Skeleton-Prompt: A Cross-Dataset Transfer Learning Approach for Skeleton Action Recognition

Mingqi Lu[a,b], Xiaobo Lu[a,b,*], Jun Liu[c]

[a]*School of Automation, Southeast University, Nanjing 210096, China.*
[b]*Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China.*
[c]*School of Computing and Communications, Lancaster University, UK*

Abstract

This paper presents Skeleton-Prompt, a novel tuning method designed to tackle cross-dataset transfer issues in skeleton action recognition models. Given the scarcity of large-scale 3D skeleton datasets and the variability in keypoint structures across datasets, existing methods often rely on training models from scratch, necessitating extensive labeled data and exhibiting high sensitivity to occlusion. Our approach aims to fine-tune pre-trained models to adapt to limited real-world skeleton data. We use 2D skeletons as inputs and leverage a large human motion dataset for 2D to 3D pose estimation to learn generalizable motion features. A lightweight prompt generator produces instance-level prompts, and we employ dynamic queries with cross-attention to refine the semantic information of the input data. Additionally, we introduce a joint-enhanced multi-stream fusion mechanism based on self-attention to improve robustness against incomplete skeletons. Skeleton-Prompt represents a significant advancement in efficient fine-tuning for skeleton action recognition, effectively addressing cross-dataset generalization challenges in a data-efficient and parameter-efficient manner.

Keywords: Action recognition, Prompt Tuning, Transfer Learning, Occluded Skeletons.

*Corresponding author
Email addresses: lumingqiseu@gmail.com (Mingqi Lu), xblu2013@126.com (Xiaobo Lu), j.liu81@lancaster.ac.uk (Jun Liu)

1. Introduction

Owing to its ability to model human joints and topological structures, skeleton sequence representation learning demonstrates significant advantages in action recognition. Over the past decade, deep learning-based methods for skeleton action recognition have evolved into various networks, such as CNNs and RNNs [1, 2]. With the introduction of graph convolutional networks (GCNs), methods like ST-GCN [3] utilize graph structures to model the spatiotemporal relationships of skeletons. However, graph-based methods are constrained by prior knowledge of input nodes and/or edges, making it difficult to handle unknown node types. The number of key joints varies across different skeleton datasets, and mainstream GCN-based methods fail to overcome the limitations posed by differing joint connection rules, resulting in challenges for scalability and transferability.

Transformers have shown great potential for processing sequential data, leading to the development of numerous Transformer-based methods for modeling the spatiotemporal information of skeleton sequences [4, 5]. The methods mentioned above are primarily based on ideal 3D skeletons from indoor simulated datasets, which are limited by depth sensors like Microsoft Kinect and are not widely applicable in real-world scenarios. Most mainstream methods rely on large amounts of labeled data, and the field of skeleton action recognition currently lacks large-scale 3D skeleton datasets. Consequently, it is challenging to train large pre-trained models akin to those used for ImageNet or BERT Each model requires training parameters from scratch for different skeleton datasets, leading to poor generalization.

Currently, there is little research on transferring pre-trained knowledge among skeleton action recognition models across different datasets. To address this gap, our work aims to fine-tune pre-trained models for transferability. We focus on exploring how far skeleton action recognition can advance in the direction of cross-dataset transfer learning without large-scale 3D skeleton data. Thanks to the rapidly developing pose estimation algorithms, 2D skeletons can be easily obtained from RGB videos, proving to be more accurate and effective for action recognition [6, 7]. Inspired by MotionBERT [8], we leverage large-scale 3D human motion data [9] to learn generalizable

skeleton motion features by recovering 3D motions from 2D skeletons. We propose SkeleFormer, based on PoseFormer [10], as a skeleton encoder that comprehensively captures the spatiotemporal relationships between skeleton joints and uses 2D-to-3D pose estimation as a pretraining task.

Traditional fine-tuning methods in transfer learning add task-specific classification heads and adjust all parameters; however, full fine-tuning on small datasets can compromise the quality of pre-trained parameters. Visual Prompt Tuning (VPT) [11] introduces a small number of task-specific visual prompts into the input space while freezing the entire pre-trained backbone. By requiring updates to only a few parameters, VPT not only significantly reduces computational and storage costs but also prevents overfitting and feature distortion. Inspired by this, we propose a novel skeleton prompt generator that allows pre-trained skeleton action recognition models to generalize across different datasets. The skeleton prompt generator dynamically generates prompts based on each input instance, effectively extracting knowledge from skeleton samples across different datasets, thereby facilitating knowledge transfer from the pre-trained model to adapt to the target data. Furthermore, we introduce a cross-attention mechanism into the skeleton prompts. Unlike the direct concatenation used in VPT, we compute the cross-attention between the pose embeddings and the generated skeleton prompts, adding the result as a residual to the pose embeddings. This cross-attention allows the prompts and pose embeddings to mutually focus on each other, enhancing the semantic information within the skeleton prompts. These prompts enable flexible adaptation across different datasets, showcasing immense potential for cross-dataset transfer, particularly for target datasets with limited large-scale data. It's worth noting that Skeleton-Prompt is the first work to apply efficient fine-tuning to skeleton action recognition.

In the real world, occlusion of skeleton data is both unavoidable and widespread. Even a few occluded points can severely impact the sparse representation of human skeletons. In this paper, we consider not only joint stream but also bone stream and velocity stream, and propose Joint-Enhanced Multi-Stream Fusion (JEMF) to learn discriminative embeddings through multi-stream fusion. In JEMF, joint embeddings serve as the query set, while bone and velocity embeddings encode the key-value set.

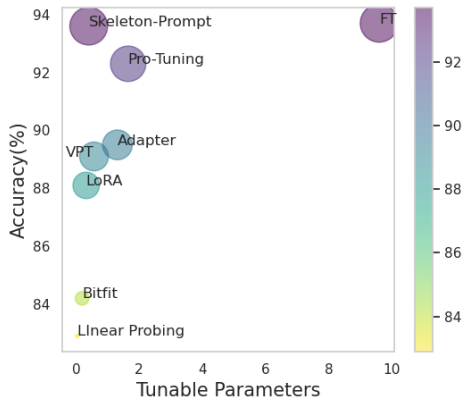


Figure 1: Comparison of transfer performance and trade-off of trainable parameters for different tuning methods under the NTU-60 CS protocol (higher is better). Using the SkeleFormer pre-trained on the Posetics dataset, Skeleton-Prompt achieves exceptional cross-dataset transfer performance and parameter efficiency trade-offs (best viewed in color).

Through the self-attention mechanism, complementary information from the bone and velocity streams is transferred to the joint stream, enhancing the model's robustness against occluded skeletons.

Our contributions can be summarized as follows:

(1) We introduce SkeleFormer, leveraging 2D-to-3D pose estimation for pretraining to capture spatiotemporal skeleton relationships across datasets.

(2) We propose a skeleton prompt generator with cross-attention to dynamically adapt pre-trained models across datasets, enhancing knowledge transfer with minimal parameter updates.

(3) We develop Joint-Enhanced Multi-Stream Fusion to improve robustness against occluded skeleton data by fusing joint, bone, and velocity streams through self-attention.

(4) We demonstrate effective cross-dataset transfer learning using 2D skeletons from RGB videos, addressing the scarcity of large-scale 3D skeleton datasets.

