Neurocomputing 500 (2022) 712–723

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An imitation learning framework for generating multi-modal trajectories from unstructured demonstrations

Jian-Wei Peng [a], Min-Chun Hu [b], Wei-Ta Chu [a]

[a] Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
[b] Dept. of Computer Science, National Tsing Hua University, Taiwan

ARTICLE INFO

Article history:
Received 6 June 2021
Revised 30 January 2022
Accepted 23 May 2022
Available online 26 May 2022
Communicated by Zidong Wang

Keywords:
Trajectory generation
Motion synthesis
Imitation learning
Reinforcement learning
Generative adversarial networks

ABSTRACT

The main challenge of the trajectory generation problem is to generate long-term as well as diverse trajectories. Generative Adversarial Imitation Learning (GAIL) is a well-known model-free imitation learning algorithm that can be utilized to generate trajectory data, while vanilla GAIL would fail to capture multi-modal demonstrations. Recent methods propose latent variable models to solve this problem; however, previous works may have a mode missing problem. In this work, we propose a novel method to generate long-term trajectories that are controllable by a continuous latent variable based on GAIL and a conditional Variational Autoencoder (cVAE). We further assume that subsequences of the same trajectory should be encoded to similar locations in the latent space. Therefore, we introduce a contrastive loss in the training of the encoder. In our motion synthesis task, we propose to first construct a low-dimensional motion manifold by using a VAE to reduce the burden of our imitation learning model. Our experimental results show that the proposed model outperforms the state-of-the-art methods and can be applied to motion synthesis.

1. Introduction

Trajectory generation is a core problem in many applications, such as motion synthesis [1], sentence generation [2], and autonomous driving [3]. In computer graphics, automatic character motion synthesis and control can help artists create animation or video games more efficiently. Recently, machine learning techniques have been used for trajectory generation and further utilized to develop applications of motion synthesis. The main challenge of trajectory generation is the ability to generate long-term as well as diverse trajectories. Previous works [4,5] have used supervised learning to autoregressively generate the current data conditioned on the data generated previously. However, this kind of method suffers from the error accumulation problem when generating long-term trajectories. Note that the word "long-term" used in this work means the ability to avoid the error accumulation problem.

Reinforcement Learning (RL) has become another popular method for modeling long-term trajectories in recent years. As one type of RL, Imitation Learning (IL) is a reliable approach to learn from expert demonstrations. IL aims to train a policy that can generate trajectories like an expert without knowing the true reward function. GAIL [6] is a recently proposed model-free IL algorithm, which minimizes the distance between the state-action distributions of the policy and the expert. It is an efficient method that can learn from demonstrations directly without estimating the reward function. However, when the expert demonstrations are multi-modal, vanilla GAIL would suffer from the mode missing problem (i.e., fail to capture multiple modalities of expert demonstrations) due to its assumption of a single expert. Some other works borrow the idea from GAN-based models, which have achieved great success in image generation. They propose latent variable models [7,8] to learn multiple modes of demonstrations. However, previous works may still have the mode missing problem in some environments (we show the examples in Section 4.3.1), which means sequence generation is still a problem worth studying.

In this work, we propose an approach to generate long-term motion trajectories that are controllable by a continuous latent variable. Compared with the original GAIL, experiments in previous works [7,9] show that VAEs do not suffer from the mode-dropping problem. This motivates us to combine GAIL with a cVAE to address the multimodality. Similar to VAE-GAN [10], to learn a generalized representation of multi-modal expert demonstrations, we introduce an encoder to encode expert trajectories. We show

E-mail addresses: andersonpen190@mislab.csie.ncku.edu.tw (J.-W. Peng), trimy@mislab.csie.ncku.edu.tw (M.-C. Hu), wtchu@gs.ncku.edu.tw (W.-T. Chu)

https://doi.org/10.1016/j.neucom.2022.05.076
0925-2312/

that a policy, a discriminator, and an encoder can be trained simultaneously with GAIL and cVAE objectives. Moreover, to incentivize the encoder to distinguish different modes, we assume that subsequences of the same trajectory should be encoded to similar locations in the latent space. To achieve this, we propose to add a contrastive loss in the training process of the encoder. For the motion synthesis task, due to the high dimensionality of motion data, we propose to first construct a low-dimensional motion manifold by using a VAE to reduce the difficulty in learning the IL model. This idea is similar to [11], where a deep generative model is trained to learn a continuous pose embedding.

Our experimental results show that the proposed model can successfully separate different modes and generate high-quality long-term trajectories compared with other state-of-the-art methods. Our contributions are summarized as follows.

- We propose a new algorithm of IL for trajectory generation, which can be controlled by a learned high-level latent variable. By assuming that subsequences of the same trajectory would be encoded to similar locations in the latent space, an additional contrastive loss is added in the training phase to successfully learn a generalized and distinguishable representation of data.
- Compared to the state-of-the-art methods, the experiments show that our method learns a better latent representation from unstructured demonstrations. Moreover, it can generate trajectories more generally by interpolating latent variables without the mode missing problem.
- We demonstrate a practical application to human motion synthesis, which is a complex high-dimensional continuous control problem. In our method, novel motions that are unseen in the dataset can be generated and controlled better by a learned latent variable.

2. Related Work

Many applications in computer vision and computer graphics have been formulated as trajectory forecasting problems. Previous works [12,13] introduce many different models to predict future trajectories of pedestrians. Some other works [14,15] consider pose sequences of human motion as high-dimensional trajectories and forecast the future poses from the past poses. Instead of using a deterministic model, some approaches [16,17] use latent variable models to generate diverse trajectories.

2.1. Motion Synthesis

RL with hand-crafted rewards has achieved great success in physics-based character animation [18,19]. However, it requires a delicate design of the reward function to precisely guide the agent for performing the desired skills. Moreover, the designed reward function can only make the agent learn a single mode of the reference motion, which is not suitable for general applications. Our work focuses on generating animation without physics simulation. Behavior Cloning (BC) [20] is an intuitive way to generate human motion sequences [21,22]. Given an input seed sequence, the model trained with the maximum likelihood objective predicts the future sequence directly. However, BC suffers from the exposure bias problem [23] as the prediction error accumulates at each time step. When the training data is insufficient, the predicted trajectory eventually diverges and can no longer be recovered.

2.2. Generative Adversarial Imitation Learning

GAIL is recently a popular IL algorithm in which a policy learns directly from expert demonstrations using adversarial training. The reward function in GAIL is provided by the discriminator, which outputs the probability that the state-action pair is generated by the expert. The policy can be trained to generate trajectories as close to the expert trajectories as possible without any provided hand-crafted reward. Many other works [24–28] aim to learn from demonstrations better based on GAIL. For example, Variational Adversarial Imitation Learning (VAIL) [29] improves GAIL by constraining information flow in the discriminator through an information bottleneck. However, these methods all assume a single expert and thus are not applicable to learn from multi-modal demonstrations.

There are some extensions of GAIL which aim to capture multiple modes of expert demonstrations. [30] extends GAIL with a context variable for multi-behavior policy. Triple-GAIL [31] introduces an auxiliary skill selector to learn an additional skill label. However, they both require ground truth context labels of expert data. InfoGAIL [32,33] both combine GAIL with a latent variable model and maximize the mutual information between the input latent variables and the generated trajectories. Although diverse behaviors can be generated by the agent with different latent variables, the mode missing problem might still occur in some environments. Another recently proposed approach [34] also results in similar performance as InfoGAIL.

2.3. Variational Autoencoder Based Trajectory Generation

VAEs [35] are latent variable generative models that optimize a lower bound of the log-likelihood. It introduces a variational inference method with an approximated distribution of latent variables. Specifically, an encoder and a decoder can be trained jointly by maximizing the Evidence Lower Bound (ELBO) with the reparameterization trick. Previous works [7,36] have used cVAEs to construct trajectory embeddings and generate diverse sequences. However, the methods based on cVAEs also suffer from the exposure bias problem as can be considered as a kind of BC method.

2.4. Motion Manifold

To generalize the limited motion data and reduce the dimensionality of the state and action space, we tend to first construct a smooth and low-dimensional motion manifold. Early works [37,38] have used the Gaussian Process Latent Variable Model (GPLVM) to learn a probability distribution function (PDF) over character poses from motion data. A recent method [11] learns a continuous pose embedding space with a deep generative model. After constructing the motion manifold, the burden in the downstream task can be reduced. We show in our experiments that motion manifold learning is relatively crucial in IL as exploration in high-dimensional state-action space can be very difficult.

