WalkTheDog: Cross-Morphology Motion Alignment via Phase Manifolds

Peizhuo Li
ETH Zurich
Switzerland
peizhuo.li@inf.ethz.ch

Sebastian Starke
Meta Reality Labs
United Kindom
sstarke@meta.com

Yuting Ye
Meta Reality Labs
USA
yuting.ye@meta.com

Olga Sorkine-Hornung
ETH Zurich
Switzerland
sorkine@inf.ethz.ch

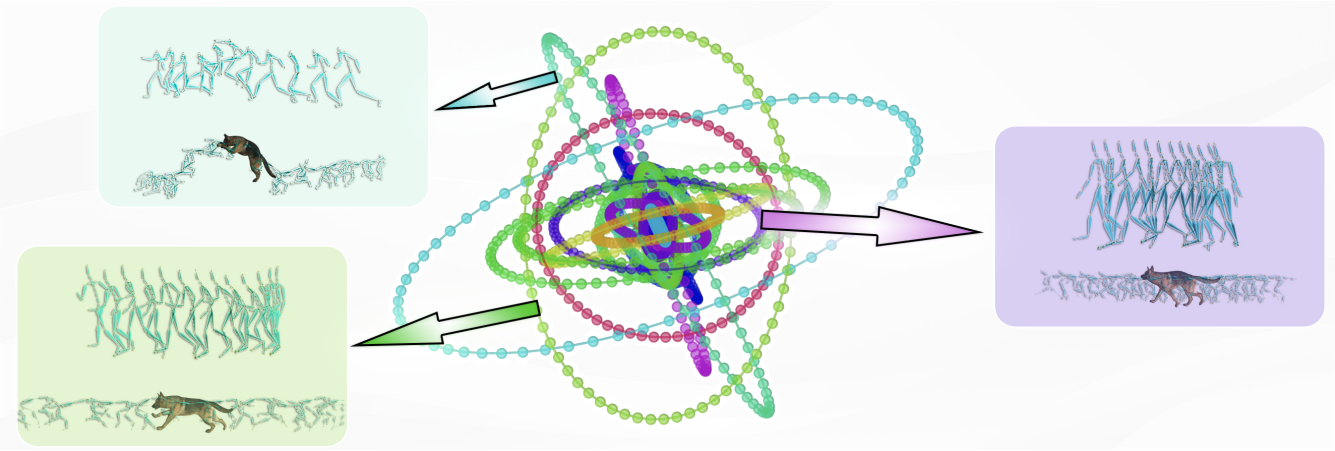

Figure 1: Our phase manifold $\mathcal{P}$ is learned from datasets with drastically different skeletal structures without any supervision. Each connected component in the manifold, visualized in a different color, is an ellipse embedded in high-dimensional space. Semantically similar motions from different characters are embedded into the same ellipse.

ABSTRACT

We present a new approach for understanding the periodicity structure and semantics of motion datasets, independently of the morphology and skeletal structure of characters. Unlike existing methods using an overly sparse high-dimensional latent, we propose a phase manifold consisting of multiple closed curves, each corresponding to a latent amplitude. With our proposed vector quantized periodic autoencoder, we learn a shared phase manifold for multiple characters, such as a human and a dog, without any supervision. This is achieved by exploiting the discrete structure and a shallow network as bottlenecks, such that semantically similar motions are clustered into the same curve of the manifold, and the motions within the same component are aligned temporally by the phase variable. In combination with an improved motion matching framework, we demonstrate the manifold's capability of timing and semantics alignment in several applications, including motion retrieval, transfer and stylization. Code and pre-trained models for this paper are available at peizhuoli.github.io/walkthedog.

CCS CONCEPTS

• **Computing methodologies** → **Motion processing**; *Machine learning*.

KEYWORDS

character animation, motion alignment, deep learning

ACM Reference Format:
Peizhuo Li, Sebastian Starke, Yuting Ye, and Olga Sorkine-Hornung. 2024. WalkTheDog: Cross-Morphology Motion Alignment via Phase Manifolds. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641519.3657508

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07.
https://doi.org/10.1145/3641519.3657508

1 INTRODUCTION

What is in common between a dog's walk and a human's walk, or that of an ogre? Understanding the intrinsic structure and semantics of motion, regardless of the character's morphology and skeletal structure, lies at the heart of character animation research. In particular, motion retargeting and style transfer often rely on precise alignment of source and target motions in the form of paired data, posing severe limitations on their applicability. To make use of large heterogeneous datasets, common approaches organize motions in a discrete graph (motion graph [Kovar et al. 2002]) or a contiguous field (motion field [Lee et al. 2010]) with great success for many control and synthesis tasks. However, they fall short of handling different character designs and diverse content in a unified space. We argue that the main drawback of these methods is that their similarity metrics are based on *extrinsic* pose features, which also encode features of the skeleton and motion semantics. In this work, we aim to learn an *intrinsic* motion representation that is agnostic to the character morphology and can disentangle motion structure from semantics without any labels or other supervision signals.

An intrinsic property of motion is its periodic structure. Common locomotion such as walking and running can be effectively parameterized by a linear *phase* variable for motion control problems [Holden et al. 2017; Peng et al. 2018]. To this end, we propose a latent representation that decomposes motions into a 1D phase and discrete amplitude vectors. This latent space forms a one-dimensional manifold that consists of multiple connected components, where each component is an ellipse corresponding to a discrete amplitude vector. We term it a *disconnected 1D manifold*. The possible choices of amplitudes are learned through vector quantization [Van Den Oord et al. 2017], similar to a clustering process. The discrete amplitude vectors serve as a narrow bottleneck to regularize unsupervised learning of semantic motion clusters. The number of amplitude vectors reflects the semantic diversity of the motion dataset.

Formally, we propose a vector quantized periodic autoencoder (VQ-PAE) that embeds motions into a disconnected 1D manifold. The encoder projects a short input sequence into a 1D continuous phase variable and a latent code from a small codebook. The decoder reconstructs the input sequence using a simple two-layer convolution network with limited capacity to prevent memorization. The codebook and the autoencoder are jointly learned end-to-end. The small codebook size and the simple decoder enforce the semantic structure in the latent space. For example, idling and running will be far apart in the codebook because the decoder cannot reconstruct both from the same or similar input. On the other hand, jogging and running may have to share the same code or be close if the codebook size is small, as they are sufficiently similar when phase-aligned. When learning VQ-PAEs from multiple characters, such as a dog and a human, each character has their own VQ-PAE to handle their unique morphology and skeletal features, but they all share the same latent codebook. As a result, they are naturally clustered semantically as enforced by the codebook size, without any explicit supervision, but solely based on the intrinsic structure of the motion. Note that the VQ-PAE is not meant to be a generative model, given the intentional bottlenecks in the codebook and the decoder. We make use of the latent representation but discard the decoder after training.