2. Related Work

2.1. Skeleton Action Recognition

In recent years, skeleton action recognition has witnessed significant progress. In this field, Nguyen et al. [12] proposed the Double-Feature Double-Motion Network, Peng et al. [13] introduced a scenario navigation framework for open-set recognition, and Yan et al. [14] addressed the challenge of one-shot 3D action recognition using large language models. To learn discriminative representations of the skeleton, GCNs explore the structural information of the skeleton and the interdependencies between joints in spatiotemporal graphs, such as ST-GCN [3]. However, GCN-based methods are constrained by the requirement for prior knowledge of input nodes and/or edges, making it difficult to handle unknown types of nodes. Recently, several studies have applied transformers to skeleton-based action recognition. ST-TR [4] computes self-attention scores for each pair of key joints in the spatiotemporal graph. RSA-Net [5] uses a relation-mining self-attention network to capture intra - and inter-frame action features. PoseC3D [15] stacks heatmaps along the temporal dimension and uses a 3D-CNN to process 2D skeletons. Regarding multi-stream fusion, Peng et al. [16] employed attention mechanisms to integrate skeletal, joint, and bone branches, while Xu et al. [17] adopted a mixture-of-experts strategy for fusion. The methods mentioned above all train model parameters from scratch in a fully supervised manner and rely on a large amount of labeled skeletal data. We consider a more realistic setup that utilizes pre-trained models to achieve action recognition across multiple skeletal datasets, making us the first to address this issue.

Most skeleton-based action recognition methods assume ideal skeleton features; however, occlusions of the skeleton are unavoidable in real-world scenarios. For incomplete skeletons, Song et al. [18] proposed a richly activated GCN to improve the robustness of action recognition models against occluded skeletons. Research in this area is still in its infancy, and performance improvements are needed.

2.2. Parameter-Efficient Transfer Learning

With the development of Large Language Models (LLMs) in Natural Language Processing (NLP), parameter-efficient transfer learning for pre-trained models has grad-

ually emerged. Full fine-tuning may distort pre-trained features, compromising robustness to distribution shifts; however, parameter-efficient transfer learning effectively mitigates this issue. Bitfit [19] freezes the weights and only adjusts the biases or a subset of biases in the pre-trained language model (PLM). Adapters [20] freeze all parameters of the PLM and sequentially insert trainable layers with residual connections after the feed-forward network (FFN) or attention blocks. LoRA [21] employs additional low-rank modules as trainable parameters to optimize the weights of the self-attention layers.

2.3. Visual Prompts

Prompt tuning has also flourished in the field of computer vision, facilitating the transfer of pre-trained visual models to downstream tasks. Jia et al. [11] introduced prompts into the ImageNet pre-trained ViT as part of the input tokens. Pro-Tuning [22] employs lightweight convolutional blocks to generate visual prompts for specific image tasks. Currently, no research has investigated the effectiveness of parameter-efficient learning in the field of skeleton action recognition. Our work is the first to explore skeleton prompts, closely related to VPT and Pro-Tuning, and aims to address the cross-dataset generalization problem of skeleton action recognition models.

3. Methodology

The proposed Skeleton-Prompt framework is shown in Figure 2. The skeleton prompt generator captures sample-specific prompts, and during cross-dataset transfer, only a small number of parameters for the skeleton prompt generator and classifier need to be optimized. The following sections introduce the skeleton encoding and skeleton prompt generation.

3.1. Revisiting PoseFormer

PoseFormer treats each 2D pose as input tokens and employs a multi-layer transformer structure similar to ViT to model the relationships between body joints within each frame and the temporal correlations across frames. For the skeletal sequence $X \in R^{f\times(J\cdot 2)}$, f denotes the number of frames in the input sequence, J represents the

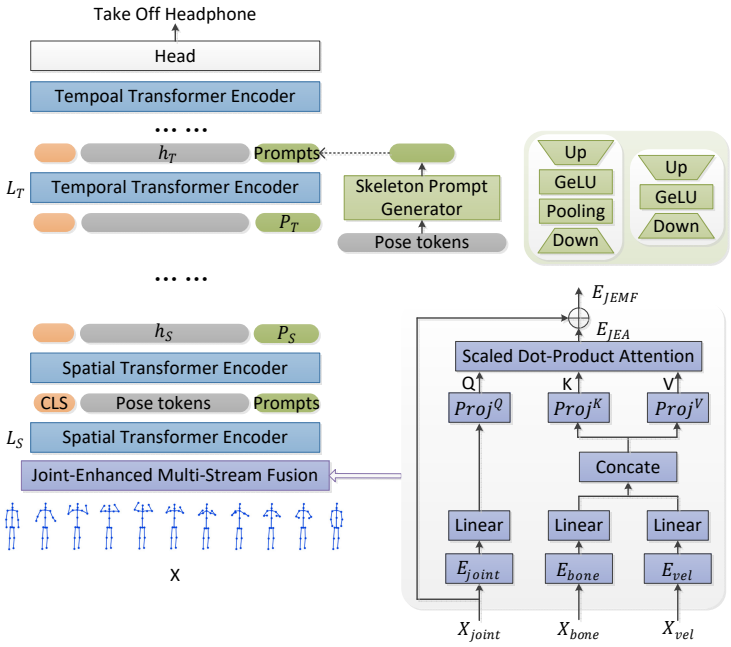


Figure 2: Tuning architecture of the proposed skeleton action recognition model. SkeletonPrompt utilizes a prompt generator for instance-level dynamic updates to achieve cross-dataset transfer, with the blue blocks representing the frozen parts of the pre-trained transformer (best viewed in color).

number of joints in each 2D pose, and 2 indicates the coordinates of the joints in the 2D space. First, each 2D pose is mapped to $h_s = [S_1, S_2, \cdots, S_J]$ using the feature embedding $E \in R^{(J \cdot 2) \times c}$, where $h_s \in R^{J \times c}$. This representation is then input into the spatial transformer L_S to encode the local relationships between the 2D joints in each frame.

$$\left(S_1^i, S_2^i, \cdots, S_J^i\right) = L_S^i\left(S_1^{i-1}, S_2^{i-1}, \cdots, S_J^{i-1}\right) \tag{1}$$

Next, the single-frame embeddings h_s from f frames are concatenated to form $h_T = \left[T_1, T_2, \cdots, T_f\right]$, where $h_T \in R^{f \times (J \cdot c)}$. This concatenated representation is then input into the temporal transformer L_T to model dependencies across the frame sequences.

$$\left(T_1^i, T_2^i, \cdots, T_f^i\right) = L_T^i\left(T_1^{i-1}, T_2^{i-1}, \cdots, T_f^{i-1}\right) \tag{2}$$

3.2. Joint-Enhanced Multi-Stream Fusion

Considering the unavoidable issue of skeleton occlusion in the real world, we propose Joint-Enhanced Multi-Stream Fusion (JEMF) to improve the robustness of the model against incomplete skeletons. The structure of JEMF is illustrated in Figure 2.

Unlike PoseFormer, we consider three input sequences in SkeleFormer: joint X_{joint}, bone X_{bone}, and velocity X_{vel}. These sequences are used to extract high-dimensional embeddings E_{joint}, E_{bone}, and X_{vel} from the three streams, which are then fed into the joint-enhanced multi-stream fusion (JEMF) module for integration, thereby leveraging complementary crossmodal dependencies. The design of JEMF is based on the self-attention mechanism, where the joint stream serves as the query set and the bone-velocity stream acts as the key-value set. We encode E_{joint} using a linear projection layer Proj_j^Q, resulting in $Q_j = \text{Proj}_j^Q\left(E_{\text{joint}}\right)$. For the keys and values of the bone and velocity streams, E_{bone} and E_{vel} are aggregated through concatenation, and each is encoded using the projection layers Proj_{bv}^K and Proj_{bv}^V, generating K_{bv} and V_{bv}, respectively.

$$K_{bv} = \text{Proj}_{bv}^K\left(\text{Concat}\left(E_{bone}, E_{vel}\right)\right) \tag{3}$$

$$V_{bv} = \text{Proj}_{bv}^V\left(\text{Concat}\left(E_{bone}, E_{vel}\right)\right) \tag{4}$$

The self-attention output provides joint-enhanced attention (JEA):

$$E_{JEA} = \text{Selfattention}\left(Q_j, K_{bv}, V_{bv}\right) \tag{5}$$

The expected Joint-enhanced Multi-stream Fusion (JEMF) is formulated as follows:

$$E_{JEMF} = E_{\text{joint}} + E_{JEA} \tag{6}$$

This transfers the key information from the bone and velocity streams to the joint stream, with the resulting embeddings fed into the SkeleFormer.

$$h = E_{JEMF} + MLP\left(LN\left(E_{JEMF}\right)\right) \tag{7}$$

Where Proj refers to the projection layer based on fully connected (fc) layers, and LN stands for layer normalization.

