Spatio-Temporal Branching for Motion Prediction using Motion Increments

Jiexin Wang
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China
jiexinwang959@gmail.com

Yujie Zhou
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China
yujiezhou@ruc.edu.cn

Wenwen Qiang
University of Chinese Academy of
Sciences
Institute of Software Chinese
Academy of Sciences
Beijing, China
wenwen2018@iscas.ac.cn

Ying Ba
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China
yingba@ruc.edu.cn

Bing Su*
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China
subingats@gmail.com

Ji-Rong Wen
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing Key Laboratory of Big Data
Management and Analysis Methods
Beijing, China
jrwen@ruc.edu.cn

ABSTRACT

Human motion prediction (HMP) has emerged as a popular research topic due to its diverse applications.Traditional methods rely on hand-crafted features and machine learning techniques, which often struggle to model the complex dynamics of human motion. Recent deep learning-based methods have achieved success by learning spatio-temporal representations of motion, but these models often overlook the reliability of motion data. Additionally, the temporal and spatial dependencies of skeleton nodes are distinct. The temporal relationship captures motion information over time, while the spatial relationship describes body structure and the relationships between different nodes. In this paper, we propose a novel spatio-temporal branching network using incremental information for HMP, which decouples the learning of temporal-domain and spatial-domain features, extracts more motion information, and achieves complementary cross-domain knowledge learning through knowledge distillation. Our approach effectively reduces noise interference and provides more expressive information for characterizing motion by separately extracting temporal and spatial features. We evaluate our approach on standard HMP benchmarks and outperform state-of-the-art methods in terms of prediction accuracy. Code is available at https://github.com/JasonWang959/STPMP.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Activity recognition and understanding**.

KEYWORDS

Human motion prediction; Incremental information; Spatio-temporal branching; Knowledge distillation

*Corresponding author

1 INTRODUCTION

Human Motion Prediction (HMP) has become an increasingly popular topic in recent years due to its ability to generate predictions for future human motion based on incomplete fragments of action execution. This technology has found widespread applications in fields such as human-robot interaction [18, 42], autonomous driving [4], and entertainment [40, 43]. Early solutions for this problem relied on state prediction methods, such as linear dynamic systems [34], Restricted Boltzmann Machines [41], hidden Markov models [21], and Gaussian process latent variable models [45]. However, these methods struggle with complex motions. In contrast, the latest approaches for HMP are predominantly data-driven, utilizing deep learning techniques. Numerous deep learning-based methods have emerged, demonstrating significant performance improvements.

Deep learning has made strides in extracting spatio-temporal dependencies of temporal data, which is critical for predicting human motion. Various neural network architectures, including Convolutional Neural Networks (CNNs) [2, 7, 22], Recurrent Neural Networks (RNNs) [6, 9, 11, 13, 27], Long Short-Term Memory

Jiexin Wang, Yujie Zhou, Wenwen Qiang, Ying Ba, Bing Su, and Ji-Rong Wen

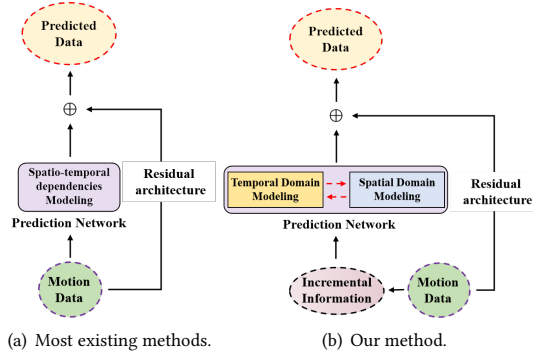(a) Most existing methods. (b) Our method.

Figure 1: Comparison between existing prediction methods and our method. (a): Most existing methods mix modelling spatio-temporal dependencies of motion data. (b): Our approach learns motion knowledge separately in the temporal and spatial domains, and distills the knowledge gained from the learning into interactions (red lines).

(LSTM) [20, 46], Transformer Networks [3], and Generative Adversarial Networks (GANs) [12, 14, 19], have been developed to model these dependencies, but there are some limitations to these methods, such as heavy reliance on filter size in CNN-based methods and prediction error accumulation in RNN-based methods. Recently, Graph Convolution Networks (GCNs) [8, 24, 25, 49] have gained widespread popularity for their ability to model inner-frame kinematic dependencies and learn spatial relations between joints in HMP. Nonetheless, these methods have mainly focused on network structure and feature extraction, little attention has been given to the underlying issues of the motion data itself, and have not fully explored the complementary nature of temporal-domain and spatial-domain information in skeleton data.

Motion data is usually collected through various types of sensors. These sensors may introduce various levels of noise and disturbance to the observation data, such as noise introduced by the sensors themselves or caused by the shooting environment. Such noise or disturbance could corrupt the original motion information and pose a challenge for deep learning models. However, the reliability of these data is often taken for granted and has not been fully explored. In this paper, to enhance the reliability and authenticity of the observation data, we propose a new learning approach. Specially, we explore the incremental information of motion data to improve the performance of the prediction model. We confirm the effectiveness of the incremental information and show that it can effectively reduce noise interference in the original observation data, thereby better characterizing action information. As demonstrated in Figure 2, incorporating incremental information results in a noticeable improvement in the prediction performance.

In addition, current approaches to modeling the skeleton data have limitations in that they only focus on modeling either temporal-domain or spatial-domain features, without fully utilizing the complementary nature of these two features [2, 10, 28]. Although previous studies have attempted to encode spatio-temporal relationships of human pose using a mixture of different convolutional kernels, they often ignore the complementarity between these two types of features [14, 29]. The mixing of temporal and spatial features may result in interference and conflicts between the two features,
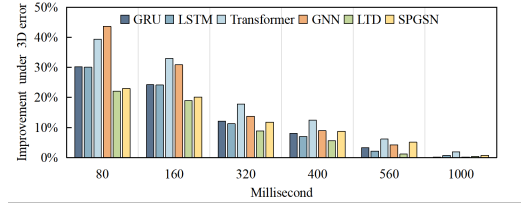

Figure 2: Performance improvement with incremental information on toy experiments. The table shows the percentage improvement in performance of the mean per joint position error for different networks using either incremental information or the original motion data as input. The experiments involve predicting 15 poses in the next 1000ms, based on 10 initial poses as input in the Human 3.6M dataset.

leading to degraded predictive performance. Moreover, skeleton data is fragile and sensitive to noise, which can be exacerbated by the mixing of features. In this paper, we propose a novel spatio-temporal dual flow branch network (as shown in Figure 1(b)), which separately considers both temporal smoothness and spatial dependencies among human body joints and then combines them through knowledge distillation. Our proposed approach provides a simple yet effective solution for modeling spatio-temporal characteristics of motion data, with promising results demonstrated in our experiments. The contributions of this paper are summarized as follows:

(1) We propose a new learning approach for human motion prediction, which effectively reduces or eliminates noise interference in the original observation data, resulting in a more accurate characterization of action information.

(2) We propose a simple spatio-temporal dual flow branch network that separately extracts the temporal-domain and spatial-domain features of human motion, and then fuses the two parts of knowledge using knowledge distillation.

(3) We conduct extensive experiments on different standard human motion prediction benchmarks demonstrating the effectiveness of the proposed method. The empirical evaluations indicate that our approach outperforms state-of-the-art methods on these datasets.

