Controlling Character Motions without Observable Driving Source

Weiyuan Li, Bin Dai, Ziyi Zhou, Qi Yao, Baoyuan Wang
Xiaobing.AI
{liweiyuan, daibin, zhouziyi, yaoqi, wangbaoyuan}@xiaobing.ai

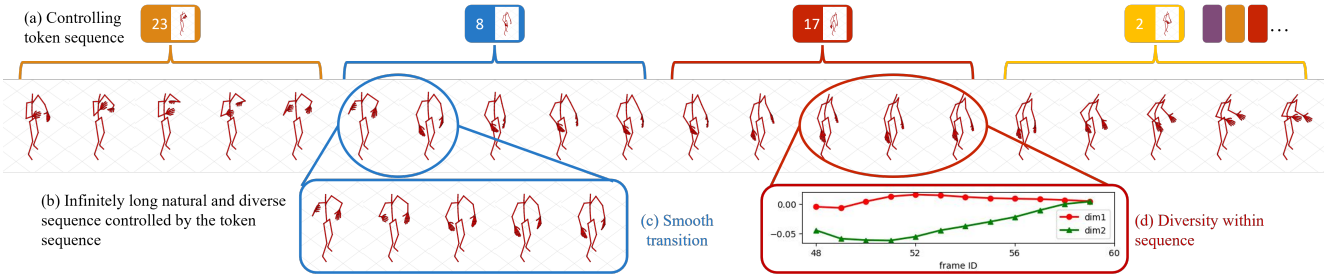


Figure 1. We propose an algorithm to generate diverse and unlimited long sequences without a driving source. The sequence is controlled by the tokens produced by a high-level policy. It makes smooth transitions between states and has natural movement within a state.

Abstract

How to generate diverse, life-like, and unlimited long head/body sequences without any driving source? We argue that this under-investigated research problem is non-trivial at all, and has unique technical challenges behind it. Without semantic constraints from the driving sources, using the standard autoregressive model to generate infinitely long sequences would easily result in 1) ***out-of-distribution (OOD) issue*** *due to the accumulated error, 2) insufficient* ***diversity*** *to produce natural and life-like motion sequences and 3) undesired* ***periodic patterns*** *along the time. To tackle the above challenges, we propose a systematic framework that marries the benefits of VQ-VAE and a novel token-level control policy trained with reinforcement learning using carefully designed reward functions. A high-level prior model can be easily injected on top to generate unlimited long and diverse sequences. Although we focus on no driving sources now, our framework can be generalized for controlled synthesis with explicit driving sources. Through comprehensive evaluations, we conclude that our proposed framework can address all the above-mentioned challenges and outperform other strong baselines very significantly.*

1. Introduction

Recently, synthesizing the character motion (both head and body, rigid or non-rigid) given a certain kind of driving source emerges as a popular topic [29,38,46,52,53]. For example, Learning2Listen [29] is able to predict both the head pose and non-rigid facial expressions of the listener when given both the audio and video sequence of the speaker. PC-AVS [52] can generate a sequence of photo-realistic talking heads when given a driving audio, a reference video for the head pose, and a target face image. Similarly, Bailando [38] can synthesize a dancing body motion sequence for a music audio input. Those models all aim to build *semantic correspondence* between the character motion and the external driving sources, i.e., the lip motion has to be aligned with the corresponding audio segment, and the body motion has to be compatible and harmonized with the rhythm and beat of the driving music. However, there are many scenarios where the character still needs to be continuously animated without an observable driving source, especially when controlling and synthesizing long and versatile character motions that are life-like. For example, in the "idle" state of a VTuber, when there is no interaction and hence not responding to any external signals, it's indispensable to control the motion behaviors to make it look natural. And perhaps the demand for naturalism is even higher for live streaming of a photo-realistic digital human avatar in order to cross the uncanny valley. Therefore, how to model and control such motion behaviors becomes an urgent yet important research problem, while we move toward the era of the metaverse.

This is not a trivial problem compared to the scenario when the driving source exists. There is one key difference between these two tasks. When the driving source is available, the model is conditioned on the driving information

at the same timestamp and thus trained to learn the correspondence between the input and output. However, when the driving source is absent, the only information that the model can condition on is the past sequence, which is also generated by the model itself. We argue that such autoregressive motion synthesis may bring multiple fatal consequences, namely 1) the out-of-distribution issue, 2) the lacking of diversity issue, and 3) the periodic repeating pattern issue. These problems will be carefully analyzed in Sec. 3.

To fill the gap and address the above challenges, in this paper, we officially define the problem of controlling character motions without a driving source. Our algorithm consists of three parts: a quantizer that encodes the sequence to a discrete token sequence, a low-level policy to decode the token sequence and a high-level policy to generate the token sequence. Each part is responsible for a challenge mentioned above. To tackle the **out-of-distribution issue**, we discretize the continuous feature space using quantization algorithms like vector-quantization variational autoencoder (VQ-VAE) [45] to a token sequence. Directly decoding the token sequence back into the continuous feature space using the VQ-VAE decoder (as in many previous works [29, 38]) will cause the **lacking of diversity issue**. Therefore, we instead employ a reinforcement learning (RL) framework to replace the VQ-VAE decoder. The policy network, which we call the low-level policy, is responsible for continuously generating the next frame based on the current token and the past frames. In this sense, the token space of the VQ-VAE can also be regarded as a task space for the low-level policy. To ensure the policy network capable of generating diverse frames given the same input token, we also add a randomly sampled Gaussian noise as input to the policy network. Three types of rewards are designed for different purposes: 1) the *realistic reward* forces the generated sequence to look natural; 2) the *diversity reward* is proposed to encourage the policy to produce diverse outputs and 3) the *correspondence reward* requires the produced result to hit the input token. Finally, to avoid the **periodic pattern issue** caused by the autoregressive generation procedure, we instead design a random generation scheme, which we call a high-level policy, to produce the token sequence. It should be noted that, though designed for the scenario without a driving source, such a framework can be generalized for other driving tasks. The discrete token space serves as an interface between the high-level policy and the low-level policy. We only need to change the high-level policy when the driving source exists without modifying the low-level policy. To sum up, we make the following contributions:

