Affordance-based Generation of Pretend Object Interaction Variants For Human-Computer Improvisational Theater

Mikhail Jacob and **Prabhav Chawla** and **Lauren Douglas** and **Ziming He** and

Jason Lee and **Tanuja Sawant** and **Brian Magerko**
Georgia Institute of Technology
Atlanta, GA 30308 USA
{mikhail.jacob, pc_1998, ldouglas7, zhe66, jlee3331, tanuja.sawant, magerko}@gatech.edu

Abstract

This paper describes DeepIMAGINATION, a neural architecture to generate variants of movement-based object interactions with props using the physical attributes of props while playing the Props game. The agent can generate these action variants while searching a learned action space in real-time to provide improvised responses to its human partner. Convolutional and recurrent variants of CVAEs are used for experimentation. The paper presents an evaluation of the architecture by benchmarking its ability to learn the human data set and generate believable, recognizable, and high-quality action variants from it. Results showed that the agent could generate believable, high-quality action variants. but that recognizability requires improvement.

Introduction

Human-agent improvisation is a challenging subset of human-computer co-creativity (mixed-initiative creativity) that requires improvisational agents to generate creative acts in near real-time within open-ended scenarios. The constraints of the task enforce severe temporal constraints on the agent while requiring the agent to possess a large amount of knowledge and reason about large action spaces without well-specified pre-defined goals at any given time. Constrained improvisational agents have been demonstrated in domains such as musical improvisation (Hoffman and Weinberg 2010), visual art (Davis et al. 2016), theater (Mathewson and Mirowski 2017), and dance (Reidsma et al. 2006); however, there are as yet few systems that focus on truly open-ended improvisation.

The authors' prior work investigated highly-constrained improvisation within theater (O'Neill et al. 2011) and pretend play (Magerko et al. 2014). Two problems became apparent from this initial work. Agents required a large amount of knowledge to be authored before they could respond meaningfully to a person's comparatively vast experiences and knowledge, also known as the knowledge-authoring bottleneck (Spierling and Szilas 2009). Previous work addressed this issue in the LuminAI installation (Jacob and Magerko 2015). It was also challenging for improvisational agents to perform meaningful real-time action selection from open-ended action spaces with ill-defined goals using learned embodied knowledge. This is the *improvisational action selection problem* that motivates this research.

Figure 1: Two actors playing the Props game from the popular TV show, "Whose Line Is It Anyway?".

A high-level solution to the improvisational action selection problem was proposed in Jacob and Magerko (2018) as "creative arc" negotiation during the improvisation as intrinsic motivation for the agent's decision-making, inspired by various aesthetic arcs across several artistic media. A *creative arc* is defined conveniently (but reductively) as a continuous trajectory through a multidimensional *creative space*, currently consisting of *novelty*, *unexpectedness*, and *quality* dimensions. Perceived or generated actions are evaluated computationally and localized to points in the creative space during the performance.

Creative arc negotiation within gestural and object-based movement improvisation was applied to the performance of an improv theater game called *Props*. Improvisers playing the Props game pretend that a given abstract prop is some real-world or fictional object and take turns to use that prop to enact imaginative mimed actions. A virtual reality (VR) installation called the *Robot Improv Circus* was created as a test bed and technical probe to study human-agent improvisation within the Props game domain.

The CARNIVAL architecture (see (Jacob and Magerko 2018) for details) enables improvisational agents to negotiate creative arcs with people. It uses interruptible search over a learned action space in order to choose the closest action to a target point on the creative arc during its turn.

CARNIVAL consists of a parameterizable action generator to perform heuristic search over the action space, improvisational reasoning strategies for guiding search, and creativity evaluation models to localize actions in the creative space.

The parameterizable action generator is a fundamental module used to search the agent's action space for candidate actions that are evaluated by the other parts of the system. Additionally, the generator's representation of the action space constrains the types and implementations of improvisational reasoning strategies for guiding search. Thus the design and evaluation of the generator are vital to the computational creativity problem of human-agent improvisational action selection studied in CARNIVAL.

The action generator for the Robot Improv Circus installation was designed to answer the following research question. ***What representations and processes enable an agent to search a learned object-based interaction space in order to generate believable, recognizable, and high-quality pretend action variants with similar abstract props?*** In order to investigate this question, a deep neural architecture was implemented and evaluated on its capacity for generating believable, recognizable, and high-quality variants of object interactions. This architecture and its variants were trained on mimed human-object actions with props in VR.

A novel feature vector representation of object physical attributes (adapted from (Varadarajan and Vincze 2012)) as an aggregation of part attributes was developed and used to learn a mapping between the physical attributes of objects and a data set of mimed human actions using those objects. Conditioning the mapping this way, enabled the application of learned actions to other physically similar objects. Additionally, the mapping and generator together, form a model of affordance-based action generation since the architecture uses it to constrain generation to actions that are physically suitable or afforded by an object (Norman 1988).

