Representation of the Semantic Structures: from Discovery to Applications

Thesis by

Serim Ryou

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2022
Defended October 18, 2021

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Pietro Perona for giving me the guidance and the opportunity to fully explore my research interests. I also would like to thank my thesis committee members, Yaser Abu-Mostafa, Victoria Kostina, and Katie Bouman, for giving me valuable academic advice and taking the time to provide feedback during my Ph.D. study. Also, I would like to thank Sarah Reisman and Yisong Yue for giving me a wonderful collaboration experience and broadening my perspective on an interdisciplinary research field.

I deeply appreciate all the Computational Vision lab members, Steve Branson, Bo Chen, Ron Appel, Eyrun Eyjolfsdottir, Krzysztof Chalupka, David Hall, Matteo Ruggero Ronchi, Mason McGill, Grant Van Horn, Joseph Marino, Alvita Tran, Oisin Mac Aodha, Cristina Segalin, Tony Zhang, Sara Beery, Eli Cole, Jennifer Sun, and Neehar Kondapaneni. My entire Ph.D. would have been impossible without your help. You guys were always a great source of motivation for me to become both a good researcher and a friendly colleague. Thank you so much for giving me memorable moments. I also would like to thank my collaborators, Michael Maser, Travis DeLano, and Alex Cui. The random ideas and discussion you brought were always very enjoyable.

I would like to thank all my Korean friends. Special thanks to Gahye Jeong, Jaebum Chung, Haemin Paik, Areum Kim, Jieun Shin, Hyeongchan Jo, Seoyoung Kim, Sanghyun Yi, and Jihong Min, for bringing me out of home and giving me emotional support during COVID. My old friends in Korea, Sukyoung, Dohee, and Joenghwa, thank you for bearing me with whining about all the difficulties I faced while I was working on my dissertation. Also, Suwon, hope you stay well and bright in peace.

Lastly, I would like to express deep gratitude to my family. My lovely nephew, Jeonghyuk, I've got so much energy from you since you were born. Above all things, I always appreciate the endless and unconditional love I got from my family. Thank you so much for always trusting me wherever I go and supporting whatever I do.

ABSTRACT

The world surrounding us is full of structured entities. Scenes can be structured as the sum of objects arranged in space, objects can be decomposed into parts, and even small molecules are composed of atoms. As humans can organize and structure many concepts into smaller components, structural representation has become a powerful tool for various applications. Computer vision utilizes the part-based representation for classical object detection and categorization tasks, and computational neuroscientists use the structural representation to achieve an interpretable and low-dimensional encoding for behavior analysis. Furthermore, structural encoding of the molecules allows the application of machine learning models to optimize experimental reaction conditions in organic chemistry.

To perform the high-level tasks described above, accurate detection of the structural component should be accomplished in advance. In this dissertation, we first propose methods to improve the pose estimation algorithm, where the task is to localize the semantic parts of the target instance from a 2D image. As the collection of a large number of human annotations is a prerequisite for the task to be successful, we aim to design a model that automatically discovers the structure information from the visual inputs without supervision. Lastly, we demonstrate the efficacy of the structural representation by applying it to various scientific applications such as behavior analysis and organic chemistry.

PUBLISHED CONTENT AND CONTRIBUTIONS

Maser*, Michael R., Alexander Y. Cui*, Serim Ryou*, Travis J. DeLano, Yisong Yue, and Sarah E. Reisman (2021). "Multilabel Classification Models for the Prediction of Cross-Coupling Reaction Conditions". In: *Journal of Chemical Information and Modeling* 61.1, pp. 156–166. DOI: 10.1021/acs.jcim.0c01234.
S.R. participated in developing the method and running the experiments.

Ryou, Serim and Pietro Perona (2021). "Weakly Supervised Keypoint Discovery". In: arXiv: 2109.13423 [cs.CV]. URL: https://arxiv.org/abs/2109.13423.
S.R. participated in designing the project, developing the method, running the experiments, and writing the manuscript.

Sun*, Jennifer J., Serim Ryou*, Roni Goldshmid, Brandon Weissbourd, John Dabiri, David J. Anderson, Ann Kennedy, Yisong Yue, and Pietro Perona (2021). "Self-Supervised Keypoint Discovery in Behavioral Videos". In: arXiv: 2112.05121 [cs.CV]. URL: https://arxiv.org/abs/2112.05121.
S.R. participated in designing the project, developing the method, running the experiments, and writing the manuscript.

Ryou*, Serim, Michael R. Maser*, Alexander Y. Cui*, Travis J. DeLano, Yisong Yue, and Sarah E. Reisman (2020). "Graph Neural Networks for the Prediction of Substrate-Specific Organic Reaction Conditions". en. In: *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRLB)*. URL: http://arxiv.org/abs/2007.04275.
S.R. participated in developing the method and running the experiments.

Ryou, Serim, Seong-Gyun Jeong, and Pietro Perona (2019). "Anchor Loss: Modulating Loss Scale Based on Prediction Difficulty". In: *The IEEE International Conference on Computer Vision (ICCV)*. DOI: doi.org/10.1109/ICCV.2019.00609.
S.R. participated in designing the project, developing the method, running the experiments, and writing the manuscript.

Ryou, Serim and Pietro Perona (2018). "Parsing Pose of People with Interaction". In: *British Machine Vision Conference (BMVC)*. URL: http://bmvc2018.org/contents/papers/0679.pdf.
S.R. participated in designing the project, developing the method, running the experiments, and writing the manuscript.

TABLE OF CONTENTS

Acknowledgements iii
Abstract iv
Published Content and Contributions v
Table of Contents v
List of Illustrations viii
List of Tables xvii
Chapter I: Introduction 1
Chapter II: Pose Estimation 7
2.1 Background 7
2.2 Introduction 8
2.3 Related Work 11
2.4 Method 13
2.5 Experiments 18
2.6 Conclusions 25
2.7 Appendix: Anchor Design 25
2.8 Appendix: Qualitative Results 27
Chapter III: Multi-Person Pose Estimation: Parsing Pose of People with Interaction 33
3.1 Abstract 33
3.2 Introduction 33
3.3 Related Work 35
3.4 COCO-Crowd dataset 37
3.5 Method 39
3.6 Experimental Results 41
3.7 Conclusions 45
Chapter IV: Weakly Supervised Keypoint Discovery 49
4.1 Abstract 49
4.2 Introduction 49
4.3 Related Work 52
4.4 Method 54
4.5 Experiments 59
4.6 Conclusions 67
4.7 Appendix: Qualitative Results 70
4.8 Appendix: Architecture Details 71
Chapter V: Neuroscience: Self-Supervised Keypoint Discovery in Behavioral Videos 81
5.1 Abstract 81
5.2 Introduction 81
5.3 Related work 84

5.4 Method 86
5.5 Experiments 90
5.6 Discussion and conclusion 98
5.7 Appendix: Additional Experimental Results 99
5.8 Appendix: Additional Implementation Details 105
5.9 Appendix: Visualizations 108
Chapter VI: Chemistry: Multi-Label Classification Models for the Prediction of Cross-Coupling Reaction Conditions 117
6.1 Abstract 117
6.2 Introduction 117
6.3 Method 119
6.4 Results and discussion 127
6.5 Conclusions 135
Chapter VII: Conclusions 143

LIST OF ILLUSTRATIONS

Number *Page*

2.1 The human pose is encoded as 2D gaussian heatmaps. Each heatmap represents the location of the target part. 7

2.2 General pipeline for the pose estimation. Given an input image, a fully convolutional neural network predicts heatmaps and these heatmaps are compared to the ground truth by mean squared error. 8

2.3 The overview of anchor loss. A network is confused about left-and-right body parts due to the symmetrical appearance of the human body, and struggles to disambiguate visually similar objects. Although the network output scores on the correct labels are relatively high, the final prediction is always chosen by the index of the highest score, resulting in a wrong prediction. Our loss function is designed to resolve this issue by penalizing more than cross entropy when the non-target (background) probability is higher than the anchor probability. 9

2.4 How the anchor probability q_* affects our loss function compared to standard cross entropy (CE) and focal loss (FL). While FL always depresses the loss values for the samples producing trivial outcomes, anchor loss dynamically re-scales its loss values based on the relative difficulties of the target and the anchor probability. For these plots, the anchor probability is chosen as the prediction score ($q_* = q_{C_1}$) on the true positive label (C_1). Thus, if the networks produce higher score on the background label compared to the anchor, our loss encourages the network to correct the relative order of the predictions by penalizing more than the cross entropy. 10

2.5 How an anchor probability is chosen for the pose estimation task. For the target body part of right shoulder (b), the maximum confidence score inside the solid red circle becomes an anchor probability to modulate the loss values in mask areas (c). 15

2.6 Gradient figure: sample gradient output of background probability distribution. Compared to the cross entropy, the magnitude of gradient increases when the prediction is higher than the anchor probability. 18

2.7 Validation curves of ResNet-110 on CIFAR-100 dataset. We compare our loss function to CE. 20

2.8 Validation curves of 2-stacked Hourglass on MPII dataset. We compare our loss function to BCE. 20

2.9 Anchor loss visualization on pose estimation. We visualize where anchor loss assigns higher loss values than the binary cross entropy and how it changes over training epochs. At the beginning, visually similar parts often get higher scores than the target body part, thus our loss function assigns higher weights on those pixel locations. Once the model is able to detect the target body part with high confidence, loss is down-weighted for most of the areas, so that the network can focus on finding more accurate location for the target body part. . . . 22

2.10 Qualitative results on human pose. The first row compares with the result from MSE loss (left) and our loss (right), and the second row contains some sample outputs. Model trained with the proposed loss function is robust at predicting symmetric body parts. 24

2.11 Double counting problem. We analyze how the anchor loss behaves when a double-counting problem occurs. 24

2.12 How an anchor probability modulates loss values. When the prediction score of target class is lower than $q_{pos} = 0.2$, anchor loss penalizes more than binary cross entropy (a). On the contrary, when the prediction score of background class is higher than $q_{neg} = 0.8$, the loss value becomes higher than the binary cross entropy (b). . . . 26

2.13 Qualitative results for human pose estimation. Top row shows the output images with baseline (MSE) and bottom row represents the outcomes with anchor loss. 27

2.14 Failure cases on human pose estimation. Network trained with anchor loss still fails to detect correct body part locations when the body part is blurred or self-occluded. 27

2.15 Image classification results on CIFAR-100. We compare the top-2 prediction scores of ResNet-110 with cross entropy (CE) and anchor loss (AL). Network trained with anchor loss successfully classifies difficult examples even though the model trained with cross entropy fails. 28

3.1 An overview of our system. For an input image (a), part localizer (Section 3.5) produces keypoint heatmaps without identity (b). Person decoding module (Section 3.5) sequentially produces instance keypoint heatmaps with distinct person identities. The network ends the prediction with generating the all-zero heatmap which represents no more instances on the scene (c). 34

3.2 Crowd Extraction Procedure. The left image (a) shows an original COCO image and annotations, and the two images on the right (b) show the corresponding images in COCO-crowd. We define the notion of "overlap" when the intersection of union (IoU) score of two bounding boxes is greater than 0.1 ([1,2], [3,4], and [4,5] pairs on left image). All interlinked boxes ([1,2] and [3,4,5]) are merged into proposals for the crowd region. Meanwhile, we discard non-crowd regions, box without an overlap. 36

3.3 Dataset configuration. After running our crowd extraction system on COCO, we observe that single-instances are dominant on the dataset (a). To overcome this dataset bias, our dataset (named COCO-crowd) consists of images with two or more instances, where its distribution can be seen in (b). To show the complexity of the our curated dataset, we count the number of overlaps arisen from each instance, and provide the distribution in (c). 38

3.4 Network architecture. The network consists of two parts: part localizer (each blue box representing ResNet-50 convolution block) and person decoder (green boxes corresponding to ConvLSTM block at each resolution). The input image is encoded with part localizer and first predicts part heatmaps. The person decoder decouples this encoded feature into distinct instance heatmaps. 38

3.5 Person Decoding Module. The green block represents two ConvLSTM layers at each resolution. The recurrent block on each scale is composed of two stacked ConvLSTMs. 40

3.6 Counting confusion matrix 43

3.7 Qualitative results. Results containing severe occlusion due to social interaction. 45

4.1 Overview of weakly supervised keypoint discovery: Our approach uses the unsupervised method to discover diverse keypoints and image-level supervision to localize the discriminative parts. By using the keypoints learned from weak-supervision to infer the viewpoint of a target instance, our model can successfully discover semantically consistent parts for instances facing in different directions. 50

4.2 System Outline: Our model uses the shared encoder for various tasks: image reconstruction, keypoint estimation, and classification. The bluish same block color represents the features generated from the shared weights of ResNet-50. Black, red, and orange arrows represent the appearance, geometry, and viewpoint stream, respectively. Target image I' is generated by applying structural transformation Tr to the source image I. The concatenation of the appearance feature from I and the geometry bottleneck from I' is used to reconstruct the target image $\hat{I}'$. Discovered keypoint heatmaps and the appearance features from I are used to generate part-based features (Sec. 4.4 and Fig. 4.3). 54

4.3 Weak-supervision module: We use the first K_w keypoints to extract part-based representation from the base features. Concatenated part-based features are fed to predict the image-level supervision task. . . 56

4.4 Spatial correlation between similar parts. Each row represents the keypoint bottleneck generated from the same keypoint channel. Ideally, heatmaps from the same row should represent the same semantic part. Keypoints discovered from the unsupervised module predict similar spatial locations for look-alike body parts regardless of the viewpoint of the animal. On the other hand, keypoints discovered from the weak-supervision module consistently find discriminative parts like the face. 57

4.5 Discovered keypoints on the AnimalPose dataset using existing methods. Unsupervised models either predict similar locations regardless of the semantic parts and viewpoint of an instance or fail to discover semantically consistent parts. 60

4.6 Qualitative results on various datasets. Our model successfully discovers semantically consistent parts for images with large viewpoint and appearance variations. 61

4.7 PCK score by finetuning the keypoint network with a different amount of supervision on CUB, AnimalPose, and StanfordExtra datasets. Despite the marginal performance differences after using 10% of supervision, the representation learned from our method gives better performance when there is an extremely limited amount of supervision. 62

4.8 Qualitative results for images with unseen categories 63

4.9 Confusion matrix and per-class accuracy for posture-based action prediction of TigDog and DogPart datasets. 64

4.10 Ground truth annotation and discovered keypoints for sitting and lying actions from TigDog dataset. 64

4.11 Qualitative result for loss ablation study. 65

4.12 Pose embedding for viewpoint-based equivariance 66

4.13 Appearance and geometry factorization: We generate the image using the scaled and flipped keypoint as a geometry bottleneck with the input image appearance feature. Given the predicted keypoints from the input, we reconstruct the images using the appearance feature from the top right images. 67

4.14 Failure cases: Our model fails to discover consistent keypoints for the rare poses and cannot handle the missing or occluded parts. The spatial correlation problem still remains for the images with large appearance variations. 67

4.15 Qualitative results on CelebA with 10 discovered keypoints. 68

4.16 Qualitative results on StanfordDogs with 24 discovered keypoints. . . 68

4.17 Qualitative results on AnimalPose with 20 discovered keypoints. . . . 69

4.18 Qualitative results on CUB with 15 discovered keypoints. 69

4.19 Qualitative results on TigDog and DogPart from the model trained on AnimalPose and StanfordDogs, respectively. 70

4.20 Qualitative results on unseen categories from the model trained on AnimalPose. 70

4.21 Appearance and geometry factorization on StanfordDogs dataset. . . 71

4.22 Appearance and geometry factorization on CUB dataset. 72

4.23 Layer description on lateral, upsample, and predict blocks. 72

4.24 Basic convolution block which is used for reconstruction module. . . 73

4.25 Reconstruction module. 74

4.26 Layer specification in the convolution block (conv_block_w) in weak supervision module. 74

5.1 Self-supervised Keypoint Discovery. Intermediate representations in the form of keypoints are frequently used for behavior analysis. We propose a method to discover keypoints from behavioral videos without the need for manual keypoint or bounding box annotations. Our method works across a range of organisms (including mice, humans, flies, jellyfish and tree), works with multiple agents simultaneously (see flies and mice above), does not require bounding boxes (boxes visualized above purely for identifying the enlarged regions of interest) and achieves state-of-the-art performance on downstream tasks. . . . 82

5.2 Our encoder-decoder approach for image difference reconstruction. Both frame I_t and frame I_{t+T} are fed to an appearance encoder Φ and a pose decoder Ψ. Given the appearance feature from I_t and geometry features from both I_t and I_{t+T} (Sec 5.4), our model reconstructs the spatiotemporal difference (Sec 5.4) computed from two frames using the reconstruction decoder ψ. 86

5.3 Behavior Classification Features. Extracting information from the raw heatmap (Section 5.4): the confidence scores and the covariance matrices are computed from normalized heatmaps. Note that the features are computed for all x, y coordinates. We visualize the zoomed area around the target instance for illustrative purposes. . . . 89

5.4 Comparison with existing methods (Jakab et al., 2018), full image, bounding box, and SSIM reconstruction (ours). "Jakab *et al*" and "full image" results are based on full image reconstruction. "White mouse bounding box" and "black mouse bounding box" show the results when the cropped bounding boxes were fed to the network for image reconstruction. 93

5.5 Keypoint data efficiency on MARS-Pose. The supervised model is based on (Segalin et al., 2020) using stacked hourglass (Newell, Yang, and Deng, 2016), while the semi-supervised model uses both our self-supervised loss and supervision. PCK is computed at $0.5cm$ threshold, averaged across nose, ears, and tail keypoints, over 3 runs. "b" and "w" indicates the black and white mouse respectively. . . . 95

5.6 Qualitative Results. Qualitative results of the keypoint discovery model trained on CalMS21 (mouse), Fly vs. Fly (fly), Human3.6M (human), jellyfish and Vegetation (tree). Additional visualizations are in the Supplementary materials. 97

5.7 Qualitative Results on CalMS21 by varying the number of keypoints. We train the keypoint discovery model with different numbers of discovered keypoints. Each row shows qualitative results with all the keypoints including the background ones. We note that there are 2 background (low-confidence) keypoints for 6 and 10 discovered keypoints, and 9 background keypoints for 20 discovered keypoints. . 100

5.8 Qualitative Results on CalMS21 for loss ablation study. With the full training objective for our discovered keypoints, we are able to track 8/10 keypoints consistently, while without rotation loss, there are only 5/10 tracked keypoints on both mice. Additionally, some of the discovered keypoints without rotation are not semantically consistent (for example, the pink and orange keypoints, two keypoints on the body of the white mouse, shift in order as the white mouse moves around). See quantitative results in Section 5.7. 102

5.9 Spectrogram from Distance of Discovered Keypoints. From a recorded video of jellyfish swimming at 48Hz, we discover keypoints at each frame using our model and compute a spectrogram based on the average distance between discovered keypoints on the jellyfish. 103

5.10 Wind Speed Regression from Discovered Keypoints. Mean wind speed, $\bar{U}$, vs. the fourth root of the sway amplitude equivalent measured from the standard deviation of the convex hull area of the 15 discovered keypoints in each clip, based on model from (J. L. Cardona and J. O. Dabiri, 2021). The scatter represents 10-mintute averages of the same data used for training the keypoint model. The black lines represent the best linear regression fit for the proportionality assumption. The proportionality coefficient and the R^2 values are presented in the legend. 104

5.11 Qualitative Results on CalMS21. We observe that keypoints are discovered for noses of both mice and generally along the spine of the mice. 109

5.12 Qualitative Results on Fly-vs-Fly. We observe that 3 keypoints are discovered on the body of the fly, with 2 on the wings (one for each wing). 109

5.13 Qualitative Results on Vegetations. Each row shows different frames with discovered keypoints from a single video. Our model can discover and track consistent keypoints within the same video. 109

5.14 Qualitative Results on Simplified Human 3.6M. We observe that keypoints are generally discovered on visible joints and end points of humans, such as head, elbows, hands, upper legs, knees and feet. We note that there is left/right swapping of body parts, since when the human is facing forwards or backwards, keypoints are generally on the same side. 110

5.15 Limitations. We visualize examples that are difficult for our model, for example from occlusion/agents being in close proximity (mouse, fly), self-occlusion (human), unusual poses (human, fly), and left-right swapping (human). 110

5.16 Confidence visualization on CalMS21. Confidence score (maximum prediction value) is shown with the normalized heatmap. Background keypoints (fourth on row 1 and second on row 2) have very low confidence. 111

6.1 Schematic modeling workflow. A) Data gathering. B) Tabulation and dictionary construction. C) Iterative model optimization. D) Inference and interpretation. 120

6.2 Samples of categorized reaction dictionaries for C-N and Negishi datasets. 121

6.3 Schematic modeling workflow. A) Tree-based methods. String and descriptor vectors for each molecule in a reaction are concatenated and used as inputs to gradient-boosting machines (GBMs). B) Deep learning methods. Molecular graphs are constructed for each molecule in a reaction, which are passed as inputs to a graph convolutional neural network (GCNN). Both model types predict probability rankings for the full reaction dictionary, which are sorted by reaction role and translated to the final output. 122

6.4 Average top-1 and top-3 categorical accuracies for each model across the four datasets. 128

6.5 Optimized prediction trellis for the Suzuki dataset. 131

6.6 AR-GCN attention weight visualization and prediction examples from randomly chosen reactions in each dataset. Darker highlighting indicates higher attention. 132

6.7 Performance dependence on reaction yield. A) Distribution of reaction yields for the four datasets. B) AR-GCN average top-1 A_c values for Suzuki predictions when trained and tested in different yield ranges (top) and dataset quartiles arranged by yield (bottom). . . 134

LIST OF TABLES

Number *Page*
2.1 Classification accuracy on CIFAR (ResNet-110) 19
2.2 Classification accuracies on ImageNet (ResNet-50) 19
2.3 Ablation studies on CIFAR-100 (ResNet-110) 21
2.4 PCK score on MPII dataset. The bottom rows show the performances of the methods built on top of hourglass network. The model trained with anchor loss shows comparative scores to the results from more complex models. 23
2.5 PCP score on LSP dataset. 23
2.6 Validation Results on MPII dataset. We report the validation score of the result using different losses with the same single-scale testing setup. 23
2.7 Hyperparameter search and comparison to other losses on MPII dataset with 2-stacked hourglass network. 23
2.8 Classification accuracies on CIFAR-100 with different anchor probabilities 27
3.1 Results (AP) on COCO-crowd. Mask-RCNN is tested using Detectron (Girshick et al., 2018) and all other methods are tested using the code and pretrained models the authors provide. Testing is held on single scale on all bottom-up methods. To see the impact of the amount of data, we also trained associative-embedding (Newell, Huang, and Deng, 2017) on COCO-crowd (AE*). 42
3.2 AP score by number of people 43
3.3 AP score by jittering bounding box 43
3.4 PCK score on Immediacy Dataset. 44
4.1 Keypoint estimation performance on the restricted setting. We train a linear regressor from the discovered keypoints for MAFL and CUB datasets. For CUB experiments, we follow the same data extraction step from the paper (Lorenz et al., 2019) and show the performance with %-MSE normalized by an edge length of the image. 62
4.2 Performance on weak-supervision task (fine-grained classification) with our baseline ResNet-50 (He et al., 2016) with an image size of 128x128 64
4.3 Architecture details about the keypoint module. 73

4.4 Architecture details about the reconstruction module. 74
4.5 Architecture details about the weak supervision module. 75
4.6 Hyperparameter settings for each dataset. 75
5.1 Behavior Classification Results on CalMS21. "Ours" represents classifiers using input keypoints from our discovered keypoints. "conf" represents using the confidence score, and "cov" represents values from the covariance matrix of the heatmap. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 runs are shown. 94
5.2 Behavior Classification Results on Fly. "FlyTracker" represents classifiers using hand-crafted inputs from (Eyjolfsdottir, S. Branson, et al., 2014). The self-supervised keypoints all use the same "generic features" computed on all keypoints: speed, acceleration, distance, and angle. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 runs are shown. 94
5.3 Comparison with state-of-the-art methods for landmark prediction on Simplified Human 3.6M. The error is in %-MSE normalized by image size. All methods predict 16 keypoints except for (Jakab et al., 2020)‡, which uses 32 keypoints for training a prior model from the Human 3.6M dataset. 96
5.4 Learning Objective Ablation, Simplified Human3.6M. %-MSE error is reported by changing the reconstruction target. 97
5.5 Hyperparameters, Simplified Human 3.6M. For frame gap experiments, the number of keypoints is set to 16. Frame gap is set to 20 for experiments with a varying number of keypoints. 97
5.6 Effect of Hyperparameters on CalMS21. For frame gap experiments, the number of keypoints is set to 10. Frame gap is set to 6 for experiments with a varying number of keypoints. All keypoints, confidence, and covariance are used as inputs, except (*) for the experiments with 20 keypoints, where only high-confidence keypoints are used (11 keypoints) since a high proportion of keypoints are discovered on the background. Mean and standard dev from 5 classifier runs are shown. 100

5.7 Effect of Varying Training Data Amount for Keypoint Discovery. We train the keypoint discovery model with different amounts of input training image pairs from video. Different training amounts are selected by choosing random video subsets from the full set of CalMS21 training videos. Image pairs are sampled from videos with a gap of 6 frames, and gap between pairs of 7 frames. All keypoints, confidence, and covariance values on 10 discovered keypoints are used. Mean and standard dev from 5 classifier runs are shown. 101

5.8 Loss Variations on CalMS21. "Ours" represents training with the full objective (reconstruction, rotation equivariance, separation) and "Reconstruction" indicates training with image difference reconstruction only. Mean and standard dev from 5 classifier runs are shown. . . . 101

5.9 Per-Class Behavior Classification Results on CalMS21. "Ours" represents classifiers using input keypoints from our discovered keypoints. "conf" represents using the confidence score, and "cov" represents values from the covariance matrix of the heatmap. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 classifier runs are shown. 103

5.10 Architecture details about the reconstruction decoder. "Conv_block" refers to a basic convolution block which is composed of 3×3 convolution, batch normalization, and ReLU activation. Note that output size for human experiments is downsampled by a factor of 2 for all the layers. 106

5.11 Hyperparameters for Keypoint Discovery. 108

6.1 Statistical summary of reaction datasets with Reaxys® queries. . . . 121

6.2 Prediction accuracy for all model types on the Suzuki dataset. 126

6.3 Prediction accuracy for all model types on the C–N, Negishi, and PKR datasets. 129

Chapter 1

INTRODUCTION

Human perceives the world by understanding the underlying structure of the surroundings. Scenes can be structured as a sum of objects arranged in space, objects can be decomposed into parts, animals can be parsed into anatomical body parts, and even small molecules are composed of atoms. This structural cognition also allows humans to easily decompose novel concepts into familiar pieces.

There is evidence in psychology (Palmer, 1977) and physiology (Wachsmuth, Oram, and Perrett, 1994) that human perception is based on the whole as a sum of parts. Early work on computational visual recognition conceptualizes understanding the objects into segments that comprise a set of basis (Biederman, 1987). From their theory, perceptual input is matched against a representation, which is composed of a set of primitives in the brain. On the other hand, Gestalt psychologists propose that human perception prioritizes the understanding of the whole rather than the sum of individual components (Koffka, 1935). Whether the human perception is based on the entirety in the context or by the sum of the parts, structural understanding plays a key role in perception and provides a powerful tool for designing engineering solutions.

The computer vision community has developed the part-based representation for classical object detection algorithms. Back in the 1970s, the pictorial structure model (Fischler and Elschlager, 1973) was introduced to provide a general statistical framework to recognize the objects in the image. Objects are represented as a sum of the components with a constraint that the attributes and the structural configuration between the parts should be preserved. By combining modern features and machine learning techniques, deformable part models (DPM) (Felzenszwalb, Mcallester, and Ramanan, 2008) emerged with the pictorial structure formulation and gained prominence for the tasks of object detection and categorization. Specifically, an object is represented as parts, which become the local appearance templates, and the springs, which encode spatial connections between the parts. DPM models have advantages in terms of computation and generalization capability; local appearance is shared across training data and flexible spatial connection enables recognizing unseen configurations.

Now we live in the era of deep learning. With the success of deep neural networks (Krizhevsky, Sutskever, and Hinton, 2012), various architectures (Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Huang et al., 2017; Tan and Le, 2019) have been proposed to generate a holistic representation of visual input. While this representation brings advances in many vision tasks, unfortunately, structural component is missing; complex information is encoded as a single high-dimensional representation. However, an in-depth understanding of the internal structure helps the system to go beyond the general classification task, especially when one has to disambiguate the marginal differences between the instances. For instance, given a pair of images with look-alike birds, one can easily identify whether the two birds are the same or not by comparing each body part and its attributes. This idea has been deployed to fine-grained classification (Branson et al., 2014), face identification (Xie, Shen, and Zisserman, 2018), and person re-identification (Zhao et al., 2017; Su et al., 2017) methods. In addition, the structural information enables the synthesis of an image with desired properties (*e.g.*, facial image generation with a specific facial expression) when used as a control input for generative models (Wang et al., 2019; Yang and Yao, 2019).

Recently, there has been an increasing effort in applying artificial intelligence (AI) to scientific problems. As the structural representation inherently has an interpretable formulation, scientific applications often rely on the structural encoding of the data as an intermediate representation. Computational neuroscientists use a pose of an animal as an input to the behavior analysis (Segalin et al., 2020), which further reduces the high dimensional video input to the low dimensional image coordinates. Chemists encode the molecules into a structural representation by using a graph-based model to perform molecular property prediction tasks such as quantum mechanical property prediction (Gilmer et al., 2017), physicochemical property prediction (Shang et al., 2018), and biological effects prediction (Xu, Pei, and Lai, 2017). Rather than treating the model as a black box, scientists can also analyze how each component of the structure contributes to the outcome.

In order to perform the high-level tasks discussed above, high accuracy for the low level-tasks such as estimating the semantic component of an instance must be established in advance. In this dissertation, we explore the structural representation on various domains ranging from computer vision, neuroscience to chemistry. As human perception largely depends on the visual input[1] , we mainly focus on advancing

[1] More than 50 percent of the cortex is devoted to visual perception.

the method for the pose estimation, which is a problem of estimating the location of semantic parts from the visual inputs (*e.g.*, image and video). We further relate the structural representation to various applications and tackle following questions:

1. How do we build a system that efficiently estimates a predefined structure from an image given enough supervisory signals? (Chapters 2 and 3)
2. What are the possible ways to learn more about the structural information with less data? Is it possible to discover the structure without human knowledge? (Chapters 4 and 5)
3. Does the structural representation itself serve as an efficient tool for various applications? (Chapters 5 and 6).

While this dissertation aims to answer the three questions above, each chapter also delivers its own independent contribution. An outline of thesis is as follows.

Chapter 2 addresses the problem of a single-person pose estimation. In this chapter, a novel loss function is proposed for training not only the keypoint estimation but also the general image classification tasks by modulating the loss scale based on the sample difficulty. The proposed loss function shows performance improvement over the standard loss functions for both image classification and pose estimation tasks.

Chapter 3 discusses the problem of multi-person pose estimation in the scenes where people are having interaction with each other. Novel network architecture is proposed to incorporate the interaction information between the instances by adopting a recurrent framework. The proposed architecture is robust at predicting the pose of people even when people are intertwined.

Chapter 4 seeks the limits of supervision for discovering the structure from an image with a weakly supervised learning approach. Image-level supervision is used to discover discriminative parts of the target object, and unsupervised learning is used to diversify the discovered keypoints. The proposed method shows consistent part discovery for images with large viewpoint and appearance variations.

Chapter 5 expands the keypoint discovery method with an emphasis on the behavior classification task for computational neuroscience experiments. From the observation of the stationary background for the videos taken from a neuroscience lab environment, we leverage the spatiotemporal difference reconstruction as an auxiliary task for discovering the keypoints. The raw discovered keypoints achieves comparable result to the behavior classification task when compared against the human annotations.

Chapter 6 explores the structural representation of molecules for chemistry applications. Among various types of molecule encoding schemes, a graph neural network, which solely encodes the structure of molecules without any chemical information, shows promising results for the task of predicting the substrate-specific cross-coupling reaction conditions for organic chemistry experiments.

Finally, Chapter 7 summarizes the dissertation and discusses the future work.

References

Biederman, I. (1987). "Recognition-by-components: a theory of human image understanding." In: *Psychological review* 94 2, pp. 115–147.

Branson, Steve, Grant Van Horn, Serge Belongie, and Pietro Perona (Sept. 1, 2014). "Bird Species Categorization Using Pose Normalized Deep Convolutional Nets". In: *British Machine Vision Conference (BMVC)*. Nottingham. URL: `http://vision.cornell.edu/se3/wp-content/uploads/2015/02/BMVC14.pdf`.

Felzenszwalb, P., D. Mcallester, and D. Ramanan (2008). "A Discriminatively Trained, Multiscale, Deformable Part Model". In: *Proc. IEEE CVPR*.

Fischler, M. A. and R. A. Elschlager (Jan. 1973). "The Representation and Matching of Pictorial Structures". In: *IEEE Trans. Comput.* 22.1, pp. 67–92. ISSN: 0018-9340.

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl (June 2017). "Neural Message Passing for Quantum Chemistry". en. In: *arXiv:1704.01212 [cs]*. URL: `http://arxiv.org/abs/1704.01212`.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proc. IEEE CVPR*.

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger (2017). "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. DOI: `10.1109/CVPR.2017.243`.

Koffka, Kurt (1935). *Principles of Gestalt psychology*. English. Routledge Lond.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*.

Palmer, S. (1977). "Hierarchical structure in perceptual representation". In: *Cognitive Psychology* 9, pp. 441–474.

Segalin, Cristina, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J. Sun, Pietro Perona, David J. Anderson, and Ann Kennedy (2020). “The Mouse Action Recognition System (MARS): a software pipeline for automated analysis of social behaviors in mice”. In: *bioRxiv*. DOI: 10.1101/2020.07.26.222299. URL: https://www.biorxiv.org/content/early/2020/07/27/2020.07.26.222299.

Shang, Chao, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi (2018). “Edge Attention-based Multi-Relational Graph Convolutional Networks”. In:

Simonyan, Karen and Andrew Zisserman (2014). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556.

Su, Chi, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian (Oct. 2017). “Pose-Driven Deep Convolutional Model for Person Re-Identification”. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

Tan, Mingxing and Quoc Le (Sept. 2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

Wachsmuth, E., M. Oram, and D. Perrett (1994). “Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque.” In: *Cerebral cortex* 4 5, pp. 509–22.

Wang, Miao, Guo-Ye Yang, Ruilong Li, Run-Ze Liang, Song-Hai Zhang, Peter. M Hall, and Shi-Min Hu (June 2019). “Example-Guided Style-Consistent Image Synthesis from Semantic Labeling”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xie, Weidi, Li Shen, and Andrew Zisserman (2018). “Comparator Networks”. In: *European Conference on Computer Vision*.

Xu, Youjun, Jianfeng Pei, and Luhua Lai (2017). “Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction.” In: *Journal of chemical information and modeling* 57 11, pp. 2672–2685.

Yang, Linlin and Angela Yao (2019). “Disentangling Latent Hands for Image Synthesis and Pose Estimation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9869–9878.

Zhao, Haiyu, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang (2017). “Spindle Net: Person Re-identification with Human Body Region Guided Feature Decomposition and Fusion”. In: *Proc. IEEE CVPR*.

Chapter 2

POSE ESTIMATION

The content of this chapter is from the peer-reviewed publication "Anchor Loss: Modulating Loss Scale based on Prediction Difficulty" by S. Ryou, S.-G. Jeong, and P. Perona, appearing at ICCV 2019.

In Chapter 2, we discuss designing a good learning signal for training pose estimation problem and demonstrate its generalization ability by applying it to the image classification task.

2.1 Background

Pose estimation is a problem of localizing a predefined set of keypoints. Here we review the basic methodology that is widely used for solving pose estimation problem. As the definition of this problem indicates, algorithms for resolving pose estimation require understanding the high-resolution input and output space. With the development of a fully convolutional neural network (FCN) (Shelhamer, Long, and Darrell, 2017), deep neural networks can be designed to predict large spatial-dimensional outputs. Recent methods (Newell, K. Yang, and Deng, 2016; Wei et al., 2016; Chen et al., 2017; Cheng et al., 2020; J. Wang et al., 2021) follow the FCN architecture and output 2D gaussian heatmap where each heatmap encodes the location of the target part. Figure 2.1 illustrates how the human pose is encoded to a gaussian-shaped heatmap formulation.

The basic pipeline of the current pose estimation methods is shown in Figure 2.2. Given an input image, the human pose is encoded as 2D gaussian heatmaps as in Fig. 2.1. The network is trained to predict the heatmaps with the L2 loss between

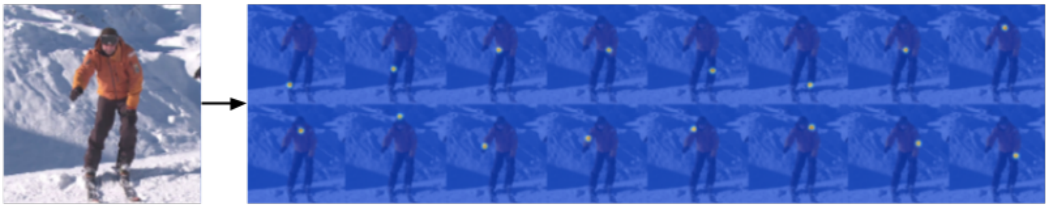


Figure 2.1: The human pose is encoded as 2D gaussian heatmaps. Each heatmap represents the location of the target part.

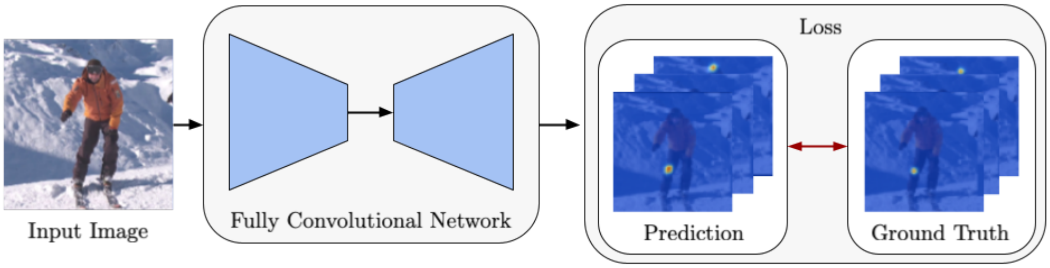


Figure 2.2: General pipeline for the pose estimation. Given an input image, a fully convolutional neural network predicts heatmaps and these heatmaps are compared to the ground truth by mean squared error.

the target and the predicted heatmap. At the inference time, we choose the location with the maximum score for each heatmap and estimate the final pose.

In this chapter, we propose a novel loss function that dynamically re-scales the cross entropy based on prediction difficulty regarding a sample. Deep neural network architectures in image classification tasks struggle to disambiguate visually similar objects. Likewise, in human pose estimation symmetric body parts often confuse the network with assigning indiscriminative scores to them. This is due to the output prediction, in which only the highest confidence label is selected without taking into consideration a measure of uncertainty. In this work, we define the prediction difficulty as a relative property coming from the confidence score gap between positive and negative labels. More precisely, the proposed loss function penalizes the network to avoid the score of a false prediction being significant. To demonstrate the efficacy of our loss function, we evaluate it on two different domains: image classification and human pose estimation. We find improvements in both applications by achieving higher accuracy compared to the baseline methods.

2.2 Introduction

In many computer vision tasks, deep neural networks produce bi-modal prediction scores when the labeled sample point is confused with the other class. Figure 2.3 illustrates some examples of network predictions with the presence of visually confusing cases. In all cases, though the network produces a non-trivial score about the correct label, the output prediction is wrong by taking the highest confidence label. For examples, human body parts are mostly composed of symmetric pairs. Even advanced deep architectures (He et al., 2016; Newell, K. Yang, and Deng, 2016) are vulnerable to mistaking subtle differences of the left-and-right body parts (Ronchi and Perona, 2017). Also, in image recognition, the output label

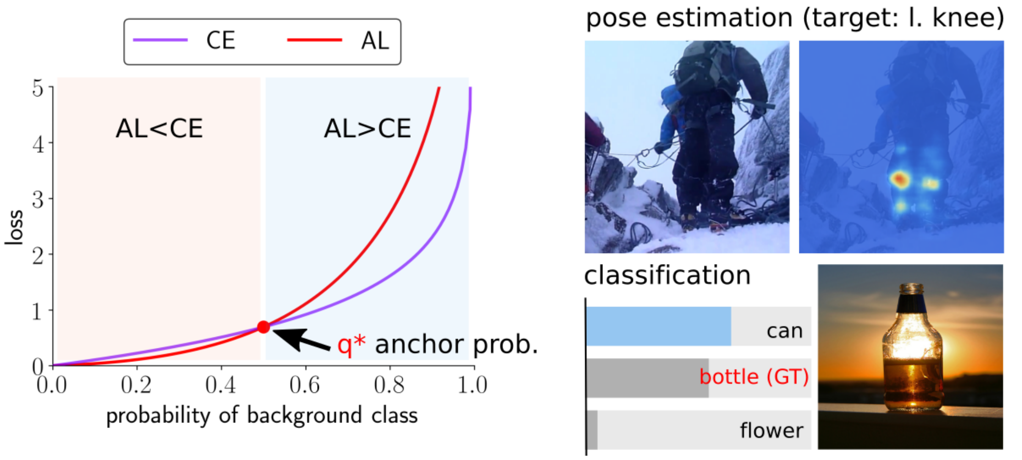
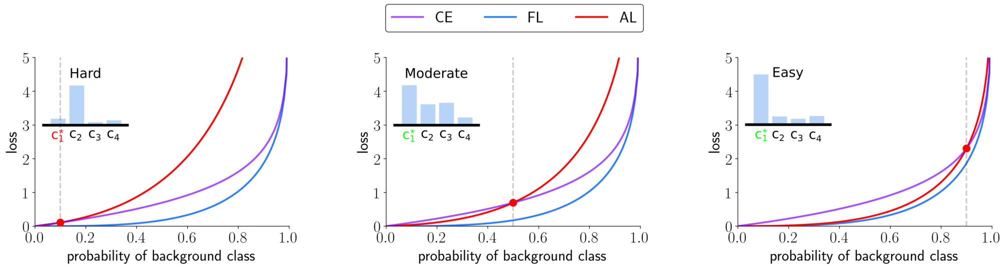


Figure 2.3: The overview of anchor loss. A network is confused about left-and-right body parts due to the symmetrical appearance of the human body, and struggles to disambiguate visually similar objects. Although the network output scores on the correct labels are relatively high, the final prediction is always chosen by the index of the highest score, resulting in a wrong prediction. Our loss function is designed to resolve this issue by penalizing more than cross entropy when the non-target (background) probability is higher than the anchor probability.

confusion of look-alike instances is an unsolved problem (Hoiem, Chodpathumwan, and Dai, 2012). Nevertheless, these tasks employ straightforward loss functions to optimize model parameters, *e.g.*, mean squared error or cross entropy.

In practice, look-alike instances incur an ambiguity in prediction scores, but it is hard to capture subtle differences in the network outputs by measuring the divergence of true and predicted distributions. Most classification tasks afterward make a final decision by choosing a label with the highest confidence score. We see that the relative score from the output distribution becomes an informative cue to resolve the confusion regarding the final prediction. We thus propose a novel loss function, which self-regulates its scale based on the relative difficulty of the prediction.

We introduce *anchor loss* that adaptively reshapes the loss values using the network outputs. Specifically, the proposed loss function evaluates the prediction difficulties using the relative confidence gap between the target and background output scores, produced by the network, to capture the uncertainty. In other words, we increase the loss for hard samples (Figure 2.4a), while we down-weight the loss when a sample leads the network to assign a relatively high confidence score about the target class (Figure 2.4c). Finally, the anchor loss alleviates the need for a post-processing step by taking the prediction difficulty into account while training.



(a) $q_* = 0.1$

(b) $q_* = 0.5$

(c) $q_* = 0.9$

Figure 2.4: How the anchor probability q_* affects our loss function compared to standard cross entropy (CE) and focal loss (FL). While FL always depresses the loss values for the samples producing trivial outcomes, anchor loss dynamically re-scales its loss values based on the relative difficulties of the target and the anchor probability. For these plots, the anchor probability is chosen as the prediction score ($q_* = q_{C_1}$) on the true positive label (C_1). Thus, if the networks produce higher score on the background label compared to the anchor, our loss encourages the network to correct the relative order of the predictions by penalizing more than the cross entropy.

This idea, adjusting the loss scales based on prediction difficulty, has been applied to the task of object detection, which inherently suffers from severe class imbalance issue (countless background vs. scarce object proposals). Focal loss (Lin et al., 2017) is designed to overcome such class imbalance by avoiding major gradient updates on trivial predictions. However, while the focal loss uniformly down-weights easy samples to ignore, the proposed loss function leverages the confidence gap between the target and non-target output values to modulate the loss scale of the samples in the training phase. We define the prediction difficulty using a reference value which we call *anchor probability* q_* obtained from the network predictions. The way to pick an anchor probability becomes a design choice. One way to use it is by taking the target prediction score as an anchor probability to modulate the background (non-target) loss values. As depicted in Figure 2.4, the proposed loss function varies based on the anchor probabilities q_*.

We propose anchor loss for improving the prediction of networks on the most semantically confusing cases at training time. Specifically, the proposed anchor loss dynamically controls its magnitude based on prediction difficulty, defined from the network outputs. We observe that our loss function encourages the separation gap between the true labeled score and the most competitive hypothesis. Our main contributions are: (i) the formulation of a novel loss function (anchor loss) for the task of image classification (Section 2.4), (ii) the adaptation of this loss function to human pose estimation (Section 2.4), and (iii) a graphical interpretation about the behavior of the anchor loss function compared to other losses (Figure 2.4 and 2.12). With extensive experiments, we show consistent improvements using anchor loss in terms of accuracy for image classification and human pose estimation tasks.

