

# 2D Motion Generation Using Joint Spatial Information with 2CM-GPT

Ryota Inoue<sup>a</sup>, Tsubasa Hirakawa<sup>b</sup>, Takayoshi Yamashita<sup>c</sup> and Hironobu Fujiyoshi<sup>d</sup>

Chubu University, 1200 Matsumoto-cho, Kasugai, Aichi, Japan  
{inoue, hirakawa}@mprg.cs.chubu.ac.jp, {takayoshi, fujiyoshi}@isc.chubu.ac.jp

**Keywords:** Deep Learning, 2D Motion Generation, Variational AutoEncoder, Language Model.

**Abstract:** Various methods have been proposed for generating human motion from text due to advancements in large language models and diffusion models. However, most research has focused primarily on 3D motion generation. While 3D motion enables realistic representations, the creation and collection of datasets using motion-capture technology is costly, and its application to downstream tasks, such as pose-guided human video generation, is limited. Therefore, we propose 2D Convolutional Motion Generative Pre-trained Transformer (2CM-GPT), a method for generating two-dimensional (2D) motion from text. 2CM-GPT is based on the framework of MotionGPT, a method for 3D motion generation, and uses a motion tokenizer to convert 2D motion into motion tokens while learning the relationship between text and motion using a language model. Unlike MotionGPT, which utilizes 1D convolution for processing 3D motion, 2CM-GPT uses 2D convolution for processing 2D motion. This enables more effective capture of spatial relationships between joints. Evaluation experiments demonstrated that 2CM-GPT is effective in both motion reconstruction and text-guided 2D motion generation. The generated 2D motion is also shown to be effective for pose-guided human video generation.

## 1 INTRODUCTION

The task of generating human motion from text is gaining attention due to its potential applications in fields such as games, movies, virtual reality, and augmented reality. With significant advancements in large language models (LLMs) and diffusion models, various methods for generating human motion from text have been proposed, e.g., methods based on LLMs, such as Generative Pre-trained Transformer (GPT) including MotionGPT (Jiang et al., 2024) and T2M-GPT (Zhang et al., 2023), and those using diffusion models such as motion diffusion model (MDM) (Tevet et al., 2023), MotionDiffuse (Zhang et al., 2024), and motion latent diffusion (MLD) (Chen et al., 2023).

Conventional human-motion-generation methods mainly focus on generating three-dimensional (3D) motions using natural-language texts describing motions as input. These methods are trained on a skinned multi-person linear (SMPL) model (Loper et al., 2015) motion-text-pair dataset to generate human motions. However, 3D motion generation

incurs high dataset-creation costs. Typically, 3D-human-motion data are created and collected using motion-capture technology, but this requires dedicated motion-capture equipment, studios, and actors. This also requires specialized knowledge for setup and post-processing. Applications such as pose-guided human-video-generation tasks also often require 2D motions as input. Therefore, it is necessary to convert 3D motions to 2D motions. In summary, conventional methods focusing on 3D motion generation incur high cost of creating datasets and are inefficient when applied to 2D applications.

We propose 2D Convolutional Motion Generative Pre-trained Transformer (2CM-GPT), a method for generating 2D motion from text. By focusing on 2D motion, datasets can be created more easily using techniques such as pose estimation, compared with 3D motion, enabling direct application of the generated 2D motion to various applications. Using text as input also enables intuitive motion generation. Our method consists of a motion tokenizer for processing motion as text and a language model for learning the relationship between text and motion. The motion tokenizer uses a vector quantised variational autoencoder (VQ-VAE) (Van Den Oord et al., 2017) to convert human 2D motion into discrete motion tokens. This enables handling motion in a feature space sim-

<sup>a</sup> <https://orcid.org/0009-0006-1546-3990>

<sup>b</sup> <https://orcid.org/0000-0003-3851-5221>

<sup>c</sup> <https://orcid.org/0000-0003-2631-9856>

<sup>d</sup> <https://orcid.org/0000-0001-7391-4725>

ilar to text. The language model uses a pre-trained text-to-text transfer Transformer (T5) (Raffel et al., 2020) to learn motion as language. This enables the generation of diverse 2D motions on the basis of different texts. We converted the HumanML3D (Guo et al., 2022) dataset, which pairs text with 3D motion data and is widely used in text-to-motion tasks, into 2D-motion data for model training.

