Optimizing Markerless Motion Tracking for the Study of Hand Kinematics

Nicholas A. Thomas

A thesis
Submitted in partial fulfillment of the
Requirements for the degree of

Master of Science

University of Washington
2021

Committee:
Amy Orsborn
Eli Shlizerman

Program Authorized to Offer Degree:
Bioengineering

University of Washington

Abstract

Optimizing Markerless Motion Tracking for the Study of Hand Kinematics

Nicholas A. Thomas

Chair of the Supervisory Committee:
Amy Orsborn
Department of Electrical Engineering, Department of Bioengineering

To facilitate studies investigating motor movements and learning mechanisms involved in complex kinematic tasks, there is need for effective real time 3D motion tracking. Current methodologies for motion tracking are insufficient for 24/7 data acquisition and can interfere with experimental conditions. Developments in deep learning have created tools capable of implementing pose estimation for neuroscience. While these tools are powerful and have promising experimental usability, they struggle to achieve stable, effective tracking on complex hand postures with significant occlusions or unusual angles. We have established these shortcomings in DeepLabCut, the deep learning pose estimation model we use in building our real time motion tracking system. To address the limitations in deep learning-based automatic pose recognition, we implement an iterative training paradigm which augments the training data of the neural networks based on performance of test data. Specifically, we aim to improve generalization and reduce training time – which are necessities for effective continuous 3D tracking - by optimizing the training data with the targeted addition of data. Using the iterative training framework, evaluated networks have their training data augmented based on the network's performance before being retrained and reevaluated. Our initial testing shows significant performance increases for the networks using this new training framework. Additionally, the networks had better tuning and generalization to complex behaviors. With the successful performance of the iterative training framework, we hope to further optimize the system by implementing the framework as a semi supervised approach.

Introduction

Overview

Brain computer interfaces (BCI) record neural signals from implanted electrode arrays and interpret the signals to control digital or physical objects. They are also useful tools for studying learning in the brain. Feedback mechanisms of the BCI can drive the brain's neuroplasticity and lead to the formation of specific cortical patterns as the user learns to use the interface [1]. Current research has gained insight into how this learning occurs in low dimensions primarily using BCI controlled computer cursor models. However, this model falls short in its ability to explore higher dimensional learning spaces. Native motor movements such as a reach and grasp, object manipulations, and even simple hand gestures occur with significantly more degrees of freedom and physically occupy three dimensions. The learning space for such movements is exponentially more complex than that of the BCI cursor models. High dimensional tasks may be more difficult to efficiently learn as people often learn "good enough" solutions rather than searching for the optimal one. The learning landscape for 2 dimensional actions is relatively constrained and has few optimal solutions. As task dimensionality and complexity increases, the learning landscape will become more complex. We currently do not understand how people learn in and explore these high dimensional spaces.

Insights into high dimensional learning will facilitate the development of BCI's with higher degrees of control and greater complexity. To study higher dimensional learning, the Orsborn lab aims to investigate hand kinematics using pose estimation during complex arm and hand movements in non-human primates. Traditionally, pose estimation is implemented using marker-based motion tracking systems or deep learning neural networks. The main benefits of markered systems are resilience to varying experimental conditions and real-time feedback; however, they can introduce noise into electrical recordings and can be prohibitively expensive. Increasing computational power and advancements in computer vision research has made deep learning based markerless systems more experimentally viable for pose estimation [22]. Our prior work focused on the development of a real time markerless motion tracking system capable of implementing 2D pose estimation across multiple cameras simultaneously. Our aim was to build a system which we could use as a basis to implement 3D tracking by triangulating between the multiple cameras. We were successful in the real time implementation of the system; however, the underlying network struggled with complex hand postures, particularly with occluded fingers and novel positionings. This uncertainty in these complex movements introduced significant stuttering in the 3D triangulations and did not produce stable 3D tracking. Manual optimization of the training data helped improve tracking, but it was extremely labor intensive and not practical for experimentation. We seek to optimize the 2D tracking performance and training methodologies of the underlying networks by implementing an iterative training framework.

Medical Motivation

Effective behavioral quantification is a necessity to implement high dimensional studies of motor movement and learning. This research would have a wide variety of applications, ranging from developing mobility oriented therapeutic diagnostic tools to improving BCI usability and efficacy for motor restoration. There is an estimated 1.9 million people in the US alone with an amputation [2] and 288,000 individuals with a spinal cord injury (SCI) of which 20.2% exhibit complete paraplegia and 11.5% exhibit complete tetraplegia [3]. Both individuals with amputations or those with SCI generally maintain their motor pathways in the brain, enabling the

restoration or supplementation of motor function. BCIs provide an alternate path to the damaged or missing nervous pathway and can help patients regain a sense of independence. For example, a tetraplegic woman was connected via a BCI to a 7 degree of freedom neuroprosthetic arm; she was able to begin movement on the second day of training and performed accurate routine movements by 13 weeks [4]. While such a case supports the feasibility of BCIs as a treatment option; they require intense resources and labor to operate effectively. For BCIs to become a more available and practical treatment, the patient must be able to reliably learn BCI skills for high dimensional control and form a cortical map for neuroprosthetic control that will be stable across time [5]. This requires comparable plasticity to that of native muscle learned movement [6]. Understanding the underlying mechanisms supporting this learning is critical to designing effective and robust BCIs.

Background

Quantifying behavior is an integral part of studying how learning occurs as neural recordings alone are insufficient for understanding the underlying mechanisms. The recordings illustrate the neural activity but do not necessarily explain causal relationships between the activity and behavior [7]. It is imperative to couple neural recordings with behavior to understand the process of learning and the neural population that governs that behavior [8]. An example of this is in a study by Nguyen et al. where they mapped the entire brain of *Caenorhabditis elegans* while simultaneously quantifying movement and behavior; they were then able to correlate the recorded neurons with the observed behaviors of the worm [9]. This is one of the first instances of this combination of complete brain mapping and behavioral recording that gave significant insights into interactions between motor mapping and control. It illustrates how powerful behavioral quantification in tandem with neural recording can be in investigating brain structure and relation. The importance of behavioral quantification in studying learning and motor control is also seen in research into the role of the sensory motor centers in flight behavior of Drosophila (fruit flies). Fyre et al. were able to hypothesize the control and dynamics of fruit fly flight using an integrative approach studying mechanosensory feedback, sensory motor interactions, quantified flight behavior, and aerodynamics [10]. Another study using both neural and behavioral data was able to isolate and determine which pairs of neurons were responsible for the fly's responses to increasing luminescence and which pair directed the response to decreasing luminescence [11]. Both models are simplistic but illustrate the importance of behavior quantification in understanding neural activity and brain function. However, they also highlight the limitations of current technologies in quantifying behavior of more complex organisms such as with hand movements in humans or monkeys. We need technologies to track significantly more complex behaviors in larger organisms.

Singular behaviors have extremely fine granularity in that they can be broken down into temporal and spatial components. For example, juggling would be a behavior, but the timing and positioning of the hands contain significant information describing the behavior. This quantification of temporal and spatial dimensions of behavior is necessary for effective motion tracking. Computer vision and deep learning convolution neural networks have shown promise in developing semi-automated markerless motion tracking systems with these characteristics. However, with increasing shifts towards and the inclusion of big data analysis in experiments, it is important to be able to quantify behavior in a scalable manner [12]. Human based observation or scoring of behavior has significant shortcomings in scalability and quality of the data; not only is it very slow, but it is subjective and, more importantly, low dimensional [13]. Computer systems have the advantage of automation, and better scalability. One such automated motion tracking

system fitted a 3-dimensional monkey skeleton to data taken using a series of Kinect sensors [14]. They created surface maps from the differing camera views and used a custom physics engine to fit a skeleton model within the surface. This methodology achieved consistent 3D tracking but was computationally expensive and slow. Thus, we looked to different frameworks to build upon to develop effective real time motion tracking and behavioral quantification.

