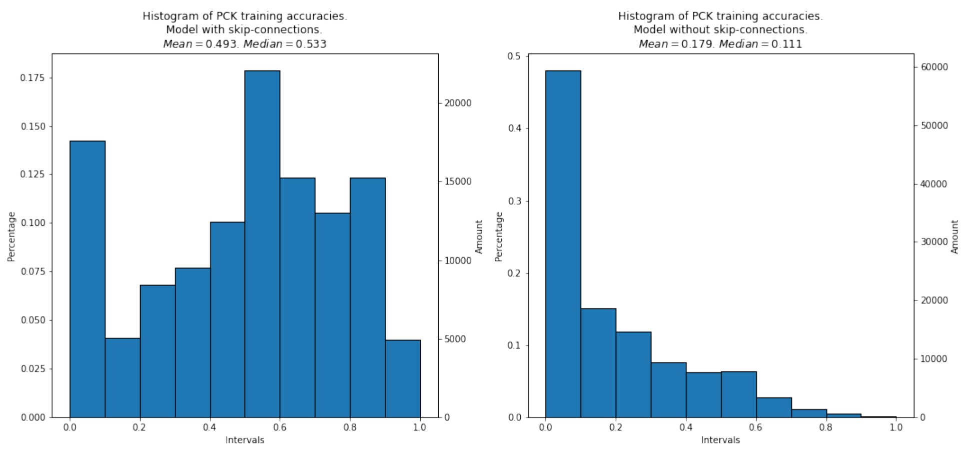


Figure 1: Histogram of PCK training accuracies of the model with the skip-connections enabled (left) and the model with the skip connections disabled (right).

1 Interpreting the Model

In the following section we will be interpreting the model developed in section **??**, with the intention of getting an understanding of what the model has learned during training, what the different parts of the model are used for, as well as checking for any redundancy in the model. In Section 1.1 an overview of the motivation behind interpreting the model is given. Section 1.2 then evaluates the effects of the skip-connections of the model. Then, Section 1.3 explores the latent space of the model with respect to getting an understanding of the principal components of the latent space. Section 1.4 also explores the latent space of the model, however, instead it uses clustering to seperate the latent space.

1.1 Motivation

Deep learning models are often complex and work like a *black boxes*. By that it is meant, that when a network is given some input, the model simply just returns some output without any explanation or reasoning behind the output. This can often be a problem, especially in cases where the output of the network can result in a life or death situation of a human. For that reason, understanding and explaining how a network works can be very important - this is what is called *explainable AI* (*XAI*)

Selvaraju *et al.* [4] argues that there are three cases for using explainable AI:

1. When the network performs worse than humans, an understanding of the network can help us improve the performance of the model.
2. When the network is on par with humans, an understanding of the network is trivial for humans to build trust in the network, as we can understand its strengths and weaknesses
3. When the network performs better than humans, an understanding of the network can teach humans how to perform better.

Throughout section 1 we will be exploring and understanding what the model, developed in section **??**, learned during training and what the different parts of the model are used for, leading towards improving the performance of the model easier.

1.2 Verifying the Effects of Skip-Connections

Olsen [3] and Newell [2] claims, that the skip-connections are used in order to recreate details that are lost during the encoder-phase. Throughout subsection 1.2 we will be verifying or refuting the claim of the effect of the skip-connections. To do so we will be using two models based on the same network:

1. The trained Stacked Hourglass from section **??**
2. The trained Stacked Hourglass from section **??**, but with the skip-connections disabled.

Thus, the second model has not been retrained and is identical to the first model, however, without its skip-connections.

In Figure 1 the distributions of the PCK training accuracies of the two models have been visualized. We have decided to make use of the training data for computing the PCK accuracies, as we want to look at the data, that the model has been trained on. By looking at the two distributions we can clearly see how to model without its skip-connections performs much worse, than the model with its skip-connections.

To further understand the decrease of accuracy in the case where the skip-connections are disabled, we have in Figure 2 visualized 20 samples from the training dataset, where the model with skip-connections has an 100% PCK accuracy score. Next to each image the ground truth heatmaps, or the prediction by the model with skip-connections, and the prediction by the model without skip-connections has been visualized.

