

# MoDi: Unconditional Motion Synthesis from Diverse Data

SIGAL RAAB, Tel-Aviv University, Israel

INBAL LEIBOVITCH, Tel-Aviv University, Israel

PEIZHOU LI, ETH Zurich, Switzerland

KFIR ABERMAN, Google Research, USA

OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

DANIEL COHEN-OR, Tel-Aviv University, Israel

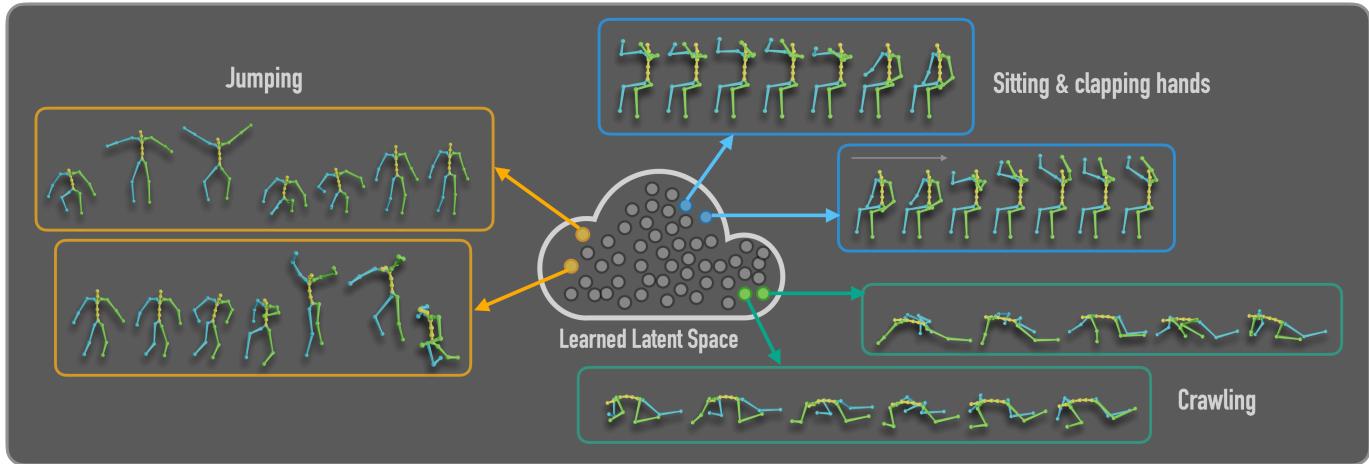


Fig. 1. Our generative prior is learned in a completely unsupervised setting from a diverse, unstructured and unlabeled motion dataset and yields a well-behaved, highly semantic latent space that facilitates synthesis operations. Note that each single latent code represents a whole motion sequence.

The emergence of neural networks has revolutionized the field of motion synthesis. Yet, learning to unconditionally synthesize motions from a given distribution remains a challenging task, especially when the motions are highly diverse. We present MoDi, an unconditional generative model that synthesizes diverse motions. Our model is trained in a completely unsupervised setting from a diverse, unstructured and unlabeled motion dataset and yields a well-behaved, highly semantic latent space. The design of our model follows the prolific architecture of StyleGAN and adapts two of its key technical components into the motion domain: a set of style-codes injected into each level of the generator hierarchy and a mapping function that learns and forms a disentangled latent space. We show that despite the lack of any structure in the dataset, the latent space can be semantically clustered, and facilitates semantic editing and motion interpolation. In addition, we propose a technique to invert unseen motions into the latent space, and demonstrate latent-based

Authors' addresses: Sigal Raab, Tel-Aviv University, Israel, sigal.raab@gmail.com; Inbal Leibovitch, Tel-Aviv University, Israel, inbal.leibovitch@gmail.com; Peizhuo Li, ETH Zurich, Switzerland, peizli@ethz.ch; Kfir Aberman, Google Research, USA, kfirberman@gmail.com; Olga Sorkine-Hornung, ETH Zurich, Switzerland, sorkine@inf.ethz.ch; Daniel Cohen-Or, Tel-Aviv University, Israel, cohenor@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

XXXX-XXXX/2022/6-ART \$15.00

<https://doi.org/XXXXXX.XXXXXXX>

motion editing operations that otherwise cannot be achieved by naive manipulation of explicit motion representations. Our qualitative and quantitative experiments show that our framework achieves state-of-the-art synthesis quality that can follow the distribution of highly diverse motion datasets. Code and trained models will be released in <https://sigal-raab.github.io/MoDi>.

CCS Concepts: • Computing methodologies → Computer vision; Animation.

Additional Key Words and Phrases: Motion synthesis, Computer Graphics, Character animation, Deep learning, Generative Prior

## ACM Reference Format:

Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. 2022. MoDi: Unconditional Motion Synthesis from Diverse Data. 1, 1 (June 2022), 11 pages. <https://doi.org/XXXXXX.XXXXXXX>

## 1 INTRODUCTION

The field of motion synthesis includes a wide range of long-standing problems whose goal is to hallucinate a sequence of temporally coherent poses that satisfy given cues and/or spatio-temporal constraints and importantly, look natural. In particular, learning to synthesize human motion from a given data distribution is a challenging task, especially when the dataset is highly diverse, unstructured and unlabeled. In recent years, deep neural networks have become a popular tool for motion generation, and their excellent performance is imputed to their ability to learn motion priors from large scale datasets.

However, learning a motion prior from a diverse dataset that contains a wide variety of motions remains a challenge.

Previous works focused on synthesizing specific types of motion of limited diversity [Holden et al. 2017b], building upon recurrent networks, conditioned by a set of initial frames [Zhou et al. 2018] or learn from a structured dataset and condition the synthesis by a label indicating a specific action [Petrovich et al. 2021].

In this work, we present MoDi, an unconditional generative model that synthesizes diverse motions. Our model is trained in a completely unsupervised setting from a diverse, unstructured and unlabeled motion dataset and yields a well-behaved, highly semantic latent space, which facilitates synthesis operations.

Our design is inspired by the powerful architecture of StyleGAN [Karras et al. 2019], which has been shown to learn a disentangled latent space and allows incredible editing capabilities [Patashnik et al. 2021; Shen et al. 2020; Wu et al. 2021]. Unlike StyleGAN, which consists of a hierarchy of convolutional layers applied on a spatial domain, here we apply the convolutions on time-vertex layers, using skeleton-aware operators [Aberman et al. 2020a], and the network generates a temporal sequence of poses rather than a single image.

In our design, there are two key components borrowed from StyleGAN: a style-code injected in each level of the generator hierarchy and a mapping function that forms the disentangled latent space. Further, we present an inversion technique that projects a given motion onto the learned latent space of our generator, enabling editing and latent-based operations to be applied on real motions.

Our results show that our generative model learns a powerful latent space without any supervision, which facilitates semantic manipulation problems on diverse motions, including semantic edition, or semantic interpolation between motion. We demonstrate that the latent space is structured and can be clustered into semantic regions, and present an inversion technique where unseen motion is projected into the latent space allowing its edit. We also qualitatively and quantitatively evaluate our synthesis quality and show that it achieves higher quality than alternative methods in similar settings.

