Multi-Scale Control Signal-Aware Transformer for Motion Synthesis without Phase

Lintao Wang[1], **Kun Hu**[1,*], **Lei Bai**[2], **Yu Ding**[3], **Wanli Ouyang**[1], **Zhiyong Wang**[1]

[1]The University of Sydney, [2]Shanghai AI Laboratory, [3]Netease Fuxi AI Lab
lwan3720@uni.sydney.edu.au, kun.hu@sydney.edu.au, baisanshi@gmail.com, dingyu01@corp.netease.com,
wanli.ouyang@sydney.edu.au, zhiyong.wang@sydney.edu.au

Abstract

Synthesizing controllable motion for a character using deep learning has been a promising approach due to its potential to learn a compact model without laborious feature engineering. To produce dynamic motion from weak control signals such as desired paths, existing methods often require auxiliary information such as phases for alleviating motion ambiguity, which limits their generalisation capability. As past poses often contain useful auxiliary hints, in this paper, we propose a task-agnostic deep learning method, namely Multi-scale Control Signal-aware Transformer (MCS-T), with an attention based encoder-decoder architecture to discover the auxiliary information implicitly for synthesizing controllable motion without explicitly requiring auxiliary information such as phase. Specifically, an encoder is devised to adaptively formulate the motion patterns of a character's past poses with multi-scale skeletons, and a decoder driven by control signals to further synthesize and predict the character's state by paying context-specialised attention to the encoded past motion patterns. As a result, it helps alleviate the issues of low responsiveness and slow transition which often happen in conventional methods not using auxiliary information. Both qualitative and quantitative experimental results on an existing biped locomotion dataset, which involves diverse types of motion transitions, demonstrate the effectiveness of our method. In particular, MCS-T is able to successfully generate motions comparable to those generated by the methods using auxiliary information.

1 Introduction

Interactively controlling a character has been increasingly demanded by various applications such as gaming, virtual reality and robotics. This task remains challenging to achieve realistic and natural poses with complex motions and environments, even with large amount of high quality motion capture (MoCap) data for modelling (Holden, Komura, and Saito 2017; Peng et al. 2018). Recently, deep learning techniques have been studied for controllable motion synthesis given their strong learning capability yet efficient parallel structures for fast runtime. Many encouraging results have been achieved using deep architectures such as multilayer perceptron (MLP) networks (Holden, Komura, and Saito 2017), recurrent neural networks (Lee, Lee, and Lee 2018), generative networks (Henter, Alexanderson, and Beskow 2020) and deep reinforcement learning architectures (Peng et al. 2018). Particularly, due to the potentials of delivering fast, responsive yet high-quality controllers, MLP networks have been devised for biped locomotion (Holden, Komura, and Saito 2017), quadruped locomotion (Zhang et al. 2018), daily interaction (Starke et al. 2019), basketball play (Starke et al. 2020) and stylised motion prediction (Mason, Starke, and Komura 2022). Since a weak control signal, which is commonly used in graphics, often corresponds to a large variation of possible motions, these studies have to rely on auxiliary phase variables in line with the character's contact states for disambiguation purposes. However, the contact states may not be available for all kinds of motions and may require manual correction during data acquisition. By contrast, recurrent neural networks, e.g. (Lee, Lee, and Lee 2018), aim to constrain the next pose prediction subject to the past motions, which can be task-agnostic in terms of motion category and demonstrate better generalisation capability. The key limitation of RNN based methods is that they often suffer from slow responsiveness issues due to the large variation of the hidden memory (Starke et al. 2019).

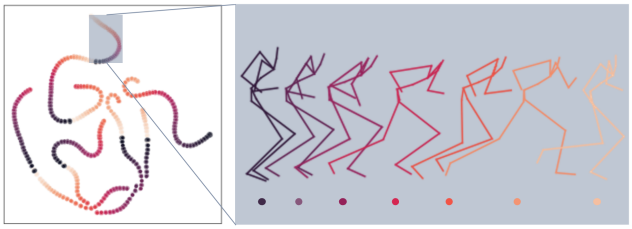

Figure 1: Illustration of the motion manifold of a running motion sequence. Joint positions and velocities of individual poses are projected into a 2D space by t-SNE and colored in line with their phases. It is noticed that auxiliary phases are continuously distributed on the manifold, which suggests the potential of inferring the phases from motion attributes.

We believe that auxiliary information can be inferred from a character's past motions. As shown in Figure 1, a walking motion sequence is represented in 2D manifolds by using different attributes (e.g. joint positions and velocities). It can

*Corresponding author.

be observed that the phases are continuously distributed on the manifolds, which enables the auxiliary information inference from the motion related attributes. Nonetheless, the past poses should be used "attentionally" since not all of them are always informative especially during motion transition, which is the reason that RNN based methods perform poorly without explicit data augmentation to the transitional cases in motion capture (MoCap) data. Therefore, in this work, we aims to study a deep learning based task-agnostic method to produce dynamic motion from trajectory-based control signals, without explicitly using additional auxiliary information such as phase.

Specifically, we propose a transformer-based encoder-decoder architecture, namely Multi-Scale Control Signal-aware Transformer (MCS-T), to attend to the motion information of past poses and trajectory with respect to various scenarios. An encoder formulates the past motion patterns of a character from multi-scale skeletons in pursuit of learning spatio-temporal patterns from different levels of dynamics. With the past motion information, the encoder is expected to formulate conventional auxiliaries implicitly. Then, a decoder guided by the control signals synthesizes and predicts the next character pose by paying trajectory-specialised attention to the encoded historical motion patterns, rather than using a long inflexible memory setting. This dynamic motion modelling pipeline helps alleviate the issues of low responsiveness and slow transition, which can be observed in existing methods not using auxiliary information. Comprehensive experiments on a biped locomotion dataset containing various motion transitions (e.g., sudden jumping and uneven terrain walking) demonstrate the effectiveness of MCS-T. It produces responsive and dynamic motion, and achieves a performance comparable to that of the methods explicitly using auxiliary information while retaining such capability for various motion categories.

