# Investigating Dynamical Representations for Human Motion Generation

Wondmgezahu Teshome Mustafa Bozdag Elaheh Motamedi
Northeastern University
teshome.w@northeastern.edu bozdag.m@northeastern.edu motamedi.e@northeastern.edu

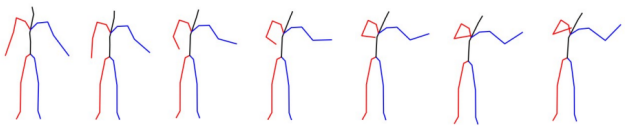
## Abstract

*The complexity of state-of-the-art models in the field of computer vision is escalating at a rapid pace. Among these, diffusion models are emerging as a focus of significant interest. These powerful models, however, often lead to computationally demanding processes, particularly when handling video data that necessitates the processing of temporal information. Moreover, processing video adds more variables to the problem to be considered such as the system dynamics. This project proposes to mitigate these challenges for diffusion-based human motion generation models by performing diffusion in a lower-dimensional dynamical space using Discrete Cosine Transform (DCT). For this purpose, we propose a variant of MotionDiffuse, a diffusion-based human motion generation framework. We utilize DCT in the forward and backward diffusion denoising processes of the original MotionDiffuse model. This approach allows us to leverage the efficiency of classical dynamical representations, thereby reducing the computational burden associated with video processing applications. Our goal is to enhance diffusion-based human motion generation models by integrating DCT within the diffusion section of the framework. We aim to give importance to more efficient and effective video processing techniques, making these tools more accessible for various applications.*

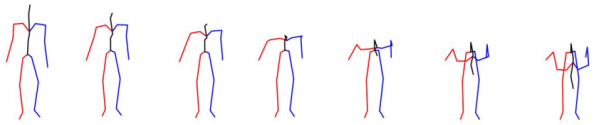
## 1. Introduction

Recent advancements in data collection and computational processing capabilities have caused exponential progress in computer vision. The availability of data and the corresponding processing capacity have allowed learning-based methods to become more than theoretical concepts and see real-world implementations with empirical validation. This growth has given rise to diverse techniques addressing fundamental challenges in the field. The initial phase of deep learning for computer vision focused on mimicking human neurons to recognize basic visual patterns, such as digits [14]. After the success of AlexNet [12], convolutional neural networks (CNNs) [13] emerged as the dominant paradigm in deep learning approaches for computer vision. Lately, the Transformer architecture with an attention mechanism, initially designed for translation tasks in natural language processing (NLP) [25], was swiftly integrated into computer vision through the Vision Transformer (ViT) [6]. This architecture was also adapted for video processing applications in the form of a Video Vision Transformer (ViViT) [3]. In tandem with techniques primarily focused on detecting features and extracting meaning from data, there have been notable achievements in generative models, exemplified by the prominence of generative adversarial networks (GANs) [7] and variational autoencoders (VAEs). More recently, diffusion models, inspired by principles of non-equilibrium thermodynamics [22], have attained state-of-the-art status in generative tasks. Numerous implementations of these models have surfaced for public use [20, 21].

In computer vision, tasks that concern human motion stand out as particularly sought after and challenging. With the success of diffusion-based models in image generation according to both qualitative and quantitative measures [20, 21, 26], the advancements were quickly followed by similar implementations in human motion generation



(a) *"A person is playing violin."*



(b) *"A person is bowing."*

Figure 1. Motions generated by the MotionDiffuse-DCT method on the HumanML3D dataset, based on their respective text descriptions.

[5, 27]. The objective of human motion generation is to create natural, realistic, and varied human movements suitable for various applications such as film production, video games, and the development of digital human characters [29]. Meeting the demand for this task often requires integrating context as a conditional signal, like text descriptions or background audio. The generated motion must not only be internally plausible but also should be synchronized with the given conditional signal. Furthermore, as a vital nonverbal communication medium, human motion reflects diverse factors, such as goals, personal styles, social norms, and cultural expressions [24]. Ideally, motion generation models should excel in capturing subtle variations while establishing a meaningful semantic connection with the provided conditional signals [29]. Computational complexity, a challenge that always persists in deep learning, remains an issue for the state-of-the-art models as they do not only use highly complex frameworks, but they utilize the combination of multiple computationally demanding techniques in a single framework [4, 5, 27]. This challenge becomes even more rigorous for video as the addition of a temporal dimension brings more than just an increase in parameter count or model size. For the temporal information to be handled properly, the connection between visual features in a series of frames needs to be detected and processed in the context of each frame and also the entirety of the sequence of frames. Many approaches were proposed to overcome the addition of a temporal dimension including utilizing classical dynamical representations [15, 17], and using the same techniques for image feature detection in various ways that would capture the dynamical connection between each frame [2, 3, 5].

