Improved Text-Driven Human Motion Generation via Out-of-Distribution Detection and Rectification

Yiyu Fu$^{1\,[0009-0009-6509-1517]}$, Baoquan Zhao$^{1,*\,[0000-0002-0574-1663]}$, Chenlei Lv$^{2\,[0000-0002-8203-3118]}$, Guanghui Yue$^{3\,[0000-0001-9020-1735]}$, Ruomei Wang$^{4\,[0000-0002-2712-4412]}$, and Fan Zhou$^{4\,[0000-0002-0400-9366]}$

[1] School of Artificial Intelligence, Sun Yat-sen University, China
[2] College of Computer Science & Software Engineering, Shenzhen University, China
[3] School of Biomedical Engineering, Shenzhen University, China
[4] School of Computer Science, Sun Yat-sen University, China
*To whom correspondence should be addressed. zhaobaoquan@mail.sysu.edu.cn

Abstract. Text-driven human motion generation is gaining momentum lately thanks to its great potential in shaping the new pathway of interactive computer graphics in the era of AI. Despite the enormous efforts made so far, existing methods still struggle to ensure fluidity and body coordination when generating motions, which seriously hinders its application in a wide spectrum of areas such as gaming, animation, and the emerging metaverse. One of the many causes is, that learning directly from motion data is prone to interference from noise within the data, resulting in reduced quality of the generated motions. In this study, we for the first time propose to promote text-to-motion generation via out-of-distribution detection in the embedding space. Leveraging the Z-score-based outlier detection algorithm, we apply masking to motion data within the motion encoder and replace target data with means, ensuring the consistency of data distribution. To verify the effectiveness of the proposed method, we have conducted extensive experiments on the widely used KIT-ML dataset. Experimental results indicate that compared to previous frameworks, our solution significantly improves the quality of text-driven human motion generation.

Keywords: Text-driven human motion generation · Z-score · Out-of-distribution detection · Outlier rectification.

1 Introduction

With the rapid advancement of AI-generated content (AIGC) technology, text-driven human motion generation is gaining increasing interest due to its unique advantage in content creation efficiency and productivity. In contrast to old-school manual approaches with the aid of computer graphics tools, AI-powered interactive motion content creation based on text revolutionizes the experience of human-computer interaction in virtual human animation by providing the most natural and convenient pathway that is accessible to both the professionals and the wide amateur end users. This has unlocked the huge application potential of digital avatars in areas such as film, animation, gaming, media, and virtual reality.

The core mission of text-to-motion generation is to create an action sequence based on a specified descriptive text, However, previous research still falls short of motion diversity and quality. Firstly, there is a lack of diversity in the generated content, That is, under the same input conditions, similar output can be produced. For instance, for the input text "a person is running," the generated results should exhibit diversities with regard to running paths, speeds, and distances. Secondly, the quality of generated motions is far from being satisfactory for challenging textural descriptions. Intricate input text embodies a more substantial amount of action semantics and motion sequence information, posing a challenge for the model to generate high-quality motions. Additionally, existing schemes exhibit limitations when dealing with long motion sequences. Constrained by the duration of the longest motion sequences in the training data and the model's learning and generative capabilities, the quality of the generated long motions becomes increasingly unstable.

With the advent of advanced AIGC techniques such as the transformers [32], diffusion models [12, 26] and large pre-trained language models [20, 7, 25, 24], the aforementioned issues have been partially addressed. Nevertheless, the generated human motions still exhibit some deficiencies, such as limb jittering, floating, and uncoordinated limb movements. This is not because the models are incapable of accurately capturing the semantics of human motion, but because they have learned noise information from the training data in the motion embedding stage. Most of the existing datasets are derived from annotated motion capture data, in which the inherent minor noise within the data may affect the data distribution, subsequently diluting the quality of generated motion.

Outliers are data items that significantly differ from the distribution of other data in the dataset [27]. They might be genuine samples that are naturally present in the dataset or could result from errors caused by human factors such as measurement or experimentation. Within human motion data, noise might be recorded when using motion capture devices. This noise data can subsequently impact the recognition of human joint points and annotations for specific tasks, thereby degrading the quality of the dataset.