3.3. 2D->3D Pre-training

In SkeleFormer, both the spatial transformer and temporal transformer consist of four layers, with an embedding dimension of $17 \times 32 = 544$. We conduct a 2D-to-3D pre-training task on the Human 3.6 M dataset, which is the most widely used indoor dataset for 3D single-person Human Pose Estimation (HPE). This dataset includes 11 professional actors performing 17 actions, comprising 3.6 million frames of video annotated with 3D ground truth. Following the approach of MotionBERT, we use orthogonal projection on the 3D motions X to obtain undisturbed 2D skeletons x, where we randomly mask 15% of the joints (setting them to zero) and add noise. We utilize SkeleFormer to extract motion features and reconstruct the 3D motions $\widehat{X}$, calculating the loss function between $\widehat{X}$ and the true X.

$$L_{2D\to 3D} = \sum_{j=1}^{J} \left\| \widehat{X}_j - X_j \right\|_2 \tag{8}$$

The feature embeddings learned by SkeleFormer can serve as spatiotemporal representations of human motion. For skeleton-based action recognition tasks, we directly apply global average pooling across different individuals and time steps. The resulting embeddings are then input into a multilayer perceptron (MLP) with a single hidden layer, and the entire network is trained end-to-end using cross-entropy classification loss.

3.4. Skeleton Prompt Generator

Our Skeleton-Prompt generates instance-level prompts for each input based on the embedded context to adapt to variations in data distribution. As shown in Figure 2, we adopt a lightweight bottleneck architecture as the skeleton prompt generator, which consists of two perceptron layers: a down-sampling projection $W_1 \in R^{m\times d}$ and an up-sampling projection $W_2 \in R^{N\times d\times m}$, where d is the input embedding dimension, m is the hidden layer dimension, and N is the length of the generated skeleton prompt. The parameter overhead introduced by the single-layer skeleton prompt generator is minimal. Therefore, for each layer of SkeleFormer, we assign a layer-specific skeleton prompt generator. In the spatial transformer, the skeleton prompt P_S generated at each

layer is as follows:

$$P_S = W_{S_2}\left(\mathrm{ReLU}\left(W_{S_1}h_S + b_{S_1}\right)\right) + b_{S_2} \tag{9}$$

Where $W_{S_1} \in R^{m\times(J\cdot c)}$ and $W_{S_2} \in R^{N\times(\cdot c)\times m}$.

In the temporal transformer, a max pooling layer is introduced within the bottleneck structure to pool the embeddings of f frames into a prompt of length N, thereby reducing the number of parameters.

$$P_T = W_{T_2}\left(\mathrm{ReLU}\left(\mathrm{MaxPool}\left(W_{T_1}h_T + b_{T_1}\right)\right)\right) + b_{T_2} \tag{10}$$

Where $W_{T_1} \in R^{m\times(\cdot c)}$ and $W_{T_2} \in R^{N\times(J\cdot c)\times m}$

The typical approach is to directly concatenate the skeleton prompt P with the embedding h, using it as the new input for each transformer layer. We introduce cross-attention to link the two, allowing the skeleton prompt to adapt to crossdataset tasks, as shown in Fig. 4. For each layer, the input embedding h passes through the self-attention layer to obtain the pose embedding $E_{\text{pose}} = \mathrm{SelfAttention}(h)$. Cross-attention is then computed between the pose embedding and the generated skeleton prompt P. The pose embedding acts as the query set, represented as $Q^C = PW_Q$, while the skeleton prompt serves as the key-value set, with $K^C = V^C = PW_K$. The cross-attention can be formulated as:

$$\begin{aligned}&\mathrm{CrossAttention}\left(E_{\text{pose}}, P\right)\\ &= \mathrm{Softmax}\left(\frac{E_{\text{pose}}W_Q \cdot PW_K}{\sqrt{d}}\right)\cdot PW_K\end{aligned} \tag{11}$$

where d represents the dimension of both the pose embedding and prompts, ensuring consistency in dimensions throughout the computation.

The cross-attention values between the pose embedding and skeleton prompts capture the semantic relationship between the two. These values are added to the pose embedding as a residual, and after passing through a Layer Norm and MLP layer, they are fed into the next transformer layer. Additionally, we introduce a weight-sharing mechanism using the parameter values from the self-attention in SkeleFormer to initialize the crossattention weights, thereby avoiding the need for a large number of learnable parameters.

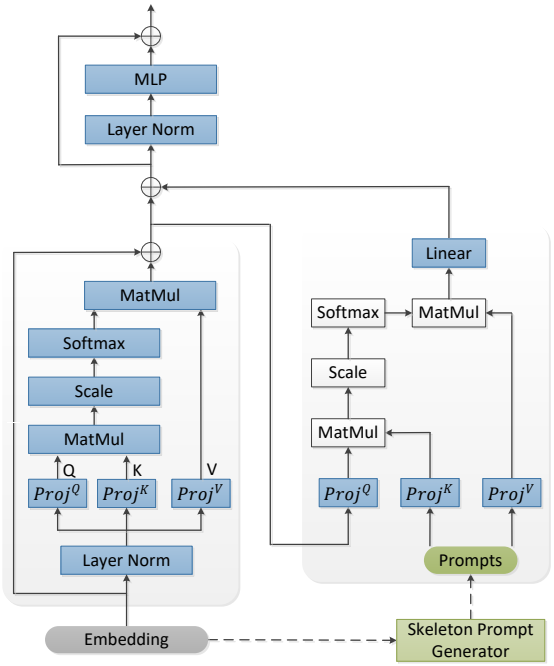


Figure 3: Structure diagram of Skeleton-Prompt, which consists of two key components: skeleton prompt generation and cross-attention calculation, with the blue blocks indicating the frozen sections of the pre-trained transformer (best viewed in color).

During cross-dataset transfer training, only the parameters of the skeleton prompt generator and the classification head are updated, while the entire transformer backbone remains frozen. Each layer's skeleton prompt generator has two input options: the hidden state of the current layer or the output from the previous layer. Our experiments have demonstrated that there is no significant difference between these two input methods.

4. Experiments

4.1. Datasets and Evaluation Protocols

We conduct experiments on nine mainstream datasets to evaluate the model's performance.

Kinetics-400 [6] is a well-known video dataset comprising 400 action categories and over 306,000 video clips, with each category containing a minimum of 400 clips. These clips, each approximately 10 seconds long, are sourced from unique YouTube videos. **Posetics** [23] was built upon the Kinetics-400 dataset. It includes 142,000 video clips across 320 action categories with corresponding 2D and 3D skeletons.

NTU-60 [24] consists of 56,880 videos across 60 action categories. The dataset provides two evaluation protocols: cross-subject (X-Sub) and cross-view (X-View). **NTU-120** [25] contains 114,000 videos across 120 action categories, and is an extended version of NTU-60. This dataset also offers two evaluation protocols: cross-subject (X-Sub) and cross-setup (X-Set).

UCF101 [26] includes 13,000 videos across 101 action categories, and **HMDB51** [27] contains 6,766 video clips across 51 action categories. Following previous work [15], split1 is used for dividing training and testing data.

FineGym99[28] is a large-scale fine-grained action recognition dataset for gymnastics, consisting of 29,000 videos across 99 fine-grained action categories.

Toyota Smarthome [7] is a real-world dataset for daily activity classification that contains 16,115 video samples across 31 action categories. The dataset includes two evaluation protocols: cross-subject (CS) and cross-view (CV1 and CV2).

