. A General Regression Technique for Learning Transductions. In *International Conference on Machine Learning*, 2005.

[48] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In *International Conference on Computer Vision*, 2017.

[49] E. Crawford and J. Pineau. Spatially Invariant Unsupervised Object Detection with Convolutional Neural Networks. In *Conference on Artificial Intelligence*, 2019.

[50] I. Croitoru, S. V. Bogolin, and M. Leordeanu. Unsupervised Learning of Foreground Object Segmentation. *International Journal of Computer Vision*, 127:1279–1302, 2019.

[51] Q. Cui and H. Sun. Towards Accurate 3D Human Motion Prediction from Incomplete Observations. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[52] Q. Cui, H. Sun, Y. Kong, X. Zhang, and Y. Li. Efficient Human Motion Prediction Using Temporal Convolutional Generative Adversarial Network. *Information Sciences*, 545:427–447, 2021.

[53] R. Dabral, A. Mundhada, U. Kusupati, S. Afaque, A. Sharma, and A. Jain. Learning 3D Human Pose from Structure and Motion. In *European Conference on Computer Vision*, 2018.

[54] A. Dehghan, A. R. Zamir, H. Idrees, and M. Shah. Automatic Detection and Tracking of Pedestrians in Videos with Various Crowd Densities. In *Proceedings of PED*, 2012.

[55] N. Deo and M. M. Trivedi. Convolutional Social Pooling for Vehicle Trajectory Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[56] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[57] M. Du and R. Chellappa. Face Association Across Unconstrained Video Frames Using Conditional Random Fields. In *European Conference on Computer Vision*, 2012.

[58] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-Less 3D Human Motion Capture with Monocular Image Sequence and Height-Maps. In *European Conference on Computer Vision*, 2016.

[59] F. Dutil, C. Gulcehre, A. Trischler, and Y. Bengio. Plan, Attend, Generate: Planning for Sequence-To-Sequence Models. In *Advances in Neural Information Processing Systems*, 2017.

[60] A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient Convnet-Based Marker-Less Motion Capture in General Scenes with a Low Number of Cameras. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[61] S.M.A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. In *Advances in Neural Information Processing Systems*, 2016.

[62] A. Faktor and M. Irani. Video Segmentation by Non-Local Consensus Voting. In *British Machine Vision Conference*, 2014.

[63] O.D. Faugeras and Q.T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[64] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with

a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.

[65] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent Network Models for Human Dynamics. In *International Conference on Computer Vision*, 2015.

[66] M. Fürst, S. T. P. Gupta, R. Schuster, O. Wasenmüller, and D. Stricker. HPERL: 3D Human Pose Estimation from RGB and LiDAR. In *International Conference on Pattern Recognition*, 2021.

[67] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-Time Human Pose Tracking from Range Data. In *European Conference on Computer Vision*, 2012.

[68] D.M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, 73(1), January 1999.

[69] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning Human Motion Models for Long-Term Predictions. In *International Conference on 3D Vision*, 2017.

[70] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *arXiv Preprint*, 2018.

[71] G. Gkioxari, A. Toshev, and N. Jaitly. Chained Predictions Using Convolutional Neural Networks. In *European Conference on Computer Vision*, 2016.

[72] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[73] P. M. Glynn. Likelihood Ratio Gradient Estimation for Stochastic Systems. *Communications of the ACM*, 33(10):75–84, 1990.

[74] K. Gong, J. Zhang, and J. Feng. PoseAug: A Differentiable Pose Augmentation Framework for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[75] A. Gopalakrishnan, A. Mali, D. Kifer, C. L. Giles, and A. G. Ororbia. A Neural Temporal Model for Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[76] A. Graves, S. Fernandez, and J. Schmidhuber. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In *International Conference on Artificial Neural Networks*, 2005.

[77] D. Grest, V. Kruger, and R. Koch. Single View Motion Tracking by Depth and Silhouette Information. In *Scandinavian Conference on Image Analysis*, 2007.

[78] L.-Y. Gui, Y.-X. Wang, X. L., and J. M. F. Moura. Adversarial Geometry-Aware Human Motion Prediction. In *European Conference on Computer Vision*, 2018.

[79] R.A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense Human Pose Estimation in the Wild. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[80] W. Guo, E. Corona, F. M.-Noguer, and X. A.-Pineda. Pi-Net: Pose Interacting Network for Multi-Person Monocular 3D Pose Estimation. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

[81] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[82] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt. In the Wild Human

Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[83] E. Haller and M. Leordeanu. Unsupervised Object Segmentation in Video by Efficient Selection of Highly Probable Positive Features. In *International Conference on Computer Vision*, 2017.

[84] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards Viewpoint Invariant 3D Human Pose Estimation. In *European Conference on Computer Vision*, 2016.

[85] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[86] M. Hassan, V. Choutas, D. Tzionas, and M. J. Black. Resolving 3D Human Pose Ambiguities with 3D Scene Constraints. In *International Conference on Computer Vision*, 2019.

[87] M. Hassan, P. Ghosh, J.Tesch, D. Tzionas, and M. J. Black. Populating 3D Scenes by Learning Human-Scene Interactions. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[88] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *International Conference on Computer Vision*, 2017.