Statistical-based outlier detection methods rely on mathematical models, where the crux is to construct a probability model or distribution for a given dataset to pinpoint anomalies. Distance-based outlier detection methods determine outliers based on the distances between sample points in the dataset. In this approach, outliers are defined as points whose distance from the majority of the sample points in the dataset exceeds a certain threshold. Various distance metrics can be chosen, such as Euclidean distance, Minkowski distance, and Manhattan distance. Density-based outlier detection primarily operates on the density variations within a dataset. The algorithm segregates sample points in the dataset into different clusters based on density differences, where sparse and anomalously dense sample points are considered outliers. Cluster-based outlier detection methods partition the original dataset into different clusters based on data features. Those sample points that are not grouped with the majority and are fewer in number can be regarded as outliers.

In light of the aforementioned observation and analysis, we propose to enhance the quality of text-driven human motion generation via outlier detection. This study for the first time introduces Z-score based out-of-distribution detection and rectification to the text-driven human motion generation task. While human motion data possesses the general characteristics of sequential data, there are relatively limited feature points on a single-frame level. Consequently, distance or density is not suitable as standard metrics for feature measurement. Instead, statistical distribution assumptions can represent these data features comprehensively. Moreover, the motion encoder maps motion data to a latent space with a Gaussian distribution. Therefore, we propose to use the Z-score, a statistical-based method, as the outlier detection algorithm under the assumption that data follows a Gaussian distribution. Our method applies outlier masks to the original motion features and rectifies outliers with mean values, effectively filtering out abnormal features from the training data and denoising features during the encoding phase. Such a solution boosts the motion encoder's ability to extract meaningful features from the data, significantly improving the quality of text-based human motion generation.

2 Related work

Methods based on GANs. The GAN [9] is a classic generative model that enhances the generator's ability to approximate the real data distribution by pitting a discriminator against a generator, thereby producing realistic data samples. Text2Action [1] utilizes a sequence-to-sequence GAN model composed of a text encoder based on an RNN structure and a motion decoder. By constructing a cross-modal joint embedding, it maps text and motion into the same space, sampling from it to generate diverse human motions. DVGANs [16] assess the output of convolutional layers at each time scale and frame through a dense validation method, improving the quality of generated motions. However, due to the irregularity of human movements and the variable-length nature of temporal data, GAN-based methods face challenges in learning motion data features in training.

Methods based on autoencoders. Autoencoders learn to encode input data into low-dimensional feature vectors and then decode them to reconstruct the original data, thereby achieving the learning of data features. Lin et al. [15] proposed a sequence-to-sequence autoencoder framework that includes a text encoder based on LSTM [19] and a motion encoder based on GRU [5]. However, a simple fusion of text and motion features leads to some generated motions not being accurately aligned with text descriptions. Language2Pose [2] employs both text and motion encoders to map input text and motion into a joint embedding space and then generates human motion sequences through a motion decoder. Ghosh et al. [8] used autoencoders to separately learn the joint embedding space between the human body's upper and lower limb movements and natural language. They then generated an overlay of whole-body movements via a motion decoder, but the different limb movements did not always successfully overlay. MotionCLIP [29] adopts a Transformer-based autoencoder to align the feature space of human movement with the feature space of CLIP (contrastive language-image pretraining) [23], thus establishing a mapping relationship between text and human motion. By leveraging the prior knowledge of CLIP in the visual domain, MotionCLIP can generate human movements using abstract and stylized language. Autoencoders encode data into fixed-length vectors, which limits the cross-modal mapping of text and motion data, reducing the quality of the generated movement.

Methods based on variational autoencoders. To introduce randomness, the variational autoencoder maps input data to a predetermined prior distribution (usually Gaussian) during the encoding process, and then samples from this distribution to generate new data samples with randomness and diversity. Guo et al. [10] proposed an autoregressive text-to-motion generation framework based on variational autoencoders, utilizing the features extracted by the text encoder to autoregressively generate variable-length motion sequences. TEMOS [21] uses Transformer-based motion and text encoders to separately learn the distributions of motion and text spaces, and aligns these two spaces into a Gaussian distribution space through KL divergence. It then samples from it, generating human motion sequences through a decoder.

Traditional variational autoencoders typically use a continuous Gaussian distribution to represent the latent space of the data. However, this representation might lead to information loss, affecting the cross-modal mapping between text and action. In contrast, the Vector Quantized Variational Autoencoder (VQ-VAE) [31] uses discrete vectors to represent data in the latent space. This helps to decouple the latent space, enabling the model to learn data features more precisely. Recent studies such as TM2T [11], T2M-GPT [33], and MotionGPT [13] also try to encode motion data into "motion tokens" using VQ-VAE, transforming the text-driven motion generation task into a translation task between natural language and motion sequences. This discrete motion representation allows for a more accurate establishment of the mapping relationship between text and motion, learning the semantic coupling in text and motion data.

