Improving Thermal Anomaly Detection Using Robot Pose Conditioning

MSc. Thesis

M.J.H. Owusu

Delft University of Technology

TU Delft

alliander

Improving Thermal Anomaly Detection Using Robot Pose Conditioning

by

M.J.H. Owusu

Student Name	Student Number
Max Owusu	4472721

Primary Supervisor:	Dr. Ir. J. Kober
External Supervisor:	MSc. Y. van Warmerdam
Project Duration:	March, 2024 - August, 2025
Faculty:	Faculty of Mechanical Engineering (ME), Delft

Cover:	Panther robot in a substation
Style:	TU Delft Report Style

TU Delft

alliander

Improving Thermal Anomaly Detection Using Robot Pose Conditioning

M.J.H. Owusu
Delft University of Technology
4472721

Abstract

Conditional variational auto-encoders showed improved anomaly detection abilities over standard variational auto-encoders in literature. This paper explores the effectiveness of robot pose conditioning on thermal anomaly detection in the context of electrical substations. We introduce a multi-modal conditional variational auto-encoder framework, capable of reconstructing thermal images and robot poses. It utilises a multi-objective loss function consisting of mean squared error image and pose reconstruction loss and Kullback-Leibler divergence. Orientation showed to be the most effective conditioning pose, in the context of anomaly detection. A well performing network effectively reconstructs the original assets based on the latent space representation, contains only slightly blurred reconstructions in cases of uncertainty, has a structured latent space as principal component analysis reveals and shows high separability between the distributions of the image reconstruction errors for normal and anomalous samples.

1. Introduction

The demand for energy has reached an all-time high, and will continue to do so for the foreseeable future. This can be concluded from the 2024 International Energy Agency (IEA) report [1], where the global energy demand between 2010-2023 was 1.4%, with an estimated growth between 2023-2035 of 0.5%. For this reason, the energy infrastructure in the Netherlands, but also globally, will be expanded in the coming years. Alliander, a Distribution System Operator (DSO) which operates in approximately a third of the Dutch medium and high-voltage grid, is required to operate and also maintain its energy infrastructure. There is a diminishing amount of technical personnel that can be found and let alone educated, as per internal Alliander documents. Due to the requirement for expansion of the electricity grid and this fact, an increasing problem can be foreseen with the indispensable inspection work that allows the infrastructure to function correctly.

In the context of energy distribution, substations play a vital role in electricity networks. They facilitate the conversion of electricity from high voltage to medium or low voltages, to ensure that many energy consumers, from smaller households to larger companies, are able to meet their individual energy needs. Current inspection work is done manually. This is done in a complex environment, and could result in possible downtime of the substation. Robotic inspection could mitigate this issue. It will also allow for more routine inspections, without the need for humans. This would allow for robot inspection, regardless of the time of day, when a substation is not functioning as it should. Also inspection would be less prone to human error, with a larger coverage. This would benefit not only Alliander, but society as a whole.

These foreseen problems create an interesting opportunity to use robotics for this not only vital, but dangerous inspection work. Applying robotics to this use case, has gained increased interest over the years. Even though there are definite benefits associated with robot inspection, it does not come without its challenges. There exists a significant research gap in streamlining this technology for wider adoption. An indispensable step in the robot inspection pipeline is Simultaneous Localisation And Mapping (SLAM), this allows the robot to map its environment, and navigate it accordingly. The robot pose information which results from this can enhance understanding of the layout of a substation. Similarly to how a human operator subconsciously would use pose information to determine whether an asset is anomalous from that specific view. This thesis will research whether information from SLAM, i.e. robot pose information, can be leveraged to aid in the anomaly detection process.

1.1 Contributions

The main contributions in this paper are the following:

- A thermal image and robot pose data acquisition pipeline, allowing for the systematic collection of segmented thermal images and pose pairs from a Gazebo simulation.
- A multi-modal Conditional Variational Auto-Encoder (CVAE) framework, which jointly

encodes thermal image and robot pose information in the latent space. Using Mean Squared Error (MSE) loss terms comparing the decoded and input representations of both modalities, to ensure a structured latent space which contains image and pose information.

- Analysing the most effective robot pose conditioning variables and network parameters for thermal anomaly detection by comparing misclassification results to baseline Variational Auto-Encoder (VAE) networks.
- Identifying well performing models by evaluating reconstructions, latent space structure and reconstruction MSE distribution separability.

2. Related Work

2.1 Visual Anomaly Detection

In order to obtain a better understanding of visual anomaly detection, a comprehensive survey was used [2]. Due to the unpredictability of anomalies, it is often impossible to create a data-set containing them, since there is no clear pattern which differentiates the anomalies from the normal instances. For this reason, unsupervised detection is often used [3] [4] [5], which also will be done in this paper. Anomaly detection can be further classified into two categories, image- and pixel-level methods. Image-level infers whether the entire image is anomalous, using the entire context that the image provides. Pixel-level detection does this using relative image location information. Since the global context is relevant for anomalies in substation assets, this will be used in this paper.

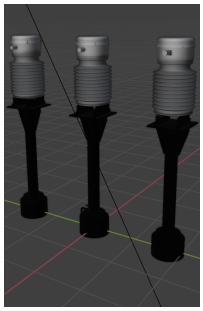
Another categorisation of the anomaly detection methods is pre- and post Deep-Learning (DL) approaches. Pre-DL methods use more traditional images features such as Scale-Invariant Feature Transform (SIFT) [6] and Histogram of Oriented Gradients (HOG) [7]. These features are then used in order to differentiate between the anomalies and normal instances. Post-DL methods use deep-convolutional neural networks due to their ability to solve computer vision tasks [8] [9] [10]. Since deep-convolutional neural networks are capable of analysis high-dimensional data, such as images, and representing its features [11], a similar architecture will be utilised in this paper.

2.1.1 Image-Level Anomaly Detection

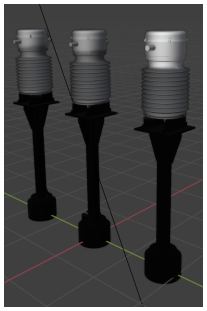
There are four ways of detecting image-level anomalies. The first method does so, by formulating a Probability Density Function (PDF). When a sample is found outside of this PDF, an anomaly has been found. Some example methods are: Gaussian Mixture Model (GMM) [12], non-parametric estimation like nearest neighbour or kernel density estimation [13]. The second approach is one-class classification. Using this method, a decision boundary is attempted to be found. Some example methods which achieve this are, one-class support vector machine [14] or support vector data description [15]. A benefit of using this approach compared to the aforementioned one, is that it can deal with a larger amount of samples, since no precise PDF is calculated. The next method is image reconstruction. In this approach, normal images are compressed in a lower dimensional latent space, based on which reconstructions can be formed. The working principle behind this, is that anomalous samples will have a higher reconstruction error than the normal samples, since they contain information which the network will not be able to reconstruct. Auto-encoder networks are used to achieve this [16]. The final method to perform image-level detection is self-supervised classification. Using this approach, supervision information is obtained using auxiliary tasks. Examples of this are finding object rotation or position [17]. This allows the model to learn visual features, which can then be used for anomaly detection. In this paper, reconstruction based anomaly detection will be used, due to its suitability for high-dimensional image data, interpretability and straightforward implementation.

2.2 Conditional Variational Auto-Encoder Anomaly Detection

The use of CVAE has been applied to the field of anomaly detection [18]. Here, a CVAE network was used in order to aid the detection of anomalies in the trigger system of the CERN particle accelerator. In this paper the loss function consists of a learned variance log-likelihood for each feature, in combination with Kullback-Leibler (KL) divergence, where the model was conditioned on the known L1 trigger rates. The mentioned paper highlights the effective use of known conditioning variables for anomaly detection, which will also be applied in this paper.



(a) Asset 5 - Baseline.



(b) Asset 5 - Anomaly.

(c) Asset 7 - Baseline.

(d) Asset 7 - Anomaly.

Figure 1: Baseline and anomaly textures for the thermal images of the two used assets.

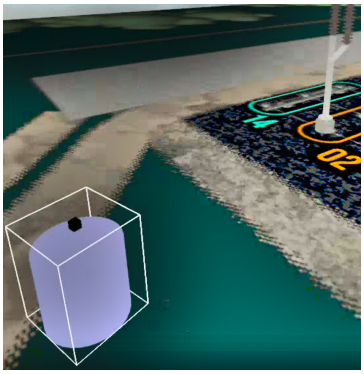


Figure 2: Cylindrical camera platform.

3. Methodology

The steps required in order to verify the effectiveness of robot pose conditioning for anomaly detection will be described in this section. They consist of data-acquisition, pre-processing, network architecture and logistic regression.

3.1 Data-Acquisition

First the thermal textures, which were needed to assess the effectiveness of the anomaly detection approach, had to be created. This was done using two models of substation assets, which were provided by Alliander. For each of the models, two textures were applied to the assets, which were then used to generate thermal images.

The primary images, which were used to train the networks, were generated using the baseline textures. Also, anomalous images were required. These images were used to verify whether anomalies can be detected. All these textures were hand-painted on the models using Blender. An overview of the applied textures can be seen in Figure 1.

These models where then imported in a Gazebo simulation, which was used to capture the thermal images and accompanying pose information. The robot used in this simulation was a simple cylindrical platform, which was 1 m high and had a radius of 0.5 m. It supported a thermal and depth camera at the same location on top of the cylinder, of identical specs. The cylindrical platform can be seen in Figure 2, and specifications of the cameras can be found in Table 1.

Capturing the various images from the simulation, was done in a systematic way which ensured reproducibility. The robot, based on some parameters which will be discussed, captured images at specific distances and angles. This approach, where the robot spawned at specific distances and angles relative to asset, ensured the captured dataset was identical for the anomaly texture compared to the baseline texture, for both of our used assets. A birds-eye-view view of this capturing strategy is given in Figure 3.

Specification	**Thermal/Depth Camera**
Horizontal FOV	1.1 / 1.1
Image Width (px)	320 / 320
Image Height (px)	256 / 256
f_x x-axis focal length (px)	277 / 277
f_y y-axis focal length (px)	277 / 277
c_x x-axis principle point (px)	160 / 160
c_y y-axis focal length (px)	120 / 120
Image Format	L16 / L16
Clip Near (m)	0.1 / 0.1
Clip Far (m)	100 / 100
Update Rate (Hz)	30 / 30

Table 1: Specifications of thermal and depth cameras.

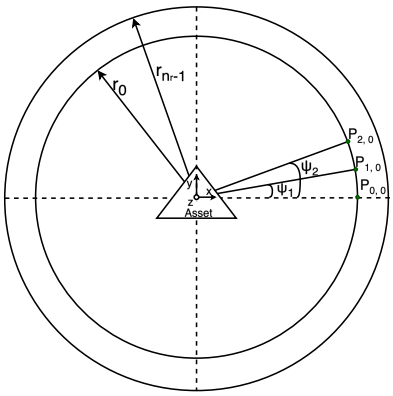


Figure 3: Birds-eye-view of image capturing strategy.

Parameter	Power Asset 05	Power Asset 07
a (start radius)	8.0	5.5
b (end radius)	10.0	8.0
n_r (Number of radii)	10	10
n_ψ (number of angles)	36	36
i	$i \in \{0, 1, \ldots, n_\psi - 1\}$	
j	$j \in \{0, 1, \ldots, n_r - 1\}$	

Table 2: Parameters used to compute the radii and angles for image generation.

