# Learning End-to-End Action Interaction by Paired-Embedding Data Augmentation

Ziyang Song[1], Zejian Yuan[1], Chong Zhang[2], Wanchao Chi[2], Yonggen Ling[2], and Shenghao Zhang[2]

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, China
songzy305@yahoo.com, yuan.ze.jian@xjtu.edu.cn
[2] Tencent Robotics X, China
aerentzhang@gmail.com, wanchaochi@tencent.com, ylingaa@connect.ust.hk, popshzhang@pku.edu.cn

**Abstract.** In recognition-based action interaction, robots' responses to human actions are often pre-designed according to recognized categories and thus stiff. In this paper, we specify a new Interactive Action Translation (IAT) task which aims to learn end-to-end action interaction from unlabeled interactive pairs, removing explicit action recognition. To enable learning on small-scale data, we propose a Paired-Embedding (PE) method for effective and reliable data augmentation. Specifically, our method first utilizes paired relationships to cluster individual actions in an embedding space. Then two actions originally paired can be replaced with other actions in their respective neighborhood, assembling into new pairs. An Act2Act network based on conditional GAN follows to learn from augmented data. Besides, IAT-test and IAT-train scores are specifically proposed for evaluating methods on our task. Experimental results on two datasets show impressive effects and broad application prospects of our method.

## 1 Introduction

Action interaction is an essential part of human-robot interaction (HRI) [1]. For robots, action interaction with human includes two levels: 1) perceiving human actions and understanding intentions behind; 2) performing responsive actions accordingly. Thanks to the development of action recognition methods [2], considerable progress has been made on the first level. As for the second level, robots often perform pre-designed action responses according to recognition results. We call this scheme as recognition-based action interaction. However, colorful appearances of human actions are mapped to a few fixed categories in this way, leading to a few fixed responses. Robots' action responses are thus stiff, lacking in human-like vividity. Moreover, annotating data for training action recognition models consumes manpower.

In this paper, we aim to learn end-to-end interaction from unlabeled action interaction data. Explicit recognition is removed, leaving the interaction implicitly guided by high-level semantic translation relationships. To achieve this

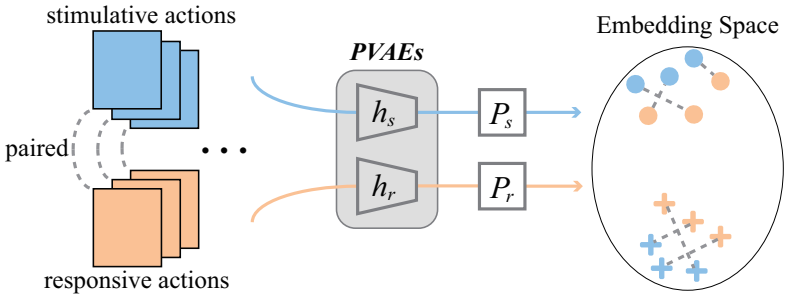


**Fig. 1.** An overview of our proposed Paired-Embedding (PE) method. Colors distinguish stimulations and responses. Circle and cross denote actions of different semantic categories. Dotted lines describe paired relationships.

goal, we specify a novel Interactive Action Translation (IAT) task: Given a set of "stimulation-response" action pairs conforming to defined interaction rules and without category labeled, learn a model to generate a response for a given stimulation during inference. The generated results are expected to manifest:

**1) reality:** indistinguishable from real human actions;

**2) precision:** conforming to defined interaction rules semantically, conditioned on the stimulation;

**3) diversity:** be various each time given the same stimulation.

For different interaction scenes and defined rules, paired action data need to be re-collected each time. Thus IAT would be more appealing if learning from a small number of samples. However, the task implicitly seeks for a high-level semantic translation relationship, which is hard to generalize from insufficient data. Moreover, the multimodal distribution of real actions is difficult to approximate without sufficient data. The contradiction between task goals and applications poses the main challenge: to achieve the three generation goals above with small-scale data.

Data augmentation is widely adopted to improve learning on small datasets. Traditional augmentation strategies apply hand-crafted transformations on existing data, thus only bring changes in limited modes. Generative Adversarial Networks (GAN) [3] emerges as a powerful technique to generate realistic samples. Nonetheless, a reliable GAN itself requires large-scale data to train. Some variants of GAN, like ACGAN [4], DAGAN [5], and BAGAN [6], are proposed to augment data for classification tasks. However, all of them need category labels that are not provided in our task. Therefore, a specially designed augmentation method is needed for small-scale unlabeled data in IAT.

We propose a novel Paired-Embedding (PE) method, as Fig. 1 shows. Through encoders in a Paired Variational Auto-Encoders (PVAEs) and PCA-based linear dimension reductions, individual action instances are projected into a low-dimension embedding space. Along with the vanilla VAE objectives [7], we employ a new PE loss utilizing paired relationships between actions to train PVAEs. Specifically, VAE loss prefers large variance of action embeddings while PE loss

pull actions within the same categories together. As a result, action instances are clustered in the embedding space in an unsupervised manner. Subsequently, both two actions in a data pair are allowed to be replaced with other instances in their respective neighborhood, assembling into new pairs conforming to defined interaction rules semantically. Therefore, the diversity of paired data is significantly and reliably enriched. Finally, we train an Act2Act network based on conditional GAN [8] on augmented data to solve our task.

