Pacific Graphics (2021)
M. Okabe, S. Lee, B. Wuensche, and S. Zollmann (Editors)

Short Paper

Human Motion Synthesis and Control via Contextual Manifold Embedding

Rui Zeng[1,2] , Ju Dai[2] , Junxuan Bai[1,2] , Junjun Pan[1,2†] and Hong Qin[3‡]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Department of Computer Science, Stony Brook University (SUNY), Stony Brook, NY, USA

Abstract

Modeling motion dynamics for precise and rapid control by deterministic data-driven models is challenging due to the natural randomness of human motion. To address it, we propose a novel framework for continuous motion control by probabilistic latent variable models. The control is implemented by recurrently querying between historical and target motion states rather than exact motion data. Our model takes a conditional encoder-decoder form in two stages. Firstly, we utilize Gaussian Process Latent Variable Model (GPLVM) to project motion poses to a compact latent manifold. Motion states could be clearly recognized by analyzing on the manifold, such as walking phase and forwarding velocity. Secondly, taking manifold as prior, a Recurrent Neural Network (RNN) encoder makes temporal latent prediction from the previous and control states. An attention module then morphs the prediction by measuring latent similarities to control states and predicted states, thus dynamically preserving contextual consistency. In the end, the GP decoder reconstructs motion states back to motion frames. Experiments on walking datasets show that our model is able to maintain motion states autoregressively while performing rapid and smooth transitions for the control.

CCS Concepts

• ***Computing methodologies*** → ***Motion processing; Motion capture; Motion path planning; Learning latent representations;***

1. Introduction and Related Works

Modeling motion dynamics as a motion controller from mocap data is challenging. The main difficulties hinge upon two aspects. On the one hand, it isn't easy to distinguish between the intra-class variation of motion states and the natural randomness of human behavior. Unlike robotic machines, motion cannot be made precisely by muscles and mind at each repeat. While adequate data are essential in data-driven methods, the gap between state variations and natural randomness becomes hard to disambiguate. On the other hand, explicit control signals are usually inadequate to explain the complex relationship among skeleton joints and temporal coherence. For example, changing the walking speed defined by singular joint velocity omits other joint positions, which could cause inconsistency when lacking contextual transitions in original data.

Many studies represent motion sequences as certain states along with clips and random variations at each posture. For motion without apparent separations, motion states are usually manually labeled. Such deterministic methods tend to perform averaged motion, regardless of a complex expression. In these cases, motion randomness is modeled by adding noises to data, generating a realistic synthesis similar to the original one. As a matter of fact, probabilistic methods are intuitive to model motion randomness. [MC12, KH10] separate clips of periodic walking into several phase states. Stable states and random variations are connected via Gaussian Process (GP) mappings in each phase. [WFH07, UFG*08] apply latent variable models (LVMs) on GP, where motion states are implicitly represented as well. GPLVM constructs a latent manifold where latent variables of motion poses are statistically distributed. The states vary consistently as the latent path transits from one region to another.

While GPLVM is able to represent states and randomness of motion, it doesn't directly model motion dynamics. First-order Markov assumptions are widely used to construct dynamic systems based on poses [WFH08, DTL11, UK12]. For fast response to control, Levine et al. [LWH*12] generates novel transitions by discovering latent manifold structure. However, precomputed strategies are not usually feasible in real-time applications since there could be a transition deficiency. Recently, neural networks are widely used for modeling dynamic systems. Variants of the recurrent neural network are applied to update model parameters by previous states and current input continuously [SZKZ20, GWE*20]. Mao et al. [MLS20] adapts motion prediction performance of RNN by adding an attention module, which morphs original RNN input by measuring sequential similarities. Holden et al. [HSK16, HKS17] use deep network structures for motion synthesis and control. These deterministic approaches do not directly model randomness, so that similar motion states could be ambiguous. Recently, probabilistic models have been proposed for character motion control.