2.3 Related Work

Class Imbalance Issue. Image classification task suffers class imbalance issue from the long-tail distribution of real-world image datasets. Typical strategies to mitigate this issue are class re-sampling (Chawla et al., 2002; Han, W.-Y. Wang, and Mao, 2005; Buda, Maki, and Mazurowski, 2018) or cost-sensitive learning (Zhou and X.-Y. Liu, 2006; Huang et al., 2016; Dong, S. Gong, and Zhu, 2017). Class re-sampling methods (Chawla et al., 2002; Buda, Maki, and Mazurowski, 2018) redistribute the training data by oversampling the minority class or undersampling the majority class data. Cost-sensitive learning (Huang et al., 2016; Dong, S. Gong, and Zhu, 2017) adjusts the loss value by assigning more weights on the misclassified minority classes. Above mentioned prior methods mainly focus on compensating scarce data by innate statistics of the dataset. On the other hand, our loss function renders prediction difficulties from network outputs without requiring prior knowledge about the data distributions.

Relative Property in Prediction. Several researchers attempt to separate confidence scores of the foreground and background classes for the robustness (Y. Gong et al., 2014; M.-L. Zhang and Zhou, 2006). Pairwise ranking (Y. Gong et al., 2014) has been successfully adopted in the multi-label image classification task, but efficient sampling becomes an issue when the vocabulary size increases. From the idea of employing a margin constraint between classes, L-softmax loss (Weiyang Liu et al., 2016) combines the last fully-connected layer, softmax, and the cross entropy loss to encourage intra-class compactness and inter-class separability in the feature space. While we do not regularize the ordinality of the outputs, our loss function implicitly embodies the concept of ranking. In other words, the proposed loss function rules out a reversed prediction about target and background classes with re-scaling loss values.

Outliers Removal vs. Hard Negative Mining. Studies about robust estimation (Huber, 1964; T. Zhang, 2004), try to reduce the contribution on model parameter optimization from anomaly samples. Specifically, noise-robust losses (Hendrycks et al., 2018; Z. Zhang and Sabuncu, 2018; Ren et al., 2018) have been introduced to support the model training even in the presence of the noise in annotations. Berrada *et al* (Berrada, Zisserman, and Kumar, 2018) address the label confusion problem in the image classification task, such as incorrect annotation or multiple categories present in a single image, and propose a smooth loss function for top-k classification. Deep regression approaches (Barron, 2019; Belagiannis et al., 2015) reduce the impact of outliers by minimizing M-estimator with various robust penalties as a loss function. Barron (Barron, 2019) proposed a generalization of common robust loss functions with a single continuous-valued robustness parameter, where the loss function is interpreted as a probability distribution to adapt the robustness.

On the contrary, there have been many studies with an opposite view in various domains, by handling the loss contribution from hard examples as a significant learning signal. Hard negative mining, originally called *Bootstrapping* (Sung, 1996), follows an iterative bootstrapping procedure by selecting background examples for which the detector triggers a false alarm. Online hard example mining (OHEM) (Shrivastava, Gupta, and Girshick, 2016) successfully adopts this idea to train deep ConvNet detectors in the object detection task. Pose estimation community also explored re-distributing gradient update based on the sample difficulty. Online Hard Keypoint Mining (OHKM) (Chen et al., 2017) re-weights the loss by sampling few keypoint heatmaps which have high loss contribution, and the gradient is propagated only through the selected heatmaps. Our work has a similar viewpoint to the latter works to put more emphasis on the hard examples.

Focal Loss. One-stage object detection task has an inherent class imbalance issue due to a huge gap between the number of proposals and the number of boxes containing real objects. To resolve this extreme class imbalance issue, some works perform sampling hard examples while training (Shrivastava, Gupta, and Girshick, 2016; Felzenszwalb, Girshick, and McAllester, 2010; Wei Liu et al., 2015), or design a loss function (Lin et al., 2017) to reshape loss by down-weighting the easy examples. Focal loss (Lin et al., 2017) also addresses the importance of learning signal from hard examples in the one-stage object detection task. Without sampling processes, focal loss efficiently rescales the loss function and prevents the gradient update from being overwhelmed by the easy-negatives. Our work is motivated by the mathematical formulation of focal loss (Lin et al., 2017), where predefined modulating term increases the importance of correcting hard examples.

Human Pose Estimation. Human pose estimation is a problem of localizing human body part locations in an input image. Most of the current works (Newell, K. Yang, and Deng, 2016; Chen et al., 2017; Wei et al., 2016; W. Yang et al., 2017; Ke et al., 2018; Tang, Yu, and Wu, 2018) use a deep convolutional neural network and generate the output as a 2D heatmap, which is encoded as a gaussian map centered at each body part location. Hourglass network (Newell, K. Yang, and Deng, 2016) exploits the iterative refinements on the predictions from the repeated encoder-decoder architecture design to capture complex spatial relationships. Even with deep architectures, disambiguating look-alike body parts remain as a main problem (Ronchi and Perona, 2017) in pose estimation community. Recent methods (W. Yang et al., 2017; Chu et al., 2017; Ke et al., 2018), built on top of the hourglass network, use multi-scale and body part structure information to improve the performance by adding more architectural components.

While there has been much interest in finding a good architecture tailored to the pose estimation problem, the vast majority of papers simply use mean squared error (MSE), which computes the L2 distance between the output and the prediction heatmap, as a loss function for this task. OHKM (Chen et al., 2017), which updates the gradient from the selected set of keypoint heatmaps, improves the performance when properly used in the refinement step. On the other hand, we propose a loss scaling scheme that efficiently redistributes the loss values without sampling hard examples.

2.4 Method

In this section, we introduce *anchor loss* and explain the design choices for image classification and pose estimation tasks. First, we define the prediction difficulty and provide related examples. We then present the generalized form of the anchor loss function. We tailor our loss function on visual understanding tasks: image classification and human pose estimation. Finally, we give theoretical insight in comparison to other loss functions.

Anchor Loss

The inference step for most classification tasks chooses the label index corresponding to the highest probability. Figure 2.3 shows sample outputs from the model trained with cross entropy. Although optimizing the networks with the cross entropy encourages the predicted distribution to resemble the true distribution, it does not convey the relative property between the predictions on each class.

Anchor loss function dynamically reweighs the loss value with respect to prediction difficulty. The prediction difficulty is determined by measuring the divergence between the probabilities of the true and false predictions. Here the anchor probability q_* becomes a reference value for determining the prediction difficulty. The definition of anchor probability q_* is arbitrary and becomes a design choice. However, in practice, we observed that setting anchor probability to the target class prediction score gives the best performance, so we use it for the rest of the paper. With consideration of the prediction difficulties, we formulate the loss function as follows:

$$\ell(p, q; \gamma) = -\underbrace{(1 + \overbrace{q - q_*}^{\text{prediction difficulty}})^\gamma}_{\text{modulator}} \underbrace{(1-p)\log(1-q)}_{\text{cross entropy}}, \tag{2.1}$$

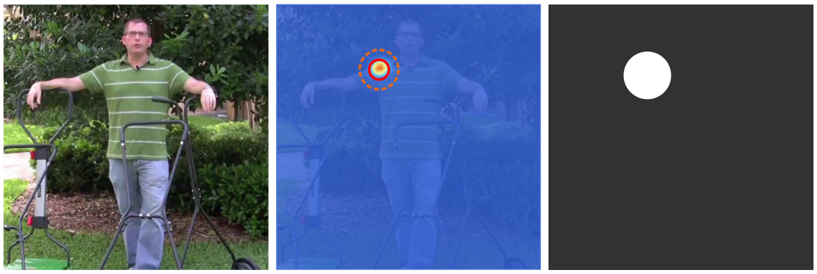
where p and q denote empirical label and predicted probabilities, respectively. The anchor probability q_* is determined by the primitive logits, where the anchor is the prediction score on the true positive label. Here, $\gamma \geq 0$ is a hyperparameter that controls the dynamic range of the loss function. Our loss is separable into two parts: modulator and cross entropy. The modulator is a monotonic increasing function that takes relative prediction difficulties into account, where the domain is bounded by $|q - q_*| < 1$. Suppose q_* be the target class prediction score. In an easy prediction scenario, the network assigns a correct label for the given sample point; hence q_* will be larger than any q. We illustrate the prediction difficulties as follows:

- **Easy case** ($q < q_*$): the loss function is suppressed, and thus rules out less informative samples when updating the model;
- **Moderate case** ($q = q_*$): the loss function is equivalent to cross entropy, since the modulator becomes 1; and
- **Hard case** ($q > q_*$): the loss function penalizes more than cross entropy for most of the range, since the true positive probability q_* is low.

As a result, we apply different loss functions for each sample.

Classification

For image classification, we adopt sigmoid-binary cross entropy as a basic setup to diversify the way of scaling loss values. Unlike softmax, sigmoid activation handles each class output probability as an independent variable, where each label represents whether the image contains an object of corresponding class or not. This



(a) input (b) heatmap (c) mask

Figure 2.5: How an anchor probability is chosen for the pose estimation task. For the target body part of right shoulder (b), the maximum confidence score inside the solid red circle becomes an anchor probability to modulate the loss values in mask areas (c).

formulation also enables our loss function to capture subtle differences from the output space by modulating the loss values on each label.

For image classification, we obtained the best performance when we set the anchor probability to the output score of the target class. The mathematical formulation becomes as follows:

$$\begin{aligned} &\ell_{cls}(p, q; \gamma) \\ &= -\sum_{k=1}^{K} p_k \log q_k + (1 - p_k)(1 + q_k - q_*)^{\gamma} \log(1 - q_k), \end{aligned} \tag{2.2}$$

where p_k and q_k represent the empirical label and the predicted probability for class k. We add a margin variable δ to anchor probability q_* to penalize the output variables which have lower but close to the true positive prediction score. Thus the final anchor probability becomes $q_* = q_t - \delta$, where t represents the target index ($p_t = 1$), and we set δ to 0.05.

Pose Estimation

Current pose estimation methods generate a keypoint heatmap for each body part at the end of the prediction stage, and predict the pixel location that has the highest probability. The main difference of pose estimation and object classification tasks is that the target has spatial dependency between adjacent pixel locations. As a result, assigning a single pixel as the true positive may incur a huge penalty on adjacent pixels. To alleviate this issue, we adopt a gaussian heatmap centered on the target keypoint as the same encoding scheme as the previous works (Newell, K. Yang, and Deng, 2016; Wei et al., 2016; Chen et al., 2017), and apply our loss function on only true negative pixels ($p_i = 0$). In other words, we use a mask variable $M(p)$ to designate the pixel locations where our loss function applies, and use standard binary cross entropy on unmasked locations.

$$M(p) = \begin{cases} 1 & \text{if } p = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

As in object classification, we found that using true-positive probability value to penalize background pixel locations gives better performance. Considering the spatial dependency, anchor probabilities are chosen spatially from the circle of high confidence, where the ground truth probability is greater than 0.5. That is,

$$q_* = \max_{i \forall p_i > 0.5} q_i. \tag{2.4}$$

We illustrate this procedure in Figure 2.5. For simplicity, we denote the standard binary cross entropy as ℓ_{BCE}. Finally, our loss function for pose estimation problem is defined as:

$$\begin{aligned} \ell_{pose}(p, q; \gamma) =& [M(p) * (1 + q - q_*)^\gamma \\ & + (1 - M(p))] * \ell_{BCE}(p, q). \end{aligned} \tag{2.5}$$

Relationship to Other Loss Functions

Our goal is to design a loss function which takes the relative property of the inference step into account. In this section, we discuss how binary cross entropy (2.6) and focal loss (Lin et al., 2017) (2.7) relate to anchor loss. Let $p \in \{0, 1\}$ denote the ground truth, and $q \in [0, 1]$ represent predicted distribution. The loss functions are

$$\ell_{CE}(p, q) = -\big[p \log(q) + (1 - p) \log(1 - q)\big], \tag{2.6}$$

$$\ell_{FL}(p, q; \gamma) = -\big[p(1 - q)^\gamma \log(q) + (1 - p) q^\gamma \log(1 - q)\big]. \tag{2.7}$$

For the sake of conciseness, we define the probability of ground truth as $q_t = pq + (1 - p)(1 - q)$. Then we replace the loss functions as follows:

$$\ell_{CE}(q_t) = -\log(q_t), \tag{2.8}$$

$$\ell_{FL}(q_t;\gamma) = -(1-q_t)^{\gamma}\log(q_t), \tag{2.9}$$

where q represents the output vector from the network. The modulating factor $(1-q_t)^{\gamma}$ with focusing parameter γ reshapes the loss function to down-weight easy samples. Focal loss was introduced to resolve the extreme class imbalance issue in object detection, where the majority of the loss is comprised of easily classified background examples. Object detection requires the absolute threshold value to decide the candidate box is foreground or background. On the other hand, classification requires the confidence score of the ground truth label to be higher than all other label scores.

If we set $q_* = 1-p$, which means $q_* = 1$ for the background classes and $q_* = 0$ for the target class:

$$q_* = \begin{cases} 1 & p = 0 \quad \text{background classes,} \\ 0 & p = 1 \quad \text{target class,} \end{cases} \tag{2.10}$$

then the modulator becomes:

$$(1 - q_t + q_*) = \begin{cases} (1-(1-q)+1) = (1-q) & p = 0, \\ (1-q+0) = q & p = 1, \end{cases} \tag{2.11}$$

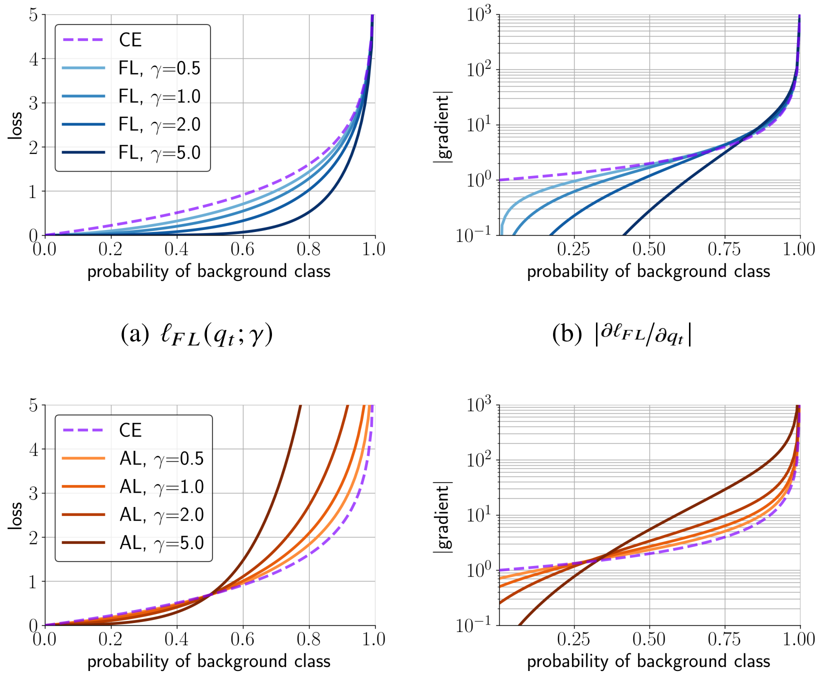
and feeding this modulator value to anchor loss becomes a mathematical formulation of focal loss:

$$\begin{aligned} &\ell_{AL}(p,q;\gamma) = -\big[p(1-q)^{\gamma}\log(q) + (1-p)q^{\gamma}\log(1-q)\big], \\ &\text{where } q_* = 1-p. \end{aligned} \tag{2.12}$$

If we set $\gamma = 0$, the the modulator term becomes 1, and anchor loss becomes binary cross entropy.

Gradient Analysis

We compute the gradient of our loss function and compare with the binary cross entropy and the focal loss. For simplicity, we focus on the loss of background label, which we discuss in Section 2.4. Note that we detach the anchor probability q_* while backpropagation and only use it as a scaling term in the modulator.



(a) $\ell_{FL}(q_t;\gamma)$

(b) $|\partial \ell_{FL}/\partial q_t|$

(c) $\ell_{AL}(q_t;\gamma), q_* = 0.5$

(d) $|\partial \ell_{AL}/\partial q_t|$

Figure 2.6: Gradient figure: sample gradient output of background probability distribution. Compared to the cross entropy, the magnitude of gradient increases when the prediction is higher than the anchor probability.

$$\ell_{AL}(q) = -(1 + q - q_*)^{\gamma} \log(1 - q) \tag{2.13}$$

$$\frac{\partial \ell_{AL}}{\partial q}(q) = -(1 + q - q_*)^{\gamma - 1} \left[\gamma \log(1 - q) - \frac{1 + q - q_*}{1 - q} \right] \tag{2.14}$$

Figure 2.12 shows the gradient of our loss function, focal loss, and cross entropy. Compared to the cross entropy, the gradient values of focal loss are suppressed for all ranges. On the other hand, our loss function assigns larger gradient values when the prediction is higher than the anchor probability, and vice versa.

2.5 Experiments

We conduct experiments on image classification and human pose estimation. In this section, we briefly overview the methods that we use in each domain, and discuss the experimental results.

Table 2.1: Classification accuracy on CIFAR (ResNet-110)

		CIFAR-10	**CIFAR-100**	
Loss Fn.	Parameter	Top-1	Top-1	Top-5
CE		93.91 ± 0.12	72.98 ± 0.35	92.55 ± 0.30
BCE		93.69 ± 0.08	73.88 ± 0.22	92.03 ± 0.42
OHEM	$\rho = 0.9, 0.9$	93.90 ± 0.10	73.03 ± 0.29	92.61 ± 0.21
FL	$\gamma = 2.0, 0.5$	94.05 ± 0.23	74.01 ± 0.04	92.47 ± 0.40
Ours				
AL	$\gamma = 0.5, 0.5$	94.10 ± 0.15	74.25 ± 0.34	**92.62** ± 0.50
AL w/ warmup	$\gamma = 0.5, 2.0$	**94.17** ± 0.13	**74.38** ± 0.45	92.45 ± 0.05

Table 2.2: Classification accuracies on ImageNet (ResNet-50)

Loss Fn.	Parameter	Top-1	Top-5
CE		76.39	93.20
OHEM	$\rho = 0.8$	76.27	93.21
FL	$\gamma = 0.5$	76.72	93.06
AL (ours)	$\gamma = 0.5$	**76.82**	93.03

Image Classification

Datasets. For the object classification, we evaluate our method on CIFAR-10/100 (Krizhevsky, 2009) and ImageNet (ILSVRC 2012) (Deng et al., 2009). CIFAR 10 and 100 each consist of 60,000 images with 32×32 size of 50,000 training and 10,000 testing images. In our experiment, we randomly select 5,000 images for the validation set. CIFAR-10 dataset has 10 labels with 6,000 images per class, and CIFAR-100 dataset has 100 classes each containing 600 images.

Implementation details. For CIFAR, we train ResNet-110 (He et al., 2016) with our loss function and compare with other loss functions and OHEM. We randomly flip and crop the images padded with 4 pixels on each side for data augmentation. All the models are trained with PyTorch (Paszke et al., 2017). Note that our loss is summed over class variables and averaged over batch. The learning rate is set to 0.1 initially, and dropped by a factor of 0.1 at 160 and 180 epochs respectively. In addition, we train ResNet-50 models on ImageNet using different loss functions. We use 8 GPUs and batch size of 224. To accelerate training, we employ a mixed-precision. We apply minimal data augmentation, *i.e.*, random cropping of 224×224 and horizontal flipping. The learning rate starts from 0.1 and decays 0.1 every 30 epoch. We also perform learning rate warmup strategy for first 5 epochs as proposed in (He et al., 2016).

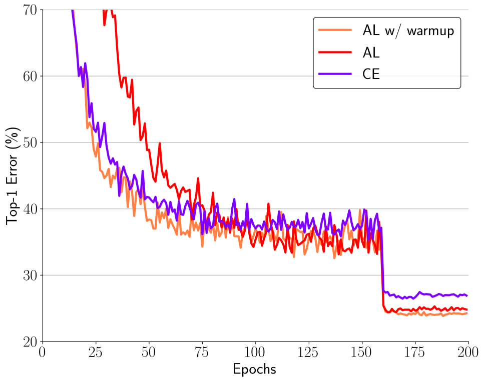


Figure 2.7: Validation curves of ResNet-110 on CIFAR-100 dataset. We compare our loss function to CE.

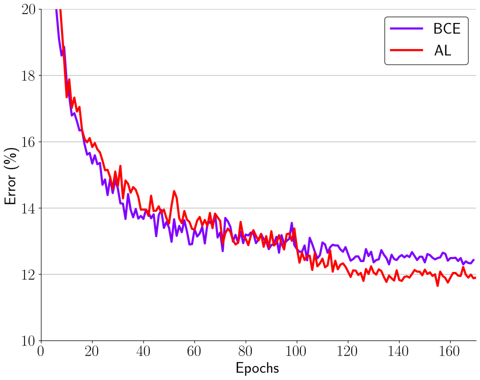


Figure 2.8: Validation curves of 2-stacked Hourglass on MPII dataset. We compare our loss function to BCE.

Results. For CIFAR, we train and test the network three times and report the mean and standard deviation in Table 2.1. We report top-1 and top-5 accuracy and compare the score with other loss functions and OHEM. OHEM computes the loss values for all samples in a batch, chooses the samples of high loss contribution with a ratio of ρ, and updates the gradient only using those samples. As we can see in the Table 2.1, our loss function has shown improvements over all loss functions we evaluated. For CIFAR 100, performance improved by simply replacing the cross entropy to the binary cross entropy, and anchor loss gives further gain by exploiting the automated re-scaling scheme. With our experimental setting, we found that sampling hard examples (OHEM) does not help. We tried out few different sampling ratio settings, but found performance degradation over all ratios.

Ablation Studies. As an ablation study, we report the top-1 and top-5 accuracy on CIFAR-100 by varying the γ in Table 2.3. For classification task, low γ yielded a good performance. We also perform experiments with fixed anchor probabilities to see how the automated sample difficulty from the network helps training. The results in Table 2.3 show that using the network output to define sample difficulty and rescale the loss based on this value helps the network keep a good learning signal.

CE warmup strategy. To accelerate and stabilize the training process, we use CE for first few epochs and then replace loss function to AL. We tested CE warmup on CIFAR-100 for the first 5 epochs (Figure 2.7). With the warmup strategy, the ratio

Table 2.3: Ablation studies on CIFAR-100 (ResNet-110)

		Top-1	Top-5
Static anchor probabilities			
$\gamma = 0.5$	$q_* = 0.8$	73.74	92.45
$\gamma = 0.5$	$q_* = 0.5$	73.77	92.30
$\gamma = 0.5$	$q_* = 0.1$	73.11	92.08
Dynamic anchor probabilities			
$\gamma = 0.5$	-	**74.25**	**92.62**
$\gamma = 1.0$	-	73.59	92.04
$\gamma = 2.0$	-	71.86	91.46

of hard samples was decreased; in other words, loss function less fluctuated. As a result, we achieved the highest top-1 accuracy of 74.38% (averaged out multiple runs) regardless of a high $\gamma = 2$ value.

Human Pose Estimation

We evaluate our method on two different human pose estimation datasets: single-person pose on MPII (Andriluka et al., 2014) and LSP (Johnson and Everingham, 2010) dataset. The single-person pose estimation problem assumes that the position and the scale information of a target person are given.

Implementation details. For the task of human pose estimation, we use the Hourglass network (Newell, K. Yang, and Deng, 2016) as a baseline and only replace the loss function with the proposed loss during training. Note that we put sigmoid activation layer on top of the standard architecture to perform classification. Pose models are trained using Torch (Collobert, Kavukcuoglu, and Farabet, 2011) framework. The input size is set to 256×256, batch size is 6, and the model is trained with a single NVIDIA Tesla V100 GPU. Learning rate is set to 0.001 for the first 100 epochs and dropped by half and 0.2 iteratively at every 20 epoch. Testing is held by averaging the heatmaps over six-scale image pyramid with flipping.

Datasets. The MPII human pose dataset consists of 20k training images over 40k people performing various activities. We follow the previous training/validation split from (Tompson et al., 2015), where 3k images from training set are used for validation. The LSP dataset (Johnson and Everingham, 2010) is composed of 11k training images with LSP extended dataset (Johnson and Everingham, 2011), and containing mostly sports activities.

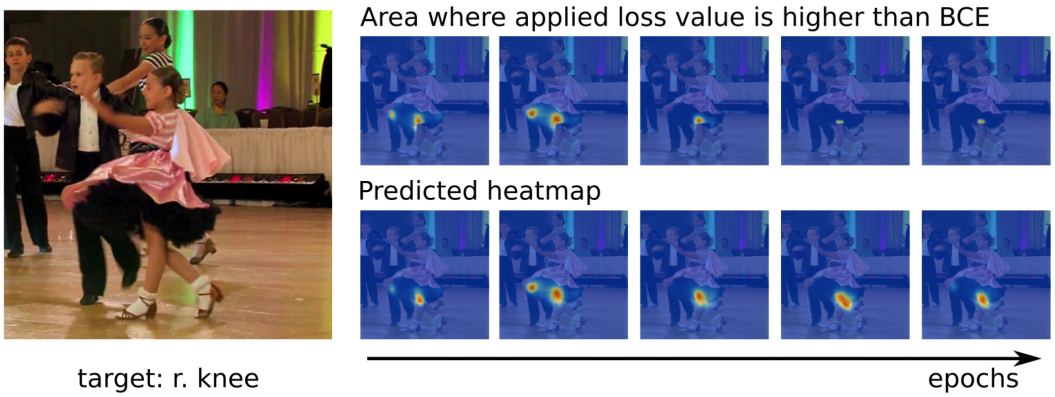


Figure 2.9: Anchor loss visualization on pose estimation. We visualize where anchor loss assigns higher loss values than the binary cross entropy and how it changes over training epochs. At the beginning, visually similar parts often get higher scores than the target body part, thus our loss function assigns higher weights on those pixel locations. Once the model is able to detect the target body part with high confidence, loss is down-weighted for most of the areas, so that the network can focus on finding more accurate location for the target body part.

Results. We evaluate the single-person pose estimation results on standard Percentage of Correct Keypoints (PCK) metric, which defines correct prediction if the distance between the output and the ground truth position lies in α with respect to the scale of the person. α is set to 0.5 and 0.2 in MPII and LSP dataset, respectively. PCK score for each dataset is reported in Table 2.4 and 2.5.

For comparison, we split the performance table by hourglass-based architecture. The bottom rows are comparison between the methods built on top of hourglass network. We achieve comparable results to the models built on top of hourglass network with more computational complexity on both datasets. We also report the validation score of the baseline method trained with mean squared error by conducting a single scale test for direct comparison between the losses in Table 2.6. We found consistent improvements over the symmetric parts; due to appearance similarity on the symmetric body parts, our loss function automatically penalizes more on those parts during training, without having any additional constraint for the symmetric parts.

Ablation Studies. We conduct ablation studies by varying γ on 2-stacked hourglass network and report the score in Table 2.7. With proper selection of $\gamma = 2.0$, we can achieve better performance over all the losses.

Table 2.4: PCK score on MPII dataset. The bottom rows show the performances of the methods built on top of hourglass network. The model trained with anchor loss shows comparative scores to the results from more complex models.

Hourglass model variants	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Hourglass + MSE (Baseline)	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Hourglass + AL (Ours)	**98.6**	96.6	92.3	87.8	90.8	88.8	86.0	91.9
Chu *et al*	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chen *et al*	98.1	96.5	92.5	88.5	90.2	**89.6**	86.0	91.9
Yang *et al*	98.5	96.7	92.5	**88.7**	**91.1**	88.6	86.0	92.0
Ke *et al*	98.5	**96.8**	**92.7**	88.4	90.6	89.3	**86.3**	**92.1**

Table 2.5: PCP score on LSP dataset.

Method	Torso	U.leg	L.leg	U.arm	F.arm	Head	Total
Yang et al., ICCV'17	99.1	95.7	93.9	91.1	84.3	96.7	92.6
Ning et al., TMM'17	98.6	95.8	93.6	90.7	84.2	96.4	92.3
Chu et al., CVPR'17	98.4	95.0	92.8	88.5	81.2	95.7	90.9
Bulat &Tzimiropoulos, ECCV'16	97.7	92.4	89.3	86.7	79.7	95.2	88.9
Wei et al., CVPR'16	98.0	92.2	89.1	85.8	77.9	95.0	88.3
Insafutdinov et al., ECCV'16	97.0	90.6	86.9	86.1	79.5	95.4	87.8
Pishchulin et al., CVPR'16	97.0	88.8	82.0	82.4	71.8	95.8	84.3
Lifshitz et al., ECCV'16	97.3	88.8	84.4	80.6	71.4	94.8	84.3
Yu et al., ECCV'16	98.0	93.1	88.1	82.9	72.6	83.0	85.4
Rafi et al., BMVC'16	97.6	87.3	80.2	76.8	66.2	93.3	81.2
Yang et al., CVPR'16	95.6	78.5	71.8	72.2	61.8	83.9	74.8
Ours	99.0	95.7	94.0	90.8	84.8	97.2	92.7

Table 2.6: Validation Results on MPII dataset. We report the validation score of the result using different losses with the same single-scale testing setup.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Hourglass + MSE	**96.73**	95.94	90.39	85.40	89.04	85.17	81.86	89.32
Hourglass + AL (Ours)	96.45	**96.04**	**90.46**	**86.00**	**89.20**	**86.84**	**83.68**	**89.93**

Table 2.7: Hyperparameter search and comparison to other losses on MPII dataset with 2-stacked hourglass network.

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
BCE	96.42	95.35	89.82	84.72	88.47	85.17	81.13	88.84
MSE	96.42	95.30	89.57	84.63	88.78	85.07	**81.77**	88.89
FL	**96.52**	**95.47**	89.71	84.87	88.38	84.75	81.25	88.81
AL, $\gamma = 5$	96.35	95.04	89.26	84.56	**88.99**	**85.51**	81.37	88.84
AL, $\gamma = 1$	96.35	95.40	89.60	85.11	88.59	84.85	**81.77**	88.94
AL, $\gamma = 2$	96.49	95.45	**90.08**	**85.42**	88.64	85.31	81.60	**89.11**

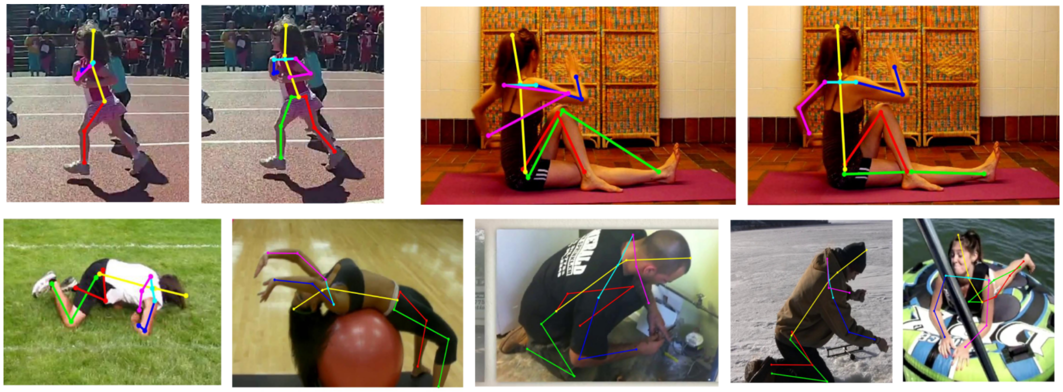


Figure 2.10: Qualitative results on human pose. The first row compares with the result from MSE loss (left) and our loss (right), and the second row contains some sample outputs. Model trained with the proposed loss function is robust at predicting symmetric body parts.

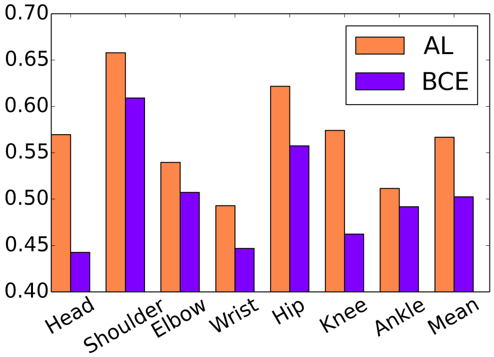


Figure 2.11: Double counting problem. We analyze how the anchor loss behaves when a double-counting problem occurs.

Qualitative Analysis. We visualize which area gets more penalty than the standard binary cross entropy in Fig 2.9. For the fist few epochs, we can see that visually similar parts of both target and non-target person get higher penalty. Once the model finds the correct body part locations, the loss function is down-weighted and the area of higher penalty is focused only on few pixel locations, which helps fine adjustments on finding more accurate locations. We also show some sample outputs in Fig 2.10. For comparison, the top row shows some outputs from the model trained with MSE (left) and anchor loss (right). We can see that the network trained with proposed loss is robust at predicting symmetric parts.

Double-counting. For the task of human pose estimation, we observe a double-counting problem, where the predicted heatmap shows multiple peaks. To analyze how AL behaves in those cases, we depict the ratio of the correct prediction when double-counting problems are encountered on MPII dataset. Overall, AL assigns correct body parts compared to BCE.

2.6 Conclusions

In this paper, we presented anchor loss function which adaptively re-scales the standard cross entropy function based on sample difficulty. The network automatically evaluates the sample difficulty by measuring the divergence between the network's true positive and false positive predictions. The proposed loss function has shown strong empirical results on two different domains: image classification and human pose estimation. A simple drop-in replacement for standard cross entropy loss gives performance improvement. With proper selection of designing the re-weighing scheme and anchor probability, we believe this loss function can generalize to other settings.

2.7 Appendix: Anchor Design

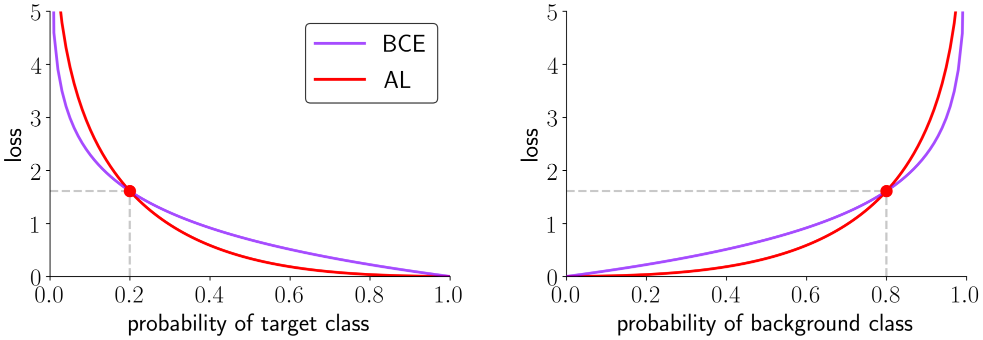
Anchor Design

In the paper, we set the anchor probability to the target class prediction score and modulate loss of the background class. Here we further study how to design anchor probability that affects behavior of the loss. We first define the basic formulation of anchor loss (AL) with sigmoid-binary cross entropy:

$$\begin{aligned}\ell(p,q;\gamma) = &-\underbrace{(1-q+q_{pos})^{\gamma_t}\, p\log(q)}_{\text{target class}} \\ &-\underbrace{(1+q-q_{neg})^{\gamma_b}(1-p)\log(1-q)}_{\text{background class}}.\end{aligned} \tag{2.15}$$

Anchor probability is a reference value for determining the prediction difficulty, which is defined as a confidence score gap between the target and background classes. The prediction difficulty is used to modulate loss values either by (i) pushing the loss of target class high, (ii) suppressing the loss of background classes, or (iii) using both ways around. The details of parameter setting for each case are as follows:

(i) **Modulate loss for target class**: We set the anchor probability to the maximum



(a) Modulate target loss

(b) Modulate background loss

Figure 2.12: How an anchor probability modulates loss values. When the prediction score of target class is lower than $q_{pos} = 0.2$, anchor loss penalizes more than binary cross entropy (a). On the contrary, when the prediction score of background class is higher than $q_{neg} = 0.8$, the loss value becomes higher than the binary cross entropy (b).

prediction score among background classes. Hence, target class loss gets more penalty when its score is lower than the anchor probability.

$$\begin{aligned} q_* &= \max_{i,\forall p_i=0} q_i, \\ \gamma_t &= \gamma \text{ and } \gamma_b = 0. \end{aligned} \tag{2.16}$$

(ii) **Modulate loss for background classes**: We set the anchor probability to prediction score of the target class. Anchor loss is penalized more when output scores of the background classes are higher than the target.

$$\begin{aligned} q_{neg} &= q_j, \text{ for } j, p_j = 1, \\ \gamma_t &= 0 \text{ and } \gamma_b = \gamma. \end{aligned} \tag{2.17}$$

(iii) **Modulate loss for both target and background classes**: We modulate loss on both directions by combining the above cases.

$$\begin{aligned} q_{pos} &= \max_{i,\forall p_i=0} q_i, \\ q_{neg} &= q_j, \text{ for } j, p_j = 1, \\ \gamma_t &= \gamma_b = \gamma. \end{aligned} \tag{2.18}$$

Table 2.8: Classification accuracies on CIFAR-100 with different anchor probabilities

loss fn.	Top-1	Top-5
BCE	73.88 ± 0.22	92.03 ± 0.42
(i)	74.06 ± 0.53	92.32 ± 0.24
(ii)	**74.25** ± 0.34	**92.62** ± 0.50
(iii)	73.90 ± 0.40	92.24 ± 0.06

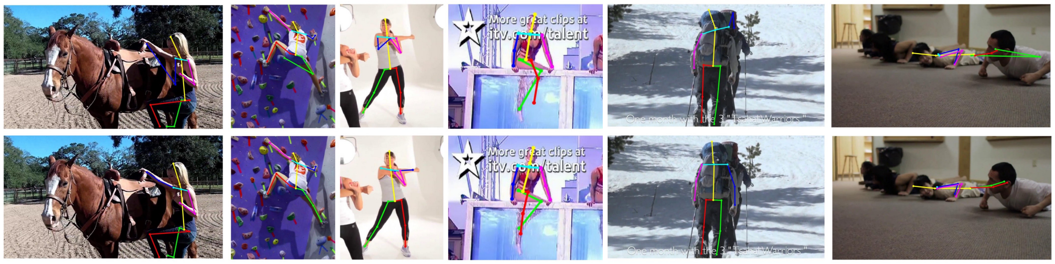


Figure 2.13: Qualitative results for human pose estimation. Top row shows the output images with baseline (MSE) and bottom row represents the outcomes with anchor loss.

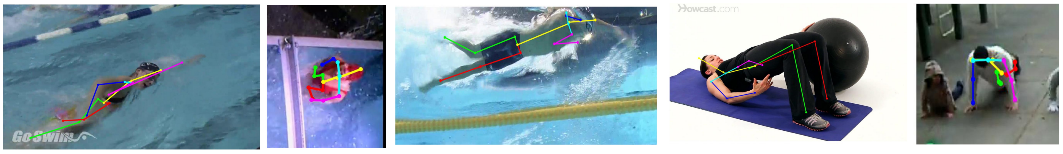


Figure 2.14: Failure cases on human pose estimation. Network trained with anchor loss still fails to detect correct body part locations when the body part is blurred or self-occluded.

We report image classification performance on CIFAR-100 by varying the way of designing anchor probability in Table 2.8. We achieve the best performance by modulating the loss for background classes (ii).

2.8 Appendix: Qualitative Results

Qualitative figures

We visualize qualitative results for human pose estimation (Fig. 2.13, 2.14) and image classification (Fig. 2.15). Network trained with anchor loss has shown im-

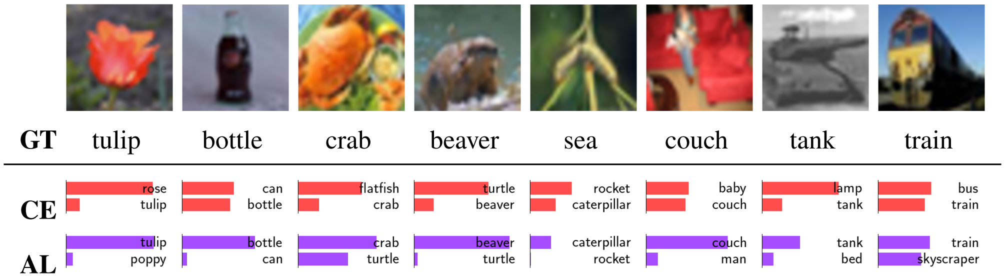


Figure 2.15: Image classification results on CIFAR-100. We compare the top-2 prediction scores of ResNet-110 with cross entropy (CE) and anchor loss (AL). Network trained with anchor loss successfully classifies difficult examples even though the model trained with cross entropy fails.

provement over the baseline losses for both tasks. Specifically, anchor loss shows its potential use for multi-person pose estimation by finding correct body parts when the target person is occluded or overlapped by other person (last two columns of Fig. 2.13).

References

Andriluka, Mykhaylo, Leonid Pishchulin, Peter Gehler, and Bernt Schiele (June 2014). "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *Proc. IEEE CVPR*.

Barron, Jonathan T. (2019). "A General and Adaptive Robust Loss Function". In: *Proc. IEEE CVPR*.

Belagiannis, Vasileios, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab (2015). "Robust Optimization for Deep Regression". In: *Proc. IEEE ICCV*.

Berrada, Leonard, Andrew Zisserman, and M. Pawan Kumar (2018). "Smooth Loss Functions for Deep Top-k Classification". In: *CoRR*. URL: `http://arxiv.org/abs/1802.07595`.

Buda, Mateusz, Atsuto Maki, and Maciej A. Mazurowski (2018). "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106, pp. 249–259. ISSN: 0893-6080. DOI: `https://doi.org/10.1016/j.neunet.2018.07.011`. URL: `http://www.sciencedirect.com/science/article/pii/S0893608018302107`.

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer (2002). "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16, pp. 321–357.

Chen, Yilun, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun (2017). "Cascaded Pyramid Network for Multi-Person Pose Estimation". In: *CoRR* abs/1711.07319.

Cheng, Bowen, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang (June 2020). "HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chu, Xiao, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang (2017). "Multi-context Attention for Human Pose Estimation". In: *Proc. IEEE CVPR*.

Collobert, R., K. Kavukcuoglu, and C. Farabet (2011). "Torch7: A Matlab-like Environment for Machine Learning". In: *BigLearn, NIPS Workshop*.

Deng, Jia, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Fei-Fei Li (2009). "ImageNet: A large-scale hierarchical image database". In: *Proc. IEEE CVPR*.

Dong, Qi, Shaogang Gong, and Xiatian Zhu (2017). "Class Rectification Hard Mining for Imbalanced Deep Learning". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 1869–1878. DOI: 10.1109/ICCV.2017.205. URL: https://doi.org/10.1109/ICCV.2017.205.

Felzenszwalb, Pedro F., Ross B. Girshick, and David A. McAllester (2010). "Cascade object detection with deformable part models". In: *Proc. IEEE CVPR*.

Gong, Yunchao, Yangqing Jia, Thomas K. Leung, Alexander Toshev, and Sergey Ioffe (2014). "deep convolutional ranking for multi label image annotation". In: *ICLR*.

Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao (2005). "Borderline-SMOTE: A New Over-sampling Method in Imbalanced Data Sets Learning". In: *Proceedings of the 2005 International Conference on Advances in Intelligent Computing - Volume Part I*. ICIC'05. Hefei, China: Springer-Verlag, pp. 878–887. ISBN: 3-540-28226-2, 978-3-540-28226-6. DOI: 10.1007/11538059_91. URL: http://dx.doi.org/10.1007/11538059_91.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proc. IEEE CVPR*.

Hendrycks, Dan, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel (2018). "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise". In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 10456–10465. URL: http://papers.nips.cc/paper/8246-using-trusted-data-to-train-deep-networks-on-labels-corrupted-by-severe-noise.pdf.

Hoiem, Derek, Yodsawalai Chodpathumwan, and Qieyun Dai (2012). “Diagnosing Error in Object Detectors”. In: *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*. ECCV’12. Florence, Italy: Springer-Verlag, pp. 340–353. ISBN: 978-3-642-33711-6. DOI: 10.1007/978-3-642-33712-3_25. URL: http://dx.doi.org/10.1007/978-3-642-33712-3_25.

Huang, Chen, Yining Li, Chen Change Loy, and Xiaoou Tang (2016). “Learning Deep Representation for Imbalanced Classification”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 5375–5384. DOI: 10.1109/CVPR.2016.580. URL: https://doi.org/10.1109/CVPR.2016.580.

Huber, Peter J. (Mar. 1964). “Robust estimation of a location parameter”. In: *Annals of Mathematical Statistics* 35.1, pp. 73–101. ISSN: 0003-4851. DOI: 10.1214/aoms/1177703732. URL: http://dx.doi.org/10.1214/aoms/1177703732.

Johnson, Sam and Mark Everingham (2010). “Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation”. In: *BMVC*.

– (2011). “Learning Effective Human Pose Estimation from Inaccurate Annotation”. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Ke, Lipeng, Ming-Ching Chang, Honggang Qi, and Siwei Lyu (2018). “Multi-Scale Structure-Aware Network for Human Pose Estimation”. In: *Proc. ECCV*.

Krizhevsky, Alex (2009). *Learning multiple layers of features from tiny images*. Tech. rep.

Lin, Tsung-Yi, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár (2017). “Focal Loss for Dense Object Detection”. In: *Proc. IEEE ICCV*.

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg (2015). “SSD: Single Shot MultiBox Detector”. In: *Proc. ECCV*.

Liu, Weiyang, Yandong Wen, Zhiding Yu, and Meng Yang (2016). “Large-Margin Softmax Loss for Convolutional Neural Networks”. In: *ICML*.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). “Stacked Hourglass Networks for Human Pose Estimation”. In: *Proc. ECCV*.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017). “Automatic differentiation in PyTorch”. In: *NIPS-W*.

Ren, Mengye, Wenyuan Zeng, Bin Yang, and Raquel Urtasun (2018). "Learning to Reweight Examples for Robust Deep Learning". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 4331–4340. URL: `http://proceedings.mlr.press/v80/ren18a.html`.

Ronchi, Matteo Ruggero and Pietro Perona (2017). "Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation". In: *Proc. IEEE ICCV*.

Shelhamer, Evan, Jonathan Long, and Trevor Darrell (2017). "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE TPAMI* 39.4, pp. 640–651.

Shrivastava, Abhinav, Abhinav Gupta, and Ross B. Girshick (2016). "Training Region-Based Object Detectors with Online Hard Example Mining". In: *Proc. IEEE CVPR*.

Sung, Kah Kay (1996). "Learning and Example Selection for Object and Pattern Detection". AAI0800657. PhD thesis. Cambridge, MA, USA.