Although IAT is formally an instance-conditional generation task like image translation [9, 10], it actually conditions on the semantic category of input action instances. Therefore, evaluation metrics for neither image translation [11, 12] nor category-conditional generation [13] is suitable for this task. Considering the three generation goals, we propose two evaluation metrics, IAT-test and IAT-train scores, to compare methods for our task from distinct perspectives. Experiments show that our proposed method gives satisfying generated action responses, both quantitatively and qualitatively.

The major contributions of our work are summarized as follows:

1) We specify a new IAT task, aiming to learn end-to-end action interaction from unlabeled interactive action pairs.

2) We design a PE data augmentation method to resolve the main challenge of our task: learning with a small number of samples.

3) We propose IAT-test and IAT-train scores to evaluate methods on our task, covering three task goals. Experiments prove the satisfying generation effects of our proposed method.

## 2 Related Work

### 2.1 Data Augmentation with GAN

It is widely accepted that in deep learning, a larger dataset often leads to a more reliable algorithm. In practical applications, data augmentation by adding synthetic data provides another way to improve performance. The most common data augmentation strategies are applying various hand-designed transformations on existing data. As GAN arises, it is a straightforward idea to use GAN to directly synthesize realistic data for augmentation. However, GAN itself always requires large-scale data for stable training. Otherwise, the quality of synthesized data is not ensured.

Several variants of conditional GAN are proposed for augmenting classification tasks, where category labels are included in GAN training. ACGAN [4] lets the generator and discriminator 'cooperating' on classification in addition to 'competing' on generation. DAGAN [5] aims to learn transformations on existing data for data augmentation. BAGAN [6] restores the dataset balance by generating minority-class samples. Unfortunately, these methods can not be applied to augmenting data without category labels given. Some other GAN-based data augmentation methods are also designed for different tasks, like [14] for emotion classification and [15, 16] for person re-identification. They are only suitable

for respective tasks but not extensible to our task. Unlike these methods, our proposed method augments IAT data by re-assigning individual actions from existing pairs into new pairs. Data synthesized in this way are undoubtedly natural and realistic. Meanwhile, PE method ensures the same interaction rules on augmented data and existing data, namely the semantic-level reality of augmented data.

### 2.2 Evaluation Metrics for Generation

Early work often relies on subjective visual evaluation of synthesized samples from generative methods like GAN. Quantitative metrics are proposed in recent years, and the most popular among them are Inception score (IS) [17] and Fréchet Inception distance (FID) [18]. Both of them are based on a pre-trained classification network (for image generation, an Inception network pre-trained on ImageNet). IS predicts category probabilities on generated samples through the classification network and evaluates generated results accordingly. FID directly measures the divergence between distributions of real and synthesized data in feature-level. CSGN [19] has extended IS and FID metrics from image generation to skeleton-based action synthesis. However, they fail to reflect the dependence of generated results upon conditions, thus are unsuitable for conditional generation tasks like ours.

GAN-train and GAN-test scores [13] are proposed for comparing category-conditional GANs. An additional classification network is also introduced. Given category information, the two metrics quantify the correlation between generated samples and conditioned categories besides generating reality and diversity. Nonetheless, category labels are missing in our task and semantic categories are implicitly reflected in paired relationships. Enlightened by GAN-train and GAN-test, we propose IAT-test and IAT-train scores to fit our task. In our metrics, binary classification on data pairs is adopted in the classification network instead of explicit multi-category classification on individual instances.

## 3 Proposed Method

Our method consists of two parts: a core Paired-Embedding (PE) method for effective and reliable data augmentation, and an Act2Act network following the former. We illustrate the two parts separately in the following.

### 3.1 Paired-Embedding Data Augmentation

Here we propose a Paired-Embedding (PE) method, which aims to cluster individual action instances in a low-dimension embedding space by utilizing paired relationships between them.