By looking at Figure 2 we can see, that the model without its skip-connections often struggles with smaller joints, such as the eyes, ears or nose, whereas it performs better, however still not always perfect, on bigger joints, such as the shoulders, hips or knees. This is probably due to the fact, that the details of the smaller joints has a bigger chance of being lost by the max pooling layers in the encoder. Without the skip-connections their information is thus lost, resulting in bad predictions. Thus, we can verify Olsen's [3] and Newell's [2] claims, that the skip-connections are used for recreating details lost in the encoder.

1.3 Dimensionality Analysis of the Latent Space

In subsection **??** we described how we decided to use the Stacked Hourglass for the pose estimation, as it is similar to Autoencoders. This makes the model useful for encoding the data into a lower dimensional representation of the input data. The space of this lower dimensional representation is known as the *latent space* of the model. By exploring the latent space we can get an understanding of what the model has learned during training.

In the following subsection we will be exploring and explaining the most important components of the training data. By doing so we will get an understanding of how each component contributes to a prediction.

We decided to only use fully-annotated observations, as not fully-annotated observations would add some variance to the data, which would confuse the following procedure. We start off by feeding the fully-annotated training data through the encoder of the model and storing the output of the third residual module in the bottleneck. We decided to make use of training data for this, as we wish to look at what features of the training data that the model has learned. Each output of the bottleneck is a $4 \times 4 \times 256$ tensor, which we flattened to a 4.096 vector. Each vector was then stacked, forming a 7.017×4.096 matrix of the latent space.