2 RELATED WORK

Human Motion Prediction. Traditional methods for HMP combine hand-crafted features with machine learning techniques [21, 26, 33, 40, 45]. However, these methods may struggle to model the complex dynamics of human motion and require substantial feature engineering for accurate predictions. Deep learning-based methods have achieved great success in HMP owing to their ability to learn complex spatio-temporal representations of human motion. RNNs [6, 9, 11, 13, 27] have been used to model the temporal dynamics of human motion, while CNNs [2, 7, 22] have been used to model the spatial information. Recent advances in HMP have focused on GCN [8, 24, 25, 29, 39], which captures the temporal and spatial dependencies of human joints and has achieved remarkable success.However, these methods can suffer from being over-smooth and cannot handle long-term motion predictions. Differently, our method aims to exploit how to better characterizing motion and focus on separately modeling the temporal-domain and spatial-domain information to achieve more accurate predictions.

Spatio-temporal Representation. Existing methods in HMP focus on improving the model or designing spatio-temporal feature extraction methods [23, 29, 39]. These methods usually mix spatio-temporal features together for learning or only use one type of feature. Despite their success, these approaches have limitations, such as low learning efficiency due to the complex spatio-temporal features, varying performance across different action categories, and sensitivity to external factors such as noise. Recently, the residual architecture learning framework based on DCT (Discrete Cosine Transform) has gradually become mainstream in HMP [30, 31]. Although DCT can model the time dependency of each joint node well, it tends to ignore the constraint relationships between joint nodes. Although GCNs have partly alleviated this issue, existing GCN models do not consider the temporal relationships in action sequences and interactions between spatio-temporal features [1, 8, 24, 29, 39, 48]. Differently, we focus on separately considering both temporal smoothness and spatial dependencies among human body joints and utilizing the complementary nature of these features to improve prediction performance.

3 PROBLEM STUDY AND ANALYSIS

3.1 Problem Formulation

The human motion prediction problem involves generating future pose sequences based on historical ones. Mathematically, the given historical poses can be represented as $\mathbf{X} = [X_1, \cdots, X_{T_p}] \in \mathbb{R}^{T_p \times J \times D}$, where T_p frames are recorded and each pose $X_t \in \mathbb{R}^{J \times D}$ depicts a human pose with J joints in D dimensions at time t. Typically, D is equal to 2 or 3, representing the 2D or 3D case, respectively. Similarly, the target future pose sequence can be defined as $\mathbf{Y} = [X_{T_p+1}, \cdots, X_{T_p+T_f}] \in \mathbb{R}^{T_f \times J \times D}$. Thus, the problem can be formally defined as the task of predicting $\mathbf{Y}$ given $\mathbf{X}$. Following [8, 30, 31], we pad the last observed pose X_{T_p} by repeating it T_f times and append the resulting poses to $\mathbf{X}$. This gives us the padded input sequence $\mathbf{X} = [X_1, \cdots, X_{T_p}, X_{T_p}, \cdots, X_{T_p}] \in \mathbb{R}^{(T_p+T_f) \times J \times D}$. Correspondingly, we construct $\mathbf{Y} = [X_1, \cdots, X_{T_p}, X_{T_p+1}, \cdots, X_{T_p+T_f}] \in \mathbb{R}^{(T_p+T_f) \times J \times D}$. Formally, the problem can be defined as follows:

$$\widetilde{\mathbf{Y}} = \mathcal{F}(\mathbf{X}; \Theta), \tag{1}$$

where $\mathcal{F}$ is a function that maps the input sequence $\mathbf{X}$ to the predicted pose $\widetilde{\mathbf{Y}}$, and Θ represents the function's parameters. The objective is to learn $\mathcal{F}$ such that the predicted motions $\widetilde{\mathbf{Y}}$ are as accurate as possible compared to the ground-truth motions $\mathbf{Y}$. To this end, we propose a spatio-temporal branching for motion prediction using incremental information, which is detailed in the following sections. An overview of our pipeline is shown in Figure 4.

3.2 Analysis from the Causal Perspective

The concepts of direct and indirect causal effects were introduced in [35], and we illustrate the corresponding causal diagram in Figure 3. Figure 3(a) denotes the direct causal effect of X on Y. In contrast, Figure 3(b) denotes the indirect causal effect of X on Y, where X affects Y through the intermediary variable M. If we set $X = x$ and $M = m$, then the indirect causal effect of X on Y can be denoted as:
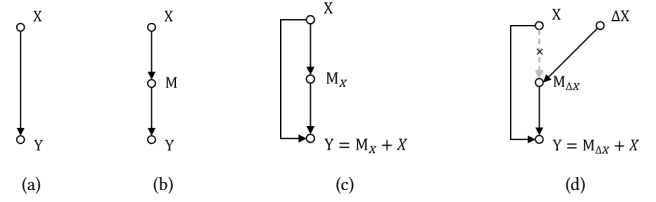
$$Y_{x,m} = Y(X = x, M = m). \tag{2}$$

(a) (b) (c) (d)

Figure 3: Comparison of different models for motion prediction. X **denotes the original observed data,** Δ_X **represents the incremental information with respect to** X**, and the intermediate variable** M **denotes the learning representation based on either** X **or** Δ_X**. The final output is denoted by** Y**.**

Figure 3(c) indicates that X has both direct and indirect causal effects on Y. Traditional methods for HMP usually involve a direct mapping from observed input sequences to future sequences. This direct mapping can be associated with the indirect causal effect. In contrast, the prediction network with residual architecture has become the primary architecture in HMP, learning to output the difference between predicted and observed data. This architecture can correspond to the combined effect of both direct and indirect causal effects shown in Figure 3(c). Therefore, we can see that the quality of the final results is mainly influenced by two factors: the input variable X and the intermediate variable M, corresponding to the motion data and the deep learning model, respectively.

Actually, during the process of data collection, noise from sensors, recording equipment, or other sources of interference can compromise the quality of motion data. Therefore, we define observed data as a combination of ground truth values and noise:

$$X = X_{gt} + \varepsilon. \tag{3}$$

In Equation 3, X_{gt} represents the ground truth motion data concerning X, while ε represents the noise that arises from various external factors during observations. Properly handling noise is crucial for improving the reliability and authenticity of motion data, as accurate data is essential for providing a solid foundation for subsequent action analysis and prediction tasks. In HMP, where multiple frames are captured per second with a small time interval between them, external noise perturbation is nearly constant over short periods. Therefore, we can assume that $\Delta \varepsilon_t \approx 0$ for adjacent frames at time t. We define the difference between the current frame and the next frame as incremental information. To simplify the description, we focus on time t, which yields:

$$\begin{aligned} \Delta X_t &= X_{t+1} - X_t \\ &= (X_{gt_{t+1}} - X_{gt_t}) + (\varepsilon_{t+1} - \varepsilon_t) \\ &= \Delta X_{gt_t} + \Delta \varepsilon_t. \end{aligned} \tag{4}$$

In Equation 4, Δ represents the variation magnitude between adjacent frames. When $\Delta \varepsilon_t \approx 0$, then $\Delta \mathbf{X}_t \approx \Delta \mathbf{X}_{gt}$, providing us with incremental information about the real data from the observed data. This suggests that incremental information can reduce or eliminate the negative impact of noise on the reliability of input samples.