† Corresponding Author: Junjun Pan, pan_junjun@buaa.edu.cn
‡ Corresponding Author: Hong Qin, qin@cs.stonybrook.edu

DOI: 10.2312/pg.20211383

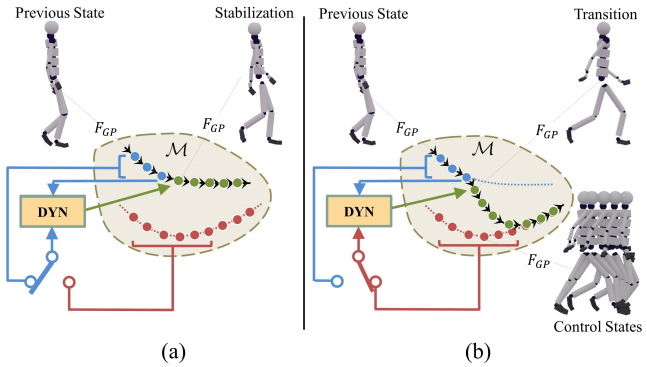

Figure 1: *Our workflow of modeling motion dynamics with (a) default control states and (b) target control states.*

Henter et al. [HAB20] proposes normalizing flow for generative locomotion synthesis. Ling et al. [LZCvdP20] apply variational autoencoders to construct two-frame manifold to disambiguate motion context. Different from them, our model constructs the manifold from motion poses. Contextual information is learned from the network afterwards to generate motion transitions.

To address the above issues, this paper proposes a novel approach to model motion dynamics for motion synthesis and control, combining neural networks with GPLVM to learn the spatio-temporal relationship from motion data. The workflow is illustrated in Figure 1. Basically, the function of motion dynamics (named DYN) are modeled on a manifold embedding $\mathcal{M}$ by GPLVM (Section 2.1). DYN receives current states (top arrow) and control (down arrow) states as context, outputting the estimation of next motion states on $\mathcal{M}$. The predicted motion states are back-projected into a specific motion pose by F_{GP}. DYN conducts motion predictions under default (blue) or target (red) control states, which performs the synthesis of stabilization (Figure 1 (a)) and transition (Figure 1 (b)) respectively. DYN is then fitted by a neural network composed of a recurrent unit and an attention unit. The recurrent unit is used to predict latent variables as output, as it maintains and updates the hidden dynamic states over time. The attention unit is employed to edit the prediction by considering control states as the target. Details of the network are in Section 2.2. The attention mechanism adaptively compares prediction states with control states, which performs better than equal-weight methods without attention. Qualitative and quantitative results are in Section 3.

In summary, the technical contributions can be listed as follows,

- We propose a novel framework for motion dynamics, which embeds observed motion data as distributions of latent variables on a manifold, then fitting the dynamical function by RNN based neural network.
- We design an attention unit in the network for target control of motion state, which adaptively combines the historical context with the target context.
- We utilize the proposed model for locomotion synthesis, in which motion states are able to be arbitrarily edited by default and target control.

2. Method

A motion sequence is composed of poses with continuous timestamps. Each pose is represented as frame data $Y_i \in \mathbb{R}^D$ with fixed joint parameters, such as positions or rotations. The human pose is constrained by skeleton topology and cooperation between relative joints, which also limits the degree of freedom of Y_i. As a matter of fact, variations of Y_i could be represented by fewer dimensions in a specific group of motion sequences. We consider a probabilistic, non-linear, non-parametric latent motion embedding:

$$Y_i = F_{GP}(X_i, \mathbf{Y}), \tag{1}$$

where the latent variable $X_i \in \mathbb{R}^d$ represents corresponding motion state of the pose. F_{GP} contains unified information shared by the pose dataset $\mathbf{Y} = \{Y_i\}$, which helps to reconstruct latent variables back to frame data. In Section 2.1, we explain how to construct $\mathbf{X} = \{X_i\}$ and F_{GP} from $\mathbf{Y}$.

Next, we apply motion dynamics to the latent embedding. Future motion is determined by its context, which refers to 1) previous motion states as we do not wish to contain future information, and 2) target states that we wish to switch. We define the next motion state as a function of current motion state X_t and control states X_t^c:

$$X_{t+1} = DYN(X_t, X_t^c), \tag{2}$$

where the previous state $X_t \in \mathbb{R}^{1\times d}$ obeying first order Markov assumption as in [WFH08, DTL11]. The first-order assumption guarantees contextual consistency while responding quickly to state variation. $X_t^c \in \mathbb{R}^{n\times d}$ are control motion states indicating the target. Algorithm 1 provides motion synthesis with default and target control states.