Similar to our work, [32,9] also propose to utilize GAIL for trajectory generation. [32] combines GAIL with a posterior approximation network. The variational lower bound of the mutual information between the input latent variables and the generated trajectories is taken as the additional reward signal. Unfortunately, their work may still have the mode missing problem as shown in our experiments. [9] first pre-trains a cVAE jointly with a state encoder, a policy, and a state decoder. Based on the trained cVAE, the policy is further trained using GAIL with latent variables encoded by the state encoder whose parameters are fixed during training. Our method is also constructed based on GAIL and a cVAE; however, we investigate a better design that can simultaneously train the encoder, the policy, and the discriminator from scratch. Moreover, we add a contrastive loss in the training process to ensure that each mode of the demonstrations is separable in the learned continuous latent space.

3. Methodology

3.1. Background of GAIL

We define an infinite-horizon, discounted Markov Decision Process (MDP) with the tuple $(\mathscr{S}, \mathscr{A}, P, r, \rho_0, \gamma)$, where $\mathscr{S}$ is the state space, $\mathscr{A}$ is the action space, $P : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \to \mathbb{R}$ is the transition probability, $r : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ is the reward function, $\rho_0 : \mathscr{S} \to \mathbb{R}$ is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. Let $\pi : \mathscr{S} \times \mathscr{A} \to \mathbb{R}$ denote a stochastic policy and π_E denote the expert policy in which only demonstrations can be accessed. RL aims to maximize the objective of the expected total cumulative reward $E_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. Similar to GAN [39], in GAIL, π aims to imitate π_E by matching their state-action distribution. A policy $\pi(a|s)$ and a discriminator $D(s, a)$ are trained jointly with the following objective function.

$$\mathscr{L}_{adv} = E_{\pi_\theta}[\log D_\omega(s, a)] + E_{\pi_E}[\log(1 - D_\omega(s, a))] - \lambda_H H(\pi_\theta), \tag{1}$$

where $H(\pi_\theta) = E_{\pi_\theta}[-\log \pi_\theta(a|s)]$ is the causal entropy of the policy and λ_H is the weight of the entropy. In practice, the policy is trained by the policy gradient theorem [40] to minimize the objective, while the discriminator is trained to maximize the objective. We refer the readers to [6] for more details of GAIL.

3.2. Overview

Our proposed framework of trajectory generation is illustrated in Fig. 1. We jointly train an encoder, a policy, and a discriminator with both GAIL and cVAE objectives. Based on the assumption that subsequences of the same trajectory would be encoded to similar locations in the latent space, an additional contrastive loss is added in the training phase. Finally, diverse trajectories can be generated by the policy with different latent variables. For the practical application of motion synthesis, we first construct an additional motion manifold with another VAE to reduce the dimensionality of the motion data. The expert motion sequences are encoded to trajectories in the learned manifold to represent the expert demonstrations.

3.3. Problem Formulation

Given a dataset of trajectories $\left\{\left(x_1^{(i)}, \ldots, x_{T_i}^{(i)}\right)\right\}_{i=1}^{N}$, our model is trained to predict trajectories $(\hat{x}_1, \ldots, \hat{x}_T)$ with multiple modes that can be controlled by a latent variable z, where $T > 0$ is a length of a trajectory and can be an arbitrary positive integer even infinite. Generally, x can be any feature vector with dimensionality d, that is, $x \in \mathscr{X} \subseteq \mathbb{R}^d$.

3.4. Multi-modal Trajectory Generation using Imitation Learning

Following the assumption in the previous work [32], we also consider the expert policy as a mixture of experts $\pi_E = \{\pi_E^0, \pi_E^1, \ldots\}$. We define $\tau_E = (s_{1:T}, a_{1:T})$ as an expert trajectory and $\zeta_{k:k+m} = (s_{k:k+m}, a_{k:k+m})$ as a subsequence of τ_E starting from a time step $k \in [1, T-1]$ with a length $m+1$ ($m \in [0, T-k]$). Our goal is to generate plausible and long-term trajectories like every expert by constructing a latent space with diverse high-level embeddings. We propose to use GAIL incorporated with an encoder. Formally, our algorithm jointly trains an encoder $q_\phi(z|\zeta_{k:k+m})$, a policy $\pi_\theta(a|s, z)$, and a discriminator $D_\omega(s, a|z)$ parameterized by ϕ, θ, and ω respectively. Here, z is a continuous embedding vector sampled from the output of the encoder.

GAIL can be conditioned on a feature that represents a goal or a particular behavior [9]. A natural choice of this feature is the embedding z obtained by encoding an expert sub-trajectory through $q_\phi(z|\zeta_{k:k+m})$. Therefore, the objective function of GAIL can be re-written as follows.

$$\mathscr{L}_{adv} = E_{\pi_\theta}[\log D_\omega(s, a|z)] + E_{\pi_E}[\log(1 - D_\omega(s, a|z))] - \lambda_H H(\pi_\theta). \tag{2}$$

As in [9], the reward function for the policy gradient training phase of $\pi_\theta(a|s, z)$ is designed as $-\log(1 - D_\omega(s, a|z))$. In addition, to ensure an encoded latent variable z can represent high-level information of an expert trajectory, we train q_ϕ and π_θ together with the cVAE objective formulated as follows.

$$\mathscr{L}_{vae} = E_{z \sim q_\phi(z|\zeta_{k:k+m})}\left[\sum_{t=1}^{T} \log \pi_\theta(a_t|s_t, z)\right] - \beta D_{KL}(q_\phi(z|\zeta_{k:k+m})||p(z)), \tag{3}$$

where β is the weight of the KL-term and $p(z) = \mathscr{N}(\mathbf{0}, \mathbf{I})$ is a standard normal distribution. We assume that subsequences of the same trajectory should be encoded to similar latent vectors. Therefore, instead of only encoding the whole trajectory τ to the corresponding latent variable, q_ϕ takes any subsequence $\zeta_{k:k+m}$ of the same trajectory as input and π_θ tries to reconstruct the whole trajectory given the encoded latent variable.

However, as shown in our experiments, we observed that the encoder may encode subsequences of the same trajectory to quite different latent points but still satisfy the above objective. To accurately guide the encoder, we introduce an additional loss for training. We use a Siamese network [41] as the encoder and train the

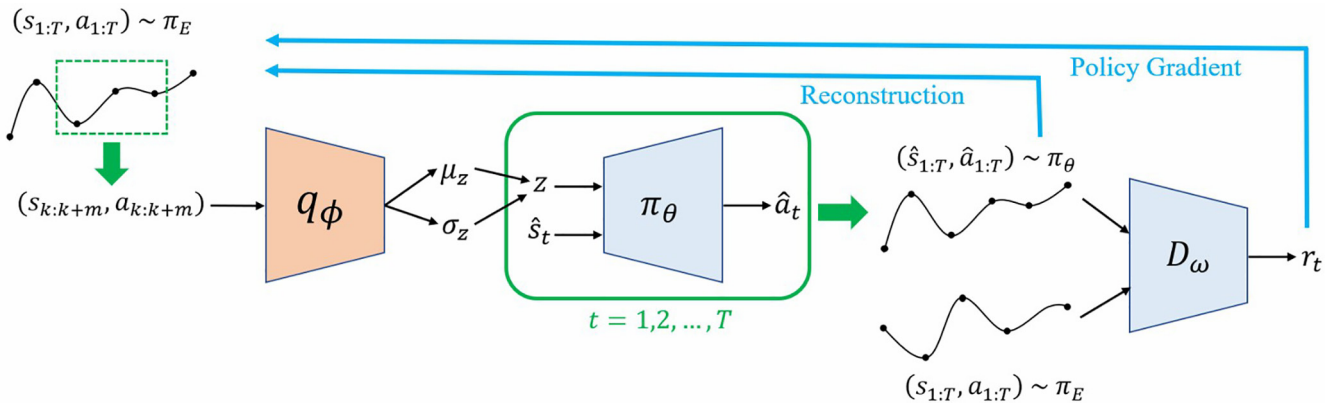

Fig. 1. Overview of the proposed method. GAIL incorporated with an encoder. Our algorithm jointly optimizes the encoder and the policy by using cVAE objective and policy gradient. (Refer to blue arrows) The encoder q_ϕ is a Siamese network trained with an additional contrastive loss.

encoder with a contrastive loss, as shown in Fig. 2. A commonly used contrastive loss function [42] is defined as follows.

$$\mathscr{L}_{sia}(z_1, z_2, Y) = (1-Y)D(z_1,z_2)^2 + Y(\max(0, c - D(z_1,z_2)))^2, \quad (4)$$

where $D(z_1,z_2) = ||z_1 - z_2||_2$ is the Euclidean distance between two latent vectors z_1 and z_2, Y is a binary indicator function denoting whether the two latent vectors belong to the same trajectory or not ($Y = 1$ if they belong to different trajectories), and c is the margin. Intuitively, the contrastive loss makes the distance of embeddings for subsequences of the same trajectories smaller and those of the different trajectories larger. Finally, the model is optimized with the combination of the above three objectives. That is,

$$\max_{\omega} \min_{\phi,\theta} \mathscr{L}(\phi,\theta,\omega) = \mathscr{L}_{adv} + \lambda_1 \mathscr{L}_{vae} + \lambda_2 \mathscr{L}_{sia}, \quad (5)$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the weights of the cVAE loss and the contrastive loss respectively.