Penn Action [29] consists of 2,326 video sequences across 15 different actions. In this paper, we use 2D skeletons for experiments and report Top-1 accuracy based on the standard train-test split.

For the Kinetics-400, NTU-60, NTU-120, UCF101, HMDB51, and FineGym99 datasets, we use the 2D skeleton sequences provided by PYSKL [30]. Unless otherwise stated, for NTU-60 and NTU-120, the results from other methods are based on 3D data experiments.

4.2. Experimental Settings

In 2D->3D pre-training, we utilize the Adam optimizer to train SkeleFormer for 100 epochs, implementing an exponential learning rate decay strategy. The initial learning rate is configured at 2e-4, with a weight decay coefficient of 0.1.

For the SkeleFormer fine-tuned from the 2D-to-3D pretraining task, the output of the skeleton encoder after global average pooling is fed into an MLP for classification. Similarly, we train a randomly initialized skeleton encoder for comparison. We set the learning rate for the skeleton encoder to 0.0001 and for the classification MLP to 0.001. For the SkeleFormer trained from scratch, we use a learning rate of 0.001 and a batch size of 32. The model is trained for 200 epochs.

4.3. Skeleton Action Recognition

We evaluate the performance of SkeleFormer against other models across multiple skeleton datasets, with the results reported in Table 1-5. It can be observed that 2D-to-3D pre-training significantly improves the accuracy of SkeleFormer on different datasets. Despite using 2D skeletons without depth information, SkeleFormer still achieves competitive performance, on par with or even surpassing state-of-the-art methods. This demonstrates the effectiveness of 2D-to-3D pre-training, because it enables the skeleton encoder to learn discriminative representations of skeleton sequences for subsequent generalization. SkeleFormer's superior performance is also attributed to its use of self-attention mechanisms to transfer complementary information from the bone and velocity streams to the joint stream, enabling the learning of discriminative skeleton feature embeddings. The results in Tables 1–5 demonstrate that SkeleFormer allows skeleton encoders to learn stronger and more robust features, making them better suited for cross-dataset transfer tasks.

We further investigate and compare the structural perturbation robustness of SkeleFormer and the GCN baseline. As shown in Table 6, we introduce perturbations to the input nodes of ST-GCN and SkeleFormer. ST-GCN, constrained by a fixed topology, is significantly affected by node deletion, shuffling, and random edge reconnection. In contrast, SkeleFormer, with its implicit structural learning and spatial-temporal decoupled attention, can flexibly capture semantic-driven structural relationships, making it highly robust to input perturbations.

4.4. Cross-Dataset Transfer

Since Kinetics-400 is not human-centered, many frames lack detectable human skeletons or the skeletons are difficult to identify. Therefore, we use the Posetics dataset

Table 1: Comparison of Accuracy with SOTA Methods on NTU-60 and NTU-120 Datasets. ”J,” ”B,” ”JM,” and ”BM” represent the joint, bone, joint motion, and bone motion data modalities, respectively. ”⋄” indicates the use of the same 2D skeleton data (17 keypoints).

Method	N60-CS	N60-CV	N120-CS	N120-CE
ST-GCN(J) [3]	81.5	88.3	70.7	73.2
2s-AGCN(J+B) [31]	88.5	95.1	82.5	84.2
MS-G3D(J+B) [32]	91.5	96.2	86.9	88.4
LST(J+B+JM+BM) [33]	92.9	97.0	89.9	91.1
ST-TR(J+B) [4]	89.9	96.1	84.3	86.7
ST-TR-agcn(J+B) [4]	90.3	96.3	85.1	87.1
RSA-Net(J+B+JM+BM) [5]	91.8	96.8	88.4	89.7
PoseC3D(J) ⋄ [15]	93.7	96.6	86.0	89.6
MS-G3D(J+B) ⋄ [32]	92.2	96.6	87.2	89.0
ST-GCN(J) ⋄ [3]	88.9	96.8	84.0	84.1
Ske2Grid (J+B+JM+BM) ⋄ [34]	**93.8**	**98.6**	87.3	90.8
MotionBert (scratch) [8]	87.7	94.1	-	-
MotionBert (finetune) [8]	93.0	97.2	-	-
SkeleFormer (scratch)	88.5	95.4	84.7	87.0
SkeleFormer (2D->3D)	93.1	97.1	**89.7**	**90.8**

Table 2: Comparison of Accuracy with SOTA Methods on the Kinetics-400 Dataset.

Methods	Kinetics Top-1(%)	Kinetics Top-5(%)
ST-GCN [3]	30.7	52.8
2s-AGCN [31]	36.1	58.7
MS-G3D [32]	38.0	60.9
MST-GCN [35]	38.1	60.8
ST-TR [4]	37.0	59.7
ST-TR-agcn [4]	38.0	60.5
4s-MST-GCN [35]	38.1	60.8
ML-STGNet [36]	38.9	62.2
PoseConv3D [15]	**47.7**	-
SkeleFormer (scratch)	36.1	59.5
SkeleFormer (2D->3D)	41.0	**64.1**

Table 3: Comparison of Accuracy with SOTA Methods on the Posetics Dataset.

Methods	Posetics Top-1(%)	Posetics Top-5(%)
ST-GCN [3]	43.3	67.3
2s-AGCN [31]	47.0	70.8
MS-G3D [32]	52.6	75.8
ST-TR [4]	47.5	71.3
UNIK [23]	47.6	71.3
UNIK(ft.) [23]	52.5	75.7
PoseC3D [15]	53.1	77.1
SkeleFormer (scratch)	51.7	76.0
SkeleFormer (2D->3D)	**56.9**	**80.5**

Table 4: Comparison of Accuracy with SOTA Methods on UCF101, HMDB51, and FineGYM99 Datasets.

Method	UCF101	HMDB51	FineGYM99
ST-GCN [3]	69.2	47.3	85.1
Pose-SlowOnly [15]	79.1	58.6	-
PoseConv3D [15]	87.0	69.7	93.2
Ske2Grid [34]	73.1	48.4	91.8
Hachiuma et al.[37]	87.8	**70.9**	-
SkeleFormer (2D->3D)	**88.1**	69.9	**94.3**

Table 5: Comparison of Accuracy with SOTA Methods on Toyota Smarthome and Penn Action Datasets.

Method	Toyota Smarthome			Penn Action
	CS	CV1	CV2	
ST-GCN [3]	53.8	15.5	51.1	89.6
2s-AGCN [31]	60.9	22.5	53.5	93.1
MS-G3D [32]	61.1	17.5	59.4	92.7
UNIK [23]	63.1	22.9	61.2	94.0
ML-STGNet [36]	64.6	29.9	63.5	-
SkeleFormer (2D->3D)	**64.7**	**35.7**	**64.5**	**97.6**

Table 6: Comparison of Structural Perturbation Robustness on the NTU RGB+D X-Sub dataset.

Method	Node Deletion (20%)	Node Shuffling	Edge Random Reconnection
ST-GCN [3]	-8.7%	-12.3%	-9.1%
SkeleFormer	-1.2%	-0.8%	-0.9%

as the source dataset for action recognition. We first train SkeleFormer on the Posetics dataset and then fine-tune it on datasets such as NTU-60 for skeleton-based action recognition. We use consistent skeleton data (2D with 17 joints) to fairly compare all models. Skeleton-Prompt is compared with several transfer learning methods. The parameters of the classifier are always updated during the training process:

(1) Full-tuning: All parameters are fully updated.

(2) Linear Probing: Parameters other than the linear classification layer are frozen.

(3) Bitfit [19]: Only the bias terms of the pre-trained backbone are fine-tuned.

(4) LoRA [21]: Optimized low-rank matrices are used in the multi-head attention of the transformer layers.

(5) Adapter [20]: Additional MLPs are inserted within the transformer layers.

