Learning Uncoupled-Modulation CVAE for 3D Action-Conditioned Human Motion Synthesis

Chongyang Zhong[1,2], Lei Hu[1,2], Zihao Zhang[1,2], and Shihong Xia[1,2]✉

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
{zhongchongyang, hulei19z, zhangzihao, xsh}@ict.ac.cn

Abstract. Motion capture data has been largely needed in the movie and game industry in recent years. Since the motion capture system is expensive and requires manual post-processing, motion synthesis is a plausible solution to acquire more motion data. However, generating the action-conditioned, realistic, and diverse 3D human motions given the semantic action labels is still challenging because the mapping from semantic labels to real motion sequences is hard to depict. Previous work made some positive attempts like appending label tokens to pose encoding and performing action bias on latent space. However, how to synthesize diverse motions that accurately match the given label is still not fully explored. In this paper, we propose the Uncoupled-Modulation Conditional Variational AutoEncoder(UM-CVAE) to generate action-conditioned motions from scratch in an uncoupled manner. The main idea is twofold: (i)training an action-agnostic encoder to weaken the action-related information to learn the easy-modulated latent representation; (ii)strengthening the action-conditioned process with FiLM-based action-aware modulation. We conduct extensive experiments on the HumanAct12, UESTC, and BABEL datasets, demonstrating that our method achieves state-of-the-art performance both qualitatively and quantitatively with potential applications.

Keywords: Human motion synthesis, Action-conditioned synthesis, CVAE, Uncoupled modulation

1 Introduction

With the development of the movie and game industry, a growing amount of motion data is required to achieve more life-like animation. To acquire the motion data, one straightforward method is to use motion capture technology. However, it is not feasible to capture all kinds of motion data because, on some occasions, rather than pre-record the motion data, the motion data is required to follow the users' intention. To solve this problem, motion synthesis is a plausible way. Especially, we focus on generating motion sequences based on the user-specific semantic action labels in this problem.

Though motion synthesis has emerged as a powerful solution for acquiring motion data, the problem of generating motion sequences based on given semantic action labels is still challenging. The reasons are mainly three-fold: (i) Firstly,

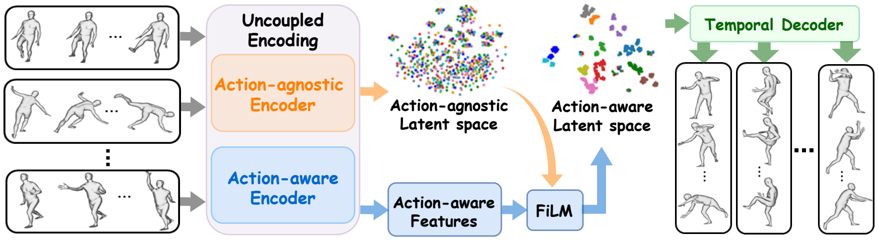


Fig. 1. Overview. We assume that a piece of white and blank paper can be drawn well, which means that it is easier to learn something as an "unskilled kid" than as an adult. Thus we perform uncoupled encoding to learn the action-agnostic latent space as the "unskilled kid" and the action-aware features as the "action skills". In the condition process, we compute FiLM parameters from the action-aware features to "teach" the action-agnostic latent representation how to perform action skills.

it is hard to learn an appropriate correspondence between motion sequences and semantic action labels, which can ensure the generated motions meet the label constraints. For example, previous work [10] tries to build the correspondence on the single frame, but the generated results may suffer from discontinuity. Petrovich et al. [30] embed labels into transformer token and train a transformer-VAE to encode the entire motion sequence and label token coupledly, then add action bias in latent representation to strengthen the action constraint. However, we argue that this coupled manner will make the learned latent representation contain more or less label-related information, which may lead to conflict with that contained in the condition labels (discussed in "Qualitative comparison" of Sec. 4.2). Imagine when you teach a person who has already learned boxing to play golf, the subconscious boxing skills in his mind may distract him from learning golf. (ii) Secondly, the spatio-temporal properties in human motion are difficult to learn, especially when dealing with complex motions within diverse action categories. While the prior works [10,30] consider temporal modeling of motion dynamics, they directly input the original motion representation into the network without considering the spatial relationship between body joints. (iii) Lastly, the generated motion requires diversity, realism, and continuity.

To solve the above problems, we introduce the UM-CVAE to learn the correspondence between action labels and motion sequences in an uncoupled manner. The main idea is twofold: (i) *reducing* the action-related information in latent representation; (ii) *extracting* spatio-temporal action-aware features and *strengthening* the action-conditioned process via FiLM.

Specifically, as shown in Fig. 1, we first encode the motion sequence into *action-agnostic*(which means easy-modulated) latent space via an action-agnostic encoder. For the *action-aware* features extraction, we perform label-motion fusion and train an extra encoder to extract the action-aware spatio-temporal features. Unlike direct concatenation or bias in previous work [10,30], we use FiLM [29] instead to obtain an action-aware latent space. Finally, the variable-length action-conditioned motion sequences are generated through temporal decoder.

Extensive experiments are conducted on HumanAct12, UESTC, and BABEL datasets. We demonstrate that our method achieves significant improvement over the state-of-the-art works both qualitatively and quantitatively. The main contributions of our work can be summarized as follow:

1. We introduce UM-CVAE, a novel sequence-level CVAE to learn the latent representation in an uncoupled manner, which makes the generated motions conform better to the given action labels;
2. We learn the action-aware features through spatio-temporal extraction and utilize FiLM as the modulation method in action-conditioned motion generation, making the condition process more reasonable and powerful;
3. We carry out extensive experiments on HumanAct12, UESTC and BABEL to demonstrate that our method outperforms state-of-the-art works and has potential applications.