[89] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[90] D. Helbing and P. Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282, 1995.

[91] M. Brand H. Hertzmann. Style Machines. In *ACM SIGGRAPH*, 2000.

[92] N. Hesse, A. S. Schröder, W. Müller-Felber, C. Bodensteiner, M. Arens, and U. G. Hofmann. Body Pose Estimation in Depth Images for Infant Motion Analysis. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2017.

[93] G. Hinton and R. Salakutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006.

[94] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[95] Hololens. https://www.microsoft.com/en-us/hololens.

[96] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang. Multimodal Deep Autoencoder for Human Pose Recovery. *IEEE Transactions on Image Processing*, 2014.

[97] M. R. I. Hossain and J. J. Little. Exploiting Temporal Information for 3D Human Pose Estimation. In *European Conference on Computer Vision*, 2018.

[98] Y. T. Hu, J. B. Huang, and A. G. Schwing. Unsupervised Video Object Segmentation Using Motion Saliency-Guided Spatio-Temporal Propagation. In *European Conference on Computer Vision*, 2018.

[99] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang. STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. In *International Conference on Computer Vision*, 2019.

[100] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[101] C. Ionescu, F. Li, and C. Sminchisescu. Latent Structured Models for Human Pose Estimation. In *International Conference on Computer Vision*, 2011.

[102] C. Ionescu, I. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[103] U. Iqbal, P. Molchanov, and J. Kautz. Weakly-Supervised 3D Human Pose Learning via Multi-view Images in the Wild. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[104] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov. Learnable Triangulation of Human Pose. In *International Conference on Computer Vision*, pages 7718–7727, 2019.

[105] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[106] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler. Learning Human Pose Estimation Features with Convolutional Networks. In *International Conference on Learning Representations*, 2014.

[107] A. Jain, A.R. Zamir, and S. Savaresea A. adn Saxena. Structural-Rnn: Deep Learning on Spatio-Temporal Graphs. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[108] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[109] E. Jang, S. Gu, and B. Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

[110] S. Jenni and P. Favaro. Self-Supervised Feature Learning by Learning to Spot Artifacts. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[111] S. Jenni and P. Favaro. Self-Supervised Multi-View Synchronization Learning for 3D Pose Estimation. In *Asian Conference on Computer Vision*, 2020.

[112] S. Jenni, H. Jin, and P. Favaro. Steering Self-Supervised Feature Learning Beyond Local Pixel Statistics. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[113] W. Jiang, N. Kolotouros, G. Pavlakos, X. Zhou, and K. Daniilidis. Coherent Reconstruction of Multiple Humans from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[114] J. Johnson, A. Alahi, and L. Fei-fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision*, pages 694–711, 2016.

[115] S. Johnson and M. Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *British Machine Vision Conference*, 2010.

[116] H. Y. Jung, Y. Suh, G. Moon, and K. M. Lee. A Sequential Approach to 3D Human Pose Estimation: Separation of Localization and Identification of Body Joints. In *European Conference on Computer Vision*, 2016.

[117] H. Kahn and A. W. Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

[118] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-To-End Recovery of Human

Shape and Pose. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[119] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-To-End Recovery of Human Shape and Pose. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[120] I. Katircioglu, H. Rhodin, V. Constantin, J. Spörri, M. Salzmann, and P. Fua. Selfsupervised Human Detection and Segmentation via Background Inpainting. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[121] I. Katircioglu, H. Rhodin, J. Spörri, M. Salzmann, and P. Fua. Self-Supervised Human Detection and Segmentation via Multi-View Consensus. In *International Conference on Computer Vision*, 2021.

[122] I. Katircioglu, B. Tekin, M. Salzmann, V. Lepetit, and P. Fua. Learning Latent Representations of 3D Human Pose with Deep Neural Networks. *International Journal of Computer Vision*, 126(12):1326–1341, 2018.

[123] M. Kaufmann, E. Aksan, J. Song, F. Pece, R. Ziegler, and O.Hilliges. Convolutional Autoencoders for Human Motion Infilling. In *International Conference on 3D Vision*, 2020.

[124] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multi-View Body Part Recognition with Random Forests. In *British Machine Vision Conference*, 2013.

[125] M. Keuper, B. Andres, and T. Brox. Motion Trajectory Segmentation via Minimum Cost Multicuts. In *International Conference on Computer Vision*, 2015.

[126] S. Kiciroglu, W. Wang, M. Salzmann, and P. Fua. Long Term Motion Prediction Using Keyposes. In *arXiv Preprint*, 2021.

[127] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2015.

[128] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

[129] M. Kocabas, N. Athanasiou, and M. J. Black. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[130] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black. PARE: Part Attention Regressor for 3D Human Body Estimation. In *International Conference on Computer Vision*, 2021.

[131] M. Kocabas, S. Karagoz, and E. Akbas. Self-supervised Learning of 3d Human Pose Using Multi-view Geometry. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[132] Y. J. Koh and C.-S. Kim. Primary Object Segmentation in Videos Based on Region Augmentation and Reduction. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[133] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting Self-Supervised Visual Representation Learning. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[134] D. Koller and N. Friedman. *Probabilistc Graphical Models*. The MIT Press, 2009.