Algorithm 1 Motion synthesis with control

Input: Initial motion state $X_0 \in \mathbb{R}^{1\times d}$, target state sequences $C_T \in \mathbb{R}^{T\times d}$
Output: Motion frames $Y_T \in \mathbb{R}^{T\times D}$

```
function DYNAMICS(X_0, C_T)
    for t = 0 → T − 1 do
        if C_t is not set then
            X_t^c ← X_t
        else
            X_t^c ← C_t
        end if
        X_{t+1} = DYN(X_t, X_t^c)
        Y_{t+1} = F_GP(X_{t+1})
    end for
    return Y_T
end function
```

2.1. Latent Motion Embedding

Motion Manifold Construction. Firstly, GPLVM maps unlabeled motion poses to a compact, dimension-reduced latent manifold. The manifold describes the prior distribution of latent variables that represent corresponding motion states. Unified motion patterns, such as the general motion style, are concluded by reconstruction GP. The unified mapping function separates shared information from motion states for a distinguishing latent manifold. By considering the pose at each frame, the original motion dataset is stacked into a data matrix $\mathbf{Y} \in \mathbb{R}^{N\times D}$, where N is the number of total frames, and D is the number of frame dimensions each.

Following Lawrence [Law05], GPLVM finds a latent space which embeds high dimensional data Y to compact, low-dimensional latent variables $\mathbf{X} \in \mathbb{R}^{N\times d}$, where $d \ll D$. It formulates

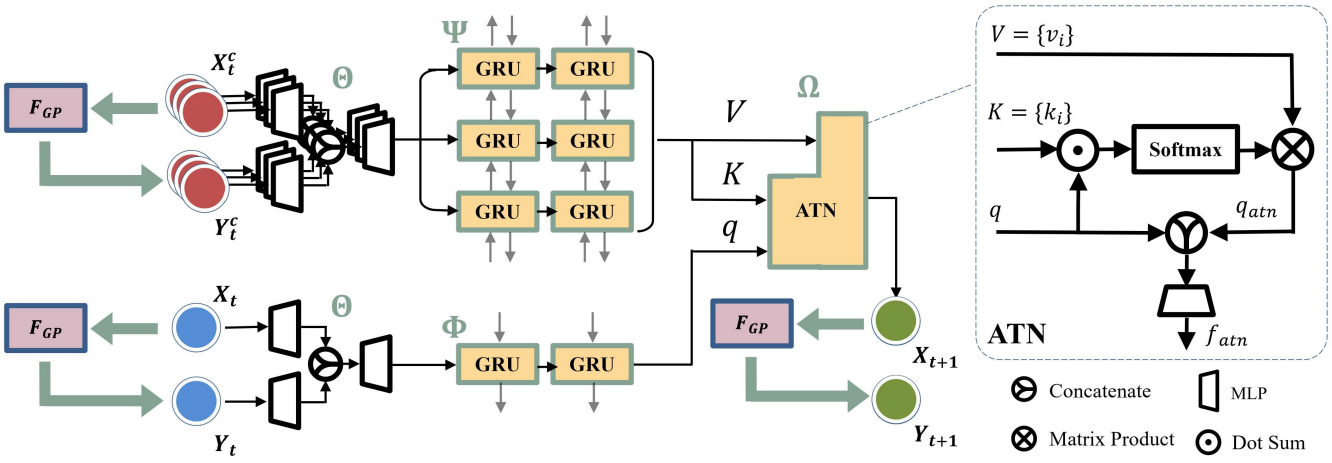

Figure 2: *The network structure. It is composed of the embedding unit Θ, the temporal context unit Φ, the control context unit Ψ, and the attention unit Ω. F_{GP} reconstructs the input and output of the network to frame data.*

Gaussian Process (GP) by the conditional probability of the observation data Y from the latent variables X, $P(\mathbf{Y}|\mathbf{X},\alpha)$. The kernel $\mathbf{K}(\alpha)$ indicates the covariance of $\mathbf{X}$, as well as the metric of motion states variation in the latent space. A standard metric of $\mathbf{K}$ is the radial basis function (RBF) kernel.

Latent variables $\mathbf{X}$ as well as kernel parameters α are optimized with the objective function of the negative log-likelihood. It is accomplished by maximum a posteriori probability estimate (MAP). Once the optimization is complete, given arbitrary latent variable X, the distribution of a motion pose Y is a Gaussian $p(Y) = \mathcal{N}(\mu,\Sigma)$ in a closed form:

$$\mu(X) = F_{GP}(X,\mathbf{Y}) = \mathbf{K}_X\mathbf{K}^{-1}\mathbf{Y}, \tag{3}$$

$$\Sigma(X) = k(X,X) - \mathbf{K}_X^T\mathbf{K}^{-1}\mathbf{K}_X, \tag{4}$$

where $\mathbf{K}_X$ measures the covariance between $\mathbf{Y}$ and X by the same kernel metric mentioned above. μ represents the mean estimation of the pose from X, which is also the reconstruction method from motion states to frame data. Σ indicates the confidence in it. The lower the variance is, the higher the confidence is in reconstructing plausible motion pose.