3.5. Motion Manifold Construction

For the motion synthesis task, a feature vector representing a pose is usually high-dimensional, which results in the failure of training a model that can mimic expert trajectories (as shown in Section 4.3.4). Using high-dimensional pose representation for IL might cause the policy to collapse to some particular states. To reduce the dimensionality of the state and action spaces, we propose to use a VAE to construct a motion manifold, in which each point represents a specific pose. The training data of the VAE contain poses collected from every frame of motion clips in the dataset. Our VAE learns a mapping between the pose data space and the latent motion manifold. After constructing the motion manifold, motion clips can be encoded to trajectories in the learned manifold by the encoder of the learned VAE. We then take the trajectories on the manifold as the expert demonstrations for IL. Fig. 3 illustrates the concept of motion manifold construction. Since the learned manifold represents poses in a continuous range, our model can generate poses that change continuously. Moreover, by considering a pose encoded to a point outside the desired range of the manifold as an invalid pose, we can have a principled way to avoid the agent from generating unnatural poses in the downstream task.

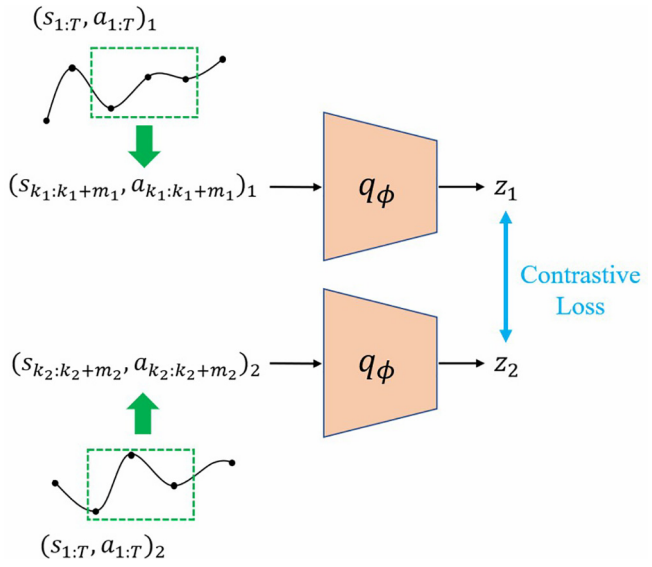

Fig. 2. Siamese architecture of the encoder for computing contrastive loss.

3.6. Implementation Details

In our implementation, VAIL is used for adversarial training and Proximal Policy Optimization (PPO) [43] is used to compute policy gradient. The overall training procedure is shown in Algorithm 1. For the motion generation task, all the joint angles of a pose are concatenated to construct a vector representation x. After training, the learned policy can generate many different trajectories by providing different latent codes. Finally, the generated trajectories are decoded into motion clips.

Algorithm 1: Multi-modal Trajectory Imitation Learning with Contrastive Loss

Input: Initial parameters of policy, discriminator and encoder θ, ω, ϕ; expert trajectories $\tau_E \sim \pi_E$
Output: Learned policy π_θ and encoder q_ϕ
repeat
  for $i = 1, 2, \ldots, N$ **do**
    Sample a sub-trajectory from expert trajectories $\zeta^{(i)}_{t:t+m} \sim \tau_E$ and sample $z^{(i)} \sim q_\phi\left(\cdot|\zeta^{(i)}_{t:t+m}\right)$.
    Run policy $\pi(\cdot|z^{(i)})$ to sample a trajectory $\hat{\tau}^{(i)}$ with $z^{(i)}$ fixed during the rollout.
  end for
  Sample a batch of state-action pairs $\hat{\chi} \sim \hat{\tau}$ and $\chi_E \sim \tau_E$.
  Sample a batch of sub-trajectory pairs from expert trajectories $\left\{\left(\zeta^{(j)}_{t:t+m}, \zeta^{(j)}_{t:t+m'}\right)\right\}_{j=1}^{M} \sim \tau_E$ with binary labels indicating whether they belong to the same trajectory or not.
  Update the discriminator D_ω with respect to ω by using the following objective:

$$\hat{\mathbb{E}}_{\hat{\chi}}[\log D_\omega(s,a|z)] + \hat{\mathbb{E}}_{\chi_E}[\log(1 - D_\omega(s,a|z))]$$

  Jointly update π_θ and q_ϕ with respect to θ and ϕ by using any policy optimization method with the reward $r_t = -\log(1 - D_\omega(s_t, a_t|z))$ and the following objective:

$$\hat{E}_{\hat{\chi}}[\log D_\omega(s,a|z)] - \lambda_H H(\pi_\theta) + \lambda_1 \mathscr{L}_{vae} + \lambda_2 \mathscr{L}_{sia}$$

until convergence

4. Experiments

To evaluate whether the proposed method can learn a continuous low-dimensional latent space that generally represents a concept across all the modes of the expert demonstration, we compared our model with the state-of-the-art methods based on three kinds of environments. The objective is to generate high-quality trajectories that are distinguishable in terms of the mode of the expert demonstration.

4.1. Environments

The experiments were conducted in two synthetic 2D plane environments and one practical motion synthesis environment. Fig. 4 shows the example of the expert trajectories in each environment. Note that the maximum lengths described in the following sub-sections are only used for the training phase. In the inference phase, there is no limitation of the value of the maximum length.

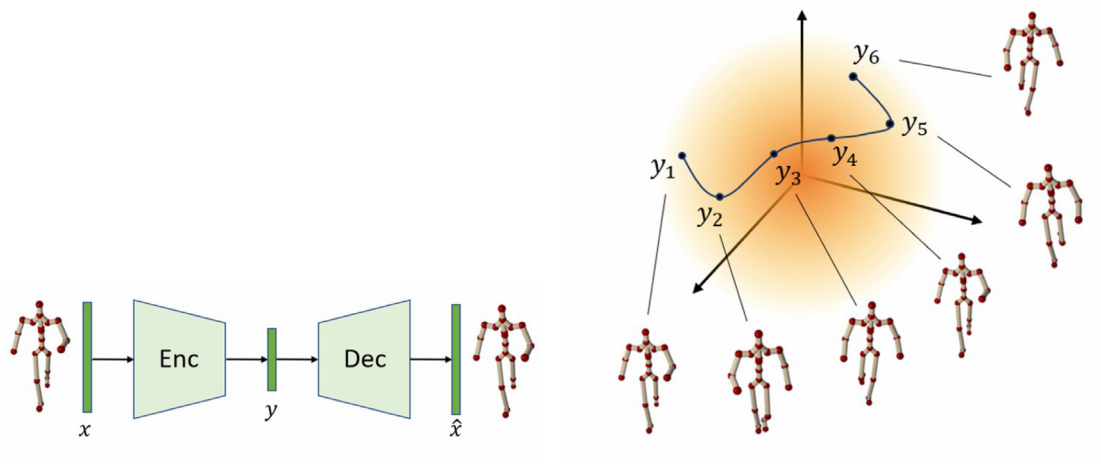

Fig. 3. Motion manifold construction before IL. (a) The proposed VAE model encodes pose data x to latent variables y. (b) After training the VAE, motion clips can be encoded to trajectories in the learned low-dimensional manifold by the encoder of the learned VAE.