In this paper, we investigate the effect of utilizing classical dynamical representations in human motion generation. We propose to achieve this by incorporating DCT into the diffusion process in the MotionDiffuse model. Our approach aims to reduce the computational complexity of a diffusion-based human motion generation model, inspired by approaches that are based on similar methods in the literature [2, 5, 15, 17]. The proposed framework encapsulates previous approaches while also extending control over the generated motion.

## 2. Related Work

### 2.1. Diffusion Models

Diffusion models are generative models inspired by non-equilibrium statistical physics, where the structure in a data distribution is destroyed iteratively, and a process that would reconstruct this distribution is learned afterward [22]. The most common framework used in computer vision is the denoising diffusion probabilistic models (DDPM) characterized by forward and reverse diffusion processes [11]. The forward process destroys the data from the true distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ using a Markov chain to add Gaussian noise at each step:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where $\beta_t$ is a small positive constant representing the noise level, $t$ is the diffusion step, and $\mathbf{I}$ is the identity matrix. Since the noise used at each step is Gaussian, $q(\mathbf{x}_t|\mathbf{x}_0)$ can be obtained in closed-form:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I}) \tag{2}$$

where $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$. The reverse process denoises $\mathbf{x}_t$ to recover $\mathbf{x}_0$, and is defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t)\mathbf{I}) \tag{3}$$

where $\mu_\theta$ and $\sigma_\theta$ are approximated by a neural network. It has been shown that this reverse process can be trained by solving the optimization problem:

$$\min_\theta \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2 \tag{4}$$

where $\epsilon_\theta$ is a trainable denoising function, and estimates the noise vector $\epsilon$ that was added to its noisy input $\mathbf{x}_t$ [11].

Diffusion models are in principle capable of modeling conditional distributions of the form $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{y}$ is a conditional input. In the context of **conditional diffusion models**, the model learns to predict the noise added to the noisy input given a set of conditions, including the time step $t$ and the conditional inputs $\mathbf{y}$. The reverse process in this case is defined as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t|\mathbf{y}), \sigma_\theta(\mathbf{x}_t, t|\mathbf{y})\mathbf{I}) \tag{5}$$

The conditional diffusion model learns a network $\epsilon_\theta$ to predict the noise added to the noisy input $\mathbf{x}_t$ with:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0),\epsilon\sim\mathcal{N}(0,I),t,\mathbf{y}}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})\|_2^2\right] \tag{6}$$

where $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + (1-\alpha_t)\epsilon$, and $\mathcal{L}$ is the overall learning objective of the diffusion model.

### 2.2. Transformers

The Transformer architecture offers a generalized approach to understanding context without predefined constraints. It was originally proposed for natural-language translation tasks [25], but with the success it had in NLP applications, it was quickly adapted to the tasks of computer vision [3, 6]. In the context of Transformers, each element in a sequence is embedded into a vector, named a token. The self-attention module is designed to capture long-range interactions among three types of inputs: queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$, where the values are linearly combined