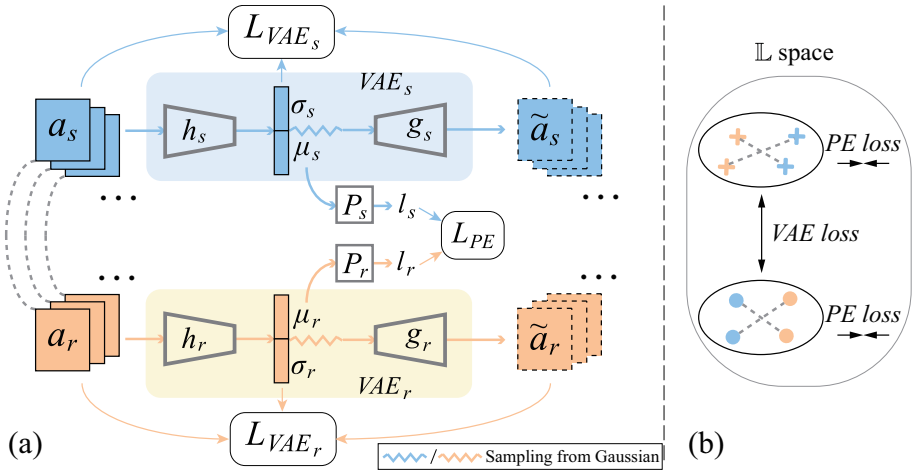


**Fig. 2.** (a) The structure of Paired Variational Auto-Encoders (PVAEs) and losses for training. (b) Effects of different losses.

**Paired Variational Auto-Encoders (PVAEs).** PE is based on a Paired Variational Auto-Encoders (PVAEs) consisting of two separate Variational Auto-Encoder (VAE) [7] networks $VAE_s$ and $VAE_r$ with the same architecture, as shown in Fig. 2(a). Following [7], a VAE network is composed of an encoder $h$ and a decoder $g$. The encoder projects each sample $a$ into ($\mu$, $\sigma$), which are parameters of a multivariate Gaussian distribution $N(\mu, \sigma^2 I)$. Then a latent variable is sampled from this distribution to generate $\widetilde{a}$ through the decoder. Reconstruction error from $\widetilde{a}$ to $a$ and a prior regularization term constitutes VAE loss, i.e.,

$$L_{VAE}(a, \widetilde{a}, \mu, \sigma) = ||a - \widetilde{a}||^2 + \lambda_{KL} D_{KL}(N(\mu, \sigma^2 I)||N(0, I)) \tag{1}$$

where $D_{KL}$ is the Kullback-Leibler divergence, with $\lambda_{KL}$ controlling its relative importance.

We extract individual action instances from original action pairs. The two networks can be respectively trained under VAE loss to model the distribution of stimulative/responsive actions.

**Paired-Embedding (PE) Loss.** Given an action set, the encoder of VAE projects each action into a $\mu$ as the mean of a Gaussian distribution. We collect Gaussian means from all the actions and compute a matrix $P$ for linear dimension reduction, using Principal Component Analysis (PCA) on them. These Gaussian means are further projected by $P$ into an extremely low-dimension embedding space $\mathbb{L}$, namely as $l = P\mu$. Owing to PCA, the variance of Gaussian means is well maintained in the $\mathbb{L}$ space. Both stimulative and responsive actions are projected into the embedding space in this way. For two actions paired in the

**Algorithm 1** Training of PVAEs

**Input:** $\mathbf{A} = \{\dots, (a_s, a_r), \dots\}$
**Output:** $h_s$, $g_s$, $h_r$, $g_r$

Initialize $h_s$, $g_s$, $h_r$, $g_r$
**for** *epoch* in [1, *Epochs*] **do**
  # First step under VAE loss
  $L_{VAE_s} = 0$, $L_{VAE_r} = 0$
  **for** $(a_s, a_r)$ in $\mathbf{A}$ **do**
    $(\mu_s, \sigma_s) = h_s(a_s)$, $(\mu_r, \sigma_r) = h_r(a_r)$
    Sample $z_s \sim N(\mu_s, {\sigma_s}^2 I)$, Sample $z_r \sim N(\mu_r, {\sigma_r}^2 I)$
    $\widetilde{a_s} = g_s(z_s)$, $\widetilde{a_r} = g_r(z_r)$
    $L_{VAE_s}$ += $L_{VAE}(a_s, \widetilde{a_s}, \mu_s, \sigma_s)$, $L_{VAE_r}$ += $L_{VAE}(a_r, \widetilde{a_r}, \mu_r, \sigma_r)$
  **end for**
  Back-prop $L_{VAE_s}$, update $h_s$, $g_s$; Back-prop $L_{VAE_r}$, update $h_r$, $g_r$
  # Second step under PE loss
  $\mathbf{M}_s = \{\}$, $\mathbf{M}_r = \{\}$, $\mathbf{M} = \{\}$, $L_P = 0$
  **for** $(a_s, a_r)$ in $\mathbf{A}$ **do**
    $(\mu_s, \sigma_s) = h_s(a_s)$, $(\mu_r, \sigma_r) = h_r(a_r)$
    $\mathbf{M}_s$.append($\mu_s$), $\mathbf{M}_r$.append($\mu_r$), $\mathbf{M}$.append($(\mu_s, \mu_r)$)
  **end for**
  $P_s$ = PCA($\mathbf{M}_s$), $P_r$ = PCA($\mathbf{M}_r$)
  **for** $(\mu_s, \mu_r)$ in $\mathbf{M}$ **do**
    $l_s = P_s \mu_s$, $l_r = P_r \mu_r$
    $L_P$ += $L_{PE}(l_s, l_r)$
  **end for**
  Back-prop $L_P$, update $h_s$, $h_r$
**end for**

original dataset $\mathbf{A}$, we push them towards each other in the embedding space using a Paired-Embedding (PE) loss, i.e.,

$$L_{PE}(l_s, l_r) = ||l_s - l_r||^2, \tag{2}$$

where $l_s$ and $l_r$ are embeddings of an interactive pair of actions in the $\mathbb{L}$ space. Fig. 2(a) illustrates such a process.