[135] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. In *International Conference on Computer Vision*, 2019.

[136] S. Kombrink and L. Burget T. Mikolov, M. Karafiat. Recurrent Neural Network based

Language Modeling in Meeting Recognition. In *Annual Conference of the International Speech Communication Association*, 2011.

[137] K. Konda, R. Memisevic, and D. Krueger. Zero-Bias Autoencoders and the Benefits of Co-Adapting Features. In *International Conference on Learning Representations*, 2015.

[138] V. Kosaraju, A. Sadeghian, R. Mart´ın-Mart´ın, I. Reid, H. Rezatofighi, and S. Savarese. Social-Bigat: Multimodal Trajectory Forecasting Using Bicycle-Gan and Graph Attention Networks. In *Advances in Neural Information Processing Systems*, 2019.

[139] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In *Advances in Neural Information Processing Systems*, 2018.

[140] P. Krähenbühl and V. Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems*, 2011.

[141] V. Kress, J. Jung, S. Zernetsch, K. Doll, and B. Sick. Human Pose Estimation in Real Traffic Scenes. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 518–523, 2018.

[142] J. N. Kundu, M. Gor, and R. V. Babu. BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN. In *AAAI Conference on Artificial Intelligence*, 2019.

[143] J. N. Kundu, S. Seth, V. Jampani, and M. Rakesh. Self-Supervised 3D Human Pose Estimation via Part Guided Novel Image Synthesis. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[144] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M.J. Black, and P.V. Gehler. Unite the People: Closing the Loop Between 3D and 2D Human Representations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[145] T. Lebailly, S. Kiciroglu, M. Salzmann, P. Fua, and W. Wang. Motion Prediction Using Temporal Inception Module. In *Asian Conference on Computer Vision*, 2020.

[146] K. Lee, I. Lee, and S. Lee. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency. In *European Conference on Computer Vision*, 2018.

[147] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[148] Y.J. Lee, J. Kim, and A. K. Grauman. Key-Segments for Video Object Segmentation. In *International Conference on Computer Vision*, 2011.

[149] A. M. Lehrmann, P. V. Gehler, and S. Nowozin. Efficient Nonlinear Markov Models for Human Motion. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[150] M. Leordeanu. *Unsupervised Learning in Space and Time*. Springer, 2020.

[151] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee. Convolutional Sequence to Sequence Model for Human Dynamics. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[152] J. Li, R. Villegas, D. Ceylan, J. Yang, Z. Kuang, H. Li, and Y. Zhao. Task-Generic Hierarchical Human Motion Prior using VAEs. In *International Conference on 3D Vision*, 2021.

[153] J. Li, F. Yang, H. Ma, S. Malla, M. Tomizuka, and C. Choi. RAIN: Reinforced Hybrid Attention Inference Network for Motion Forecasting. In *International Conference on Computer Vision*, 2021.

[154] L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun. End-To-End Contextual Perception and Prediction with Interaction Transformer. In *International Conference on Intelligent Robots and Systems*, 2020.

[155] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Symbiotic Graph Neural Networks for 3D Skeleton-Based Human Action Recognition and Motion Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[156] S. Li and A.B. Chan. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In *Asian Conference on Computer Vision*, 2014.

[157] S. Li, L. Ke, K. Pratama, Y.-W. Tai, C.-K. Tang, and K.-T. Cheng. Cascaded Deep Monocular 3D Human Pose Estimation With Evolutionary Training Data. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[158] S. Li, B. Seybold, A. Vorobyov, A. Fathi, Q. Huang, and C.-C. J. Kuo. Instance Embedding Transfer to Unsupervised Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[159] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. J. Kuo. Unsupervised Video Object Segmentation with Motion-Based Bilateral Networks. In *European Conference on Computer Vision*, 2018.

[160] S. Li, W. Zhang, and A. B. Chan. Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2015.

[161] S. Li, W. Zhang, and A. B. Chan. Maximum-Margin Structured Learning with Deep Networks for 3D Human Pose Estimation. In *IJCV*, 2016.

[162] Y. Li, Z. Shen, and Y. Shan. Fast Video Object Segmentation Using the Global Context Module. In *European Conference on Computer Vision*, 2020.

[163] M. Liang and X. Hu. Recurrent Convolutional Neural Network for Object Recognition. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[164] M. Lin, L. Lin, X. Liang, K.Wang, and H. Cheng. Recurrent 3D Pose Sequence Machines. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[165] T-.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[166] Z. Lin, Y.-F. Wu, S. V. Peri, W. Sun, G. Singh, F. Deng, J. Jiang, and S. Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *International Conference on Learning Representations*, 2020.

[167] D. Lingwei, N. Yongwei, L. Chengjiang, Z. Qing, and L. Guiqing. MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction. In *International Conference on Computer Vision*, 2021.

[168] R. Liu, J. Shen, H. Wang, C. Chen, S.-C. Cheung, and V. Asari. Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[169] S. Liu, H. Guo, H. Pan, P. Wang, X. Tong, and Y. Liu. Deep Implicit Moving Least-Squares Functions for 3D Reconstruction. In *Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2021.