2 Related works

In this section, we briefly review related works, including research on diverse motion prediction, constrained motion synthesis, and sequence-level VAE.

2.1 Diverse motion prediction

Human motion prediction aims to predict the human motion in the future period from a given historical motion sequence, which we refer to here as motion synthesis conditioned on the historical sequence. From traditional statistical methods [4,39] to deep neural networks based works like RNN [27,14,44], GCN [26,8,34,45], VAE [38,42] and GAN [3,17], exciting progress has been made in motion prediction. Among these works, diverse human motion prediction based on generative models is more relevant to our work. Using past sequences as conditions, CVAE-based works [38,42] build a probability model on the existing motion data to predict a variety of results through sampling and prediction. Moreover, GAN-based works [17,3] are conditioned on the latent space modeled by standard normal distribution to generate various motions and use GAN to optimize the quality of the prediction. In addition, some other condition ways include conditioned on contextual cues and interaction with objects [7], conditioned on music [20], etc. Unlike these works, the problem we want to solve is conditioned on action labels to generate motion sequences without any initial frame or past sequence.

2.2 Constrained motion synthesis

Generating human motion that meets user constraints has always been a challenging problem. Depending on the type of the given constraint, we divide the related work into content-constrained and semantic-constrained motion synthesis.

Content-constrained motion synthesis Content constraints refer to motion content information, such as velocity, direction, joint trajectory, etc. Through training an unconstrained generative model using RNN [40] or TCN [13] and then performing an optimization-based approach to constrain and edit the motion generation, some works use a two-stage method to generate content-constrained motion, which is too time-consuming to realize real-time generation. To solve this problem, researchers directly parameterize the control signal as the input of the generative model [12,43,35,36], which is equivalent to making an unconstrained generative model conditioned on the control parameters. This class of work relies on sufficiently informative motion representations and achieves high-quality locomotion generation. Unlike the generative methods, another kind of work is named ”motion matching” [6,11], which generates motion sequences by searching the animation database based on users-input in real-time to find the most appropriate next frame or next clip. These works generate high-quality motions but badly depend on the quantity and quality of the dataset while being computationally intensive.

Semantic-constrained motion synthesis Rather than specific content constraints, sometimes users prefer to generate motions using semantic constrain like action labels, styles, descriptive sentences, and music, etc, which can help users without professional skills get the data more conveniently. Text2Action [1]and Language2Pose [2] investigate how to generate a motion sequence from a text description, and some other work generates motion conditioned on music [18,19] or styles [28,41]. Some works focus on more detailed aspects, generating motion from semantic action labels. DVGAN [21] uses a language model to encode the action labels and generates motion by RNN and CNN, and GAN is used to improve the realism of the generation. Their work cannot generate high-quality motions due to the difficulty of learning the correspondence between labels and sequences. Recently, A2M [10] and ACTOR [30] propose to use the CVAE-based framework, which not only performs coupled encoding on labels and motion sequences but also treats action labels as conditions to modulate the latent space, resulting in natural and diverse generated motions.

While most previous works learn the label-sequence correspondence in a coupled manner, we observe that the learned label-related features in latent representation and that in the given condition-label may conflict with each other during the condition process. Therefore, we perform uncoupled encoding to learn the action-agnostic latent space and the action-aware features, then adopt FiLM as the condition method instead of concatenation or bias to perform a more powerful action modulation.

2.3 Sequence-level VAE

Action labels usually correspond to the whole motion sequence rather than a single frame, so to better learn the correspondence between labels and sequences, we use sequence-level VAE. The main problem of sequence-level VAE is how to

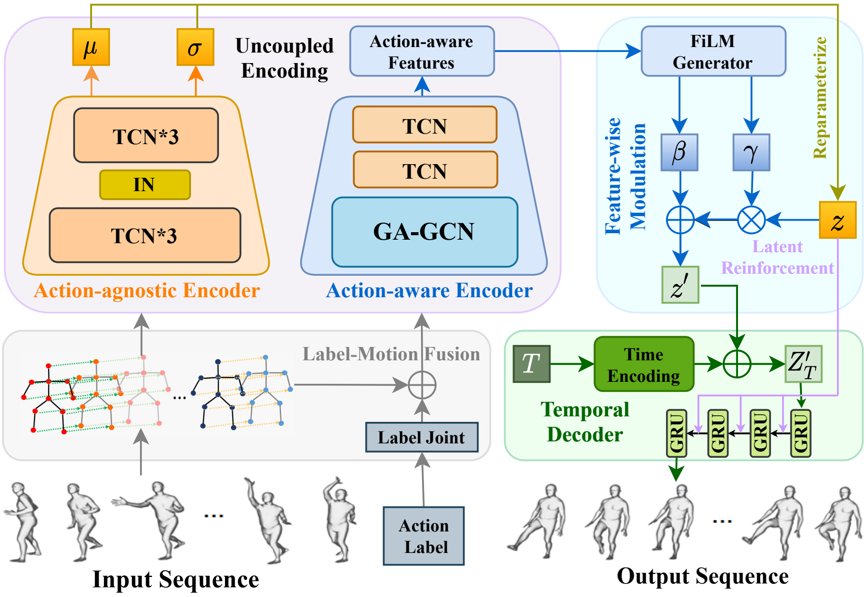


Fig. 2. UM-CVAE architecture. We encode the motion sequence into action-agnostic latent space via TCN and employ instance normalization to weaken action-related information. To strengthen the action-conditioned process, we learn the action-aware spatio-temporal features from action labels and motion sequences through label-motion fusion and spatio-temporal modeling. Then we use FiLM to compute modulation parameters γ and β to perform linear modulation on the learned latent representation. Finally, the GRU decoder is used to generate variable-length motion sequences in conjunction with temporal encoding and latent reinforcement.

