DOI: 10.1111/cgf.14641
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)
Volume 41 (2022), Number 8

Facial Animation with Disentangled Identity and Motion using Transformers

Prashanth Chandran[1,2] Gaspard Zoss[2] Markus Gross [1,2] Paulo Gotardo[2] Derek Bradley[2]

[1]ETH Zurich [2]DisneyResearch|Studios

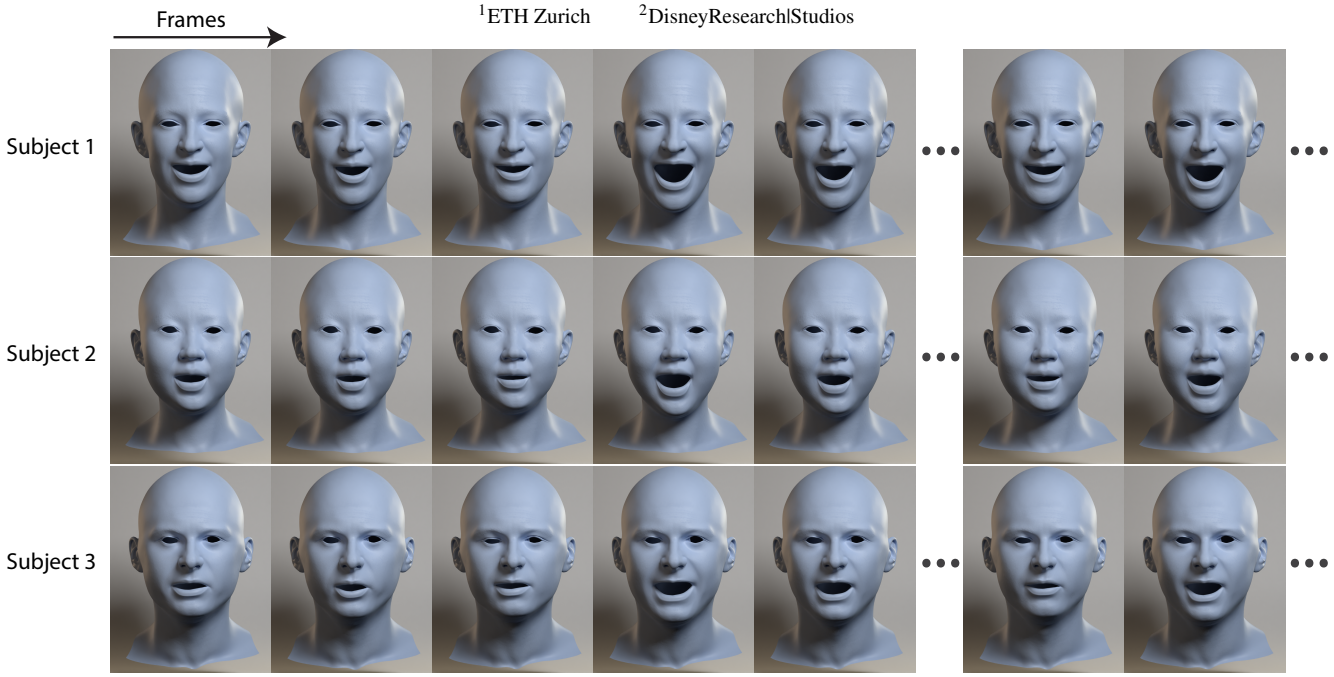

Figure 1: *We extend 3D morphable models to include the time dimension, with a novel transformer network that can synthesize and analyze 3D geometry sequences of arbitrary length. Our method disentangles facial identity from motion, allowing one to generate arbitrary animations for different subject identities. Each row above corresponds to a unique identity and the columns correspond to frames from a randomly sampled animation from the learned motion manifold of our performance transformer.*

Abstract

We propose a 3D+time framework for modeling dynamic sequences of 3D facial shapes, representing realistic non-rigid motion during a performance. Our work extends neural 3D morphable models by learning a motion manifold using a transformer architecture. More specifically, we derive a novel transformer-based autoencoder that can model and synthesize 3D geometry sequences of arbitrary length. This transformer naturally determines frame-to-frame correlations required to represent the motion manifold, via the internal self-attention mechanism. Furthermore, our method disentangles the constant facial identity from the time-varying facial expressions in a performance, using two separate codes to represent neutral identity and the performance itself within separate latent subspaces. Thus, the model represents identity-agnostic performances that can be paired with an arbitrary new identity code and fed through our new identity-modulated performance decoder; the result is a sequence of 3D meshes for the performance with the desired identity and temporal length. We demonstrate how our disentangled motion model has natural applications in performance synthesis, performance retargeting, key-frame interpolation and completion of missing data, performance denoising and retiming, and other potential applications that include full 3D body modeling.

CCS Concepts

• ***Computing methodologies*** → ***Motion processing; Shape modeling; Mesh geometry models;***

1. Introduction

In the past several years, we have witnessed a steady increase of data-driven algorithms for 3D human motion modeling. Examples include deep neural networks for solving problems like facial performance capture from monocular video [FFBB21], dynamic hand tracking [BdBT19], and full body motion reconstruction [ZYW*19]. The common thread among all data-driven methods is the need for high-quality training data. When it comes to modeling the non-rigid motion of 3D shapes in facial performances, training data can be difficult to acquire, often involving synchronized multi-camera setups, scheduling of human performers, and then time-consuming reconstruction techniques and quality checks. As a result, today's methods for data-driven facial performance applications often require strategies for dealing with the "small sample size problem", such as augmenting the dataset with synthetic examples.

When it comes to synthetic human modeling, recent generative neural networks have excelled at synthesizing images depicting realistic full-head portraits of people in static poses (*e.g.*, StyleGAN [KLA19] and its variants). But these methods are not yet able to synthesize dynamic faces with realistic non-rigid motion, as seen in a real facial performance. In terms of three-dimensional representations, morphable 3D face models [BV99] can be readily and easily sampled to synthesize novel identities and poses. However, once again we face the challenge of synthesizing dynamic motion for the sampled 3D shapes. Motion synthesis, in particular for 3D shapes, is an area that is considerably less explored. A niche area that has received some attention is the class of methods that can animate a 3D face model from audio input [KAL*17, TKY*17, RZW*21], but often using a model without proper motion priors. To date, there is still no comprehensive framework for unconstrained dynamic motion synthesis that can output arbitrarily long 3D facial performances. In the absence of such a framework, applications that require data augmentation must settle for simpler solutions for sampling facial expressions, such as random walks within the latent space of 3D models that lack temporal coherence and realism. The result is unsatisfactory data and a large domain gap for applications that target real human behavior.