Related Work

Gestural creativity has been modeled in several disciplines, such as choreography synthesis, robotics, and embodied conversational agents. Gesture synthesis systems try to create parameterized, natural, and expressive gestures by following a similar pipeline: input to gesture planner, selection by statistical model, and modification by final component (Ng-Thow-Hing, Luo, and Okita 2010).

Generative choreography systems such as Ikeuchi (2008) and Ofli et al. (2012) used segmented music measures as a conditioning input to their generative choreography systems. The most statistically likely candidate dance segments from a pre-authored database were then chosen based on the music inputs and combined to create smooth transitions. Embodied conversational agents create gestures from speech, text, or video clips. Mancini and Castellano (2007) used video tracking and analysis to create an agent capable of mimicking detected expressivity. Kipp et al. (2007) focused on creating natural gestures in virtual agents by using g-units to create continuous flowing movements from gesture segments. Previous models of gestural creativity have been successful in mimicking tasks, but because of the open-ended

Figure 2: A view of the virtual agent miming an action using a prop in the Robot Improv Circus VR installation

action space, a deep generative model was chosen instead of a traditional statistical model.

Gesture synthesis has made significant advances through deep generative models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). They have proven to be particularly useful for generating novel gestures and choreography with minimal feature engineering by hand. Augello et al. (2017) employed a vanilla VAE trained on a data set of human dance movements to generate robot dance movements. Similar work by Kiasari, Moirangthem, and Lee (2018) focused on combining VAEs and GANs to produce sequences of stylized actions. Their model utilized latent variables from the autoencoder as input to the GAN's discriminator network, while the input to the GAN's generator network was conditioned using action labels and initial poses of the generated action sequences. Our architecture also seeks to control the mode of the generated data through conditioning but adds conditioning both at input and latent space sampling stages since we draw inference directly from the latent space.

Recurrent Neural Networks (RNNs), notably Long Short-Term Memory (LSTM) networks, have been more commonly used for sequential motion generation. Researchers have exploited the hidden Markov model process underlying motion and choreography by using RNN models that combine distributed hidden states and non-linear dynamics. The results are evident in choreographic support (Crnkovic-Friis and Crnkovic-Friis 2016; Tang, Jia, and Mao 2018) and motion synthesis (Holden, Saito, and Komura 2016; Habibie et al. 2017). Our approach extends previous work by conditioning RNN-based generative models for gesture synthesis and preserving local/regional coherence by grouping multiple poses within temporal proximity.

Robot Improv Circus

The Robot Improv Circus is a VR installation for people to play the Props game with a virtual agent. The experience takes place on the stage of a robot circus, where improv is the main event. Participants take turns with the virtual agent to mime pretend actions using abstract props as a real-world or fictional object in imaginative ways in order to create a proto-narrative with the agent.

The VR experience consists of a trial round followed by

a small number of game rounds. Each performer is given a new prop every turn and each round consists of 5-7 turns. The goal of the game is to create a proto-narrative by taking turns miming actions with the prop. Performers hit a buzzer after enacting their actions to signal the end of their turn.

As an example, after receiving a prop shaped like a cube, the VR user might pretend that the prop is a hat and mime putting it on. She then hits the buzzer to end her turn. A new prop, shaped like a long flattened cone, appears in front of the agent who pretends to comb its hair using it as a comb. The agent speaks and displays a speech bubble that reads, "I am combing with a comb" (like in fig. 2). The speech and speech bubbles were added to encourage dialogue and increase recognizability of the mimed actions.

The Robot Improv Circus is exhibited in a circus tent. The installation has two large displays that act as portals for a real-world audience to view the virtual circus stage. They can watch, applaud, and provide positive feedback to participants in VR, visible as floating emoji above the robot audience in VR.

DeepIMAGINATION

DeepIMAGINATION (Deep IMprovised Action Generation through INteractive Affordance-based exploraTION) is responsible for generating candidate actions for consideration elsewhere in the CARNIVAL architecture. The module represents and reasons about props using their physical attributes and a learned model of object feature vectors allowing learned actions to be generalized to props with similar componential physical attributes that it may not have seen before. The search through the agent's action space is implemented as strategy-guided sampling from the latent space of a conditional variational autoencoder (CVAE) (Sohn, Lee, and Yan 2015) conditioned on the physical attributes of the props making use of the properties of the VAE latent space. The followings sections describe the representations and CVAE architectural variants explored.