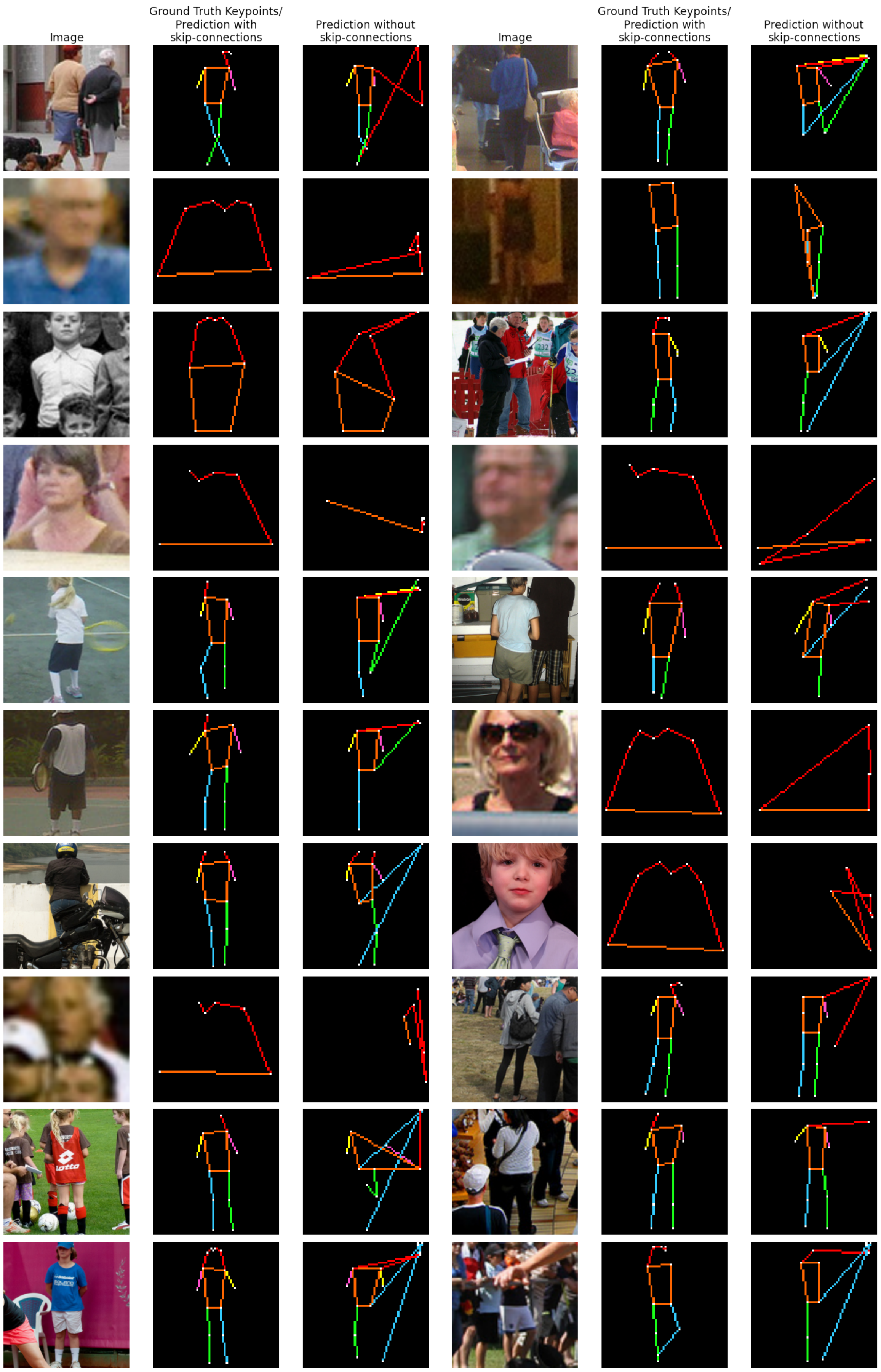


Image
Ground Truth Keypoints/ Prediction with skip-connections
Prediction without skip-connections
Image
Ground Truth Keypoints/ Prediction with skip-connections
Prediction without skip-connections

Figure 2: 20 samples of images correctly predicted by the model with skip-connections enabled, the corresponding ground truth heatmap and predictions by the model with skip-connections disabled.

We start off by finding the 4.096 principal components and the corresponding explained variance ratio of the data by using PCA. By doing so we get an understanding of which components for predictions are the most important. Next, the mean coordinate of the principal components, $\bar{\boldsymbol{x}}$, is found. The closest real data point is then stored. We then explore each principal component by "walking" from $\bar{\boldsymbol{x}}$ along the principal component in positive and negative direction, with various step sizes. After each "walk" the closest real data point is found and stored. By comparing the ground truth heatmaps of these real data points, we can see the transition of the heatmaps along the principal component, which will give us an idea of what effects the component has.

When doing the walk along a principal component, we made the step size be equal to $c \cdot \sqrt{\lambda}$, where c is a constant and λ is the explained variance of the principal component. By doing so we ensure, that we are not walking too far, ending up with misleading visualizations due to long distance between the end point to the nearest observation. Some of the results have been visualized in Figure 3.

By looking at Figure 3 we can see, that the first principal component is used for determining if the person is sitting or standing up. This is very clear, as the person sits down if we are "walking" in the negative direction from the mean coordinate, and on the otherhand the person straightens up if we are "walking" in the positive direction from the mean coordinate.

We can also see, that principal component 2, 3, 10 and 30 do not have an easy-to-see pattern, like it is the case with the first principal component. This could probably be because they explain a very little amount of the variance in the data, resulting in a very small step size, as well as patterns contributing very little to the prediction of the model.

Lastly, we can see, that principal component 50, which explains 0.165% of the variance in the data, and the following principal components do not have any variations in their corresponding results. This results in them acting as noise, and hints towards how the model could be using many fewer filters in the bottleneck.