The main contributions of this paper can be summarised as follows:

- A novel real-time motion controller, namely Multi-Scale Control Signal-aware Transformer, is proposed to improve the responsiveness and motion dynamics over existing methods not explicitly using auxiliary information. It is also task-agnostic, compared with the methods explicitly using auxiliary information. To the best of our knowledge, our task-agnostic method is one of the first studies utilising transformer based encoder-decoder scheme for controllable motion synthesis.
- A multi-scale graph modelling scheme is devised to exploit rich skeleton dynamics.
- A novel control signal-aware self-attention is devised to inject control signals for motion prediction.
- Comprehensive experiments were conducted to demonstrate the effectiveness of our proposed MCS-T.

2 Related Work

In this section, we review related studies in terms of kinematic-based controllable motion synthesis, transformer based motion learning and multi-scale skeleton.

2.1 Kinematics Based Controllable Motion Synthesis

The kinematics based methods focus on the motion of character bodies including the joints without considering the physics that cause them to move. Four major categories of methods are reviewed as follows.

Search-based Methods: Early studies were based on graphs (Arikan and Forsyth 2002; Lee et al. 2002; Kovar, Gleicher, and Pighin 2008; Lee et al. 2010), where each frame of a motion database was treated as a vertex and edges represented possible transitions between two frames. A graph search can find a path to produce an expected motion. Motion matching (Clavet 2016; Holden et al. 2020) simplified the graph search by finding transitional frames directly in animation databases and produced the state-of-art gaming animation (Buttner 2019; Zinno 2019). However, the matching criterion are often required to be devised by experienced animators for a wide range of motion scenarios.

Recurrent Neural Network based Methods: Fragkiadaki et al. (2015) constructed an encoder-decoder structure based on recurrent neural network (RNN) and directly adopted 3D body joint angles to predict character poses. Li et al. (2017) addressed the error accumulation issue by introducing a teacher forcing-like mechanism (Williams and Zipser 1989). Lee, Lee, and Lee (2018) incorporated control signals into RNNs. To alleviate the low responsiveness and slow motion transition issues caused by the inflexible RNN memory state, comprehensive data augmentation was conducted to enrich transitional patterns. However, less motion diversity was observed during the runtime, since the augmented knowledge was still limited.

Phase-based Methods: Phase-functioned neural network (Holden, Komura, and Saito 2017) adopted a multilayer perceptron (MLP) to predict biped locomotion with an auxiliary foot contact phase, which clusters motion with similar timing to disambiguate motion predictions. The phase-based frameworks were further extended to quadruped locomotion (Zhang et al. 2018), environmental interaction (Starke et al. 2019), basketball game (Starke et al. 2020) and martial arts (Starke et al. 2021). Nevertheless, the acquisition of phase information relied on the expertise of animators and the contact information of characters, which may not be universally available. Mason, Starke, and Komura (2022) proposed a heuristic principal component analysis based strategy to compute the phase of a stylised motion, where the arms often exhibited special movements without contact states. However, it was still a task-specific solution.

Generative Methods: Instead of predicting a single motion pose, modelling the conditional pose distribution and conducting sampling could avoid the averaging pose from vastly different poses (Habibie et al. 2017; Ling et al. 2020; Henter, Alexanderson, and Beskow 2020; Liu et al. 2021; Li et al. 2022; Kania, Kowalski, and Trzciński 2021). Ling et al. (2020) used a variational autoencoder (VAE) to estimate the next pose distribution and draw user control-conditioned samples through reinforcement learning. Normalising flow was also introduced for this purpose, which modelled motion distribution and control signal together (Henter, Alexan-

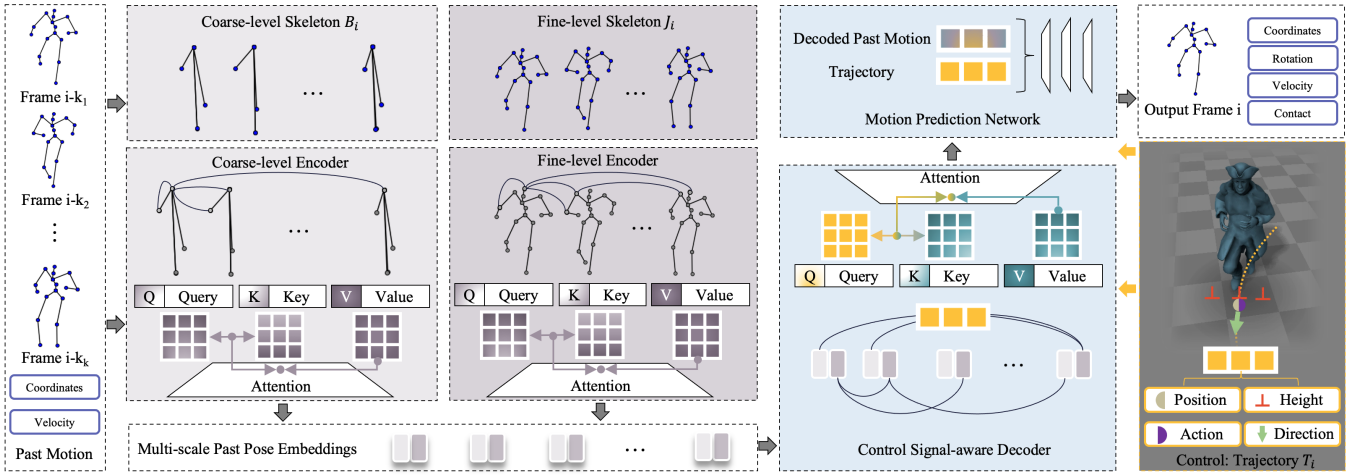

Figure 2: Illustration of our proposed MCS-T method, which is based on an encoder-decoder architecture to formulate the past motion patterns with multi-scale skeleton representations and predict the next motion with the guidance of the control signals.

derson, and Beskow 2020). Although the generative approach did not require auxiliary information, it heavily depended on balanced MoCap data distributions (Ling et al. 2020), not designed for trajectory-based control signal (Liu et al. 2021) or less controlled on produced motion gait (Henter, Alexanderson, and Beskow 2020).