Our contributions are as follows.

- We propose a generation method for modifying the convolution method of motion tokenizer, a conventional method that relies on 3D data, to enable 2D motion generation from text.
- The 2D motion generated with our method was applied to a pose-guided human-video-generation method to demonstrate the effectiveness of our method.

## 2 RELATED WORK

In this section, we introduce datasets that pair text with 3D motion and methods for generating 3D motion from text.

### 2.1 3D Human Motion-Language Dataset

Datasets that pair human motion with text annotations play a crucial role in text-to-motion tasks. These datasets are essential for training models that generate motion based on natural-language descriptions, and the quality and diversity of the datasets significantly impact the performance of generation models. Representative datasets widely used in text-to-motion tasks include HumanML3D and the KIT Motion-Language (ML) dataset (Plappert et al., 2016).

HumanML3D is a combined dataset of HumanAct12 (Guo et al., 2020) and the Archive of Mocap as Surface Shapes (Mahmood et al., 2019), comprising 14,616 motions and 44,970 text annotations. This dataset includes everyday motions covering a wide range of actions, such as walking and jumping; sports motions, such as swimming and karate; acrobatic motions, such as rotating; and artistic motions, such as dancing. Therefore, HumanML3D is a versatile dataset that can be used in various contexts and is frequently used as a benchmark for diverse motion-generation tasks. The KIT-ML dataset consists of 3,911 motions and 6,353 text annotations, including more detailed motions, e.g., gesture motions, such as pointing and waving; movements, such as walking and crawling; manipulations, such as throwing and

wiping; and sports motions such as martial arts and tennis.

### 2.2 Text-Guided 3D Motion Generation

Text-guided 3D motion generation aims to generate human motion in 3D space using natural-language text as input. Text2Action (Ahn et al., 2018) uses a recurrent neural network (Graves and Graves, 2012)-based sequence-to-sequence (Sutskever et al., 2014) architecture to generate motion from text. Joint language to pose (Ahuja and Morency, 2019) combines a text encoder based on long short-term memory (Hochreiter and Schmidhuber, 1997) and a motion encoder-decoder using a gated recurrent unit (Cho et al., 2014) that learns text and motion in a shared feature space for more accurate motion generation. MotionCLIP (Tevet et al., 2022) leverages the text-image latent space of contrastive language-image pre-training (Radford et al., 2021), enabling flexible and broad mappings from language to motion. This approach is particularly adaptable to diverse instructions given in language, contributing to a wide range of motion-generation tasks. TEMOS (Petrovich et al., 2022) uses a combination of a VAE and Transformer (Vaswani et al., 2017) for sequence-level embeddings of text and motion, generating diverse motion. MLD, a method based on diffusion models, executes diffusion processes in the latent space of motion to generate plausible human-motion sequences. MotionGPT uses a motion tokenizer using a VQ-VAE to build a motion vocabulary and an LLM to learn the relationship between text and motion vocabularies. This enables MotionGPT to handle multiple motion tasks such as text-to-motion, motion-to-text, motion prediction, and motion in-between in a unified framework.

In summary, research in text-to-motion is currently active and has made significant progress through these previous studies. However, most of these methods focus only on 3D motion generation, and research on 2D motion generation has not progressed much.

## 3 PROPOSED METHOD

In this section, we introduce 2CM-GPT, our method for generating 2D motion from text.

### 3.1 Overview

Previous methods focusing on 3D motion generation incur high costs for dataset creation and are inefficient when applied to 2D applications. Thus, 2CM-GPT