Tang, Wei, Pei Yu, and Ying Wu (Sept. 2018). "Deeply Learned Compositional Models for Human Pose Estimation". In: *The European Conference on Computer Vision (ECCV)*.

Tompson, Jonathan, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler (2015). "Efficient object localization using Convolutional Networks". In: *CVPR*. IEEE Computer Society, pp. 648–656.

Wang, Jingdong, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, D. Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao (2021). "Deep High-Resolution Representation Learning for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, pp. 3349–3364.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional Pose Machines". In: *Proc. IEEE CVPR*.

Yang, Wei, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang (2017). "Learning Feature Pyramids for Human Pose Estimation". In: *Proc. IEEE ICCV*.

Zhang, Min-Ling and Zhi-Hua Zhou (Oct. 2006). "Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization". In: *IEEE Trans. on Knowl. and Data Eng.* 18.10, pp. 1338–1351. ISSN: 1041-4347. DOI: `10.1109/TKDE.2006.162`. URL: `http://dx.doi.org/10.1109/TKDE.2006.162`.

Zhang, Tong (2004). "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms". In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: ACM, pp. 116–. ISBN: 1-58113-838-5. DOI: `10.1145/1015330.1015332`. URL: `http://doi.acm.org/10.1145/1015330.1015332`.

Zhang, Zhilu and Mert Sabuncu (2018). “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels”. In: *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 8778–8788. URL: `http://papers.nips.cc/paper/8094-generalized-cross-entropy-loss-for-training-deep-neural-networks-with-noisy-labels.pdf`.

Zhou, Zhi-Hua and Xu-Ying Liu (Feb. 2006). “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *Knowledge and Data Engineering, IEEE Transactions on* 18, pp. 63–77. DOI: `10.1109/TKDE.2006.17`.

Chapter 3

MULTI-PERSON POSE ESTIMATION: PARSING POSE OF PEOPLE WITH INTERACTION

The content of this chapter is from the peer-reviewed publication "Parsing Pose of People with Interaction" by S. Ryou and P. Perona, appearing at BMVC 2018.

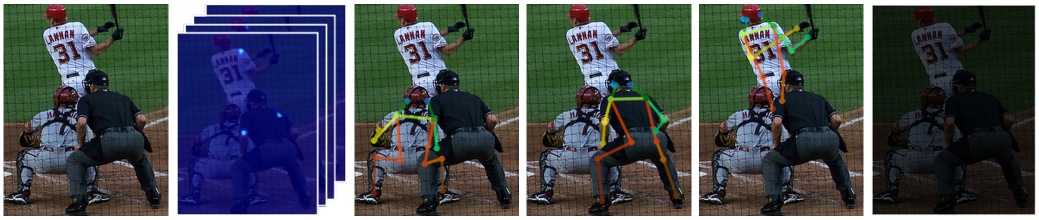
In Chapter 3, we propose a novel architecture for the multi-person pose estimation problem in the scenes with people having interaction.

3.1 Abstract

We propose an end-to-end multi-person pose estimation model that learns to predict keypoint locations for each person in the scene, regardless of the complexity of their social interactions. While recent multi-person pose estimation algorithms achieve high performance on scenes where people do not overlap, these algorithms produce undesired outcomes, *e.g.*, merging two people or swapping similar parts of different people, when the people in the scene are heavily occluded. To attack this issue, we have curated a subset of COCO (Lin, Maire, et al., 2014) containing such scenes and call it COCO-crowd. We formulate multi-person pose estimation as a sequential prediction problem that first generates heatmaps of the potential part locations and then assembles the parts into separate instances, each representing a single person, using convolutional LSTMs. Despite using a small-scale dataset (relative to all of COCO), we achieved comparable performance to state-of-the-art methods trained on the full COCO dataset. We also evaluate our method on the Immediacy dataset (Chu, Ouyang, et al., 2015), which consists of images with diverse social interactions, *e.g.*, standing shoulder to shoulder or hugging, and achieve state-of-the-art results.

3.2 Introduction

Pose estimation, localizing joint locations in an input image, is an important building block for high level computer vision tasks such as human action recognition (C. Wang, Y. Wang, and A. L. Yuille, 2013), human re-identification (Zhao et al., 2017), and proxemics inference (Chu, Ouyang, et al., 2015; Y. Yang, Baker, et al., 2012). Multi-person pose estimation focuses on predicting a distinct keypoint skeleton for each person in an input image. Recent multi-person pose estimation methods produce promising results with deep convolutional neural networks and



(a) Input (b) Heatmaps (c) Corresponding ordered outputs of the model

Figure 3.1: An overview of our system. For an input image (a), part localizer (Section 3.5) produces keypoint heatmaps without identity (b). Person decoding module (Section 3.5) sequentially produces instance keypoint heatmaps with distinct person identities. The network ends the prediction with generating the all-zero heatmap which represents no more instances on the scene (c).

large-scale datasets (Lin, Maire, et al., 2014). These methods are categorized into 1) *top-down approaches* (Papandreou et al., 2017; Fang et al., 2017; Y. Chen et al., 2017; He, Gkioxari, et al., 2017) that independently run single-person pose estimation algorithms subsequent to human detection results, and 2) *bottom-up approaches* (Cao et al., 2017; Newell, Huang, and Deng, 2017; Insafutdinov et al., 2016; Pishchulin, Insafutdinov, et al., 2016) that group the estimated joint locations into instances representing individual people.

Although current pose algorithms work well for scenes with minimal occlusion, it is still a challenging problem to cluster the correct parts in cluttered scenes. In scenes with human interaction, multi-person pose estimation becomes a challenging task, as body parts are often partially occluded and/or intertwined. These scenarios have been identified as a challenge by the pose-estimation community, and methods have been suggested to improve performance. For instance, the winning entry of 2017 COCO keypoint challenge (Y. Chen et al., 2017) (a top-down approach) defines "hard" keypoints and performs a refinement process on the initial prediction in difficult cases. Bottom-up approaches have suggested methods that attempt to learn additional cues for the succeeding inference step, *i.e.*, pairwise terms (Insafutdinov et al., 2016), identity embedding (Newell, Huang, and Deng, 2017), or part affinity field (Cao et al., 2017). However, these methods ignore the possibility that multiple people can have the same part located at the same image coordinate, which is more likely in highly crowded scenes.

Ronchi and Perona (Ronchi and Perona, 2017) provided in-depth analysis of the performance drops on pose estimation algorithms; they concluded that state of the art methods vastly underperform in crowd scenes, because parts are mostly occluded due to overlapped instances. In addition, since crowd scenarios are rarely found in COCO, which is the most commonly used pose estimation dataset, the community has suffered from scarcity of appropriate training data for developing accurate multi-person pose estimation algorithms.

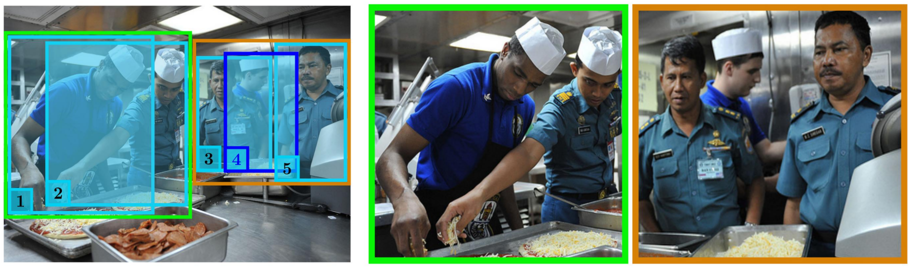
In this paper, we aim to develop a multi-person pose estimation algorithm that is able to decouple human poses despite considerable overlaps between interacting instances. COCO dataset has non-overlapping instance bias, which we discuss in section 3.4. To overcome this bias, we analyze the COCO keypoint dataset and extract all images which show overlapped instances. Also, we revisit the Immediacy dataset (Chu, Ouyang, et al., 2015) which contains images with significant overlap between people. We propose a single-pipeline framework that is trainable in a fully end-to-end fashion. Unlike (Newell, Huang, and Deng, 2017), our method directly renders the final instance heatmaps so that an additional inference step is unnecessary. We let the network have the global encoding of the entire scene and sequentially recognize individual instances to improve the overall pose estimation performance in crowd scenarios. Storing the memory of the entire scene and the histories of the instances, we empower the network to handle occluded parts in the overlapped areas. Figure 3.1 illustrates an overview of our system.

We summarize the main contributions of this paper as follows:

- We tackle a challenging problem in multi-person pose estimation that deals with the severe overlaps arisen from human interactions; and
- We achieve comparable performance with the state-of-the-art methods despite training with significantly less data.

3.3 Related Work

Single-Person Pose Estimation. Earlier approaches in pose estimation (Y. Yang and Deva Ramanan, 2013; X. Chen and A. Yuille, 2014; Pishchulin, Micha Andriluka, et al., 2013; Kiefel and Gehler, 2014) employ graphical models, where each node represents a keypoint and each edge encodes limb information. Deformable Part Models (DPM) (Felzenszwalb, Mcallester, and D. Ramanan, 2008) decompose objects into parts and use spatial relations among the parts to build computationally tractable inference steps. With the advent of deep convolutional neural networks (DCNN), researchers began to apply DCNNs for keypoint feature extraction and limb



(a) COCO

(b) COCO-Crowd

Figure 3.2: Crowd Extraction Procedure. The left image (a) shows an original COCO image and annotations, and the two images on the right (b) show the corresponding images in COCO-crowd. We define the notion of "overlap" when the intersection of union (IoU) score of two bounding boxes is greater than 0.1 ([1,2], [3,4], and [4,5] pairs on left image). All interlinked boxes ([1,2] and [3,4,5]) are merged into proposals for the crowd region. Meanwhile, we discard non-crowd regions, box without an overlap.

representation. Chen *et al* (X. Chen and A. Yuille, 2014) define limb configurations using pairwise clusters of adjacent keypoints, and employ a DCNN to extract the unary and pairwise scores for each keypoint. A single unified model (Tompson et al., 2014) was proposed to combine a DCNN part detector with a spatial model enforcing implicit constraints on the body parts.

Bulat *et al* (Bulat and Tzimiropoulos, 2016) proposed a cascaded CNN architecture that performs regression on the first predicted part heatmap. Several approaches (Newell, K. Yang, and Deng, 2016; Wei et al., 2016; Carreira et al., 2016) exploit iterative refinements and show significant improvement. In particular, the stacked hourglass network (Newell, K. Yang, and Deng, 2016) consists of repeating multi-scale modules and performs sequential refinements to capture complex spatial relationships. Chu *et al* (Chu, W. Yang, et al., 2017) exploited attention mechanisms at multiple resolutions and applied Conditional Random Fields (CRF) to model the correlations in neighboring regions. Yang *et al* (W. Yang et al., 2017) proposed a pyramid residual module on skip connections of the hourglass block to learn multi-scale features.

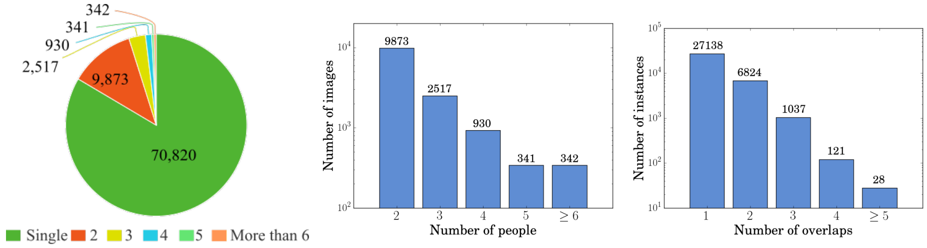
Multi-person Pose Estimation. Current multi-person pose estimation methods can be classified into two main categories: *top-down approaches* (Papandreou et al., 2017; Fang et al., 2017; Y. Chen et al., 2017; He, Gkioxari, et al., 2017) and *bottom-up approaches* (Newell, Huang, and Deng, 2017; Cao et al., 2017; Insafutdinov et al., 2016). Top-down approaches first detect candidate human bounding boxes, then run a single-person pose estimation algorithm on each box. Papandreou *et al* (Papandreou et al., 2017) followed this two-step pipeline with Faster-RCNN (Ren et al., 2015) as a human detector and fully convolutional ResNet (He, Zhang, et al., 2016) as a pose estimator. Fang *et al* (Fang et al., 2017) proposed a symmetric spatial transformer network to produce a high quality single-person region. Mask-RCNN (He, Gkioxari, et al., 2017) proposed a framework for both instance segmentation and pose estimation by predicting an object mask and keypoint locations in parallel with the existing branch for bounding box recognition.

On the other hand, bottom-up approaches first predict part locations, then assemble the parts into distinct people. Pishchulin *et al* (Pishchulin, Insafutdinov, et al., 2016) proposed a partitioning and labeling formulation based on the CNN part detectors and Integer Linear Programming (ILP). DeeperCut (Insafutdinov et al., 2016) extended this work by incorporating image-conditioned pairwise probabilities that consider body part configurations into the deep network. Cao *et al* (Cao et al., 2017) exploited a two-stage pipeline, which first generated part heatmaps and part affinity fields along the limbs, and then assigned part identity through a bipartite graph matching algorithm. Newell *et al* (Newell, Huang, and Deng, 2017) proposed an end-to-end system which directly output part identity tags along with the part locations.

Recurrent Model with Spatial Sequence Prediction. Our work formulates multi-person pose estimation as a sequential problem using spatial variants of recurrent neural networks. Gkioxari *et al* (Gkioxari, Toshev, and Jaitly, 2016) adopted a sequential model for single-person pose estimation by predicting each joint location dependent on the previous output, allowing the network to learn complex body structure. Shi *et al* (Shi et al., 2015) proposed the Convolutional LSTM (ConvLSTM), a convolutional variant of the standard LSTM (Hochreiter and Schmidhuber, 1997), to capture spatiotemporal correlation within precipitation forecasting. Romera-Paredes and Torr (Romera-Paredes and Torr, 2016) proposed a class-specific instance segmentation and counting method by sequentially segmenting one instance of the scene at a time using ConvLSTM.

3.4 COCO-Crowd dataset

Our work focuses on parsing the poses of people in crowd scenes. With this perspective, we analyze the COCO dataset. Previous work on COCO keypoint evaluation Ronchi and Perona, 2017 defines "overlap" between instances if a pairwise



(a) (b) (c)

Figure 3.3: Dataset configuration. After running our crowd extraction system on COCO, we observe that single-instances are dominant on the dataset (a). To overcome this dataset bias, our dataset (named COCO-crowd) consists of images with two or more instances, where its distribution can be seen in (b). To show the complexity of the our curated dataset, we count the number of overlaps arisen from each instance, and provide the distribution in (c).

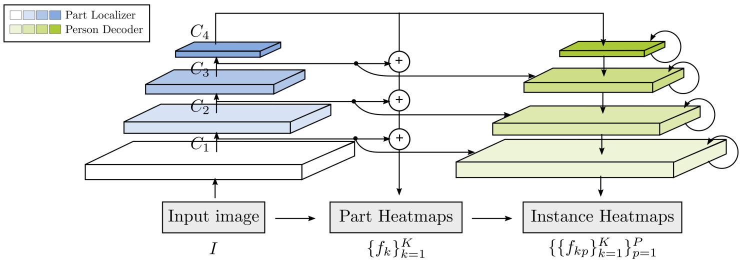


Figure 3.4: Network architecture. The network consists of two parts: part localizer (each blue box representing ResNet-50 convolution block) and person decoder (green boxes corresponding to ConvLSTM block at each resolution). The input image is encoded with part localizer and first predicts part heatmaps. The person decoder decouples this encoded feature into distinct instance heatmaps.

instance shows an intersection over union (IoU) score greater than 0.1. We borrow this definition to extract regions of images that exhibit overlap to use in our dataset. In order to discover these regions, we iterate over all possible pairs of bounding boxes containing a person in each image. If a pair of boxes have IoU≥0.1, then we tag that pair of boxes as a crowd. After all crowd pairs are obtained, we merge all pairs that share at least one common instance into sets. Figure 3.2 describes this process in detail. We also summarize the resulting data distribution in Figure 3.3. Our dataset, named as COCO-crowd, has 14,003 training images containing 35,148 total instances. Test and validation images are also produced by following this procedure on 2014 COCO validation data, which results in 3,336 validation and 3,336 test images.

3.5 Method

Figure 3.4 provides an overview of our system composed of two parts: part localizer and person decoder. The proposed framework is a single pipeline that encodes the input image to predict K keypoint heatmaps using a fully convolutional network, and sequentially produces instance heatmaps using a convolutional recurrent neural network. We use ConvLSTM in order to decode the individual instances. We describe the details of the part prediction in Section 3.5 and explain how we apply ConvLSTM to our framework in Section 3.5.

Part localizer

We use ResNet-50 (He, Zhang, et al., 2016) as our building block for keypoint detection. Similar to (Y. Chen et al., 2017; Lin, Dollár, et al., 2017), we utilize feature pyramid structure to preserve both semantic information and the localization quality. An input image I is encoded with ResNet conv blocks, and transformed into feature maps in different scales as C_1, C_2, C_3, and C_4, respectively (see Figure 3.4). We apply 1×1 kernel convolution to match the dimension of the all feature maps to 64. Then, we resize and sum these feature maps to produce the final part heatmap. We apply a sigmoid to the summed feature maps. The output of the part localization module has the form of K heatmaps, each representing a single part location, with an output stride of 4. We denote the final output heatmap as $f_k(x_i)$, where k represents the k-th keypoint (out of K) and $x_i \in \{1, \dots, N\}$ represents the index of 2D pixel location.

Person decoder

We model multi-person pose estimation as a sequential prediction problem with variable length of output. In our problem setting, the model should keep track of the number of people, and individuate an instance from a set of human candidates. Since pose estimation requires high localization quality, we adopt a spatial variant on LSTM, ConvLSTM. To preserve multi-resolution information, we apply ConvLSTM units at every scale encoded from the part localization step.

The architecture of the person decoding module is displayed in Figure 3.5. The person decoder consists of a chain of ConvLSTMs at every scale. All of the ConvLSTM kernels are 3×3. The features C_1, C_2, C_3, and C_4 generated from the part localization module are fed through all the subsequent recurrent stages to prevent the network from forgetting keypoint information. In particular, we halve the dimensionality of part features by applying one convolutional block of ResNet,

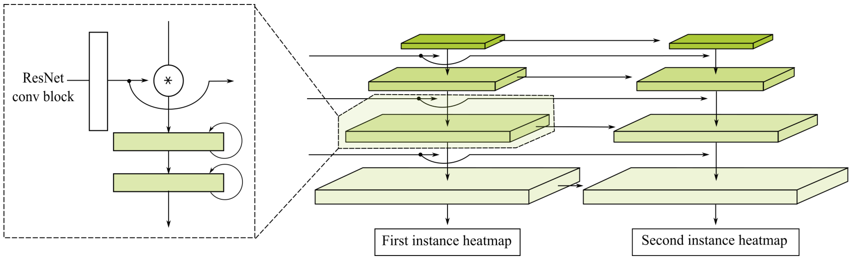


Figure 3.5: Person Decoding Module. The green block represents two ConvLSTM layers at each resolution. The recurrent block on each scale is composed of two stacked ConvLSTMs.

which we explain the details in the supplementary material. These features are concatenated to the input for each ConvLSTM block. We employ two stacked ConvLSTM layers for each scale block, so that the output from the first ConvLSTM acts as an input to the second unit. After passing the ConvLSTM block, features are upsampled by 2, and the final output of person decoder has an output stride of 4, producing K keypoint heatmaps of the target person. We denote the output as $f_{kp}(x_i)$ where $p \in \{1, \ldots, P\}$ represents p-th person over the total number of people P in an input image. In the same manner as the part localization step, we apply a sigmoid on top of the outputs. When the network finishes prediction, it is trained to output an all-zero heatmap. We provide the details of the person decoder in the supplementary material.

Loss function

In our experimental setting, the network first predicts candidate part locations and finally produces a heatmap for each instance. Thus, the loss function consists of two parts.

Part localization. Let x_i be a 2D location on the image, where $i \in \{1, \ldots, N\}$ indexing the pixel locations. For each part type $k \in \{1, \ldots, K\}$, we denote $h_k(x_i)$ as the k-th keypoint heatmap at location x_i. The ground truth heatmap $h_k(x_i) = 1$, when $\|x_i - y\| \leq R$ for $y \in \{y_{k0}, y_{k1}, \ldots, y_{kP}\}$, each y_{kp} representing part-k location of the p-th person over all P people, and zero otherwise. In our experiments, we set $R = 3$ pixels. This part heatmap encodes all part locations without identity. We apply pixelwise binary cross entropy to the output of the part localization module with $h_k(x_i)$. The part localization loss is as follows:

$$\mathcal{L}_{part} = \frac{1}{NK} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathcal{L}_{BCE}(f_k(x_i), h_k(x_i)), \tag{3.1}$$

where $\mathcal{L}_{BCE}$ denotes pixelwise binary cross entropy.

Person decoder. Let $h_{kp}(x_i)$ be the k-th keypoint heatmap of p-th person at location x_i. The keypoint heatmap $h_{kp}(x_i) = 1$ when $\|x_i - y_{kp}\| \leq R$ with y_{kp} ground truth location of part-k of p-th person, and zero otherwise. The network produces set of keypoint heatmaps at each step, encoding part locations of each person. In order to make the network decide the order in which to predict each instance, we use the Hungarian algorithm (Kuhn and Yaw, 1955), as in (Stewart, Mykhaylo Andriluka, and Ng, 2016; Romera-Paredes and Torr, 2016). Given a cost matrix, the Hungarian algorithm finds an optimal matching between the output and the target heatmaps and re-orders the target heatmaps in a matched order. We construct our cost matrix by computing binary cross entropy for each prediction-target pair. Given the re-ordered heatmaps from the Hungarian algorithm, we again apply binary cross entropy in order to compute our loss. We additionally apply loss for the following two steps, as in (Romera-Paredes and Torr, 2016), with zero heatmaps, so that the network learns the stop criterion.

$$\mathcal{L}_{person} = \frac{1}{NK(P+2)} \sum_{p=1}^{P+2} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathcal{L}_{BCE}(H(f_{kp}(x_i), h_{kp}(x_i))), \tag{3.2}$$

where $H(\cdot)$ denotes the Hungarian algorithm, which returns the re-ordered target and input. The final loss is as follows:

$$\mathcal{L} = \lambda_0 \, \mathcal{L}_{part} + \mathcal{L}_{person}, \tag{3.3}$$

where $\lambda_0 = 0.5$ is a hyperparameter which controls the relative importance of two terms.

3.6 Experimental Results

Experimental setup

Training Setup. We have implemented our system in PyTorch. We optimize eq. (3.3) with Adam and train for 130 epochs. For COCO-crowd, the learning rate is set to 1e-3 and is decayed by 0.1 at epoch 60 and 90, respectively. With the same initial setting, the learning rate is dropped by 0.1 at 40 and 60 for the Immediacy dataset. We use a batch size of 32 on 8 GPUs for COCO-crowd, whereas a batch size of 12 on a single GPU for Immediacy. For the part encoding backbone (*i.e.*, ResNet), we employ the initial weights pretrained on ImageNet (Deng et al., 2009). The input size is set to 512×512. We augment the data with random flips, rotations (±40°), and scalings on the fly. When training the model on COCO-crowd, we use the

Method	AP	AP.5	AP.75	AP M	AP L	AR	AR.5	AR.75	AR M	AR L
Mask-RCNN	0.364	0.598	0.362	0.371	0.398	0.497	0.706	0.519	0.505	0.528
CMU-pose	0.365	0.599	0.369	0.367	0.378	0.418	0.628	0.429	0.422	0.438
AE	**0.438**	0.664	**0.456**	**0.440**	0.451	**0.532**	0.740	**0.560**	**0.538**	**0.554**
AE*	0.396	0.663	0.402	0.409	0.420	0.486	0.728	0.507	0.493	0.515
Ours	0.433	**0.709**	0.447	**0.440**	**0.454**	0.520	**0.761**	0.549	0.526	0.545

Table 3.1: Results (AP) on COCO-crowd. Mask-RCNN is tested using Detectron (Girshick et al., 2018) and all other methods are tested using the code and pretrained models the authors provide. Testing is held on single scale on all bottom-up methods. To see the impact of the amount of data, we also trained associative-embedding (Newell, Huang, and Deng, 2017) on COCO-crowd (AE*).

corresponding original COCO images for the scale augmentation to contain various backgrounds. The cropped box is enlarged when the scaling factor is greater than 1.0.

We follow the curriculum learning scheme used in (Bengio et al., 2009; Romera-Paredes and Torr, 2016) by gradually increasing number of people after the loss converges. Therefore, the network learns to predict at most M instances in iteration M, even when more instances are present. In our experiments, we train the network to predict at most two people until convergence, then increase the maximum number of people by 1 every two epochs. For COCO-crowd, the loss is masked to avoid penalizing instances without annotation.

Testing Setup. Testing is performed on a single scale with both the original and a flipped version of each image. If the maximum value of the heatmap is less than the threshold (0.05), the network produces all-zero heatmap to stop the prediction.

Evaluation

COCO-crowd

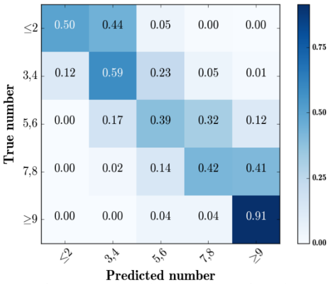
COCO keypoint dataset has 17 keypoint labels of 5 facial landmarks (nose, left/right ear and eye) and 12 body parts (left/right shoulder, elbow, wrist, hip, knee, and ankle). We report the performance with three different algorithms using the official evaluation metric, average precision (AP) and average recall (AR) in Table 3.1. Two bottom-up methods (Newell, Huang, and Deng, 2017; Cao et al., 2017) are tested in a single scale using the code and pretrained models that the authors provide. Mask-RCNN (He, Gkioxari, et al., 2017) is tested using Detectron (Girshick et al., 2018) with the ResNeXt-101 encoding backbone, which showed the highest mAP score among all detectron models.

Method	2	3	4	5	≥ 6
AE	0.503	0.435	0.367	**0.405**	**0.419**
Ours	**0.512**	**0.439**	**0.393**	0.386	0.364

Table 3.2: AP score by number of people

Jittering (px)	± 0	± 5	± 10	± 15
AP	0.433	0.426	0.412	0.392

Table 3.3: AP score by jittering bounding box



	≤2	3,4	5,6	7,8	≥9
≤2	0.50	0.44	0.05	0.00	0.00
3,4	0.12	0.59	0.23	0.05	0.01
5,6	0.00	0.17	0.39	0.32	0.12
7,8	0.00	0.02	0.14	0.42	0.41
≥9	0.00	0.00	0.04	0.04	0.91

Figure 3.6: Counting confusion matrix

With a small amount of data, we outperform two different methods and achieve comparable performance to the state-of-the-art method trained on the full COCO dataset. To see the impact of the amount of data, we also train AE (Newell, Huang, and Deng, 2017) from scratch, using COCO-crowd (AE*). When compared against state-of-the-art methods trained on the same amount of data, our method shows promising results. We also perform an additional experiment to gauge the importance of part localization module. When training without the part localization loss, we observed a huge performance drop, AP score of *0.271* compared to the original score *0.433*.

Counting. To see if our method successfully learns when to stop, we visualize the confusion matrix for the number of predictions and the number of ground truth instances in Figure 4.9. We observe that most of the elements are in diagonal, which implies our method can approximately count the number of people in an image.

Performances over the number of people. We evaluate the score by varying number of people in an image and compare against the state-of-the-art method in Table 3.2. Our method performs better on predicting relatively small number of people. Due to the recurrent architecture, we observed failure cases as the sequence length increases.

Bounding box jittering test. COCO-crowd dataset is composed of the cropped regions for the crowd, thus it requires crowd detections in advance. To show the feasibility of our framework as a full system, we show how our method is robust at bounding box jittering in Table 3.3. While some part locations can be eliminated from the bounding box as jittering, our method faithfully estimates pose of people in the box.

Immediacy Dataset

The Immediacy dataset was originally designed to analyze visual interaction between people. In this dataset people are mostly present in pairs, either holding one another from behind, hugging, holding hands, giving each other a high five or putting arms over each other's shoulders. It contains 7,500 training images and 2,500 testing images. We used 500 images from the training set for validation. The total number of instance is 20,499, each having 12 keypoint labels of upper body (head top, neck, left/right shoulder, left/right elbow, left/right wrist, left/right hand, left/right hip). This dataset is challenging for inferring the arm locations, since social interaction makes significant arm occlusions.

We followed the percentage of correct keypoints (PCK) metric (Y. Yang and Deva Ramanan, 2013) used in the dataset paper (Chu, Ouyang, et al., 2015). PCK measure is for a single-person pose estimation problem, where an estimated body part location is defined to be correct when it falls within $\alpha\ max(height, width)$ pixels. We used $\alpha = 0.2$ as in the original setup of (Chu, Ouyang, et al., 2015). Since the results from the paper are evaluated given the bounding box of ground truth upper body, we match the result to corresponding ground truth and report the mean PCK of matched keypoints, for fair comparison. To show how current methods perform at this dataset, we also test Mask-RCNN (He, Gkioxari, et al., 2017) and report score of the parts in common. Even without using the person location, current methods significantly outperform all previous methods. In particular, our method improves wrist and hand predictions by a wide margin. Even without exhibiting the Immediacy dataset, our model trained on COCO-crowd still shows huge performance gains on wrist compared to Mask-RCNN. We provide qualitative results on both datasets in Figure 3.7.

Method	head	shoulder	elbow	wrist	hand	torso	mean
Yang	69.5	63.0	42.6	31.8	29.0	43.9	47.0
Ouyang	67.7	61.3	46.4	35.4	32.5	48.9	49.0
Chu	82.5	74.6	50.1	38.8	37.1	55.4	56.4
Mask-RCNN	-	81.0	64.3	55.3	-	-	-
Ours (crowd)	-	86.8	65.5	63.7	-	-	-
Ours	**95.6**	**88.8**	**72.4**	**74.0**	**73.3**	**75.1**	**79.9**

Table 3.4: PCK score on Immediacy Dataset.

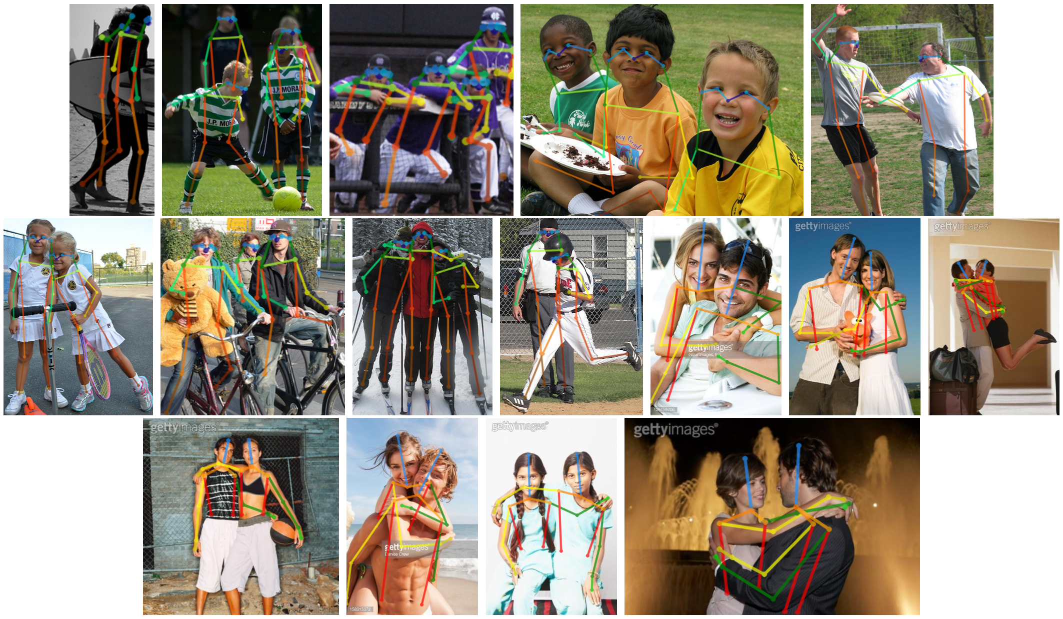


Figure 3.7: Qualitative results. Results containing severe occlusion due to social interaction.

3.7 Conclusions

In this paper, we addressed the problem of pose estimation in crowd scenes and proposed a multi-person pose estimation method which sequentially decouples each instance. We tested our approach with two challenging datasets and showed that the proposed method is able to infer human poses regardless of complex interactions. With considerably small amount of data, our method achieved a comparable performance to the state-of-the-art method trained on the full COCO dataset. Furthermore, we significantly improved the performance on the Immediacy dataset, containing heavily occluded scenes due to social interactions, and produced faithful predictions on the arm locations. We believe our approach to be applicable to general multi-person pose estimation followed by crowd detection.

Acknowledgement

This work was supported by funding from Disney, Inc and Allen E. Puckett Endowment Fund.

References

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum Learning". In: *ICML*.

Bulat, Adrian and Georgios Tzimiropoulos (2016). “Human Pose Estimation via Convolutional Part Heatmap Regression”. In: *Proc. ECCV*.

Cao, Zhe, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2017). “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *Proc. IEEE CVPR*.

Carreira, João, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik (2016). “Human Pose Estimation with Iterative Error Feedback”. In: *Proc. IEEE CVPR*.

Chen, Xianjie and Alan Yuille (2014). “Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations”. In: *NIPS*.

Chen, Yilun, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun (2017). “Cascaded Pyramid Network for Multi-Person Pose Estimation”. In: *CoRR* abs/1711.07319.

Chu, Xiao, Wanli Ouyang, Wei Yang, and Xiaogang Wang (2015). “Multi-task Recurrent Neural Network for Immediacy Prediction”. In: *Proc. IEEE ICCV*.

Chu, Xiao, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang (2017). “Multi-context Attention for Human Pose Estimation”. In: *Proc. IEEE CVPR*.

Deng, Jia, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Fei-Fei Li (2009). “ImageNet: A large-scale hierarchical image database”. In: *Proc. IEEE CVPR*.

Fang, Hao-Shu, Shuqin Xie, Yu-Wing Tai, and Cewu Lu (2017). “RMPE: Regional Multi-person Pose Estimation”. In: *Proc. IEEE ICCV*.

Felzenszwalb, P., D. Mcallester, and D. Ramanan (2008). “A Discriminatively Trained, Multiscale, Deformable Part Model”. In: *Proc. IEEE CVPR*.

Girshick, Ross, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He (2018). *Detectron*. `https://github.com/facebookresearch/detectron`.

Gkioxari, Georgia, Alexander Toshev, and Navdeep Jaitly (2016). “Chained Predictions Using Convolutional Neural Networks”. In: *Proc. ECCV*.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick (2017). “Mask R-CNN”. In: *Proc. IEEE ICCV*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *Proc. IEEE CVPR*.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780.

Insafutdinov, Eldar, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele (2016). “DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model”. In: *Proc. ECCV*.

Kiefel, Martin and Peter Gehler (2014). “Human Pose Estimation with Fields of Parts”. In: *Proc. ECCV*.

Kuhn, H. W. and Bryn Yaw (1955). “The Hungarian method for the assignment problem”. In: *Naval Res. Logist. Quart*, pp. 83–97.

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie (2017). “Feature Pyramid Networks for Object Detection”. In: *Proc. IEEE CVPR*.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). “Microsoft COCO: Common Objects in Context”. In: *Proc. ECCV*.

Newell, Alejandro, Zhiao Huang, and Jia Deng (2017). “Associative Embedding: End-to-End Learning for Joint Detection and Grouping”. In: *NIPS*.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). “Stacked Hourglass Networks for Human Pose Estimation”. In: *Proc. ECCV*.

Papandreou, George, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Christoph Bregler, and Kevin Murphy (2017). “Towards Accurate Multi-person Pose Estimation in the Wild”. In: *Proc. IEEE CVPR*.

Pishchulin, Leonid, Micha Andriluka, Peter Gehler, and Bernt Schiele (2013). “Poselet conditioned pictorial structures”. In: *Proc. IEEE CVPR*.

Pishchulin, Leonid, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele (2016). “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation”. In: *Proc. IEEE CVPR*.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *NIPS*.

Romera-Paredes, Bernardino and Philip Hilaire Sean Torr (2016). “Recurrent Instance Segmentation”. In: *Proc. ECCV*.

Ronchi, Matteo Ruggero and Pietro Perona (2017). “Benchmarking and Error Diagnosis in Multi-Instance Pose Estimation”. In: *Proc. IEEE ICCV*.

Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo (2015). “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *NIPS*.

Stewart, Russell, Mykhaylo Andriluka, and Andrew Y. Ng (2016). “End-to-End People Detection in Crowded Scenes”. In: *Proc. IEEE CVPR*.

Tompson, Jonathan J, Arjun Jain, Yann LeCun, and Christoph Bregler (2014). “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”. In: *NIPS*.

Wang, Chunyu, Yizhou Wang, and Alan L. Yuille (2013). “An Approach to Pose-Based Action Recognition.” In: *Proc. IEEE CVPR*.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). “Convolutional Pose Machines”. In: *Proc. IEEE CVPR*.

Yang, Wei, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang (2017). “Learning Feature Pyramids for Human Pose Estimation”. In: *Proc. IEEE ICCV*.

Yang, Yi, Simon Baker, Anitha Kannan, and Deva Ramanan (2012). “Recognizing proxemics in personal photos”. In: *Proc. IEEE CVPR*.

Yang, Yi and Deva Ramanan (2013). “Articulated Human Detection with Flexible Mixtures of Parts.” In: *IEEE TPAMI* 35.

Zhao, Haiyu, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang (2017). “Spindle Net: Person Re-identification with Human Body Region Guided Feature Decomposition and Fusion”. In: *Proc. IEEE CVPR*.

Chapter 4

WEAKLY SUPERVISED KEYPOINT DISCOVERY

The content of this chapter is from the manuscript "Weakly Supervised Keypoint Discovery" by S. Ryou and P. Perona 2021.

In Chapter 4, we investigate the method to automatically discover the keypoints without requiring human annotation.

4.1 Abstract

In this paper, we propose a method for keypoint discovery from a 2D image using image-level supervision. Recent works on unsupervised keypoint discovery reliably discover keypoints of aligned instances. However, when the target instances have high viewpoint or appearance variation, the discovered keypoints do not match the semantic correspondences over different images. Our work aims to discover keypoints even when the target instances have high viewpoint and appearance variation by using image-level supervision. Motivated by the weakly-supervised learning approach, our method exploits image-level supervision to identify discriminative parts and infer the viewpoint of the target instance. To discover diverse parts, we adopt a conditional image generation approach using a pair of images with structural deformation. Finally, we enforce a viewpoint-based equivariance constraint using the keypoints from the image-level supervision to resolve the spatial correlation problem that consistently appears in the images taken from various viewpoints. Our approach achieves state-of-the-art performance for the task of keypoint estimation on the limited supervision scenarios. Furthermore, the discovered keypoints are directly applicable to downstream tasks without requiring any keypoint labels.

4.2 Introduction

Keypoints are a convenient intermediate representation towards final tasks, such as action recognition (Li et al., 2020), fine-grained classification (Branson et al., 2014; Guo and Farrell, 2019), face identification (Xie, Shen, and Zisserman, 2018), and person re-identification (Su et al., 2017; Zhao et al., 2017). However, collecting keypoint annotations is labor-intensive and time-consuming compared to the image-level or bounding box annotations. Recently, unsupervised keypoint discovery (Jakab et al., 2018; Jakab et al., 2020; Lorenz et al., 2019; Y. Zhang et al.,

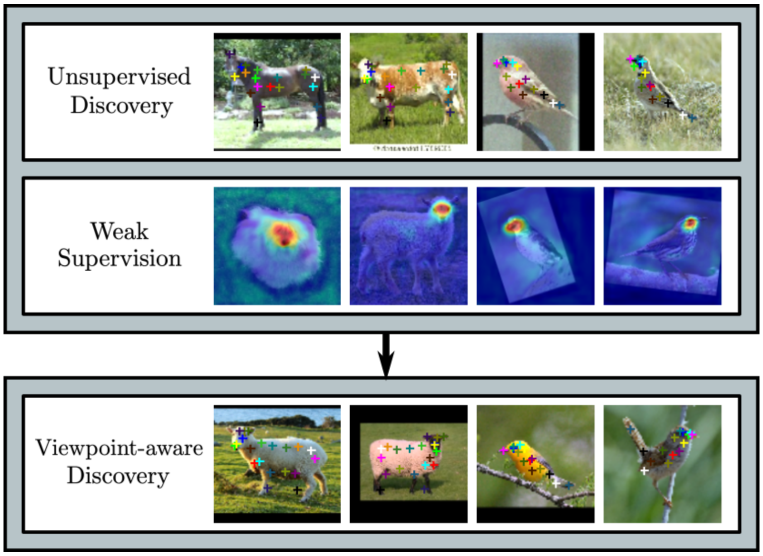


Figure 4.1: Overview of weakly supervised keypoint discovery: Our approach uses the unsupervised method to discover diverse keypoints and image-level supervision to localize the discriminative parts. By using the keypoints learned from weak-supervision to infer the viewpoint of a target instance, our model can successfully discover semantically consistent parts for instances facing in different directions.

2018) has been proposed to reduce the annotation effort and has shown successful results for the images with aligned instances and humans with a mostly upright pose. However, these methods struggle to find consistent keypoints when the target objects have severe viewpoint and shape variation (See Figure 5.4). On the other hand, weakly-supervised learning methods on object localization (Jie et al., 2017; Zhou et al., 2016; Oquab et al., 2015; Singh and Lee, 2017; P. Tang et al., 2019; Chong Wang et al., 2014) easily identify the discriminative parts of the target object by using the features trained from the deep neural networks with class labels. In this work, we propose a weakly-supervised keypoint discovery method by exploiting the image-level supervision to guide the network to discover discriminative parts and a viewpoint. To discover diverse keypoints from the target instance, our method adopts the unsupervised methods (Jakab et al., 2018; Jakab et al., 2020; Lorenz et al., 2019; Y. Zhang et al., 2018) which are based on the image reconstruction method, conditioning on the structural bottleneck. Figure 4.1 illustrates the overview of our method.

Most of the current unsupervised keypoint discovery approaches (Jakab et al., 2018; Jakab et al., 2020; Lorenz et al., 2019; Y. Zhang et al., 2018) share the idea of disentangling the representation of appearance and structure from an input image; here, the structure is represented as a set of keypoints. Given a pair of source and target images, where the target image is generated by applying a structural transformation to the source image, these works (Jakab et al., 2018; Jakab et al., 2020) extract a keypoint information from a target image and the appearance representation from a source image. With the appearance feature from the source image, the network is trained to reconstruct the target image by using the keypoint bottleneck computed for its structural representation. These methods automatically discover semantically meaningful parts for the images with aligned instances (*e.g.*, images only contain the same species of animal or the viewpoint is restricted). However, empirical results show that when the instance has high viewpoint variation, the model fails to find semantically consistent parts (Fig. 5.4). Specifically, animals usually have diverse poses with high appearance and viewpoint variation. The discovered keypoints from animal images show a high correlation on the spatial coordinates and lose the semantic correspondence across different images.

On the other hand, weakly-supervised learning methods on object localization (Jie et al., 2017; Zhou et al., 2016; Oquab et al., 2015; Singh and Lee, 2017; P. Tang et al., 2019; Chong Wang et al., 2014) easily identify the most discriminative parts, while suffering from localizing only the dominant region (*e.g.*, face of animal). Our method exploits this idea to discover the discriminative parts when the target instance has a large viewpoint and shape variation. We use the part-based representation by extracting the local features from the discovered keypoint locations and train these features to predict the image-level labels. The parts discovered from this process are simultaneously used for the unsupervised image reconstruction task as well.

After adopting these two approaches, we observe that the discovered keypoints still show a high spatial correlation between different parts that have a similar appearance, *e.g.*, parts from the torso or front and back legs (Fig. 4.11). To resolve this issue, we propose a viewpoint-based equivariance constraint, where the keypoint representation should move according to the structural deformation. Unlike using the equivariance constraint on all pairs of images (Lorenz et al., 2019; Y. Zhang et al., 2018), our method applies this constraint only to the viewpoint-augmented images. We use discovered parts from the image-level supervision to infer the viewpoint of an instance and enforce the equivariance constraint based on the model prediction.

We evaluate our method in various experimental settings. To compare with existing keypoint discovery methods, we first test on datasets with small viewpoint variation, *e.g.*, facial keypoint, and animals with a consistent viewpoint. Moreover, we demonstrate the robustness of our method to a large viewpoint and appearance variation by applying it to challenging datasets that include diverse species of animals. When trained with datasets with large shape diversity, our model can handle high appearance variation and discover the keypoints from the images with unseen categories. For both cases, our method achieves state-of-the-art performance in the limited supervision scenarios. Finally, we analyze the distribution of the discovered keypoints and demonstrate its representation power by applying it to a simple behavior classification task.

4.3 Related Work

Our goal is to build a keypoint discovery model which is robust across viewpoints by incorporating information from pose estimation, weakly-supervised learning, and unsupervised keypoint discovery.

Keypoint Estimation. Keypoint estimation is a problem of localizing a predefined set of keypoints from an input image. Pose is a convenient intermediate representation for various applications. Applications range from human pose estimation (Chen et al., 2017; Newell, Yang, and Deng, 2016; Ryou, Jeong, and Perona, 2019; W. Tang, Yu, and Wu, 2018; Wei et al., 2016), facial landmark detection (Belhumeur et al., 2011; Burgos-Artizzu, Perona, and Dollár, 2013; X. Cao et al., 2014; Z. Zhang et al., 2014) to animal pose estimation (Branson et al., 2014; Guo and Farrell, 2019; Mathis et al., 2018). With the development of fully convolutional neural networks (Shelhamer, Long, and Darrell, 2017), the pose estimation community gained huge success by estimating the part locations using a heatmap, where each location is encoded as a 2D gaussian map centered at each body part location. Specifically, most of the existing approaches exploited iterative refining steps (Newell, Yang, and Deng, 2016; Wei et al., 2016), multi-scale information (Chen et al., 2017), and learning signals (Chen et al., 2017; Ryou, Jeong, and Perona, 2019) for further improvement.