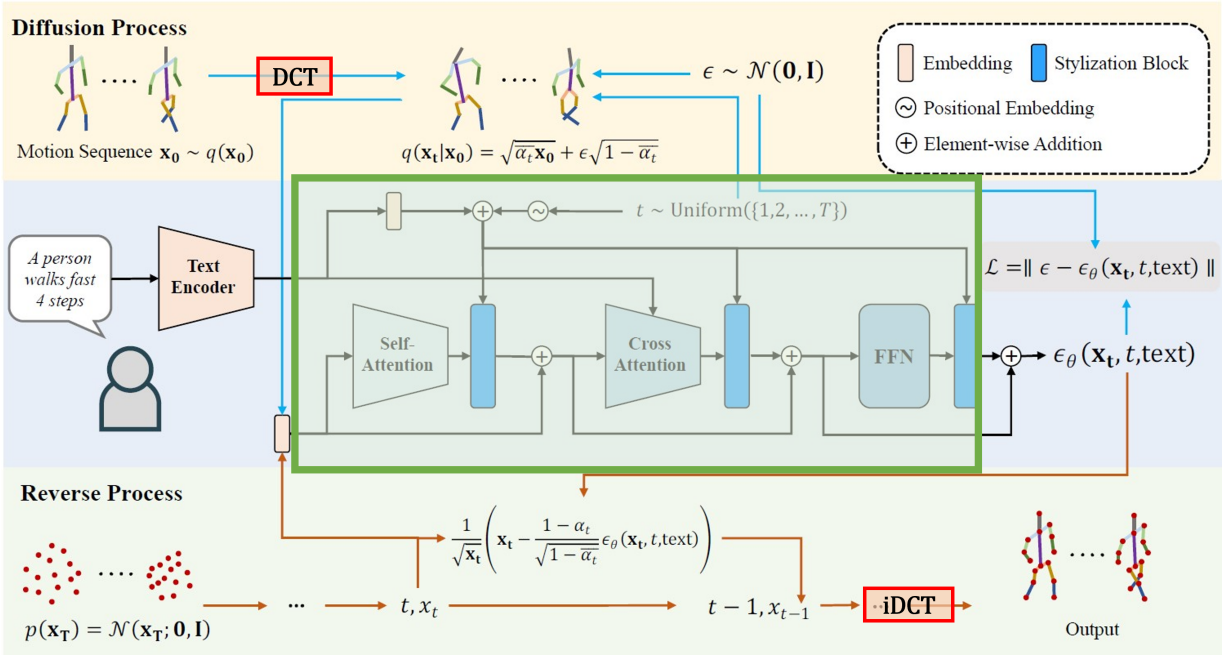



Figure 2. MotionDiffuse pipeline. Arrow colors indicate different stages: blue for training, red for inference, and black for both [27]. The approach in this work is performing the diffusion in the DCT domain and converting the final representation back with iDCT (red boxes). The motion decoder section that consists of the CMLT is enclosed with a green box.

according to the importance of each key representing each of the queries. The original mechanism proposed is a scaled dot-product attention in the form:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = SoftMax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (7)$$

where $d_k$ is the dimension of the keys $\mathbf{K}$, used for scaling the dot-product of key-query pairs $\mathbf{Q}\mathbf{K}^T$. In self-attention, the key and queries are derived from the same input sequence with a linear transformation that is learned by the network in the form of weights $W_\mathbf{Q}$ and $W_\mathbf{K}$ [25]. This mechanism is used to learn the dependencies between the elements of a given sequence. Another commonly used version of this mechanism is the **cross-attention**, where the key and queries are derived from two different input sequences to learn the relationship between them. The most prominent use cases of cross-attention are text-based applications that combine the information from a text sequence with the information extracted from an image [19]. The main benefit of this approach is the ability to control a process using the main medium of interaction for humans, natural language, and understanding the effects more clearly.

### 2.3. Dynamical Representations

The extraction of system dynamics is a multifaceted topic spanning disciplines such as dynamical systems, control theory, and computer vision [2, 5, 15]. Various implementations have endeavored to leverage classical approaches for extracting temporal information from image and video data. Notably, the dynamical atoms-based network (DYAN) employs dynamics-based affine invariants within a shallow neural network structure to capture both long and short-term temporal information from videos [15]. This is achieved by encoding video frames into sparse embeddings constructed as the poles of a discrete linear time-invariant (LTI) system. Such an architecture allows the use of fewer parameters for computationally demanding tasks, such as action recognition, where prevailing methods often rely on resource-intensive models [15]. In addition to DYAN, various approaches in the literature leverage the discrete cosine transform (DCT) [1], a transformation technique widely employed in signal processing and data compression. The DCT represents a finite sequence of data points as a summation of cosine functions oscillating at different frequencies. Distinct from the Discrete Fourier Transform (DFT), the DCT employs only real numbers [1]. The DCT operation extracts both current and periodic temporal properties from motion sequences, proving beneficial for capturing continuous motions. Recent works utilize DCT for both human motion detection and generation tasks [2, 5, 17]. For instance, in [5], the authors train a diffusion model with DCT in a masked completion task. Each coefficient produced in this manner encodes information from the entire sequence at a specific temporal frequency. In another

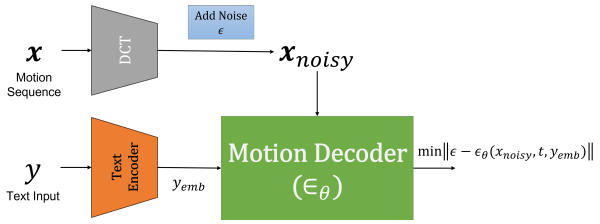



Figure 3. Applying DCT on the MotionDiffuse Pipeline.

study [10], the authors combine DCT with standard practices, such as predicting residual displacement of joints and optimizing velocity as an auxiliary loss. This integrated approach showcases the versatility of DCT in enhancing various aspects of human motion representation and generation within deep learning models.

## 3. Methodology

In this work, we focus on using DCT with the MotionDiffuse [27] model to analyze the effects of using a dynamical representation with a resource-demanding human motion generation framework. This decision was driven by the availability of the model implementation, featuring a highly complex architecture that incorporates multiple techniques, thereby increasing computational complexity.

