PoseGPT: Quantization-Based 3D Human Motion Generation and Forecasting

Thomas Lucas$^{(\boxtimes)}$, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez

NAVER LABS Europe, Meylan, France
`thomas.lucas@naverlabs.com`
`https://europe.naverlabs.com/research/computer-vision/posegpt`

Abstract. We address the problem of action-conditioned generation of human motion sequences. Existing work falls into two categories: forecast models conditioned on observed past motions, or generative models conditioned on action labels and duration only. In contrast, we generate motion conditioned on observations of arbitrary length, including none. To solve this generalized problem, we propose PoseGPT, an auto-regressive transformer-based approach which internally compresses human motion into quantized latent sequences. An auto-encoder first maps human motion to latent index sequences in a discrete space, and vice-versa. Inspired by the Generative Pretrained Transformer (GPT), we propose to train a GPT-like model for next-index prediction in that space; this allows PoseGPT to output distributions on possible futures, with or without conditioning on past motion. The discrete and compressed nature of the latent space allows the GPT-like model to focus on long-range signal, as it removes low-level redundancy in the input signal. Predicting discrete indices also alleviates the common pitfall of predicting averaged poses, a typical failure case when regressing continuous values, as the average of discrete targets is not a target itself. Our experimental results show that our proposed approach achieves state-of-the-art results on HumanAct12, a standard but small scale dataset, as
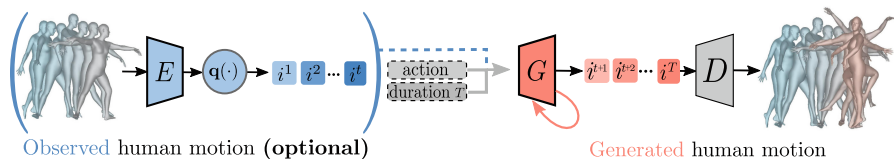
Observed human motion (**optional**)   Generated human motion

Fig. 1. Method Overview. PoseGPT generates a human motion sequence, conditioned on an action label, a duration T, and optionally on an observed past human motion. A GPT-like [52] model G sequentially predicts discrete latent indices, which are decoded using a decoder D into a generated human motion. When conditioning also on past human motion, the input human motion is encoded with E and quantized using $q(.)$ into the discrete latent space.

T. Lucas and F. Baradel—Equal contribution.

---

well as on BABEL, a recent large scale MoCap dataset, and on GRAB, a human-object interactions dataset.

1   Introduction

Generating realistic and controllable human motion is still an open research question despite decades of efforts in this domain [5,6]. In this work, we tackle the task of action-conditioned generation of realistic human motion sequences of varying length, with or without observation of past motion. Most of the effort in human motion synthesis has been focused on future motion prediction, typically conditioned on a sequence of past frames [4,9,29,69,70]; however, this requirement is a limiting constraint. In particular applications to virtual reality or character control [31,59] ideally should not require real world observations. And indeed, recent works [27,49] have shown that deep models can handle the highly multi-modal nature of human motion sequences, without conditioning on the past to narrow it down. Nevertheless, many possible applications of human motion modeling do require conditioning. In particular, vision based human-robot interactions may require robots to observe humans and predict likely future movements to successfully avoid them or interact with them. Therefore, we propose a class of models flexible enough to approach the more general problem of motion generation conditioned on observations of *arbitrary length*, including none.

Auto-regressive generative models [45,46] are natural candidates to handle this task. By factorizing distributions over the time dimension, they can be conditioned on past sequences of arbitrary length. However when applied to human motion sequences their potential is limited in at least two ways by the nature of the data. First, they are costly and inefficient to train on data captured at high frame rates, *e.g.* 30 frames per second (fps), in particular when using state-of-the-art transformer architectures. Second, long-term future is highly multi-modal; in a continuous target space this leads to average unrealistic predictions and, in turn, to error drift when sampling from auto-regressive models. Indeed, related previous works that have proposed auto-regressive approaches (based on LSTMs [22] and GRUs [43]), have shown that they are subject to error drift and prone to regress unrealistic average poses.

Therefore, we propose to compress human motion into a space that is lower dimensional and *discrete*, to reduce input redundancy. This allows training an auto-regressive model using discrete targets rather than to regress in a continuous space, such that the average of targets is not a valid output itself. We propose an auto-encoder transformer-based network which maps the human motion to a low dimensional space, discretized using a quantization bottleneck [47], and vice versa. Importantly, we ensure that the causal structure of the time dimension is kept in the latent representations such that it *respects the arrow of the time* (i.e. only the past influences the present). To do so we rely on causal attention in the encoder. This is crucial to enable conditioning of our model on observed past motions of arbitrary length, unlike in [49].

Then, we employ an auto-regressive GPT-like model to capture human motion directly in the learned discrete space. Transformer models have become the de-facto architecture for language tasks [51,52,65] and are increasingly adopted in computer vision [15,20]. This requires adaptations to deal with continuous and locally redundant data, which is not well suited to the quadratic computational cost induced by the lack of inductive prior in transformers. The input data used in this work falls into this category: we employ parametric 3D models [40,48] which represent human motion as a sequence of human 3D meshes, a continuous, high-dimensional and redundant representation. Our proposed discretization of the human motion alleviates the need for the auto-regressive model to capture low-level signal and enables it to concentrate on long-range relations. Indeed, while the space of human body model parameters [48] is high-dimensional and sparse – random samples are unlikely to be realistic – the quantization step concentrates useful regions into a finite set of points. In particular, random sequences in that space produce locally realistic sequences that lack temporal coherence. The GPT-like component of our method, called PoseGPT, is trained to predict a distribution over the next index in the discrete space. This allows probabilistic modeling of possible futures, with or without conditioning on past motion.

Motion capture (MoCap) datasets with action labels are costly to create [32,62]. We have been able to learn models from several orders of magnitude more data than prior art [27,49], owing to the recent availability of the BABEL [50] dataset, and also relying on the smaller HumanAct12 for fair comparison with previous works. In addition, we propose an evaluation protocol which we believe aggregates the best practices from prior art [49] and from the generative image modeling literature [8,41,44,57]. It is based on three principles; first, sample quality is evaluated using metrics based on classifiers, inspired from the GAN literature. Second, we strive to account for over-fitting together with sample quality. Indeed, sample quality metrics typically compare synthetic data to train data, without employing a validation set, which rewards over-fitting. While this is harmless when working with plentiful and complex data that deep models are unlikely to over-fit, we show that is not the case with small human motion datasets such as HumanAct12. Finally, we report likelihood based metrics to evaluate mode coverage. Indeed, while it is notoriously difficult to measure diversity from samples alone [44,57], that is in principle not the case for models that allow likelihood computations on test data. Using these principles we show that our proposed approach outperforms existing ones while being more flexible.