- To our best knowledge, we are the first to define and study the problem of controlling character motion without any driving source. We unveil the problem's importance and its technical challenges.
- We design a framework consisting of a high-level policy, a token/task space, and a low-level policy to solve this problem. The token space can also be suited for other high-level policies with driving sources. The low-level policy with carefully designed reward functions can produce natural and diverse results.
- We conducted extensive experiments on two public body skeleton datasets and a self-collected VTuber face dataset. Empirical results show that our framework achieves better performance than previous algorithms.

2. Related work

Face/head/body driving Prior works mainly focus on driving the face/head/body motion with observable and semantically meaningful sources, which includes speech audio [41, 46, 52, 53], music [6, 21, 25, 38], video [5, 18–20, 44] and even text [13, 17, 50]. Among them, the works that aim to accurately control the lip motion to align with the speech audio [22, 23, 33, 41, 43, 51] has received much attention. To better control the results, PC-AVS [52] also depends on a separate head pose driving source from a reference video. StyleTalker [28] learns an audio-to-motion latent space to produce the head motion. Different from the audio-driven head sequence generation task, Learning2Listen [29] tries to generate the head sequence of a listener given both the video and the audio of the speaker. As for the body driving, [3] drives the body and gesture given the speech audio. Bailando [38] produces a sequence of dancing skeletons based on the input music. Unlike all these tasks which require either one or a few types of observable driving sources, our method aims to generate life-like motion sequences without such driving signals.

Reinforcement learning in character control Physics-based character control task [16, 24] has a long history. Recently, there are many works trying to solve this problem using reinforcement learning [30–32, 47]. MotionPrior [30] regards the controlling problem as a Markov Decision Process (MDP) and trains a policy net to generate the skeleton sequence. It manually defines the state-similarity metric as the reward function. AMP [32] replaces the similarity reward function with an adversarial network [12] to get rid of the cumbersome human-designed metric. ASE [31] further improves diversity by introducing noise into the policy network. Our method is inspired by this line of work. However, Instead of taking a pre-defined goal like *location* or *strike* as input, our policy network takes the token as the task. The reward function corresponding to the task/token is also different. In the pre-defined task case, the reward function is hand designed. However, in our case, we can directly use the VQ-VAE encoder as the reward function.

Quantization + Prior Models. A two-stage quantization + prior learning model is first proposed in the image synthesis task [9, 36, 45, 49]. These algorithms first learn a discrete

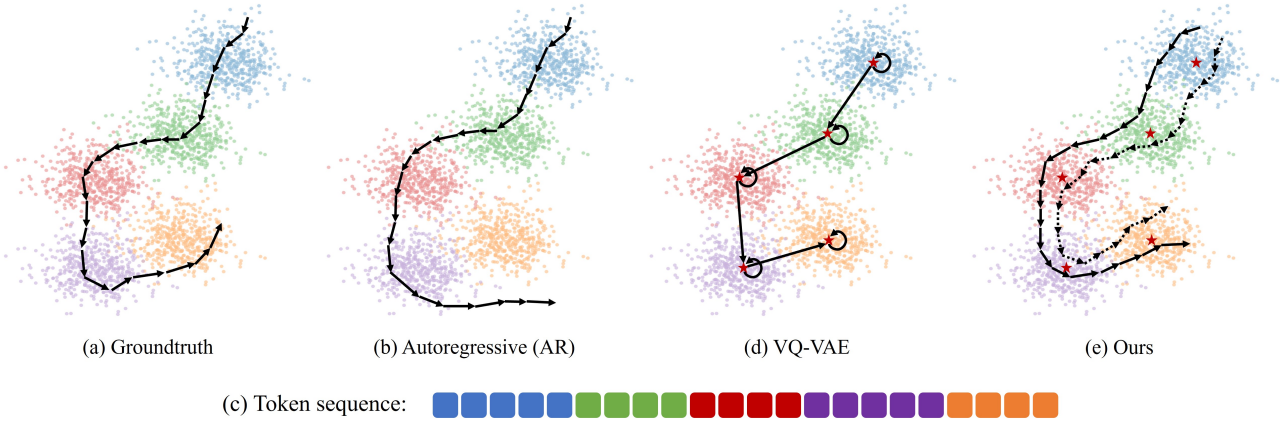


Figure 2. (a) A ground-truth trajectory in the feature space. (b) A trajectory generated by an autoregressive model may suffer from the OOD issue. (c) The token sequence by applying the VQ-VAE encoder and quantizer to the ground-truth trajectory. (d) The trajectory is decoded from the token sequence using the VQ-VAE decoder. (e) The trajectories are decoded from the token sequence using our low-level policy. Our algorithm can produce diverse smooth trajectories.

representation of an image and then train a prior model on this representation. Some works adapted this fashion into text-to-image generation [1, 8, 34, 35]. In these works, a prior condition on the text embedding is learned after obtaining the image discrete representations. This philosophy is also applied to the driving tasks [17, 29, 38], where a conditional probability distribution is learned on the image tokens conditioned on the driving source. Though our work also uses a similar quantization algorithm as the first step, we have a different purpose for such a design, where we wish to use a discrete space to avoid the OOD issue.