2.2 Transformer in Motion Learning

Transformers (Vaswani et al. 2017) have achieved great success in a wide range of tasks such as natural language processing (Devlin et al. 2018) and computer vision (Dosovitskiy et al. 2020; Arnab et al. 2021). Compared with traditional recurrent neural networks, self-attention mechanisms perform more effectively and efficiently to address sequential patterns. Therefore, various transformer based methods were proposed for many motion related tasks such as motion prediction (Mao, Liu, and Salzmann 2020; Martínez-González, Villamizar, and Odobez 2021; Aksan et al. 2021; Wang et al. 2021), action recognition (Plizzari, Cannici, and Matteucci 2021; Mazzia et al. 2022), 3D pose estimation (Zheng et al. 2021) and motion synthesis (Petrovich, Black, and Varol 2021). However, there are few studies utilising transformer for controllable motion synthesis.

2.3 Multi-scale Skeleton

To better explore rich spatial skeleton representations of human poses, many studies introduced multi-scale skeletons by using higher order polynomials of adjacency matrices (Liu et al. 2020), graph convolutions (Jang, Park, and Lee 2022) or heuristic rules (Li et al. 2020; Dang et al. 2021; Ghosh et al. 2021; Bhattacharya et al. 2021). We address the multi-scale graphs with transformers to provide multi-scale tokens with trajectories, which is the first attempt in controllable motion synthesis for more responsive motions.

3 Methodology

Figure 2 illustrates the proposed MCS-T architecture, which addresses the motion control problem as a regression task. The motion data is first parameterised as pose and trajectory embeddings. The pose embedding is formulated by multi-scale skeleton graphs for comprehensively exploiting the past spatio-temporal relations. It encodes the motion sequence for each skeleton scale by specialised transformer encoders for latent motion representation. The representation is then utilised by a transformer decoder queried by trajectory information, i.e., control signal, for a control-conditioned integration with past motion states. Finally, a motion prediction network predicts the character's next pose and potential future trajectory.

3.1 Multi-scale Skeleton Poses

A virtual character is animated upon a skeleton of rotational joints, of which the coordinates and velocities can be defined regarding the motion. Each pose skeleton in a motion sequence can be viewed as a graph, where the joints are vertices and the bones are edges. Based on such graph representation, multi-scale skeletons can be constructed for a pose by aggregating the adjacent vertices as a pooled coarse-level vertex. As illustrated in Figure 2, two scales of skeletons in a fine-to-coarse scheme are adopted in this study. This scheme aims to comprehensively characterise the spatial patterns, by which the additional coarse-level representation enables global observations of motions and improves motion dynamics especially during a motion transition.

The fine-level representation is the same as the original skeleton structure obtained from MoCap, which contains 31 joints. Specifically, we denote j_i^p and j_i^v as the vectors representing the coordinates local to the corresponding root transformation and velocity values of the fine-level vertices (i.e., joints) of the i-th frame, respectively. For the coarse-level representation, we denote b_i^p and b_i^v as the vectors representing the coordinates and velocity values of the vertices (i.e., aggregated joints) of the i-th frame, respectively.

Particularly, for the motion prediction of the i-th frame, we construct an input X_i, which consists of two components regarding the past pose information of the two skeleton

scales: J_i and B_i. In detail, we denote $J_i = \{(j^p_{i-k}, j^v_{i-k})\}$, $B_i = \{(b^p_{i-k}, b^v_{i-k})\}$, $k = k_1, ..., k_K$, as the fine and coarse sequences, respectively. In total, K frames are adopted instead of using all frames for the consideration of both efficiency purpose and model complexity.

3.2 Multi-scale Motion Encoder

A motion encoder aims to formulate the past motion patterns X_i as a reference for predicting the future motion. Compared with the methods using auxiliary information, our encoder only depends on the past motion information available and can be generalized to all kinds of actions. However, trivially using all past motion information could result in issues of low responsiveness and slow motion transition. In other words, using all past information can introduce redundancy for predicting the next pose and sometimes even disturb the prediction, which may fall into the historical motion states. Thus, a transformer-based multi-scale encoder is proposed to formulate the past motion patterns in an adaptive manner.

