LuKAN: A Kolmogorov-Arnold Network Framework for 3D Human Motion Prediction

Md Zahidul Hasan
zadidhasan11@gmail.com
A. Ben Hamza
hamza@ciise.concordia.ca
Nizar Bouguila
nizar.bouguila@concordia.ca

Concordia University
Montreal, QC, Canada

Abstract

The goal of 3D human motion prediction is to forecast future 3D poses of the human body based on historical motion data. Existing methods often face limitations in achieving a balance between prediction accuracy and computational efficiency. In this paper, we present LuKAN, an effective model based on Kolmogorov-Arnold Networks (KANs) with Lucas polynomial activations. Our model first applies the discrete wavelet transform to encode temporal information in the input motion sequence. Then, a spatial projection layer is used to capture inter-joint dependencies, ensuring structural consistency of the human body. At the core of LuKAN is the Temporal Dependency Learner, which employs a KAN layer parameterized by Lucas polynomials for efficient function approximation. These polynomials provide computational efficiency and an enhanced capability to handle oscillatory behaviors. Finally, the inverse discrete wavelet transform reconstructs motion sequences in the time domain, generating temporally coherent predictions. Extensive experiments on three benchmark datasets demonstrate the competitive performance of our model compared to strong baselines, as evidenced by both quantitative and qualitative evaluations. Moreover, its compact architecture coupled with the linear recurrence of Lucas polynomials, ensures computational efficiency. Code is available at: https://github.com/zadidhasan/LuKAN

1 Introduction

The task of 3D human motion prediction is to forecast the future 3D poses of a human body over a specified time horizon based on historical motion data. It empowers diverse applications requiring dynamic and responsive interaction with human movements, including human-object interaction [7], animation [2], and autonomous driving [8, 32]. In recent years, substantial progress has been made in 3D human motion prediction [3, 6, 11, 17, 19, 21, 28, 31], yet accurately forecasting future motion remains a major challenge due to the intrinsic complexity and variability of human movements. The spatio-temporal nature of human motion further compounds these challenges, requiring models to effectively capture both spatial inter-joint relationships and temporal dynamics across sequential frames.

© 2025. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

State-of-the-art methods have embraced diverse neural network architectures tailored to the spatio-temporal nature of motion data, including Recurrent Neural Networks (RNNs) [10, 13, 23], Graph Convolutional Networks (GCNs) [6, 19, 21, 22, 33], Transformers [1, 5, 22], and Multi-Layer Perceptrons (MLPs) [3, 11]. RNNs excel at modeling sequential dependencies but struggle with long-term sequences. GCN-based approaches capture spatial relationships through graph convolutions, but are prone to oversmoothing. Transformers, leveraging the self-attention mechanism, have quadratic computational complexity with respect to sequence length, requiring substantial computation for effective training. MLP-based models achieve reduced computational overhead, but use fixed activation functions and require deep architectures to model complex relationships. More recently, Kolmogorov-Arnold networks (KANs) have emerged as a compelling alternative to MLPs, demonstrating superior performance in function representation across various tasks, including regression [18], while mitigating spectral bias [30]. Unlike MLPs, KANs leverage learnable activation functions on the edges. Existing GCN- and MLP-based approaches employ the discrete cosine transform (DCT) to encode motion in the frequency domain [11, 21]. However, the reliance on DCT may limit their flexibility in capturing localized motion patterns. Moreover, most GCN- and Transformer-based models incorporate MLPs as their core components for feature learning, inheriting a fundamental drawback of MLPs, namely spectral bias [26].

Proposed Work and Contributions. In this paper, we propose LuKAN, a robust model for 3D human motion prediction based on KANs. It integrates a KAN layer that learns univariate functions parameterized by Lucas polynomials to capture interactions between temporal patterns across joints, and spatial projections that model inter-joint relationships. We summarize our contributions as follows: (1) We propose a novel architecture, leveraging KANs and the discrete wavelet transform to encode temporal information in the motion sequence by decomposing the trajectory of each body joint into low-frequency (coarse-scale) components and high-frequency components (fine-scale). Wavelet functions excel at capturing transient and rapidly changing features in a signal, offering a significant advantage over DCT, particularly for motion data where localized variations and dynamic changes are crucial. For instance, rapid hand gestures (high-frequency components) can be captured at fine scales, while slower, more gradual movements like walking (low-frequency components) can be captured at coarser scales; (2) We design a Temporal Dependency Learner to model both localized motion variations and global trends in human motion; (3) We conduct extensive experiments on benchmark datasets, showing that LuKAN achieves competitive performance with minimal computational overhead.