In this work, we address the problem of realistically modeling and synthesizing the non-rigid deformation of 3D faces, presenting a framework that is directly applicable in several scenarios involving 3D facial performances. Key to our approach is the disentanglement of the constant facial identity component from the time-varying performance itself, given a sequence of deforming 3D shapes. To this end, our method is inspired by (and extends) the semantic model in [CBGB20] that represents static faces individually, within an *identity-agnostic expression latent space*. Here, we further consider the temporal dimension and propose a new model that can represent entire sequences of expressions in facial performances as points within an *identity-agnostic performance latent space*. This new model is designed as a transformer autoencoder, building upon transformer networks [VSP*17] that are naturally suitable for operating on data sequences with arbitrary length, such as facial performances in 3D animation. First, our transformer-based encoder converts an input neutral 3D face and a temporal sequence of blendweight vectors into an identity code and a performance code. Thanks to the transformer's self-attention mechanism, our model can automatically determine important temporal correlations between arbitrary pairs of frames, in order to learn an identity-agnostic motion manifold. As a result, we demonstrate how this model provides a means to synthesize new performances that generalize over arbitrary identities and sequence lengths (see Fig. 1). This is done using our new identity-modulated decoder, which transforms the performance code into an output performance with the desired identity and length. Our transformer autoencoder allows for arbitrary-length inputs and outputs at both training and inference time, allowing us to train on captured performances of any length. We train our model on two distinct face datasets, and further illustrate its generalization capabilities on a third dataset consisting of full 3D bodies.

While the main motivation for our work lies in facial animation and data augmentation for deep learning, our method has potential value in other fields as well. The ability to generate synthetic human motion can aid the entertainment industry in synthesizing realistic performances of background characters in films or video games. In the fast moving telepresence and metaverse field, it may be useful to generate synthetic motion of personal avatars or digital assistants. Our method can also be used for temporal data processing, offering tools like compression, denoising, and temporal upsampling. In the following, we also demonstrate applications such as performance synthesis, performance retargeting and retiming, key-frame interpolation and completion of missing performance data.

2. Related Work

So far, most work on generating 3D shapes such as faces and bodies has focused on modeling geometric variability over sets of individual shapes without any notion of temporal ordering, *e.g.*, [RBSB18, GLP*19, CBGB20, ABWB19, LBZ*20, JWCZ19]. We begin our review of related work starting with conventional 3D morphable models followed by their recent neural counterparts.

Linear Parametric Shape Models. Blend shapes [LiAR*14] are a popular, artist-friendly representation for navigating the span of a specific class of shapes. Blanz and Vetter [BV99] used principal component analysis (PCA) and proposed a 3D morphable model (3DMM) of human faces. Vlasic et al. [VBPP05] later proposed a global multi-linear model that disentangles facial identity and expression, which was extended by Wang et al. [WBZB20] to a local multilinear model offering greater expressiveness. FLAME [LBB*17] is another practical face model that incorporates skinning to articulate the jaw, neck and eyeballs. Recently, Ploumpis et al. [PVO*20] extended the human head 3DMMs to also include parts other than the face like the cranium, ears, eyes, teeth, and tongue. An excellent review of 3DMMs for human faces is given in [EST*20]. For human bodies, SMPL [LMR*15] is perhaps the most well-known, articulated linear model that has proven to be immensely useful in several applications.

Deep Shape Models. While linear 3D shape models are easy to control, they are severely limited by their expressiveness. Nonlinear, variational autoencoders were successfully adopted for modeling human faces and bodies

[BWS*18, TGLX18]. Researchers working on neural geometry processing have also leveraged graph convolutional networks [RBSB18, HHF*19, BBP*19, GCBZ19, ZWL*20], as well as other network architectures used to model static 3D shapes such as point nets operating on point clouds [QSKG17, QYSG17], Generative Adversarial Networks (GANs) [GLP*19, ABWB19], and recent diffusion-based techniques [SACOXX]. On witnessing the success of neural shape models, researchers have also attempted to semantically control them [LBZ*20, JWCZ19, CBGB20, BODO20, ABWB19, FAWB18]. In the context of faces, by disentangling facial identity and expression, either in supervised [LBZ*20, CBGB20, BODO20] or unsupervised fashion [JWCZ19, ABWB19, FAWB18], these powerful nonlinear models can be intuitively controlled by a human artist. However, unlike our model, none of these linear or deep shape models include a representation of temporal dynamics. As a result, these techniques capture solely spatial shape correlations, but not temporal correlations. Simply traversing the parametric space induced by these models does not generally provide sequences of 3D shapes showing realistic temporal deformations.

Motion Modeling. To synthesize temporal sequences with facial performances, previous works have explored the use of audio to drive a 3D face [KAL*17, TKY*17]. While the method of [KAL*17] directly outputs a 3D mesh, the method of [TKY*17] outputs animation parameters that can be used to animate a generic face rig using a static 3D morphable model. In modeling and learning the dynamics of full human bodies, DYNA [PMRMB15] extends the SMPL model by modeling soft tissue dynamics with an auto-regressive model. SoftSMPL [SGOC20] extends DYNA with an LSTM based architecture to model secondary dynamics. By reasoning about the hierarchy of joints in the human body [AKH19], researchers have also explored unconstrained human body motion generation and key-frame inpainting with recurrent models [HKPP20, HYNP20, MBR17], VAEs [YRV*18, LZCVDP20], transformers [LYC*20], generative networks [ZLB*20], and even normalizing flows [HAB20]. Recently, Li et al. [LVC*21] proposed hierarchical motion VAEs for learning a prior over human body movements. While their work shares the spirit of learning a motion manifold with ours, they learn a prior over fixed-length sequences and operate on a set human skeletal topology. Likewise another 4D model specifically tied to the SMPL body model [JZW*22] uses gated recurrent units to model temporal dynamics of human shapes. To our knowledge, no generic disentangled 4D morphable shape model like ours exists for human faces.

Transformers in Shape Modeling. Transformers were originally introduced in the context of natural language modelling [VSP*17]. Lin et al. [LWL21a] use a vanilla transformer to reconstruct coarsely posed human bodies and hands from images, and a learnable MLP to upsample the meshes to full resolution. In a follow up work [LWL21b], the authors coupled their previous vanilla transformer with graph-convolutional layers and showed better accuracy in body and hand reconstruction. More recently, Chandran et al. [Cha22] also proposed the use of a transformer architecture to capture spatial correlations across vertices in static 3D shapes. In contrast, our work uses a transformer architecture to learn temporal correlations over sequences of shapes. Transformers for generating sequences of human bodies has also been recently explored by Song et al. [SWJ*22] who concentrate on a multi-person skeleton generation use case and by Hong et al. [HZP*22] for the generation of human body animations from text input. The recent work by Petrovich et al. [PBV21] is closest in spirit to ours: it introduces *Actor*, a transformer variational autoencoder for action-conditioned generation of human body poses. In contrast to their work, our model also serves a 4D morphable model for shapes and as shown in Section 4.2, our network design converges faster and provides more accurate reconstruction on validation sequences, thanks to our novel performance encoder and styled transformer decoder.

In summary, we believe our work presents the first 4D morphable model that can represent rich, coherent human shapes, with a variety of applications, as demonstrated in the following.