(6) Visual Prompt tuning [11]: A series of learnable prompt tokens are added before the input patch tokens.

(7) Pro-Tuning [22]: The lightweight convolutional blocks are fine-tuned, which generate task-specific prompts.

(8) Skeleton-Prompt without CA: The generated skeleton prompts are directly concatenated with pose tokens and input into the next transformer layer.

For each transfer method, we test the following learning rates: {0.0001, 0.0005, 0.001, 0.005}. For the prompt-based methods, we fix the weight decay at 0.0001 . For other methods (non-prompt-based), we vary the weight decay values in the set {0.001, 0.0001}. We train for a total of 100 epochs, with an initial warm-up period of 10 epochs. We use the AdamW optimizer and cosine-decay learning rate scheduler. We select an appropriate learning rate (non-prompt-based), we vary the weight decay values in the set {0.001, 0.0001}. We train for a total of 100 epochs, with an initial warm-up period of 10 epochs. We use the AdamW optimizer and the cosine-decay learning rate scheduler.

Table 7: Comparison of Transfer Results of Different Tuning Methods Across Multiple Skeleton Datasets.

Method	N60-CS	N60-CV	N120-CS	N120-CE	UCF101	HMDB51	FineGYM	Smarthome	Penn	Params (M)
Full fine-tuning	93.7	97.6	90.4	91.5	89.1	70.8	94.7	65.2	98.3	9.58
Linear Probing	82.9	85.5	78.0	78.6	77.9	59.7	79.8	53.5	87.9	0.04
Bitfit [19]	84.2	87.6	81.1	82.9	80.3	61.3	80.7	56.3	89.4	0.19
LoRA [21]	88.1	90.3	83.5	84.4	81.3	62.4	82.5	58.6	90.5	0.32
VPT [11]	89.1	92.9	84.6	84.9	82.0	63.9	83.3	61.1	92.4	0.57
Adapter [20]	89.5	93.1	85.5	86.4	84.2	64.3	84.4	62.0	92.3	1.31
Pro-Tuning [22]	92.3	95.1	87.7	88.8	86.3	67.8	90.1	63.3	96.4	1.65
Skeleton-Prompt w/o CA	91.4	94.2	87.0	87.9	85.0	66.2	88.8	62.5	94.6	0.40
Skeleton-Prompt	93.6	96.8	90.2	91.0	88.8	70.7	93.1	65.0	97.5	0.40

We select an appropriate learning rate and keep the batch size fixed at 64.

Table 7 displays the transfer results of Skeleton-Prompt across multiple datasets. The proposed method achieves performances comparable to those of full fine-tuning. Skeleton-Prompt without CA also demonstrates good transfer performance, with instance-based dynamic skeleton prompts showing strong generalization for cross-dataset recognition. Skeleton-Prompt has significantly fewer trainable parameters than full fine-tuning, making it deployable across multiple datasets without the need to redundantly store a large number of fundamental parameters, providing a substantial advantage in real-world applications. By introducing cross-attention in prompt tuning, Skeleton-Prompt outperforms Skeleton-Prompt without CA, and achieves state-of-the-art performance compared to previous PETL methods across different evaluation protocols. Previous work [32] demonstrates weaker transferability, as dataset-specific model configurations do not always adapt well to the transferred datasets.

Table 8: Efficiency Comparison of Different Fine-Tuning Methods in the Posetics → NTU-60 Transfer Task.

Method	Trainable Params (M)	Training Time / Epoch (min)	Inference FPS	Accuracy (%)
Full Fine-Tuning	9.58	25.3	31	93.7
LoRA [21]	0.32	18.7	29	88.1
VPT (p = 10) [11]	0.57	16.2	28	89.1
Skeleton-Prompt	0.40	14.5	27	93.6

As shown in Table 8, Skeleton-Prompt requires training only 0.40M parameters

(4.2% of the full fine-tuning baseline), reducing per-epoch training time by 42.7% (14.5 vs. 25.3 minutes), with only a 12.9% drop in inference speed (27 vs. 31 FPS). In cross-dataset transfer, Skeleton-Prompt achieves the highest accuracy (93.6%) while maintaining the lowest parameter count and shortest training time. By comparison, LoRA suffers from impaired temporal modeling due to its low-rank approximation (88.1% accuracy), and VPT is limited by the inflexibility of static prompts (89.1% accuracy).

4.5. Different Numbers and Structures of Nodes

We further investigate the inconsistency in the number and structure of skeletal nodes in cross-dataset transfer tasks. We conduct experiments using the original NTU-60 CS dataset (with 25 3D keypoints) and the Posetics dataset (with 17 2D keypoints), applying a full fine-tuning approach for transfer. We compare the performance of SkeleFormer with the GCN baseline ST-GCN, and the performance differences are shown in Table 9.

Table 9: Comparison of Transfer Performance Across Different Numbers and Structures of Nodes.

Datasets	ST-GCN	SkeleFormer
Original NTU-60	81.5%	90.4%
Posetics → Original NTU-60	81.6%	92.6%
Posetics	43.3%	57.8%
Original NTU-60 → Posetics	43.3%	57.8%

Due to the differences in node count and structure between the two datasets, transfer training does not yield performance gains for the GCN method. However, SkeleFormer overcomes the fixed topology limitation of GCN through sequence modeling and a pretraining strategy that decouples motion semantics, systematically addressing the challenge of inconsistent node numbers and structures across different skeletal datasets.

4.6. Qualitative Analysis

4.6.1. Joint-Frame Heatmaps

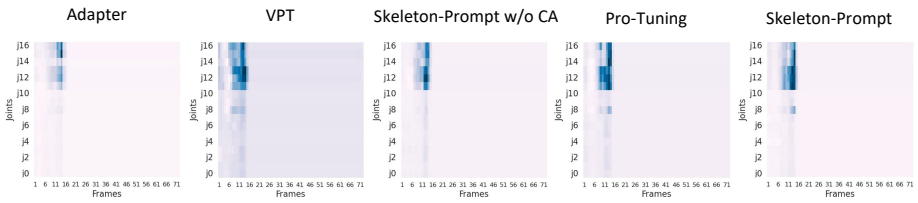
As shown in Figure 4, we conduct an in-depth analysis of the tuning performance on NTU-120 for Adapter (1st column), VPT (2nd column), Skeleton-Prompt w/o CA (3rd column), Pro-Tuning (4th column), and Skeleton-Prompt (5th column) using the attribution algorithm BIG [38]. The resulting attribution scores are visualized as joint-frame heatmaps, where the brightness of each grid reflects its importance in the prediction process. Compared to other methods, Skeleton-Prompt more accurately clusters action-relevant joints in the spatial dimension and achieves clearer boundaries in the temporal dimension, enabling it to learn more discriminative features (best viewed in color). The visualization results demonstrate that Skeleton-Prompt provides a more comprehensive action representation and exhibits a superior capability for cross-dataset representation learning.

4.6.2. Feature Space Visualization

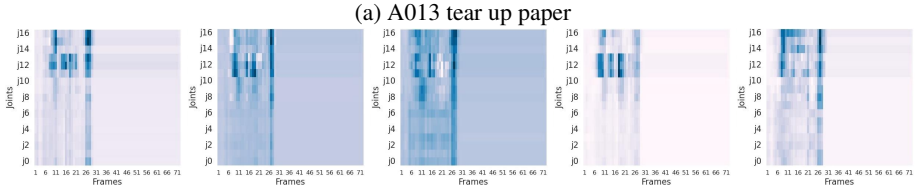
In the Posetics → NTU-60 transfer task, we use t-SNE to visualize the output features from the last layer of the Transformer model. We randomly selected 20 categories from the NTU-60 dataset, with 100 samples per category. As shown in Figure 5, the t-SNE visualization clearly validates the significant advantage of Skeleton-Prompt in decision boundary clarity. Compared to other fine-tuning methods, where similar actions are loosely clustered, Skeleton-Prompt shows clear separation between different categories.