2 Related Work

RNN-based Methods. RNNs have been extensively used in the early stages of human motion prediction research due to their ability to model temporal dependencies in sequential data [10, 13, 15, 23]. These models excel at capturing temporal patterns, making them suitable for tasks where the sequence order and history play a vital role. However, RNN-based methods are often limited by their inability to effectively capture long-term dependencies and are prone to gradient instability during training, particularly for complex motion sequences.

GCN- and MLP-based Methods. GCNs represent human poses as graphs, with joints as nodes and bones as edges. This graph structure enables GCN-based methods to encode inter-joint dependencies naturally. Mao *et al.* [21] proposed a spatio-temporal network that

applies DCT to input motion sequences to encode the temporal dynamics of joint coordinates in the trajectory space. The network uses GCNs with learnable adjacency matrices to capture spatial dependencies between body joints. Guo *et al.* [11] introduced an effective approach using MLPs on the spatial and temporal dimensions of the DCT-transformed input. However, relying on DCT may constrain the flexibility of these models in capturing localized motion patterns effectively. Moreover, MLPs use fixed activation functions at their nodes, limiting their flexibility to adapt to diverse data patterns. Feng *et al.* [9] introduced MotionWavelet, leveraging 2D wavelet transforms to model human motion patterns in the spatial-frequency domain. However, its reliance on diffusion models with guidance mechanisms to control prediction refinement results in higher computational cost. Our proposed LuKAN framework differs from existing methods in that it employs learnable 1D functions on its edges, allowing the network to adaptively model complex temporal dependencies in motion data. It also employs DWT to encode temporal dependencies in the joint trajectory, allowing the model to capture both coarse and fine-grained motion patterns. While both our model and MotionWavelet leverage wavelet transforms for human motion prediction, they differ significantly in terms of their architectural design and learning methodology. Unlike MotionWavelet [9], which modifies motion signals repeatedly through the diffusion process, LuKAN retains high-frequency details. Moreover, using a KAN layer parameterized with Lucas polynomials provides flexibility and computational efficiency, as they are more efficient to evaluate than the piecewise construction of B-splines used in standard KANs.

3 Method

In this section, we first describe the task at hand. Next, we provide a preliminary background on KANs [18, 30]. Then, we introduce the key building blocks of our network architecture.

Problem Description. Let $\mathbf{X}_{1:L} = (\mathbf{x}_1, \ldots, \mathbf{x}_L)^\intercal \in \mathbb{R}^{L \times K}$ be a history motion sequence of L consecutive 3D human poses, where L is the look-back window, $K = 3J$ in the feature dimension, and J is the total number of body joints. At each time step t, each pose $\mathbf{x}_t \in \mathbb{R}^{1 \times K}$ is a flattened vector formed by concatenating the 3D coordinates of all joints in a single frame. The objective is to construct a predictive model that estimates a motion sequence $\hat{\mathbf{X}}_{L+1:L+T} = (\hat{\mathbf{x}}_{L+1}, \ldots, \hat{\mathbf{x}}_{L+T}) \in \mathbb{R}^{T \times K}$ for the subsequent T timesteps. To this end, we design an efficient model based on Kolmogorov-Arnold networks [18].

Kolmogorov-Arnold Networks. KANs are inspired by the Kolmogorov-Arnold representation theorem [4, 27], which states that any continuous multivariate function on a bounded domain can be represented as a finite composition of continuous univariate functions of the input variables and the binary operation of addition. A KAN layer is a fundamental building block of KANs [18], and is defined as a matrix of 1D functions $\Phi = (\phi_{q,p})$, where each trainable activation function $\phi_{q,p}$ is defined as a weighted combination, with learnable weights, of a sigmoid linear unit (SiLU) function and a spline function. Given an input vector $\mathbf{x}$, the output of an L-layer KAN is given by

$$\mathrm{KAN}(\mathbf{x}) = (\Phi^{(\mathrm{L}-1)} \circ \cdots \circ \Phi^{(1)} \circ \Phi^{(0)})\mathbf{x}, \tag{1}$$

where $\Phi^{(\ell)}$ is a matrix of learnable functions associated with the ℓ-th KAN layer.

Model Architecture. The overall framework of our network architecture is depicted in Figure 1. LuKAN is designed to efficiently predict 3D human motion by modeling both spatial relationships and temporal dependencies in motion data.