learn the mappings in the sequence-latent-sequence transformation. To solve this problem, [38] and [42] use GRU to encode and recover sequences, embedding the hidden state of the last frame to the latent space, which may result in temporal information not being fully learned. Some works use a transformer instead to apply attention mechanism on feature extraction [5,16,9,30]. Since transformer cannot get a single latent code directly, [16] learns latent code for each frame, [5] simply averages the hidden states to get a single latent code, and [9] uses attention average at encoding stage and maps the latent code to L vectors at decoding stage. [30] learns the latent space by taking the first two outputs of transformer encoder corresponding to the distribution parameter tokens, then directly using the latent representation as the key and value of transformer decoder and the temporal positional encoding as the query to generate motion sequences. These works mainly focus on temporal modeling of the input but lack spatial terms. Therefore, we use GAGCN [45] to learn the complex spatial properties of human motion and use TCN for smooth temporal modeling, which learns more expressive action-aware features.

3 Methods

Problem formulation The problem we aim to solve is to generate a motion sequence $X_{1:T} = \{x_1, x_2, ..., x_T\}$ according to the given action label $a \in A$. Here A is the predefined action categories set, T is the desired length and $x_t \in \mathbb{R}^{24\times6+3}$ is the SMPL [23] pose of a single frame, including the root joint translation and 24 joint rotations represented by the continuous 6D rotation representation.

Overview As shown in Fig. 2, we encode the motion sequence into action-agnostic latent space via TCN and employ instance normalization [37] on the intermediate features to weaken the action-related features. For the feature extraction, we embed the action labels into a latent representation as an extra *label joint* of the human body for the purpose of reinforcing the action information, which we call "label-motion fusion". After that, GAGCN [45] and TCN are used to extract the action-aware spatio-temporal features of the fused input, where spatial feature extraction is lacking in previous works. For the action-condition method, we use FiLM instead of direct concatenation or bias in previous work. Specifically, we input the learned spatio-temporal features into a fully connected layer to compute modulation parameters γ and β, and perform linear modulation on the action-agnostic latent representation, so that the action-aware features are better integrated into the latent representation. Finally, the GRU is used to generate variable-length motion sequences in conjunction with temporal encoding and latent reinforcement.

3.1 Uncoupled-Modulation CVAE

Action-agnostic latent representation Following previous work [10,30], we adopt a CVAE-based framework for action-conditioned motion generation. Previous works encode the labels and sequences coupledly to learn the latent representation, which makes the learned latent representation contain more or less action-related information. When random sampling from the latent space, the latent representation may contain the information of an arbitrary action label a, and if we modulate it with another action label b, the conflict between them will result in the generated motion not matching action label b.

We assume that a piece of white and blank paper can be drawn well, which means that it is easier to learn something as an "unskilled kid" than as an adult. So we need to learn an action-agnostic latent representation, just like an unskilled kid who only knows the basic movement laws. As shown in the upper left corner of Fig. 2, we use 6 TCN layers as an action-agnostic encoder(AAGE) denoted as Ψ_{AAGE} and add an instance normalization layer between the third and fourth layers to weaken the action-related information contained in the input $X_{1:T}$. Finally, the encoder outputs μ and σ to obtain $z \in \mathbb{R}^{256}$ by reparameterization:

$$\mu, \sigma = \Psi_{AAGE}(X_{1:T}), \quad p(z) = N(z|\mu, \sigma) \tag{1}$$

Action-aware modulation Since the learned latent space is action-agnostic, the action condition method is required to be more powerful to make the generated motion match the given label. A professional teacher needs two capabilities: being good at learning action skills from motion sequences; knowing how to teach the skills to others, i.e. feature extraction and action condition.

In order to extract action-aware features, we need to fuse the label and motion and learn the spatio-temporal dynamics of the input motion sufficiently. Intuitively, the action label of a motion sequence is related to whole-body joints, so we encode the action label into an embedding vector by a linear projection, treating it as an "*label joint*". Then we concatenate the embedded vector and other joint features to get the labeled poses $X^l_{1:T}$, which we call it as "label-motion fusion" shown in the lower left corner of Fig. 2. GAGCN is a novel variant of GCN proposed by [45] to learn the complex spatial characteristics of human motion. Here we train an action-aware encoder(AAWE) with spatial GAGCN (denoted as Ψ^s_{AAWE}) and TCN (denoted as Ψ^t_{AAWE}) to learn the spatial dependencies between joints and the temporal dynamics to get the action-aware spatio-temporal features f_{st}. The whole encoder can be formulated as follows:

