Unified Pose Sequence Modeling

Lin Geng Foo\(^1\)*, Tianjiao Li\(^1\)*, Hossein Rahmani\(^2\), Qiuhong Ke\(^3\), Jun Liu\(^1\)*

\(^1\)Singapore University of Technology & Design \(^2\)Lancaster University \(^3\)Monash University

\{lingeng.fool,tianjiao.li\}@mymail.sutd.edu.sg, h.rahmani@lancaster.ac.uk, qiuhong.ke@monash.edu, jun.liu@sutd.edu.sg

Abstract

We propose a Unified Pose Sequence Modeling approach to unify heterogeneous human behavior understanding tasks based on pose data, e.g., action recognition, 3D pose estimation and 3D early action prediction. A major obstacle is that different pose-based tasks require different output data formats. Specifically, the action recognition and prediction tasks require class predictions as outputs, while 3D pose estimation requires a human pose output, which limits existing methods to leverage task-specific network architectures for each task. Hence, in this paper, we propose a novel Unified Pose Sequence (UPS) model to unify heterogeneous output formats for the aforementioned tasks by considering text-based action labels and coordinate-based human poses as language sequences. Then, by optimizing a single auto-regressive transformer, we can obtain a unified output sequence that can handle all the aforementioned tasks. Moreover, to avoid the interference brought by the heterogeneity between different tasks, a dynamic routing mechanism is also proposed to empower our UPS with the ability to learn which subsets of parameters should be shared among different tasks. To evaluate the efficacy of the proposed UPS, extensive experiments are conducted on four different tasks with four popular behavior understanding benchmarks.

1. Introduction

Pose sequences, which capture the movements of the human body via human joint coordinates, are well-known to be an efficient and effective representation of human motion and behaviour [58,80]. This is mainly because pose sequences often provide enough information to characterize complex motion patterns [31], while being robust against superficial visual variations such as the background, clothing texture and illumination conditions [43,44]. At the same time, by using depth sensors such as the Kinect, pose data can also be conveniently obtained in real-time to facilitate downstream applications. Therefore, the potential of pose sequences to tackle behaviour understanding has attracted a lot of attention in recent years.

Notably, the usage of pose sequences has been widely explored across many practical applications, including human-robot interaction [1,59], augmented reality [4,51] and security surveillance [18,71]. Specifically, pose sequences, as informative inputs, can facilitate certain aspects in these applications, such as action recognition [12,44,48,66,67,86–88], 3D pose estimation [41,45,84,92,95,96] and early action prediction [23,35,39,77,78], making these tasks popular and important areas of research.

However, existing methods for each task still often require task-specific architectures, e.g., hourglass networks for pose estimation [84] and specialized GCN architectures for action recognition [11,69], while the performing of multiple pose-based tasks with a single model is not well explored. Therefore, in order to perform multiple tasks, users will often need to design and train multiple separate models, which can be inconvenient and inefficient.

Hence, in this work, we seek to simplify and unify the modeling for several popular and important pose-based tasks: 3D action recognition, 2D action recognition, 3D pose estimation and 3D early action prediction. This is a challenging goal that has not been achieved before, requiring a single model to cover a large scope involving 2D tasks, 3D tasks, as well as 2D to 3D lifting. By unifying these pose-based tasks and removing the need to design and train separate task-specific models to tackle different pose-based tasks, we can greatly reduce the difficulty and complexity involved in tackling these tasks. Moreover, a unified model is also an elegant way of handling multiple tasks that brings us one step closer in our pursuit of general purpose vision systems [25], i.e., an efficient multi-purpose AI model akin to the human brain.

To this end, we propose a Unified Pose Sequence (UPS) model to unify the architecture and output format for multiple popular pose-based tasks. Our UPS is a single uni-
fied model that simultaneously tackles multiple tasks without task-specific designs or branches, i.e., with a unified decoder. In order to unify the output formats of different tasks (which can be very different) to be produced by a single decoder, our UPS predicts a sequence of output tokens, similar to language modeling tasks. Specifically, our UPS’s decoder auto-regressively produces a sequence of output tokens, such that the output sequence can potentially be of different lengths to meet the requirements of multiple tasks. Additionally, these output tokens can be interpreted as text embeddings, which are a powerful and general representation that can be mapped into various predictions as required. Moreover, to mitigate the potential destructive interference [60, 90] brought by the heterogeneity between different tasks, we propose a dynamic routing mechanism for our UPS that facilitates parameter sharing between tasks.