3. Disentangled Motion Model

In this section, we describe our transformer-based architecture that introduces a notion of time into deep geometry models. Although we describe the method in the context of facial performance, we also show in Section 4 that our method can also be used to model dynamic motion of other shapes as well, such as full human bodies.

3.1. Network Architecture

An overview of our network, a performace autoencoder for disentangled motion modeling, is shown in Fig. 2. At a high level, the input to our method is (*i*) a neutral 3D face shape with a particular identity, and (*ii*) a sequence of blend weights that describe a facial performance. Note that, by design, identity and performance are already disentangled on the input side. The two inputs are separately fed into an identity shape encoder and a performance encoder, yielding an identity latent code, $\mathbf{z}_{id}$, and a performance latent code, $\mathbf{z}_{perf}$. These two codes are then supplied to the single decoder, which in turn reproduces the output performance with the chosen facial identity.

While the encoders allow us to represent the identity and the overall performance with a single pair of codes, the decoder allows us to regenerate the performance with optionally different identity and temporal length. To accomplish this decoding task, the output $\mathbf{z}_{id}$ and $\mathbf{z}_{perf}$ are first position-encoded (for the desired output performance duration) and fed into the decoder, which has a transformer architecture with style-based modulation. The decoder transforms the sequence of position-encoded inputs into an output sequence of latent shape codes. Finally, this sequence of output codes are individually passed through a shape decoder to produce the 3D shapes for the output performance. All modules in our architecture are trained end-to-end, in a fully supervised manner, by encoding and decoding the training performances with the original (same) identity and temporal length.

Once trained, our model offers a disentangled latent space of facial identities and performances that can be freely combined to generate previously unseen, new outputs. When decoding a performance with a single 3D shape, our model can behave as a conventional 3D morphable shape model. But most important here is the added ability to model the dimension of motion over time. Our

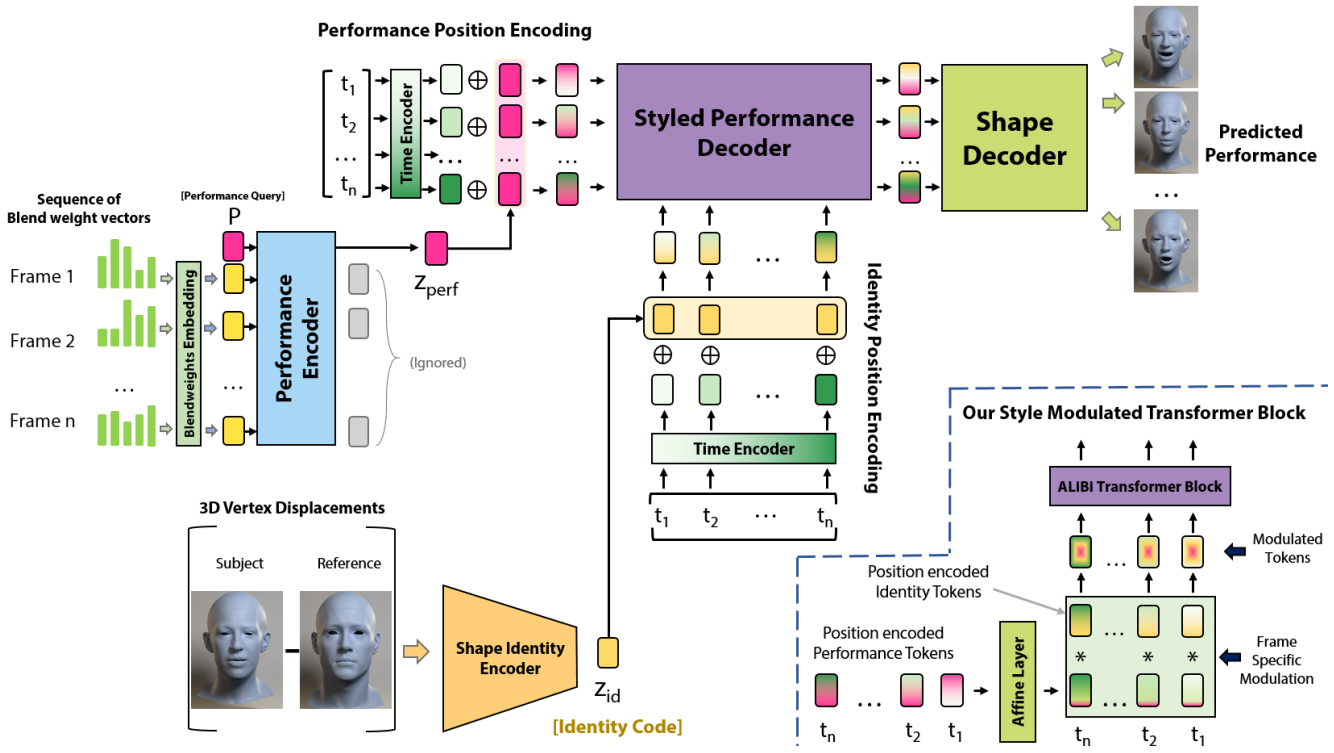

Figure 2: *Our disentangled motion model is designed as a transformer-based autoencoder and leverages self-attention to capture temporal correlations in sequences of vectors with blend shape weights. The two separate encoders yield a pair of identity and performance codes ($\mathbf{z}_{id}$, $\mathbf{z}_{perf}$), which the single decoder transforms back into an output sequence of 3D shapes with the desired identity and performance length.*

4-D morphable model can therefore serve as a powerful prior for many applications in deformable shape modeling, as illustrated in Section 4. Next, we discuss the various components of our network and their designs in further detail.

3.1.1. Identity Shape Encoder

Following Chandran et al. [CBGB20], we model our identity shape encoder as a simple multilayer perceptron (MLP) with 4 linear residual layers followed by GeLU activations. The identity is provided as a neutral shape mesh (i.e., without expression) and we subtract a canonical face shape (*e.g.*, the mean of the dataset), in order to obtain small per-vertex 3D displacements, which are flattened into an input vector. The output of the shape encoder is a single 128-dimensional latent vector which we refer to as the identity code $\mathbf{z}_{id}$. The identity code captures the shape of the given subject in the neutral expression. It is important to note that this shape encoder only ever receives neutral faces of subjects (with different identities), helping us achieve explicit disentanglement between facial identities and expressions over time. In contrast, and as described next, the performance encoder only receives subject-agnostic performance data (codes describing generic expressions, such as blend shape weights). It is the task of the transformer decoder to combine the information and model the subject-specific expressions and motion of the face. While all results in our paper use this simple MLP as the identity encoder, we evaluate different architecture choices in our supplemental document.

3.1.2. Performance Encoder

The goal of this network module is to encode a facial performance into a condensed latent representation. As facial performances can be of arbitrary duration, it is important for the performance encoder to be able to handle input sequences of varying lengths, which can also represent small parts of longer performances. For these reasons, we model our performance encoder as a transformer, an architecture that is naturally suited for handling sequences of arbitrary length. As such, the transformer encoder takes an arbitrarily long performance as input and always generates a single 128-dimensional latent performance code $\mathbf{z}_{perf}$.

