DOI: 10.1111/cgf.14647
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (2022), Number 8

Interaction Mix and Match: Synthesizing Close Interaction using Conditional Hierarchical GAN with Multi-Hot Class Embedding

Aman Goel[1,2], Qianhui Men[2] and Edmond S. L. Ho[†3]

[1]International Institute of Information Technology, Hyderabad, India
[2]Department of Engineering Science, Oxford University, United Kingdom
[3]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, United Kingdom

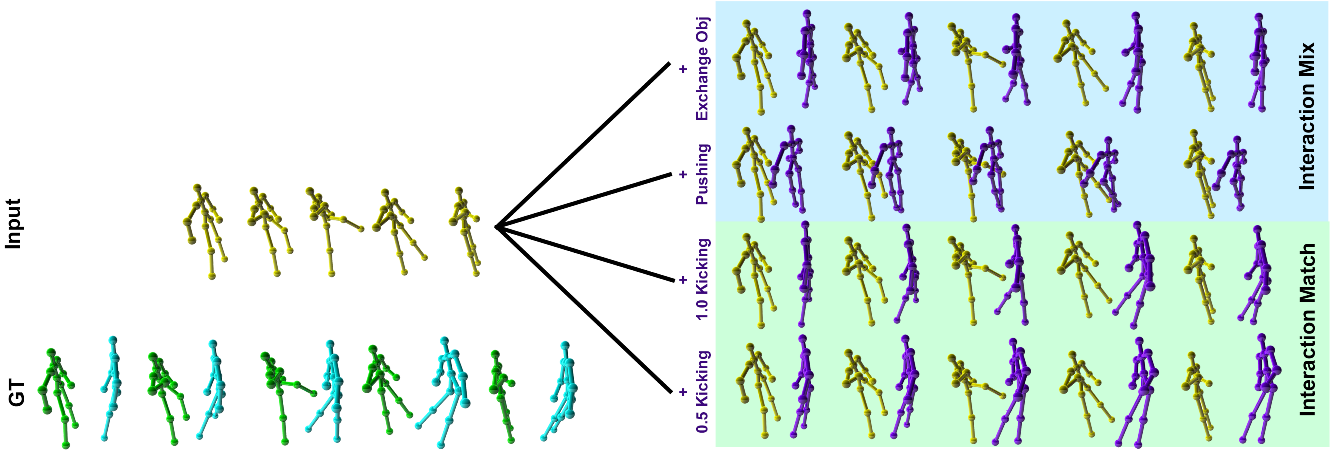


Figure 1: *Examples of Mix And Match Interactions. Given the input motion (yellow) of one character, different reactive motions (purple) can be generated by specifying the interaction labels.*

Abstract

Synthesizing multi-character interactions is a challenging task due to the complex and varied interactions between the characters. In particular, precise spatiotemporal alignment between characters is required in generating close interactions such as dancing and fighting. Existing work in generating multi-character interactions focuses on generating a single type of reactive motion for a given sequence which results in a lack of variety of the resultant motions. In this paper, we propose a novel way to create realistic human reactive motions which are not presented in the given dataset by mixing and matching different types of close interactions. We propose a Conditional Hierarchical Generative Adversarial Network with Multi-Hot Class Embedding to generate the Mix and Match reactive motions of the follower from a given motion sequence of the leader. Experiments are conducted on both noisy (depth-based) and high-quality (MoCap-based) interaction datasets. The quantitative and qualitative results show that our approach outperforms the state-of-the-art methods on the given datasets. We also provide an augmented dataset with realistic reactive motions to stimulate future research in this area.

CCS Concepts
• ***Computing methodologies*** → ***Motion capture; Machine learning; Motion processing; Animation;***

1. Introduction

Animating virtual human-like characters has been playing an important role in a wide range of applications, including computer

[†] Corresponding author: e.ho@northumbria.ac.uk

games, CGI/3D anime movies, virtual reality, etc. Character animation was traditionally a labour-intensive task as it requires extensive manual intervention from experienced animators to generate high-quality animations. With the advancement of motion acquisition technology, pre-recorded human motions are more accessible and can be used for animating virtual characters to semi-automate the animation production pipeline. However, a significant amount of manual intervention and post-processing is still required to remove artefacts caused by noise, marker swaps, and marker occlusions in the captured motions [PHMP19].

Although encouraging results are demonstrated by using data-driven and deep learning techniques [HSK16, LZCVDP20, SZZK21] in animating virtual characters in recent years, most of the existing work focuses on generating single-character or character-object interactions. On the other hand, generating multi-character (two or more characters) interactions is less researched in the literature. Synthesizing close interactions, such as dancing and hugging, between multi-character is a challenging research problem since the motions of the characters have to be precisely aligned spatially and temporally to avoid artefacts such as interpenetration of body parts while preserving the contextual meaning in the interaction (such as the contact at the hands in high-five). To better support animators in generating a wide variety of two-character interactions efficient with a high degree of controllability, we propose a method for *Interaction Mix and Match* in this research. Specifically, given the motion of a single character as an input, our method enables the synthesis of 1) *diverse reactive motions* by adjusting the scale of input label from the training data, and 2) *new interactions types* by mixing the interaction types informed by the multi-class label embeddings.

An intuitive solution will be combining individually captured motions [SKY07, SKSY08, KHL05] and editing the interaction according to additional constraints given by the users. However, careful design of the motion editing algorithms (such as [HKT10, HK09]) and parameter tuning are required to avoid interpenetration of body parts. Synthesizing a virtual partner/opponent from the user's movement has been explored in VR dancing [HCKL13] and sword fighting [DVLP20]. While the aforementioned approaches can generate the reactive motion (i.e. motion reacts to the human user) of the virtual character interactively, the reaction is highly similar to the pre-recorded motions in the dataset and results in a lack of variety in the synthesized interactions. On the other hand, modelling two-character interaction using Recurrent Neural Networks (RNNs) [KBM*20] have demonstrated encouraging results in predicting the future movements. The key insight is to model the correlation between the motions of the two characters by a 2-stream cross-conditioned network [WXXF22]. Although these approaches model interactions effectively for recognition and prediction tasks, they are less desirable for animation synthesis due to the low controllability of the resultant motions. Aristidou et al. [AYA*22] recently proposed a music-driven approach for synthesizing dancing motion including partnered dance (e.g. Salsa). Although this work shares similar interests as ours in generating close interactions with high-level control, [AYA*22] is specifically designed for dancing while our proposed method can be applied to synthesizing different kinds of leader-follower interactions.

The most relevant research to our work is the GAN-based interaction synthesis framework proposed by Men et al. [MSHL22]. To the best of our knowledge, [MSHL22] is the most recent method that generates the reactive motion of the follower according to the input motion of the leader. [MSHL22] considered a seq2seq reactive motion generator and adversarially rectify the synthesized motion with a binary and a multi-class discriminator. However, with no label information guided, their model can only produce a fixed reactive pattern learned from the input motion data and fails to create a mixture of reactions.

In this paper, we introduce a novel concept of using multi-hot class embedding in a conditional GAN to generate higher quality reactive motions with a larger degree of controllability over recent research [MSHL22]. The main goal of this work is to enable *Interaction Mix and Match* which is illustrated in Figure 1. For *interaction mix*, we aim to create reactions that combine different types of reactive styles directed by the multi-label indicator. By modifying the multi-hot class embedding, we can synthesize different reactive motions which are not presented in the motion datasets. For *interaction match*, we aim to create a single type of reactive motion in response to the input motion of the other character. By adjusting the numeric scale of the label indicator, our trained reaction generator can also create different levels of reactive variations.

Experimental results indicate that the interactions synthesized by our proposed method outperformed the state-of-the-art reaction synthesis model [MSHL22] as well as the baselines qualitatively and quantitatively. Numerically, lower Average Frame Distance (AFD) and Fréchet Inception Distance (FID) were obtained using our method which indicates our synthesized motions better resemble the original data. Qualitatively, our generated reaction has better synthesis quality with natural movements and flexible reactive styles. The synthesized reactions also show an improved representation space with clearer classification boundaries compared with non-label guided generations. We further demonstrate the positive impact of using our synthesized motions to improve the robustness of interaction recognition models through data augmentation.