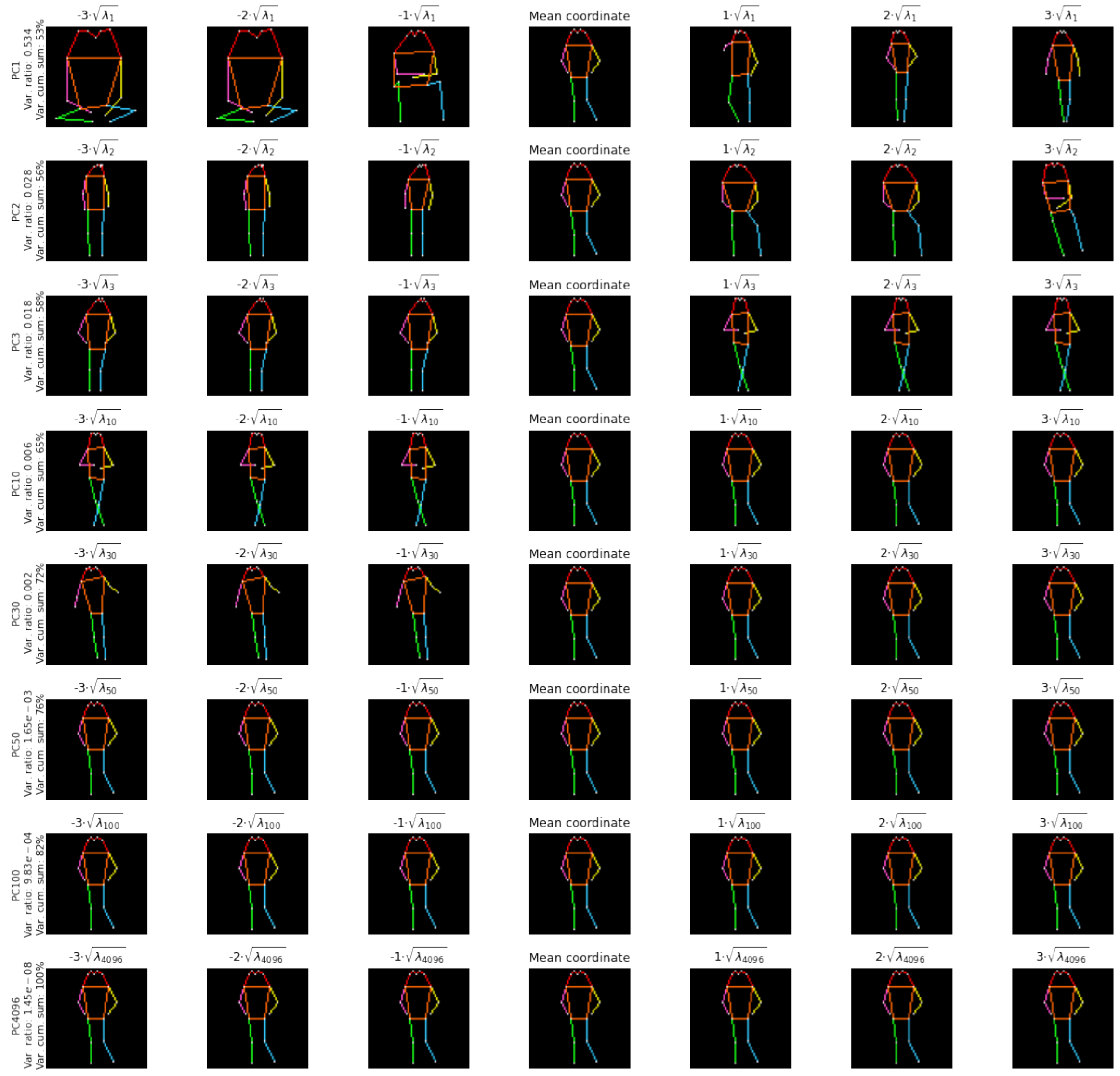
All in all, by doing the shape analysis of the latent space of the hourglass, we have learned how the model has learned the difference between people sitting down and standing up (and the poses in between), as well as possibly have identified some redundancy, in the form of the model using more filters in the bottleneck than necessary.

1.4 Using Clustering to Seperate the Latent Space

Similar to subsection 1.3, we will be exploring the latent space of the model to further get an understanding of what the model has learned during training, where we again will be using the training observations. Instead of exploring the principal components of the latent space, we will instead be grouping the training observations to get an understanding of how the model relate similar data to each other.