2   Related Work

Human Motion Forecasting. Predicting future human poses given a past motion is a topic of interest in human motion analysis [5,6]. The first successful methods were based on statistical models [11,23], with most recent work relying on deep learning based methods [26,34]. In particular image generation methods such as GANs [26] and VAEs [34] have been extended to human motion forecasting [9,28,39]. In DLow [69], a pretrained model is employed to enforce diversity in the predicted motions. In Cao *et al.* [14], the scene context is also

taken into account to predict future human motion. However they both show limitations when it comes to predicting long-term future horizons; in particular they tend to predict average poses, which is a known issue for methods trained by predicting continuous values [25]. In contrast, we propose a method able to predict future motion without error drift by quantizing motions.

Human Motion Synthesis. The task of human motion synthesis, given a class query, was first tackled with a focus on simple and cyclic human actions such as walking [61,63]. More recently, a lot of focus has been devoted to human pose and motion generation conditioned on a rich query representation such as a short textual description [2,3,19,24,38,39] or an audio representation such as music [36,37]. Class labels can be seen as a coarse case of textual descriptions; they bring less information about the motion than detailed descriptions, but are simpler to acquire and use. A few recent propositions have tackled 3D human motion generation given action classes [27,49,54], and in particular ACTOR [49] shows impressive results at generating human motion for non-periodic actions. However only small scale-datasets were available at the time [27,73,74]. The generated human motions are always front view, and the trajectories in the training data lack diversity. In [54], action sequences are modeled by conditioning predictions at each time frame on the last; this performs well for short sequences but does not allow conditioning on observations of arbitrary length. To go beyond these limitations, we develop a method trained on large-scale datasets, with long-tailed class distributions such as the recently released BABEL [50]. Our method can optionally be conditioned on past observations of arbitrary length, and obtains state-of-the-art performance. Most similar to ours, the concurrent work in [58] also relies on a quantization step and a GPT model for successfully learning dance motion conditioned on music.

Pose Representation. Human body representations are often expressed as skeleton representations, where a known kinematic structure is available. Most work in human modeling, ranging from human pose estimation [1,56,67,71] to human pose modeling [25,30], have used this type of representations for a while. However recent works are moving toward 3D body shape models [7,35,40,48] which are more realistic and enable more powerful applications such as augmented and virtual reality. Representing the 3D human body, and in particular the pose, is not straightforward. One can express a human pose as a set of 3D joint locations in the Euclidean space or as a set of bone angles encoding the rotations necessary to obtain the pose. However the lack of continuity in the space of rotation representations is a commonly observed issue [12,72] for deep learning methods. There has not been convergence towards a unified human pose representation format so far. In this work, we do not explicitly enforce any human pose representation but rather propose a model that can learn to embed and quantize any representation to a discrete latent space learned by the model.

Generative Modeling. Deep generative models can be broadly classified in two categories: maximum-likelihood based models, trained to maximize the like-
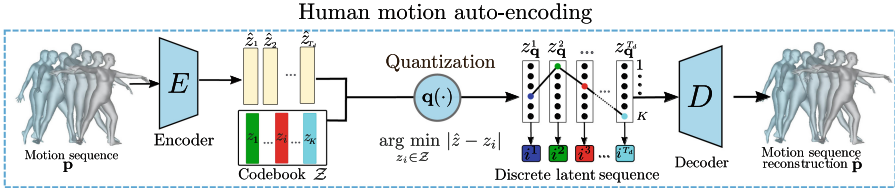
Human motion auto-encoding

Fig. 2. Discrete latent representation for human motion. The encoder E maps a human motion $\boldsymbol{p}$ to a latent representation $\hat{z}$ which is then quantized using a codebook $\mathcal{Z}$. The decoder D reconstructs the human motion $\hat{p}$ from the quantized latent sequence $z_{\mathbf{q}}$.

lihood of generating training data, and adversarial models [26] trained to maximize the quality of generated images as evaluated by a discriminator model. In the maximum-likelihood based literature, which is most relevant to our work, there are two dominant paradigms to handle the highly multi-modal nature of perceptual data. The first family is that of variational auto-encoders (VAEs) [34,55], which relies on an encoder, and the second that of autoregressive models [16,46] which relies on the chain rule decomposition of high-dimensional data. Both paradigms are leveraged in our work: in the first stage of our approach we adapt a flavour of auto-encoders called VQVAEs [64], which uses quantized latent variables, to our problem. In the second stage, we train a transformer based auto-regressive model to sequentially predict discrete latent sequences. Similar recipes have been applied to high-resolution image generation in [18,21,53], to video prediction [66,68] and to speech modeling in [17]. Note that while GANs generally display an impressive aptitude to generate high quality samples [13], they are not well suited to the task of human future pose/motion prediction. Indeed, they suffer from mode-collapse [57], *i.e.*, the inability to cover the full variability of the training data. This ability is critical for example for applications such as human-robot interactions where likely modes of the distribution of possible futures must not be ignored. Thus, this class of models is not a good candidate on its own.

3 The PoseGPT Model

In this section we describe PoseGPT, our proposed approach for generative modeling of human pose sequences. First, we present how we compress human motion to a discrete space, and reconstruct motion from it (Sect. 3.1). Second, we introduce a GPT-like model trained for next-index probabilistic prediction in that space (Sect. 3.2).

3.1 Learning a Discrete Latent Space Representation

Human actions defined by body-motions can be characterized by the rotations of body parts, disentangled from the body shape. This allows the generation
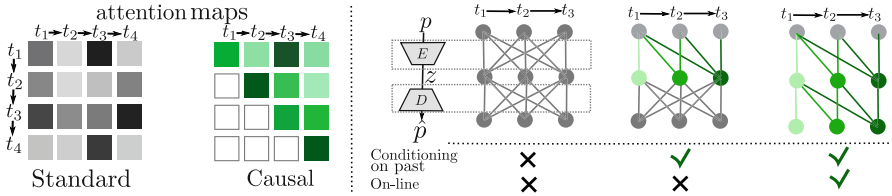
Fig. 3. Conditioning on past with causal attention. Masking attention maps in the encoder leads to models that can be conditioned on past observations. Masking the attention maps in the decoder as well allows models that can make on-line predictions.

of motions with actors of different morphology. For this, we rely on parametric differential body models – SMPL [40] and SMPL-X [48] – which disentangle body parts rotations from body shape; a human motion $\boldsymbol{p}$ of length T is represented as a sequence of body poses and translations of the root joints: $\boldsymbol{p} = \{(\theta_1, \delta_1), \ldots, (\theta_T, \delta_T)\}$ where θ and δ represent the body pose and the translation respectively. We use an encoder E and a quantization operator $\mathbf{q}$ to encode pose sequences and a decoder D to reconstruct $\hat{\boldsymbol{p}} = D(\mathbf{q}(E(\boldsymbol{p})))$. We use causal attention mechanisms to maintain a temporally coherent latent space and neural discrete representation learning [47] for quantization. An overview of the training procedure is shown in Fig. 2.