Methods based on diffusion models. The fundamental principle of diffusion models is to sample initial noise from a known simple distribution (usually a Gaussian distribution), and incrementally add noise during the diffusion process to approximate the probability distribution of the target data, and finally produce high-quality samples by progressively removing the noise [12, 26]. MotionDiffuse [34] is the first text-to-motion generation method based on diffusion models, capable of producing motions of any desired duration. Through "noise interpolation," it achieves fine-grained control between different body parts. MoFusion [6], FLAME [14], and ReMoDiffuse [35] all utilize Transformer-based encoders and diffusion model architectures to achieve diverse, high-quality, and variable-length motion generation. While diffusion models can produce high-quality and diverse motions, their training process is generally time-consuming and computationally intensive. Moreover, controlling noise during the diffusion process is challenging, leading to potential issues in generated results such as uncoordinated movements and limb clipping. Some research has optimized diffusion models to address these concerns. MDM [30] makes the generation process of diffusion models more controllable by predicting the motion sequence itself during the denoising process. MLD [4] executes the diffusion process on the latent code of low-dimensional motions in the latent space, significantly reducing computational overhead during the training and inference phases, making it two orders of magnitude faster than other diffusion models that operate directly on the original motion sequences.

The core focus of current research on text-driven human motion generation is the cross-modal fusion of text features and motion features. Researchers generally adopt attention mechanisms [32], diffusion models [12, 26], and pre-trained models to improve the quality of generated motions. However, current methods still struggle to effectively handle anomalous data. Even if the model has a strong learning capacity, it can easily be disrupted by abnormal features, thus affecting the quality of the generated motion. Therefore, this study is dedicated to promoting text-driven motion generation with the newly developed out-of-distribution detection and rectification scheme.

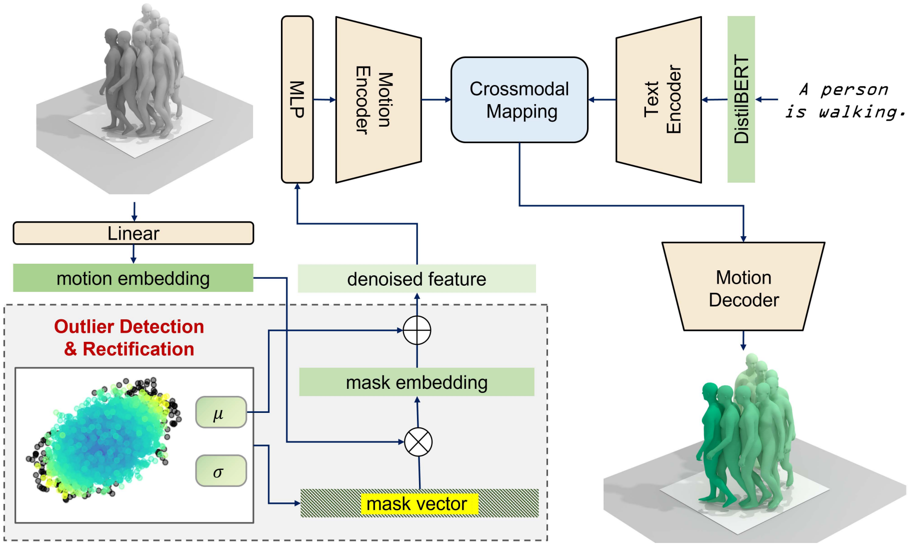


Fig. 1: Framework of the proposed text-driven human motion generation method via out-of-distribution detection and rectification. First, the motion data is mapped through a linear layer to produce a motion embedding. Next, outlier detection algorithms are used to identify values with anomalous distributions, and denoising of the input features is achieved through masking and mean replacement. The data then undergoes a non-linear feature transformation via a Multi-Layer Perceptron (MLP). The denoised features are fed into the motion encoder, while the text encoder provides textual features, facilitating cross-modal feature fusion between the two. Finally, samples are drawn from the joint embedding space, and a motion sequence is generated via the motion decoder.

3 The proposed method

The framework of the proposed text-driven human motion generation method is shown in Figure 1. A core idea of our method is to detect outliers in the motion data and then rectify the features to improve the quality of generated motions. In our work, TEMOS [21] serves as the foundation architecture, which learns the distribution of the motion space from motion data through a Transformer-based variational autoencoder. Details of the proposed pipeline are as follows.