Another important component of our model is that the performance encoder should be identity-agnostic in order to achieve the desired disentanglement. A convenient, identity-agnostic input representation of a facial shape is a set of blend shape weights, which can be used to blend subject-specific expressions to generate desired final shapes [LiAR*14]. Facial models based on blend shapes are very common, as they tend to be semantically meaningful and offer artists an intuitive means of interaction. Therefore, we represent facial performances as sequences of blend weight vectors, which will be encoded by our performance encoder.

Each frame of a performance is represented as a blend weight vector, which we referred to as a *token*, following the transformer literature. These vectors, on their own, present no notion of time. To be processed by a transformer in a meaningful manner, the blend

weight vectors have to be position-encoded. Two types of position-encoding are commonly used, absolute and relative. Absolute sinusoidal position-encoding involves the addition of a fixed set of sinusoids, each corresponding to a unique position, onto the input tokens. However, recent work has shown that relative position-encoding via the ALiBi attention mechanism [PSL21] gives superior performance and extrapolates better to longer sequences at inference time. We thus adopt relative encoding and first perform blend weight embedding by applying linear projections to each input token, independently, before passing them through our performance encoder, which uses the ALiBi position encoding scheme at each layer. Our performance encoder has 4 transformer blocks with ALiBi position encoding. The performance encoder then mixes information across its input tokens and produces an equal number of output tokens. To extract a single performance latent code, we follow a common strategy [DCLT19, RBK21, PBV21] and append an additional token **P** to the input. **P** is a global, 128-dimensional *performance query* code that is optimized with the network weights. Given the transformer output tokens, we extract only the one corresponding to **P**, which gives the desired 128-dimensional performance code $\mathbf{z}_{perf}$. This code encapsulates the temporal dynamics of the input blend weight vectors, in a condensed manner, within the learned latent space of performances. Note that the identity and performance codes need not be of the same dimension.

3.1.3. Style-Modulated Transformer Decoder

So far we have reduced the neutral shape of a subject and a performance sequence of blend weight vectors into two latent codes, $\mathbf{z}_{id}$ and $\mathbf{z}_{perf}$. Our goal now is to combine the identity and performance to obtain a subject-specific motion representation. To allow for variable-length outputs and to properly leverage temporal correlations during the decoding of 3D performances, we also model the decoder as a transformer. Here, we introduce a novel style-based transformer architecture that achieves better reconstructions and faster convergence when compared to a standard transformer decoder (see results in Section 4.2).

We query the decoder by providing it with a sequence of input tokens, whose number indicate the length of the desired output performance. Each *identity token* is a position-encoded version of the identity code $\mathbf{z}_{id}$, while each *performance token* is a position-encoded version of $\mathbf{z}_{perf}$. These tokens are injected into each transformer layer of the decoder and combined via style-based modulation, Fig. 2 (bottom-right). The output of the decoder is a sequence of latent shape tokens, each of which encodes information on the desired identity and the expression at the particular frame in time.

Time Encoder. As discussed in Section 3.1.2 for the encoder, relative position-encoding (PE) using ALiBi can outperform absolute PE with sinusoids. However, here, decoding with relative PE would result in a sequence of constant input tokens that simply duplicates $\mathbf{z}_{id}$ and $\mathbf{z}_{perf}$ to achieve the desired output length; the decoder would hardly be able to successfully reconstruct a performance. For this reason, we resort to absolute PE of $\mathbf{z}_{id}$ and $\mathbf{z}_{perf}$ in our decoder. Standard PE defines a fixed set of sinusoids for discrete positions in time and adds these to the input tokens. In our case, however, this discretization affects our ability to freely and continuously sample (interpolate) our temporal domain to decode sequences of arbitrary length. We thus adopt an alternative PE scheme that is simple, yet powerful: we define the decoder input as a sequence of scalars $t_i \in [0,1]$ that represent normalized time indices of the desired frames to be decoded. We then learn the PE $\gamma(t_i)$ of each t_i together with our network, by modeling the mapping $\gamma(\cdot)$ as an additional *time encoder* MLP $\gamma(\cdot)$. More specifically, we model $\gamma(\cdot)$ as an MLP with sinusoidal SiRen activations [SMB*20]. Each encoded token $\gamma(t_i)$ is then added with a $\mathbf{z}_{id}$ or $\mathbf{z}_{perf}$ to complete our continuous PE. We validate the performance of our new PE scheme versus standard sinusoidal PE in Section 4.2 and in the supplementary material.

Style-Based Modulation. As illustrated in Fig. 2 (bottom-right), the position-encoded *performance tokens* are further individually passed through an additional affine layer that extracts frame-specific information from each performance token, leading to an equal number of *expression tokens*. Each per-frame expression token is then modulated by the position-encoded *identity token* at their corresponding instance in time, before they are mapped onto queries, keys and values inside the transformer.

Decoder Architecture. Our performance decoder consists of 4 style-modulated transformer layers. The performance tokens are converted into per-frame expression tokens at each of the 4 layers of the decoder, resulting in layer-specific tokens that are modulated by the corresponding identity token. Our style modulation with performance tokens can be thought of as skip connections from the performance latent space into different levels of the transformer decoder. We empirically observe that, analogous to the effect of skip connections in residual networks, our style modulation at multiple stages of the decoder allows for faster convergence and better performance, likely due to better gradient flow during training. Other than the style modulation, our decoder uses standard transformer blocks with residual connections, layer normalizations, and GeLU activations. The output of our transformer decoder is a sequence of latent tokens which have both identity and frame-specific expression information. This sequence of output tokens are then passed independently through a shape decoder to reconstruct the output sequence of shapes.

3.1.4. Shape Decoder

The shape decoder performs the final step of converting the per-frame output tokens from the decoder into per-frame subject-specific 3D shapes. The architecture of our shape decoder is similar to that of the identity shape encoder. We use a residual MLP of 4 layers and GeLU activations. The shape decoder processes each token independently and predicts a list of 3D vertex offsets, which are added to the canonical shape to produce the desired geometry for each frame. We evaluate and compare alternative architecture choices for both the shape encoder and decoder in our supplemental material.

4. Results

Our disentangled motion model naturally lends itself to several applications in 4D shape modeling. This section first describes the datasets our network was trained on, and then shows reconstruction results as validation. It also evaluates some of our design

choices, presents ablation studies, and finally highlights applications of our motion model. For training details, additional applications (re-timing performances and mixing styles across performances), and encoding robustness, please refer to the supplementary document.

4.1. Datasets

To show generality of our method, we apply our motion model separately on three different datasets, including 3D faces and full bodies, as described below.