We validate our design by learning VQ-PAEs from both a human dataset and a dog dataset with a shared codebook. Examining the average pose at each point on the manifold reveals that the learned embeddings are both timing- and semantics-aligned between the two characters. This highly structured and aligned phase manifold opens up new possibilities for motion data organization, retrieval, transfer and stylization. The phase manifold embedding can be flexibly integrated with existing motion synthesis pipelines. For example, given an unseen human motion, we can search the shared manifold for the nearest neighbor of dog motion with similar semantics and timing. We can further combine motion matching [Büttner and Clavet 2015] with linear time warping supported by the 1D phase variable to transfer semantically similar motions between the human and the dog, without any paired data or pre-defined mapping among the skeletal structures. In addition, we demonstrate applications of motion characterization on the MOCHA dataset [Jang et al. 2023].

Our key contributions are summarized as follows:

- A novel phase manifold designed for both timing and semantics alignment. We also show that the manifold is compact, disentangled, and highly structured.
- A demonstration of using narrow bottlenecks and intrinsic structure of motions to achieve alignment among heterogeneous datasets, without any supervision, self-supervised losses, or skeletal structure correspondences.
- Applications with an improved motion matching framework on the phase manifold for motion retrieval, transfer and stylization.

2 RELATED WORK

In this section, we review the related work mainly on clustering and organizing motion capture datasets. We take a deeper look into the works related to *phase* in terms of motion organization. Motion retargeting and style transfer are also related, in the sense of bridging different characters and distilling the core content of motions. We briefly review them at the end of this section.

Organizing and clustering motion dataset. Organizing a large-scale motion capture dataset is a difficult yet important task for applications. Graph-based methods [Kovar et al. 2002; Arikan and Forsyth 2002] find similar patterns of poses, cluster them into the same node, and use the edges to represent transition motions between nodes. This approach allows interactive control by mapping the user control to paths on the graph. Min and Chai [2012] use key-frame-based segmentation to construct the graph structure and build probabilistic-based to increase the expressiveness and diversity of generated motion. At the same time, similar probabilistic models on graph structures are proposed. Park et al. [2011] organize a motion capture dataset using context-free grammar learned from segments clustered with Partitioning Around Medoids (PAM) algorithm based on pose level similarity. Aristidou et al. [2018] notice that semantic similarity may not be reflected by low-level representations such as poses and propose to learn a high-dimensional representation of motion motifs and motion signatures. Since the discrete structures lack expressiveness and responsiveness, Lee

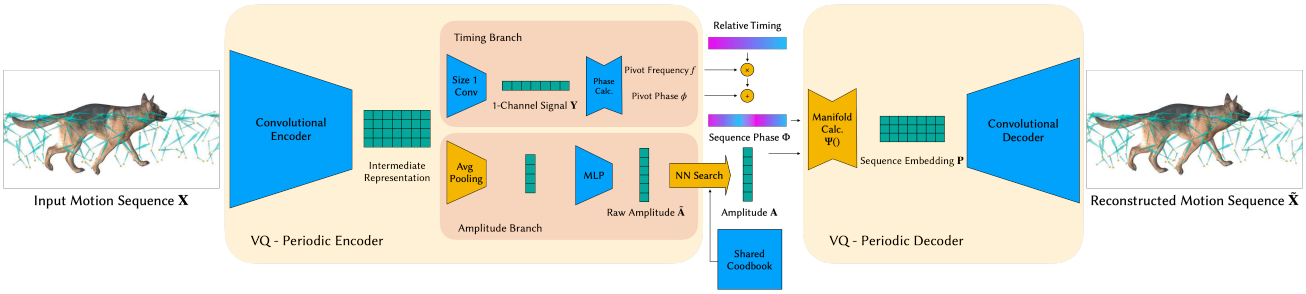

Figure 2: Architecture of VQ-PAE. Starting with a short motion sequence $\mathbf{X} \in \mathbb{R}^{J \times T}$, the encoder learns an intermediate representation using convolution. The representation is fed into the timing and the amplitude branch for predicting the phase ϕ, the frequency f and the amplitude A of the pivot frame (rendered with mesh). A vector quantization (i.e. nearest neighbor search) is used in the amplitude branch to ensure the structure of the phase manifold. Note the codebook $\mathcal{A}$ is shared among multiple VQ-PAEs. We calculate the embedding P of the sequence assuming the frequency and amplitude stay constant in the sequence. The predicted phase manifold sequence is then passed through a convolutional decoder to reconstruct the input motion. Components with learnable parameters are marked in blue.

et al. [2010] take another approach and learn a continuous field for motion. However, to generate motion, reinforcement learning is required to progress in the learned field. Motion matching [Büttner and Clavet 2015] skips the organization of data and directly finds the best match of the current state and control signal in the dataset and replays the sequence. It is among the methods with the highest quality and is widely used in industry. With the progress of tokenization [Dhariwal et al. 2020; Rombach et al. 2022] with VQ-VAE [Van Den Oord et al. 2017], it is becoming more and more popular for organizing human motion [Geng et al. 2023], and demonstrated great success in multi-modal tasks [Guo et al. 2022; Siyao et al. 2022; Zhang et al. 2023]. However, completely discretizing the latent space makes it difficult to capture the continuous nature of motion, and the learned latent space is usually less compact, making it difficult to construct a shared latent space for multiple characters.

Exploiting periodicity and phase. Using phase and frequency domains to organize motion is closely related to our method. Park et al. [2002] propose to align motions by the key-frames such as foot-contact as key poses, and warping the motion with the guidance of key poses so motions at different speeds can be interpolated. It serves as an early inspiration for the introduction of phase and is part of the inspiration for our frequency-scaled motion matching in Section 4. Unuma et al. [1995] demonstrate style transfer can be performed in the frequency domain. The introduction of *phase* into neural networks demonstrated great success. It originally started with 1D phase [Holden et al. 2017] coming from a semi-automated labeling process and quickly expanded into multiple dimension hand-crafted phase that is able to handle complex and non-periodic motions [Starke et al. 2019]. Starke et al. [2020] attach a phase to each limb to deal with complex multiple contacts. DeepPhase [Starke et al. 2022] proposes periodic autoencoders (PAEs), enabling learning on a continuous and expressive multi-dimensional phase manifold. It has been proven successful in applications like pose estimation [Shi et al. 2023] and motion in-betweening [Starke et al. 2023]. However, the learned phases and amplitudes are usually entangled, making it difficult to separate the timing and high-level semantics of motion. The sparsity of motion data leaves a large portion of the phase manifold invalid and can lead to implausible motions when used for synthesis, and it will be even more challenging to learn a shared phase manifold for multiple characters. We provide a comparison with DeepPhase of the disentanglement of phase manifolds in Section 5.3.