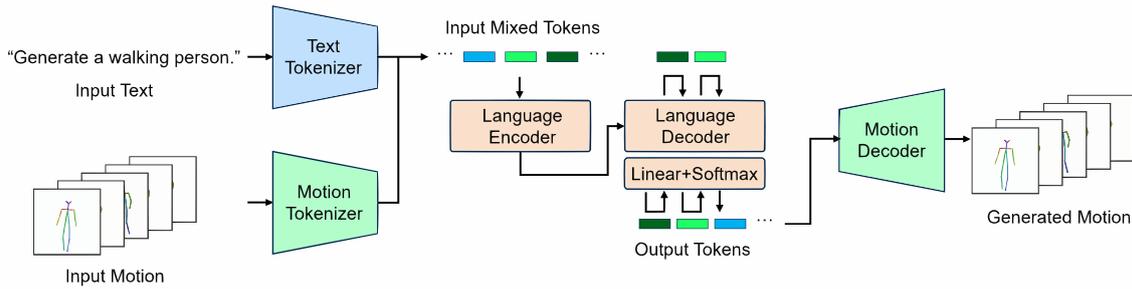


Figure 1: 2CM-GPT architecture. The model consists of a motion tokenizer for processing motion as text and a language model for learning the relationship between text and motion. The input to the language model is a text-motion vocabulary that combines two types of tokens: text tokens that describe motion and motion tokens that represent 2D motion.

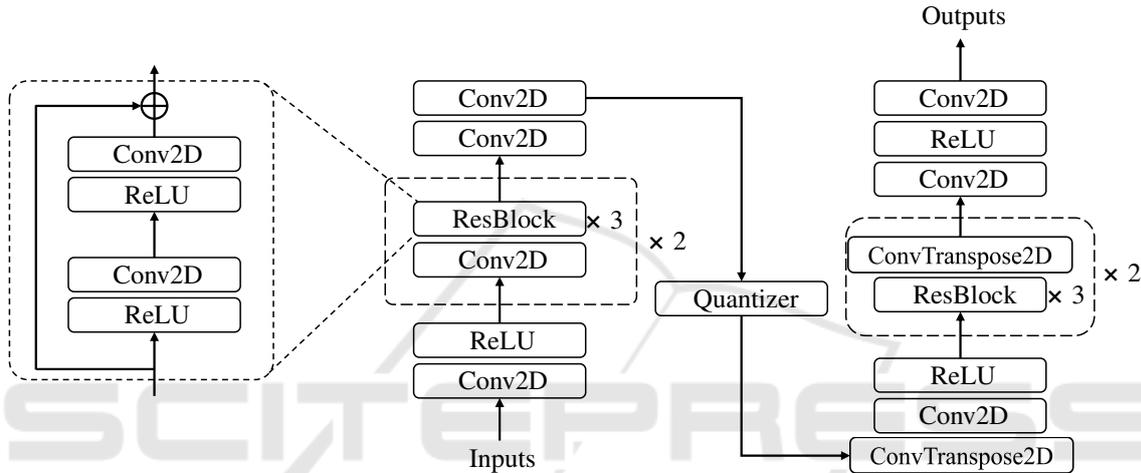


Figure 2: Motion tokenizer architecture. The model uses a CNN-based architecture with 2D convolution (Conv2D), residual block (ResBlock), ReLU activation, and nearest neighbor interpolation (Upsample).

was developed for generating 2D motion from text. By focusing on 2D motion, it becomes possible to create datasets using methods, such as pose estimation, enabling easier and more precise data creation compared with focusing on 3D motion. Using text as input enables intuitive generation, and the generated 2D motion can be directly applied to 2D applications. The architecture of 2CM-GPT is shown in Figure 1. 2CM-GPT is based on the framework of the 3D-motion-generation method MotionGPT and consists of two main components: a motion tokenizer and language model, with modifications to the convolution method in the motion tokenizer and training method of the language model. The next sections describe these components: Section 3.2 explains the architecture, role, and training of the motion tokenizer, and Section 3.3 discusses the role and training of the language model when learning the relationship between text and motion.

### 3.2 Motion Tokenizer

The motion tokenizer in 2CM-GPT uses a VQ-VAE architecture to build a vocabulary of 2D motion. VQ-VAE is an extension of VAE with a discrete latent space and consists of an encoder-decoder architecture. The motion tokenizer first uses an encoder to estimate latent variables from the input motion. It then quantizes the latent variables using a codebook of discrete vectors. Specifically, it finds the nearest vector  $b_k$  in the codebook  $B := \{b_k\}_{k=1}^K$  for each latent variable  $z_e$  output by the encoder. The codebook is constructed with  $K$  latent variable vectors of dimension  $d$ . The vector quantization is shown in Equation (1):

$$z_q = b_k, \text{ where } k = \operatorname{argmin}_k \|z_e - b_k\|_2 \quad (1)$$

Next, the quantized latent variables are used by the decoder to reconstruct the input motion.