In summary, our contributions are as follows:

- We propose a Unified Pose Sequence (UPS) model that can tackle several popular pose-based tasks through a single unified framework. UPS simultaneously tackles multiple tasks without task-specific designs or branches by modeling the output as a sequence of tokens, enabling it to handle different output formats.

- On four popular pose-based tasks (3D action recognition, 2D action recognition, 3D pose estimation and 3D early action prediction), UPS achieves good performance that is comparable to state-of-the-art methods.

2. Related Work

Pose-based Tasks. Due to the practicality and effectiveness of pose sequences in capturing human motion and behaviour [58, 80], how to better leverage upon them to perform various tasks has become an increasingly popular area of research. There are a few important tasks in particular that have received a lot of attention. 2D and 3D Action Recognition [12, 13, 20, 21, 36, 38, 44, 48, 66, 67, 88] is where we predict the action class of the input 2D and 3D pose sequence respectively. In 3D Pose Estimation [24, 41, 45, 84, 95, 96], we predict the 3D coordinates of a human’s joints, with the input being either RGB images [53, 70] or 2D poses [24, 41, 45, 84, 95, 96]. In this work, we use 2D pose sequences as input. Besides, in 3D Early Action Prediction [23, 27, 35, 39, 76–78], we would like to predict the action performed by the subject, after observing only the front parts of each pose sequence. However, these existing methods often require task-specific architectures, and the performing of multiple pose-based tasks with a single model is not well explored. Hence, in this work, we seek a unified model to tackle these tasks simultaneously.

Sequence Modeling. Sequence modeling is an important concept in the field of NLP, particularly for the generation of a sentence as a sequence of words [57, 73]. In general, a Transformer is used in an auto-regressive manner [57, 73], taking previous tokens as input to predict the next token, and thus generating a sequence of tokens. Recently, such sequence modeling has also been explored for some vision-language tasks [8, 14, 25, 75, 97]. Different from previous works, we explore the crucial challenge of unifying several popular pose-based tasks. Not only do these skeleton-based tasks often require different task-specific designs to successfully tackle, they also require vastly different output formats (e.g., video-level classification vs joint-level coordinates) and input formats (e.g., 2D vs 3D pose).

Multi-task Learning. A related field is multi-task learning [15, 93], where models are trained to perform multiple tasks simultaneously. In general, there are several approaches to multi-task learning, including multi-task architecture designs [16, 50, 62, 94], optimization methods [22, 90], and learning of task relationships [3, 17, 26, 79]. However, in existing works on multi-task learning, each task still requires dedicated task-specific branches, thus extending new tasks in this setting will require additional sets of parameters, e.g., task-specific heads. Differently, our UPS unifies multiple skeleton-based tasks into one single model by integrating all output formats into a language-based format, and can conveniently handle various tasks without needing any modification to the model architecture.

Language Models for Vision Tasks. Language models have often been applied to facilitate vision-based tasks. For instance, language is used as input in text-to-image generation [61, 85] and visual grounding [32, 83, 89], while language is an output for the image captioning [9, 56] and visual question answering [2, 82] tasks. Here, we propose a novel paradigm integrating language and pose into a unified model to handle various pose and action tasks.

3. Method

3.1. Overview

In order to unify diverse pose-based human behavior understanding tasks (e.g., action recognition, 3D pose estimation, early action prediction) with a single model, we propose a novel Unified Pose Sequence model as shown in Fig. 1. A major obstacle we face is that different tasks require different output formats. For instance, the action recognition task requires class predictions, while the 3D pose estimation task requires 3D locations of human joints.

Therefore, to jointly represent the output sequences from our UPS and heterogeneous ground-truth formats from different tasks, we utilize sequence modeling [8] to unify different target data formats and avoid multiple task-specific output heads. Specifically, we tokenize the text-based class labels and coordinate-based joint locations following the standard language modeling setup and establish a unified vocabulary. Then, given the UPS output token sequences,
The UPS decoder will stop generating outputs if any of the three ending tokens (i.e., EOT, TEM, SEM) are encountered. Note that after quantization, we have $3 \times n_{bins}$ discrete bins for $X$-dim, $Y$-dim and $Z$-dim in total.