We selected to use DeepLabCut (DLC) [15], an implementation of deep learning convolutional neural network used for pose estimation, as the basis for our system. DeepLabCut has been demonstrated to be able to achieve tracking accuracy comparable to that of a human, and it has the potential for real time processing and 3D tracking [15]. 3D tracking is the future goal of the system which necessitates stable tracking across multiple cameras simultaneously. In our initial work, we successfully developed a real time multi-camera implementation of DLC achieving framerates of over 30 Hz across three synthetic camera streams and latency ranging from 7-25 ms [21]. This performance exceeded our experimental goals for the system; however as stated before, our research highlighted that DLC struggles to generalize to novel postures and complex hand movements particularly when the fingers are occluded.

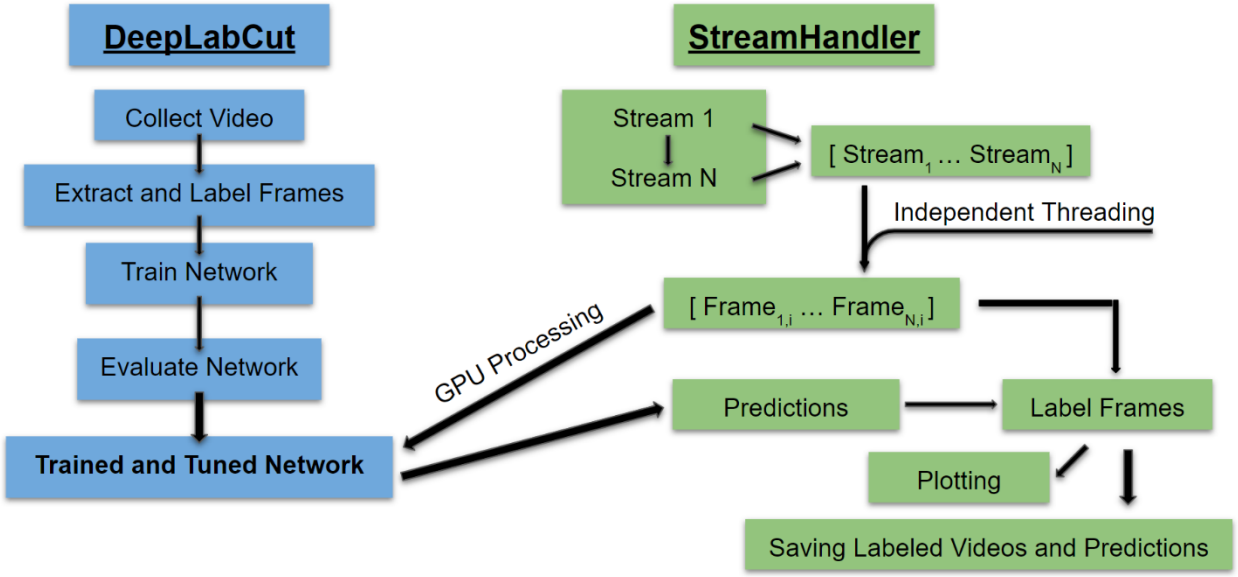

Figure 1: Workflow for the Stream Handler for real time processing. Once data is collected, labeled, and then used to train a network. The stream handler then handles retrieving the frames from batched streams and passing them to the trained network for labeling and plotting in order to minimize latency and maximize speed.

Prior Work: Real Time Tracking

Our initial work focused on the development of a real time implementation of motion tracking using DLC. We built a wrapper around DLC to take in multiple video streams, batch them, and extract frames as they are generated. The frames themselves are also batched to maintain congruity between frames taken at the same time. This batch of frames is then analyzed via DLC which returns predictions and confidences. If the confidences are above a threshold, then the labels are plotted, and the frames are displayed and or saved. We then benchmarked the system and analyzed three underlying networks for their performance and speed tradeoffs by spatially rescaling the original 640x480 frames. We calculated score and accuracy as:

Score = [1% frame diagonal x Accuracy / RMSE Distance]

Accuracy = [Number of Correct Labels / Total Labels]

As seen in figure 2A, the mobileNet has the best score vs framerate tradeoff. Each data point is 10% smaller rescaling starting at 100% original frame size down to 30%. Based on these results, we used a rescaling factor of 70% which maintained approximately 85% of the performance and increased the speed by approximately 40% for a single stream. Using these settings, the mobileNet once again significantly outperformed the other networks and achieved the target performance of greater than 30Hz processing for 3 streams simultaneously. Overall, we achieved the target specifications and created a successful real time implementation of DeepLabCut. However as previously stated, the DLC network struggled with generalization and complex hand gestures which is what we aim to address with the following experimentation. Lastly, we selected to use the mobileNet as the underlying network for our experimentation.

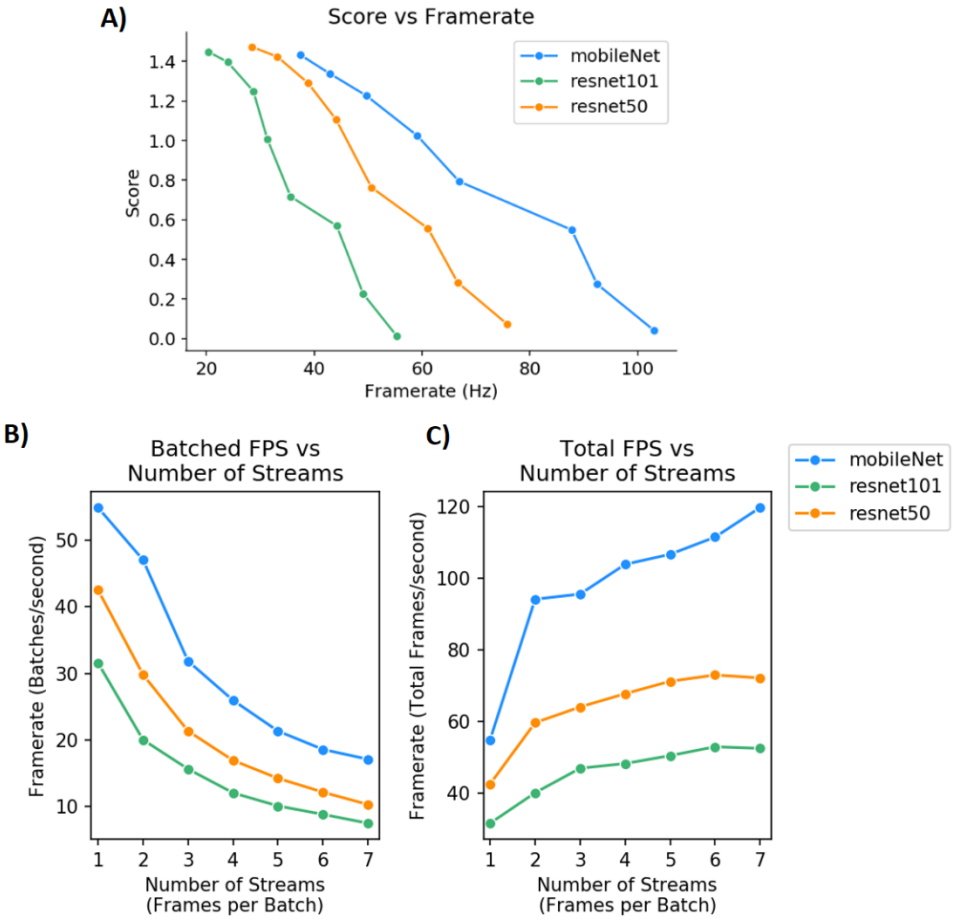

*Figure 2: Performance metrics for the Stream Handler. A) Score (Accuracy * 8 / Avg Distance) for each underlying network calculated on a test set of 200 Frames. B, C) Frame rate for each underlying network calculated on 4 minutes (7200 frames) of video. The video is synthetically replicated for each stream to simulate multiple cameras being processed simultaneously.*

Design Solution

To address the performance issues regarding generalization to novel postures and complex gestures with significant occlusion, we focused on developing a framework to optimize the training of DeepLabCut. Qualitative investigation into iteratively curating tailored datasets of increasingly complex movements showed that these datasets helped improve performance, stability, and reliability of tracking through complex gesturing. However, manually curating such a dataset is labor intensive, subjective, and not practical for implementing with animal models. Thus, we aim to automate the development of tailored datasets and quantify the potential performance gains of an iterative training framework.