## 2 RELATED WORK

The emergence of neural networks has transformed the field of motion synthesis, and many novel neural models have been developed in recent years [Holden et al. 2016, 2015]. Most of these models focus on specific human motion related tasks, conditioned on some limiting factors, such as motion prefix [Aksan et al. 2019; Barsoum et al. 2018; Habibie et al. 2017; Hernandez et al. 2019; Yuan and Kitani 2020; Zhang et al. 2021a], in-betweening [Duan et al. 2021; Harvey and Pal 2018; Harvey et al. 2020; Kaufmann et al. 2020], motion retargeting or style transfer [Aberman et al. 2020a,b, 2019; Holden et al. 2017a; Villegas et al. 2018], music [Aristidou et al. 2021; Lee et al. 2018; Li et al. 2021; Sun et al. 2020], or text [Tevet et al. 2022; Zhang et al. 2021b].

A large number of models focus on action conditioned generation. Since these works are closer in spirit with ours, in the following we elaborate more about them. These models can be roughly divided to autoregressive [Fragkiadaki et al. 2015; Guo et al. 2020; Habibie et al. 2017; Maheshwari et al. 2022; Petrovich et al. 2021; Zhou et al.

2018] and generative [Degardin et al. 2022; Wang et al. 2020b; Yan et al. 2019; Yu et al. 2020].

Petrovich et al. [2021] learn an action-aware latent representation by training a VAE. They sample from the learned latent space and query a series of positional encodings to synthesize motion sequences conditioned on a categorical action. They employ a transformer for encoding and decoding a sequence of parametric SMPL human body models. Maheshwari et al. [2022] generate single or multi-person pose-based action sequences with locomotion. They present generations conditioned by 120 action categories. They use a Conditional Gaussian Mixture Variational Autoencoder to enable intra and inter-category diversity. Wang et al. [2020a] employ a sequence of recurrent autoencoders. They replace the KL divergence loss by a discriminator to ensure the bottle neck distribution.

Some generative models are combined with factors that limit their generalization, such as Gaussian processes [Yan et al. 2019] or auto encoders [Wang et al. 2020b; Yu et al. 2020]. Degardin et al. [2022] fuse the architectures of GANs and GCNs to synthesise the kinetics of the human body. Like us, they borrow a mapping network from StyleGAN [Karras et al. 2019, 2020]. However, their model does not utilize important aspects of StyleGAN such as multi-level style injection. As we demonstrate, these aspects significantly ameliorate the quality of the synthesized motions.

Unlike the above conditional models, we present an unconditional method. Only a few works enable pure unconditioned synthesis. Holden et al. [2016] presented a pioneering work in deep motion. They introduce an unconditional autoencoder, which unlike our model, is not skeleton-aware, and trained on a rather limited set of activities in which the character is standing on its feet. Their latent space is not sufficiently disentangled, so they train a separate autoencoder for each editing task, while MoDi performs editing in the latent space with no need to train an encoder whatsoever.

In order to process motion in a deep learning framework, many existing works convert the motion into a pseudo image, where the joints and time-frames are equivalent to image height and width, and joint features (e.g., coordinates) are equivalent to RGB channels [Hernandez et al. 2019; Holden et al. 2016; Maheshwari et al. 2022; Petrovich et al. 2021]. While this approach is straightforward and intuitive, joints are fundamentally different from image pixels in that they are not necessarily adjacent to each other as pixels are. A partial solution for this problem is presented in Tree Structure Skeleton Image (TSSI) [Yang et al. 2018], where some of the joints are replicated to ensure skeletal continuity in convolution. However, TSSI cannot reflect all neighborhood degrees.

The emergence of Graph-based convolutional networks has been adopted by the motion research community [Yan et al. 2019], since the human skeleton is naturally represented by a graph, where the joints and bones are represented with vertices and edges, respectively. A full motion is then considered as a spatio-temporal graph [Degardin et al. 2022; Yu et al. 2020].

Since a single kernel shared by all joints cannot capture the fine nuances of each joint, more advanced techniques [Aberman et al. 2020a; Yan et al. 2019] exploit the advantage of using finite size skeletons with predefined topology. Each skeletal joint is unique in the way it relates to its neighbors. In our work, we adopt this approach and dedicate a unique kernel for each joint.

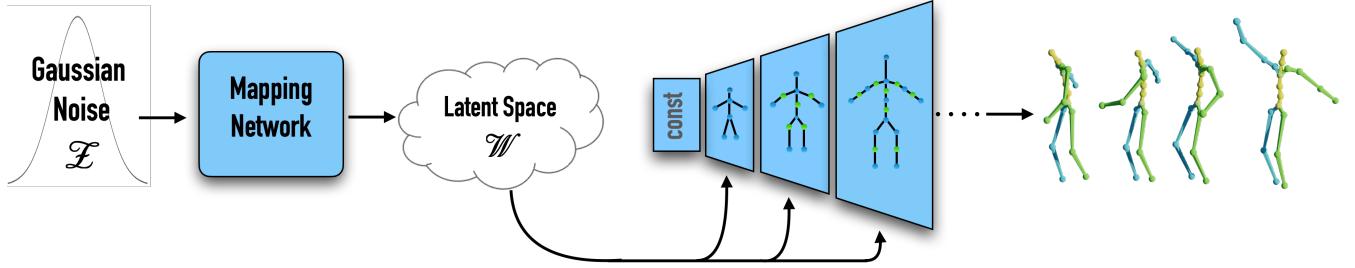


Fig. 2. MoDi’s motion generator combines structure-aware neural modules with a mapping network and style-codes injected to multiple levels of the generator.

### 3 OVERVIEW

At the crux of our approach lies a deep generative model that is trained in an unsupervised fashion on an extremely diverse, unstructured, motion dataset. Our framework is inspired by recent works in the imaging domain that present generative models which learn a well behaved latent space [Karras et al. 2019, 2020]. However, there is a significant domain gap between images and motion data. Images possess a regularized spatial structure, while motion data is an irregular one, which consists of a skeletal graph axis, and a temporal one. As we shall elaborate next, our architectural design bridges this gap. We represent the motion data by a temporal sequence of 3D joint rotations, a sequence of global positions of the root joint, and foot contact labels. Furthermore, we employ 3D convolutions rather than 2D ones, embed structure-aware neural modules, and use a different kernel for each skeletal joint. Note however, that an image-based progressive generative network is a lot deeper, with an order of magnitude more parameters, compared to our skeletal network whose depth depends on the topological complexity of the skeleton.

As illustrated in Figure 2, our network receives a noise vector drawn from an i.i.d Gaussian distribution and outputs a natural, temporally coherent, human motion sequence. In the first part, the framework contains a mapping network that learns to map the Gaussian distribution into a latent distribution that corresponds to the distribution of the training data. Note that each point in the latent space represents a sequences of poses and not a single frame.

In the second part, we introduce a hierarchical framework that learns to convert a learnable constant tensor into a motion representation via a series of skeleton-aware convolutional and unpooling layers [Aberman et al. 2020a; Degardin et al. 2022]. The intermediate layers in the network are modulated by style-codes that are injected in each level and modify second order statistics of the channels, in a spatially invariant manner [Huang and Belongie 2017]. Due to the disentangled nature of the trained latent space, our generative model learns a powerful motion representation, which facilitates semantic manipulation problems on diverse motions.

Next, we discuss our motion representation (Section 4) and the architecture of our framework (Section 5), show that our latent space is semantically clustered, present an inversion technique that projects a given motion onto the learned latent space, and demonstrate semantic editing capabilities applications (Section 6). Finally, we show

quantitative and qualitative evaluation of our framework and compare it to state-of-the-art alternatives (Section 7). We refer the reader to the supplementary video to see the results of our work.