4.6.3. Action Dynamics

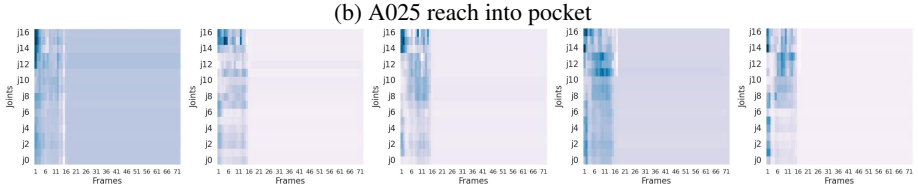
As shown in Figure 6, using NTU-RGB+D as an example, we categorize actions into three groups based on their dynamic levels: low, medium, and high. We further explore the accuracy differences (in %) between Skeleton-Prompt and the baseline ST-GCN for each action category on the NTU RGB+D X-Sub dataset. From the analysis of Figure 6, it can be observed that Skeleton-Prompt performs significantly better than the baseline method on medium and low dynamic actions, i.e., actions with medium to long temporal dependencies (e.g., ”reading,” ”writing”). The advantages



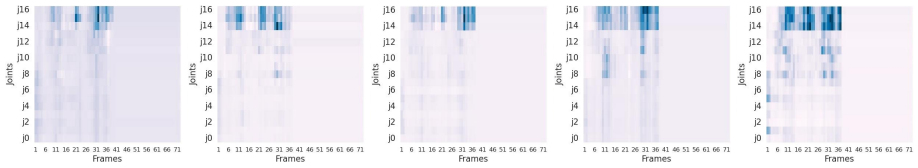
(a) A013 tear up paper



(b) A025 reach into pocket



(c) A079 sniff/smell



(d) A085 apply cream on face

Figure 4: Joint-frame heatmaps for (a) A013 “tear up paper”, (b) A025 “reach into pocket”, (c) A079 “sniff/smell” and (d) A085 “apply cream on face” (best viewed in color).

Figure 5: Visualization of Transfer Feature Distributions for Different Tuning Methods (best viewed in color).

of global attention in modeling long-range dependencies are evident. Additionally, the prompt generator identifies key stages through inter-frame velocity features (e.g.,

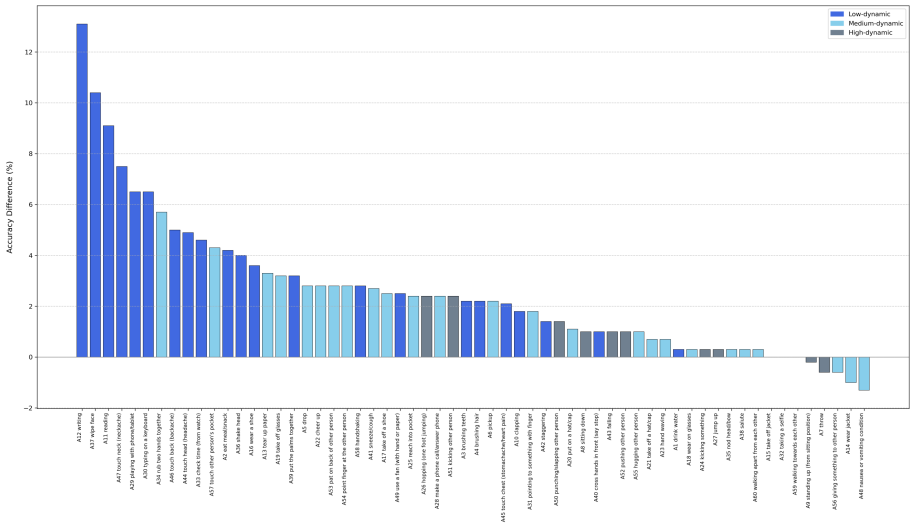


Figure 6: Top-1 accuracy differences (in %) between Skeleton-Prompt and ST-GCN on joint input modality in the NTU RGB+D X-Sub dataset (best viewed in color).

sudden changes in acceleration). Skeleton-Prompt's performance is limited on high-dynamic fast actions (e.g., "throw"), as the fixed positional encoding in the Transformer results in information loss when discretizing high-frequency movements (such as rapid wrist rotation).

4.6.4. Failure Cases

Figure 7 presents typical examples of errors made by the proposed method when recognizing skeletal actions, including "reading" vs. "playing with phone/tablet," "take off glasses" vs. "wipe face," "wear a shoe" vs. "take off a shoe," and "pointing to something with finger" vs. "taking a selfie." These failure cases occur when action categories are determined based on fine finger movements, which involve subtle spatial features and insufficient discriminative cues in the skeletal feature space.

4.7. Ablation Experiments

4.7.1. Different Components

In Table 10, we perform ablation experiments on the components of the proposed method. Even without 2D-to-3D lifting, SkeleFormer pre-trained on the Posetics dataset

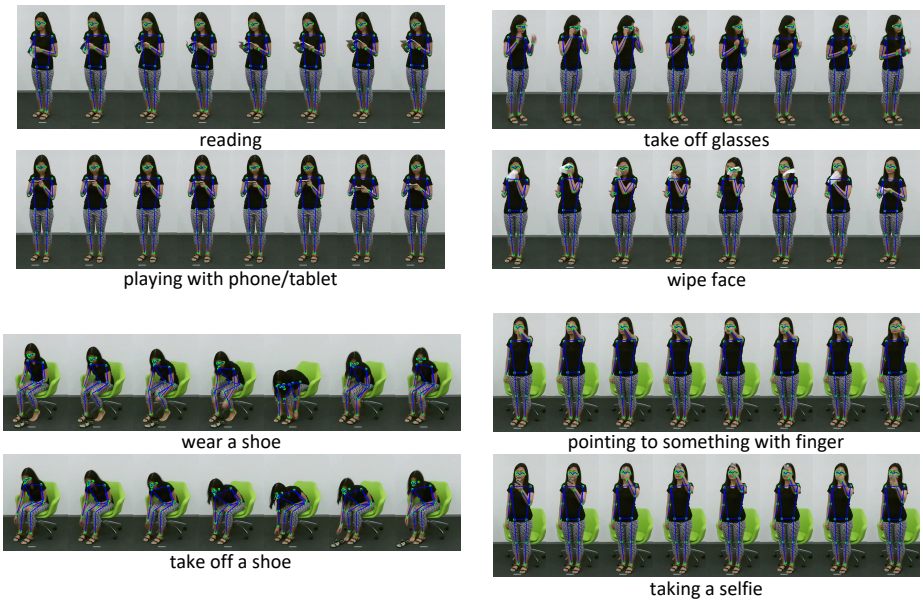


Figure 7: Failure Case Analysis in Cross-Dataset Transfer (best viewed in color).

Table 10: Comparison of the transfer performance of different components of Skeleton-Prompt across multiple skeleton datasets.

w/ 2D->3D	w/ JEMF	w/ CA	N60-CS	N60-CV	N120-CS	N120-CE	UCF101	HMDB51	FineGYM	Smarthome	Penn
	✓	✓	89.5	92.4	84.5	85.7	83.5	65.0	86.0	61.1	93.6
✓	✓	✓	93.6	96.8	90.2	91.0	88.8	70.7	93.1	65.0	97.5
✓		✓	91.7	94.6	87.5	88.6	85.2	66.2	89.4	62.5	94.5
✓	✓		91.4	94.2	87.0	87.9	85.0	66.2	88.8	62.5	94.6

still performs exceptionally well across multiple datasets. Notably, on smaller datasets such as HMDB51 and Penn, Skeleton-Prompt shows particularly outstanding transfer performance. Our analysis of the cross-attention between pose embeddings and skeleton prompts reveals that it enhances mutual focus and semantic richness within the prompts, further improving the flexibility and effectiveness of cross-dataset transfer learning. Moreover, the introduction of JEMF significantly improves accuracy across multiple skeleton datasets by enhancing the robustness of skeleton features through the fusion of joint, bone, and velocity streams.

