Action to Motion with VQ-VAE and Transformer

Hongxuan Liu
University of Alberta
hongxua4@ualberta.ca

Liwei Feng
University of Alberta
liwei4@ualberta.ca

Wendi Wu
University of Alberta
wendi7@ualberta.ca

Abstract

The technique of action recognition has been widely used in our daily life, especially in security and filming areas. At the same time, a reverse technique of action recognition is also developing rapidly, which is, given an action category or a textual description of the action, the technique can visualize the action motion based on the input information. In general, given the differences in input, this technique includes two different topics, text-to-motion, and action-to-motion. Our project will focus on the action-to-motion technique and implement it by combining VQ-VAE and Transformer model together.

1. Introduction

Motion synthesis and generation is a growing field, and many pieces of research are related to this topic. Motion synthesis can be used in various different areas, for instance, the 3D game is one of the potential fields. 3D games have become a trend in the gaming industry as they could provide users with a better gaming experience. And to make the game more competitive, developers need to spend some time on the 3D modeling of gaming characters to make sure the motion of characters is smooth. Thus, the motion generation technique can make the development process much easier.

The key challenge of this problem is that we should not only ensure the accuracy of each generated motion but also the diversity of generated motions in the same action type. In fact, many attempts related to this technique have been made. For example, in [17], given an initial pose and an action type, it can generate 2D motions and come up with a video by putting the 2D motions together frame by frame. [3] is also built for generating 2D human motion only. We can see that those models only target on 2D space, which is clearly not enough to meet the needs of the 3D generation. Compared with these models, a model to generate 3D motions was successfully built by Guo Chuan [6]. This model does not need any prior conditions of initial poses of motions [6]. Instead, a framework that consists of a conditional temporal Variational Auto-Encoder (VAE) based on Lie algebra representation is used to generate a pose sequence given only the action category input [6].

To address the action-to-motion problem, based on our knowledge, Transformer is the ideal model for sequential data prediction. Furthermore, an autoregressive model requires discrete result representation for the probability model. Actually, there are works using VQ-VAE to translate sequential data into a discrete representation. For example, [5] use VQ-VAE to translate motion data to discrete motion tokens, then input them into a Gated Recurrent Unit(GRU) or Transformer. Also, [16] uses VQ-VAE and Transformer to synthesize videos. Inspired by these works, we believe that the model combining VQ-VAE and Transformer together can be used to implement action-to-motion.

In this paper, we propose our model based on VQ-VAE and Transformer and demonstrate the detailed algorithm behind the method. Also, we compare its performance with other related works using 3 different datasets and discuss about the results.

2. Related Work

2.1. Action to Motion

Previously, human motion analysis mainly focused on future human motion prediction, which is given one pose or a sequence of poses, the future motions can be predicted [8]. Barsoum proposed to predict the future motion by HP-GAN [1]. The drawback is that we must have at least one pose. A new method, which uses an action label to generate human motion sequences, has been successfully realized by Mathis using VAE and Transformer in 2021 [8]. Given a sequence of body poses, P_1,....P_T, and an action label **a**, the transformer encoder outputs distribution parameters μ and Σ [8]. To obtain μ and Σ from the encoder, they used extra learnable tokens per action μ_a^{token} and Σ_a^{token}. Using the two distribution parameters, a motion latent representation is sampled and is used as the input of the decoder. Additionally, the decoder also takes an action label and a duration as its input. In the end, the decoder outputs the whole sentence $\hat{P_1}$,....$\hat{P_T}$ on which a reconstruction loss is computed [8].