**Training PVAEs.** We train $VAE_s$ and $VAE_r$ synchronously and divide each epoch into two steps, as in Algorithm 1. During the first step, the two networks are independently optimized towards minimizing respective VAE loss. In the second step, PE loss serves to guide encoders in two networks.

Such an alternating strategy drives PVAEs from two opposite directions, as Fig. 2(b) shows.

– On the one hand, Gaussian means should scatter for the reconstruction of different action instances. In other words, Gaussian means must maintain a sufficiently large variance, which is transfered almost losslessly to $\mathbb{L}$ space by PCA. Consequently, the first learning step under VAE loss requires a large variance among $\mathbb{L}$ embeddings of stimulative/responsive actions respectively.

– On the other hand, each defined interaction rule is shared among several action pairs. For these action pairs, semantic category information is unified while other patterns in action instances are diverse. Since $\mathbb{L}$ space has an extremely low dimension, embeddings of paired actions can not be close for all pairs if the space mostly represents patterns apart from semantics. In other words, PE loss pushes the space towards representing semantic categories of actions only. Thus, stimulative or responsive actions within the same semantic category are pulled together in $\mathbb{L}$ space, guided by PE loss.

As a result, actions with similar semantics tend to cluster in the embedding space. Meanwhile, different clusters are far away from each other to maintain large variance. Experimental results in Sec. 4.4 further verify this effect.

**Data Augmentation with PVAEs.** Given a set of individual action instances (either stimulative or responsive) and the corresponding VAE network from trained PVAEs, an $N \times N$ matrix $C$ is computed as,

$$C(i,j) = exp(-\frac{||l^{(i)} - l^{(j)}||^2}{2||s \cdot (P\sigma^{(i)})||^2}), \tag{3}$$

where $N$ is the number of action instances, with $i$ and $j$ indexing two samples. A pre-set scale factor $s$ controls the neighborhood range. After that, we normalize the sum of each row in $C$ to 1, i.e.,

$$NC(i,j) = \frac{C(i,j)}{\sum_{k=1}^{N} C(i,k)}. \tag{4}$$

The computed $NC$ matrix represents confidence in replacing one action with another under defined interaction rules. An action is believed to express semantics similar to other actions in its neighborhood, owing to clustering effects in $\mathbb{L}$ space. We respectively compute two $NC$ matrices for stimulative and responsive action instances and use them to augment action pairs. Two actions from each action pair in the original dataset are replaced with other samples in their respective neighborhood, according to $NC$ matrices. Assume that $N$ data pairs in the original set are evenly distributed in $K$ semantic categories. With replacement, we can optimally attain $\frac{N}{K} \times \frac{N}{K} \times K = \frac{N^2}{K}$ various data pairs conforming to defined interaction rules. Such an increase in data diversity will significantly boost the learning effects of IAT task.

### 3.2 Act2Act: Encoder-Deoder Network under Conditional GAN

IAT is similar to paired image translation in the task form and goals. Both of them can be regarded as an instance-conditional generation task. They differ in that image translation conditions on the structured content of input instance, while our task implicitly conditions on the higher-level semantics of input instance. In recent years, GAN-based methods have been successful in image translation, generating photorealistic results. A similar GAN-based scheme is applied to our task.

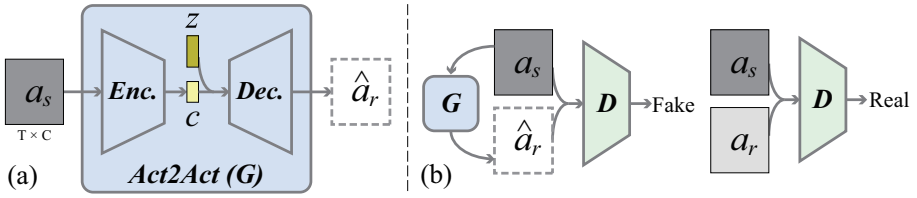


**Fig. 3.** (a) The Act2Act network and (b) training under conditional GAN.

Our Act2Act network is stacked with an encoder-decoder architecture, as in Fig. 3(a). It receives a stimulative action $a_s$ as input, and gives an output $\widehat{a}_r$ with the same form. Through the encoder, a low-dimension code $c$ is extracted from $a_s$. A random noise vector $z$ is sampled from zero-mean Gaussian distribution with unit variance, and then combined with $c$ to decode $\widehat{a}_r$.