Physical Attributes Feature Vector Representation

The physical attributes of a given prop are represented as a fixed-length feature vector. The encoded value is obtained by decomposing the prop into a set of parts that each correspond to a shape primitive with (optional) deformations applied to it. These individual parts are then coded/parsed to obtain a set of binary physical attributes features.

The physical attributes feature set represents the part's shape primitive, size, thickness, flatness, concavity, taper, rigidity, curvature, hole size, and whether a digit/symbol is signified. The feature set was chosen by extending from affordance representation ontologies such as (Varadarajan and Vincze 2012). The physical attributes feature values for each part are then aggregated for the entire prop by summing them together and normalizing them using the maximum count for any feature in the data set. The encoded value represents the normalized counts of each physical attribute feature for the prop across all parts. For example, a barbell-shaped prop might be two flattened spheres connected by a long, thin cylinder. The encoding is currently done by hand given the small number of props and focus of the research.

Gesture Feature Vector Representation

The conditional variational autoencoder (CVAE) models were trained on almost 900 mimed actions of length from 3.3 to 10 seconds collected from five novice improvisers pretending the ambiguously shaped props to be real-world objects (e.g., ladles, golf clubs, and swords) within a VR data collection environment. Each training data point was represented as a single vector with sequential body poses concatenated together and zero-padded as necessary. Models were trained using either a 27000-dimensional or a 16000-dimensional vector representation. The 27000-dimensional vector used 30 features per frame, recorded at 90 FPS for 10 seconds. The 16000-dimensional vector used 35 features per frame at 45 FPS for 10 seconds with 250 entries of zero padding. Each pose consisted of normalized location data (position and rotation) for the user's head and hands in the VR system (and the character's pelvis, calculated using inverse kinematics). The 27000-dimensional representation also had two flags for the VR controllers' grab object button states. The 16000-dimensional representation directly included normalized location data for the prop instead.

Conditional Variational Autoencoder (CVAE) Architecture

The encoder and decoder were both conditioned on the physical attribute vectors of the props used to perform the actions using input concatenation. The encoder reduces the high-dimensional input into a low-dimensional latent space, and the decoder reconstructs a sampled latent vector back into the input space. Fig. 3 depicts how this was done using 1-dimensional convolutional layers and 1-dimensional transposed convolutional layers in the encoder and decoder, respectively. Dropout layers were used for regularization. A recurrent CVAE variant is described in a later subsection.

The network was implemented in TensorFlow and trained with the ADAM optimizer (Kingma and Ba 2014). Given an input distribution X, a latent distribution z and a conditioning distribution c, the CVAE loss function is defined as:

$$L(X, z, c) = E[\log P(X|z, c)] + D_{KL}[Q(z|X, c) \;||\; P(z|c)] \quad (1)$$

In other words, the loss function is the sum of the decoder's reconstruction loss and the encoder's Kullback-Leibler divergence loss, both conditioned on the physical attributes distribution. Training the network is made possible by using the re-parameterization trick (Kingma and Welling 2013):

$$z = \mu(X, c) + \epsilon \Sigma^{\frac{1}{2}}(X, c), \text{ where } \epsilon \sim \mathcal{N}(0, 1) \quad (2)$$

During generation, the model's latent space is repeatedly sampled at specific locations provided by the CARNIVAL architecture's improvisational response strategies, based on the current improvisational context occurring. The DeepIMAGINATION module generates candidate actions conditioned on the physical attributes of the given prop. Candidate actions are evaluated by the creativity evaluation models of the CARNIVAL architecture, and the candidate action that is closest to the next target point on the agent's creative