## 4 MODEL

### 4.1 Motion representation

We describe motion using temporally coherent 3D joint rotations,  $\mathbf{R} \in \mathbb{R}^{T \times J \times K}$ , where  $T$  is the number of frames,  $J$  is the number of joints and  $K$  is the number of rotation features. The root joint position is represented by a sequence of global translations,  $\mathbf{P} \in \mathbb{R}^{T \times 3}$ .

The human eye is sensitive to the slightest appearance of foot sliding. Thus, we encourage our network to refrain from synthesising sliding artefacts by appending foot contact labels to our motion representation. We hold binary values for the two feet,  $\mathbf{F} \in \{0, 1\}^{T \times 2}$ , indicating whether each foot touches the ground at each frame.

We zero-pad the feature dimension of the root location and the feet contact labels to the size of the rotation feature,  $K$ , and add an extra dimension so  $\mathbf{P}$  and  $\mathbf{F}$  can be considered as three additional entities. Altogether we have  $\mathbf{R} \in \mathbb{R}^{T \times J \times K}$  (unchanged),  $\hat{\mathbf{P}} \in \mathbb{R}^{T \times 1 \times K}$ , and  $\hat{\mathbf{F}} \in \mathbb{R}^{T \times 2 \times K}$ . Once all features share the same size, we concatenate them and denote the full motion space by

$$\mathcal{M}_{full} = \mathbb{R}^{T \times (J+3) \times K}. \quad (1)$$

Let  $\mathcal{M}_{nat}$  denote the space of natural motions that are plausible for humans to enact. Each motion  $m \in \mathcal{M}_{nat}$  is represented by a tuple,  $[\mathbf{R}_m, \hat{\mathbf{P}}_m, \hat{\mathbf{F}}_m]$ . Note that the sub-space of all human motions,  $\mathcal{M}_{nat} \subset \mathcal{M}_{full}$ , is extremely sparse, as most of the values in  $\mathcal{M}_{full}$  correspond to unnatural or impossible human motions.

In order to operate in a deep environment, we match a level subscript to each of the aforementioned elements. Hence the number of frames, joints and features at level  $\ell$  is denoted by  $T_\ell$ ,  $J_\ell$  and  $K_\ell$ , respectively. Accordingly, the representation of rotations, root positions, and foot contact labels, is changed to  $\mathbf{R}_\ell \in \mathbb{R}^{T_\ell \times J_\ell \times K_\ell}$ ,  $\mathbf{P}_\ell \in \mathbb{R}^{T_\ell \times 3}$  and  $\mathbf{F}_\ell \in \{0, 1\}^{T_\ell \times 2}$ , respectively. While the number of frames  $T_\ell$  changes at each level in the conventional neural way, that is, by multiplies of two, the number of joints is changed to obtain a meaningful reduction/increase in the skeletal topology [Aberman et al. 2020a; Degardin et al. 2022; Yan et al. 2019; Yu et al. 2020]. Additional representation considerations are detailed in the appendix.

### 4.2 Structure-aware Neural Modules

The human skeleton can be considered as a directed graph, where the joints stand for vertices and the bones stand for directed edges.

We refer to each skeletal joint as associated with the edge that is directed towards it, thus they share the same features. The root joint, to which no edge is directed, is associated with an abstract edge that starts at the origin and ends at the root joint. In the following parts we refer to joints and edges seamlessly, according to the context.

Some works [Degardin et al. 2022; Yu et al. 2020] use Graph Convolutional Networks (GCNs) for neural computation. Like GCNs, they employ the same kernel to all graph vertices. Unlike general graphs, the topology of the skeleton is known in advance, and it has a finite size. These facts can be exploited to get better sensitivity to each joint’s unique role in the skeleton. We follow the works that exploit the knowledge of skeletal topology [Aberman et al. 2020a; Yan et al. 2019] and dedicate a separate kernel for each joint.

## 5 NETWORK ARCHITECTURE

In recent years, generative works in the image domain have attained unprecedented synthesis quality [Brock et al. 2018; Ho et al. 2020; Kingma and Dhariwal 2018]. One of the prominent methods is StyleGAN [Karras et al. 2021, 2019, 2020]. StyleGAN as is *cannot* be used for motion synthesis: (1) it is designated for 2D images, where the inductive bias of pixel neighborhood is strongly exploited; this bias cannot be used for skeletal joints that are not geometrically continuous, and (2) it uses standard 2D convolutional kernels, whereas MoDi achieves significantly better results with skeleton-aware kernels (see Section 7). We adapt StyleGAN’s mapping network, and provide our own synthesis network, which we call *motion synthesis network*. Our motion synthesis network is based on skeleton-aware neural modules, combined with StyleGAN’s style-code injection. The combination of the mapping network and the synthesis network yields our generator,  $G$ . We denote motions synthesized out of a random noise  $z \in \mathcal{Z}$  by  $G(z)$ , and motions synthesized out of a latent variable  $w \in \mathcal{W}$  by  $G(w)$ , where  $w$  can be generated either by running  $z$  in the mapping network or by inverting a given motion (see Section 6.3). In order to synthesise motions directly from  $\mathcal{W}$ , the generator skips the mapping block and goes directly to the synthesis network. Our generator architecture is described below and is illustrated in Figure 2.

*Mapping network.* Let  $\mathcal{Z} = \mathcal{N}(\vec{0}, \mathbf{I})$  be a multivariate normal distribution. Given a latent code  $z \in \mathcal{Z}$ , a non-linear mapping network produces a latent value,  $w \in \mathcal{W}$ .  $\mathcal{W}$  is known to be disentangled and well behaved, as studied by many works [Nitzan et al. 2021; Shen et al. 2020] for images and by Degardin et al. [2022] for motions.  $w$  is then converted into a set of *style* standard deviation vectors, using learned affine transformations.

*Motion synthesis network.* Our motion synthesis network is based on structure-aware neural modules, described in Section 4.2. We briefly describe style-code injection here, for information completeness. The style is a standard deviation to be applied on the data in the motion synthesis network. It is produced out of  $w$ , using learned affine transformations. The output of each affine transformation is used for the demodulation of a different layer. The effect of style-injection is a control of the motion synthesis process via scale-specific modifications. One unique characteristic of style injection is that the

style is not taken from a given reference, but is sampled out of  $z$  followed by the mapping network. Hence, the generated motion stems from a learned constant rather than a reference motion.

We employ a discriminator [Goodfellow et al. 2014],  $D$ , that holds the reverse architecture of the synthesis network. That is, it receives a generated or real motion, and processes it in neural blocks that gradually decrease in size. Like the synthesis network, our discriminator is based on structure-aware neural modules. A detailed description of training setups and hyperparameters is given in the appendix.

### 5.1 Loss Functions

Our main loss is adversarial. In addition, we regularize the generator with foot contact and with path length regularization, and the discriminator with  $R1$  regularization.