Ethical and Regulatory Considerations

In considering the design of the system, we took into account the ethical impact and regulations that it would be subjected to. The system is not intended to be used in a clinical setting and solely in a research setting; thus, it will be subject to the host institutes individual regulatory processes. For the Orsborn lab, the intent is to use this toolbox to facilitate behavioral studies in non-human primates which does subject it to regulation by the Washington Nation Primate Research Center (WaNPRC). All experiments and data taken using this system will have to comply with any and all standards and requirements the center sets [19]. It is also important to note that the data generated would be publicly requestable under Washington State's open access laws. As the system is intended to be used in animal or human studies, it is important that the implementation of the system does not add risk or harm any of the subjects. Additionally, any experiment using animal subjects has its own body of regulations and ethical considerations such as whether animals need to be used, does the experiment cause undue harm, does it subject the animal to extreme duress or pain. For human studies, it is important to ensure that all participants privacy is protected, and informed consent is given for recording individuals. Consent was given for all data collected to be used in experimentation and displayed publicly

Materials and Methods

Hardware and OS

Two computers were used to conduct all training and testing. The first was a Razer Blade 2019 Advanced Model. The system had an Intel i7-8750H CPU locked @ 2.2Ghz, a Nvidia RTX 2070 Max-Q 8Gb Graphics Card, and 64 GB DDR4 Dual Channel SODIMM Memory. The second is a desktop had a Threadripper 1950x CPU, a Nvidia RTX 2080 8GB Graphics Card, and 32 GB DDR4 3200Hz Quad Channel Memory. All the video data for training and testing was taken using an external NEXIGO 1080p 60 FPS webcam set to 640x480 at 30 FPS. All tests were conducted using Ubuntu 16.04. All image analysis for labeling was done using DeepLabCut 2.08. All code was written using Python 3.6.

Data Format

For all experiments, we tracked a single individuals left hand and wrist with 24 points of interest as seen in Figure 3 to the right. To create the datasets for training DLC, 5 videos were taken comprising of different behaviors and hand motions. Each of these videos were manually annotated by two individuals (credit to Caroline Johnson and Toma Itagaki). It should be noted that different labelers naturally introduce additional variation. To minimize this, each labeler was trained on practice datasets to reduce discrepancies on their labeling. Four videos consisting of a total of 4356 frames were compiled into a dataset denoted SIMPLE. Each video had increasing complexity with composition outlined in table 1 below. The last video consisted of 1439 frames and is denoted as COMPLEX. It should be noted that the video in COMPLEX included positions and occlusions unique to the video and were not present in SIMPLE. Lastly, the frames were rescaled to a factor of .6 for faster processing. This rescaling was demonstrated to decrease accuracy between 10% and 20% but was determined to be appropriate for rapid experimentation [21].

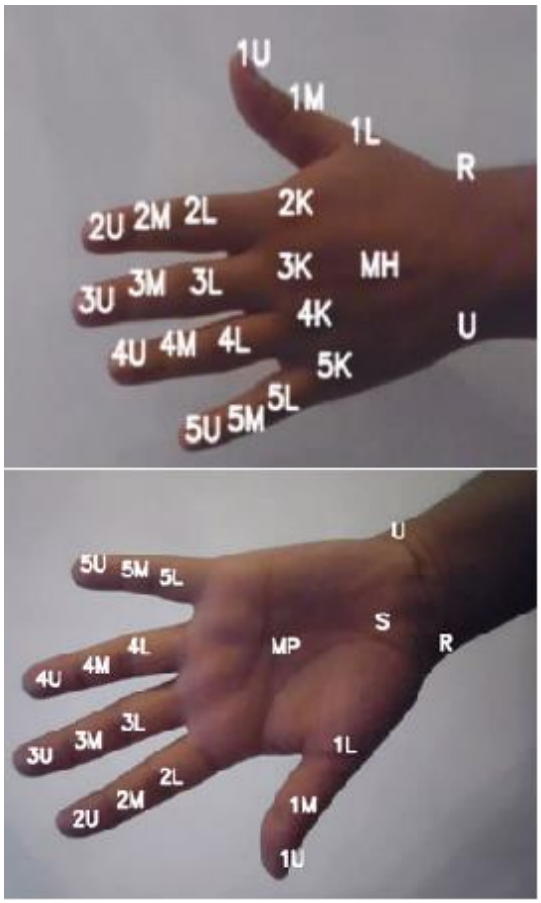

Figure 3: Labels for the 24 points of Interest. Radius (R), Ulna (U), Middle Hand (MH), Knuckles (2-5K), Lowest Finger Joints (1-5L), Middle Finger Joints (1-5M), Fingertips/Uppermost Finger Segment (1-5U), Middle Palm (MP), and Scaphoid (S).

TABLE 1: Summary of Video Composition with Ascending Complexity

Video	**Abb.**	**Frame Count**	**Dataset**	**Description**
Simple Movements	M0	977	SIMPLE	Hand spread with fingers fixed, no motion other than wrist rotation
Open Close	M1	743	SIMPLE	Front and back view of the hand under simple open close motions, no occlusions between fingers
Simple Finger Movements	M2	888	SIMPLE	Back view of the hand, simple finger articulations. No occlusions between fingers
Independent Finger Movements	M3	1748	SIMPLE	Front and back view of the hand with simple finger articulations. Minimal Finger occlusions
Natural Complex Movements	M4	1439	COMPLEX	Full Range of motions of the hand and finger, full occlusions. Rotations with complex finger movements.

Training Framework

To train each network, 50 frames were randomly selected from the training dataset to form a training set. These frames were then used to train a network for 250000 iterations. That network was then reevaluated on the training dataset and 50 frames were selected and added to the training set based on the chosen methodology: Ordered, Random, and Mixed. Ordered selected the worst performing frames in terms of scores, Random selected frames randomly from the dataset with each frame having equal chance of being selected (this serves as the control group), and Mixed was a split between the two methodologies with 25 frames first being selected by Ordered, and the from the remaining frames, 25 more were randomly selected. All methodologies selected without replacement and any frames selected that were already within the training dataset were skipped so that there were no repeated frames. Networks were trained for 14 iterations where an iteration is one addition of new frames into the training set. This gave a total of 15 networks for each type with a total of 750 frames in the final datasets. We trained two sets of networks on the SIMPLE dataset (S-Networks) and two sets of networks trained on the entire dataset (E-Networks). The first set of networks is denoted Set 1, the second as Set 2. This number of iterations and networks was determined due to computational limitations.