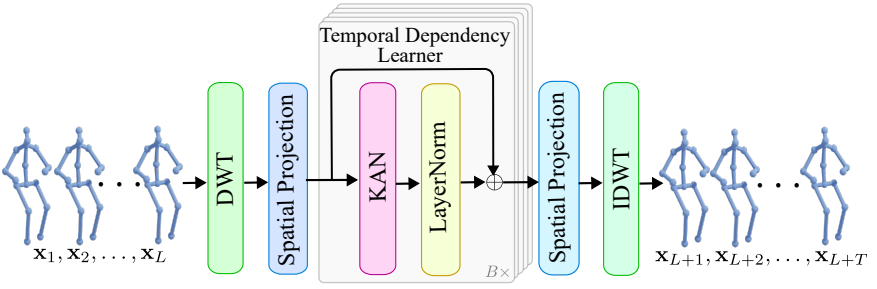

Figure 1: **Overview of Model Architecture**. LuKAN processes input 3D motion data by applying DWT to encode temporal information. A spatial projection is applied both before and after the Temporal Dependency Learner block (repeated B times). Each block consists of a KAN layer, LayerNorm, and a residual skip connection. The inverse DWT (IDWT) reconstructs the motion in the time domain, outputting a sequence of predicted 3D poses.

3.1 Temporal Encoding

Joint Trajectory. The ith column of the history motion sequence $\mathbf{X}_{1:L}$, denoted as $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_L^{(i)})^\intercal$, represents the trajectory of the i-th skeleton joint over the L consecutive frames in the sequence. The coordinates $x_\ell^{(i)}$ at each time step ℓ represent the position of the i-th joint in 3D space at that specific moment. This representation allows for capturing the motion of each joint individually over the observed time window.

Discrete Wavelet Transform Encoding. To encode temporal information of the human motion in the trajectory, we employ DWT, which decomposes a signal into its approximate and detail components using wavelets, ensuring that localized temporal variations in the motion sequence are captured effectively. Specifically, given a wavelet (e.g., Daubechies wavelet), applying a three-level DWT to the i-th joint trajectory $\mathbf{x}^{(i)}$ yields

$$\mathbf{c}^{(i)} = \mathrm{DWT}(\mathbf{x}^{(i)}), \tag{2}$$

where $\mathbf{c}^{(i)} = (\mathbf{a}^{(i)}, \mathbf{d}^{(i)})^\intercal$ is an $(L_a + L_d)$-dimensional vector of wavelet coefficients that describe the signal's approximation and detail components. The approximation coefficients $\mathbf{a}^{(i)} \in \mathbb{R}^{L_a}$ represent the low-frequency (coarse-scale) components of the trajectory, while the detail coefficients $\mathbf{d}^{(i)} \in \mathbb{R}^{L_d}$ represent the high-frequency (fine-scale) variations in the trajectory. Unlike cosine waves, which oscillate indefinitely, wavelet functions are compact, with oscillations that diminish over time, enabling them to localize effectively and capture transient or rapidly changing features in a trajectory, which DCT cannot address as efficiently. The original trajectory can be reconstructed from its wavelet coefficients using the Inverse Discrete Wavelet Transform (IDWT) as follows:

$$\hat{\mathbf{x}}^{(i)} = \mathrm{IDWT}(\mathbf{a}^{(i)}, \mathbf{d}^{(i)}), \tag{3}$$

which takes as input an $(L_a + L_d)$-dimensional vector of wavelet coefficients and returns an L-dimensional reconstructed trajectory, ensuring that essential motion characteristics are preserved while enabling a more localized representation of human motion sequences.

3.2 Spatial Projection

The spatial projection maps the DWT-transformed history motion sequence into an embedding space of dimension D, capturing inter-joint dependencies and providing an expressive representation of the spatial structure of the human body. Its output is an $(L_a + L_d) \times D$ matrix given by

$$\mathbf{Z}_1 = \mathrm{DWT}(\mathbf{X}_{1:L})\mathbf{W}_1 \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{K \times D}$ is a learnable weight matrix, which defines a linear projection along the spatial (i.e., joint) dimension, and D is the embedding dimension. For notational simplicity, the bias term is omitted here and throughout the following subsections.

3.3 Temporal Dependency Learner

The Temporal Dependency Learner is a core component of LuKAN, designed to capture temporal relationships within the motion sequence data. It operates as a sequence modeling block, emphasizing both local and global temporal dependencies to effectively predict future motion, while maintaining computational efficiency. This component consists of three key elements: a single KAN layer, LayerNorm, and a residual skip connection. The design choices are motivated as follows: (1) unlike MLPs, our proposed KAN effectively captures dependencies with its Lucas polynomials as learnable activation functions, providing flexibility in modeling both localized variations (such as fast changes in pose) and global trends (like slow transitions in motion), while reducing the need for excessively deep architectures; (2) LayerNorm helps stabilize training and ensures feature consistency across different motion sequences; and (3) a residual skip connection enhances gradient flow and prevents information loss, mitigating the limitations of purely feedforward architectures.