Another model to carry out action-to-motion is developed by Chuan Guo in 2020 using VAE [6]. Given a real motion or a pose sequence $\mathbf{M} = [\mathbf{p}_1, ..., \mathbf{p}_T]$, VAE aims to maximize the probability of $\mathbf{M}$ sampled from the learned model distribution [6]. The whole model includes 3 parts: posterior network, prior network, and generator. Both the posterior network and prior network have the same structure, the only difference is the parameter values in GRU and decoders. By inputting a one-hot vector **a** of action category, a time counter that records the sequence generation process, and a subvector of the current pose $\mathbf{p}_t$ into the posterior network, the GRU will come up with a set of distribution parameter $N(\mu_{\phi(t)}, \sigma_{\phi(t)})$. Similarly, the prior network will take the same input as the posterior network except for replacing $\mathbf{p}_t$ by $\mathbf{p}_{t-1}$ which is the ground truth of the pose at time t-1, its GRU will also come up with a set of distribution parameters $N(\mu_{\psi(t)}, \sigma_{\psi(t)})$. A prior loss will be computed on the two different sets of distribution parameters to enforce the posterior network to be close to the prior [6]. Using the input of the prior network and $\mathbf{z}_t \sim N(\mu_{\phi(t)}, \sigma_{\phi(t)})$ from the posterior network, the generator outputs a prediction $\hat{\mathbf{p}_t}$, this prediction is compared with the ground truth $\mathbf{p}_t$ while calculating the reconstruction loss.

Guo evaluated this model and other approaches using the same datasets and proved that his model generates the most congruent set of motions to action inputs. We will use this method as our baseline and compare it with our new approach.

2.2. VQ-VAE (Vector Quantized Variational Autoencoder)

Many methods could be used for reconstruction, such as autoencoder [2], VAEs [18], generative adversarial networks(GANs) [12] and VQ-VAE [14] which has become a very popular method in construction research, for example in [10] and [9].

VQ-VAE is a variant of variational autoencoder that uses vector quantization to obtain a discrete latent representation. Compared with VAEs whose outputs are continuous and whose prior is static, VQ-VAE has a discrete output with a categorical distribution and a learned prior [14]. Moreover, it does not suffer from ”posterior collapse” and has no variance issues [14].

VQ-VAE contains three components including an encoder, a quantizer, and a decoder, among which the encoder as well as quantizer map inputs onto a latent embedding space and the decoder decodes the quantization result and reconstructs the original data. The latent embedding space is defined by the size of the discrete latent space and the dimensionality of each latent embedding vector. For every output of the encoder, it is passed through the quantizer and the quantization is used to map it to its nearest vector in the learnable codebook.

The loss function of VQ-VAE contains three parts, which are reconstructed loss, codebook loss and commit loss. Reconstructed loss is used to limit the distance between the input and output [9]. Codebook loss is associated with the distance between the outputs of the encoder and their nearest neighbors from the codebook after quantization [9]. Commitment loss controls fluctuations of encodings [9].

2.3. Transformer

The Transformer is a transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution [15]. It is an artificial neural network architecture that can be used to transform input sequences into output sequences, which makes it suitable for sequence data prediction. In real-life, Transformers can be applied in many fields such as speech recognition and machine translation.