TABLE 2: Selection Methodologies

Selection Methodology	Description
Ordered	Selects 50 worst performing frames
Mixed	Selects 25 worst performing frames, 25 random frames
Random	Selects 50 frames with uniform randomness

TABLE 3: Summary of Error Types

Error Type	Criteria
True Positive (TP)	Label had to be correctly included and within 12 pixels of the ground truth label
True Negative (TN)	Label had to be correctly omitted
False Positive (FP)	Label was either correctly included and beyond the 12 pixel threshold or incorrectly included
False Negative (FN)	Label was incorrectly omitted

Evaluation Metrics

To measure performance, we classified each label as a specific error type determined by whether the label was correctly including or omitted within a specified frame. Distance was calculated only for true positive labels as that is the only error with both a ground truth label and a network prediction. It was normalized to the original frame dimensions of 640x480. As the aim of the project is to automate tuning the training data, we generated a singular score metric to evaluate frames. It should be noted, this score metric is different than that described in prior work. Score is calculated as:

$$\text{Score} = [\ \text{Count}(TP < \text{Max Error Distance}) + \text{Count}(TN)\ ]\ /\ 24$$

The Max Error Distance is a threshold that determines whether a label is reclassified as a FP if the label's distance from ground truth exceeds the selected value. Additionally, we set a threshold of minimum correct points for a frame to considered 'Well Labeled' which we selected to be 18. In benchmarking a network iteration for both distance and score, no frames that were used in training were included in the sets used for evaluation. It should be noted that score is a better indicator of tracking quality compared to distance.

Despite these evaluation metrics, it was difficult to assess whether the chosen values were representative of the actual performance of the labeling. Even with a ground truth label, distance does not necessarily state whether the point is correctly labeling a body part. This is especially true with the hand where fingers can be relatively close to one another. Thus, we varied the Max Error Distance and Minimum Correct Points to estimate which combination was most representative actual proportion of frames which were well labeled. As the thresholds became more restrictive, the number of frames that could be considered 'Well Labeled' approached 0. As Max Distance Error increased, misclassifications of fingers increased. Thus, to evaluate the role of Max Distance Error and misclassifications, we would manually iterate through the datasets with labels plotted as circles with a radius of the Max Distance Error centered on the ground truth predictions. Iterating through the datasets in this manner and analyzing for overlaps, where an overlap represents the potential of one label being incorrectly classified as another, we assessed a Max Distance Error of 12 to be appropriate for our dataset. It minimized the majority of overlaps between points, and thus misclassification errors, without being too restrictive so that points were incorrectly classified as FP.

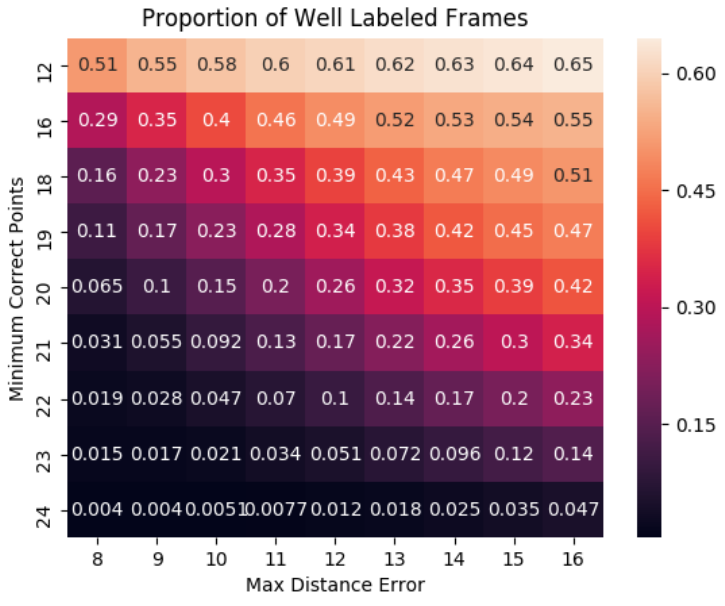

Figure 4: Proportion of Frames that Pass the Well Labeled Criteria for the given pairs of Minimum Correct Points and Max Distance Error. The true proportion was qualitatively estimated to be approximately 40% +- 5%

Several of the larger points – R, U, S, MP, MH, 1L – had higher frequencies of misclassifications as FP due to the higher variances in

the manually labeling of those points. We estimate an average of 2-3 of those points would be incorrectly classified as FP. This would automatically penalize score 2-3 points. Factoring this in, we aimed for a true target of 20-22 of points correct, which when adjusted gave us target scores of 18-20. In qualitatively assessing the true proportion of well labeled frames, we believe that 18 was the appropriate threshold of Minimum Correct Points to best estimate the proportion of Well Labeled frames. In summary, TP positive labels had to have a distance less than 12 pixels, and 18 points had to be correct for the frame to be Well Labeled.

SVM Prediction of Frame Quality

We first sought to predict whether a frame was well labeled. We selected to use an SVM with speed – pixels per frame - of the labels for the input as initial testing showed a correlation between high velocities and error. While testing the cutoff for a well labeled frame, we additionally evaluated an SVM for each combination of Minimum Correct Points and Max Distance Error. The SVMs were trained using 30% of the total dataset and tested on the remaining 70%. Each training and test set were randomly generated.

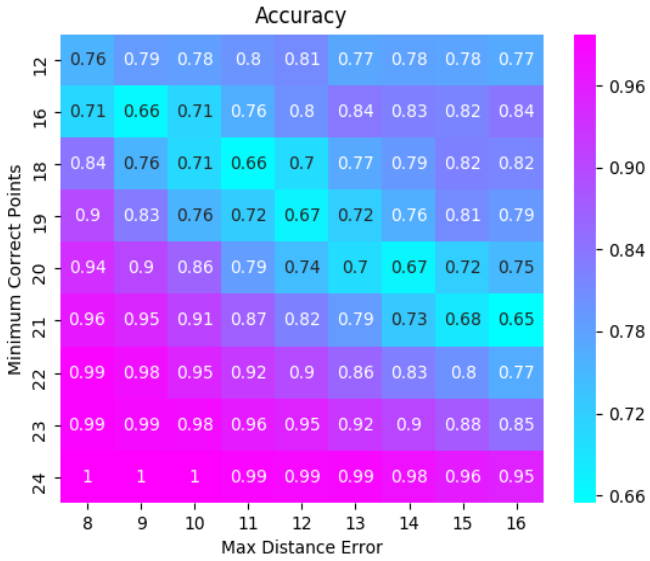

Figure 5: SVM accuracy for in labeling frames as Well Labeled or not for given pairs of Minimum Correct Points and Max Distance Error.

To use the SVM in frame selection within the iterative Labeled training framework, frames would be predicted as Well Labeled (Good) or Bad based on velocities. Frames would be then randomly selected from the Bad frames and added to the training dataset. Due to time constraints, the SVM was not fully tested and compared to the other networks and initial testing on its results within the Iterative Training Framework is omitted due to lack of proper controls.

Results

We evaluated the performance of the iterative training framework by evaluating the average frame score and distance for each network iteration. We calculated these metrics for the entire dataset, the SIMPLE dataset, and the COMPLEX dataset. Note, referring to Random, Mixed, or Ordered selection is referring to the networks trained using that selection methodology in the iterative training methodologies. Due to the small sample size for each network, statistical significance could not be meaningfully calculated.