Then, we can represent each bin by a text-prompted token extracted by the same language model used for action token (e.g., RoBERTa [47]). Here, we take the $X$-dim as an example: we describe the first location on the $X$-dim as “The first horizontal coordinate” and so forth, where the last location on the $X$-dim can be queried by “The \(n_{bins}-th\) horizontal coordinate”. Then we send these descriptions to the pre-trained language model and extract the three $n_{bins}$ discrete joint tokens. In such a way, an arbitrary joint $p_{n,j}$ can be denoted using three discrete tokens $s^X_{n,j}, s^Y_{n,j}$ and $s^Z_{n,j} \in \mathbb{R}^d$ along each dimension. Therefore, the target coordinate-format pose sequence $P_{\text{TAR}} \in \mathbb{R}^{N \times J \times V}$ can be further discretely tokenized into a pose token sequence $S_{\text{TAR}}$, i.e., $\{s^X_{1,1}, s^Y_{1,1}, \ldots, s^X_{N,J}, s^Y_{N,J}, s^Z_{N,J}\}$.

**EOTs: End of Task Tokens.** Another challenge we face is that the different tasks can require output sequences of different lengths. For example, for action recognition we only require one output token to represent the class prediction, while for 3D pose estimation we need to produce a longer sequence of $J$ joint coordinates. To adaptively produce the output token sequence as required while avoiding the usage of multiple task-specific output heads, we introduce end of task tokens (EOTs) to indicate when to stop the decoding process for the UPS model. Similar to joint tokens, we leverage text prompts to produce these tokens, e.g., the description “action recognition ends here” is sent to the pretrained language model to generate the ending token for action recognition task $EOT_{\text{AR}}$. Here we define 3 types of
our routing mechanism, which is described in Sec. 3.4. Below, we introduce a joint coordinates tokenizer, which is composed of three TCN-GCN layers [11]. The joint tokenizer takes \( P_{\text{IN}} \in \mathbb{R}^{N \times J \times d} \) as input, and produces input token sequence \( S_{\text{IN}} \in \mathbb{R}^{N \times J \times d} \).

**Joint Coordinates Tokenizer.** To tokenize the input pose sequence \( P_{\text{IN}} \) (which is in the coordinates format), we introduce a joint coordinates tokenizer, which is comprised of three TCN-GCN layers [11]. The joint tokenizer takes \( P_{\text{IN}} \in \mathbb{R}^{N \times J \times V} \) as input, and produces input token sequence \( S_{\text{IN}} \in \mathbb{R}^{N \times J \times d} \).

**UPS Encoder.** Our UPS encoder consists of \( L_{\text{encoder}} \) stacks of SEM-TEM blocks, where each SEM-TEM block is a SEM module followed by a TEM module (as described in more detail below). The UPS encoder takes \( S_{\text{IN}} \) as inputs and produce encoded hidden token features \( S_{\text{EN}} \in \mathbb{R}^{N \times J \times d} \). We remark that, as shown in Fig. 1, we design a routing mechanism and send an additional task token to the UPS encoder, to adaptively determine which subsets of parameters should be shared for different tasks. Details of the routing mechanism are outlined in Sec. 3.4. Below, we describe the SEM and TEM in detail.

1. **Spatial Encoding Module (SEM).** To obtain representative topology information among different human joints, we use the GCN block [11, 44, 67] as a basic component for our SEM. Each SEM is comprised of two basic GCN blocks, where the input tokens \( X_{\text{IN}} \in \mathbb{R}^{N \times J \times d} \) are fed to the GCN blocks sequentially to produce tokens \( X_{\text{LAT}} \in \mathbb{R}^{N \times J \times d} \) for the following TEM.

2. **Temporal Encoding Module (TEM).** The TEM performs includes a Multi-Head Self-Attention (MHSA) layer followed by a 2-layer MLP. Importantly, the MHSA is conducted between “frame-level” tokens in order to efficiently encode temporal information over a long sequence of frames. Specifically, we first reshape \( X_{\text{LAT}} \in \mathbb{R}^{N \times J \times d} \) as \( X_{\text{LAT}} \in \mathbb{R}^{N \times J \times d} \), where we now have \( N \) “frame-level” tokens of length \( J \times d \). The input \( X_{\text{LAT}} \) is sent into the TEM, and we encode the temporal relationships across frames, resulting in an output \( X_{\text{OUT}} \in \mathbb{R}^{N \times J \times d} \). The input \( X_{\text{LAT}} \) is then reshaped to \( X'_{\text{IN}} \in \mathbb{R}^{N \times J \times d} \) to be sent into the next module.