The fine and coarse-level pose information J_i and B_i can be treated as matrices, where each row represents the position and velocity of a particular temporal motion frame. Our multi-scale encoder is based on self-attentions (Vaswani et al. 2017) using the concepts of query, value and key, which can be formulated as:

$$\begin{aligned} Q^J_i = J_i W^{Q,J}, K^J_i = J_i W^{K,J}, V^J_i = J_i W^{V,J}, \\ Q^B_i = B_i W^{Q,B}, K^B_i = B_i W^{K,B}, V^B_i = B_i W^{V,B}, \end{aligned} \tag{1}$$

where $W^{\cdot,\cdot}$ are projection matrices containing trainable weights with an output dimension γ, J related matrices formulate the fine-level pose patterns and B related matrices formulate the coarse-level pose patterns. Then, the temporal patterns can be computed for each level as follows:

$$Z^J_i = \text{softmax}(\frac{Q^J_i K^{J\intercal}_i}{\sqrt{\gamma}})V^J_i, Z^B_i = \text{softmax}(\frac{Q^B_i K^{B\intercal}_i}{\sqrt{\gamma}})V^B_i. \tag{2}$$

To this end, the temporal relations of motion frames can be formulated by observing the entire sequence based on the weights obtained using the softmax function in Eq. (2). In practice, multiple independent self-attentions can be adopted to increase the capability of modelling and feed-forward components are followed, which is known as a transformer encoder layer. By stacking multiple transformer encoder layers for each observation level, the final spatio-temporal patterns can be obtained. For the convenience of notations, we still use the symbols Z^J_i and Z^B_i to indicate the encoded sequential representations. By concatenating Z^J_i and Z^B_i in a frame-wise manner, a sequence $\{Z_i\}$ can be obtained as the encoded multi-scale past motion patterns.

3.3 Control Signal & Trajectory

The trajectory of a character's movement is based on the user's control signals. We denote a trajectory vector:

$$\begin{aligned} T_i = (&t^p_{i,s-S}, ..., t^p_{i,s}, ..., t^p_{i,S-1}, t^d_{i,s-S}, ..., t^d_{i,s}, ..., t^d_{i,S-1}, \\ &t^h_{i,s-S}, ..., t^h_{i,s}, ..., t^h_{i,S-1}, t^g_{i,s-S}, ..., t^g_{i,s}, ..., t^g_{i,S-1}), \end{aligned} \tag{3}$$

which represents the sampled discrete trajectory patterns for the prediction of the frame i. Particularly, the indices of the sampled points are specified as $s \in \{s_{-S}, ..., s_0, ..., s_{S-1}\}$. In this study, we empirically adopt $S = 6$ and the sampled points are evenly distributed around the current frame to cover the trajectories 1 second before and 1 second after. In detail, the trajectory includes four aspects:

- $t^p_{i,s}$ represents the sampled s-th trajectory position in the 2D horizontal plane of the i-th frame.
- $t^d_{i,s}$ indicates the trajectory direction in the 2D horizontal plane, which is the facing direction of the character.
- $t^h_{i,s}$ is a sub-vector contains the trajectory height in line with the terrain to characterise the geometry information, which are obtained from three locations regarding the sampled point including the center, left and right offset.
- $t^g_{i,s}$ is a one-hot encoding sub-vector regarding the action category for the sampled trajectory point. For our locomotion settings, we have five action categories including standing, walking, jogging, jumping and crouching.

3.4 Control Signal-aware Decoder

Based on the past motion embeddings from the multi-scale motion encoder, a control signal-aware decoder is proposed to formulate a latent embedding for motion prediction. The trajectory information is involved by the decoder to attend to the past encoded motion patterns through a control signal-aware attention mechanism. This allows the decoded patterns being relevant to the user's control signals. In detail, we adopt the trajectory T_i as a query to the past motions:

$$q^D_i = T_i W^{Q,D}, K^D_i = Z_i W^{K,D}, V^D_i = Z_i W^{V,D}, \tag{4}$$

where $W^{\cdot,\cdot}$ are projection matrices containing trainable weights with an output dimension γ. Hereafter, the past motion information with user control can be summarised into a vector as follows:

$$z^D_i = \text{softmax}(\frac{q^D_i K^{D\intercal}_i}{\sqrt{\gamma}})V^D_i. \tag{5}$$

Particularly, we call the attention in Eq. (4-5) as a control signal-aware attention and multi-heads of it are adopted with feed-forward networks to characterise the motions from multiple aspects. For the simplification of notations, we continue to use z^D_i to denote this multi-head output.

3.5 Motion Prediction Network

To predict and synthesize the motion of the i-th frame, which we denote as Y_i, an additional motion prediction network (MPN) component is introduced. Y_i contains pose $\{(j^p_i, j^v_i, j^r_i)\}$, trajectory T_{i+1} and contact information C_i. Particularly, j^r_i represents local joint rotation additional to position and velocity. The prediction $\hat{T}_{i+1}$ of T_{i+1} is only for the trajectory after the i-th frame, where the sampled trajectory points before the current frame already exist. C_i is a vector, which indicates the labels of foot contact for each heel and toe joint of the two feet. It can be used to perform Inverse Kinematics (IK) post-processing to better fit the character with terrain geometry.

Our MPN is based on feed-forward layers with Exponential Linear Unit (ELU) activation function (Clevert, Unterthiner, and Hochreiter 2015). In detail, we have an estimation $\hat{Y}_i$ of Y_i:

$$\hat{Y}_i = \text{MPN}(z_i^D, T_i), \tag{6}$$

where the decoded output and motion trajectory are considered as the input. Note that the trajectory information is also used for MPN besides the decoder, which helps the control signals to be fully formulated for providing highly responsive motion synthesis.

3.6 MCS-T Training and Runtime Inference

By defining the computations of the proposed MCS-T as a function $\mathcal{F}$ with trainable parameters $\mathbf{\Theta}$, where $\hat{Y}_i = \mathcal{F}(X_i, T_i)$. A mean squre error (MSE) loss with ℓ_1 regularization is adopted to optimize $\mathbf{\Theta}$. In detail, we solve the following optimization problem during the training:

$$\arg\min_{\mathbf{\Theta}} \parallel Y_i - \boldsymbol{\mathcal{F}}(X_i, T_i; \mathbf{\Theta}) \parallel_2^2 + \lambda|\mathbf{\Theta}|, \tag{7}$$

where λ is a hyper-parameter controlling the scale of the regularization.