The minimum and maximum radii, which were used to capture images for each of the assets, were r_0 and r_{n-1} respectively. All the individual radii and angles which were used to capture images, is described in Equation (1) and (2), using the accompanying variables described in Table 2.

$$\psi_i = \frac{i \cdot 2\pi}{n_\psi - 1} \tag{1}$$

$$r_j = a + j \cdot \frac{b - a}{n_r - 1} \tag{2}$$

These were used to describe the robot pose, which later was used to condition the CVAE networks. Here x and y were the position relative to the asset, and ψ was the z-axis orientation of the robot. The points in Figure 3, which describe the robot pose for each of the radii, are shown in Equation (3).

$$\mathbf{P}_{i,j} = \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ \psi_i \end{bmatrix} = \begin{bmatrix} r_j \cdot \cos(\psi_i)) \\ r_j \cdot \sin(\psi_i)) \\ \psi_i \end{bmatrix} \tag{3}$$

3.1.1 *Masking Thermal Images*

A mask was created based on the depth image, which was needed to isolate the regions of interest in the thermal images. The depth image contained some invalid data due to the limits of the virtual sensor, like ∞ values, which needed to be filtered from the image prior to the utilisation of the depth image as a mask. Removing these values effectively filtered out all the pixels which showed the sky-box. Subsequently, the remaining pixels were converted to 3D-coordinates using the pinhole model. As shown in Equation (4).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{D}(u, v) \cdot \begin{bmatrix} \frac{u - c_x}{f_x} \\ -\frac{v - c_y}{f_y} \\ 1 \end{bmatrix} \tag{4}$$

Equation (4) contains the following intrinsic parameters of the camera. The 2D pixel coordinates described by (u, v). The x and y focal length are given using the variables f_x and f_y. The camera's principle point in x- and y-direction are c_x and c_y respectively. The values used for these variables are given in Table 1. After the pixels were converted to real-world coordinates, the ground-plane was filtered using a vertical threshold. The last step needed to create the final mask, was isolating specific depth values. This ensured that the mask only contained the asset. During data collection, only one asset was present in the virtual world. This ensured the possibility of using a simple depth threshold to isolate the region of interest. A clustering algorithm such as DBSCAN would have been more appropriate, had the virtual word contained multiple assets.

The final step in obtaining the segmented thermal image data, was applying a bit-wise masking function to the thermal image. The masking approach effectively processed the depth-data, converted the pixels to real-world coordinates, then filtered the points based on spatial coordinates, in order to prepare our dataset for anomaly detection.

3.2 Pre-Processing

Some pre-processing was required before our dataset could be used for training. The first step when processing the already masked thermal images, was finding the smallest cropping rectangle, which preserved the thermal data. This valid thermal data could be identified, since these pixels contained non-zero values. After this rectangle was determined, there still were some empty pixels, for instance around the base of the asset, due to the masking approach utilised. For these empty pixels, zero's were inserted. At this stage in the pre-processing pipeline, not all images had an uniform size. This could be attributed to the fact that not all images were captured at the same distance from the asset, which resulted in an imbalance in the data. To mitigate this, zero-padding was used to ensure the aspect ratio of the images was maintained, while the images were converted to the target size of 128 x 64 px. After the target size was achieved, the images were normalized. Subsequently, the data structure was parsed into a tabular form, which contained the pre-processed thermal image in the first column, and the robot pose as given in Equation (3), in the last three column elements. This effectively resulted in 360 table rows, which contained different thermal image views for each of the textures given in Figure 1, captured using the parameters as described in Table 2. These four tables then resulted in the complete data-set.

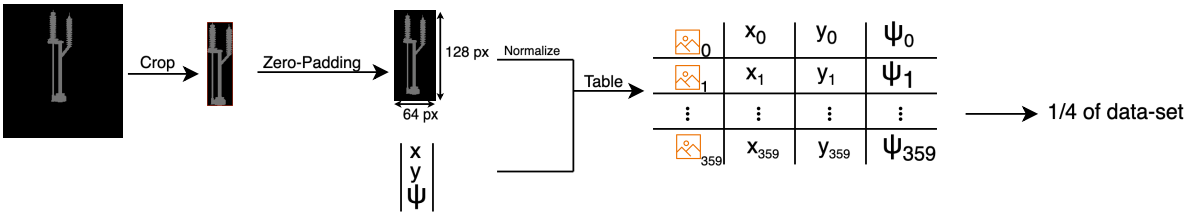


Figure 4: The full pre-processing pipeline.

An overview of the entire pre-processing pipeline is given in Figure 4.

3.3 Network Architecture

The main evaluation network followed the CVAE framework, as introduced in [19], but this was extended for multiple modalities i.e., pose and image reconstruction. CVAE networks have showed improved anomaly detection over their VAE counterpart [18], which was validated for thermal anomaly detection utilising robot pose conditioning in this paper. The implemented CVAE networks had a shared latent space based on the input variables, from which the network jointly reconstructed images and poses. Capacity influenced generalization, and thus over- or under-fitting of the model, as mentioned in [20]. For this reason, multiple size configurations were used in order to asses the effect of capacity, as determined by the latent space size and encoder/decoder width, on the reconstruction abilities and thus anomaly detection. The influence of the size parameter regarding the network structure is given in Table 3. An overview of the CVAE network structure is given in Figure 5b, in which the variable s is the size parameter, and p the amount of poses used per pose mode given in Table 4.

The effectiveness of robot pose conditioning on anomaly detection was compared to a baseline VAE model, which only used images to represent the latent space. This network can be seen in Figure 5a, which used the same parameters as the CVAE network presented in Figure 5b, but this only contained an image encoder and decoder. The assessment strategy regarding the pose modes will be explained in Section 3.3.7.

3.3.1 *Image Encoder*

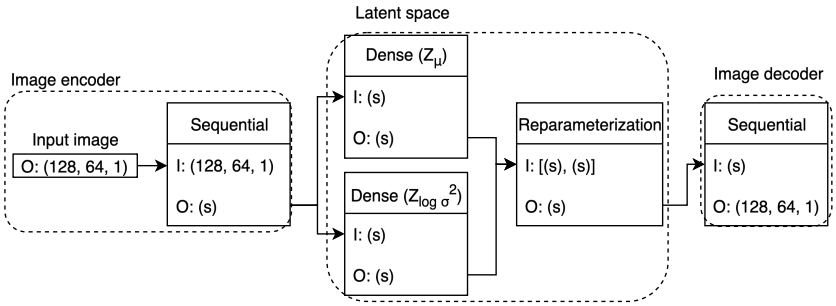
The image encoder architecture was designed in order to utilize the thermal images, which has a shape of $(128, 64, 1)$, this was always the size of the input layer. It utilised a Convolutional Neural Network (CNN) format, which based on the size parameter had more convolutional filters per layer, which can be seen in Table 3. This increased representational capacity as mentioned in [20]. Three $2D$ convolutional layers were utilised, which ensured sufficient representative capacity while avoiding over-fitting, due to the relatively small input image size. A kernel size of 3×3 and a stride of 2×2 was used, which is commonly used in auto-encoder networks due to its balance between computational efficiency and its contribution to representational capacity. This ensured that the network effectively captured features such as textures and edges. The layers utilised Rectified Linear Unit (ReLU) activation, which introduced sparsity and thus was computationally efficient. This also ensured that vanishing gradients would not occur during training, as mentioned in [21]. Finally the feature tensor was flattened and passed to a fully connected dense layer, which resulted in a vector with dimensions equal to the latent space.

3.3.2 *Pose Encoder*

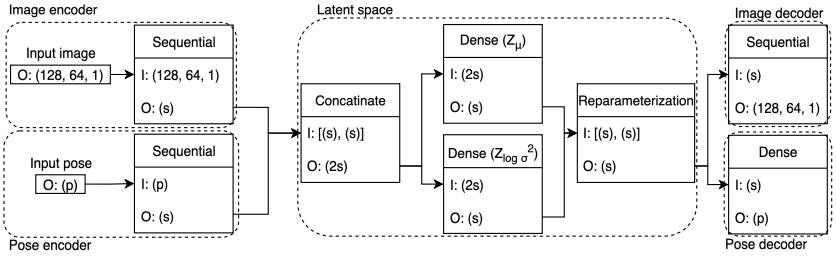
The pose encoder captured the pose component in the network, in order to represent it in the shared latent space. The input size of the pose encoder vector, scaled based on the pose mode used. The used pose modes will be further clarified in the Section 3.3.7. The number of neurons in the fully connected dense layers of the pose encoder, similarly to the image encoder, scaled with the size parameter, as can be seen in Table 3. Again, this increased representational capacity as mention in [20]. These layers utilised ReLU activation, similarly to the image encoder. Due to stability issues in the pose encoder for the largest network size, one additional layer was utilised, as can be seen in Table 3.

3.3.3 *Latent Space*

The goal of the latent space, was to capture information from the encoders(s), in a compressed format. In the case of the VAE networks, this was solely embedded by the image encoder, and in the case of the CVAE networks, this became a jointly embedded representation of the images and the poses. The latent space was then parametrized by two vectors, namely $\mathbf{z}_{\mu}$ and $\mathbf{z}_{\log \sigma^2}$, which were the mean and logarithmic variance respectively. In the case of the CVAE networks, both encoders were



(a) VAE network structure.



(b) CVAE network structure.

Figure 5: Comparison of VAE and CVAE network structures.

concatenated and fed into the dense layers to form the latent distribution parameters, while for the VAE networks the encoded image features were used directly.

In order to facilitate gradient-based optimization, direct sampling of the latent space was not possible. This was due to the fact that direct sampling is non-differentiable, which was a requirement needed in order to allow for back-propagation of the gradients. The reparametrization trick [22], as seen in Equation (5), allowed for sampling of the latent space, while still remaining differentiable to enable back-propagation of the gradients.

$$\mathbf{z} = \boldsymbol{\mu} + \exp\left(\tfrac{1}{2} \cdot \log \boldsymbol{\sigma}^2\right) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

3.3.4 *Image Decoder*

The image decoder was required in order to obtain reconstructed images, based on the latent representation. It mirrored the structure of the image encoder, but in reverse. Initially, the image decoder started with a dense layer. This dense layer mapped the latent vector to a flattened representation of size $16 \cdot 4 \cdot s$, with s being the size parameter as mentioned in Table 3. This vector was then reshaped to $(16, 4, s)$, which was then fed into a sequence of 2D transposed convolutional layers, where the number of filters per layer can be found in Table 3. A kernel size of 4×4 and stride of 2×2 was utilised for these layers. A slightly larger kernel size was used compared to the encoder, which ensured the latent representation was smoothly upscaled towards the target shape of the original input images. Similarly to the image encoder, ReLU activation was used. A final single 2D transposed convolutional filter layer was used, with a kernel size of 4×4 and stride of 1×2, to upscale towards the target shape of $(128, 64, 1)$. For this final layer sigmoid activation was used, which ensured the output values were in the range of $[0, 1]$.

3.3.5 *Pose Decoder*

The final part of the CVAE architecture, was the pose decoder. Its goal was to reconstruct the originally used pose, based on the jointly formed latent space structure, as deduced from the input images and poses. The use of the pose decoder, in combination with the loss function will be described in Section 3.3.6. It ensured that the information retained in the latent space was relevant to image reconstructions, but also captured pose based features. The pose decoder consisted of one to three fully connected layers, based on the pose mode used as described in Section 3.3.7. The input of this layer was a vector from the latent space of dimensions equal to size parameter s from Table 3, and the output was a vector of the same dimensions as the poses used in Table 4. The layer in the pose decoder used linear activation, since the poses were continuous values.

Size (s)	Latent Space	Image Encoder / Decoder	Pose Encoder
16	16	[4, 8, 16] / [16, 8, 4]	[16, 32]
32	32	[8, 16, 32] / [32, 16, 8]	[32, 64]
64	64	[16, 32, 64] / [64, 32, 16]	[64, 128]
128	128	[32, 64, 128] / [128, 64, 32]	[128, 256]
256	256	[64, 128, 256] / [256, 128, 64]	[128, 256, 512]

Table 3: Influence of the size parameter on the network components.

Network	Pose Mode	$\mathbf{p}_{mode}$
VAE	0: default	-
CVAE	1: orientation	$[\psi]^T$
CVAE	1: distance	$[\sqrt{x^2+y^2}]^T$
CVAE	2: position	$[x, y]^T$
CVAE	2: distance-orientation	$[\sqrt{x^2+y^2}, \psi]^T$
CVAE	3: default	$[x, y, \psi]^T$

Table 4: Overview of network configurations, pose modes, and $\mathbf{p}_{mode}$. The prefix p: indicates the number of poses used.

3.3.6 *Training Implementation and Loss*

The training implementation relied on Adaptive Moment Estimation (ADAM) [23], with a base learning rate of 0.0005, which was computationally efficient, and required no tuning. Each network was trained over 200 epochs, which was empirically deduced since the training losses of the stable networks converged sufficiently. Smaller batch sizes resulted in regularization, as mentioned in [24]. Since this was desirable in an auto-encoder architecture, a batch size of 32 was used. A standard 80/20 train-test split was utilised, where the samples were shuffled. In order to train the networks, a multi-objective loss function needed to be devised, which can be seen in Equation (6). These three objectives consist of: image reconstruction, latent space structure, and pose reconstruction.

The first loss term was $\mathcal{L}_{\text{image}}$, and is defined in Equation (7), as the MSE between the flattened input image $\mathbf{x} = [x_1, x_2, ..., x_D]^T$ and flattened reconstructed image $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_D]^T$. Maximising the log-likelihood of our data given a Gaussian distribution, was equivalent to minimising the MSE of the reconstructions, as mentioned in [22]. For this reason MSE was used for our reconstruction terms. As standard in multi-objective loss functions, weights were introduced, where the image loss was being scaled by a factor $\alpha = 1.0$.