[170] Y. Liu, Q. Yan, and A. Alahi. Social NCE: Contrastive Learning of Socially-Aware Motion Representations. In *International Conference on Computer Vision*, 2021.

[171] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou. Multimodal Motion Prediction with Stacked Transformers. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[172] S. Lombardi, J. Saragih, T. Simon, and Y. Sheikh. Deep Appearance Models for Face Rendering. *ACM Transactions on Graphics*, 37:1–13, 2018.

[173] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M.J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM SIGGRAPH Asia*, 34(6), 2015.

[174] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. V. Gool. Video Object Segmentation with Episodic Graph Memory Networks. In *European Conference on Computer Vision*, 2020.

[175] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli. See More, Know More: Unsupervised Video Object Segmentation with Co-Attention Siamese Networks. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[176] X. Lu, W. Wang, J. Shen, Y. Tai, D. Crandall, and S.C.H. Hoi. Learning Video Object Segmentation from Unlabeled Videos. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[177] D. C. Luvizon, D. Picard, and H. Tabia. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[178] A. M.-Gonzalez and M. A. J.-M.Odobez. Pose Transformers (POTR): Human Motion Prediction with Non-Autoregressive Transformers. In *International Conference on Computer Vision*, 2021.

[179] S. Ma, T. Simon, J. Saragih, D. Wang, Y. Li, F. D. L. Torre, and Y. Sheikh. Pixel Codec Avatars. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[180] L.v.d. Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[181] W. Mao, M. Liu, and M. Salzmann. History Repeats Itself: Human Motion Prediction via Motion Attention. In *European Conference on Computer Vision*, 2020.

[182] W. Mao, M. Liu, and M. Salzmann. Generating Smooth Pose Sequences for Diverse Human Motion Prediction. In *International Conference on Computer Vision*, 2021.

[183] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning Trajectory Dependencies for Human Motion Prediction. In *International Conference on Computer Vision*, 2019.

[184] W. Mao, M. Liu, M. Salzmann, and H. Li. Multi-Level Motion Attention for Human Motion Prediction. *International Journal of Computer Vision*, pages 1–23, 2021.

[185] J. Martinez, M.J. Black, and J. Romero. On Human Motion Prediction Using Recurrent Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[186] J. Martinez, R. Hossain, J. Romero, and J.J. Little. A Simple Yet Effective Baseline for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2017.

[187] R. Mehran, A. Oyama, and M. Shah. Abnormal Crowd Behavior Detection Using Social Force Model. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[188] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. In

International Conference on 3D Vision, 2017.

[189] D. Mehta, S. S., O. Sotnychenko, H. Rhodin, M. Shafiei, H. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-Time 3D Human Pose Estimation with a Single RGB Camera. In *ACM SIGGRAPH*, 2017.

[190] G. Moon, J. Y. Chang, and K. M. Lee. V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[191] F. Moreno-Noguer. 3D Human Pose Estimation from a Single Image via Distance Matrix Regression. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[192] G. Mori and J. Malik. Estimating Human Body Configurations Using Shape Context Matching. In *European Conference on Computer Vision*, 2002.

[193] G. Mori and J. Malik. Recovering 3D Human Body Configurations Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[194] A. Newell, K. Yang, and J. Deng. Stacked Hourglass Networks for Human Pose Estimation. In *European Conference on Computer Vision*, 2016.

[195] E. Ng, D. Xiang, H. Joo, and K. Grauman. You2me: Inferring Body Pose in Egocentric Video via First and Second Person Interactions. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[196] B.X. Nie, P. Wei, and S.-C. Zhu. Monocular 3D Human Pose Estimation by Predicting Depth on Joints. In *International Conference on Computer Vision*, 2017.

[197] M. Noroozi and P. Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision*, pages 69–84, 2016.

[198] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting Self-Supervised Learning via Knowledge Transfer. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[199] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a Feedback Loop for Hand Pose Estimation. In *International Conference on Computer Vision*, 2015.

[200] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele. Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation. In *International Conference on 3D Vision*, 2018.

[201] A. Papazoglou and V. Ferrari. Fast Object Segmentation in Unconstrained Video. In *International Conference on Computer Vision*, pages 1777–1784, 2013.

[202] S. Park, J. Hwang, and N. Kwak. 3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information. In *European Conference on Computer Vision*, 2016.

[203] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning Features by Watching Objects Move. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[204] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[205] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal Depth Supervision for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[206] G. Pavlakos, X. Zhou, K. Derpanis, G. Konstantinos, and K. Daniilidis. Coarse-To-Fine

Volumetric Prediction for Single-Image 3D Human Pose. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[207] G. Pavlakos, X. Zhou, K. Derpanis. G. Konstantinos, and D. Kostas. Harvesting Multiple Views for Marker-Less 3D Human Pose Annotations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[208] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[209] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-Supervised Training. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[210] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In *International Conference on Computer Vision*, 2009.

[211] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[212] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[213] F. Perazzi, O. Wang, M. Gross, and A. S.-Hornung. Fully Connected Object Proposals for Video Segmentation. In *International Conference on Computer Vision*, 2015.