Conditional GAN is applied for training, as Fig. 3(b) shows. The encoder-decoder network is treated as Generator $G$, with another Discriminator $D$ receives a combination of two action sequences and outputs a score. Given paired training data $(a_s, a_r)$, $G$ is trained to produce $\widehat{a}_r$ indistinguishable from $a_r$. Meanwhile, $D$ is trained to differentiate $(a_s, \widehat{a}_r)$ from $(a_s, a_r)$ as well as possible.

Behind the above design lies our understanding of IAT task. We consider the task as an implicit series connection of recognition and category-conditional generation. Therefore, we do not introduce $z$ until input is extracted into $c$, unlike in [9, 10] for image translation. The code $c$ has a very low dimension since we expect it to encode high-level semantics. Correlation between $a_s$ and $a_r$ exists only in semantics, but not low-level appearance. Thus the encoder-decoder network is supervised by conditional GAN only, without reconstruction error from $\widehat{a}_r$ to $a_r$.

## 4 Experiments

### 4.1 IAT-test and IAT-train

Inspired by [13], we propose IAT-test and IAT-train scores to evaluate methods on our task, as illustrated in Fig. 4. Besides the training set $\mathbf{A}$ for the task, another set $\mathbf{B}$ composed of individual actions is introduced. Categories of actions in set $\mathbf{B}$ are annotated. Based on annotations, we can pair actions in $\mathbf{B}$ and assign pairs to $\mathbf{B}_{pos}$ or $\mathbf{B}_{neg}$. The former contains action pairs under the same interaction rules as $A$, while the latter contains the rest, as Fig. 4(a1) shows. Given a model $G$ trained on set $\mathbf{A}$, we select stimulative actions from $\mathbf{B}$ and generate responses for them, resulting in paired action set $\mathbf{B}_g$. Fig. 4(a2) illustrates such a process. We evaluate the model $G$ according to $\mathbf{B}_g$ samples in the following ways.

**IAT-test.** With positive samples from $\mathbf{B}_{pos}$ and negative samples from $\mathbf{B}_{neg}$, we train a binary classifier $E$ to judge whether an action pair accords to the

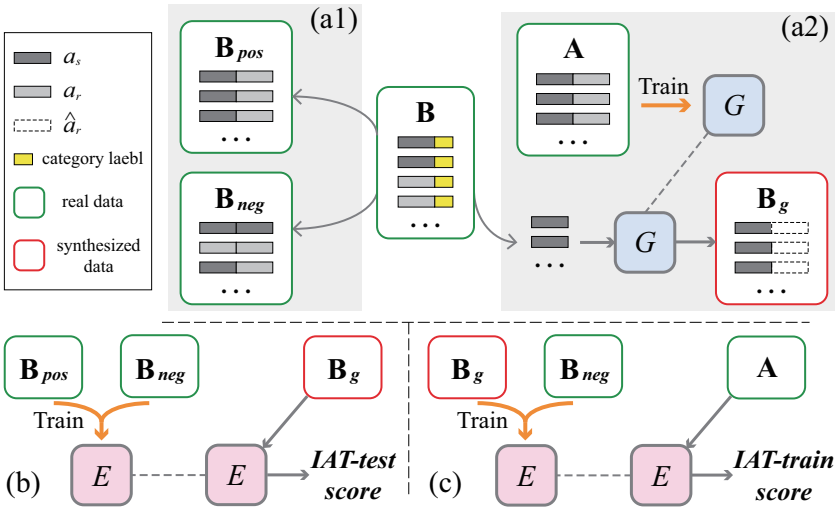


**Fig. 4.** Illustration of our proposed evaluation metrics.

defined interaction rules and give a 1/0 score accordingly. K-fold cross-validation is adopted to investigate and ensure the generalization performance of $E$.

IAT-test is the test score of model $E$ on set $\mathbf{B}_g$, as shown in Fig. 4(b). If $\mathbf{B}_g$ is provided by a perfect model $G$, IAT-test score should approximate the K-fold validation accuracy of model $E$ during training. Otherwise, a lower score can be attributed to: 1) Generated responses are not realistic enough; 2) Semantic translation relationships captured by $G$ are not precise, especially when generalized to stimulative actions in set $\mathbf{B}$. In other words, IAT-test quantifies how well the generation goals of reality and precision are achieved.

**IAT-train.** Here a classifier $E$ similar to the above is trained, with positive samples from $\mathbf{B}_g$ and negative samples from $\mathbf{B}_{neg}$.

IAT-train is the test score of model $E$ on set $\mathbf{A}$, as shown in Fig. 4(c). A low score can appear due to: 1) From unrealistic generation results, $E$ learns features useless for classifying real samples; 2) Incorrect interaction relationships in $\mathbf{B}_g$ misleads the model $E$. 3) Lack of diversity in $\mathbf{B}_g$ impairs the generalization performance of $E$. Overall, IAT-train reflects the achievement of all three goals.

Combining the two metrics helps separate diversity from the other generation goals. In other words, when the model $G$ receives a high IAT-test score and a low IAT-train score, the latter can be reasonably attributed to a poor generation diversity.