With 2CM-GPT, we change the convolution used in the encoder and decoder of the motion tokenizer from 1D convolution to 2D convolution. While 3D motion data are represented in the same dimension

for joint positions, velocities, and angles, and MotionGPT applies 1D convolution, 2D motion data represent joints and xy-coordinates in two different dimensions. Applying 1D convolution after aggregating these into a single dimension fails to properly model the relationships between joints. Instead, by using 2D convolution while keeping the two dimensions separate, 2CM-GPT can directly account for the spatial relationships between joints. The architecture of the motion tokenizer and motion decoder is shown in Figure 2.

The motion tokenizer in MotionGPT uses three different loss functions for training: reconstruction loss, embedding loss, and commitment loss. Of these losses, embedding loss is computed by extracting joint velocity information from the embedding. However, 2CM-GPT uses 2D motion, which is represented only by the coordinates of the joints, thus has no velocity information. Therefore, we use two different loss functions for training: reconstruction loss and commitment loss. To improve codebook utilization, we use the exponential moving average and the codebook-resetting technique (Razavi et al., 2019). The L1 smooth loss is used for reconstruction loss.

### 3.3 Language Model

2CM-GPT uses a pre-trained T5 model to learn the relationship between text tokens and motion tokens generated by the motion tokenizer. The input to the language model is a text-motion vocabulary that combines two types of tokens: text tokens that describe motion and motion tokens that represent 2D motion. Depending on the task, the text-motion vocabulary can represent text tokens, motion tokens, and tokens that represent both text and motion, enabling MotionGPT to generate diverse and flexible text and motion.

The language model in MotionGPT involves pre-training that learns the relationship between text tokens and motion tokens and instruction tuning for various motion-related tasks. During pre-training, 15% of the input tokens are randomly replaced with special sentinel tokens, and the model learns to generate the corresponding tokens in the output. It also learns the relationship using paired text and motion tokens. For instruction tuning, prompts for tasks, such as text-to-motion, motion-to-text, motion prediction, and motion in-between, are used. However, to achieve higher accuracy in the text-to-motion task, we fine-tune the language model in 2CM-GPT using only the instruction prompts for that task. The instruction prompts used as input are those for text-to-motion from MotionGPT.

## 4 EXPERIMENT

We quantitatively and qualitatively evaluated the effectiveness of 2CM-GPT by comparing the reconstructed 2D motion using the motion tokenizer and the motion generation guided by text. We also applied the 2D motion generated by 2CM-GPT to the downstream task of pose-guided human video generation and qualitatively assessed the accuracy of the generated 2D motion.

### 4.1 2D Human-Motion-Language Dataset

Text-guided 2D motion generation requires a dataset consisting of pairs of 2D motion and text, but no suitable dataset currently exists. Therefore, creating a dataset is a major challenge, and it is essential to convert datasets into a new data format. For human video generation, a 2D motion with 18 joints based on the Common Objects in Context (COCO) (Lin et al., 2014) dataset is widely used, so a dataset based on this format is desirable. Therefore, we used a 3D motion dataset to create a 2D motion dataset in the COCO format. Specifically, we converted 3D motion to 2D motion on the basis of HumanML3D, which consists of pairs of text and 3D motion. The conversion procedure begins with estimating an SMPL mesh from the 3D motion data included in HumanML3D using the SMPL model. The estimated SMPL mesh is then multiplied using a regressor matrix to obtain the coordinates of 18 joints on a 2D plane. This method enables us to obtain accurate joint information based on the COCO format. Our 2D motion dataset was constructed on the basis of the text, type of motion, and length of the action, similar to HumanML3D. This enables the generation of diverse 2D motions guided by text.

### 4.2 Experimental Settings

In this experiment, we used the 2D Human Motion-Language dataset as the training and evaluation dataset. To enhance the diversity of the dataset, we applied left-right mirrored data augmentation to all motions and specific texts (such as "right hand," "left hand," "right foot," "left foot"), effectively doubling the dataset size. For training, we used 23,384 motions, 1,460 for validation and 4,384 for evaluation.

For training the motion tokenizer, the batch size was set to 256, training iterations to 3,000 epochs, learning rate to  $2 \times 10^{-4}$ , and the codebook was  $128 \times 512$ . For pre-training and fine-tuning the language model, the batch size was set to 16, with 300

epochs for pre-training, 100 epochs for fine-tuning, and a learning rate of  $1 \times 10^{-4}$ . The optimizer used for all training processes was AdamW.