In terms of the runtime inference, a trajectory blending scheme is adopted for post-processing. In detail, the trajectory positions $\hat{t}^d_{i+1,s}$ and directions $\hat{t}^d_{i+1,s}$, $s = s_0, ..., s_{S-1}$, after the i-th frame are further blended with the user control signal for the $(i+1)$-th frame's motion prediction:

$$\begin{aligned} t^p_{i+1,s} &= (1 - \tau^p_s)\bar{t}^p_{i+1,s} + \tau^d_s \hat{t}^p_{i+1,s}, \\ t^d_{i+1,s} &= (1 - \tau^d_s)\bar{t}^d_{i+1,s} + \tau^d_s \hat{t}^d_{i+1,s}, \end{aligned} \tag{8}$$

where $\bar{t}_{i+1,s}$ is the trajectory computed by the user's control signal, τ^p_s and τ^d_s are hyper-parameters to control the blending level. That is, the user control signal is blended with higher weights in near trajectory for more responsive motion, and with lower weights in far trajectory in pursuit of smoother transition. In terms of $t^p_{i+1,s}$ and $t^d_{i+1,s}$, $s = s_{-S}, ..., s_{-1}$, they are in line with the actual existing trajectory. Additionally, the trajectory height $t^h_{i+1,s}$ can be derived based on $t^p_{i+1,s}$ within the virtual scene, and the action category $t^g_{i+1,s}$ is set directly by the user.

4 Experimental Results and Discussions

4.1 Dataset

We evaluate our proposed method on a public dataset (Holden, Komura, and Saito 2017) for a fair comparison with the state-of-the-art methods. The dataset consists of biped locomotion data of various gaits, terrains, facing directions and speeds, which helps evaluate the quality of the common character motion controller in terms of responsiveness and motion transition. A biped character with 31 joints and MoCap techniques were adopted to collect these data. In total, we obtained around 4 million samples for training.

4.2 Implementation Details

In total, $K = 5$ past frames with indices $k_1 = 1$, $k_2 = 10$, $k_3 = 20$, $k_4 = 30$ and $k_5 = 40$ were selected as input to predict the motion of the i-th frame. Note that this setting was found to provide the best quality of prediction (see Section 4.5). Two independent transformer-encoders were used for the fine-level and coarse-level motion sequences, respectively. Each of them consisted of three transformer-encoder layers using six self-attention heads of a dimension 186 and the the feed-forward layers were of a dimension 1024. A dropout rate of 0.1 was applied to the encoders. The transformer decoder was using the same configurations as the encoder. The motion prediction network was modelled as a three-layer MLP with a hidden dimension 512 and a dropout rate 0.3. $\tau^p_s = (s/S)^{0.5}$ and $\tau^d_s = (s/S)^2$ were defined for Eq(8) empirically. (see Supplementary Material for details).

During the training, the input and output were firstly normalised by their mean and standard deviation. Additionally, the input features related to the joints were all scaled by 0.1, which helped produce dynamic motions in certain scenarios to enlarge the proportion of the trajectory related inputs. In terms of the loss function, λ for ℓ_1 regularization was set to 0.01. The model was implemented by PyTorch (Paszke et al. 2019) and trained with an Adam optimisier (Kingma and Ba 2014). The learning rate was set to 10^{-4} and the batch size was 32. In total, MCS-T was trained with 20 epochs, which took around 50 hours on an Nvidia GTX 1080Ti GPU.

4.3 Comparisons with State-of-the-art Methods

Qualitative and quantitative evaluations of MCS-T were conducted against a number of baseline methods, in terms of motion quality, especially from the aspects of responsiveness and motion transition. The baseline methods include MLP with a single past pose, MLP with multiple past poses, RNN (Lee, Lee, and Lee 2018) and PFNN (Holden, Komura, and Saito 2017) methods. Overall, we show that our MCS-T was able to produce motions in line with the-state-of-the-art results with a task-agnostic design and to alleviate the fundamental issues of the baseline methods. More results are available in the supplementary material.

MLP with single past pose: We trained an MLP to synthesize motion using a single past pose with trajectory information. The experimental results show that the overall motion produced was quite stiff especially when changing the direction and could have weird artifacts such as floating as shown in the ceiling scenario in Figure 3. As expected, motion prediction from vague control signal can be difficult without auxiliary information and various possible predictions can exist, which leads to an average pose.

MLP with multiple past poses: Similar to the first baseline, except that additional pose information from multiple past frames was considered for an MLP. The results show that the generated motion was improved, as the past frames provided the auxiliary information implicitly. Nonetheless, the synthesized motion suffers from the slow motion transition issue. This problem became obvious when the character was traversing through rocky terrain as shown in Figure 3. While it was able to adapt the character motion correctly to