Motivated by the above analysis, we utilize the counterfactual operation shown in Figure 3(d). This operation enables us to replace the neural network's training data with incremental information that eliminates the impact of noise while preserving the residual learning structure in HMP. Additionally, residual learning in HMP
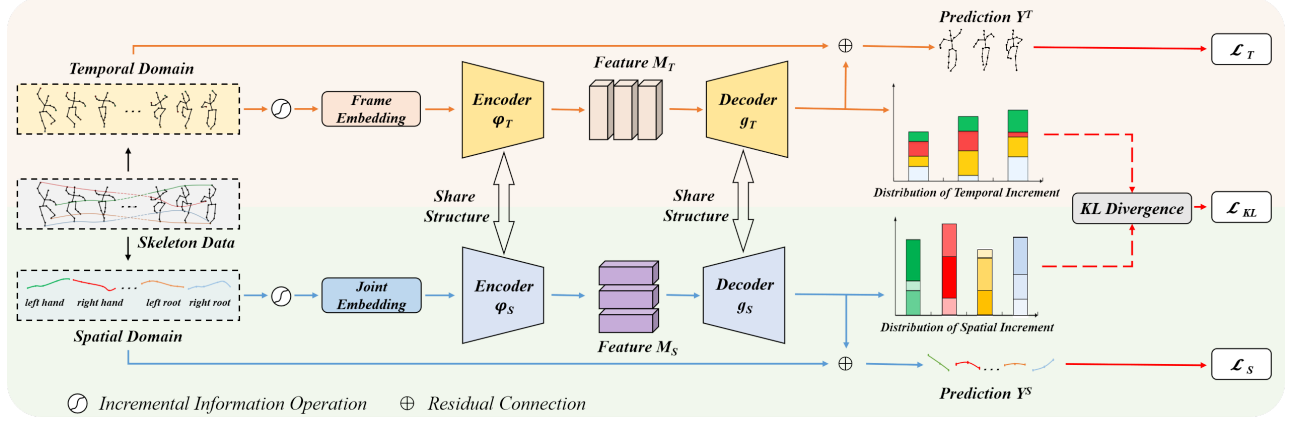
Figure 4: Overview of the proposed method. Orange lines and blue lines indicate the pipeline of the prediction based on the temporal domain data and the prediction based on the spatial domain data (Section 4.1), respectively. Except for the difference in the input dimension and the output dimension, the network structures of φ_t and φ_s, g_t and g_s are the same. Red lines indicate the pipeline of the computation to loss (Section 4.3), which are only performed during training.

involves learning a residual vector between the padding sequence and the ground-truth sequence, which suggests that incorporating incremental information into the prediction process may improve performance by better aligning the input and output. Specifically, compared to Figure 3(c), in Figure 3(d), Δx is given as input to the intermediary variable M, resulting in $M = f(X = \Delta x)$. This enables us to obtain both the indirect causal effect of Δx on Y through M and the direct causal effect of X on Y. We refer to the resulting value of Y after the counterfactual operation as:

$$Y_{x,M_{\Delta x}} = Y(X = x, M = f(X = \Delta x)) \tag{5}$$

To verify the positive effect of the counterfactual operation with the replacement of incremental information in Figure 3(d), we use the concept of indirect causal effects, which measures the impact of an intermediate variable M on variable Y when all other variables remain constant. Specifically, we compare the difference between $Y_{x,M_{\Delta x}}$ in Figure 3(d) and Y_{x,M_x} in Figure 3(c). The former represents the value of Y when $M = f(X = \Delta x)$, while the latter represents the value of Y when $M = f(X = x)$. Although both quantities have the same direct causal effect $X \rightarrow Y$, they differ in the indirect causal effect $X \rightarrow M \rightarrow Y$. The indirect causal effect accounts for the difference in the output Y when the intermediate variables are changed. Specifically, the indirect causal effect is defined as follows:

$$ITE = \mathbb{E}[Y_{x,M_{\Delta x}} - Y_{x,M_x}]. \tag{6}$$

In HMP, the performance metric used for evaluation is the prediction error value e. The indirect causal effect can be defined as the proportional reduction rate of the error value, which can be expressed mathematically as $ITE = \mathbb{E}[Y_{x,M_{\Delta x}} - Y_{x,M_x}] \propto \frac{e_x - e_{\Delta x}}{e_x}$. To confirm the validity of our proposed counterfactual structural model, we conducted a toy experiment on several baselines, shown in Figure 2, to obtain indirect causal effects. The results demonstrate that the counterfactual operation enhances the model's indirect causal effect by providing incremental information. In the following section, we propose a design for the intermediary variable M to enhance the prediction performance of Y.

4 METHODOLOGY

4.1 Spatio-temporal Branching Framework

Existing methods for HMP have mainly focused on improving the model or designing spatio-temporal feature extraction methods. These methods combine spatio-temporal features or rely on a single type of feature [24, 25, 39], thus ignore the complementary nature of the temporal and spatial properties of human motion sequences. Additionally, these methods can have low learning efficiency due to the complexity of the convolutional or multi-stage learning design. In contrast, the effectiveness of decoupling spatio-temporal information has been demonstrated in some tasks [5, 47].

Motivated by [36–38], we propose a straightforward spatio-temporal branching framework, as depicted in Figure 4. The model comprises a temporal branch and a spatial branch, which encodes the input from the temporal domain and the spatial domain, respectively. The temporal branch encodes a sequence in the temporal dimension, resulting in a temporal-domain feature. Similarly, the spatial branch encodes it in the spatial dimension, resulting in a spatial-domain feature. Both representations are employed for cross-domain mutual distillation to improve the space-time cross-interaction. This decoupling approach simplifies the complexity of a joint model across the diverse spatial and temporal dimensions.

Temporal Domain Modeling. Given a sequence $\mathbf{X}$, we first reshape it into the time-majored domain as $\mathbf{X}^T = \{x_i^T\}_{i=1}^{T_p+T_f} \in \mathbb{R}^{(T_p+T_f) \times J \cdot D}$, where x_i^T denotes the i-th frame that is essentially an entire human skeleton. Based on the analysis in Section 3.2, we extract incremental information $\Delta \mathbf{X}^T = \{\Delta x_i^T\}_{i=1}^{T_p+T_f} \in \mathbb{R}^{(T_p+T_f) \times J \cdot D}$ from $\mathbf{X}^T$, which is then used as input to the designed model. Besides, a frame embedding is employed to project $\Delta \mathbf{X}^T$ into a C-dimensional dense feature space through two fully connected layers. Formally, we obtain the projected feature from Δx_i^T as:

$$\hat{\Delta x_i^T} = W_{t2}(\sigma(W_{t1}\Delta x_i^T + b_1)) + b_2, \tag{7}$$

where $W_{t1} \in \mathbb{R}^{C \times J \cdot D}$ and $W_{t2} \in \mathbb{R}^{C \times C}$ are transformation matrix, $b_1 \in \mathbb{R}^C$ and $b_2 \in \mathbb{R}^C$ indicate the bias vectors, and σ is the ReLU activation function. All projected features are jointly denoted as

$\Delta \hat{\mathbf{X}}^T = \{\Delta \hat{x}_i^T\}_{i=1}^{T_p+T_f} \in \mathbb{R}^{(T_p+T_f) \times C}$. However, since each frame is independently represented in $\Delta \hat{\mathbf{X}}^T$, it lacks the temporal dependencies required for HMP. To overcome this limitation of traditional sequence modeling methods such as LSTM, and RNN, which mainly focus on information transmission between adjacent frames, we use GCN to model the temporal dependencies in the skeleton data. GCN can capture interactions between frames over longer distances in the temporal domain and model each temporal frame as a node, effectively capturing relationships between different frames. This approach allows for better extraction of global information.