### 3.1. MotionDiffuse

The MotionDiffuse model [27], a diffusion-based approach, employs an 8-layer Cross-Modality Linear Transformer (CMLT) network to integrate text inputs for controllable human motion generation. The CMLT architecture comprises self-attention, cross-attention, and feed-forward network (FFN) blocks. The self-attention block takes as input the noisy motion sequence:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \sqrt{\overline{\alpha_t}}\mathbf{x}_0 + \epsilon\sqrt{1-\overline{\alpha_t}} \tag{8}$$

where $\mathbf{x}_t$ is the ground truth distribution, $\alpha_t = 1 - \beta_t$, $\overline{\alpha_t} = \prod_{s=0}^{t} a_s$, and $q(\mathbf{x}_t|\mathbf{x}_0)$ is the sample from the noisy distribution that comes from the forward diffusion process, similar to the original formulation in 2. Here, $\mathbf{x}$ refers to joint data containing information such as joint position, linear velocity, and height. The output of the attention block is then combined with positional embeddings for each sample vector. This operation is executed using a Stylization block, where the original output of any previous block $\mathbf{Y}$ undergoes the following process:

$$\mathbf{B} = \psi_b(\phi(e)),\ \mathbf{W} = \psi_w(\phi(e)),\ \mathbf{Y}' = \mathbf{Y}\cdot\mathbf{W} + \mathbf{B} \tag{9}$$

where $(\cdot)$ denotes Hadamard product, $e$ represents the positional embeddings, and $\psi_b, \psi_w$ and $\phi$ are different linear projections. This stylization process is applied after each block in the CMLT, and it incorporates the positional embeddings $e$. Following the conventional Transformer architecture, a residual connection links the input of each block to its output. Subsequently, the output of the self-attention block is directed to a cross-attention block along with text embeddings. The text embeddings are created using a text encoder with the original Transformer architecture in [25], where the first several layer weights are initialized with pre-trained CLIP weights [19] to enhance the generalization ability of the model. Finally, the output of the cross-attention block feeds into an FFN layer, with the resulting output of the FFN block providing the denoising function $\epsilon_0$. The entire model is trained on a single loss function:

$$\mathcal{L} = \mathcal{E}_{t\in[1,T],\mathbf{x}_0\sim q_0(\mathbf{x}_0),\epsilon\sim\mathcal{N}(0,\mathbf{I})}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, text)\|_2^2\right] \tag{10}$$

which essentially is the same as the loss function defined for conditional diffusion models given in 6, with the condition vector $\mathbf{y}$ replaced with $text$. The overall architecture can be seen in Figure 2.

### 3.2. DCT Implementation

Implementing DCT to a computer vision framework to improve the understanding of dynamical information from the data has been done in recent works in the literature [5, 17]. However, the main focus is generally on the aforementioned dynamical aspects of DCT, and the computational benefits of using a lower-dimensional space that could potentially encapsulate all the information related to the dynamics of the system is often overlooked. In this work, we are trying to focus on this perspective of DCT by applying it to the MotionDiffusion framework and comparing it with the original model in terms of different performance benchmarks and computation time. Given a motion sequence $\mathbf{x}$, the DCT operation simply projects the motion sequence into a frequency space, commonly referred as the DCT space as the following:

$$\mathbf{X}_k = \sum_{n=0}^{N-1} \mathbf{x}_n cos\left[\frac{k\pi}{N}\left(n + \frac{1}{2}\right)\right] \tag{11}$$

This formulation is known as the DCT-II and it is simply referred to as "the DCT" as it is the most commonly used formulation [1]. To implement DCT with the MotionDiffusion framework, we take the initial motion sequence $\mathbf{x}$ or $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ in Figure 2, and apply the DCT operation before starting the forward-diffusion process. Thus, the entire motion-decoder block that learns the denoising function, enclosed with a green rectangular frame in Figure 2, performs the diffusion denoising process in the DCT space. The main benefit of using DCT is the ability to disregard high-frequency components in a dynamical sequence that does not contribute much to the information we get from the data. Thus, we can define the dimensions of the DCT

space as we like, within the limitations of the algorithm. It also allows the user to understand which components contribute more to the overall motion generation process, and keep them while disregarding the rest without a critical information loss. Furthermore, since the DCT operation is an orthogonal transform, the motion sequence generated by the diffusion process can be recovered from the DCT space by applying the inverse DCT (iDCT) operation to the output of the motion decoder. Leveraging these benefits, we apply the DCT before the motion decoder as shown in 3 and 2 and recover the output of the decoder using the iDCT operation seen in the last step of the overall pipeline in Figure 2.

## 4. Experiments

In this section, we present a series of experiments designed to evaluate the performance and effectiveness of our proposed method. Our primary goal is to provide a comprehensive understanding of the strengths and potential limitations of our approach. To this end we have conducted a series of rigorous tests and evaluations. These include both quantitative and qualitative assessments, as well as comparisons with the original motionDiffuse [27] methods.