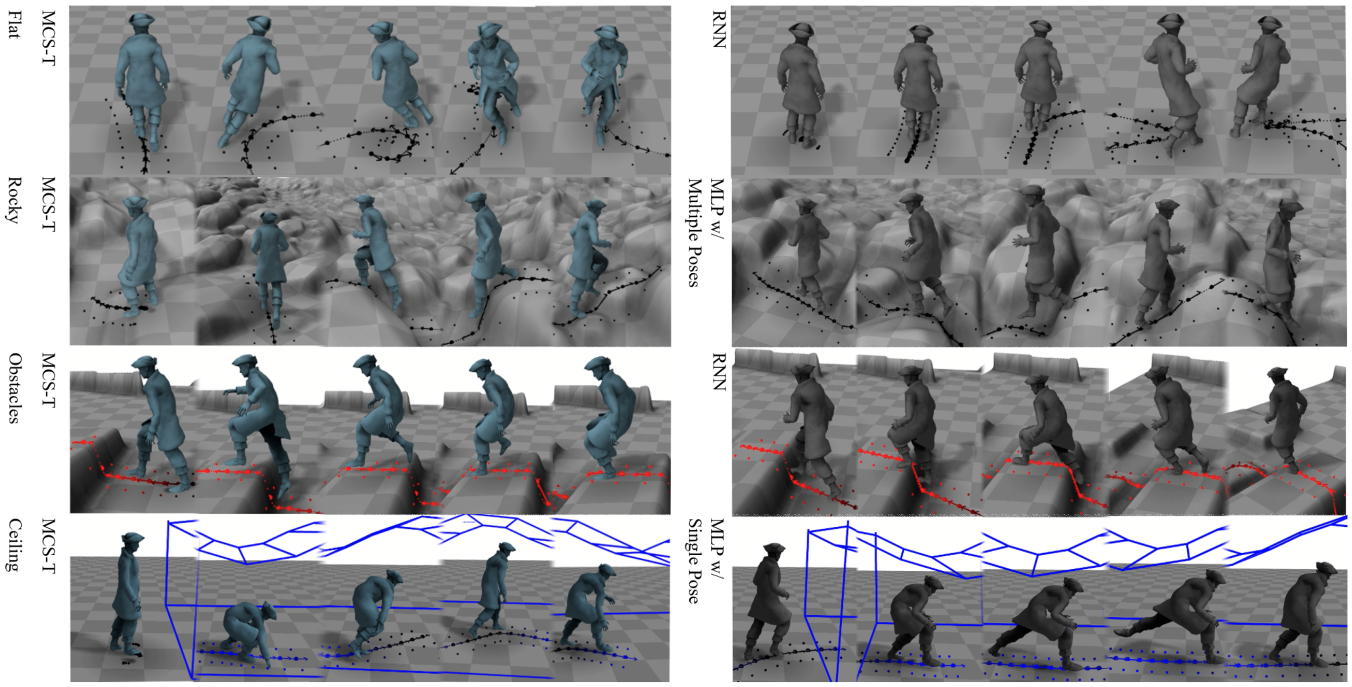

Figure 3: Qualitative results of our MCS-T and other baselines under four scenarios: flat, rocky, obstacles and ceiling. The left side shows the motions synthesized by MCS-T and the right side provides examples that demonstrate the baseline limitations.

the new geometry, the motion was performed as smooth as the regular locomotion on a flat terrain. The reason could be that using several past poses in a simple manner is limited to the large redundant variations in the past.

RNN: An LSTM architecture (Lee, Lee, and Lee 2018) was adopted for this biped locomotion dataset. The past memory enables LSTM to predict motion of higher quality. Nevertheless, it still suffered from the slow motion transition issue. As shown in Figure 3, the character could be floating when transiting between motion and was unable to jump over the obstacles timely and obviously. The reason is that the hidden memory prevented the RNN model from quickly reaching transitional states of a jumping motion.

Phase-functioned neural network: Rather than relying on past poses to constrain motion prediction, PFNN (Holden, Komura, and Saito 2017) utilised the foot contact phase for motion disambiguation. The qualitative results of our MCS-T were very closely to those of PFNN in a wide range of scenarios (see supplementary materials). Our method does not require the task-specific auxiliary information, which only relies on the past motion data generally available.

Moreover, to quantitatively evaluate whether the produced motion is responsive to control signals and transits to different motions timely, the average joint angle update per second as a metrics for motion dynamics was compared. A higher joint angle update represents more dynamic motion produced and faster transition between frames. The results are listed in Table 1, which indicate that MCS-T is able to produce much more agile motion than the task-agnostic RNN, while being comparable to the task-specific PFNN (Holden, Komura, and Saito 2017) method.

4.4 Ablation Study

An ablation study was conducted to demonstrate the effectiveness of the multi-scale skeleton representation and the control signal-aware mechanism in our encoder and decoder, respectively. The quantitative evaluation is listed in Table 1.

Multi-scale skeletons with an extra middle scale: In addition to the two skeleton scales, we experimented with one extra scale called as a middle scale. It aggregated the joints into a level between the two existing levels. However, the three-scale scheme did not contribute to the overall performance and produced stiff motions especially under scenarios with quick and frequent transitions such as obstacles and ceiling scene. The potential reason could be that the increased model complexity deteriorates the capability of motion prediction and produces sub-optimal solution.

Multi-scale skeletons: Without multi-scale skeletons, the motion dynamics dropped significantly, especially in the obstacles scene. Jumping motion became less responsive and sometimes the dynamics were too weak to observe. Thus, incorporating coarse-level skeletons helped exploit the motion patterns during a transition from a global perspective.

Control signal-aware decoder: Besides the motion prediction network, our decoder is driven by the control signals as well, which are adopted as the queries of the decoding attentions. Alternatively, by simply using a conventional self-attention mechanism to construct this decoder, it led to less motion dynamics. The most obvious case is in the ceiling scenario, where the motion appeared to be jittery and unstable during the transition between the walking and the crouching in the ceiling scenario.