[214] T. Pfister, J. Charles, and A. Zisserman. Flowing Convnets for Human Pose Estimation in Videos. In *International Conference on Computer Vision*, 2015.

[215] P.O. Pinheiro and R. Collobert. Recurrent Neural Networks for Scene Labelling. In *International Conference on Machine Learning*, 2014.

[216] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[217] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-Time Identification and Localization of Body Parts from Depth Images. In *International Conference on Robotics and Automation*, 2010.

[218] G. Pons-Moll, J. Taylor, J. Shotton, A. Hertzmann, and A. Fitzgibbon. Metric Regression Forests for Correspondence Estimation. *International Journal of Computer Vision*, 2015.

[219] A-.I. Popa, M. Zanfir, and C. Sminchisescu. Deep Multitask Architecture for Integrated 2D and 3D Human Sensing. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[220] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng. Cross View Fusion for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2019.

[221] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. In *European Conference on Computer Vision*, 2012.

[222] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Conference on Computer Vision and Pattern Recognition*,

2016.

[223] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang. Lightweight Multi-View 3D Pose Estimation through Camera-Disentangled Representation. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[224] H. Rhodin, V. Constantin, I. Katircioglu, M. Salzmann, and P. Fua. Neural Scene Decomposition for Human Motion Capture. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[225] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation. In *European Conference on Computer Vision*, 2018.

[226] H. Rhodin, J. Spoerri, I. Katircioglu, V. Constantin, F. Meyer, E. Moeller, M. Salzmann, and P. Fua. Learning Monocular 3D Human Pose Estimation from Multi-View Images. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[227] A. Richard, C. Lea, S. Ma, J. Gall, F. D. L. Torre, and Y. Sheikh. Audio- and Gaze-driven Facial Animation of Codec Avatars. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

[228] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *International Conference on Machine Learning*, 2011.

[229] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. In *European Conference on Computer Vision*, pages 549–565, 2016.

[230] G. Rogez and C. Schmid. Mocap Guided Data Augmentation for 3D Pose Estimation in the Wild. In *Advances in Neural Information Processing Systems*, 2016.

[231] A. Rohan, M. Rabah, T. Hosny, and S.-H. Kim. Human Pose Estimation-Based Real-Time Gait Analysis Using Convolutional Neural Network. *IEEE Access*, 8:191542–191550, 2020.

[232] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241, 2015.

[233] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2001.

[234] A. H. Ruiz, J. Gall, and F. Moreno-Noguer. Human Motion Prediction via Spatio-Temporal Inpainting. In *International Conference on Computer Vision*, 2019.

[235] C. Russell, R. Yu, and L. Agapito. Video Pop-Up: Monocular 3D Reconstruction of Dynamic Scenes. In *European Conference on Computer Vision*, 2014.

[236] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, S. H. Rezatofighi, and S. Savarese. Sophie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[237] M. Salzmann and R. Urtasun. Implicitly Constrained Gaussian Process Regression for Monocular Non-Rigid Pose Estimation. In *Advances in Neural Information Processing Systems*, December 2010.

[238] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In *European Conference on Computer Vision*, 2020.

[239] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A Simple Neural Network Module for Relational Reasoning. In *Advances in Neural Information Processing Systems*, 2017.

[240] M. Sanzari, V. Ntouskos, and F. Pirri. Bayesian Image Based 3D Pose Estimation. In *European Conference on Computer Vision*, 2016.

[241] H. Sarbolandi, D. Lefloch, and A. Kolb. Kinect Range Sensing: Structured-Light Versus Time-Of-Flight Kinect. *Computer Vision and Image Understanding*, 139:1–20, 2015.

[242] A. Sengupta, I.Budvytis, and R. Cipolla. Probabilistic 3D Human Shape and Pose Estimation from Multiple Unconstrained Images in the Wild. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[243] S. Seo, J.-Y. Lee, and B. Han. URVOS: Unified Referring Video Object Segmentation Network with a Large-Scale Benchmark. In *European Conference on Computer Vision*, 2020.

[244] H. Seong, J. Hyun, and E. Kim. Kernelized Memory Network for Video Object Segmentation. In *European Conference on Computer Vision*, 2020.

[245] N. Shafiee, T. Padir, and E. Elhamifar. Introvert: Human Trajectory Prediction via Conditional 3D Attention. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[246] A. M. Sharma, K. S. Venkatesh, and A. Mukerjee. Human Pose Estimation in Surveillance Videos Using Temporal Continuity on Static Pose. In *International Conference on Image Information Processing*, 2011.

[247] J. Shotton, A. Fitzgibbon, M. Cook, and A. Blake. Real-Time Human Pose Recognition in Parts from a Single Depth Image. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[248] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient Human Pose Estimation from Single Depth Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013.

[249] X. Shu, L. Zhang, G.-J. Qi, W.Liu, and J. Tang. Spatiotemporal Co-Attention Recurrent Neural Networks for Human-Skeleton Motion Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[250] L. Sigal and M.J. Black. Humaneva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. Technical report, Department of Computer Science, Brown University, 2006.

[251] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A Joint Model for 2D and 3D Pose Estimation from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[252] Snapchat Lens Studio. https://lensstudio.snapchat.com.