2.2. Contextual Motion Dynamics

Dynamical Net. In this section, we explain how to fit equation 2 with our DYN. The whole structure of DYN is illustrated in Figure 2. DYN is composed of the embedding unit Θ, the temporal context unit Φ, the control context unit Ψ, and the attention unit Ω.

Firstly, the embedding unit Θ extracts feature of X_t before temporal prediction:

$$\Theta(X_t) = f_m\{f_c\,[f_m(X_t), f_m(F_{GP}(X_t))]\}, \tag{5}$$

where f_m indicates Multilayer Perceptron (MLP) and f_c refers to vector concatenation. From the experiment, we find that adding the features of reconstructed frame data speeds up network convergence. The embedding unit Θ is also used for feature extraction of X_t^c. Next, the temporal context unit Φ makes a prediction from previous motion states X_t at each timestamp t. Φ consists of two-layer Gated Recurrent Unit (GRU) f_{G1}, f_{G2}:

$$\Phi_{t+1}(X_t) = \Phi(\Theta(X_t), h_t) = f_{G2}\{f_{G1}\,[\Theta(X_t), h_{t1}]\,, h_{t2}\}, \tag{6}$$

where $h_t = (h_{t1}, h_{t2})$ indicating the hidden states of GRU at current timestamp t. GRU updates h_t for each iteration. Since h_t is calculated recurrently, hidden states could simultaneously encode short-term and long-range context information. For example, h_K iterates K times from initialization, accumulating context at each $t \in [0, K-1]$. Note that when predicting Φ_1, the hidden states $h_{t,0}$ are determined by Ψ, as the same technique by Cho et al. [CvMG*14].

The network introduces control states X_t^c to morph the prediction Φ_{t+1} from previous states. After the embedding unit, Θ, the control context unit Ψ is designed to recover the temporal context. The order of X_t^c is crucial for states consistency but not explicitly encoded in the feature yet. The formulation of Ψ is almost the same with Φ. Different from Equation 6, $\Theta(X_t^c)$ take place of $\Theta(X_t)$ as the input. The output of $\Theta(X_t^c)$ scale n times as the input contains n continuous states. Another difference is the bi-directional because we hope to extract as much context forward time as well as backward time.

Finally, feature of control states Ψ_{t+1} and prediction states Φ_{t+1} are merged by an attention unit Ω. Attention module takes a query (q), key (K), value (V) as input to obtain attentioned query (q_{atn}). After that, (q, q_{atn}) are concatenated for keeping complete features through forwarding:

$$q_{atn}(q,K,V) = \sum_{i=1}^{n} softmax(q^T k_i)v_i, \tag{7}$$

$$f_{atn}(q,K,V) = f_m\{f_c\,[q_{atn}(q,K,V), q]\}. \tag{8}$$

Taking Φ_{t+1} as q, Ψ_{t+1} as K and V , Ω obtains X_{t+1} as the output of DYN. Here we set $K = V$ to directly compare between Φ_{t+1} and Ψ_{t+1}, since they encodes X_t^c and feature prediction of X_{t+1} to the same domain:

$$X_{t+1} = \Omega_{t+1}(X_t, X_t^c) \tag{9}$$

$$= f_{atn}\left[\Phi_{t+1}(X_t), \Psi_{t+1}(X_t^c), \Psi_{t+1}(X_t^c)\right]. \tag{10}$$

The controlled data frame Y_{t+1} is obtained by Equation 1.

Network Training. We measure the prediction accuracy in both the latent manifold space and the observed motion frame domain. The loss of our network is defined as

$$L = \sum_{i=1}^{T}\left(\left\|X_{i+1} - DYN(X_i, X_i^h)\right\|^2 + \lambda\left\|Y_{i+1} - F_{GP}\left[DYN(X_i, X_i^h)\right]\right\|^2\right), \tag{11}$$