Motion retargeting and style transfer. Gleicher [1998] proposed one of the earliest method for motion retargeting, by directly optimizing on low-level motion representations. Other optimization-based methods [Lee and Shin 1999; Choi and Ko 2000; Tak and Ko 2005] are also proposed to improve the result. However, those methods mainly focus on transferring motions to a new skeleton, instead of building a common representation for different characters. This is only addressed with deep learning based methods [Villegas et al. 2018; Lim et al. 2019; Aberman et al. 2020a; Li et al. 2023], where a common latent space among different characters is learned. Although they may not need paired data, the same or homeomorphic skeletons are required such that the learning and auxiliary losses can be applied, while our method does not have this constraint. It is also demonstrated by Kim et al. [2022] that with paired examples, it is possible to retarget between bipeds and quadrupeds. In combination with the view of dynamic systems, Kim et al. [2020] show that a common latent space for two similar dynamic systems for bipeds or pendulums can be learned with a pair of autoencoders. For style transfer, Xia et al. [2015] propose to use KNN search to build the style regression model. Aberman et al. [2020b] disentangles the style code and content code with a labeled dataset. Jang et al. [2023] make a further step to distinguish stylization and characterization, pushing the boundary of style transfer further. Our method can achieve a similar effect by treating each style as a separate dataset and using the alignment ability to transfer the content.

3 PHASE MANIFOLD

In this section, we introduce the design of our disconnected 1D phase manifold, which allows us to align motions with a single timing variable while creating a narrow bottleneck and forcing our framework to cluster semantically similar motions into the same connected component of the phase manifold. We then describe

our vector quantized periodic autoencoder (VQ-PAE) to learn the embedding of motions from one dataset. Finally, we explain the approach for training multiple VQ-PAEs on different datasets into a common phase manifold.

3.1 Disconnected 1D phase manifold

We construct a phase manifold such that the timing is controlled by a 1D phase variable. Given an input motion sequence $\mathbf{X} \in \mathbb{R}^{J \times T}$, where J and T indicate the degree of freedom and the number of frames, respectively, we aim at mapping each frame $\mathbf{X}_i$ to a point $p = \Psi(\mathbf{A}, \phi) \in \mathbb{R}^d$ on the phase manifold $\mathcal{P}$, parameterized by a 1D phase variable $\phi \in (-\frac{1}{2}, \frac{1}{2}]$ and a vector amplitude $\mathbf{A} \in \mathbb{R}^{2d}$. We choose the mapping Ψ to be

$$\Psi(\mathbf{A}, \phi) = \mathbf{A}^0 \sin(2\pi\phi) + \mathbf{A}^1 \cos(2\pi\phi), \tag{1}$$

an ellipse embedded in $\mathbb{R}^d$, where $\mathbf{A}^0, \mathbf{A}^1 \in \mathbb{R}^d$ are the first and second half of $\mathbf{A}$, respectively. In contrast to ϕ, which can take any value in $(-\frac{1}{2}, \frac{1}{2}]$, $\mathbf{A}$ can only be chosen from a finite codebook $\mathcal{A} \subset \mathbb{R}^{2d}$ with size K. Thus, our phase manifold $\mathcal{P}$ can be formally defined as $\{\Psi(\mathbf{A}, \phi) \mid \mathbf{A} \in \mathcal{A}, \phi \in (-\frac{1}{2}, \frac{1}{2}]\}$. This construction gives us a latent space that is a collection of ellipses, as shown in Figure 1, where we collect samples of $\mathcal{P}$ by uniformly sampling the phase ϕ on each ellipse $\mathcal{P}_i = \{\Psi(\mathbf{A}_i, \phi) \mid \phi \in (-\frac{1}{2}, \frac{1}{2}]\}$ and use PCA to reduce dimension for visualization. In this manifold, a class of motions with similar semantics is embedded into the same ellipse. Note there is a one-to-one mapping between ellipses $\mathcal{P}_i$ and amplitudes $\mathbf{A}_i$. This allows us to flexibly scale the size of the bottleneck by changing the size of $\mathcal{A}$. A properly chosen bottleneck size is the key to learning an expressive yet semantically aligned phase manifold.

3.2 Vector quantized periodic autoencoder

Starke et al. [2022] introduce periodic autoencoder (PAE) for learning a continuous phase manifold. To learn a discrete amplitude space, we utilize the vector quantization technique to cluster the amplitude vectors into a learnable codebook $\mathcal{A}$. The architecture of our vector quantized periodic autoencoder (VQ-PAE) is demonstrated in Figure 2.

A desired mapping from motion to phase manifold should satisfy the following properties for an input motion sequence $\mathbf{X} \in \mathbb{R}^{J \times T}$ containing roughly a cyclic motion:

- *Phase linearity*: the phase ϕ should increase as linearly as possible over time.
- *Amplitude constancy*: the amplitude $\mathbf{A}$ should be as constant as possible over time.

To achieve those two properties, we use a similar approach as PAE [Starke et al. 2022] by using an encoder to predict the amplitude $\mathbf{A}$, the phase ϕ and the frequency f, which is the change rate of phase over time, at the center frame, *i.e.* the *pivot* frame, of a short input motion sequence $\mathbf{X}$. We then assume the two properties hold for the whole input sequence $\mathbf{X}$ and extrapolate the phase linearly with the predicted frequency to the whole sequence. We calculate the embeddings using Equation (1) with extrapolated phases and amplitudes. A decoder is then used to reconstruct the input motion sequence from the predicted embedding. A decent reconstruction

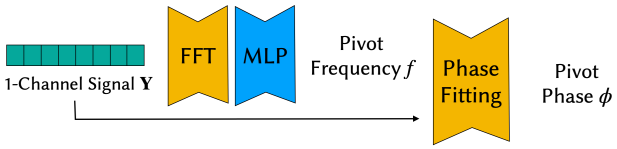

Figure 3: Details of phase calculation module.

can only be achieved if the learned mapping is close to phase linear and amplitude constant.

Encoder. The encoder consists of a 2-layer 1D convolutional network mapping the input to an intermediate representation. The intermediate representation is then fed into two branches, namely the timing branch and the amplitude branch, each responsible for the prediction of phase, frequency and amplitude, respectively. We denote the relative timing of each frame in the sequence w.r.t. the pivot frame as $\mathcal{T} = \{t_i\}_{i=1}^{T}$, where $t_i = [i - (T+1)/2]\,\Delta_T$. Note we choose T to be an odd number such that the pivot frame is unique, and Δ_T is the frame time of the dataset.

Timing branch. The timing branch starts with a 1D convolution with kernel size 1, mapping the multi-dimensional intermediate representation to a 1-channel temporal signal. A phase calculation module is used on the temporal signal to predict the phase ϕ and frequency f. The detailed architecture of the phase calculation module is shown in Figure 3. PAE [Starke et al. 2022] uses the power of each frequency bin calculated by fast Fourier transform (FFT) as weights to calculate the average frequency. However, it produces unstable frequencies as the input phase shifts even when it is a sinusoidal signal with a non-integer frequency. We find that using a small multi-layer perceptron (MLP) on the powers produces more robust frequency prediction. We use the equations presented by Mason [2022] to calculate the phase ϕ, which helps with the fact that ϕ is not a continuous parameterization of the phase manifold. Please refer to the supplementary material for more details.