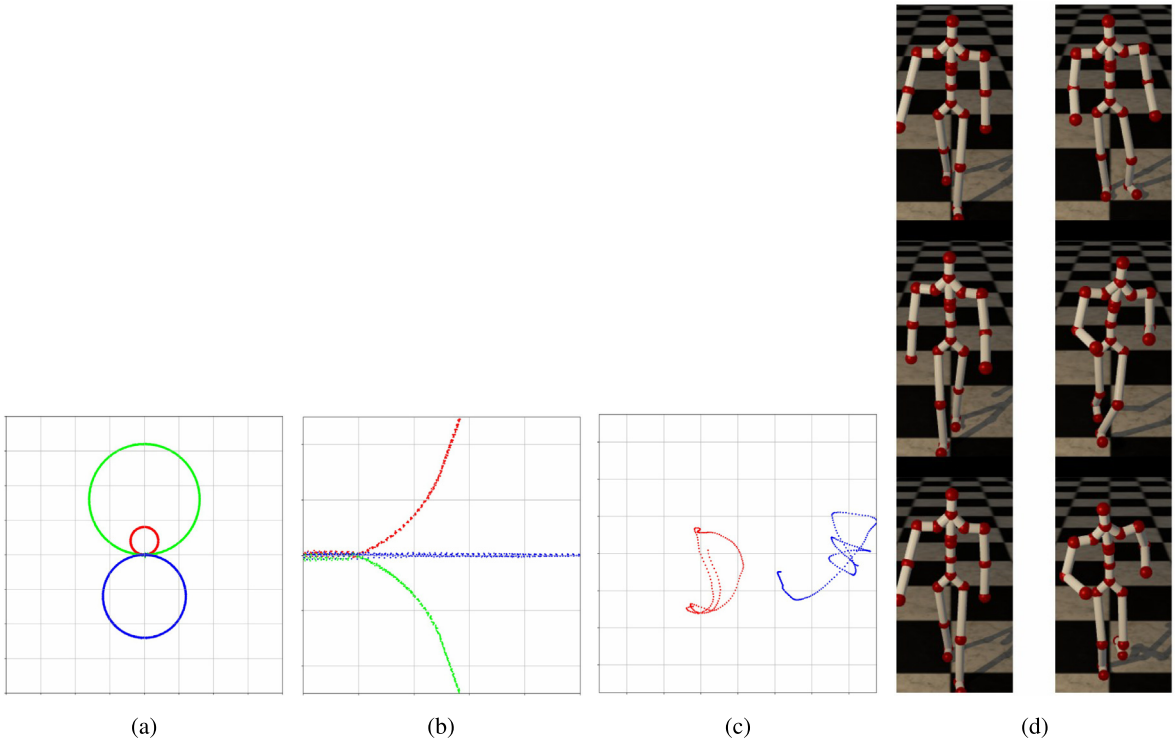

Fig. 4. Expert demonstrations in each environment. (a) Three different modes (red, green, blue) in 2D circle trajectories. (b) Three different modes (red: left, green: right, blue: forward) in 2D traffic trajectories. (c) Two different modes (red: walking, blue: running) in motion synthesis environment. (d) Motion clips used to generate expert trajectories. Left: walking. Right: running.

4.1.1. 2D Circle Trajectories

In this environment, the agent generates a trajectory on a 2D plane. We follow the same settings as in [32], and the state at time t is defined as the positions from time $t-4$ to time t. The action is defined as the moving direction at the current time step with a constant velocity. As in [32], we used three different stochastic expert policies to generate demonstrations with three distinct modes. As shown in Fig. 4a, the demonstration of each mode is a circle-like trajectory with a specific radius and moving direction. In the experiments, we drew 1,024 demonstrations with a maximum length of 256 for each mode.

4.1.2. 2D Traffic Trajectories

In this environment, the agent simulates a vehicle to choose a driving direction. The definition of the state and action is the same as in the 2D circle trajectories environment. We also generated demonstrations with three different modes by using three different stochastic expert policies. As shown in Fig. 4b, the three experts perform turning left, turning right, and going forward respectively. In the experiments, we drew 1,024 demonstrations with a maximum length of 80 for each mode.

4.1.3. Motion Synthesis

We used the motion clips from the AMASS dataset [44] as our raw motion data. The AMASS dataset consists of 11,265 motion clips collected by tracking visual markers placed on humans. All the joint angles of a pose are concatenated to construct a 63-dimensional vector. VAE was used to learn an 8-dimensional motion manifold with all the poses in the dataset. The motion clips were split into segments with a maximum length of 256 and encoded to trajectories in the learned motion manifold, which are treated as expert demonstrations. We use the positions of the generated latent points from time $t-4$ to time t to represent the state at time t. On the other hand, we use the current velocity, which is obtained by subtracting the position at time t from the position at time $t-1$, to represent the action at time t. We drew both 3,974 trajectories of walking and running as two different kinds of expert demonstrations. Fig. 4c shows the trajectories of the two modes on the motion manifold and Fig. 4d shows some key poses of the corresponding motion.

4.2. Baselines

We compared our proposed model with the following baselines:

- **BC:** We used BC to train a single policy with a state as input and action as output. For comparison, the policy is conditioned on a latent variable sampled from a standard normal distribution.
- **cVAE-BC:** We constructed a cVAE with a bi-directional GRU encoder and a policy decoder which is similar to the one used in [7] except that we used a continuous latent variable instead of a discrete one. This model is also taken as a BC method.
- **GAIL:** We implemented GAIL that uses an additional latent variable as a condition. The latent variable is drawn from a Gaussian prior and fed into the policy.
- **InfoGAIL:** We also implemented InfoGAIL, in which the posterior approximation network has the same architectures as our encoder network. Note that the original InfoGAIL paper uses a discrete latent variable while we use a continuous one.

For the implementation of the above baseline methods, the network architecture of the encoders, policies, and discriminators, if any, are all the same as those used in our model. More details of model architectures and hyperparameters are available in the appendix.

4.3. Results

4.3.1. Distinguishing Demonstrations

To demonstrate the ability to distinguish different modes, we compared our model with the baselines by showing the reconstruction results and visualizing the learned latent space. For the reconstruction task, the model first encodes demonstrations of each mode to the corresponding latent codes. The policy then generates the reconstructed results given the latent codes. To visualize the results for the motion synthesis environment, we applied Principal Component Analysis (PCA) to transform the reconstructed trajectories onto a 2D space. For latent space visualization, we drew demonstrations of each mode and generate the corresponding latent codes by the encoder of the learned model, which are then plotted on a 2D plane. Note that due to the lack of an encoder in BC and GAIL, we could not encode demonstrations for the two models. Instead, we simply picked a fixed set of latent codes to generate trajectories.

The reconstruction results are shown in Fig. 5. Please note that in the original papers of cVAE-BC and InfoGAIL, they only show that successful results can be generated by using a discrete latent variable. However, Fig. 5 shows that when the latent variable is continuous, BC and cVAE-BC fail to distinguish all the modes and the reconstructed trajectories are of low quality. Fig. 5 also shows that GAIL can generate more realistic trajectories than BC but has the mode missing problem. Moreover, the lack of an encoder makes it difficult to reconstruct demonstrations in practice. InfoGAIL generates similar results as GAIL and can reconstruct demonstrations with its encoder. However, it also has the mode missing problem in some environments. Our method can not only distinguish all modes in each environment but also reconstruct high-quality trajectories.

The visualization of the learned latent space is shown in Fig. 6. Compared with the baseline models, our encoder tends to encode subsequences of the same trajectories to similar points in the latent space. The distance between data of different modes also increases and thus clusters of different modes can be distinguished clearly.

4.3.2. Learning a Generalized Representation

A good high-level representation should not only memorize the training data but also possess the general concept over the training data. For example, given training data of two different modes, our model should learn a common representation of these two modes and be able to infer the outcomes between them. To evaluate the ability of generalization for the learned latent space, we report the results of interpolation between the two latent codes of different modes.

As shown in Fig. 7, BC fails to learn a smooth latent space and therefore cannot generate intermediate results when interpolating two latent codes. cVAE-BC can somehow generate intermediate outcomes in the synthetic 2D environments while the resulting trajectories are mostly of low quality. GAIL and InfoGAIL both generate plausible results of interpolation. However, the mode missing problem is also reflected in the interpolated results. Our approach can generalize over demonstrations for the complex motion synthesis environment even when the demonstrations are from different modes. Interpolating in the latent space results in smooth interpolation in the data space correspondingly, and the resulting trajectories are still plausible. We also demonstrate the decoded motion clips of interpolation results generated by our model. Fig. 8 shows that our model can generate motions between walking and running by continuously changing the latent variable.

4.3.3. Quantitative Evaluation

In addition to qualitative evaluation, we also conducted quantitative evaluation to measure the quality of the generated trajectories by using the Reconstruction Minimum Squared Error (R-MSE) and Sampling Minimum Squared Error (S-MSE) as in [45]. R-MSE is defined as the squared error of the closest reconstruction to the ground truth, which measures the quality of the reconstruction result. S-MSE is defined as the squared error of the closest sample to ground truth when sampling the latent variables from prior, which measures the quality of the generated trajectories compared with the ground truth. Note that to calculate the R-MSE and S-MSE, we drew 256 samples and used maximum lengths of 80, 256, and 256 for 2D traffic, 2D circle, and motion synthesis environments respectively. For comparison, the error value is further normalized using the minimum and maximum values of the expert demonstrations in each environment.