Unsupervised Keypoint Discovery. Despite the success in estimating the part location with supervision, one of the major drawbacks is that it requires a huge amount of annotations. Recently, methods to discover the landmarks without manual annotation emerged with the shared idea of image reconstruction with encoder-decoder architecture by adopting the geometry bottleneck (Jakab et al., 2018; Lorenz et al., 2019; Y. Zhang et al., 2018). Jakab *et al* (Jakab et al., 2018) used the discovered keypoints as a geometry bottleneck for conditional image generation. Zhang *et al* (Y. Zhang et al., 2018) proposed an autoencoder-based architecture by explicitly using the feature representation from the discovered keypoints. Lorenz *et al* (Lorenz et al., 2019) disentangled appearance and structure representation by exploiting the shape and appearance transform separately. Jakab *et al* (Jakab et al., 2020) incorporated prior knowledge about the pose of the target by using unpaired keypoint data from existing datasets to discover the keypoints for other datasets within the same domain. Our work does not require any prior knowledge about the structure of the target instance and tackles a more challenging problem where the target instance has a large viewpoint and shape variation.

Weakly-supervised Learning. Weakly-supervised learning methods have been adopted for various vision tasks including object localization (Jie et al., 2017; Oquab et al., 2015; Singh and Lee, 2017; P. Tang et al., 2019; Chong Wang et al., 2014), semantic segmentation (Huang et al., 2018; Pathak, Krähenbühl, and Darrell, 2015), and semantic matching (Novotný, Larlus, and Vedaldi, 2017). Zhou *et al* (Zhou et al., 2016) proposed Class Activation Map (CAM) for object localization only with the class labels and demonstrated that the image-level supervision gives a cue to find the most discriminative region of the objects. At the same time, weakly-supervised methods struggle from predicting only the dominant parts rather than the entire object. While previous work aims to resolve this issue by manipulating the image patches (Singh and Lee, 2017) or iteratively refining the classifiers (Jie et al., 2017; P. Tang et al., 2019), our work exploits this idea to discover the consistent discriminative parts.

Part-based Representation. Part-based features have been useful representations for many computer vision applications, especially for the tasks of disambiguating the marginal visual differences: fine-grained classification (Branson et al., 2014; Guo and Farrell, 2019; Sun et al., 2018; J. Zhang et al., 2019) and facial identification (Xie, Shen, and Zisserman, 2018). Fine-grained image classification works use keypoint information either by explicitly estimating the keypoint locations using the ground truth (Branson et al., 2014; Guo and Farrell, 2019) or implicitly discovering the parts (Sun et al., 2018; J. Zhang et al., 2019). While the latter works also

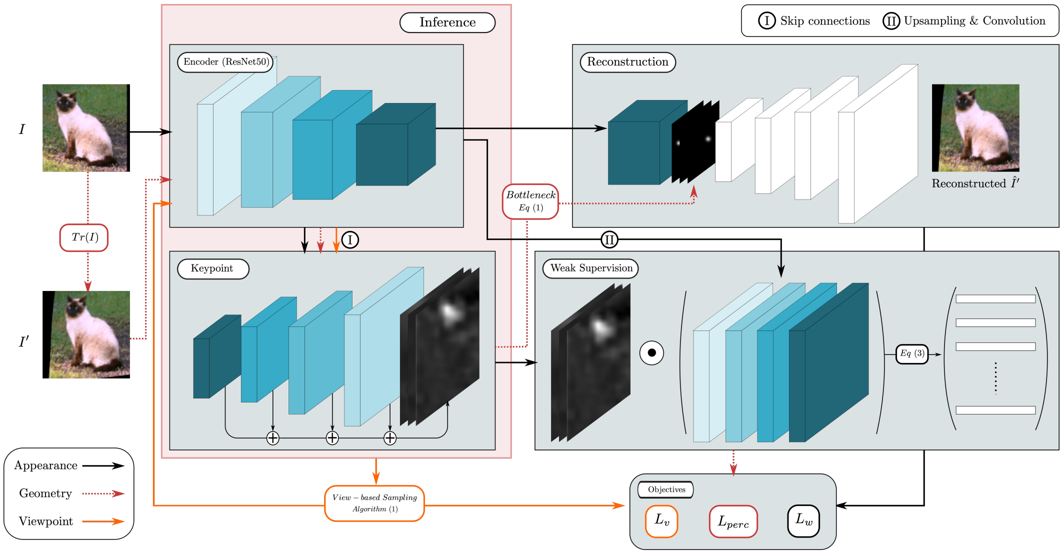


Figure 4.2: System Outline: Our model uses the shared encoder for various tasks: image reconstruction, keypoint estimation, and classification. The bluish same block color represents the features generated from the shared weights of ResNet-50. Black, red, and orange arrows represent the appearance, geometry, and viewpoint stream, respectively. Target image I' is generated by applying structural transformation Tr to the source image I. The concatenation of the appearance feature from I and the geometry bottleneck from I' is used to reconstruct the target image $\hat{I}'$. Discovered keypoint heatmaps and the appearance features from I are used to generate part-based features (Sec. 4.4 and Fig. 4.3).

automatically learn the parts, the keypoints are the byproduct of the final task, thus they do not measure the semantic consistency over different images. On the other hand, the goal of our work is to discover the keypoints which are consistent over different images and species.

Recent works on action recognition (Du, W. Wang, and L. Wang, 2015; Li et al., 2020; Chunyu Wang, Y. Wang, and Yuille, 2013) use only the coordinate-based representation as an input to the activity classification. Our work shows potential for the discovered keypoints to be used as an input representation to the simple behavior classification tasks.

4.4 Method

Our work is based on 1) weakly-supervised learning by detecting coarse parts with image-level supervision and 2) unsupervised learning by discovering fine parts with an unsupervised reconstruction module. The overall pipeline of our method is shown in Figure 5.2. In this section, we explain the architecture of each module.

Unsupervised Keypoint Discovery

In our experimental setting, we use ResNet-50 (He et al., 2016) as a backbone for keypoint discovery, image reconstruction, and the weak-supervision modules. The convolution feature blocks $\{C_1, C_2, C_3, C_4\}$ from each resolution of ResNet-50 are used throughout all modules.

Given a pair of source and target images (I, I'), where the target image is generated by applying structural transformation to the source image, the model learns to reproduce the target image by using the appearance representation from the source image and the geometry information from the target image. The dotted red line in Fig 5.2 represents the pipeline for the unsupervised keypoint discovery.

Specifically, the source image I and the transformed target image I' are fed to the shared image encoder. Here, we use Thin Spline Transformation (TPS) (Duchon, 1977; Wahba, 1990) as a transformation function to generate a target image (See I' in Fig. 5.2). The encoder generates the feature representations $\{C_1, C_2, C_3, C_4\}$ and $\{C'_1, C'_2, C'_3, C'_4\}$ for the source and target images, respectively. The feature C_4 from the source image is used as an appearance feature and the features generated from the target image are used to extract the geometry information from the keypoint module.

Keypoint module. We use the GlobalNet architecture from Cascaded Pyramid Network (CPN) (Chen et al., 2017), which exploits multi-scale features from ResNet-50 (He et al., 2016), for the keypoint discovery module. From each resolution of convolutional blocks $\{C'_1, C'_2, C'_3, C'_4\}$ from the target image I', the network generates heatmaps and upsamples them to the final output size. The sum of the heatmaps from each resolution becomes the final heatmap, which is used for generating a geometry bottleneck.

Image reconstruction. The geometry information from the target image should capture the structural differences from the source image. We use the discovered keypoints as a geometry bottleneck by generating a gaussian heatmap. From the predicted raw heatmaps of the keypoint module, we apply spatial softmax to each channel and use these normalized heatmaps H_k to compute the weighted sum over x, y coordinates to get the $p_k = (u_k, v_k)$ locations for $k = \{1, \ldots, K\}$ keypoints. To explicitly localize the target part, we generate 2D Gaussian heatmaps centered on the keypoint locations and these heatmaps become a structure bottleneck B_k for the target instance:

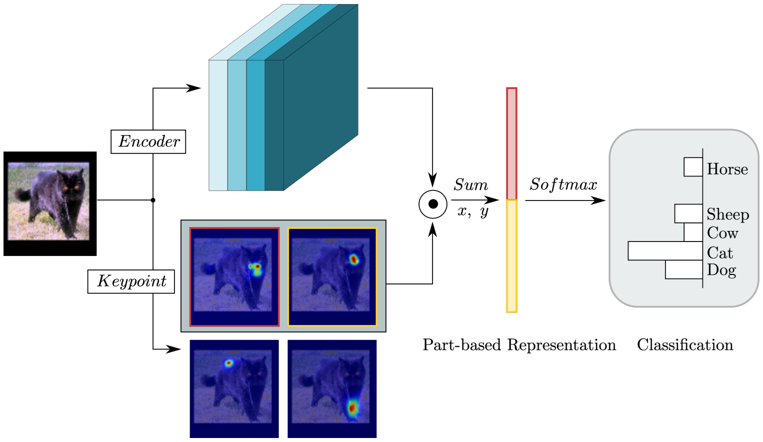


Figure 4.3: Weak-supervision module: We use the first K_w keypoints to extract part-based representation from the base features. Concatenated part-based features are fed to predict the image-level supervision task.

$$B_k(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\|\mathbf{x} - p_k\|^2}{2\sigma^2}\right). \tag{4.1}$$

The concatenation of the last feature from the encoded source image C_4 and the structure bottleneck B_k from the target image becomes an input to the reconstruction module, which consists of convolution and upsampling layers. We feed geometry bottleneck to each resolution of the reconstruction module. The architecture details about this module are in the Appendix 4.8. For the image reconstruction, we use perceptual loss L_{perc} (Johnson, Alahi, and Fei-Fei, 2016), which compares the features computed from VGG (Simonyan and Zisserman, 2014) network f with the target I' and the reconstructed images $\hat{I}'$.

$$L_{perc} = \sum_i \left\| f(I') - f(\hat{I}'; B(I'))) \right\|_2 \tag{4.2}$$

Weak Supervision

We extract the part-based representation from the features obtained by the shared encoder. In order to combine multi-resolution information of the image, we upsample features from each convolutional block $\{C_1, C_2, C_3, C_4\}$ to the final heatmap size and apply few convolution layers to reduce the feature dimension. These feature blocks C become the base representation for the image-level supervision tasks. Then, we

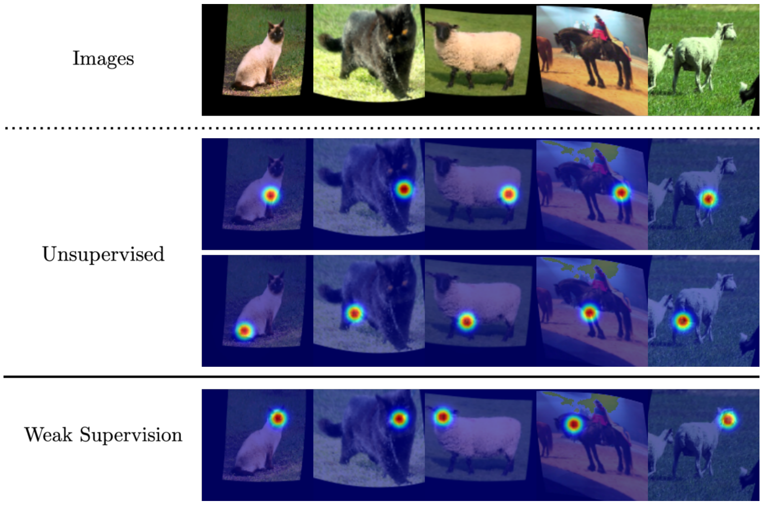


Figure 4.4: Spatial correlation between similar parts. Each row represents the keypoint bottleneck generated from the same keypoint channel. Ideally, heatmaps from the same row should represent the same semantic part. Keypoints discovered from the unsupervised module predict similar spatial locations for look-alike body parts regardless of the viewpoint of the animal. On the other hand, keypoints discovered from the weak-supervision module consistently find discriminative parts like the face.

generate a keypoint-based representation by applying the Hadamard product to the base features and the discovered keypoint heatmap:

$$h_k = \sum_i \sum_j H_k(i,j) \odot C(i,j). \tag{4.3}$$

Here we use the $K_w < K$ number of heatmaps to extract features for the target task. Figure 4.3 illustrates this process. Each vector represents a localized feature for each keypoint. The concatenated vectors are fed to the final fully connected layer. Here we apply cross-entropy for the classification task as our weak supervision loss:

$$L_w = -\sum_i^N y_i \log \hat{y}_i. \tag{4.4}$$

Viewpoint-based Equivariance

We observe that the discovered keypoints have a spatial correlation on the parts that have a similar appearance, except for the discriminative parts that are tied to the weak supervision module. Figure 4.4 shows that the keypoints discovered from the

```
Algorithm 1: Viewpoint-based Equivariance
Input: Images $I$ in a mini-batch at iteration t; Keypoint network $\phi(I)$
Result: Images for training the equivariance loss $I^v$; Corresponding keypoint
        coordinates $p^*$
for $i = 1, ..., N$ do
    i. Discover the keypoints: $(u_i^k, v_i^k) = \phi(I_i)$ for $k = \{1, ..., K\}$.
    ii. Compute the $x$-variance of the discovered keypoints: $s_i = \frac{\sum(u_i^k - \bar{u}_i)^2}{K-1}$.
end
Sample one-side facing images using $S = \{s_1, ...s_N\}$
1. Sort S in descending order and choose $N_s$ images with high $x$ variance
2. For $N_s$ images, compute the mean of $u_i$ for $k = \{1, \ldots, K_w\}$: $\mu_i = \frac{\sum u_i^k}{K_w}$
3. Pick $N_v$ samples facing the same direction by sorting $\mu = \{\mu_1, ...\mu_{N_s}\}$.
4. Generate the view-augmented images $I^v$ by flipping $N_v$ images
5. For $N_v$ images, set equivariance label $p^*$ by flipping the discovered
   keypoints.
```

torso and legs are predicting similar spatial locations although the animals are facing in opposite directions. In order to resolve this issue, we propose an equivariance constraint based on the model predictions from the same viewpoint to generate the keypoint labels for the opposite viewpoint by data augmentation. Note that we use “viewpoint” as the direction where the target instances are facing. Unlike previous works using equivariance constraint on the pair of images (Lorenz et al., 2019; Y. Zhang et al., 2018) with small deformation, our work applies it to viewpoint-based data augmentation like mirrored version of the image.

Sampling based on weak supervision. To generate the labels for the viewpoint-augmented image, we have to sample images which share the same viewpoint. This sampling process involves the model prediction and the keypoints discovered from the image-level supervision task to infer the facing direction of a target instance. The procedure of sampling and training is explained in Algorithm 1.

For the samples obtained by this procedure, we apply MSE loss L_v as an equivariance constraint:

$$L_v = \frac{1}{N_v} \sum_{i}^{N_v} \|\phi(I_i^v) - p_i^*\|_2. \tag{4.5}$$

Loss

The final objective function is composed of three parts: the perceptual loss L_{perc} for image reconstruction, image-level supervision loss L_w, and viewpoint-based equivariance loss L_v. Since the equivariance constraint depends on the model prediction, we adopt curriculum learning (Bengio et al., 2009) for training L_v loss. The hyperparameter settings are in the Appendix 4.8.

$$L = w_p L_{perc} + w_w L_w + w_v L_v \mathbb{1}_{\{epoch>n\}} \tag{4.6}$$

4.5 Experiments

In this section, we evaluate our method on keypoint estimation and downstream tasks. First, we show the qualitative results with discovered keypoints on various datasets. To quantitatively measure the performance of our method, we evaluate two different experimental settings: linear regression and finetuning. Secondly, we analyze our model output by showing the distribution of the predictions and visualize the pose embedding. We also show the performance on the weak supervision task of fine-grained classification. Finally, we demonstrate the efficacy of our discovered keypoints by directly predicting simple animal behaviors from the discovered keypoints without any keypoint label.

Datasets

We conduct experiments on various datasets with large viewpoint and appearance variations. To compare with existing unsupervised methods, we run the experiments on the images with a consistent viewpoint (CelebA, CUB) and images with various viewpoints (CUB, AnimalPose, StanfordDogs). In addition, we test our method by applying the discovered keypoints to a simple activity prediction task (DogPart, TigDog). We briefly explain each dataset here.

CelebA (Liu et al., 2015) is a dataset of 200k facial images with 10k identities. We follow the same training and testing split of (Lorenz et al., 2019; Y. Zhang et al., 2018), which excludes the train and test set from MAFL. We train the linear regressor for 5 keypoints using the MAFL training set 19k images and test on 1k MAFL test set. Since this dataset has a fixed viewpoint with small appearance variation, we only use the image reconstruction loss for training this dataset.

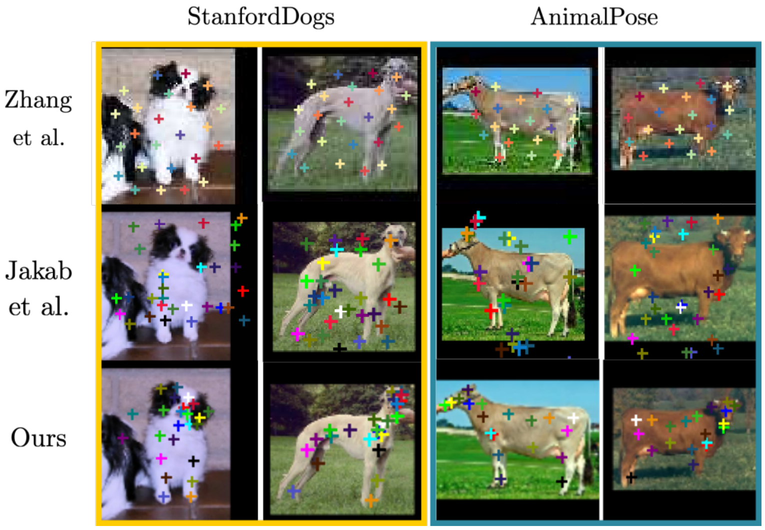


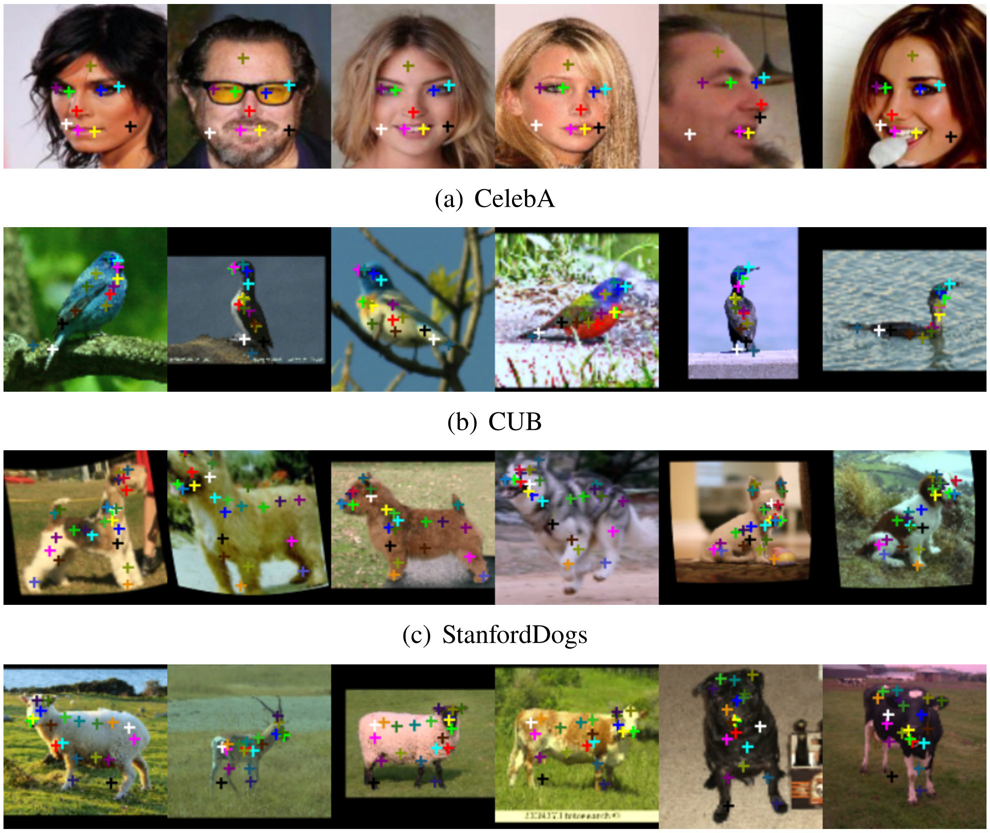
Figure 4.5: Discovered keypoints on the AnimalPose dataset using existing methods. Unsupervised models either predict similar locations regardless of the semantic parts and viewpoint of an instance or fail to discover semantically consistent parts.

CUB (Welinder et al., 2010) is a dataset of fine-grained classification of the bird species with 200 categories and 15 keypoint labels. We test on two different settings with the CUB dataset. First, to compare with the unsupervised methods (Lorenz et al., 2019), we exclude the seabird species and align the parity using the visibility of the eye landmark. In addition, we test on the full dataset including the images with all species and various viewpoints by finetuning the keypoint network.

StanfordDogs (Khosla et al., 2011) is a dataset for fine-grained classification of 120 dog species with 20k images. Recently, the StanfordExtra dataset (Biggs et al., 2020) has been released with silhouette and 24 keypoint labels. We evaluate the keypoint performance on the StanfordExtra dataset.

Animal Pose (J. Cao et al., 2019) is a dataset for a cross-domain adaptation task with 12 different species of animals. This dataset contains bounding box annotations for 7 animal categories and the pose labels for 5 different species in a total of 6k instances in 4k images. 20 keypoints are labeled for the animals with the pose label.

DogPart (Barnard et al., 2016) is a dataset for automatic animal behavior classification. This dataset is composed of 10 videos taken from a zoo or indoor environment and each frame is labeled with 3 different posture-based action categories: standing, sitting, and lying. In our experiments, we extract the frames that have keypoint and action labels and loosely crop the bounding box area, which brings to a total number of 1k images.



(a) CelebA

(b) CUB

(c) StanfordDogs

(d) AnimalPose

Figure 4.6: Qualitative results on various datasets. Our model successfully discovers semantically consistent parts for images with large viewpoint and appearance variations.

TigDog (Del Pero et al., 2015) is a dataset for behavior analysis. We use the subset of the action categories that can be identified by each frame: standing, sitting, and rolling for horse images. The process to extract the images is in the appendix and the curated dataset for our experimental setup has around 2k images.

Implementation Details. We set the input image size to 128x128 and the number of keypoints for the weakly supervised task to 5 for all experiments. Our model is based on pretrained model on ImageNet (Deng et al., 2009). For the restricted setting, we do not apply the viewpoint-equivariance loss since there is no viewpoint change in the dataset. We adopt curriculum learning for the full dataset experiments. Hyperparameter settings are in the supplementary material. We discover the same number of keypoints provided by each dataset unless otherwise specified.

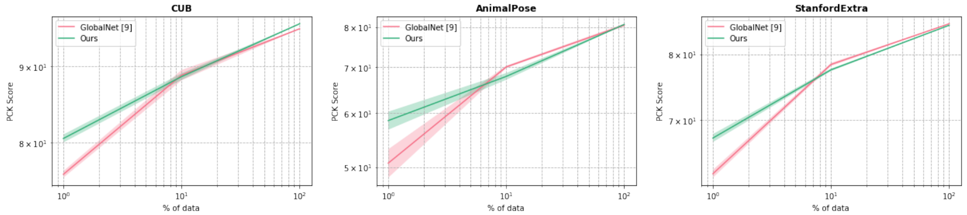


Figure 4.7: PCK score by finetuning the keypoint network with a different amount of supervision on CUB, AnimalPose, and StanfordExtra datasets. Despite the marginal performance differences after using 10% of supervision, the representation learned from our method gives better performance when there is an extremely limited amount of supervision.

Table 4.1: Keypoint estimation performance on the restricted setting. We train a linear regressor from the discovered keypoints for MAFL and CUB datasets. For CUB experiments, we follow the same data extraction step from the paper (Lorenz et al., 2019) and show the performance with %-MSE normalized by an edge length of the image.

Dataset K	MAFL 10	CUB 10
Thewlis (Thewlis, Bilen, and Vedaldi, 2017)	6.32	-
Jakab (Jakab et al., 2018)	3.19	-
Zhang (Y. Zhang et al., 2018)	3.46	5.36
Lorenz (Lorenz et al., 2019)	3.24	3.91
Ours	**2.66**	**3.77**

Keypoint Estimation

Figure 5.4 and 4.6 show qualitative results on the discovered keypoints with zero keypoint annotation compared to the result from existing methods (Fig 5.4). Our method can successfully discover keypoints when the target instances have large viewpoint variations. To quantitatively measure the performance of our method, we follow the same evaluation protocol from previous unsupervised works (Jakab et al., 2018; Lorenz et al., 2019; Y. Zhang et al., 2018) by learning a linear regressor from the model prediction to keypoint annotations for the viewpoint-constrained datasets. For animal datasets, a simple linear regressor cannot capture the relation between the prediction and the annotations due to significant viewpoint changes across the images. Thus, we finetune the keypoint network and evaluate the keypoint estimation performance by varying the number of keypoint annotations with the supervised models.

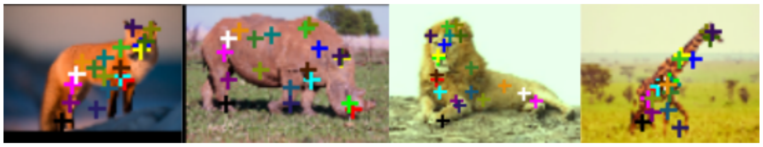


Figure 4.8: Qualitative results for images with unseen categories

Restricted setting. Table 4.1 shows the performance of keypoint estimation for the restricted setting. We use inter-ocular distance (IOD) error as a metric for MAFL and edge distance normalized error for CUB. Although our model uses the features learned from image-level supervision, our method shows the state-of-the-art performance on both datasets compared with the unsupervised methods.

Full dataset. To show the sample efficiency of the representation from our model, we finetune the keypoint estimation network by varying the amount of keypoint annotation with 1%, 10%, and 100%. We use the Percentage of Correct Keypoints (PCK) metric, which defines correct prediction if the distance between the ground truth and the prediction is within $\alpha = 0.1$ with respect to the bounding box size. Figure 4.7 shows the average PCK score over 3 different runs with supervised baseline GlobalNet (Chen et al., 2017), which is the same architecture for our keypoint module. Although the performance reaches almost the same after using 10% of the data, our model shows better performances when there is an extremely limited amount of supervision.

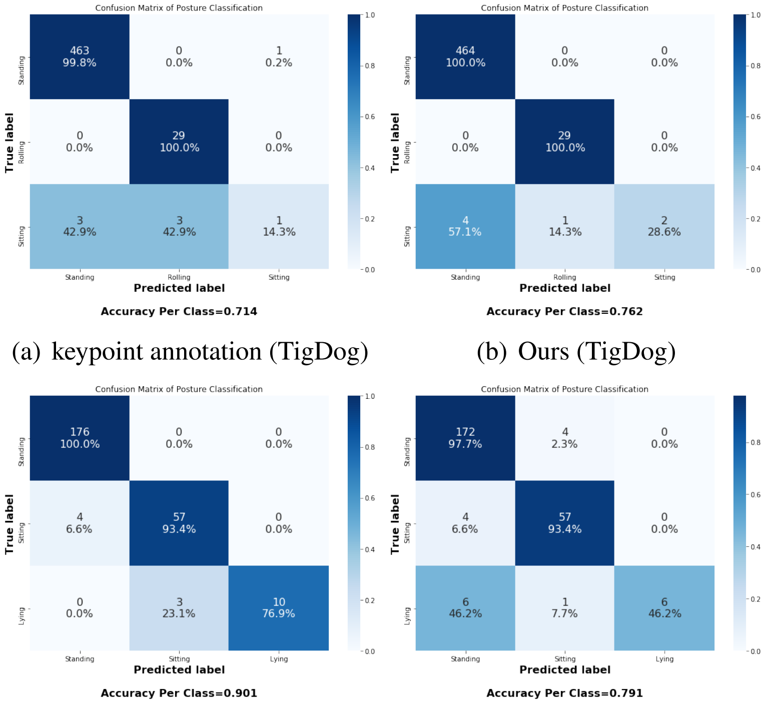
Keypoint discovery from unseen categories. We show qualitative results on the animals from unseen categories in Figure 4.8 using the model trained with the AnimalPose dataset. Since the species in AnimalPose contain diverse animals, our model can handle animals with various appearances across the species. We test on fox, rhino, lion, and giraffe images, which never appeared in the training dataset, and observe consistent part discovery across different species.

Downstream Tasks

We show the performance of the fine-grained classification, which is the weak-supervision task, in Table 4.2. Since the goal of our method is to discover keypoints, this representation does not necessarily give the performance gain on all tasks. However, fine-grained classification on the CUB dataset shows improvement over the baseline, which is trained with the size of 128x128 images on ResNet-50 (He et al., 2016).

Table 4.2: Performance on weak-supervision task (fine-grained classification) with our baseline ResNet-50 (He et al., 2016) with an image size of 128x128

Method	Dataset	Accuracy
ResNet-50 (He et al., 2016) (our baseline)	CUB	67.9
Ours	-	**68.9**
ResNet-50 (He et al., 2016) (our baseline)	StanfordDogs	**71.5**
Ours	-	69.7



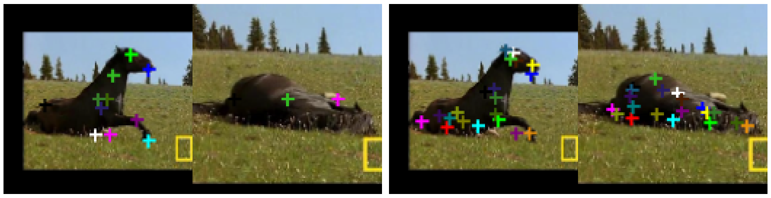
(a) keypoint annotation (TigDog)

(b) Ours (TigDog)

(c) Keypoint annotation (DogPart)

(d) Ours (DogPart)

Figure 4.9: Confusion matrix and per-class accuracy for posture-based action prediction of TigDog and DogPart datasets.



(a) Keypoint annotation

(b) Ours

Figure 4.10: Ground truth annotation and discovered keypoints for sitting and lying actions from TigDog dataset.

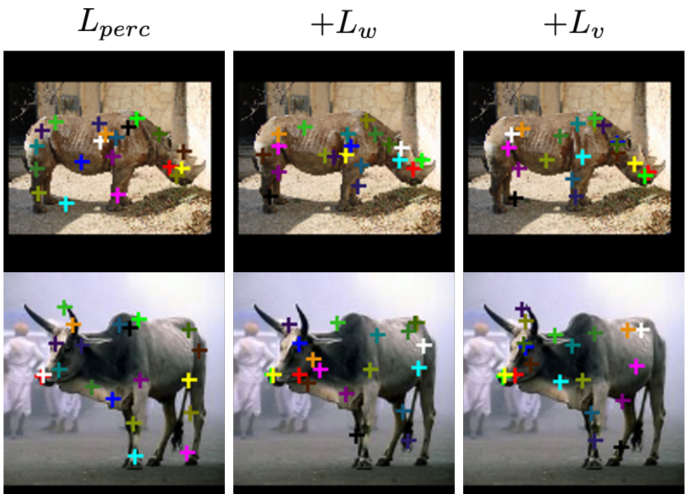


Figure 4.11: Qualitative result for loss ablation study.

Posture-based activity prediction. To demonstrate the representation power of the discovered keypoints, we directly apply the discovered keypoints, without any keypoint labels, to a simple behavior classification task (*e.g.*sitting, standing, and lying) for two different datasets. We train two fully-connected layers with an input of the keypoint locations. For these experiments, we did not train the keypoint discovery model for each dataset due to the limited size of curated datasets. We used the model trained using AnimalPose (J. Cao et al., 2019) to TigDog (Del Pero et al., 2015) and StanfordDogs (Biggs et al., 2020; Khosla et al., 2011) to DogPart (Barnard et al., 2016) experiments, which further demonstrates the generalization ability of our trained models. Note that we do not use the regressed or finetuned keypoints. In Figure 4.9, we provide the behavior classification results trained from human-annotated ground truth keypoints as a baseline, which is expected to be an upper bound performance. Our model achieves comparable performance in most of the categories. Surprisingly, our method on a simplified TigDog behavior task shows better performance. This is due to the scarce keypoint annotations for the occluded parts (Fig. 4.10). Since our model always predicts all the parts around the animal location, it gives more information for the lying or rolling behaviors.

Discussion

Loss ablation study. We show qualitative results for the loss ablation study in Figure 4.11. When the model is trained only with the reconstruction loss $\mathcal{L}_{perc}$, keypoints are discovered around the shape of the target instance without the semantic consistency over different images. While several keypoints are discovered

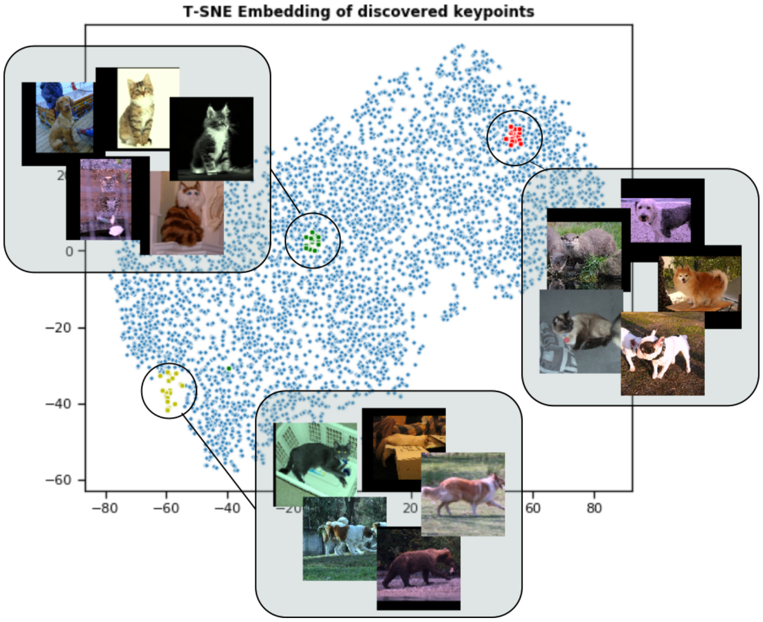


Figure 4.12: Pose embedding for viewpoint-based equivariance

consistently (*e.g.*, red keypoint for facial area, green keypoint for nose) by adopting the weak supervision loss $\mathcal{L}_w$, we observe a spatial correlation problem: violet and turquoise keypoints do not move correspondingly as the animal changes the viewpoint. Finally, the model trained with the full objectives discovers semantically consistent keypoints for the animals having different viewpoints.

Pose embedding. Since our model uses the discovered keypoints to apply a viewpoint-based equivariance constraint, it is important to check whether the model can capture the viewpoint variation. We visualize the T-SNE (Maaten and Hinton, 2008) embedding of the discovered keypoints from AnimalPose (J. Cao et al., 2019) dataset and the corresponding images from three different random locations. Embedding based on the discovered keypoints shows a high correlation with the viewpoint.

Appearance and geometry factorization. Our model uses image reconstruction to discover the keypoints. Although image generation is not a primary goal of our method, our model can manipulate images with a huge viewpoint and appearance variation. We visualize the generated image given the geometry and the appearance bottleneck in Figure 4.13.

Failure cases. Figure 4.14 shows the failure cases. Since the discovered keypoints heavily depend on the training data distribution, our model struggles to predict

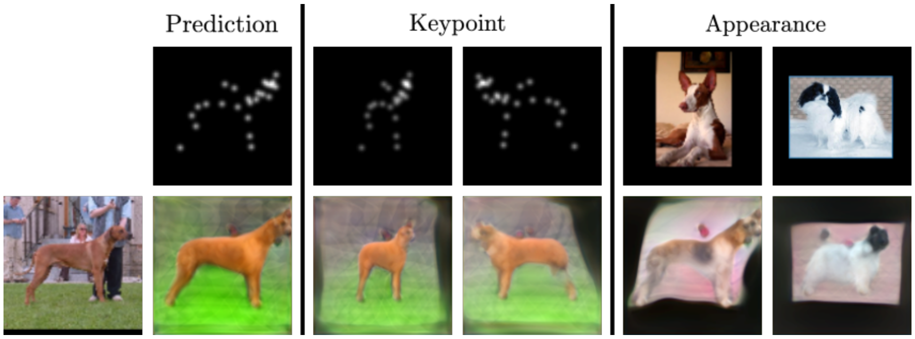


Figure 4.13: Appearance and geometry factorization: We generate the image using the scaled and flipped keypoint as a geometry bottleneck with the input image appearance feature. Given the predicted keypoints from the input, we reconstruct the images using the appearance feature from the top right images.

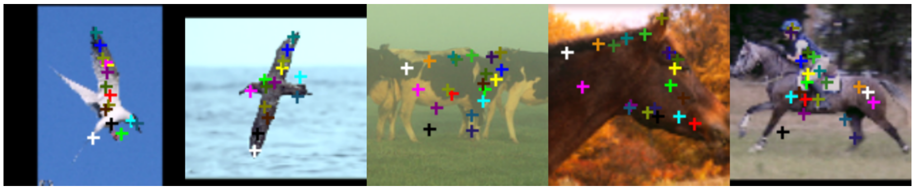


Figure 4.14: Failure cases: Our model fails to discover consistent keypoints for the rare poses and cannot handle the missing or occluded parts. The spatial correlation problem still remains for the images with large appearance variations.

the poses that do not frequently appear in the training data. Also, the model is trained to predict all the keypoint locations without considering the occlusion or missing parts. When the appearance variation is too high, the model fails to predict consistent keypoints.

4.6 Conclusions

We proposed a method to discover the keypoints from the images with various viewpoints by exploiting the weak labels. Our method can successfully discover keypoints when the target instances show large appearance and viewpoint variations. The proposed method has shown strong empirical results for the task of keypoint estimation with a limited amount of supervision. Furthermore, we demonstrated the representation power of our discovered keypoints by running an off-the-shelf model to downstream tasks from different datasets.

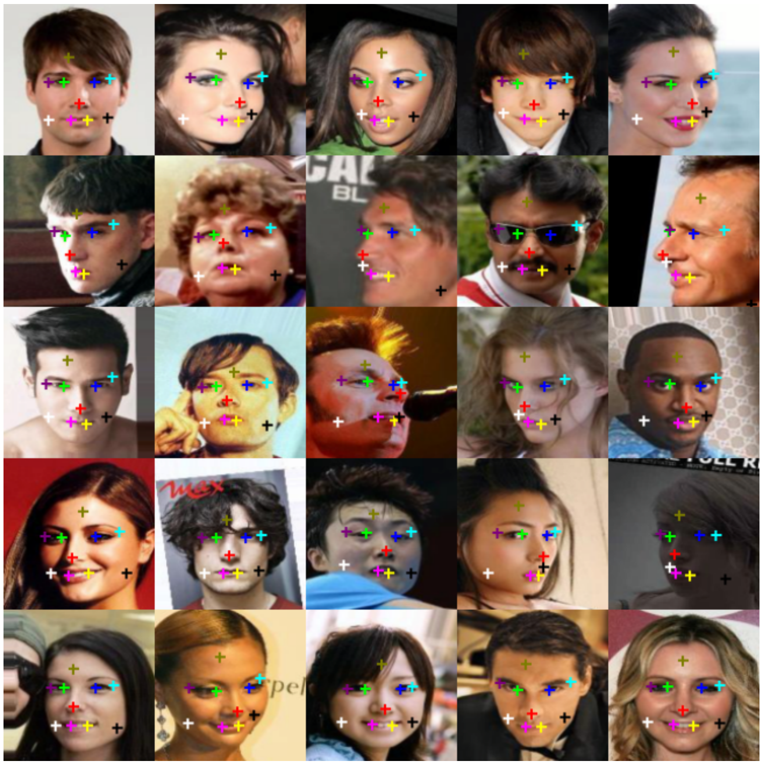


Figure 4.15: Qualitative results on CelebA with 10 discovered keypoints.

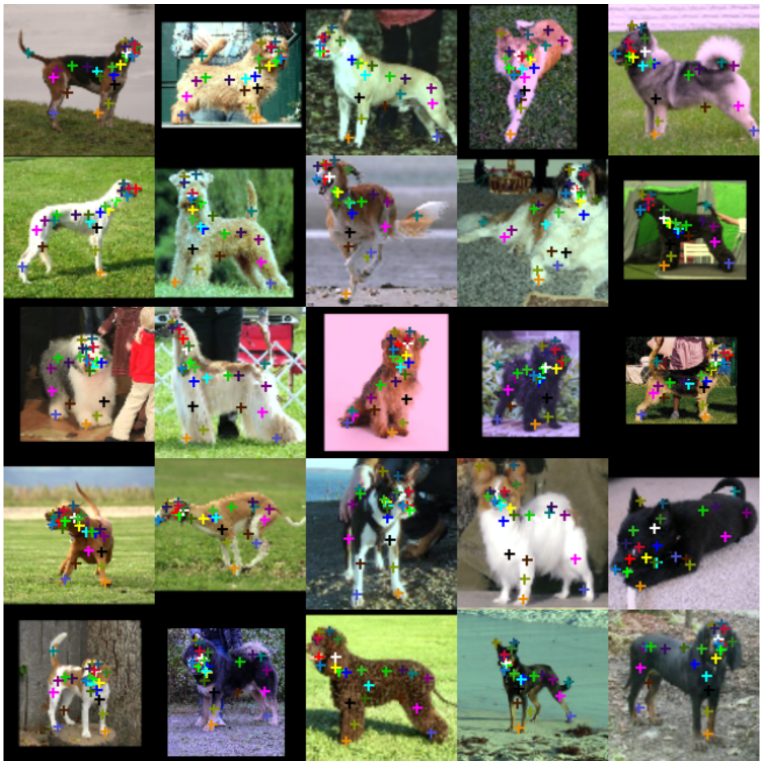


Figure 4.16: Qualitative results on StanfordDogs with 24 discovered keypoints.

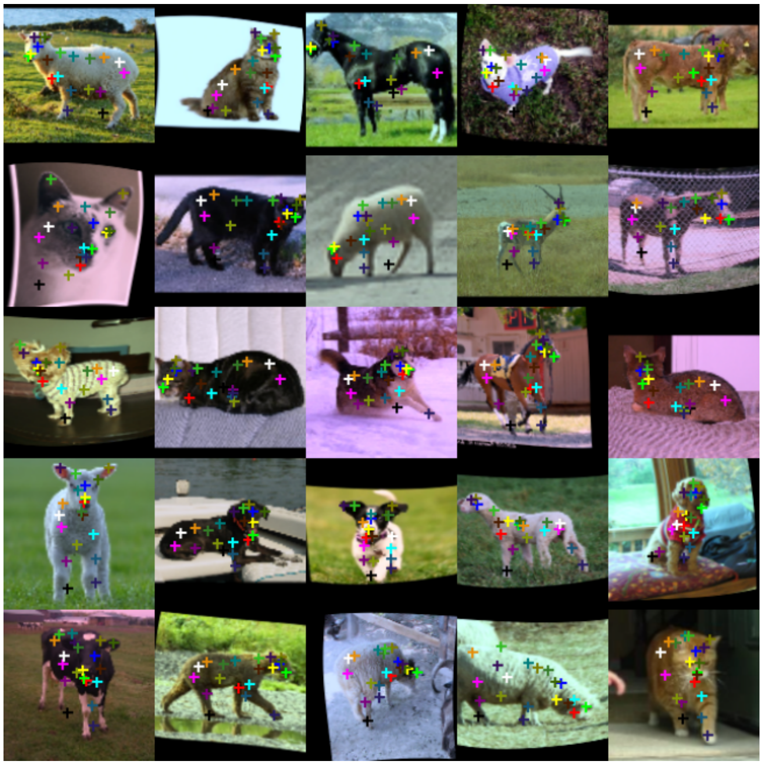


Figure 4.17: Qualitative results on AnimalPose with 20 discovered keypoints.

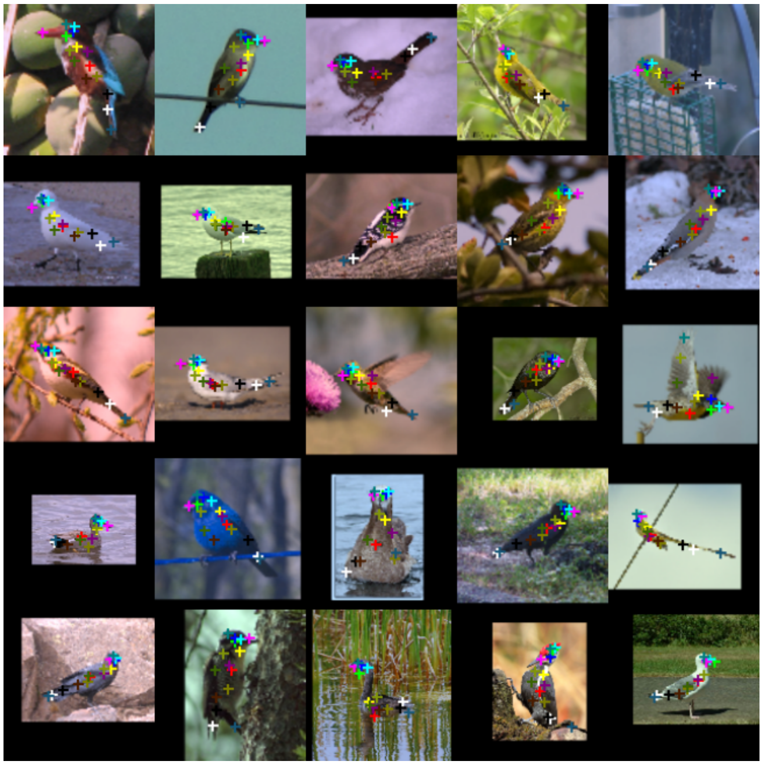


Figure 4.18: Qualitative results on CUB with 15 discovered keypoints.