1.1. Contributions

The contributions of this work can be summarized as follows:

- We proposed a new framework for generating diverse reaction given an input action using Conditional Hierarchical GAN
- We proposed using Multi-Hot Class Embedding to enable users to specify the action class to enhance controllability
- A new synthetic 2-character close interactions dataset will be available to stimulate the research in this area

2. Related Work

In this section, we will first review the related research in synthesizing multi-person interactions, which is roughly divided into algorithmic (Section 2.1) and data-driven (Section 2.2) approaches. The action-based motion synthesis approaches will then be reviewed in Section 2.3.

2.1. Algorithmic Interaction Synthesis

Synthesizing close interactions with two or multiple characters such as dancing and wrestling has been a challenging task in character animation. Early work in this area combines individually captured kickboxing motions to create two-character fighting scenes by constructing an action level motion graph [SKY07]. By further obtaining the potential future actions between the characters through game tree expansion, the best actions are selected by the min-max algorithm. Such close interaction patterns can be precomputed and stored as *Interaction Patches* [SKSY08] for synthesizing new multi-character scenes by concatenating different patches. To simulate hit-and-react interactions, momentum-based inverse kinematics [KHL05] is proposed to edit pre-recorded motions according to the strength of the external perturbation and the point of contact on the body.

Another stream of research mainly focuses on synthesizing motions, such as wrestling, which are difficult to be captured even if the motions are captured individually. Ho and Komura [HK07b] proposed using *tangle* and Gauss Linking Integral (GLI) to model the entangling body parts in close interactions. Such a topologically-based pose representation can be used for synthesizing tangling or detangling body parts by increasing or decreasing the magnitude of the GLI value accordingly. The topological approach is further combined with Rapidly-exploring random trees (RRT) [HK07a] to synthesize interactions such as carrying and piggybacking, as well as extending to *Topology Coordinates* [HK09] for synthesizing human-human and human-object close interactions by linearly interpolating key poses in the topology-based coordinate system.

To further provide animators with the flexibility to generate a wide range of close interactions, *Interaction Mesh* [HKT10] is proposed to preserve the spatial relations (i.e relative distances) between the character and its surroundings (including other characters and objects [HS13]) for motion adaption and motion retargeting. A volumetric mesh is constructed by applying Delaunay Tetrahedralization on a 3D point cloud sampled from the key locations on the character(s) and the object(s). By minimizing the deformation of the mesh during motion adaptation, the spatial relations between the character(s) and objects can be maintained. *Aura Mesh* [JKL18] is proposed to capture and preserve the spatial relations at skin-level when retargeting close interactions. Naghizadeh and Cosker [NC19] proposed giving higher priority to local connections over global ones in Interaction Mesh to enable large-scale transformation in multi-character motion retargeting. Kim et al. [KSK21] recently proposed using As-Rigid-As-Possible (ARAP) deformation on the pose graph for retargeting multiple characters. By avoiding computationally expensive spacetime optimization, multi-character motions can be retargeted interactively.

While the aforementioned approaches can effectively synthesize and edit close interactions, careful design of the interaction representations and parameter tuning are required which result in difficulties in generating large-scale close interactions efficiently.

2.2. Data-driven Interaction Synthesis

With the availability of interaction datasets [YHC*12, SYHS20, WXXF22, CCFB17], data-driven approaches are becoming more popular. By retrieving pre-recorded interactions, virtual partner/opponent can be synthesized based on the user's motion in dancing [HCKL13] and sword fighting [DVLP20] in VR. Kundu et al. [KBM*20] proposed using Cross-Conditioned Recurrent Networks for synthesizing human-human interactions. Specifically, given the observations (i.e pose sequence) of each character, two recurrent neural networks are used for predicting the motion of a character using another character's motion as input. Wen et al. [WXXF22] also proposed a 2-stream network with cross-interaction attention (XIA) module for predicting the motion based on the previous movements of the interacting characters. Huang et al. [HFKB15] generated reactive motion with maximum-entropy inverse optimal control (ME-IOC). However, their model can only sample the reaction from the training dataset. The most relevant work is the interaction synthesis framework proposed by Men et al. [MSHL22] in which a GAN-based model is used for generating the reactive motion of a character in response to the motion of another character given as input. By having an addition discriminator to predict the interaction class, the GAN-based model generates motions with better quality by taking into account the class information.

Data-driven approaches showed promising results in synthesizing high-quality interaction with minimal human intervention. However, the existing work lacks the controllability required by animators. Our proposed method addresses this problem by proving users with a high-level control (i.e. action label) to generate the desired close interactions easily.

2.3. Action-level Motion Synthesis

Synthesizing animation based on high-level controls such as 'walk', 'run' and 'jump' has been an active research area in character animation. Such an intuitive control is similar to instructing actors and actresses by the director in the real world. Arikan et al. [AFO03] proposed an optimization-based method to select relevant poses from the database to construct the resultant motion according to the input annotations (i.e. actions) while satisfying spatiotemporal constraints (such as reaching a particular location at a particular frame). A recent work proposed by Lee et al. [LMLL21] focuses on using reinforcement learning to synthesize time-critical motions from interactive character control including high-level action labels. Specifically, the teacher policy is first learned to achieve the tasks in optimal ways and then the time-critical student policy is followed to improve the responsiveness of the interactive control. Battan et al. [BAR*21] synthesize motion from the input label and give an initial set of frames using a 2-stage approach on an encoder-decoder architecture. In particular, the first stage predicts a sparse set of keyframes for the whole motion while the second stage generates the dense motion trajectories from the output of the first stage.

Action-conditional generative models have also formed a popular stream for motion synthesis in recent years. Guo et al. [GZW*20] proposed a Lie Algebra based Variational Auto-

Encoder (VAE) framework to generate motions according to the input action label. Petrovich et al. [PBV21] proposed a Transformer-based conditional VAE for 3D motion synthesis. The authors further demonstrated the effectiveness of using the learned sequence-level latent space for denoising noisy input such as the 3D pose sequence estimated from monocular video.

While the aforementioned action-conditioned approaches have been widely used in synthesizing single-person motions, less attention has been paid to synthesizing multi-person interactions using high-level controls. The recent MUGL [MGS22] adopted the Gaussian Mixture VAE (GMVAE) [DMG*16] which can generate single and multi-person 3D motions directly from an action label. However, the method does not 1) generate the reactive motion according to the input motion and 2) support the generation of motion from multi-class labels to further increase the diversity.

3. Methodology

In this section, we propose an end-to-end framework to synthesize stylized reactive motion informed by multi-hot action labelling. The goal of our model is to synthesize reactive patterns given an input action and its label indicator. For *interaction mix*, we aim to generate reactions that combine different classes of reactive styles directed by the multi-label indicator. For example, when given an input action of *kicking*, the user can specify the reactive motion to be *hugging* while *avoiding*. For *interaction match*, we aim to generate the reactive motion corresponding to the interaction type and motion of the input. With the generative nature of our model, diverse interaction variations can be created. The overview of the proposed framework can be found in Figure 2. The generator is formed by a seq2seq attentive network with the label embedding that learns the class-specific patterns, thus creating the controllable reactions when given the multi-hot labelling during inference. The multi-class discriminator with multi-layer sequential encoding is to generate high-quality reactive patterns and improves the representation space for reaction extrapolation.

3.1. Class Embedding

We propose a multi-hot class indicator to control the pattern of the generated reactive motion. Class encoding, which is usually modelled as a one-hot vector, is frequently adopted in conditional GAN-based models [MO14] to control the generator that only generates a certain class of samples. Here, we extend the one-hot encoding to multi-hot scenarios that softened the constrain to be multi-labelled. More specifically, for example, “1" is a positive label that the reactive motion is expected to perform, “0" is a neutral label while “-1" is a negative label that avoids generating the corresponding reaction. Theoretically, the multi-hot labelling can also be extended to floating-point numbers to show the continuous effect of content change in the generated motion. With the Multi-Hot Class Embedding, users can specify the reactive motion that combines the characteristics of different reactive styles.