$$M_T = \varphi_T(\Delta \hat{\mathbf{X}}^T) \in \mathbb{R}^{(T_p+T_f) \times C}, \tag{8}$$

where φ_T is a GCN-based encoder, which is used to model the temporal dependency of $\Delta \mathbf{X}^T$ and generate the temporal domain representation M_T. Instead of using a pre-defined graph structure, the encoder learns the graph connectivity during training, enabling it to capture long-range dependencies in the time dimension. Following the encoder, we employ a one-layer GCN as the decoder g_T to predict the incremental information of the entire sequence simultaneously. The predicted result is as following:

$$Y^T = g_T(M_T) + \mathbf{X} \in \mathbb{R}^{(T_p+T_f) \times J \times D}. \tag{9}$$

Spatial Domain Modeling. Modeling the input data in the spatial domain is similar to modeling in the temporal domain. Given $\mathbf{X}$, we reshape it into the space-majored domain as $\mathbf{X}^S = \{x_j^S\}_{j=1}^J \in \mathbb{R}^{(J \cdot D) \times (T_p+T_f)}$, where x_j^S denotes the j-th joint that is essentially a joint trajectory along the time. Similarly, Incremental information $\Delta \mathbf{X}^S$ is extracted, and a joint embedding is used to obtain the joint representation $\Delta \hat{\mathbf{X}}^S \in \mathbb{R}^{(J \cdot D) \times C}$. As each joint is represented independently in $\Delta \hat{\mathbf{X}}^S$, they naturally lack spatial dependencies that are crucial for HMP. To capture these dependencies, we treat the human pose as a generic graph, perform GCN on the graph, and obtain the joint representation M_S incorporating both temporal and spatial information. This joint representation serves as input to the decoder for generating the predicted motion sequence.

$$M_S = \varphi_S(\Delta \hat{\mathbf{X}}^S) \in \mathbb{R}^{J \cdot D \times C}, \tag{10}$$

$$Y^S = g_S(M_S) + \mathbf{X} \in \mathbb{R}^{(T_p+T_f) \times J \times D}, \tag{11}$$

φ_S and g_S are the encoder and decoder of the spatial domain representation, respectively. We employ the learned graph connectivity to enable the network to capture long-range dependencies beyond the human kinematic tree.

4.2 Cross-domain Knowledge Distillation

Motivated by the fact that spatial and temporal domain modelings share the same underlying objective despite having different forms, and the knowledge learned from each domain can complement each other. To enhance the interplay of spatial joints and temporal patterns and reduce modal bias, cross-domain knowledge distillation is essential. By distilling knowledge between domains, each domain can receive information from other perspectives, reducing the modal bias in the learned representation. We propose a knowledge distillation mechanism that leverages spatial information to refine the temporal branch and vice versa, resulting in a better integration of spatial-domain and temporal-domain information.

We propose to represent the output of our method (the incremental information: $\Delta \hat{\mathbf{Y}}^T = g_T(M_T)$ and $\Delta \hat{\mathbf{Y}}^S = g_S(M_S)$) to properly model the knowledge learned in each domain. This incremental information can be viewed as the motion energy of each joint, which reflects the activity level of different joints and the characteristics of skeleton movements. To capture the temporal variation trend of joints and reveal the dynamic changes of the skeleton in the temporal dimension, we model the motion energy of the same joint at different time steps as a distribution, and we model each frame's skeleton as a motion energy distribution.

Let $Q^{TT} = \{q_t^{TT}\} \in \mathbb{R}^{(T_p+T_f) \times J}, t = 1, 2, \cdots, (T_p+T_f)$. The $q_{t,i}^{TT}$ is computed as follows:

$$q_{t,i}^{TT} = \frac{exp(\Delta \hat{\mathbf{Y}}^T{}_{t,i})}{\sum_{j=1}^J exp(\Delta \hat{\mathbf{Y}}^T{}_{t,j})}, i = 1, 2, \cdots, J. \tag{12}$$

Similarly, let $Q^{TS} = \{q_j^{TS}\} \in \mathbb{R}^{J \times (T_p+T_f)}, j = 1, 2, \cdots, J$. The $q_{j,t}^{TS}$ is computed as follows:

$$q_{j,t}^{TS} = \frac{exp(\Delta \hat{\mathbf{Y}}^T{}_{t,j})}{\sum_{t=1}^{T_p+T_f} exp(\Delta \hat{\mathbf{Y}}^T{}_{t,j})}, t = 1, 2, \cdots, (T_p+T_f). \tag{13}$$

Moreover, let $Q^{ST} = \{q_t^{ST}\} \in \mathbb{R}^{(T_p+T_f) \times J}, t = 1, 2, \cdots, (T_p+T_f)$. The $q_{t,i}^{ST}$ is computed as follows:

$$q_{t,i}^{ST} = \frac{exp(\Delta \hat{\mathbf{Y}}^S{}_{t,i})}{\sum_{j=1}^J exp(\Delta \hat{\mathbf{Y}}^S{}_{t,j})}, i = 1, 2, \cdots, J. \tag{14}$$

Finally, let $Q^{SS} = \{q_j^{SS}\} \in \mathbb{R}^{J \times (T_p+T_f)}, j = 1, 2, \cdots, J$. The q_i^{SS} is computed as follows:

$$q_{j,t}^{SS} = \frac{exp(\Delta \hat{\mathbf{Y}}^S{}_{t,j})}{\sum_{t=1}^{T_p+T_f} exp(\Delta \hat{\mathbf{Y}}^S{}_{t,j})}, t = 1, 2, \cdots, (T_p+T_f). \tag{15}$$

Q^{TT}, Q^{TS}, Q^{ST}, and Q^{SS} describe the distribution characteristics of each domain's incremental information. To perform knowledge distillation, one intuitive approach is to establish consistency constraints directly between skeleton domains based on these probability distributions. In contrast to previous knowledge distillation methods, which transfer knowledge from a fixed and well-trained teacher model to a student, our approach continuously updates knowledge during different domains, with each domain acting as both student and teacher. To enable knowledge distillation between the temporal and spatial domains, we minimize the KL divergence in the loss function, which facilitates joint interaction in both domains. Specifically, given Q^{TT}, Q^{TS}, Q^{ST}, and Q^{SS}, we have:

$$\mathcal{L}_{KL} = \sum_{t=1}^{T_p+t_f} KL(q_t^{TT} \parallel q_t^{ST}) + \sum_{j=1}^J KL(q_j^{SS} \parallel q_j^{TS}), \tag{16}$$

where KL denotes the KL divergence and $q_t^{TT}, q_t^{ST}, q_j^{TS}$, and q_j^{SS} denote the distributions of the incremental information for each joint or frame in the respective domains. This loss function enables direct consistency constraints between the skeleton domains and allows continuous knowledge transfer between domains.

Table 1: Comparisons of short-term prediction MPJPEs on H3.6M. Results for motion prediction at 80ms, 160ms, 320ms, and 400ms in the future are shown, along with the average MPJPEs across all actions. The best results are highlighted in bold.