Method	Phase	Flat			Rocky			Obstacles			Ceiling			Average		
		Full	Arm	Leg	Full	Arm	Leg	Full	Arm	Leg	Full	Arm	Leg	Full	Arm	Leg
PFNN	✓	106.5	100.6	135.9	**128.7**	145.0	**156.0**	109.1	110.9	143.9	139.9	130.9	**187.0**	121.1	121.9	155.7
MLP w/ Single Pose	✗	71.5	65.4	90.2	86.5	90.3	110.5	78.7	71.2	108.9	109.7	103.7	142.0	86.6	82.7	112.9
MLP w/ Multiple Poses	✗	94.0	88.1	122.7	95.2	91.3	131.0	85.7	76.3	122.6	115.1	100.8	161.8	97.5	89.1	134.5
RNN	✗	83.3	78.4	107.1	83.2	76.4	115.5	85.5	80.4	122.3	123.7	107.9	174.2	93.9	85.8	129.8
MCS-T (ours)	✗	**110.9**	**107.5**	**142.8**	126.7	**149.0**	151.6	**116.1**	**121.4**	**150.7**	**140.0**	137.0	184.6	**123.4**	**128.7**	**157.4**
+ Middle scale	✗	105.5	101.6	135.4	109.2	117.2	140.8	104.6	108.6	140.1	122.0	111.5	164.5	110.3	109.7	145.2
- Multi-scale skeleton	✗	96.3	87.6	127.1	112.6	123.8	143.4	91.9	89.4	127.2	124.1	114.1	167.1	106.2	103.7	141.2
- Control signal-aware attention	✗	94.6	87.7	125.2	105.6	109.1	140.2	90.7	85.2	129.3	137.5	**145.8**	173.7	107.1	107.0	142.1

Table 1: Quantitative comparison in terms of the average joint angle update per second (degree/s) ↑ for different methods including MCS-T under four motion scenarios: Flat, Rocky, Obstacles, and Ceiling. The angle updates are further divided into full body with all joints, arm and leg joints. The highest value is in bold and the second highest value is underlined.

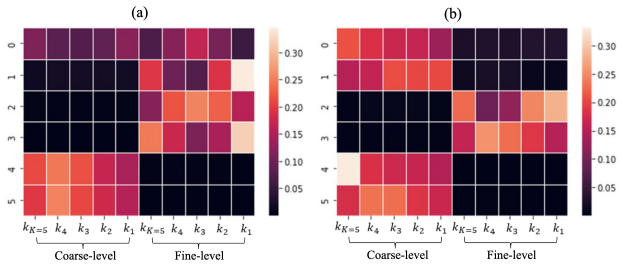

Figure 4: Visualization of the attention map of MCS-T, where the first layer of the decoder is shown, including (a) transitional and (b) non-transitional scenarios. The x-axis represents the past motion indices and the y-axis indicates the 6 attention heads.

4.5 Multi-scale and Control Signal-aware Motion Attentions

Our experiments show that MCS-T is able to synthesize motions with the highest quality and alleviate the slow transition issue. This lies in the attention mechanisms of MCS-T, which adaptively addresses the sequential motion context. The attention map of the decoder's first layer is visualised in Figure 4 to show how MCS-T performs attentions for different cases. Figure 4 (a) is for a frame of motion transition from a jumping state to a jogging state. Most attention heads focused on the fine-level skeletons, especially in more recent frames, as the further past frames were not very relevant during this motion transition. Additionally, two attention heads paid even attentions to the coarse-scale motion, which learned global motion patterns for faster motion transition. Figure 4 (b) is for a non-transitional case where the character remains the jogging state, the attentions are evenly distributed on all positions of the past poses, especially with more attention heads focusing on the coarse-level. The reason could be that the coarse motion sequence provides sufficient spatio-temporal patterns for predicting this kind of motions with strong recurring patterns.

4.6 Limitations & Future Work

There are two major limitations of our proposed MCS-T. First, our MCS-T method may not always synthesize the beam walking motion well. For example, as shown in Figure 5, informed by the special terrain geometry, the character performed a hand balancing motion. However, MCS-T did not always launch this motion. It could be due to the small percentage of beam walking motion in the training data ($\tilde{2}\%$) and imbalance learning strategies should be considered. Second, since MCS-T exploits the past motion history, the error accumulation could happen with a very low chance. The character motion could get stuck in weird poses for a very short period but can escape from it by providing new control signals. Robust noise-based learning could be conducted for alleviating such error accumulation. In our future work, besides addressing these limitations, we will investigate an adaptive strategy for selecting past frames, such as exploring network architecture search (NAS) (Zimmer, Lindauer, and Hutter 2021) and token evaluation strategies.

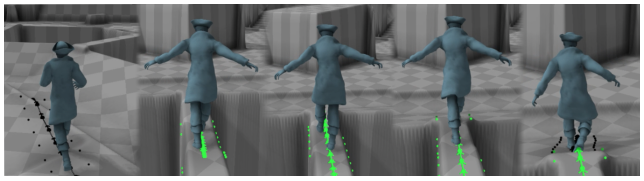

Figure 5: Illustration of a limitation of MCS-T, where the hand balancing motion is not well synthesized when the character is walking on a beam.

5 Conclusion

In this paper, we present MCS-T as a transformer-based task-agnostic character motion control method. With multi-scale graph representation, it aims to produce responsive and dynamic motions without explicitly using auxiliary information. Specifically, MCS-T involves an encoder-decoder design, where the encoder formulates the spaio-temporal motion patterns of past poses from multi-scale perspectives and the decoder takes a control signal into account for predicting the next pose. Our experiments on a public dataset have demonstrated that MCS-T can produce results comparable to those of the state-of-the-art methods which explicitly using auxiliary information. We also investigate the limitations of our method for future improvement.

Acknowledgments

This study was partially supported by Australian Research Council (ARC) grant #DP210102674.

References

Aksan, E.; Kaufmann, M.; Cao, P.; and Hilliges, O. 2021. A spatio-temporal transformer for 3D human motion prediction. In *International Conference on 3D Vision*, 565–574. IEEE.

Arikan, O.; and Forsyth, D. A. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics*, 21(3): 483–490.

Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lučić, M.; and Schmid, C. 2021. Vivit: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision*, 6836–6846.

Bhattacharya, U.; Childs, E.; Rewkowski, N.; and Manocha, D. 2021. Speech2affectivegestures: Synthesizing co-speech gestures with generative adversarial affective expression learning. In *29th ACM International Conference on Multimedia*, 2027–2036.