### 4.2 Dataset

We evaluate our method on UTD-MHAD [20] and AID [21] datasets, both composed of skeleton-based single-person daily interactive actions. For each dataset,

action categories are firstly paired to form our defined interaction rules, such as "tennis serve - tennis swing", "throw - catch", etc. Then action clips in the dataset are divided into two parts: clips in one part are randomly paired according to interaction rules to form set **A** for learning our task; clips in the other part are reserved as set **B** for evaluation.

**UTD-MHAD** consists of 861 action clips from 27 categories performed by 8 subjects. Each frame describes a 3D human pose with 20 joints of the whole body. We select 10 of 27 action categories and pair them into 5 meaningful interaction rules. Moreover, we choose to use 9 joints of the upper body only since other joints hardly participate in selected actions. Finally, we obtain a set **A** of 80 action pairs and a set **B** of 160 individual action instances.

**AID** consists of 102 long sequences, each containing several short action clips. Each frame describes a 3D human pose with 10 joints of the upper body. After removing 5 corrupted sequences, we have 97 sequences left, performed by 19 subjects and covering 10 action categories. Subsequently, 5 interaction rules are defined on the 10 categories. Finally, we obtain a set **A** of 282 action pairs and a set **B** of 407 individual action instances.

**Implementation Details.** Similar to [22], action data are represented as normalized limb vectors instead of original joint coordinates. This setting brings two benefits. On the one hand, it eliminates the variance of body sizes of subjects in datasets. On the other hand, it ensures that the lengths of human limbs in each generated sequence are consistent.

Action instances (whether at input or output) in our method are $T \times C$ skeleton action sequences. $T$ indicates the temporal length (unified to 32 frames long on both two datasets) and $C$ is the dimension of a 3D human pose in one frame (normally $C = number\ of\ limbs \times 3$). 1D convolutions are performed in our various networks. All GAN-based models in the following experiments are trained under WGAN-GP [23].

### 4.3 Comparison with GAN-based Data Augmentation

As discussed in Sec. 1 and 2.1, GAN-based augmentation methods for classification and other specified tasks can not be applied to our task. Therefore, training an unconditional GAN for directly generating action pairs is left as the only choice for GAN-based data augmentation. We select CSGN [19], which is promising to generate high-quality human action sequences unconditionally. A comparison of data augmentation effects between our PE method and this method is shown in Table. 1.

Learning without augmentation gives generation results that are acceptable from reality and precision (a 85.32/87.29 IAT-test score), but extremely disappointing in diversity (a 53.92/51.17 IAT-train score). For augmentation, a

**Table 1.** Quantitative comparison of data augmentation effects between CSGN and PE.

| | UTD-MHAD | | AID | |
|---|---|---|---|---|
| Data Augmentation | IAT-test | IAT-train | IAT-test | IAT-train |
| – | 85.32 | 53.92 | 87.29 | 51.17 |
| CSGN [19] | 87.86 | 58.97 | 89.96 | 68.82 |
| **PE (Ours)** | **91.03** | **64.94** | **90.69** | **75.65** |

CSGN network is first trained to model the distribution of paired action data. Then we mix generated action pairs with existing data to train our Act2Act network. This method benefits the learning of the task followed, especially visible from a significant increase in IAT-train score. However, it still lags behind our method 3.17/0.73 and 5.97/6.83 respectively in two metrics. We examine generated actions from CSGN and find them to be realistic but not diverse enough, thus provide limited modes for augmentation. Such results keep in line with the fact that GAN-based methods need large-scale training data to ensure multi-modal generation quality. As a comparison, our PE method is more friendly to this small-scale data. Considerable improvements in diversity of generated action responses reflect similar improvements brought by PE in diversity of paired training data.

### 4.4 Ablation Study

**Embedding Space.** Fig. 5 visualizes the distribution of actions in the embedding space, projected by PVAEs trained with/without PE loss. Groundtruth category labels are utilized to color data points for comparison. As can be seen, additional PE loss brings much better clustering effects in both gatherings within categories (especially in Fig. 5(a)) and distances between categories (in Fig. 5(b)).

We analyze two critical hyper-parameters affecting PE data augmentation: the scale factor $s$ and the dimension of $\mathbb{L}$ embedding space $d_{\mathbb{L}}$. Augmentation