$$f_{st} = \Psi^t_{AAWE}(\Psi^s_{AAWE}(X^l_{1:T})) \tag{2}$$

In order to make our action-aware feature to know how to "teach" the latent representation, we use FiLM as the modulation method instead of the simple concatenation or bias in previous work(shown in the upper right corner of Fig. 2). We use a fully connected layer(denoted as Φ) as FiLM generator to compute the modulation parameters $\gamma \in \mathbb{R}^{256}$ and $\beta \in \mathbb{R}^{256}$ from action-aware features, and then perform linear modulation on the latent representation z to obtain the action-aware latent representation $z' \in \mathbb{R}^{256}$:

$$\gamma, \beta = \Phi(f_{st}), \quad z' = \gamma * z + \beta \tag{3}$$

Temporal decoder To carry out action-conditioned motion generation, we pre-compute FiLM parameters for each action offline and save them with the corresponding labels. At runtime, given a prescribed action label, we randomly select a corresponding FiLM parameter to modulate the randomly sampling z. For more details, please refer to the supplementary material. Once we get the modulated latent representation z', to better recover a temporal sequence from a single latent z', we perform time encoding on the input length T via positional encoding and project $z' \in \mathbb{R}^d$ into $Z'_T \in \mathbb{R}^{d\times T}$ with the help of time encoding and a linear layer to get T vectors $[z'_1, \ldots, z'_T]$. After that, we input Z'_T into 4 GRU layers to generate an action-conditioned motion sequence(shown in the lower right corner of Fig. 2). Our uncoupled modulation is able to further enhance the diversity of the generated motion by operation on the modulation parameter like interpolation(seeing Sec. 4.4). In our experiments, we find that the strong condition may lead to posterior collapse, so we take a "latent reinforcement" approach [22] by adding the unmodulated z to the input of the last three layers of GRU to make the decoder's attention focus more on z. This trick solves the problem to some extent.

3.2 Training

The implementation details of our UM-CVAE are shown in the supplementary material. Here we only formulate our training loss. Following [30], we adopt 3 loss terms to train out model, the total loss is the weighted sum of these loss terms: $L = \lambda_{pr} L_{pr} + \lambda_{vr} L_{vr} + \lambda_{kl} L_{kl}$.

Pose reconstruction loss L_{pr}: We use L2 parametric to calculate the loss between our reconstruction results $\widetilde{X}_{1:T} = \{\widetilde{x}_1, \widetilde{x}_2, ..., \widetilde{x}_T\}$ and Ground Truth $X_{1:T} = \{x_1, x_2, ..., x_T\}$, formulated as: $L_{pr} = \frac{1}{T}\sum_{i=1}^{T} \|\widetilde{x}_i - x_i\|_2$.

Vertex coordinates reconstruction loss L_{vr}: For a more refined reconstruction results, we adopt an extra reconstruction loss on mesh Vertex coordinates. Specifically, we use differentiable SMPL layer to transform poses $X_{1:T} = \{x_1, x_2, ..., x_T\}$ into mesh vertices $V_{1:T} = \{v_1, v_2, ..., v_T\}$, the loss is formulated as: $L_{vr} = \frac{1}{T}\sum_{i=1}^{T} \|\widetilde{v}_i - v_i\|_2$.

KL divergence loss L_{kl}: We adopt the standard KL divergence loss, i.e. minimizing the KL divergence between Θ-parametrized approximate posterior distribution$q_{\Theta}(z|X_{1:T})$ and standard Gaussian distribution $p(z)$, it can be formulated as: $L_{kl} = D_{kl}\{q_{\Theta}(z|X_{1:T})\|p(z)\}$.

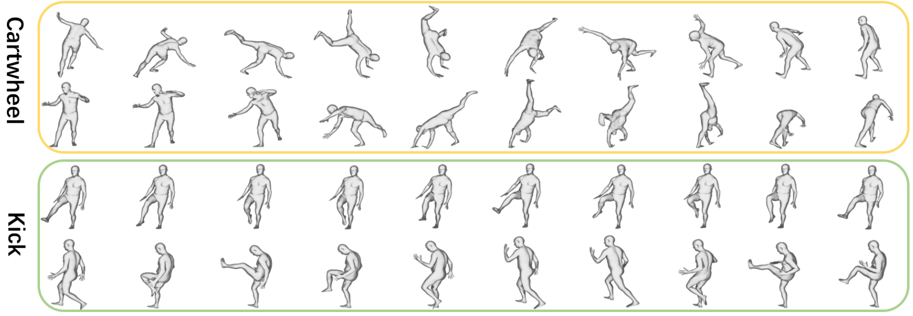


Fig. 3. Qualitative results of our method. We illustrate our generations of "Cartwheel" and "Kick" actions from BABEL, and each action consists of 2 sequences. These results demonstrate that our method can generate complex, realistic, diverse, and label-compliant motions. More results are shown in the supplementary material.

4 Experiments

In this section, we evaluate the proposed method. First, we will show the details of the used benchmark dataset and evaluation metrics in Sec. 4.1. The quantitative and qualitative comparison results with the state-of-the-art method will be given in Sec. 4.2. Then, we will analyze the main components of our method in Sec. 4.3. Finally, we will show the applications of our method in Sec. 4.4.

4.1 Datasets and evaluation metrics

Datasets Here we briefly introduce the datasets we used. Please refer to supplementary material for more details about the datasets

HumanAct12 [10]is adopted from an existing dataset PHSPD [46], consisting of 1,191 motion clips and 90,099 frames in total. All motions are organized into 12 action categories.

UESTC [15]consists of 25K sequences across 40 action categories and 118 persons collected using Microsoft Kinect v2 sensors. [30] use VIBE to obtain SMPL sequences, which we use for training and testing. The processed dataset has 10650 sequences for training and 13350 sequences for testing.