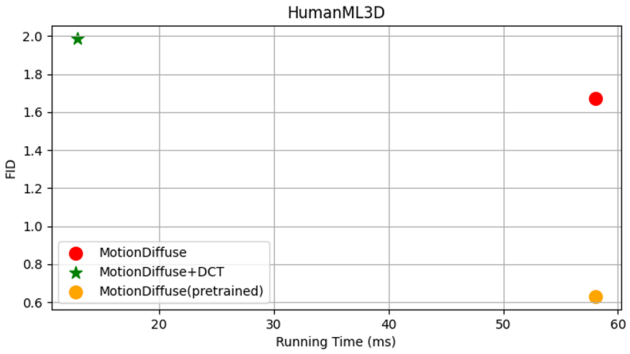



Figure 4. Comparison of FID versus running time per frame for all baseline models on the HumanML3D dataset.

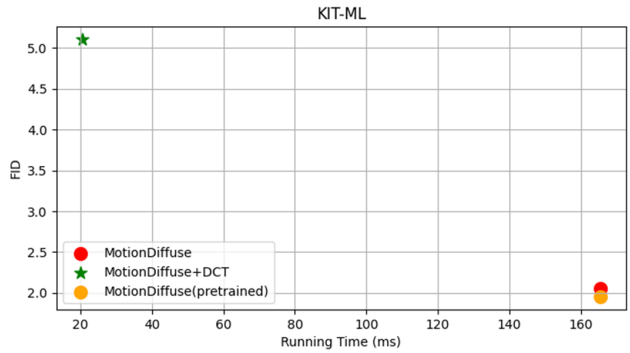



Figure 5. Comparison of FID versus running time per frame for all baseline models on the KIT-ML dataset.

The experiments are structured as follows: First, we describe the datasets and evaluation metrics used in our experiments. Next, we detail the experimental setup, including the implementation details and the parameter settings. We then present the results of our method and compare them with the original model and also with other approaches for performance. Finally, we provide an in-depth analysis and discussion of the results, highlighting the key findings and insights.

**Datasets used:** In this study, two main datasets were used: the KIT dataset [18] and the HumanML3D dataset [8]. The KIT dataset is a comprehensive collection of 3911 motion sequences encompassing various actions such as walking, running, and jumping. It also includes 6353 sequence-level descriptions that provide detailed and diverse annotations for these motion sequences.

On the other hand, the HumanML3D dataset is a combination of the HumanAct12 [9] and AMASS [16] datasets. It comprises 14,616 motions and 44,970 descriptions composed of 5,371 distinct words. This dataset covers a broad range of human actions, including daily activities (e.g., 'walking', 'jumping'), sports (e.g., 'swimming', 'playing golf'), acrobatics (e.g., 'cartwheel'), and artistry (e.g., 'dancing'). The preparation of the HumanML3D dataset followed the procedures outlined in a GitHub repository: [https://github.com/EricGuo5513/HumanML3D].

**Metrics used:** In the evaluation of our work, we employed quantitative metrics to assess the computational complexity and evaluate the generated results. For measuring model complexity, we utilized FLOPs (Floating Point Operations), MACs (Multiply-Accumulate Operations), and inference time. FLOPs and MACs quantify the number of arithmetic operations required by the model, while inference time measures the time taken to generate results. To evaluate the generated motions, we employed the following metrics:

**FID (Fréchet Inception Distance)**: Measures the similarity between real and generated data based on feature representations extracted from a pre-trained Inception model.

**Diversity**: This metric assesses the variety and distinctiveness of the generated motions.

**MultiModality**: Quantifies the presence of multiple modes in the generated motion distribution.

**Multimodal Distance (MM Dist)**: This metric is used in the context of human motion generation to calculate the distance between motions and texts.

**R precision**: Evaluates the precision of the top-ranked samples compared to the real data. These metrics provide a comprehensive assessment of the quality and diversity of the generated motions, aiding in the thorough evaluation of the proposed approach.

**Inference Time Performance:** This metric measures the efficiency of various baseline models by quantifying their

| Methods | FID↓ | Diversity→ | MultiModality↑ | MM Dist↓ | R precision↑ | | |
|---|---|---|---|---|---|---|---|
| | | | | | Top1 | Top2 | Top3 |
| Real Motions | 0.002 | 9.503 | - | 2.974 | 0.511 | 0.703 | 0.797 |
| MotionDiffuse (pre-trained) | 0.630 | 9.410 | 1.553 | 3.113 | 0.491 | 0.681 | 0.782 |
| MotionDiffuse | 1.6727 | 8.8121 | 2.2302 | 3.4470 | 0.4244 | 0.6106 | 0.7159 |
| MotionDiffuse+DCT | 1.9873 | 8.2921 | 2.2814 | 3.7238 | 0.3981 | 0.5858 | 0.6940 |