SDFM: The SDFM dataset consists of 3D face meshes introduced for semantic deep face modeling [CBGB20]. The data includes both static facial expressions as well as tracked dynamic performances for a subset of the individuals. In this work, we use only the subset of data corresponding to the 20 subjects with dynamic performances (which includes both dialog speech and dynamic expressions). To make the data compatible with our network, we used the 24 expressions to build a blend shape model for each actor, and then converted the performances from mesh sequences to blend weight sequences by fitting the blend shapes to each frame of geometry (following [CBGB20]). The meshes contain 5257 vertices in correspondence, and in total we obtained 114 performance sequences totaling approximately 23000 frames. We used a random sample of 90 sequences for training, and the rest for validation. Fig. 3 (left) shows the reconstruction for 2 frames of a validation performance for the SDFM dataset.

COMA: This dataset also contains 3D face meshes [RBSB18], each with 5023 vertices. The data includes 12 individuals each performing 12 dynamic expressions, for a total of 144 performances and 20465 combined frames of geometry. All 144 performances were used for training, minus randomly chosen sequences of 60 consecutive frames from 20 different performances that were used for validation. Following a similar strategy as with SDFM, we chose the 12 extreme expressions to create a blend shape model per actor, and converted the mesh sequences to blend weight sequences for our network. Fig. 3 (center) shows the reconstruction for 2 frames of a validation performance for COMA.

AMASS: This is a large database of human motion capture [MGT*19] with SMPL parameterization [LMR*15] for its dynamic sequences. Although this parameterization is inherently shape and pose disentangled, the human body spans a much more diverse space of movements than faces and might bring forth different challenges. To demonstrate that our method can learn motion manifolds under such challenges scenarios as well, we train our method on a subset (CMU, DanceDB, KIT) of the AMASS dataset. Our performance encoder now receives a sequence of SMPL pose parameters as input instead of blend weights (Kindly refer to our supplementary material for details on how we modify our architecture to train on human bodies). We train our performance decoder by using the SMPL model as a fixed differentiable module to decode vertex positions identical to [PCG*19, PBV21]. We leave out a random subset of 20 sequences for validation. Fig. 3 (right) shows the reconstruction for 2 frames of a validation performance for the AMASS dataset.

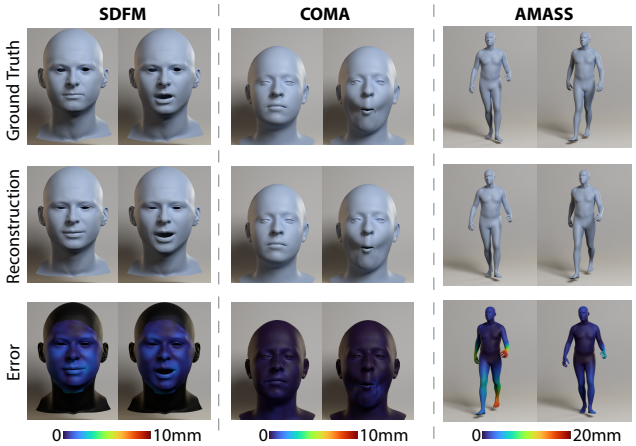

Figure 3: *We show 2 frames from reconstructed validation performances for each of the datasets we apply our method on. Given that the entire performance is compressed to a single 128-dimensional code, reconstruction results are comparatively close.*

Table 1: *Performance Reconstruction Errors on 3 different datasets*

Dataset	Validation error (mm)
SDFM [CBGB20]	1.47
COMA [RBSB18]	0.62
AMASS (CMU, KIT, DanceDB) [MGT*19]	7.19

For all three separately-trained models, a single 128-dimensional performance code was able to reproduce the sequences of movements after position-encoding. In Table 1, we show the validation reconstruction error on the three datasets. Please also refer to the reconstructed performances in the supplemental video. These reconstruction results demonstrate that our transformer architecture can serve as a compressed motion manifold for the generation of moving 3D shapes, including facial and full-body performances.

4.2. Ablation Studies

We now present experiments that motivate several of our design choices. To keep compute costs low, we perform our ablation on a subset of the SDFM dataset consisting of 23 dynamic facial performances (≈5000 frames) of a single subject. We leave out 4 performances for validating the performance of our different variants.

Architecture Design. As mentioned earlier, a related technique for generating human motion sequences is the *Actor* model [PBV21], which is based on a transformer variational autoencoder. We aim to understand if a similar network design would perform sufficiently well in our setting, and therefore we replaced our performance encoder with the variational encoder from [PBV21] and our style-modulated transformer decoder with a standard transformer decoder identical to the one used in *Actor*. For a fair comparison, we adjusted the size of the both models to keep their capacities approximately the same. We refer to this modified version of our architecture as the *Simple* variant, and we

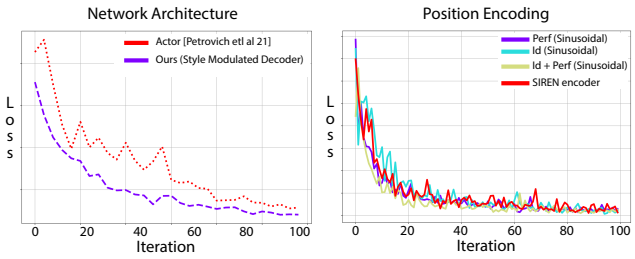

Figure 4: *(Left) Our style modulated transformer decoder converges to a lower error faster than a simple transformer encoder - decoder architecture. (Right) Position-encoding both the identity and performance codes results in marginally the best performance. Our SiRen time encoder extrapolates better to unseen sequence lengths without sacrificing performance.*

trained both architectures on a dataset of real world facial performances from a single subject. Fig. 4 analyzes the convergence and reconstruction behavior of the two architectures. As we can see, our proposed architecture not only converges to a lower error much faster at training time, but also achieves lower error on validation performances, justifying our novel network design.

Position-Encoding. In Section 3.1 we describe that the latent codes corresponding to the identity and the performance are both position-encoded before being passed to the transformer decoder. The decoder requires a minimum of at least one of the codes to be position-encoded, and thus we have three options: (1) position-encode only the identity code, (2) position-encode only the performance code, or (3) position-encode both. We evaluate all three of these options with a small experiment on our ablation dataset, where we use conventional sinusoids that are added on top of the corresponding latent codes. Fig. 4 illustrates that position-encoding both the shape code and the performance code results in the fastest convergence of the transformer and also provides the highest reconstruction quality. Finally, replacing the fixed set of sinusoids with the SIREN-based MLP, as described in Section 3.1, has little effect on reconstruction quality and convergence speed. However, it makes our position encoding compatible with continuous interpolation and allows us to optimize for temporal shifts as explained in our key-frame projection experiment below.

Shape Encoder and Decoder. Our method itself is agnostic to the choice of the encoder and decoder that are used to represent 3D shapes. This allows us to readily leverage advances in graph/mesh convolution and other deep learning techniques, extending them to model motion as well. In the supplemental document, we present an additional ablation study where we replace the fully connected shape encoder and decoder modules with a state-of-the-art graph convolution technique [GCBZ19]. The supplemental material also contains additional ablation studies on the effect of network capacity and variable sequence length training on our model.