Causal Latent Space. The encoder first represents human motion sequences as a latent sequence representation $\hat{\boldsymbol{z}} = \{\hat{z}^1, \ldots, \hat{z}^{T_d}\} = E(\boldsymbol{p})$ where $T_d \leq T$ is the temporal dimension of the latent sequence. By default, we require that our latent representation respects the arrow of time, i.e., that for any $t \leq T_d$, $\{\hat{z}_1, \ldots, \hat{z}_t\}$ depends only on $\{p_1, \ldots, p_{\lfloor t \cdot T/T_d \rfloor}\}$; such as illustrated in Fig. 3. For this, we rely on transformers with causal attention; it avoids any inductive prior besides causality, by modeling interactions between all inputs using self-attention [65], modified to respect the arrow of time. Intermediate representations are mapped using three feature-wise linear projections, into query $Q \in \mathbb{R}^{N \times d_k}$, key $K \in \mathbb{R}^{N \times d_k}$ and value $V \in \mathbb{R}^{N \times d_v}$; in addition, a causal mask is defined as $C_{i,j} = -\infty \cdot [\![i > j]\!] + [\![i \leq j]\!]$, and the output is computed as:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top \cdot C}{\sqrt{d_k}}\right) V \in \mathbb{R}^{N \times d_v}. \tag{1}$$

The causal mask ensures that all entries below the diagonal of the attention matrix do not contribute to the final output and thus that the arrow of time is respected. This is crucial to allow conditioning on past observations when sampling from the model: if latent variables depend on the full sequence, they are impossible to compute from past observations alone.

Quantizing the Latent Space. To build an efficient latent representation of human motion sequences, we then rely on a discrete codebook of learned temporal representations; more precisely a latent space sequence $\hat{\boldsymbol{z}} \in \mathbb{R}^{T_d \times n_z}$
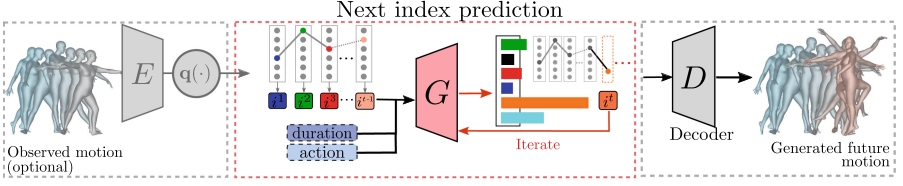
Fig. 4. Future motion prediction. In the discrete latent space, an auto-regressive transformer model G predicts the next latent index given previous ones. We condition on a human action label, a sequence duration and optionally on an observed motion.

is mapped to a sequence of codebook entries $z_{\mathbf{q}} \in \mathcal{Z}^{T_d}$, where $\mathcal{Z}$ is a set of C codes of dimension n_z. Equivalently, this can be summarized as a sequence of T_d indices corresponding to the code entries in the codebook. A given sequence $\boldsymbol{p}$ is approximately reconstructed by $\hat{\boldsymbol{p}} = D(z_{\mathbf{q}})$ where $z_{\mathbf{q}}$ is obtained by encoding $\hat{z} = E(x) \in \mathbb{R}^{T_d \times n_z}$ and mapping each temporal element of this tensor with $\mathbf{q}(\cdot)$ to its closest codebook entry z_k:

$$z_{\mathbf{q}} = \mathbf{q}(\hat{z}) := \left(\arg\min_{z_k \in \mathcal{Z}} \|\hat{z}_t - z_k\| \right) \in \mathbb{R}^{T_d \times n_z} \qquad (2)$$

$$\hat{\boldsymbol{p}} = D(z_{\mathbf{q}}) = D\left(\mathbf{q}(E(\boldsymbol{p}))\right). \qquad (3)$$

Equation (3) is non differentiable; the standard way to backpropagate through it is to rely on the straight-through gradient estimator, which during the backward pass simply approximates the quantization step as an identity function by copying the gradients from the decoder to the encoder [10]. Thus the encoder, decoder and codebook can be trained by optimizing:

$$\mathcal{L}_{\mathrm{VQ}}(E, D, \mathcal{Z}) = \|\boldsymbol{p} - \hat{\boldsymbol{p}}\|^2 + \|\mathrm{sg}[E(\boldsymbol{p})] - z_{\mathbf{q}}\|_2^2 + \beta \|\mathrm{sg}[z_{\mathbf{q}}] - E(\boldsymbol{p})\|_2^2, \qquad (4)$$

with $\mathrm{sg}[\cdot]$ the stop-gradient operator. The term $\|\mathrm{sg}[z_{\mathbf{q}}] - E(\boldsymbol{p})\|_2^2$, dubbed the "commitment loss" [47], has been shown necessary to stable training.

Product Quantization. To increase the flexibility of the discrete representations learned by the encoder E, we propose using product quantization [33]: each element $\hat{z}_i \in \mathbb{R}^{n_z}$ in the sequence of latent representation is cut into K chunks $(\hat{z}_i^1, \ldots, \hat{z}_i^K) \in \mathbb{R}^{n_z/K \times K}$, and each chunk is discretized separately using K different codebooks $\{\mathcal{Z}_1, \ldots, \mathcal{Z}_K\}$. The size of the discrete space learned increases exponentially with K, for a total of $C^{T_d \cdot K}$ combinations. We empirically validate the utility of using product quantization in our experiments. Instead of one index target per time step, product quantization produces K targets. To capture relations between them, we propose a prediction head that models the K factors sequentially rather than in parallel, called 'auto-regressive' head and evaluated in Sect. 3.2; see the supplementary material for more details.
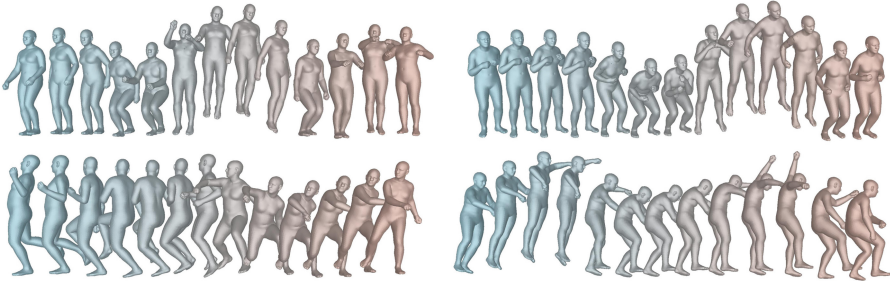
Fig. 5. Samples generated from scratch. Samples generated without any observed motion for the action labels 'jumping' (top) and for the action 'dancing' (bottom). Note: Times flows from left to right (*i.e.*, the blue texture corresponds to the first frame and the red texture to the last frame).