Table 1. Text-to-motion task performance on HumanML3D test split. The right arrow → indicates higher similarity to real motion is better, and MM Dist stands for MultiModal Distance.

| Methods | FID↓ | Diversity→ | MultiModality↑ | MM Dist↓ | R precision↑ | | |
|---|---|---|---|---|---|---|---|
| | | | | | Top1 | Top2 | Top3 |
| Real Motions | 0.031 | 11.08 | - | 2.788 | 0.424 | 0.649 | 0.779 |
| MotionDiffuse (pre-trained) | 1.954 | 11.10 | 0.730 | 2.958 | 0.417 | 0.621 | 0.739 |
| MotionDiffuse | 2.0525 | 10.5476 | 2.0987 | 3.5326 | 0.2315 | 0.3991 | 0.5241 |
| MotionDiffuse+DCT | 5.1040 | 9.4313 | 1.5356 | 4.2916 | 0.3509 | 0.5156 | 0.6250 |

Table 2. Text-to-motion task performance on KIT-ML test split. The right arrow → indicates higher similarity to real motion is better, and MM Dist stands for MultiModal Distance.

inference time per frame, expressed in milliseconds. It provides an assessment of the computational efficiency of the models, with lower inference times indicating faster, more efficient models.

**Implementation Details:**
For both of the datasets (HumanML3D and KIT-ML), we follow the implementation of [27]. We used an 8-layer Transformer as the motion decode, where as for the text encoder, CLIP ViTB/32 [19] along with four more Transformer encoder layers were used. The detailed hyperparameter configuration is mentioned below in Tables 7 and 8. Regarding the GPUs utilized for training, both models - MotionDiffuse and MotionDiffusion+DCT - were trained on the KIT-ML dataset using a single NVIDIA GeForce RTX 2080 Ti GPU for an approximate duration of 2 days. For the HumanML3D dataset, we employed a single NVIDIA TITAN Xp and the training lasted for about 4.5 days. For the pretrained MotionDiffuse model, which was used as a baseline in this study, they used 8 Tesla V100 GPUs during training. Each GPU handled 128 samples, resulting in a total batch size of 1024. The training involved 40K iterations for the KIT-ML dataset and 100K iterations for the HumanML3D dataset. We trained For comparing the different baselines, we train the original MotionDiffuse [27], with the same training setting as ours to have a fair comparison. We also report the values of the pre-trained MotionDiffuse model. For the computation of FLOPs and MACs, we employed a batch size of 1. The maximum sequence length was set to 20 for the MotionDiffuse+DCT model on the KIT-ML dataset, 40 for the HumanML3D dataset, and 196 for the MotionDiffuse model.

In the series of experiments conducted by [8], the pose states primarily consist of seven distinct components: $(r_{va}, r_{vx}, r_{vz}, r_h, j_p, j_v, j_r)$. The root joint's angular velocity along the $Y$-axis, linear velocity along the $X$-axis, and the $Z$-axis are represented by $(r_{va}, r_{vx}, r_{vz}) \in R$, respectively. The height of the root joint is denoted by $r_h \in R$. The position and linear velocity of each joint are represented by $(j_p, j_v) \in R^{Jx3}$, where $J$ is the number of joints. The 6D rotation of each joint is denoted by $j_r \in R^{Jx6}$ [28]. In the HumanML3D dataset, $J$ is 22, and in the KIT-ML dataset, $J$ is 21.

### 4.1. Comparisons on Text-to-motion

Our model is evaluated against several baselines including Real Motions (ground truth), MotionDiffuse, and MotionDiffuse with their pre-trained models for the text-to-motion task. Tables 1 and 2 present a quantitative comparison of our model with these baselines on the HumanML3D and KIT-ML datasets, respectively. For the HumanML3D dataset, our model (MotionDiffuse+DCT) outperforms MotionDiffuse (trained under the same settings as ours) in terms of MultiModality, while achieving comparable results on other metrics. On the KIT-ML dataset, our model demonstrates superior precision compared to MotionDiffuse, and exhibits comparable performance on other metrics. Furthermore, we see that the scores of the MotionDiffuse+DCT model trained on HumanML3D are closer to the original model in all metrics compared to the version trained on the KIT-ML dataset. This might be caused due to the extent of the content of these datasets, and training times. However, we need to point out a significant difference in hypermarameter configurations between the two models that can be seen in Tables 7 and 8. The model was initially trained for the KIT-ML dataset with the first 20 DCT coefficients. Later on, we expanded our usage to the first 40

| Datasets | Methods | Params(M) | ITPF(ms) | FLOPs(G) | | MACs(G) | |
|---|---|---|---|---|---|---|---|
| | | | | fwd | fwd+bwd | fwd | fwd+bwd |
| KIT-ML | MotionDiffuse | 87.14 | *165.4065* | 33 | 99 | 16.3 | 48.91 |
| | MotionDiffuse+DCT | 87.14 | 20.5837 | 16.27 | 48.82 | 8.09 | 24.28 |
| HumanML3D | MotionDiffuse | 87.15 | 58.03 | 33.01 | 99.03 | 16.31 | 48.93 |
| | MotionDiffuse+DCT | 87.15 | 12.8796 | 18.17 | 54.52 | 9.03 | 27.08 |