Motion Prediction. Motion prediction [2, 4, 11, 27, 48] aims to predict the human motions in the near future based on the past motions. Both deterministic methods [11, 27] and probabilistic models [2, 4, 48] are designed. The key difference between our problem and motion prediction is that we focus on generating unlimited long natural and diverse sequence rather than the near future motion.

3. Problem Definition and Analysis

Problem Definition. The existing driving problem assumes there is a driving source c_t at each timestep t. The driving engine tries to model the probability distribution $p(x_t|c_t)$[1], where x_t is the representation of the face/body. In this paper, we define the driving problem from a different perspective. Suppose the virtual human is placed in an environment (i.e. live streaming). It should decide its facial expression and body pose at each timestep no matter whether the driving source exists or what the driving source is. Generally speaking, we should model $p(x_t|x_{<t}, c_t)$ for each t. This procedure should continuously go on until the whole event ends. Unlike previous driving problems that ignores $x_{<t}$ and only models $p(x_t|c_t)$, we consider another scenario when $c_t = \emptyset$ and model $p(x_t|x_{<t})$. Such a setting is even more common in practice since the underlying driving source is often not observable. We argue that this problem is both challenging and important for further building more complicated decision models.

Challenges. The first challenge is the OOD issue. Our model $p(x_t|x_{<t})$ is trained on the ground-truth dataset. However, during the inference phase, we apply the model to the dataset generated by the model itself. That being said, the distributions of $x_{<t}$ in the training and inference phases are different, making the output x_t following an even more different distribution. An illustration is shown in Fig. 2(a)-(b). The groundtruth trajectory is shown in Fig. 2(a). The trajectory generated by an autoregressive model is shown in Fig. 2(b). These two trajectories look similar in the beginning. However, as the number of steps increases, the path becomes significantly different and the generated path may go to the OOD region.

Using VQ-VAE to constrain the output space of the autoregressive model in a discrete inlier set can avoid the OOD issue. Suppose we have trained a VQ-VAE to cluster the dataset into 5 different clusters, each represented in different colors. Fig. 2(c) shows the corresponding token sequence of the gourndtruth trajectory. However, directly decoding the token sequence will produce a trajectory lacking of diversity, as shown in Fig. 2(d), which is the second challenge of the problem. A good model should be able to produce diverse and smooth trajectories as shown in Fig. 2(e).

Lastly, there is also the periodic pattern or the fixed point problem. Though the prior model is conditioned on all the history $x_{<t}$ theoretically, we usually use a fixed length of

[1]Sometimes there is also an identity input.

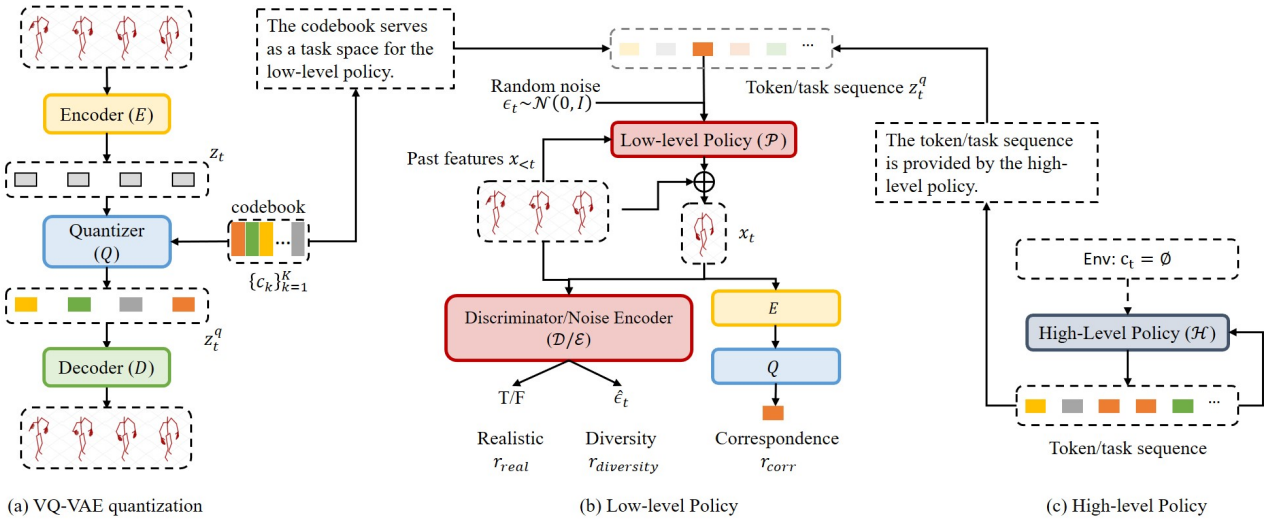


Figure 3. Pipeline. (a) VQ-VAE discretizes the sequence. The token space serves as both the input of the low-level policy and the output of the high-level policy. (b) The low-level policy takes the token, the past features, and a random noise as input and produces the next feature. (c) The high-level policy generates the token sequence.