Table 1 shows the result of R-MSE for the models containing an encoder. The error of cVAE-BC is relatively high because it tends to ignore the condition provided by the encoder. InfoGAIL fails to reconstruct some modes as shown in the previous section. Our model reconstructs trajectories with better quality and achieves lower R-MSE. Table 2 shows the results of S-MSE for all the models. We also provide the model size to contrast our model with other baselines. BC and cVAE-BC suffer from the error accumulation

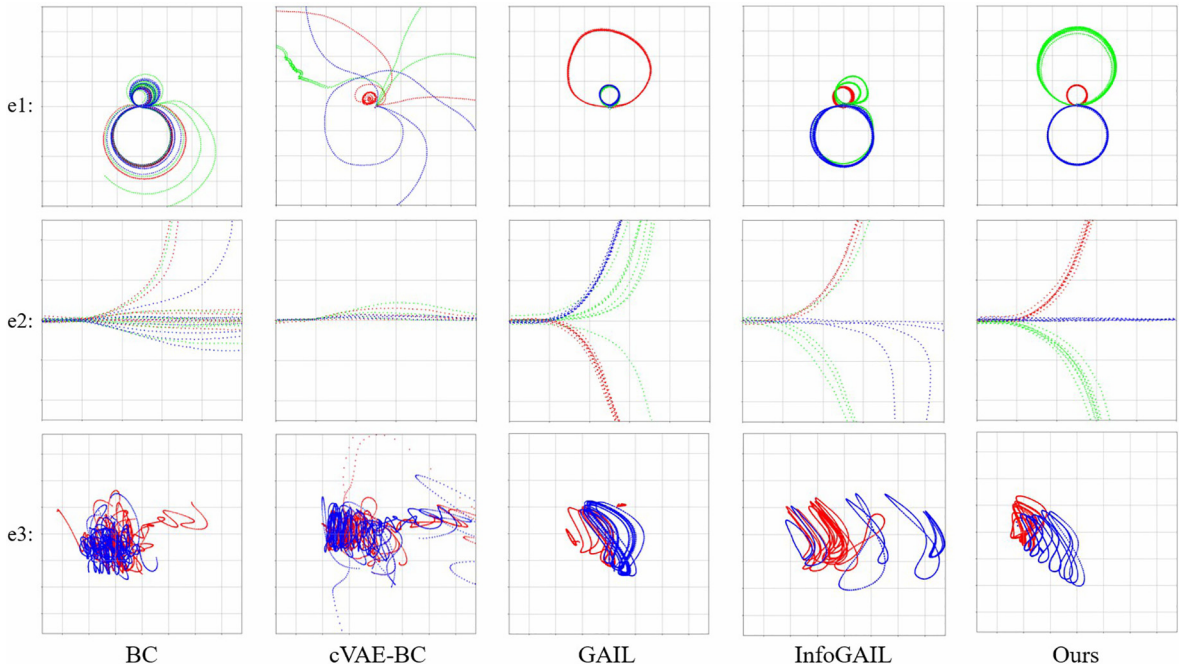

Fig. 5. Reconstruction results in each environment (e1: 2D circle trajectories, e2: 2D traffic trajectories, e3: Motion synthesis environment). Each color denotes specific latent codes for each mode respectively.

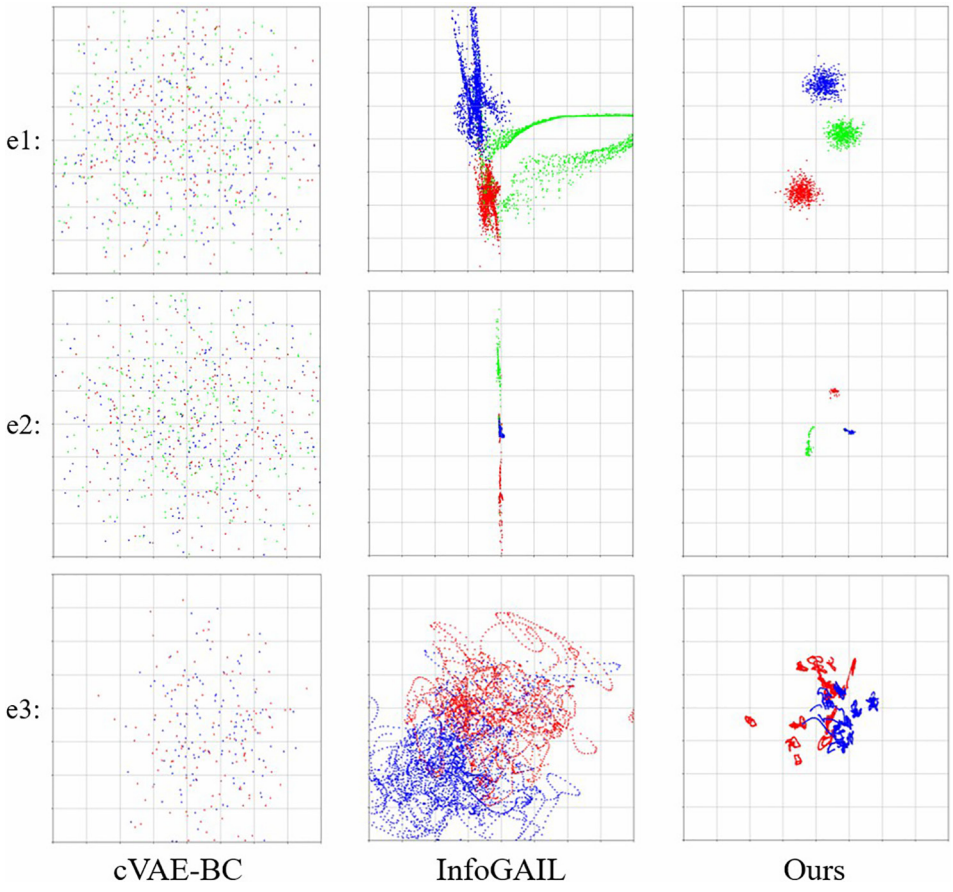

Fig. 6. The learned latent space in each environment. Each color denotes a different mode of expert demonstrations.

problem when generating long sequences. The S-MSE results of GAIL and InfoGAIL are generally lower than those of BC and cVAE-BC. However, they may only generate trajectories of some specific modes as shown in the previous experiment. Our method achieves the lowest S-MSE in all the environments as it learns a generalized representation of the demonstrations.

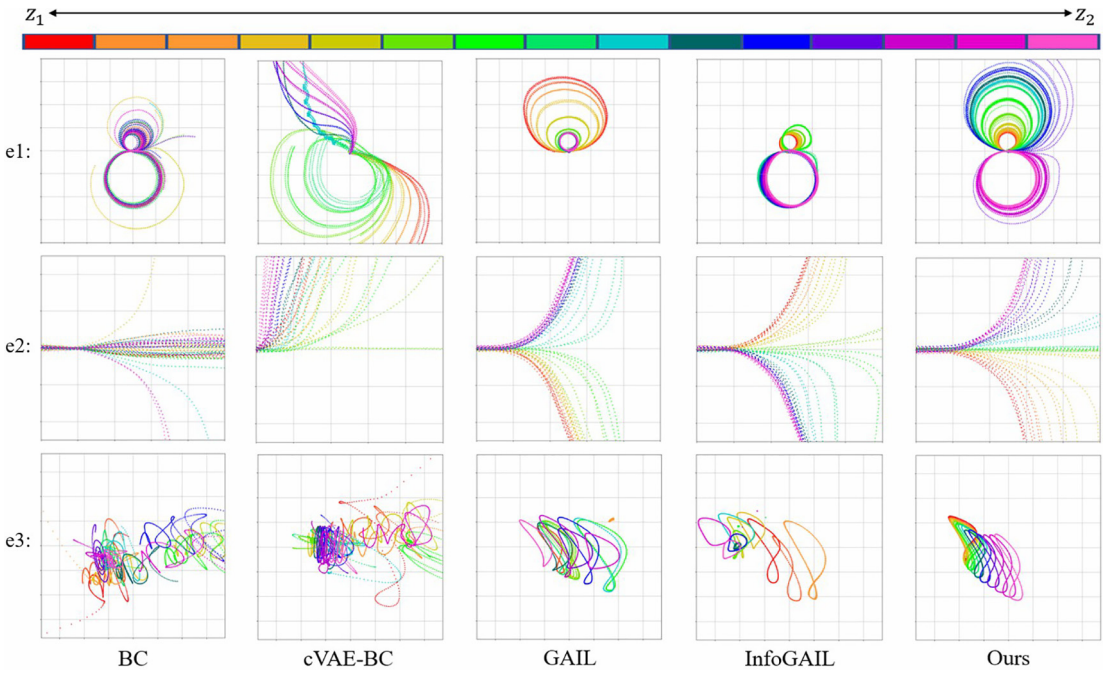

Fig. 7. Interpolation results in each environment. As shown at the top, each color denotes a different linear combination of two latent codes z_1 and z_2.