### 4.3 Evaluation Metrics

We evaluated the accuracy of the motion tokenizer’s 2D motion reconstruction with 2CM-GPT using mean per joint position error (MPJPE), Fréchet inception distance (FID) (Heusel et al., 2017), and diversity (Guo et al., 2020).

MPJPE measures the Euclidean distance between the generated joints and ground-truth joints, taking the mean of these distances. A value closer to 0 indicates that the generated motion is accurate. MPJPE is useful for measuring the reconstruction accuracy of motion because it directly reflects the joint-position errors. MPJPE is expressed as

$$MPJPE = \frac{1}{N} \sum_{i=1}^N \|j_i - j'_i\|_2 \quad (2)$$

where  $N$  is the number of joints,  $j_i$  is the ground-truth-joint coordinates, and  $j'_i$  is the predicted joint coordinates.

FID measures the Fréchet distance between the distribution of generated motions and that of ground-truth motions, with values closer to 0 indicating that the distribution of generated motions is close to the ground-truth distribution. FID is useful for evaluating the overall quality and realism of generated motions. FID is expressed as

$$FID = \|\mu - \mu'\|_2^2 + \text{Tr}(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{1/2}) \quad (3)$$

where  $\mu$  and  $\Sigma$  respectively represent the mean and covariance of the ground-truth-motion distribution, and  $\mu'$  and  $\Sigma'$  represent those of the generated motion distribution.

Diversity calculates the variance of randomly sampled pairs from the generated motions, with larger values indicating greater diversity. Diversity is expressed as

$$Diversity = \frac{1}{S} \sum_{i=1}^S \|v_i - v'_i\|_2 \quad (4)$$

where  $S$  is the number of sampled pairs, and  $v_i$  and  $v'_i$  are pairs of motion vectors.

For FID and diversity evaluation, feature vectors extracted from current evaluation models are typically used. However, there is currently no suitable feature extractor for the 2D Human Motion-Language dataset. Therefore, we used the motion tokenizer of

MotionGPT and 2CM-GPT as feature extractors, enabling fair comparison of the inherent performance of each method. This enables accurate evaluation of the reconstruction and generation accuracy of 2CM-GPT, clearly demonstrating performance differences with other methods.

### 4.4 Motion Reconstruction

We quantitatively and qualitatively evaluated the 2D-motion-reconstruction accuracy of 2CM-GPT’s motion tokenizer.

**Quantitative Evaluation.** The accuracy of motion reconstruction using the motion tokenizer is shown in Table 1, calculated using MPJPE, FID, and diversity. The table also shows that 2CM-GPT improved reconstruction accuracy compared with MotionGPT. Specifically, MPJPE improved by 0.018 points, indicating improved joint-position accuracy, enabling closer reconstruction to the ground truth. To complement the quantitative evaluation, we also show MPJPE results at individual joints in table 2. The table shows that 2CM-GPT consistently outperforms MotionGPT at all joints. Notably, joints such as the hips, shoulder, and neck exhibit the lowest MPJPE values. These joints typically involve less complex movement patterns, making them more predictable for the model. In contrast, joints like the wrists and elbows show higher MPJPE values, suggesting that these are more challenging to reconstruct. This is likely due to their greater range of motion and higher degrees of freedom, as well as the fact that they are involved in more intricate and dynamic movements. FID improved by over 7.894 points, indicating that the reconstructed motion distribution is similar to that of real motions. Diversity improved by over 0.859 points, indicating enhanced diversity in the generated motions, enabling 2CM-GPT to reconstruct a broader range of motion patterns. These results indicate that using 2D convolution for processing 2D motion data is more effective than using 1D convolution for motion reconstruction.

**Qualitative Evaluation.** The reconstructed motion using the motion tokenizer is shown in Figure 3. The figure also shows that 2CM-GPT reconstructed motion is closer to the input motion compared with MotionGPT. Focusing on the movements of each joint, the generated results from MotionGPT indicate noticeable deviations in joint positions and orientations. For example, even when the input motion faces left, the reconstructed motion from MotionGPT

Table 1: Comparison of accuracy of motion reconstruction by motion tokenizer between MotionGPT and 2CM-GPT. The method names in the columns for FID and diversity indicate the motion tokenizer used as the feature extractor. “Real” represents the evaluation results of ground-truth 2D motions.