Given an input action for character A denoted as $\mathbf{X}^A = \{x_t^A\}_{t=1}^T$, we separately model its skeleton hierarchy to encode the spatial features. To better analyze the input action, we consider a hierarchical encoder to model the spatial dynamics for each of the five body parts as well as the whole body, which results in six body structures, as illustrated in Figure 2. This is because modelling the actions within a group of joints can better learn the spatial correlations than modelling the whole body structure at once. For example, arm motions are more informative than other body parts in the upper body movements in interactions such as *shaking hands*. Instead of attaching the label information to each body slice, we only concatenate with the label embedding with the whole skeleton. This is to avoid the abuse of class labelling which may lead to an over-fitted generation. The one-hot labelling during training is defined by:

$$1_C(x) = \begin{cases} 1 & x \in C \\ 0 & x \notin C \end{cases} \tag{1}$$

Here, C is the corresponding class label for the input action $\mathbf{X}^A$. Before feeding to the generator, each of the five body part and the label-concatenated whole body skeleton is further encoded by a fully-connected layer respectively to encode the structural feature.

During inference, the interaction label can either be one-hot encoding or multi-hot encoding to realize *interaction mix* by combining different types of reactive motion. An example is shown in the upper-left corner of Figure 2. The multi-hot inference with one-hot training also ensures an extrapolative representation space (as shown in Figure 6) that a multi-pattern reactive motion can be blended to satisfy the specialized reactive conditions.

3.2. Generator Backbone

To generate realistic reactive motion, we utilize the conditional GAN with a multi-class discriminator to improve the quality of the synthesized class-informed reaction. Men et al. [MSHL22] designed two discriminators of a binary and a multi-class discriminator that cooperate to create realistic motion reactive to the input motion. In contrast, we discard the binary discriminator to distinguish real or fake generated motion to better train the GAN network with high variation in the generative model, which helps alleviate the mode collapse [SGZ*16] so that the reactive motions are less likely to be reduced to the same patterns.

Our generator G is formed by an attentive Long Short-Term Memory (LSTM)-based sequence-to-sequence (seq2seq) architecture. The encoder consists of one-layer bidirectional LSTM (Bi-LSTM) [ZZHY15] for each of the six body structures to model the temporal dynamics. The six output hidden states of the encoder are concatenated and fed into the decoder which is formed by a single-layer bidirectional LSTM to create the reactive motion. A frame-level attention mechanism is further introduced to enhance the connectivity between the encoder and the decoder, such that the model can recognize the most informative frames from input A and present similar importance to the reactive B. Assume that the concatenated hidden state of the encoder at frame t is h_t and $\hat{h}_t$ is the hidden state for the decoder, the seq2seq attention learns a contextual embedding c_t for decoder that allocates information from all steps in the encoded states::

$$c_t = \sum_{t_e=1}^{T} \phi(t_e, t) h_t. \tag{2}$$

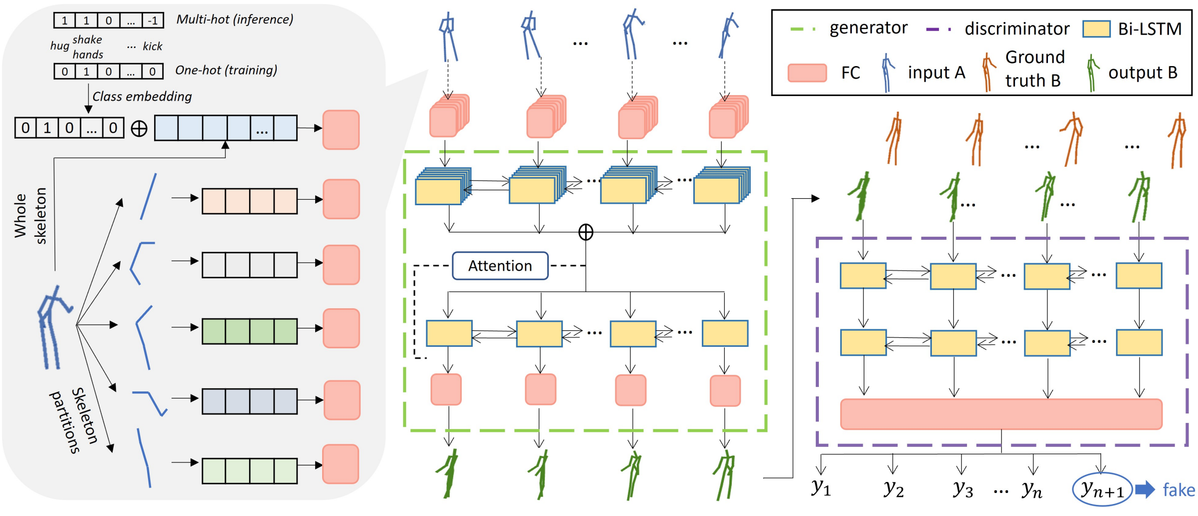


Figure 2: *The overview of the proposed framework. The call-out box on the left shows the label embedding and the body partition embedding of the input character. We then illustrate the attentive-based seq2seq architecture of our reactive motion generator in the middle. On the right hand side, the multi-class discriminator formed by Bi-LSTM layers is used for improving the quality of the synthesized reaction.*

where $\phi(t_e,t)$ is the attention weight that evaluates the correspondence between the state h_{t_e} at current encoder step t_e and the previous decoder state $\hat{h}_{t-1}$:

$$\phi(t_e,t) = \sigma_1(W_1\sigma_2(W_2[h_{t_e};\hat{h}_{t-1}])), \tag{3}$$

where W_1 and W_2 are the learnable parameters to increase capability of the seq2seq attention, and σ_1 and $\sigma 2$ are *softmax* and *tanh* activation functions, respectively. Here, we consider a global seq2seq attention where all timestamps of the encoder are included. However, it is also feasible to a local embedding where only partial input motion is observed for online predicting. The contextual embedding c_t is updated along with the hidden state at every time step, which sets up a prompt reaction by assigning step-wise significance to the decoder. Since the performance length of the input action of character A and the generated reaction for character B should be equal as in real-world interaction, T_e is the same as T in our seq2seq model. Furthermore, we connect a fully-connected layer to the end of the attentive Bi-LSTM decoder to reconstruct the reactive pose at every frame t.

3.3. The Multi-class Discriminator

We propose a multi-class discriminator that recognizes which interaction type the synthesized reactive motion $\hat{x}_B$ belongs to. The class-wise discriminator can help increase diversity in the reactive patterns to cater for different types of input action. The architecture of the multi-class discriminator is presented on the right hand side of Figure 2. Besides the n classes of the reactive labels $y_i, i = 1,\ldots,n$, we also include an extra fidelity label y_{n+1} to compensate for the functionality of a binary discriminator. To this end, the reaction to be judged will belong to either one of the real types of interaction class y_i or a fake class y_{n+1}.

Instead of feeding in the interaction with motions from both characters A and B, the discriminator only recognizes the reactive motion for B to justify the discrimination of different reactive patterns. It reduces the dependency for the discriminator on the extracted features from input action A and avoids generating collapsed results [CLJ*16, SGZ*16].

Specifically, our multi-class discriminator D consists of two-layer bidirectional LSTMs to classify the synthesized reactive motion from both the forward and backward movements. Compared to a single-directional LSTM, a discriminator using Bi-LSTM can extract high-level semantic features that significantly improve the performance for sequential classification. The hidden output at every time step is concatenated and embedded into a fully-connected layer with *softmax* activation to output the class probability.

3.4. Loss Function

During training, the synthesized reactive motion corresponding to B's motion $\hat{\mathbf{X}}^B$ is expected to be as close as possible to $\mathbf{X}^B$ with the L_1 norm to contrast their intensity similarities. The generated $\hat{\mathbf{X}}^B$ is also expected to be classified as the same class of $\mathbf{X}^A$. The corresponding adversarial loss for the proposed conditional GAN is defined as:

$$\begin{aligned}\mathcal{L}_{CGAN} = &-\mathbb{E}_{x,y\sim G}\log p(y|x,y<N+1)\\ &+\mathbb{E}_{x,y\sim G}\log p(y|x,y=N+1)\\ &+\mathbb{E}_{x,y\sim p_B}\log p(y|x,y<N+1),\end{aligned} \tag{4}$$

where p_B represents the real distributions of the character B's motion, y is the class label, and $p(y|x)$ stands for the probability of x being recognized as the synthesized interaction class.