To create the latent space matrix we follow the same procedure as in 1.3, but instead also use not fully-annotated observations. Due to memory constraints only 10.000 random samples were used, resulting in an 10.000×4.096 matrix of the latent space

If we take this latent space matrix, project it down to 2 dimensions using PCA and visualize the samples with their corresponding ground truth heatmaps, we get the plot visualized in



-3·√λ1
-2·√λ1
-1·√λ1
Mean coordinate
1·√λ1
2·√λ1
3·√λ1
PC1 Var. ratio: 0.534 Var. cum. sum: 53%
PC2 Var. ratio: 0.028 Var. cum. sum: 56%
PC3 Var. ratio: 0.018 Var. cum. sum: 58%
PC10 Var. ratio: 0.006 Var. cum. sum: 65%
PC30 Var. ratio: 0.002 Var. cum. sum: 72%
PC50 Var. ratio: 1.65e−03 Var. cum. sum: 76%
PC100 Var. ratio: 9.83e−04 Var. cum. sum: 82%
PC4096 Var. ratio: 1.45e−08 Var. cum. sum: 100%

Figure 3: Nearest observations to the end point of "walking" along various principal components with varying step sizes.

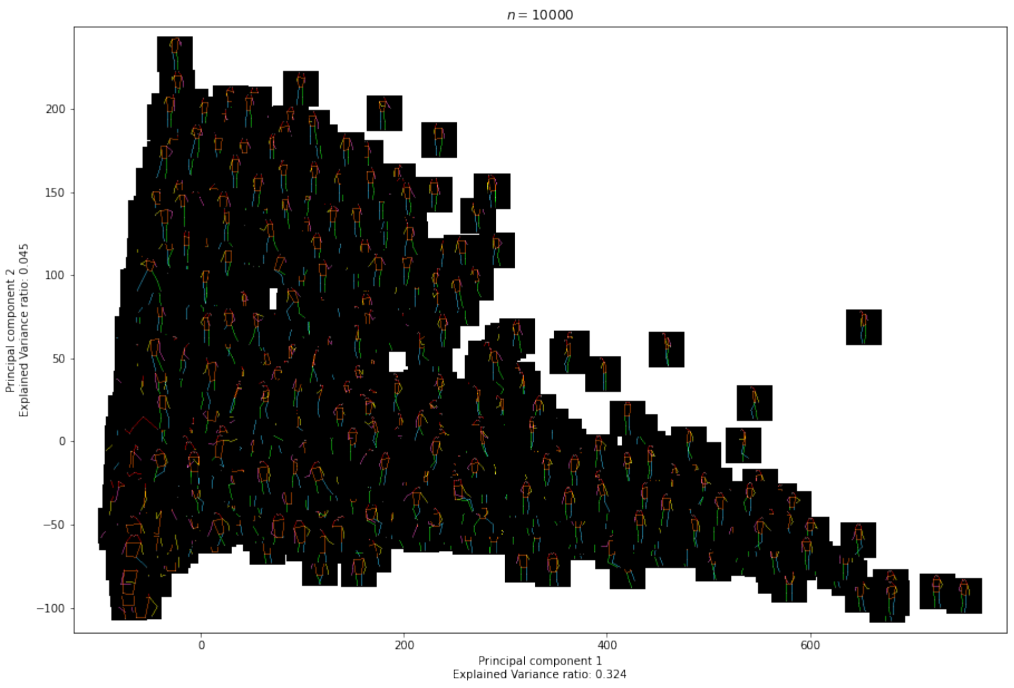


Figure 4: Plot of 10.000 samples of the latent space of the trained model, with the corresponding ground truth heatmaps

Figure 4. The plot only explains about 37% of the variance of the original data, however, we can clearly see how there is some specific structure in the data, as samples that are somewhat similar are close to each other, however, with a few outliers.