3.2 Learning a Density Model in the Discrete Latent Space

The latent representation $z_\mathbf{q} = \mathbf{q}(E(\boldsymbol{p})) \in \mathbb{R}^{T_d \times n_z}$ produced by composing the encoder E and the quantization operator $\mathbf{q}(\cdot)$ can be represented as the sequence of codebook indices of the encodings, $\boldsymbol{i} \in \{0, \ldots, |\mathcal{Z}| - 1\}^{T_d}$, by replacing each code by its index in the codebook $\mathcal{Z}$, *i.e.*, $i_t = k$ such that $(z_\mathbf{q})_t = z_k$. Indices of $\boldsymbol{i}$ can be mapped back to the corresponding codebook entries and decoded to a sequence $\hat{\boldsymbol{p}} = D(z_{i_1}, \ldots, z_{i_{T_d}})$.

Learning to Predict Next Pose Index. As a second step to our method, we propose to learn a prior distribution over learned latent code sequences. A motion sequence $\boldsymbol{p}$ of the human action a is encoded into $(i_t)_{1..T_d}$. We then formulate the problem of latent sequence generation as auto-regressive index prediction; for this we keep the natural temporal ordering, which can be interpreted as time due to the use of causal attention in the encoder. We train a transformer model [65] denoted G – well suited to discrete sequential data – using maximum-likelihood estimation, similar in spirit to GPT [52].

Given $\boldsymbol{i}_{<j}$, the action a and the sequence length T, the transformer outputs a softmax distribution over the next indices, *i.e.*, $p_G(i_j|\boldsymbol{i}_{<j}, a, T)$, the likelihood of the latent sequence is $p_G(\boldsymbol{i}) = \prod_j p_G(\mathbf{i}_j|\mathbf{i}_{<j}, a, T)$ and the model is trained to minimize:

$$\mathcal{L}_{\text{GPT}} = \mathbb{E}_{\boldsymbol{i}} \left[-\sum_j \log p_G(i_j|\boldsymbol{i}_{<j}, a, T) \right]. \tag{5}$$

An overview of the training procedure is shown in Fig. 4 and, in the supplementary material, we discuss different input sequence embeddings for processing by the GPT.

Sampling Human Motion. Human motion is generated sequentially by sampling from $p(s_i|\boldsymbol{s}_{<i}, a, T)$ to obtain a sequence of pose indices $\tilde{\boldsymbol{z}}$ given an action
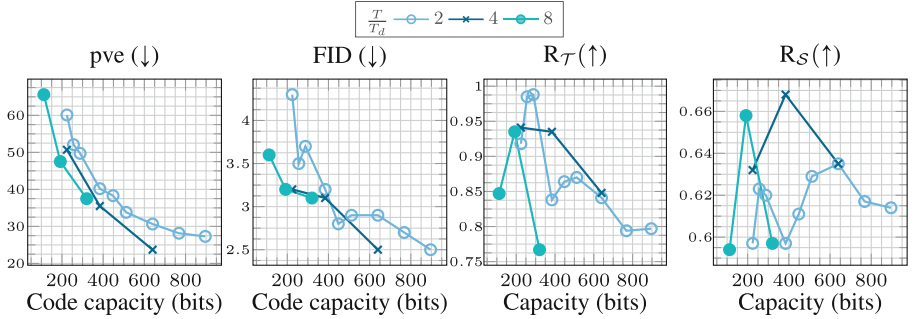
Fig. 6. Latent space design. We define models for $T/T_d \in \{2, 4, 8\}$ by varying K and C and present results as a function of the capacity of latent sequence.

and sequence length, and decoding it into a sequence of pose $\tilde{\boldsymbol{p}} = D(\tilde{\boldsymbol{z}})$ (see Fig. 5 for samples).

4   Experiments

We experiment with two parametric 3D models: SMPL [40] for comparison to state-of-the-art approaches, and SMPL-X [48] to enable control of the face and hands. We now present the three datasets considered for evaluation; architectural and implementation details are in the supplementary material.

HumanAct12 allows comparison to prior art [27,49], but its small size and the absence of train/val/test splits are limiting. It contains 1191 videos and SMPL pose parameters, 12 action classes and a single action per video. The poses, automatically optimized from estimated 3D joints, are noisier than annotations from capture environments.

BABEL [50] is a subset of AMASS [42], a large collection of MoCap data captured in controlled environments for high quality annotations. It contains 28K sequences (43 hours of motion in total); sequence length varies from 3 s to several minutes and there are 120 manually annotated human actions in total. The action distribution is very long-tailed so we use only the 60 most common actions as proposed by the authors. In short, BABEL is over 40 times bigger than HumanAct12, has a train/val/test split, no noise in the SMPL parameters and a rich variety of human actions; we believe this makes it a dataset of choice to move forward.

GRAB [60] contains whole-body SMPL-X of people grasping objects, with 11 persons performing 29 motions with 51 different rigid objects, for a total of 1500 sequences of 8 seconds on average, with 7 persons for training and 2 for testing.

4.1   Evaluation Metrics

Generative models can be evaluated through generated data; a perfect set of samples contains data that is *as realistic* and *as diverse* as real *unseen* test

data. These aspects are not always trivial to quantify, and we now discuss how they are measured in practice.

Sample Quality Evaluation. The dominant approach [27, 49] to measure sample quality relies on pretrained classifiers. In particular the Frechet Inception Distance (FID), which we report, measures a distance between distributions of classifier features obtained from a set of samples D_{samples} and real data. Following [49], we also rely on a classifier $\mathcal{T}$ pre-trained on train data and report the ratio between accuracies on sampled and test data:

$$R_\mathcal{T}(D_{\text{samples}}, D_{\text{test}}) = \frac{|D_{\text{samples}}|}{|D_{\text{test}}|} \cdot \frac{\sum_{x \in D_{\text{test}}} \text{acc}_\mathcal{T}(x)}{\sum_{x \in D_{\text{samples}}} \text{acc}_\mathcal{T}(x)}. \tag{6}$$

This metric is not sensitive to diversity – the model can drop modes as long as the rest is very well classified. The ratio normalizes values that otherwise depend on choices orthogonal to sample quality; we refer to the supplementary material for details on the action classifier.