Table 3. Comparative analysis of model complexities across different models and datasets. The DCT model utilizes the first 20 rows of DCT coefficients for the KIT-ML dataset and the first 40 rows for the HumanML3D dataset. Metrics such as Inference Time Per Frame (ITPF), number of parameters (Params), with 'M' standing for Million, 'G' for Giga, 'ms' for milliseconds, 'fwd' for forward pass, and 'bwd' for backward pass are included.

| Methods | FID↓ | Diversity→ | MultiModality↑ | MM Dist↓ | R precision↑ | | |
|---|---|---|---|---|---|---|---|
| | | | | | Top1 | Top2 | Top3 |
| Real Motions | 0.002 | 9.503 | - | 2.974 | 0.511 | 0.703 | 0.797 |
| MotionDiffuse+DCT (40) | 1.9873 | 8.2921 | 2.2814 | 3.7238 | 0.3981 | 0.5858 | 0.6940 |
| MotionDiffuse+DCT (20) | 9.7189 | 7.3259 | 3.0548 | 5.6472 | 0.2498 | 0.3744 | 0.4662 |

Table 4. Ablation for sampling on HumanML3D dataset using the first 20 rows of DCT coefficients, which was trained on the first 40 rows.

coefficients while training the model on the HumanML3D dataset. This increase was motivated by the idea of using more coefficients to increase the performance while remaining significantly efficient compared to the original model. Due to time constraints, the effect of the number of DCT coefficients could not be analyzed deeply. This difference in the number of coefficients might be causing a drop in performance in the KIT-ML dataset, which remains to be investigated thoroughly in future work. However, as pointed out before, the model still outperforms the original in precision scores and achieves comparable scores in other metrics as well.

### 4.2. Inference Time Costs

We present a comparative analysis of inference time costs for two methods, MotionDiffuse and MotionDiffuse+DCT, on two datasets, KIT-ML and HumanML3D. The results in Table 3 indicate that our approach, specifically the MotionDiffuse+DCT method, outperforms the original model in terms of Inference Time Per Frame (ITPF), Floating Point Operations Per Second (FLOPs), and Multiply-Accumulate operations (MACs). For the HumanML3D dataset, MotionDiffuse+DCT takes about 12.87 ms per frame, which is significantly more efficient compared to the MotionDiffuse method. Similarly, for the KIT-ML dataset, MotionDiffuse+DCT requires approximately 20.5837 ms per frame, showcasing its superior performance. In terms of FLOPs and MACs, our method excels in both forward and backward process calculations, further solidifying its efficiency and effectiveness.

### 4.3. Ablation studies

We undertake straightforward ablation studies on both our proposed MotionDiffuse+DCT version and the MotionDiffuse model to assess the impact on both our approach and the original model. Initially, we conduct inference with a reduced number of coefficients for the MotionDiffuse+DCT model to investigate how the quantity of coefficients influences model performance. As depicted in Table 4, when sampling is performed with half the number of DCT coefficients the model was initially trained on, all metrics exhibit a substantial decrease in model performance. This observation highlights the sensitivity of the model to the difference in the number of DCT coefficients between the training and inference stages. Hence, it becomes evident that determining the optimal number of DCT coefficients is crucial, and a mechanism to identify the detrimental effects of using a specific number of coefficients should be explored prior to training. This process could potentially involve the utilization of an attention-map-like structure, a concept that warrants further investigation.

We then try Denoising Diffusion Implicit Models (DDIM) [23] sampling for both MotionDiffusion+DCT and MotionDiffusion models. DDIM is a class of iterative models that share the same training procedure with DDPMs. DDIM essentially acts as an efficient scheduler that results in faster sampling for models trained by the DDPM training structure [23]. In Tables 5 and 6, the model, trained on DDPM, uses the DDIM sampling technique with 50 sampling steps. While this approach reduces the inference time per frame, it compromises on accuracy. Specifically, the Fréchet Inception Distance (FID), Diversity, and Multimodality metrics underperform when compared to the DDPM counterpart. This indicates that while the DDIM

| Methods | FID↓ | Diversity→ | MultiModality↑ | MM Dist↓ | ITPF(ms) |
|---|---|---|---|---|---|
| MotionDiffuse | 55.3806 | 1.4947 | 0.9341 | 8.0004 | 2.8719 |
| MotionDiffuse+DCT | 59.4548 | 1.2919 | 1.0993 | 8.1486 | 0.61481 |