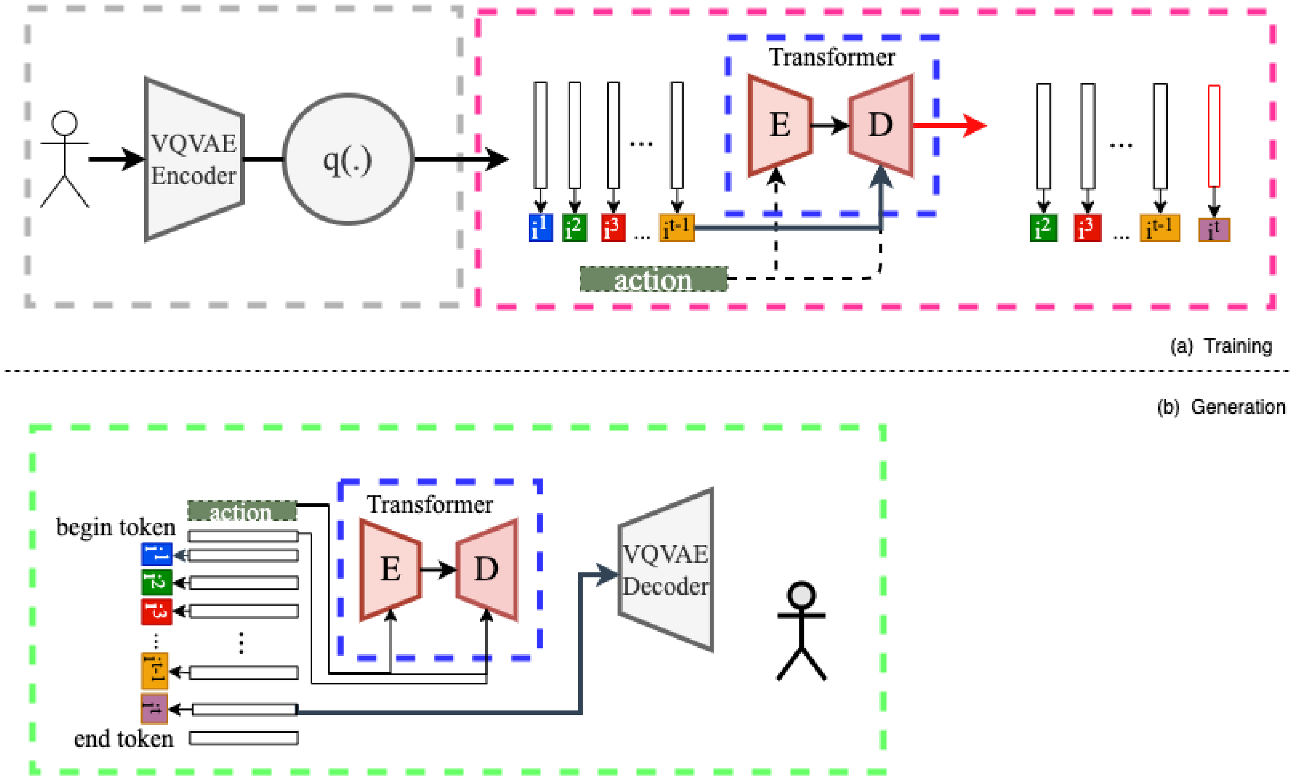
Transformer is composed of two parts: an encoder and a decoder. In most cases, an encoder has 6 small built-in encoders and a decoder has 6 small decoders [11]. Within each encoder, each independent small encoder is composed of a self-attention layer and a feedforward neural net. The first encoder would take the input vectors and process them with the self-attention layer before passing them to the feed-forward neural net [15]. The other encoder would take the output of the previous encoder as input. In the end, the encoder encodes the input sequence to be a vector that contains features about the whole sequence.

Similar to the encoder, each small decoder is composed of three parts: the self-attention layer, the decoder attention layer, and the feed-forward neural net. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack [15]. The output vector of the encoder works as input for the decoder which combines information from both the encoder and decoder and generates meaningful results. Once the work of the first decoder is done the result is passed to the next decoder and so on, until the information is sent to the final decoders [11]. The output of the decoder is the probability of the next component of the sequence.

Furthermore, to improve the result, both the encoder and decoder take attention into consideration, which refers to the relation between words. And Transformer uses positioning encoding to take care of the order of words in the sentence.

3. Methodology

As data of motions is a sort of sequential data, it is straightforward to use recurrent neural network(RNN) to tackle this kind of problem and there are already some



(a) Training

(b) Generation

Figure 1. We designed the VQ-VAE and Transformer architecture. We train the VQ-VAE in advance, input motion data, and train the codebook of Quantization. (a) After training VQ-VAE, we use Encoder and Quantization to generate motion tokens. Then we use a one-hot action label and motion tokens to train the transformer. (b) While generating, we input the label into the transformer and put the output into the decoder to get the corresponding motion.

works based on RNN. For example, [6] use GRU which is a variant of RNN to address action-to-motion problem. In our project, we prefer to use the state-of-the-art(SOTA) model Transformer as our base model, because with the help of position embedding and multi-head attention, Transformer has a better global context and can grasp semantic features from the global context. The global context is crucial in action-to-motion problems given that a different current state may have the same previous state and previous frame motions have a great impact on subsequent motions. Besides, the representation of input and output for Transformer is vital. We can not directly input motion data into the Transformer since it is tough to fit continuous coordinates if we use a non-autoregressive model. Therefore we prefer to use the autoregressive model with an output of probability of motions and we demand a discrete representation of motion data, then we come up with VQ-VAE. The codebook of VQ-VAE can help us translate data of motions into discrete numbers and we can use those numbers in our probability model.