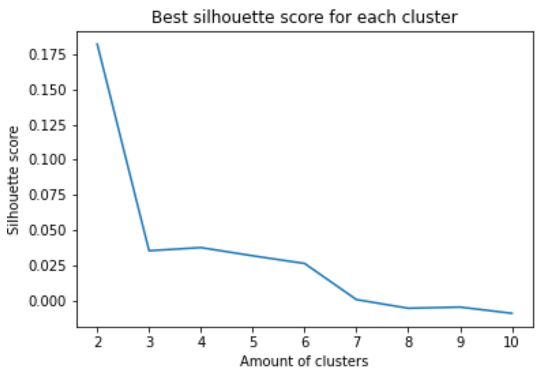
To see how the model seperates the data in the latent space, we will be using K-Means. Choosing the optimal K can often be difficult, as it is often not clear how many clusters there are in the data. For choosing the optimal k the *Silhouette score* is often computed, following the pseudocode visualized in Algorithm 1. For computing the Silhouette score, various values of K are used for training various K-Means models. After each model has been trained, let a_i be the average distance of the ith sample to the other samples in the same cluster as the ith sample. Then, let b_i be the average distance of the ith sample to the samples in the nearest cluster. Ideally, we want $a_i < b_i$, as $b_i < a_i$ means that the ith sample probably has been grouped to the wrong cluster. For that reason, the ith silhouette score is set to $1 - \frac{a_i}{b_i}$ if $a_i < b_i$ or $\frac{b_i}{a_i} - 1$ if $a_i > b_i$. By the end of the algorithm the mean silhouette score is returned. By computing the silhouette score for various values of K, the K with the silhouette score closest to 1 is chosen as the optimal K [1].

When running the K-Means algorithm on the latent space, we use $K = 2, 3, ..., 10$, where the algorithm is retrained 10 times with different initial centroid position for each K. For each run we record the Silhouette score, where the highest Silhouette score for each K has been visualized in Figure 5a. By looking at Figure 5a we can clearly see, how the optimal K for the model is when $K = 2$.

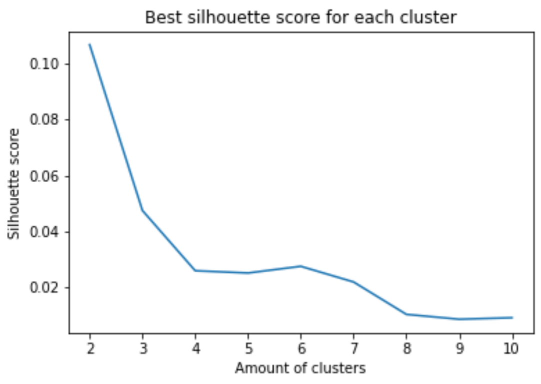
The results of running the K-Means model with $K = 2$ has been visualized in Figure 6. The K-Means model was run on the latent space in all of the 4.096 dimensions and only each clus-

Algorithm 1 Compute Silhouette Score [1]

```
Require: Clusters $C_0, C_1, ..., C_{k-1}$
1: for each cluster $C_i$ do
2:     for each sample $x \in C_i$ do
3:         Compute the mean euclidean distance from $x$ to the other samples in the same cluster: $a(x) = \frac{1}{|C_i|-1} \sum_{y \in C_i} D(x, y)$
4:         Compute the mean euclidean distance from $x$ to the nearest other cluster: $C_j$ $b(x) = \frac{1}{|C_j|} \sum_{z \in C_j} D(x, z)$
5:         Compute the Silhouette of $x$: $s(x) = \begin{cases} 1 - \frac{a(x)}{b(x)} & \text{if } a(x) < b(x) \\ 0 & \text{if } a(x) = b(x) \text{ or } |C_i| = 1 \\ \frac{b(x)}{a(x)} - 1 & \text{if } a(x) > b(x) \end{cases}$
6: return mean of the Silhouettes
```



(a) Latent space consists of 10.000 randomly chosen training samples



(b) Latent space consists of 7.017 fully-annotated training samples.

Figure 5: Silhouette score of running various K-Means models on different data from the latent space.