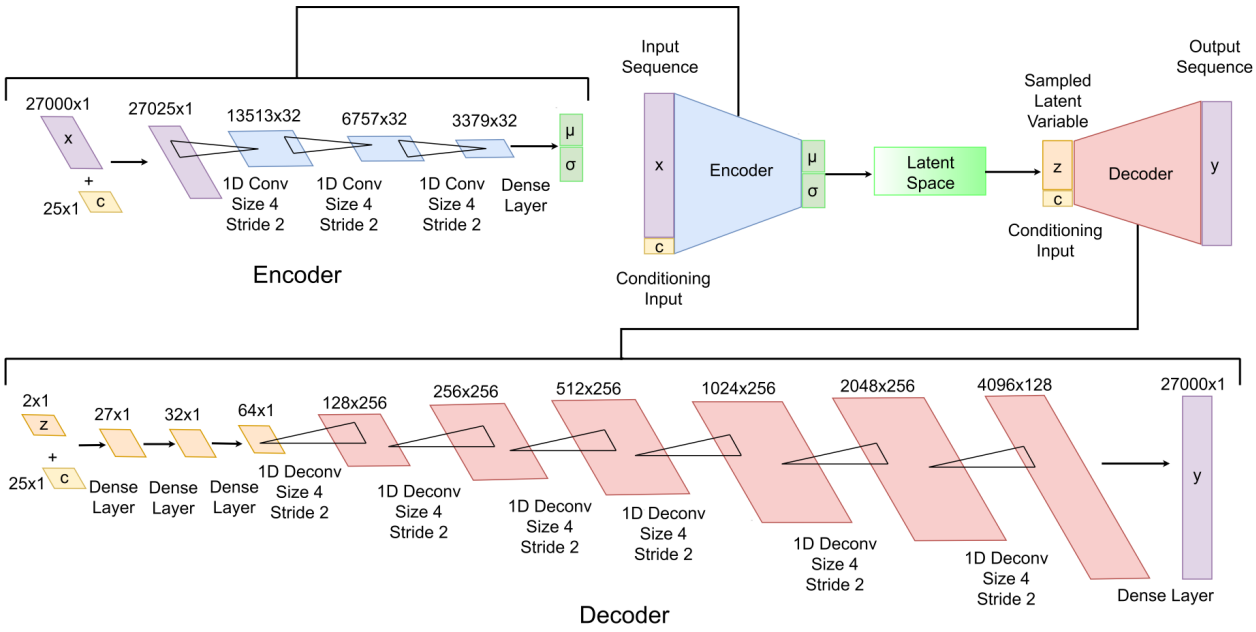

Figure 3: A convolutional variant of the DeepIMAGINATION architecture with $(27000, 1)$ shaped input gesture and 2D latent space. General CVAE architecture shown in upper right quadrant. Zoomed-in views of encoder and decoder in upper left and bottom respectively. Dropout layers not shown but applied between every 1D convolution and transposed convolution layer.

arc for its current turn is selected. The generated output is then used within the Robot Improv Circus VR experience to control the rigged character model of a robot avatar using inverse kinematics (to control the character's other joints). Exponential moving average smoothing of joint trajectories is applied due to the IK-induced shakiness of the actions.

A total of eight architecture variants were designed and trained, including four convolutional models and four recurrent models. The variants were implemented for performance evaluation and selection. The convolutional architectures only differed in their input vector representations. The four recurrent models used either a standard RNN architecture or an architecture based on the MusicVAE network (Roberts et al. 2018). The two groups of RNN variants were also trained on different input vector representations.

Convolutional Variants It is helpful to think of the different input vector representations ($(27000, 1)$, $(16000, 1)$, $(900, 30)$, and $(450, 35)$) for convolutional models in terms of the number of channels in the input data. The data was first represented with one channel, that is, 27000 and 16000 dimensional vectors were reshaped to $(27000, 1)$ and $(16000, 1)$ dimensional tensors, respectively. In another representation, the number of channels corresponded to the number of features per body pose frame - i.e., 27000 dimensional vectors were reshaped to $(900, 30)$ tensors while the 16000 dimensional vectors were reshaped to $(450, 35)$ tensors (disregarding the zero padding).

Recurrent Variants The Recurrent Neural Network versions of CVAE were implemented using long short-term (LSTM) layers. Both the encoders and decoders of the Vanilla RNN implementation include single layers of bidirectional LSTMs that represented information for each frame concatenated with the physical attributes vector. Based on results from Roberts et al., where vanilla RNN-based decoders sometimes had poor sampling and reconstruction performance, a hierarchical RNN architecture for the decoder was designed based on their MusicVAE architecture. In this variant, the latent vector z is first passed through a fully connected layer to initialize the state of the Conductor layer, which is composed of a unidirectional LSTM layer. The output of the conductor layer is then passed as initialization for the bottom LSTM layers, where each frame vector from Conductor layer, concatenated with the output of previous bottom layer LSTM, is used as initialization for the bottom layer LSTM of next time interval. The outputs of each bottom layer LSTM are then concatenated and flattened to match the input tensor shape.

Methodology

The DeepIMAGINATION module is the parameterizable action generator for searching the agent's action space in the CARNIVAL architecture. It was designed to investigate, "what representations and processes enable an agent to search a learned object-based interaction space in order to generate believable, recognizable, and high-quality pretend action variants with similar abstract props?" Therefore, our evaluation plan for the DeepIMAGINATION module consisted of the following questions.

EQ1 Which CVAE variant best learned the distribution of human-object interactions?

EQ2 Does the architecture allow an agent to generate action variants that are believable, recognizable, and high-quality compared to human actions?