Benchmarking the Performance of the S-Networks (Networks Trained on the SIMPLE Dataset)

We first analyzed the performance of the S-Networks (networks trained only on the SIMPLE dataset). When evaluated on the entire dataset, the S-Networks showed increases in

Evaluations of S-Networks

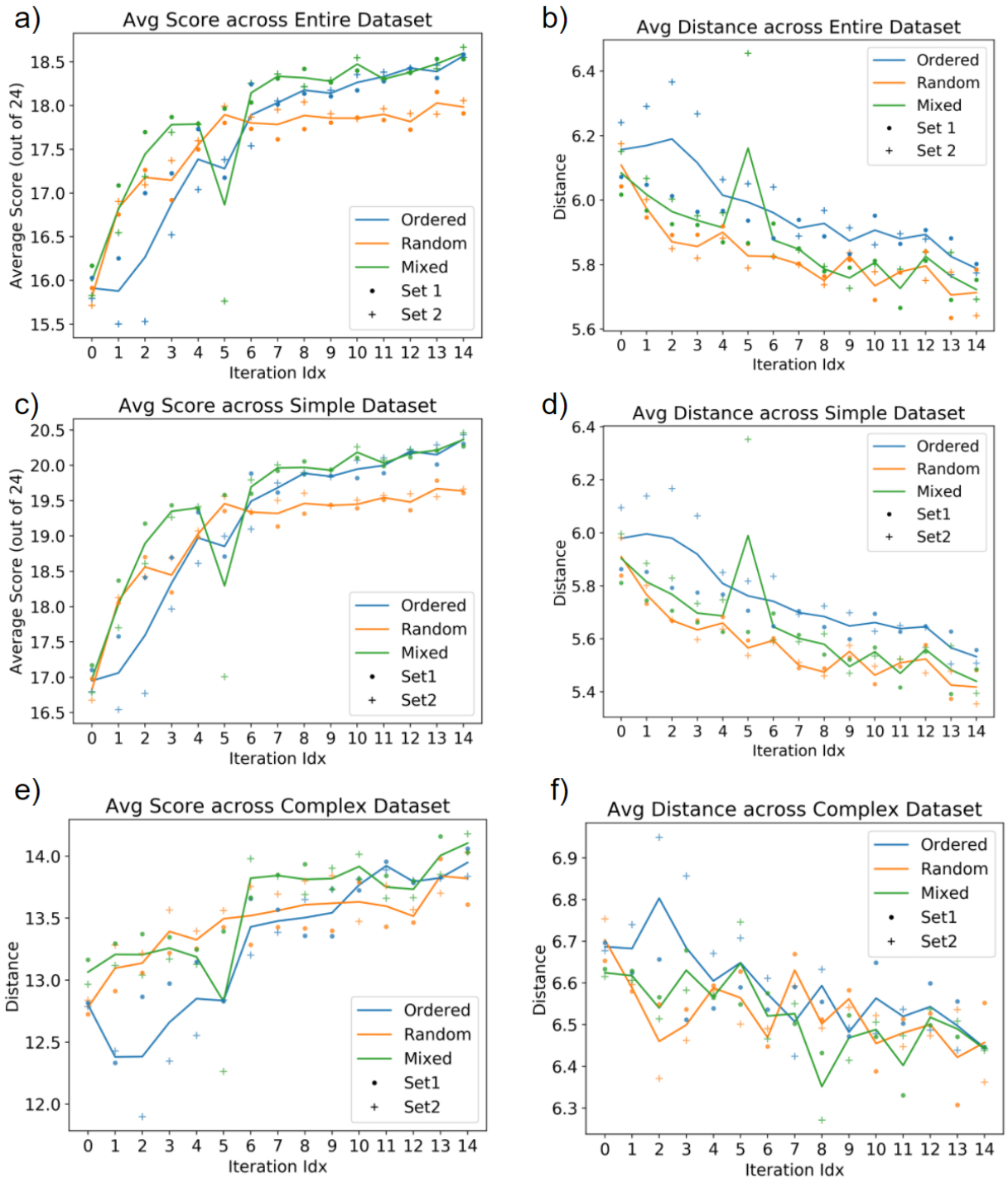

Figure 6: Performance of networks trained on the SIMPLE Dataset. Each iteration of each network was evaluated on the Entire Dataset, SIMPLE Dataset, and COMPLEX Dataset. No frames used in training were used in the Evaluation. The line for each figure is the average performance for the corresponding iteration from each set (Set 1 or 2)

performance for the Mixed and Ordered Selection compared to the Random selection (Fig 6a). The performance shows trending divergence for both sets at iteration 6. Relative to the Random network, the final iterations of the Ordered and Mixed selections had an average performance increase of .58 (3.24%) and .62 (3.42%) respectively (Fig 6a). For the S-Networks evaluation on the entire dataset, we saw that the distance was comparable across all selection methodologies with the Ordered selection tending to have a slightly higher distance (Fig 6b). However, the scale of the distances is less than the variance of the labels themselves and thus are not accurate representations of differences in performance for the training methodologies. This was also true for the evaluations on the SIMPLE dataset and the COMPLEX dataset (Fig 6d, 6f). When benchmarking the S-Networks on the SIMPLE dataset, we see that there is a large increase in performance for both the Ordered and Mixed selections. They trended towards the same score of 20.4 with an increase of .73 (~3.7%) points for both over the Random selection (Fig 6c). Lastly, we looked at the performance of the S-networks on the COMPLEX dataset to assess out of domain labeling performance. We saw no distinguishable trends in the performance on the COMPLEX dataset. All the selection methodologies performed similarly to one another with no significant deviations in their trends or final values. Overall, the S-Networks performance on the COMPLEX dataset is much lower (~ 4-5 points for score) compared to the performance on the SIMPLE datasets.

Overall, we do see significant performance gains with the Mixed and Ordered selection. However, the Ordered selection tends to have initially worse performance compared to the Mixed and Random selections before converging to similar performance. The Mixed selection tended to track more closely with the Random selection before overtaking performance. Lastly, the differences in performances between Set1 and Set2 decreases with increasing iterations with both sets converging to similar final performance values.

Analyzing Training Data Composition of S-Networks

After benchmarking the performance of the S-networks, we sought to analyze the training data to see how the iterative training framework affected training set composition (what proportion

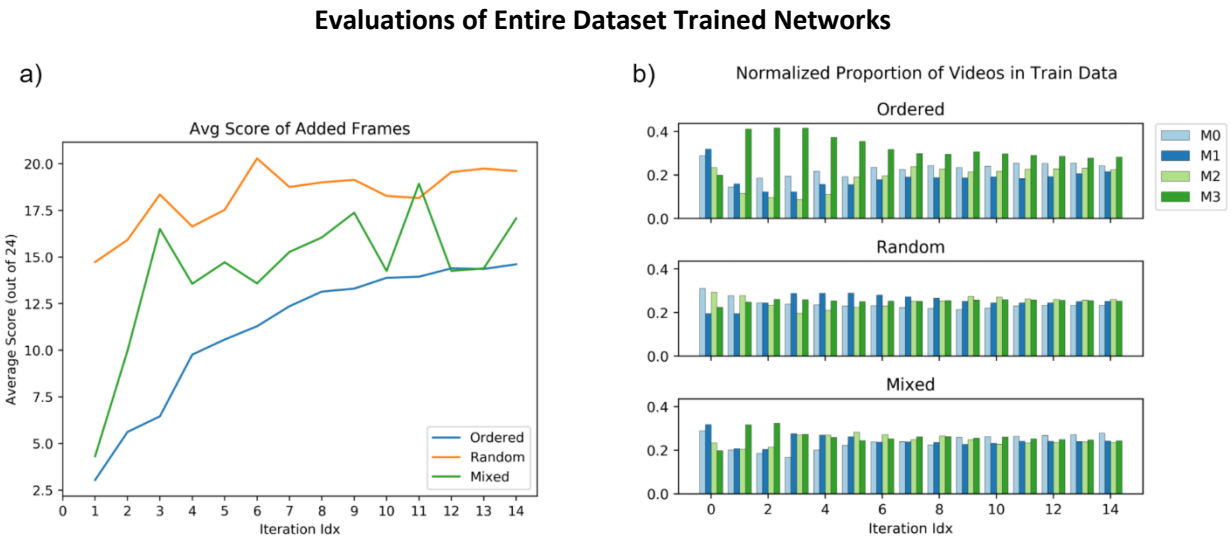

Figure 7: Training set composition data for Set1 S-Networks A) Average scores for the 50 frames added for each training iteration B) Proportion of video representation in the training data for each iteration for each network normalized to their representation in the SIMPLE Dataset