BABEL [31]leverages the recently introduced AMASS dataset [25] for mocap sequences. BABEL contains action annotations for about 43.5 hours of mocap performed by over 346 subjects from AMASS represented by SMPL-H [32], with 15472 unique language labels.

Evaluation metrics Following [10], we measure Frechet Inception Distance(FID), action recognition accuracy(Acc.), overall diversity(Div.), and multimodality(MM.) for quantitative evaluations. For HumanAct12 and UESTC, we use the provided recognition models of [10] and [30] to extract motion features to compute evaluation metrics. For BABEL, since the dataset is complicated and challenging for action recognition, we only use their provided recognition models [33] to compute the recognition accuracy for evaluation, including Top-1, Top-5, and Top-1-norm accuracy.

4.2 Comparisons with the state-of-the-art methods

To the best of our knowledge, the prior works focus on action-conditioned motion generation are A2M [10] and ACTOR [30], so we compare with their works qualitatively and quantitatively on HumanAct12, UESTC, and BABEL.

Quantitative comparison We used their publicly available code and pretrained model to obtain results. It is worth noting that A2M does not experiment on UESTC, so we use their code to train on UESTC for 1500 epochs. And it is mentioned in [30] that their model can get better results by training more epochs. To be fair, we use their code to train 1500 epochs on UESTC, so the results we report are slightly different from those in ACTOR(better than the results demonstrated in [30]). We use the evaluation metrics in 4.1 to perform quantitative comparison on HumanAct12 and UESTC, the results are shown in Tab. 1. And we evaluate the comparison of recognition accuracy on BABEL. The results are shown in the third to sixth rows of Tab. 2

Thanks to our uncoupled latent spatial learning strategy and action-aware modulation, our results show a significant improvement over A2M and ACTOR on all datasets, especially on UESTC and BABEL. These two datasets have more motion types and motion sequences compared to HumanAct12, which indicates

Table 1. Quantitative comparison to the state-of-the-art works on UESTC and HumanAct12. We use the evaluation metrics in 4.1 and the best results are marked in bold. We can see that our method outperforms the state-of-the-art in both datasets.

	UESTC					HumanAct12			
Methods	$\text{FID}_{tr}\downarrow$	$\text{FID}_{te}\downarrow$	Acc.↑	Div.→	MM.→	$\text{FID}_{tr}\downarrow$	Acc.↑	Div.→	MM.→
Original	2.93 ±0.26	2.79 ±0.29	98.8 ±0.10	33.34 ±0.32	14.16 ±0.06	0.02 ±0.00	99.4 ±0.00	6.86 ±0.03	2.60 ±0.01
A2M [10]	25.78 ±1.31	27.01 ±1.99	88.1 ±0.57	31.78 ±0.41	15.44 ±0.11	2.46 ±0.08	92.3 ±0.20	7.03 ±0.04	2.87 ±0.04
ACTOR [30]	16.81 ±1.70	18.95 ±1.41	91.7 ±0.31	32.70 ±0.59	14.53 ±0.08	0.12 ±0.00	95.5 ±0.80	6.84 ±0.03	2.53 ±0.02
Ours	**9.12** ±0.30	**8.58** ±0.23	**93.0** ±0.24	31.85 ±0.29	15.08 ±0.09	**0.09** ±0.00	**95.8** ±0.42	6.81 ±0.02	2.93 ±0.01

Table 2. Quantitative results on BABEL. The fourth to sixth rows show the comparison with state-of-the-art, and the best results are underlined. The recognition accuracy of our method is the highest and also the closest to the real data. The comparison between the "Original" row and the "Augmented" row shows that our method can augment the action recognition dataset and improve the action recognition accuracy.

	BABEL-60		
Methods	Top-5	Top-1	Top-1-norm
Original	67.83	33.41	30.42
A2M [10]	52.34	26.17	24.06
ACTOR [30]	48.24	25.37	23.49
Ours	57.14	29.04	27.81
Augmented	**70.01**	**35.14**	**32.18**

that our approach can generate realistic, diverse, and label-constrained motion on the more challenging dataset. It is worth noting that because there are more categories and more complex data in BABEL, the evaluation results of all methods are relatively worse than the real data, where our method is still the best. Also, we found that ACTOR performs worse than A2M on BABEL. We guess it is because ACTOR uses a single z as key and value for transformer decoder, which is not enough to learn the multi-head attention, resulting in unsatisfactory results on the complex dataset.

Qualitative comparison We visualize the generated results to demonstrate the advantages of our approach. Because of the limitations of previous methods, the motions they generate may have problems such as unnatural, falling into a stationary state, and confusing action types, etc. Our experimental results indicate that we solve these problems well. We demonstrate qualitative comparison results here in Fig 4 and the motion generation results on BABEL in Fig. 3.