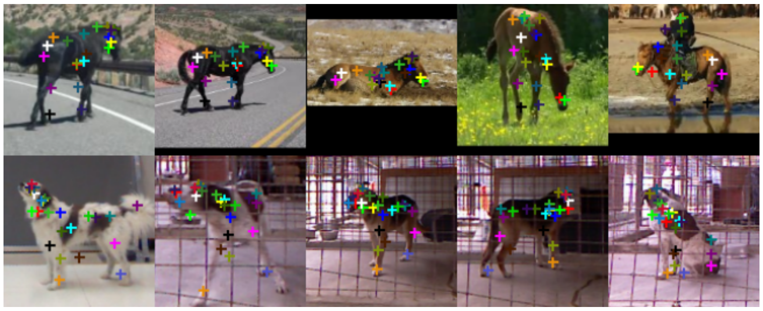


Figure 4.19: Qualitative results on TigDog and DogPart from the model trained on AnimalPose and StanfordDogs, respectively.

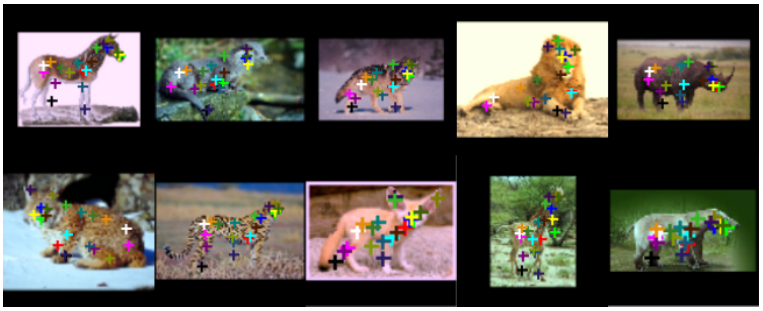


Figure 4.20: Qualitative results on unseen categories from the model trained on AnimalPose.

4.7 Appendix: Qualitative Results

We provide additional qualitative results on the following datasets: CelebA (Fig 4.15), StanfordDogs (Fig 4.16), AnimalPose (Fig 4.17), and CUB (Fig 4.18). To show the robustness of our trained model, we visualize the discovered keypoints on DogPart and TigDog datasets (Fig 4.19), where the model was originally trained on StanfordDogs and AnimalPose datasets, respectively. We also test our method on the images with unseen categories in Fig 4.20. Our method can reliably discover keypoints when the instances have large shape and viewpoint variations. However, since our model is trained to predict all the keypoints without considering the presence of occlusion or missing parts, the discovered keypoints from the cropped images are partially mapped to look-alike parts. Also, the model cannot distinguish the front and the back legs for the images with front-facing animals.

Appearance and geometry factorization

Our model uses image generation to discover the keypoints by disentangling the appearance and geometry features. Though image reconstruction is not a primary goal of our work, our method can generate images with large appearance and

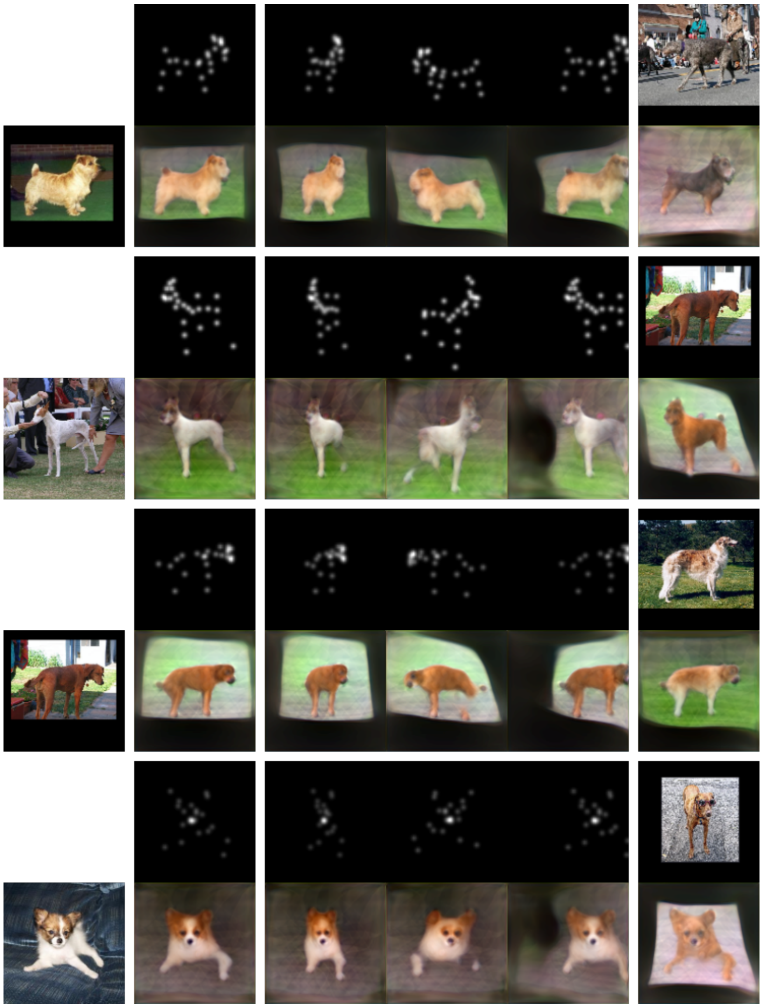


Figure 4.21: Appearance and geometry factorization on StanfordDogs dataset.

viewpoint variations. We visualize the generated images by scaling, flipping, and moving the geometry bottleneck on StanfordDogs (Fig 4.21) and CUB (Fig 4.22) datasets. We also show the generated images using the appearance feature from the top images with the same geometry bottleneck.

4.8 Appendix: Architecture Details

Here we provide the architecture details about the keypoint, reconstruction, and weak supervision modules. We specify the block type and the feature dimension of each layer in Tables 4.3 to 4.5.

Keypoint module

Figure 4.23 shows the description of the lateral, upsample, and predict blocks that are used for the keypoint module. For the convolution, the stride is always set to 1.

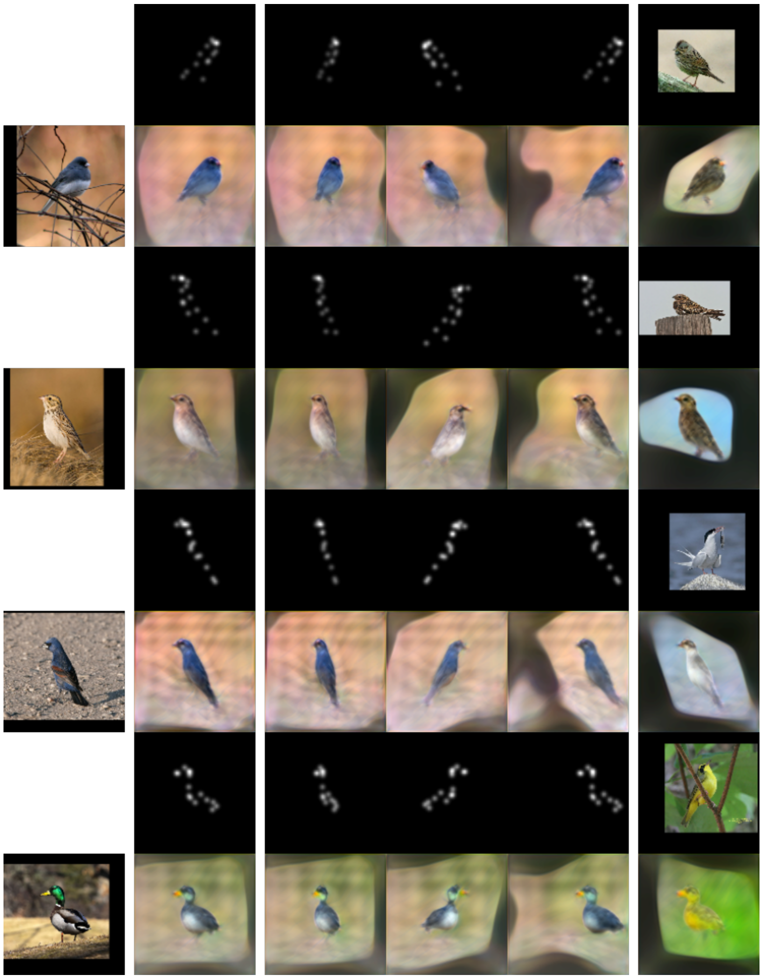
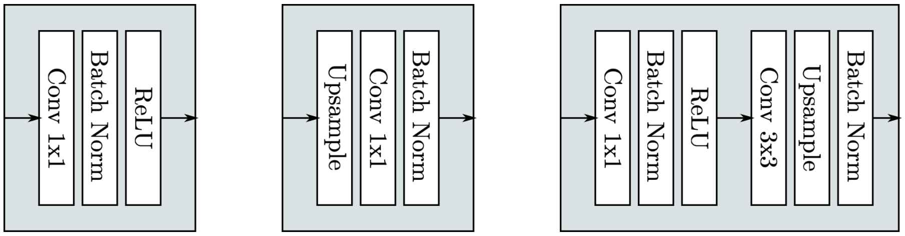


Figure 4.22: Appearance and geometry factorization on CUB dataset.



(a) Lateral (b) Upsample (c) Predict

Figure 4.23: Layer description on lateral, upsample, and predict blocks.

Table 4.3: Architecture details about the keypoint module.

Type	input_dim	out_ch_dim	output_size
Lateral	2048	256	4x4
Upsample	256	256	8x8
Predict	256	# parts	64x64
Lateral	1024	256	8x8
Upsample	256	256	16x16
Predict	256	# parts	64x64
Lateral	512	256	16x16
Upsample	256	256	32x32
Predict	256	# parts	64x64
Lateral	256	256	32x32
Upsample	256	256	64x64
Predict	256	# parts	64x64

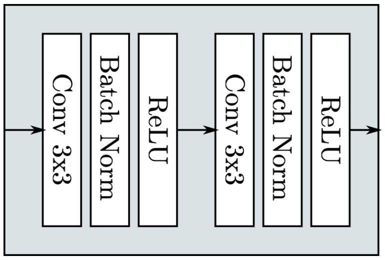


Figure 4.24: Basic convolution block which is used for reconstruction module.

Note that the input for each lateral block is the output from the convolution block of the ResNet-50 encoder. The output from the previous lateral block is added to the next lateral block so that the network does not lose the semantic information from the deep layers and spatial information from the shallow layers. The final heatmap, which is the output from the final predict block, is used to generate the geometry bottleneck.

Reconstruction module

We feed the geometry bottleneck to each resolution in the reconstruction module. In this procedure, we use a different normalization factor for generating the gaussian heatmap bottleneck so that the layers from higher resolution get more concentrated geometry features. We use $[0.1, 0.1, 0.01, 0.01, 0.001]$ to the σ variable in Eq 1 for each resolution. Figure 4.24 shows the basic convolution block (conv_block in Table 4.4), which is composed of 3x3 convolution, batch normalization, and ReLU

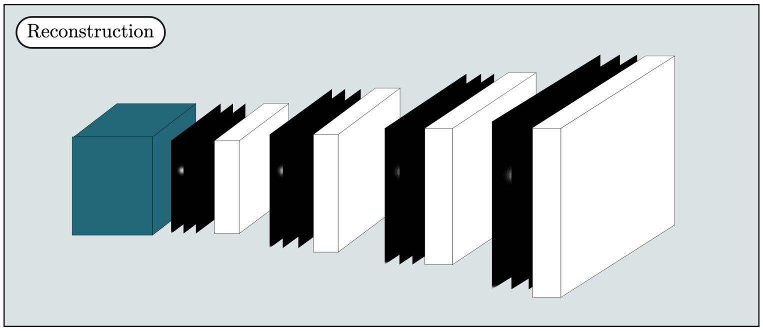


Figure 4.25: Reconstruction module.

Table 4.4: Architecture details about the reconstruction module.

Type	input_ch_dim	kernel_size	out_ch_dim	output_size
Upsampling	-	-	-	8x8
Conv_block	2048+# parts	-	1024	8x8
Upsampling	-	-	-	16x16
Conv_block	1024+# parts	-	512	16x16
Upsampling	-	-	-	32x32
Conv_block	512+# parts	-	256	32x32
Upsampling	-	-	-	64x64
Conv_block	256+# parts	-	128	64x64
Upsampling	-	-	-	128x128
Conv_block	128+# parts	-	64	128x128
Convolution	64	1	3	128x128

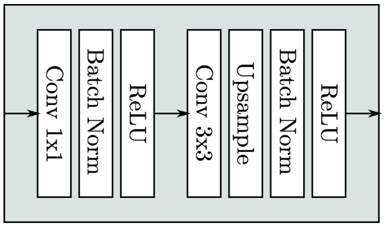


Figure 4.26: Layer specification in the convolution block (conv_block_w) in weak supervision module.

activation. The upsampling layer in Table 4.4 is a single upsampling layer which is different from the upsampling block in Table 4.3.

Weak supervision module

Figure 4.26 shows the layer description of the convolution block in the weak supervision module. The input for each convolution block is the output from the

Table 4.5: Architecture details about the weak supervision module.

Type	input_ch_dim	out_ch_dim	output_size
Conv_block_w	2048	256	64x64
Conv_block_w	1024	256	64x64
Conv_block_w	512	256	64x64
Conv_block_w	256	256	64x64

Table 4.6: Hyperparameter settings for each dataset.

Dataset	lr	L_{perc}	L_w	L_v	epoch	# parts
CelebA	0.001	1	-	-	-	10
CUB	0.001	1	1	1	30	15
AnimalPose	0.001	1	1	1	40	20
StanfordDogs	0.001	1	1	1	30	24

ResNet-50 encoder. We generate the base feature for the final classification task using the concatenation of the features from the module in Table 4.5.

Dataset

TigDog We use the subset of the TigDog dataset for the posture-based action classification task, where the action category consists of sitting, rolling, and standing. Most of the activity classes require temporal information, thus we used the images from the category of *walking*, *rolling*, *standing up* and *sitting*, where the action can be identified by watching a single frame. From the frames of *standing up* and *sitting* behaviors, we manually selected the images that have sitting posture for the sitting category.

Hyperparameter settings

Our method does not require an extensive hyperparameter search. The experimental results were obtained by applying the same weight for each loss. We show the starting epoch number for the curriculum learning of the viewpoint equivariance loss in Table 4.6. We use the SGD optimizer for all the experiments.

References

Barnard, Shanis, Simone Calderara, Simone Pistocchi, Rita Cucchiara, Michele Podaliri-Vulpiani, Stefano Messori, and Nicola Ferri (July 2016). "Quick, Accurate, Smart: 3D Computer Vision Technology Helps Assessing Confined Animals' Behaviour". eng. In: *PloS one* 11.7, e0158748–e0158748. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0158748. URL: https://doi.org/10.1371/journal.pone.0158748.

Belhumeur, P. N., D. W. Jacobs, D. J. Kriegman, and N. Kumar (2011). "Localizing Parts of Faces Using a Consensus of Exemplars". In: *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. ISBN: 9781457703942.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum Learning". In: *ICML*.

Biggs, Benjamin, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla (2020). "Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop". In: *ECCV*.

Branson, Steve, Grant Van Horn, Serge Belongie, and Pietro Perona (Sept. 1, 2014). "Bird Species Categorization Using Pose Normalized Deep Convolutional Nets". In: *British Machine Vision Conference (BMVC)*. Nottingham. URL: http://vision.cornell.edu/se3/wp-content/uploads/2015/02/BMVC14.pdf.

Burgos-Artizzu, Xavier P., Pietro Perona, and Piotr Dollár (2013). "Robust Face Landmark Estimation under Occlusion". In: *Proceedings of the 2013 IEEE International Conference on Computer Vision*. ISBN: 9781479928408.

Cao, Jinkun, Hongyang Tang, Haoshu Fang, Xiaoyong Shen, Cewu Lu, and Yu-Wing Tai (2019). "Cross-Domain Adaptation for Animal Pose Estimation". In: *CoRR* abs/1908.05806. arXiv: 1908.05806. URL: http://arxiv.org/abs/1908.05806.

Cao, Xudong, Yichen Wei, Fang Wen, and Jian Sun (Apr. 2014). "Face Alignment by Explicit Shape Regression". In: *Int. J. Comput. Vision* 107.2, pp. 177–190. ISSN: 0920-5691.

Chen, Yilun, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun (2017). "Cascaded Pyramid Network for Multi-Person Pose Estimation". In: *CoRR* abs/1711.07319.

Del Pero, L., S. Ricco, R. Sukthankar, and V. Ferrari (2015). "Articulated motion discovery using pairs of trajectories". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, Jia, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Fei-Fei Li (2009). "ImageNet: A large-scale hierarchical image database". In: *Proc. IEEE CVPR*.

Du, Yong, Wei Wang, and Liang Wang (2015). “Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Duchon, Jean (1977). “Splines minimizing rotation-invariant semi-norms in Sobolev spaces”. In: *Constructive Theory of Functions of Several Variables*. Ed. by Walter Schempp and Karl Zeller. Springer Berlin Heidelberg. ISBN: 978-3-540-37496-1.

Guo, Pei and Ryan Farrell (2019). “Aligned to the Object, Not to the Image: A Unified Pose-Aligned Representation for Fine-Grained Recognition”. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. DOI: 10.1109/WACV.2019.00204. URL: https://doi.org/10.1109/WACV.2019.00204.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *Proc. IEEE CVPR*.

Huang, Zilong, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang (2018). “Weakly-Supervised Semantic Segmentation Network With Deep Seeded Region Growing.” In: *CVPR*. URL: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2018.html#HuangWWLW18.

Jakab, Tomas, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi (2018). “Unsupervised Learning of Object Landmarks through Conditional Image Generation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

– (2020). “Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jie, Zequn, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu (2017). “Deep Self-Taught Learning for Weakly Supervised Object Localization”. In: *CVPR*.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). “Perceptual losses for real-time style transfer and super-resolution”. In: *European Conference on Computer Vision*.

Khosla, Aditya, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei (June 2011). “Novel Dataset for Fine-Grained Image Categorization”. In: *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Maosen, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian (June 2020). “Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang (Dec. 2015). “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*.

Lorenz, Dominik, Leonard Bereska, Timo Milbich, and Björn Ommer (2019). “Unsupervised Part-Based Disentangling of Object Shape and Appearance”. In: *CVPR*.

Maaten, Laurens van der and Geoffrey Hinton (2008). “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9, pp. 2579–2605. URL: `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.

Mathis, Alexander, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge (2018). “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning”. In: *Nature Neuroscience*. URL: `https://www.nature.com/articles/s41593-018-0209-y`.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). “Stacked Hourglass Networks for Human Pose Estimation”. In: *Proc. ECCV*.

Novotný, David, Diane Larlus, and Andrea Vedaldi (2017). “AnchorNet: A Weakly Supervised Network to Learn Geometry-Sensitive Features for Semantic Matching”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: `10.1109/CVPR.2017.306`. URL: `https://doi.org/10.1109/CVPR.2017.306`.

Oquab, M., L. Bottou, I. Laptev, and J. Sivic (2015). “Is object localization for free? - Weakly-supervised learning with convolutional neural networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: `10.1109/CVPR.2015.7298668`.

Pathak, Deepak, Philipp Krähenbühl, and Trevor Darrell (2015). “Constrained Convolutional Neural Networks for Weakly Supervised Segmentation”. In: *International Conference on Computer Vision (ICCV)*.

Ryou, Serim, Seong-Gyun Jeong, and Pietro Perona (2019). “Anchor Loss: Modulating Loss Scale Based on Prediction Difficulty”. In: *The IEEE International Conference on Computer Vision (ICCV)*. DOI: `doi.org/10.1109/ICCV.2019.00609`.

Shelhamer, Evan, Jonathan Long, and Trevor Darrell (2017). “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE TPAMI* 39.4, pp. 640–651.

Simonyan, Karen and Andrew Zisserman (2014). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556.

Singh, Krishna Kumar and Yong Jae Lee (2017). “Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-supervised Object and Action Localization.” In: *CoRR* abs/1704.04232. URL: `http://dblp.uni-trier.de/db/journals/corr/corr1704.html#SinghL17`.

Su, Chi, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian (Oct. 2017). “Pose-Driven Deep Convolutional Model for Person Re-Identification”. In: *The IEEE International Conference on Computer Vision (ICCV)*.

Sun, Ming, Yuchen Yuan, Feng Zhou, and Errui Ding (2018). "Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition". In: *CoRR* abs/1806.05372. arXiv: 1806.05372. URL: http://arxiv.org/abs/1806.05372.

Tang, Peng, Xinggang Wang, Song Bai, Wei Shen, Wenyu Liu, and Alan Yuille (2019). "PCL: Proposal Cluster Learning for Weakly Supervised Object Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Tang, Wei, Pei Yu, and Ying Wu (Sept. 2018). "Deeply Learned Compositional Models for Human Pose Estimation". In: *The European Conference on Computer Vision (ECCV)*.

Thewlis, James, Hakan Bilen, and Andrea Vedaldi (Oct. 2017). "Unsupervised Learning of Object Landmarks by Factorized Spatial Embeddings". In: *The IEEE International Conference on Computer Vision (ICCV)*.

Wahba, G. (1990). *Spline Models for Observational Data*. Philadelphia: Society for Industrial and Applied Mathematics.

Wang, Chong, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan (2014). "Weakly Supervised Object Localization with Latent Category Learning". In: *ECCV*. ISBN: 978-3-319-10599-4.

Wang, Chunyu, Yizhou Wang, and Alan L. Yuille (2013). "An Approach to Pose-Based Action Recognition." In: *Proc. IEEE CVPR*.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional Pose Machines". In: *Proc. IEEE CVPR*.

Welinder, P., S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona (2010). *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology.

Xie, Weidi, Li Shen, and Andrew Zisserman (2018). "Comparator Networks". In: *European Conference on Computer Vision*.

Zhang, Jian, Runsheng Zhang, Yaping Huang, and Qi Zou (2019). "Unsupervised Part Mining for Fine-grained Image Classification". In: *CoRR* abs/1902.09941. arXiv: 1902.09941. URL: http://arxiv.org/abs/1902.09941.

Zhang, Yuting, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee (2018). "Unsupervised Discovery of Object Landmarks as Structural Representations". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. DOI: 10.1109/CVPR.2018.00285.

Zhang, Zhanpeng, Ping Luo, Chen Change Loy, and Xiaoou Tang (2014). "Facial Landmark Detection by Deep Multi-task Learning". In: *ECCV*. ISBN: 978-3-319-10599-4.

Zhao, Haiyu, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang (2017). “Spindle Net: Person Re-identification with Human Body Region Guided Feature Decomposition and Fusion”. In: *Proc. IEEE CVPR*.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba (2016). “Learning Deep Features for Discriminative Localization”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chapter 5

NEUROSCIENCE: SELF-SUPERVISED KEYPOINT DISCOVERY IN BEHAVIORAL VIDEOS

The content of this chapter is from the manuscript "Self-Supervised Keypoint Discovery in Behavioral Videos" by J. J. Sun*, S. Ryou*, R. Goldshmid, B. Weissbourd, J. Dabiri, D. J. Anderson, A. Kennedy, Y. Yue, and P. Perona 2021.

Following the approaches in Chapter 4, we discuss an extended algorithm for keypoint discovery with an emphasis on the application of behavior analysis in this chapter.

5.1 Abstract

We propose a method for learning the posture and structure of agents from unlabelled behavioral videos. Starting from the observation that behaving agents are generally the main sources of movement in behavioral videos, our method uses an encoder-decoder architecture with a geometric bottleneck to reconstruct the difference between video frames. By focusing only on regions of movement, our approach works directly on input videos without requiring manual annotations, such as keypoints or bounding boxes. Experiments on a variety of agent types (mouse, fly, human, jellyfish, and trees) demonstrate the generality of our approach and reveal that our discovered keypoints represent semantically meaningful body parts, which achieve state-of-the-art performance on keypoint regression among self-supervised methods. Additionally, our discovered keypoints achieve comparable performance to supervised keypoints on downstream tasks, such as behavior classification, suggesting that our method can dramatically reduce the cost of model training vis-a-vis supervised methods.

5.2 Introduction

Automatic recognition of object structure, for example in the form of keypoints and skeletons, enables models to capture the essence of the geometry and movements of objects. Such structural representations are more invariant to background, lighting, and other nuisance variables and are much lower-dimensional than raw pixel values, making them good intermediates for downstream tasks, such as behavior classification (K. Branson et al., 2009; Eyjolfsdottir, S. Branson, et al., 2014; Segalin et al.,

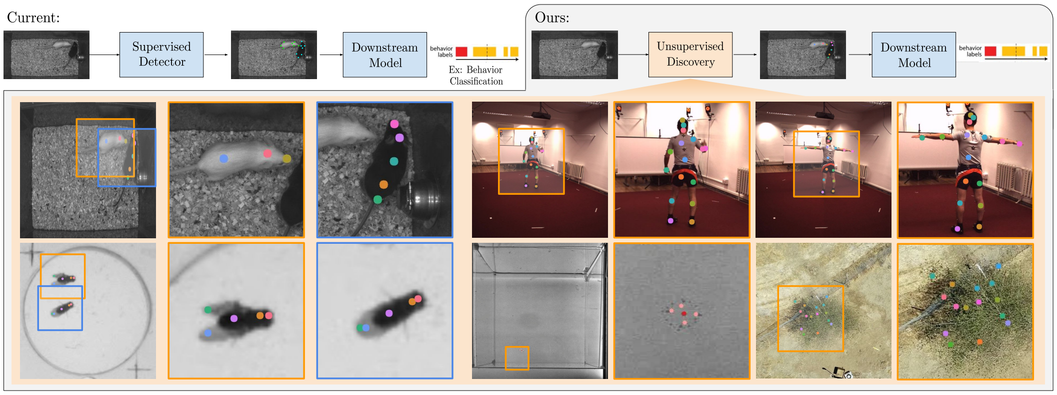


Figure 5.1: Self-supervised Keypoint Discovery. Intermediate representations in the form of keypoints are frequently used for behavior analysis. We propose a method to discover keypoints from behavioral videos without the need for manual keypoint or bounding box annotations. Our method works across a range of organisms (including mice, humans, flies, jellyfish and tree), works with multiple agents simultaneously (see flies and mice above), does not require bounding boxes (boxes visualized above purely for identifying the enlarged regions of interest) and achieves state-of-the-art performance on downstream tasks.

2020; Sun, Kennedy, et al., 2021; Dankert et al., 2009), video alignment (Sun, Zhao, et al., 2020; Liu et al., 2021), and physics-based modeling (J. L. Cardona and J. O. Dabiri, 2021; Silva et al., 2020).

However, obtaining annotations to train supervised pose detectors can be expensive, especially for applications in behavior analysis. For example, in behavioral neuroscience (Pereira, Shaevitz, and Murthy, 2020), datasets are typically small and lab-specific, and the training of a custom supervised keypoint detector presents a significant bottleneck in terms of cost and effort. Additionally, once trained, supervised detectors often do not generalize well to new agents with different structures without new supervision. The goal of our work is to enable keypoint discovery on new videos without manual supervision, in order to enable behavior analysis to be more easily carried out on novel settings and different agents.

Previous work on unsupervised/self-supervised methods for keypoint discovery (Jakab et al., 2020; Jakab et al., 2018; Zhang et al., 2018) (see also Section 5.3) has a few limitations when applied to behavioral videos. In particular, these methods do not address the case of multiple agents, which is fundamental to behavior analysis. Existing methods often require inputs as cropped bounding boxes around the object of interest, which would require an additional detector module to run on real-world videos. Furthermore, these methods do not exploit relevant structural properties in behavioral videos (e.g., the camera and the background are typically stationary, as observed in many real-world behavioral datasets (Segalin et al., 2020; Eyjolfsdottir, S. Branson, et al., 2014; Burgos-Artizzu et al., 2012; Marstaller, Tausch, and Stock, 2019; Pereira, Shaevitz, and Murthy, 2020; Jhuang et al., 2010)).

To address these challenges, the key to our approach is to learn keypoints based on reconstructing the *image difference* between two video frames. Similar to previous works based on image reconstruction (Jakab et al., 2018; Ryou and Perona, 2021), we use an encoder-decoder setup to encode input images into a keypoint bottleneck, and use the decoder for reconstruction. Our method then takes a novel approach in defining the reconstruction target as the difference between two frames instead of the full video frame as in previous work (Jakab et al., 2020; Ryou and Perona, 2021; Jakab et al., 2018). By focusing on agent movement, our model discovers keypoints for multiple agents directly from behavioral videos without requiring additional supervision.

Our self-supervised approach works without manual supervision across diverse organisms (Figure 5.1) and we find that our discovered keypoints achieve state-of-the-art performance on downstream tasks among other self-supervised keypoint discovery methods. We demonstrate the performance of our keypoints on behavior classification (Sun, Karigo, et al., 2021), keypoint regression (Jakab et al., 2018), and physics-based modeling (J. L. Cardona and J. O. Dabiri, 2021). Thus, our method has the potential for transformative impact in behavior analysis: first, one may discover keypoints from behavioral videos for new settings and organisms; second, unlike methods that predict behavior directly from video, our low-dimensional keypoints are semantically meaningful so that users can directly compute behavioral features; finally, our method can be applied to videos without the need for manual annotations.

To summarize, our main contributions are:

1. Self-supervised method for discovering keypoints from real-world behavioral videos recorded from largely stationary cameras, without requiring manual annotations.
2. Experiments across a range of organisms (mice, flies, human, jellyfish, and tree) demonstrating the generality of the method and showing that the discovered keypoints are semantically meaningful.
3. Quantitative benchmarking on downstream behavior analysis tasks showing performance that is comparable to supervised keypoints.

5.3 Related work

Analyzing Behavioral Videos. Video data collected for behavioral experiments often consists of moving agents recorded from stationary cameras (Anderson and Perona, 2014; Segalin et al., 2020; Eyjolfsdottir, S. Branson, et al., 2014; Burgos-Artizzu et al., 2012; Nilsson et al., 2020; Dankert et al., 2009; K. Branson et al., 2009; Jhuang et al., 2010). These behavioral videos contain different model organisms studied by researchers, such as fruit flies (Eyjolfsdottir, S. Branson, et al., 2014; Kabra et al., 2013; K. Branson et al., 2009; Dankert et al., 2009) and mice (Hong et al., 2015; Segalin et al., 2020; Jhuang et al., 2010; Burgos-Artizzu et al., 2012). From these recorded video data, there has been an increasing effort to automatically estimate poses of agents and classify behavior (Kabra et al., 2013; Hong et al., 2015; Eyjolfsdottir, K. Branson, et al., 2017; Mathis et al., 2018; Egnor and K. Branson, 2016; Segalin et al., 2020).

Pose estimation models that were developed for behavioral videos (Mathis et al., 2018; Graving et al., 2019; Segalin et al., 2020; Pereira, Tabris, et al., 2020) require human annotations of anatomically defined keypoints, which are expensive and time-consuming to obtain. In addition to the cost, not all data can be crowd-sourced due to the sensitive nature of some experiments. Furthermore, organisms that are translucent (jellyfish) or with complex shapes (tree) can be difficult for non-expert humans to annotate. Our goal is to enable keypoint discovery on videos for behavior analysis, without the need for manual annotations.

After pose estimation, behavior analysis models generally compute trajectory features and train behavior classifiers in a fully supervised fashion (Burgos-Artizzu et al., 2012; Hong et al., 2015; Eyjolfsdottir, S. Branson, et al., 2014; Sun, Kennedy, et al., 2021; Segalin et al., 2020). Some works have also explored using unsupervised methods to discover new motifs and behaviors (Berman et al., 2014; Wiltschko et al., 2015; Hsu and Yttri, 2021; Luxem et al., 2020). Here, we apply our discovered keypoints to supervised behavior classification and compare against baseline models using supervised keypoints for this task.

Keypoint Estimation. Pose estimation is the problem of localizing a predefined set of keypoints from visual data, and many works in this area focus on human pose. With the success of fully convolutional neural networks (Shelhamer, Long, and Darrell, 2017), recent methods (Newell, Yang, and Deng, 2016; Wei et al., 2016; Chen et al., 2017; Tang, Yu, and Wu, 2018) employ encoder-decoder networks by predicting high-resolution outputs encoded with 2D Gaussian heatmaps representing each part. To improve model performance, (Newell, Yang, and Deng, 2016; Wei et al., 2016; Tang, Yu, and Wu, 2018) propose an iterative refinement approach, (Chen et al., 2017; Ryou, Jeong, and Perona, 2019) design efficient learning signals, and (Cheng et al., 2020; J. Wang et al., 2021) exploit multi-resolution information. Beyond human pose, there are also works that focus on animal pose estimation, notably (Mathis et al., 2018; Graving et al., 2019; Pereira, Tabris, et al., 2020). Similar to these works, we also use 2D Gaussian heatmaps to represent parts as keypoints, but instead of using human-defined keypoints, we aim to discover keypoints from video data without manual supervision.

Unsupervised Part Discovery. Though keypoints provide a useful tool for behavior analysis, collecting annotations is time-consuming and labor-intensive especially for new domains that have not been previously studied. Unsupervised keypoint discovery (Jakab et al., 2018; Zhang et al., 2018; Jakab et al., 2020) has been proposed to reduce keypoint annotation effort and there have been many promising results on aligned objects, such as facial images and humans with an upright pose. These methods train and evaluate on images where the object of interest is centered in an input bounding box. Most of the approaches (Zhang et al., 2018; Jakab et al., 2018; Lorenz et al., 2019) use an autoencoder-based architecture to disentangle the appearance and geometry representation for the image reconstruction task. Our setup is similar in that we also use an encoder-decoder architecture, but crucially, we reconstruct image difference between video frames, instead of the full image as in previous works. We found that this enables our discovered keypoints to track semantically-consistent parts without manual supervision, requiring neither keypoints nor bounding boxes.

There are also works for parts discovery that employ other types of supervision (Jakab et al., 2020; Schmidtke et al., 2021; Ryou and Perona, 2021). For example, (Ryou and Perona, 2021) proposed a weakly-supervised approach using class label to discriminate parts to handle viewpoint changes, (Jakab et al., 2020) incorporated pose prior obtained from unpaired data from different datasets in the same domain, and (Schmidtke et al., 2021) proposed a template-based geometry bottleneck based on a pre-defined 2D Gaussian-shaped template. Different from these approaches, our method does not require any supervision beyond the behavioral videos. We chose to focus on this setting since other supervisory sources are not readily available for emerging domains (ex: jellyfish, trees).

In previous works, keypoint discovery has been applied to downstream tasks, such

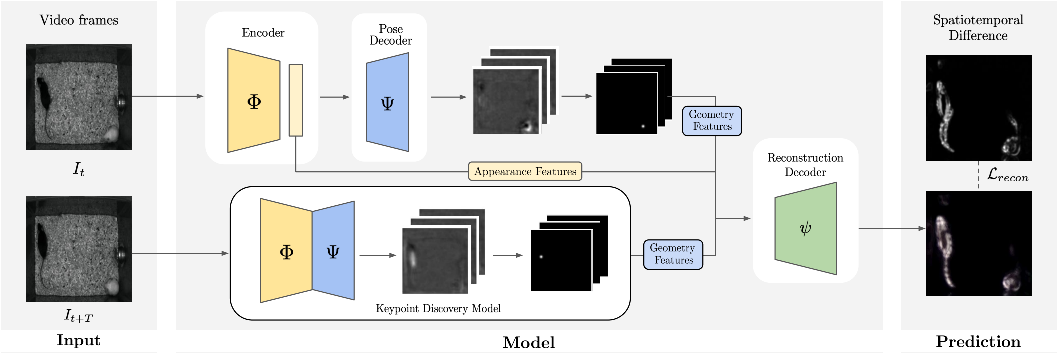


Figure 5.2: Our encoder-decoder approach for image difference reconstruction. Both frame I_t and frame I_{t+T} are fed to an appearance encoder Φ and a pose decoder Ψ. Given the appearance feature from I_t and geometry features from both I_t and I_{t+T} (Sec 5.4), our model reconstructs the spatiotemporal difference (Sec 5.4) computed from two frames using the reconstruction decoder ψ.

as image and video generation (Minderer et al., 2019; Jakab et al., 2020), keypoint regression to human-annotated poses (Zhang et al., 2018; Jakab et al., 2018), and video-level action recognition (Kim et al., 2019; Minderer et al., 2019). While we also apply keypoint discovery to downstream tasks, we note that our work differs in approach (we discover keypoints directly on behavioral videos using image difference reconstruction), focus (behavioral videos of diverse organisms from largely stationary cameras), and application (real-world behavior analysis tasks (Sun, Karigo, et al., 2021; J. L. Cardona and J. O. Dabiri, 2021)).

5.4 Method

The goal of our approach (Figure 5.2) is to discover semantically meaningful keypoints in behavioral videos of diverse organisms without manual supervision. In behavioral videos, the camera is generally fixed with respect to the world, such that the background is largely stationary and the agents (*e.g.*, mice and flies moving in an enclosure) are the only moving components of the scene. Thus spatiotemporal differences provide a strong cue to infer location and movements of agents.

Self-supervised keypoint discovery

Given a behavioral video, our work aims to reconstruct regions of motion between a reference frame I_t (the video frame at time t) and a future frame I_{t+T} (the video frame T timesteps later, for some set value of T.) We accomplish this by extracting appearance features from frame I_t and keypoint locations ("geometry features") from both frames I_t and I_{t+T} (Figure 5.2). In contrast, previous works (Jakab et al., 2018; Lorenz et al., 2019; Jakab et al., 2020; Ryou and Perona, 2021; Schmidtke et al., 2021) only use appearance features from I_t and geometry features from I_{t+T} to reconstruct the full image I_{t+T} (instead of difference between I_t and I_{t+T}).

We use an encoder-decoder architecture, with shared appearance encoder Φ, geometry decoder Ψ, and reconstruction decoder ψ. During training, the pair of frames I_t and I_{t+T} are fed to the appearance encoder Φ to generate appearance features, and those features are then fed into the geometry decoder Ψ to generate geometry features. In our approach, the reference frame I_t is used to generate both appearance and geometry representations, and the future frame I_{t+T} is only used to generate a geometry representation. The appearance feature h_a^t for frame I_t are defined simply as the output of Φ: $h_a^t = \Phi(I_t)$.

The pose decoder Ψ outputs K raw heatmaps $\mathbf{X}_i \in \mathbb{R}^2$, then applies a spatial softmax operation on each heatmap channel. Given the extracted $p_i = (u_i, v_i)$ locations for $i = \{1, \dots, K\}$ keypoints from the spatial softmax, we define the geometry features h_g^t to be a concatenation of 2D Gaussians centered at (u_i, v_i) with variance σ.

Finally, the concatenation of the appearance feature h_a^t and the geometry features h_g^t and h_g^{t+T} is fed to the decoder ψ to reconstruct the learning objective $\hat{S}$ discussed in the next section: $\hat{S} = \psi(h_a^t, h_g^t, h_g^{t+T})$.

Learning formulation

Spatiotemporal difference

Our method works with different types of spatiotemporal differences as reconstruction targets. For example:

Structural Similarity Index Measure (SSIM) (Z. Wang et al., 2004). This is a method for measuring the perceived quality of the two images based on luminance, contrast, and structure features. To compute our reconstruction target based on SSIM, we apply the SSIM measure locally on corresponding patches between I_t and I_{t+T} to build a similarity map between frames. Then we compute dissimilarity by taking the negation of the similarity map.

Frame differences. When the video background is static with little noise, simple frame differences, such as absolute difference ($S_{|d|} = |I_{t+T} - I_t|$) or raw difference ($S_d = I_{t+T} - I_t$), can also be directly applied as a reconstruction target.

Reconstruction loss

We apply perceptual loss (Johnson, Alahi, and Fei-Fei, 2016) for reconstructing the spatiotemporal difference S. Perceptual loss compares the L2 distance between the features computed from VGG network ϕ (Simonyan and Zisserman, 2014). The reconstruction $\hat{S}$ and the target S are fed to VGG network, and mean squared error is applied to the features from the intermediate convolutional blocks:

$$\mathcal{L}_{recon} = \left\| \phi(S(I_t, I_{t+T})) - \phi(\hat{S}(I_t, I_{t+T})) \right\|_2 . \tag{5.1}$$

Rotation equivariance loss

In cases where agents can move in many directions (*e.g.*, mice filmed from above can translate and rotate freely), we would like our keypoints to remain semantically consistent. We enforce rotation-equivariance in the discovered keypoints by rotating the image with different angles and imposing that the predicted keypoints should move correspondingly. We apply the rotation equivariance loss on the generated heatmap.

Given reference image I and the corresponding geometry bottleneck h_g, we rotate the geometry bottleneck to generate pseudo labels $h_g^{R^\circ}$ for rotated input images I^{R° with degree $R = \{90^\circ, 180^\circ, 270^\circ\}$. We apply mean squared error between the predicted geometry bottlenecks $\hat{h_g}$ from the rotated images and the generated pseudo labels h_g:

$$\mathcal{L}_r = \left\| h_g^{R^\circ} - \hat{h}_g(I^{R^\circ}) \right\|_2 . \tag{5.2}$$

Separation loss

Empirical results show that rotation equivariance encourages the discovered keypoints to converge at the center of the image. We apply separation loss to encourage the keypoints to encode unique coordinates, and prevent the discovered keypoints from being centered at the image coordinates (Zhang et al., 2018). The separation loss is defined as follows:

$$\mathcal{L}_s = \sum_{i \neq j} \exp\left(\frac{-(p_i - p_j)^2}{2\sigma_s^2} \right). \tag{5.3}$$

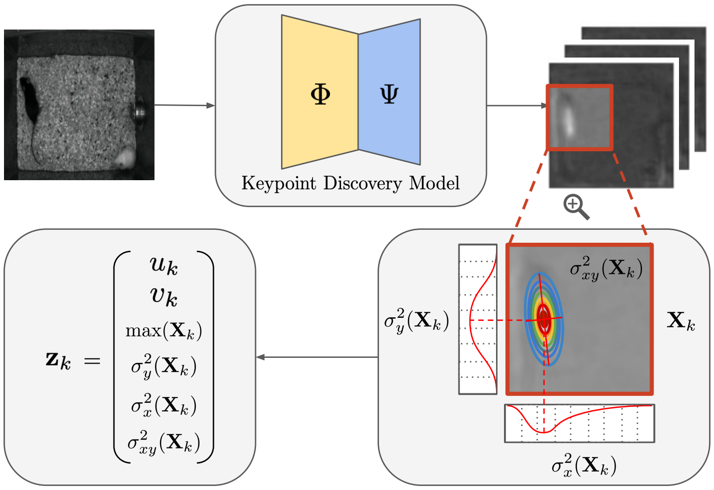

Φ
Ψ
Keypoint Discovery Model
$\mathbf{z}_k = \begin{pmatrix} u_k \\ v_k \\ \max(\mathbf{X}_k) \\ \sigma_y^2(\mathbf{X}_k) \\ \sigma_x^2(\mathbf{X}_k) \\ \sigma_{xy}^2(\mathbf{X}_k) \end{pmatrix}$
$\sigma_y^2(\mathbf{X}_k)$
$\sigma_{xy}^2(\mathbf{X}_k)$
$\mathbf{X}_k$
$\sigma_x^2(\mathbf{X}_k)$

Figure 5.3: Behavior Classification Features. Extracting information from the raw heatmap (Section 5.4): the confidence scores and the covariance matrices are computed from normalized heatmaps. Note that the features are computed for all x, y coordinates. We visualize the zoomed area around the target instance for illustrative purposes.

Final objective

Our final loss function is composed of three parts: reconstruction loss $\mathcal{L}_{recon}$, rotation equivariance loss $\mathcal{L}_r$, and separation loss $\mathcal{L}_s$:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathbb{1}_{epoch>n}(w_r\mathcal{L}_r + w_s\mathcal{L}_s). \tag{5.4}$$

We adopt curriculum learning (Bengio et al., 2009) and apply $\mathcal{L}_r$ and $\mathcal{L}_s$ once the keypoints are consistently discovered from the semantic parts of the target instance.

Feature extraction for behavior analysis

Following standard approaches (Segalin et al., 2020; Burgos-Artizzu et al., 2012; Hong et al., 2015), we use the discovered keypoints as input to a behavior quantification module: either supervised behavior classifiers or a physical model. Note that this is a separate process from keypoint discovery; we feed discovered geometry information into a downstream model.

In addition to discovered keypoints, we extracted some additional features from the raw heatmap (Figure 5.3) to be used as input to our downstream modules. For instance, we extracted the uncertainty and confidence of the network prediction of keypoint location, as we found these features to be informative. When a target part is well localized, our keypoint discovery network produces a heatmap with a single high peak with low variance; conversely, when a target part is occluded, the raw heatmap contains a blurred shape with lower peak value. This "confidence" score (heatmap peak value) is also a good indicator for whether keypoints are discovered on the background (blurred over the background with low confidence) or tracking anatomical body parts (peaked with high confidence), visualized in Appendix. The shape of a computed heatmap can also reflect shape information of the target (*e.g.*, stretching).

Given a raw heatmap $\mathbf{X}_k$ for part k, the confidence score is obtained by choosing the maximum value from the heatmap, and the uncertainty measure is obtained by computing the covariance matrix from the heatmap. Figure 5.3 visualizes the features we extract from the raw heatmaps. Using the normalized heatmap as the probability distribution, additional geometric features are computed:

$$\begin{aligned}
\sigma_x^2(\mathbf{X}_k) &= \sum_{ij}(x_i - u_k)^2 \mathbf{X}_k(i,j), \\
\sigma_y^2(\mathbf{X}_k) &= \sum_{ij}(y_j - v_k)^2 \mathbf{X}_k(i,j), \\
\sigma_{xy}^2(\mathbf{X}_k) &= \sum_{ij}(x_i - u_k)(y_j - v_k) \mathbf{X}_k(i,j).
\end{aligned} \tag{5.5}$$

5.5 Experiments

We demonstrate that our method is able to discover consistent keypoints in real-world behavioral videos across a range of organisms (Section 5.5). We evaluate our keypoints on downstream tasks for behavior classification (Section 5.5) and pose regression (Section 5.5), then illustrate additional applications of our keypoints (Section 5.5).

Experimental setting

Datasets

CalMS21. CalMS21 (Sun, Karigo, et al., 2021) is a large-scale dataset for behavior analysis consisting of videos and trajectory data from a pair of interacting mice. Every frame is annotated by an expert for three behaviors: sniff, attack, mount. There are 507k frames in the train split, and 262k frames in the test split (video frame: 1024 × 570, mouse: approx 150 × 50). We use only the train split on videos without miniscope cable to train our keypoint discovery model. Following (Sun, Karigo, et al., 2021), the downstream behavior classifier is trained on the entire training split, and performance is evaluated on the test split.