[253] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam. Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection. In *European Conference on Computer Vision*, 2018.

[254] O. Stretcu and M. Leordeanu. Multiple Frames Matching for Object Discovery in Video. In *British Machine Vision Conference*, 2015.

[255] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid. Actor-

Centric Relation Network. In *European Conference on Computer Vision*, 2018.

[256] C. Sun, A. Shrivastava, C. Vondrick, R. Sukthankar, K. Murphy, and C. Schmid. "relational Action Forecasting. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[257] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[258] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional Human Pose Regression. In *International Conference on Computer Vision*, 2017.

[259] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral Human Pose Regression. In *European Conference on Computer Vision*, 2018.

[260] I. Sutskever, J. Martens, and G. Hinton. Generating Text with Recurrent Neural Networks. In *International Conference on Machine Learning*, 2011.

[261] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, 2014.

[262] Y. Tang, L. Ma, W. Liu, and W.Zheng. Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamic. In *International Joint Conference on Artificial Intelligence*, 2018.

[263] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling Human Motion Using Binary Latent Variables. In *Advances in Neural Information Processing Systems*, 2006.

[264] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured Prediction of 3D Human Pose with Deep Neural Networks. In *British Machine Vision Conference*, 2016.

[265] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *International Conference on Computer Vision*, 2017.

[266] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct Prediction of 3D Body Poses from Motion Compensated Sequences. In *Conference on Computer Vision and Pattern Recognition*, pages 991–1000, 2016.

[267] P. Tokmakov, K. Alahari, and C. Schmid. Learning Motion Patterns in Videos. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[268] P. Tokmakov, K. Alahari, and C. Schmid. Learning Video Object Segmentation with Visual Memory. In *International Conference on Computer Vision*, 2017.

[269] D. Tome, T. Alldieck, P. Peluse, G. Pons-Moll, L. Agapito, H. Badino, and F. De la Torre. SelfPose: 3D Egocentric Pose Estimation from a Headset Mounted Camera. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[270] D. Tome, P. Peluse, L. Agapito, and H. Badino. xR-EgoPose: Egocentric 3D Human Pose from an HMD Camera. In *International Conference on Computer Vision*, 2019.

[271] D. Tome, C. Russell, and L. Agapito. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[272] D. Tome, M. Toso, L. Agapito, and C. Russell. Rethinking Pose in 3D: Multi-stage Refinement and Recovery for Markerless Motion Capture. In *International Conference on 3D Vision*, 2018.

[273] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint Training of a Convolutional Network

and a Graphical Model for Human Pose Estimation. In *Advances in Neural Information Processing Systems*, 2014.

[274] A. Toshev and C. Szegedy. Deeppose: Human Pose Estimation via Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[275] H. Tu, C. Wang, and W.-J. Zeng. VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment. In *European Conference on Computer Vision*, 2020.

[276] H.-Y. Tung, A.W. Harley, W. Seto, and K. Fragkiadaki. Adversarial Inverse Graphics Networks: Learning 2D-To-3D Lifting and Image-To-Image Translation from Unpaired Supervision. In *International Conference on Computer Vision*, 2017.

[277] R. Urtasun, D. Fleet, and P. Fua. 3D People Tracking with Gaussian Process Dynamical Models. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[278] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *International Conference on Machine Learning*, 2008.

[279] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 2010.

[280] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll. Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera. In *European Conference on Computer Vision*, 2018.

[281] B. Wandt and B. Rosenhahn. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[282] B. Wang, E. Adeli, H.-K. Chiu, D.-A. Huang, and J. C. Niebles. Imitation Learning for Human Pose Prediction. In *International Conference on Computer Vision*, pages 7123–7132, 2019.

[283] C. Wang, J. Li, W. Liu, C. Qian, and C. Lu. HMOR: Hierarchical Multi-Person Ordinal Relations for Monocular Multi-Person 3D Pose Estimation. In *European Conference on Computer Vision*, 2020.

[284] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[285] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models. In *Advances in Neural Information Processing Systems*, 2005.

[286] J. Wang, K. Qiu, H. Peng, J. Fu, and J. Zhu. AI Coach: Deep Human Pose Estimation and Analysis for Personalized Athletic Training Assistance. In *ACM International Conference on Multimedia*, 2019.

[287] J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao. Deep 3D Human Pose Estimation: A Review. *Computer Vision and Image Understanding*, 210:103225, 2021.

[288] J. Wang, S. Yan, B. Dai, and D. Lin. Scene-Aware Generative Network for Human Motion Synthesis. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[289] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei. 3D Human Pose Machines with Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

42:1069–1082, 2020.

[290] W. Wang, J. Shen, and F. Porikli. Saliency-Aware Geodesic Video Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[291] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C. Hoi, and H. Ling. Learning Unsupervised Video Object Segmentation through Visual Attention. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[292] X. Wang, A. Jabri, and A.A. Efros. Learning Correspondence from the Cycle-Consistency of Time. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[293] Y. Wang, Y. Liu, X. Tong, Q. Dai, and P. Tan. Outdoor Markerless Motion Capture with Sparse Handheld Video Cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24:1856–1866, 2018.