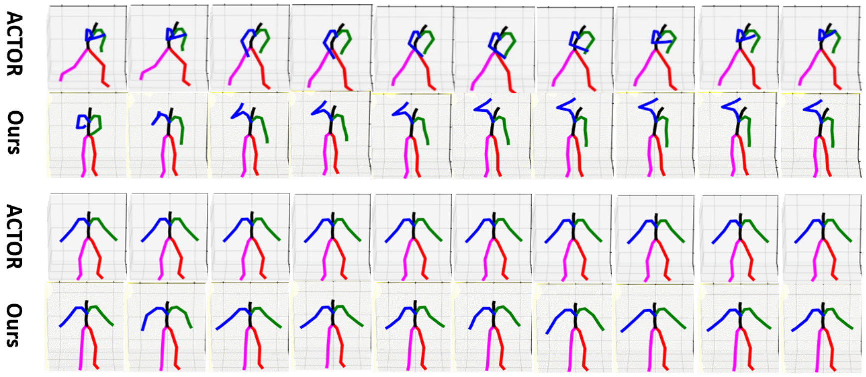


Fig. 4. Qualitative comparison of our method and ACTOR [30]. We illustrate the generations of "Drinking"(Top) on HumanAct12 and "Wrist-circling"(Bottom) on UESTC. In order to make the contrast effect more intuitive, we used the skeleton instead of the shape. While ACTOR have some issues like falling into a stationary state and confusing action types, our method solves these problems well.

The comparison of results generated on "Drinking" is worthy to notice. When given the "drinking" action label as the condition to the latent representation, ACTOR [30] generates a motion sequence in which the lower body is "warming up" and the upper body is lifting just like "drinking", and the generated motion of ours meets the labels well. This means that the action-related features learned in the latent representation indeed conflict with that contained in the user-specified label, which is strong support for our motivation.

The comparison of "Wrist-circling" is also convincing. In the movement of "Wrist-circling", only the wrist is circling with a small amplitude, and other joints are basically static. Generating this kind of motion requires us to learn the spatio-temporal characteristics of the motion well and reconstruct the temporal dynamics of the motion when decoding. We can see that the motion generated by ACTOR is stuck in a stationary state, while the motion generated by our method can normally perform wrist-circling. This is because we take the temporal dynamics into account when performing the reconstruction by mapping z' into ${Z'}_T$, while a single z is used as key and value in ACTOR to recover the motion sequence. More qualitative results(figure and video) are shown in the supplementary material.

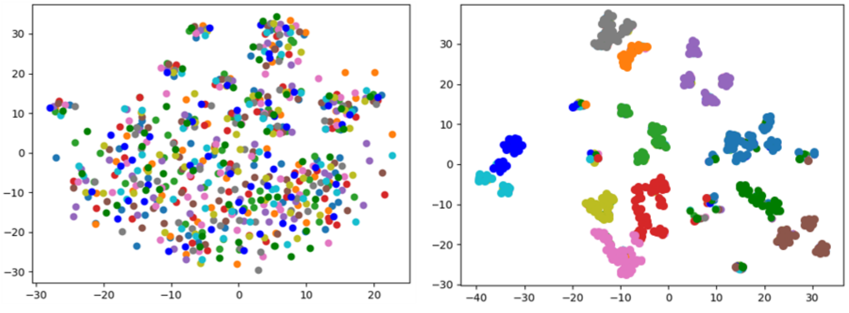
4.3 Ablation study

We do ablation studies to evaluate the effect of key components in our method: action-aware modulation, instance normalization, and latent reinforcement.

Effect of action-aware modulation: For better action-conditioned motion generation, we propose action-aware modulation to learn spatio-temporal features and modulate the latent space using FiLM. To demonstrate the advantages of our action-aware modulation, we remove action-aware encoder and

Table 3. Ablation study. We perform ablation studies to evaluate the effect of key components in our method on UESTC and HumanAct12, including action-aware modulation(FiLM), instance normalization(IN), and latent reinforcement(LR).

	UESTC					HumanAct12			
Methods	FID$_{tr}$ ↓	FID$_{te}$ ↓	Acc.↑	Div.→	MM.→	FID$_{tr}$ ↓	Acc.↑	Div.→	MM.→
w/o IN	16.89 ±0.74	16.42 ±0.55	91.6 ±0.59	34.51 ±0.53	16.58 ±0.21	0.23 ±0.04	93.9 ±0.57	6.92 ±0.12	3.07 ±0.04
w/o FiLM	23.94 ±1.58	25.17 ±2.14	88.9 ±0.68	32.52 ±0.54	16.21 ±0.13	0.52 ±0.42	92.7 ±0.44	5.97 ±0.06	2.27 ±0.03
w/o LR	11.88 ±0.41	12.71 ±0.38	92.7 ±0.30	26.78 ±0.29	12.07 ±0.07	0.12 ±0.00	95.0 ±0.34	5.18 ±0.02	2.17 ±0.01
Ours	**9.12** ±0.30	**8.58** ±0.23	**93.0** ±0.24	31.85 ±0.29	15.08 ±0.09	**0.09** ±0.00	**95.8** ±0.42	6.81 ±0.02	2.93 ±0.01



(a) Before modulation (b) After modulation

Fig. 5. Latent space visualization. We visualize the latent space before and after modulation using T-SNE [24]. The latent space before modulation is randomly distributed(action-agnostic). After modulation, the latent variables are separated from each other according to the action types, making the latent space action-aware.

directly input the motion sequences and action labels into action-agnostic encoder without IN, and use the action bias method in [30] for modulation. The experimental results are shown in the row starting with "w/o FiLM" in Tab. 3. It can be seen that the experimental results are significantly worse on FID and Acc. without our action-aware modulation, which means that the generated results have worse conformity to the action types of the original dataset. Moreover, the Div. and MM. increase instead due to the possibility of generating some artificial motions without action-aware modulation.To illustrate the effect of action-aware modulation more intuitively, we visualize the latent space before and after modulation with TSNE [24], and the results are shown in Fig. 5. The latent space is clearly clustered according to action types after modulation, which indicates that our method knows how to "teach" the latent space action skills.