KAN Layer. We employ a single KAN layer, with associated matrix $\Phi = (\phi_{q,p})$ whose (q,p)-th entry is a function with learnable parameters. Each trainable function $\phi_{q,p}$ is parameterized by a weighted linear combination of Lucas polynomials

$$\phi_{q,p}(x_p) = \sum_{r=0}^{R} \gamma_{q,p,r} P_r(x_p), \tag{5}$$

where x_p represents the p-th element of the joint trajectory vector, and $\gamma_{q,p,r}$ is the learnable coefficient of the r-th Lucas polynomial $P_r(x_p)$ for the q-th output element. These learnable parameters are adjusted during training to optimize the network's performance with the aim of improving the accuracy of the function approximation. In it important to mention that Lucas polynomials are defined recursively, making them computationally efficient to evaluate [25]. Specifically, Lucas polynomials $P_r(x)$ are defined by the linear recurrence relation

$$P_r(x) = xP_{r-1}(x) + P_{r-2}(x), \tag{6}$$

with initial conditions $P_0(x) = 2$ and $P_1(x) = x$. The degree of $P_r(x)$ is equal to r.

Layer Normalization (LN). LN is applied immediately after the KAN layer to standardize the output by normalizing feature activations.

Residual Skip Connection. This skip connection links the input of KAN directly to its output, creating a residual pathway. Specifically, the output of the Temporal Dependency Learner is an $(L_a + L_d) \times D$ matrix given by

$$\mathbf{Z_2} = \mathrm{LN}(\mathrm{KAN}(\mathbf{Z_1})) + \mathbf{Z_1}, \tag{7}$$

where KAN and LN are applied along the temporal dimension.

3.4 Spatial Projection and Inverse Discrete Wavelet Transform

The spatial projection, applied after the Temporal Dependency Learner, refines the spatial relationships between human body joints, ensuring structural consistency in the predicted poses. It models inter-joint dependencies, complementing the initial spatial projection. On the other hand, IDWT maps the temporally processed data back to the time domain. Together, the spatial projection and IDWT refine joint relationships and reconstruct the motion sequence in the time domain, resulting in an $L \times K$ output expressed as:

$$\mathbf{Z}_3 = \text{IDWT}(\mathbf{Z}_2\mathbf{W}_2), \tag{8}$$

where $\mathbf{W}_2 \in \mathbb{R}^{D\times K}$ is a learnable weight matrix. As pointed out in Subsection 3.1, IDWT restores the temporal length from $L_a + L_d$ to the original L, thereby generating an $L \times K$ output $\mathbf{Z}_3$. The spatial projection corrects and reinforces joint relationships after KAN has processed the temporal dependencies, while IDWT ensures that these relationships are translated back into the time domain for motion reconstruction.

Model Prediction. The predicted sequence is a $T \times K$ matrix given by

$$\hat{\mathbf{X}}_{L+1:L+T} = \widetilde{\mathbf{Z}}_3 + \mathbf{X}_L, \tag{9}$$

where T is the prediction horizon, $\widetilde{\mathbf{Z}}_3$ consists of the first T rows of $\mathbf{Z}_3$, and $\mathbf{X}_L \in \mathbb{R}^{T\times K}$ is constructed by replicating the final pose $\mathbf{x}_L$ of the historical motion sequence T times.

Model Training. We train our model using the following loss function

$$\mathcal{L} = \frac{1}{T}\sum_{t=L+1}^{L+T}(\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 + \|\mathbf{v}_t - \hat{\mathbf{v}}_t\|_2), \tag{10}$$

where $\|\cdot\|$ denotes the ℓ_2-norm, $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$ are the predicted and ground truth poses for the t-th predicted frame, $\mathbf{v}_t$ and $\hat{\mathbf{v}}_t$ are the associated velocities, respectively.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experimental evaluations on three standard datasets: Human3.6M [12], Archive of Motion Capture as Surface Shapes (AMASS) [20], and 3D Pose in the Wild dataset (3DPW) [29]. We follow standard protocols [22] for data preprocessing and splitting. Additional results and ablation studies are provided in the supplementary material.

Evaluation Metric and Baselines. We assess the model's performance using the Mean Per Joint Position Error (MPJPE), measured in millimeters, where lower values correspond to better prediction performance. We benchmark LuKAN against several state-of-the-art approaches for 3D human motion prediction, including ConvSeq2Seq [15], Learning Trajectory Dependencies (LTD) [21], History repeats (Hisrep) [22], Dynamic Multiscale Graph Neural Networks (DMGNN) [16], MultiScale Residual Graph Convolution Network (MSR-GCN) [6], Spatial and Temporal Dense Graph Convolutional Network (ST-DGCN) [19],

Context-based Interpretable Spatio-Temporal Graph Convolutional Network (CIST-GCN) [24], MotionMixer [3], Skeleton-Parted Graph Scattering Networks (SPGSN) [17], and Simple Multi-Layer Perceptron (SiMLPe) [11].