Motion	Walking				Eating				Smoking				Discussion			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
DMGNN	17.32	30.67	54.56	65.20	10.96	21.39	36.18	43.88	8.97	17.62	32.05	40.30	17.33	34.78	61.03	69.80
HisRep	10.53	19.96	34.88	42.05	7.39	15.53	31.26	38.58	7.17	14.54	28.83	35.67	10.89	25.19	56.15	69.30
MSR-GCN	12.16	22.65	38.64	45.24	8.39	17.05	33.03	40.43	8.02	16.27	31.32	38.15	11.98	26.76	57.08	69.74
STSGCN	16.26	24.63	40.06	45.94	14.32	22.14	37.91	45.03	13.10	20.20	37.71	44.65	14.33	24.28	52.62	68.53
SPGSN	10.14	19.39	34.80	41.47	7.07	14.85	30.48	37.91	6.72	13.79	27.97	34.61	10.37	23.79	53.61	67.12
PGBIG	11.20	22.15	37.77	43.19	6.73	14.89	30.03	37.42	7.40	15.88	31.30	38.30	50.19	23.85	52.12	65.42
Ours	**4.54**	**14.51**	**32.32**	**40.02**	**2.82**	**10.24**	**27.41**	**35.66**	**2.71**	**9.64**	**24.77**	**32.00**	**3.89**	**15.79**	**46.91**	**61.50**

Motion	Directions				Greeting				Phoning				Posing			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
DMGNN	13.14	24.62	64.68	81.86	23.30	50.32	107.30	132.10	12.47	25.77	48.08	58.29	15.27	29.27	71.54	96.65
HisRep	7.77	18.23	41.34	51.61	15.47	34.04	73.77	88.90	9.78	20.98	39.81	50.87	13.23	27.70	63.68	81.82
MSR-GCN	8.61	19.65	43.28	53.82	16.48	36.95	77.32	93.38	10.10	20.74	41.51	51.26	12.79	29.38	66.95	85.01
STSGCN	14.24	24.27	44.24	53.21	15.02	30.70	67.11	87.63	14.88	21.40	46.55	52.03	15.01	25.69	58.38	73.08
SPGSN	7.35	17.15	39.80	50.25	14.64	32.59	70.64	86.44	8.67	18.32	38.73	48.46	10.73	25.31	59.91	76.46
PGBIG	7.61	18.91	44.63	56.39	14.28	31.48	64.27	**77.91**	8.85	18.84	39.26	49.11	10.40	24.89	57.34	73.59
Ours	**2.65**	**11.22**	**34.98**	**46.56**	**6.02**	**22.95**	**63.55**	81.73	**3.41**	**12.52**	**33.83**	**44.37**	**3.88**	**16.42**	**51.89**	**70.36**

Motion	Purchases				Sitting				Sitting Down				Taking Photo			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
DMGNN	21.35	38.71	75.67	82.74	11.92	25.11	44.59	50.20	14.95	32.88	77.06	93.00	13.61	28.95	45.99	58.76
HisRep	14.75	32.39	66.13	79.64	10.53	21.99	46.26	57.80	16.10	31.63	62.45	76.84	9.89	21.01	44.56	56.30
MSR-GCN	14.63	32.81	65.18	78.27	10.21	20.36	43.68	53.62	15.54	29.97	59.31	72.25	9.09	20.10	44.60	55.72
STSGCN	15.26	26.26	63.45	74.25	15.19	22.95	46.82	58.34	16.70	28.05	56.15	72.03	16.61	24.84	45.98	61.79
SPGSN	12.75	28.58	61.01	74.38	9.28	19.40	42.25	53.56	14.18	27.72	56.75	70.74	8.79	18.90	41.49	52.66
PGBIG	13.43	30.37	61.34	74.87	9.08	19.72	42.96	54.51	14.81	30.92	58.60	71.00	8.41	18.66	40.90	51.89
Ours	**5.01**	**19.14**	**52.93**	**67.77**	**4.05**	**13.08**	**36.39**	**48.36**	**7.09**	**20.28**	**50.24**	**65.27**	**3.63**	**12.89**	**36.33**	**48.47**

Motion	Waiting				Walking Dog				Walking Together				Average			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
DMGNN	12.20	24.17	59.62	77.54	47.09	93.33	160.13	171.20	14.34	26.67	50.08	63.22	16.95	33.62	65.90	79.65
HisRep	10.58	23.75	49.30	60.26	21.77	43.38	78.53	90.21	9.88	19.51	35.91	42.60	11.50	24.40	49.75	60.78
MSR-GCN	10.68	23.06	48.25	59.23	20.65	42.88	80.35	93.31	10.56	20.92	37.40	43.85	12.11	25.56	51.64	62.93
STSGCN	16.30	27.33	48.12	59.79	16.48	37.63	70.60	86.33	11.38	22.39	39.90	47.48	15.34	25.52	50.64	60.61
SPGSN	9.21	19.79	43.10	54.14	17.83	37.15	71.74	84.91	8.94	18.19	33.84	40.88	10.44	22.33	47.07	58.26
PGBIG	8.88	19.89	43.66	54.53	20.78	41.55	72.66	84.53	9.22	19.64	36.42	43.60	10.76	23.44	47.55	58.42
Ours	**3.56**	**13.32**	**37.80**	**50.01**	**7.91**	**27.70**	**66.12**	**81.48**	**3.53**	**13.24**	**31.84**	**39.61**	**4.31**	**15.53**	**41.82**	**54.21**

4.3 Loss Function

To train the proposed model, we use the average L_2 distance between the ground-truth joint positions and the predicted ones as the loss function. Formally, for one training sample, we have:

$$\mathcal{L}_T = \frac{1}{(T_p + T_f) \cdot J} \sum_{n=1}^{T_p+T_f} \sum_{j=1}^{J} \| Y_{n,j}^T - \mathbf{Y}_{n,j} \|^2, \qquad (17)$$

$$\mathcal{L}_S = \frac{1}{(T_p + T_f) \cdot J} \sum_{n=1}^{T_p+T_f} \sum_{j=1}^{J} \| Y_{n,j}^S - \mathbf{Y}_{n,j} \|^2. \qquad (18)$$

To jointly train the model, we define the final loss function $\mathcal{L}$ as the sum of $\mathcal{L}_t$, $\mathcal{L}_s$, and the KL-divergence based loss $\mathcal{L}_{kl}$, weighted by a hyperparameter λ. The final loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s + \lambda \cdot \mathcal{L}_{KL}. \qquad (19)$$

This loss function considers both the temporal smoothness and spatial dependencies among human body joints. Additionally, to combine the knowledge learned from both domains, we use the weighted average of Y^T and Y^S as the final prediction:

$$\widetilde{\mathbf{Y}} = \frac{1}{2} \cdot (Y^T + Y^S). \qquad (20)$$

5 EXPERIMENTS

5.1 Datasets and baselines

Human 3.6M (H3.6M). H3.6M [16] consists of motion capture data of 11 actors performing 15 common actions. Following the standard paradigm [25, 32], we train the models on the segmented clips in the 6 subjects and test on the specific clips in the 5th subject.

CMU-Mocap. The CMU-Mocap dataset has been widely used in various applications. Following the evaluation protocol used in previous works [8, 24, 31], we also evaluated the performance on 8 human action categories.

3D Pose in the Wild (3DPW). 3DPW [44] is the first dataset in the wild to provide accurate 3D pose annotations for evaluation. It contains more than 51,000 frames for challenging indoor and outdoor activities, and official organizers have defined data splits for training, validation, and testing, along with evaluation protocols.

Baselines. We compare our approach with several state-of-the-art baselines, including Res-sup [32], Traj-GCN [31], DMGNN [25], HisRep [30], STSGCN [39], MSR-GCN [8], SPGSN [24], PGBIG [29].