3.1. Motion representation

During the period of project development, we tried different representations of motions. In each dataset, data of motions are represented in values of absolute coordinates in the beginning. However, we found that these absolute coordinates values are not proper for the VQ-VAE training as they are in high variance and it is difficult for the model to converge and fit. Relative coordinates are the final choice. For example, in Humanact12 dataset, we first get t frame of motion data $M_t \in R^{24\times3}$, let root point $P \in R_t^3$ be the first point of t frame. Then we set P_0 as 0, so we minus P_0 for each point.

$$M_t = M_t - P_0 * 24 \tag{1}$$

Finally, for each frame we minus the root of the previous frame.

$$M_t = M_{t-1} - P_{t-1} * 24 \tag{2}$$

In this way, the values of each point vary in a limited range and it becomes much simpler for VQ-VAE to learn how to reconstruct them.

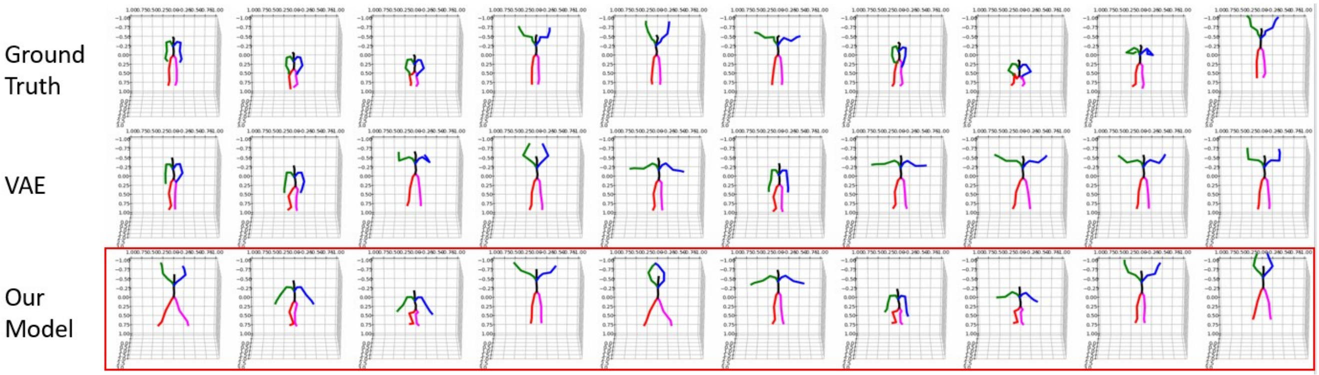


Figure 2. This is a set of results for action jump in HumanAct12 dataset.

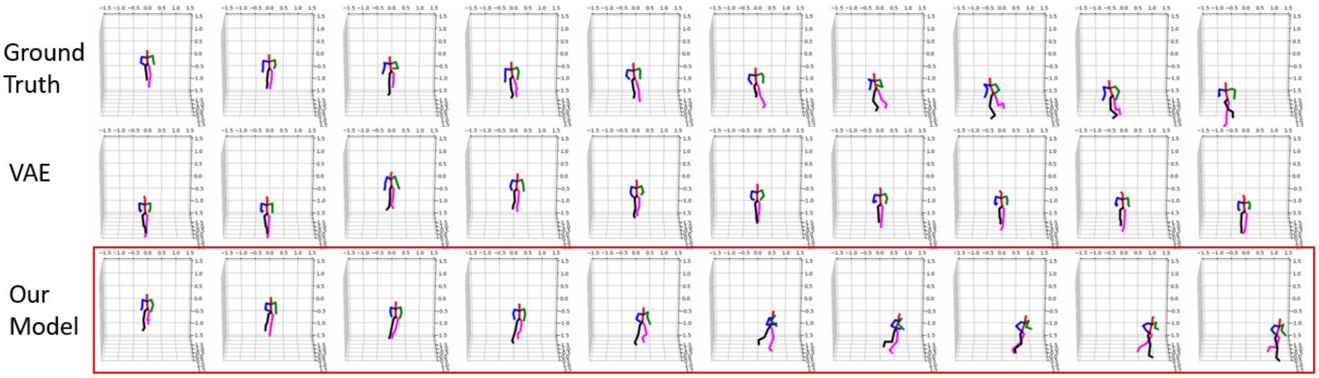


Figure 3. This is a set of results for action run in CMU MoCap dataset.