of the training data came from which video) and if there were trends relating to performance. To do so, we looked at the analyzed the training data for the Set1. First, we looked at the training data composition of the frames selected and added each iteration in the training framework which showed each network performing as expected. The Ordered network had the lowest average score composition of frames added, the Random selection had the highest, and the Mixed selection trended between the two (Fig 7a). For the Ordered selection, the train data biased the most complex video (M3) in the SIMPLE dataset heavily before incorporating more of the simpler videos (M0-2) (Fig 7b). Notably, the frames selected in the simpler videos tended to be frames that were harder to label – the hand in the middle of a rotation, side profiles, more occlusions. The Mixed selection networks saw a similar trend at a lesser scale with more representation of the M0-2 videos in the earlier iterations. The Random network had proportions that mimicked those of the makeup of the SIMPLE set. The lower representation of M0-M2 videos in the Ordered selection likely explains the lower performance of the Ordered network in early iterations as it was likely overfit in those early stages.

Benchmarking the Performance of the E-Networks (Networks Trained on the Entire Dataset)

After the analysis of the S-Networks, we sought to see how the iterative training framework performed with more complex videos. We trained networks on the entire dataset (E-Networks) and also benchmarked their performance on the entire dataset, SIMPLE dataset, and COMPLEX dataset. The E-networks showed increases in score for the Ordered selection when evaluated on the entire dataset with an increase in score of .5 points (2.63%) relative to the Random selection (Fig 8a). The Ordered selection and Mixed selection trended above the random selection past iteration 5, but the Mixed selection performance dipped in the final iterations converging to the same score as the Random selection (18.8). Once again, we saw the networks have similar performance regarding distance converging to similar values with the Ordered selection having slightly higher distances; the scale of the differences has minimal impact on tracking quality as it is less than the variance of the labels themselves. We then looked at the performance of the E-Networks on the SIMPLE dataset. All the selection methodologies performed comparably to one another for score with the Ordered selection having the highest performance (Fig 8c). Notably, we saw a differences in score between the performance of the S-Networks and E-Networks in their respective performance on the SIMPLE dataset (Fig 6c, 8c). For both the S-Networks and the E-Networks, we saw that the Random selection performed comparably both trending towards a score of 19.5. Both the Mixed and Ordered selections saw a decrease in average score going from the S-Networks to E-Networks. The Ordered selection score dropped .73 points (3.8%) and the Mixed selection score dropped 1.2 points (6.15%). Lastly, for the evaluations on the entire dataset, the Mixed selection final iteration for Set1 was the lowest overall final score; the Mixed selection also had the highest overall best performing network for iteration 10 Set2.

Lastly, we analyzed the performance of the E-Networks on the COMPLEX dataset to assess the potential performance gain on more complex hand gestures using the iterative training framework. The performance of the E-Networks on the COMPLEX dataset showed a large difference in performance between the selection methodologies fig (8e). The Ordered and Mixed selections trended above the Random selection after iteration 3. Both Ordered selections (Set1 and Set2) outperformed all other networks in the final iteration supporting a deviating trend in performance for the Ordered selection. The Ordered selection had an increase in score of 1.1 points (6.14%) over the Random selection and an increase of .76 points (4.3%) over the Mixed Selection. As stated, the Mixed selection trended better than the Random selection but only had an

Evaluations of Entire Dataset Trained Networks

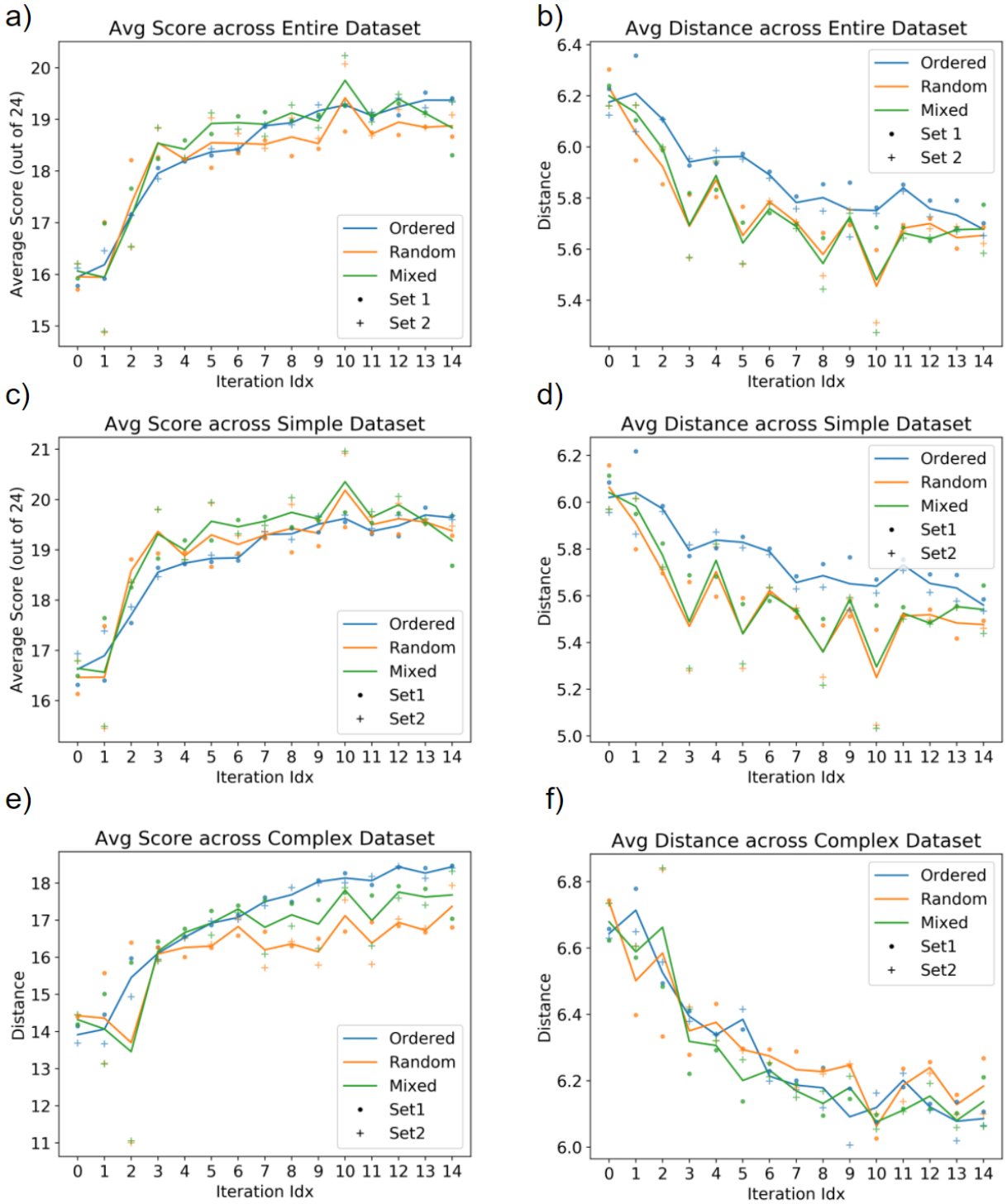

Figure 8: Performance of networks trained on the entire dataset (SIMPLE + COMPLEX). Each iteration of each network was evaluated on the Entire Dataset, SIMPLE Dataset, and COMPLEX Dataset. No frames used in training were used in the Evaluation. The line for each figure is the average performance for the corresponding iteration from each set (Set 1 and 2)

increase in score of .31 points (1.8%). Additionally, the Ordered selection had the lowest distance on the COMPLEX dataset – this was the only case where Ordered had the lowest distance. However, as stated, the scale of these differences in distance performance are not significant as an indicating tracking quality. Overall, the E-Networks performed several points better in score compared to the S-Networks for the COMPLEX dataset.