Firstly, we identify outliers in the motion data and perform rectification using the proposed masking scheme before being fed into the motion encoder. The selection of outlier detection algorithms can have an influence on the quality of generated content. As aforementioned, there are various outlier detection algorithms in the literature such as the density-based method Local Outlier Factor (LOF) [3], the machine learning-based method Isolation Forest [17], and the statistical method Z-score. In our implementation, we adopted Z-score, a.k.a. the standard score, as the out-of-distribution detection algorithm. The reason is that the Z-score assumes that the data is normally distributed, which is in line with the distribution of data features from the motion encoder. The principle of using the Z-score for outlier detection is that the larger the absolute value of a data point's Z-score, the greater its deviation from the sample mean. If the Z-score exceeds the set outlier threshold, then that data point is considered an outlier. The calculation formula is as follows:

$$\mathrm{Z} = \frac{x^M - \mu^M}{\sigma^M} \tag{1}$$

Where Z represents the Z-score value, x^M denotes the motion data, μ^M is the mean of motion, and σ^M is the standard deviation of motion.

The purpose of outlier masking is to retain non-outlier data in the input. This is achieved by generating a binary tensor of the same dimension as the input data to indicate the status of each data point. In this tensor, positions with a value of 0 correspond to data identified as outliers, while positions with a value of 1 indicate normal data points. This binary mask is then multiplied with the original data, setting all data positions flagged as outliers to 0, while the values of normal data points remain unchanged.

Secondly, to prevent the removed outliers from affecting subsequent computations and to maintain the Gaussian distribution characteristics of the data, the values at the positions of outliers are replaced with the mean value of the original motion data. This is done by adding a mean tensor that is multiplied by the inverse mask, where 1 becomes 0 and 0 becomes 1.

Lastly, the processed data undergoes a nonlinear transformation through a multi-layer perceptron to compensate for potential information loss during outlier processing, aiming to approximate the latent features present in the original data as closely as possible. Figure 2 visualizes the out-of-distribution detection and rectification. Although the outlier detection algorithm based on Z-score is relatively straightforward in principle, it exhibited superior performance compared to LOF and Isolation Forest. This can be attributed to the Z-score algorithm's assumption that data follows a normal distribution, and in the text-to-motion generation framework, the Actor encoder also maps motion data features to a space with a normal distribution.

For the input text, we use the pre-trained language model DistilBERT [25] to transform it into text features, which are then passed to the text encoder to learn the latent space of the text. Subsequently, a cross-modal fusion is performed with the latent space output from the motion encoder. The specific method aligns these two latent spaces into the shared feature space using KL divergence. Afterward, we can randomly sample from this shared feature space and generate motion sequences via the motion decoder.

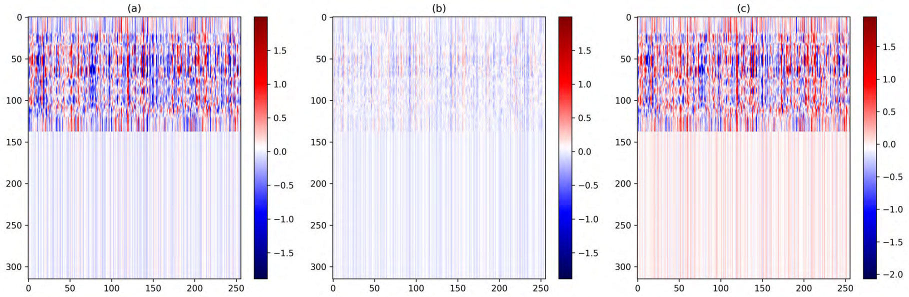


Fig. 2: Visualization of out-of-distribution detection and rectification. (a) the original motion features; (b) the denoised motion features; and (c) the removed noise features. The x-axis and y-axis represent the latent space dimension (256) and the number of frames in motion data, respectively. The varying shades of color indicate the degree of deviation from the normal values. It can be observed that there is a significant difference between the motion features before and after denoising. The denoised motion features show a smoother color distribution compared to the original motion features.

Following TEMOS [21], the total loss is composed of three terms as follow:

$$\mathcal{L}_{tot} = \mathcal{L}_{gen} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{cme}\mathcal{L}_{cme}, \tag{2}$$

where $\mathcal{L}_{gen}$ is used to measure the consistency between the generated motion and the groud-truth one, $\mathcal{L}_{KL}$ is designed to minimize the Kullback-Leibler (KL) divergences between the text embedding and motion embedding, and $\mathcal{L}_{cme}$ is the cross-modal embedding similarity loss.