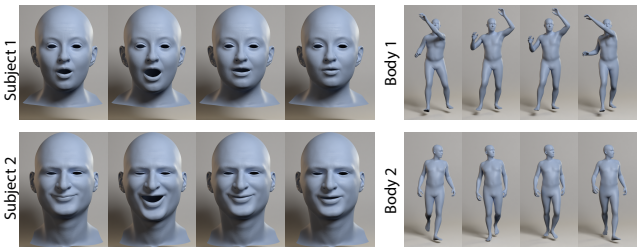

Figure 5: *One of our main applications is generating novel performances, which we demonstrate here (and in the suppl. material). Left: novel performances generated by our method trained on the SDFM dataset. Right: two different motions sampled from our model trained on the AMASS dataset.*

Figure 6: *Our method allows to easily retarget performances from one individual (row 1) to any others (rows 2 and 3).*

4.3. Applications

We now illustrate different applications of our disentangled motion model for 3D+time shape manipulations. As we demonstrate example applications on human motion, please refer to the supplemental video where the animated results can be properly appreciated.

Generating Novel Performances. A natural application of our method is the generation of novel performances. Generating automatic animation and synthetic data for training neural networks are main motivations for our work. Importantly, the generated motions should be coherent, smooth and look natural. Once our network is trained, we can sample the latent space to obtain new performance codes, which can be mapped onto any identity through our transformer decoder. We treat the performance latent space as a multivariate Gaussian distribution for sampling. Fig. 5 shows novel performances on sampled identities from the SDFM dataset and two novel performances generated by our model trained on the AMASS subset. The movements generated by our model are smooth, realistic and also capture a wide variety of deformations.

Performance Retargeting. Our method naturally disentangles the latent identity space from the performance space. This means that we can fix the performance code and simply vary the identity code to retarget a performance onto different identities. Fig. 6 shows performance retargeting results, where we encoded a captured performance from one actor (row 1) and then sampled two different latent identity codes to reconstruct the corresponding performance

Figure 7: *Two input performances (rows 1 and 3) can be interpolated, as we show here at* 50% *interpolation (row 2). The trajectories of a point on the lower lip (last column) show that interpolation is more than just simple blending of the input frames.*

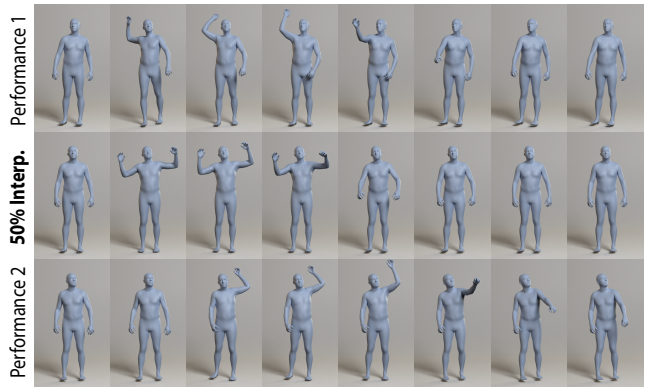

Figure 8: *Our method can seamlessly interpolate complex performances on human bodies too. In this figure, two counteracting performances; throwing with the right hand (rows 1) and throwing with the left (row 3) can be interpolated, to result in interesting inbetweens as shown here at* 50% *interpolation (row 2).*

on the target identities (rows 2 and 3). Note that the decoder mixes identity- and expression-specific information, allowing it to capture identity-specific facial deformation, which is an important aspect for maintaining realism. Performance retargeting is an important application on its own, or can also be used as a means to generate additional large amounts of 3D animations (by retargeting a corpus of captured performances onto a variety of synthetic characters).

Interpolation and Extrapolation. An interesting application of 4D motion models is interpolating between two or more different performances, which can be challenging in the case of performances with different lengths. In our framework, performance interpolation is easily achieved by interpolating the performance codes in the latent space of the model. Once decoded, the result is a non-linear interpolation of the inputs. Interpolating in latent space trivially addresses the case of performances with different lengths. In Fig. 7, we show the result of 50% interpolation between two different performance codes. We also visualize the trajectory of a point on the lower lip for both the original and interpolated performances, as well as 25% and 75%. Note how the interpolated performance is more than just a simple blending of the per frame shapes, and produces a completely new performance with new timing and expression transitions, yet still captures the essence of the original performances. Note that while interpolating performances as a whole, the model does not interpolate expressions in a pairwise manner, but actually allows for both neighboring and distant frames to be influenced through self-attention. As such, while one cannot intuitively control the interpolated performance due to the nonlinear nature of our transformer decoder, linear interpolation of performance codes always produces a plausible, temporally smooth performance, highlighting the smooth motion manifold learned by our model. In Fig. 8, we show a similar example of performance interpolation on human bodies too. Please refer to our supplementary material for more animations.

Performance extrapolation is also easily enabled by our model. As the length of the generated output sequence is dictated by the length of the position-encoding of the latent codes, one can artistically lengthen a performance by feeding an additional number of position-encoded identity codes as input to the performance decoder. As an experiment, we take our model trained on sequences of 60 frames, and then queried longer performances of up to 120 frames at inference time. Our method produces plausible shape deformations even for sequences much longer than what it was trained on, Fig. 9 (left).

Inpainting by Projection. Like any other morphable model, our temporal motion model allows for the projection of new shapes into its latent space. We formulate this projection step as an optimization problem. Specifically, we optimize for a latent identity and performance code that, when position-encoded and fed through the pre-trained decoder, reproduces the given (potentially incomplete or corrupted) performance data. Thus, our motion manifold can be used to not only project full performances but also to in-paint partial animations with missing data, or even sparse key-frame animations. In these cases the optimization objective is modified such that the reconstructed performance matches the available frames only, naturally filling in the rest with coherent motion. In Fig. 9 (right), we show an example of projecting a set of evenly spaced key-frames into our motion manifold. Existing morphable models have no notion of time and can only linear interpolate between key-frame poses. In contrast, our result produces more interesting non-linear interpolation of the key-frames.

Performance Blending and De-noising. Noisy or implausible performances can also be projected into our motion manifold to obtain more natural motions. To demonstrate this effect we stack the blend weight vector sequences of two discontinuous facial performances as a single temporal sequence and feed this to our performance encoder. The resulting performance reconstructed through the decoder shows a smooth transition from one performance to the other (Fig. 9).