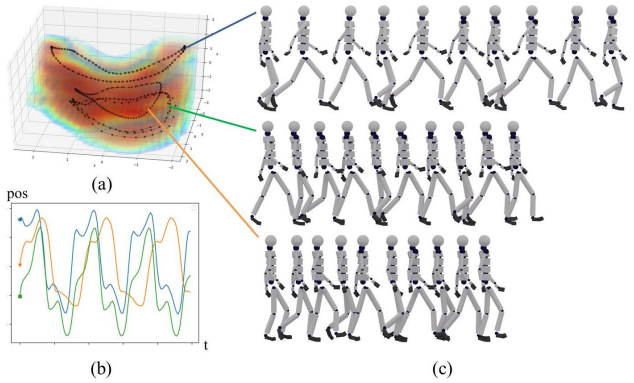

Figure 3: *Qualitative result for synthesis with stabilization, including (a) locations of latent variables, (b) curves of the right knee height changing with time and (c) samples of avatar movement.*

where X_{i+1} is corresponding latent variables of frame data Y_{i+1}. X_i^h is the historical state sequence from training data that matches to X_i. The first term is the loss in the latent manifold space and the second term is in the motion pose space. The hyperparameter λ balances the relative importance of the two loss terms, which we emperically set to 0.1 in optimization.

Our model learns the contextual distribution in the latent space. During training, control states X_i^h match to historical state X_i in the dataset. During synthesis, control states could be arbitrary target states. Equation 11 forces the network output to share the same distribution in both cases, even though the input states in the test stage don't appear in the training stage. Both original and control states contribute to the next state of the system. After training, the network learns to output by the combination of state inputs. For motion transitions under control, we manually set control states so that the output of DYN morphs to match the new target. Since the target states gradually take part in the network input, the influence on the output presents a smooth transition.

For network training, we need to prepare training pairs, including motion clips, latent variables. We cut motion clips by a sliding window with fixed width. We set $T = 20$ for our walking dataset. Overlap between adjacent clips is set to 1 for continuous contextual dependency. All sequences downsample from 120 Hz to 60 Hz. We design several standard techniques for the training. Adam algorithm is employed to automatically calculates the derivatives. We set the learning rate to 0.0001 and then multiplied it by 0.95 after each epoch. Dropout strategy is applied, and we set the dropping rate to 0.1. The model is optimized in a mini-batch manner with a batch size of 32 for 200 epochs. The training consumes around 20 hours on an NVIDIA Geforce RTX 2080Ti GPU.

3. Experiments and Evaluations

Our DYN is trained with the CMU locomotion dataset. We choose 12 sequences, 4000 frames of walking data from the same subject. Each sequence varies in moving speed and stride length. We compare the proposed model with networks without the attention unit and the aid of GPLVM. Stabilization synthesis tests how motion states preserve stable for long sequential synthesis, while transition synthesis tests on how states varies from the original to the target. The performance is evaluated both qualitatively and quantitatively.

3.1. Stabilization Synthesis

Once the initial states are set, DYN is able to synthesize motion with any length. When no control states are received, the dynamics of DYN are calculated as Figure 1 (a). In this case, synthetic motion sequences are supposed to preserve stable motion states for different initialization, such as movement speed and stride length, while the walking phase changes periodically.

Figure 3 shows the synthetic results by three initial sequences with different speeds, indicated by color lines including the slow (orange), the medium (green), and the fast (blue). Latent variables of synthetic motion sequences are showed in Figure 3 (a). Different colors indicate different confidence scores on the GP reconstruction of plausible motion pose, high with red and low with blue. We see all the latent variables locates on regions with high confidence, which means plausible reconstruction quality of poses. Also, each group of the latent variables gathers in a specific region, indicating that poses with different walking speeds are clearly separated without intersection. The circles formed by latent paths explain the periodic nature of walking. Movements of an avatar visualize the reconstructed results. Figure 3 (c) shows reconstruction quality in a complete view. Frames are sampled after constructing 1.5 seconds in 6 Hz from 60 Hz of synthetic results. Considering the variation of neighboring frames, the difference of speeds is still distinguishing, demonstrating the synthesis stabilization. The curves in Figure 3 (b) indicate how the heights of the right knee change with time. The movement patterns of a single joint repeat periodically, demonstrating the synthesis stabilization in a partial view.