The second loss term $\mathcal{L}_{\text{pose}}$ was defined in Equation (8), as the MSE between the poses used for each pose mode as described in Table 4, and their reconstructions. These are described by $\mathbf{p}_{mode} = [p_{mode,1}, ..., p_{mode,p}]^T$ and $\hat{\mathbf{p}}_{mode} = [\hat{p}_{mode,1}, ..., \hat{p}_{mode,p}]^T$ respectively. The pose loss was being scaled by $\beta \in \{0.2, 0.4, 0.6, 0.8\}$.

The final loss term is described in Equation (9), as KL loss [25]. This term ensured that the latent space had a similar distribution to a multivariate Gaussian. This equation utilised the two vectors which parametrize the latent space, as mentioned in Section 3.3.3, namely $\mathbf{z}_\mu$ and $\mathbf{z}_{\log \sigma^2}$. The KL-loss was being scaled by $\lambda_{\text{KL}} \in \{0.0001, 0.001, 0.01, 0.1, 1.0\}$. It should be noted that the total loss as described in Equation (6) was utilised by the CVAE networks. The VAE networks used only the image- and KL-loss terms, and their weights. This will be further clarified in Section 3.3.7. When training the VAE networks, only non-anomalous thermal images were used. While for the CVAE networks, non-anomalous image and pose pairs were both used. The reasoning for the use of the VAE and CVAE networks, as of which poses were used for the CVAE networks, will be further clarified in Section 3.3.7. The hardware which was used for training consisted of an AMD Ryzen 7 5800H CPU and a NVIDIA RTX 3070 laptop GPU.

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{image}} + \beta \cdot \mathcal{L}_{\text{pose}} + \lambda_{\text{KL}} \cdot \mathcal{L}_{\text{KL}} \tag{6}$$

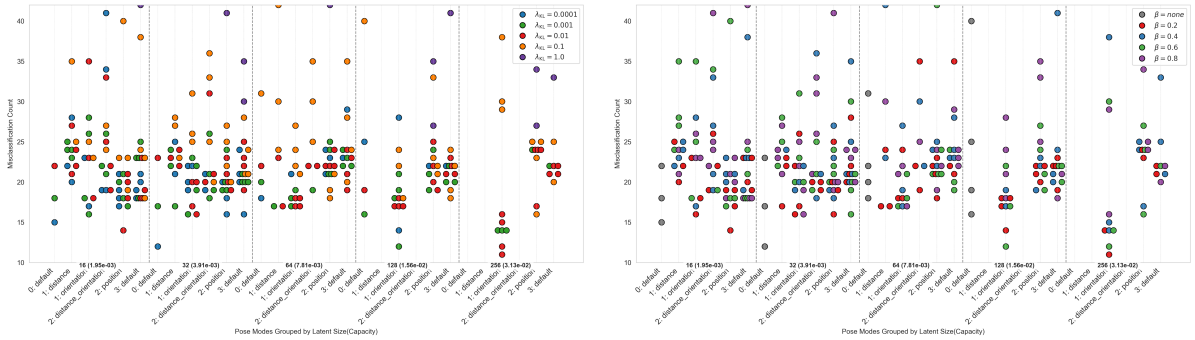
$$\mathcal{L}_{\text{image}} = \frac{1}{D} \sum_{k=1}^{D} (x_k - \hat{x}_k)^2 \tag{7}$$

$$\mathcal{L}_{\text{pose}} = \frac{1}{p} \sum_{l=1}^{p} (p_{mode,l} - \hat{p}_{mode,l})^2 \tag{8}$$

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{m=1}^{s} \left(1 + \mathbf{z}_{\log \sigma^2,m} - \mathbf{z}_{\mu,m}^2 - \exp(\mathbf{z}_{\log \sigma^2,m})\right) \tag{9}$$

3.3.7 *Pose Evaluation*

In order to evaluate the effectiveness of pose conditioning on the anomaly detection abilities of the networks, multiple pose configurations were eval-



(a) Influence of KL-weight on misclassification count grouped by latent size(capacity) and pose mode.

(b) Influence of β-weight on misclassification count grouped by latent size(capacity) and pose mode.

Figure 6: Network parameter influence on misclassification count grouped by latent size(capacity) and pose mode.

uated. This ranged from a VAE baseline network, which only used image information to structure the latent space, to the one to three pose CVAE networks, where different combinations of the robot poses as described in Equation (3), were used in order to form a structured latent space which captured both image and pose features. An overview of these different pose configurations is given in Table 4. In order to determine how to scale the pose- and KL-loss, different scaling parameter combinations were evaluated.

3.4 Logistic Regression

Binary classification was implemented, in order to evaluate how well the reconstruction errors could be utilised for anomaly detection. This was achieved using logistic regression, which resulted in an anomaly or normal label for the samples. First the anomaly- and normal-dataset MSE reconstructions and their ground-truth binary labels were collected, ensuring a balance between both classes. Subsequently, the data-set was split into a 80/20 train and test-split, after which the training data was again split into a 80/20 training and validation split. Subsequently, a five-fold stratified cross validation implementation was utilised in order to optimize the decision boundary, which preserved the balance between anomaly and normal data in each fold. For every fold, four folds were used to train the logistic regression model, where a 20% penalty was implemented for misclassifying anomalies compared to the misclassification of normal samples. The final fold was then used to validate the model. The decision boundary which maximised the F1-score of the validation fold was then stored. This process was then repeated five times. The five optimal decision boundaries were then averaged, and this was used on the final test-set. For this test-set, the misclassified data-points and their corresponding MSE were captured.

4. Results

4.1 Pose Mode Evaluation

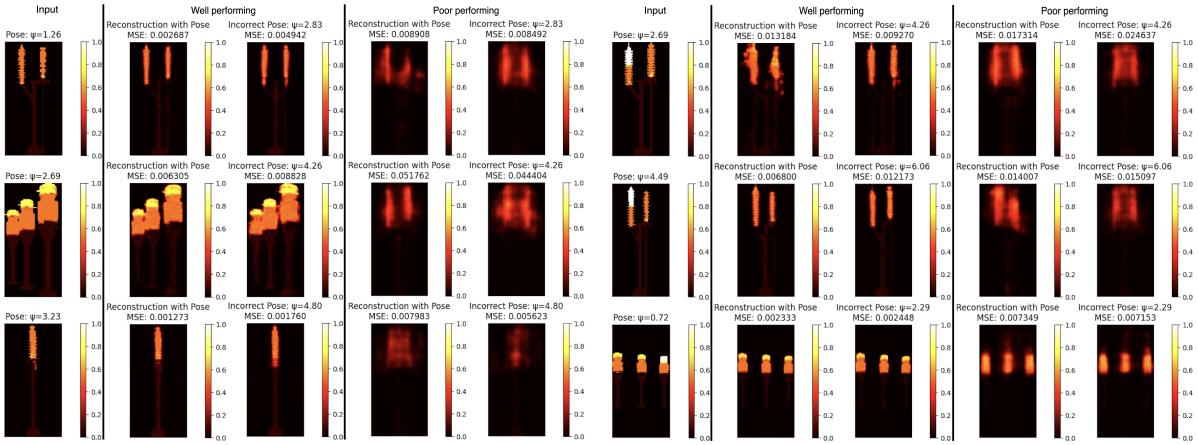
The misclassifications which occurred during the logistic regression step, as described in Section 3.4, have been summarized in Figure 6. These plots describe the influence of network parameters, indicated by the point colours, on misclassification count, grouped by network capacity and pose mode. The plots were derived from the Table in Appendix A.

Networks where more than 60% of the image reconstruction MSEs exceeded 0.01, were classified as failed attempts, these were not used to generate the plots in Figure 6. In order to relate the influence of the latent space size, to the input image size of 128×64 px, a capacity metric is introduced. In which the latent space size is divided by the number of pixels in the input images. This will yield a per-pixel capacity, as can be seen in Equation (10).

$$\text{Network Capacity} = \frac{\text{Latent Size}}{128 \times 64} \tag{10}$$

The default VAE network showed better performance, for smaller network capacities of $\frac{16}{8192} \approx 1.95 \cdot 10^{-3}$, $\frac{32}{8192} \approx 3.91 \cdot 10^{-3}$ and $\frac{64}{8192} \approx 7.81 \cdot 10^{-3}$. These networks seemed to have the best results using lower KL weights, which indicates that these unconditioned networks benefit from less regularization, which are lower in capacity.

Networks utilising one pose, namely distance and orientation seemed to show mixed results. Using distance many failed reconstructions occurred, as can be seen by the small amount of points for this mode in Figure 6, and the strike-through values in Appendix A. This intuitively seems to make sense. This parameter is the distance from the asset where the image was captured, but this information is mostly lost in our pre-processing pipeline due to the padding approach. This leads to a high level of confusion in these networks, due to the



(a) Reconstruction of normal images.

(b) Reconstructions of anomaly images.

Figure 7: Comparison between the reconstructions of normal and anomalous images, for well and poor performing networks, using correct and incorrect pose information.

conditioning on the distance metric.

Orientation however, is the best performing pose mode, as can be seen based on low occurring points in Figure 6. Even-though some well performing networks occurred at lower capacity networks, the majority of them occurred at higher ones, indicating that this additional representational capacity can benefit this pose mode. The best performing network, with 11 misclassifications, occurred at a capacity of $\frac{256}{8192} \approx 3.13 \cdot 10^{-2}$. Larger capacity networks tend to benefit from moderate to high KL regularization, as can be seen for the other pose modes. The influence of β weights seems relatively stable, indicating that the pose information is successfully captured in the latent space in these networks.

The two pose modes again seemed to show mixed results. Just as for the one pose distance metric, we can again see the same effect of the distance metric. Resulting in a high level of confusion caused by the conditioning. The position pose mode, was the best performing lowest capacity network, of 14 misclassifications. Overall, showing relatively robust performance at a capacity of $\frac{16}{8192} \approx 1.95 \cdot 10^{-3}$ and $\frac{32}{8192} \approx 3.91 \cdot 10^{-3}$, for all β parameters. Using higher capacity networks of $\frac{128}{8192} \approx 1.56 \cdot 10^{-2}$ and $\frac{256}{8192} \approx 3.13 \cdot 10^{-2}$, more regularization seems to be needed, as of moderate β values. For lower capacity networks, position occasionally had similar or better performance compared to the other pose modes. However, for high capacity networks this was not the case.

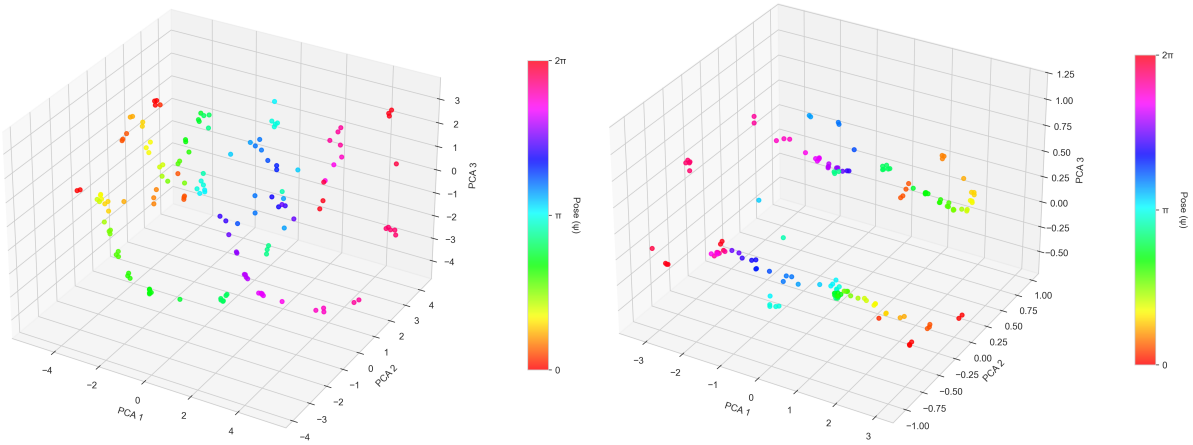
The final pose mode, 3: default, seems to be relatively robust, as indicated by the high amount of points and the low misclassification fluctuation of the different β parameters. This mode seems to have the general trend of having better performance using low capacity networks, with most β weights and low regularization, but again almost never outperforming orientation. The well performing instances can most-likely be accredited to the use of the orientation metric, which is the most telling factor in terms of when an anomaly can be seen or not, since orientation can drastically increase the networks ability to reconstruct different asset viewing angles, and thus increasing the MSE when an anomaly is used in the image input.