Evaluation Metric. Mean Per Joint Position Error (MPJPE) is the most widely used evaluation metric in HMP, which computes the average ℓ_2 distances between ground truth and predicted joint positions in 3D Euclidean space.

Table 2: Comparison of long-term motion prediction results on the H3.6M dataset. The table presents the prediction performance at 560ms and 1000ms in the future.

Motion	Walking		Eating		Smoking		Discussion		Directions		Greeting		Phoning		Posing	
millisecond	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K
DMGNN	71.36	85.82	58.11	86.66	50.85	72.15	**81.90**	106.32	102.06	135.75	144.51	170.54	71.33	108.37	125.45	188.18
MSR-GCN	52.72	63.05	52.54	77.11	49.45	71.64	88.59	117.59	71.18	100.59	116.24	147.23	68.28	104.36	116.26	174.33
STSGCN	57.64	66.74	58.46	75.08	55.55	74.13	84.20	**107.74**	75.61	109.89	**79.32**	**103.75**	79.19	109.88	**80.82**	**107.58**
SPGSN	46.89	53.59	49.76	**73.39**	46.68	68.62	89.68	118.55	70.05	**100.52**	110.98	143.21	66.70	**102.52**	110.34	165.39
PGBIG	49.78	58.44	50.01	73.67	50.19	72.10	87.17	117.88	74.08	105.73	100.74	135.61	66.61	103.32	103.87	167.99
Ours	**44.71**	**53.35**	**48.07**	73.49	**44.51**	**68.12**	86.89	121.45	**67.60**	102.34	109.71	147.09	**64.05**	103.81	106.49	167.75

Motion	Purchases		Sitting		Sitting Down		Taking Photo		Waiting		Walking Dog		Walking Together		Average	
millisecond	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K	560	1K
DMGNN	104.86	146.09	75.51	115.44	118.04	174.05	78.38	123.65	85.54	113.68	183.20	210.17	70.46	86.93	93.57	127.62
MSR-GCN	101.63	139.15	78.19	120.02	102.83	155.45	77.94	121.87	76.33	106.25	111.87	148.21	52.93	65.91	81.13	114.18
STSGCN	**87.10**	**119.26**	82.32	119.83	**92.60**	**129.67**	87.70	119.79	86.41	118.04	**86.79**	**118.33**	75.33	95.83	80.66	113.33
SPGSN	96.53	133.88	75.00	116.24	98.94	149.88	75.58	118.22	73.50	**103.62**	102.37	137.96	49.84	60.86	77.40	**109.64**
PGBIG	96.95	133.56	75.16	114.18	95.21	139.71	**72.44**	**115.96**	72.13	103.57	104.44	144.20	53.77	65.03	76.84	110.06
Ours	94.51	138.03	**70.49**	**113.87**	94.54	149.20	73.30	119.26	72.80	105.63	103.41	139.83	**49.40**	**60.23**	**75.36**	110.96

Table 3: Comparison of short-term prediction on 8 action categories from the CMU-Mocap dataset. Results at 80ms, 160ms, 320ms, and 400ms in the future are shown.

Motion	basketball				basketball signal				directing traffic				jumping			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	15.45	26.88	43.51	49.23	20.17	32.98	42.75	44.65	20.52	40.58	75.38	90.36	26.85	48.07	93.50	108.90
Traj-GCN	11.68	21.26	40.99	50.78	3.33	6.25	13.58	17.98	6.92	13.69	30.30	39.97	17.18	32.37	60.12	72.55
DMGNN	15.57	28.72	59.01	73.05	5.03	9.28	20.21	26.23	10.21	20.90	41.55	52.28	31.97	54.32	96.66	119.92
MSR-GCN	10.28	18.94	37.68	47.03	3.03	5.68	12.35	16.26	5.92	12.09	28.36	38.04	14.99	28.66	55.86	69.05
STSGCN	12.56	23.04	41.92	50.33	4.72	6.69	14.53	17.88	6.41	12.38	29.05	38.86	17.52	31.48	58.74	72.06
SPGSN	10.24	18.54	38.22	48.68	2.91	5.25	11.31	15.01	5.52	11.16	25.48	37.06	14.93	28.16	56.72	71.16
PGBIG	10.14	18.53	36.88	45.97	2.83	4.92	10.58	14.23	5.17	10.01	23.46	31.75	14.56	28.59	56.98	70.21
Ours	**6.57**	**13.83**	**33.43**	**44.93**	**1.60**	**3.40**	**8.54**	**11.99**	**3.09**	**7.31**	**20.76**	**29.73**	**9.28**	**20.99**	**50.29**	**65.52**

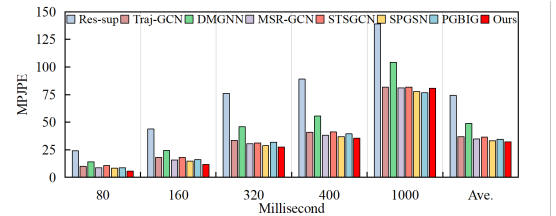| Motion	running				soccer				walking				washing window			
millisecond	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	25.76	48.91	88.19	100.80	17.75	31.30	52.55	61.40	44.35	76.66	126.83	151.43	22.84	44.71	86.78	104.68
Traj-GCN	14.53	24.20	37.44	41.10	13.33	24.00	43.77	53.20	6.62	10.74	17.40	20.35	5.96	11.62	24.77	31.63
DMGNN	17.42	26.82	38.27	40.08	14.86	25.29	52.21	65.42	9.57	15.53	26.03	30.37	7.93	14.68	33.34	44.24
MSR-GCN	12.84	20.42	30.58	34.42	10.92	19.50	37.05	46.38	6.31	10.30	17.64	21.12	5.49	11.07	25.05	32.51
STSGCN	16.70	27.58	36.15	36.42	13.49	25.24	39.87	51.58	7.18	10.99	17.84	22.61	6.79	12.10	24.92	36.66
SPGSN	10.75	**16.67**	26.07	30.08	10.86	18.99	35.05	45.16	6.32	10.21	**16.34**	**20.19**	4.86	9.44	21.50	28.37
PGBIG	13.18	24.72	44.17	50.17	11.89	21.96	42.00	51.50	6.83	11.59	19.49	22.78	5.00	9.58	21.86	28.57
Ours	**8.97**	18.70	37.55	43.24	**7.53**	**15.39**	**33.93**	**43.78**	**4.75**	**8.59**	16.51	20.37	**2.92**	**6.80**	**18.40**	**25.50**

Table 4: The average prediction MPJPEs across the test set of 3DPW at various prediction time steps.

millisecond	100	200	400	500	600	800	900	1000	Average
DMGNN	17.80	37.11	70.38	83.02	94.12	109.67	117.25	123.93	81.66
HisRep	15.88	35.14	66.82	78.49	93.55	107.63	114.59	114.75	78.36
MSR-GCN	15.70	33.48	65.02	77.59	93.81	108.15	114.88	116.31	78.12
STSGCN	18.32	37.79	67.51	77.34	92.75	106.65	113.14	112.22	78.22
SPGSN	15.39	32.91	**64.54**	**76.23**	91.62	103.98	109.41	111.05	75.64
Ours	**10.52**	**27.59**	65.10	78.78	**88.63**	**101.43**	**105.86**	**109.78**	**73.46**

Figure 5: Comparison of different motion forecasting models in CMU-Mocap.