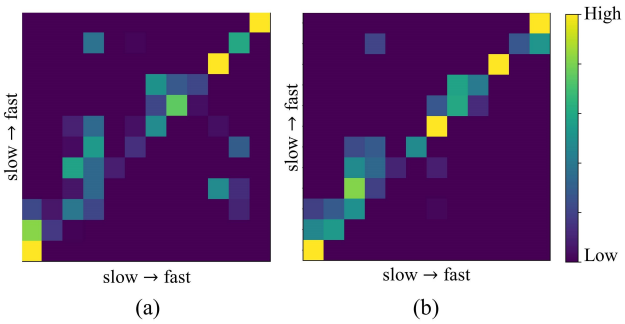

Figure 4: *Heatmaps for measuring stabilization by MMD, (a) for the model without attention and (b) for DYN.*

Furthermore, we report quantitative results by statistics. The direct comparison by frame error can not be made because original motion sequences have fixed lengths that are much shorter than synthetic sequences. We introduce MMD for quantitative comparison with inadequate ground truth. Instead of analyzing frame by frame, Maximum Mean Discrepancy (MMD) considers data distribution between sequences. Recent work has extensively used adversarial ideas [LCC*17, WSH19] as discriminators to distinguish between real and false data samples. In this paper, we randomly choose 3 phases for each of the 12 original motion sequences as initialization to synthesize 1000 length of motion. MMD calculates every 40 frames of synthetic sequences of the 12 original motion sequences. When the lowest value is obtained, we choose its index to label the motion states of the clip. Then we count the ratio of synthetic labels for a group with the same initialization label. The results are shown in Figure 4 as heat maps. It compares the synthesis stabilization of DYN (b) with the network without attention (a).

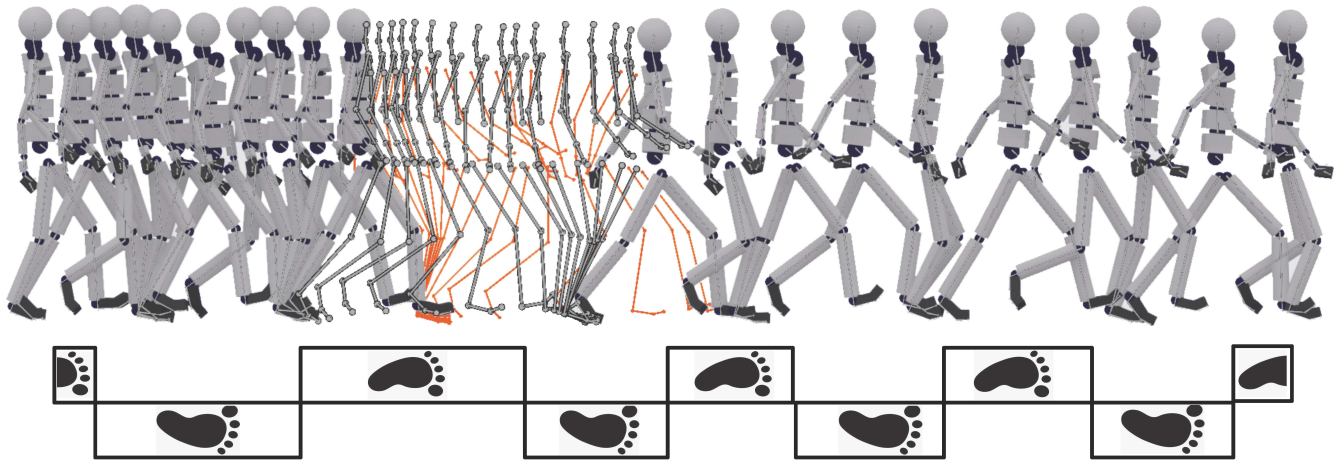

Figure 5: *Motion transition from one states to another. Footprints indicates touching on the ground.*

The element (i, j) stands for the MMD ratio of label j in initialization group i. The order of labels ascends as walking accelerates so that motion states are similar to their neighbors. From Figure 4 (b), we can see that the MMD ratio distributes higher at diagonal, indicating that the synthesis is distributed most closed to the original initialization. However, Figure 4 (a) shows regions with high ratios beyond diagonal, which means weak stabilization of motion states. The reason for the large deviation might be caused by the lack of attention unit. The network without attention treats previous states equally, regardless of the difference of motion dynamics at every phase. The attention unit Ω in DYN adaptively learns motion dynamics by judging the importance at each timestamp, which is more flexible for handling contextual variations.

3.2. Transition Synthesis

The transition synthesis results are illustrated in Figure 5. It can be oberserved that the initial (left avatars) is relaxed and slow, while the target (right avatars) is intense and fast. The transition (skeleton) is smooth without acute variation. Even without ground truth, our model still performs smooth transitions from one state to another. The reasons contribute to latent embedding and contextual state control. On the one hand, the latent embedding by F_{GP} distinguishes between minor noises from motion randomness and significant variations of motion state, which helps to find the intrinsic difference and blend them little by little as time proceeds. On the other hand, the attention unit in *DYN* adaptively chooses between the control states that contribute the most to the realistic context, thus preventing sharp steer of the joint path.