[294] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt. Probabilistic Monocular 3D Human Pose Estimation with Normalizing Flows. In *International Conference on Computer Vision*, 2021.

[295] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[296] X.-S. Wei, C.-L. Zhang, J. Wu, C. Shen, and Z.-H. Zhou. Unsupervised Object Discovery and Co-Localization by Deep Descriptor Transforming. In *arXiv Preprint*, 2017.

[297] D. Weinland, M. Ozuysal, and P. Fua. Making Action Recognition Robust to Occlusions and Viewpoint Changes. In *European Conference on Computer Vision*, pages 635–648, 2010.

[298] R.J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 1992.

[299] Z. Wu, Y. Xiong, X. Y. Stella, and D. Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[300] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *Proceedings of Machine Learning Research*, 37:2048–2057, 2015.

[301] Y. Xu, S.-C. Zhu, and T. Tung. DenseRaC: Joint 3D Pose and Shape Estimation by Dense Render-and-Compare. In *International Conference on Computer Vision*, 2019.

[302] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who Are You with and Where Are You Going? In *Conference on Computer Vision and Pattern Recognition*, 2011.

[303] X. Yan, A. Rastogi, R. Villegas, K. Sunkavalli, E. Shechtman, S. Hadap, E. Yumer, and H. Lee. MT-VAE: Learning Motion Transformations to Generate Multimodal Human Dynamics. In *European Conference on Computer Vision*, 2018.

[304] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3D Human Pose Estimation in the Wild by Adversarial Learning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[305] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised Moving Object Detection via Contextual Information Separation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[306] Y. Yang and D. Ramanan. Articulated Pose Estimation with Flexible Mixtures-Of-Parts.

In *Conference on Computer Vision and Pattern Recognition*, 2011.

[307] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. S. Torr. Anchor Diffusion for Unsupervised Video Object Segmentation. In *International Conference on Computer Vision*, 2019.

[308] Z. Yang, Y. Wei, and Y. Yang. Collaborative Video Object Segmentation by Foreground-Background Integration. In *European Conference on Computer Vision*, 2020.

[309] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou. Video Object Segmentation and Tracking: A Survey. *ACM Transactions on Intelligent Systems and Technology*, 11:1–47, 2020.

[310] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. A Dual-Source Approach for 3D Pose Estimation from a Single Image. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[311] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D Pose Estimation from a Single Depth Image. In *International Conference on Computer Vision*, 2011.

[312] M. Ye and R. Yang. Real-time Simultaneous Pose and Shape Estimation for Articulated Objects Using a Single Depth Camera. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[313] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative Image Inpainting with Contextual Attention. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[314] Y. Yuan and K. Kitani. Dlow: Diversifying Latent Flows for Diverse Human Motion Prediction. In *European Conference on Computer Vision*, 2020.

[315] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-Tracker: Global Multi-Object Tracking Using Generalized Minimum Clique Graphs. In *European Conference on Computer Vision*, pages 343–356, 2012.

[316] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[317] F. Z. Zhang, D. Campbell, and S. Gould. Spatially Conditioned Graphs for Detecting Human-Object Interactions. In *International Conference on Computer Vision*, 2021.

[318] L. Zhang, J. Zhang, Z. Lin, R. Měch, H. Lu, and Y. He. Unsupervised Video Object Segmentation with Joint Hotspot Tracking. In *European Conference on Computer Vision*, 2020.

[319] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *European Conference on Computer Vision*, 2016.

[320] Z. Zhang, C. Wang, W. Qin, and W. Zeng. Fusing Wearable IMUs with Multi-View Images for Human Pose Estimation: A Geometric Approach. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[321] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D.N. Metaxas. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[322] M. Zhen, S. Li, L. Zhou, J. Shang, H. Feng, T. Fang, and L. Quan. Learning Discriminative Feature with CRF for Unsupervised Video Object Segmentation. In *European Conference on Computer Vision*, 2020.

[323] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu. HEMlets Pose: Learning Part-Centric

Heatmap Triplets for Accurate 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2019.

[324] T. Zhou, J. Li, X. Li, and L. Shao. Target-Aware Object Discovery and Association for Unsupervised Video Multi-Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[325] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach. In *International Conference on Computer Vision*, 2017.

[326] X. Zhou, A. S. Liu, A. G. Pavlakos, A. V. Kumar, and K. Daniilidis. Human Motion Capture Using a Drone. In *International Conference on Robotics and Automation*, 2018.

[327] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep Kinematic Pose Regression. In *European Conference on Computer Vision*, 2016.

[328] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis. Sparseness Meets Deepness: 3D Human Pose Estimation from Monocular Video. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[329] D. Zou and P. Tan. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):354–366, 2013.

[330] Z. Zou and W. Tang. Modulated Graph Convolutional Network for 3D Human Pose Estimation. In *International Conference on Computer Vision*, 2021.