MARS-Pose. This dataset consists of a set of videos with similar recording conditions to the CalMS21 dataset. We use a subset of the MARS pose dataset (Segalin et al., 2020) with keypoints from manual annotations to evaluate the ability of our model to predict human-annotated keypoints, with $\{10, 50, 100, 500\}$ images for train and 1.5k images for test.

Fly vs. Fly. These videos consists of interactions between a pair of flies, annotated per frame by domain experts. We use the Aggression videos from the Fly vs. Fly dataset (Eyjolfsdottir, S. Branson, et al., 2014), with the train and test split having 1229k and 322k frames respectively (video frame: 144×144, fly: approx 30×10). Similar to (Sun, Kennedy, et al., 2021), we evaluate on behaviors of interest with more than 1000 frames in the training set (lunge, wing threat, tussle).

Human 3.6M. The Human 3.6M dataset (Ionescu et al., 2013) is a large-scale motion capture dataset, which consists of 3.6 million human poses and images for 17 different activities taken from 4 viewpoints. To quantitatively measure the pose regression performance against baselines, we use the Simplified Human 3.6M dataset, which consists of 800k training and 90k testing images with 6 activities in which the human body is mostly upright. We follow the same evaluation protocol from (Zhang et al., 2018) to use subjects 1, 5, 6, 7, and 8 for training and 9 and 11 for testing.

Jellyfish. The jellyfish data is an in-house video dataset containing 30k frames of recorded swimming jellyfish (video frame: 928×1158, jellyfish: approx 50 pix in diameter). We use this dataset to qualitatively test the performance of our model on a new organism, and apply our keypoints to detect the pulsing motion of the jellyfish.

Vegetation. This is an in-house dataset acquired over several weeks using a drone to record the motion of swaying trees. The dataset consists of videos of an oak tree and corresponding wind speeds recorded using an anemometer, with a total of 2.41M video frames (video frame: 512×512, oak tree: varies, approx $\frac{1}{4}$ of the frame). We evaluate this dataset using a physics-based model (J. L. Cardona and J. O. Dabiri, 2021) that relates the visually observed oscillations to the average wind speeds.

Training and evaluation procedure

We train our keypoint discovery model using the full objective in Section 5.4. During training, we rescale images to 256 × 256 and use T of around 0.2 seconds, except the Human dataset, where we use 128 × 128. Unless otherwise specified, all experiments are ran with all keypoints discovered from our keypoint discovery model with 10 keypoints for mouse, fly, and jellyfish, 16 keypoints for the human dataset, and 15 keypoints for the vegetation dataset. We train on the train split of each dataset as specified, except for jellyfish and vegetation, where we use the entire dataset. Additional details are in the Appendix.

After training the keypoint discovery model, we extract the keypoints and use it for different evaluations based on the labels available in the dataset: behavior classification (CalMS21, Fly), keypoint regression (MARS-Pose, Human), and physics-based modeling (Vegetation).

For keypoint regression, similar to previous works (Jakab et al., 2018; Jakab et al., 2020), we compare our regression with a fully supervised 1-stack hourglass network (Newell, Yang, and Deng, 2016). We evaluate keypoint regression on Simplified Human 3.6M dataset by using a linear regressor without a bias term, following the same evaluation setup from previous works (Zhang et al., 2018; Lorenz et al., 2019). On MARS-Pose, we train our model in a semi-supervised fashion with $10, 50, 100, 500$ supervised keypoints to test data efficiency. For behavior classification, we evaluate on CalMS21 and Fly, using available frame-level behavior annotations. To train behavior classifiers, we use the specified train split of each dataset. For CalMS21 and Fly, we train the 1D Convolutional Network benchmark model provided by (Sun, Karigo, et al., 2021) on our keypoints. We evaluate using mean average precision (MAP) weighted equally over all behaviors of interest.

Behavior classification results

CalMS21 Behavior Classification. We evaluate the effectiveness of our discovered keypoints for behavior classification (Table 5.1). Compared to supervised keypoints trained for this task, our keypoints (without supervision), is comparable when using both pose and confidence as input. Compared to other self-supervised methods, even those that use bounding boxes, our discovered keypoints on the full image generally achieve better performance.

Keypoints discovered through image reconstruction, similar to baselines (Jakab et al., 2018; Ryou and Perona, 2021) cannot track the agents well without using bound-

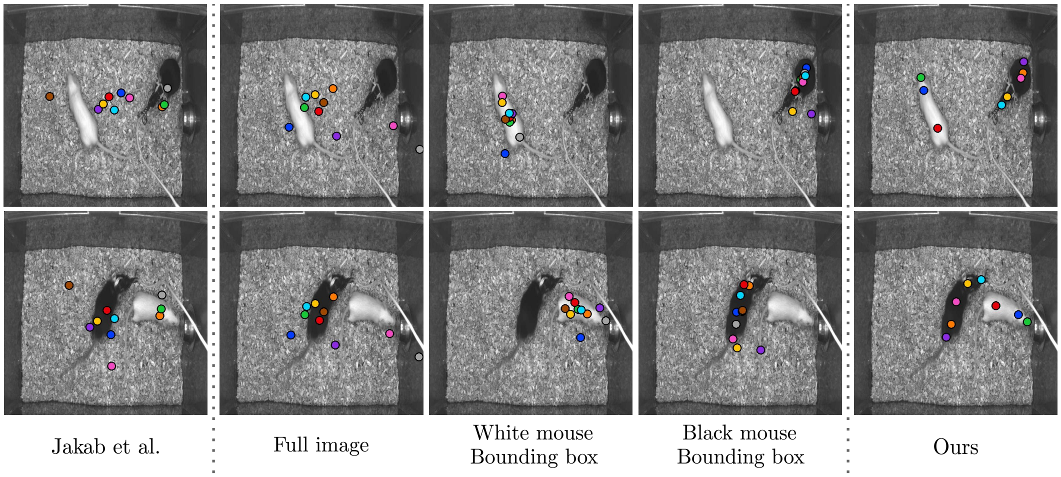


Figure 5.4: Comparison with existing methods (Jakab et al., 2018), full image, bounding box, and SSIM reconstruction (ours). "Jakab *et al*" and "full image" results are based on full image reconstruction. "White mouse bounding box" and "black mouse bounding box" show the results when the cropped bounding boxes were fed to the network for image reconstruction.

ing box information (Figure 5.4) and does not perform well for behavior classification (Table 5.1). When we provide bounding box information to the image-based keypoint discovery module, the performance is significantly improved, but this model does not perform as well as our keypoints from image difference reconstruction.

For the per-class performance (see the appendix), the biggest gap exists between our keypoints and MARS on the "attack" behavior. This is likely because during attack, the mice are moving quickly, and there exists a lot of motion blur and occlusion which is difficult to track without supervision. However, once we extract more information from the heatmap, through computing keypoint confidence, our keypoints perform comparably to MARS.

Fly Behavior Classification. The FlyTracker (Eyjolfsdottir, S. Branson, et al., 2014) uses hand-crafted features computed from the image, such as contrast, as well as features from tracked fly body parts, such as wing angle or distance between flies. Using discovered keypoints, we compute comparable features without assuming keypoint identity, by computing speed and acceleration of every keypoint, distance between every pair, and angle between every triplet. For all self-supervised methods, we use keypoints, confidence, and covariance for behavior classification. Results demonstrate that while there is a small gap in performance to the supervised estimator, our discovered keypoints perform much better than image reconstruction,

CalMS21	Pose	Conf	Cov	MAP
Fully supervised				
MARS † (Segalin et al., 2020)	✓			.856 ± .010
	✓	✓		.874 ± .003
	✓	✓	✓	.880 ± .005
Self-supervised				
Jakab et al. (Jakab et al., 2018)	✓			.186 ± .008
Image Recon.	✓			.182 ± .007
	✓	✓		.184 ± .006
	✓	✓	✓	.165 ± .012
Image Recon. bbox†	✓			.819 ± .008
	✓	✓		.812 ± .006
	✓	✓	✓	.812 ± .010
Ours	✓			.814 ± .007
	✓	✓		.857 ± .005
	✓	✓	✓	.852 ± .013

Table 5.1: Behavior Classification Results on CalMS21. "Ours" represents classifiers using input keypoints from our discovered keypoints. "conf" represents using the confidence score, and "cov" represents values from the covariance matrix of the heatmap. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 runs are shown.

Fly	MAP
Hand-crafted features	
FlyTracker (Eyjolfsdottir, S. Branson, et al., 2014)	.809 ± .013
Self-supervised + generic features	
Image Recon.	.500 ± .024
Image Recon. bbox†	.750 ± .020
Ours	.727 ± .022

Table 5.2: Behavior Classification Results on Fly. "FlyTracker" represents classifiers using hand-crafted inputs from (Eyjolfsdottir, S. Branson, et al., 2014). The self-supervised keypoints all use the same "generic features" computed on all keypoints: speed, acceleration, distance, and angle. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 runs are shown.

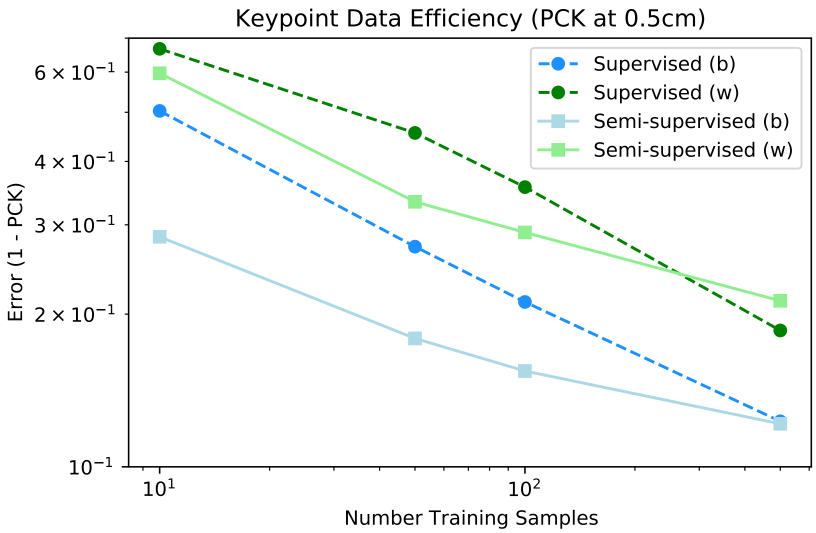


Figure 5.5: Keypoint data efficiency on MARS-Pose. The supervised model is based on (Segalin et al., 2020) using stacked hourglass (Newell, Yang, and Deng, 2016), while the semi-supervised model uses both our self-supervised loss and supervision. PCK is computed at $0.5cm$ threshold, averaged across nose, ears, and tail keypoints, over 3 runs. “b” and “w” indicates the black and white mouse respectively.

and is comparable to models that require bounding box inputs (Table 5.2).

Pose regression results

MARS Pose Regression. We evaluate the pose estimation performance of our method in the setting where some human annotated keypoints exist (Figure 5.5). For this experiment, we train our model in a semi-supervised fashion, where the loss is a sum of both our keypoint discovery objective (Section 5.4) as well as standard keypoint estimation objectives based on MSE (Segalin et al., 2020). For both black and white mouse, when using our keypoint discovery objective in a semi-supervised way during training, we are able to track keypoints more accurately compared to the supervised method (Segalin et al., 2020) alone. We note that the performance of both methods converge at around 500 annotated examples.

Simplified Human 3.6M Pose Regression. To compare with existing keypoint discovery methods, we evaluate our discovered keypoints on the human dataset (a standard benchmarking dataset) by regressing to annotated keypoints (Table 5.3). Though our method is directly applicable to full images, we train the discovery model using cropped bounding box for a fair comparison with baselines, which all use

Simplified H36M	all	wait	pose	greet	direct	discuss	walk
Fully supervised:							
Newell (Newell, Yang, and Deng, 2016)	2.16	1.88	1.92	2.15	1.62	1.88	2.21
Self-supervised + unpaired labels							
Jakab (Jakab et al., 2020)‡	2.73	2.66	2.27	2.73	2.35	2.35	4.00
Self-supervised + template							
Schmidtke (Schmidtke et al., 2021)	3.31	3.51	3.28	3.50	3.03	2.97	3.55
Self-supervised + regression							
Thewlis (Thewlis, Bilen, and Vedaldi, 2017)	7.51	7.54	8.56	7.26	6.47	7.93	5.40
Zhang (Zhang et al., 2018)	4.14	5.01	4.61	4.76	4.45	4.91	4.61
Lorenz (Lorenz et al., 2019)	2.79	–	–	–	–	–	–
Ours	2.44	2.50	2.22	2.47	2.22	2.77	2.50

Table 5.3: Comparison with state-of-the-art methods for landmark prediction on Simplified Human 3.6M. The error is in %-MSE normalized by image size. All methods predict 16 keypoints except for (Jakab et al., 2020)‡, which uses 32 keypoints for training a prior model from the Human 3.6M dataset.

cropped bounding boxes centered on the subject. Compared to both self-supervised + prior information and self-supervised + regression, our method shows state-of-the-art performance on the keypoint regression task, suggesting image difference is an effective reconstruction target for keypoint discovery.

Ablation Study

Learning Objective Ablation Study. We report the pose regression performance on the Human dataset in Table 5.4 by varying the spatiotemporal difference reconstruction target. Here, image reconstruction also performs well since cropped bounding box is used as an input to the network. Overall, spatiotemporal difference reconstruction yield better performance over image reconstruction.

Effect of Hyperparameters. We evaluate the effect of number of keypoints and frame gaps on the human dataset (Table 5.5). Note that we use pure frame difference as a reconstruction target for studying the effect of hyperparameters. When the frame gap is too small, the region of motion becomes too narrow, which results in slightly lower performance. Also, discovering more keypoints does not always guarantee better performance. Empirical results show that informative keypoints are discoverable with 16 keypoints.

Additional applications

We show qualitative performance and demonstrate additional downstream tasks

	Image Recon.	SSIM	Abs. Difference	Difference
%-MSE	2.67	**2.44**	2.46	2.57

Table 5.4: Learning Objective Ablation, Simplified Human3.6M. %-MSE error is reported by changing the reconstruction target.

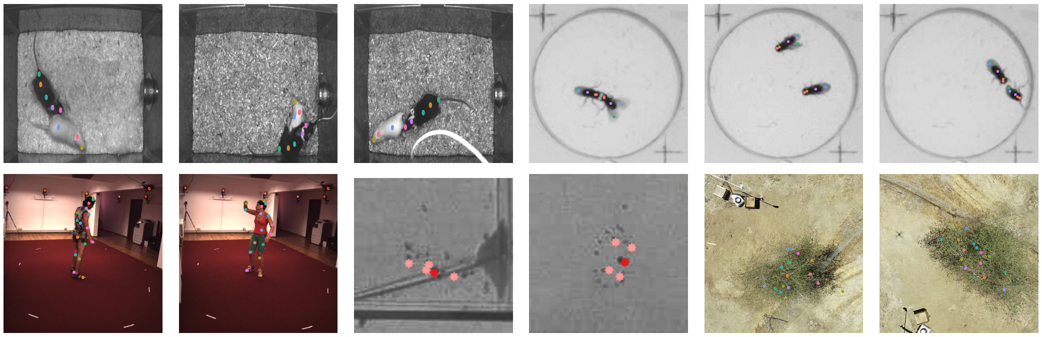


Figure 5.6: Qualitative Results. Qualitative results of the keypoint discovery model trained on CalMS21 (mouse), Fly vs. Fly (fly), Human3.6M (human), jellyfish and Vegetation (tree). Additional visualizations are in the Supplementary materials.

Hyperparam.	Value	%-MSE	Hyperparam.	Value	%-MSE
Frame Gap	10	2.81	# keypoints	10	2.96
	20	**2.57**		16	**2.57**
	30	2.64		30	2.63

Table 5.5: Hyperparameters, Simplified Human 3.6M. For frame gap experiments, the number of keypoints is set to 16. Frame gap is set to 20 for experiments with a varying number of keypoints.

using our discovered keypoints, on pulse detection for jellyfish and on wind speed regression for the Vegetation data. Additional details for all experiments are in the Appendix.

Qualitative Results. From our qualitative results (Figure 5.6), we see that our keypoints are able to track some body parts consistently, such as the nose of both mice and keypoints along the spine; the body and wings of the flies; the mouth and gonads of the jellyfish; and points on the arms and legs of the human. For visualization only, we show only keypoints discovered with high confidence values (Section 5.4); for all other experiments, we use all discovered keypoints.

Pulse Detection. Jellyfish swimming is among the most energetically efficient forms of transport, and its control and mechanics are studied in hydrodynamics research (Costello et al., 2021). Of key interest is the relationship between body plan and swim pulse frequency across diverse jellyfish species. By computing distance between our discovered keypoints, we are able to extract a frequency spectrogram to study pulsing of the jellyfish, with a visible band at the swimming frequency. This provides a way to automatically annotate swimming behavior, which could be quickly applied to video from multiple species to characterize the relationship between swimming dynamics and body plan.

Wind Speed Modeling. Measuring local wind speed is useful for tasks such as tracking air pollution and weather forecasting (J. Cardona, Howland, and J. Dabiri, 2019). Oscillations of trees encode information on wind conditions, and as such, videos of moving trees could function as wind speed sensors (J. Cardona, Howland, and J. Dabiri, 2019; J. L. Cardona and J. O. Dabiri, 2021). Using the Vegetation dataset, we evaluate the ability of our keypoints to predict wind speed using a physics-based model (J. L. Cardona and J. O. Dabiri, 2021). This model defines the relationship between the mean wind speed and the structural oscillations of the tree, and requires tracking these oscillations from video, which was previously done manually. We show that the keypoint detection model can accomplish this task automatically. Using our keypoints, we are able to regress the measured ground truth wind speed with an $R^2 = 0.79$, suggesting there is a good agreement between the proportionality assumption from (J. L. Cardona and J. O. Dabiri, 2021) and the experimental results using the keypoint detection model.

5.6 Discussion and conclusion

We propose a self-supervised method to discover meaningful keypoints from unlabelled videos for behavior analysis. We observe that in many settings, behavioral videos have stationary cameras which contain agents moving against a (quasi) stationary background. Our proposed method is based on reconstructing image difference between video frames, can handle videos with multiple agents, and does not require manual annotations. Our approach is general, and works well across a range of organisms.

Results show that our discovered keypoints are semantically meaningful, informative, and enable performance comparable to supervised keypoints on the downstream task of behavior classification. Our method will reduce the time and cost dramatically for video-based behavior analysis, thus accelerating scientific progress in fields such as ethology and neuroscience.

Limitations. One issue we did not explore in detail, and which will require further work, is keypoint discovery for agents that may be partially or completely occluded at some point during observation (Ohayon et al., 2013). Additionally, similar to other keypoint discovery models (Zhang et al., 2018; Lorenz et al., 2019; Schmidtke et al., 2021), we observe left/right swapping of some body parts, such as the legs in a walking human. One approach that might overcome these issues would be to extend our model to discover the 3D structure of the organism, for instance by using data from multiple cameras. Despite these challenges, our model performs comparably to supervised keypoints for behavior classification.

Benefits and risks of this technology. Automating the analysis of behavior is useful across many fields: in neuroscience, to study the neural control of behavior; in ethology and conservation, to study animal behavior and their response to human encroachment; in rehabilitation, to track patients' recovery of motor function; and in helping improve safety in the workplace. Risks are inherent in any application where humans behavior is analyzed, and care must be taken to respect privacy and human rights. Responsible use in research requires following all applicable rules and policies, including filing for permission with the relevant internal review board (IRB), and obtaining written informed consent from human subjects being filmed.

We present additional experimental results (Section 5.7), additional implementation details (Section 5.8), and visualizations (Section 5.9).

5.7 Appendix: Additional Experimental Results

CalMS21 Ablation Study

Similar to the main paper, we evaluate CalMS21 on the behavior classification task 1 train/test split provided by (Sun, Karigo, et al., 2021), and show results on the Mean Average Precision (MAP) across the annotated behavior classes. We use our self-supervised keypoints as input to behavior classification to compare against supervised and other self-supervised baselines.

Effect of Hyperparameters. For all experiments on CalMS21, we use a frame gap of 6 and 10 discovered keypoints. Here, we vary the number of discovered keypoints and frame gap for our model, and apply the learned keypoints to behavior classification (Table 5.6). There are small variations in performance, in particular, the downstream performance generally improves with increasing the number of keypoints, and a frame gap of 6 or 12 works better than larger frame gaps. We note that the number of low confidence background keypoints also increases with the number of discovered keypoints (Figure 5.7), and due to the large proportion of background keypoints, we do not use background keypoints in the 20 keypoints

Hyperparam.	Value	MAP	Hyperparam.	Value	MAP
Frame Gap	6	.852 ± .013	# keypoints	6	.850 ± .017
	12	.862 ± .012		10	.852 ± .013
	30	.839 ± .003		20*	.868 ± .008

Table 5.6: Effect of Hyperparameters on CalMS21. For frame gap experiments, the number of keypoints is set to 10. Frame gap is set to 6 for experiments with a varying number of keypoints. All keypoints, confidence, and covariance are used as inputs, except (*) for the experiments with 20 keypoints, where only high-confidence keypoints are used (11 keypoints) since a high proportion of keypoints are discovered on the background. Mean and standard dev from 5 classifier runs are shown.

case for the classification task. In all cases, we note that we do better than other self-supervised baselines even with bounding box information (MAP = .819) for this task.

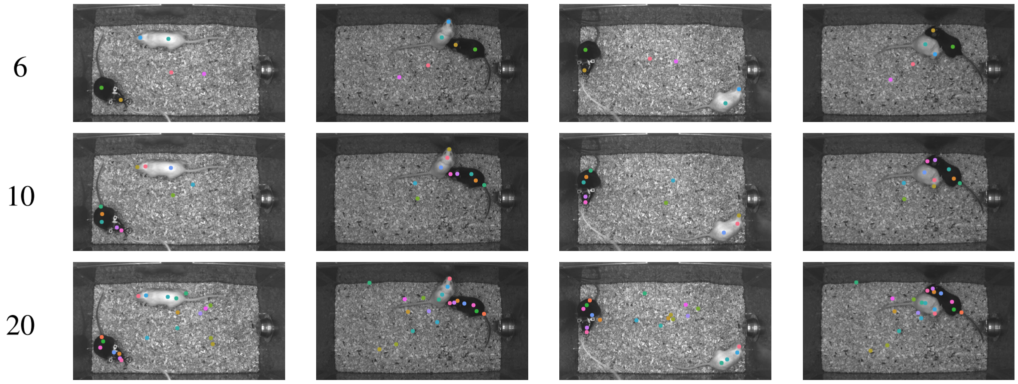


Figure 5.7: Qualitative Results on CalMS21 by varying the number of keypoints. We train the keypoint discovery model with different numbers of discovered keypoints. Each row shows qualitative results with all the keypoints including the background ones. We note that there are 2 background (low-confidence) keypoints for 6 and 10 discovered keypoints, and 9 background keypoints for 20 discovered keypoints.

Varying Amount of Unlabeled Video Data. We vary the amount of input data (unlabelled image pairs) used to train the keypoint discovery model, and observe comparable performance at different amounts of data availability (Table 5.7). In particular, we are able to achieve comparable performance on behavior classification to supervised keypoints (Table 5.9) by using only $7.8k$ input training pairs in our model (approximately 4 minutes of video recorded at 30Hz; approximately 30 minutes of video considering no overlaps on selected image pairs). We note that this experiment

# Training Pairs	Corresponding Video Length (30Hz)	MAP
7.8k	4.3 min	.867 ± .003
18k	10 min	.840 ± .016
26k	14 min	.852 ± .013

Table 5.7: Effect of Varying Training Data Amount for Keypoint Discovery. We train the keypoint discovery model with different amounts of input training image pairs from video. Different training amounts are selected by choosing random video subsets from the full set of CalMS21 training videos. Image pairs are sampled from videos with a gap of 6 frames, and gap between pairs of 7 frames. All keypoints, confidence, and covariance values on 10 discovered keypoints are used. Mean and standard dev from 5 classifier runs are shown.

CalMS21	Pose	Conf	Cov	Ours (MAP)	Reconstruction (MAP)
	✓			.814 ± .007	.695 ± .022
Loss Variation	✓	✓		.857 ± .005	.776 ± .012
	✓	✓	✓	.852 ± .013	.794 ± .008

Table 5.8: Loss Variations on CalMS21. “Ours” represents training with the full objective (reconstruction, rotation equivariance, separation) and “Reconstruction” indicates training with image difference reconstruction only. Mean and standard dev from 5 classifier runs are shown.

is varying the amount of unlabelled data for training the keypoint discovery model (the train/test split for evaluating the behavior classifier stays constant).

Loss Ablation Study. We compare our discovery model trained with the full objective (reconstruction, rotation equivariance, separation) to one trained only on image difference reconstruction (Table 5.8). The rotation equivariance loss is qualitatively important for tracking semantically consistent parts of the mouse (Figure 5.8) and the separation loss prevents the model from predicting keypoints at the center of the image, which are rotationally consistent but do not track semantic body parts. The full objective is important to achieving comparable performance to supervised baselines. We would like to note that the image reconstruction baselines in our main results are also trained with the full objective, except the reconstruction is based on image reconstruction. Additionally, since keypoint locations are not consistent for the reconstruction only case, we note that adding confidence and covariance significantly improves the performance of the reconstruction loss only model (Table 5.8).

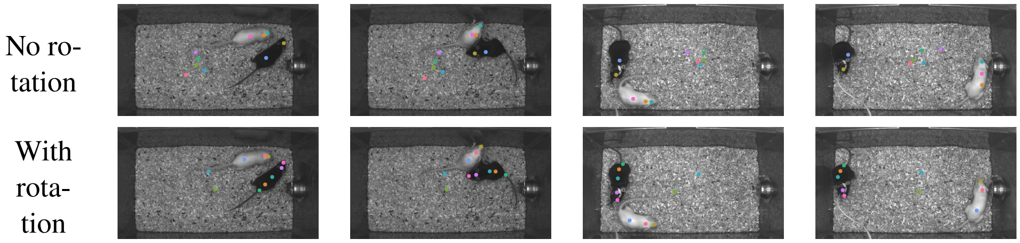


Figure 5.8: Qualitative Results on CalMS21 for loss ablation study. With the full training objective for our discovered keypoints, we are able to track 8/10 keypoints consistently, while without rotation loss, there are only 5/10 tracked keypoints on both mice. Additionally, some of the discovered keypoints without rotation are not semantically consistent (for example, the pink and orange keypoints, two keypoints on the body of the white mouse, shift in order as the white mouse moves around). See quantitative results in Section 5.7.

CalMS21 Per-Class Performance

Our discovered keypoints achieve comparable performance to supervised keypoints when using pose and confidence features from the heatmap (Table 5.9). For both supervised keypoints and our keypoints, the behavior classes with the biggest improvement when adding confidence features is on the “Attack” class, which contains frames with occlusion and motion blur since the mice are moving quickly and chasing/tussling. Heatmap confidence and covariance values provides more information about the detected part (Figure 5.16). For example, when a part is well localized (ex: visible nose of mouse), our keypoint discovery network produces a heatmap with a single high peak with low variance; conversely, when a target part is occluded, the heatmap contains a blurred shape with lower peak value. We note that performance is similar for the supervised keypoints and our keypoints on the “Investigation” and “Mount” classes.

Jellyfish Pulse Detection

The energy efficiency of swimming jellyfish combined with their structural simplicity makes them a good organism for understanding the hydrodynamics of animal propulsion (Costello et al., 2021). In particular, researchers would like to study the relationship between body plan and swim pulse frequency across jellyfish species. This has applications in ethology, hydrodynamics, as well as bio-inspired vehicles. Here, we use Clytia hemisphaerica as our jellyfish species to study jellyfish pulsing during swimming using our discovered keypoints. After videos are recorded from a swimming jellyfish from in a tank, we apply our keypoint discovery model to track keypoints automatically on the jellyfish. We also compute the swim pulse

CalMS21	Pose	Conf	Cov	MAP	Attack AP	Investigation AP	Mount AP
				Fully supervised			
	✓			.856 ± .010	.724 ± .023	.893 ± .005	.950 ± .004
MARS † (Segalin et al., 2020)	✓	✓		.874 ± .003	.790 ± .004	.890 ± .006	.943 ± .004
	✓	✓	✓	.880 ± .005	.804 ± .012	.902 ± .004	.934 ± .006
				Self-supervised			
Jakab et al. (Jakab et al., 2018)	✓			.186 ± .008	.135 ± .019	.254 ± .019	.170 ± .029
	✓			.182 ± .007	.111 ± .016	.217 ± .011	.219 ± .021
Image Recon.	✓	✓		.184 ± .006	.114 ± .006	.209 ± .012	.229 ± .021
	✓	✓	✓	.165 ± .012	.110 ± .016	.218 ± .013	.167 ± .038
	✓			.819 ± .008	.680 ± .028	.861 ± .007	.918 ± .007
Image Recon. bbox†	✓	✓		.812 ± .006	.694 ± .011	.818 ± .016	.923 ± .013
	✓	✓	✓	.812 ± .010	.709 ± .008	.806 ± .019	.922 ± .013
	✓			.814 ± .007	.654 ± .025	.861 ± .003	.925 ± .014
Ours	✓	✓		.857 ± .005	.763 ± .015	.879 ± .009	.928 ± .006
	✓	✓	✓	.852 ± .013	.751 ± .025	.870 ± .009	.935 ± .010

Table 5.9: Per-Class Behavior Classification Results on CalMS21. “Ours” represents classifiers using input keypoints from our discovered keypoints. “conf” represents using the confidence score, and “cov” represents values from the covariance matrix of the heatmap. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 classifier runs are shown.

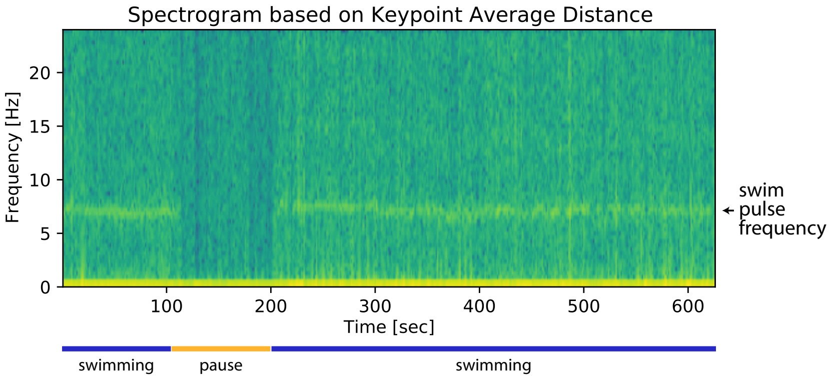


Figure 5.9: Spectrogram from Distance of Discovered Keypoints. From a recorded video of jellyfish swimming at 48Hz, we discover keypoints at each frame using our model and compute a spectrogram based on the average distance between discovered keypoints on the jellyfish.

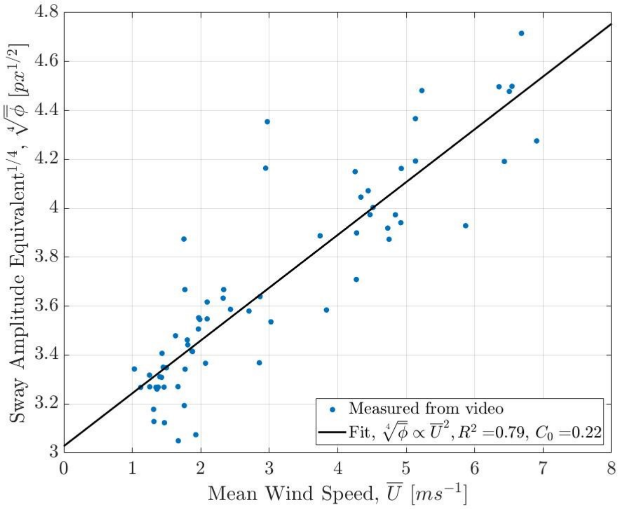


Figure 5.10: Wind Speed Regression from Discovered Keypoints. Mean wind speed, $\bar{U}$, vs. the fourth root of the sway amplitude equivalent measured from the standard deviation of the convex hull area of the 15 discovered keypoints in each clip, based on model from (J. L. Cardona and J. O. Dabiri, 2021). The scatter represents 10-mintute averages of the same data used for training the keypoint model. The black lines represent the best linear regression fit for the proportionality assumption. The proportionality coefficient and the R^2 values are presented in the legend.

frequency by computing the distance between all pairs of our discovered keypoints with high confidence (5 keypoints) and extracting a frequency spectrogram based on average keypoint distance (Figure 5.9). We observe a visible band at the swimming frequency around 7Hz, and we note that between 110 to 200 seconds, the jellyfish is not swimming (floating), and thus the swimming frequency band is not visible in that duration. Since our discovered keypoints are able to detect pulsing, this provides a way to automatically annotate swimming behavior. This method can be applied to videos from other jellyfish species to study the relationship between swimming dynamics and body plan.

Vegetations Wind Speed Regression

Videos of oscillation of tree branches and leaves encode information on local wind conditions, and could function as wind speed sensors. Local wind speed measurements are useful for a variety of tasks, including air pollution monitoring, weather forecasting, and predicting movement of forest fires (J. Cardona, Howland, and J. Dabiri, 2019; J. L. Cardona and J. O. Dabiri, 2021). We use the Vegetation dataset to study the effectiveness of our discovered keypoints for capturing oscillating movement of trees. This dataset consists of videos of swaying trees recorded from an overhead camera from a drone, while the wind speed is measured using an anemometer. We observe that the discovered keypoints from our approach are of different parts of the tree in separate views but are consistent within a single clip, as to capture oscillations of branches/leaves.

We use a physics-based model (J. L. Cardona and J. O. Dabiri, 2021) to study the relationship between oscillations of trees and wind speed. This model defines the relationship between structural oscillation and wind speed as:

$$\sigma \sim I_u \bar{U}$$

where σ is the standard deviation of the amplitude of the structural oscillations, $\bar{U}$ is the mean wind speed, and I_u is the measure of the turbulence intensity of the streamwise component, defined as the standard deviation of the streamwise velocity fluctuations normalized to the mean wind speed. The model requires tracing of the structural oscillations of the branches/leaves, which was previously done manually and we show that the keypoint discovery model can do this automatically. The 15 detected keypoints track these oscillations in a 2D space and a representative measure of these oscillations in both coordinates is calculated using the convex hull area, or the sway amplitude equivalent, ϕ. The average sway amplitude equivalent of the keypoints, $\bar{\phi}$, provides the following proportionality relationship:

$$C_0\sqrt{\bar{\phi}} \sim \bar{U}$$

where C_0 is the coefficient of proportionality. The best regression fit of the experimental data calculated using the least squares method has $R^2 = 0.79$ suggesting there is a good agreement between the proportionality assumption and the experimental results using the keypoint detection model (Figure 5.10).

5.8 Appendix: Additional Implementation Details

Architecture Details Our method uses ResNet-50 (He et al., 2016) as an encoder Φ, GlobalNet (Chen et al., 2017) as a pose decoder Ψ, and a series of convolution blocks as a reconstruction decoder ψ, following the unsupervised keypoint discovery model from (Ryou and Perona, 2021). Architecture details about reconstruction decoder is shown in Table 5.10. We also release the code for mouse and human experiments, so please refer to the code for more implementation details.

Table 5.10: Architecture details about the reconstruction decoder. “Conv_block” refers to a basic convolution block which is composed of 3×3 convolution, batch normalization, and ReLU activation. Note that output size for human experiments is downsampled by a factor of 2 for all the layers.

Type	Input dimension	Output dimension	Output size
Upsampling	-	-	16x16
Conv_block	2048 + # keypoints × 2	1024	16x16
Upsampling	-	-	32x32
Conv_block	1024 + # keypoints × 2	512	32x32
Upsampling	-	-	64x64
Conv_block	512 + # keypoints × 2	256	64x64
Upsampling	-	-	128x128
Conv_block	256 + # keypoints × 2	128	128x128
Upsampling	-	-	256x256
Conv_block	128 + # keypoints × 2	64	256x256
Convolution	64	3	256x256

The hyperparameters for the keypoint discovery model is included in Table 5.11. All models use SSIM image as the reconstruction target, unless stated otherwise. All keypoint discovery models are trained until convergence of the training loss on a NVIDIA V100 Tensor Core GPU. Below, we include a additional details on the keypoint discovery model and downstream task used to evaluate each dataset.

CalMS21. The CalMS21 dataset (Sun, Karigo, et al., 2021) consists of videos and trajectory data from a pair of interacting mice, annotated with behavior labels at each frame by neuroscientists. There is one black mouse and one white mouse engaging in social behaviors, recorded at 1024×570 at 30 Hz. The supervised keypoints provided with CalMS21 are from the MARS detector (Segalin et al., 2020) developed for this dataset, which detects 7 anatomically-defined keypoints for each mouse. For training keypoint discovery, we use a subset of the training split without miniscope cable (26k images), and we use the full train/test split defined by (Sun, Karigo, et al., 2021) on Task 1 for evaluating behavior classification. For behavior classification, we use the same setup (1D Conv Net architecture, hyperparameters, random seeds, data split, etc.) as the CalMS21 dataset benchmarks, except we replace the supervised input keypoints with our discovered keypoints for evaluation. We additionally experiment with adding heatmap confidence and convariance during classification by appending these additional features to input keypoints during classifier training. This dataset is available under the CC-BY-NC-SA license.

MARS-Pose. MARS-Pose is a set of mouse interaction images with human keypoint annotations (Segalin et al., 2020) and these images are recorded in similar recording conditions to CalMS21 (Sun, Karigo, et al., 2021). We use a subset of the images for training (10,50,100,500) and test on the full 1.5k images test set. We evaluate this dataset based on pose estimation performance to the human-annotated keypoints. For the supervised model, we use the stacked hourglass model (Newell, Yang, and Deng, 2016) and for the semi-supervised model, we add a supervised keypoint estimation loss based on MSE to our keypoint discovery framework.

Fly vs. Fly. This dataset consists of videos of two interacting flies (Eyjolfsdottir, S. Branson, et al., 2014) with frame-level behavior annotations. We use the "Aggression" videos from this dataset (144 × 144 at 30 Hz) and use the behaviors with more than 1000 annotated training samples, same as (Sun, Kennedy, et al., 2021). The provided FlyTracker with this dataset computes hand-crafted behavioral features directly from video for behavior classification. Since keypoints may be discovered from any body part, we compute corresponding generic features not based on keypoint identity: speed of every keypoint, acceleration of every keypoint, distance between every pair, and angle between every triplet. Additionally, since the flies are similar in appearance, for the keypoint discovery model when extracting keypoint locations from the heatmaps, we detect 2 max locations for the 2 peaks. We then take the spatial softmax over the region around each max location, instead of taking the spatial softmax over the whole heatmap. In terms of identity, we always use the fly with smaller y values at centroid as the first fly, and the fly with larger y values as the second. For the classifier model, we use the same setup (1D Conv Net architecture (except frame gap in the Conv Net is 1 instead of 2 since flies have faster behaviors), hyperparameters, random seeds, data split, etc.) as the CalMS21 dataset benchmarks, except using the fly features as input to classify annotated behavior at each frame. This dataset is available under the CC0 1.0 Universal license.

Human3.6M. Human 3.6M dataset (Ionescu et al., 2013) is a large-scale dataset containing 3.6 million 3D and 2D human poses with corresponding images. The videos are taken from 4 different viewpoints for 17 scenarios (discussion, taking photo, walking, ...) with the same background. This dataset is available for academic use, and the dataset license is provided by the Human 3.6M authors on the dataset website, link available within (Ionescu et al., 2013). Simplified Human 3.6M dataset, introduced by (Zhang et al., 2018), consists of 6 different activities with

Dataset	# Keypoints	Batch size	Resolution	Frame Gap	Learning Rate
CalMS21	10	5	256	6	0.001
Fly	10	5	256	3	0.001
Human	16	36	128	20	0.001
Jellyfish	10	5	256	20	0.001
Vegetations	15	5	256	60	0.001

Table 5.11: Hyperparameters for Keypoint Discovery.

mostly upright poses by cropping the full image using bounding box. Since our method requires static background assumption, we crop a pair of full images using the same bounding box for training a keypoint discovery model. The final image has 128×128 resolution. We evaluate the pose regression performance on the same testing set from the Simplified Human 3.6M dataset.

Jellyfish. This is an in-house video dataset consisting of a freely swimming Clytia hemisphaerica in a water tank. We train and run our keypoint discovery model on the same 30k frames, recorded at 48Hz, to demonstrate our keypoints on new organisms and on detecting swimming frequency. Since the jellyfish is very small ($\sim$ 50 pix) relative to the size of the image (928×1158), we first use the SSIM image to identify a rough bounding box around the jellyfish (150×150) before re-scaling the input to the keypoint discovery model to 256×256. We note that this step would not be necessary given a GPU with more memory, since the jellyfish would still be visible at higher resolutions. More details on the pulse detection is in Section 5.7.

Vegetations. This is an in-house video dataset captured from a drone flying overhead of an Oak tree as the tree is swaying in the wind, and local wind speed is recorded using an anemometer. The video frames are processed at 512×512 and 120 Hz, and re-scaled to be 256×256 for the keypoint discovery model. The drone may shift slightly over the video recording, and we use existing image alignment methods (Thévenaz, 1998) to align video frames before computing the image difference reconstruction target for our method. More details on the wind speed regression is in Section 5.7.

5.9 Appendix: Visualizations

We present additional visualization results on mouse (Figure 5.11), fly (Figure 5.12), tree (Figure 5.13), and human (Figure 5.14).

Confidence Visualizations. We observe that keypoints discovered on the background and not tracking agent parts generally have very low confidence (Figure 5.16).

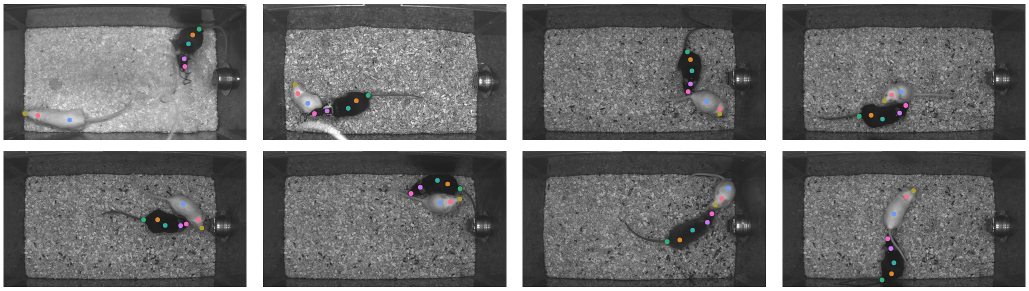


Figure 5.11: Qualitative Results on CalMS21. We observe that keypoints are discovered for noses of both mice and generally along the spine of the mice.

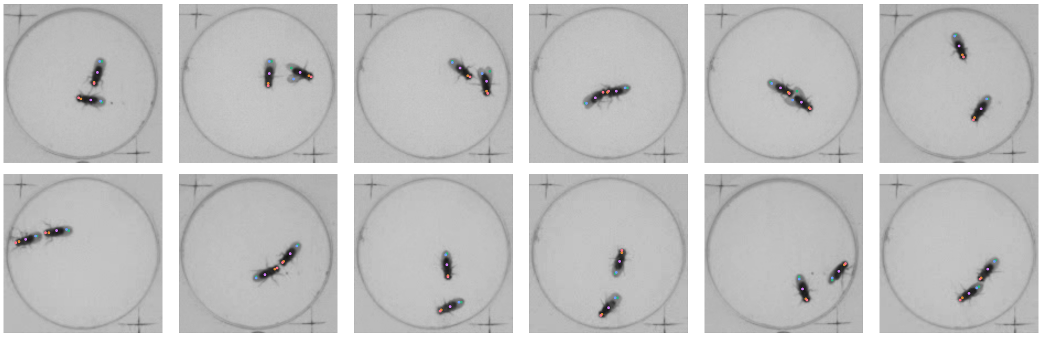


Figure 5.12: Qualitative Results on Fly-vs-Fly. We observe that 3 keypoints are discovered on the body of the fly, with 2 on the wings (one for each wing).

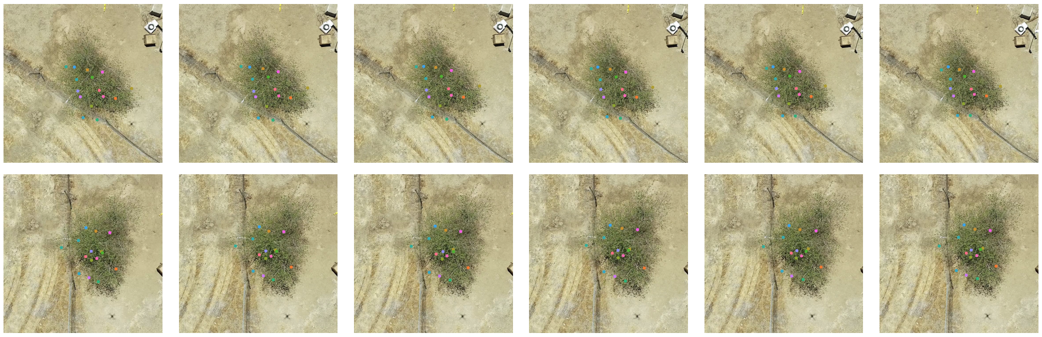


Figure 5.13: Qualitative Results on Vegetations. Each row shows different frames with discovered keypoints from a single video. Our model can discover and track consistent keypoints within the same video.

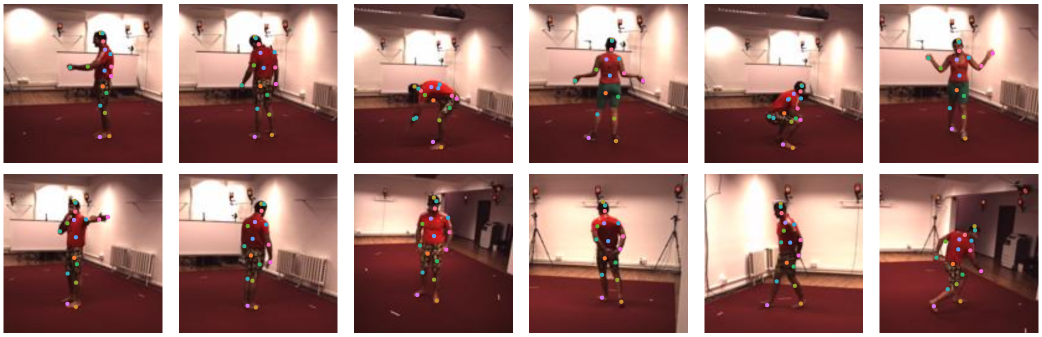


Figure 5.14: Qualitative Results on Simplified Human 3.6M. We observe that keypoints are generally discovered on visible joints and end points of humans, such as head, elbows, hands, upper legs, knees and feet. We note that there is left/right swapping of body parts, since when the human is facing forwards or backwards, keypoints are generally on the same side.