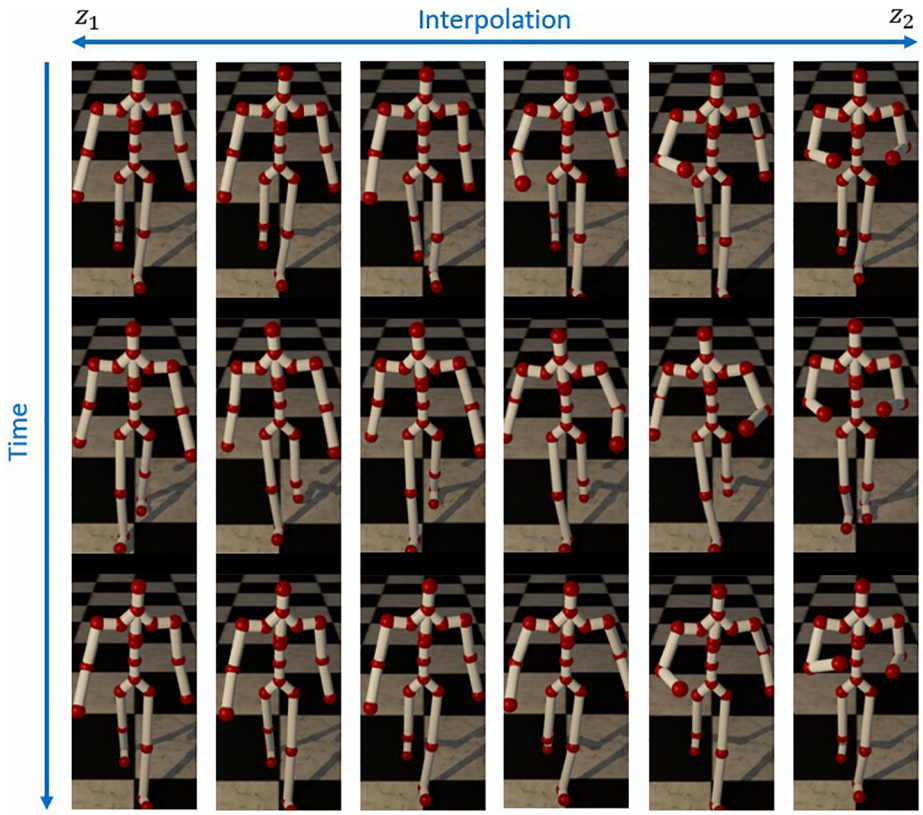

Fig. 8. Decoded motion clips of the interpolation results generated by our model. Each column shows a motion clip generated by our model conditioned on a different linear combination of two latent codes z_1(walking) and z_2(running). We show three frames for each motion clip.

4.3.4. Ablation Study

In the ablation study, we first analyze the effect of the contrastive loss $\mathscr{L}_{sia}$ defined as in Eq. 4. We compare the results of the final proposed model by removing the contrastive loss. Note that if we also remove the cVAE loss $\mathscr{L}_{vae}$, the encoder should also be removed and the model reduces to the original GAIL. The R-MSE and S-MSE results of removing contrastive loss are also listed in Table 1 and Table 2. To further understand the effect of contrastive

Table 1
R-MSE results in each environment. ($\times 10^{-3}$).

Model	2D Circle	2D Traffic	Motion	Motion w/o Manifold
cVAE-BC	54.252	48.814	8.858	11.711
InfoGAIL	2.628	0.109	7.952	284.219
Ours w/o $\mathscr{L}_{sia}$	67.075	1.102	7.245	294.618
Ours	**0.156**	**0.040**	**5.481**	245.541

loss, we also show the corresponding results of interpolation, reconstruction, and latent space (See Fig. 9). When training our model without the contrastive loss, the model collapses to generate the same trajectories of a specific mode, except in the 2D traffic trajectories environment, which is a relatively simple environment. This experiment indicates that our proposed contrastive loss can effectively separate modes in the latent space to help generate diverse trajectories.

On the other hand, we also report the motion synthesis results without constructing a motion manifold beforehand. As shown in the rightmost column of Table 1 and Table 2, without the help of the motion manifold, the R-MSE and S-MSE results are much worse for almost all the models. For BC and cVAE-BC, the evaluated error does not change too much when the motion manifold is not constructed beforehand. We think this is because BC directly minimizes the reconstruction error so that the training process can still be stable. For other GAIL-based models, the training process collapses due to the instability of RL and adversarial training. We found our method trained with the original high-dimensional motion data fails to generate any reasonable motion. Therefore, our proposed motion manifold construction is indispensable to reduce the dimensionality of the motion data and stabilize the training process of IL.

4.4. Limitation and Discussion

Currently, our model can only be used to generate cyclic motions or simple acyclic trajectories such as the 2D traffic trajectories. Our model is still unsuitable to generate complex acyclic motions because the ending state of the multi-modal acyclic demonstrations cannot be determined in RL. In our model, a latent code is randomly sampled and fed into the policy to generate trajectories. The generated trajectory corresponding to the given latent code can be an unseen one in the dataset. For example, in Fig. 7, e2, our model generates "interpolated" trajectories given different latent codes. In this 2D environment, we can simply define ending states as states that the current position reaches the boundary of the 2D plane. However, for the more complex motion synthesis environment, the ending states of the interpolated trajectories cannot be determined since they may not exist in the dataset and may be possibly anywhere (e.g., jumping, squatting, ...). We currently use a finite horizon T and a fixed boundary as our ending condition for cyclic motions (e.g., walking, running, ...).

On the other hand, when training our proposed model, we assume that two sub-trajectories sampled from different trajectories should be dissimilar, but it is possible that the sampled sub-trajectories might still be similar, which might affect the reliability

Table 2
S-MSE results and model sizes in each environment. ($\times 10^{-3}$).

Model	2D Circle	2D Traffic	Motion	Motion w/o Manifold	Model Size
BC	41.856	0.243	12.238	11.704	0.070 MB
cVAE-BC	29.532	8.907	9.429	12.674	1.625 MB
GAIL	22.142	0.544	8.522	186.731	0.209 MB
InfoGAIL	0.699	0.077	7.823	228.180	0.280 MB
Ours w/o $\mathscr{L}_{sia}$	66.970	2.679	8.915	217.066	0.282 MB
Ours	**0.076**	**0.046**	**7.354**	193.486	0.282 MB

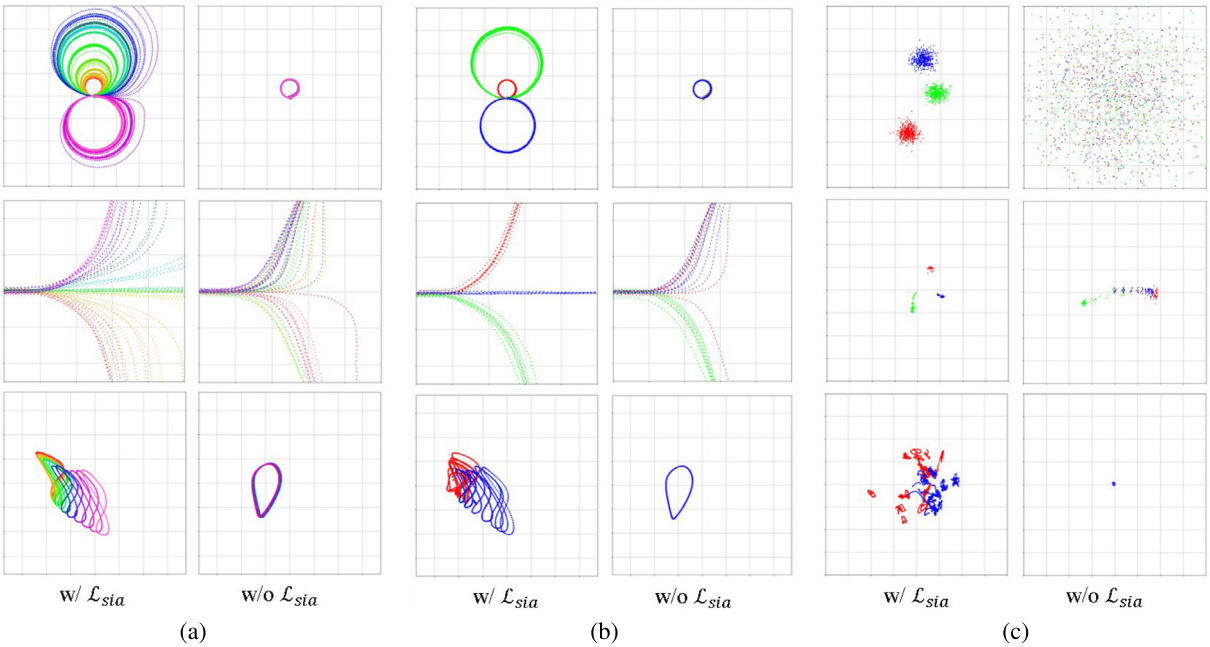

Fig. 9. Comparison between the results of using and not using constrastive loss for ablation study. (a) Interpolation. (b) Reconstruction. (c) Latent space visualization.

of the contrastive loss calculation and cause two similar sub-trajectories to be pushed away from each other in the latent space. Nevertheless, we did not observe this phenomenon in our experiments.