past window $x_{t-\Delta T:t-1}$ in practice. Suppose we model the distribution $p(x_t|x_{t-\Delta T:t-1})$ as Gaussian and use the mean of the Gaussian distribution as x_t during the inference time, as in many autoregressively regression tasks. Then x_t becomes completely determined by $x_{t-\Delta T:t-1}$. Further, x_{t+1} is then determined by $x_{t-\Delta T+1:t}$, which is again a deterministic function of $x_{t-\Delta T:t-1}$. So we can write $x_{t:t+\Delta T-1} = f(x_{t-\Delta T:t-1})$, where $f(\cdot)$ is a function correlated to the autoregressive model. The autoregressive model is like applying the same function again and again. Such a process will make the same sequence appear repeatedly. Though we usually add some noise during the sampling procedure in practice, such an issue cannot be completely avoided, as will be presented in our experiments.

4. Method

Our method consists of three parts: 1) a discrete token space yielded from a VQ-VAE model, 2) a low-level policy network that decodes the token sequence to the continuous feature space, and 3) a high-level policy that produces the token sequence. The overview pipeline is illustrated in Fig. 3. We then introduce more details for each part.

4.1. Token Space Derived from VQ-VAE

The token space serves as both the output space of the high-level policy and the input space of the low-level policy. Once the token space is determined, the high-level policy and the low-level policy can be disentangled and separately developed. The low-level policy only needs to focus on how to generate a natural and diverse sequence based on the provided token sequence, while the high-level policy only cares about how to produce the token sequence based on the environment. In this paper, we use the quantized latent space of a VQ-VAE model as the token space, which can effectively avoid the out-of-distribution issue since it constrains the output of the high-level policy into a discrete inlier set.

Technically, either a clip-level VQ-VAE or a frame-level VQ-VAE can be adopted. In our current implementation, we adopt a frame-level VQ-VAE. For each frame, the feature $x_t \in \mathbb{R}^d$ is encoded into a continuous latent vector $z_t \in \mathbb{R}^\kappa$ via an encoder $E(\cdot)$, *i.e.* $z_t = E(x_t)$, where d is the feature space dimension while κ is the latent space dimension. The latent vector z_t is then assigned to the nearest code in a learnable codebook $\{c_k\}_{k=1}^K$, where K is the codebook size. Let $Q(\cdot)$ be the quantizer and q_t stand for the code index, then

$$q_t := Q(z_t) = \arg\min_k ||z_t - c_k||_2. \quad (1)$$

Denote z_t^q as the q_t-th entry in the codebook $\{c_k\}_{k=1}^K$. It is also known as the quantized version of the latent vector z_t. The quantized z_t^q is then decoded to the original feature space via a decoder $D(\cdot)$, *i.e.* $\hat{x}_t = D(z_t^q)$.

The VQ-VAE, including the encoder, decoder, and codebook, is optimized using the objective

$$\mathcal{L}_{VQ} = ||x_t - \hat{x}_t|| + ||\text{sg}[z_t] - z_t^q||_2 + \beta||z_t - \text{sg}[z_t^q]||_2, \quad (2)$$

where sg$[\cdot]$ means stop gradient and β is a hyperparameter. Following [45], we use $\beta = 0.25$ in all our experiments.

4.2. Low-Level Policy

Most prior works [29, 38] directly use the VQ-VAE decoder to generate the output given the discrete token sequence. Such a design at least has two drawbacks. Firstly,

it may produce unnatural (i.e., flicking) sequences lacking diversity, as shown in Fig. 2(d). Secondly, it only considers the token at/around the current timestamp when decoding. No long-term dependency of the generated sequence is considered, degrading the generation quality when the sequence becomes infinitely long.

To tackle the issues of the VQ-VAE decoder, we use a reinforcement learning framework, which we call a low-level policy, to decode the token sequence. The state at the current step t includes not only the current token z_t^q, but also the past frames $x_{t-\Delta T:t-1}$ for smoothness, and a random sample ϵ_t for diversity. The policy network $\mathcal{P}(\cdot)$ takes the state $s_t = [z_t^q, x_{t-\Delta T:t-1}, \epsilon_t]$ as input and outputs a d-dimensional deviation vector δx_t as action. The next frame feature then becomes $x_t = \delta x_t + x_{t-1}$. An illustration is shown in Fig. 3(b).

We design three rewards regarding different constraints on the future frame x_t. The realistic reward corresponds to whether the sequence $(x_{t-\Delta T:t-1}, x_t)$ looks natural or not. A discriminator is adapted to produce a realistic score. It is an MLP that takes $(x_{t-\Delta T:t-1}, x_t)$ as input. The realistic reward at step t then becomes

$$r_{t,real} = \mathcal{D}(x_{t-\Delta T:t-1}, x_t). \tag{3}$$

The reward is normalized to the range $(0, 1)$ using a sigmoid activation layer. The discriminator is trained in an adversarial manner [12]. The loss for the discriminator is

$$\mathcal{L}_{gan} = \text{CE}\left(\mathcal{D}(x^r_{t'-\Delta T:t'}), 1\right) + \text{CE}\left(\mathcal{D}(x_{t-\Delta T:t}), 0\right), \tag{4}$$

where $\text{CE}(\cdot,\cdot)$ stands for the cross entropy function, x^r is a random sequence from the training dataset and t' is a random start frame index.