4 Experiments

4.1 Dataset and evaluation metrics

Dataset. In this paper, we use the widely-used public KIT-ML (Motion Language) [22] as our dataset for experimental studies. This dataset was constructed using the motion capture data processing framework, the Master Motor Map (MMM) [28], which builds standardized human models from motion capture data and appends textual descriptions to these motions. The dataset comprises 111 distinct subjects, with a total of 3,911 motion actions that cover a combined duration of 11.23 hours. In addition, there are 6,278 text annotations (52,903 words in total) in this dataset.

Evaluation Metrics. To evaluate the effectiveness of the proposed method, we adopted the same metrics as those in [21], including Average Positional Error (APE) and Average Variance Error (AVE). These two metrics assess the quality of the generated motion by calculating the average positional error and average variance error between the generated motion and the ground truth motion at corresponding joints, respectively. Formally, we calculate the APE $\mathcal{L}_j^{APE}$ and AVE $\mathcal{L}_j^{AVE}$ of the j-th joint as follows [21]:

$$\mathcal{L}_j^{APE} = \frac{1}{N_s N_f} \sum_{n \in N_s} \sum_{f \in N_f} \left\| \mathbf{M}_j^f - \hat{\mathbf{M}}_j^f \right\|_2, \tag{3}$$

$$\mathcal{L}_j^{AVE} = \frac{1}{N_s} \sum_{n \in N_s} \| \sigma_j - \hat{\sigma}_j \|_2, \tag{4}$$

where

$$\sigma_j = \frac{1}{N_f - 1} \sum_{f \in N_f} \left(\mathbf{M}_j^f - \tilde{\mathbf{M}}_j^f \right)^2 \in \mathbb{R}^3. \tag{5}$$

Here, N_s and N_f represent the total numbers of samples and time frames, respectively; n means the sample index in N_s and f means the frame index in N_f; $\mathbf{M}$ represents the motion sequence while $\hat{\mathbf{M}}$ is the average value of the joint over time.

We evaluated the AVE and APE of root joint error, global trajectory error, mean local error, and mean global error. Among them, the root joint error means the error of the human body's root joint in the 3D Cartesian coordinate system. The global trajectory error represents the global trajectory error of the human body's root joint in the XOY plane coordinates during movement, assessing the trajectory accuracy of human movement on the horizontal plane. Mean local error indicates the average error of joints in the human body's local coordinate system, used to evaluate whether the positions of various parts relative to the root joint are accurate. Mean global error represents the average error of joints in the human body's global coordinate system, assessing the position accuracy of all joints in the global coordinate system. The smaller these evaluation values, the higher the quality of the generated motion.

4.2 Experiment configuration and training details

For a fair comparison, we adopted the same experiment settings as TEMOS with regard to data preprocessing, training parameters, and evaluation. All experiments were conducted on a workstation equipped with a 12th Gen Intel(R) Core(TM) 9-12900K CPU, with CUDA 12.2 supporting an NVIDIA GeForce RTX 4090 GPU, and running Ubuntu 20.04 as its operating system.

Our encoders and decoders all consist of 6 Transformer layers, with each layer utilizing 6 multi-head attention heads. The dimension of the intermediate layer is set at 1024, and a dropout rate of 0.1 is applied. For model training, we employed the AdamW optimizer with a learning rate of 0.0001 and a batch size of 32, conducting training over 1000 epochs. Figure 3 illustrates the training loss curves with regards to the APE and AVE of root joint and mean pose.

4.3 Comparisons between different text-driven human motion generation methods

To demonstrate the superiority of the proposed method, we compared it with three peer models, including Lin et al. [15], Language2Pose [2] and Ghosh et al. [8]. For the methods of Lin et al., Language2Pose, and Ghosh et al., we adopted the open-source

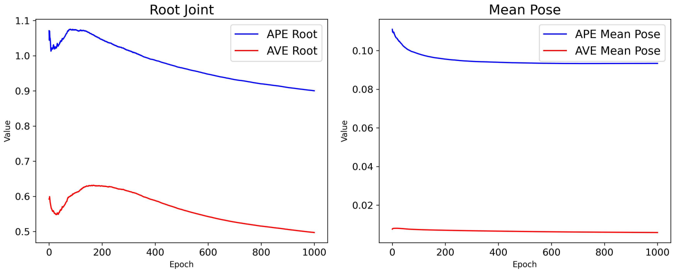


Fig. 3: The training loss curves with regards to the APE and AVE of root joint and mean pose.