Method	MPJPE ↓	FID ↓		Diversity ↑	
		MotionGPT	2CM-GPT	MotionGPT	2CM-GPT
Real	0.000	0.000	0.000	16.981	16.954
MotionGPT	0.179	33.753	34.545	11.385	10.896
2CM-GPT	<b>0.161</b>	<b>25.859</b>	<b>25.150</b>	<b>12.244</b>	<b>12.129</b>

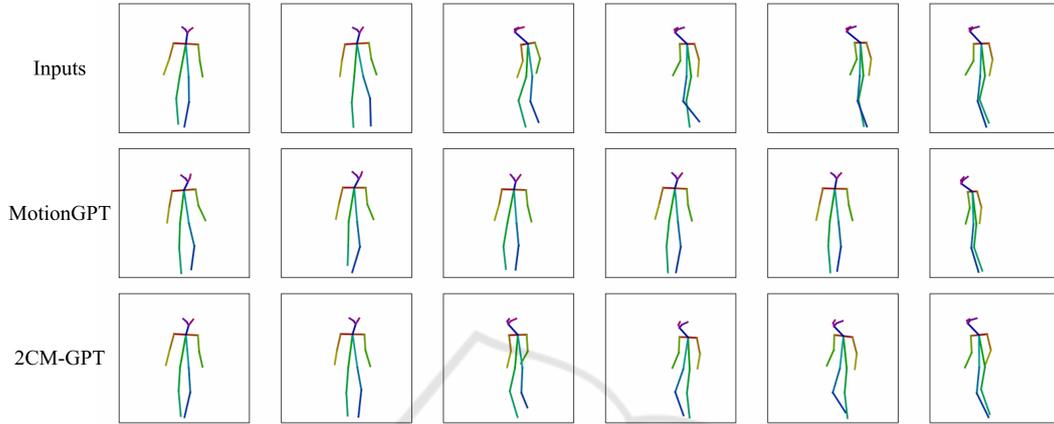


Figure 3: Comparison of reconstructed motions by motion tokenizer between MotionGPT and 2CM-GPT. The left image shows the initial frame, and the right image shows the final frame.

Table 2: Comparison of MPJPE for each joint of motion reconstruction by motion tokenizer between MotionGPT and 2CM-GPT.

Joint	Method	
	MotionGPT	2CM-GPT
Nose	0.180	<b>0.162</b>
Neck	0.155	<b>0.139</b>
Shoulder_Right	0.168	<b>0.149</b>
Elbow_Right	0.196	<b>0.171</b>
Wrist_Right	0.247	<b>0.223</b>
Shoulder_Left	0.168	<b>0.155</b>
Elbow_Left	0.195	<b>0.180</b>
Wrist_Left	0.245	<b>0.234</b>
Hip_Right	0.151	<b>0.133</b>
Knee_Right	0.156	<b>0.137</b>
Ankle_Right	0.173	<b>0.151</b>
Hip_Left	0.151	<b>0.135</b>
Knee_Left	0.156	<b>0.141</b>
Ankle_Left	0.172	<b>0.155</b>
Eye_Right	0.177	<b>0.159</b>
Eye_Left	0.177	<b>0.161</b>
Ear_Right	0.170	<b>0.154</b>
Ear_Left	0.171	<b>0.156</b>

might face forward, resulting in incorrect motion reconstruction. In contrast, 2CM-GPT maintains joint positions and orientations closer to the input motion, accurately reproducing the overall motion. This suggests that using 2D convolution in 2CM-GPT enables

better modeling of spatial relationships between joints compared with using 1D convolution in MotionGPT.

#### 4.5 Text-Guided Motion Generation

We qualitatively evaluated the accuracy of text-guided 2D motion generation using 2CM-GPT, which uses a motion tokenizer with improved motion-reconstruction accuracy. The 2D motions generated by inputting various texts is shown in Figure 4. The results in Figure 4 indicate that the generated 2D motions align with the input texts. Notably, the start and end points of actions, such as leg kicks and hand claps, are accurately represented, and the consistency of movements is maintained. 2CM-GPT effectively interprets the semantic information in the text and translates it into 2D motion, demonstrating high performance in both accuracy and realism.