The valid CVAE misclassifications have been compared to their VAE counterparts. This resulted in the following metrics, which show how often the pose modes outperform the VAE networks; 1: distance - $\frac{6}{25} = 24.0\%$, 1: orientation - $\frac{27}{47} = 57.4\%$, 2: distance-rotation - $\frac{6}{26} = 23.1\%$, 2: position - $\frac{16}{50} = 32.0\%$ and 3: default - $\frac{15}{50} = 30.0\%$. Highlighting that orientation was the best performing pose mode.

4.2 Reconstruction Analysis

A more detailed assessment of the network's outputs will be had, in order to obtain a better understanding of the performance differences between CVAE networks. To achieve this, two networks will be compared, both from the best performing pose mode orientation. Namely, size 32, $\beta = 0.4$, $\lambda_{KL} = 0.1$ with 48 misclassifications and size 256, $\beta = 0.2$, $\lambda_{KL} = 0.01$ with 11 misclassifications.

First we will look at the model's ability to reconstruct images. These reconstructions for normal images can be seen in Figure 7a, and for anomaly images in Figure 7b. For both these figures, the first image column contain the images and pose used as an input to the trained networks. The second and third column contain reconstructions of this input for a correct and incorrect pose, for



(a) $\mathbf{Z}_\mu$ PCA of well performing model.

(b) $\mathbf{Z}_\mu$ PCA of poor performing model.

Figure 8: Comparison of PCA for a well and poor performing network.

the well forming network, while the fourth and fifth column contain the reconstructions for the poor performing model. More of these reconstructions can be found in Appendix B and C, for the 11 and 48 misclassification networks respectively. A colour-mapping was applied to the thermal images in order to easily see the thermal differences in them.

The differences between the well and poor performing models is immediately apparent. In both Figure 7a and 7b, we can see that the 11 misclassification network is able to almost always fully reconstruct the assets based on their latent representation, while the 48 misclassification network fails to do so. In both networks we can see the effect of uncertainty on the network, where the reconstruction contains blurred sections where the networks fails to fully reconstruct the images. The 11 misclassification network only has this blurring at sections of high complexity, such as the individual isolator disks. While for the 48 misclassification network has blurring in almost the entire image. This blurring only occurs less for this network when the asset can easily be distinguished, as can be seen in the bottom row of images in Figure 7b. Here, the three isolator heads are much different than other image representations. It can still be seen that the network is uncertain about the height of the centremost isolator though. The effect of pose information on the well performing network can be seen in the top row of Figure 7a. The network forms a mixed representation between the input image and the image which would occur at $\psi = 2.83$. This results in the rightmost isolator head being elongated, and the diagonal frame which connects the head to the base appearing in the right of the image. This shows the networks ability to interpret image and pose features in order to form a mixed representation of them. This only occurs when the network unable to distinguish the asset's representation based on image features alone, this can be seen from the middle row of Figure 7a, where the three isolator heads are easy to differentiate, resulting in only slight blurring around the heads. The general trend of the well performing network is that it can reconstruct the normal image from the anomalous ones. In the top row of Figure 7b, we can see that this is not always the case. Here, some artifacts can be seen, which possibly are representations where asset 5 can be seen. This can occur when some ambiguity is seen between the two assets. The presence of these reconstruction artifacts can be desirable, since this increases the reconstruction error, which allows the anomalies to be detected more easily. There is a trade-off though, the interpretability of the reconstructions could diminish, when too many unrelated features constantly appear in the image.

4.3 Latent Space Analysis

The latent space of the models mentioned in Section 4.2 will be analysed, to gain more insights on what the differentiating factors are between well and poor performing models. In order to do this, a Principal Component Analysis (PCA) was conducted on the $\mathbf{Z}_\mu$ variable of the latent space. This results in the dimensionality of the latent variable being reduced to the three main contributing ones. The can be seen for the 11 and 48 misclassification networks in Figure 8a and 8b respectively.

A clear distinction between the well and poor performing model, is the distribution of the points. The well performing model seems to follow cyclical patterns, where the pose is distributed relatively smoothly, with better separability. The poor performing model has clear discontinuities, especially

at $\psi = 0 = 2\pi$. This highlights the network's ability to represent data points, and interpret intermediate ones. We can also determine that the well performing model uses more volume to represent these points, where a larger 3D domain of the PCA axis is utilised. The PCA on the models' $\mathbf{z}_{\log \sigma^2}$ latent variable follows the same general trend as the $\mathbf{Z}_\mu$ ones. Where the well performing model structures the uncertainty variable in well organized cyclical patterns, in a larger domain. For the poor performing model this is more discontinuous and compressed. These Can be found in Appendix D.

4.4 Sample Distribution Analysis

Another way to differentiate well from poor performing models is the Gaussian distributions between the normal and the anomalous image reconstruction MSEs. When an error is calculated, which could correspond to both normal and anomalous distributions, i.e. when there is overlap between them, logistic regression performance suffers.

Two formulas will be introduced, to evaluate the performance of all valid models with regards to separability between normal and anomalous samples. These two functions describe the distance between the mean of the normal and anomalous distributions, normalized over the normal standard deviation in Equation (11) and the relative uncertainty of the anomaly distribution in Equation (12). These two functions can be used as measure to determine separability between the distributions. However, since Gaussian distributions are assumed, while multi-modal distributions can appear in the MSE distribution, inconsistencies can occur.

$$S_\mu = \frac{(\mu_A - \mu_N)}{\sigma_N} \quad (11) \qquad S_\sigma = \frac{\sigma_A}{\sigma_N} \quad (12)$$

The presented plot in Figure 9 shows the effect of β pose weight during training on misclassifications. Where the centre point colour represents the misclassification count, and the outer colour the used β weights. The misclassification count of the top 3% performing models is written as a label close to the points. More separability analysis plots can be found in Appendix E, but these will not be discussed since these align with previously discussed results.

The top performing models generally speaking appear in the top right quadrant of the plot, where there is a large difference between the means of the distributions, and the normal distribution is less skewed than the anomalous one. The most appearing β values here, range from VAE networks no β to moderate values of $\beta = 0.6$. From this

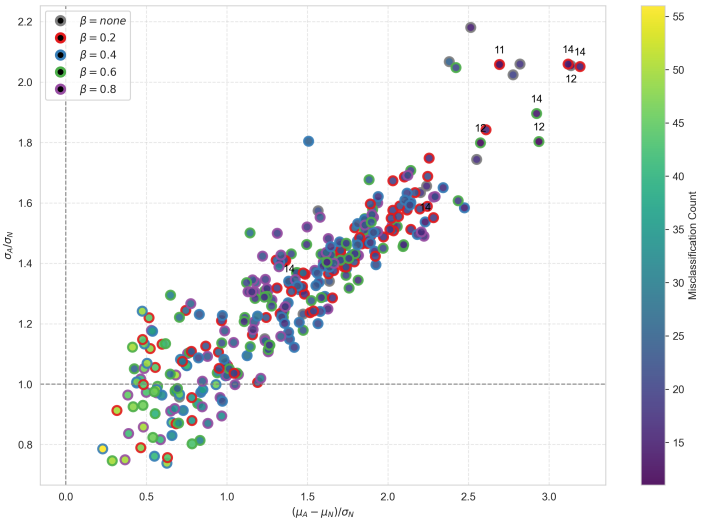


Figure 9: Separability analysis of misclassification count and β-weight influence.

we can conclude that β influence can lead to well performing models, but it is not strictly necessary.

5. Conclusions

In this work, a multi-modal CVAE framework was presented, which was used to evaluate the performance of thermal-image based anomaly detection. The effect and influence of robot pose based conditioning and network parameters was compared to a baseline VAE network, using empirical results to evaluate the most effective conditioning variable which enhances anomaly detection abilities. These empirical results include misclassification results, reconstruction abilities, PCA of the latent variables and MSE distribution separability analysis.

The most informative conditioning variable was orientation. This showed improved anomaly detection abilities by roughly 57.4% compared to the VAE baseline models. Distance based conditioning showed to be the worst performing model, resulting in significantly worse anomaly detection compared to orientation. This can likely be accredited to the pre-processing pipeline, where distance information is lost due to the zero padding approach.

The effect of network capacity, which is related to network size, was also analysed. Networks which benefit from lower KL regularization are of smaller capacity, especially VAE models. CVAE networks, especially using the well performing orientation conditioning, moderate regularization and pose reconstruction weight during training show the best results. Stable performance for different pose reconstruction weights, are a tell for a suitable conditioning variable.

The effect of well and poor performing models

on the structure of the latent variables was also conducted using PCA. Well performing models showed cyclical patterns, where the conditioning variable was continuously represented. This resulted in correct reconstructions, where blurring could be seen in places of uncertainty. In poor performing models the PCA of the latent variables showed discontinuous pose representation, where less of the 3D domain was effectively utilised, resulting in reconstructions which almost entirely showed blurring.

A separability metric was introduced, which showed how much the image reconstruction MSE distributions of the normal and anomalous samples differed. This revealed a general trend which well performing models have, where separability is high. This was not always accurate, since this metric assumes Gaussian distributions. In practise this metric may not always hold, since the reconstruction distributions may exhibit multi-modality.

To conclude, this work shows that pose conditioning can have a positive effect on anomaly detection abilities, when the weights of the KL divergence and the reconstruction terms are properly scaled. It also revealed that a simple, weighted multi-component loss function is difficult to balance. In order to fully leverage the benefit of pose condition on anomaly detection, future work should explore different approaches which balance these loss terms. Additionally, a more robust separability metric, which takes the multi-modal nature of the MSE image reconstruction distributions into account, should be implemented to gain further insights into when a model shows better anomaly detection performance.

References

[1] *World Energy Outlook 2024*, en-GB, 2024. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2024 (visited on 10/29/2024).

[2] J. Yang, R. Xu, Z. Qi, and Y. Shi, *Visual Anomaly Detection for Images: A Survey*, Sep. 2021. DOI: 10.48550/arXiv.2109.13157. [Online]. Available: http://arxiv.org/abs/2109.13157 (visited on 04/12/2024).

[3] B. Zong, Q. Song, M. R. Min, *et al.*, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection," en, Feb. 2018. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0- (visited on 06/04/2025).

[4] S. Mei, H. Yang, and Z. Yin, "An Unsupervised-Learning-Based Approach for Automated Defect Inspection on Textured Surfaces," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 6, pp. 1266–1277, Jun. 2018, ISSN: 1557-9662. DOI: 10.1109/TIM.2018.2795178. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8281622 (visited on 06/04/2025).

[5] X. Yan, H. Zhang, X. Xu, X. Hu, and P.-A. Heng, "Learning Semantic Context from Normal Samples for Unsupervised Anomaly Detection," en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3110–3118, May 2021, Number: 4, ISSN: 2374-3468. DOI: 10.1609/aaai.v35i4.16420. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16420 (visited on 06/04/2025).

[6] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/790410 (visited on 06/04/2025).

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, ISSN: 1063-6919, vol. 1, Jun. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1467360 (visited on 06/04/2025).

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016, pp. 770–778. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html (visited on 06/04/2025).

[9] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Super-Resolution," 2018, pp. 2472–2481. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Residual_Dense_Network_CVPR_2018_paper.html (visited on 06/04/2025).

[10] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," 2015, pp. 1395–1403. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Xie_Holistically-Nested_Edge_Detection_ICCV_2015_paper.html (visited on 06/04/2025).

[11] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556 [cs], Apr. 2015. DOI: 10.48550/arXiv.1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556 (visited on 06/04/2025).

[12] C. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics), en. Springer New York, NY, Aug. 2006, ISBN: 978-0-387-31073-2. [Online]. Available: https://link.springer.com/book/9780387310732 (visited on 06/04/2025).

[13] S. S. Khan and M. G. Madden, "A Survey of Recent Trends in One Class Classification," en, in *Artificial Intelligence and Cognitive Science*, ISSN: 1611-3349, Springer, Berlin, Heidelberg, 2010, pp. 188–197, ISBN: 978-3-642-17080-5. DOI: 10.1007/978-3-642-17080-5_21. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-17080-5_21 (visited on 06/04/2025).

[14] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, ISSN: 0899-7667. DOI: 10.1162/089976601750264965. [Online]. Available: https://doi.org/10.1162/089976601750264965 (visited on 06/04/2025).