**UPS Decoder.** Our decoder produces the prediction of the next token in an auto-regressive manner, while aggregating information from the input tokens as well as the partial sequence of output tokens that have already been produced. Here, our decoder consists of \( L_{\text{decoder}} \) basic transformer blocks [19]. At inference, our decoder takes in the tokens corresponding to the input and sequentially produce unified
output tokens until the EOT token is encountered. Details on training and testing are elaborated in Sec. 3.5.

3.4. Routing Mechanism

Our proposed UPS architecture can tackle several popular skeleton-based tasks (e.g., action recognition, 3D pose estimation and early action prediction) with a unified model and output format. However, the various tasks described can require very different types of knowledge, and simultaneously tackling these tasks altogether can be challenging. In particular, using the exact same set of parameters to tackle multiple different tasks can lead to destructive interference \([60, 90]\), and a lowered performance. Thus, we further extend our UPS encoder with a dynamic routing mechanism. Our dynamic routing mechanism allows tasks to either share blocks of parameters or use separate sets of parameters, depending on which one is more beneficial for performance. This encourages knowledge sharing, while mitigates the destructive interference issue.

Firstly, we introduce \(H\) parallel blocks in each layer of the UPS encoder, where each of the \(H\) blocks in each layer has the same architecture. Thus, our UPS encoder will consist of a stack of \(L\) layers consisting of \(H\) blocks each. Furthermore, we introduce block embeddings \(B_{l,h} \in \mathbb{R}^q\) for the \(h\)-th block in the \(l\)-th layer \(\theta_{l,h}\). Then, during forward inference, a Block Selection step is conducted to select the most suitable block.

In the Block Selection step, we make use of our task embedding \(\tau\) that is defined for each task, and use them to selectively activate blocks at each layer to perform our dynamic routing. We note that such a design is unlike previous works \([97]\) that use their task embeddings as input tokens that are directly processed along with \(P_{1:N}\) in SEM and TEM.

We compute the dot product between the task embedding and the block embeddings to calculate scores to determine which block to activate. Specifically, given a task embedding \(\tau\), we perform the following steps at a layer \(l\) to select the most suitable block among \(\{\theta_{l,h}\}_{h=1}^H\), as follows:

\[
\begin{align*}
s_{l,h} &= B_{l,h} \cdot \tau, \quad h \in [1, H] \\
m_l &= \text{GumbelSoftmax}\{s_{l,h}\}_{h=1}^H
\end{align*}
\]

where the selected block for the \(l\)-th layer is \(\theta_{l,m_l}\). In other words, the block \(\theta_{l,m_l}\) is chosen because its block embedding \(B_{l,m_l}\) is the closest (and hence most suitable) to the task embedding \(\tau \in \mathbb{R}^q\). Importantly, this mechanism leads to the samples of a task sharing the same route, while different tasks can potentially share or not share the same route, depending on the learned task and block embeddings. Note that, as the Argmax operation is non-differentiable, we use the Gumbel-Softmax operation \([30]\) so that the entire model can be trained in an end-to-end manner.

![Figure 3. Task embeddings \(\tau\) are learned to dynamically select the optimal blocks to use during training. At each \(l\)-th layer of the encoder, there are \(H\) blocks \(\{\theta_{l,h}\}_{h=1}^H\) to choose from (indicated in beige and blue), with corresponding block embeddings \(\{B_{l,h}\}_{h=1}^H\) (indicated in green). To select the most suitable block, we compute the dot products between task embedding \(\tau\) and block embeddings \(\{B_{l,h}\}_{h=1}^H\) and send them into the Gumbel-Softmax operator (indicated by \(\mathbb{G}\)). By optimizing the embeddings \(\tau\) and \(\{B_{l,h}\}_{h=1}^H\) during training, our dynamic routing mechanism can alleviate the issue of destructive interference and improve the sharing of knowledge.