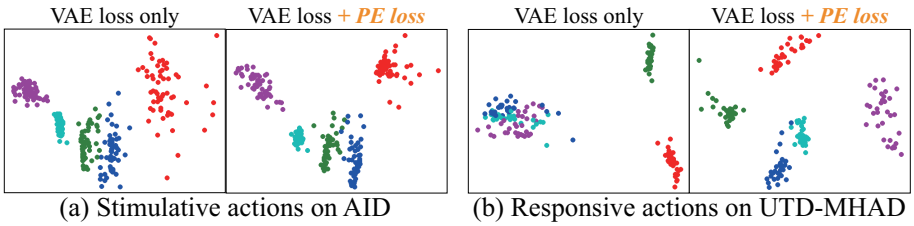


**Fig. 5.** Action embeddings projected by PVAEs trained with VAE loss only and with PE loss also.

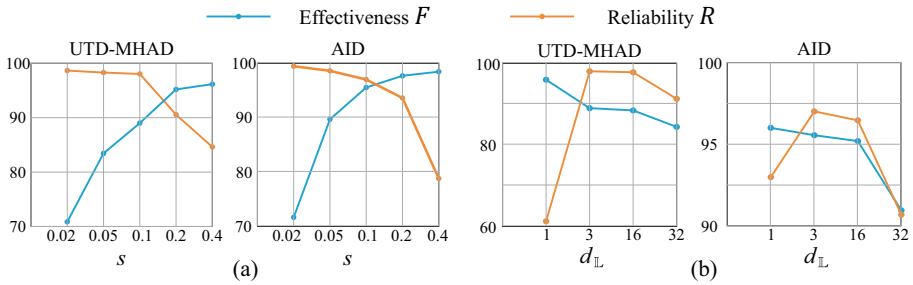


**Fig. 6.** Data augmentation effects with different (a) scale factors and (b) dimensions of $\mathbb{L}$ space.

effects reflected in $NC$ matrices are evaluated from effectiveness $F$ and reliability $R$. Specifically, $F$ is represented as the probability that each sample is replaced by others to form new pairs, i.e.,

$$F = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} 1(i \neq j) \cdot NC(i,j). \tag{5}$$

Meanwhile, we import groundtruth category labels to calculate the probability of category unchanged after replacement as $R$, i.e.,

$$R = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} 1(cat^{(i)} = cat^{(j)}) \cdot NC(i,j), \tag{6}$$

where $cat$ is the category of action.

As the neighborhood range controlled by $s$ expands, the effectiveness of PE data augmentation increases while the reliability decreases. Fig. 6(a) suggests $s = 0.1$ to be the equilibrium point of $F$ and $R$ on both two datasets. Changes brought by different $d_{\mathbb{L}}$ are more complicated. As Fig. 6(b) shows, when $\mathbb{L}$ is a 1-d space, learning PVAEs to cluster actions in it can be difficult. The low reliability reflects relatively weak clustering effects at this time. Then the subtle difference between 3-d and 16-d suggests a very flexible selection range for a reasonable embedding space dimension. When the dimension further increases, augmentation effects start to corrupt, mostly due to the imbalance between PE loss and VAE loss during training PVAEs.

**Comparison with Label-Given Methods.** Here experiments are conducted in label-given situations to give an upper bound of performance of our method:

**1) Re-assign:** Actions are re-assigned into new pairs according to groundtruth labels. All paired relationships conforming to defined interaction rules are exhausted for the training of Act2Act.

**2) Split:** The network is explicitly split into two parts: a classification part for stimulative actions and a category-conditional generation part for responsive

**Table 2.** Quantitative comparison of generation effects between our proposed method and methods in label-given situations.

| | | UTD-MHAD | | AID | |
|---|---|---|---|---|---|
| Data | Label-given | IAT-test | IAT-train | IAT-test | IAT-train |
| Original | × | 85.32 | 53.92 | 87.29 | 51.17 |
| **PE aug.** | × | 91.03 | 64.94 | 90.69 | 75.65 |
| Re-assign | ✓ | 90.97 | 68.93 | 93.05 | 82.15 |
| Split | ✓ | 91.35 | 71.64 | 95.04 | 85.89 |

actions. The two parts are independently trained with category labels given and connected in series during inference.

As Table. 2 shows, methods augmented by PE is very close to label-given methods in performance, compared to the original baseline. With category labels given, we can attain more satisfactory generation results.

### 4.5 Qualitative Evaluation

Generated responses conditioned on some stimulative actions are shown in Fig. 7. Three samples for random noise vector $z$ in Act2Act are involved in each generation. It is surprising that given the same stimulative action, generated responses from our method are various due to randomness from $z$. Such variety of actions manifests in several aspects like pose, movement speed and range. In contrast, generation results from the baseline lack such diversity. Take the "knock - push" interaction for instance. Our generated actions tend to "push" towards various directions, while actions from the baseline seem to place hands always at the same height.

Besides, all generated responses from our method belong to respective categories expected by interaction rules. This indicates that within our method, latent code $c$ in Act2Act precisely controls semantic translation. In addition, human-like vividity shown in these generated actions is impressive. Overall, qualitative evaluation further verifies the effectiveness of our method in meeting all three generation goals. There is still distortion in some instances, like the final generated response in "basketball shoot - baseball swing" interaction. We attribute it to the tiny scale of data in UTD-MHAD and the complexity of "baseball swing" action (in such action, hands may overlap each other).