#### 4.6 Application of Text-Guided 3D Motion Generation

We applied the 2D motion generated with 2CM-GPT from text to pose-guided human video generation, demonstrating the effectiveness of 2CM-GPT through qualitative evaluation. For pose-guided human video generation, we used DisCo (Wang et al., 2024), a method characterized by faithfulness, generalizabil-

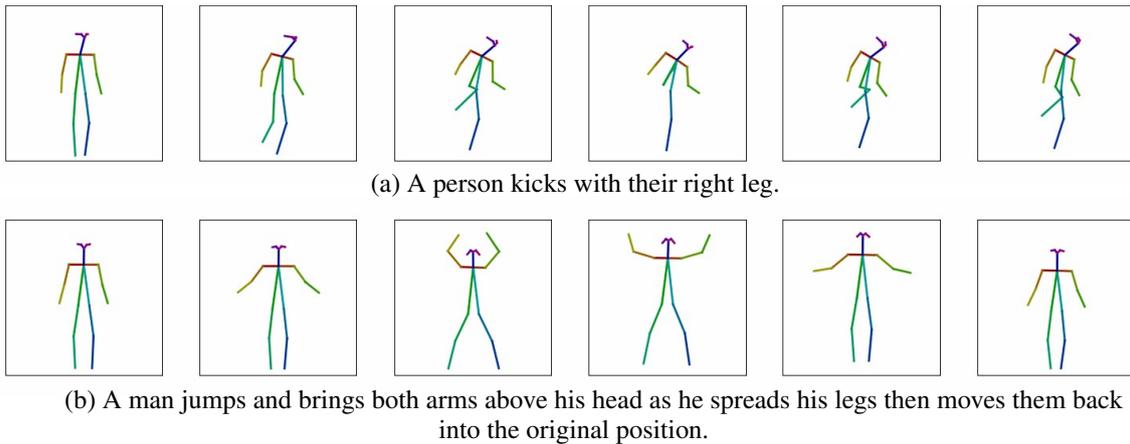


Figure 4: Text-guided motion generation results from 2CM-GPT. The corresponding input texts are shown in (a) and (b).

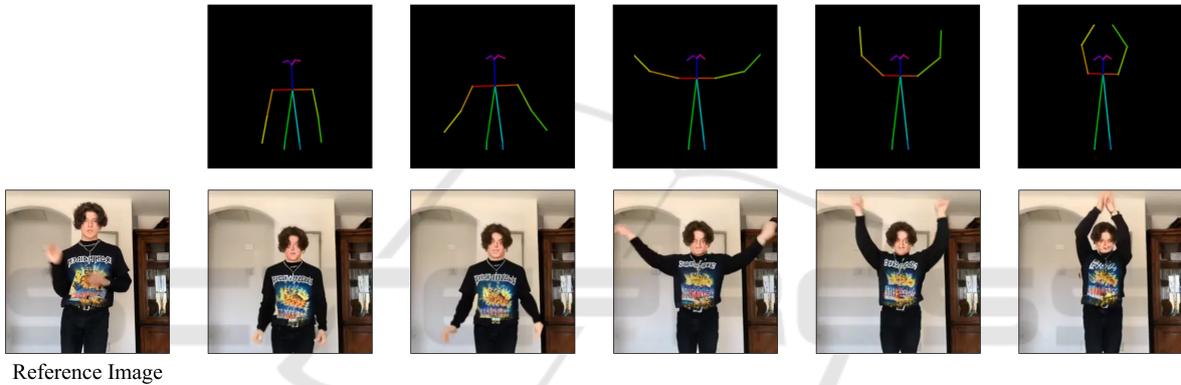


Figure 5: Human video generation results using 2D motion generated with 2CM-GPT. The input text used for generating the 2D motion is “A person claps their hands together well above their head.”

ity, and compositionality . The result of inputting the 2D motion generated with 2CM-GPT into DisCo is shown in Figure 5. The results in this figure indicate that the input 2D motion results in natural actions in the generated human video. The 2D motion generated with 2CM-GPT was seamlessly integrated into the output of DisCo, confirming the effectiveness of 2CM-GPT.