Benchmarking Error

Standard quantitative evaluation metrics were chosen to benchmark the eight architecture variants and evaluate EQ1. These metrics were Euclidean distance, root mean squared error, and cosine similarity, as well as the final epoch mean loss. Each architecture was evaluated using a model trained over 40000 iterations (approximately 2850 epochs with a batch size of 64) using a 10% validation split. The trained model was then fed the entire data set, calculating metrics between each input vector and the reconstructed vector created by the CVAE. The mean, median, and standard deviation of each metric along with training and validation loss from the final epoch of training are reported in the next subsection (see Table 1). For qualitative comparisons, visual inspection within our Unity3D environment was used to compare the different generated actions.

Mechanical Turk Study

We created multiple surveys using Amazons Mechanical Turk platform that assessed the believability, quality, and recognizability of four data sets related to actions from DeepIMAGINATION. The experiment was conducted to address the evaluation question EQ2 described above. Each of the four data sets consisted of 40 gestures performed by a robot character in VR across 20 props from the Robot Improv Circus. A GIF was recorded of the robot character performing two actions with each prop for a total of 160 actions across all four datasets. These GIFs were then evaluated by remote workers on the Mechanical Turk platform.

Data Sets The human-generated data set comprised actions performed by a human in VR with a robot avatar. This set of human gestures was then passed through DeepIMAGINATION in various conditions to generate three additional data sets of actions with the same robot avatar. The direct output of the autoencoding made up the agent mimicry data set as it represented the agent's interpretation of human gestures. The third and fourth data sets were made up of near and far action variants (respectively) of the agent mimicry data set. They were generated by sampling points that were nearby and further away (respectively) from the mimicry gestures in the CVAE model's latent space. The same robot avatar performed these actions as well.

Each survey required the participant to watch either one or two recorded GIFs of actions (depending on the task) from one of the four datasets and answer a few questions about that GIF. In each survey, the human data set made up the human actions, and the other three data sets made up the computer generated actions. There were 80 participants for tasks with single GIFs and 60 participants for tasks with two GIF comparisons. Each participant worked on 20 GIFs out of the entire data set of GIFs.

Believability In order to assess the believability of the actions, two survey tasks were given to Mechanical Turk workers. In the first survey, each participant watched a single GIF at a time and answered if they believed the action was performed by a human in VR or generated by a computer program. The comparison was made in order to evaluate whether participants could tell the difference between computer-generated (CG) actions and human actions between each data set. The hypothesis was that differences would be seen between the discrimination accuracy of generated actions according to which of the three CG data sets was being evaluated (indicating that at least some groups of CG actions were as believable as human actions).

A second study was run that asked people to compare a human action from the human actions data set with a CG action from one of the other three datasets and asked the participant to identify which action they believed was generated by a computer, if both were generated by a computer, or neither was generated by a computer. The test helped to clarify whether participants thought that computer-generated actions were human actions when directly comparing the two. The test also indicated how believable the CG GIFs were. If the participants had low accuracy in determining the identity of the computer-generated GIF, it would indicate that the computer-generated GIFs were believable. The hypothesis was that there would be significant differences in recognition accuracy across groups, indicating that the CG actions were mistaken for human actions in some of the groups.

Recognizability The recognizability of the actions in our data sets was assessed in terms of how identifiable the pretend object and pretend action were that the character in the GIF was portraying. The survey asked participants to select what they believed the robot character was most likely enacting from a list of three options. The options were similar to stabbing with a sword or eating with a spoon. High accuracy in identifying the actions and objects shown in the GIF would indicate that the portrayal was recognizable overall. Our hypothesis was that comparable recognition accuracy across groups would be seen showing that the CG action sets were equally recognizable to human actions.

Quality Participants were asked to determine the quality of the GIFs through two tasks. In the first one, participants were asked to rate the smoothness and quality on a 5-point Likert scale by looking at a GIF and evaluating it on its own. They were also asked to state what they thought were criteria for quality in this domain before rating any GIFs and were asked to use those criteria strictly during rating.

The second task was a forced choice condition. Participants were asked which action they thought was smoother and of higher quality. Each participant was asked to define quality at the beginning of the survey and use those criteria strictly while rating the GIFs for quality.

The two measures (smoothness and user-defined quality) were used together to assess the overall quality of each action in both tasks. If smoothness and user-defined quality were high for each action, it would indicate that the overall quality was high. Our hypothesis was that there would be comparable quality and smoothness ratings across groups.

Results

Benchmarking Error

The results of the standard evaluation metrics that were used on the eight architecture variants are shown in Table 1 and

the final epoch mean losses are shown in Table 2. Comparing the Conv. $(16\text{K}, 1)$ architecture with the Conv. $(27\text{K}, 1)$ architecture, the former performs better in all metrics except for training loss. However, Conv. $(16\text{K}, 1)$ model's validation loss is lower, which provides stronger evidence that the $(16000, 1)$ representation allows for a better reconstruction. Both of the reshaped convolutional feature vector representations outperformed their respective un-reshaped versions. The result shows more support that the reshaping helped the CVAE learn more beneficial relationships when the per pose features were split up across channels.