## 5 Conclusion

In this paper, we specify a novel task to learn end-to-end action interaction and propose a PE data augmentation method to enable learning with small-scale unlabeled data. Another Act2Act network learns from augmented data. Two

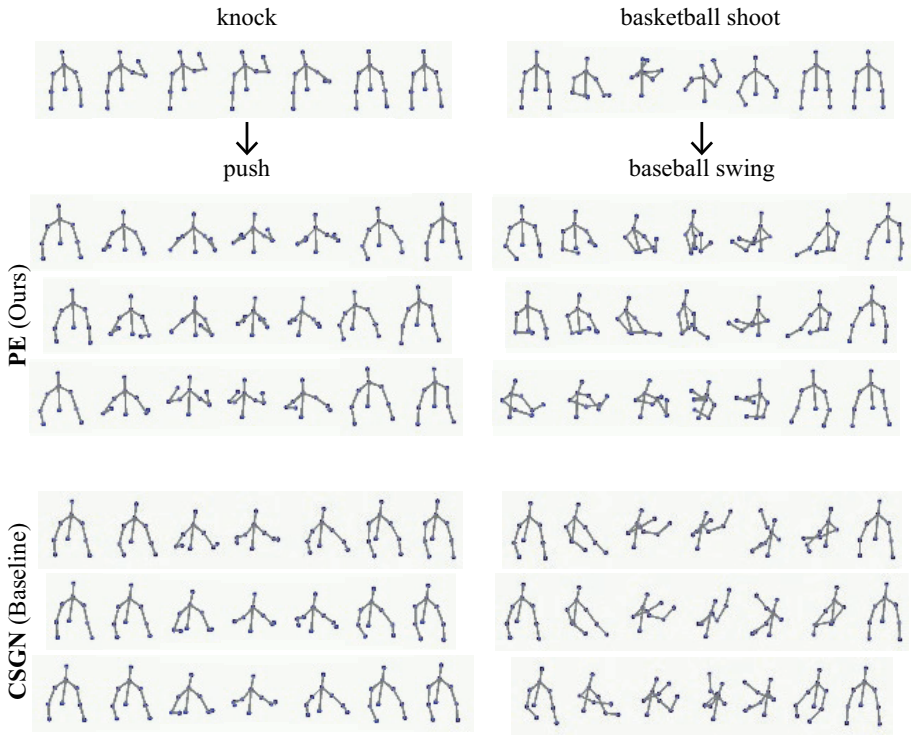


**Fig. 7.** Examples of generation on UTD-MHAD from our PE method and baseline augmented by CSGN [19]. For each example, the given stimulative action and generated responses corresponding to three random noise vectors are shown. Visualized actions are meanly sampled from 32-frame sequences.

new metrics are also specially designed to evaluate methods on our task from generation goals of reality, precision and diversity. Our PE method manages to augment paired action data significantly and reliably. Experimental results show its superiority to baseline and other GAN-based augmentation methods, approximating the performance of label-given methods. Given impressively high-quality action responses generated, our work shows broad application prospects in action interaction. We also hope our PE method to enlighten other unsupervised learning tasks with weak information like paired relationships in our task.

## Acknowledgement

This work was supported by the National Key R&D Program of China (2016YFB 1001001), the National Natural Science Foundation of China (61976170, 91648121, 61573280), and Tencent Robotics X Lab Rhino-Bird Joint Research Program (201902, 201903).

## References


1. Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., Šabanović, S.: Human-Robot Interaction: An Introduction. Cambridge University Press (2020)
2. Ji, Y., Yang, Y., Shen, F., Shena, H., Li, X.: A survey of human action analysis in hri applications. IEEE Transactions on Circuits and Systems for Video Technology (2019)
3. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. CoRR **abs/1406.2661** (2014)
4. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: ICML. (2017)
5. Antoniou, A., Storkey, A.J., Edwards, H.: Data augmentation generative adversarial networks. CoRR **abs/1711.04340** (2017)
6. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, A.C.I.: BAGAN: data augmentation with balancing GAN. CoRR **abs/1803.09655** (2018)
7. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR. (2014)
8. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR **abs/1411.1784** (2014)
9. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017)
10. Zhu, J., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: NIPS. (2017)
11. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. (2016)
12. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. (2018)
13. Shmelkov, K., Schmid, C., Alahari, K.: How good is my gan? In: ECCV. (2018)
14. Zhu, X., Liu, Y., Li, J., Wan, T., Qin, Z.: Emotion classification with data augmentation using generative adversarial networks. In: PAKDD. (2018)
15. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In: ICCV. (2017)
16. Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y., Kautz, J.: Joint discriminative and generative learning for person re-identification. In: CVPR. (2019)
17. Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS. (2016)
18. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS. (2017)
19. Yan, S., Li, Z., Xiong, Y., Yan, H., Lin, D.: Convolutional sequence generation for skeleton-based action synthesis. In: ICCV. (2019)
20. Chen, C., Jafari, R., Kehtarnavaz, N.: UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: ICIP. (2015)
21. Song, Z., Yin, Z., Yuan, Z., Zhang, C., Chi, W., Ling, Y., Zhang, S.: Attention-oriented action recognition for real-time human-robot interaction. CoRR **abs/2007.01065** (2020)
22. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. In: ICRA. (2018)
23. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS. (2017)