[15] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," en, *Machine Learning*, vol. 54, no. 1, pp. 45–66, Jan. 2004, Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Kluwer Academic Publishers-Plenum Publishers, ISSN: 1573-0565. DOI: 10.1023/B:MACH.0000008084.60811.49. [Online]. Available: https://link.springer.com/article/10.1023/B:MACH.0000008084.60811.49 (visited on 06/04/2025).

[16] M. Sakurada and T. Yairi, *Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction | Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, Dec. 2014. [Online]. Available: https://dl-acm-org.tudelft.idm.oclc.org/doi/abs/10.1145/2689746.2689747?casa_token=sEe905wrnNAAAAAA:ShyxOrx8idPnIPuZSq8EvYVUUN-_P3vw6PTtnOXdyka9-vLVJK-ZhbTD6sm9g54ktcivRg99nc3p (visited on 06/04/2025).

[17] I. Golan and R. El-Yaniv, "Deep Anomaly Detection Using Geometric Transformations," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/5e62d03aec0d17facfc5355dd90d441c-Abstract.html (visited on 06/04/2025).

[18] A. A. Pol, V. Berger, C. Germain, G. Cerminara, and M. Pierini, "Anomaly Detection with Conditional Variational Autoencoders," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 1651–1657. DOI: 10.1109/ICMLA.2019.00270. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8999265 (visited on 06/05/2025).

[19] K. Sohn, H. Lee, and X. Yan, "Learning Structured Output Representation using Deep Conditional Generative Models," in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html (visited on 06/11/2025).

[20] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning*. MIT press Cambridge, MA, USA, Oct. 2015, vol. 1.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," en, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ISSN: 1938-7228, JMLR Workshop and Conference Proceedings, Jun. 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html (visited on 06/10/2025).

[22] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, arXiv:1312.6114 [stat], Dec. 2022. DOI: 10.48550/arXiv.1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114 (visited on 05/27/2025).

[23] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 05/29/2025).

[24] L. N. Smith, *A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay*, en, Mar. 2018. [Online]. Available: https://arxiv.org/abs/1803.09820v2 (visited on 06/11/2025).

[25] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951, Publisher: Institute of Mathematical Statistics, ISSN: 0003-4851. [Online]. Available: https://www.jstor.org/stable/2236703 (visited on 05/30/2025).

Terms and Abbreviations

ADAM Adaptive Moment Estimation. 7

CNN Convolutional Neural Network. 5
CVAE Conditional Variational Auto-Encoder. 1, 2, 4–9, 11, 16

DL Deep Learning. 2
DSO Distribution System Operator. 1

GMM Gaussian Mixture Model. 2

HOG Histogram of Oriented Gradients. 2

IEA International Energy Agency. 1

KL Kullback-Leibler. 2, 7–9, 11, 12, 16, 26

MSE Mean Squared Error. 2, 7–9, 11, 12, 16

PCA Principal Component Analysis. 10–12, 23
PDF Probability Density Function. 2

ReLU Rectified Linear Unit. 5, 6

SIFT Scale Invariant Feature Transform. 2
SLAM Simultaneous Localisation and Mapping. 1

VAE Variational Auto-Encoder. 2, 5–9, 11, 16

Appendix

A Misclassification Count

size (s)	λ_{KL}	0: Default	1: Distance	1: Orientation	2: Distance-Orientation	2: Position	3: Default
16	0.000100	15	20 | 22 | 28 | ~~23~~	23 | 17 | ~~21~~ | ~~28~~	19 | 19 | 34 | 41	19 | 21 | 17 | 18	20 | 19 | 21 | 18
16	0.001000	18	25 | 23 | 24 | 24	16 | 28 | 18 | 26	26 | 21 | ~~27~~ | 22	17 | 20 | 18 | 21	23 | 23 | 18 | 25
16	0.010000	22	22 | 24 | 27 | 21	18 | ~~23~~ | 35 | 23	22 | 33 | 19 | 25	14 | 21 | 18 | 20	19 | 23 | 18 | 18
16	0.100000	~~29~~	~~46~~ | 25 | 35 | 50	~~29~~ | ~~30~~ | 25 | 23	44 | 27 | 46 | 24	19 | 23 | 40 | 23	23 | 38 | 24 | 18
16	1.000000	~~33~~	~~28~~ | ~~22~~ | ~~22~~ | ~~24~~	~~32~~ | ~~32~~ | ~~29~~ | ~~32~~	~~31~~ | ~~49~~ | ~~26~~ | ~~24~~	~~31~~ | 50 | 46 | 50	~~39~~ | 49 | 46 | 42
32	0.000100	12	~~33~~ | ~~25~~ | 25 | 21	22 | 19 | 20 | 22	~~33~~ | ~~22~~ | 19 | 21	16 | 18 | 20 | ~~27~~	20 | 21 | 16 | 24
32	0.001000	17	17 | ~~20~~ | 22 | 24	16 | 22 | 23 | 19	19 | 18 | 21 | 26	20 | 19 | 24 | 19	21 | 20 | 20 | 20
32	0.010000	23	22 | 24 | ~~33~~ | 23	17 | 20 | 20 | 16	20 | 21 | ~~33~~ | 31	19 | 23 | 21 | 20	22 | 25 | 19 | 23
32	0.100000	~~36~~	23 | 46 | 28 | 27	26 | 48 | 31 | 25	20 | 36 | 25 | 33	25 | 27 | 22 | 20	28 | 21 | 21 | 24
32	1.000000	~~33~~	~~26~~ | ~~26~~ | ~~21~~ | ~~24~~	~~31~~ | ~~28~~ | ~~32~~ | ~~29~~	~~33~~ | ~~27~~ | ~~25~~ | ~~26~~	48 | 45 | 48 | 41	44 | 35 | 30 | 45
64	0.000100	18	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~46~~ | ~~33~~ | 21	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	23 | 24 | 21 | ~~33~~	~~33~~ | 23 | ~~33~~ | 29
64	0.001000	20	17 | ~~33~~ | ~~33~~ | ~~33~~	18 | 19 | 17 | ~~33~~	19 | ~~33~~ | ~~33~~ | ~~33~~	21 | 25 | 21 | 24	22 | 23 | 24 | 22
64	0.010000	22	17 | 23 | ~~33~~ | ~~33~~	18 | 17 | 18 | 17	22 | 22 | ~~33~~ | ~~33~~	22 | 22 | 24 | 22	21 | 24 | 19 | 23
64	0.100000	31	24 | 42 | ~~22~~ | 30	24 | 27 | 21 | 22	35 | 30 | ~~33~~ | 25	18 | 49 | 21 | 19	35 | 28 | 20 | 24
64	1.000000	~~33~~	~~23~~ | ~~25~~ | ~~27~~ | ~~22~~	~~30~~ | ~~31~~ | ~~28~~ | ~~30~~	~~45~~ | ~~50~~ | ~~45~~ | ~~45~~	44 | 46 | 42 | 43	44 | 56 | 50 | 46
128	0.000100	25	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	14 | ~~33~~ | ~~33~~ | 28	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	22 | ~~33~~ | ~~33~~ | ~~33~~	22 | ~~33~~ | ~~33~~ | ~~33~~
128	0.001000	16	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	18 | 19 | 12 | 21	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	21 | 19 | 24 | ~~33~~	22 | 20 | 20 | ~~33~~
128	0.010000	19	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	17 | 17 | 17 | 18	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	20 | 22 | 19 | 25	23 | 21 | 22 | 21
128	0.100000	40	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	18 | 46 | 22 | 24	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	22 | 23 | 21 | 33	19 | 24 | 22 | 18
128	1.000000	~~32~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~30~~ | ~~30~~ | ~~28~~ | ~~45~~	~~33~~ | ~~33~~ | ~~25~~ | ~~33~~	44 | 49 | 27 | 35	50 | 41 | 50 | 47
256	0.000100	~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~
256	0.001000	~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	14 | 14 | 14 | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | 24 | ~~33~~ | ~~33~~	22 | ~~33~~ | ~~33~~ | ~~33~~
256	0.010000	~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	11 | 15 | 12 | 16	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	24 | 17 | 24 | 24	21 | 21 | 22 | 22
256	0.100000	~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | 38 | 30 | 29	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	23 | 25 | 16 | 25	44 | 25 | 43 | 20
256	1.000000	~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	~~33~~ | ~~33~~ | ~~33~~ | ~~33~~	47 | 45 | 27 | 34	44 | 33 | 46 | 50

Table 5: All misclassifications of various pose modes and network parameters.

The first two columns indicate which size and KL weight were used. The remaining columns indicate which pose modes were used, as described in Table 4. The VAE networks contain one number, while the CVAE networks have four numbers per configuration. This due to the different β parameters used: $\beta = 0.2|\beta = 0.4|\beta = 0.6|\beta = 0.8$, while the VAE equivalent only used KL- and image reconstruction loss, and thus no β parameter was utilised. The misclassifications were highlighted to show their significance. Red numbers indicate the best performing model per row, i.e. per size and λ_{KL} combination, while blue numbers indicate the best performing network per pose mode. Green numbers show the best performance for both row and pose mode. Some networks yielded sub-optimal reconstructions, were the images failed to show the original asset. Networks where more than 60% of the image reconstruction MSEs exceeded 0.01, were classified as failed attempts. In such cases, the corresponding misclassification counts are shown with a strike-through.

B Well Performing Network

B.1 Normal Image Reconstructions

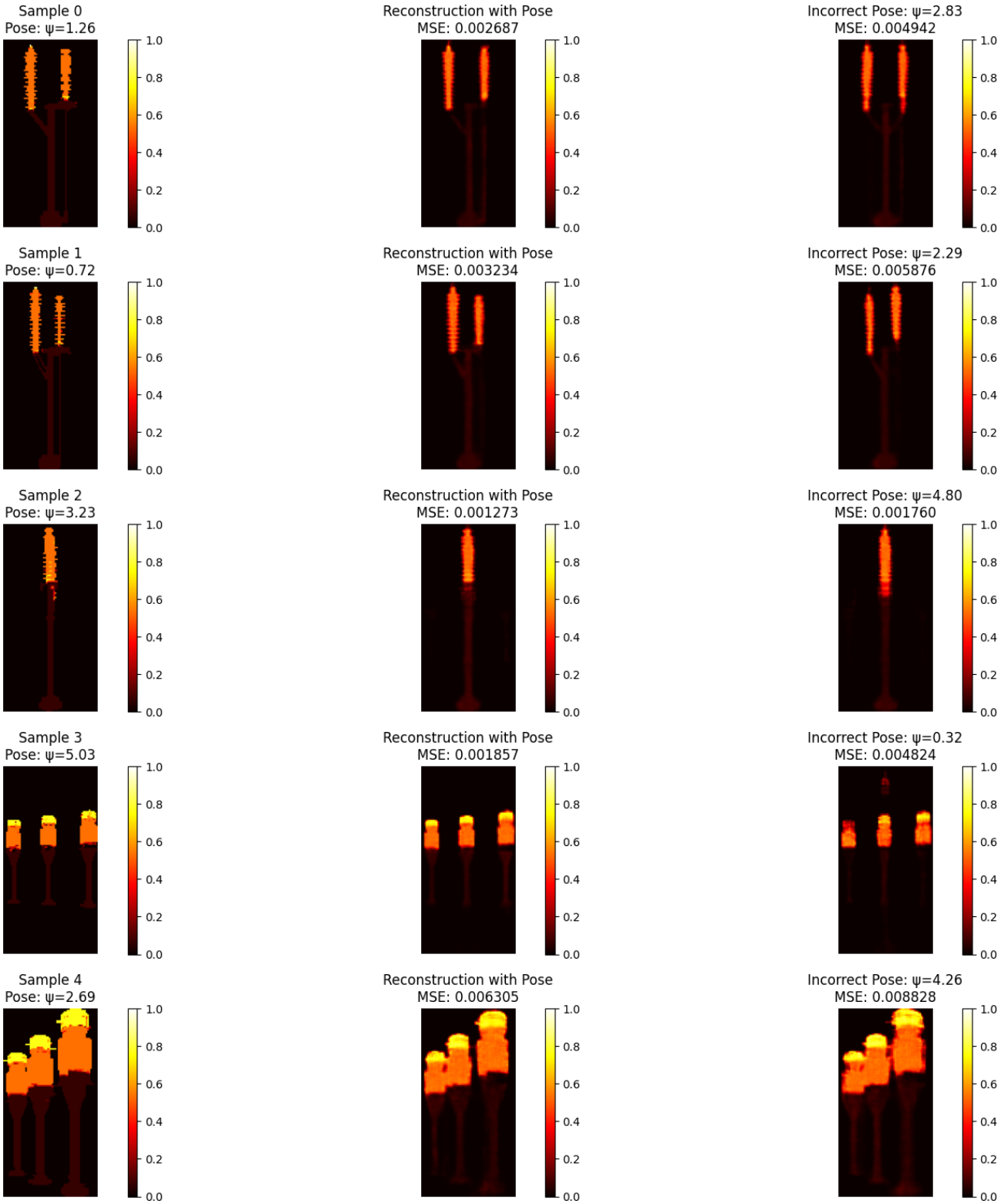


Figure 10: Normal reconstructions of 1: orientation, size 256, $\beta = 0.2$, $\lambda_{KL} = 0.01$ with 11 misclassifications.