Amplitude branch. As the amplitude should be nearly constant over time, we first apply an average pooling on the temporal axis on the intermediate representation. An MLP is followed to get a raw prediction of amplitude $\tilde{\mathbf{A}}$. Since the possible choices of amplitude are finite, we use a vector quantization layer to find the nearest neighbor $\mathbf{A} = \arg\min_{\mathbf{A}_i \in \mathcal{A}} \|\tilde{\mathbf{A}} - \mathbf{A}_i\|_2$.

Decoder. With phase linearity and amplitude constancy assumptions, the phase variable of the input motion can be calculated by $\Phi = \phi + f \cdot \mathcal{T}$ with the relative timing $\mathcal{T}$. The embedding of the input motion sequence can then be calculated by $\mathbf{P} = \Psi(\mathbf{A}, \Phi)$. The decoder is a 2-layer 1D convolutional network that maps the embedding back to the original motion space.

Loss function. We use the following loss function to train our VQ-PAE:

$$\begin{aligned} \mathcal{L}_{\text{rec}} &= \|\mathbf{X} - \tilde{\mathbf{X}}\|_2, \\ \mathcal{L}_{\text{vq}} &= \|\text{sg}(\tilde{\mathbf{A}}) - \mathbf{A}\|_2 + \|\tilde{\mathbf{A}} - \text{sg}(\mathbf{A})\|_2, \end{aligned} \tag{2}$$

where $\tilde{\mathbf{X}}$ is the reconstructed motion sequence, $\text{sg}(\cdot)$ is the stop gradient operator. The first loss is the reconstruction loss of the VQ-PAE the second loss is the vector quantization loss [Van Den Oord

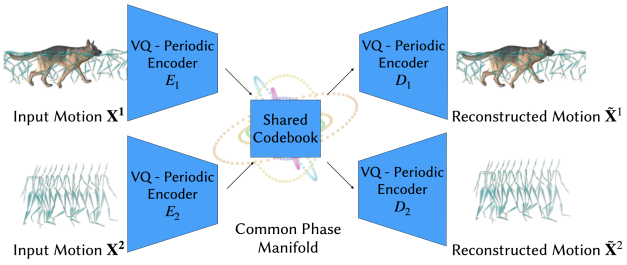

Figure 4: Overview of training multiple VQ-PAEs on heterogeneous datasets. A common phase manifold is guaranteed by using a shared codebook $\mathcal{A}$.

et al. 2017]. The total loss is

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}}, \tag{3}$$

and λ_{vq} is a hyperparameter. For a detailed network architecture and hyperparameter settings, please refer to the supplementary material.

3.3 Learning a common phase manifold among VQ-PAEs

To align motions among different datasets, a common phase manifold can be learned with a shared codebook $\mathcal{A}$ and no additional supervision as shown in Figure 4. Without loss of generality, we illustrate the training process of two VQ-PAEs on two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ with different skeletal structures in this section. The training process can be easily extended to more datasets.

The loss for training two VQ-PAEs can be written as

$$\mathcal{L} = \mathcal{L}_{\text{rec1}} + \mathcal{L}_{\text{rec2}} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}}, \tag{4}$$

where $\mathcal{L}_{\text{rec1}}$ and $\mathcal{L}_{\text{rec2}}$ are the reconstruction losses of the two VQ-PAEs, and $\mathcal{L}_{\text{vq}}$ is the vector quantization loss of the shared codebook $\mathcal{A}$. During training, we optimize two VQ-PAEs at the same time. Note that we do not need any skeletal topology correspondences due to the use of simple 1D convolution and MLPs.

However, directly optimizing Equation (4) can lead to situations where part of the entries in $\mathcal{A}$ are only used by one VQ-PAE, causing disparity in the embeddings of $\mathcal{D}_1$ and $\mathcal{D}_2$. This is also a common problem for regular VQ-VAEs that many entries in the codebook are not used. Zheng and Vedaldi [2023] propose a simple yet effective reinitialization technique to solve this problem for training with one VQ-VAE. We adapt their method to the training of multiple VQ-PAEs.

Reinitialization of $\mathcal{A}$. At the beginning of training, $\mathcal{A}$ is initialized with uniform ditribution $\mathcal{U}[-1/K, 1/K]$ and $K = |\mathcal{A}|$. For simplicity, we discuss the reinitialization of $\mathcal{A}$ for one VQ-PAE. At each training iteration step, the decayed running average usage $N_i^{(t)}$ at the t-th iteration of each entry $\mathbf{A}_i$ in $\mathcal{A}$ by the VQ-PAE is updated by

$$N_i^{(t)} = \gamma N_i^{(t-1)} + (1-\gamma)\frac{n_i^{(t)}}{N}, \tag{5}$$

where $n_i^{(t)}$ is the number of times $\mathbf{A}_i$ is used by the VQ-PAE at the t-th iteration, N is the number of amplitudes produced by the encoder being quantized at each iteration and γ is the decay rate. Intuitively, entries with low usage are more likely to be reinitialized. We choose to reinitialize the less frequently used entries to a randomly chosen amplitude produced by the encoder. Formally, the reinitialization target Z_i of entry $\mathbf{A}_i$ is sampled such that closer outputs are preferred to maximize the utilization of the codebook by

$$\mathbb{P}(Z_i = \tilde{A}_k) \propto \exp(-\|\mathbf{A}_i - \tilde{\mathbf{A}}_k\|_2), \tag{6}$$

where $\{\tilde{\mathbf{A}}_k\}$ are the raw amplitudes predicted by the encoder in this iteration. At an update step, every entry in the codebook is linearly interpolated to the reinitialization target with a weight α_i by

$$\alpha_i = \exp\left(-N_i \frac{10}{1-\gamma} - \epsilon\right), \tag{7}$$

$$\mathbf{A}_i = (1-\alpha_i)\mathbf{A}_i + \alpha_i Z_i, \tag{8}$$

where ϵ is a small constant acting as a regularizer. We set α_i such that less frequently used $\mathbf{A}_i$ is interpolated more towards a randomly picked output of the encoder. Note the temporal superscript (t) is omitted for simplicity. Since the codebook is shared among multiple VQ-PAEs, the reinitialization of $\mathcal{A}$ is performed as the average update of all VQ-PAEs produced by Equation (8). An entry will converge to a stable value only if it is frequently used by all VQ-PAEs. For more details and reasoning of the setting of ϵ and γ, we refer the readers to the work of Zheng and Vedaldi [2023]. $\mathcal{A}$ is reinitialized at every training iteration before the gradient descent step.

Existing methods for learning a common latent space for motions with different skeletons [Villegas et al. 2018; Aberman et al. 2020a] usually require at least partially specified skeletal topology correspondences and additional implicit supervision such as cycle consistency [Zhu et al. 2017] and adversarial training [Goodfellow et al. 2020]. In contrast, our method achieves a common phase manifold with only a shared codebook $\mathcal{A}$ and no additional supervision, while semantics and timing alignment are naturally provided. Relying on the intrinsic periodicity of motions, this phase manifold can be used to model different character topologies including biped and quadruped without extra class-specific designs.