*Adversarial Loss.* We train our generator and discriminator with a non-saturating adversarial loss [Goodfellow et al. 2014],

$$\mathcal{L}_{adv}^G = -\mathbb{E}_{z \sim \mathcal{Z}} [\log D(G(z))], \quad (2)$$

$$\mathcal{L}_{adv}^D = -\mathbb{E}_{m \sim \mathcal{M}_{nat}} [\log D(m)] - \mathbb{E}_{z \sim \mathcal{Z}} [\log(1 - D(G(z)))] . \quad (3)$$

*Foot contact regularization.* Accurate foot contact is one of the major factors of motion quality. There is already a special care for foot contact in the adversarial loss, as  $\mathcal{M}_{nat}$  contains a foot contact label component. However, we noticed that encouraging the contact between the feet and the ground improves the naturalness of the the motions, and discourages the phenomenon of ‘floating’ feet. Hence we add an encouragement regulation

$$\mathcal{L}_{touch}^G = \mathbb{E}_{z \sim \mathcal{Z}} [-\log s(G(z)_F)], \quad (4)$$

where  $(\cdot)_F$  is the contact-label component of the motion, and  $s(\cdot)$  is the sigmoid function.

We use an additional regularization for the foot contact, which is the contact consistency loss [Li et al. 2022; Shi et al. 2020]. This regularization encourages a consistency between contact label and feet velocity, by requiring that in every frame either the contact label or the foot velocity will be minimized. That is, a high velocity should not be possible while a foot is touching the ground

$$\mathcal{L}_{fcon}^G = \mathbb{E}_{z \sim \mathcal{Z}} \left[ \left\| FK(G(z))_f \right\|_2^2 \cdot s(G(z)_F) \right], \quad (5)$$

where  $(\cdot)_F$  is the contact-label component of the motion,  $FK(\cdot)$  is the forward kinematic operator on a set of rotations and global position yielding joint locations,  $(\cdot)_f$  is the feet velocity extracted from joint locations, and  $s(\cdot)$  is the sigmoid function.

Although our foot contact loss significantly mitigates foot sliding artifacts, we further clean foot contact with a fully automatic procedure using standard IK optimization [Li et al. 2022].

*Path Length Regularization.* Path length regularization [Karras et al. 2020] encourages that a fixed-size step in  $\mathcal{W}$  results in a non-zero, fixed-magnitude change in the generated motion.

$$\mathcal{L}_{path}^G = \mathbb{E}_{w \sim \mathcal{W}, r \sim \mathcal{R}} \left[ \left\| J_w^T G(w) * r \right\|_2 - a \right]^2, \quad (6)$$

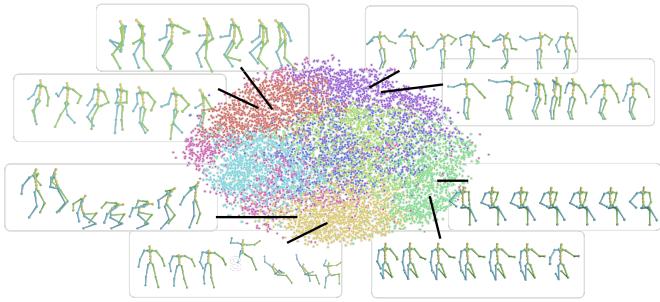


Fig. 3. The latent space  $\mathcal{W}$ , split into 8 clusters using K-means, and visualized via the T-SNE method. Each point relates to one  $\mathcal{W}$  space instance, generated from random noise  $z \in \mathcal{Z}$ . The visualized motions are the result of running those latent variables through our motion generator. We observe that the clusters indeed represent semantic grouping of the data.

where  $\mathcal{R}$  is a unit Gaussian space normalized by the number of joints and frames,  $J_w = \partial G(w)/\partial(w)$ , and  $a$  is the accumulated mean gradient length.

*R1 Regularization.* R1 regularization [Mescheder et al. 2018] is designed to improve the functioning of the discriminator.

$$\mathcal{L}_{R1}^D = \mathbb{E}_{m \sim \mathcal{M}_{nat}} [\|\nabla_m D(m)\|_2^2]. \quad (7)$$

Altogether, the loss for the generator is

$$\mathcal{L}^G = \mathcal{L}_{adv}^G + \lambda_{touch} \mathcal{L}_{touch}^G + \lambda_{fcon} \mathcal{L}_{fcon}^G, \quad (8)$$

and the loss for the discriminator is

$$\mathcal{L}^D = \mathcal{L}_{adv}^D. \quad (9)$$

We activate the regularization  $\mathcal{L}_{path}^G$  and  $\mathcal{L}_{R1}^D$  in a lazy fashion, as done by Karras et al. [2020].

## 6 LATENT SPACE ANALYSIS AND APPLICATIONS

In the following, we analyze the latent space  $\mathcal{W}$ , and explore its favorable characteristics.

### 6.1 Latent Clusters

We demonstrate that  $\mathcal{W}$  can be clustered into meaningful collections of motions. Recall that MoDi is designed to learn highly diverse datasets, whose data is unstructured and cannot be semantically clustered using common attributes.

In Figure 3 we observe the latent space  $\mathcal{W}$ , split into 8 clusters using K-means. The  $\mathcal{W}$  values belong to 10,000 randomly synthesized motions. We randomly choose several motions from each cluster and depict them. As one can see, motions represented by different clusters are semantically different, and motions that belong to the same cluster, are semantically similar.

### 6.2 Latent interpolation

Let  $\bar{w}$  be the mean value of all  $w \in \mathcal{W}$ , and let *mean motion* denote  $G(\bar{w})$ , the motion generated by it. The mean motion is depicted at the bottom row of Figure 4(a). This motion is similar for all variations

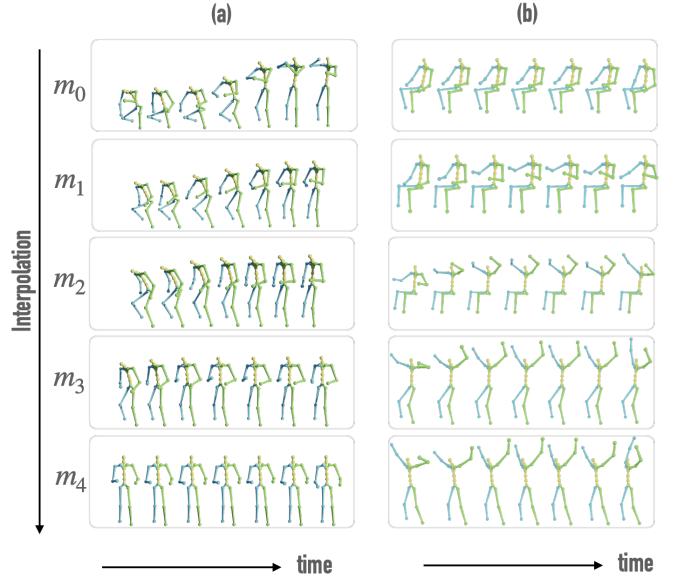


Fig. 4. Latent space interpolation with  $C = 4$ . Column (a): interpolation to the mean motion (truncation); column (b): interpolation between two sampled motions. The interpolated motion is notably smooth and natural.

of trained networks, and is what we would intuitively expect: an idle standing, front facing, human. We have noticed that interpolation towards the mean motion never seems to cause artifacts.

We demonstrate the linearity of the latent space  $\mathcal{W}$  by interpolating between the latent values and observing the motions generated out of the interpolated values. A special case, called *truncation*, is when the interpolation target is  $\hat{w}$ . Let  $\hat{w}$  be any latent value created by sampling a value  $z$  and running it in the mapping network. A truncated sequence is denoted by  $w_i = \hat{w} + \frac{i}{C}(\bar{w} - \hat{w})$ , where  $C$  is the number of interpolation steps, and  $i \in [0 \dots C]$ . Clearly  $w_0 = \hat{w}$  and  $w_C = \bar{w}$ . If we replace  $\bar{w}$  by another sampled value,  $\tilde{w}$ , we obtain an interpolated sequence,  $w_i = \hat{w} + \frac{i}{C}(\tilde{w} - \hat{w})$ . Let  $m_i = G(w_i)$  denote the motion generated out of each  $w_i$ . Figure 4 (a) and (b) shows the motions created out of truncation and interpolation, respectively.