B.2 Anomaly Asset 7 Reconstructions

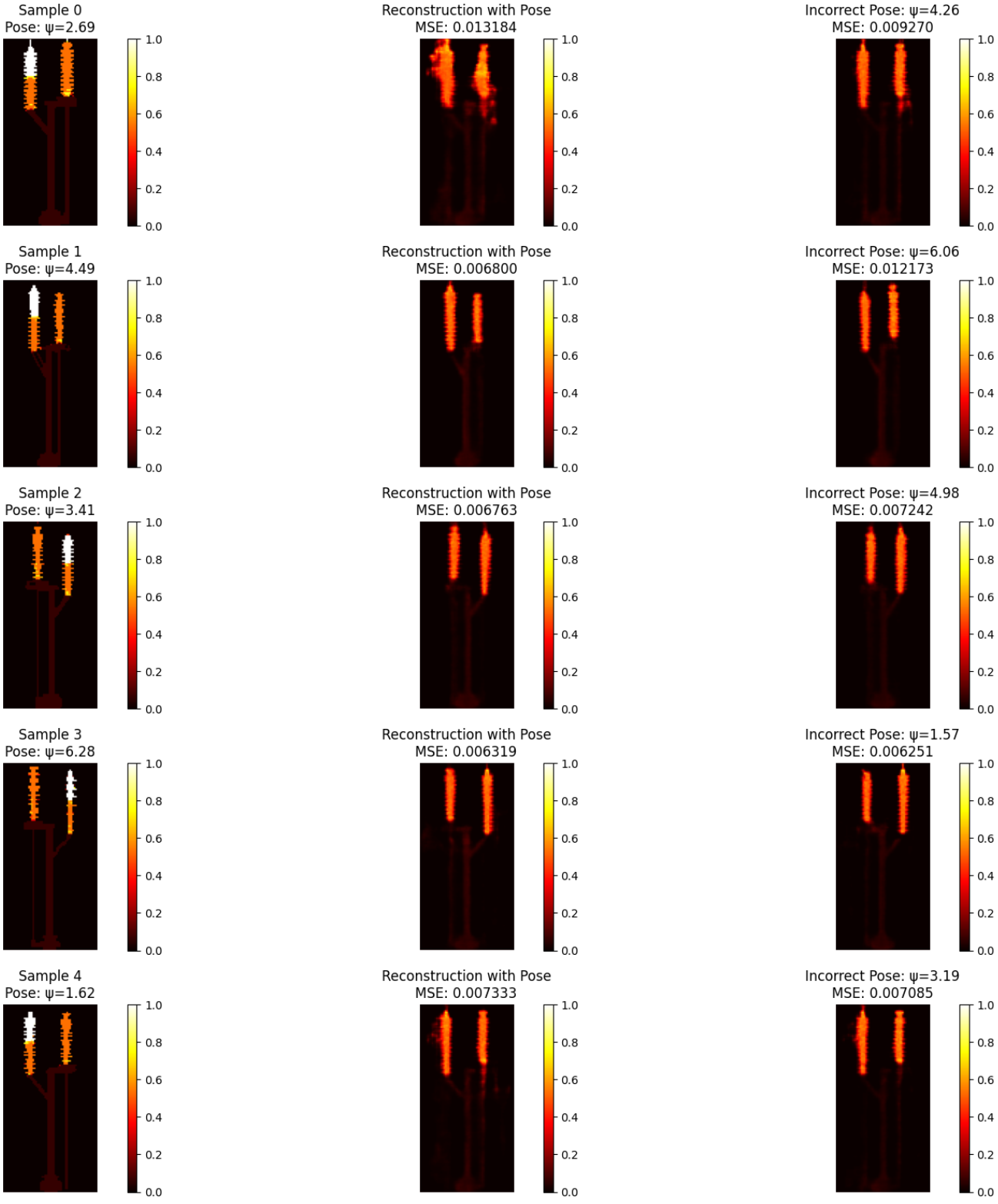


Figure 11: Asset 7 reconstruction of 1: orientation, size 256, $\beta = 0.2$, $\lambda_{KL} = 0.01$ with 11 misclassifications.

B.3 Anomaly Asset 5 Reconstructions

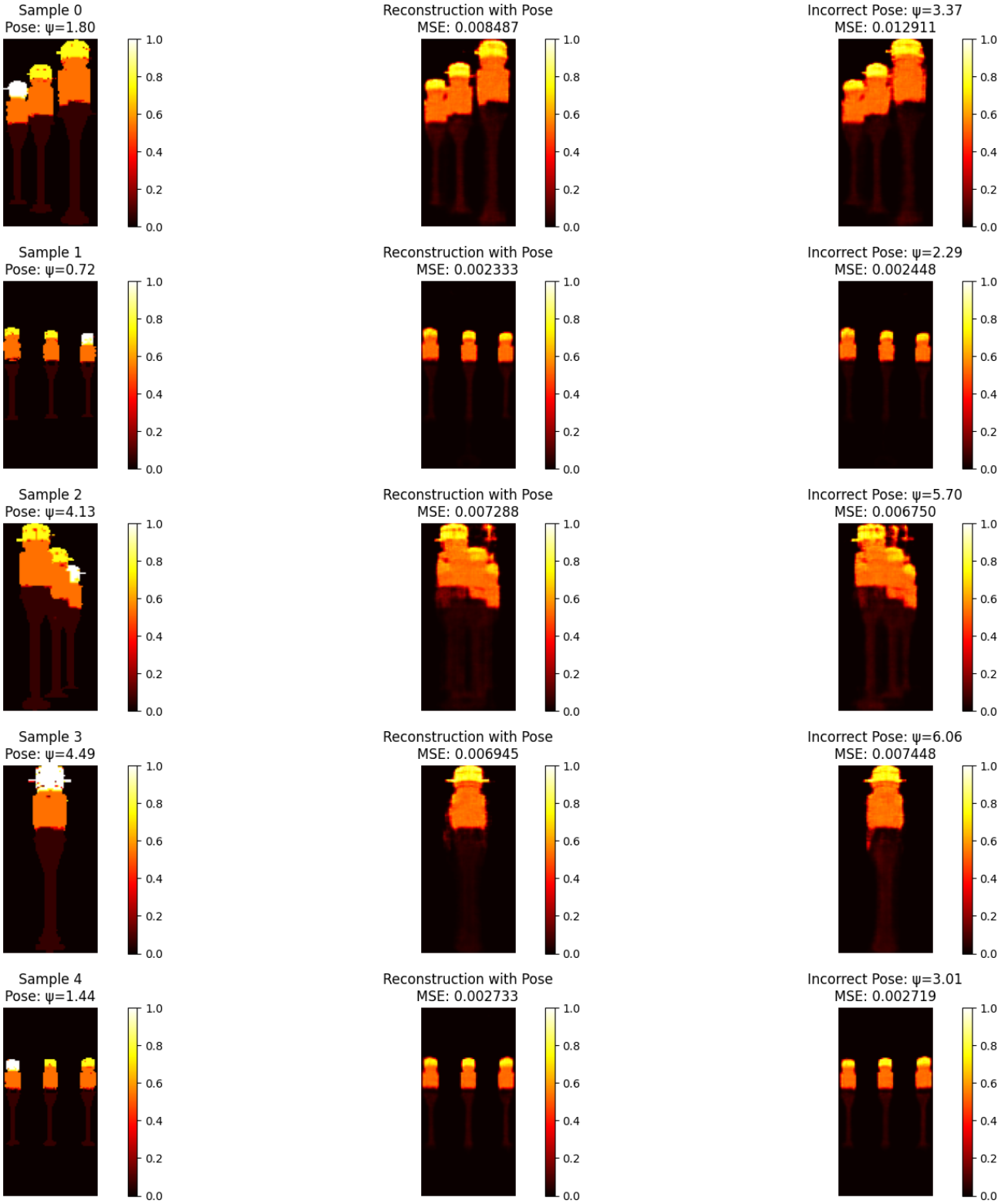


Figure 12: Asset 5 reconstructions of 1: orientation, size 256, $\beta = 0.2$, $\lambda_{KL} = 0.01$ with 11 misclassifications.

C Poor Performing Network

C.1 Normal Image Reconstructions

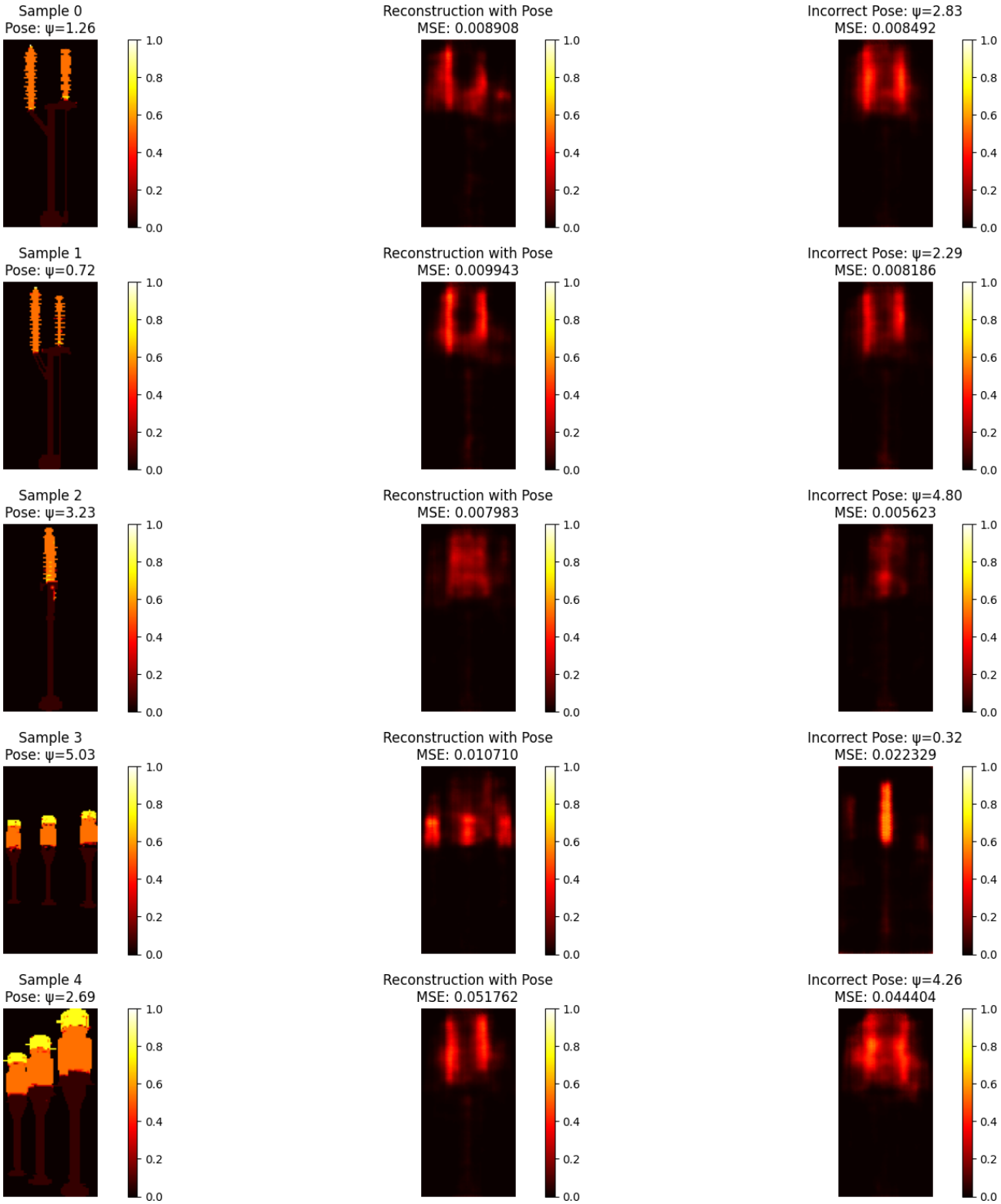


Figure 13: Normal reconstructions of 1: orientation, size 32, $\beta = 0.4$, $\lambda_{KL} = 0.1$ with 48 misclassifications.