Diversity Evaluation. First, we evaluate sample diversity by training a classifier $\mathcal{S}$ on *samples* and evaluating it on unseen *test data*, following [57]. Intuitively, for $\mathcal{S}$ to perform as well as $\mathcal{T}$, samples need to be as diverse and as realistic as real data; we measure it with:

$$R_\mathcal{S}(D_{\text{test}}) = \sum_{x \in D_{\text{test}}} \frac{\text{acc}_\mathcal{S}(x)}{\text{acc}_\mathcal{T}(x)}. \tag{7}$$

This metric is sensitive to diversity as real data modalities not captured by the generator will not be seen by $\mathcal{S}$ and misclassified, but not by $\mathcal{T}$, which will degrade the ratio. The pair $(R_\mathcal{S}, R_\mathcal{T})$ is best considered together [57]: if $R_\mathcal{S}$ is close to one, we consider sample quality to be high, and gains in $R_\mathcal{T}$ can be attributed to diversity [44]. Note that $\mathcal{S}$ and $\mathcal{T}$ have the same architecture and are trained with the same hyper parameters. More classically, we also report likelihood based metrics; dropped modes will lead to data points with very low likelihood, so they are sensitive to mode coverage [8]. In particular, we report the test reconstruction error of the auto-encoder using the Per-Vertex Error (pve), and the test likelihood of the GPT on encoded test sequences. As these metrics do not guarantee realistic samples, we consider them together with classifier based quality metrics.

Over-Fitting. Sample quality metrics typically used – standard FID or classification accuracy [49] – measure differences between train data and generated data, without involving a test set. This does not account for over-fitting and rewards models that perfectly copy train data: on small datasets, all metrics will monotonically improve with model capacity. To remedy this, we keep unseen data on BABEL and compute the FID, $R_\mathcal{S}$ ratio and maximum-likelihood based metrics using that test data. Our only metric not sensitive to over-fitting is $R_\mathcal{T}$; we rely on the others to detect over-fitting.

Table 1. Impact of latent space capacity on BABEL for $T/T_d = 2$ (left) and for $C = 256$ (right). **bold** denotes best in column (across K); <u>underlined</u> denotes best in row (across C).

K (nb. codebooks)	pve↓	R_T↑	R_S↑	FID↓	pve↓	R_T↑	R_S↑	FID↓	pve↓	R_T↑	R_S↑	FID↓
		$C = 128$				$C = 256$				$C = 512$		
1	60.1	**0.91**	0.60	4.3	52.1	**0.98**	0.61	<u>3.5</u>	49.7	**0.99**	0.62	3.7
2	40.2	0.84	0.60	3.2	38.3	0.86	0.61	<u>2.8</u>	33.8	<u>0.87</u>	**0.63**	2.9
4	30.6	<u>0.84</u>	**<u>0.64</u>**	**2.9**	28.2	0.79	<u>0.62</u>	**2.7**	27.3	0.80	0.61	**<u>2.5</u>**
8	27.2	**0.48**	**0.49**	<u>3.0</u>	23.7	0.49	0.46	4.4	26.7	0.41	0.47	4.3

$(T/T_d = 2)$

K (nb. codebooks)	pve↓	R_T↑	R_S↑	FID↓	pve↓	R_T↑	R_S↑	FID↓
		$T/T_d = 4$				$T/T_d = 8$		
2	50.7	**0.94**	0.63	3.2	47.5	**0.93**	**0.66**	3.2
4	35.5	0.93	**0.67**	3.1	37.5	0.77	0.60	**3.1**
8	28.8	0.64	0.54	**2.5**	65.6	0.85	0.59	3.6

$(C = 256)$

4.2 Ablative Study of Design Choices

We now ablate the main design choices made in PoseGPT. The first is the design of the discrete latent space, in particular the quantization bottleneck and its capacity. The second regards the GPT component, trained for next index prediction; in particular we ablate the choice of input embedding method and prediction head. Finally, we evaluate the impact of using causal attention in the auto-encoder. Note that as there is no test split on HumanAct12; because it is too small to define one of reasonable size without severely degrading performance, we compute the FID using train data on this dataset.

Latent Sequence Space Design. The main design choice regarding the latent sequence space is the quantization bottleneck. We now study the impact of its capacity, mostly controlled by T_d (latent sequence length), K (nb. of product quantization factor) and C (total number of centroids). More capacity yields lower reconstruction errors at the cost of less compressed representations. In our case, that means more indices to predict for the GPT, which impacts sampling, and we now explore this trade-off.

In Table 1 (left), models trained on BABEL show that as expected, pve goes down monotonously with both K and C, but not the R_S and R_T ratios, as also shown in Fig. 6. Models with $K = 1$ obtain high sample classification accuracy but poor reconstruction on test data and lower R_S; this suggests insufficient

Table 2. Latent space design on HumanAct12. **Bold** denotes best value.

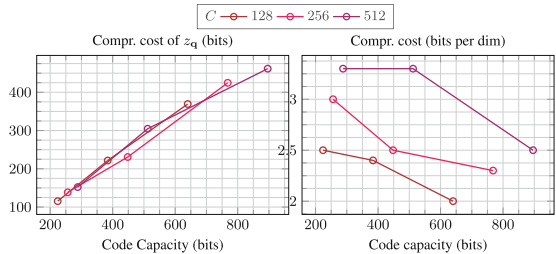| K (nb. codebooks)	FID ↓	R_S (↑)	FID ↓	R_S (↑)
	$C = 256$		$C = 512$	
8	0.12	93.8	0.11	93.7
16	0.11	94.5	0.10	94.9
32	**0.09**	**95.1**	**0.08**	**95.2**

Fig. 7. Cost of compressing z_q using the GPT, in bits and bits per dimension.

Table 3. GPT design on BABEL and Human-Act12. **Bold** text denotes best value. ar denotes auto-regressive.