3.2. VQ-VAE Training

VQ-VAE is a critical part of our model. If we can not obtain decent results of VQ-VAE, we would never get satisfying results for the Transformer. As we mentioned in the last section, we adopt a relative coordinates motion representation which is the input of VQ-VAE. Taking motion data M_t as the input, our encoder has 2 convolution layers with 2 residual layers and outputs a latent vector v. The convolution layers can not only extract features from motion data but also compress these data to a smaller codebook size. Then we use the nearest neighbor search to find the corresponding embedding vector in the embedding space. The embedding space is our codebook B and embedding vectors are entries of the codebook, we set them as $\{b\}_k^K$ where K is the codebook size. We use the index k of the embedding vector as our token of motions T_n. In our model, the input size is compressed by half in each layer, so for input size S, we have $n = S/4$ tokens. In addition, inspired by [5] and [4], we use Generative Adversarial Networks(GANs) in our VQ-VAE training to ensure that the features of motions are enough to reconstruct a motion that is similar to the ground truth. We use L_1 loss function between the reconstructed motion $\hat{M}$ and ground-truth M, with the commitment loss L_E of VQ-VAE [13] and loss of GANs L_c, at the end we can get our total loss L_G.

$$L_G = L_1 + L_E + \lambda * L_c \tag{3}$$

λ is a hyperparameter and we set it as 0.5.

3.3. Transformer training

In our model, we use a simple encoder with only word embedding and then forward the output to the decoder. The decoder receives a vector including action and tokens information. As shown in Figure 1.(a), in our Transformer training, we use one-hot embedding of action A as the input of encoder, and use both A as well as generated tokens T_n as the input of the decoder. We also include the cross entropy function as the loss function to calculate loss between T_0 to T_{n-1} and T_1 to T_n. In our experiments, we set both encoder layers and decoder layers as 4 and set the head count as 8.

3.4. Generation

The generation process is shown in Figure 1.(b). The one-hot embedding of action is inputted into both the encoder and decoder. Also, we input a begin token into the decoder as the stub of start. Then the decoder outputs a

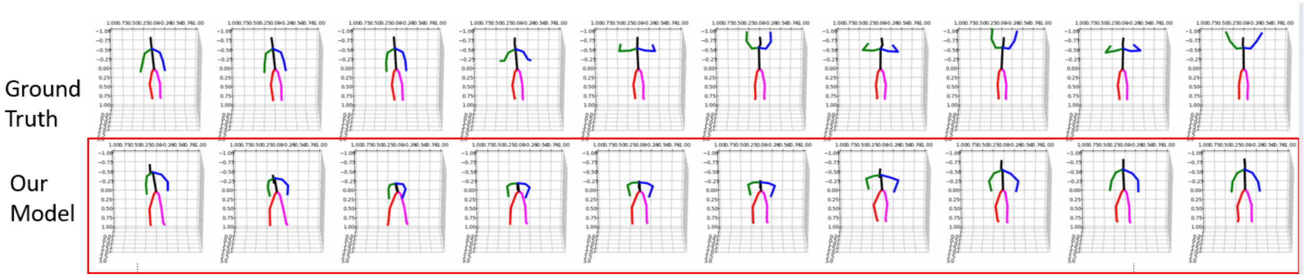


Figure 4. This is a set of results for action Cheer Up in NTU-RGB-D dataset.

probability model of tokens, and we can sample the first motion token from this probability model. After that, we concat begin token and the first token and input them into the decoder together to predict the next token. The generation process repeats this procedure until reaching the end token or the max step. Finally, we input these predicted tokens into the decoder of VQ-VAE and get the corresponding motions.

4. Evaluation

4.1. Dataset

HumanAct12 [6] is a widely-used motion dataset that contains 12 different action categories with 34 subcategories and 1191 motion clips. Each body pose has 24 body joints. And our experiments use the whole 12 action categories.

CMU MoCap is a motion dataset that contains 2605 motion sequences with 22 body joints. 8 actions are included in our experiments.

NTU-RGB-D [7] is a dataset proposed by Nanyang Technological University which contains 120 action categories and each pose has 18 joints. A subset of 13 distinct actions is used in our experiments.

4.2. Metrics

To compare with our baseline method, we used the same metrics as stated in [6], including frechet inception distance(FID), accuracy, diversity, and multimodality.