Motion states should respond as quickly as possible when control happens. To compare the performance of state transition, we randomly set initial and target control states to produce motion transitions. The control sequence $X_t^c (t \in [1, K+C])$ is constructed to control motion from states of index label p to q, where the index label of X_t^c equals to p if $0 < t < K$ and equals to q if $K < t < K+C$. We sample 400 sequences to apply 1-time variation by randomly selecting the initial phase and label pair (p, q). Frame length of initial states and control states are set to constant, $K = 100$, $C = 20$. Samples that fail to preserve initial states are deleted. We propose similar metric in Section 3.1 by utilizing MMD as the measurement. The MMD is calculated in a window size of 10 with a sliding step of 1, and the current motion state is denoted by the label with the lowest value. Because the initial and target label is not constant, we represent them as L_p and L_q, respectively.

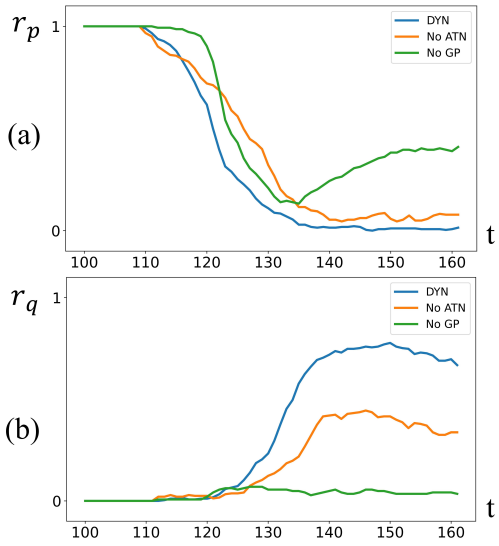

Figure 6: *Agility of response to target control, comparing among the model of DYN (blue), without attention (orange) and without GPLVM (green). The ratio of initial states r_p (a) and target states r_q vary along time.*

Figure 6 shows the statistical comparison results of the proposed DYN, model without attention, and model without GPLVM. The label of L_t at each timestamp t equals L_p, L_q, or others, indicating before, after, and during the transition, respectively. Their ratio of them at t are named as $r_{p,t}$, $r_{q,t}$, $r_{o,t}$ respectively, $r_{p,t} + r_{q,t} + r_{o,t} = 1$. Here we only draw figures of r_p and r_q. Figure 6 (a) shows curves of r_p changes by timestamp t. It reveals how the motion leaves from initial states. The target states are set at $t \in [110, 130]$ named L_q, where $t \in [0, 110]$ refers to initial states named L_p. All three curve starts to decrease when target control happens. DYN (blue) and the model without attention (orange) drops earlier than the model without GP (green), which indicates quicker responses to various con-

trol states. Two of them reach the lowest at about $t = 140$, while most sample sequences do not preserve states r_p. Note that $r_{p,t}$ of the model without GP increases after $t = 140$, which indicates the sample sequences that fail to transit to target states as their states recover to L_p. Figure 6 (b) shows curves of r_q by timestamp t. It reveals how the motion reaches target states. We see the curve r_q of DYN (blue) raises the fastest, followed by the model without attention (orange). The former reaches a higher ratio than the latter after $t = 40$, indicating that more sample sequences complete the transition to target states with a control length of $C = 20$. At the end of each curve, the ratio drops a bit. The reasons are that some sample sequences fail to preserve target states after X_t^c disappears. In this case, more prolonged control states are required. In addition, the curve of the model without GPLVM (green) barely not raises, as most of the sample sequences fail the transition to the target states.

4. Limitations and Discussions

The dataset we use only contains walking samples with different speeds, which are represented as swing strides of arm and leg and the frequency of step. More variant motion data, *e.g.*, running and direction turning, can be used to explore the model's capabilities. In this case, direction invariant representation is required for reasonable transition across motion clips. Furthermore, the motion manifold is constructed by GPLVM, the required storage and optimization time increase in proportion to the dataset scale. The Scalability of the latent embedding is under investigation for various motion datasets. In addition, our network structure is quite simple, while powerful and informative networks with deep layers often lead to better performance.