4.7.2. Prompt Design

In the Posetics → NTU-60 cross-dataset transfer task, we conduct ablation experiments on prompt structure, length, and position. As shown in Tables 11 and 12, the optimal accuracy (93.6%) is achieved with a prompt length of 10. Longer prompts (20) lead to a performance decrease (-0.4%) due to the introduction of redundant information, while shorter prompts (5) fail to effectively capture the semantic differences across datasets (-1.5%). Cross-attention significantly outperforms direct concatenation (+4.5%) and element-wise addition (+4.2%) due to its ability to focus on key semantic areas using learnable attention weights.

Table 11: Ablation Experiment on Prompt Length in the Posetics → NTU-60 Transfer Task.

Prompt Length	Top-1 Accuracy(%)	Inference FPS
5	92.1	29
10	93.6	27
20	93.2	24

Table 12: Ablation Experiment on Prompt Position and Interaction Mechanism in the Posetics → NTU-60 Transfer Task.

Prompt Position	Interaction Mechanism	Top-1 Accuracy(%)
Layer-wise Concatenation	Direct Concatenation	89.1
Layer-wise Concatenation	Element-wise Addition	89.6
Layer-wise Concatenation	Cross-Attention	93.6
Mid-layer Insertion	Cross-Attention	92.9

4.7.3. Multi-Stream Fusion Strategies

As shown in Table 13, we compared the following fusion strategies (with other modules kept consistent) on the NTU-60 dataset.

(1) Concatenation: The prompt vector and the original pose embedding are concatenated along the channel dimension and then input into the Transformer.

(2) Addition: The prompt vector and the pose embedding are added element-wise.

(3) Multiplication: The prompt vector and the pose embedding are multiplied element-wise.

(4) Cross-Attention (Ours): The cross-attention between the pose embedding and the prompt is computed, and the result is added in residual form.

Table 13: Performance Comparison of Multi-Stream Fusion Strategies on the Posetics → NTU-60 Transfer Task.

Fusion Strategy	Top-1 Accuracy(%)	Trainable Params (M)
Concatenation	89.0	0.51
Addition	89.4	0.40
Multiplication	88.8	0.41
Cross-Attention (Ours)	93.6	0.40

In the Posetics → NTU transfer task, the accuracy of the cross-attention mechanism (93.6%) is significantly higher than that of other strategies. We analyze that cross-attention adaptively adjusts the prompt-embedding correlation based on the characteristics of the target dataset, while static fusion strategies (such as concatenation) fail to adapt to domain differences. Residual learning overlays the attention results in residual form, preserving the cross-domain generalization ability of the original embedding and avoiding feature distortion caused by direct modification.

4.8. Varying Data Scales

We evaluate the transfer performance of different tuning methods across varying dataset sizes, gradually increasing the training set size from 10% to 100%. As shown in Table 14, Skeleton-Prompt consistently outperforms other baselines under limited data conditions. When only 30% of the training data is used, Skeleton-Prompt achieves a performance comparable to the classic Bitfit, which requires 50% of the training data. However, when the training data are extremely scarce, Skeleton-Prompt struggles to effectively train the prompt generator, resulting in a sharp decline in performance. Once the training data exceeds 50%, Skeleton-Prompt can generate meaningful skeletal prompts, surpassing other methods and exhibiting stronger generalization ability.

Table 14: Comparison of the transfer performance of tuning methods at different training data scales under the NTU-120 CS evaluation protocol.

Fraction	Bitfit	LoRA	VPT	Adapter	Pro-Tuning	Skeleton-Prompt
10%	**66.5**	61.6	65.8	65.0	66.2	64.8
20%	72.2	70.0	71.5	72.1	**75.4**	69.3
30%	74.3	74.8	75.5	75.9	**80.9**	77.3
40%	76.4	77.9	77.1	78.3	**82.4**	81.9
50%	77.4	79.7	79.5	80.2	**84.5**	84.3
60%	78.6	80.9	80.6	81.6	85.1	**86.2**
70%	79.4	81.9	81.7	82.4	85.9	**87.6**
80%	80.0	82.5	82.6	83.4	86.7	**88.8**
90%	80.8	83.1	83.8	84.6	87.3	**89.6**
100%	81.1	83.5	84.6	85.5	87.7	**90.2**

4.9. Robustness to Occlusion

Referring to the experimental setup in [18], we evaluate the model's robustness to occlusion under the NTU-60 CS protocol, categorizing it into two cases: part occlusion and frame occlusion. We train the models on unobstructed skeleton data and test them on occluded data. The types of part occlusion are categorized as None, without left arm (1), right arm (2), two hands (3), two legs (4), and trunk (5). The number of occluded frames is set to 0, 10, 20, 30, 40, and 50. Tables 8 and 9 demonstrate the superiority of our method for recognizing skeleton actions under occlusion. Compared with RA-GCN and PDGCN, which are specifically designed to address skeletal occlusion, our method still exhibits significant advantages under various occlusion conditions.

We compare the proposed JMSF with the input fusion and post-fusion methods. In the input fusion baseline, joint embeddings, bone embeddings, and velocity embeddings from the three streams are merged into a single stream and input into the skeleton encoder. In the post-fusion baseline, the three streams are input into the skeleton encoder separately, and the final embeddings are summed afterward. As shown in Tables 15-16, under various occlusion conditions, JMSF outperforms both input fusion and post-fusion baselines while using smaller model sizes for both inference and training. This demonstrates the robustness of the JMSF in handling occluded skeleton data.

Table 15: Comparison of results of different methods under part occlusion using the NTU-60 CS protocol.

Part Occlusion	Occluded Part					
	None	1	2	3	4	5
ST-GCN [3]	80.7	71.4	60.5	62.6	77.4	50.2
2s-AGCN [31]	88.5	72.4	55.8	82.1	74.1	71.9
3s RA-GCN [18]	87.3	74.5	59.4	74.2	83.2	72.3
STIGCN [39]	88.8	12.7	11.5	18.3	45.5	20.9
MS-G3D [32]	87.3	31.3	23.8	17.1	78.3	61.6
3s PDGCN [40]	87.5	76.0	62.0	75.4	85.0	73.0
SkeleFormer (input)	91.8	81.1	71.9	72.7	82.8	81.6
SkeleFormer (post)	92.1	81.8	72.8	73.1	83.3	82.0
SkeleFormer (JMSF)	**93.1**	**84.2**	**75.4**	**76.6**	**87.1**	**86.5**

Table 16: Comparison of results of different methods under frame occlusion using the NTU-60 CS protocol.

Frame Occlusion	Number of Occluded Frames					
	0	10	20	30	40	50
ST-GCN [3]	80.7	69.3	57.0	44.5	34.5	24.0
2s-AGCN [31]	88.5	74.8	60.8	49.7	38.2	28.0
3s RA-GCN [18]	87.3	83.9	76.4	66.3	53.2	38.5
STIGCN [39]	88.8	70.4	51.0	38.7	23.8	8.0
MS-G3D [32]	87.3	77.6	65.7	54.3	41.9	30.1
3s PDGCN [40]	87.5	83.9	76.6	66.7	53.9	40.0
SkeleFormer (input)	91.8	82.8	74.5	63.5	50.2	36.6
SkeleFormer (post)	92.1	83.3	75.6	65.1	52.3	38.6
SkeleFormer (JMSF)	**93.1**	**85.0**	**77.9**	**67.9**	**55.5**	**41.7**

5. Limitations and Future Work

Although Skeleton-Prompt reduces the number of parameters by freezing the backbone network, the cross-attention mechanism in the prompt generator introduces additional computational costs. In real-time scenarios (such as online action recognition), inference speed may be constrained. In future work, we will adopt Neural Architecture Search (NAS) or knowledge distillation techniques to reduce the computational complexity of the cross-attention module.