Besides the L_1 and L_{CGAN} loss, we also adopt the continuity loss L_c and bone loss L_b introduced from [YXN*17] and [MSHL22] to preserve the motion consistency and the bone length nature. Here, we set the weight of the continuity loss to 1 instead of 0.01 used in

[MSHL22] and resulted in better motion quality as demonstrated in the accompanying video demos. The overall loss function follows a min-max optimization scheme:

$$\min_{G}\max_{D}\mathcal{L}_{CGAN}+\lambda_b\mathcal{L}_b+\lambda_c\mathcal{L}_c+\lambda_1\mathcal{L}_1, \tag{5}$$

4. Experimental Settings

In this section, the settings of the experiments will be presented. We first introduce the datasets used in this work in Section 4.1. The corresponding experimental protocols, such as data split, are explained in Section 4.2. Finally, the implementation details will be given in Section 4.3.

4.1. Datasets

4.1.1. SBU Two-person Interaction Dataset

The SBU dataset [YHC*12] contains eight classes of two-person interactions, including *approaching, departing, pushing, kicking, punching, exchanging objects, hugging,* and *shaking hands*. The *approaching* and *departing* interactions are excluded in our experiments since their reactive motions are standing still without meaningful movements. The interactions were captured using the Microsoft Kinect (depth-based sensor) and the 3D skeletal motions are provided. There are 7 subjects participated in the data collection with 197 motion sequences being used in total, and each character is represented by the locations of 15 joints in each frame.

4.1.2. Character-Character (2C) Dataset

The 2C dataset [SYHS20] contains high-quality kickboxing motions which are captured using an optical MoCap system. It contains 2 classes of two-person interactions, i.e., *kicking* and *punching* with diverse reactive patterns such as avoiding or being hit. 44 motions are used in the model and each character is represented by 20 joints in each frame. The joint information of the two characters is represented by 3D angular values with a skeleton hierarchy.

4.2. Dataset Settings

Same as [MSHL22], we perform leave-on-subject-out cross-validation on the SBU dataset. For 2C, we pre-process it by converting the joint angles into joint positions with forward kinematics (FK). We also normalize its skeleton by a scaling factor of 100 for both training and evaluation. The train:test ratio in 2C is 3:1. The 3D joint positions for both datasets are made relative to the root joint of character A, and the global root translation and rotation about the upward vector (i.e. y-axis) at the root joint of character A are removed as data standardization.

4.3. Implementation Details

The code base is built upon Keras platform, running on a PC with AMD Ryzen 7 3700x 8-Core Processor 4.4GHz, 64GB Memory and Nvidia RTX 2080 Ti Graphics card. RMSprop with a learning rate of 0.01 is used as the optimizer. There are 80 and 200 LSTM Neurons for each spatial slice and 480 and 1200 for the attentive layer for SBU and 2C, respectively. A batch size of 16 was used for both datasets, and we train 1600 epochs and 2000 epochs on SBU and 2C, respectively. For the weights of the network Loss, we set λ_b = 0.01, $\lambda_c = 1$, λ_1 = 1. Since we emphasize the continuity loss to create motions of better quality, a higher weight is given to it to encourage more smooth joint trajectories to be produced.

5. Experimental Results

Extensive experiments have been conducted to evaluate the effectiveness of our proposed method. Firstly, we evaluate the quality of the motion synthesized by our method quantitatively and qualitatively in Section 5.1. Secondly, we visualize the learned latent space in Section 5.2 to demonstrate how our method facilitates the synthesis of two-character interactions with high-level controls. Thirdly, we justify the design of our proposed framework by conducting an ablation study in Section 5.3. Finally, we demonstrate another application of our proposed method as a data augmentation approach to improving the robustness of interaction recognition models by providing a wider variety of synthesized training data in Section 5.4. We also compare our results with those obtained using the implementation of the state-of-the-art reaction synthesis model [MSHL22] provided by the authors to highlight the superior performance of our method.

5.1. Motion Synthesis

In this section, the quantitative and qualitative evaluations of the results generated by our methods and the baselines are presented in the following subsections.

5.1.1. Quantitative Analysis

For quantitative evaluation, we adopted the deterministic metric Average Frame Distance (AFD) to compare the synthesized skeletal motion with the ground truth in terms of geometric similarity:

$$AFD := \frac{1}{T}\sum^{t}||x_t^{B'}-x_t^{B}|| \tag{6}$$

where $x_t^{B'}$ and x_t^{B} are the synthesized and ground truth skeletal pose at time t, respectively, and $1 \leq t \leq T$. The AFD on different interaction classes on the SBU dataset is shown in Table 1. We compare our model with the traditional machine learning methods including Nearest Neighbour (NN), Hidden Mixture Model (HMM), Discrete Markov Decision Process [KZBH12], kernel-based reinforcement learning [HK14], maximum-entropy inverse optimal control [HFKB15], and the recent deep-learning based method [MSHL22]. The results show that our method outperformed all existing methods by achieving the lowest AFD in all 6 classes. Furthermore, our results show a more consistent performance across different class with a small range of AFD (0.32-0.50) while a large range of AFD (0.44-0.72) is obtained by the state of the art [MSHL22].

In addition to AFD, we further compute the Fréchet Inception Distance (FID) to measure the difference in the distribution between the synthesized and original motions in the SBU and 2C datasets. The results are presented in Table 2 and 3. Again, our method outperformed [MSHL22] by archiving a lower FID in all

	AFD (↓)						
Action	NN	HMM	[KZBH12]	[HK14]	[HFKB15]	[MSHL22]	OURS
Kick	0.81	0.92	0.65	0.92	0.67	0.53	**0.50**
Push	0.51	0.60	0.45	0.61	0.48	0.52	**0.43**
Shake hand	0.48	1.41	0.42	0.54	0.42	0.44	**0.40**
Hug	0.61	0.67	0.48	0.81	0.47	0.72	**0.42**
Ex. obj.	0.63	3.84	0.53	0.74	0.54	0.45	**0.40**
Punch	0.56	0.66	0.48	0.66	0.52	0.45	**0.32**

Table 1: *The AFD of different actions in the SBU dataset. Our method achieves the lowest AFD for all types of interactions.*

classes on both the SBU and 2C datasets. This indicates the motions synthesized using our method can consistently better resemble the motion quality as in the original motions in different datasets.

	FID (↓)					
Method	Kick	Push	Shake hand	Hug	Ex. obj.	Punch
[MSHL22]	10.8	20.8	23.8	29.5	16.7	11.2
OURS	**9.3**	**16.8**	**15.1**	**19.3**	**11.7**	**10.8**

Table 2: *The FID of different actions in the SBU dataset. Our method achieves a lower FID in all interaction classes.*