5.2 Comparison to State-of-the-Art Methods.

We validate the effectiveness of our method by presenting quantitative results for short-term and long-term motion prediction on the H3.6M, CMU-Mocap, and 3DPW datasets.

Short-term motion prediction. Short-term motion prediction aims to predict future poses within 500 milliseconds. Table 1 presents the MPJPEs of our method and several previous methods on 15 representative actions at multiple prediction timestamps for H3.6M. Our method achieves the best performance at all timestamps among all representative actions. Compared to the baselines, our method shows a significant improvement.Table 3 presents the MPJPEs of our method and several previous methods on 8 action

categories at multiple prediction timestamps. Our method outperforms most methods at all timestamps. The average results of the CMU-Mocap dataset are presented in Figure 5, which shows that our method achieves almost optimal results at almost every timestamp, with varying degrees of MPJPEs reduction compared to other baselines. These results prove the effectiveness of our proposed model in achieving significant improvements in short-term prediction.

Long-term motion prediction. Long-term motion prediction aims to predict the poses over 500 milliseconds, which is a challenging task due to the variability of poses and the difficulty in capturing human intention. In Table 2, we present the prediction result of various methods at 560 ms and 1000 ms on 15 actions in

Table 5: Results of our methods in the ablation study, where MPJPE is shown and lower values is better.

Increment	$\mathcal{L}_t$	$\mathcal{L}_s$	$\mathcal{L}_{kl}$	80ms	160ms	320ms	400ms	Average
	✓			10.57	23.65	50.16	61.91	36.57
		✓		9.99	22.51	48.10	59.57	35.04
	✓	✓		9.99	22.50	47.87	59.18	34.88
	✓	✓	✓	9.53	21.66	46.83	58.20	34.06
✓	✓			5.29	16.61	43.17	55.91	30.24
✓		✓		4.91	15.94	42.87	54.98	29.68
✓	✓	✓		4.55	15.95	42.62	55.18	29.58
✓	✓	✓	✓	**4.31**	**15.53**	**41.82**	**54.21**	**28.96**

Table 6: Impact of the fusion coefficient λ on motion prediction results. MPJPE is reported. Mean and variance values are reported, excluding the case where $\lambda = 0$.

λ	80ms	160ms	320ms	400ms	Average
0	4.55	15.95	42.62	55.18	29.58
0.001	4.34	15.61	41.86	54.29	29.02
0.01	4.35	15.59	41.91	54.35	29.05
0.1	**4.31**	**15.53**	**41.82**	**54.21**	**28.96**
1.0	4.41	15.68	42.08	54.55	29.18
Average	4.35	15.60	41.92	54.36	29.05
Variance	0.03	0.05	0.10	0.13	0.09
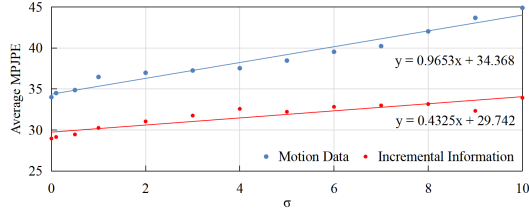
Figure 6: Results under different levels of Gaussian noise (with mean of 0 and standard deviation of σ). "Motion Data" indicates the model trained using raw motion data, and "Incremental Information" indicates the model trained using the incremental information of motion. The straight lines represent the results of robust linear regression [15].

H3.6M. We can see that our proposed method achieves more effective prediction on most actions and has lower MPJPEs at 560 ms and achieve a comparable prediction result at 1000 ms. To further validate the effectiveness of our method, we conducted experiments on the 3DPW dataset. The average MPJPEs across all the test samples at different prediction steps are presented in Table 4. Our method outperforms state-of-the-art methods, reducing the MPJPE by an average of 4.97%. However, we also found that our approach did not show a significant improvement in long-term prediction performance. There are likely two reasons for this state of affairs: 1) long-term action prediction tasks themselves are highly variable and unpredictable, and 2) our method may have lost some semantic information about the motion by using incremental information to represent the motion characteristics. Overall, our method demonstrates promising results in both short-term and long-term motion prediction, which confirms that our model is effective in utilizing incremental information and achieving impressive outcomes.

5.3 Ablation study

The influence of spatio-temporal branching framework. To gain a better understanding of our approach, we conducted additional experiments on the H3.6M dataset to assess the impact of incremental information, temporal and spatial domain modeling,

and cross-domain knowledge distillation. The results are summarized in Table 5, where L_t and L_s represents scenarios where the temporal branch or the spatial branch is utilized. Our experiments show that the use of incremental information can lead to a significant improvement in prediction accuracy, indicating that it can provide more expressive information for characterizing motion. Moreover, our experiments show that different domain modeling methods have varying capabilities in modeling motion data. In addition, our experiments reveal that the best results are achieved when cross-domain knowledge distillation is used, indicating that this approach can result in better integration of spatial-domain and temporal-domain information. Overall, These results highlight the importance of considering multiple modeling approaches and using knowledge distillation in motion prediction tasks.

The influence of hyper-paramter λ. We varied the value of λ in our approach to examine its impact on the H3.6M dataset. Specifically, we varied the value of λ from $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The results presented in Table 6 demonstrates that our method achieved the lowest MPJPEs when $\lambda = 0.1$, indicating that knowledge distillation is effective. Additionally, we observe that in the absence of the knowledge distillation module ($\lambda = 0$), the MPJPEs are the highest. Besides, there was little difference in performance when λ took other values, suggesting that knowledge distillation is effective and our method is not sensitive to the choice of λ

The influence of incremental information. To verify the effectiveness of the incremental information in reducing the impact of noise on model learning, we conducted an experiment where we added Gaussian noise to the data at various levels and measured the resulting change in MPJPEs. The results, shown in Figure 6, indicate that our method has a smaller slope, indicating that the amplitude of the change in MPJPEs is smaller as the noise level increases. This demonstrates that our approach can effectively mitigate the interference of noise on model learning.

6 CONCLUSION

In this paper, we propose a novel spatio-temporal branching framework for human motion prediction utilizing the incremental information from the motion data to effectively reduce noise interference in the observation data. Different from previous works, our method separately extracts the temporal-domain and spatial-domain features to capture more useful information from the motion data. The interplay of spatial joints and temporal patterns is also enhanced, and modal bias is reduced through cross-domain knowledge distillation. Extensive experiments conducted on three benchmarks demonstrate the effectiveness of our method. In our future work, we intend to investigate the potential of our approach in other applications such as action recognition and pose estimation.

REFERENCES

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

[2] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. 2017. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6158–6166.

[3] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. 2020. Learning progressive joint propagation for human motion prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 226–242.

[4] Siheng Chen, Baoan Liu, Chen Feng, Carlos Vallespi-Gonzalez, and Carl Wellington. 2020. 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. *IEEE Signal Processing Magazine* 38, 1 (2020), 68–86.

[5] François Chollet. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1251–1258.

[6] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. 2020. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6992–7001.

[7] Qiongjie Cui, Huaijiang Sun, Yue Kong, Xiaoqian Zhang, and Yanmeng Li. 2021. Efficient human motion prediction using temporal convolutional generative adversarial network. *Information Sciences* 545 (2021), 427–447.

[8] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. 2021. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11467–11476.

[9] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*. 4346–4354.

[10] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*. PMLR, 1243–1252.

[11] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. 2019. A neural temporal model for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12116–12125.