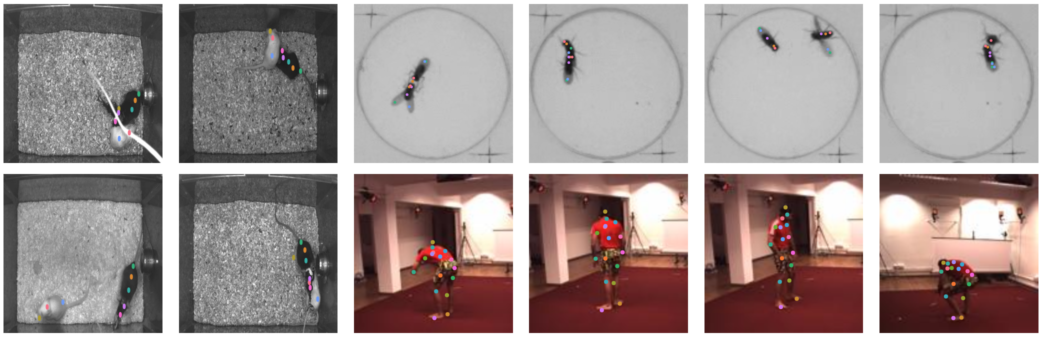


Figure 5.15: Limitations. We visualize examples that are difficult for our model, for example from occlusion/agents being in close proximity (mouse, fly), self-occlusion (human), unusual poses (human, fly), and left-right swapping (human).

This is because heatmaps of background keypoints are not well-localized, and is spread over the image, thus have a low peak value (low confidence). In comparison, discovered keypoints on body parts (such as the nose), is localized to a specific part of the image and has higher peak values. Additionally, confidence values can provide information on occluded parts. For example, for the nose of the white mouse (third column, first row, Figure 5.16), the confidence varies from 0.5 $\sim$ 0.6 when the nose is visible in the first two examples to 0.3 $\sim$ 0.4 when the nose is harder to see in the last two examples.

Challenges. Difficult examples for our model are visualized in Figure 5.15. When there is occlusion, such as in the mouse examples, the keypoint is generally discovered on the visible parts, and when there is heavy occlusion, such as from the miniscope cable, discovered keypoint location may be shifted. This is likely why

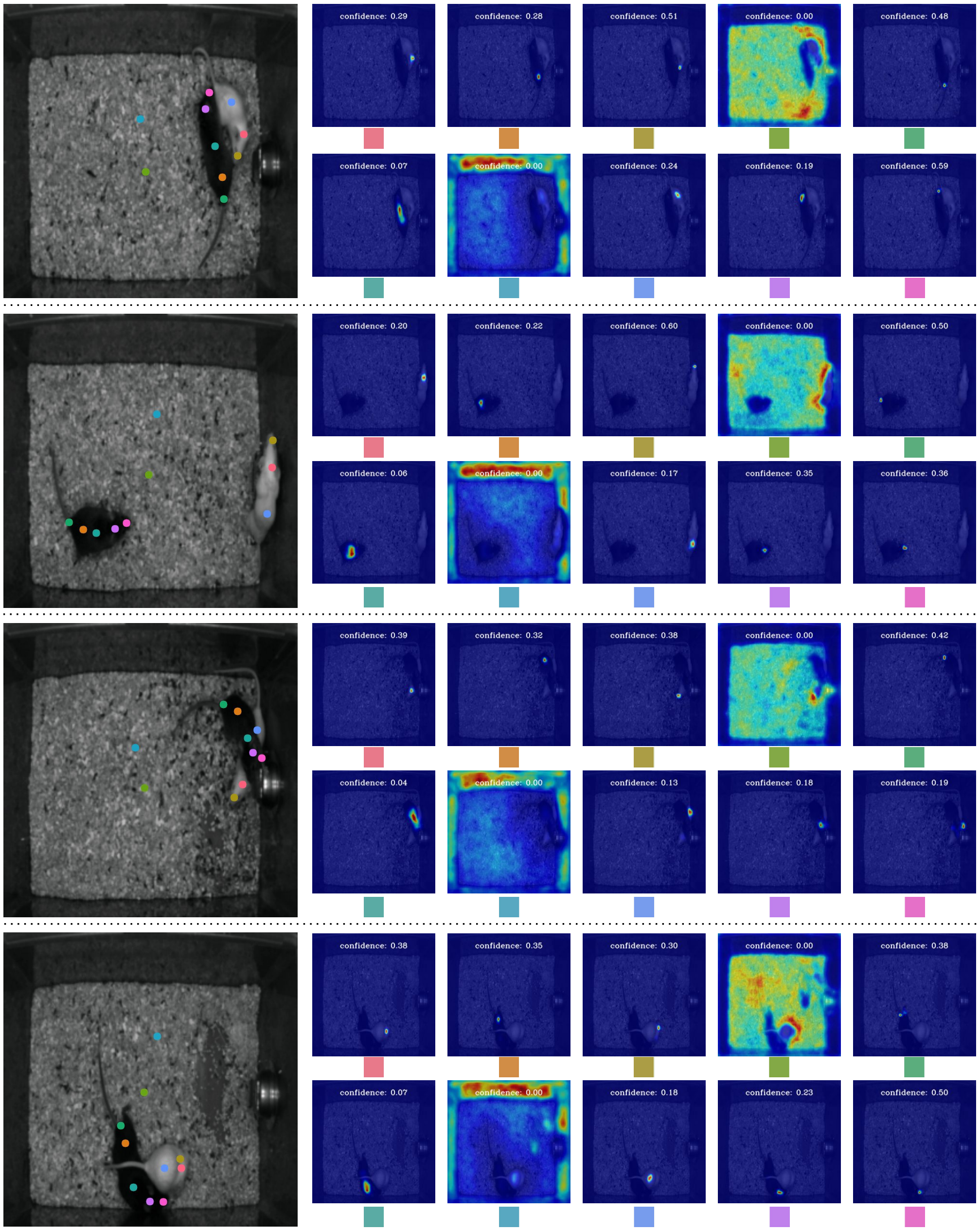


Figure 5.16: Confidence visualization on CalMS21. Confidence score (maximum prediction value) is shown with the normalized heatmap. Background keypoints (fourth on row 1 and second on row 2) have very low confidence.

including additional information from the heatmap, such as confidence (Figure 5.16) is helpful for behavior classification. We can see similar effects on self-occlusion for humans, and also left-right swapping of some keypoints for when humans are facing towards or away from the camera (this has also been observed with other keypoint discovery models (Zhang et al., 2018; Lorenz et al., 2019; Schmidtke et al., 2021)).

Unusual poses may also be difficult, such as when the fly is completely tilted towards the camera in the last column of row 1. Future directions to integrate 3D structure, for instance by using multi-view videos, could help address these issues. Despite this, we note that our current discovery model achieves state-of-the-art results among other self-supervised methods for behavior classification and keypoint regression.

References

Anderson, David J and Pietro Perona (2014). “Toward a science of computational ethology”. In: *Neuron* 84.1, pp. 18–31.

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). “Curriculum Learning”. In: *ICML.*

Berman, Gordon J, Daniel M Choi, William Bialek, and Joshua W Shaevitz (2014). “Mapping the stereotyped behaviour of freely moving fruit flies”. In: *Journal of The Royal Society Interface* 11.99, p. 20140672.

Branson, Kristin, Alice A Robie, John Bender, Pietro Perona, and Michael H Dickinson (2009). “High-throughput ethomics in large groups of Drosophila”. In: *Nature methods* 6.6, pp. 451–457.

Burgos-Artizzu, Xavier P, Piotr Dollár, Dayu Lin, David J Anderson, and Pietro Perona (2012). “Social behavior recognition in continuous video”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1322–1329.

Cardona, Jennifer, Michael Howland, and John Dabiri (2019). “Seeing the Wind: Visual Wind Speed Prediction with a Coupled Convolutional and Recurrent Neural Network”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/a9ad5f2808f68eea468621a04c49efe1-Paper.pdf.

Cardona, Jennifer L and John O Dabiri (2021). “Wind speed inference from environmental flow-structure interactions, part 2: leveraging unsteady kinematics”. In: *arXiv preprint arXiv:2107.09784*.

Chen, Yilun, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun (2017). “Cascaded Pyramid Network for Multi-Person Pose Estimation”. In: *CoRR* abs/1711.07319.

Cheng, Bowen, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang (June 2020). “HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

Costello, John H, Sean P Colin, John O Dabiri, Brad J Gemmell, Kelsey N Lucas, and Kelly R Sutherland (2021). “The hydrodynamics of jellyfish swimming”. In: *Annual Review of Marine Science* 13, pp. 375–396.

Dankert, Heiko, Liming Wang, Eric D Hoopfer, David J Anderson, and Pietro Perona (2009). “Automated monitoring and analysis of social behavior in Drosophila”. In: *Nature methods* 6.4, pp. 297–303.

Egnor, SE Roian and Kristin Branson (2016). “Computational analysis of behavior”. In: *Annual review of neuroscience* 39, pp. 217–236.

Eyjolfsdottir, Eyrun, Kristin Branson, Yisong Yue, and Pietro Perona (2017). “Learning recurrent representations for hierarchical behavior modeling”. In: *ICLR*.

Eyjolfsdottir, Eyrun, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona (2014). “Detecting social actions of fruit flies”. In: *European Conference on Computer Vision*. Springer, pp. 772–787.

Graving, Jacob M, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin (2019). “DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning”. In: *Elife* 8, e47994.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *Proc. IEEE CVPR*.

Hong, Weizhe, Ann Kennedy, Xavier P Burgos-Artizzu, Moriel Zelikowsky, Santiago G Navonne, Pietro Perona, and David J Anderson (2015). “Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning”. In: *Proceedings of the National Academy of Sciences* 112.38, E5351–E5360.

Hsu, Alexander I and Eric A Yttri (2021). “B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors”. In: *Nature communications* 12.1, pp. 1–13.

Ionescu, Catalin, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu (2013). “Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.7, pp. 1325–1339.

Jakab, Tomas, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi (2018). “Unsupervised Learning of Object Landmarks through Conditional Image Generation”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

– (2020). “Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jhuang, Hueihan, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D Steele, and Thomas Serre (2010). “Automated home-cage behavioural phenotyping of mice”. In: *Nature communications* 1.1, pp. 1–10.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). “Perceptual losses for real-time style transfer and super-resolution”. In: *European Conference on Computer Vision*.

Kabra, Mayank, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson (2013). “JAABA: interactive machine learning for automatic annotation of animal behavior”. In: *Nature methods* 10.1, p. 64.

Kim, Yunji, Seonghyeon Nam, In Cho, and Seon Joo Kim (2019). “Unsupervised keypoint learning for guiding class-conditional video prediction”. In: *arXiv preprint arXiv:1910.02027*.

Liu, Jingyuan, Mingyi Shi, Qifeng Chen, Hongbo Fu, and Chiew-Lan Tai (2021). “Normalized Human Pose Features for Human Action Video Alignment”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11521–11531.

Lorenz, Dominik, Leonard Bereska, Timo Milbich, and Björn Ommer (2019). “Unsupervised Part-Based Disentangling of Object Shape and Appearance”. In: *CVPR*.

Luxem, Kevin, Falko Fuhrmann, Johannes Kürsch, Stefan Remy, and Pavol Bauer (2020). “Identifying Behavioral Structure from Deep Variational Embeddings of Animal Motion”. In: *bioRxiv*.

Marstaller, Julian, Frederic Tausch, and Simon Stock (2019). “DeepBees-Building and Scaling Convolutional Neuronal Nets For Fast and Large-Scale Visual Monitoring of Bee Hives”. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0.

Mathis, Alexander, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge (2018). “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning”. In: *Nature Neuroscience*. URL: https://www.nature.com/articles/s41593-018-0209-y.

Minderer, Matthias, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee (2019). “Unsupervised learning of object structure and dynamics from videos”. In: *arXiv preprint arXiv:1906.07889*.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). “Stacked Hourglass Networks for Human Pose Estimation”. In: *Proc. ECCV*.

Nilsson, Simon RO, Nastacia L Goodwin, Jia J Choong, Sophia Hwang, Hayden R Wright, Zane Norville, Xiaoyu Tong, Dayu Lin, Brandon S Bentzley, Neir Eshel, et al. (2020). “Simple Behavioral Analysis (SimBA): an open source toolkit for computer classification of complex social behaviors in experimental animals”. In: *BioRxiv*.

Ohayon, Shay, Ofer Avni, Adam L Taylor, Pietro Perona, and SE Roian Egnor (2013). "Automated multi-day tracking of marked mice for the analysis of social behaviour". In: *Journal of neuroscience methods* 219.1, pp. 10–19.

Pereira, Talmo D, Joshua W Shaevitz, and Mala Murthy (2020). "Quantifying behavior to understand the brain". In: *Nature neuroscience* 23.12, pp. 1537–1549.

Pereira, Talmo D, Nathaniel Tabris, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Z Yan Wang, David M Turner, Grace McKenzie-Smith, Sarah D Kocher, Annegret Lea Falkner, et al. (2020). "SLEAP: Multi-animal pose tracking". In: *BioRxiv*.

Ryou, Serim, Seong-Gyun Jeong, and Pietro Perona (2019). "Anchor Loss: Modulating Loss Scale Based on Prediction Difficulty". In: *The IEEE International Conference on Computer Vision (ICCV)*. DOI: `doi.org/10.1109/ICCV.2019.00609`.

Ryou, Serim and Pietro Perona (2021). "Weakly Supervised Keypoint Discovery". In: arXiv: `2109.13423 [cs.CV]`. URL: `https://arxiv.org/abs/2109.13423`.

Schmidtke, Luca, Athanasios Vlontzos, Simon Ellershaw, Anna Lukens, Tomoki Arichi, and Bernhard Kainz (2021). "Unsupervised Human Pose Estimation Through Transforming Shape Templates". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, pp. 2484–2494.

Segalin, Cristina, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J. Sun, Pietro Perona, David J. Anderson, and Ann Kennedy (2020). "The Mouse Action Recognition System (MARS): a software pipeline for automated analysis of social behaviors in mice". In: *bioRxiv*. DOI: `10.1101/2020.07.26.222299`. URL: `https://www.biorxiv.org/content/early/2020/07/27/2020.07.26.222299`.

Shelhamer, Evan, Jonathan Long, and Trevor Darrell (2017). "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE TPAMI* 39.4, pp. 640–651.

Silva, Brian M de, David M Higdon, Steven L Brunton, and J Nathan Kutz (2020). "Discovery of physics from data: universal laws and discrepancies". In: *Frontiers in artificial intelligence* 3, p. 25.

Simonyan, Karen and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556.

Sun, Jennifer J, Tomomi Karigo, Dipam Chakraborty, Sharada P Mohanty, David J Anderson, Pietro Perona, Yisong Yue, and Ann Kennedy (2021). "The Multi-Agent Behavior Dataset: Mouse Dyadic Social Interactions". In: *arXiv preprint arXiv:2104.02710*.

Sun, Jennifer J, Ann Kennedy, Eric Zhan, David J Anderson, Yisong Yue, and Pietro Perona (2021). "Task programming: Learning data efficient behavior representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2876–2885.

Sun, Jennifer J, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu (2020). "View-invariant probabilistic embedding for human pose". In: *European Conference on Computer Vision*. Springer, pp. 53–70.

Tang, Wei, Pei Yu, and Ying Wu (Sept. 2018). "Deeply Learned Compositional Models for Human Pose Estimation". In: *The European Conference on Computer Vision (ECCV)*.

Thévenaz, Philippe (1998). "StackReg: an ImageJ plugin for the recursive alignment of a stack of images". In: *Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne* 2012.

Thewlis, James, Hakan Bilen, and Andrea Vedaldi (Oct. 2017). "Unsupervised Learning of Object Landmarks by Factorized Spatial Embeddings". In: *The IEEE International Conference on Computer Vision (ICCV)*.

Wang, Jingdong, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, D. Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao (2021). "Deep High-Resolution Representation Learning for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, pp. 3349–3364.

Wang, Zhou, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli (2004). "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *IEEE TRANSACTIONS ON IMAGE PROCESSING* 13.4, pp. 600–612.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional Pose Machines". In: *Proc. IEEE CVPR*.

Wiltschko, Alexander B, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta (2015). "Mapping sub-second structure in mouse behavior". In: *Neuron* 88.6, pp. 1121–1135.

Zhang, Yuting, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee (2018). "Unsupervised Discovery of Object Landmarks as Structural Representations". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. DOI: 10.1109/CVPR.2018.00285.

Chapter 6

CHEMISTRY: MULTI-LABEL CLASSIFICATION MODELS FOR THE PREDICTION OF CROSS-COUPLING REACTION CONDITIONS

The content of this chapter is from the peer-reviewed publication "Multi-Label Classification Models for the Prediction of Cross-Coupling Reaction Conditions" by M. R. Maser*, A. Y. Cui*, S. Ryou*, T. J. DeLano, Y. Yue, and S. E. Reisman, appearing at JCIM 2021, and

"Graph Neural Networks for the Prediction of Substrate-Specific Organic Reaction Conditions" by S. Ryou*, M. R. Maser*, A. Y. Cui*, T. J. DeLano, Y. Yue, and S. E. Reisman, appearing at ICML Workshop on Graph Representation Learning and Beyond 2020.

In Chapter 7, we apply the structural representation of the molecules for predicting the experimental conditions of substrate-specific cross-coupling reaction conditions for the organic chemistry field.

6.1 Abstract

Machine-learned ranking models have been developed for the prediction of substrate-specific cross-coupling reaction conditions. Datasets of published reactions were curated for Suzuki, Negishi, and C–N couplings, as well as Pauson–Khand reactions. String, descriptor, and graph encodings were tested as input representations, and models were trained to predict the set of conditions used in a reaction as a binary vector. Unique reagent dictionaries categorized by expert-crafted reaction roles were constructed for each dataset, leading to context-aware predictions. We find that relational graph convolutional networks and gradient-boosting machines are very effective for this learning task, and we disclose a novel reaction-level graph-attention operation in the top-performing model.

6.2 Introduction

A common roadblock encountered in organic synthesis occurs when canonical conditions for a given reaction type fail in complex molecule settings (Dreher, 2019). Optimizing these reactions frequently requires iterative experimentation that can slow progress, waste material, and add significant costs to research (Blakemore et al., 2018). This is especially prevalent in catalysis, where the substrate-specific nature of reported conditions is often deemed a major drawback, leading to the slow adoption of new methods (Mahatthananchai, Dumas, and Bode, 2012; Dreher, 2019; Blakemore et al., 2018). If, however, a transformation's structure-reactivity relationships (SRRs) were well-known or predictable, this roadblock could be avoided and new reactions could see much broader use in the field (Reid and Sigman, 2018).

Machine learning (ML) algorithms have demonstrated great promise as predictive tools for chemistry domain tasks (Butler et al., 2018). Strong approaches to molecular property prediction (Z. Wu et al., 2018; Yang et al., 2019; Withnall et al., 2020) and generative design (Blaschke et al., 2018; Elton et al., 2019; Prykhodko et al., 2019; Moret et al., 2019) have been developed, particularly in the field of medicinal chemistry (Panteleev, Hua Gao, and Jia, 2018). Some applications have emerged in organic synthesis, geared mainly towards predicting reaction products (Skoraczyński et al., 2017; Coley, Barzilay, et al., 2017), yield (Ahneman et al., 2018; Nielsen et al., 2018; Simón-Vidal et al., 2018; Granda et al., 2018), and selectivity (Hughes, Miller, and Swamidass, 2015; Peng, Duarte, and Paton, 2016; Banerjee, Sreenithya, and Sunoj, 2018; Beker et al., 2019; Zahrt et al., 2019). Significant effort has also been invested in computer-aided synthesis planning (CASP) (Coley, Green, and Jensen, 2018) and the development of retrosynthetic design algorithms (Segler, Preuss, and Waller, 2018; Coley, Green, and Jensen, 2019; Badowski et al., 2020; Nicolaou et al., 2020).

To supplement these tools, initial attempts have been made to predict reaction conditions in the forward direction based on the substrates and products involved (Hanyu Gao et al., 2018). Thus far, studies have focused on global datasets with millions of data points of mixed reaction types. Advantages of this approach include ample training data and the ability to query any transformation with a single model. However, the sparse representation of individual reactions is a major drawback, in that reliable predictions can likely only be expected for the most common reactions and conditions within. This precludes the ability to distinguish subtle variations in substrate structures that lead to different condition requirements, which is critical for SRR modeling.

In recent years, it has become a goal of ours to develop predictive tools to overcome challenges in selecting substrate-specific reaction conditions. Towards this end, we recently reported a preliminary study of graph neural networks (GNNs) as multi-label classification (MLC) models for this task (Ryou* et al., 2020). We selected four high-value reaction types from the cross-coupling literature as testing grounds: Suzuki, C–N, and Negishi couplings, as well as Pauson-Khand reactions (PKRs) (Huerta, Hallinder, and Minidis, 2020). Modeling studies indicated relational graph convolutional networks (R-GCNs) (Schlichtkrull et al., 2017) as uniquely suited for our learning problem. We herein report the full scope of our studies, including improvements to the R-GCN architecture and an alternative tree-based learning approach using gradient-boosting machines (GBMs) (Friedman, 2001).

6.3 Method

A schematic representation of the overall approach is included in Figure 6.1. We direct the reader to our initial report (Ryou* et al., 2020) for additional procedural explanations.[1]

Data acquisition and pre-processing

A summary of the datasets studied here is shown in Table 6.1. Each dataset was manually pre-processed using the following procedure:

1. Reaction data was exported from Reaxys® query results (*Reaxys* 2019; Huerta, Hallinder, and Minidis, 2020).

2. SMILES strings (Weininger, 1988) of coupling partners and major products were identified for each reaction entry (i.e., data point).

3. Condition labels including reagents, catalysts, solvents, temperatures, etc. were extracted for each data point.

4. All unique labels were enumerated into a dataset dictionary, which was sorted by reaction role and trimmed at a threshold frequency to avoid sparsity.

5. Labels were re-indexed within categories and applied to the raw data to construct binary condition vectors for each reaction. We refer to this process as binning.

The reactions studied here were chosen for their ubiquity and value in synthesis, breadth of known conditions, and range of dataset size and chemical space.[2] It

[1] We make our full modeling and data processing code freely available at `https://github.com/slryou41/reaction-gcnn`.

[2] Detailed molecular property distributions for each dataset can be found with our previous studies (Ryou* et al., 2020).

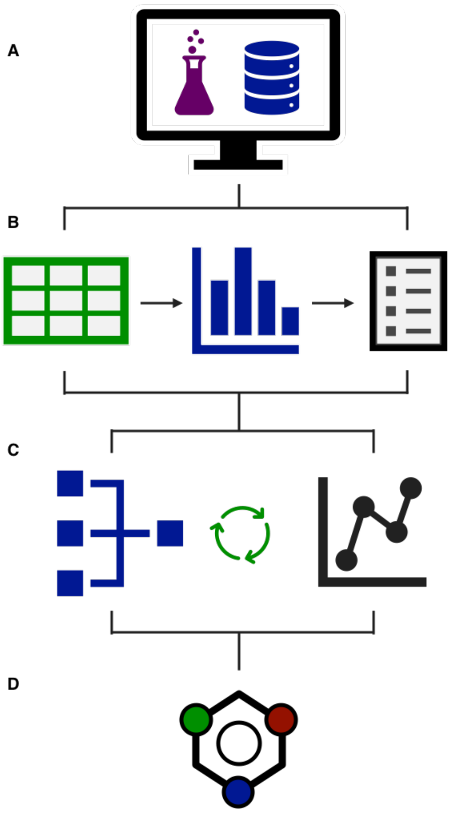
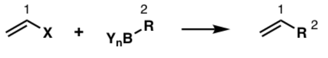
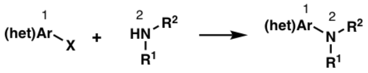
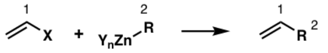
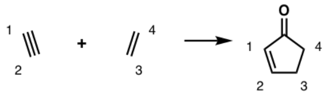


Figure 6.1: Schematic modeling workflow. A) Data gathering. B) Tabulation and dictionary construction. C) Iterative model optimization. D) Inference and interpretation.

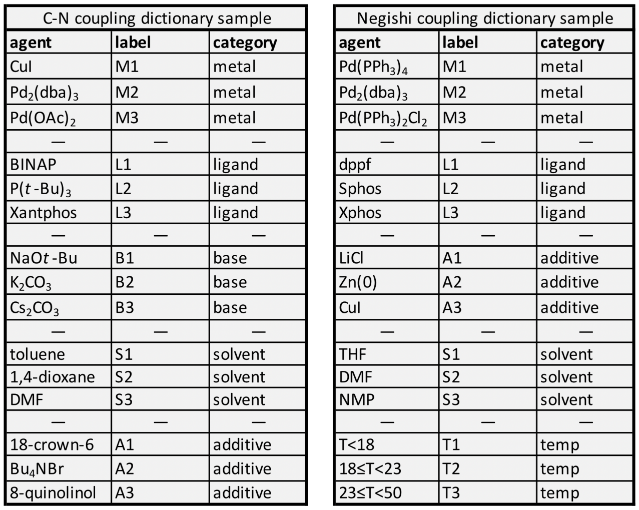
should be noted that certain parameters (e.g. temperature, pressure, etc.) were more fully recorded in some datasets than others. In cases where this data was well-represented, reactions with missing values were simply removed, or in the case of temperature and pressure were assumed to occur ambiently. However, when appropriate, these parameters were dropped from the prediction space to avoid discarding large portions of data.

The Suzuki dataset (Table 6.1, line 1) was obtained from a search of C–C bond-forming reactions between C(sp^2) halides or pseudohalides and organoboron species. Data processing returned 145k reactions with 118 label bins in 5 categories. Similarly, the C–N coupling dataset (line 2) details reactions between aryl (pseudo)halides and amines, with 37k reactions and 205 bins in 5 categories. The Negishi dataset (line 3) contains C–C bond-forming reactions between organozinc compounds and C(sp^2) (pseudo)halides. After processing, this dataset gave 6.4k reactions with 105

Table 6.1: Statistical summary of reaction datasets with Reaxys® queries.

name	depiction	reactions	raw labels	label bins	categories
Suzuki		145,413	3,315	118	5
C–N		36,519	1,528	205	5
Negishi		6,391	492	105	5
PKR		2,749	335	83	8

bins in 5 categories. The PKR dataset (line 4) describes couplings of C–C double bonds with C–C triple bonds to form the corresponding cyclopentenones, containing 2.7k reactions with 83 bins in 8 categories. For all datasets, atom mapping was used as depicted in Table 6.1 to ensure only the desired transformation type was obtained.[3] Samples of the C–N and Negishi label dictionaries are included in Figure 6.2, and full dictionaries for all reactions are provided in the code repository.



C-N coupling dictionary sample

agent	label	category
CuI	M1	metal
$Pd_2(dba)_3$	M2	metal
$Pd(OAc)_2$	M3	metal
—	—	—
BINAP	L1	ligand
$P(t\text{-Bu})_3$	L2	ligand
Xantphos	L3	ligand
—	—	—
NaOt-Bu	B1	base
K_2CO_3	B2	base
Cs_2CO_3	B3	base
—	—	—
toluene	S1	solvent
1,4-dioxane	S2	solvent
DMF	S3	solvent
—	—	—
18-crown-6	A1	additive
Bu_4NBr	A2	additive
8-quinolinol	A3	additive

Negishi coupling dictionary sample

agent	label	category
$Pd(PPh_3)_4$	M1	metal
$Pd_2(dba)_3$	M2	metal
$Pd(PPh_3)_2Cl_2$	M3	metal
—	—	—
dppf	L1	ligand
Sphos	L2	ligand
Xphos	L3	ligand
—	—	—
LiCl	A1	additive
Zn(0)	A2	additive
CuI	A3	additive
—	—	—
THF	S1	solvent
DMF	S2	solvent
NMP	S3	solvent
—	—	—
$T<18$	T1	temp
$18\leq T<23$	T2	temp
$23\leq T<50$	T3	temp

Figure 6.2: Samples of categorized reaction dictionaries for C-N and Negishi datasets.

[3] Given their relative frequency and to maintain consistent formatting, intramolecular couplings were dropped from the first three reactions but were retained for the PKR dataset.

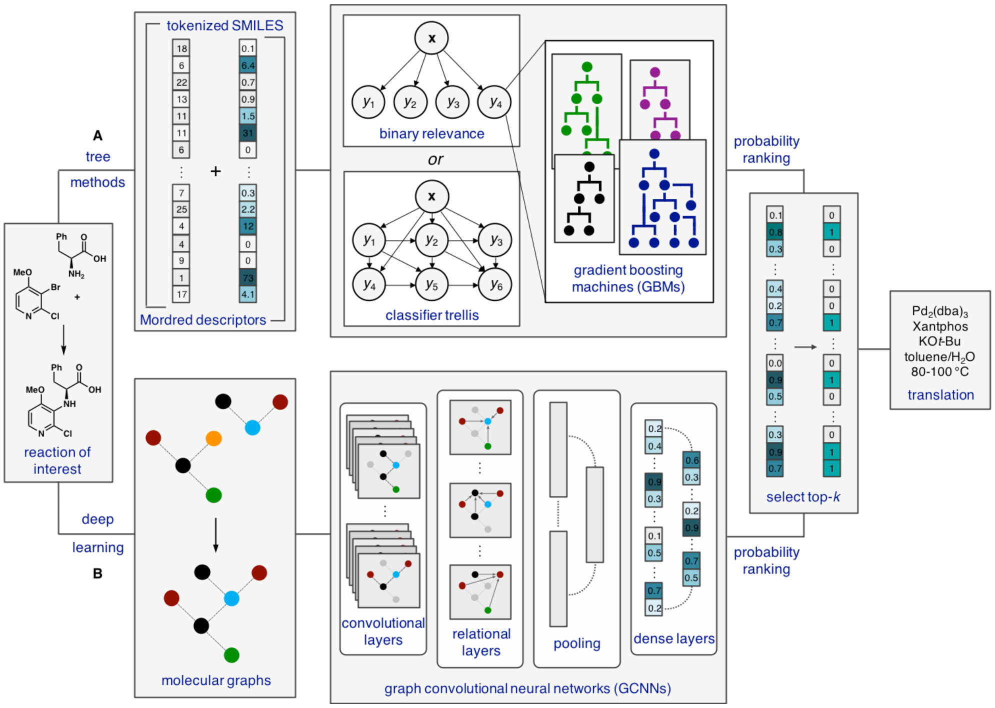


Figure 6.3: Schematic modeling workflow. A) Tree-based methods. String and descriptor vectors for each molecule in a reaction are concatenated and used as inputs to gradient-boosting machines (GBMs). B) Deep learning methods. Molecular graphs are constructed for each molecule in a reaction, which are passed as inputs to a graph convolutional neural network (GCNN). Both model types predict probability rankings for the full reaction dictionary, which are sorted by reaction role and translated to the final output.

Model setup

For each dataset, an 80/10/10 train/validation/test split was used in modeling. Training and test sets were kept consistent between model types for sake of comparability. Model inputs were prepared as reactant/product structure tuples, with encodings tailored to each learning method. Models were trained using binary cross-entropy loss to output probability scores for all reagent/condition labels in the reaction dictionary. The top-*k* ranked labels in each dictionary category were selected as the final prediction, where *k* is user-determined.

We define an accurate prediction as one where the ground-truth label appears in the top-*k* predicted labels. Given the variable class-imbalance in each dictionary category (Cui et al., 2019; Ryou* et al., 2020), accuracy is evaluated at the categorical level as follows:

$$A_c = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[\hat{Y}_i \cap Y_i] \,, \tag{6.1}$$

where $\hat{Y}_i$ and Y_i are the sets of top-k predicted and ground truth labels for the i-th sample in category c, respectively. The correct instances are summed and divided by the number of samples in the test set, N, to give the overall test accuracy in the category, or A_c (X.-Z. Wu and Z.-H. Zhou, 2017).

As a general measure of a model's performance, we calculate its average error reduction (AER) from a baseline predictor ("dummy") that always predicts the top-k most frequently occurring dataset labels in each category:

$$\mathrm{AER} = \frac{1}{C}\sum_{c=1}^{C} \frac{A_c^g - A_c^d}{1 - A_c^d} \,, \tag{6.2}$$

where A_c^g and A_c^d are the accuracies of the GNN and dummy model in the c-th category, respectively, and C is the number of categories in the dataset dictionary. AER represents a model's average improvement over the naive approach that one might use as a starting point for experimental optimization. In other words, AER is the percent of the gap closed between the naive model and a perfect predictor of accuracy 1.

Model construction

Both tree- and deep learning methods were explored for this MLC task (Figure 6.3), and their individual development is discussed below.

Gradient-boosting machines

GBMs are decision-tree-based learning algorithms that are popular in the ML literature for their performance in modeling numerical data (Natekin and Knoll, 2013). We explored several string and descriptor-based encodings as numerical inputs (see SI) and found that a hybrid encoding scheme provided the greatest learnability (Figure 6.3A).[4] The hybrid inputs are a concatenation of tokenized SMILES strings for each molecule in a reaction (coupling partners and products), further concatenated with molecular property vectors obtained from the Mordred descriptor calculator (Moriwaki et al., 2018). GBMs consistently outperformed other tree-based learners such as random forests (RFs) (Breiman, 2001), perhaps owing to their use of

[4] Gradient boosting was implemented using Microsoft's LightGBM (Ke et al., 2017).

sequential ensembling to improve in poor-performance regions (Natekin and Knoll, 2013).

In our GBM experiments, a separate classifier was trained for all bins in a dataset dictionary, predicting whether or not they should be present in each reaction. Two general strategies have been developed for related MLC tasks, known as the binary relevance method (BM) and classifier chaining (CC) (Zhang and Z.-H. Zhou, 2014). The BM approach considers each classifier as an independent model, predicting the label of its bin irrespective of the others. Conversely, CCs make predictions sequentially, taking the output of each label as an additional input for the next one, where the optimal order of chaining is a learned parameter (Jesse Read et al., 2009). While the BM approach is significantly simpler from a computational perspective, CCs offer the potential for higher accuracy by modeling interdependencies between labels (Zhang and Z.-H. Zhou, 2014).

We saw this as prudent in our studies given that reagent correlations are frequently observed in synthesis. Some examples relevant to this work include using a polar protic solvent with an inorganic base, excluding exogenous ligand when using a preligated metal source, setting the temperature below the boiling point of the solvent, etc. We decided to explore both methods, testing BM against a modern update to CCs introduced by Read and coworkers known as classifier trellises (CTs) (J. Read et al., 2015). In the CT method, instead of fully sequential propagation, models are fit in a pre-defined grid structure (the “trellis”), where the output of each prediction is passed to multiple downstream classifiers at once (Figure 6.3A, center). This eliminates the cost of chain structure discovery, while still benefiting from nesting predictions (Zhang and Z.-H. Zhou, 2014).

The ordering of a CT is enforced algorithmically starting from a seed label, chosen randomly or by expert intervention. From Read et al. (J. Read et al., 2015), the trellis is populated by maximizing the mutual information (MI) between source and target labels (s_ℓ) at each step (ℓ) as follows:

$$s_\ell = \text{argmax}_{k \in S} \sum_{j \in \mathsf{pa}(\ell)} I(y_j; y_k) \, , \tag{6.3}$$

where S and $\mathsf{pa}(\ell)$ are the set of remaining labels and the available trellis structure at the current step, respectively, and y_j and y_k are the j-th and k-th target labels, respectively. Here, $I(y_j; y_k)$ represents the MI between labels j and k based on their co-occurrences in the dataset. The matrix of *all* pairwise label dependencies

$I(Y_j;Y_k)$ is constructed as below:

$$I(Y_j;Y_k) = \sum_{y_j \in \mathcal{Y}_j} \sum_{y_k \in \mathcal{Y}_k} p(y_j, y_k) \log \left(\frac{p(y_j, y_k)}{p(y_j)p(y_k)} \right), \tag{6.4}$$

where $p(y_j, y_k)$, and $p(y_j)$ and $p(y_k)$ are the joint and marginal probability mass functions of y_j and y_k, respectively. $\mathcal{Y}_j$ and $\mathcal{Y}_k$ represent the possible values y_j and y_k can each assume, which for our task of binary classification are both $\{0,1\}$. Full MI matrices and optimized trellises for each dataset are included in the SI, and an example is discussed with the results.

Relational graph convolutional networks

Originally reported by Schlichtkrull et al. (Schlichtkrull et al., 2017), R-GCNs are a subclass of message passing neural networks (MPNNs) (Gilmer et al., 2017) that explicitly model relational data such as molecular graphs. This is achieved by constructing sets of *relation* operations, where each relation $r \in \mathcal{R}$ is specific to a type and direction of edge between connected nodes. In our setting, the relations operate on atom-bond-atom triples using a learned, sparse weight matrix $\mathbf{W}_r^{(l)}$ in each layer l (Schlichtkrull et al., 2017). In a propagation step, each current node representation $h_i^{(l)}$ is transformed with all relation-specific neighboring nodes $h_j^{(l)}$ and summed over all relations such that:

$$h_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^{(l)} h_j^{(l)} + \mathbf{W}_0^{(l)} h_i^{(l)} \right), \tag{6.5}$$

where $\mathcal{N}_i^r$ is the set of applicable neighbors and σ is an element-wise non-linearity, for us the tanh. The self-relation term $\mathbf{W}_0^{(l)} h_i^{(l)}$ is added to preserve local node information, and $c_{i,r}$ is a normalization constant (Schlichtkrull et al., 2017). Unlike traditional GCNs, R-GCNs intuitively model edge-based messages in local subgraph transformations (Schlichtkrull et al., 2017). This is potentially very powerful for reaction learning in that information on edge types (i.e., single, double, triple, aromatic, and cyclic bonds) is crucial for modeling reactivity.

Here, we extend the R-GCN architecture with an additional graph attention layer (GAL) at the final readout step inspired by graph attention networks (GATs) from Veličković (Veličković et al., 2018) and Busbridge (Busbridge et al., 2019). As described by Veličković et al. (Veličković et al., 2018), GALs compute pair-wise node attention coefficients α_{ij} for each node h_i in a graph and its neighbors h_j.

Table 6.2: Prediction accuracy for all model types on the Suzuki dataset.

dataset	top-k	category	dummy	BM-GBM	CT-GBM	R-GCN	AR-GCN
Suzuki	top-1	**AER**	-	–0.1919[a]	–0.1766[b]	0.2767	**0.3115**
		metal	0.3777	0.5665	0.5526	0.6306	**0.6499**
		ligand	0.8722	0.8306	0.8490	0.9036	**0.9081**
		base	0.3361	0.4831	0.4912	0.5455	**0.5896**
		solvent	0.6377	0.6683	0.6725	0.7049	**0.7217**
		additive	0.9511	0.8903	0.8870	**0.9624**	0.9621
	top-3	**AER**	-	0.4119	0.3741	0.4936	**0.5246**
		metal	0.6744	0.8534	0.8453	0.8482	**0.8597**
		ligand	0.9269	0.9639	0.9602	0.9644	**0.9676**
		base	0.7344	**0.8347**	0.8309	0.8123	0.8285
		solvent	0.8013	0.8619	0.8564	0.8836	**0.8897**
		additive	0.9771	0.9844	0.9828	**0.9934**	0.9931

[a] AER excluding *additive*: 0.0710. [b] AER excluding *additive*: 0.1073.

Two nodes' features are first transformed *via* a shared weight matrix **W**, the results of which are concatenated before applying a learned weight vector and softmax normalization. The final update rule is simply a linear combination of α_{ij} with the newly transformed node vectors ($\mathbf{W}h_j$), summed over all neighboring nodes and averaged over a set of parallel attention mechanisms (Veličković et al., 2018).

In our recent studies (Ryou* et al., 2020), we observed that existing relational GATs (R-GATs) (Busbridge et al., 2019) using atom-level attention layers were less effective for our task than simple R-GCNs.[5] Inspired nonetheless by the chemical intuition of graph attention, we adapted existing GALs to construct a *reaction-level* attention mechanism. Instead of pair-wise α_{ij}, we construct self-attention coefficients α_i^m for all nodes h_i^m in a molecular graph $\boldsymbol{h}^m = \{h_0^m, h_1^m, ..., h_L^m\}$. As in GATs, we take a linear combination of α_i^m for all L nodes in $\boldsymbol{h}^m$ after further transformation by matrix $\mathbf{W}^g$:

$$\alpha_i^m = \sigma\left(\mathbf{W}^s h_i^m\right), \forall i \in \{1, 2, ..., L\}, \tag{6.6}$$

$$h_i^a = \alpha_i^m \mathbf{W}^g h_i^m \ , \tag{6.7}$$

where $\mathbf{W}^s$ is the learned attention weight matrix, σ is the sigmoid activation function, and h_i^a is the updated node representation. The convolved graphs $\boldsymbol{h}^a = \{h_0^a, h_1^a, ..., h_L^a\}$ for each molecule m are then concatenated on the node feature axis

[5] We found it necessary to reduce the hidden dimension of R-GATs to avoid excessive memory requirements relative to other GCNs (Veličković et al., 2018), and thus do not make a direct comparison of their performance.

to give an overall reaction representation $\boldsymbol{h}^r$ that we term the attended reaction graph (ARG):

$$\mathrm{ARG} = \boldsymbol{h}^r = \left[{}_{m=1}^{M}\boldsymbol{h}_{m^a}\right], \tag{6.8}$$

where M is the number of molecules in the reaction (reactants and products) and $||$ denotes concatenation. Similar to the attention mechanism above, reaction-level attention coefficients α_i^r are then constructed and linearly combined with the ARG nodes h_i^r after transformation with $\mathbf{W}^v$. The final readout vector $\boldsymbol{v}_r$ is obtained from the attention layer by summative pooling over the nodes:

$$\alpha_i^r = \sigma\left(\mathbf{W}^r h_i^r\right), \forall i \in \{1, 2, ..., H\}, \tag{6.9}$$

$$\boldsymbol{v}_r = \sum_{i=1}^{H} \alpha_i^r \mathbf{W}^v h_i^r\ , \tag{6.10}$$

where H is the total number of nodes and $\mathbf{W}^r$ is the reaction attention weight matrix. This construction differs from standard R-GCNs, which output readout vectors for individual molecules and concatenate them to form the ultimate reaction representation. Altogether, we term our hybrid architecture as an *attended relational graph convolutional network*, or AR-GCN.

In all deep learning experiments, with or without attention, the reaction vector readouts were passed to a multi-layer perceptron (MLP) of depth = 2.[6] The final prediction is made as a single output vector with one entry for each label in the reaction dictionary, and the result is translated as described in section 6.3.

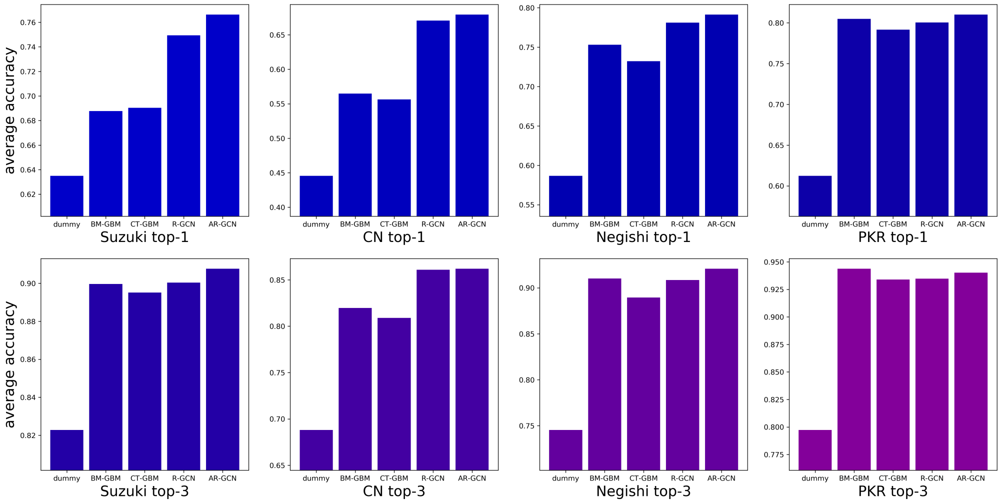
6.4 Results and discussion

Model performance

Our modeling pipeline was first tested on the Suzuki coupling dataset, the largest of the four. Table 6.2 summarizes top-1 and top-3 categorical accuracies (Equation 6.1) and AERs (Equation 6.2) for the following models: GBMs with no trellising (BM-GBM), GBMs with trellising (CT-GBM), standard R-GCNs as reported by Schlichtkrull et al. (R-GCN) (Ryou* et al., 2020; Schlichtkrull et al., 2017), our AR-GCNs developed here (AR-GCN), and the dummy predictor as a baseline control (dummy).

For this dataset, GCN models significantly outperformed GBMs across categories for both top-1 and top-3 predictions. While GBMs actually gave negative top-1 AERs over baseline, these scores were dominated by the *additive* contribution;

[6] All NN models were implemented using the Chainer Chemistry (ChainerChem) deep learning library (Tokui et al., 2015).



Suzuki top-1

CN top-1

Negishi top-1

PKR top-1

Suzuki top-3

CN top-3

Negishi top-3

PKR top-3

Figure 6.4: Average top-1 and top-3 categorical accuracies for each model across the four datasets.

excluding this category the BM- and CT-GBMs gave modest 7% and 11% AERs, respectively. Despite struggling with top-1 predictions, GBMs gave significant AERs for top-3, with BM-GBMs at 41% and CT-GBMs at 37%. The AR-GCNs gave the best accuracy of all models, providing 31% and 52% top-1 and top-3 AERs, respectively. AR-GCNs gave roughly 3% AER gain over the R-GCN in both top-1 and top-3 predictions, demonstrating the value of the added attention layer.