The RNN-based models seem to generalize better than the convolutional variants, as the Vanilla 16K model performs the best in validation loss(shown in Table 2). In contrast to RNN variants, convolutions based models overfit drastically on the training set. The variance in losses between 27K and 16K RNN based models in Table 1 and 2 shows that RNN-based models are resilient to reductions in input dimensions.

Mechanical Turk Study

Believability The task of detecting whether a given GIF was human performed or CG was treated as a binary classification task between the performance of the participants on the human data set in comparison to their performance on each of the other three data sets. The lower the participant accuracy, the stronger would be the evidence that the CG actions were believable. In order to analyze the participant responses, a confusion matrix was created for the four sets of comparisons: human vs. all CG, human vs. agent mimicry, human vs. near variant, and human vs. far variant. The F1 scores for the four conditions were: 0.5251, 0.7154, 0.7163, and 0.671. Additionally, the Matthews Correlation Coefficients for the four conditions were: 0.3308, 0.4237, 0.426, and 0.2912, respectively (all weak positive correlations).

The results above showed that the believability of the CG actions was comparable to that of the human actions in the single GIF rating task when human vs. all CG or human vs. far variant conditions were considered. The fact that far variants scored the highest in comparison to agent mimicry and near variants was surprising since it was the least close to the corresponding human point in the latent space. However, it possible that it was close to some other human point and thus ended up generating believable actions.

The claim that human-performed and CG GIFs could be confused for each other was further bolstered by treating the participants as raters and calculating an inter-rater reliability (IRR) score for how they rated whether the GIFs were human-performed or CG. The IRR score, Krippendorffs alpha, across all data types, was calculated to be 0.23925, showing only a slight agreement between participants about the origin of the action. The result was interpreted to mean that the human-performed and CG GIFs could not reliably be determined across participants and data sets.

Responses from the forced choice believability evaluation task between two action GIFs was assessed by treating it as a multi-class classification problem. The options given to participants were – CG action on the left, CG action on the right, both CG actions, and neither CG actions. As a reminder, poor participant performance on this task would be indicative that the CG actions were highly believable.

A four-class confusion matrix was created for the four responses possible, once each for human vs. agent mimicry, human vs. near variant, and human vs. far variant. In that order, the F1 scores were 0.8157, 0.7925, and 0.7678, respectively. The Matthews Correlation Coefficient was calculated respectively, to be 0.5414, 0.5938, and 0.4931 (strong positive correlations). Both results were calculated using micro-averaging due to the multi-class condition. The result indicated that when compared directly side-by-side to a human-performed action, participants were able to identify the human-performed action with relatively high accuracy, indicating that the actions were not as believable as desirable when compared directly against a human-performed action.

Recognizability Participants were asked to identify the actions performed by robot characters when assessing the recognizability of actions. Their mean accuracy (standard deviation in parenthesis) was determined across the different data sets ordered as human, agent mimicry, near variant, and far variant as 0.64 (0.26), 0.37 (0.24), 0.41 (0.23), and 0.33 (0.27). The median accuracy values for the same groups were 0.66, 0.4, 0.4, and 0.30. This outcome is a negative result that shows that recognizability for CG actions was comparable to random guessing, while human-performed actions were twice as likely to be recognized correctly.

A Shapiro-Wilk test found a non-normal distribution for the accuracy data. Therefore, a Kruskal-Wallis omnibus rank sum test was computed on the data. The results were found to be significant, and the null hypothesis was rejected with a confidence level $= 5.505771 \cdot 10^{-17}$. A Dunns test adjusted with Benjamini-Hochberg FDR showed that all the negative result relationships between the human data and the CG data were significant (all confidence levels < 0.019641).

Quality When assessing forced choice smoothness and quality of each action, the medians were calculated for the Likert scale responses and chi-squared tests were calculated for the human data compared to each of the three data types to see if there were significant associations between the types of data and the Likert scale responses for smoothness (or high quality respectively). For single GIF smoothness, the median values for human, agent mimicry, near variants, and far variants were 4, 2, 2, 3 on a 1 - 5 scale from not at all smooth to very smooth. The chi-squared test reported significance with $\tilde{\chi}^2$ = 304.9299 and a confidence level of $<$ 0.00001. For single GIF user-defined quality, the median scores reported for the same data sets were 4, 3, 3, 3 on a similar scale from very poor quality to very high quality. The chi-squared test reported significance with $\tilde{\chi}^2$ = 265.4731 and a confidence level of $<$ 0.00001.