3.5. Training and Testing

**Training Loss.** We generate our ground truth sequences using the techniques described in Sec. 3.2. We also use the negative log-likelihood loss to optimize our sequence prediction capabilities, following previous works \([8, 14]\) on language modeling. Specifically, the loss is formulated as:

\[
L = -\frac{1}{K} \sum_{k=1}^{K} \log P_\phi(y_k | y_{<k}, X)
\]  

where \(K\) is the length of the ground truth sequence, \(\phi\) refers to all trainable parameters (i.e., \(\phi = \{\theta, B, \tau\}\)), \(X\) is the input pose sequence, \(y_k\) refers to the \(k\)-th output token and \(y_{<k}\) refers to all output tokens before the \(k\)-th output token. Intuitively, we are training our model \(\phi\) to accurately predict the next token, with only access to the inputs and previously predicted tokens. We note that all tasks can be optimized through this loss.

**UPS Training.** Here, we rotate the training among the four different tasks with every iteration. Specifically, we cycle through the tasks in this manner: 3D pose estimation \(\rightarrow\) 2D action recognition \(\rightarrow\) 3D early action prediction \(\rightarrow\) 3D action recognition. We also train a corresponding task encoder for each task.

**Inference.** After obtaining the trained UPS model, we test the single unified model on all the tasks. For each task, we perform model inference according to the optimized routes learned by the corresponding task token.
4. Implementation Details

We leverage RoBERTa$_{\text{Base}}$ [47] as the pre-trained language model to extract word embeddings from text prompts. For the UPS decoder vocabulary, it holds action tokens, joint coordinate tokens, and EOT tokens, and thus it has a size of $(n_{cls} + 3 \cdot n_{bins} + 3)$. To obtain representative features from the human topology, a GCN-based [11,88] tokenizer is utilized. For UPS encoder and decoder, both of them consist of 3 stacked SEM-TEM blocks, i.e., $L_{\text{encoder}} = L_{\text{decoder}} = 3$. In each layer of the UPS encoder, by setting $H = 2$, our dynamic design has 2 parallel SEM-TEM blocks. For all block embeddings and task embeddings, we set $q = 256$, and randomly initialize them. All experiments are conducted on 8 Nvidia V100 GPUs, and the batch size is set as 1,024. We use AdamW [49] optimizer with weight decay of $5 \cdot e^{-4}$. The initial learning rate is set to $1 \cdot e^{-2}$ and gradually decays to 0.

5. Experiments

We conduct experiments on four tasks with the same unified model. The tasks are: 3D action recognition, 2D action recognition, 3D pose estimation, and 3D early action prediction. We experiment on NTU RGB+D 60 (NTU60) [64] and NTU RGB+D 120 (NTU120) [42] datasets for 3D action recognition and 3D early action prediction, Kinetics 400 [33] dataset for 2D action recognition and 3D early action prediction, and Human3.6M [28] dataset for 3D pose estimation.

We conduct experiments on the following variants: (1) UPS$_{\text{separate}}$, which is optimized separately on each task. (2) UPS, which represents our full model; it is trained on all tasks at the same time and then fine-tuned on each task based on our task embedding. When we train on all tasks at the same time, our dynamic routing mechanism is leveraged in each layer to encourage different tasks to either share common knowledge by selecting same set of parameters, or to mitigate the destructive interference issue by using separate sets of parameters.

5.1. 3D Action Recognition

In 3D action recognition, we are given a 3D pose sequence, and want to predict its action class.

**Dataset.** NTU RGB+D 60 [64] is a large dataset that has been widely used for 3D action recognition. It consists of about 56k RGB+D sequences from 60 activity classes. NTU RGB+D 120 [42] is an extension of [64], and is currently the largest dataset for 3D action analysis. It is a challenging dataset that contains more than 114k pose sequences across 120 activity classes. We follow the standard evaluation protocol of previous works [42, 64] to evaluate the Cross-Subject (xsub) and Cross-View (xview) protocols for NTU60, and the Cross-Subject (xsub) and Cross-Setup (xset) protocols for NTU120.

<table>
<thead>
<tr>
<th>Methods</th>
<th>NTU60 Top-1 (%)</th>
<th>NTU120 Top-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-GCN [88]</td>
<td>37.2</td>
<td>58.7</td>
</tr>
<tr>
<td>2s-AGCN [67]</td>
<td>37.8</td>
<td>61.0</td>
</tr>
<tr>
<td>DSTA-Net [68]</td>
<td>36.9</td>
<td>59.6</td>
</tr>
<tr>
<td>CTR-GCN [11]</td>
<td>37.1</td>
<td>60.1</td>
</tr>
<tr>
<td>MS-AAGCN [69]</td>
<td>37.8</td>
<td>61.0</td>
</tr>
<tr>
<td>SYbio-GNN [37]</td>
<td>37.2</td>
<td>58.0</td>
</tr>
</tbody>
</table>