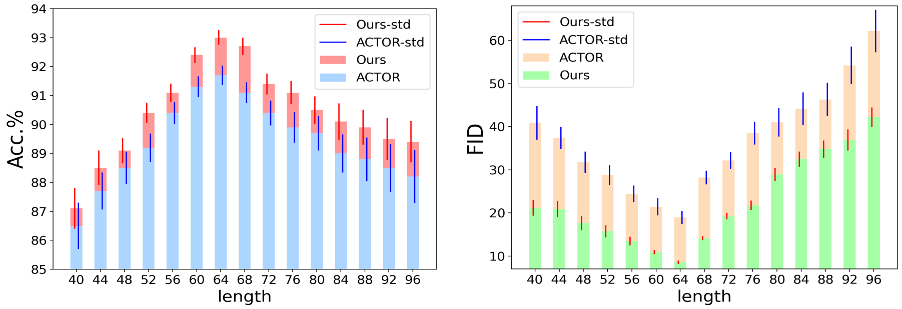
Effect of instance normalization: According to the above description, the learned latent space should be action-agnostic as the way to avoid conflicting with modulation. Inspired by [37], we use instance normalization to weaken the action-related information contained in the input sequence. To verify the effectiveness of this component, we directly remove it and the results are shown in the row starting with "w/o IN" in Tab. 3. When our encoder does not contain instance normalization, FID and Acc. become significantly worse, while Div. and MM. rise instead. The reason may be that the encoder without instance normalization learns more or less information related to the label and conflicts with the given label when modulating. The generated results will be different from the real human motion and increase the "diversity", which is obviously not what we want indeed. The visualized latent space in Fig.5(a) is scattered, which also proves that the learned latent space is action-agnostic to some extent.

Effect of latent reinforcement: In our experiments, we find that since our action-aware modulation is too strong and weakens the effect of the latent space, which may reduce the diversity of generated motions. It is a problem known as posterior collapse. Inspired by [22], we add the unmodulated z to the input of the last three layers of GRU decoder to emphasize the importance of the latent variable, which is named latent reinforcement. The ablation results are shown in the row starting with "w/o LR" in Tab. 3, where FID and Acc. change very slightly and Div. and MM. decrease a lot. This indicates that the lack of latent reinforcement does reduce the diversity of the generated motions.

4.4 Applications

Generating variable-length motions Similar to [30], we add length T as input to the encoder stage to achieve variable-length motion generation through temporal encoding, latent variable expansion, and GRU decoder. We use a model trained on UETSC with fixed length 64 to generate motions of different lengths from 40 to 100(with 4 frames interval), and then use our evaluation metrics to compute the FID_{te} and Acc. of the generated motions. To show the superiority of our model, we compare it with ACTOR, and the experimental results are plotted in Fig. 6. The results show that our method ensures higher Acc. and lower FID_{te} for different lengths of motion, which is a significant improvement compared to ACTOR. Besides, the standard deviation of our method is smaller, which indicates that our method is more robust.

Augmentation for action recognition Our method generates high-quality action-conditioned motions, thus a very straightforward application is augmenting the action recognition dataset to help improve action recognition accuracy. We report the augmentation of BABEL because the current state-of-the-art action recognition method gets unsatisfactory accuracy on it. Specifically, we generate 30 motion clips for each motion category in BABEL-60(1800 in total), adding 9.11% data to BABEL-60. The augmented data is then used to train the motion recognition model [33]. The Acc. in the row starting with "Augmented"



(a) Acc. for variable-length (b) FID$_{te}$. for variable-length

Fig. 6. Generating variable-length motions. We report the Acc.(left) and FID$_{t}e$ (right) of the motion from 40 to 100 frames at 4 frame intervals generated by a model trained on 64 frames and compare them with ACTOR [30]. The bars indicate the mean and the lines indicate the standard deviation. The results show that our method generates high-quality variable-length motions without training on other lengths and outperforms previous work.

in Tab. 2 is higher than the "Original" row, which demonstrates that the data generated by our method can indeed augment the action recognition dataset.

Motion interpolation Thanks to our action-aware modulation method, we can interpolate different motions of the same action label to get more diverse results. Specifically, we select several different modulation parameters learned from a given label and interpolate them to generate interpolated motion. In this way, the motions we generate will be more diverse while maintaining the properties of the given label. The results are shown in the supplementary material.

5 Conclusions

We propose a novel action-conditioned motion generation model called UM-CVAE. Specifically, we learn the action-agnostic latent representation and action-aware spatio-temporal feature in an uncoupled manner, then perform action-aware feature-wise linear modulation via FiLM. The generated motions are realistic, diverse, and label-compliant, which show significant improvement over prior works and have some useful applications. However, there are still limitations to our approach. For example, although our decoding approach is carefully designed, how to better recover motion sequence from a single latent variable remains to be explored. Also, our method relies on the quality of the dataset and cannot be conveniently extended to new actions or unseen body size.

Acknowledgements. This work was supported by the National Key R&D Program of Science and Technology for Winter Olympics (No.2020YFF0304701) and the National Natural Science Foundation of China (No.61772499).