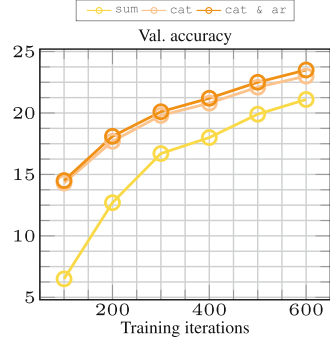| t^0: extra token at $t=0$. +: sum at all t. :: concat at all t.					
Input tokens		Prediction head		BABEL	HumanAct12
a	T	mlp	ar	R_S(↑) R_T(↑) gpt-acc.(↑)	FID(↓) R_T(↑)
t^0	×	×	×	0.65 0.51 22.4 (20.8)	0.56 0.74
+	×	×	×	0.79 0.55 23.1 (16.3)	0.19 0.91
+	+	×	×	0.79 0.57 23.9 (16.2)	**0.10** **0.94**
::	::	×	×	**0.86** **0.61** 24.9 (22.9)	0.22 0.86
+	+	✓	×	- - -	0.09 95.1
::	::	✓	×	0.86 0.61 25.1 (22.7)	- -
::	::	✓	✓	**0.98** **0.64** **25.9** (22.6)	- -

Fig. 8. Index pred. accuracy using concatenation *vs.*summation.

capacity to capture the full diversity of the data. On the other hand, models with the most capacity (*e.g.*, $K = 8$) yield sub-par performance. The best trade-offs are achieved with $(K, C) \in \{(2, 256), (2, 512), (4, 128), (4, 256)\}$. The table on the right shows that the model can handle decreased temporal resolution. Note that using $K = 8$ works better at coarser resolutions, as it compensates for the loss of information. In Table 2, all metrics improve monotonically with K and C; this is expected as over-fitting is not factored out by the metrics and the dataset is small enough to over-fit. Finally, in Fig. 7, we report the cost of compressing $z_\mathbf{q}$ using the GPT model. We observe that the absolute compression cost in bits (left) increases, *i.e.*, $z_\mathbf{q}$ contains more information, while the cost per dimension decreases: each sequence index is easier to predict individually.
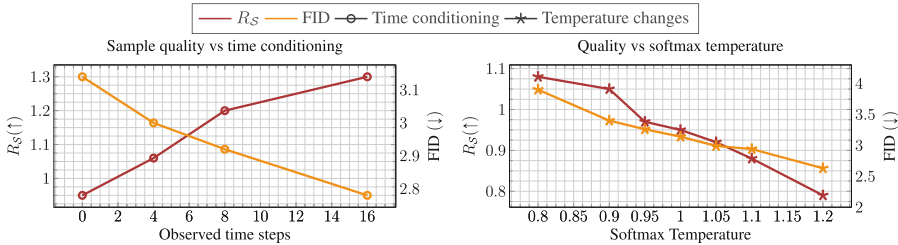

Fig. 9. Sample quality with our best model for different amounts of observed motion, and different temperatures, measure with the FID and R_S metrics.

Ablations on Next Index Prediction. We now study two design choices made in the GPT component of PoseGPT: the choices of input embedding and predictions head. Using the proper input embedding has a strong impact on the

performance of transformer architectures [52], and depends on the input data. In Table 3, we study this impact when working with human motion. For this ablation, we fix the latent sequence space, *i.e.*, the auto-encoder hyper-parameters and weights, and train our GPT model with different input embeddings. We measure sample quality, and the accuracy of the GPT model at predicting discrete sequence indices. Note that this accuracy is directly comparable across models, as the latent space is frozen and identical.

In the first row, we observe that embedding the action at each timestep, rather than as an extra transformer input, has significant positive impact. Conditioning on the sequence length is also beneficial (Row 3 *vs.*Row 4); this is expected, as it relieves the model from having to predict when to stop generating new poses. As an added benefit, it also allows extra control at inference time. We also see that a concatenation of the embedded information, followed by a linear projection – which can be seen as a learned weighted sum – is better than simple summation on BABEL. On the other hand this extra model capacity is not beneficial on HumanAct12, which may be due to the size of the dataset. In Fig. 8, we also observe that models using concatenation rather than summation train significantly faster.

Having determined the best input configuration for both datasets, we further experiment with more expressive output layers for the model; we show that having a MLP head rather than a single fully-connected layer is beneficial, and we obtain further gains using an auto-regressive layer (see Sect. 3.2). This can be explained by the fact that with product quantization, several codebook indices are extracted simultaneously from a single input vector, but are not independent, and using an MLP and/or an auto-regressive layer better captures the correlations between them.

Causal Attention. In Table 4, we study the impact of using causal attention in the auto-encoder, for $K \in \{2, 4\}$ and $C = 256$. Causal attention is a restriction on model flexibility, as it limits the inputs used by features in the encoder. Empirically, we observe that adding causal attention indeed degrades performance. Adding it to the encoder, which is mandatory to create a model that can be conditioned on past observations, causes only a mild degradation. Adding it in the decoder as well allows to run the model on-line, *i.e.*, make observations and predictions in parallel, but strongly degrades performance.

Conditioning and Temperature. Conditioning the model on past observation is expected to improve the quality of generated samples. In Fig. 9 (left), we see that indeed both $R_{\mathcal{S}}$ and the FID improve monotonically as the length of observations increases. In the right plot, we see that increasing or decreasing the softmax temperature leads to a trade-off between the two metrics; this behaviour can be expected: decreasing the temperature improves sample quality by concentrating the mass on major modes of the distribution, and thus increases mode-dropping.

Table 4. Impact of causal attention in the encoder and decoder for $C = 256$.

K (nb. codebooks)	Causal E	Causal D	pve ($\downarrow$)
2	×	×	**35.7**
2	✓	×	38.3
2	✓	✓	50.2
4	×	×	**25.3**
4	✓	×	28.2
4	✓	✓	38.6
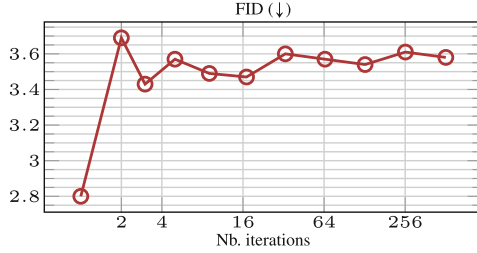
Fig. 10. Evaluation of error drift. Model iteratively conditioned on last predictions made.

Error-Drift in Long-Term Horizon Generation. In Fig. 10, we study the robustness of PoseGPT to error drift, a typical failure case of models that make auto-regressive predictions in continuous space. To this end, we sample from our model several times consecutively, by conditioning the last pose generated by the model. To initiate this process, the first motion is generated without temporal conditioning. Empirically, we observe that in this setting, PoseGPT is robust to long-term error drift: the FID initially degrades but remains stable even when we repeat the generation process many times.

4.3   Comparison to the State of the Art

In Table 5, we compare PoseGPT against the state-of-the-art results. For fair comparison, these metrics are computed without conditioning on past observations. We find that PoseGPT outperforms the state-of-the-art method, namely ACTOR [49], when looking at the FID metric with a relative gain of 33% (0.12 *vs*.0.08) on HumanAct12 and over 50% on BABEL. The performance in diversity and the multimodality indicates that PoseGPT covers the human motion distribution of this dataset. On BABEL, the gains are around 50% in terms of both FID and classification accuracy. The gains in classification accuracy indicate both higher quality samples, and a richer distribution.