| UPS$_{\text{separate}}$ | 36.2 | 59.4 |
|UPS | 40.5 | 63.3 |

**Results.** Following previous works [13, 42, 48], we employ the Top-1 classification accuracy metric. As shown in Tab. 1, our UPS model achieves good performance that is comparable to the state-of-the-art, demonstrating the efficacy of our method for 3D action recognition. Across all evaluation protocols on NTU60 and NTU120, we observe that by sharing one single model, UPS achieves better performances compared to UPS$_{\text{separate}}$. This demonstrates the efficacy of our method in incorporating diverse tasks into one model.

5.2. 2D Action Recognition

The 2D skeleton action recognition task is where we predict the action class of a 2D pose sequence.

**Dataset.** Kinetics 400 [33] is a widely used dataset that contains 400 action classes. It consists of more than 306k video clips. Following previous works [88], we extract 2D pose sequences using the OpenPose [6] toolbox. We follow the train-test split of previous works [37, 67, 88].

<table>
<thead>
<tr>
<th>Methods</th>
<th>Top-1 (%)</th>
<th>Top-5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-GCN [88]</td>
<td>30.7</td>
<td>52.8</td>
</tr>
<tr>
<td>CTR-GCN [11]</td>
<td>37.1</td>
<td>60.1</td>
</tr>
<tr>
<td>MS-AAGCN [69]</td>
<td>37.8</td>
<td>61.0</td>
</tr>
<tr>
<td>SYbio-GNN [37]</td>
<td>37.2</td>
<td>58.1</td>
</tr>
</tbody>
</table>

| UPS$_{\text{separate}}$ | 36.2 | 59.4 |
|UPS | 40.5 | 63.3 |

**Results.** Following previous works [88], we employ the Top-1 and Top-5 accuracy metrics. Note that, to unify the input formats for our joint tokenizer, for 2D skeletons we manually pad the third axis (Z-dim) with all zeros. As shown in Tab. 2, even when trained separately, our
UPS-separate obtains good recognition accuracy. For instance, as compared to Sybio-GNN [37], our basic UPS-separate achieves a 1.3% higher Top-5 recognition accuracy. By unifying all tasks together, our full unified UPS model outperforms UPS-separate by a large margin, and achieves state-of-the-art performance compared to all previous approaches on both Top-1 and Top-5 accuracy. This suggests that by integrating different heterogeneous tasks together, our UPS model can take advantage of each task to further benefit the learning process for 2D skeleton-based action recognition.

### 5.3. 3D Pose Estimation

In 3D pose estimation, we take 2D pose sequences as inputs and predict the corresponding 3D joint coordinates.

**Dataset.** Human3.6M [28] contains 3.6 million human poses, and is one of the largest motion capture datasets. In this dataset, skeleton data of the subjects performing various activities are captured via motion capture. We follow the train-test split of prior works [5, 54, 65], and use 5 subjects for training and 2 subjects for testing. Following previous works [5, 46, 54, 65, 96], we use CPN [10] to extract 2D keypoints, and train our model on the detected 2D pose inputs and predict the corresponding 3D joint coordinates.

#### 5.3.1. 3D Pose Estimation

In 3D pose estimation, we evaluate our UPS using xsub protocol.

### 5.4. Early Action Prediction

In early action prediction, our goal is to correctly predict action positions and the ground truth positions. P-MPJPE computes the predicted results after the predicted 3D poses are aligned to the ground truth via a rigid transformation. As shown in Tab. 3 and Tab. 4, our UPS model achieves good performance on both metrics, and obtains lower error as compared to the separately-trained UPS-separate, demonstrating the efficacy of our method for 3D pose estimation.

#### 5.4.1. Early Action Prediction

In early action prediction, we follow the previous work [23] and evaluate our UPS using xsub protocol.
RGB+D 120 [42] datasets. We follow the existing works [23, 35, 39] and evaluate our UPS on xsub protocol.

**Results.** We evaluate the prediction performance of our UPS when only 20%, 40% and 60% of frames are observed, and the results are shown in Tab. 5. Our UPS achieves promising prediction results compared to other state-of-the-art approaches. It is noteworthy that all the existing approaches are specifically designed for early action prediction task while our UPS is able to tackle different tasks at the same time.