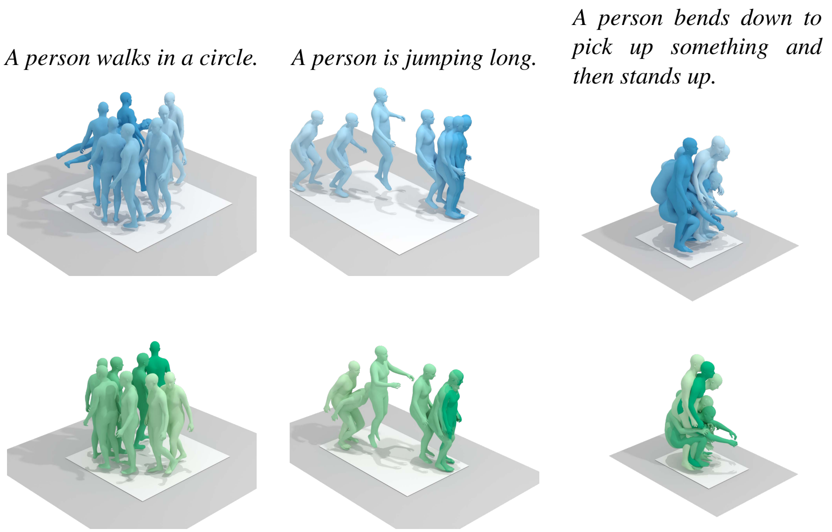
Table 1: Comparison between different text-driven human motion generation methods.

Methods	Average Positional Error $\mathcal{L}^{APE}$ ↓				Average Variance Error $\mathcal{L}^{AVE}$ ↓			
	root joint	global traj	mean local	mean global	root joint	global traj	mean local	mean global
Ghosh [8]	1.291	1.242	0.206	1.294	0.564	0.548	0.024	0.563
Language2Pose [2]	1.622	1.616	**0.097**	1.630	0.669	0.669	0.006	0.672
Lin [15]	1.966	1.956	0.105	1.969	0.790	0.789	0.007	0.791
TEMOS [21]	1.153	1.144	0.105	1.166	0.534	0.533	**0.005**	0.537
Ours	**1.040**	**1.031**	0.105	**1.054**	**0.453**	**0.453**	**0.005**	**0.456**

evaluation code provided by TEMOS [21], ensuring that the results are consistent with the data presented in the original papers of TEMOS.

As shown in Table 1, the proposed method shows significant performance gains in terms of most of the selected evaluation metrics. Specifically, for the metric of APE $\mathcal{L}^{APE}$, our method reduces the APEs with regards to root joint, global traj, and mean global by 9.8%, 9.9%, and 9.6%, respectively, compared to TEMOS [21]. While for the mean local APE, Language2Pose [2] achieves the best performance among all peer methods. In addition, compared to TEMOS, the proposed method lowers the APEs $\mathcal{L}^{AVE}$ with regards to root joint, global traj, and mean global by 15.2%, 15.0%, and 15.1%, respectively, and is comparable to TEMOS in terms of mean local. These experimental results demonstrate the superiority of the proposed method in text-driven human motion generation.

To further investigate the efficacy of different methods, we have also carried out a qualitative study and compared the generated motions using our method with those using the state-of-the-art TEMOS. Figure 4 illustrates three groups of generated motion sequences using TEMOS (rendered in blue, top row) and our method (rendered in green, bottom row). To showcase the details of human motion, we utilized the human parametric model SMPL [18]. The different shades of color are used to visualize the movement over time of a 3D human model where lighter blue or green color indicates the early stage of a motion. For the instance of *"A person walks in a circle"*, the result generated by TEMOS has issues with abnormal body positioning. In the example of *"A person is jumping long"*, our method depicts the process of a person squatting down, gathering strength, and jumping up more accurately. For the case of *"A person bends*



A person walks in a circle.

A person is jumping long.

A person bends down to pick up something and then stands up.

Fig. 4: Visual comparison between the generated motion sequences using TEMOS (rendered in blue, top row) and our method (rendered in green, bottom row). The different shades of color are used to visualize the movement over time of a 3D human model where lighter blue or green color indicates the early stage of a motion.

down to pick up something and then stands up", the action is expected to be completed in place, but the motion generated by TEMOS exhibits a significant displacement from the lighter frames to the darker ones. These results further verify the superiority of the proposed method.