- FID measures the distance between features of real images and features of generated images.
- Accuracy shows the correlation between the motion and its action type.
- Diversity measures the distance between some pairs of sampled motion features from various action types.
- Multimodality measures the distance between some pairs of sampled motion features from the same action type.

Additionally, a pre-trained RNN action recognition classifier from [6] is used to classify the motions while calculating accuracy. The related equation for calculating diversity and multimodality is:

$$Diversity = \frac{1}{S_d} \sum_{i=1}^{S_d} \|v_i - v_i'\|_2 \tag{4}$$

where S_d is the size of two randomly sampled subsets,
v_i and v_i' are extracted motion feature vector of two subsets

$$Multimodality = \frac{1}{C \times S_l} \sum_{c=1}^{C} \sum_{i=1}^{S_l} \|v_{c,i} - v_{c,i}'\|_2 \tag{5}$$

where C is number of action types,S_l is the size of two randomly sampled subsets,$v_{c,i}$ and $v_{c,i}'$ are extracted motion feature vector of two subsets

4.3. Evaluation Results

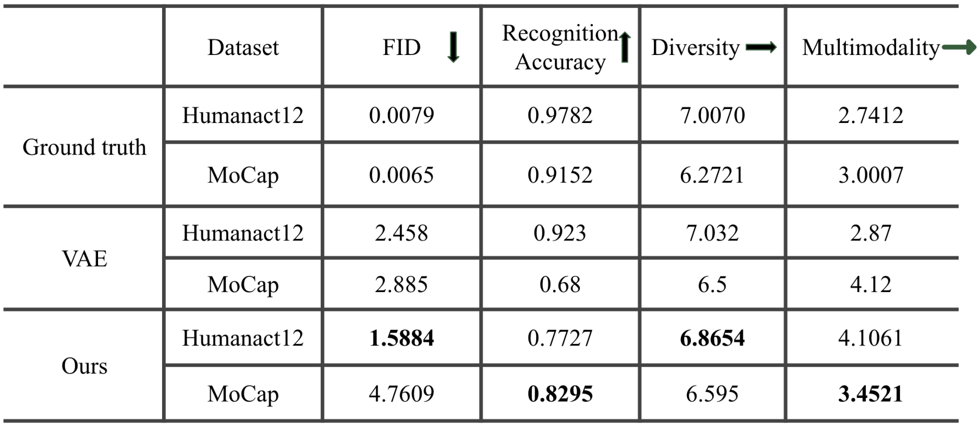
To see how our approach performs, we evaluated our model using HumanAct12, CMU MoCap, and NTU-RGB-D datasets and compare our results with the ground truth as well as the method in [6] which uses VAE to carry out action-to-motion. We name [6] as VAE in our evaluation results.

4.3.1 Generated Results

Figure 2 shows the generated results of the action ”jump” in the Humanact12 dataset of our model and the comparison with the ground truth and the results of the VAE model. It is clear to see that the motion our model generated is congruent with the action type.

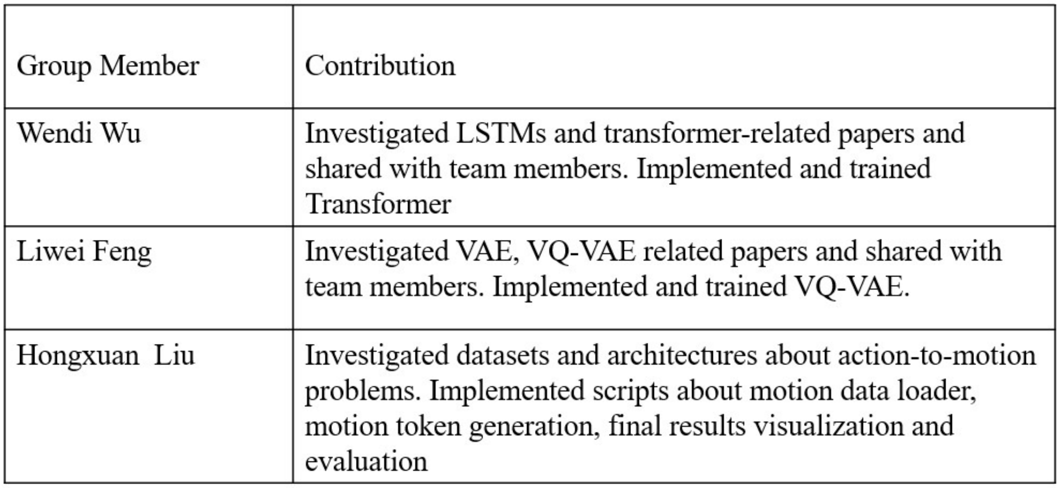
Figure 3 shows the generated results of the action ”run” in the CMU Mocap dataset of our model and the comparison with the ground truth and the results of VAE model. Our model has a reasonable generation for action ”run”, which is even better than the results of VAE model.