	FID (↓)	
Method	Kick	Punch
[MSHL22]	194.2	148.1
OURS	**164.2**	**122.4**

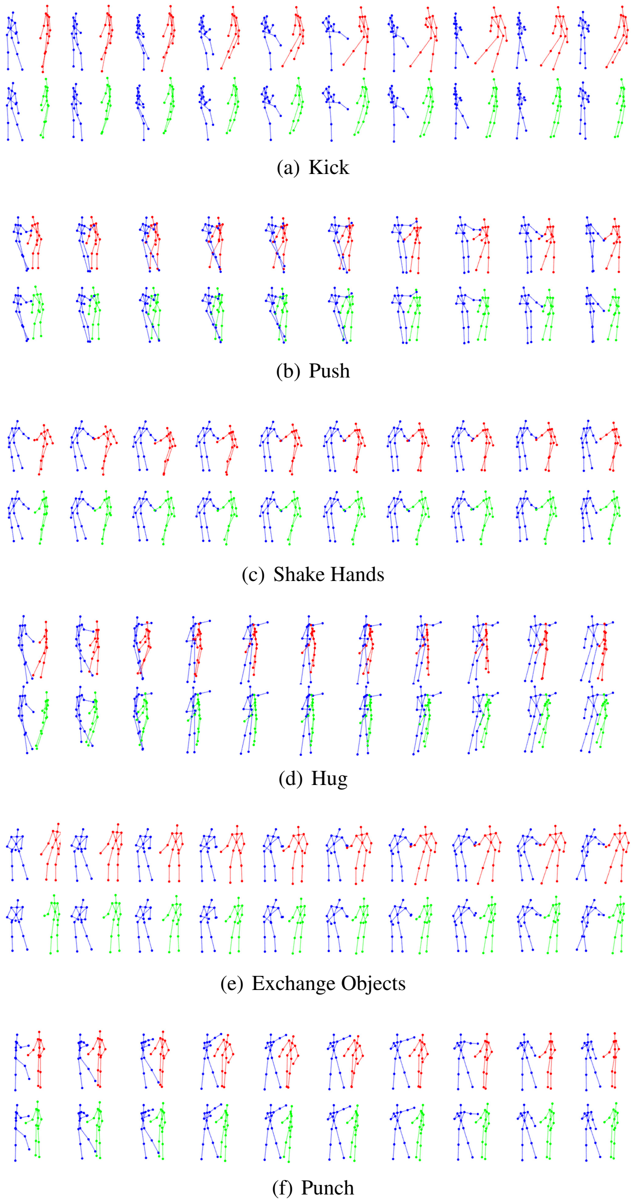
Table 3: *The FID of different actions in the 2C dataset. Our method achieves a lower FID in both kick and punch classes.*

5.1.2. Qualitative Analysis

To assess the visual quality of the interactions synthesized by our method, readers are referred to the accompanying video demo. In this section, we will qualitatively evaluate the synthesized motions in different experimental settings.

5.1.2.1. Interaction Match - Comparing with Ground Truth Data We first demonstrate the resemblance of data from the SBU dataset and the results are illustrated in Figure 3. Specifically, we employ a leave-one-subject-out cross-validation approach to train our model. At the inference stage, the input motion (colored in blue) alongside the ground truth interaction label is used for generating the reactive motion (colored in green). It can be seen that the synthesized reactive motions resemble the corresponding interactions as in the ground truth data (colored in red). Since our proposed method is based on a generative model, some variations are introduced to the reactive motions when compared with the ground truth data. For example, the leg movements of *Kick* (Figure 3(a)) and *Exchange Objects* (Figure 3(e)) are different from their corresponding ground truth motions. Nevertheless, the context of the interactions is correctly preserved.

We further demonstrate the capability and the generality of our proposed method by generating reactive motions on the high-quality 2C dataset. Examples of the synthesized motions are illustrated in Figure 4. Similar to the results obtained in the SBU dataset, our method can resemble the interactions in the 2C dataset while introducing some variations to the synthesized reactive motions.


(a) Kick

(b) Push

(c) Shake Hands

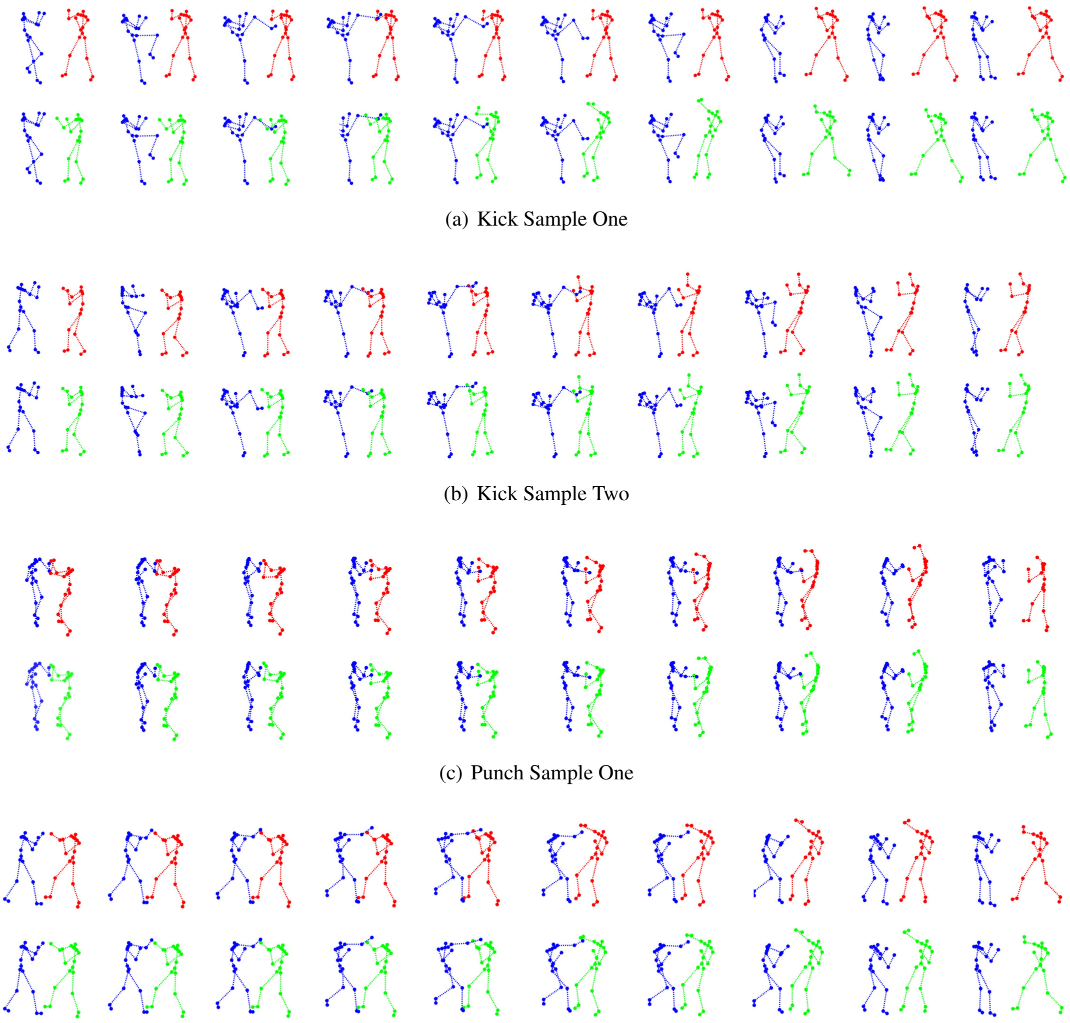
(d) Hug

(e) Exchange Objects

(f) Punch

Figure 3: *Examples of real and synthesized reactive motion of the six interaction classes in the SBU dataset. Given the input motion (blue) and the interaction label, the reactive motion (green) can be synthesized. The ground truth reactive motion (red) is also included for comparison.*

5.1.2.2. Interaction Mix With the multi-hot embedding in our proposed framework, users can control the reactive motions to be synthesized by specifying the interaction labels as a multi-hot vector. Some examples generated from the SBU dataset are shown in Figure 5. It can be seen that the generations show hybrid reactive