### 4.7 Ablation Studies

We evaluated the appropriate size  $K$  of codebook using motion tokenizer with different codebook sizes. The accuracy of motion reconstruction using the motion tokenizer is shown in Table 3, calculated using MPJPE, FID, and diversity. The table shows that  $K = 32$  is appropriate for motion tokenizer. This suggests that a too large size  $K$  of codebook cannot represent 2D motion well, which has few features. However, if the codebook size  $K$  is too small, the motions generated by the language model become monotonous and less diverse. It is suggested that the language model

cannot learn the relationship between text and motion well because there are not enough tokens to represent motion.

Table 3: Comparison of accuracy of motion reconstruction by motion tokenizer in 2CM-GPT with different codebook sizes. 2CM-GPT used as the feature extractor.

2CM-GPT	MPJPE ↓	FID ↓	Diversity ↑
K=16	0.122	<b>7.816</b>	<b>14.251</b>
K=32	<b>0.114</b>	8.313	14.049
K=64	0.134	15.933	13.104
K=128	0.161	25.150	12.129
K=256	0.176	34.037	11.054
K=512	0.175	33.580	11.213
K=1024	0.185	40.502	10.952

## 5 CONCLUSIONS

We proposed 2CM-GPT for generating 2D motion from text. By adopting 2D convolution for the motion tokenizer, 2CM-GPT is able to more accurately

model spatial relationships between joints compared with methods using 1D convolution. Evaluation experiments demonstrated that 2CM-GPT achieved superior accuracy in motion-reconstruction tasks both quantitatively and qualitatively and showed high performance in text-guided 2D motion generation. Applying the motions generated with 2CM-GPT to pose-guided human video generation confirmed that the resulting videos exhibited natural movements. These results indicate the practicality of 2CM-GPT in motion-generation tasks. Future work will include training with more diverse motion datasets, introducing advanced architectures to further strengthen the relationship between text and motion, and exploring other possible applications for 2D motion generation.

## REFERENCES

- Ahn, H., Ha, T., Choi, Y., Yoo, H., and Oh, S. (2018). Text2Action: Generative Adversarial Synthesis from Language to Action. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5915–5920. IEEE.
- Ahuja, C. and Morency, L.-P. (2019). Language2Pose: Natural Language Grounded Pose Forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE.
- Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., and Yu, G. (2023). Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010.
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. ACL.
- Graves, A. and Graves, A. (2012). Long Short-Term Memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.
- Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., and Cheng, L. (2022). Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161.
- Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., and Cheng, L. (2020). Action2Motion: Conditioned Generation of 3D Human Motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in neural information processing systems*, 30.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.
- Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., and Chen, T. (2024). MotionGPT: Human Motion as a Foreign Language. *Advances in Neural Information Processing Systems*, 36.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16.
- Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). AMASS: Archive of Motion Capture as Surface Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451.
- Petrovich, M., Black, M. J., and Varol, G. (2022). TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pages 480–497. Springer.
- Plappert, M., Mandery, C., and Asfour, T. (2016). The KIT Motion-Language Dataset. *Big Data*, 4(4):236–252.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research*, 21(140):1–67.
- Razavi, A., Van den Oord, A., and Vinyals, O. (2019). Generating Diverse High-Fidelity Images with VQ-VAE-2. *Advances in neural information processing systems*, 32.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3104–3112.
- Tevet, G., Gordon, B., Hertz, A., Bermano, A. H., and Cohen-Or, D. (2022). MotionCLIP: Exposing Human Motion Generation to CLIP Space. In *European Conference on Computer Vision*, pages 358–374. Springer.
- Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., and Bermano, A. H. (2023). Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*.
- Van Den Oord, A., Vinyals, O., et al. (2017). Neural Discrete Representation Learning. *Advances in neural information processing systems*, 30.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.

- (2017). Attention Is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 6000–6010. Curran Associates Inc.
- Wang, T., Li, L., Lin, K., Zhai, Y., Lin, C.-C., Yang, Z., Zhang, H., Liu, Z., and Wang, L. (2024). DisCo: Disentangled Control for Realistic Human Dance Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9326–9336.
- Zhang, J., Zhang, Y., Cun, X., Huang, S., Zhang, Y., Zhao, H., Lu, H., and Shen, X. (2023). T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., and Liu, Z. (2024). MotionDiffuse: Text-Driven Human Motion Generation with Diffusion Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

