MoveAE: Modifying Affective Robot Movements Using Classifying Variational Autoencoders

Michael Suguitan
mjs679@cornell.edu
HRC^2 Lab, Cornell University
Ithaca, New York, United States

Randy Gomez
r.gomez@jp.honda-ri.com
Honda Research Institute Japan
Wako, Saitama, Japan

Guy Hoffman
hoffman@cornell.edu
HRC^2 Lab, Cornell University
Ithaca, New York, United States

ABSTRACT

We propose a method for modifying affective robot movements using neural networks. Social robots use gestures and other movements to express their internal states. However, a robot's interactive capabilities are hindered by the predominant use of a limited set of preprogrammed or hand-animated behaviors, which can be repetitive and predictable, making sustained human-robot interactions difficult to maintain. To address this, we developed a method for modifying existing emotive robot movements by using neural networks. We use hand-crafted movement samples and a classifying variational autoencoder trained on these samples. Our method then allows for adjustment of affective movement features by using simple arithmetic in the network's latent embedding space. We present the implementation and evaluation of this approach and show that editing in the latent space can modify the emotive quality of the movements while preserving recognizability and legibility in many cases. This supports neural networks as viable tools for creating and modifying expressive robot behaviors.

CCS CONCEPTS

• **Artificial intelligence** → **Cognitive robotics**; • **Machine learning** → *neural networks*; • **Human-centered computing** → HCI theory, concepts and models.

KEYWORDS

Social robots; deep learning; neural networks; affective generation

ACM Reference Format:
Michael Suguitan, Randy Gomez, and Guy Hoffman. 2020. MoveAE: Modifying Affective Robot Movements Using Classifying Variational Autoencoders. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI '20), March 23–26, 2020, Cambridge, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3319502.3374807

1 INTRODUCTION

In this work, we demonstrate the use of neural networks to modify the affective qualities of movements for an expressive robot. Current robot movement generation methods demand a deep understanding of the domain and its feature space, rendering these processes costly and hard to implement. Conversely, neural networks used in deep learning are able to learn the feature space on their own, reducing the dependency on domain knowledge. Neural networks may thus be applicable to the creation of expressive robot movements.

Robots designed for social interaction are becoming more common in spaces such as homes and storefronts. Movements and gestures are important modes of nonverbal communication that are unique to robots compared to other agents without physical bodies [11, 15]. There are various methods for creating expressive movements, from manual trajectory editing interfaces to learning from demonstration (LfD) [2]. However, these methods can be slow and often require prior knowledge of a specific robot platform. These techniques are thus difficult to implement and lacking in generalizability. To more quickly create new behaviors, roboticists sometimes turn to adjusting affective qualities of existing robot movements [4, 8, 21, 27]. Adjustment is easier than authoring new movements, but still requires technical knowledge of kinematics and movement theory. These pitfalls lead to robot behaviors usually being preprogrammed, creating a novelty effect that stunts long-term interaction and conveys a lack of intelligence [7, 28].

At the same time, advancements in deep learning have enabled the creation of data-driven neural network models that can learn complex features given sufficient data. These have enabled various applications ranging from temporal forecasting to image generation [29]. While these methods have seen success in tasks such as audio-visual perception and generation, they have remained largely unadopted for generating robot behaviors, where most algorithms are based on traditional machine learning methods or rely on problem-specific heuristics [13]. Neural networks can reduce the dependency on domain knowledge and heuristics by learning the features directly from the input data. Recently, neural networks have been developed for modifying high-level features in domains such as images [17] and audio [24] by editing low-level parameters in a learned "latent" embedding space. These works used the same approach for both images and audio, showing that neural networks can be more domain-agnostic and generalizable than heuristic methods.

To address the problem of repetitive movements in interactive robots, we propose to use deep learning techniques, particularly variational autoencoders (VAEs), classification networks, and latent space editing methods, to modify affective movement features for a low-degree-of-freedom (DoF) robot. We first learn low-dimension latent representations of the robot's affective movements. These latent representations can be used to both reconstruct the original movement and classify the movements by the intended emotion (happy, sad, angry). We then modify the valence and arousal features of the movements by using simple arithmetic operations in the latent embedding space. Our contributions are:

HRI '20, March 23–26, 2020, Cambridge, United Kingdom

ACM ISBN 978-1-4503-6746-2/20/03…$15.00
https://doi.org/10.1145/3319502.3374807

- A classifying variational autoencoder neural network architecture that compresses expressive robot movements into a lower-dimension latent space. The lower-dimension latent representations can reconstruct the original movements and are separated by emotion class.
- A method using linear regression to map the latent space representations into the circumplex emotion model dimensions of valence and arousal.
- An algorithm and interface for modifying the valence and arousal of the movements.
- Objective and subjective evaluations to assess the validity of this approach, in the form of neural network performance metrics and an online survey.

2 RELATED WORK

We review works in affective robot movements and neural network applications for affective robotics and latent feature modification.

2.1 Affective robot movements

Many prior works in human-robot interaction (HRI) categorize robot emotions into discrete classes according to Ekman's six categories: happiness, sadness, anger, surprise, fear, disgust [6]. In contrast, the circumplex model places emotion classes on the continuous dimensions of valence and arousal [22], with valence corresponding to positivity and negativity and arousal corresponding to high and low energy. The circumplex model illustrates the qualitative relationships between the emotions and its continuous dimensions are conducive for quantitative operations, making it suitable for adoption in numerical models.