We observe interesting characteristics in all interpolation sequences. First,  $m_i$  is semantically similar to  $m_{i-1}$ , but it also changed towards the semantics of the target motion  $m_C$ . When dealing with truncation,  $m_i$  is always milder than  $m_{i-1}$ . For example, if there is a wild move, say, jump, in  $m_{i-1}$ , it will become a milder jump in motion  $m_i$ .

Second, we notice that the interpolation is semantic and agnostic to frame indices. That is, the interpolation is not between frames, but between whole sequences. For example, if in  $m_{i-1}$  the character jumps occasionally, then in  $m_i$  the character jumps in the same semantic fashion, but not necessarily during the same frames.

Lastly, there are no unnatural motions in the sequence, although if we had just geometrically interpolated the joints, unnatural motions would have been obtained. Figure 4(b) demonstrates this: It shows our model’s latent interpolation, which obtains natural motions at all stages. A naïve geometric interpolation of edge rotations would obtain an abnormal pose, between sitting to standing, with a vertical spine (see supplementary video).

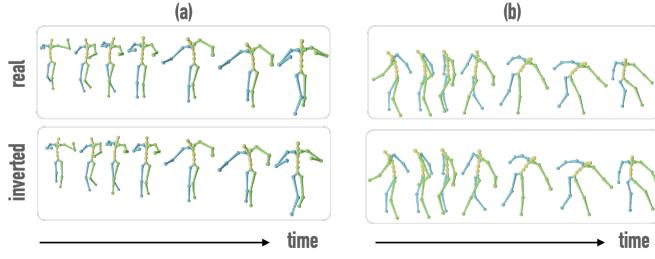


Fig. 5. Inversion of two motions, (a) and (b). The original motions are depicted at the top row, and the reconstructed ones are visualized at the bottom row. The reconstructed motions are created by first converting the real motions into  $\mathcal{W}^+$  and then running the obtained latent values in the generator.

### 6.3 Motion Inversion

We chose our inversion module to be an optimizer that minimizes the difference between reconstructed motion  $G(w)$  and the target motion  $m$ :

$$I(m) = \arg \min_w \|G(w) - m\|_2 + \lambda_{\text{FK}} \|FK(G(w)) - FK(m)\|_2 \quad (10)$$

Thus, the final inverted latent vector and its reconstruction are denoted by  $I(m)$  and  $G(I(m))$  respectively. We use gradient descent to minimize the target similarity metric starting with a random Gaussian noise. We find that the inversion quality improves when using  $\mathcal{W}^+$  [Abdal et al. 2019] rather than  $\mathcal{W}$ , and when using an  $L2$  loss on both rotation angles and joint locations. Figure 5 visualizes the quality of the obtained inversions.

Inversion is an important basis for further applications. Once an unseen motion is inverted to the  $\mathcal{W}^+$  space, it can be, for example, edited in a latent manner, as shown in the next paragraph.

### 6.4 Editing in the latent space

If a latent space is sufficiently disentangled, it should be possible to find direction vectors that consistently correspond to individual factors of variation. Let  $a$  be an attribute related to motion.  $a$  can be any semantic attribute, such as motion speed, verticality measurement of parts in the body, or a motion style. Inspired by Shen et al. [2020], we compute a score that measures  $a$  in a motion. For example, when measuring the verticality of a motion, a character doing a handstand would get a score of  $-1$ , lying down would get a score of  $0$ , and standing would get a score of  $1$ . Using a given score, we train an SVM, yielding a hyperplane that would serve as a separation boundary. Denote the unit normal of the hyperplane by  $n$ . Then  $G(w + n)$  would possess increased score of attribute  $a$  comparing to the  $G(w)$ . In a sufficiently disentangled latent space, the only attribute that should change in such an edited motion is  $a$ , preserving the rest of the motion intact. Unlike image datasets, that hold labeling for various attributes (age, gender, pose,...), there is not much labeling in motion datasets. We created our own simple classifiers, and here we elaborate regarding one of them, measuring *gradual right arm lifting*. Gradual right arm lifting, denoted *gral*, means that the right arm is lifted as time advances. An example that would get a high score is a standing character that starts its motion with its arm vertical towards the ground, and gradually lifts it such that at the end of the motion it is vertical towards the sky. Computing a score for the *gral* attribute

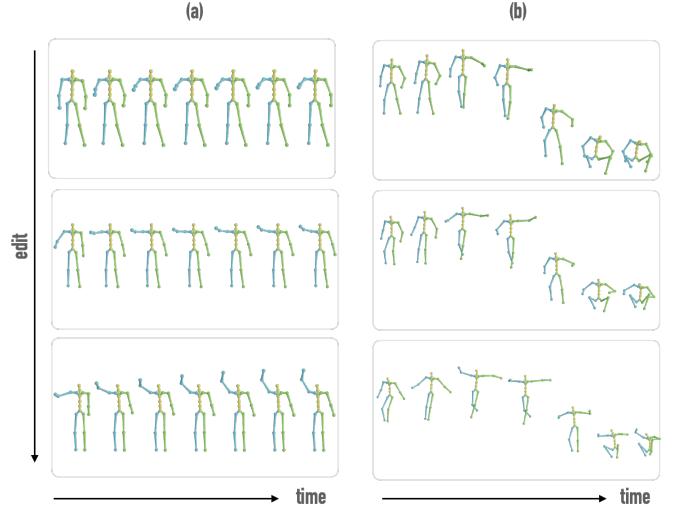


Fig. 6. Editing motions using latent space directions. The motion remains intact except for the edited attribute, *gradual right arm lifting* (*gral*). Note that character's right arm is to the reader's left. The *gral* attribute gets stronger as we go down in each column. The jumping motion (b) remains natural even at the expense of arm lifting, owing to our model's generative priors.

is not straight forward, and is detailed in the appendix. Results of editing the *gral* attribute are visualized in Figure 6, where we show that the *gral* attribute gets stronger while stepping in the latent space, but not on the expense of the naturalness of motions, and without changing their semantics. In our video clip we show that when such an attribute is artificially applied via geometric editing, the results are unnatural. Obtaining manual natural results would require an artist's hard work, while our generator does it effortlessly.

## 7 EXPERIMENTS

### 7.1 Datasets

*Mixamo*. We construct our 3D motion dataset using the Mixamo [Adobe Systems Inc. 2021] 3D animation collection, which contains approximately 2500 extremely diverse motions that are not constrained by any set of categories. These motions are applied on 70 characters. Examples of the motions in the dataset are elementary actions (jumping, kicking, walking), dance moves (samba, hip-hop), martial arts (boxing, capoeira), acrobatics (back/front flips, acrobatic jumps), and non-standard motions (running on a wall, flying).

We generate our data samples by first extracting the relevant edges from each motion (e.g., we drop the fingers). Then we crop each motion to partially overlapping sequences of frames, hence increasing the amount of data.

*HumanAct12*. HumanAct12 [Guo et al. 2020] is a less diverse dataset compared to Mixamo, and offers approximately 1200 motion clips, organized into 12 action categories and 34 subcategories. Due to its small number of motions, we use HumanAct12 for quantitative comparison only.