C.2 Anomaly Asset 7 Reconstructions

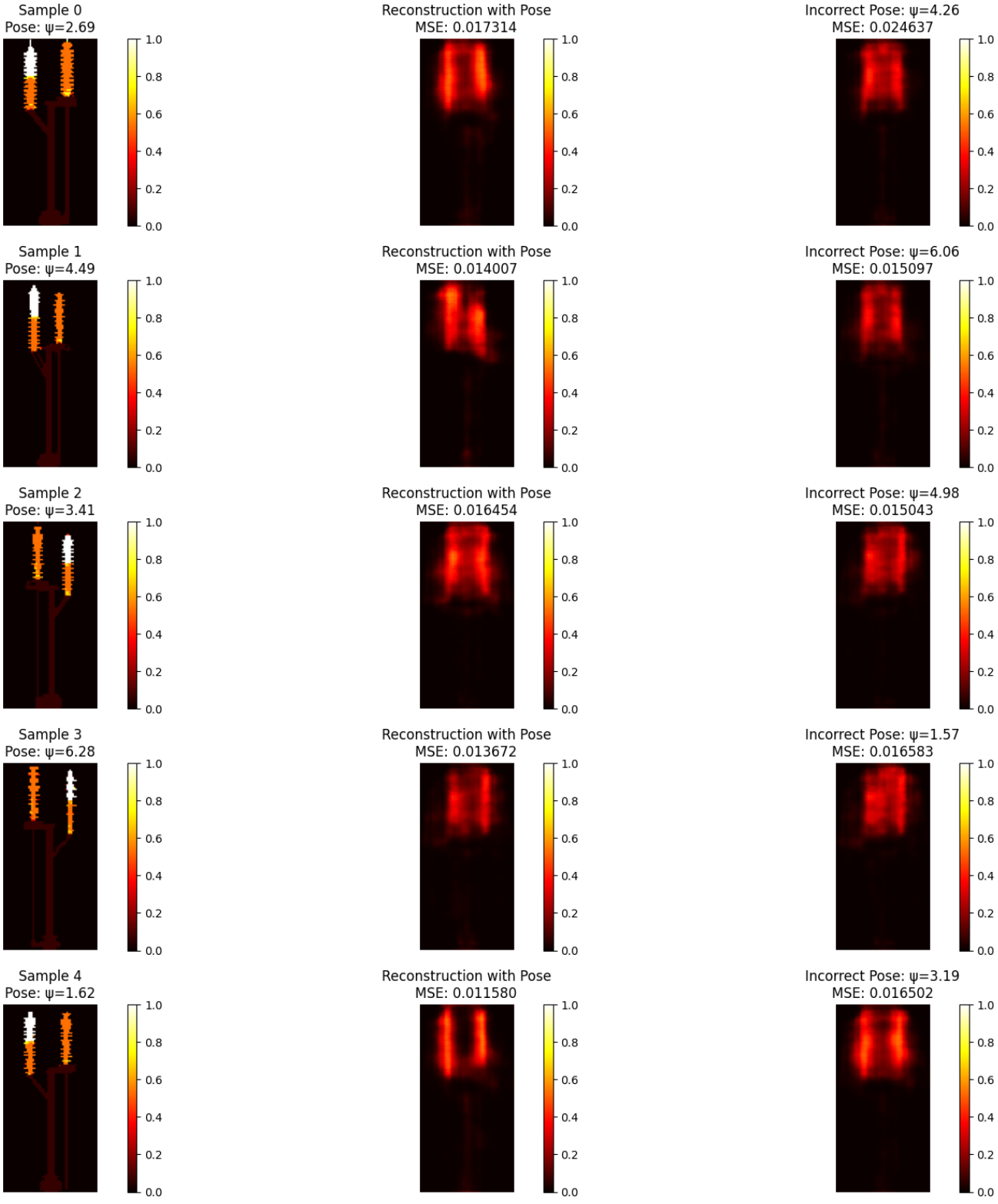


Figure 14: Asset 7 reconstructions of 1: orientation, size 32, $\beta = 0.4$, $\lambda_{KL} = 0.1$ with 48 misclassifications.

C.3 *Anomaly Asset 5 Reconstructions*

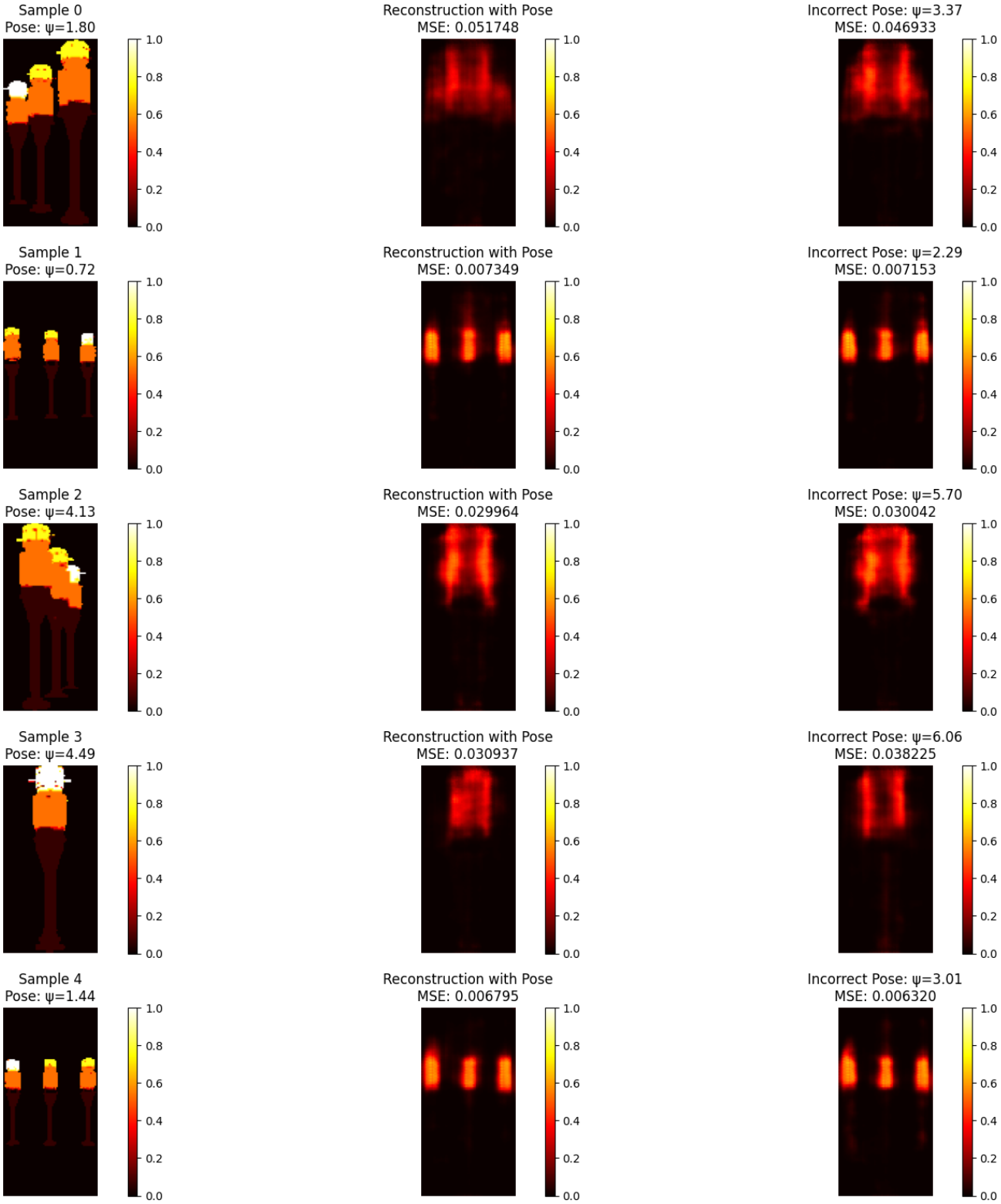


Figure 15: Asset 5 reconstructions of 1: orientation, size 32, $\beta = 0.4$, $\lambda_{KL} = 0.1$ with 48 misclassifications.

D $\mathbf{z}_{\log \sigma^2}$ PCA

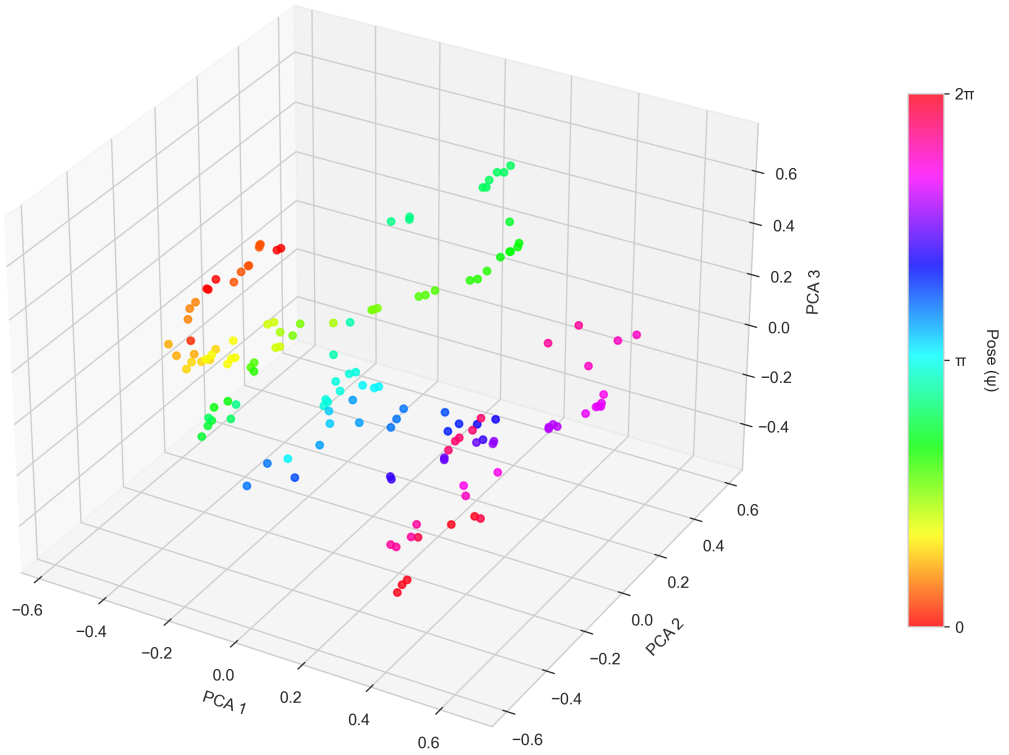


Figure 16: $\mathbf{z}_{\log \sigma^2}$ PCA of 1: orientation, size 256, $\beta = 0.2$, $\lambda_{KL} = 0.01$ with 11 misclassifications.