2.2 Robot movement generation / modification

Movements and gestures are primary ways for robots to express their internal emotive states, and methods for designing affective robot movements have been extensively studied [13].

2.2.1 Generation. There are many approaches for generating robot movements, from low-level manual trajectory editing to high-level demonstrative techniques such as LfD [2]. These methods, however, have several drawbacks. Editing trajectories is time-consuming and unintuitive for non-roboticists, while directly manipulating a robot for LfD may be difficult to perform in real-time. LfD can be performed indirectly by attaching sensors to a human demonstrator, but this introduces the correspondence problem of mapping a human movement to a non-human embodiment. This has been addressed in many works within the graphics community, often using heuristic mappings from human poses to animate animals or other creatures [23, 30, 35]. Alissandrakis et al. explored heuristic methods to address this correspondence problem for robots [1], though their approach required extensive knowledge of the embodiment's kinematics. These difficulties lead robot movements to be largely preprogrammed and repetitive.

2.2.2 Modification. Modifying existing movements can be used to quickly expand a robot's library of movements, but still demands a high level of technical knowledge. As discussed by Karg et al. [13], most techniques used to modify affective robot movements rely on prior heuristic knowledge of robotics and the kinematics of a specific platform. These approaches typically adjust movement features that have been empirically found to be important for conveying affect, such as gaze direction [8, 21] or speed [27]. Desai et al. used a simulation of a quadrupedal robot with editable movement parameters such as walking pattern, speed, and body angle to adjust the affective quality of its gait [4]. The interface and method used was accessible compared to manual trajectory editing techniques, but still required a high level of domain knowledge.

2.3 Neural network applications for affective robotics and latent feature modification

The strength of neural networks compared to heuristic methods is their ability to learn complex and intractable data features with less dependence on domain knowledge and manual feature engineering. Neural networks have found success in complex applications for affective computing, primarily in perceptual tasks such as emotion recognition [12, 20], though some works have explored using neural networks for affective speech and expression generation [5, 33].

2.3.1 Affective robotics. Apart from emotion recognition, there have been few applications of neural networks for affective robotics. With regards to movement generation, Rodriguez et al. used a generative adversarial network (GAN) to generate talking gestures for a Pepper robot [25], but mostly generated random movements that did not consider affect. In more affect-oriented work, Zhou et al. compared hand-designed and network-learned feature costs for editing affective handovers [38]. The results showed that the hand-designed features were more suitable for expressing simple styles such as happy and sad, but the network could be preferable for complicated styles such as hesitant. This suggests that neural networks may be a better option for more complex affect expression.

2.3.2 Latent feature modification. Autoencoders are neural networks that learn a latent space to compress high-dimensional data into low-dimensional representations. The learned latent space can also be used to modify high-level features by editing the low-level parameters. Larsen et al. used this approach to modify discrete features of face images, such as gender and facial hair [17]. Roberts et al. extended this technique to modify continuous features of music, such as note density and pitch [24]. These works used the same general techniques for two very different domains, demonstrating the potential to use neural networks for modifying data features with less domain knowledge compared to heuristic methods.

The capabilities of neural networks for feature modification can be applied to affective robot movements. This intersection of HRI and deep learning can mitigate the novelty effect by continuously updating a robot's behavior library. An ever-growing repertoire of behaviors would help imbue robots with a sense of affective autonomy and may promote prolonged human-robot interactions.

3 NEURAL NETWORK BACKGROUND

Neural networks are the foundational models used in deep learning, approximating a transfer function from input data to output predictions. Compared to simple linear perceptrons [26], modern neural networks use varied activation functions, convolutions, and recurrence in their layers to create a non-linear model between the input and output. These layers can be arranged into various network

components such as encoders, decoders, or classifiers. Network components can then be combined into larger architectures such as image classifiers [16], recurrent networks [19], and autoencoders for dimensionality reduction [10]. Neural networks are trained by defining loss functions for the desired objectives, such as categorical cross-entropy for classification or mean error for reconstruction. Before training, the input data is split into training and testing sets. The training set is repeatedly passed through the network to optimize the layer parameters to minimize the loss functions and achieve the objectives. The test set is held out and does not update the network parameters, but is instead used to validate the model's performance on unseen data.

3.1 Variational autoencoders (VAE)

The primary network architecture used for this work is a variational autoencoder (VAE), which compresses input data into a latent embedding space while also giving this space a known structure.

Autoencoders are comprised of two components: encoders to compress the input data into a latent space, and decoders to decompress the latent space into reconstructions of the original inputs. Traditional autoencoders seek to minimize the reconstruction loss, which is defined as the difference between the input data and output reconstruction. VAEs additionally implement a Kullback-Leibler (KL) divergence objective, which structures the latent space into a Gaussian distribution. β-VAE is a further modification that implements weighing between the reconstruction and KL loss, allowing for the relative importance of the objectives to be tuned [9].

The combination of the reconstruction loss and KL divergence ensures that decoding from a random sample in the known latent distribution results in a valid realistic data sample. In lieu of random sampling, the original data can also be edited in the latent space to modify high-level features. This has enabled the use of VAEs in various applications such as image modification [17] and musical style transfer [24]. GANs were also considered and can extend VAEs to achieve better results [17], but their notorious training difficulty makes simple VAEs a better choice for our purpose.