Analyzing the Training Data Composition of the E-Networks

Lastly, we investigated the trends in the training data of the Set1 E-networks to look for how the training framework was selecting videos and to see if any trends correlated with performance. We found that the trends in the training data of the E-Networks mimicked those of the S-Networks. The Random selection had the highest average scores of frames being added, the Ordered had the lowest, and the Mixed generally trended between the two (Fig 9a). For the breakdown of the proportions of videos in the training data, the selection again was biased toward M3 (the most complex video in the SIMPLE dataset) initially for the Ordered and Mixed selections (Fig 9b). However, the trends started returning to what was expected at iteration 2 with M4 becoming more and more represented in the dataset until it become the predominant video for both the Ordered and Mixed networks. There was also less representation of M0-M2 compared to both the Random selection and the Ordered and Mixed selection methodologies trained on the SIMPLE dataset. This supports that the iterative training framework biases complexity in its selection. For the Random network, the distributions follow that of relative proportions of the videos in the training set.

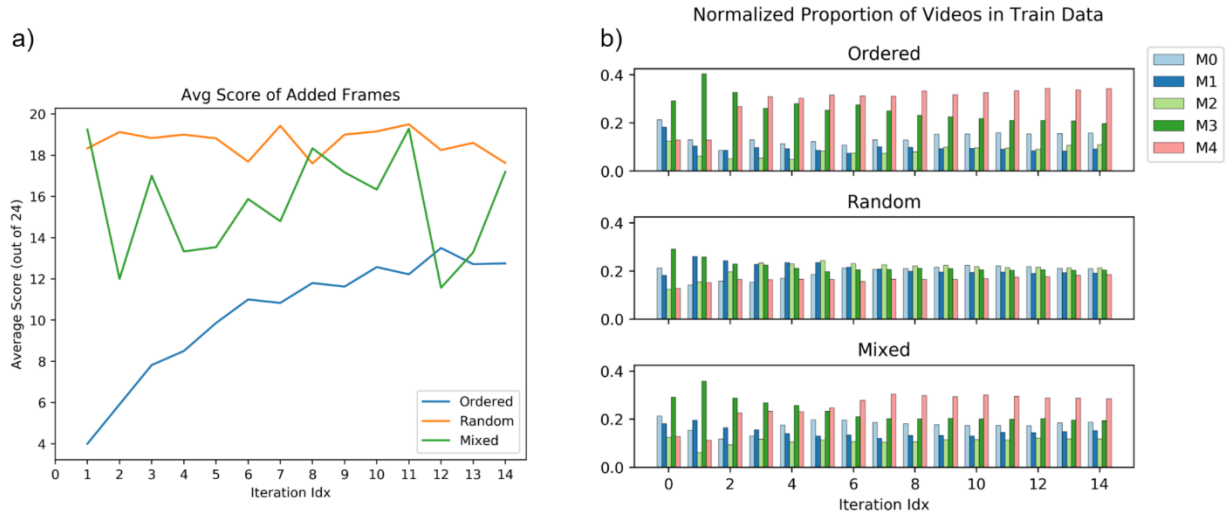

Figure 9: Training set composition data for Set1 E-Networks A) Average scores for the 50 frames added for each training iteration B) Proportion of video representation in the training data for each iteration for each network normalized to their representation in the training data (SIMPLE + COMPLEX Datasets)

Discussion

Performance and Tuning for Complexity

First looking at the Ordered selection for both the S and E Networks, we saw improved performance on both training datasets. However, it initially had worse performance trends

compared to the other networks. This is likely due to the selection of frames with outlier behaviors – i.e. part of, or the entire, hand being out of the frame. Frames like these tended to be added early in the selection process and the behaviors in them made up a very small portion of the training data. Additionally, the Ordered selection heavily biased the M3 video in its selection which could account for part of the lower initial performance. For the E-Networks Ordered selection, there was a large improvement of performance on the COMPLEX dataset (Fig 8e). The S-Network Ordered selection also performed better on the SIMPLE dataset (Fig 6c). In comparison, its E-Network counterpart performed relatively worse on that dataset (Fig 8c). This trend in the E-Network supports that the Ordered selection within the framework is optimizing for complexity. However, there was a decrease in relative performance on the SIMPLE dataset going from the S to E networks. However, the E-Networks Ordered selection still had performance comparable to the Random selection (Fig 8c). This suggest that the performance gain in the COMPLEX dataset did not directly penalize performance on the SIMPLE dataset, but rather changed the target of the optimization. This is further supported by the trends in the distributions of the training data used for the Ordered selection. (Fig 9b).

The Mixed selection had similar trends to the Ordered selection. It also saw improved performance and complexity optimizations. It did perform better initially and had better performance on the SIMPLE dataset evaluations for the S-Networks. This is correlated with the higher representations of videos M0-M2 compared to the Ordered selections (Fig 7b). The E-Network Mixed selection saw a decrease in performance on the last two iterations. This does not follow the generally upward trend of the performance and could be attributed to poor random frame selection. Further testing and network iterations are needed to better understand and evaluate the trends in performance. The E-Network Mixed selection also had less performance gain compared the E-Network Ordered selection when evaluated on the COMPLEX dataset (Fig 8e). This is likely due to the smaller relative proportion of M4 in the training datasets for the two networks. The trends support a correlation between representation in the training data (proportion of video) and performance of within domain behaviors for that representation. However, further research is needed to understand how much can be attributed to quality of frame (some frames might have more impact on performance) and the actual number of frames (frame quality is not relevant to performance). Lastly, as previously stated, more sets of networks need to be run to be able to calculate significance of results and better establish trends.

Source of Error

Due to time constraints, a limited number of models were run and tested. For each network type, only two sets of iterations were run and evaluated. Thus, there could be outlier performance due to chance. Additionally, each network was supposed to be initialized on the same dataset; however, for Set1, the Random selection trained on the SIMPLE dataset had different frames due to human error. This could be a source of potential error by limiting the effectiveness of the control.

Semi-Supervised Training Framework

While the iterative training framework did show significant performance gains, the current implementation is still very labor intensive. To reduce labor and training time, we aim to develop

metrics to assess frame quality without labeled ground truth data. The current implementation of the SVM is not sufficient for experimental needs. We will be looking at both optimizing the binary predictions of Well Labeled frames and implementing scalar predictions of score using a model such as a Support Vector Regression (SVR) or Ranked SVM for predicting relative score. Lastly, we will be assessing the iterative training framework with different metrics to understand the roles of the ranking metrics used regarding tuning, optimization, and generalization.