ISINSU KATIRCIOGLU

(+41) 78 808 6113 ⋄ Lausanne, Switzerland
isinsu.katircioglu@gmail.com ⋄ linkedin.com/isinsu-katircioglu

RESEARCH INTERESTS

Computer Vision, Machine Learning, Deep Learning, Self-supervised Learning, Human Pose Estimation, Image Segmentation, 2D/3D Object Detection, Inpainting

EDUCATION

Ph.D. in Computer Science, École Polytechnique Fédérale de Lausanne (EPFL), 2016 - 2021
Computer Vision Laboratory (CVLAB)
Dissertation Topic: Encoder-decoder Networks for Human Segmentation and Motion Analysis
Thesis directors: Dr. Mathieu Salzmann, Prof. Pascal Fua

M.Sc. in Computer Science, École Polytechnique Fédérale de Lausanne (EPFL), 2014 - 2016
CGPA: 5.32/6
Master Thesis: Structured Prediction of 3D Human Pose with Deep Neural Networks
Thesis director: Prof. Pascal Fua

B.Sc. in Computer Science, Middle East Technical University (METU), 2010 - 2014
CGPA: 3.91/4.0

PUBLICATIONS

- **I. Katircioglu**, H. Rhodin, J. Spörri, M. Salzmann, P. Fua. Human Detection and Segmentation via Multi-view Consensus. In ICCV, 2021.
- **I. Katircioglu**, H. Rhodin, V. Constantin, J. Spörri, M. Salzmann, P. Fua. Self-supervised Segmentation via Background Inpainting. In TPAMI, 2021.
- H. Rhodin, V. Constantin, **I. Katircioglu**, M. Salzmann, and P. Fua. Neural Scene Decomposition for Multi-Person Motion Capture. In CVPR, 2019.
- H. Rhodin, J. Spörri, **I. Katircioglu**, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua. Learning Monocular 3D Human Pose Estimation from Multi-view Images. In CVPR, 2018.
- **I. Katircioglu***, B. Tekin*, M. Salzmann, V. Lepetit, P. Fua. Learning Latent Representations of 3D Human Pose with Deep Neural Networks. International Journal of Computer Vision (IJCV). 126(12): 1326-1341 (2018).
- B. Tekin*, **I. Katircioglu***, M. Salzmann, V. Lepetit, and P. Fua. Structured Prediction of 3D Human Pose with Deep Neural Networks. In BMVC, 2016.

PROJECTS

Self-supervised Segmentation via Background Inpainting

- Proposed a self-supervised detection and segmentation approach that relies on importance sampling based optimization, new foreground and background objectives, and their joint training on unlabeled images captured by static, rotating and handheld cameras.

3D Human Pose Estimation Based on Person-to-Person Interactions

- Worked on recurrent neural network and transformer based models that exploit the motion dependencies between the past poses of a group of interacting people to predict their future poses. Acquired multi-view boxing and lindy hop video datasets with 2D and 3D body poses to evaluate our method.

3D Pose Based Motion Correction for Physical Exercises

- Worked on a deep learning based motion correction model to analyze and identify the common mistakes in yoga exercises for personal training. Gathered a yoga dataset containing videos, 2D and 3D body poses of semi-professional and amateur yoga practitioners.

Structured 3D Human Pose Estimation

- Introduced a deep learning regression model for structured prediction of 3D human pose from monocular images that relies on an overcomplete auto-encoder to learn a high-dimensional latent pose representation and account for joint dependencies.

SKILLS

Programming	Python, C/C++, Java, Matlab
Frameworks	PyTorch, TensorFlow, CUDA, OpenCV, Kubernetes, Git, SVN

WORK EXPERIENCE

Software Engineer Intern June - Sep 2015
NVISO *Lausanne, Switzerland*

- Developed a hyperparameter optimization framework for facial landmark detection using Spearmint Bayesian optimization library. Setup multi-node training with Redis and MongoDB.

Software Engineer Feb - Apr 2014
ASELSAN *Ankara, Turkey*

- Developed a framework in C++ for the communication of the infrared seeker head and target tracking missile. Processed the image data obtained from the optical system of the infrared seeker head and used it for the guidance of the missile.

Software Engineer Intern Jul - Aug 2013
ASELSAN *Ankara, Turkey*

- Built a framework for automatic target recognition and tracking from satellite images using SIFT and HOG features. Worked on multi-view geometry for finding corresponding points between two cameras.

Software Engineer Intern Aug - Sep 2012
Media & Medical Information Technology Solutions *Ankara, Turkey*

- Developed a CRM web application for the use of doctors and pharmaceutical companies. Implemented the framework in C# using .NET.

TEACHING & SERVICES

- Reviewer in ECCV 2020 and ICCV 2021.
- Mentorship in GirlsCoding Lausanne (2018).
- Supervised master's thesis of Costa Georgantas and Hugues Vinzant.
- Teaching assistant in Computer Vision (2018-2020), Information, Computation and Communication (ICC) Programming (2017-2020) and Linear Algebra (2020) courses.

HONOURS & AWARDS

- EDIC Doctoral Fellowship (École Polytechnique Fédérale de Lausanne, Sep 2016 - Aug 2017)
- Excellence Masters Fellowship (École Polytechnique Fédérale de Lausanne, Sep 2014 - Feb 2016)
- Computer Engineering Department Salutatorian (Middle East Technical University, June 2014)

LANGUAGES

- English (Fluent)
- French (Intermediate)
- Turkish (Native)