Table 1. Quantitative results for state-of-the-art ACTOR [Petrovich et al. 2021] and MoDi on the HumanAct12 [Guo et al. 2020] dataset. Best results are emphasized in **bold**. The grayed lines show the original algorithms, without the changes that make them comparable. Note that in all variations our model outperforms ACTOR.

Model	FID ↓	KID ↓	Precision ↑ Recall ↑	diversity ↑
ACTOR [2021] unconstrained	48.8	0.53	0.72, 0.74	14.1
ACTOR [2021] constrained	18.8	0.18	0.70, 0.56	17.52
MoDi (ours) with mixing	15.55	0.14	<b>0.72, 0.75</b>	17.36
MoDi (ours) without mixing	<b>13.03</b>	<b>0.12</b>	0.71, <b>0.81</b>	<b>17.57</b>

## 7.2 Quantitative Results

7.2.1 *Metrics.* We use the following metrics: FID, KID, precision-recall and diversity, detailed in the appendix. Some of the other metrics described in the literature, e.g., accuracy, cannot be used for an unconstrained model and are hence ignored. The metrics build upon the latent features of an action recognition network. However, training an action recognition network on Mixamo is a challenging task, as there is no action labeling in this dataset.

To address this challenge, we employ an interdisciplinary approach. Mixamo has a textual label and description for each motion. We employ an NLP model, namely sentence-BERT [Reimers and Gurevych 2019], to obtain latent features representing the textual characteristics of each motion. Then we use K-means to cluster the embedding, and use each class as a pseudo action label.

With action classes at hand, we train an STGCN [Yan et al. 2018] action recognition model. The features that are extracted from this trained model are then used for the metrics calculations. To calculate the metric scores, we randomly sample 2000 and 1000 synthesized motions on the Mixamo and HumanAct12 datasets, respectively.

7.2.2 *Results.* MoDi differs from most other works due to training without any constraints. Although Holden et al. [2016] introduced unconstrained training, we prefer comparing with more recent works, and since they are all conditioned, we make minor changes in the training procedure to make the models comparable, as detailed next.

Changes to constrained models: We ‘convert’ these models to unconstrained by assigning the same label to all motion instances, as if they all belong to the same class.

Changes to MoDi: MoDi uses style-mixing [Karras et al. 2019], which encourages disentanglement between hierarchical layers. However, mixing may change the distribution of synthesised motions, yielding degradation in the metric score. Hence, for the sole purpose of comparison with other works, we train our model with no mixing.

We compare our model with state-of-the-art action synthesis network, ACTOR [Petrovich et al. 2021], on the HumanAct12 [Guo et al. 2020] dataset. The results are shown in Table 1. We compare to two trained versions of ACTOR. The first, is where we make ACTOR comparable to MoDi by removing the action constraints. The numeric

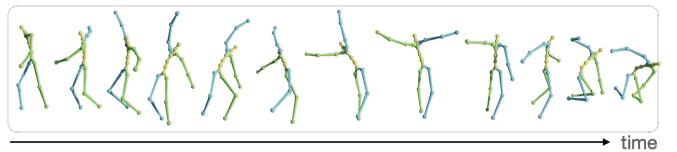


Fig. 7. Qualitative result. See more in the video and the supplemental.

results testify that training with no constraints makes it more difficult for ACTOR to converge. The second, used only as a sanity check, is where we run the constrained model provided by the authors. We observe that even the constrained version is outperformed by MoDi.

We provide results for two variations of our trained model. The first, is our regular setting, trained with style-mixing. The second is the one that should be compared to action synthesis models, with no mixing. Both versions outperform ACTOR by a large margin.

## 7.3 Qualitative Results

The reader is asked to watch our supplementary video in order to get the full impression of our networks qualitative results. For completeness, we show one special motion in Figure 7, and several more in the supplemental material.

## 7.4 Ablation

Table 2. Quantitative results for various generator designs, on the Mixamo dataset. The arrows next to the metric names indicate whether lower or higher values are better. Best scores are emphasised in **bold**.

Architecture	FID ↓	KID ↓	Precision ↑ Recall ↑	diversity ↑
Pseudo images Joint loc.	22.5	0.17	<b>0.49, 0.49</b>	13
+ Skeleton-aware without pooling	14.0	0.13	0.45, 0.62	14.4
+ Edge rot.	12.6	<b>0.09</b>	0.46, 0.60	15.2
+ Skeletal pooling	<b>11.24</b>	<b>0.09</b>	0.48, <b>0.70</b>	<b>15.3</b>

In this section we describe the various stages and improvements that have lead to the current result. Our ablation quantitative results are displayed in Table 2.

The first version of MoDi is inspired by image generative works, hence the motion is represented in a pseudo-image structure. The drawback of pseudo-images are detailed in Section 2. In the second variation, we replace the pseudo-images by a skeleton-aware joint structure. In the appendix we detail why generating edge rotations is better than generating joint locations, hence in the third variation we employ edge rotations. Up to the fourth version, we were using convolutional neural modules only. That is, the up-sampling of frames as well as joints was done using transposed convolutions, and the down sampling was done using convolutions. In the fourth version, we follow Aberman et al. [2020a] and control the joints up/down sampling with pooling and un-pooling neural modules.

Altogether, we see an improvement in each version, which shows that our direction of development is correct.

## 8 CONCLUSION

One of the most fascinating phenomena of deep learning is that it can learn knowledge, including semantics, from unsupervised data. In this work, we have presented a deep neural architecture that learns motion prior in a completely unsupervised setting. The main challenge has been to learn a generic prior from a diverse, unstructured and unlabeled motion dataset. This necessarily requires a careful design of a neural architecture to process the unlabeled data. We have presented MoDi, an architecture that distills a powerful, well-behaved latent space, which then facilitates downstream latent-based motion manipulations.

Like any data-driven method, the quality of the generalization power of MoDi is a direct function of the training data, which, at least compared to image datasets, is still lacking. Another limitation is that skeleton-aware kernels, with a dedicated kernel per joint, consume large volumes, resulting in relatively large running time.

A challenging research direction is to enable learning motion priors from videos. Building upon networks like MoDi, with inductive bias, may open the way to learn motion directly from videos.