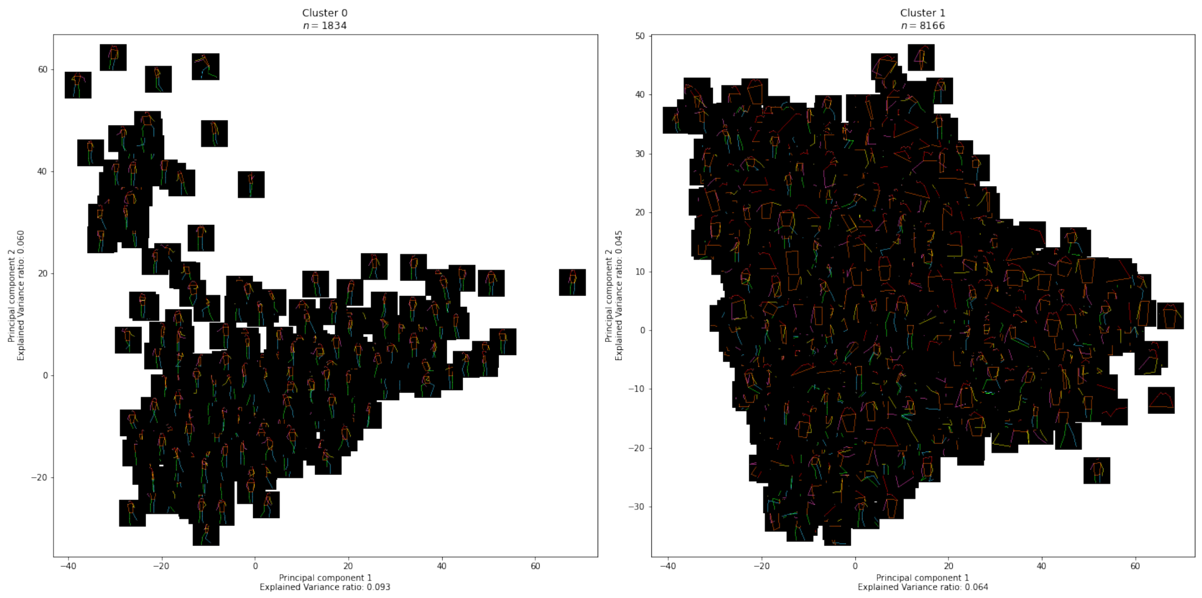


Figure 6: The resulting clusters of running a K-Means model with $K = 2$ on the latent space consisting of 10.000 random training samples

ter were projected down to 2 dimensions for the purpose of visualization. By looking at Figure 6 we can see how the two clusters has different content: where Cluster 0 focuses more on almost fully-annotated samples, Cluster 1 focuses more on samples that have a lot of keypoints missing. This is also easy to see if we look at the ground truth heatmaps of the samples closest to the centroids of the two clusters, as visualized in Figure 7. By doing so we can see, that the ground truth heatmap of the closest sample to the centroid of Cluster 0 almost has all of its joints annotated, whereas the ground truth heatmap of the closest sample to the centroid of Cluster 1 only consists of 2 keypoints.

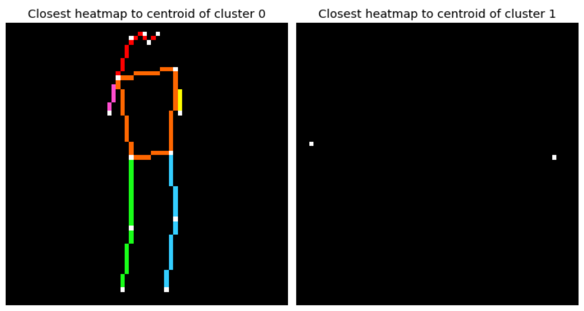

Closest heatmap to centroid of cluster 0
Closest heatmap to centroid of cluster 1

Figure 7: Closest points to the centroids of the two clusters from running K-Means on the latent space consisting of 10.000 random training samples

Although there are differences in the two clusters in Figure 6, there are still quite a lot of missclassified samples. To overcome this problem we remove all of the not fully-annotated samples and instead use all of the 7.017 fully-annotated samples of the training set, again fed through the network and outputted by the third residual module in the bottleneck. By doing so we get the Silhouette scores visualized in Figure 5b, where we again clearly see, that $K = 2$ is the optimal value of K.