4 FREQUENCY-SCALED MOTION MATCHING

After the training of our VQ-PAEs, we can obtain the corresponding manifold embedding $p_i \in \mathcal{P}$ for every frame i in the dataset, by using the encoder to encode a 1-second motion sequence centered at frame i. Although relying on a single point on the manifold to represent a pose can be ambiguous, since the manifold is designed to be compact, a sequence of manifold points contains rich information to retrieve a motion sequence from the database. In fact, within a single cycle, the possible progress of phase, characterized by all possible mappings from time to phase $g\colon [0,1] \to (-\frac{1}{2}, \frac{1}{2}]$, is very expressive. To exploit the expressiveness in a sequence, we demonstrate that it is possible to use motion matching [Büttner and Clavet 2015] on the phase manifold and improve it with the explicit phase variable.

Given the phase embedding sequence $\mathbf{P}$ of an input motion sequence, we use motion matching to retrieve a motion sequence from the database, with phase embedding as the control signal in the classical motion matching algorithm. For more details of the

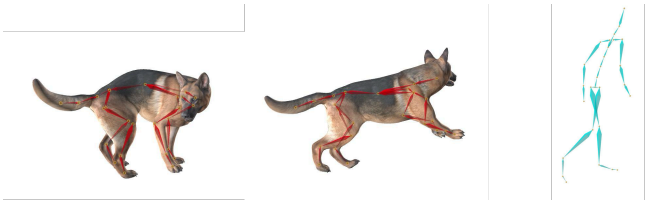

Figure 5: The running motions in Dog and Human-Loco dataset are of different frequencies. With frequency scaling, the motion with correct semantics is matched.

implementation, please refer to the supplementary material. We also compare the result performance of motion transfer on the dog-human setup with skeleton-aware networks (SAN) [Aberman et al. 2020a], the state-of-the-art for skeletal motion retargeting between different skeletal structures. SAN heavily requires end-effector velocity consistency between the source and target characters and struggles to transfer motion with a large difference in skeletal structure as shown in Figure 7.

A common problem in motion matching is there is a trade-off between responsiveness and smoothness. This can be mitigated by using variable replay lengths T_0 depending on the control signal. However, this requires a lot of manual tuning and is not robust to different inputs. In addition to this problem, directly applying vanilla motion matching for motion transfer is not ideal, as there might not be a motion clip in the database sharing the same semantics and frequency as the input motion, causing timing or semantics misalignment.

Algorithm 1 Frequency-scaled motion matching

$i \leftarrow 1$
$\mathbf{J}_{\text{start}} \leftarrow$ initial pose descriptor
while $i < T$ **do**
  $k = \arg\min_k \mathrm{c}(i,k)$
  Output $\mathbf{Y}_{k:k+t(k)}$ linearly interpolated to length $t(i)$
  $i \leftarrow i + t(i)$
  $\mathbf{J}_{\text{start}} \leftarrow \mathbf{J}_{k+t(k)}$
end while

With the help of our phase manifold, we can solve both problems by performing matching on a fixed number of cycles instead of a fixed number of frames. We demonstrate the details with 1 cycle and this can be easily extended to arbitrary cycles. Given a motion sequence $\mathbf{X}$ and its corresponding frequencies $\mathbf{F} = \{f_i\}$ predicted by the VQ-PAE, for each starting frame i, we define its period $t(i)$ as the first frame j such that $\sum_{k=i}^{i+j} f_k \Delta_T \geq 1$, thus $\mathbf{X}_{i:i+t(i)}$ roughly corresponds to one cycle of motion. During motion matching, instead of using a fixed number of frames T_0, we query every period of the input manifold, while the query is conducted on sequence with 1-period length in the database, as shown in Algorithm 1 and Equation (9). We denote the phase sequences of the database with $\mathbf{Q}$, the pose descriptor used to measure the similarity between frames with $\mathbf{J}$ and the pose with $\mathbf{Y}$. As a result, when more agile motions, *i.e.* motions with higher frequency and lower period, are involved, the matching steps will be carried out more frequently and thus the motion will be more responsive. On the other hand, by allowing interpolating the output motion to the same frequency as the input, we achieve a more accurate timing and semantics alignment, as shown in Figure 5. The transition cost function $\mathrm{c}(i,k)$ is defined as:

$$\mathrm{c}(i,k) = d(\mathbf{P}_{i:i+t(i)}, \mathbf{Q}_{k:k+t(k)}) + \lambda_1 \|\mathbf{J}_{\text{start}} - \mathbf{J}_k\|_2^2 + \lambda_2 \|t(i) - t(k)\|^2, \tag{9}$$

where $d(\mathbf{P}, \mathbf{Q})$ can be calculated by linearly interpolating their phases to the same length, chosen to be $1/\Delta_T$, and calculating the squared distance between them. The third term is introduced because we favor the motion with similar frequency and discourage large temporal interpolation. Note that $t(i)$ in the database and the fixed length interpolation of $\mathbf{Q}_{i:i+t(i)}$ can be precomputed, so the commonly used acceleration techniques for motion matching can still be applied to speed up the search.

5 APPLICATIONS AND EVALUATIONS

We evaluate our disconnected 1D phase manifold in terms of timing alignment and semantic alignment on several datasets. We show that our method can be used for improving motion matching with the predicted 1D phase. With our improved motion matching, we show that it is possible to achieve motion transfer and motion stylization by performing motion matching on the phase manifold.

5.1 Datasets

We use three datasets in our experiment. The Dog dataset [Zhang et al. 2018] and Human-Locomotion dataset [Starke et al. 2019] contain mostly locomotion including walking, running, jumping and idling. The MOCHA dataset [Jang et al. 2023] is a recently proposed highly stylized and characterized motion dataset. It contains a wide range of motions on different characters, including clown, ogre, princess, robot and zombie. For a detailed demonstration of the dataset, we refer the readers to Jang et al. [2023]. In the following sections, we train our VQ-PAEs with two combinations of datasets: Dog and Human-Locomotion and MOCHA-Clown and MOCHA-Ogre. We refer to the former as *human-dog* setting and the latter as *stylized* setting. In addition, we show that it is possible to learn a shared latent space for multiple datasets with different characters, such as Dog, Human-Locomotion, and MOCHA by extending Equation (4) with additional reconstruction losses and training multiple VQ-PAEs together. Please refer to 3:10 in the accompanying video for a demonstration.

5.2 Motion alignment

We examine the average pose at each point of the manifold to verify its alignment effect. Since our 1D phase manifold is a compact embedding of motions, the mapping from p_i to pose space is naturally a one-to-many mapping. However, it is not trivial to obtain the average on a continuous space. We propose to train a small MLP for each dataset that minimizes the following loss:

$$\mathcal{L}_{\text{pose}} = \mathbb{E}_{(p_i, \mathbf{Y}_i) \sim \mathcal{D}_k} \|\mathbf{Y}_i - M_k(p_i)\|_2, \tag{10}$$

where M_k is the MLP for dataset $\mathcal{D}_k$ that maps a point in $\mathcal{P}$ to pose space, and $(p_i, \mathbf{Y}_i)$ are pairs of manifold embedding and the corresponding pose in the dataset $\mathcal{D}$.