Table 5. Text-to-motion task performance on HumanML3D test split with *DDIM* sampling steps of 50. The right arrow → indicates higher similarity to real motion is better, and MM Dist stands for MultiModal Distance.

| Methods | FID↓ | Diversity→ | MultiModality↑ | MM Dist↓ | ITPF(ms) |
|---|---|---|---|---|---|
| MotionDiffuse | 83.7064 | 1.9093 | 1.5613 | 9.8833 | 8.015 |
| MotionDiffuse+DCT | 75.384 | 2.0248 | 1.46 | 9.4881 | 0.9805 |

Table 6. Text-to-motion task performance on KIT-ML test split with *DDIM* sampling steps of 50. The right arrow → indicates higher similarity to real motion is better, and MM Dist stands for MultiModal Distance.

sampling technique enhances computational efficiency, it does so at the expense of model performance.

## 5. Conclusion

In summary, this work presents a novel approach to text-driven human motion generation by integrating the Discrete Cosine Transform (DCT) with the MotionDiffuse methodology. By performing diffusion in a lower-dimensional dynamical space using DCT, we have significantly reduced the model complexity and inference time per frame, thereby enhancing computational efficiency. Our MotionDiffuse+DCT model exhibits notable strengths across different datasets. It excels in MultiModality on the HumanML3D dataset and demonstrates superior precision on the KIT-ML dataset compared to the MotionDiffuse model.

**Limitations and Future Work.** It is important to acknowledge instances where the model does not yield optimal quantitative results on other metrics. Additionally, the presented experimental results are constrained by computational and time limitations. Determining the optimal number of DCT coefficients to have the best efficiency/performance trade-off remains an open topic. This issue could be addressed with an attention-map-like structure that would determine how the self and cross-attention blocks in the CMLT attend to the DCT coefficients, potentially providing deeper insights into the approach taken. Furthermore, other dynamical representations, such as the atoms-based representation in DYAN [15] could be investigated as an alternative to DCT to confirm the benefits of using a dynamics-based representation before the diffusion process. In conclusion, we believe that our work highlights a significant issue in human motion generation that necessitates further exploration.

| Configuration | MotionDiffuse | +DCT |
|---|---|---|
| Optimizer | Adam | Adam |
| Batch size | 64 | 64 |
| Learning rate | 0.0002 | 0.0002 |
| First DCT rows | - | 40 |
| Diffusion steps | 1000 | 1000 |
| Motion decoder $L_{dim}$ | 512 | 512 |
| Text encoder $L_{dim}$ | 256 | 256 |
| Max text length | 20 | 20 |
| Text encoding model | bigru | bigru |

Table 7. Hyperparameter configuration for *HumanML3D* dataset. $L_{dim}$ represents the latent space dimension, and +DCT refers to the proposed MotionDiffusion+DCT model.

| Configuration | MotionDiffuse | +DCT |
|---|---|---|
| Optimizer | Adam | Adam |
| Batch size | 32 | 128 |
| Learning rate | 0.0002 | 0.0002 |
| First DCT rows | - | 20 |
| Diffusion steps | 1000 | 1000 |
| Motion decoder $L_{dim}$ | 512 | 512 |
| Text encoder $L_{dim}$ | 256 | 256 |
| Max text lenght | 20 | 20 |
| Text encoding model | bigru | bigru |

Table 8. Hyperparameter configuration for *KIT-ML* dataset. Same format as Table 7.

## References


[1] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974. 3, 4

[2] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. *Advances in Neural Information Processing Systems*, 21, 2008. 2, 3

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 1, 2

[4] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2317–2327, 2023. 2

[5] Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. Humanmac: Masked motion completion for human motion prediction. *arXiv preprint arXiv:2302.03665*, 2023. 2, 3, 4

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[8] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 5, 6

[9] 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. 5

[10] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction, 2022. 4

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[14] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a backpropagation network. *Advances in neural information processing systems*, 2, 1989. 1

[15] Wenqian Liu, Abhishek Sharma, Octavia Camps, and Mario Sznaier. Dyan: A dynamical atoms-based network for video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 170–185, 2018. 2, 3, 8

[16] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 5

[17] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 4

[18] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, dec 2016. 5

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 3, 4, 6

[20] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 1

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[22] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1, 2

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 7

[24] Nicolás Salazar Sutil. *Motion and representation: The language of human movement*. MIT Press, 2015. 2

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3, 4

[26] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1

[27] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2, 3, 4, 5, 6

[28] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 6

[29] Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2