## REFERENCES

- Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2stylegan: How to embed images into the stylegan latent space?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4432–4441.
- 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 transfer from video to animation. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 64–1.
- Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. 2019. Learning character-agnostic motion for motion retargeting in 2d. *arXiv preprint arXiv:1905.01680* (2019).
- Adobe Systems Inc. 2021. Mixamo. <https://www.mixamo.com> Accessed: 2021-12-25.
- Emre Aksan, Manuel Kauffmann, and Otmar Hilliges. 2019. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, 7144–7153.
- Andreas Aristidou, Anastasios Yiannakidis, Kfir Aberman, Daniel Cohen-Or, Ariel Shamir, and Yiorgos Chrysanthou. 2021. Rhythm is a Dancer: Music-Driven Motion Synthesis with Global Structure. *arXiv preprint arXiv:2111.12159* (2021).
- Emad Barsoum, John Kender, and Zicheng Liu. 2018. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 1418–1427.
- Sourav Biswas, Kangxue Yin, Maria Shugrina, Sanja Fidler, and Sameh Khamis. 2021. Hierarchical Neural Implicit Pose Network for Animation and Motion Retargeting. *arXiv preprint arXiv:2112.00958* (2021).
- Mikolaj Binkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=r1IUOzWCW>
- Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).
- Bruno Degardin, João Neves, Vasco Lopes, João Brito, Ehsan Yaghoubi, and Hugo Proença. 2022. Generative Adversarial Graph Convolutional Networks for Human Action Synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE Computer Society, Los Alamitos, CA, USA, 1150–1159.
- Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. 2021. Single-shot motion completion with transformer. *arXiv preprint arXiv:2103.00776* (2021).
- Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE Computer Society, 4346–4354.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM New York, NY, USA, 2021–2029.
- Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. 2017. A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference*.
- Félix G Harvey and Christopher Pal. 2018. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs*. 1–4.
- Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.
- Alejandro Hernandez, Jürgen Gall, and Francesc Moreno-Noguer. 2019. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, 7134–7143.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
- Daniel Holden, Ikhsanul Habibie, Ikuo Kusajima, and Taku Komura. 2017a. Fast neural style transfer for motion data. *IEEE computer graphics and applications* 37, 4 (2017), 42–49.
- Daniel Holden, Taku Komura, and Jun Saito. 2017b. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.
- Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.
- Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*. 1–4.
- Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*. 1501–1510.
- Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34 (2021).
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4401–4410.
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.
- Manuel Kauffmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. 2020. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 918–927.
- Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* 31 (2018).
- Juheon Lee, Seohyun Kim, and Kyogu Lee. 2018. Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network. *arXiv preprint arXiv:1811.00818* (2018).
- Peizhuo Li, Kfir Aberman, Zihan Zhang, Rana Hanocka, and Olga Sorkine-Hornung. 2022. GANimator: Neural Motion Synthesis from a Single Sequence. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 138.
- Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. 2021. Learn to dance with aist++: Music conditioned 3d dance generation. *arXiv e-prints* (2021), arXiv-2101.
- Shubh Maheshwari, Debutan Gupta, and Ravi Kiran Sarvadevabhatla. 2022. MUGL: Large Scale Multi Person Conditional Action Generation with Locomotion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. IEEE Computer Society, Los Alamitos, CA, USA, 257–265.
- Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which Training Methods for GANs do actually Converge?. In *International Conference on Machine learning (ICML)*.
- Yotam Nitzan, Rinon Gal, Ofir Brenner, and Daniel Cohen-Or. 2021. LARGE: Latent-Based Regression through GAN Semantics. *arXiv preprint arXiv:2107.11186* (2021).
- Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2085–2094.
- Mathis Petrovich, Michael J Black, and Güldem Varol. 2021. Action-conditioned 3D human motion synthesis with transformer VAE. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, 10985–10995.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.
- Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. 2018. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems* 31 (2018).

- Yujun Shen, Ceyuan Yang, Xiaou Tang, and Bolei Zhou. 2020. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *arXiv preprint arXiv:2005.09635* (2020).
- Mingyi Shi, Kfir Aberman, Andreas Aristidou, Taku Komura, Dani Lischinski, Daniel Cohen-Or, and Baoguan Chen. 2020. Motionet: 3d human motion reconstruction from monocular video with skeleton consistency. *ACM Transactions on Graphics (TOG)* 40, 1 (2020), 1–15.
- Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S Kankanhalli, Weidong Geng, and Xiangdong Li. 2020. DeepDance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia* 23 (2020), 497–509.
- Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. 2022. MotionCLIP: Exposing Human Motion Generation to CLIP Space. *arXiv preprint arXiv:2203.08063* (2022).
- Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8639–8648.
- Qi Wang, Thierry Artières, Mickael Chen, and Ludovic Denoyer. 2020a. Adversarial learning for modeling human motion. *The Visual Computer* 36, 1 (2020), 141–160.
- Zhenyi Wang, Ping Yu, Yang Zhao, Ruiyi Zhang, Yufan Zhou, Junsong Yuan, and Changyou Chen. 2020b. Learning diverse stochastic human-action generators by learning smooth latent transitions. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 12281–12288.
- Zongze Wu, Dani Lischinski, and Eli Shechtman. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12863–12872.
- Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. 2019. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, 4394–4402.
- Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*.
- Zhengyuan Yang, Yuncheng Li, Jianchao Yang, and Jiebo Luo. 2018. Action recognition with spatio-temporal visual attention on skeleton image sequences. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 8 (2018), 2405–2415.
- Ping Yu, Yang Zhao, Chunyu Li, Junsong Yuan, and Changyou Chen. 2020. Structure-aware human-action generation. In *European Conference on Computer Vision*. Springer, 18–34.
- Ye Yuan and Kris Kitani. 2020. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*. Springer, 346–364.
- Jia-Qi Zhang, Xiang Xu, Zhi-Meng Shen, Ze-Huan Huang, Yang Zhao, Yan-Pei Cao, Pengfei Wan, and Miao Wang. 2021b. Write-An-Animation: High-level Text-based Animation Editing with Character-Scene Interaction. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 217–228.
- Yan Zhang, Michael J Black, and Siyu Tang. 2021a. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3372–3382.
- Yi Zhou, Zimo Li, Shuangjin Xiao, Chong He, Zeng Huang, and Hao Li. 2018. Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis. In *International Conference on Learning Representations*.

Table 3. Architecture: Dimension at hierarchy levels.

Name	Hierarchy level	channels × joints × frames
Generator - Motion Synth. Net.	0	256 × 1 × 4
	1	128 × 2 × 8
	2	64 × 7 × 16
	3	64 × 12 × 32
	4	32 × 20 × 64
Discriminator	0	32 × 20 × 64
	1	64 × 12 × 32
	2	64 × 7 × 16
	3	128 × 2 × 8
	4	256 × 1 × 4

Table 4. Architecture: Building blocks in hierarchical levels. Skeletal operators are based on [Aberman et al. 2020a].

Name	Neural building blocks
Generator	Upsample (frames only)
Motion Synth. Net.	Skeletal Unpool Skeletal Conv (channels increase) Skeletal Conv (in-place)
Discriminator -	Skeletal Conv (in-place) Skeletal Conv (channels+frames decrease) Skeletal Pool Add Residual

## A HYPER-PARAMETERS AND TRAINING DETAILS

In this section, we describe the details for the network architectures. Table 3 describes the architecture for our generator and discriminator networks. The sizes of the kernel are configurable by hyper-parameters, and in the table we specify which hyper-parameters we have used for our best model. Note that the number of joints varies according to the topology of the skeleton on which the network is trained. The values in Table 3 belong to the skeleton used by the model presented in this work. The structure of a hierarchical level in our generator and discriminator is described in Table 4. A hierarchy level in the motion synthesis network possesses input/output skips, and hierarchy level in the discriminator possesses a residual skip, both base on Karras et al. [2020].

In our experiments, we use  $\lambda_{fcon} = 1$ ,  $\lambda_{touch} = 0.01$ , batch size 16, learning rate 0.002 for both generator and discriminator, mixing 0.9, and train for 80,000 iterations. We use pytorch version 1.5.0, CUDA version 10.1 on GeForce GTX 1080 Ti GPU.