Figure 6: Motion retrieval. We retrieve motions at different frequencies in the same connected component containing motions of a dog moving up and down. From left to right the frequency decreases, corresponding to fast jumping, jumping up and sitting back, and slowly standing up and sitting back. Please refer to 1:17 in the accompanying video for a more comprehensive result.

We uniformly sample phase variables with different amplitudes to get the embeddings and use the learned MLP to predict the corresponding poses. The results are shown in Figure 1. It can be seen that even for drastically different characters like a dog and a human, where neither the semantic nor the timing alignment is well defined, the average poses from different datasets at the same manifold point provide a reasonable alignment on the semantic level. This is only possible if semantically similar motions are mapped into the same amplitude and poses with similar timing are mapped into the same phase, otherwise, the average poses would be noisy and meaningless. For more results, please refer to the accompanying video.

5.3 Disentangling phase and amplitude

In both phase manifolds designed by us and by DeepPhase [Starke et al. 2022], the phase represents timing and the amplitude represents motion content. We examine the phase-amplitude entanglement by training the same MLP mapping from the phase manifold to pose space as in Section 5.2. By taking the amplitude from one motion sequence or a static pose and the phase from another motion sequence, we predict the corresponding pose using the trained MLP. It can be seen in Figure 8 that our method can learn a disentangled phase manifold, but the manifold from DeepPhase fails due to the entanglement and non-compactness in using a multi-dimensional phase.

5.4 Motion retrieval

We show a simple example that by varying the frequency f, we can retrieve semantically similar motion at different frequencies by searching the nearest neighbor in the phase embeddings of the dataset, as shown in Figure 6. Formally speaking, given an amplitude $\mathbf{A} \in \mathcal{A}$ and a frequency f, we generate a uniformly distributed phase sequence $\Phi_f = \{\phi_i\}_{i=1}^N$ with $\phi_i = if\Delta_T$, and $N = 1/(f\Delta_T)$ such that Φ covers exact one cycle with frequency f. We then retrieve the desired motion with nearest neighbor search by comparing the constructed embedding sequence $\Psi(\mathbf{A}, \Phi_f)$ and the embedding sequences of the motions with length N from the dataset. Please refer to the accompanying video for a detailed result.

Table 1: Per-frame mean joint position error (cm) using MLP.

Size of $\mathcal{A}$	8	16	32	64	128	512
Dog [2018]	1.86	1.67	1.41	1.24	1.19	0.87
Human-Locomotion [2019]	1.29	1.26	1.19	1.13	1.08	1.01
MOCHA-Clown [2023]	11.6	10.1	9.97	9.86	9.29	6.50
MOCHA-Ogre [2023]	12.2	11.3	10.9	9.85	9.42	7.70

Table 2: Manifold overlapping percentage.

Size of $\mathcal{A}$	8	16	32	64	128	512
human-dog	100	100	100	67.8	5.42	0.00
stylized	100	100	100	100	92.3	10.3
human-dog (no reinit.)	100	73.6	64.2	52.5	1.32	0.00

5.5 Motion stylization and characterization

An immediate application of our improved motion matching can be motion stylization and characterization. We show that by training different VQ-PAEs on different characters from MOCHA [Jang et al. 2023] dataset in a shared phase manifold, we can transfer the core content of motion among different characters, and stylize the motion according to a specific character dataset as shown in Figure 9. We are able to achieve a similar effect as the motion stylization method proposed by Jang et al. [2023] with a much simpler setup. Since the code for MOCHA [Jang et al. 2023] is not available, we provide a qualitative comparison in the accompanying video.

5.6 Ablation study

We study the impact of codebook size and usage of reinitialization of $\mathcal{A}$ on the performance of our method.

Codebook size. Choosing an appropriate codebook size is critical for our framework, as a small codebook size will not be able to capture the different semantics, and a large codebook makes the alignment on semantics less accurate. We measure the expressiveness of a learned phase manifold by calculating the mean joint position error when using MLP to reconstruct the input motion from the phase manifold embeddings, using the same setting as in Section 5.2. Note that MOCHA datasets have a larger error due to a large number of transitions between amplitudes, which cannot be captured by the per-frame decoding MLP, but can be faithfully reconstructed by the motion matching algorithm using a sequence of embeddings as input. As shown in Table 1, the expressiveness reaches a plateau when the codebook size is larger than 64 for Dog and Human-Loco dataset, and 64 for MOCHA dataset, but peaks at 512. However, we also show that the percentage of embeddings in the dataset that lies on a shared connected component *decreases* with the codebook size, as shown in Table 2. This indicates that a large codebook size can cause a disparity in the learned manifold embeddings, in favor of higher expressiveness. Although size 512 improves on expressiveness, it fails to create sufficient overlapping between datasets. Thus, we choose $|\mathcal{A}| = 32$ for the human-dog setting and $|\mathcal{A}| = 64$ for the stylized setting in our experiments according to the results.

Reinitialization of $\mathcal{A}$. With the help of reinitialization adapted from Zheng and Vedaldi [2023], every entry in $\mathcal{A}$ is used by both VQ-PAEs, which is crucial for building a common phase manifold. When disabled, the phase manifold overlapping percentage drops as shown in Table 2.

6 DISCUSSION AND CONCLUSION

In this work, we present a disconnected 1D phase manifold for motion alignment, leveraging the intrinsic periodicity of motions. We show that the alignment can be achieved thanks to the carefully designed *structure* of the latent space. With the proposed vector quantized periodic autoencoder, we can embed motions from different characters with different skeletal structures or morphologies into the same phase manifold without any supervision or skeletal structure correspondences. We demonstrate that when integrated with motion matching, various applications such as motion retrieval, transfer, and stylization can be achieved.

The key success of our simple motion alignment lies in the limited capability of the shallow VQ-PAE, which prevents a large distortion between the motion representation and the latent embeddings, and the design of the compact latent space, a collection of ellipses embedded in $\mathbb{R}^d$. For semantic alignment, the structural similarity between motion datasets is explicitly reflected in the latent space through the amplitudes. For example, running motions are clustered into ellipses with larger amplitudes, while idling motions are clustered into ellipses with smaller amplitudes. As for timing alignment, the anisotropic structure of the ellipses (Equation (1)) is crucial. Although we expect the phase variable to progress linearly through an entire motion cycle, the progress of the phase manifold is not linear. This guarantees, for example, that crucial points in motions such as foot contacts, are mapped to the vertices of ellipses. However, this alignment is not always perfect: as can be seen at 3:01 in the accompanying video, a mismatch of the left and right foot contact exists, since no joint correspondence is provided, so the left and right body parts are indistinguishable.