Alternative Tracking Methods, Data Augmentation, and Future Directions

With the success of the iterative training framework, it is important to understand the relevance of the work and overall tracking quality compared to alternate pose estimation frameworks, algorithms, and systems. The majority of pose estimation – particularly regarding hands - has been predominantly focused on human tracking. One of the key reasons we chose to use DeepLabCut, was its demonstrated ability to generalize to different animals and subjects including both primates and humans. This supports the use of humans as a proxy for potential performance on non-human primates. However, recent research has led to significant advancements in both general pose estimation and targeted hand tracking which necessitates reevaluating our system design and approach. One particular area of interest and comparison is modeled approaches. Three dimensional kinematic models for pose estimation have demonstrated to be effective tool for tracking the complex motions of the hand [23, 24]. Mueller et al. used a combination of synthetic data, GANs, and a 3D kinematic model to successfully increase generalization and improve robustness to occlusions for 3D tracking using a monocular 3D camera [23]. Using a modified segmented hand model composing of a palm detector and hand landmark model, Zhang et al. were able to implement a real time, on device 3D hand tracking model for mobile GPU's [24]. In both cases, the skeleton models helped improve overall tracking stability and targeted the issues of tracking hands through occlusions. Another important feature of skeleton based models is the ability to calculate distance for all points giving a singular metric that can more easily represent tracking performance.

An important direction of our research is to assess the validity of more state-of-the-art augmentations, models, and approaches in tracking non-human primate hands. This could include testing the performance of such systems "out of the box" in addition to modifying the networks and using transfer learning to tune for non-human primate tracking. Additionally, looking at alternatives for either data augmentation or synthetic generation could both help improve DeepLabCut performance or make alternative large data approaches more feasible. Systems such as PoseAug [25], an auto-augmentation framework which targets training data diversity, could help supplement deficiencies in training data and produce better tracking. Other approaches such as domain adaption for animal pose estimation show significant increases in performance and generalization using synthetic data [26]. Alternative systematic active learning frameworks which incorporate uncertainty and confidence targeted for pose estimation have been demonstrated to be effective ways to improve tracking performance [27]. Overall, a key component of future work will be in drawing from systems such as these to augment, substitute, and supplement parts of our tracking framework to achieve our end goal of robust, real time, 3D, markerless motion tracking.

Conclusion

Initial testing shows strong potential in the Iterative Training Framework as it showed both improvement of within domain performance in SIMPLE datasets and better generalization in complex movements in the analysis of the networks trained on the whole dataset. However, the iterative training framework did not show any advantages for out of domain generalization. Despite the Random selection performing better regarding distance, there are clear performance benefits with the Ordered and Mixed selection methodologies. We were successful in improving both the overall performance and generalization to complex movements, but further benchmarking and testing is to further optimize the training framework and to better understand both the limitations and tradeoffs

Acknowledgments

I would like to thank my PI, Prof Amy Orsborn, for her guidance and support of this project in addition to her excellent mentoring. I also want to thank the other members of the Orsborn lab for their input, experience, and advice throughout the project – particularly Caroline Johnson and Toma Itagaki for their assistance with the labeling of the datasets. Lastly, I would like to thank Prof Eli Shlizerman as a member of my advisory board.

Citations

References

1) A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, "Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control," *Neuron*, vol. 82, no. 6, pp. 1380–1393, Jun. 2014.
2) G. McGimpsey and T. C. Bradford, "Limb Prosthetics Services and Devices - NIST." [Online]. Available: https://www.nist.gov/document/239limbprostheticsservicesdevicespdf.
3) National Spinal Cord Injury Statistical Center, Facts and Figures at a Glance. Birmingham, AL: University of Alabama at Birmingham, 2018
4) Collinger, J. L., Wodlinger, B., Downey, J. E., Wang, W., Tyler-Kabara, E. C., Weber, D. J., McMorland, A.J., Velliste, M., Boninger, M., Schwartz, A.B., "7 degree-of-freedom neuroprosthetic control by an individual with tetraplegia," *Lancet*, vol. 381, no. 9866, pp. 557–564, Feb. 2013.
5) K. Ganguly and J. M. Carmena, "Emergence of a Stable Cortical Map for Neuroprosthetic Control," *PLoS Biol*, vol. 7, no. 7, Jul. 2009.
6) J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-Computer Interfaces in Medicine," *Mayo Clin Proc*, vol. 87, no. 3, pp. 268–279, Mar. 2012.
7) J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel, "Neuroscience Needs Behavior: Correcting a Reductionist Bias," *Neuron*, vol. 93, no. 3, pp. 480–490, Feb. 2017.
8) A. Gomez-Marin, J. J. Paton, A. R. Kampff, R. M. Costa, and Z. F. Mainen, "Big behavioral data: psychology, ethology and the foundations of neuroscience," *Nature Neuroscience*, vol. 17, no. 11, pp. 1455–1462, Nov. 2014.
9) J. P. Nguyen *et al.*, "Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans," *Proc Natl Acad Sci U S A*, vol. 113, no. 8, pp. E1074–E1081, Feb. 2016.
10) M. A. Frye and M. H. Dickinson, "Closing the loop between neurobiology and flight behavior in Drosophila," *Current Opinion in Neurobiology*, vol. 14, no. 6, pp. 729–736, Dec. 2004.
11) R. Behnia, D. A. Clark, A. G. Carter, T. R. Clandinin, and C. Desplan, "Processing properties of ON and OFF pathways for *Drosophila* motion detection," *Nature*, vol. 512, no. 7515, pp. 427–430, Aug. 2014.
12) A. J. Calhoun and M. Murthy, "Quantifying behavior to solve sensorimotor transformations: advances from worms and flies," *Current Opinion in Neurobiology*, vol. 46, pp. 90–98, Oct. 2017.
13) D. J. Anderson and P. Perona, "Toward a Science of Computational Ethology," *Neuron*, vol. 84, no. 1, pp. 18–31, Oct. 2014.
14) T. Nakamura *et al.*, "A Markerless 3D Computerized Motion Capture System Incorporating a Skeleton Model for Monkeys," *PLOS ONE*, vol. 11, no. 11, p. e0166154, Nov. 2016.
15) A. Mathis *et al.*, "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, no. 9, p. 1281, Sep. 2018.

16) B. Forys, D. Xiao, P. Gupta, J. D. Boyd, and T. H. Murphy, “Real-time markerless video tracking of body parts in mice using deep neural networks.,” *bioRxiv*, p. 482349, Nov. 2018.
17) A. Orsborn and J. M. Carmena, “Creating new functional circuits for action via brain-machine interfaces,” *Front. Comput. Neurosci.*, vol. 7, 2013.
18) A. L. Orsborn and B. Pesaran, “Parsing learning in networks using brain–machine interfaces,” *Current Opinion in Neurobiology*, vol. 46, pp. 76–83, Oct. 2017.
19) “Project Review Process,” Washington National Primate Research Center. [Online]. Available: https://www.wanprc.org/research-opportunities/project-review-process/
20) “Licensing a repository,” Licensing a repository - GitHub Help. [Online]. Available: https://help.github.com/en/articles/licensing-a-repository
21) N. Thomas, “Implementing Real Time Markerless Motion Tracking” *Bioengineering Capstone,* 2020.
22) Toshev, Alexander, and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653-1660. 2014.
23) Mueller, Franziska, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. "Ganerated hands for real-time 3d hand tracking from monocular rgb." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49-59. 2018.
24) Zhang, Fan, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. "Mediapipe hands: On-device real-time hand tracking." *arXiv preprint arXiv:2006.10214* (2020).
25) Gong, Kehong, Jianfeng Zhang, and Jiashi Feng. "PoseAug: A Differentiable Pose Augmentation Framework for 3D Human Pose Estimation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8575-8584. 2021.
26) Li, Chen, and Gim Hee Lee. "From Synthetic to Real: Unsupervised Domain Adaptation for Animal Pose Estimation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1482-1491. 2021.
27) Liu, Buyu, and Vittorio Ferrari. "Active learning for human pose estimation." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4363-4372. 2017.