3.2 Latent space editing to modify features

Latent space modification in the aforementioned prior works [17, 24] was achieved by calculating "attribute vectors" $\vec{a}_f$ in the latent space for modifying high-level features f (e.g., hair color, musical pitch). The vector $\vec{a}_f$ can be seen as a latent-space translation in the direction of data points that contain the feature of interest.

Given a latent-space representation of a data sample $\vec{x}_0$, the high-level features are modified by adding these attribute vectors. The degree of modification for a given feature is controlled with a weight parameter c_f.

$$\vec{x} = \vec{x}_0 + \sum_f c_f \vec{a}_f$$

The modified latent representation $\vec{x}$ is then passed through the decoder of the VAE to generate the new modified data sample.

In the face image modification work mentioned above [17], the features were binary (e.g. mustache or no mustache, blonde or not blonde). The attribute vectors were calculated as the difference between the mean latent vectors $\vec{\mu}_f$ of the "yes" and "no" groups.

$$\vec{a}_f = \vec{\mu}_{f,yes} - \vec{\mu}_{f,no}$$

For music modification [24], features were continuous (e.g. note density, pitch, average interval). The attribute vectors were calculated by first ranking the samples in terms of intensity (e.g. high vs low note density) and taking the difference between the mean latent vectors of the highest and lowest quartiles.

$$\vec{a}_f = \vec{\mu}_{f,Q_{high}} - \vec{\mu}_{f,Q_{low}}$$

4 IMPLEMENTATION

To illustrate our approach, we implemented a system to generate gestures expressing three emotions on a small desktop robot.

4.1 Robot platform

We used the social robot Blossom as a test platform (Figure 1) [32]. Blossom features four DoFs of head motion achieved by four motors: yawing through one motor in the base, and pitching, rolling, and vertical translation through three motors at the front, left, and right sides of the central tower structure. The tower motors actuate the head by reeling in cables connected to the head platform. Blossom has few DoFs compared to other robots but is still capable of expressive movements, making it suitable as a simple testbed.

We collected a dataset of emotive Blossom movements by asking volunteers to puppeteer the robot to display three main emotions: happy, sad, and angry. Movements are created with a phone application that translates the movement of the phone directly into the movement of Blossom's head. The dataset consists of approximately 25 movements per emotion class, each recorded at 10 Hz. Because neural networks require the input data to be consistently-sized, the movements are cut down by chunking them into sliding three-second windows every 1.5 seconds (Figure 2). The resulting dataset thus contains over 5,000 120D samples [1].

[1] 30 (3 seconds, 10 Hz) points × 4 DoF = 120D.

Figure 1: The Blossom robot. The exterior (left) is made of soft materials while the interior mechanism (right) consists of a central tower structure from which the head platform is suspended by elastic bands. The head platform has four degrees of freedom.

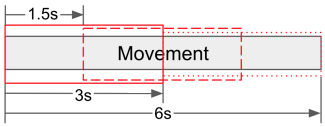

Figure 2: Illustration of how the movement data is "chunked" into three-second windows with 1.5-second overlaps to be used by the network. In this example, this six-second movement will yield three samples.

4.2 Neural network

We constructed the neural network with Keras and TensorFlow [3].

4.2.1 Classifying VAE architecture. Figure 3 shows the network architecture which consists of a VAE with an additional emotion classifier. The role of the VAE is to compress the 120-dimensional input movement data into a lower-dimension latent space. The classifier ensures that this latent space is separable by the emotion classes (happy, sad, angry). The network is based on convolutional layers and the parameters are detailed in Table 1. We chose convolutions over recurrence due to easier training and adjustable temporal reception [34]. The number of filters corresponds to the number of kinematic features to detect. Kernel size controls the receptive field, with a larger size denoting increased temporal correspondence between timesteps. Dropout was used to reduce overfitting given the small data size. The training objectives are:

- Reconstruction loss to ensure that the output reconstructions are identical to the input data.
- KL divergence to give the latent space a Gaussian structure.
- Classification loss to separate the learned latent space by emotion class.

4.2.2 Latent space → circumplex model. The dimensions in the learned latent space do not meaningfully represent human-readable affect. In order to both visualize the gestures and allow users to modify them, we use linear regression to map the latent space onto the circumplex model's valence and arousal dimensions. First, we calculate the centroids of each emotion class in the nD latent space. Each centroid is then recalculated by weighing each sample by the inverse of its distance to the original unweighted centroid. We use these weighted centroids to diminish the importance of movement samples that may be confused with another emotion class. An ordinary least squares linear regression model fits the nD centroids of each emotion to their locations on the 2D circumplex model. The circumplex model does not numerically define the emotion locations, so they were arbitrarily chosen as:

- Happy: valence = 1, arousal = 1
- Sad: valence = -1, arousal = -1
- Angry: valence = -1, arousal = 1

After fitting the centroids to their locations, the linear regression model is used to transform all movements into the circumplex space.