The second reward regards the correspondence between the output feature x_t and the input token q_t. We leverage the trained VQ-VAE encoder E and quantizer Q for this reward. It is designed as

$$r_{t,corr} = \mathbb{I}(q_t, Q(E(x_t))), \tag{5}$$

where $\mathbb{I}(\cdot,\cdot)$ is an identical function that equals to 1 when the two arguments are equal and 0 otherwise. It requires that x_t should be assigned to q_t by the VQ-VAE encoder and quantizer. However, only using this term will make the training procedure problematic because all the possible x_t are equally bad as long as it does not hit q_t. To tackle this issue, we also calculate the change of the ℓ_1 distance between x_t and z_t. This change is further clipped into range $[-1, 0.8]$. If $q_t \neq Q(E(x_t))$, we use the clipped change as the reward, which encourages x_t to move towards z_t when it is assigned to a different token.

The last reward encourages the policy to produce diverse outputs given different noise ϵ_t. Though the policy network takes ϵ_t as input, it is very likely to completely ignore the noise input without such a diversity reward. To enforce the policy network to encode the information of ϵ_t, we train another noise encoder to reconstruct ϵ_t given $(x_{t-\Delta T:t-1}, x_t)$. The noise encoder is trained using the L2-Loss

$$\mathcal{L}_{diverse} = \frac{1}{2}||\epsilon_t - \mathcal{E}(x_{t-\Delta T:t-1}, x_t)||_2^2, \tag{6}$$

where $\mathcal{E}$ is the noise encoder. In practice, the noise encoder and the discriminator share the same architecture and weights. The diversity reward then becomes

$$r_{t,diversity} = -||\epsilon_t - \mathcal{E}(x_{t-\Delta T:t-1}, x_t)||_2^2. \tag{7}$$

The final reward function can be written as

$$r_t = w_r \cdot r_{t,real} + w_c \cdot r_{t,corr} + w_d \cdot r_{t,diversity}, \tag{8}$$

where w_r, w_c and w_d are the weights of each reward. The value at step t is then defined as

$$V_t = r_t + \sum_{dt=1}^{+\infty} \gamma^{dt} r_{t+dt}, \tag{9}$$

where γ is the discount factor (set as 0.98).

The low-level policy network is trained using proximal policy optimization (PPO) [42]. We use GAE(λ) [37] to compute the advantage function and TD(λ) [40] to update the approximate value function. The algorithm for training the low-level policy is shown in Algorithm 1. The discriminator $\mathcal{D}$ and the noise encoder $\mathcal{E}$ are optimized during the training of the policy network.

Note that our low-level policy can also be trained using supervised learning, regarding the reward function (8) as the loss function. However, using supervised learning will degrade the performance since it ignores the long term dependency. We will empirically demonstrate that using RL will produce better results than supervised learning.

It is interesting to notice that our low-level policy has a connection with the popular diffusion models [15, 39]. We also have a random noise input, and two condition inputs (the token and the past feature sequence). The advantage of our method is that we only need a single iteration, which is more efficient. Moreover, with the RL training strategy, it is able to achieve better long term dependency.

Algorithm 1 Algorithm of Low-Level Policy

Input: VQ-VAE encoder E, VQ-VAE quantizer Q, VQ-VAE codebook, dataset.
Output: Low-level policy network.
while Not converge **do**
  Update $\mathcal{D}$ using loss function 4.
  Update $\mathcal{E}$ using loss function 6.
  Update low-level policy network $\mathcal{P}$ using PPO.
end while

4.3. High-Level Policy

Generally speaking, the high-level policy, denoted as $\mathcal{H}$, takes both the driving source c_t and the past feature $x_{<t}$ as input and outputs the token for the next step z_t^q. So we can write the general form as $z_t^q = \mathcal{H}(x_{<t}, c_t)$. In this paper, we focus on the scenario when $c_t = \emptyset$ and leave the the other types of high-level policies for future work.

A straightforward way to design the high-level policy is to use an autoregressive model that takes the past tokens $z_{<t}^q$ as input and continuously generate the next token, *i.e.* $z_t^q = \mathcal{H}(z_{<t}^q)$. However, as discussed in Section 3, using such a model will have the periodic pattern issue. Many previous works manually add some randomness in the sampling procedure to avoid the issue. For example, instead of selecting the token with the highest probability, we can uniformly sample from the top K tokens [45]. Considering that our low-level policy can add more diversity to the decoded sequence, we adopt this scheme as one of our high-level policies, denoted as *Ours-A(utoregressive)*.

We also consider using a random prior. Specifically, for each 20-frame clip, we randomly choose a token from the codebook. Interestingly, such a simple random strategy can produce the best results in most cases. This scheme is denoted as *Ours-R(andom)*.

5. Experiments

We evaluate our algorithm on two public body datasets, namely the Trinity Gesture dataset [10] and the AIST++ dataset [26], and a self-collected face dataset. The Trinity Gesture dataset includes 224 minutes of body motion and the corresponding audio. The AIST++ dataset contains 1,408 sequences of 3D human dance motion along with the music. In our experiments, we assume that the audio/music is unavailable and only use the body motion data. We also collect 46.4 hours of live-streaming data of a female VTuber. In most of the time, there is indeed no observable driving source but the anchor still have some natural expression and movements. This dataset is split into a training set with 37.4 hours and a test set with 9 hours. Then we detect the face region of the anchor and extract the expression and pose features using EMOCA [7]. This dataset is named VTuber-EMOCA. The implementation including the network architectures and the training details are described in the supplemental material.