5. Conclusion

We present a method of imitation learning to generate trajectories that are controllable by a continuous latent variable. Our model can learn a generalized representation from expert demonstrations by adding a contrastive loss. The experimental results show that our approach can generate diverse and plausible trajectories and can be applied in practical applications of motion synthesis. In the future, we will emphasize generating acyclic motions and extending the model to deal with text-to-motion tasks.

CRediT authorship contribution statement

Jian-Wei Peng: Conceptualization, Methodology, Validation, Software, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Visualization. **Min-Chun Hu:** Conceptualization, Methodology, Validation, Resources, Writing - review & editing, Supervision, Project administration. **Wei-Ta Chu:** Writing - review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Experimental Details

For motion manifold construction, we trained our VAE model with a batch size of 256 for 32 epochs. For BC and cVAE-BC, we trained the models with a batch size of 256 for 100,000 iterations. For GAIL, InfoGAIL, and our proposed model, we trained the models for 3,000 PPO steps in both the 2D synthetic environments and 10,000 PPO steps in the motion synthesis environment. The detailed settings of PPO hyperparameters are described in Table A.3. For training our proposed model, we used $\beta = 0.005, \lambda_H = 0.001, \lambda_1 = 1.0$, and $\lambda_2 = 0.5$. In all the environments, the dimension of latent vectors is set to 2. We used Adam optimizer for network optimization in all the experiments. All the models were trained on an 8-core machine with an NVIDIA GeForce RTX 2060 SUPER graphics card.

Table A.3
PPO hyperparameters in our experiments.

Hyperparameter	Value
Horizon	128
Clip value	0.2
Adam stepsize	1e-4
Num. epochs	4
Minibatch size	64
Discount(γ)	0.99
GAE parameter(λ)	0.95
Number of actors	8

Appendix B. Model Architectures

In all the baseline models, the network architecture of the encoders, policies, and discriminators, if any, are all the same as those used in our model. We represented the policy as a three-layer fully connected neural network with 128 hidden units and ReLU activation for each layer except for the last one. The policy output a diagonal Gaussian distribution with learned variance. The discriminator was a three-layer fully connected neural network with 128 hidden units and ReLU activation for each layer except for the last one. The output of the discriminator was passed through a sigmoid function. For the RNN encoder used in cVAE-BC, it was a two-layer bidirectional GRU with 128 hidden units.

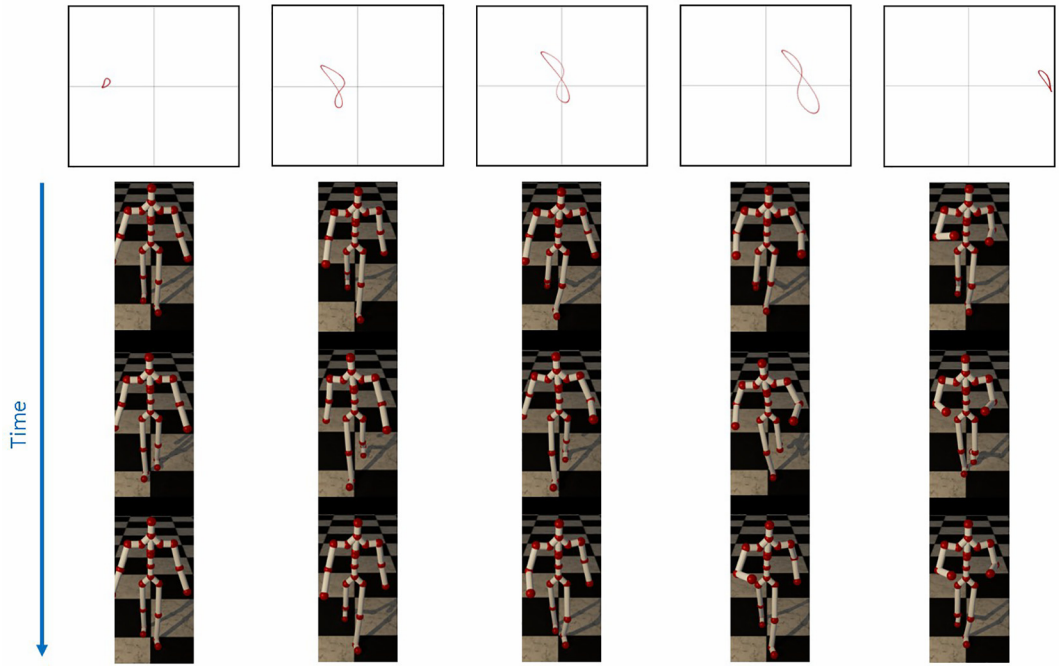

Fig. B.10. Visualization of motion generated by randomly sampling from the latent space of our model. Top: Generated trajectories in the motion manifold. (visualized in 2D) Bottom: Decoded motion.

For the MLP encoder used in InfoGAIL and our model, it was a three-layer fully connected neural network with 128 hidden units and ReLU activation for each layer except for the last one. Both the RNN and MLP encoders output a mean and variance of a Gaussian distribution. When training the policy using PPO, we introduced an additional value network to estimate the value of states. The value network architecture was the same as the discriminator except that the output was not passed through a sigmoid function.

Appendix C. More Qualitative Results

We provide additional visualization of results generated by our model. Fig. B.10 shows different motions generated by randomly sampling from the latent space.

References

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

[2] W. Zhou, T. Ge, K. Xu, F. Wei, M. Zhou, Self-adversarial learning with comparative discrimination for text generation, in: International Conference on Learning Representations, 2020. https://openreview.net/forum?id=B1l8L6EtDS..

[3] N. Rhinehart, R. McAllister, S. Levine, Deep imitative models for flexible inference, planning, and control, in: International Conference on Learning Representations, 2020. https://openreview.net/forum?id=Skl4mRNYDr..

[4] X. Guo, J. Choi, Human motion prediction via learning local structure representations and temporal dependencies, in: AAAI, 2019..

[5] C. Li, Z. Zhang, W.S. Lee, G.H. Lee, Convolutional sequence to sequence model for human dynamics, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5226–5234. doi:10.1109/CVPR.2018.00548..

[6] J. Ho, S. Ermon, Generative adversarial imitation learning, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates Inc, 2016, pp. 4565–4573. https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf..

[7] F. Hsiao, J. Kuo, M. Sun, Learning a multi-modal policy via imitating demonstrations with mixed behaviors, CoRR abs/1903.10304. arXiv:1903.10304. http://arxiv.org/abs/1903.10304.

[8] J.T. Springenberg, K. Hausman, M. Riedmiller, N. Heess, Z. Wang, Learning an embedding space for transferable robot skills, in: International Conference on Learning Representations, 2018..

[9] Z. Wang, J. Merel, S. Reed, G. Wayne, N. de Freitas, N. Heess, Robust imitation of diverse behaviors, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 5326–5335.

[10] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, Vol. 48 of Proceedings of Machine Learning Research, New York, USA, 2016, pp. 1558–1566.

[11] J.N. Kundu, M. Gor, R.V. Babu, Bihmp-gan: Bidirectional 3d human motion prediction gan, in: AAAI, 2019, pp. 8553–8560. https://doi.org/10.1609/aaai.v33i01.33018553.

[12] P. Zhang, W. Ouyang, P. Zhang, J. Xue, N. Zheng, Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12077–12086. doi:10.1109/CVPR.2019.01236..

[13] Y. Xu, Z. Piao, S. Gao, Encoding crowd interaction with deep neural network for pedestrian trajectory prediction, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5275–5284. doi:10.1109/CVPR.2018.00553..

[14] W. Mao, M. Liu, M. Salzmann, H. Li, Learning trajectory dependencies for human motion prediction, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9488–9496. doi:10.1109/ICCV.2019.00958..

[15] J. Martinez, M.J. Black, J. Romero, On human motion prediction using recurrent neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4674–4683. doi:10.1109/CVPR.2017.497..