The two clusters, resulted by only using fully-annotated samples, have been visualized in Figure 8 and the corresponding closest ground truth heatmaps for the samples closest to the centroids have been visualized in Figure 9. Like before, the K-Means model were ran on the data in full dimension to create the two clusters, which then were projected down to 2 dimensions using PCA for the purpose of visualization. By looking at the figure we clearly see how the content of Cluster 0 contains samples that are stationary, whereas the samples of Cluster 1 carry a lot more movement. This is also the case for the ground truth heatmaps of the samples closest to the centroids, visualized in Figure 9, as we can see, that the heatmap for Cluster 0 is more straighten, whereas the heatmapf for Cluster 1 is more bent and looks like it is in more movement. The two clusters does have a lot less missclassifications, than it was the case with the two clusters in Figure 6. The missclassifications could explain the not-optimal performance of the model as this could mean, that the network has not fully learned the differences between certain positions and where the positions should be placed in the latent space.

All in all we have learned, that the model, during training, has learned to seperate almost fully-annotated people and not fully-annotated people, as well as learned to seperate stationary people and moving people. However, the seperations are not perfect, as there are some missclassifications, hinting towards the inaccuracies of the model.

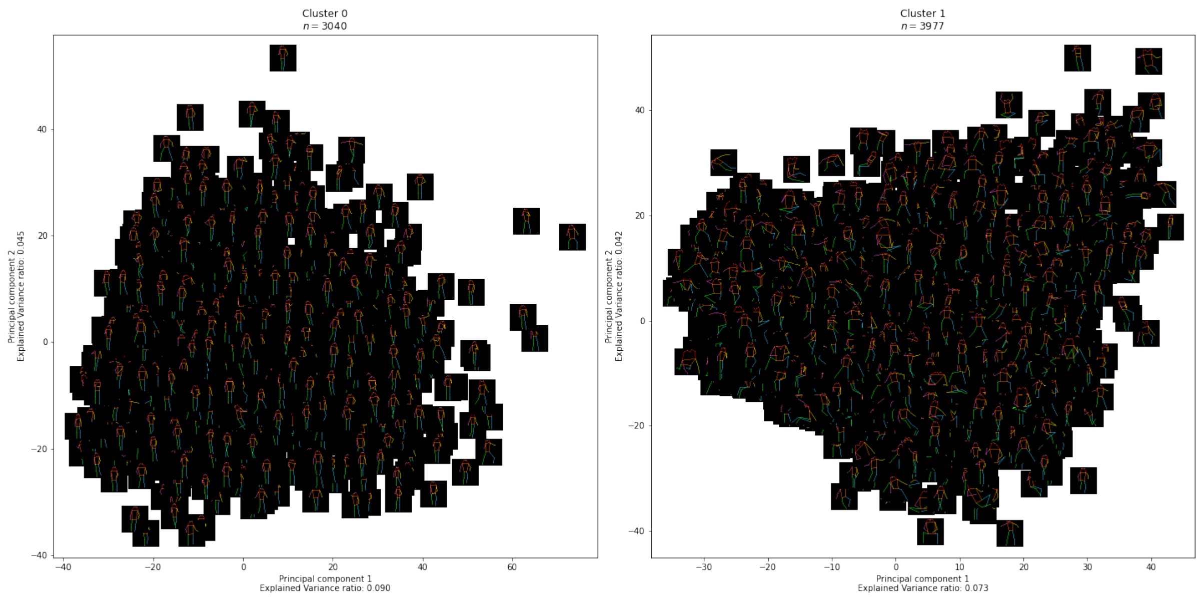
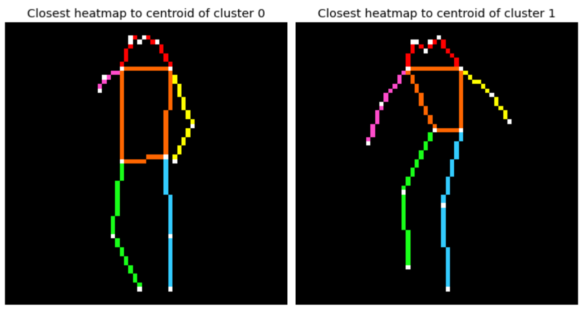


Figure 8: The resulting clusters of running a K-Means model with $K = 2$ on the latent space consisting of 7.017 fully-annotated training samples



Closest heatmap to centroid of cluster 0
Closest heatmap to centroid of cluster 1

Figure 9: Closest points to the centroids of the two clusters from running K-Means on the latent space consisting of 7.017 fully-annotated training samples