When assessing forced choice smoothness and quality of each action, the percentage of results that were considered smoother (or higher-quality respectively) was recorded along with chi-squared tests that were calculated for human data compared to each of the three data types. The test was conducted to see if there were significant associations between the types of data and the selection of the human or computer action as more smooth (or high qual-

Table 1: Evaluation metrics. Note: (+) means higher is better and (-) means lower is better.

Architecture	Euclidean Distance (-)			Root Mean Squared Error (-)			Cosine Similarity (+)		
	Mean	Median	Std Dev	Mean	Median	Std Dev	Mean	Median	Std Dev
Conv. $(27\text{K}, 1)$	11.160	9.628	4.933	0.068	0.059	0.030	0.972	0.981	0.002
Conv. $(16\text{K}, 1)$	3.404	2.640	2.523	0.027	0.021	0.020	0.997	0.999	0.005
Conv. $(900, 30)$	2.975	1.871	3.125	**0.019**	**0.011**	0.019	0.996	0.999	0.007
Conv. $(450, 35)$	**2.763**	**1.656**	3.033	0.022	0.013	0.024	**0.997**	**1.000**	0.007
Vanilla 16K	6.634	6.473	1.278	0.052	0.051	0.010	0.993	0.993	0.003
Conductor 16K	6.114	5.976	1.294	0.048	0.047	0.010	0.994	0.994	0.002
Vanilla 27K	6.473	5.830	2.491	0.039	0.035	0.015	0.991	0.993	0.006
Conductor 27K	6.469	5.811	2.433	0.039	0.035	0.015	0.991	0.993	0.006

Table 2: Final epoch mean loss (lower is better).

Architecture	Training Loss	Validation Loss
Conv. $(27\text{K}, 1)$	**9.83**	145.28
Conv. $(16\text{K}, 1)$	14.65	119.00
Conv. $(900, 30)$	20.88	139.81
Conv. $(450, 35)$	21.13	138.49
Vanilla 16K	34.81	**38.01**
Conductor 16K	42.39	51.09
Vanilla 27K	39.86	48.56
Conductor 27K	47.44	56.17

ity respectively). For smoothness, human data was chosen as smoother 75.63% against agent mimicry, 77.54% against near variants, 75.30% against far variants, and 76.14% overall against all CG actions. There were no significant differences found between the groups, with $\tilde{\chi}^2 = 0.6701$ at a confidence level of < 0.05. For user-defined quality, the percentage of responses where human data was chosen as higher-quality was 73.58%, 78.26%, 76.74%, and 76.14% for the same ordering as smoothness. There was no significant association found either, with $\tilde{\chi}^2 = 2.6957$ at a confidence level of < 0.05.

Discussion

The action generation module described in this article is a vital part of the CARNIVAL architecture that enables improvisational embodied agents to improvise with people. Thus the evaluation was conducted based on its capabilities as a generator that could create believable, recognizable, and high-quality outputs. However, the larger evaluation design for an architecture that models creativity in such an open-ended and ill-defined domain has been challenging.

The task of evaluating generator outputs out of context could have been unusual for many human evaluators (though perhaps less so for those familiar with prop-based improv theatre). Therefore, the results of the human evaluation task may not truly reflect the agent's performance within the context of the entire CARNIVAL architecture. Additionally, the benchmarking experiments did evaluate how well the model learned the distribution of human actions, but the perceivable difference between models also needs to be evaluated. As a result, further studies will culminate in observational and in-person evaluation of the entire CARNIVAL architecture as an improvisational partner.

Our CVAE models all significantly overfit the data set due to the small size of data set used for training. Regularization only partially mitigated overfitting. We are currently conducting data collection and doing annotation on collected data to increase the amount of training data available.

A redesigned representation of physical attributes considering prop part ordering and spatial relationships is planned. The added nuance was not an initial priority. The suitability of the representation and its capacity for supporting transfer of learned actions to other props with similar physical attributes will also be evaluated.

We are planning to experiment with adversarial training of our architecture variants. The experiment would solve some of the challenges with the CVAE generation though it does introduce other difficulties like increased modal collapse that will need to be addressed. Additionally, adversarial training is challenging to perform at the moment, since we have minimal data. Another potential solution around the lack of data could be self-supervision on the unlabeled examples that have been collected but not annotated.

Additionally, even though the RNN based models show exceptional performances in terms of robustness and generalization, the loss values for the models are still relatively high compared to some of the best convolutional models in terms of training loss, mean Euclidean distance, and root mean squared loss. One potential improvement that can be made for the RNN based architectures is to increase the hidden layer size as well as stacking multiple RNN layers together to increase potential expressiveness of the models.