[16] Y. Yuan, K.M. Kitani, Diverse trajectory forecasting with determinantal point processes, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020, OpenReview.net, 2020. https://openreview.net/forum?id=ryxnY3NYPS..

[17] S.H. Park, G. Lee, J. Seo, M. Bhat, M. Kang, J. Francis, A.R. Jadhav, P.P. Liang, L. Morency, Diverse and admissible trajectory forecasting through multimodal context understanding, in: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (Eds.), Computer Vision - ECCV 2020–16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI, Vol. 12356 of Lecture Notes in Computer Science, Springer, 2020, pp. 282–298. doi:10.1007/978-3-030-58621-8_17. doi: 10.1007/978-3-030-58621-8_17..

[18] K. Bergamin, S. Clavet, D. Holden, J.R. Forbes, Drecon: Data-driven responsive control of physics-based characters 38 (6). doi:10.1145/3355089.3356536. doi: 10.1145/3355089.3356536..

[19] X.B. Peng, P. Abbeel, S. Levine, M. van de Panne, Deepmimic: Example-guided deep reinforcement learning of physics-based character skills, ACM Trans. Graph. 37 (4). doi:10.1145/3197517.3201311. doi: 10.1145/3197517.3201311..

[20] M. Bain, C. Sammut, A framework for behavioural cloning, in: Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995], Oxford University, GBR, 1999, p. 103–129..

[21] S. Starke, H. Zhang, T. Komura, J. Saito, Neural state machine for character-scene interactions, ACM Trans. Graph. 38 (6). doi:10.1145/3355089.3356505. doi: 10.1145/3355089.3356505..

[22] S. Starke, Y. Zhao, T. Komura, K. Zaman, Local motion phases for learning multi-contact character movements, ACM Trans. Graph. 39 (4). doi:10.1145/3386569.3392450. doi: 10.1145/3386569.3392450..

[23] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled sampling for sequence prediction with recurrent neural networks, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, MIT Press, Cambridge, MA, USA, 2015, p. 1171–1179..

[24] W. Jeon, S. Seo, K.-E. Kim, A bayesian approach to generative adversarial imitation learning, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates Inc, 2018, pp. 7429–7439. https://proceedings.neurips.cc/paper/2018/file/943aa0fcda4ee2901a7de9321663b114-Paper.pdf..

[25] V. Tangkaratt, B. Han, M.E. Khan, M. Sugiyama, Variational imitation learning with diverse-quality demonstrations, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, Vol. 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 9407–9417. http://proceedings.mlr.press/v119/tangkaratt20a.html.

[26] F. Yang, A. Vereshchaka, Y. Zhou, C. Chen, W. Dong, Variational adversarial kernel learned imitation learning, Proceedings of the AAAI Conference on Artificial Intelligence 34 (04) (2020) 6599–6606, https://doi.org/10.1609/aaai.v34i04.6135.

[27] K. Kim, H.S. Park, Imitation learning via kernel mean embedding, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, AAAI Press, 2018, pp. 3415–3422. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16807..

[28] B. Wang, E. Adeli, H. Chiu, D. Huang, J.C. Niebles, Imitation learning for human pose prediction, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7123–7132. doi:10.1109/ICCV.2019.00722..

[29] X.B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, S. Levine, Variational discriminator bottleneck: Improving imitation learning, inverse RL, and GANs by constraining information flow, in: International Conference on Learning Representations, 2019. https://openreview.net/forum?id=HyxPx3R9tm..

[30] J. Merel, Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, N. Heess, Learning human behaviors from motion capture by adversarial imitation, CoRR abs/1707.02201. arXiv:1707.02201. http://arxiv.org/abs/1707.02201.

[31] C. Fei, B. Wang, Y. Zhuang, Z. Zhang, J. Hao, H. Zhang, X. Ji, W. Liu, Triple-gail: A multi-modal imitation learning framework with generative adversarial nets, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 2929–2935, main track. doi:10.24963/ijcai.2020/405. https://doi.org/10.24963/ijcai.2020/405..

[32] Y. Li, J. Song, S. Ermon, Infogail: Interpretable imitation learning from visual demonstrations, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 3815–3825.

[33] K. Hausman, Y. Chebotar, S. Schaal, G. Sukhatme, J.J. Lim, Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates Inc, 2017, pp. 1235–1245. https://proceedings.neurips.cc/paper/2017/file/632cee946db83e7a52ce5e8d6f0fed35-Paper.pdf..

[34] M. Sharma, A. Sharma, N. Rhinehart, K.M. Kitani, Directed-info GAIL: Learning hierarchical policies from unsegmented demonstrations using directed information, in: International Conference on Learning Representations, 2019. https://openreview.net/forum?id=BJeWUs05KQ..

[35] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014. arXiv:http://arxiv.org/abs/1312.6114v10..

[36] N. Mishra, P. Abbeel, I. Mordatch, Prediction and control with temporal segment models, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, International Convention Centre, Sydney, Australia, 2017, pp. 2459–2468.

[37] K. Grochow, S.L. Martin, A. Hertzmann, Z. Popović, Style-based inverse kinematics 23 (3) (2004) 522–531, https://doi.org/10.1145/1015706.1015755.

[38] S. Levine, J.M. Wang, A. Haraux, Z. Popović, V. Koltun, Continuous character control with low-dimensional embeddings, ACM Trans. Graph. 31 (4). doi:10.1145/2185520.2185524. doi: 10.1145/2185520.2185524..

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates Inc, 2014. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf..
[40] R.S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: S. Solla, T. Leen, K. Müller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, MIT Press, 2000, pp. 1057–1063. https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf..
[41] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 539–546 vol. 1. doi:10.1109/CVPR.2005.202..
[42] S. Dey, A. Dutta, J.I. Toledo, S.K. Ghosh, J. Lladós, U. Pal, Signet: Convolutional siamese network for writer independent offline signature verification, CoRR abs/1707.02131. arXiv:1707.02131. http://arxiv.org/abs/1707.02131.
[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR abs/1707.06347. arXiv:1707.06347. http://arxiv.org/abs/1707.06347.
[44] N. Mahmood, N. Ghorbani, N. Troje, G. Pons-Moll, M.J. Black, Amass: Archive of motion capture as surface shapes, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 5441–5450.
[45] X. Yan, A. Rastogi, R. Villegas, K. Sunkavalli, E. Shechtman, S. Hadap, E. Yumer, H. Lee, Mt-vae: Learning motion transformations to generate multimodal human dynamics, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

Jian-Wei Peng received the B.S. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 2017. Since 2017, he has been pursuing the Ph.D. degree in the Multimedia Information System Lab, Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests include computer graphics, computer vision, and machine learning.

Min-Chun Hu is also known as Min-Chun Tien and Ming-Chun Tien. She received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, in 2011. She is now an Associate Professor in the Department of Computer Science, National Tsing Hua University, Taiwan. From 2012 to 2017, she was an assistant professor in the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, and she was promoted as an associate professor in 2018. She has published more than 90 papers in international journals and conferences. She was awarded the Exploration Research Award from Pan Wen Yuan Foundation, the Outstanding Youth Award from the Computer Society of the Republic Of China (CSROC), and the Best Young Professional Member Award of IEEE Tainan Section in 2015, 2017, and 2018, respectively. Her research interests include digital signal processing, multimedia content analysis, machine learning, computer vision, computer graphics, virtual reality and augmented reality.

Wei-Ta Chu received the B.S. and M.S. degrees in Computer Science from National Chi Nan University, Taiwan, in 2000 and 2002, and received the Ph.D. degree in Computer Science from National Taiwan University, Taiwan, in 2006. He was a Professor in National Chung Cheng University from 2007 to 2019. He is now a Professor in the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. His research interests include multimedia analysis, computer vision, and deep learning. He won the Best Full Technical Paper Award in ACM Multimedia 2006. He was awarded Outstanding Youth Electrical Engineer Award by the Chinese Institute of Electrical Engineering in 2017, the Distinguished Alumni Award presented by National Chi Nan University in 2014, Best GOLD Member Award presented by IEEE Tainan Section in 2013, the K. T. Li Young Researcher Award presented by Institute of Information & Computing Machinery in 2012, and the Young Faculty Awards presented by National Chung Cheng University in 2011. He was a visiting professor at Nagoya University in 2017 and Columbia University in 2008. He was an associate editor of IEICE Transactions on Information and Systems from 2016 to 2020. He serves as a TPC of IEEE ICME 2022, ACM ICMR 2021, MMM 2020, and IEEE MMSP 2019; a Publication Co-Chair of ACCV 2020, IEEE VCIP 2018, and ICS 2016.