(a) Kick Sample One

(b) Kick Sample Two

(c) Punch Sample One

(d) Punch Sample Two

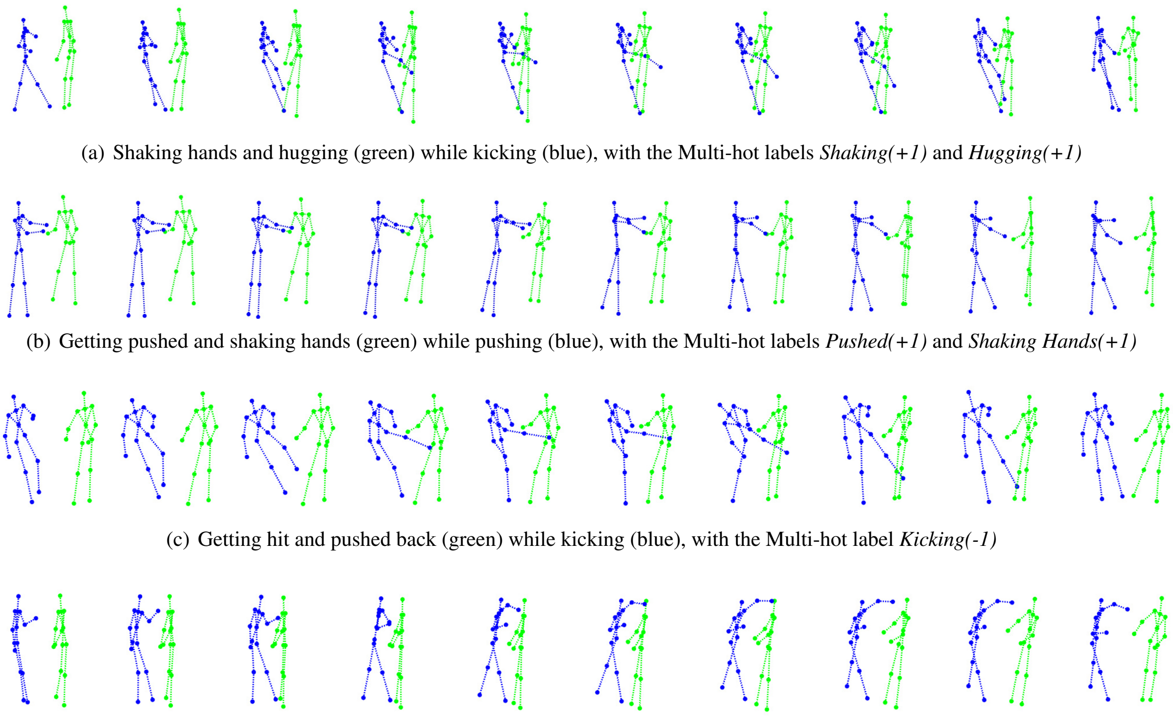
Figure 4: *Examples of real and synthesized reactive motion of kicking and punching interaction in the 2C dataset, respectively. Given the input motion (blue) and the interaction label, the reactive motion (green) can be synthesized. The ground truth reactive motion (red) is also included for comparison.*

patterns. For example, we set the labels of shaking hands and hugging to be positive to generate the reaction in (a). We observe that the synthesized character B is getting close to character A meanwhile ready to shake hands (as shown in the last few frame stamps). In (b), the generated motion for character B is shaking hands with A while laying back to avoid being pushed.

5.2. Analysis of the Latent Space

To analyze the quality of the latent space learned, the t-SNE of the embedding space is illustrated in Figure 6. Specifically, we take the embedding of the reactive interaction generated, just after the concatenation of the Conditional Hierarchical Bi-LSTM outputs and run t-SNE (t-distributed stochastic neighbor embedding). Again, we follow the protocol to have a leave-one-subject-out cross-validation in this experiment.

The results indicate that most of the interaction classes formed clusters which show a low degree of inter-class similarity. In particular, 4 out of 6 classes, including *Exchange Objects, Shake Hands, Hug* and *Kick* are not overlapping with the other classes. Although there is an overlapping between the regions covered by the clusters of the *Punch* and *Push* classes, it can be seen that the movements of the character who punches or pushes the other are very similar in terms of the skeletal motions. The corresponding reactive motions are also similar in those two classes. Furthermore, the samples of the *Push* are essentially outside the cluster of *Punch*. This highlight the effectiveness of the learning of the latent space for representing different types of interactions. In addition to separating different



(a) Shaking hands and hugging (green) while kicking (blue), with the Multi-hot labels *Shaking(+1)* and *Hugging(+1)*

(b) Getting pushed and shaking hands (green) while pushing (blue), with the Multi-hot labels *Pushed(+1)* and *Shaking Hands(+1)*

(c) Getting hit and pushed back (green) while kicking (blue), with the Multi-hot label *Kicking(-1)*

(d) Getting hit and pushed back (green) while punching (blue), with the Multi-hot label *Punching(-1)*

Figure 5: *Examples of Interaction Mix (a and b) and Match (c and d) on the SBU dataset along with their Multi-hot Labels. The input and the synthesized reaction are colored blue and green, respectively.*

types of interactions, Figure 6 also highlights our latent space can effectively capture the similarity between different classes. For example, *Shake Hand* and *Exchange Object* are more similar as well as the *Punch* and *Push* pair discussed above.

We further compare the t-SNE of the embedding space obtained using [MSHL22] in Figure 7. It can be seen that samples from different interaction types are mixed together in the latent space. We argue that a more well-constructed latent space not only informing the user about which interaction types are more suitable for *Interaction Mix* based on their similarity, but also facilitates the extrapolation of different types of interaction with the multi-hot label in *Interaction Mix*.

5.3. Ablation Study

In this section, we justify the design of different components in our proposed framework by conducting an ablation study. The results are presented in Table 4 and 5 for the SBU and 2C datasets, respectively.

From the results in Table 4, our proposed network without the bone loss ($\mathcal{L}_b$) achieved the lowest AFD across all 6 classes. The complete version of our framework achieved the lowest AFD in the *Punch* class and second-lowest in the rest of the 5 classes. Since the SBU dataset is captured using Microsoft Kinect, we observed that the skeleton motions captured are not of high-quality and the bone lengths of the characters vary over time. As a result, having the bone loss maintains the bone lengths over time will not give our method a favourable AFD on this dataset, although it is essential to enforce such a constraint in rigid-body character animation.

We also show the ablation test on the high-quality 2C dataset with the results given in Table 5. By combining all constraints, our model shows the best motion quality with the lowest FID. Since the bone length captured in 2C is more stable, the advantage of continuity constraint $\mathcal{L}_c$ is more advantageous than $\mathcal{L}_b$ compared to SBU dataset. Furthermore, the effectiveness of the multi-hot encoding is also significant when generating between these two highly-similar interaction.

	AFD (↓)					
Method	Kick	Push	Shake hand	Hug	Ex. Obj.	Punch
OURS	0.5	0.43	0.4	0.42	0.4	**0.32**
OURS w/o *D*	0.53	0.45	0.42	0.49	0.44	0.38
OURS w/o $\mathcal{L}_b$	**0.45**	**0.42**	**0.39**	**0.40**	**0.39**	**0.32**
OURS w/o $\mathcal{L}_c$	0.59	0.45	0.55	0.46	0.41	0.4
OURS w/o FC	0.55	0.47	0.46	0.49	0.41	0.4
OURS w/o Multi-Hot Embed	0.51	0.48	0.50	0.66	0.46	0.45

Table 4: *Ablations of main components in the proposed model evaluated by AFD on the SBU dataset. The model without the bone loss ($\mathcal{L}_b$) gives the lowest AFD*