5.1. Quantitative Evaluation

We first quantitatively compare our algorithm with many previous works that can also be adapted to generate an infinitely long sequence. These baseline methods include

- **Random-T:** Randomly select a sequence of length T from the training set at each step and combine all the sequences together as a single long sequence. Though it looks trivial, this is actually a strong baseline because every clip is from the real dataset.
- **SRandom-T:** A stronger baseline than *Random-T*. We linearly interpolate between every two randomly selected clips in *Random-T* to further improve the smoothness.
- **Autoregressive:** An autoregressive transformer is directly trained on the continuous data using the L2 loss. Compared to *Random-T*, it does not need to maintain a huge memory pool for the data clips.
- **VQ-VAE [45]:** Following the quantization + prior model fashion, a VQ-VAE model is first trained on the sequences. Then an autoregressive prior model is further learned on the token sequences. We use both frame-level VQ-VAE as our algorithm does and a clip-level VQ-VAE as Learning2Listen [29] does, which are respectively denoted as VQ-VAE-F(rame) and VQ-VAE-C(lip).
- **AMP [32]:** Adversarial motion prior (AMP) is also an autoregressive model. It learns a discriminator to distinguish the ground-truth and the generated sequences. The output of the discriminator is used as a reward in the reinforcement learning framework.
- **ASE [31]:** Adversarial skill embedding (ASE) further adds noise as input to improve the diversity compared to AMP. Neither AMP nor ASE can use the token sequence to control the output sequence.

We evaluate the quality of the generated sequence using the commonly adopted Frechet distance [14] (FD) between the test dataset and the generated sequences. We first generate a set of sequences with 2000 frames, which we consider as infinitely long[2]. These sequences are then randomly sliced into T-frame clips. The mean and variance are computed on the $(T \times d)$-dimensional space. We use FD-T to stand for the FD between the generated sequence and the test sequence with clip length T. In our evaluation, we use $T = 5$ and 10.

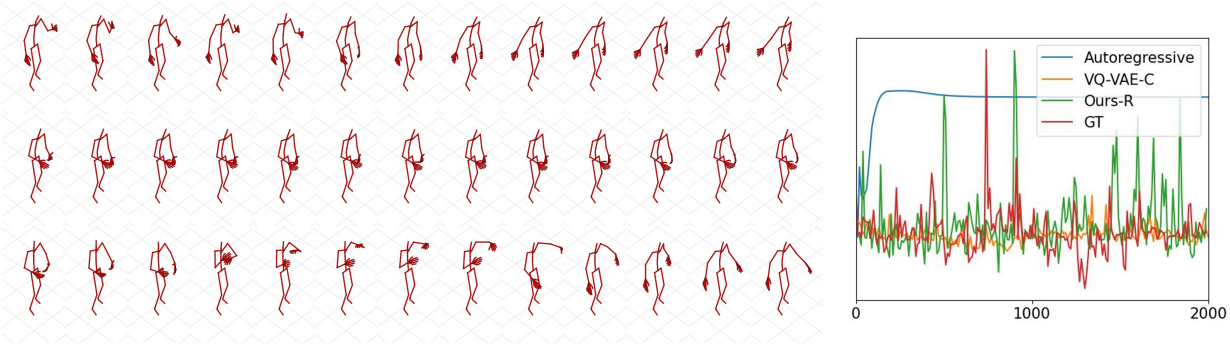
The performance of different algorithms are shown in Tab. 1. The best performance is shown in boldface while the second-best method is underlined. Besides comparing baselines, we also provide the ground-truth (GT) performance as a reference by dividing the test set into two groups and calculating the FD between them. We find that the FD distance becomes relatively large in the raw space since the dimension of the raw space is high. To achieve a better sense of these quantitative numbers, we extract the principal components using PCA[3] and calculate the FD in the PCA space. For Trinity Gesture dataset, we present FD score in both

[2]We observe similar results with longer sequences. So we can regard 2000 frames as infinitely long.

[3]For Trinity Gesture and VTuber-EMOCA expression, we use 20 componets. For the simpler VTuber-EMOCA pose, we use 5 components. For the more complicated AIST++, we use 40 components.

dataset	Trinity Gesture				AIST++		VTuber EMOCA			
Method	Raw space		PCA space		PCA space		Expr.-PCA		Pose-PCA	
	FD-5	FD-10	FD-5	FD-10	FD-5	FD-10	FD-5	FD-10	FD-5	FD-10
GT	62.86	134.1	2.25	2.18	8.42	7.75	1.68	1.68	0.51	0.55
Random-5	99.84	278.8	6.64	8.01	46.76	30.83	11.28	8.38	5.19	5.03
SRandom-5	76.55	185.8	2.91	2.51	19.52	10.18	5.86	6.24	1.02	0.84
Autoregressive	1503	3019	43.96	44.86	85.37	47.97	59.19	53.36	10.32	10.13
VQ-VAE-F [45]	136.1	280.2	8.92	7.61	22.75	21.74	4.81	3.89	2.37	2.71
VQ-VAE-C [29]	130.1	262.8	9.19	7.99	32.25	33.17	16.95	16.79	2.47	2.56
AMP [32]	143.4	294.0	8.65	7.51	22.97	19.64	9.71	8.28	3.85	3.92
ASE [31]	81.14	171.0	3.44	3.30	13.58	12.70	1.82	2.47	**0.42**	0.70
Ours R (No RL)	71.63	155.3	4.33	5.43	12.15	11.69	3.21	4.37	0.79	0.80
Ours A	47.70	101.8	3.25	3.03	10.68	11.07	1.64	**1.64**	0.53	0.69
Ours R	**41.54**	**92.00**	**2.15**	**2.25**	**9.22**	**8.89**	**1.50**	2.15	0.50	**0.58**

Table 1. Quantitative Evaluation on Trinity Gesture, AIST++ and VTuber-Emoca Datasets.