4.2.3 Latent feature modification. We use a similar approach to feature modification as prior works (see: Section 3.2). First, the circumplex representations of the data samples are ranked from high to low intensity for both valence and arousal features. For each feature f, the latent space means for the higher and lower halves are calculated as $\vec{\mu}_{f,high}$ and $\vec{\mu}_{f,low}$ Compared to the quartiles used in prior work [24], splitting into halves was empirically found to yield better performance. A feature's attribute vector $\vec{a}_f$ is calculated as the difference between its high and low mean vectors.

$$\vec{a}_f = \vec{\mu}_{f,high} - \vec{\mu}_{f,low}$$

To modify the valence and arousal of a movement, its original latent representation $\vec{m}_0$ is summed with a linear combination of the attribute vectors and the feature weights c_f.

$$\vec{m} = \vec{m}_0 + \sum_{f=\{V,A\}} c_f \vec{a}_f$$

4.2.4 Modification interface. We created an interface for visualizing the circumplex model and modifying the movements (Figure 4). Each point on the scatter plot represents a three-second movement sample projected from the latent space into the circumplex model using the regression parameters described in Section 4.2.2. The emotion classes are color-coded and the projected centroids are marked. In the graph, green is happy (h), blue is sad (s), and red is angry (a). The user selects a movement $\vec{m}_0$ and adjusts the attribute vector weights c_f using the valence (V) and arousal (A) sliders. The projected modified movement $\vec{m}$, denoted by the large X marker, updates in real time. In addition to directly adjusting

Table 1: Network layers and parameters

	Layer	Parameters
Encoder	**Input**	**Movement (30x4)**
	Dropout	10%
	Conv1D+BN	7 filters, kernel size 5
	Leaky ReLU	α = 0.01
	Dropout	5%
	Conv1D+BN	4 filters, kernel size 3
	Leaky ReLU	α = 0.01
	Flatten	–
	Dropout	5%
	KL Resample	–
Decoder	Dense	60
	Upsample1D	2
	BatchNorm	–
	Leaky ReLU	α = 0.01
	Conv1D+BN	4 filters, kernel size 3
	Leaky ReLU	α = 0.01
	Conv1D+BN	6 filters, kernel size 5
	Leaky ReLU	α = 0.01
	Conv1D+BN	6 filters, kernel size 5
	Leaky ReLU	α = 0.01
	Dense	30
	Output	**Movement (30x4)**
Classifier	Dropout	5%
	Dense	13
	Leaky ReLU+BN	α = 0.01
	Dropout	5%
	Dense	3
	SoftMax	–
	Output	**Emotion**

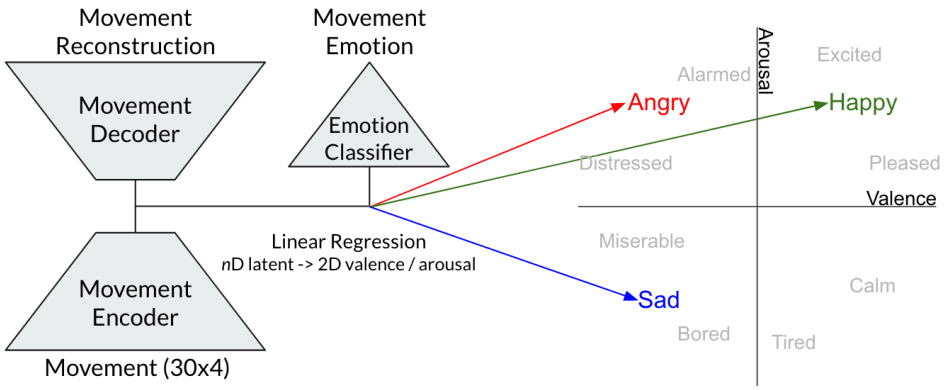

Figure 3: The network architecture consists of a variational autoencoder (left) with an emotion classifier (center). Once the network is sufficiently trained to reconstruct the movements and classify the latent representations by emotion class, linear regression is used to map the nD latent space into the 2D circumplex model (right) with the valence and arousal dimensions.

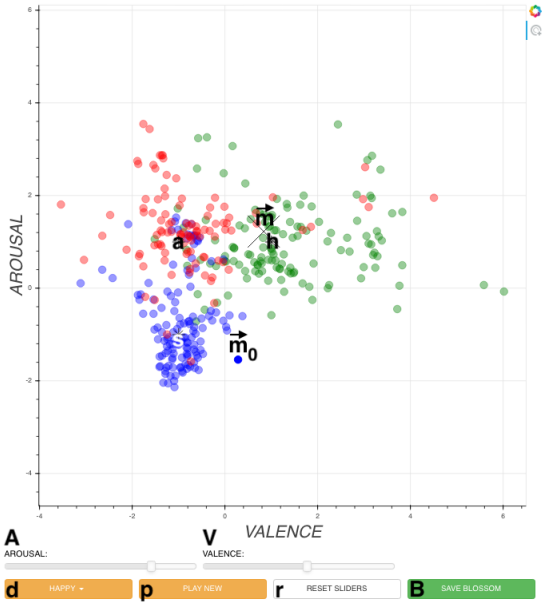

Figure 4: Movement modification interface. The emotions are separated by color and their centroids are marked: (*h*)appy is green, (*s*)ad is blue, and (*a*)ngry is red. The selected movement $\vec{m}_0$ is modified by either adjusting the (V)alence and (A)rousal sliders or by selecting an emotion from the (d)ropdown menu, and $\vec{m}$ denotes the location of the modified movement. In this case, the dropdown menu was used to modify a sad movement to be happy, and the sliders updated accordingly. The (p)lay button plays $\vec{m}$ on Blossom, the (r)eset button resets the sliders, and the (B)lossom button saves $\vec{m}$ to a file for later use.

the feature weight sliders, users can also use a dropdown emotion selector (d) to update the attribute vector weights based on the emotion centroids. The dropdown selector uses the valence-arousal distance between the target emotion's centroid and the original movement $\vec{m}_0$ to indirectly update the sliders and c_f. In Figure 4, a sad movement at [0.4,-1.6] was modified to be happy, whose centroid lies at [1,1]. Selecting "happy" from the dropdown thus sets the valence and arousal sliders to 0.6 and 2.6, respectively, and updates the movement $\vec{m}$ close to the target centroid. Once modified, the VAE decoder generates a three-second gesture in the form of motor trajectories. The interface also includes buttons to play the movements on Blossom (p), save the modified Blossom movement to a file (B), and reset the sliders (r).