6. Conclusion

In this work, we present SkeleFormer, a novel framework that addresses critical challenges in skeleton action recognition by leveraging 2D-to-3D pose estimation for robust pretraining. Our proposed skeleton prompt generator with cross-attention enhances model adaptability across datasets, enabling efficient knowledge transfer with minimal parameter updates. Additionally, the Joint-Enhanced Multi-Stream Fusion method improves robustness against occluded skeleton data by integrating joint, bone, and velocity streams through self-attention. While our approach effectively mitigates issues of scalability and dataset variability, challenges remain regarding the generalization to highly noisy or incomplete skeleton data. The lack of large-scale 3D skeleton datasets also limits broader applicability. Nevertheless, this study demonstrates significant potential for advancing cross-dataset transfer learning and inspires avenues for future research. These include expanding datasets, exploring unsupervised or semi-supervised approaches, and integrating contextual information. Our contributions provide a robust foundation for researchers and practitioners to refine skeleton-based action recognition and extend its applicability to real-world scenarios.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62271143, in part by Frontier Technologies R&D Program of Jiangsu under Grants BF2024060, in part by the Big Data Computing Center of Southeast University.

References

[1] M. Liu, H. Liu, C. Chen, Enhanced skeleton visualization for view invariant human action recognition, Pattern Recognition 68 (2017) 346–362.

[2] M. Naveenkumar, S. Domnic, Deep ensemble network using distance maps and body part features for skeleton based action recognition, Pattern Recognition 100 (2020) 107125.

[3] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Proceedings of the AAAI conference on artificial intelligence, 2018.

[4] C. Plizzari, M. Cannici, M. Matteucci, Skeleton-based action recognition via spatial and temporal transformer networks, Computer Vision and Image Understanding 208 (2021) 103219.

[5] K. Gedamu, Y. Ji, L. Gao, Y. Yang, H. T. Shen, Relation-mining self-attention network for skeleton-based human action recognition, Pattern Recognition 139 (2023) 109455.

[6] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2017, pp. 6299–6308.

[7] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, G. Francesca, Toyota smarthome: Real-world activities of daily living, in: Proceedings of the IEEE international conference on computer vision, 2019, pp. 833–842.

[8] W. Zhu, X. Ma, Z. Liu, L. Liu, W. Wu, Y. Wang, Motionbert: A unified perspective on learning human motion representations, in: Proceedings of the IEEE international conference on computer vision, 2023, pp. 15085–15099.

[9] C. Ionescu, D. Papava, V. Olaru, C. 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 (2013) 1325–1339.

[10] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, Z. Ding, 3d human pose estimation with spatial and temporal transformers, in: Proceedings of the IEEE international conference on computer vision, 2021, pp. 11656–11665.

[11] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.-N. Lim, Visual prompt tuning, in: European Conference on Computer Vision, 2022, pp. 709–727.

[12] T.-T. Nguyen, D.-T. Pham, H. Vu, T.-L. Le, A robust and efficient method for skeleton-based human action recognition and its application for cross-dataset evaluation, IET Computer Vision 16 (8) (2022) 709–726.

[13] K. Peng, C. Yin, J. Zheng, R. Liu, D. Schneider, J. Zhang, K. Yang, M. S. Sarfraz, R. Stiefelhagen, A. Roitberg, Navigating open set scenarios for skeleton-based action recognition, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 38, 2024, pp. 4487–4496.

[14] T. Yan, W. Zeng, Y. Xiao, X. Tong, B. Tan, Z. Fang, Z. Cao, J. T. Zhou, Crossglg: Llm guides one-shot skeleton-based 3d action recognition in a cross-level manner, in: European Conference on Computer Vision, Springer, 2024, pp. 113–131.

[15] C. K. Duan H, Zhao Y, Revisiting skeleton-based action recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, p. 2969–2978.

[16] K. Peng, A. Roitberg, K. Yang, J. Zhang, R. Stiefelhagen, Delving deep into one-shot skeleton-based action recognition with diverse occlusions, IEEE Transactions on Multimedia 25 (2023) 1489–1504.

[17] Y. Xu, K. Peng, D. Wen, R. Liu, J. Zheng, Y. Chen, J. Zhang, A. Roitberg, K. Yang, R. Stiefelhagen, Skeleton-based human action recognition with noisy labels, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2024, pp. 4716–4723.

[18] Y.-F. Song, Z. Zhang, C. Shan, L. Wang, Richly activated graph convolutional network for robust skeleton-based action recognition, IEEE Transactions on Circuits and Systems for Video Technology 31 (2020) 1915–1925.

[19] E. B. Zaken, S. Ravfogel, Y. Goldberg, Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, in: ACL, 2022, pp. 1–9.

[20] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International conference on machine learning, 2019, pp. 2790–2799.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022.

[22] X. Nie, B. Ni, J. Chang, G. Meng, C. Huo, S. Xiang, Q. Tian, Pro-tuning: Unified prompt tuning for vision tasks, IEEE Transactions on Circuits and Systems for Video Technology (2023).

[23] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, F. Brémond, Unik: A unified framework for real-world skeleton-based action recognition, in: The British Machine Vision Conference, 2021.

[24] A. Shahroudy, J. Liu, T.-T. Ng, G. Wang, Ntu rgb+ d: A large scale dataset for 3d human activity analysis, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2016, pp. 1010–1019.

[25] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, A. C. Kot, Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding, IEEE transactions on pattern analysis and machine intelligence 42 (10) (2019) 2684–2701.

[26] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402 (2012).

[27] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: Proceedings of the IEEE international conference on computer vision, 2011, pp. 2556–2563.

[28] D. Shao, Y. Zhao, B. Dai, D. Lin, Finegym: A hierarchical video dataset for fine-grained action understanding, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2616–2625.

[29] W. Zhang, M. Zhu, K. G. Derpanis, From actemes to action: A strongly-supervised representation for detailed action understanding, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 2248–2255.

[30] H. Duan, J. Wang, K. Chen, D. Lin, Pyskl: Towards good practices for skeleton action recognition, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 7351–7354.

[31] L. Shi, Y. Zhang, J. Cheng, H. Lu, Two-stream adaptive graph convolutional networks for skeleton-based action recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12026–12035.

[32] C. Z. Liu Z, Zhang H, Disentangling and unifying graph convolutions for skeleton-based action recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, p. 143–152.

[33] W. Xiang, C. Li, Y. Zhou, B. Wang, L. Zhang, Generative action description prompts for skeleton-based action recognition, in: Proceedings of the IEEE international conference on computer vision, 2023, pp. 10276–10285.

[34] D. Cai, Y. Kang, A. Yao, Y. Chen, Ske2grid: skeleton-to-grid representation learning for action recognition, in: International conference on machine learning, 2023, pp. 3431–3441.

[35] Z. Chen, S. Li, B. Yang, Q. Li, H. Liu, Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 1113–1122.

[36] Y. Zhu, H. Shuai, G. Liu, Q. Liu, Multilevel spatial–temporal excited graph network for skeleton-based action recognition, IEEE Transactions on Image Processing 32 (2022) 496–508.

[37] R. Hachiuma, F. Sato, T. Sekii, Unified keypoint-based action recognition framework via structured keypoint pooling, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 22962–22971.

[38] Z. Wang, M. Fredrikson, A. Datta, Robust models are more interpretable because attributions look normal, in: International conference on machine learning, Vol. 162, 2021, pp. 22625–22651.

[39] Z. Huang, X. Shen, X. Tian, H. Li, J. Huang, X.-S. Hua, Spatio-temporal inception graph convolutional networks for skeleton-based action recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2122–2130.

[40] Z. Chen, H. Wang, J. Gui, Occluded skeleton-based human action recognition with dual inhibition training, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 2625–2634.