Table 5. State-of-the-art comparison. On HumanAct12 (left), PoseGPT obtains better FID and comparable classification accuracy. On BABEL (center) and on GRAB (right), PoseGPT obtains substantial gains for all metrics. [*] means trained by us based on official code. Note that the FID of real data is not 0 due to data augmentations. For consistency with [27,49], we report diversity and multimodality metrics on Human-Act12; these metrics are considered good when close to the values obtained on real data.

Model	FID$\downarrow$	R_T (%).$\uparrow$	Div.	Multimod.
Real	0.02	99.4	6.86	2.60
Action2Motion [27]	2.46	92.3	7.03	2.87
ACTOR [49]	0.12	95.5	6.84	2.53
PoseGPT	**0.08**	**95.8**	6.85	2.82

Model	action cond.	length cond.	future pred.	$R_S(\uparrow)$	FID$\downarrow$	$R_T(\uparrow)$
Real	-	-	-	1.0	0.01	1.0
ACTOR[*]	✓	✓	X	0.35	9.5	0.56
PoseGPT	✓	✓	✓	**0.64**	**2.7**	**0.98**

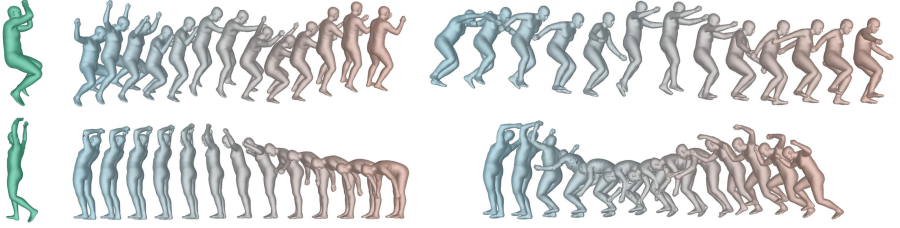| Model	FID $\downarrow$	$Acc_S(D)\uparrow$	$Acc_D(S)\uparrow$
Real	0.01	0.99	-
ACTOR[*]	20.7	0.20	-
PoseGPT	**5.1**	**0.86**	-

Fig. 11. Samples conditioned on past observation. On the left in green, we show an observed initial pose, then we sample two different future human motions that we show side by side. The top row corresponds to the human action 'jumping' and the bottom row is sampled from the human action 'stretching'.
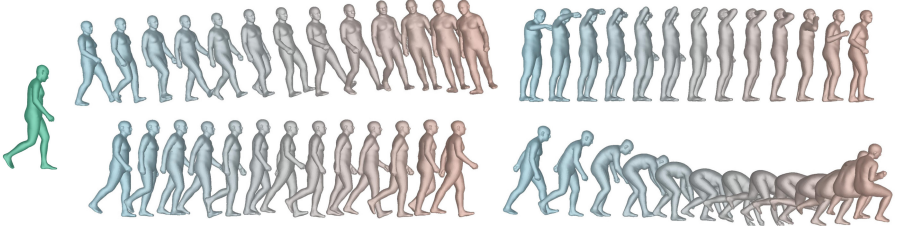

Fig. 12. Samples conditioned on an initial pose and with four different actions. Given an initial pose shown in green, we generate four different human motions conditioned on four different actions. What are these actions?[1]

Qualitative Examples. Finally, we show samples of human motions generated by PoseGPT. In Fig. 5, we show samples of human motion generated by conditioning on a human action only. We observe that human motions are realistic and diverse for both actions. Then in Fig. 11, we display two possible future motions given an initial pose and an action. The generated human motions are diverse which demonstrates that PoseGPT is able to handle the multimodal nature of the future. Finally in Fig. 12, given a initial pose, we generate four human motions with four different actions[1]. This demonstrates that the action information is taken into account and impacts the human motion generation. We provide more visualizations in the supplementary material.

5 Conclusion

This work introduces PoseGPT, an auto-regressive transformer-based approach which quantizes human motion into latent sequences. Given a human action, a duration and an arbitrarily long past observation, it outputs realistic and diverse 3D human motions. We provide quantitative and qualitative experiments to show the strengths of our proposed method. In particular, ablations demonstrate that

---

[1] From left to right and top to bottom: 'turning', 'touching face', 'walking', 'sitting'.

quantization is a key component, and we study each part of our approach in detail. PoseGPT reaches state-of-the-art performance on three different benchmarks and is able to generate human motions given an action label, conditioned on observed past motion of arbitrary length.

References

1. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 44–58 (2005)
2. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2Action: generative adversarial synthesis from language to action. In: ICRA, pp. 5915–5920 (2018)
3. Ahuja, C., Morency, L.: Language2Pose: natural language grounded pose forecasting. In: 3DV, pp. 719–728 (2019)
4. Aksan, E., Kaufmann, M., Hilliges, O.: Structured prediction helps 3D human motion modelling. In: ICCV, pp. 7144–7153 (2019)
5. Badler, N.: Temporal scene analysis: conceptual descriptions of object movements. PhD thesis, University of Toronto (1975)
6. Badler, N.I., Phillips, C.B., Webber, B.L.: Simulating Humans: Computer Graphics Animation and Control. Oxford University Press, NY (1993)
7. Baradel, F., Groueix, T., Weinzaepfel, P., Brégier, R., Kalantidis, Y., Rogez, G.: Leveraging mocap data for human mesh recovery. In: 3DV, pp. 586–595 (2021)
8. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
9. Barsoum, E., Kender, J., Liu, Z.: HP-GAN: probabilistic 3D human motion prediction via GAN. In: CVPRW, pp. 1418–1427 (2018)
10. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013)
11. Bowden, R.: Learning statistical models of human motion. In: CVPRW (2000)
12. Brégier, R.: Deep regression on manifolds: a 3D rotation case study. In: 3DV, pp. 166–174 (2021)
13. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019)
14. Cao, Z., Gao, H., Mangalam, K., Cai, Q.-Z., Vo, M., Malik, J.: Long-term human motion prediction with scene context. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 387–404. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_23
15. Chen, M., et al.: Generative pretraining from pixels. In: ICML, pp. 1691–1703 (2020)
16. Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: PixelSNAIL: an improved autoregressive generative model. In: ICML, pp. 864–872 (2018)
17. Chorowski, J., Weiss, R.J., Bengio, S., Van Den Oord, A.: Unsupervised speech representation learning using WaveNet autoencoders. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(12), 2041–2053 (2019)
18. De Fauw, J., Dieleman, S., Simonyan, K.: Hierarchical autoregressive image models with auxiliary decoders. arXiv preprint arXiv:1903.04933 (2019)
19. Delmas, G., Weinzaepfel, P., Lucas, T., Moreno-Noguer, F., Rogez, G.: PoseScript: 3D human poses from natural language. In: ECCV (2022)

20. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: ICLR (2021)
21. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR, pp. 12873–12883 (2021)
22. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: ICCV, pp. 4346–4354 (2015)
23. Galata, A., Johnson, N., Hogg, D.: Learning variable-length Markov models of behavior. Comput. Vis. Image Underst. **81**(3), 398–413 (2001)
24. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Synthesis of compositional animations from textual descriptions. In: CVPR, pp. 1396–1406 (2021)
25. Ghosh, P., Song, J., Aksan, E., Hilliges, O.: Learning human motion models for long-term predictions. In: 3DV, pp. 458–466 (2017)
26. Goodfellow, I., et al.: Generative adversarial nets. Commun. ACM **63**(11), 139–144 (2014)
27. Guo, C., et al.: Action2Motion: conditioned generation of 3D human motions. In: ACMMM, pp. 2021–2029 (2020)
28. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: CVPR, pp. 2255–2264 (2018)
29. Habibie, I., Holden, D., Schwarz, J., Yearsley, J., Komura, T.: A recurrent variational autoencoder for human motion synthesis. In: BMVC (2017)
30. Herda, L., Fua, P., Plankers, R., Boulic, R., Thalmann, D.: Skeleton-based motion capture for robust reconstruction of human motion. In: Proceedings Computer Animation 2000, pp. 77–83 (2000)
31. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. ACM Trans. Graph. **36**(4), 1–13 (2017)
32. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
33. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 117–128 (2010)
34. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
35. Kocabas, M., Athanasiou, N., Black, M.J.: VIBE: video inference for human body pose and shape estimation. In: CVPR, pp. 5253–5263 (2020)
36. Lee, H.Y., et al.: Dancing to music. Adv. Neural Inf. Process. Syst. **32** (2019)
37. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Learn to dance with AIST++: music conditioned 3D dance generation. arXiv preprint arXiv:2101.08779 (2021)
38. Lin, A.S., Wu, L., Rodolfo, C., Kevin Tai, Q.H.R.J.M.: Generating animated videos of human activities from natural language descriptions. In: Proceedings of the Visually Grounded Interaction and Language Workshop at NeurIPS (2018)
39. Lin, X., Amer, M.R.: Human motion modeling using DVGANs. arXiv preprint arXiv:1804.10652 (2018)
40. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graph. **34**(6), 1–16 (2015)
41. Lucas, T., Shmelkov, K., Alahari, K., Schmid, C., Verbeek, J.: Adaptive density estimation for generative models. Adv. Neural Inf. Process. Syst. **32** (2019)
42. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: archive of motion capture as surface shapes. In: ICCV, pp. 5442–5451 (2019)
43. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: CVPR, pp. 2891–2900 (2017)

44. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: ICML, pp. 7176–7185 (2020)
45. Van den Oord, A., et al.: Conditional image generation with PixelCNN decoders. Adv. Neural Inf. Process. Syst. **29** (2016)
46. van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: ICML, pp. 1747–1756 (2016)
47. van den Oord, A., Oriol, V., Kavukcuoglu, K.: Neural discrete representation learning. In: ICML (2018)
48. Pavlakos, G., et al.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR, pp. 10975–10985 (2019)
49. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. In: ICCV, pp. 10985–10995 (2021)
50. Punnakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: BABEL: bodies, action and behavior with English labels. In: CVPR, pp. 722–731 (2021)
51. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
52. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
53. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. Adv. Neural Inf. Process. Syst. **32** (2019)
54. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: HuMoR: 3D human motion model for robust pose estimation. ICCV, pp. 11488–11499 (2021)
55. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep, generative models. In: ICML, pp. 1278–1286 (2014)
56. Rogez, G., Weinzaepfel, P., Schmid, C.: LCR-Net++: multi-person 2D and 3D pose detection in natural images. IEEE Trans. Pattern Anal. Mach. Intell. **42**(5), 1146–1161 (2019)
57. Shmelkov, K., Schmid, C., Alahari, K.: How good is my GAN? In: ECCV, pp. 213–229 (2018)
58. Siyao, L., et al.: Bailando: 3D dance generation by actor-critic GPT with choreographic memory. In: CVPR, pp. 11050–11059 (2022)
59. Starke, S., Zhang, H., Komura, T., Saito, J.: Neural state machine for character-scene interactions. ACM Trans. Graph. **38**(6), 1–14 (2019)
60. Taheri, O., Ghorbani, N., Black, M.J., Tzionas, D.: GRAB: a dataset of whole-body human grasping of objects. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 581–600. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_34
61. Taylor, G.W., Hinton, G.E., Roweis, S.: Modeling human motion using binary latent variables. Adv. Neural Inf. Process. Syst. **19** (2006)
62. Carnegie Mellon University: CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/
63. Urtasun, R., Fleet, D.J., Lawrence, N.D.: Modeling human locomotion with topologically constrained latent variable models. In: Elgammal, A., Rosenhahn, B., Klette, R. (eds.) HuMo 2007. LNCS, vol. 4814, pp. 104–118. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75703-0_8
64. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Adv. Neural Inf. Process. Syst. **30** (2017)
65. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)

66. Walker, J., Razavi, A., Oord, A.V.D.: Predicting video with VQVAE. arXiv preprint arXiv:2103.01950 (2021)
67. Weinzaepfel, P., Brégier, R., Combaluzier, H., Leroy, V., Rogez, G.: DOPE: distillation of part experts for whole-body 3D pose estimation in the wild. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12371, pp. 380–397. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58574-7_23
68. Weissenborn, D., Täckström, O., Uszkoreit, J.: Scaling autoregressive video models. In: ICLR (2020)
69. Yuan, Y., Kitani, K.: DLow: diversifying latent flows for diverse human motion prediction. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12354, pp. 346–364. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58545-7_20
70. Zhang, Y., Black, M.J., Tang, S.: We are more than our joints: predicting how 3D bodies move. In: CVPR, pp. 3372–3382 (2021)
71. Zheng, C., et al.: Deep learning-based human pose estimation: a survey. arXiv preprint arXiv:2012.13392 (2020)
72. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR, pp. 5745–5753 (2019)
73. Zou, S., et al.: Polarization human shape and pose dataset. arXiv preprint arXiv:2004.14899 (2020)
74. Zou, S., et al.: 3D human shape reconstruction from a polarization image. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12359, pp. 351–368. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_21