5 EVALUATION

We evaluate the performance of the neural network and the modification method using objective metrics for each of our training objective, as well as using an online user survey.

5.1 Network parameters for evaluation

We empirically derived most of the network parameters. The test set hold-out rate was set to 20%. The size of the nD latent space was derived empirically. $n = 40$ was found to be the maximum possible reduction while still achieving the training objectives. For the movement reconstruction objective, using simple mean-squared or mean-absolute error functions resulted in a lack of base motion (yawing) and side-to-side movement (rolling). This may have been due to augmenting the data by mirroring the left-right motions, causing the network to ignore these DoFs and simply default to looking straight ahead. To overcome this issue, we used a custom loss function that weighs each movement DoF differently and uses squared error for the front and base motor and absolute error for the left and right motors. The weights for the front, left and right, and base motors were empirically set to 5, 7, and 20. The KL divergence loss was implemented according to β-VAE [9], and the classifier used categorical cross-entropy as its loss function. During network training, we monitored the following objectives:

- Reconstruction - Monitor loss and plot comparisons of original and reconstructed movements for visual inspection.
- KL - Not monitored, but β-VAE recommends adjusting the weight according to the task [9].

Figure 5: Filmstrips of a happy movement (top) modified into sad (middle) and angry (bottom).

- Classification - Monitor accuracy and plot latent embeddings in TensorFlow Projector to visually inspect emotion class separation in the latent space [31].

We tuned the loss weights iteratively by increasing weights for underperforming objectives, e.g. increasing the reconstruction weight if the movement characteristics are not being preserved or increasing the classification weight if the emotions are being confused. We settled on 5, 0.1, and 7 for the reconstruction, KL, and classification loss weights, respectively. We empirically tuned the remaining training parameters: learning rate of 0.1, batch size of 30, Adam optimizer [14], and mixup with a factor of 0.2 [36]. 100 epochs was sufficient to stabilize the losses.

5.2 Online survey

We evaluated the subjective effectiveness of our method using an online survey, which presented videos of gestures along with a questionnaire for each gesture. The movements shown in the online user survey were chosen by randomly selecting five samples within the held-out test sets of the three emotion classes, resulting in a dataset of 15 original movements. We then modify each movement into the two other emotion classes by using the dropdown interface described above, e.g. a happy sample was modified into both sad and angry, as in Figure 5. This provides two modified movements for each original movement, resulting in a survey dataset of 45 movements, 15 original and 30 reconstructed.

We had two main hypotheses. If the latent representation of a movement is modified to lie in another target emotion space on the circumplex model, then the modified movement's new emotion:

H1) is consistently recognized as the target emotion.
H2) expresses the target emotion as legibly as an original movement with the same emotion.

For each survey question, a video of a movement was followed by Likert scales for how well it represented each emotion class and a multiple choice selection for which emotion it best represented (Figure 6). Each survey showed 30 random movements from the original 45. We distributed the survey using Amazon Mechanical Turk offering $2 compensation and received 100 responses.

Figure 6: Online survey questions.

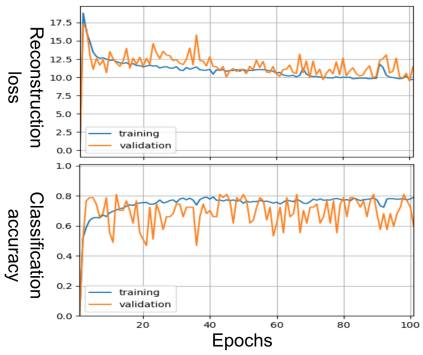

Figure 7: Movement reconstruction loss (top) and emotion classification accuracy (bottom) over 100 epochs.

6 RESULTS

The performance of this approach was evaluated using both objective metrics for the technical implementation and statistical significance tests for the survey results.

6.1 Objective metrics of network performance

We used traditional neural network training metrics to objectively evaluate the technical implementation. The movement reconstruction loss and emotion classification accuracy are the primary training objectives. The KL divergence was weighed lowly as it is comparatively unimportant and primarily provides the Gaussian structure for the latent space.

Figure 7 shows the training curves for the movement reconstruction loss and emotion classification accuracy. Both curves leveled off by the end of training. The validation curves, while noisy, are

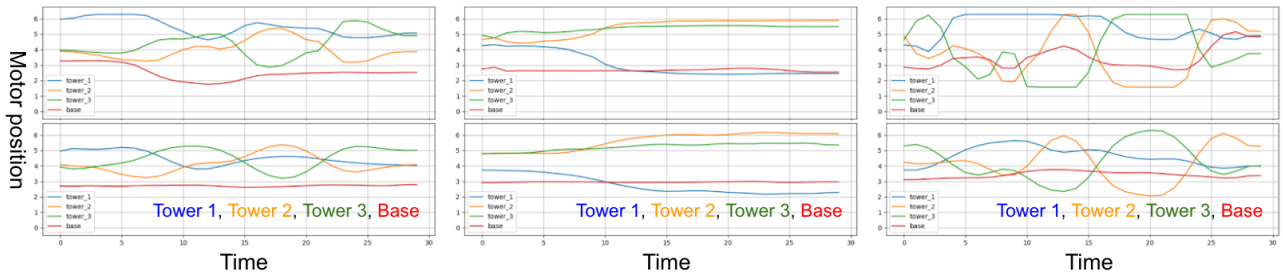

Figure 8: DoF curves for original (top) and reconstructed (bottom) movements for each emotion (happy left, sad center, angry right). The blue, yellow, green, and red lines represent the front, right, left, and base motors, respectively. The reconstructions have difficulty achieving the same exaggeration as the original movements, but retain the overall trajectory characteristics.