5. Conclusion

We propose a new 3D+time framework for modeling and realistically synthesizing arbitrary dynamic motion for 3D shapes like

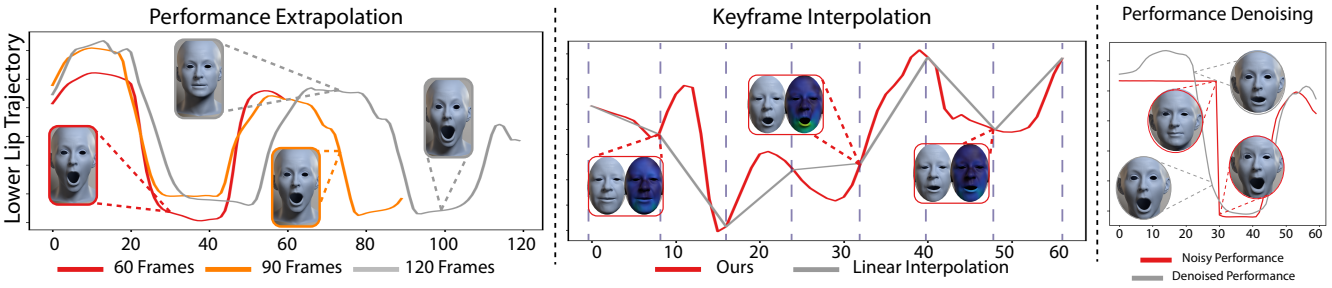

Figure 9: ***Extrapolation****: (left) We trace the (y-axis) trajectory of a point on the lower lip when using the same performance code to query output performances of 60, 90 and 120 frames respectively. Here, the performance code corresponds to random transitions between extreme facial expressions, creating large non-linear movements. Our method is able to continue the performance for durations never seen during training.* ***Key-frame interpolation****: (middle) Our pre-trained network can be used as a motion prior for projecting key-framed facial expressions. Our motion manifold allows for non-linear inpainting of missing frames, while also respecting the key-frame constraints, compared to the linear interpolation which results in a more robotic performance.* ***Performance Denoising****: (right) We de-noise a discontinuous performance by projecting a sequence with a strong discontinuity into our performance latent space. The reconstructed performance plausibly smooths out the transition, as shown by the trajectory of a lower lip point.*

human faces, with demonstrated extension to full 3D bodies. By design, our transformer-based architecture naturally models the motion manifold of performances while disentangling the time-varying shape deformation from the constant identity component in the performance. This capability allows our model to generalize better and to synthesize performances with arbitrary identity and length. We show applications of novel performance generation, retargeting, interpolation, extrapolation, projection and more. The main limitation of our method is the tradeoff between performance compression and reconstruction quality. Naturally, a single 128-dimensional performance code cannot represent all the information in a very long performance. An interesting direction for future work is thus the optimal partitioning of long performances into segments that are better suited for encoding. Another limitation of our work is that we currently do not model physically based constraints. As a result, we cannot always guarantee high-quality geometry around regions like the lips (*e.g.*, lip contacts, lip stickiness, *etc.*) for faces, and cannot prevent self intersections in the case of human bodies. Incorporating physical (anatomical) constraints into our method could be very beneficial in future work, to further improve the quality of results, especially for human bodies. Nevertheless, our disentangled motion model shows great potential in the automatic generation of realistic animation, in the 4D manipulation of animation data, and is also ideally suited for augmenting datasets with coherent synthetic 3D performances for deep learning applications.

References

[ABWB19] ABREVAYA V. F., BOUKHAYMA A., WUHRER S., BOYER E.: A generative 3d facial model by adversarial training. *CoRR abs/1902.03619* (2019). arXiv:1902.03619. 2, 3

[AKH19] AKSAN E., KAUFMANN M., HILLIGES O.: Structured prediction helps 3d human motion modelling. In *The IEEE International Conference on Computer Vision (ICCV)* (Oct 2019). First two authors contributed equally. 3

[BBP*19] BOURITSAS G., BOKHNYAK S., PLOUMPIS S., ZAFEIRIOU S., BRONSTEIN M.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *Int. Conf. Comput. Vis.* (2019), pp. 7212–7221. 3

[BdBT19] BOUKHAYMA A., DE BEM R., TORR P. H.: 3d hand shape and pose from images in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2019). 2

[BODO20] BAILEY S. W., OMENS D., DILORENZO P., O'BRIEN J. F.: Fast and deep facial deformations. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 4 (Aug. 2020), 94:1–15. Presented at SIGGRAPH 2020, Washington D.C. doi:10.1145/3386569.3392397. 3

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *ACM SIGGRAPH* (1999), vol. 99, pp. 187–194. 2

[BWS*18] BAGAUTDINOV T. M., WU C., SARAGIH J. M., FUA P., SHEIKH Y.: Modeling facial geometry using compositional vaes. *IEEE Conf. Comput. Vis. Pattern Recog.* (2018), 3877–3886. 3

[CBGB20] CHANDRAN P., BRADLEY D., GROSS M., BEELER T.: Semantic deep face models. In *International Conference on 3D Vision* (2020), pp. 345–354. 2, 3, 4, 6

[Cha22] Shape transformers: Topology-independent 3d shape models using transformers. In *Eurographics* (2022). 3

[DCLT19] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805. 5

[EST*20] EGGER B., SMITH W. A. P., TEWARI A., WUHRER S., ZOLLHOEFER M., BEELER T., BERNARD F., BOLKART T., KORTYLEWSKI A., ROMDHANI S., THEOBALT C., BLANZ V., VETTER T.: 3d morphable face models - past, present and future. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 5 (2020). 2

[FAWB18] FERNÁNDEZ ABREVAYA V., WUHRER S., BOYER E.: Multilinear autoencoder for 3d face model learning. In *Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on* (2018). 3

[FFBB21] FENG Y., FENG H., BLACK M. J., BOLKART T.: Learning an animatable detailed 3d face model from in-the-wild images. *ACM Trans. Graphics (Proc. SIGGRAPH) 40*, 4 (2021). 2

[GCBZ19] GONG S., CHEN L., BRONSTEIN M., ZAFEIRIOU S.: Spiralnet++: A fast and highly efficient mesh convolution operator. In *Int. Conf. Comput. Vis. Workshops* (2019). 3, 7

[GLP*19] GECER B., LATTAS A., PLOUMPIS S., DENG J., PAPAIOANNOU A., MOSCHOGLOU S., ZAFEIRIOU S.: Synthesizing coupled 3d

face modalities by trunk-branch generative adversarial networks. *ArXiv abs/1909.02215* (2019). 2, 3

[HAB20] HENTER G. E., ALEXANDERSON S., BESKOW J.: Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 6 (nov 2020). 3

[HHF*19] HANOCKA R., HERTZ A., FISH N., GIRYES R., FLEISHMAN S., COHEN-OR D.: Meshcnn: A network with an edge. *ACM Trans. Graphics (Proc. SIGGRAPH) 38*, 4 (2019). 3

[HKPP20] HOLDEN D., KANOUN O., PEREPICHKA M., POPA T.: Learned motion matching. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 4 (jul 2020). 3

[HYNP20] HARVEY F. G., YURICK M., NOWROUZEZAHRAI D., PAL C.: Robust motion in-betweening. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 4 (jul 2020). 3

[HZP*22] HONG F., ZHANG M., PAN L., CAI Z., YANG L., LIU Z.: Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–19. doi:10.1145/3528223.3530094. 3

[JWCZ19] JIANG Z.-H., WU Q., CHEN K., ZHANG J.: Disentangled representation learning for 3d face shape. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2019). 2, 3

[JZW*22] JIANG B., ZHANG Y., WEI X., XUE X., FU Y.: H4d: Human 4d modeling by learning neural compositional representation, 2022. 3