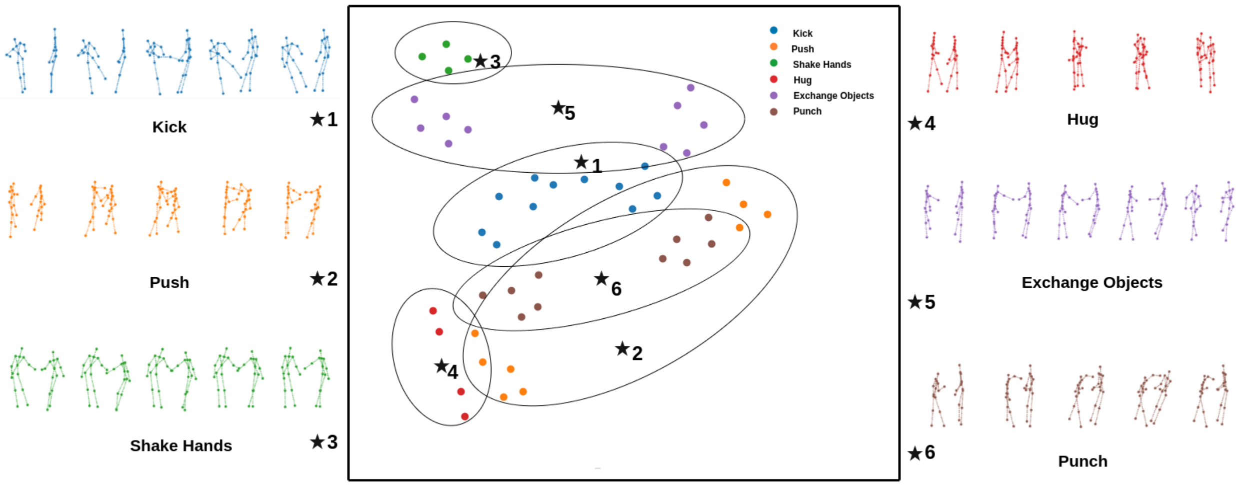


Figure 6: *t-SNE of the embedding space obtained using our proposed method.*

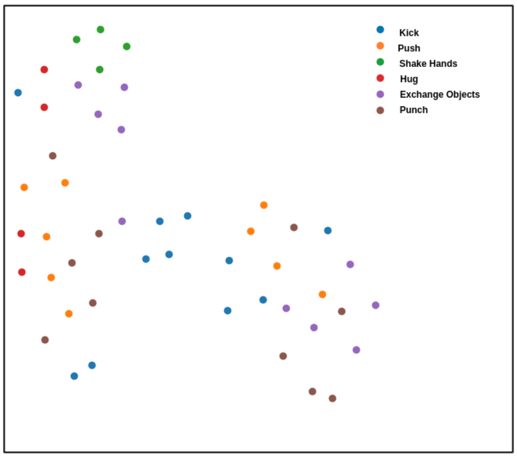


Figure 7: *t-SNE of the embedding space obtained using [MSHL22].*

5.4. Data Augmentation for Interaction Recognition

To demonstrate another application of our proposed method, we conduct a series of experiments to evaluate how the interactions synthesized by our method can be used for enhancing the datasets for training interaction recognition algorithms. Specifically, we generate a new set of 300 (50 per class) interactions from the SBU dataset and the dataset will be available to the public to stimulate the research in this area. We train an interaction classifier to test the quality of the synthesized interaction. The classifier has the same structure as the multi-class discriminator but outputs N classes instead of N+1. Its train-test split settings are as follows:

	FID (↓)	
Method	Kick	Punch
OURS	**164.2**	**122.4**
OURS w/o D	174.4	128.2
OURS w/o $\mathcal{L}_b$	166.4	124.8
OURS w/o $\mathcal{L}_c$	180.2	131.4
OURS w/o FC	175.3	129.1
OURS w/o Multi-Hot Embed	175.1	129.7

Table 5: *Ablations of main components in the proposed model evaluated by FID on the 2C dataset. Our model with all loss terms shows the lowest FID*

.

- **Original**: We follow the half-half data split [YHC*12] widely used in the interaction recognition protocol in the SBU dataset
- **Augmented**: We evenly divided the synthesized dataset and add them to the original SBU training and testing set

The classification accuracies are reported in Table 6. The results show that the variations introduced by our augmented dataset have a positive impact on improving the interaction classification performance over the original SBU dataset. In particular, 5 out of 6 classes have an increase in the classification accuracy with a range from 1.33% to 21.39%.

6. Discussion and Conclusions

In this paper, we propose a novel GAN-based framework for synthesizing 2-character interactions based on the input motion of one character and the interaction label. By incorporating the multi-hot class embedding, our method enables *Interaction Mix* which generates new interaction classes by extrapolating single-hot labelled in-

			Accuracy					
Different Splits	# Train Seq.	# Test Seq.	Kick	Push	Shake Hand	Hug	Exchange Object	Punch
Original	99	98	**0.947368**	0.941176	0.692308	0.538462	0.667367	0.888889
Augmented	124	123	**0.996013**	0.877035	**0.705583**	**0.640724**	**0.881263**	**0.911242**

Table 6: *Accuracy of the classifier on different splits on the original and augmented SBU datasets. The augmented dataset helps improve the recognition performance over most of the interaction classes in the original SBU dataset.*

teractions for the training data, as well as *Interaction Match* which synthesizes diverse interaction variations from the original interaction classes in the dataset. Experimental results show that our method outperforms the existing work including the most relevant work [MSHL22].

While the proposed method provides users with a large degree of high-level control, mixing very dissimilar interaction types may result in artefacts such as interpenetration of body parts (see the example illustrated in Figure 8). This can happen since our method does not handle collisions explicitly. Further exploring the usage of the latent space, such as interpolation and extrapolation, learned using our model as well as learning a topology-aware latent space [HSCY13] for avoiding interpenetration are potential future directions. Using existing close interaction editing methods such as *Interaction Mesh* [HKT10] and *Aura Mesh* [JKL18] as a post-processing step to clean up the interpenetration as well as maintain the contact points between the characters can be another solution as demonstrated in [HCKL13]. In terms of the quality of the synthesized motion, artefacts such as foot sliding can be found in the synthesized motions since there is no explicit loss term on the stepping pattern in our proposed network and this is quite common to other GAN-based motion synthesis methods [AYA*22] such as MotionCLIP [TGH*22]. We are interested in incorporating the contact consistency loss proposed in GANimator [LAZ*22] or an additional post-processing step [WHSZ21] to clean up the foot sliding artefacts in the results as an extension of the proposed work.

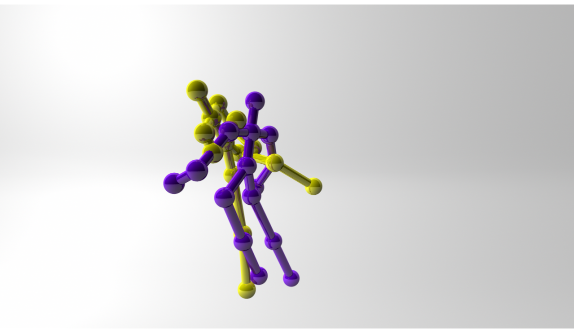


Figure 8: *Artefacts such as interpenetration of body parts can be found when mixing very dissimilar interaction types, e.g. mixing Kicking and Hugging.*

In the future, we could use a classifier [ZLX17] to generate the Multi-Hot Class Embedding during training and inference time. The Multi-Hot Class Embedding could also be extended to single body motion synthesis [BAR*21] to generate controllable synthesized motions.

References

[AFO03] Arikan O., Forsyth D. A., O'Brien J. F.: Motion synthesis from annotations. *ACM Trans. Graph. 22*, 3 (jul 2003), 402–408. doi:10.1145/882262.882284. 3

[AYA*22] Aristidou A., Yiannakidis A., Aberman K., Cohen-Or D., Shamir A., Chrysanthou Y.: Rhythm is a dancer: Music-driven motion synthesis with global structure. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–1. doi:10.1109/TVCG.2022.3163676. 2, 11

[BAR*21] Battan N., Agrawal Y., Rao S. S., Goel A., Sharma A.: Glocalnet: Class-aware long-term human motion synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2021), pp. 879–888. doi:10.1109/WACV48630.2021.00092. 3, 11

[CCFB17] Coppola C., Cosar S., Faria D. R., Bellotto N.: Automatic detection of human interactions from rgb-d data for social activity classification. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (2017), IEEE Press, p. 871–876. doi:10.1109/ROMAN.2017.8172405. 3

[CLJ*16] Che T., Li Y., Jacob A. P., Bengio Y., Li W.: Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136* (2016). 5

[DMG*16] Dilokthanakul N., Mediano P. A. M., Garnelo M., Lee M. C. H., Salimbeni H., Arulkumaran K., Shanahan M.: Deep unsupervised clustering with gaussian mixture variational autoencoders, 2016. URL: https://arxiv.org/abs/1611.02648, doi:10.48550/ARXIV.1611.02648. 4

[DVLP20] Dehesa J., Vidler A., Lutteroth C., Padget J.: Touché: Data-driven interactive sword fighting in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2020), Association for Computing Machinery, p. 1–14. URL: https://doi.org/10.1145/3313831.3376714. 2, 3