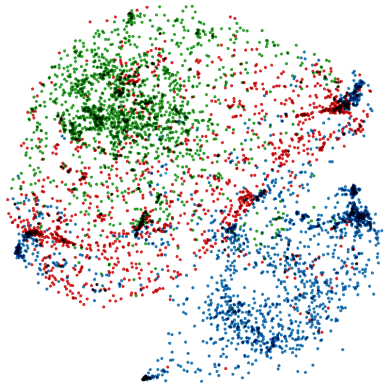

Figure 9: t-SNE representation of all of the movement samples in the latent space. The latent space is visibly separated by emotions (happy is green, sad is blue, angry is red).

very close in performance to the training curves, suggesting that the model did not overfit to the training set. We achieve close to 80% classification accuracy, which is promising considering the abstract nature of the movement data and simplicity of the network.

The reconstruction objective is further evaluated by comparing the original and reconstructed movements. Figure 8 contrasts original (top) and reconstructed (bottom) samples for each emotion class. The reconstructions are less exaggerated, but capture the overall trajectory characteristics of the original movements.

The classification objective is further evaluated with visualization of the latent space. Figure 9 is a dimensionality reduction of all movement samples in the latent space using t-SNE [18]. The emotion regions are visibly separated in this space even before applying the transformation into the circumplex space.

6.1.1 Feature sliders. The performance of the feature sliders can also be objectively measured. A slider would ideally modify a movement along only its intended feature axis (e.g. the valence slider moves a movement sample only along the horizontal valence axis in the interface). However, editing in the latent space may induce coupling in the features, i.e. modifying valence may indirectly modify arousal, and vice-versa. This coupling was also present in prior work [17], where adding mustaches also added masculine features due to these features being highly correlated in the input dataset.

The degree of feature coupling is highly dependent upon the emotion class and specific movement sample. To test this, each slider was maximized individually and the unit difference vector $\hat{m}_\Delta$ from the original $\vec{m}_0$ to modified $\vec{m}$ movement was calculated.

$$\hat{m}_\Delta = \frac{\vec{m} - \vec{m}_0}{|\vec{m} - \vec{m}_0|}$$

The dot product between $\hat{m}_\Delta$ and the unit feature vector (<1,0> for valence, <0,1> for arousal) denotes the alignment of the modification direction and the intended axis, with a dot product of 1 denoting perfect alignment. This was calculated for every movement in the held-out test set, and the mean dot products for all emotion-feature combinations are presented in Table 2. All of the results are almost 1, indicating that both sliders move primarily in their respective axes and perform as intended.

Table 2: Slider evaluation results.

		Feature	
		Valence	Arousal
Emotion	Happy	0.999	0.996
	Sad	0.995	0.989
	Angry	0.995	0.992

6.1.2 Dropdown. The performance of the dropdown menu for modifying a movement towards a target emotion can also be objectively measured. The dropdown emotion selector indirectly adjusts the sliders by setting the valence-arousal distance from the movement to the target emotion's centroid as the slider values. As visualized on Figure 4, the effectiveness of this method can be calculated by measuring the distance between the final modified movement $\vec{m}$ and the target emotion centroid (h in this example), with a distance of 0 denoting ideal performance. This distance was calculated for every movement in the held-out test set, and the mean distances for each original-target emotion combination are presented in Table 3.

Modifying a movement towards its original emotion yields the best performance. For cross-emotion modification, sad→happy performs the best, followed by angry→happy and happy→sad. Interestingly, happy and sad both have difficulty modifying into angry.

6.2 Survey

In addition to the above, we analyzed the subjective metrics collected in the survey in light of the hypotheses laid out above.

Table 3: Dropdown evaluation results. Bolded values indicate best performance for each original emotion class. Italicized values indicate second-best performance.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	**0.126**	*0.353*	0.507
	Sad	*0.237*	**0.098**	0.328
	Angry	*0.317*	0.405	**0.193**

6.2.1 H1. For the first hypothesis, there should be no difference in the recognition accuracy for the target emotions between the original and modified movements. For example, movements modified to be happy should be recognized as happy with the same accuracy as original happy movements. TOST (two one-sided tests) equivalence tests were performed between the original and modified movements for each target emotion. Given the range of the accuracies (0 for wrong, 1 for correct), the equivalence test α was set to 0.1. The results (Table 4) show that **H1** is supported ($p < 0.05$) for happy→sad and sad→angry and implied ($p < 0.1$) for angry→sad, but is not supported for the other modifications.