(a) Generated Sequences. From top to bottom: Autoregressive, VQ-VAE-C and Ours-R

(b) Evolution of the 1st dimension.

Figure 4. Visualization on Trinity Gesture dataset.

spaces to show consistency. Ours-R achieves the best performace among all the algorithms in both the raw and the PCA spaces. The autoregressive algorithm completely fails since it drops into a fixed point very quickly as we will see in Sec. 5.2. Both VQ-VAE-F and VQ-VAE-C suffer from poor performance since they cannot produce diverse trajectories as demonstrated in Fig. 2. Replacing the reinforcement learning framework with supervised learning also degrades the performance, though it still outperforms most of the comparing methods. One may observe that the FD for the generated sequence is sometimes lower than that of the groundtruth in the raw space. This is caused by the unstable covariance matrix of the data. Using the PCA space clearly mitigate the issue.

Considering that the FD score in the PCA space is more meaningful, we only report the FD in the PCA space on AIST++ dataset. *Ours-R* again produces the best results among all the algorithms. We further validate the performance on our VTuer-EMOCA dataset. Following Learning2Listen [29], we divide the 56-dimensional EMOCA feature into a 53-dimensional expression part and a 3-dimensional pose part, the results of which are reported separately. Only ASE produces similar results to our methods. Note that ASE does not have a simple interface for high-level policies, yet our algorithm still produces better results on the expression part, which contains most of the dimensions of EMOCA.

5.2. Qualitative Evaluation

Generation Comparison. Fig. 4(a) shows the sequence generated by Autoregressive (top), VQ-VAE-C (middle) and Ours-R (bottom) on the Trinity Gesture dataset. The Autoregressive model produces a reasonable sequence in the beginning. However, it gradually moves to the OOD region and stucks in a completely still state. VQ-VAE-C clearly suffers from the lacking of diversity issue. The hands always keep in front of the hip, though every small segment looks natural. We observe that VQ-VAE results sometimes produces flickering results. This may be caused by the trade-off between the window size and the codebook size. Ours-R produces natural and diverse results. We also plot the evolution of the 1st dimension in Fig. 4(b). Ours-

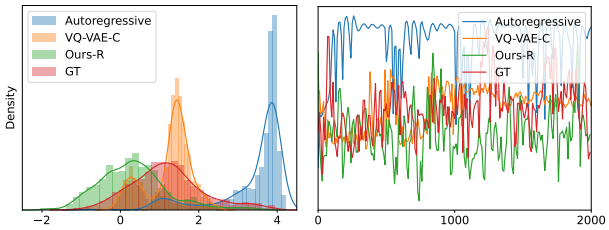


Figure 5. (left) Distribution and (right) evolution of the first dimension of the generated VTuber-EMOCA sequences.

R (green curve) has the similar pattern as the groundtruth (red). Autoregressive (blue) soon degenerates to a completely still state while VQ-VAE-C (orange) has relatively small amplitude.

Controllable Generation. Fig. 1 shows how we can control the generation by specifying the token sequence. The top row is the high-level token sequenceWe visualize the direct decoding of the first four tokens via the VQ-VAE decoder. The middle row shows the frames generated by our low-level policy based on the provided token sequence. Each token corresponds to 20 frames and we display one of every 4 frames. Fig. 1(c) shows the details from the 21st to 25th frames. Our algorithm produces smooth transitions between different tokens. Fig. 1(d) plots the evolution of the first two dimensions from the 48th to 59th frames. These frames are all similar to the direct decoding. However, they do not keep completely still. Rather, they have a natural tiny movement around the specified token. More visualization results on both Trinity Gesture dataset and VTube-EMOCA dataset are included in the supplementary materials.

Addressing Challenges. In Sec. 3, we analyzed three challenges in this problem. We then empirically address all the challenges. We visualize the distribution of the first dimension of the generated sequences on VTuber-EMOCA dataset in Fig. 5(a). Autoregressive clearly has the OOD issue. Using a discrete space can effectively avoid this issue. Both VQ-VAE-C and our algorithm range in the region where the groundtruth has high density. However, the distribution of VQ-VAE-C is more concentrated, meaning that it suffers from the lacking of diversity issue. We also plot the evolution of the first dimension in Fig. 5(b). The autoregressive algorithm generates a periodic pattern though the pattern in each period is slightly different. The periodic pattern issue is not that obvious in the VQ-VAE-C algorithm because there are also some additional randomness during sampling. However, we do observe that some patterns repeatedly appear in the token sequence. In addition, we apply a user study to evaluate the quality of the generated sequences in the supplementary file.