For NTU-RGB-D datasets, we found that the generated results of our model are not as satisfying as those of other datasets. For example, in Figure 4 the generated results of



	Dataset	FID ↓	Recognition Accuracy ↑	Diversity →	Multimodality →
Ground truth	Humanact12	0.0079	0.9782	7.0070	2.7412
	MoCap	0.0065	0.9152	6.2721	3.0007
VAE	Humanact12	2.458	0.923	7.032	2.87
	MoCap	2.885	0.68	6.5	4.12
Ours	Humanact12	**1.5884**	0.7727	**6.8654**	4.1061
	MoCap	4.7609	**0.8295**	6.595	**3.4521**

Figure 5. Performance evaluation on HumanAct12, CMU Mocap. The results of VAE are from [6], and ⟶ means the closer to the real motion the better.



Group Member	Contribution
Wendi Wu	Investigated LSTMs and transformer-related papers and shared with team members. Implemented and trained Transformer
Liwei Feng	Investigated VAE, VQ-VAE related papers and shared with team members. Implemented and trained VQ-VAE.
Hongxuan Liu	Investigated datasets and architectures about action-to-motion problems. Implemented scripts about motion data loader, motion token generation, final results visualization and evaluation

Figure 6. Group Member Contributions of this Project

the action ”cheer up” in our model are not consistent with the action type and are quite different from the ground truth. Because of the unsatisfactory results, we did not compare them with those of the VAE model and calculate the metrics.

4.3.2 Quantitative Comparison

Figure 5 indicates the quantitative comparison result between ground truth, baseline method and our model on three different datasets. According to the preference on different metrics stated in the first row, we can clearly see that our model has a lower FID than VAE on the HumanAct12 dataset. Even though VAE has a higher accuracy on HumanAct12, our model performs much better when it comes to the MoCap dataset. For diversity and multimodality, both our model and VAE have similar performance.

5. Conclusion

We proposed using VQ-VAE and Transformer in the action-to-motion scenario in this paper. We presented the detailed framework and parameters of our experiment, and also showed our results and analysis in different datasets. As novices of computer vision, we studied a lot in this project, not only knowledge about action-to-motion, but also research methodology. In the future, we will try different representations of motions such as SMPL, also we may get more diversity of generated motions by changing the codebook of VQ-VAE.

6. Group Member Contributions

Please see 6 for details.

References

[1] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1499–149909, 2018. 1

[2] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. *CoRR*, abs/1804.04488, 2018. 2

[3] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. *CoRR*, abs/1711.08682, 2017. 1

[4] 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. 4

[5] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pages 580–597. Springer, 2022. 1, 4

[6] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 1, 2, 3, 5, 6

[7] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. 5

[8] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, pages 10985–10995, October 2021. 1, 2

[9] Walter H.L. Pinaya, Petru-Daniel Tudosiu, Robert Gray, Geraint Rees, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Unsupervised brain imaging 3d anomaly detection and segmentation with transformers. *Medical Image Analysis*, 79:102475, 2022. 2

[10] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *CoRR*, abs/2006.10704, 2020. 2

[11] Dr J Rogel-Salazar. Transformers models in machine learning: Self-attention to the rescue, Aug 2022. 2

[12] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Margarethe Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. 2

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

[14] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6306–6315. Curran Associates, Inc., 2017. 2

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. 2

[16] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 1

[17] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1

[18] David Zimmerer, Simon A. A. Kohl, Jens Petersen, Fabian Isensee, and Klaus H. Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection. *CoRR*, abs/1812.05941, 2018. 2

A. Appendix

A.1. Resipotory and Dataset

Our code is in `https://github.com/hideak1/action2motion`, and you can download the dataset and our pre-trained model by following our README.