Table 4: Mean emotion recognition accuracies and equivalence test p-values (italicized). Bolded p-values support H1.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	0.59, ——	0.63, ***0.03***	0.18, *0.13*
	Sad	0.44, *0.91*	0.66, ——	0.21, ***0.01***
	Angry	0.44, *0.91*	0.61, *0.08*	0.24, ——

6.2.2 H2. For the second hypothesis, there should be no difference in the legibility scores for the target emotions between the original and modified movements. For example, the legibility scores for movements modified to be happy should not be significantly different than the scores for original happy movements. The legibility score is the Likert score for the target emotion, e.g. the legibility score for a sad movement modified to be happy would be the Likert score for the happy slider in Figure 6. Equivalence tests between the original and modified movements for each target emotion were performed. Given the range of the Likert scores (1-5), the equivalence test α was set to 0.2. The results (Table 5) show that **H2** is supported ($p < 0.05$) for all modifications.

Table 5: Mean emotion legibility scores and equivalence test p-values (italicized). Bolded p-values support H2.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	3.33, ——	3.42, ***0.02***	2.07, ***0.02***
	Sad	2.77, ***0.02***	3.27, ——	2.27, ***0.02***
	Angry	2.81, ***0.02***	3.54, ***0.02***	2.26, ——

7 DISCUSSION

The objective results show that the network achieves the learning goals well: the reconstructions look similar to the original movements but with less exaggeration. This is in line with challenges reported for generative networks in other domains [37]. It was particularly interesting that the network and regression model were able to map the movements into the circumplex space with minimal domain knowledge apart from the emotion centroid locations. Qualitatively, valence corresponds to looking upwards or downwards while arousal corresponds to exaggeration.

The subjective results show that **H2** is supported for all modifications, but **H1** is only partially supported. Using the dropdown menu for automatic modification may have been a limiting factor, as evidenced by its mediocre cross-emotion performance (Table 3). Manually moving the sliders while monitoring the output for fine-tuning may have yielded better modified samples.

The subjective results also imply that not all emotions are equally conveyable. Anger is consistently recognized below the chance level of 0.33, implying that Blossom may have difficulty conveying anger compared to happy and sad. When creating the movements, anger was the most ambiguous while sad movements were usually a slow lowering of the head. This shows the relationship between a robot's expressive capabilities and its embodiment, which renders certain emotions harder to convey. Movements modified to be happy scored considerably lower in terms of both accuracy and legibility than their original counterparts. This implies that this modification method may not be able to retain some qualities of hand-crafted movements. These observations highlight the difficulty in quantifying emotions, especially with the simple circumplex model.

Plans for future work include a usability study for the interface and testing with other robots to evaluate generalizability. The study would assess the ease-of-use of the interface and address the issues with the automated modification. We would also test the method using robots with more complex modalities such as sounds or face gestures. These modalities may better convey emotions that are difficult to express through movements alone. Additionally, we could choose to imbue affect into non-emotive or task-oriented gestures such as hesitating or signaling. We also want to explore using other starting points in the latent space, such as neutral movements or random samples, to generate new gestures.

8 CONCLUSION

We presented a method for modifying affective movements for an expressive robot using neural networks. Using a dataset of hand-crafted movements, we trained a classifying VAE to learn a latent space to compactly represent the movements and classify them by their intended emotions. We then used linear regression to map the abstract latent space into the comprehensible valence and arousal dimensions on the circumplex emotion model. Applying simple arithmetic in the latent space enables us to modify the valence and arousal of the movements. We evaluated this approach with objective and subjective metrics which showed that the method performs well along learning objectives and to some extent supported the hypotheses that the modified movements are comparable to the originals in terms of recognizability and legibility. Compared to heuristic approaches for creating movements, we used little domain knowledge of kinematics and robotics. This suggests that using neural networks for generating robot behaviors is more generalizable and accessible, enabling faster and easier methods for expanding a robot's behavior library for prolonged interaction.

REFERENCES

[1] Aris Alissandrakis, Chrystopher L Nehaniv, and Kerstin Dautenhahn. 2007. Correspondence Mapping Induced State and Action Metrics for Robotic Imitation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37, 2 (April 2007), 299–307. https://doi.org/10.1109/TSMCB.2006.886947

[2] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57, 5 (2009), 469–483. https://doi.org/10.1016/j.robot.2008.10.024

[3] François Chollet. 2015. Keras. https://github.com/fchollet/keras.

[4] Ruta Desai, Fraser Anderson, Justin Matejka, Stelian Coros, James McCann, George Fitzmaurice, and Tovi Grossman. 2019. Geppetto: Enabling Semantic Design of Expressive Robot Behaviors. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 369, 14 pages. https://doi.org/10.1145/3290605.3300599

[5] Dongkeon Lee, Kyo-Joong Oh, and Ho-Jin Choi. 2017. The chatbot feels you - a counseling service using emotional response generation. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 437–440. https://doi.org/10.1109/BIGCOMP.2017.7881752

[6] Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion* 6, 3-4 (1992), 169–200. https://doi.org/10.1080/02699939208411068

[7] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C Schultz, et al. 2005. Designing robots for long-term social interaction. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1338–1343. https://doi.org/10.1109/IROS.2005.1545303

[8] Minoru Hashimoto, Hiromi Kondo, and Yukimasa Tamatsu. 2008. Effect of emotional expression to gaze guidance using a face robot. In *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*. 95–100. https://doi.org/10.1109/ROMAN.2008.4600649

[9] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR* 2, 5 (2017), 6.