4.4 Comparison between different outlier detection algorithms

To evaluate the performance of different outlier detection algorithms, we conducted an experiment comparing the Actor encoder from the original TEMOS [21] with the Actor encoder integrated with various outlier detection algorithms. Given that the Actor encoder maps motion data to a normal distribution space, and in a normal distribution, approximately 95% of the data lies within two standard deviations from the mean, we uniformly set the outlier threshold for the outlier detection algorithms at 5%. For fair comparison, we adopted the same random seed for all methods in each round of test, and performed the experiments with 10 different seeds for the average. The experimental results are shown in Table 2.

From the experimental results, it can be observed that on both APE and AVE metrics, the performance of the Actor+Z-score and Actor+LOF methods surpassed the original Actor method. Among the four methods, Actor+Z-score exhibited the best performance, showing the lowest errors in all evaluation metrics except for "mean local AVE". Actor+iForest only had a slight improvement over Actor in the AVE metric. Therefore, this study ultimately chose Z-score as the outlier detection algorithm.

Table 2: Comparison between different outlier detection methods.

Methods	Average Positional Error $\mathcal{L}^{APE} \downarrow$				Average Variance Error $\mathcal{L}^{AVE} \downarrow$			
	root joint	global traj	mean local	mean global	root joint	global traj	mean local	mean global
Actor	1.153	1.144	**0.105**	1.166	0.534	0.533	0.005	0.537
Actor+Z-score	**1.074**	**1.065**	**0.105**	**1.087**	**0.466**	**0.465**	0.005	**0.469**
Actor+LOF	1.113	1.104	**0.105**	1.126	0.486	0.485	0.005	0.489
Actor+iForest	1.196	1.187	0.107	1.210	0.531	0.530	0.005	0.533

4.5 Evaluation of different thresholds for outlier detection

In statistics, the Z-score indicates how many standard deviations a data point deviates from the population mean. To investigate the impact of different Z-score thresholds for outlier detection on motion generation quality, we empirically selected 1.00, 1.96, 2.00, and 3.00 as the test conditions. The resultant outlier proportions using these three thresholds are 31.73%, 5%, 4.55%, and 0.27%, respectively, in which the proportion of 5% is widely used in previous studies, and thus was also included in our evaluation. The experimental results are shown in Figure 5. As can be seen from the figure, the performance with regards to the APE and AVE metrics is comparable with 1.00, 1.96, and 2.00 as the thresholds. However, when the Z-score threshold is set to 3, we can observe a noticeable performance gain in all eight groups. This is in line with the assumption that in a normal distribution, outliers are distributed beyond three standard deviations from the mean.

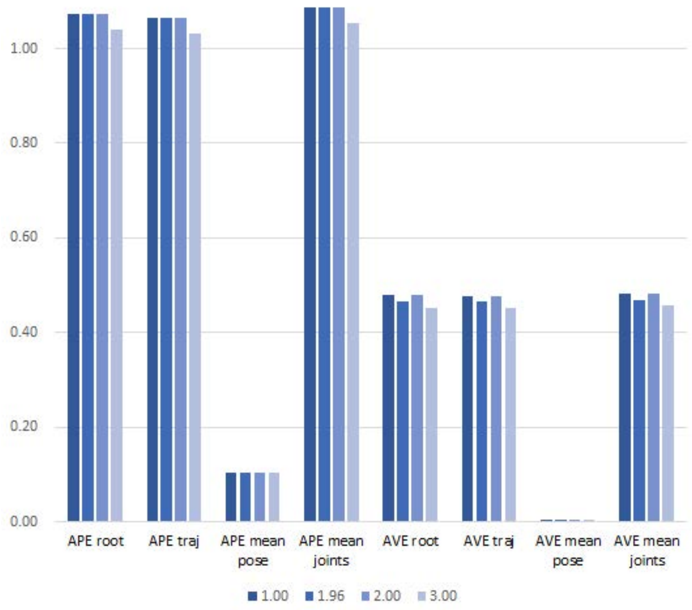


Fig. 5: Evaluation of different thresholds for outlier detection.

Table 3: Ablation study.