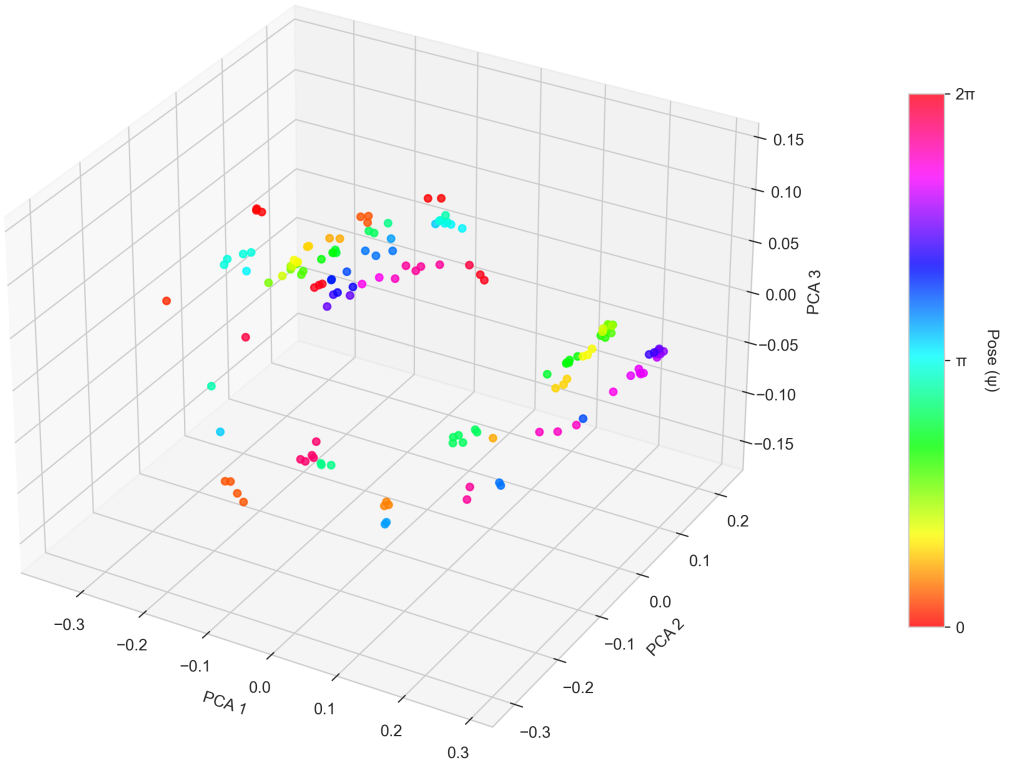


Figure 17: $\mathbf{z}_{\log \sigma^2}$ PCA of 1: orientation, size 32, $\beta = 0.4$, $\lambda_{KL} = 0.1$ with 48 misclassifications.

E Separability Analysis

E.1 Pose Mode Separability Analysis

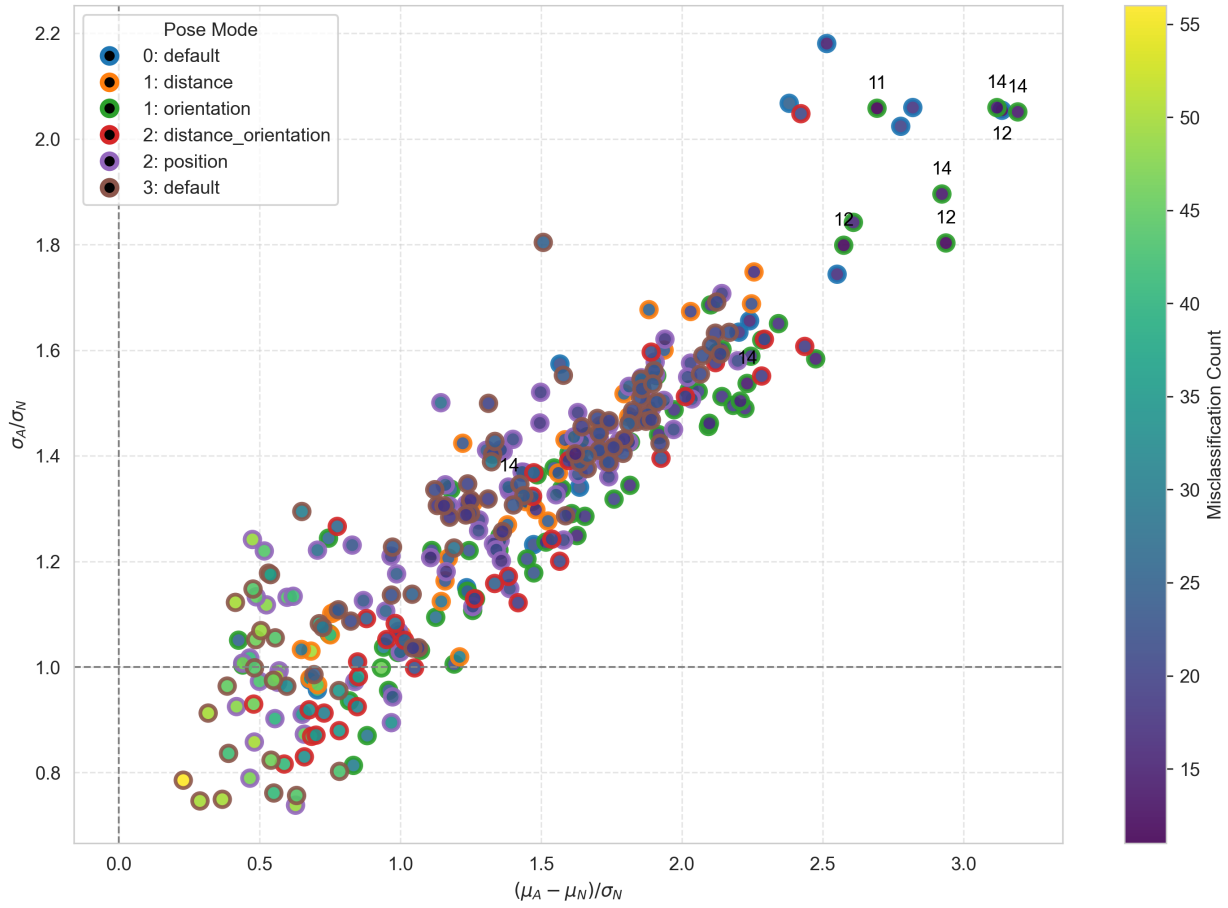


Figure 18: Separability analysis of misclassification count and pose mode influence.

E.2 Size Separability Analysis

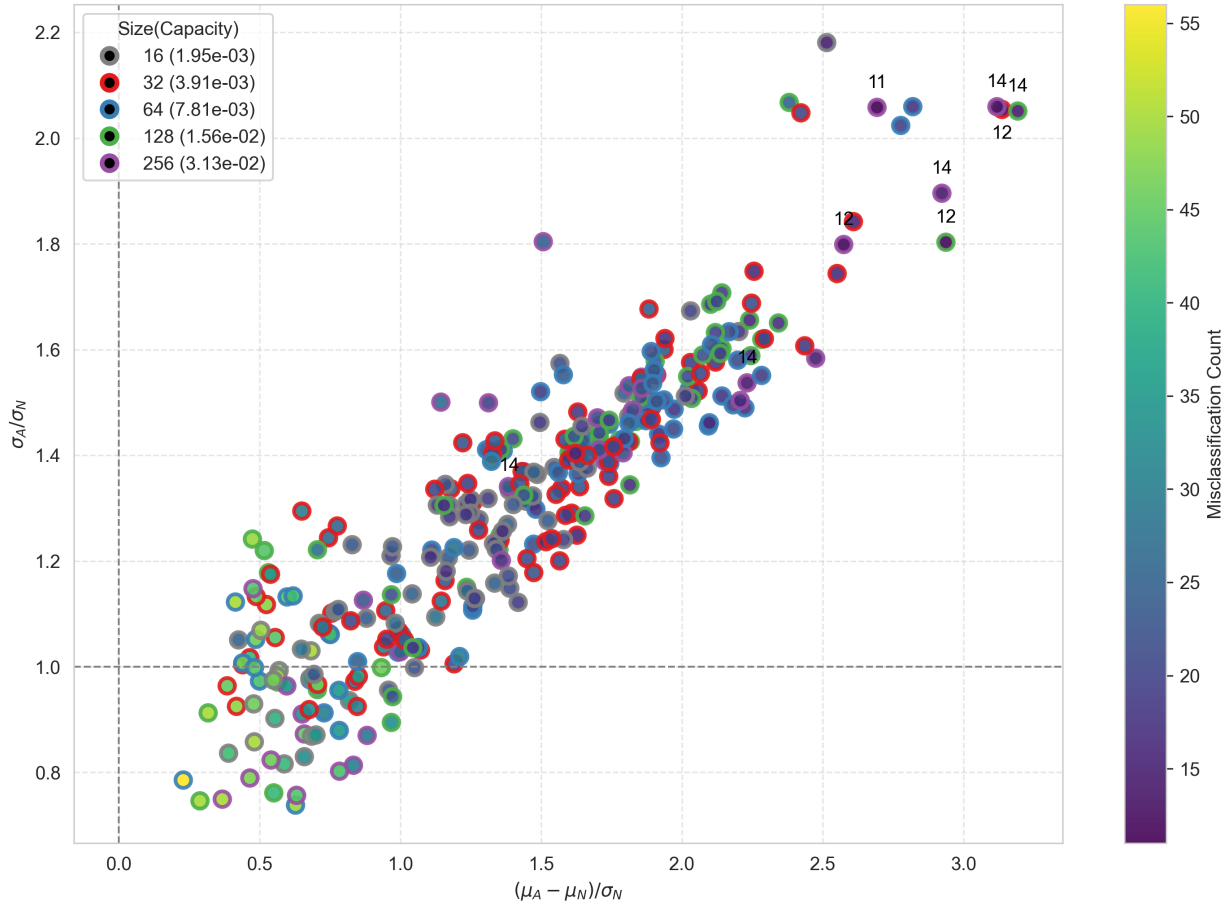


Figure 19: Separability analysis of misclassification count and size influence.

E.3 KL-weight Separability Analysis

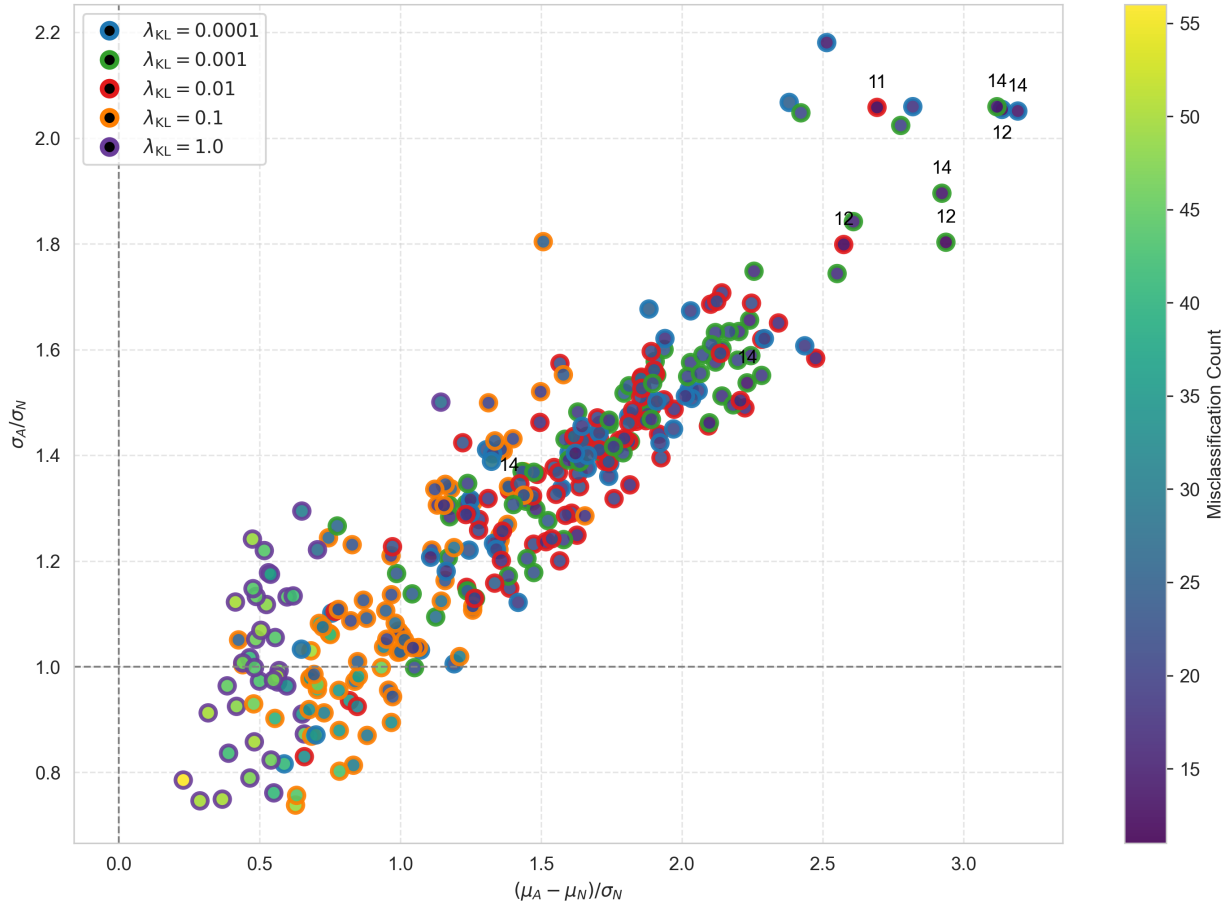


Figure 20: Separability analysis of misclassification count and λ_{KL} influence.