[10] Geoffrey E. Hinton and Richard S. Zemel. 1993. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'93)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3–10. http://dl.acm.org/citation.cfm?id=2987189.2987190

[11] Guy Hoffman and Wendy Ju. 2014. Designing Robots with Movement in Mind. *J. Hum.-Robot Interact.* 3, 1 (Feb. 2014), 91–122. https://doi.org/10.5898/JHRI.3.1.Hoffman

[12] Samira Ebrahimi Kahou, Christopher Pal, Xavier Bouthillier, Pierre Froumenty, Çaglar Gülçehre, Roland Memisevic, Pascal Vincent, Aaron Courville, Yoshua Bengio, Raul Chandias Ferrari, Mehdi Mirza, Sébastien Jean, Pierre-Luc Carrier, Yann Dauphin, Nicolas Boulanger-Lewandowski, Abhishek Aggarwal, Jeremie Zumer, Pascal Lamblin, Jean-Philippe Raymond, Guillaume Desjardins, Razvan Pascanu, David Warde-Farley, Atousa Torabi, Arjun Sharma, Emmanuel Bengio, Myriam Côté, Kishore Reddy Konda, and Zhenzhou Wu. 2013. Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction (ICMI '13)*. ACM, New York, NY, USA, 543–550. https://doi.org/10.1145/2522848.2531745

[13] Michelle Karg, Ali-Akbar Samadani, Rob Gorbet, Kolja Kühnlenz, Jesse Hoey, and Dana Kulić. 2013. Body Movements for Affective Expression: A Survey of Automatic Recognition and Generation. *IEEE Transactions on Affective Computing* 4, 4 (Oct 2013), 341–359. https://doi.org/10.1109/T-AFFC.2013.29

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:cs.LG/1412.6980

[15] Andrea Kleinsmith and Nadia Bianchi-Berthouze. 2013. Affective Body Expression Perception and Recognition: A Survey. *IEEE Transactions on Affective Computing* 4, 1 (Jan 2013), 15–33. https://doi.org/10.1109/T-AFFC.2012.16

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[17] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. arXiv:cs.LG/1512.09300

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[19] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association*.

[20] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. 2015. Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15)*. ACM, New York, NY, USA, 443–449. https://doi.org/10.1145/2818346.2830593

[21] Mannes Poel, Dirk Heylen, Anton Nijholt, M Meulemans, and A Van Breemen. 2009. Gaze behaviour, believability, likability and the iCat. *AI & SOCIETY* 24, 1 (01 Aug 2009), 61–73. https://doi.org/10.1007/s00146-009-0198-1

[22] Jonathan Posner, James A. Russell, and Bradley S. Peterson. 2005. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology* 17, 3 (2005), 715–734. https://doi.org/10.1017/S0954579405050340

[23] Helge Rhodin, James Tompkin, Kwang In Kim, Kiran Varanasi, Hans-Peter Seidel, and Christian Theobalt. 2014. Interactive motion mapping for real-time character control. *Computer Graphics Forum* 33, 2 (2014), 273–282. https://doi.org/10.1111/cgf.12325 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12325

[24] Adam Roberts, Jesse H. Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. *CoRR* abs/1803.05428 (2018). arXiv:1803.05428 http://arxiv.org/abs/1803.05428

[25] Igor Rodriguez, José María Martínez-Otzeta, Itziar Irigoien, and Elena Lazkano. 2019. Spontaneous talking gestures using Generative Adversarial Networks. *Robotics and Autonomous Systems* 114 (2019), 57–65. https://doi.org/10.1016/j.robot.2018.11.024

[26] Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (1958), 386–408.

[27] Martin Saerbeck and Christoph Bartneck. 2010. Perception of Affect Elicited by Robot Motion. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction (HRI '10)*. IEEE Press, Piscataway, NJ, USA, 53–60. http://dl.acm.org/citation.cfm?id=1734454.1734473

[28] Tamie Salter, Kerstin Dautenhahn, and R Bockhorst. 2004. Robots moving out of the laboratory - detecting interaction levels and human contact in noisy school environments. In *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*. 563–568. https://doi.org/10.1109/ROMAN.2004.1374822

[29] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[30] Yeongho Seol, Carol O'Sullivan, and Jehee Lee. 2013. Creature Features: Online Motion Puppetry for Non-human Characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '13)*. ACM, New York, NY, USA, 213–221. https://doi.org/10.1145/2485895.2485903

[31] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. ViÃľgas, and Martin Wattenberg. 2016. Embedding Projector: Interactive Visualization and Interpretation of Embeddings. arXiv:stat.ML/1611.05469

[32] Michael Suguitan and Guy Hoffman. 2019. Blossom: A Handcrafted Open-Source Robot. *ACM Trans. Hum.-Robot Interact.* 8, 1, Article 2 (March 2019), 27 pages. https://doi.org/10.1145/3310356

[33] Joshua M Susskind, Geoffrey E Hinton, Javier R Movellan, and Adam K Anderson. 2008. Generating facial expressions with deep belief nets. In *Affective Computing*. InTech.

[34] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499* (2016).

[35] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. 2010. Animating Non-humanoid Characters with Human Motion Data. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, Goslar Germany, Germany, 169–178. http://dl.acm.org/citation.cfm?id=1921427.1921453

[36] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2017. mixup: Beyond Empirical Risk Minimization. arXiv:cs.LG/1710.09412

[37] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658* (2017).

[38] Allan Zhou and Anca D Dragan. 2018. Cost Functions for Robot Motion Style. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3632–3639. https://doi.org/10.1109/IROS.2018.8594433