A few interesting categorical trends can be seen across model types. For instance, models provide the best error reduction (ER $= \frac{A_c^g - A_c^d}{1 - A_c^d}$, see Equation 6.2) in the *metal* category, with the AR-GCN at 44% and 57% for top-1 and top-3, respectively. Similarly, models perform well in the *base* category, where the AR-GCN gave the best top-1 ER and BM-GBMs gave the best top-3 ER. Less consistent ERs between top-1 and top-3 predictions were obtained for the remaining three categories. For example, with *solvent*s, the AR-GCN improved baseline by 23% in top-1 predictions, but 44% in top-3. Likewise, for AR-GCN *ligand* predictions, a 28% ER was obtained for top-1 versus a 56% gain in top-3. Finally, although the baseline *additive* accuracy is high as the majority of reactions are `null` in this category, the AR-GCN still gave a 23% top-1 ER and a 70% top-3 ER.

The trends and differences between top-1 and top-3 performance gains are reflective of the frequency distributions in each label category. (Ryou* et al., 2020) These intuitively resemble long-tail or Pareto-type distributions (Newman, 2005), with the bulk of the cumulative density contained in a small number of bins and the

Table 6.3: Prediction accuracy for all model types on the C–N, Negishi, and PKR datasets.

dataset	top-*k*	category	dummy	BM-GBM	CT-GBM	R-GCN	AR-GCN
C–N	top-1	**AER**	-	–0.0416[a]	–0.0929[b]	0.3453	**0.3604**
		metal	0.2452	0.4972	0.4822	0.5989	**0.6162**
		ligand	0.5219	0.5891	0.5964	0.6981	**0.7068**
		base	0.2479	0.5125	0.5111	0.5932	**0.6066**
		solvent	0.3219	0.4730	0.4655	0.5647	**0.5674**
		additive	0.8904	0.7526	0.7265	0.8984	**0.8997**
	top-3	**AER**	-	0.3835	0.3430	0.5391	**0.5471**
		metal	0.6526	0.8017	0.7813	0.8479	**0.8490**
		ligand	0.6647	0.8033	0.8050	0.8605	**0.8688**
		base	0.6400	0.8081	0.7997	**0.8452**	0.8370
		solvent	0.5677	0.7549	0.7348	0.7973	**0.7997**
		additive	0.9156	0.9304	0.9237	0.9534	**0.9559**
Negishi	top-1	**AER**	-	0.3492	0.2466	0.4439	**0.4565**
		metal	0.2887	0.5606	0.5363	0.6555	**0.6730**
		ligand	0.7879	0.8078	0.8013	0.8724	**0.8772**
		temperature	0.3317	0.6721	**0.6769**	0.6188	0.6507
		solvent	0.6938	0.8546	0.8498	0.8868	**0.8915**
		additive	0.8309	0.8708	0.7964	**0.8724**	0.8644
	top-3	**AER**	-	0.6141	0.4949	0.6590	**0.6833**
		metal	0.5008	0.7868	0.7625	0.8086	**0.8517**
		ligand	0.8549	0.9548	0.9144	0.9522	**0.9553**
		temperature	0.5885	**0.9160**	0.9031	0.8517	0.8708
		solvent	0.8788	0.9418	0.9273	**0.9537**	**0.9537**
		additive	0.9043	0.9515	0.9402	**0.9761**	0.9729
PKR	top-1	**AER**	-	**0.4396**	0.3744	0.3973	0.4199
		metal	0.4302	**0.7901**	0.7863	0.7132	0.7057
		ligand	0.8792	**0.9389**	0.9237	0.9057	0.9094
		temperature	0.2830	0.5916	0.5878	0.6528	**0.6642**
		solvent	0.3321	0.6450	0.5992	0.6792	**0.6981**
		activator	0.6906	0.8168	0.8092	0.8415	**0.8491**
		CO (g)	0.7245	0.8817	0.8779	0.8717	**0.8868**
		additive	0.9057	**0.9084**	0.8893	0.8906	0.8491
		pressure	0.6528	**0.8664**	0.8588	0.8491	0.8491
	top-3	**AER**[c]	-	**0.7205**	0.6877	0.6844	0.7145
		metal	0.7132	**0.9504**	0.9389	0.9057	0.8906
		ligand	0.9019	0.9924	0.9924	0.9849	**0.9962**
		temperature	0.5962	0.8550	0.8473	0.8528	**0.8604**
		solvent	0.5925	0.8855	0.8473	0.8679	**0.8981**
		activator	0.8830	0.9542	0.9466	**0.9774**	**0.9774**
		CO (g)	1.0000	1.0000	1.0000	1.0000	1.0000
		additive	0.9321	**0.9885**	0.9809	0.9698	0.9736
		pressure	0.9623	0.9809	0.9847	**0.9849**	**0.9849**

[a] AER excluding *additive*: 0.2623. [b] AER excluding *additive*: 0.2578. [c] Excludes *CO(g)*.

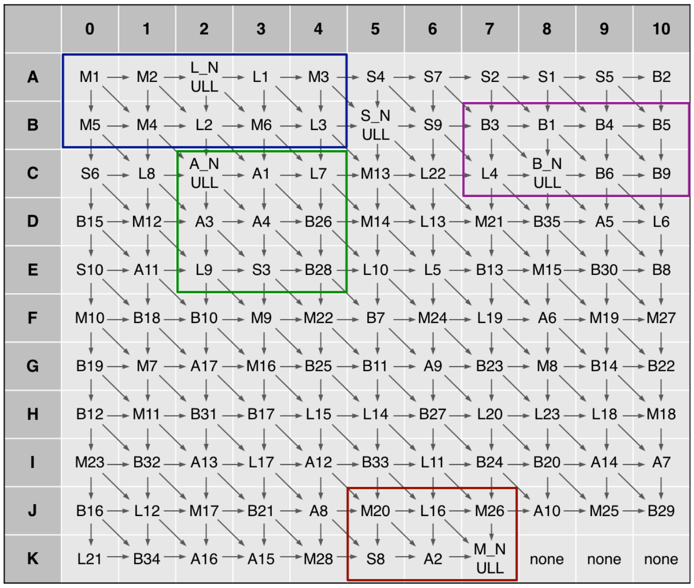
remaining bins supporting smaller frequencies. The distribution shapes are likely to influence the relative top-1 and top-3 AERs, where the highly skewed distributions could be more difficult to improve over baseline.

Having demonstrated the utility of our predictive framework, we turned to the remaining datasets to assess its scope. Modeling results for C–N, Negishi, and PKRs are detailed in Table 6.3 and Figure 6.4. Notable observations for each dataset are discussed below.

C–N coupling. Similar to the Suzuki results, the AR-GCN was the top performer for C–N couplings in almost all categories, and slightly higher AERs were observed overall. The AR-GCN afforded 36% and 55% top-1 and top-3 AERs, respectively, again providing slight gains over R-GCNs at 35% and 54%. As above, GBMs struggled with this relatively large dataset (36,519 reactions) due to difficulties with the *additive* category. Models again made strong improvements in the *metal* and *base* categories, but also gave consistently strong gains for *ligand*s and *solvent*s, especially for top-3 predictions. For example, the AR-GCN returned top-3 ERs of 57% for *metal*s, 61% for *ligand*s, 55% for *base*s, and 54% for *solvent*s. Note that these ERs correspond to very high accuracies (A_c) of 85%, 87%, 84%, and 80%, respectively.

Negishi coupling. The highest AERs of all modeling experiments came with the Negishi dataset. The AR-GCN again gave the strongest performance, with top-1 and top-3 AERs of 46% and 68%, respectively. However, the R-GCN and even GBM models gave the highest accuracies in some categories. Interestingly, BM- and CT-GBMs performed significantly better than the GCNs for *temperature* predictions, though the strongest ER for most models came from the *solvent* category.

PKR. For the PKR dataset—the smallest of the four—simple BM-GBMs gave the best top-1 AER at 44%, followed closely by the AR-GCN at 42%. Similarly for top-3 predictions, these models gave AERs of 72% and 71%, respectively. Compared to the other reactions, GCNs are perhaps more prone to overfitting this small of a dataset (K. Zhou et al., 2020), making tree-based modeling more suitable. It is interesting to note that in general for PKRs, the GCN models were better at predicting physical parameters like *temperature*, *solvent*, and *CO(g)* atmosphere, whereas GBMs gave better performance for reaction components such as *metal*, *ligand*, and *additive*.



	0	1	2	3	4	5	6	7	8	9	10
A	M1	M2	L_NULL	L1	M3	S4	S7	S2	S1	S5	B2
B	M5	M4	L2	M6	L3	S_NULL	S9	B3	B1	B4	B5
C	S6	L8	A_NULL	A1	L7	M13	L22	L4	B_NULL	B6	B9
D	B15	M12	A3	A4	B26	M14	L13	M21	B35	A5	L6
E	S10	A11	L9	S3	B28	L10	L5	B13	M15	B30	B8
F	M10	B18	B10	M9	M22	B7	M24	L19	A6	M19	M27
G	B19	M7	A17	M16	B25	B11	A9	B23	M8	B14	B22
H	B12	M11	B31	B17	L15	L14	B27	L20	L23	L18	M18
I	M23	B32	A13	L17	A12	B33	L11	B24	B20	A14	A7
J	B16	L12	M17	B21	A8	M20	L16	M26	A10	M25	B29
K	L21	B34	A16	A15	M28	S8	A2	M_NULL	none	none	none

Figure 6.5: Optimized prediction trellis for the Suzuki dataset.

Interpretability

Given the results described above, we sought an understanding of the chemical features informing our predictions. Tree-based learning is often favored in this regard in that feature importances (FIs) can be directly extracted from models. We found that FIs for our GBMs were roughly uniform across the SMILES regions of the encodings (see SI for detailed rankings). The most informative physical descriptors from the Mordred vectors pertained to two classes: topological charge distributions (Galvez et al., 1994) correlated with local molecular dipoles; and Moreau–Broto autocorrelations (Moreau and Broto, 1980) weighted by polarizability, ionization potential, and valence electrons. The latter class is particularly intriguing as they are calculated from molecular graphs in what have been described as atom-pair convolutions (Hollas, 2003), not unlike the GCN models used here (Schlichtkrull et al., 2017).

An advantage to using CTs is the ability to extract their MI matrices and trellis structures for interpretation (J. Read et al., 2015). The optimized trellis for the Suzuki CT-GBMs is included in Figure 6.5, where several chemically intuitive features can be noted:

1. Block A0–B4 (blue): The results of `M1` ($Pd(PPh_3)_4$) and `M2` ($Pd(OAc)_2$) are used to predict exogenous ligand (L_NULL), and if `M4` ($Pd(dppf)Cl_2 \cdot DCM$) or `M5` ($Pd(PPh_3)_2Cl_2$) are used. Based on this, `L1` (PPh_3) and `L2` (Sphos) are predicted, then feeding models of `M3` ($Pd(dppf)Cl_2$), `M6` ($Pd_2(dba)_3$), and `L3`

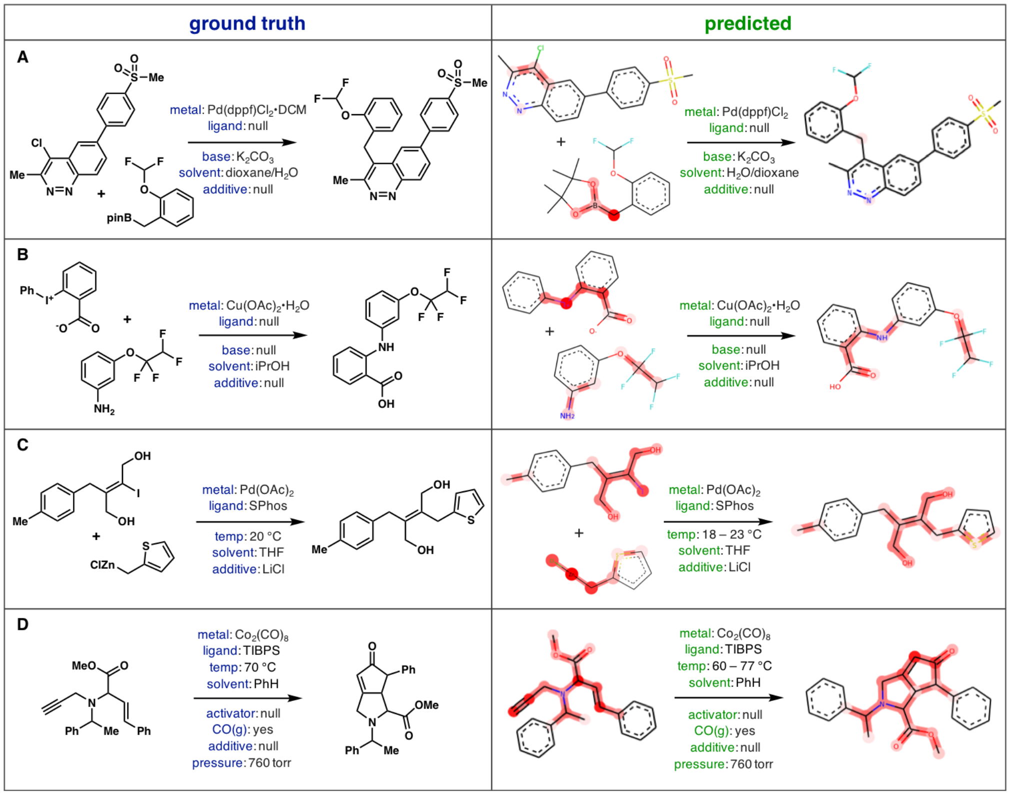


Figure 6.6: AR-GCN attention weight visualization and prediction examples from randomly chosen reactions in each dataset. Darker highlighting indicates higher attention.

(Xphos).

2. Block C2–E4 (green): Whether or not an additive is needed (A_NULL) informs the use of A1 (Bu_4NBr), A3 (LiCl), and A4 (HCl). Interestingly, acid A4 then informs the prediction of S3 (MeOH) and bases B26 ($KF \cdot 2H_2O$) and B28 ($LiOH \cdot H_2O$).

3. Block B7–C10 (purple): Several bases are connected, where the predictions of B3 (K_3PO_4) and B1 (K_2CO_3) inform whether or not a base is even needed (B_NULL). These subsequently feed classifiers of B4 (Cs_2CO_3) and B6 (CsF), which in turn feed B5 ($NaHCO_3$) and B9 (KF).

4. Block J5–K7 (red): The prediction of M20 ($NiCl_2 \cdot$ DME) informs the use of L16 (di-*t*Bubpy), commonly employed in Ni-catalyzed cross-couplings. These results then feed the prediction of another Ni source, M26 ($NiNO_3 \cdot H_2O$).

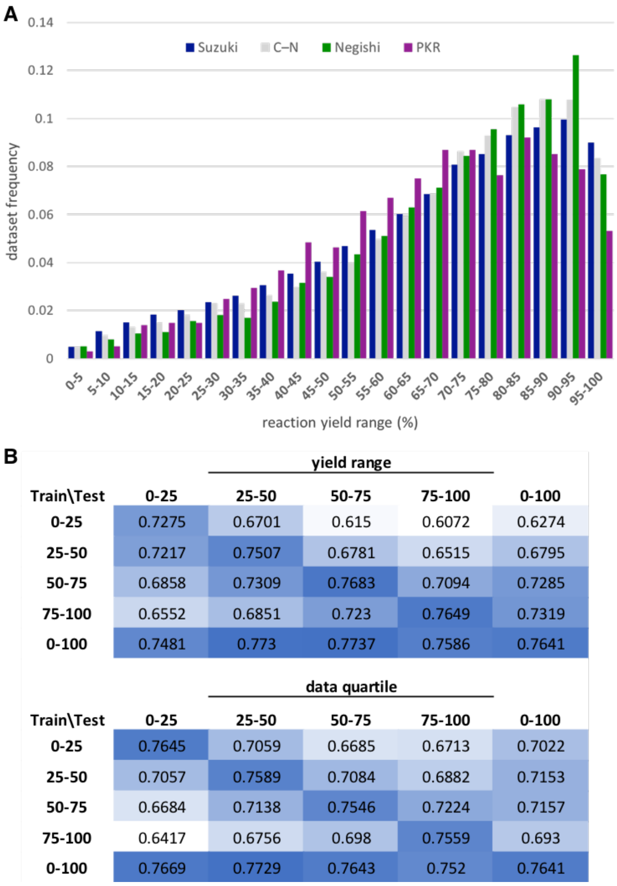
As a control experiment, we withheld the propagated predictions from the CT-GBMs to test whether the MI was actually being used. Indeed, model accuracy dropped off markedly, even below baseline in some categories (see SI). While this suggests that CT-GBMs do learn reagent correlations, the sharp performance loss may also indicate overfitting to this information (J. Read et al., 2015). Further studies are necessary to uncover the optimal molecule featurization in combination with CTs, though the results here suggest their promise in modeling structured reaction data.

For AR-GCNs, a valuable interpretability feature lies in the learned feature weights α_i^r (Equation 6.9). Intuitively, the weights represent the model's assignment of importance on an atom, as they re-scale node features in the final graph layer before inference. When extracted, the weights can be mapped back onto a molecule's atoms and displayed by color scale using RDKit (Landrum, 2016). This gives a visual interpretation of the functional groups most heavily informing the predictions. Example visualizations from random reactions in each dataset with the resulting AR-GCN predictions are included in Figure 6.6.

In the Suzuki example (Figure 6.6A), the attention is dominated by the sp^3 carbon bearing the Bpin group, with additional contributions from the bis-*o*-substituted heteroaryl-chloride and its cinnoline nitrogen, all of which could be reasonably expected to influence reactivity. It is interesting that weights on the *o*-difluoromethoxy group, the sulfone, and the majority of the product are suppressed, perhaps indicating that an alkyl nucleophile is sufficient to predict the required conditions. The AR-GCN predictions are correct in each category besides the *metal*, where the model erroneously identifies the metal source Pd(dppf)Cl$_2$ instead of its ground truth DCM adduct Pd(dppf)Cl$_2$ $\cdot$ DCM.

Conversely, the weights in the C–N coupling example are more evenly distributed (Figure 6.6B). Intuitively, the chemically active iodonium benzoate is given strong attention in the electrophile, as is the nucleophilic aniline nitrogen. Here, the *m*-tetrafluoroethoxy group is also weighted significantly and these groups are given similar attention in the product. All categories are predicted correctly in this example, though three of them are `null`.

The Negishi example (Figure 6.6C) is an interesting C(sp^3)–C(sp^2) coupling of a fully substituted alkenyl-iodide and thiophenyl-methylzinc chloride. Similar to A, the strongest weights correspond to the sp^3 nucleophilic carbon, though similarly strong attention is distributed over the electrophilic alkene including the pendant alcohols. These weights are again reflected in the product and all five condition



yield range

Train\Test	0-25	25-50	50-75	75-100	0-100
0-25	0.7275	0.6701	0.615	0.6072	0.6274
25-50	0.7217	0.7507	0.6781	0.6515	0.6795
50-75	0.6858	0.7309	0.7683	0.7094	0.7285
75-100	0.6552	0.6851	0.723	0.7649	0.7319
0-100	0.7481	0.773	0.7737	0.7586	0.7641

data quartile

Train\Test	0-25	25-50	50-75	75-100	0-100
0-25	0.7645	0.7059	0.6685	0.6713	0.7022
25-50	0.7057	0.7589	0.7084	0.6882	0.7153
50-75	0.6684	0.7138	0.7546	0.7224	0.7157
75-100	0.6417	0.6756	0.698	0.7559	0.693
0-100	0.7669	0.7729	0.7643	0.752	0.7641

Figure 6.7: Performance dependence on reaction yield. A) Distribution of reaction yields for the four datasets. B) AR-GCN average top-1 A_c values for Suzuki predictions when trained and tested in different yield ranges (top) and dataset quartiles arranged by yield (bottom).

categories are predicted correctly, including *temperature* and use of a LiCl *additive*.

Lastly, an intramolecular PKR (Figure 6.6D) showed the most uniformly distributed attention of the four examples. Still, the strongest weights are given to the participating alkyne and alkene, with additional emphasis on the amino ester bridging group. Weights are similarly distributed in the product, though strongest attention is intuitively assigned to the newly formed enone. Here, all 8 categories are predicted correctly including the use of an ambient carbon monoxide atmosphere (*CO(g)* and *pressure*).

Yield Analysis

Having explored our models' chemical feature learning, we lastly investigated the effect of reaction yield, as it is a critical feature of synthesis data. Unsurprisingly, plotting the distribution of reaction yields in each dataset showed a uniformly strong bias towards high-yielding reactions (Figure 6.7A). Given the skewness of the data in this regard, we hypothesized that models would perform best at predicting conditions for high-yielding reactions.

We divided the dataset into quartiles by reaction yield and re-trained the AR-GCN with each sub-set, subsequently testing in each region and on the full test set (Figure 6.7B). Intuitively, models trained in any yield range tended to give highest accuracy when tested in the same range, occupying the confusion matrix diagonal in Figure 6.7B (top). To our surprise, however, the standard model trained on the full dataset gave consistently high accuracies, regardless of the test set (bottom row).

Since the yield bins contain varying amounts of data, we re-split the dataset, again ordered by yield but with equal sub-set sizes (Figure 6.7B bottom). A similar trend was observed where the highest accuracies were found on the diagonal and bottom row of the confusion matrix. Interestingly, the worst performing model was that trained in the highest yield range and tested in the lowest. We recognize that making "inaccurate" predictions on low-yielding reactions offers an avenue for predictive reaction optimization and future studies will explore this objective.

6.5 Conclusions

In summary, we present a multi-label classification approach to predicting experimental reaction conditions for organic synthesis. We successfully model four high-value reaction types using expert-crafted label dictionaries: Suzuki, C–N, and Negishi couplings, and Pauson–Khand reactions. We explore and optimize two model classes: gradient boosting machines and graph convolutional networks. We find that GCN models perform very well in larger datasets, while GBMs show success for smaller datasets.

We report the first use of classifier trellises in molecular machine learning, and in some cases find them to give improvements over binary relevance algorithms by incorporating label correlations in modeling. We introduce a novel reaction-level graph attention mechanism that provides significant accuracy gains when coupled with relational GCNs, and construct a hybrid GCN architecture called *attended relational GCNs*, or AR-GCNs. We further provide an analytical framework for the chemical interpretation of our models, extracting the trellis structures and mutual information matrices of the CT-GBMs, and visualizing the attention weights assigned in AR-GCN predictions.

Experimental studies are currently underway assessing the feasibility of model predictions on novel reactions. Additionally, efforts to apply our modeling framework to less-structured reaction types such as oxidations and reductions are ongoing. Future studies will address the interplay between structure representation and classifier chaining, as well as the extension of our reaction attention mechanism to other tasks. We expect the work herein to be very informative for future condition prediction studies, a highly valuable but underexplored learning task.

Acknowledgement

We thank Prof Pietro Perona for mentorship guidance and helpful project discussions, and Chase Blagden for help structuring the GBM experiments. Fellowship support was provided by the NSF (M.R.M., T.J.D Grant No. DGE- 1144469). S.E.R. is a Heritage Medical Research Institute Investigator. Financial support from Research Corporation is warmly acknowledged.

References

Ahneman, Derek T., Jesús G. Estrada, Shishi Lin, Spencer D. Dreher, and Abigail G. Doyle (Apr. 2018). “Predicting reaction performance in C–N cross-coupling using machine learning”. en. In: *Science* 360.6385, pp. 186–190. ISSN: 0036-8075, 1095-9203. DOI: `10.1126/science.aar5169`. URL: `http://science.sciencemag.org/content/360/6385/186` (visited on 11/20/2018).

Badowski, Tomasz, Ewa P. Gajewska, Karol Molga, and Bartosz A. Grzybowski (Jan. 2020). “Synergy Between Expert and Machine-Learning Approaches Allows for Improved Retrosynthetic Planning”. en. In: *Angewandte Chemie International Edition* 59.2, pp. 725–730. ISSN: 1433-7851, 1521-3773. DOI: `10.1002/anie.201912083`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201912083` (visited on 07/01/2020).

Banerjee, Sayan, A. Sreenithya, and Raghavan B. Sunoj (July 2018). “Machine learning for predicting product distributions in catalytic regioselective reactions”. en. In: *Physical Chemistry Chemical Physics* 20.27, pp. 18311–18318. ISSN: 1463-9084. DOI: `10.1039/C8CP03141J`. URL: `https://pubs.rsc.org/en/content/articlelanding/2018/cp/c8cp03141j` (visited on 08/06/2018).

Beker, Wiktor, Ewa P. Gajewska, Tomasz Badowski, and Bartosz A. Grzybowski (Mar. 2019). “Prediction of Major Regio-, Site-, and Diastereoisomers in Diels–Alder Reactions by Using Machine-Learning: The Importance of Physically Meaningful Descriptors”. en. In: *Angewandte Chemie International Edition* 58.14, pp. 4515–4519. ISSN: 1433-7851, 1521-3773. DOI: `10.1002/anie.201806920`. URL:

https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201806920 (visited on 07/01/2020).

Blakemore, David C., Luis Castro, Ian Churcher, David C. Rees, Andrew W. Thomas, David M. Wilson, and Anthony Wood (Apr. 2018). “Organic synthesis provides opportunities to transform drug discovery”. en. In: *Nature Chemistry* 10.4, pp. 383–394. ISSN: 1755-4330, 1755-4349. DOI: 10.1038/s41557-018-0021-z. URL: http://www.nature.com/articles/s41557-018-0021-z (visited on 07/02/2020).

Blaschke, Thomas, Marcus Olivecrona, Ola Engkvist, Jürgen Bajorath, and Hongming Chen (2018). “Application of Generative Autoencoder in De Novo Molecular Design”. en. In: *Molecular Informatics* 37.1-2, p. 1700123. ISSN: 1868-1751. DOI: 10.1002/minf.201700123. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201700123 (visited on 09/04/2019).

Breiman, Leo (2001). “Random Forests”. In: *Machine Learning* 45, pp. 5–32. URL: https://doi.org/10.1023/A:1010933404324.

Busbridge, Dan, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla (Apr. 2019). “Relational Graph Attention Networks”. en. In: *arXiv:1904.05811 [cs, stat]*. URL: http://arxiv.org/abs/1904.05811 (visited on 04/30/2020).

Butler, Keith T., Daniel W. Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh (July 2018). “Machine learning for molecular and materials science”. en. In: *Nature* 559.7715, pp. 547–555. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0337-2. URL: https://www.nature.com/articles/s41586-018-0337-2 (visited on 04/23/2019).

Coley, Connor W., Regina Barzilay, Tommi S. Jaakkola, William H. Green, and Klavs F. Jensen (May 2017). “Prediction of Organic Reaction Outcomes Using Machine Learning”. In: *ACS Central Science* 3.5, pp. 434–443. ISSN: 2374-7943. DOI: 10.1021/acscentsci.7b00064. URL: https://doi.org/10.1021/acscentsci.7b00064 (visited on 11/18/2019).

Coley, Connor W., William H. Green, and Klavs F. Jensen (May 2018). “Machine Learning in Computer-Aided Synthesis Planning”. In: *Accounts of Chemical Research* 51.5, pp. 1281–1289. ISSN: 0001-4842. DOI: 10.1021/acs.accounts.8b00087. URL: https://doi.org/10.1021/acs.accounts.8b00087 (visited on 07/30/2018).

– (June 2019). “RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application”. en. In: *Journal of Chemical Information and Modeling* 59.6, pp. 2529–2537. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/acs.jcim.9b00286. URL: https://pubs.acs.org/doi/10.1021/acs.jcim.9b00286 (visited on 04/28/2020).

Cui, Yin, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie (Jan. 2019). “Class-Balanced Loss Based on Effective Number of Samples”. en. In: *arXiv:1901.05555 [cs]*. URL: http://arxiv.org/abs/1901.05555 (visited on 05/12/2020).

Dreher, Spencer D. (2019). “Catalysis in medicinal chemistry”. en. In: *Reaction Chemistry & Engineering* 4.9, pp. 1530–1535. ISSN: 2058-9883. DOI: 10.1039/C9RE00067D. URL: http://xlink.rsc.org/?DOI=C9RE00067D (visited on 07/02/2020).

Elton, Daniel C., Zois Boukouvalas, Mark D. Fuge, and Peter W. Chung (2019). “Deep learning for molecular design—a review of the state of the art”. en. In: *Molecular Systems Design & Engineering* 4.4, pp. 828–849. ISSN: 2058-9689. DOI: 10.1039/C9ME00039A. URL: http://xlink.rsc.org/?DOI=C9ME00039A (visited on 05/08/2020).

Friedman, Jerome H. (Oct. 2001). “Greedy Function Approximation: A Gradient Boosting Machine”. en. In: *The Annals of Statistics* 29.5, pp. 1189–1232. URL: https://www.jstor.org/stable/2699986?seq=1#metadata_info_tab_contents.

Galvez, J., R. Garcia, M. T. Salabert, and R. Soler (May 1994). “Charge Indexes. New Topological Descriptors”. en. In: *Journal of Chemical Information and Modeling* 34.3, pp. 520–525. ISSN: 1549-9596. DOI: 10.1021/ci00019a008. URL: https://pubs.acs.org/doi/abs/10.1021/ci00019a008 (visited on 03/04/2020).

Gao, Hanyu, Thomas J. Struble, Connor W. Coley, Yuran Wang, William H. Green, and Klavs F. Jensen (Nov. 2018). “Using Machine Learning To Predict Suitable Conditions for Organic Reactions”. In: *ACS Central Science* 4.11, pp. 1465–1476. ISSN: 2374-7943. DOI: 10.1021/acscentsci.8b00357. URL: https://doi.org/10.1021/acscentsci.8b00357 (visited on 11/18/2019).

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl (June 2017). “Neural Message Passing for Quantum Chemistry”. en. In: *arXiv:1704.01212 [cs]*. URL: http://arxiv.org/abs/1704.01212.

Granda, Jarosław M., Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin (July 2018). “Controlling an organic synthesis robot with machine learning to search for new reactivity”. en. In: *Nature* 559.7714, pp. 377–381. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0307-8. URL: https://www.nature.com/articles/s41586-018-0307-8 (visited on 07/20/2018).

Hollas, Boris (2003). “An Analysis of the Autocorrelation Descriptor for Molecules”. en. In: *Journal of Mathematical Chemistry* 33.2, pp. 91–101. URL: https://link.springer.com/article/10.1023/A:1023247831238.

Huerta, Fernando, Samuel Hallinder, and Alexander Minidis (July 2020). *Machine Learning to Reduce Reaction Optimization Lead Time – Proof of Concept with Suzuki, Negishi and Buchwald-Hartwig Cross-Coupling Reactions*. en. preprint ChemRxiv.12613214. URL: https://chemrxiv.org/articles/Machine_Learning_to_Reduce_Reaction_Optimization_Lead_Time_Proof_of_Concept_with_Suzuki_Negishi_and_Buchwald-

Hartwig_Cross-Coupling_Reactions/12613214/1 (visited on 07/16/2020).

Hughes, Tyler B., Grover P. Miller, and S. Joshua Swamidass (July 2015). “Modeling Epoxidation of Drug-like Molecules with a Deep Machine Learning Network”. In: *ACS Central Science* 1.4, pp. 168–180. ISSN: 2374-7943. DOI: 10.1021/acscentsci.5b00131. URL: https://doi.org/10.1021/acscentsci.5b00131 (visited on 12/18/2018).

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu (2017). “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 3146–3154. URL: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

Landrum, Gregory A. (2016). *RDKit: Open-Source Cheminformatics Software*. (accessed Nov 20, 2016). (Visited on 11/20/2016).

Mahatthananchai, Jessada, Aaron M. Dumas, and Jeffrey W. Bode (Oct. 2012). “Catalytic Selective Synthesis”. en. In: *Angewandte Chemie International Edition* 51.44, pp. 10954–10990. ISSN: 14337851. DOI: 10.1002/anie.201201787. URL: http://doi.wiley.com/10.1002/anie.201201787 (visited on 05/11/2020).

Moreau, G and P Broto (1980). “The Autocorrelation of a Topological Structure: A New Molecular Descriptor”. In: *New Journal of Chemistry* 4.6, pp. 359–360.

Moret, Michael, Lukas Friedrich, Francesca Grisoni, Daniel Merk, and Gisbert Schneider (Nov. 2019). “Generating Customized Compound Libraries for Drug Discovery with Machine Intelligence”. en. In: ISSN: doi:10.26434/chemrxiv.10119299.v1. DOI: 10.26434/chemrxiv.10119299.v1. URL: https://chemrxiv.org/articles/Generating_Customized_Compound_Libraries_for_Drug_Discovery_with_Machine_Intelligence/10119299 (visited on 11/08/2019).

Moriwaki, Hirotomo, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi (Feb. 2018). “Mordred: a molecular descriptor calculator”. In: *Journal of Cheminformatics* 10.1, p. 4. ISSN: 1758-2946. DOI: 10.1186/s13321-018-0258-y. URL: https://doi.org/10.1186/s13321-018-0258-y (visited on 01/28/2019).

Natekin, Alexey and Alois Knoll (2013). “Gradient boosting machines, a tutorial”. en. In: *Frontiers in Neurorobotics* 7. ISSN: 1662-5218. DOI: 10.3389/fnbot.2013.00021. URL: http://journal.frontiersin.org/article/10.3389/fnbot.2013.00021/abstract (visited on 07/16/2020).

Newman, M. E. J. (Sept. 2005). "Power laws, Pareto distributions and Zipf's law". en. In: *Contemporary Physics* 46.5, pp. 323–351. ISSN: 0010-7514, 1366-5812. DOI: 10.1080/00107510500052444. URL: http://arxiv.org/abs/cond-mat/0412004 (visited on 09/07/2020).

Nicolaou, Christos A., Ian A. Watson, Mark LeMasters, Thierry Masquelin, and Jibo Wang (Apr. 2020). "Context Aware Data-Driven Retrosynthetic Analysis". en. In: *Journal of Chemical Information and Modeling*. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/acs.jcim.9b01141. URL: https://pubs.acs.org/doi/abs/10.1021/acs.jcim.9b01141 (visited on 04/28/2020).

Nielsen, Matthew K., Derek T. Ahneman, Orestes Riera, and Abigail G. Doyle (Apr. 2018). "Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning". en. In: *Journal of the American Chemical Society* 140.15, pp. 5004–5008. ISSN: 0002-7863, 1520-5126. DOI: 10.1021/jacs.8b01523. URL: http://pubs.acs.org/doi/10.1021/jacs.8b01523 (visited on 08/01/2018).

Panteleev, Jane, Hua Gao, and Lei Jia (Sept. 2018). "Recent applications of machine learning in medicinal chemistry". en. In: *Bioorganic & Medicinal Chemistry Letters* 28.17, pp. 2807–2815. ISSN: 0960894X. DOI: 10.1016/j.bmcl.2018.06.046. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960894X1830547X (visited on 05/08/2020).

Peng, Qian, Fernanda Duarte, and Robert S. Paton (Nov. 2016). "Computing organic stereoselectivity – from concepts to quantitative calculations and predictions". en. In: *Chemical Society Reviews* 45.22, pp. 6093–6107. ISSN: 1460-4744. DOI: 10.1039/C6CS00573J. URL: https://pubs.rsc.org/en/content/articlelanding/2016/cs/c6cs00573j (visited on 08/05/2018).

Prykhodko, Oleksii, Simon Viet Johansson, Panagiotis-Christos Kotsias, Josep Arús-Pous, Esben Jannik Bjerrum, Ola Engkvist, and Hongming Chen (Dec. 2019). "A de novo molecular generation method using latent vector based generative adversarial network". In: *Journal of Cheminformatics* 11.1, p. 74. ISSN: 1758-2946. DOI: 10.1186/s13321-019-0397-9. URL: https://doi.org/10.1186/s13321-019-0397-9 (visited on 01/08/2020).

Read, J., L. Martino, P. Olmos, and D. Luengo (June 2015). "Scalable Multi-Output Label Prediction: From Classifier Chains to Classifier Trellises". en. In: *Pattern Recognition* 48.6, pp. 2096–2109. ISSN: 00313203. DOI: 10.1016/j.patcog.2015.01.004. URL: http://arxiv.org/abs/1501.04870 (visited on 01/24/2020).

Read, Jesse, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank (2009). "Classifier Chains for Multi-label Classification". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 254–269. ISBN: 978-3-642-04174-7.

Reaxys (2019). (accessed on May 13, 2019). URL: https://new.reaxys.com/ (visited on 05/13/2019).

Reid, Jolene P. and Matthew S. Sigman (Oct. 2018). “Comparing quantitative prediction methods for the discovery of small-molecule chiral catalysts”. en. In: *Nature Reviews Chemistry* 2.10, pp. 290–305. ISSN: 2397-3358. DOI: 10.1038/s41570-018-0040-8. URL: http://www.nature.com/articles/s41570-018-0040-8 (visited on 07/16/2020).

Ryou*, Serim, Michael R. Maser*, Alexander Y. Cui*, Travis J. DeLano, Yisong Yue, and Sarah E. Reisman (2020). “Graph Neural Networks for the Prediction of Substrate-Specific Organic Reaction Conditions”. en. In: *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRLB)*. URL: http://arxiv.org/abs/2007.04275.

Schlichtkrull, Michael, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling (Oct. 2017). “Modeling Relational Data with Graph Convolutional Networks”. en. In: *arXiv:1703.06103 [cs, stat]*. URL: http://arxiv.org/abs/1703.06103 (visited on 05/01/2020).

Segler, Marwin H. S., Mike Preuss, and Mark P. Waller (Mar. 2018). “Planning chemical syntheses with deep neural networks and symbolic AI”. en. In: *Nature* 555.7698, pp. 604–610. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature25978. URL: http://www.nature.com/doifinder/10.1038/nature25978 (visited on 08/05/2018).

Simón-Vidal, Lorena, Oihane García-Calvo, Uxue Oteo, Sonia Arrasate, Esther Lete, Nuria Sotomayor, and Humberto González-Díaz (July 2018). “Perturbation-Theory and Machine Learning (PTML) Model for High-Throughput Screening of Parham Reactions: Experimental and Theoretical Studies”. In: *Journal of Chemical Information and Modeling* 58.7, pp. 1384–1396. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.8b00286. URL: https://doi.org/10.1021/acs.jcim.8b00286 (visited on 08/14/2018).

Skoraczyński, G., P. Dittwald, B. Miasojedow, S. Szymkuć, E. P. Gajewska, B. A. Grzybowski, and A. Gambin (June 2017). “Predicting the outcomes of organic reactions via machine learning: are current descriptors sufficient?” In: *Scientific Reports* 7. ISSN: 2045-2322. DOI: 10.1038/s41598-017-02303-0. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5472585/ (visited on 07/30/2018).

Tokui, Seiya, Kenta Oono, Shohei Hido, and Justin Clayton (2015). “Chainer: a Next-Generation Open Source Framework for Deep Learning”. en. In: URL: https://chainer-chemistry.readthedocs.io/en/latest/index.html.

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio (Feb. 2018). “Graph Attention Networks”. en. In: *arXiv:1710.10903 [cs, stat]*. URL: http://arxiv.org/abs/1710.10903 (visited on 04/30/2020).

Weininger, David (Feb. 1988). “SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules”. en. In: *Journal of Chemical Information and Modeling* 28.1, pp. 31–36. ISSN: 1549-9596. DOI: 10.1021/ci00057a005. URL: https://pubs.acs.org/doi/abs/10.1021/ci00057a005 (visited on 05/13/2020).

Withnall, M., E. Lindelöf, O. Engkvist, and H. Chen (Dec. 2020). “Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction”. en. In: *Journal of Cheminformatics* 12.1. ISSN: 1758-2946. DOI: 10.1186/s13321-019-0407-y. URL: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-019-0407-y (visited on 05/08/2020).

Wu, Zhenqin, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande (Jan. 2018). “MoleculeNet: a benchmark for molecular machine learning”. en. In: *Chemical Science* 9.2, pp. 513–530. ISSN: 2041-6539. DOI: 10.1039/C7SC02664A. URL: https://pubs.rsc.org/en/content/articlelanding/2018/sc/c7sc02664a (visited on 10/06/2019).

Wu, Xi-Zhu and Zhi-Hua Zhou (Sept. 2017). “A Unified View of Multi-Label Performance Measures”. en. In: *arXiv:1609.00288 [cs]*. URL: http://arxiv.org/abs/1609.00288 (visited on 05/12/2020).

Yang, Kevin, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay (Aug. 2019). “Analyzing Learned Molecular Representations for Property Prediction”. In: *Journal of Chemical Information and Modeling* 59.8, pp. 3370–3388. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.9b00237. URL: https://doi.org/10.1021/acs.jcim.9b00237 (visited on 11/18/2019).

Zahrt, Andrew F., Jeremy J. Henle, Brennan T. Rose, Yang Wang, William T. Darrow, and Scott E. Denmark (Jan. 2019). “Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning”. en. In: *Science* 363.6424, eaau5631. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aau5631. URL: http://www.sciencemag.org/lookup/doi/10.1126/science.aau5631 (visited on 01/31/2019).

Zhang, Min-Ling and Zhi-Hua Zhou (Aug. 2014). “A Review on Multi-Label Learning Algorithms”. en. In: *IEEE Transactions on Knowledge and Data Engineering* 26.8, pp. 1819–1837. ISSN: 1041-4347. DOI: 10.1109/TKDE.2013.39. URL: http://ieeexplore.ieee.org/document/6471714/ (visited on 05/03/2020).

Zhou, Kuangqi, Yanfei Dong, Wee Sun Lee, Bryan Hooi, Huan Xu, and Jiashi Feng (June 2020). “Effective Training Strategies for Deep Graph Neural Networks”. en. In: *arXiv:2006.07107 [cs, stat]*. URL: http://arxiv.org/abs/2006.07107 (visited on 08/24/2020).

Chapter 7

CONCLUSIONS

In this dissertation, we presented the structural representation of data and its application across a range of domains including computer vision, computational neuroscience, and organic chemistry. With a primary focus on the visual input, we investigated the methods to detect the structure of the target and improved the performance in single and multi-person pose estimation problems. In addition, we proposed the structure discovery methods with weakly-supervised and unsupervised approaches and demonstrated its representation power by applying the raw discovered keypoints to the behavior classification task. Furthermore, we have shown promising results in the field of organic chemistry by relying solely on the structural encoding of the molecules to predict the experimental condition for coupling reactions. To summarize, this dissertation addressed the following questions:

1. **Pose Estimation** Given enough supervisory signals, how do we design a system that can efficiently estimate the semantic parts of the target instance from images?
2. **Structure Discovery** Without human knowledge, how does the model discover the semantically meaningful components from visual input in an automatic fashion?
3. **Application** Does the structural representation of data encode enough information to perform downstream tasks?

We summarize the contribution of each chapter for answering the questions above and also explain the contribution for advancing domain-specific problems.

In Chapter 2, we proposed a loss function to improve the performance for the single-person pose estimation and the image classification tasks. In human pose estimation, symmetric body parts often confuse the network by assigning indiscriminative scores to them. We define the prediction difficulty as a relative property coming from the confidence score gap between positive and negative labels and penalize the network to avoid the score of a false prediction being significant. We demonstrated the efficacy of the proposed loss by achieving comparable results to the methods that require more computational overhead. Also, the model trained with the proposed loss showed performance improvement on the LSP dataset over the baselines which require more model parameters. This chapter addressed question 1 by designing a good learning signal and demonstrated the generalization ability of the proposed method by further improving the performance on the image classification task.

In Chapter 3, we proposed a novel architecture for multi-person pose estimation, which is robust at predicting the pose of people having social interaction. For the Immediacy dataset, which is composed of the images with interacting people (*e.g.*, hugging and standing shoulder to shoulder), our model achieves 23.5% performance improvement over the baselines. This chapter sought to answer question 1 by proposing a network architecture to improve the challenging scenario of multi-person pose estimation in crowded scenes.

In Chapter 4, we proposed a keypoint discovery method for the images with large viewpoint and appearance variations. The proposed method not only achieves the state-of-the-art performance for the keypoint regression task but also tackles more challenging scenarios where the images exhibit diverse categories and viewpoints. This chapter addressed question 2 by exploiting the class label as a cue to infer the semantically meaningful parts from an image. The proposed method has also shown promising results for a simple downstream task such as a posture-based action recognition task.

In Chapter 5, we proposed an unsupervised keypoint discovery method with a specific focus on behavior analysis for computational neuroscience experiments. Since the videos for behavior analysis are often taken from the lab environment with stationary background, we leverage the spatiotemporal difference as a learning signal to extract the location of the organisms by focusing on the region of motion. By combining the network architecture proposed in Chapter 4, the proposed method successfully estimates the location of the target instance and consistently tracks the semantic parts without requiring any bounding box and keypoint annotations. In addition, the raw discovered geometric information has shown competitive performance to the supervised keypoints for behavior classification downstream task. This chapter addressed questions 2 and 3 by automatically discovering the structure and applying it to complex downstream tasks.

In Chapter 6, we proposed several methods to predict the substrate-specific cross-coupling reaction conditions in organic synthesis. Synthesizing a complex molecule often requires significant research efforts by iterative experimentation that can slow progress and waste materials. We aim to automate the optimization process by using machine learning models and address several different schemes for encoding the molecules. Among the string-based, chemical descriptor-based, and structure-based encodings, structural representation was the top-performing model by achieving the average error reduction rate of 31% from a baseline predictor. Also, domain experts can analyze and interpret the model prediction by looking at the activation scores for each component in the graph. This chapter addressed question 3 by investigating the structural representation of the molecules and conducting the downstream task of reaction condition prediction in the organic chemistry field.

There are still important research questions to be explored. This dissertation mainly investigated the 2D keypoint as a representation to encode the structure of visual input. Though this representation has demonstrated its efficiency in various applications, there are still remaining challenges. The results in Chapter 5 implied that the current models are not robust at predicting consistent keypoints for the occluded parts. As the real-world objects and scenes are in the 3D space, building a robust 3D structural representation will carry rich information about the target. Likewise, graph-based molecular encoding also lacks the 3D spatial information between the atoms. Since the chemical reactions occur in the bonds between the atoms in the 3D space, incorporating more complex spatial information will improve the prediction task. We believe this dissertation showed the promising direction for applying structural representation to diverse domains, and hope it works as a building step for new problems.