[12] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. 2018. Adversarial geometry-aware human motion prediction. In *Proceedings of the european conference on computer vision (ECCV)*. 786–803.

[13] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José MF Moura. 2018. Few-shot human motion prediction via meta-learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 432–450.

[14] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. 2019. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7134–7143.

[15] Peter J Huber. 2011. Robust statistics. In *International encyclopedia of statistical science*. Springer, 1248–1251.

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. 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 (2013), 1325–1339.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Hema Swetha Koppula and Ashutosh Saxena. 2013. Anticipating human activities for reactive robotic response.. In *IROS*. Tokyo, 2071.

[19] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. 2019. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 8553–8560.

[20] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. 2021. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3054–3063.

[21] Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. 2014. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1314–1321.

[22] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. 2018. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5226–5234.

[23] Maosen Li, Siheng Chen, Zihui Liu, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. 2021. Skeleton graph scattering networks for 3d skeleton-based human motion prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. 854–864.

[24] Maosen Li, Siheng Chen, Zijing Zhang, Lingxi Xie, Qi Tian, and Ya Zhang. 2022. Skeleton-Parted Graph Scattering Networks for 3D Human Motion Prediction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*. Springer, 18–36.

[25] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 214–223.

[26] Hongyi Liu and Lihui Wang. 2017. Human motion prediction for human-robot collaboration. *Journal of Manufacturing Systems* 44 (2017), 287–294.

[27] Zhenguang Liu, Shuang Wu, Shuyuan Jin, Qi Liu, Shijian Lu, Roger Zimmermann, and Li Cheng. 2019. Towards natural and accurate future motion prediction of humans and animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10004–10012.

[28] Wenjie Luo, Bin Yang, and Raquel Urtasun. 2018. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 3569–3577.

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

[30] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2020. History repeats itself: Human motion prediction via motion attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 474–489.

[31] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. 2019. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9489–9497.

[32] Julieta Martinez, Michael J Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2891–2900.

[33] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 911–918.

[34] Vladimir Pavlovic, James M Rehg, and John MacCormick. 2000. Learning switching linear models of human motion. *Advances in neural information processing systems* 13 (2000).

[35] Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the seventeenth conference on uncertainty in artificial intelligence*. Morgan Kaufmann, San Francisco, 411–420.

[36] Wenwen Qiang, Jiangmeng Li, Bing Su, Jianlong Fu, Hui Xiong, and Ji-Rong Wen. 2023. Meta attention-generation network for cross-granularity few-shot learning. *International Journal of Computer Vision* 131, 5 (2023), 1211–1233.

[37] Wenwen Qiang, Jiangmeng Li, Changwen Zheng, and Bing Su. 2021. Auxiliary task guided mean and covariance alignment network for adversarial domain adaptation. *Knowledge-Based Systems* 223 (2021), 107066.

[38] Wenwen Qiang, Jiangmeng Li, Changwen Zheng, Bing Su, and Hui Xiong. 2021. Robust local preserving and global aligning network for adversarial domain adaptation. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[39] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. 2021. Space-time-separable graph convolutional network for pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11209–11218.

[40] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Trans. Graph.* 38, 6 (2019), 209–1.

[41] Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. 2006. Modeling human motion using binary latent variables. *Advances in neural information processing systems* 19 (2006).

[42] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. 2018. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters* 3, 3 (2018), 2394–2401.

[43] Herwin Van Welbergen, Ben JH Van Basten, Arjan Egges, Zs M Ruttkay, and Mark H Overmars. 2010. Real time animation of virtual humans: a trade-off between naturalness and control. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2530–2554.

[44] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In *European Conference on Computer Vision (ECCV)*.

[45] Jack Wang, Aaron Hertzmann, and David J Fleet. 2005. Gaussian process dynamical models. *Advances in neural information processing systems* 18 (2005).

[46] Ye Yuan and Kris Kitani. 2019. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10082–10092.

[47] Ru Zhang, Feng Zhu, Jianyi Liu, and Gongshen Liu. 2019. Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. *IEEE Transactions on Information Forensics and Security* 15 (2019), 1138–1150.

[48] Chongyang Zhong, Lei Hu, Zihao Zhang, Yongjing Ye, and Shi hong Xia. 2022. Spatio-temporal gating-adjacency GCN for human motion prediction. 2022 IEEE. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6437–6446.
[49] Yujie Zhou, Haodong Duan, Anyi Rao, Bing Su, and Jiaqi Wang. 2023. Self-supervised Action Representation Learning from Partial Spatio-Temporal Skeleton Sequences. *arXiv preprint arXiv:2302.09018* (2023).

A APPENDIX

We report more experimental results and more technical details which are not included in the paper due to space limit.

A.1 Implementation details.

Our model is implemented in PyTorch and trained on a single NVIDIA A40 GPU using the ADAM optimizer [17]. We set the learning rate to 0.001 with a decay rate of 0.96 every two epochs. We used a batch size of 32 for training and applied gradient clipping with a maximum ℓ2-norm of 1 to prevent exploding gradients.

A.2 Effects of KL divergence in Cross-domain Knowledge Distillation

To study the effects of KL divergence in facilitating knowledge distillation between the spatial and temporal domains, we evaluate different approaches, including using Jensen-Shannon (JS) divergence and Euclidean distance instead of KL divergence on the H3.6M dataset. The experimental results presented in the Table 7 demonstrate the effectiveness of all these methods in promoting knowledge transfer. Among these alternatives, KL divergence achieved the best performance. This can be attributed to the unique characteristics of KL divergence, which quantifies the divergence between motion energy distributions in our method. In our study, KL divergence effectively captures the distinctions and similarities

between the spatial and temporal features, enabling a more efficient transfer of knowledge between the two domains and reducing modal bias in the learned representation.

Table 7: Impact of different knowledge distillation methods on motion prediction results. MPJPE is reported.

Method	80ms	160ms	320ms	400ms	560ms	1000ms	Average
Temporal branch (w/o KL)	5.29	16.61	43.17	55.91	78.47	113.26	52.12
Spatial branch (w/o KL)	4.91	15.94	42.87	54.98	76.85	112.34	51.32
Ours (Euclidean distance)	4.43	15.81	42.31	54.76	76.24	111.68	50.87
Ours (JS)	4.37	15.63	42.03	54.51	75.75	111.37	50.61
Ours (KL)	**4.31**	**15.53**	**41.82**	**54.21**	**75.36**	**110.96**	**50.37**

A.3 Time and Model Size Comparisons.

In Table 8, we compare our method to existing methods in terms of prediction time and model size and present the prediction results for short-term motion prediction on the H3.6M dataset. Our method achieves the lowest MPJPE while maintaining efficient running time. Our method's main advantage is its simple network architecture, without any complex module design or arithmetic mechanism. Additionally, we use a dual-branch network that can run in parallel, further improving efficiency.

Table 8: Time and model size comparisons. The average MPJPE at 80ms, 160ms, 320ms, 400ms, 560ms, and 1000ms is also reported.

Method	Train(Per batch)	Test(Per batch)	Model Size	MPJPE
Traj-GCN	51ms	41ms	2.55M	58.12
MSR-GCN	107ms	52ms	6.30M	57.93
STSGCN	68ms	39ms	0.04M	57.69
PGBIG	117ms	39ms	1.23M	54.51
SPGSN	461ms	376ms	7.60M	54.19
Ours	118ms	62ms	7.33M	**50.37**