References

1. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 5915–5920. IEEE (2018)
2. Ahuja, C., Morency, L.P.: Language2pose: Natural language grounded pose forecasting. In: 2019 International Conference on 3D Vision (3DV). pp. 719–728. IEEE (2019)
3. Barsoum, E., Kender, J., Liu, Z.: Hp-gan: Probabilistic 3d human motion prediction via gan. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 1418–1427 (2018)
4. Brand, M., Hertzmann, A.: Style machines. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 183–192 (2000)
5. Cheng, X., Xu, W., Wang, T., Chu, W.: Variational semi-supervised aspect-term sentiment analysis via transformer. arXiv preprint arXiv:1810.10437 (2018)
6. Clavet, S.: Motion matching and the road to next-gen animation. In: Proc. of GDC (2016)
7. Corona, E., Pumarola, A., Alenya, G., Moreno-Noguer, F.: Context-aware human motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6992–7001 (2020)
8. Cui, Q., Sun, H., Yang, F.: Learning dynamic relationships for 3d human motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6519–6527 (2020)
9. Fang, L., Zeng, T., Liu, C., Bo, L., Dong, W., Chen, C.: Transformer-based conditional variational autoencoder for controllable story generation. arXiv preprint arXiv:2101.00828 (2021)
10. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 2021–2029 (2020)
11. Holden, D., Kanoun, O., Perepichka, M., Popa, T.: Learned motion matching. ACM Transactions on Graphics (TOG) **39**(4), 53–1 (2020)
12. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG) **36**(4), 1–13 (2017)
13. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)
14. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 5308–5317 (2016)
15. Ji, Y., Xu, F., Yang, Y., Shen, F., Shen, H.T., Zheng, W.S.: A large-scale rgb-d database for arbitrary-view human action recognition. In: Proceedings of the 26th ACM international Conference on Multimedia. pp. 1510–1518 (2018)
16. Jiang, J., Xia, G.G., Carlton, D.B., Anderson, C.N., Miyakawa, R.H.: Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 516–520. IEEE (2020)
17. Kundu, J.N., Gor, M., Babu, R.V.: Bihmp-gan: Bidirectional 3d human motion prediction gan. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8553–8560 (2019)
18. Lee, H.Y., Yang, X., Liu, M.Y., Wang, T.C., Lu, Y.D., Yang, M.H., Kautz, J.: Dancing to music. Advances in Neural Information Processing Systems **32** (2019)

19. Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., Li, H.: Learning to generate diverse dance motions with transformer. arXiv preprint arXiv:2008.08171 (2020)
20. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13401–13412 (2021)
21. Lin, X., Amer, M.R.: Human motion modeling using dvgans. arXiv preprint arXiv:1804.10652 (2018)
22. Ling, H.Y., Zinno, F., Cheng, G., Van De Panne, M.: Character controllers using motion vaes. ACM Transactions on Graphics (TOG) **39**(4), 40–1 (2020)
23. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015)
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
25. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5442–5451 (2019)
26. Mao, W., Liu, M., Salzmann, M.: History repeats itself: Human motion prediction via motion attention. In: European Conference on Computer Vision. pp. 474–489. Springer (2020)
27. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2891–2900 (2017)
28. Mason, I., Starke, S., Komura, T.: Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. arXiv preprint arXiv:2201.04439 (2022)
29. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
30. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3d human motion synthesis with transformer vae. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10985–10995 (2021)
31. Punnakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: Babel: Bodies, action and behavior with english labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 722–731 (2021)
32. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **36**(6) (Nov 2017)
33. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12026–12035 (2019)
34. Sofianos, T., Sampieri, A., Franco, L., Galasso, F.: Space-time-separable graph convolutional network for pose forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11209–11218 (2021)
35. Starke, S., Zhang, H., Komura, T., Saito, J.: Neural state machine for character-scene interactions. ACM Trans. Graph. **38**(6), 209–1 (2019)
36. Starke, S., Zhao, Y., Komura, T., Zaman, K.: Local motion phases for learning multi-contact character movements. ACM Transactions on Graphics (TOG) **39**(4), 54–1 (2020)

37. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6924–6932 (2017)
38. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: Proceedings of the IEEE international conference on computer vision. pp. 3332–3341 (2017)
39. Wang, J.M., Fleet, D.J., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE transactions on pattern analysis and machine intelligence **30**(2), 283–298 (2007)
40. Wang, Z., Chai, J., Xia, S.: Combining recurrent neural networks and adversarial training for human motion synthesis and control. IEEE transactions on visualization and computer graphics **27**(1), 14–28 (2019)
41. Xia, S., Wang, C., Chai, J., Hodgins, J.: Realtime style transfer for unlabeled heterogeneous human motion. ACM Transactions on Graphics (TOG) **34**(4), 1–10 (2015)
42. Yuan, Y., Kitani, K.: Dlow: Diversifying latent flows for diverse human motion prediction. In: European Conference on Computer Vision. pp. 346–364. Springer (2020)
43. Zhang, H., Starke, S., Komura, T., Saito, J.: Mode-adaptive neural networks for quadruped motion control. ACM Transactions on Graphics (TOG) **37**(4), 1–11 (2018)
44. Zhong, C., Hu, L., Xia, S.: Spatial-temporal modeling for prediction of stylized human motion. Neurocomputing (2022)
45. Zhong, C., Hu, L., Zhang, Z., Ye, Y., Xia, S.: Spatio-temporal gating-adjacency gcn for human motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6447–6456 (2022)
46. Zou, S., Zuo, X., Qian, Y., Wang, S., Xu, C., Gong, M., Cheng, L.: 3d human shape reconstruction from a polarization image. In: European Conference on Computer Vision. pp. 351–368. Springer (2020)