Dis.	Cor.	Noi.	FD-5↓	FD-10↓	Hit↑	Div↑
✓			8.65	7.51	–	2.67
	✓		6.88	5.78	0.16	4.14
✓	✓		5.31	5.11	0.05	4.27
✓		✓	3.44	3.30	–	5.20
	✓	✓	8.15	6.36	0.19	4.27
✓	✓	✓	2.15	2.25	0.14	5.01
✓			9.71	8.28	–	2.10
	✓		12.54	11.24	0.19	4.11
✓	✓		2.26	2.78	0.15	5.76
✓		✓	1.82	2.47	–	6.10
	✓	✓	9.36	7.85	0.18	5.42
✓	✓	✓	1.50	2.15	0.19	5.99

Table 2. Ablation Study on Trinity Gesture (first section) and VTuber-EMOCA-expr (second section).

5.3. Ablation Study

We designed three different rewards for different purposes in the low-level policy framework. In this subsection, we apply an ablation study to demonstrate how each reward contributes to the whole framework. Besides the quality metrics FD-5 and FD-10, we use another two metrics regarding correspondence and diversity. We use the hit rate to evaluate the correspondence between the generated sequence and the controlling task token. It counts how many percentages of the generated frames actually hits the task token. To evaluate the diversity, we first cluster the frames in the training data into 100 clusters. During generation, we generate multiple sequences for each initialization. The entropy of the cluster-ID histogram is calculated for each initialization. We report the average entropy over different initialization as *Div*.

As shown in tab. 2, using all the three rewards produces the best FD score. Removing the discriminator significantly degrades the performance. Comparing the Div between the first part and the second part in each sections indicates that $r_{diversity}$ indeed benefits improving the diversity of the generated sequence. Lastly, we will not be able to control the low-level policy if we remove the r_{corr}.

6. Conclusion

We define the problem of generating diverse, life-like, and unlimited long head/body sequences without any driving source. The challenges of the problem are analyzed and a pipeline is proposed to solve this problem. Empirical results show that our algorithm produces significantly better results than previous methods. Moreover, our task space and low-level policy can be re-used for further building more complicated decision modules with multiple driving sources, which will be future work.

References

[1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022. 3

[2] Hyemin Ahn, Esteve Valls Mascaro, and Dongheui Lee. Can we use diffusion probabilistic models for 3d motion prediction? *arXiv preprint arXiv:2302.14503*, 2023. 3

[3] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. In *Computer Graphics Forum*, volume 39, pages 487–496. Wiley Online Library, 2020. 2

[4] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018. 3

[5] Stella Bounareli, Vasileios Argyriou, and Georgios Tzimiropoulos. Finding directions in gan's latent space for neural face reenactment. *arXiv preprint arXiv:2202.00046*, 2022. 2

[6] Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 2

[7] Radek Danecek, Michael J Black, and Timo Bolkart. Emoca: Emotion driven monocular face capture and animation. *arXiv preprint arXiv:2204.11312*, 2022. 6

[8] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 3

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[10] Ylva Ferstl and Rachel McDonnell. Investigating the use of recurrent motion modelling for speech gesture generation. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 93–98, 2018. 6

[11] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015. 3

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2, 5

[13] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 5

[16] Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O'Brien. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78, 1995. 2

[17] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022. 2, 3

[18] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3397–3406, 2022. 2

[19] Gee-Sern Hsu, Chun-Hung Tsai, and Hung-Yi Wu. Dual-generator face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 642–650, 2022. 2

[20] Po-Hsiang Huang, Fu-En Yang, and Yu-Chiang Frank Wang. Learning identity-invariant motion representations for cross-id face reenactment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7084–7092, 2020. 2

[21] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. *arXiv preprint arXiv:2006.06119*, 2020. 2

[22] Prajwal KR, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and CV Jawahar. Towards automatic face-to-face translation. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1428–1436, 2019. 2

[23] Avisek Lahiri, Vivek Kwatra, Christian Frueh, John Lewis, and Chris Bregler. Lipsync3d: Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2755–2764, 2021. 2

[24] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Limit cycle control and its application to the animation of balancing and walking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 155–162, 1996. 2

[25] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1272–1279, 2022. 2

[26] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021. 6

[27] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2891–2900, 2017. 3

[28] Dongchan Min, Minyoung Song, and Sung Ju Hwang. Styletalker: One-shot style-based audio-driven talking head video generation. *arXiv preprint arXiv:2208.10922*, 2022. 2

[29] Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20395–20405, 2022. 1, 2, 3, 4, 6, 7

[30] Xue Bin Peng. *Acquiring Motor Skills Through Motion Imitation and Reinforcement Learning*. PhD thesis, University of California, Berkeley, 2021. 2

[31] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *arXiv preprint arXiv:2205.01906*, 2022. 2, 6, 7

[32] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, 2021. 2, 6, 7

[33] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 484–492, 2020. 2

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[36] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[37] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 5

[38] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022. 1, 2, 3, 4

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[40] Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998. 5

[41] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2

[42] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017. 5

[43] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *European conference on computer vision*, pages 716–731. Springer, 2020. 2

[44] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 2

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2, 4, 6, 7

[46] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. *arXiv preprint arXiv:2107.09293*, 2021. 1, 2

[47] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 2

[48] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *Proceedings of the European conference on computer vision (ECCV)*, pages 265–281, 2018. 3

[49] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2

[50] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2

[51] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9299–9306, 2019. 2

[52] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4176–4186, 2021. 1, 2

[53] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 1, 2