Implementation Details. All experiments are performed on a single NVIDIA RTX 3070 GPU with 8GB of memory using PyTorch. Our model is trained for 50K epochs on Human3.6M and 115K epochs on AMASS, using a batch size of 128. We use Adam optimizer [14] with a weight decay of 10^{-4}. The learning rate is initialized at 3×10^{-4} and decayed to 10^{-5} after 30K epochs. The look-back window is set to $L = 50$, with a prediction horizon of $T = 10$ for Human3.6M, and $T = 25$ for AMASS and 3DPW. We employ Daubechies wavelets with 4 vanishing moments in both DWT and IDWT, and we set the number of levels of decomposition to 3. We also set the number of temporal dependency learner blocks to $B = 48$.

4.2 Results and Analysis

Results on Human3.6M. We report the MPJPE errors averaged across all time steps in Table 1 for both short-term (80ms - 400ms) and long-term (560ms - 1000ms) predictions. The results demonstrate the effectiveness of LuKAN compared to the best-performing baseline, SiMLPe. LuKAN consistently achieves lower MPJPE errors across all time steps, with notable relative error reductions. For instance, at the 720ms prediction horizon, LuKAN achieves an MPJPE of 89.9mm compared to 90.1mm for SiMLPe, yielding a relative error reduction of approximately 0.22%. Similarly, at the 1000ms horizon, LuKAN reduces the MPJPE to 109.3mm from SiMLPe's 109.4mm, resulting in a relative error reduction of approximately 0.09%. These results highlight LuKAN's capability to improve upon the state-of-the-art, while maintaining its simple and efficient architecture.

Table 1: Average MPJPE results of our model and baseline methods on Human3.6M for different prediction time steps in milliseconds (ms) ranging from 80ms to 1000ms. These MPJPE errors, measured in millimeters (mm), are averaged across all different actions in the dataset. The best results are shown in **bold**, and the second best results are underlined.

	MPJPE (mm)↓							
	80	160	320	400	560	720	880	1000
ConvSeq2Seq [15]	16.6	33.3	61.4	72.7	90.7	104.7	116.7	124.2
LTD-10-10 [21]	11.2	23.4	47.9	58.9	78.3	93.3	106.0	114.0
Hisrep [22]	10.4	22.6	47.1	58.3	77.3	91.8	104.1	112.1
DMGNN [16]	17.0	33.6	65.9	79.7	103	-	-	137.2
MSR-GCN [6]	11.3	24.3	50.8	61.9	80.0	-	-	112.9
ST-DGCN [19]	10.6	23.1	47.1	57.9	76.3	90.7	102.4	109.7
SPGSN [17]	10.4	22.3	47	58.2	77.4	-	-	109.6
CIST-GCN [24]	10.5	23.2	47.9	59.0	77.2	-	-	110.3
MotionMixer [3]	11	23.6	47.8	59.3	77.8	91.4	106	111
SiMLPe [11]	9.6	21.7	46.3	57.3	**75.7**	90.1	101.8	109.4
LuKAN (ours)	**9.4**	**21.5**	**46.2**	**57.2**	**75.7**	**89.9**	**101.6**	**109.3**

Results on AMASS and 3DPW. We train our model on the AMASS dataset and test it on on the AMASS-BMLrub and 3DPW datasets, adhering to the standard evaluation protocol outlined in [22]. The results in Table 2 provide a comprehensive comparison of our

model against strong baseline methods on the AMASS-BMLrub and 3DPW datasets, evaluated in terms of MPJPE across different prediction horizons. On AMASS-BMLrub, LuKAN achieves competitive results, particularly excelling in short-term predictions. At 80ms and 160ms, LuKAN matches the best-performing LTD-10-10 with MPJPEs of 10.6mm and 19.3mm, respectively. For longer horizons, LuKAN consistently demonstrates robust performance, achieving the second-best MPJPE scores, such as 34.4mm at 320ms and 66.4mm at 1000ms. Compared to SiMLPe at 320ms, for example, LuKAN yields comparable performance, highlighting its ability to stay on par with state-of-the-art models. On the more challenging 3DPW dataset, which evaluates the generalization ability of prediction models, LuKAN consistently outperforms all baselines across all prediction horizons. For instance, at 320ms, LuKAN achieves an MPJPE of 37.9mm, outperforming SiMLPe's 38.1mm with a relative error reduction of 0.52%. At 1000ms, LuKAN achieves an MPJPE of 72.2mm, matching SiMLPe and further underscoring its robustness in generalization. Overall, the combination of competitive performance in short-term predictions and robust results in long-term horizons highlights LuKAN's versatility and ability to balance prediction accuracy and efficiency across different time horizons.