## B MOTION REPRESENTATION CONSIDERATIONS

Some methods generate a sequence of 3D poses [Degardin et al. 2022], where each location is specified by the 3D coordinates of each joint. However, the resulting representation is incomplete, since it does not reflect the self rotations of the bones. In particular, it does not contain all the information necessary to drive a rigged virtual 3D character, and the temporal consistency of the skeleton’s bone lengths

is not guaranteed. While joint rotations may be recovered from joint positions via inverse kinematics (IK), the solution is not unique. and thus ill-posed. Furthermore, models that predict location tend to yield temporally jittery output, and require a post processing smoothing stage. Following these considerations, we follow numerous recent works that are based on joint rotation representation [Li et al. 2022; Maheshwari et al. 2022; Tevet et al. 2022]. The motion generated by MoDi can be directly converted into an animation sequence without the need to apply neither IK nor temporal smoothing.

Our network is trained on a single set of bone lengths. Once a motion is generated, it can be retargeted to any other set of bone lengths using existing motion retargeting methods [Aberman et al. 2020a,b; Biswas et al. 2021].

### C QUANTITATIVE METRICS

We briefly describe each metric that we use for quantitative results.

*FID.* Frèchet inception distance is the distance between the feature distribution of generated motions and that of the real motions, namely the difference in mean and variance. Despite its simplicity, FID is an important metric widely used to evaluate the overall quality of generated motions [Guo et al. 2020; Petrovich et al. 2021]. FID is borrowed from the image domain, where the inception network is used for features. To adjust this metric to the motion domain, the inception network is replaced by an action recognition network. A lower value implies better FID results.

*KID.* Kernel Inception Distance (KID), proposed by Bińkowski et al. [2018], compares skewness as well as the values compared in FID, namely mean and variance. KID is known to work better for small and medium size datasets. Lower values are better.

*Precision and Recall.* These measures are adopted from the discriminative domain to the generative domain [Sajjadi et al. 2018]. Precision measures the probability that a randomly generated motion falls within the support of the distribution of real images, and is closely related with fidelity. Recall measures the probability that a real motion falls within the support of the distribution of generated images, and is closely related with diversity. Higher precision and recall values imply better results.

*Diversity.* This metric measures the variance of generated motions [Guo et al. 2020; Petrovich et al. 2021]. In the context of the action recognition model, it measures the variance across all action categories, and therefore it suits an unconstrained generator. The diversity value is considered good if it is close to the diversity of the ground truth. In all our experiments, the diversity of the generated data was lower than the ground truth, so for clarity we mark it with an upwards pointing arrow, implying that for our case, higher is better.

### D ADDITIONAL QUALITATIVE RESULTS

In Figure 8 we show additional qualitative results. The reader is encouraged to watch the video in order to get the full impression from our results.

### E COMPUTING THE GRAL SCORE

Our classifier computes the *gral* score in the following way. Let  $m = [R, S, F]$  be a selected motion. Recall  $R$  represents the rotation angles of the motion. Let  $R_{rs,t}$  and  $R_{re,t}$  denote the rotations of the right shoulder and the right elbow at time  $t$ , respectively. Let  $[R_{rs,t}, \dots, R_{rs,t+8}]$  be a temporal window of size 8. A similar window is created for  $R_{re}$ . We compute the average angle in each window, and slide the window with stride of 4. Altogether we get the average computed  $T/4$  times for both the right shoulder and the right elbow. Denote the sequence of average angles by  $\alpha_{rs}$  and  $\alpha_{re}$ . The next step is to compute the difference between each element to the one preceding it, and obtain

$$score_{rs_i} = \begin{cases} 1, & \text{if } \alpha_{rs_i} > \alpha_{rs_{i-1}}, \\ 0, & \text{otherwise} \end{cases} \quad score_{re_i} = \begin{cases} 1, & \text{if } \alpha_{re_i} > \alpha_{re_{i-1}}, \\ 0, & \text{otherwise} \end{cases}$$

where  $i \in [1, T/4 - 1]$ .

Clearly, if all scores are one, the arm is opening, and if they are all zero, the arms are closing. The average of all the values in the two score vectors is used as the final attribute score.

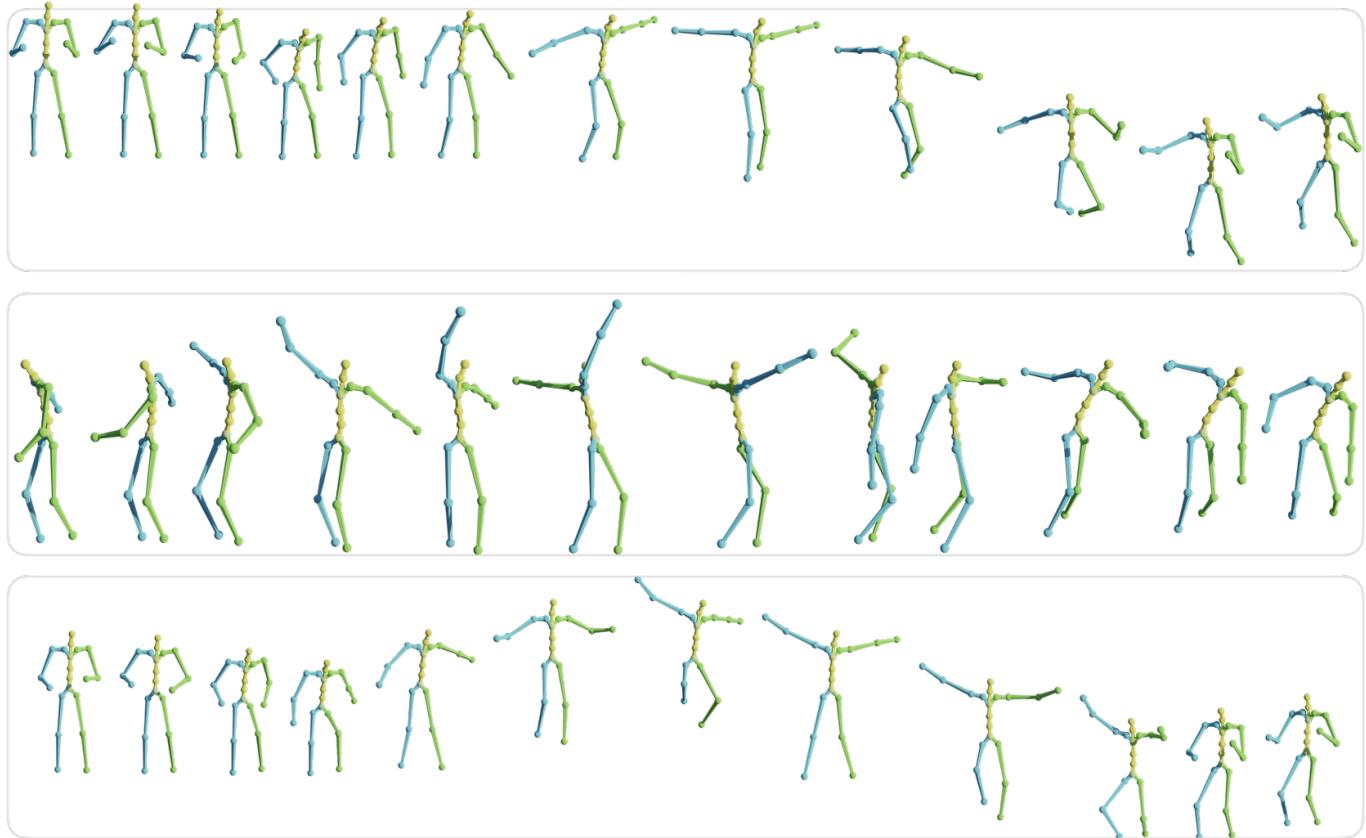


Fig. 8. Qualitative results of synthesised jumps and wild dances. See more results in the supplementary video.