[KAL*17] KARRAS T., AILA T., LAINE S., HERVA A., LEHTINEN J.: Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. Graphics (Proc. SIGGRAPH) 36*, 4 (2017). 2, 3

[KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2019), pp. 4401–4410. 2

[LBB*17] LI T., BOLKART T., BLACK M. J., LI H., ROMERO J.: Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 36*, 6 (2017). 2

[LBZ*20] LI R., BLADIN K., ZHAO Y., CHINARA C., INGRAHAM O., XIANG P., REN X., PRASAD P., KISHORE B., XING J., LI H.: Learning formation of physically-based face attributes. In *IEEE Conf. Comput. Vis. Pattern Recog.* (June 2020). 2, 3

[LiAR*14] LEWIS J. P., ICHI ANJYO K., RHEE T., ZHANG M., PIGHIN F. H., DENG Z.: Practice and theory of blendshape facial models. In *Eurographics* (2014). 2, 4

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia) 34*, 6 (2015), 248:1–248:16. 2, 6

[LVC*21] LI J., VILLEGAS R., CEYLAN D., YANG J., KUANG Z., LI H., ZHAO Y.: Task-generic hierarchical human motion prior using vaes, 2021. arXiv:2106.04004. 3

[LWL21a] LIN K., WANG L., LIU Z.: End-to-end human pose and mesh reconstruction with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2021). 3

[LWL21b] LIN K., WANG L., LIU Z.: Mesh graphormer. In *Int. Conf. Comput. Vis.* (2021). 3

[LYC*20] LI J., YIN Y., CHU H., ZHOU Y., WANG T., FIDLER S., LI H.: Learning to generate diverse dance motions with transformer. *ArXiv abs/2008.08171* (2020). 3

[LZCVDP20] LING H. Y., ZINNO F., CHENG G., VAN DE PANNE M.: Character controllers using motion vaes. *ACM Trans. Graphics (Proc. SIGGRAPH) 39*, 4 (jul 2020). 3

[MBR17] MARTINEZ J., BLACK M. J., ROMERO J.: On human motion prediction using recurrent neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 4674–4683. 3

[MGT*19] MAHMOOD N., GHORBANI N., TROJE N. F., PONS-MOLL G., BLACK M. J.: AMASS: Archive of motion capture as surface shapes. In *Int. Conf. Comput. Vis.* (Oct. 2019), pp. 5442–5451. 6

[PBV21] PETROVICH M., BLACK M. J., VAROL G.: Action-conditioned 3D human motion synthesis with transformer VAE. In *Int. Conf. Comput. Vis.* (2021), pp. 10985–10995. 3, 5, 6

[PCG*19] PAVLAKOS G., CHOUTAS V., GHORBANI N., BOLKART T., OSMAN A. A. A., TZIONAS D., BLACK M. J.: Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2019). 6

[PMRMB15] PONS-MOLL G., ROMERO J., MAHMOOD N., BLACK M. J.: Dyna: A model of dynamic human shape in motion. *ACM Trans. Graphics (Proc. SIGGRAPH) 34*, 4 (Aug. 2015), 120:1–120:14. 3

[PSL21] PRESS O., SMITH N. A., LEWIS M.: Train short, test long: Attention with linear biases enables input length extrapolation, 2021. arXiv:2108.12409. 5

[PVO*20] PLOUMPIS S., VERVERAS E., O'SULLIVAN E., MOSCHOGLOU S., WANG H., PEARS N., SMITH W., GECER B., ZAFEIRIOU S. P.: Towards a complete 3d morphable model of the human head. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020). 2

[QSKG17] QI C. R., SU H., KAICHUN M., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2017), pp. 77–85. 3

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (2017), p. 5105–5114. 3

[RBK21] RANFTL R., BOCHKOVSKIY A., KOLTUN V.: Vision transformers for dense prediction, 2021. arXiv:2103.13413. 5

[RBSB18] RANJAN A., BOLKART T., SANYAL S., BLACK M. J.: Generating 3d faces using convolutional mesh autoencoders. In *Eur. Conf. Comput. Vis.* (2018). 2, 3, 6

[RZW*21] RICHARD A., ZOLLHOEFER M., WEN Y., DE LA TORRE F., SHEIKH Y.: Meshtalk: 3d face animation from speech using cross-modality disentanglement, 2021. arXiv:2104.08223. 2

[SACOXX] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graphics (Proc. SIGGRAPH) XX*, X (20XX). 3

[SGOC20] SANTESTEBAN I., GARCES E., OTADUY M. A., CASAS D.: SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans. *Computer Graphics Forum (Proc. Eurographics)* (2020). 3

[SMB*20] SITZMANN V., MARTEL J. N., BERGMAN A. W., LINDELL D. B., WETZSTEIN G.: Implicit neural representations with periodic activation functions. In *Proc. NeurIPS* (2020). 5

[SWJ*22] SONG Z., WANG D., JIANG N., FANG Z., DING C., GAN W., WU W.: Actformer: A gan transformer framework towards general action-conditioned 3d human motion generation, 2022. 3

[TGLX18] TAN Q., GAO L., LAI Y. K., XIA S.: Variational Autoencoders for Deforming 3D Mesh Models. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2018). 3

[TKY*17] TAYLOR S., KIM T., YUE Y., MAHLER M., KRAHE J., RODRIGUEZ A. G., HODGINS J., MATTHEWS I.: A deep learning approach for generalized speech animation. *ACM Trans. Graphics (Proc. SIGGRAPH) 36*, 4 (2017). 2, 3

[VBPP05] VLASIC D., BRAND M., PFISTER H., POPOVIĆ J.: Face transfer with multilinear models. *ACM Trans. Graphics (Proc. SIGGRAPH) 24*, 3 (July 2005), 426–433. 2

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U., POLOSUKHIN I.: Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), vol. 30. 2, 3

[WBZB20] WANG M., BRADLEY D., ZAFEIRIOU S., BEELER T.: Facial expression synthesis using a global-local multilinear framework. *Eurographics 39*, 2 (2020), 235–245. 2

[YRV*18] YAN X., RASTOGI A., VILLEGAS R., SUNKAVALLI K., SHECHTMAN E., HADAP S., YUMER E., LEE H.: Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *European Conference on Computer Vision* (2018), Springer, pp. 276–293. 3

[ZLB*20] ZHOU Y., LU J., BARNES C., YANG J., XIANG S., LI H.: Generative tweening: Long-term inbetweening of 3d human motions. *ArXiv abs/2005.08891* (2020). 3

[ZWL*20] ZHOU Y., WU C., LI Z., CAO C., YE Y., SARAGIH J., LI H., SHEIKH Y.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. In *Advances in Neural Information Processing Systems* (2020). 3

[ZYW*19] ZHENG Z., YU T., WEI Y., DAI Q., LIU1 Y.: Deephuman: 3d human reconstruction from a single image. In *Int. Conf. Comput. Vis.* (2019). 2