Conclusion

This article described a deep learning approach to generating candidate actions within the CARNIVAL architecture in order to address the improvisational action selection problem within human-agent embodied improvisation with objects. Four convolutional variants and four RNN variants of the proposed CVAE model were used to generate agent movements based on human inputs and prop physical attributes. In addition, the performance of those models was

analyzed using benchmarking metrics as well as Mechanical Turk studies to evaluate their believability, recognizability, and quality among observers.

The results showed that our models could successfully learn the distribution of training data. In terms of a subjective evaluation from human subjects on Mechanical Turk, the generated results were relatively believable as long as they were not simultaneously compared against human actions. They did not seem very recognizable, however. Therefore, we have implemented an agent speech bubble in the Robot Improv Circus that serves as a way to increase recognizability and communicate about agent intent. Finally, while human actions were consistently rated smoother and higher quality than CG actions, the ratings themselves, especially user-defined quality, were positive for all CG actions.

Acknowledgements

The authors would like to acknowledge the other researchers who supported this project as well as the reviewers who provided valuable feedback. This project is funded by a Creative Curricular Initiatives grant from the Georgia Tech Office of the Arts and Georgia Tech Arts Council.

References

Augello, A.; Cipolla, E.; Infantino, I.; Manfré, A.; Pilato, G.; and Vella, F. 2017. Creative robot dance with variational encoder. *CoRR* abs/1707.01489.

Crnkovic-Friis, L., and Crnkovic-Friis, L. 2016. Generative choreography using deep learning. *CoRR* abs/1605.06921.

Davis, N.; Hsiao, C.-P.; Yashraj Singh, K.; Li, L.; and Magerko, B. 2016. Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 196–207. ACM.

Habibie, I.; Holden, D.; Schwarz, J.; Yearsley, J.; and Komura, T. 2017. A recurrent variational autoencoder for human motion synthesis. In *BMVC*.

Hoffman, G., and Weinberg, G. 2010. Gesture-based human-robot jazz improvisation. In *Proceedings - IEEE International Conference on Robotics and Automation*, 582–587.

Holden, D.; Saito, J.; and Komura, T. 2016. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.* 35(4):138:1–138:11.

Ikeuchi, T. S. K. 2008. Synthesis of dance performance based on analyses of human motion and music.

Jacob, M., and Magerko, B. 2015. Interaction-based Authoring for Scalable Co-creative Agents. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*.

Jacob, M., and Magerko, B. 2018. Creative arcs in improvised human-computer embodied performances. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 62. ACM.

Kiasari, M. A.; Moirangthem, D. S.; and Lee, M. 2018. Human action generation with generative adversarial networks. *CoRR* abs/1805.10416.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipp, M.; Neff, M.; Kipp, K. H.; and Albrecht, I. 2007. Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In *IVA*.

Magerko, B.; Permar, J.; Jacob, M.; Comerford, M.; and Smith, J. 2014. An Overview of Computational Co-creative Pretend Play with a Human. In *Proceedings of First Workshop on Playful Virtual Characters at the Fourteenth Annual Conference on Intelligent Virtual Agents*.

Mancini, M., and Castellano, G. 2007. Real-time analysis and synthesis of emotional gesture expressivity.

Mathewson, K. W., and Mirowski, P. 2017. Improvised theatre alongside artificial intelligences. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Ng-Thow-Hing, V.; Luo, P.; and Okita, S. Y. 2010. Synchronized gesture and speech production for humanoid robots. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* 4617–4624.

Norman, D. A. 1988. The psychology of everyday things.

Ofli, F.; Erzin, E.; Yemez, Y.; and Tekalp, A. M. 2012. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE Transactions on Multimedia* 14:747–759.

O'Neill, B.; Piplica, A.; Fuller, D.; and Magerko, B. 2011. A knowledge-based framework for the collaborative improvisation of scene introductions. In *Proceedings of the 4th International Conference on Interactive Digital Storytelling*, volume 7069 LNCS, 85–96.

Reidsma, D.; van Welbergen, H.; Poppe, R.; Bos, P.; and Nijholt, A. 2006. Towards Bi-directional Dancing Interaction. In *Entertainment Computing - ICEC 2006*, 1–12.

Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and Eck, D. 2018. A hierarchical latent vector model for learning long-term structure in music. In *ICML*.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 3483–3491.

Spierling, U., and Szilas, N. 2009. Authoring issues beyond tools. In *Joint International Conference on Interactive Digital Storytelling*, 50–61. Springer.

Tang, T.; Jia, J.; and Mao, H. 2018. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, 1598–1606. New York, NY, USA: ACM.

Varadarajan, K. M., and Vincze, M. 2012. Afnet: The affordance network. In *Asian Conference on Computer Vision*, 512–523. Springer.