Table 2: **Performance comparison of our model and baselines on AMASS-BMLrub and 3DPW** for various prediction horizons.

	AMASS-BMLrub								3DPW							
	80	160	320	400	560	720	880	1000	80	160	320	400	560	720	880	1000
ConvSeq2Seq [13]	20.6	36.9	59.7	67.6	79.0	87.0	91.5	93.5	18.8	32.9	52.0	58.8	69.4	77.0	83.6	87.8
LTD-10-10 [21]	**10.3**	**19.3**	36.6	44.6	61.5	75.9	86.2	91.2	12.0	22.0	38.9	46.2	59.1	69.1	76.5	81.1
LTD-10-25 [21]	11.0	20.7	37.8	45.3	57.2	65.7	71.3	75.2	12.6	23.2	39.7	46.6	57.9	65.8	71.5	75.5
Hisrep [22]	11.3	20.7	35.7	42.0	51.7	58.6	63.4	67.2	12.6	23.1	39.0	45.4	56.0	63.6	69.7	73.7
SiMLPe [11]	10.8	19.6	**34.3**	**40.5**	**50.5**	**57.3**	**62.4**	**65.7**	12.1	22.1	38.1	44.5	**54.9**	62.4	68.2	**72.2**
Ours	10.6	**19.3**	34.4	40.8	50.9	57.6	62.7	66.4	**11.9**	**21.8**	**37.9**	**44.4**	**54.9**	**62.2**	**68.1**	**72.2**

Qualitative Results. In Figure 2, we present a comparison of our predicted poses with those generated by SiMLPe for the Directions and Eating actions from Human3.6M. To facilitate visual assessment, the predicted frames are overlaid on the ground truth poses, highlighting any deviations. For both actions, our model demonstrates superior alignment with the ground truth, particularly for the Directions action. Notably, the predicted leg positions from our model are closer to the ground truth compared to those predicted by SiMLPe.

4.3 Ablation Study

Effect of Temporal Encoding. The results in Table 3 compare the performance of DWT and DCT for temporal encoding across Human3.6M, AMASS, and 3DPW datasets in terms of MPJPE. On Human3.6Mt, DWT achieves an MPJPE of 89.9mm at 720ms, outperforming DCT's 90.2mm with a relative error reduction of 0.33%. Similarly, at 1000ms, DWT achieves a lower MPJPE of 109.3mm compared to DCT's 109.5mm, yielding a relative error reduction of 0.18%. On AMASS, DWT consistently outperforms DCT across all time steps. For instance, at 400ms, DWT achieves an MPJPE of 40.8mm compared to 41.5mm for DCT, resulting in a relative error reduction of 1.69%. At 1000ms, DWT reduces the MPJPE to 66.4mm compared to DCT's 66.9mm, with a relative error reduction of 0.75%. On 3DPW, the difference between DWT and DCT is less pronounced, but DWT achieves slightly better

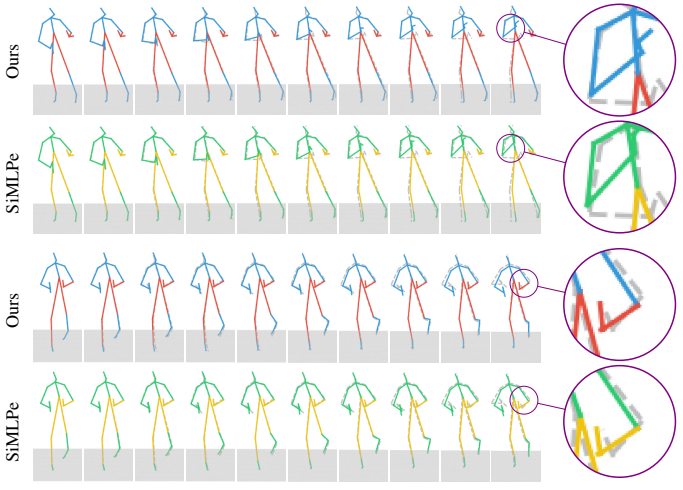

Figure 2: **Visual comparison results of our model and the SiMLPe baseline** on two actions: Directions (top) and Eating (bottom). Predicted poses from our model are depicted in red and blue, while those from SiMLPe are shown in yellow and green. Ground truth poses, represented by dashed lines, are overlaid with the predictions to highlight deviations.

results for most time steps. At 320ms, DWT achieves an MPJPE of 37.9mm compared to 38.4mm for DCT, yielding a relative error reduction of 1.3%. Overall, DWT demonstrates consistent improvements over DCT across all datasets, particularly in short-term predictions.