[GZW*20] Guo C., Zuo X., Wang S., Zou S., Sun Q., Deng A., Gong M., Cheng L.: *Action2Motion: Conditioned Generation of 3D Human Motions*. Association for Computing Machinery, New York, NY, USA, 2020, p. 2021–2029. URL: https://doi.org/10.1145/3394171.3413635. 3

[HCKL13] Ho E. S. L., Chan J. C. P., Komura T., Leung H.: Interactive partner control in close interactions for real-time applications. *ACM Trans. Multimedia Comput. Commun. Appl. 9*, 3 (jul 2013). doi:10.1145/2487268.2487274. 2, 3, 11

[HFKB15] Huang D.-A., Farahmand A.-m., Kitani K. M., Bagnell J. A.: Approximate maxent inverse optimal control and its application for mental simulation of human interactions. In *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015). 3, 6, 7

[HK07a] Ho E. S. L., Komura T.: Planning tangling motions for humanoids. In *2007 7th IEEE-RAS International Conference on Humanoid Robots* (2007), pp. 507–512. doi:10.1109/ICHR.2007.4813918. 3

[HK07b] Ho E. S. L., Komura T.: Wrestle alone : Creating tangled motions of multiple avatars from individually captured motions. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)* (2007), pp. 427–430. doi:10.1109/PG.2007.54. 3

[HK09] Ho E. S. L., Komura T.: Character Motion Synthesis by Topology Coordinates. *Computer Graphics Forum 28*, 2 (2009), 299–308. doi:10.1111/j.1467-8659.2009.01369.x. 2, 3

[HK14] Huang D.-A., Kitani K. M.: Action-reaction: Forecasting the dynamics of human interaction. In *European Conference on Computer Vision* (2014), pp. 489–504. 6, 7

[HKT10] Ho E. S. L., Komura T., Tai C.-L.: Spatial relationship preserving character motion adaptation. *ACM Trans. Graph. 29*, 4 (jul 2010). doi:10.1145/1778765.1778770. 2, 3, 11

[HS13] Ho E. S. L., Shum H. P. H.: Motion adaptation for humanoid robots in constrained environments. In *2013 IEEE International Conference on Robotics and Automation* (2013), pp. 3813–3818. doi:10.1109/ICRA.2013.6631113. 3

[HSCY13] Ho E. S. L., Shum H. P. H., Cheung Y.-m., Yuen P. C.: Topology aware data-driven inverse kinematics. *Computer Graphics Forum 32*, 7 (2013), 61–70. doi:https://doi.org/10.1111/cgf.12212. 11

[HSK16] Holden D., Saito J., Komura T.: A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph. 35*, 4 (jul 2016). doi:10.1145/2897824.2925975. 2

[JKL18] Jin T., Kim M., Lee S.-H.: Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters. *Computer Graphics Forum 37*, 2 (2018), 311–320. doi:https://doi.org/10.1111/cgf.13363. 3, 11

[KBM*20] Kundu J. N., Buckchash H., Mandikal P., V R. M., Jamkhandi A., Babu R. V.: Cross-conditioned recurrent networks for long-term synthesis of inter-person human motion interactions. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2020), pp. 2713–2722. doi:10.1109/WACV45572.2020.9093627. 2, 3

[KHL05] Komura T., Ho E. S. L., Lau R. W. H.: Animating reactive motion using momentum-based inverse kinematics: Motion capture and retrieval. *Comput. Animat. Virtual Worlds 16*, 3-4 (2005), 213–223. doi:http://dx.doi.org/10.1002/cav.v16:3/4. 2, 3

[KSK21] Kim J., Seol Y., Kwon T.: Interactive multi-character motion retargeting. *Computer Animation and Virtual Worlds 32*, 3-4 (2021), e2015. doi:https://doi.org/10.1002/cav.2015. 3

[KZBH12] Kitani K. M., Ziebart B. D., Bagnell J. A., Hebert M.: Activity forecasting. In *European Conference on Computer Vision* (2012), pp. 201–214. 6, 7

[LAZ*22] Li P., Aberman K., Zhang Z., Hanocka R., Sorkine-Hornung O.: Ganimator: Neural motion synthesis from a single sequence. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 138. 11

[LMLL21] Lee K., Min S., Lee S., Lee J.: Learning time-critical responses for interactive character control. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459826. 3

[LZCVDP20] Ling H. Y., Zinno F., Cheng G., Van De Panne M.: Character controllers using motion vaes. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392422. 2

[MGS22] Maheshwari S., Gupta D., Sarvadevabhatla R.: Mugl: Large scale multi person conditional action generation with locomotion. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Los Alamitos, CA, USA, jan 2022), IEEE Computer Society, pp. 747–755. doi:10.1109/WACV51458.2022.00082. 4

[MO14] Mirza M., Osindero S.: Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014). 4

[MSHL22] Men Q., Shum H. P., Ho E. S., Leung H.: Gan-based reactive motion synthesis with class-aware discriminators for human–human interaction. *Computers & Graphics 102* (2022), 634–645. doi:https://doi.org/10.1016/j.cag.2021.09.014. 2, 3, 4, 5, 6, 7, 9, 10, 11

[NC19] Naghizadeh M., Cosker D.: Multi-character motion retargeting for large-scale transformations. In *Advances in Computer Graphics* (Cham, 2019), Gavrilova M., Chang J., Thalmann N. M., Hitzer E., Ishikawa H., (Eds.), Springer International Publishing, pp. 94–106. 3

[PBV21] Petrovich M., Black M. J., Varol G.: Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)* (2021). 4

[PHMP19] Perepichka M., Holden D., Mudur S. P., Popa T.: Robust marker trajectory repair for mocap using kinematic reference. In *Motion, Interaction and Games* (New York, NY, USA, 2019), MIG '19, Association for Computing Machinery. doi:10.1145/3359566.3360060. 2

[SGZ*16] Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., Chen X.: Improved techniques for training gans. In *Advances in neural information processing systems* (2016), pp. 2234–2242. 4, 5

[SKSY08] Shum H. P. H., Komura T., Shiraishi M., Yamazaki S.: Interaction patches for multi-character animation. In *ACM SIGGRAPH Asia 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH Asia '08, Association for Computing Machinery. doi:10.1145/1457515.1409067. 2, 3

[SKY07] Shum H. P. H., Komura T., Yamazaki S.: Simulating competitive interactions using singly captured motions. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2007), VRST '07, Association for Computing Machinery, p. 65–72. doi:10.1145/1315184.1315194. 2, 3

[SYHS20] Shen Y., Yang L., Ho E. S. L., Shum H. P. H.: Interaction-based human activity comparison. *IEEE Transactions on Visualization and Computer Graphics 26*, 8 (2020), 2620–2633. doi:10.1109/TVCG.2019.2893247. 3, 6

[SZZK21] Starke S., Zhao Y., Zinno F., Komura T.: Neural animation layering for synthesizing martial arts movements. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459881. 2

[TGH*22] Tevet G., Gordon B., Hertz A., Bermano A. H., Cohen-Or D.: Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063* (2022). 11

[WHSZ21] Wang H., Ho E. S. L., Shum H. P. H., Zhu Z.: Spatio-temporal manifold learning for human motions via long-horizon modeling. *IEEE Transactions on Visualization and Computer Graphics 27*, 1 (2021), 216–227. doi:10.1109/TVCG.2019.2936810. 11

[WXXF22] Wen G., Xiaoyu B., Xavier A.-P., Francesc M.-N.: Multi-person extreme motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2022). 2, 3

[YHC*12] Yun K., Honorio J., Chattopadhyay D., Berg T. L., Samaras D.: Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2012), pp. 28–35. doi:10.1109/CVPRW.2012.6239234. 3, 6, 10

[YXN*17] Yan Y., Xu J., Ni B., Zhang W., Yang X.: Skeleton-aided articulated motion generation. In *Proceedings of the 25th ACM international conference on Multimedia* (2017), pp. 199–207. 5

[ZLX17] Zhang S., Liu X., Xiao J.: On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2017), pp. 148–157. doi:10.1109/WACV.2017.24. 11

[ZZHY15] Zhang S., Zheng D., Hu X., Yang M.: Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation* (2015), pp. 73–78. 4