Buttner, M. 2019. Machine Learning for Motion Synthesis and Character Control in Games. *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.

Clavet, S. 2016. Motion matching and the road to next-gen animation. In *Game Developer Conference*.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Dang, L.; Nie, Y.; Long, C.; Zhang, Q.; and Li, G. 2021. MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction. In *IEEE/CVF International Conference on Computer Vision*, 11467–11476.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent network models for human dynamics. In *IEEE International Conference on Computer Vision*, 4346–4354.

Ghosh, A.; Cheema, N.; Oguz, C.; Theobalt, C.; and Slusallek, P. 2021. Synthesis of compositional animations from textual descriptions. In *IEEE/CVF International Conference on Computer Vision*, 1396–1406.

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

Henter, G. E.; Alexanderson, S.; and Beskow, J. 2020. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics*, 39(6): 1–14.

Holden, D.; Kanoun, O.; Perepichka, M.; and Popa, T. 2020. Learned motion matching. *ACM Transactions on Graphics*, 39(4): 53–1.

Holden, D.; Komura, T.; and Saito, J. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics*, 36(4): 1–13.

Jang, D.-K.; Park, S.; and Lee, S.-H. 2022. Motion Puzzle: Arbitrary Motion Style Transfer by Body Part. *ACM Transactions on Graphics*.

Kania, K.; Kowalski, M.; and Trzciński, T. 2021. TrajeVAE–Controllable Human Motion Generation from Trajectories. *arXiv preprint arXiv:2104.00351*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kovar, L.; Gleicher, M.; and Pighin, F. 2008. Motion graphs. In *ACM SIGGRAPH 2008 classes*, 1–10.

Lee, J.; Chai, J.; Reitsma, P. S.; Hodgins, J. K.; and Pollard, N. S. 2002. Interactive control of avatars animated with human motion data. In *Annual Conference on Computer Graphics and Interactive Techniques*, 491–500.

Lee, K.; Lee, S.; and Lee, J. 2018. Interactive character animation by learning multi-objective control. *ACM Transactions on Graphics*, 37(6): 1–10.

Lee, Y.; Wampler, K.; Bernstein, G.; Popović, J.; and Popović, Z. 2010. Motion fields for interactive character locomotion. In *ACM SIGGRAPH Asia 2010 papers*, 1–8.

Li, M.; Chen, S.; Zhao, Y.; Zhang, Y.; Wang, Y.; and Tian, Q. 2020. Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 214–223.

Li, P.; Aberman, K.; Zhang, Z.; Hanocka, R.; and Sorkine-Hornung, O. 2022. GANimator: Neural Motion Synthesis from a Single Sequence. *arXiv preprint arXiv:2205.02625*.

Li, Z.; Zhou, Y.; Xiao, S.; He, C.; Huang, Z.; and Li, H. 2017. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*.

Ling, H. Y.; Zinno, F.; Cheng, G.; and Van De Panne, M. 2020. Character controllers using motion vaes. *ACM Transactions on Graphics*, 39(4): 40–1.

Liu, Z.; Lyu, K.; Wu, S.; Chen, H.; Hao, Y.; and Ji, S. 2021. Aggregated multi-gans for controlled 3D human motion prediction. In *AAAI Conference on Artificial Intelligence*, volume 35, 2225–2232.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 143–152.

Mao, W.; Liu, M.; and Salzmann, M. 2020. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, 474–489. Springer.

Martínez-González, A.; Villamizar, M.; and Odobez, J.-M. 2021. Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In *IEEE/CVF International Conference on Computer Vision*, 2276–2284.

Mason, I.; Starke, S.; and Komura, T. 2022. Real-Time Style Modelling of Human Locomotion via Feature-Wise Transformations and Local Motion Phases. *ACM on Computer Graphics and Interactive Techniques*, 5(1): 1–18.

Mazzia, V.; Angarano, S.; Salvetti, F.; Angelini, F.; and Chiaberge, M. 2022. Action Transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, 124: 108487.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Peng, X. B.; Abbeel, P.; Levine, S.; and van de Panne, M. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4): 1–14.

Petrovich, M.; Black, M. J.; and Varol, G. 2021. Action-conditioned 3D human motion synthesis with transformer vae. In *IEEE/CVF International Conference on Computer Vision*, 10985–10995.

Plizzari, C.; Cannici, M.; and Matteucci, M. 2021. Spatial temporal transformer network for skeleton-based action recognition. In *International Conference on Pattern Recognition*, 694–701. Springer.

Starke, S.; Zhang, H.; Komura, T.; and Saito, J. 2019. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6): 209–1.

Starke, S.; Zhao, Y.; Komura, T.; and Zaman, K. 2020. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics*, 39(4): 54–1.

Starke, S.; Zhao, Y.; Zinno, F.; and Komura, T. 2021. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics*, 40(4): 1–16.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wang, J.; Xu, H.; Narasimhan, M.; and Wang, X. 2021. Multi-Person 3D Motion Prediction with Multi-Range Transformers. *Advances in Neural Information Processing Systems*, 34: 6036–6049.

Williams, R. J.; and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2): 270–280.

Zhang, H.; Starke, S.; Komura, T.; and Saito, J. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics*, 37(4): 1–11.

Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; and Ding, Z. 2021. 3D human pose estimation with spatial and temporal transformers. In *IEEE/CVF International Conference on Computer Vision*, 11656–11665.

Zimmer, L.; Lindauer, M.; and Hutter, F. 2021. Auto-Pytorch: Multi-fidelity metalearning for efficient and robust AutoDL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9): 3079–3090.

Zinno, F. 2019. ML Tutorial Day: From Motion Matching to Motion Synthesis, and All the Hurdles In Between. *Game Developer Conference*.