While our current framework provides good timing alignment, the semantics alignment is not always perfect. It requires carefully picking the right codebook size to balance between expressiveness and the amount of overlap among datasets. It also implicitly requires the datasets to contain semantically similar motion distributions. For example, the backward motion is presented in the Human-Loco dataset but not in the Dog dataset, so the Human backward walking is aligned with forward walking for Dog. In the future, it would be interesting to automatically learn the size of $\mathcal{A}$ and filter out motions that are not semantically similar. In addition, the residual amplitude, removed by the quantization, could be potentially used for representing "styles" of motions within the same semantics.

Our current framework is not generative. It would be interesting to explore the possibility of generating new motions from the phase manifold. Another promising direction for future research is training the PAEs with other 1D input signals, such as a music dataset, e.g. for a tightly aligned music-to-dance generation.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. We also thank Heyuan Yao and Alexander Winkler for the insightful discussions. This work was supported in part by the ERC Consolidator Grant No. 101003104 (MYCLOTH).

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020a. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. 2020b. Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 64–1.

Okan Arikan and David A Forsyth. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 483–490.

Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Deep motifs and motion signatures. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–13.

Michael Büttner and Simon Clavet. 2015. Motion Matching - The Road to Next Gen Animation. https://www.youtube.com/watch?v=z_wpgHFSWss

Kwang-Jin Choi and Hyeong-Seok Ko. 2000. Online motion retargetting. *The Journal of Visualization and Computer Animation* 11, 5 (2000), 223–235.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341* (2020).

Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. 2023. Human Pose as Compositional Tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 660–671.

Michael Gleicher. 1998. Retargetting motion to new characters. In *Proc. 25th annual conference on computer graphics and interactive techniques*. ACM, 33–42.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. 2022. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*. Springer, 580–597.

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

Deok-Kyeong Jang, Yuting Ye, Jungdam Won, and Sung-Hee Lee. 2023. MOCHA: Real-Time Motion Characterization via Context Matching. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Nam Hee Kim, Zhaoming Xie, and Michiel van de Panne. 2020. Learning to Correspond Dynamical Systems. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control (Proceedings of Machine Learning Research)*, Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger (Eds.), Vol. 120. PMLR, 105–117. https://proceedings.mlr.press/v120/kim20a.html

Sunwoo Kim, Maks Sorokin, Jehee Lee, and Sehoon Ha. 2022. Humanconquad: human motion control of quadrupedal robots using deep reinforcement learning. In *SIGGRAPH Asia 2022 Emerging Technologies*. 1–2.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion Graphs. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 473–482. https://doi.org/10.1145/566570.566605

Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proc. 26th annual conference on computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 39–48.

Yongjoon Lee, Kevin Wampler, Gilbert Bernstein, Jovan Popović, and Zoran Popović. 2010. Motion fields for interactive character locomotion. 1–8.

Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023. Ace: Adversarial correspondence embedding for cross morphology motion retargeting from human to nonhuman characters. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Jongin Lim, Hyung Jin Chang, and Jin Young Choi. 2019. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting.. In *BMVC*, Vol. 2. 7.

Ian Mason. 2022. Periodic Autoencoder - Explanation and Addendum. https://www.ianxmason.com/posts/PAE/

Jianyuan Min and Jinxiang Chai. 2012. Motion graphs++ a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 1–12.

Jong Pil Park, Kang Hoon Lee, and Jehee Lee. 2011. Finding syntactic structures from human motion data. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, 2183–2193.

Sang Il Park, Hyun Joon Shin, and Sung Yong Shin. 2002. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 105–111.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)* 37, 4 (2018), 1–14.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

Mingyi Shi, Sebastian Starke, Yuting Ye, Taku Komura, and Jungdam Won. 2023. Phasemp: Robust 3d pose estimation via phase-conditioned human motion prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14725–14737.

Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. 2022. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11050–11059.

Paul Starke, Sebastian Starke, Taku Komura, and Frank Steinicke. 2023. Motion In-Betweening with Phase Manifolds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6, 3 (Aug. 2023), 1–17. https://doi.org/10.1145/3606921

Sebastian Starke, Ian Mason, and Taku Komura. 2022. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–13.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Trans. Graph.* 38, 6 (2019), 209–1.

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 54–1.

Seyoon Tak and Hyeong-Seok Ko. 2005. A physically-based motion retargeting filter. *ACM Trans. Graph.* 24, 1 (2005), 98–117.

Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. 1995. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 91–96.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *Proc. IEEE CVPR*. 8639–8648.

Shihong Xia, Congyi Wang, Jinxiang Chai, and Jessica Hodgins. 2015. Realtime style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2m-gpt: Generating human motion from textual descriptions with discrete representations. *arXiv preprint arXiv:2301.06052* (2023).

Chuanxia Zheng and Andrea Vedaldi. 2023. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22798–22807.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

source motion

transfered motion

SAN [2020a]

Figure 7: Motion transfer. Our framework can transfer motions between different characters preserving the semantics. However, SAN [2020a] produces implausible results because of unstable adversarial training.