5. Conclusions

In this paper, we propose a novel framework to model motion dynamics by integrating GPLVM with a neural network. We apply GPLVM, a probabilistic, non-linear, non-parametric latent variable model, to construct a motion manifold. Distributions on the manifold describe the randomness of motion and movements of latent variables on the manifold represent the intra-class variation of motion states. To interpret motion dynamics in manifold space, we further design a recurrent-based neural network. The recurrent unit maintains historical context to keep temporal consistency, while the attention unit adaptively responds to the control context to produce a smooth transition to target motion states.

We have evaluated our model for synthesis stabilization and motion transition qualitatively and quantitatively. The qualitative results validate our model's ability to maintain walking speed, stride, and period for long-term motion synthesis, while changes smoothly when arbitrary control occurs. The quantitative comparison with models without attention and GP further proves that our model can better match the features of motion behavior and quickly responds to instant target control.

6. Acknowledgments

This work was supported by National Natural Science Foundation of China (No.61872020, U20A20195), Beijing Natural Science Foundation Haidian Primitive Innovation Joint Fund (L182016), Shenzhen Research Institute of Big Data, Shenzhen, 518000, China Postdoctoral Science Foundation (2020M682827), Baidu academic collaboration program, and Global Visiting Fellowship of Bournemouth University.

References

[CvMG*14] CHO K., VAN MERRIENBOER B., GÜLÇEHRE Ç., BAHDANAU D., BOUGARES F., SCHWENK H., BENGIO Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing* (2014), pp. 1724–1734. 3

[DTL11] DAMIANOU A. C., TITSIAS M. K., LAWRENCE N. D.: Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems* (2011), pp. 2510–2518. 1, 2

[GWE*20] GHORBANI S., WLOKA C., ETEMAD A., BRUBAKER M. A., TROJE N. F.: Probabilistic character motion synthesis using a hierarchical deep latent variable model. *Computer Graphics Forum 39*, 8 (2020), 225–239. 1

[HAB20] HENTER G. E., ALEXANDERSON S., BESKOW J.: Moglow: probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics 39*, 6 (2020), 236:1–236:14. 2

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Transactions on Graphics 36*, 4 (2017), 42:1–42:13. 1

[HSK16] HOLDEN D., SAITO J., KOMURA T.: A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics 35*, 4 (2016), 138:1–138:11. 1

[KH10] KWON T., HODGINS J. K.: Control systems for human running using an inverted pendulum model and a reference motion capture sequence. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2010), pp. 129–138. 1

[Law05] LAWRENCE N. D.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research 6* (2005), 1783–1816. 2

[LCC*17] LI C., CHANG W., CHENG Y., YANG Y., PÓCZOS B.: MMD GAN: towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems* (2017), pp. 2203–2213. 4

[LWH*12] LEVINE S., WANG J. M., HARAUX A., POPOVIC Z., KOLTUN V.: Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics 31*, 4 (2012), 28:1–28:10. 1

[LZCvdP20] LING H. Y., ZINNO F., CHENG G., VAN DE PANNE M.: Character controllers using motion vaes. *ACM Transactions on Graphics 39*, 4 (2020), 40. 2

[MC12] MIN J., CHAI J.: Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics 31*, 6 (2012), 153:1–153:12. 1

[MLS20] MAO W., LIU M., SALZMANN M.: History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision* (2020), vol. 12359, pp. 474–489. 1

[SZKZ20] STARKE S., ZHAO Y., KOMURA T., ZAMAN K. A.: Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics 39*, 4 (2020), 54. 1

[UFG*08] URTASUN R., FLEET D. J., GEIGER A., POPOVIC J., DARRELL T., LAWRENCE N. D.: Topologically-constrained latent variable models. In *International Conference on Machine Learning* (2008), vol. 307, pp. 1080–1087. 1

[UK12] UKITA N., KANADE T.: Gaussian process motion graph models for smooth transitions among multiple actions. *Computer Vision and Image Understanding 116*, 4 (2012), 500–509. 1

[WFH07] WANG J. M., FLEET D. J., HERTZMANN A.: Multifactor Gaussian process models for style-content separation. In *International Conference on Machine Learning* (2007), vol. 227, pp. 975–982. 1

[WFH08] 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 (2008), 283–298. 1, 2

[WSH19] WANG W., SUN Y., HALGAMUGE S. K.: Improving MMD-GAN training with repulsive loss function. In *International Conference on Learning Representations* (2019). 4