Table 3: Ablation study on the choice of temporal encoding: DWT vs. DCT across all datasets for various prediction horizons. DWT consistently outperforms DCT.

		MPJPE (mm)↓							
		80	160	320	400	560	720	880	1000
Human3.6M	DCT	**9.4**	**21.4**	**45.8**	**56.8**	**75.7**	90.2	101.8	109.5
	DWT	**9.4**	21.5	46.2	57.2	**75.7**	**89.9**	**101.6**	**109.3**
AMASS	DCT	10.9	19.7	34.9	41.5	51.6	58.7	63.6	66.9
	DWT	**10.6**	**19.3**	**34.4**	**40.8**	**50.9**	**57.6**	**62.7**	**66.4**
3DPW	DCT	12.2	22.2	38.4	44.9	55.1	62.3	**68.1**	**72.2**
	DWT	**11.9**	**21.8**	**37.9**	**44.4**	**54.9**	**62.2**	**68.1**	**72.2**

Effect of Polynomial Basis. The results in Table 4 highlight the superior performance of Lucas polynomials compared to B-splines, used in standard KANs, and other polynomial bases. At 400ms, Lucas polynomials outperform the next best basis, Hermite, yielding a relative error reduction of 0.17%. Similarly, at 1000ms, Lucas polynomials achieve an MPJPE of 109.3mm, outperforming Hermite's 110.1mm by a relative reduction of 0.73%. In comparison to B-splines, the improvements are more pronounced, yielding a relative error reduction of 2.5%. At 320ms, Lucas polynomials achieve an MPJPE of 46.2mm compared to 49.0mm for B-splines, resulting in a relative error reduction of 5.71%. These results demonstrate that Lucas polynomials yield significant improvements over B-splines and other polynomial bases, for both short- and long-term predictions.

Table 4: Ablation study on the choice of the polynomial basis in KAN for various prediction horizons. Lucas polynomials yield significant improvements over B-splines.

Polynomials	MPJPE (mm)↓							
	80	160	320	400	560	720	880	1000
B-Splines	10.3	23.3	49.0	60.1	78.7	92.7	104.5	112.1
Chebyshev	9.7	22.1	47.2	58.3	77.1	91.7	103.7	111.6
Legendre	9.6	21.8	46.8	57.9	76.3	90.4	102.4	110.1
Hermite	9.5	21.6	46.3	57.3	76.0	90.3	102.2	110.1
Lucas	**9.4**	**21.5**	**46.2**	**57.2**	**75.7**	**89.9**	**101.6**	**109.3**

4.4 Model Complexity Analysis

In this section, we analyze the time and memory complexity of LuKAN by considering its main architectural components: spatial projections, DWT and its IDWT, and the Temporal Dependency Learner based on KAN with Lucas polynomial activations.

Time Complexity. Each spatial projection involves a matrix multiplication of complexity $\mathcal{O}(DJL)$, where J is the number of joints, D is the embedding dimension, and L is the length of the input sequence. DWT and its inverse are applied along the temporal dimension. As these are linear-time operations per sequence and per feature, their total complexity is $\mathcal{O}(JL)$. The core component of LuKAN is a B-layer KAN with Lucas polynomial activations, where B is the total number of blocks. Its time complexity is $\mathcal{O}(BDRL^2)$, where R is the degree of the Lucas polynomial. Hence, the time complexity of LuKAN is $\mathcal{O}(DJL + BDRL^2)$.

Memory Complexity. In terms of memory complexity, the model maintains a lightweight parameter count. Each spatial projection require $\mathcal{O}(JD)$ parameters, while the B-layer KAN contributes $\mathcal{O}(BRL^2)$ parameters, giving a total parameter complexity of $\mathcal{O}(JD + BRL^2)$. During runtime, memory is also allocated for storing intermediate activations and for evaluating the polynomial basis, yielding a total runtime memory complexity of $\mathcal{O}(DL + JL)$. Overall, LuKAN achieves a compelling balance between expressive power and computational efficiency.