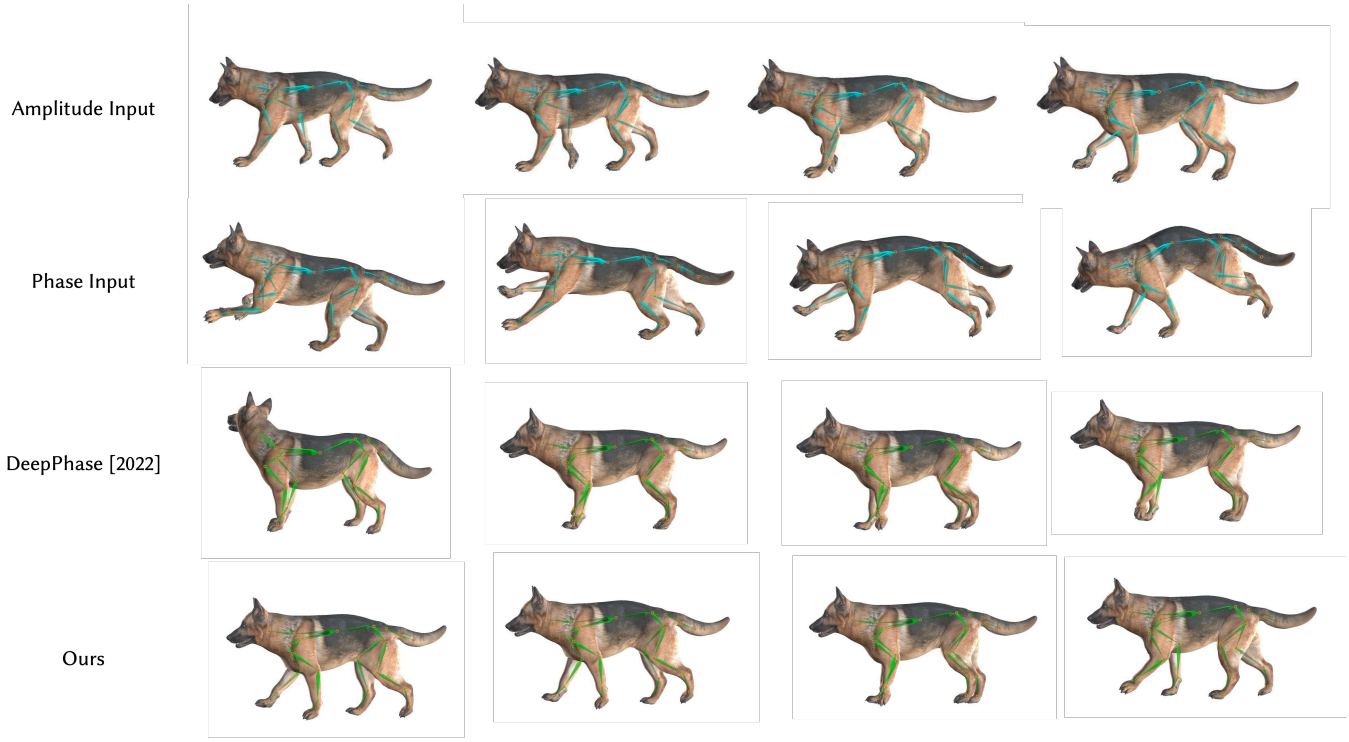

Figure 8: Phase and amplitude disentanglement. Our method generates motion combining the semantics from the amplitude input and the timing from the phase input, while DeepPhase [2022] generates implausible motions due to the entangled phase manifold.

source motion (ogre)

transfered motion (clown)

Figure 9: Motion characterization. The walking motion of the ogre is transferred to the clown. Our method preserves the semantics of the motion, while the result motion is highly characterized.

Supplementary material: WalkTheDog: Cross-Morphology Motion Alignment via Phase Manifolds

Peizhuo Li
ETH Zurich
Switzerland
peizhuo.li@inf.ethz.ch

Sebastian Starke
Meta Reality Labs
United Kindom
sstarke@meta.com

Yuting Ye
Meta Reality Labs
USA
yuting.ye@meta.com

Olga Sorkine-Hornung
ETH Zurich
Switzerland
sorkine@inf.ethz.ch

1 IMPLEMENTATION DETAILS

In this section, we give a detailed description of the phase calculation module and the motion matching (without frequency scaling) algorithm.

1.1 Phase Calculation

We use the equations presented by Mason [2022] to calculate the phase ϕ using the entries y_i from 1-channel signal $\mathbf{Y}$ and the relative timing t_i from $\mathcal{T}$:

$$\begin{aligned} s_x &= \sum_{i=1}^{T} y_i \cos(2\pi f t_i), \\ s_y &= \sum_{i=1}^{T} y_i \sin(2\pi f t_i), \\ \phi &= \frac{\operatorname{atan2}(s_y, s_x)}{2\pi}, \end{aligned} \tag{1}$$

where $\operatorname{atan2}(y, x)$ is the argument of the complex number $x + iy$. This equation helps us avoid dealing with the fact that ϕ is not a continuous parameterization of the phase manifold.

1.2 Motion Matching on Phase Manifolds

Given the phase embedding sequence $\mathbf{P}$ of an input motion sequence with T frames, we use Algorithm 1 to retrieve the motion sequence from the database, using phase embedding as the control signal in the classical motion matching algorithm. We denote the length of replay after each match with T_0, the phase sequences of the database with $\mathbf{Q}$, the pose descriptor used to measure the similarity between frames with $\mathbf{J}$ and the pose with $\mathbf{Y}$.

In our experiment, we use a simple setup where normalized joint positions are chosen as our pose descriptor $\mathbf{J}$ to ensure a smooth transition between different replays. We also apply inertialization at each transition. The cost function in Algorithm 1 can be customized depending on the exact application.

Algorithm 1 Phase Manifold Motion Matching

$i \leftarrow 1$
$\mathbf{J}_{\text{start}} \leftarrow$ initial pose descriptor
while $i < T$ **do**
  $k = \arg\min_k \|\mathbf{P}_{i:i+T_0} - \mathbf{Q}_{k:k+T_0}\|_2^2 + \lambda \|\mathbf{J}_{\text{start}} - \mathbf{J}_k\|_2^2$
  Output $\mathbf{Y}_{k:k+T_0}$
  $i \leftarrow i + T_0$
  $\mathbf{J}_{\text{start}} \leftarrow \mathbf{J}_{k+T_0}$
end while

Table 1: Details of the datasets used in our experiments.

Name	Framerate	# of Frames
Dog [2018]	60	151k
Human-Locomotion [2019]	60	186k
MOCHA-Clown [2023]	120	486k
MOCHA-Ogre [2023]	120	500k
MOCHA-Princess [2023]	120	501k

Since skeleton-aware networks [Aberman et al. 2020] require homeomorphic skeletons, we remove the tail of the dog skeleton when compared with it. In addition, we specify a correspondence between the forelegs and arms, and the hindlegs and legs. SAN heavily requires end-effector velocity consistency between the source and target characters and struggles to transfer motion with a large difference in skeletal structure.

1.3 Datasets

A detailed description of datasets is listed in Table 1. The training time on Dog and Human-Locomotion is around 40 minutes while on MOCHA-Clown and MOCHA-Ogre is around 2 hours, proportional to the number of frames. Wehen training on all the datasets together, it takes around 2 hours and 40 minutes.

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07.
https://doi.org/10.1145/3641519.3657508

2 NETWORK ARCHITECTURE AND HYPERPARAMETERS

In this section, we describe the network architecture and the hyperparameters used to train the network.

2.1 Architecture Details

The convolutional encoder and decoder share the same architecture of two-layer 1D convolutions with kernel size 23 with ELU as activation. The MLP producing raw amplitude $\tilde{\mathbf{A}}$ has 5 layers and the hidden units are of the same size as the amplitude. The MLP in the phase calculation module also has 5 layers. The hidden units share the size as the input frequency bin powers produced by FFT. The MLP used in learning average poses has 8 layers, and the hidden units are of the same size as the pose. LeakyReLU with a negative slope 0.2 is used as activation for all aforementioned MLPs.

2.2 Hyperparameters

Our VQ-PAE is implemented in PyTorch [Paszke et al. 2019], and the experiments are performed on NVIDIA GeForce RTX 3090 GPU. We optimize the parameters of our network using the Adam optimizer [Kingma and Ba 2014]. We set the learning rate to 1×10^{-4} and the batch size to 32. We choose to use joint velocity in the local coordinate of the character as the input feature $\mathbf{X}$. We choose the size of the input motion sequence during training to correspond to 1 second. The exact size is different for different datasets depending on their framerate. For the hyperparameters used in frequency-scaled motion matching, we set $\lambda_1 = 0.5$ and $\lambda_2 = 1$.

REFERENCES

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

Deok-Kyeong Jang, Yuting Ye, Jungdam Won, and Sung-Hee Lee. 2023. MOCHA: Real-Time Motion Characterization via Context Matching. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ian Mason. 2022. Periodic Autoencoder - Explanation and Addendum. https://www.ianxmason.com/posts/PAE/

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Trans. Graph.* 38, 6 (2019), 209–1.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.