Methods	Average Positional Error $\mathcal{L}^{APE} \downarrow$				Average Variance Error $\mathcal{L}^{AVE} \downarrow$			
	root joint	global traj	mean local	mean global	root joint	global traj	mean local	mean global
Z-score	1.072	1.063	0.105	1.085	0.489	0.488	0.005	0.491
Z-score+Linear	1.140	1.131	0.105	1.153	0.498	0.498	0.005	0.501
Z-score+MLP	**1.040**	**1.031**	0.105	**1.054**	**0.453**	**0.453**	0.005	**0.456**

4.6 Ablation study

During the process of integrating the outlier detection algorithm into the action encoder, we employed a Multi-Layer Perceptron (MLP) to perform a nonlinear transformation on the data. To verify the effectiveness of this structure, we designed three test conditions, including the standalone outlier detection algorithm "Z-score", the outlier detection algorithm with an added linear layer "Z-score+Linear", and the outlier detection algorithm with an added Multi-Layer Perceptron "Z-score+MLP". Table 3 shows that with regard to both APE and AVE metrics, the Z-score+MLP method consistently achieves the lowest errors. In contrast, the Z-score+Linear method exhibited relatively higher errors in multiple subcategories, especially in terms of average positional error, while the performance of the Z-score method was in between. This ablation study validates the effectiveness of the use of an MLP module in our framework.

5 Conclusion

In this paper, we have proposed an improved text-driven human motion generation method via out-of-distribution detection and rectification. It aims to boost the quality of generated motions by integrating outlier detection algorithms within the motion encoder for the first time. Extensive experiment results on the KIT-ML dataset demonstrate that compared to peer methods, the proposed Z-score-based outlier detection and rectification solution significantly enhances the quality of generated motions. Future work will focus on adaptive adjustments to different data distributions and parameters of the outlier detection algorithm.

Acknowledgments. This study was funded by Natural Science Foundation of Guangdong Province (2023A1515011639), National Key R&D Program of the Ministry of Science and Technology, China (2022YFF0903103); Natural Science Foundation of China (62371305), and Fundamental Research Funds for the Central Universities, Sun Yat-sen University (23xkjc019).

Disclosure of Interests. None.

References

1. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. In: 2018 IEEE International Conference on Robotics and Automation. pp. 5915–5920. IEEE (2018)

2. Ahuja, C., Morency, L.P.: Language2pose: Natural language grounded pose forecasting. In: 2019 International Conference on 3D Vision. pp. 719–728. IEEE (2019)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
4. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18000–18010 (2023)
5. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Dabral, R., Mughal, M.H., Golyanik, V., Theobalt, C.: Mofusion: A framework for denoising-diffusion-based motion synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9760–9770 (2023)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Synthesis of compositional animations from textual descriptions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1396–1406 (October 2021)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
10. Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., Cheng, L.: Generating diverse and natural 3d human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5152–5161 (June 2022)
11. Guo, C., Zuo, X., Wang, S., Cheng, L.: Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In: European Conference on Computer Vision. pp. 580–597. Springer (2022)
12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
13. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. arXiv preprint arXiv:2306.14795 (2023)
14. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis & editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8255–8263 (2023)
15. Lin, A.S., Wu, L., Corona, R., Tai, K., Huang, Q., Mooney, R.J.: Generating animated videos of human activities from natural language descriptions. Learning **2018**(1) (2018)
16. Lin, X., Amer, M.R.: Human motion modeling using dvgans. arXiv preprint arXiv:1804.10652 (2018)
17. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)
18. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866 (2023)
19. Memory, L.S.T.: Long short-term memory. Neural computation **9**(8), 1735–1780 (2010)
20. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., Roth, D.: Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys **56**(2), 1–40 (2023)
21. Petrovich, M., Black, M.J., Varol, G.: Temos: Generating diverse human motions from textual descriptions. In: European Conference on Computer Vision. pp. 480–497. Springer (2022)

22. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. Big data **4**(4), 236–252 (2016)
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
24. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
25. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
26. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
27. Souiden, I., Omri, M.N., Brahmi, Z.: A survey of outlier detection in high dimensional data streams. Computer Science Review **44**, 100463 (2022)
28. Terlemez, Ö., Ulbrich, S., Mandery, C., Do, M., Vahrenkamp, N., Asfour, T.: Master motor map (mmm)—framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In: 2014 IEEE-RAS International Conference on Humanoid Robots. pp. 894–901. IEEE (2014)
29. Tevet, G., Gordon, B., Hertz, A., Bermano, A.H., Cohen-Or, D.: Motionclip: Exposing human motion generation to clip space. In: European Conference on Computer Vision. pp. 358–374. Springer (2022)
30. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022)
31. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., Shen, X.: T2m-gpt: Generating human motion from textual descriptions with discrete representations. arXiv preprint arXiv:2301.06052 (2023)
34. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. arXiv preprint arXiv:2208.15001 (2022)
35. Zhang, M., Guo, X., Pan, L., Cai, Z., Hong, F., Li, H., Yang, L., Liu, Z.: Remodiffuse: Retrieval-augmented motion diffusion model. arXiv preprint arXiv:2304.01116 (2023)