5 Conclusion

In this work, we proposed LuKAN, an effective model for predicting 3D human motion, inspired by Kolmogorov-Arnold networks. Our model captures both localized temporal dependencies and complex motion dynamics effectively. The model's spatial projections ensure that LuKAN maintains structural consistency while remaining computationally efficient. Through extensive experiments on three benchmark datasets, we demonstrated that our model achieves competitive or superior prediction performance compared to state-of-the-art methods, with significantly fewer parameters and lower computational cost. Notably, LuKAN strikes a good balance between prediction accuracy, efficiency, and model simplicity. For future work, we will explore extending LuKAN to handle multi-person scenarios, and further optimizing its architecture for broader applicability.

References

[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3D human motion prediction. In *Proc. International Conference on 3D Vision*, pages 565–574, 2021.

[2] Samaneh Azadi, Akbar Shah, Thomas Hayes, Devi Parikh, and Sonal Gupta. Make-An-Animation: Large-scale text-conditional 3D human motion generation. In *Proc. IEEE International Conference on Computer Vision*, pages 15039–15048, 2023.

[3] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. MotionMixer: MLP-based 3D human body pose forecasting. In *Proc. International Joint Conference on Artificial Intelligence*, pages 791–798, 2022.

[4] Jürgen Braun and Michael Griebel. On a constructive proof of Kolmogorov's superposition theorem. *Constructive Approximation*, 30:653–675, 2009.

[5] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu, and Nadia Magnenat Thalmann. Learning progressive joint propagation for human motion prediction. In *Proc. European Conference on Computer Vision*, 2020.

[6] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. MSR-GCN: Multi-scale residual graph convolution networks for human motion prediction. In *Proc. IEEE International Conference on Computer Vision*, pages 11447–11456, 2021.

[7] Christian Diller and Angela Dai. CG-HOI: Contact-guided 3D human-object interaction generation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 19888–19901, 2024.

[8] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneiders. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, 2020.

[9] Yuming Fenga, Zhiyang Dou, Ling-Hao Chen, Yuan Liu, Tianyu Li, Jingbo Wang, Zeyu Cao, Wenping Wang, Taku Komura, and Lingjie Liu. MotionWavelet: Human motion prediction via wavelet manifold learning. *arXiv:2411.16964*, 2024.

[10] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. *Proc. IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[11] 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. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, pages 4809–4819, 2023.

[12] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1325–1339, 2014.

[13] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[15] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018.

[16] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3D skeleton-based human motion prediction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 215–223, 2020.

[17] Maosen Li, Siheng Chen, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. Skeleton-parted graph scattering networks for 3D human motion predictiont. In *Proc. European Conference on Computer Vision*, pages 18–36, 2022.

[18] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-arnold networks. In *International Conference on Learning Representations*, 2025.

[19] Tiezheng Ma, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6437–6446, 2022.

[20] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proc. IEEE International Conference on Computer Vision*, 2019.

[21] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proc. IEEE International Conference on Computer Vision*, pages 9489–9497, 2019.

[22] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *Proc. European Conference on Computer Vision*, pages 474–489, 2020.

[23] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4674–4683, 2017.

[24] Edgar Medina, Leyong Loh, Namrata Gurung, Kyung Hun Oh, and Niels Heller. Context-based interpretable spatio-temporal graph convolutional network for human motion forecasting. In *Proc. IEEE Winter Conference on Applications of Computer Vision*, 2024.

[25] Ömer Oruç. A new algorithm based on lucas polynomials for approximate solution of 1D and 2D nonlinear generalized Benjamin-Bona-Mahony-Burgers equation. *Computers and Mathematics with Applications*, 74:3042–3057, 2017.

[26] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proc. International Conference on Maching Learning*, 2019.

[27] Johannes Schmidt-Hieber. The Kolmogorov-Arnold representation theorem revisited. *Neural Networks*, 137:119–126, 2021.

[28] Xiaoning Sun, Huaijiang Sun, Bin Li, Dong Wei, Weiqing Li, and Jianfeng Lu. DeFeeNet: Consecutive 3D human motion prediction with deviation feedback. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5527–5536, 2023.

[29] Timo von Marcard, Roberto Henschel, Michael Black Bodo J., Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *Proc. European Conference on Computer Vision*, 2018.

[30] Yixuan Wang, Jonathan W. Siegel, Ziming Liu, and Thomas Y. Hou. On the expressiveness and spectral bias of KANs. In *International Conference on Learning Representations*, 2025.

[31] Dong Wei, Huaijiang Sun, Xiaoning Sun, and Shengxiang Hug. NeRMo: Learning implicit neural representations for 3D human motion prediction. In *Proc. European Conference on Computer Vision*, pages 409–427, 2024.

[32] Pengxiang Wu, Siheng Chen, and Dimitris Metaxas. MotionNet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 11385–11395, 2020.

[33] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. T2M-GPT: Generating human motion from textual descriptions with discrete representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2023.