### 6. Ablation Study

To show the efficacy of our proposed UPS, we conduct extensive ablation experiments on three tasks, i.e., 3D action recognition, 3D early action prediction and 3D pose estimation. For 3D action recognition and 3D early action prediction, we conduct experiments on the xsub protocol of NTU RGB+D 120 dataset and report Top-1 accuracy. As for 3D pose estimation, the ablation experiments are conducted on the Human 3.6M dataset (using CPN to extract 2D pose) and we report the MPIPE metric.

**Impact of depth of UPS encoder.** We conduct ablation studies on the impact of UPS encoder’s depth ($L_{encoder}$) in Tab. 7. We find that, when we increase the depth above 3, the performance does not improve further for all three tasks.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Action Recognition↑</th>
<th>Pose Estimation↓</th>
<th>Early Action Prediction↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.0</td>
<td>42.0</td>
<td>52.05</td>
</tr>
<tr>
<td>3</td>
<td>89.3</td>
<td>40.8</td>
<td>53.25</td>
</tr>
<tr>
<td>5</td>
<td>89.2</td>
<td>40.8</td>
<td>53.21</td>
</tr>
</tbody>
</table>

**Impact of number of blocks in a layer ($H$).** In Tab. 8 we ablate the decision regarding our setting of $H$. We find that, when we increase the $H$ above 2, to 5, the results do not improve further. We also note that setting $H = 2$ provides a significantly improved performance from $H = 1$, which is equivalent to not having dynamic routing.

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Action Recognition↑</th>
<th>Pose Estimation↓</th>
<th>Early Action Prediction↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.6</td>
<td>41.7</td>
<td>51.80</td>
</tr>
<tr>
<td>2</td>
<td>89.3</td>
<td>40.8</td>
<td>53.25</td>
</tr>
<tr>
<td>5</td>
<td>89.1</td>
<td>40.9</td>
<td>53.24</td>
</tr>
</tbody>
</table>

**Impact of Routing Mechanism.** We evaluate the efficacy of our proposed routing mechanism qualitatively and quantitatively. The qualitative visualization is shown in Fig. 4, which show that our tasks can share blocks or use separate blocks in each layer. As we can see, different tasks tend to share different blocks in different layers. The pose estimation and action prediction tasks tend to share the same block in the earlier two layers. However, for the last two layers, the action recognition and action prediction tasks tend to share parameters. This is possibly because the action recognition and action prediction tasks both require action classes as outputs. We also quantitatively evaluate the impact of the dynamic routing mechanism in Tab. 9. We find that our UPS consistently outperforms the variant without dynamic routing (UPS w/o DR), which demonstrates the efficacy of the dynamic routing mechanism.

### 7. Conclusion

In this paper, we propose a novel Unified Pose Sequence Modeling approach to unify three different pose-based human behavior understanding tasks, which are action recognition, 3D pose estimation and early action prediction. We leverage sequence modeling and text prompts to unify text-based activity categories and coordinate-based human joints into one single model. We also propose a dynamic routing mechanism to encourage different tasks to share common knowledge and avoid unwanted interference by adaptively sharing different subsets of parameters.

**Acknowledgments**

This work is supported by MOE AcRF Tier 2 (Proposal ID: T2EP20222-0035), National Research Foundation Singapore under its AI Singapore Programme (AISG-100E-2020-065), and SUTD SKI Project (SKI 2021_02_06). This work is also supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

---

Table 7. Evaluation of depth of UPS encoder.

<table>
<thead>
<tr>
<th>Depth</th>
<th>Action Recognition↑</th>
<th>Pose Estimation↓</th>
<th>Early Action Prediction↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.0</td>
<td>42.0</td>
<td>52.05</td>
</tr>
<tr>
<td>3</td>
<td>89.3</td>
<td>40.8</td>
<td>53.25</td>
</tr>
<tr>
<td>5</td>
<td>89.2</td>
<td>40.8</td>
<td>53.21</td>
</tr>
</tbody>
</table>

Table 8. Evaluation of number of blocks in each layer.

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Action Recognition↑</th>
<th>Pose Estimation↓</th>
<th>Early Action Prediction↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.6</td>
<td>41.7</td>
<td>51.80</td>
</tr>
<tr>
<td>2</td>
<td>89.3</td>
<td>40.8</td>
<td>53.25</td>
</tr>
<tr>
<td>5</td>
<td>89.1</td>
<td>40.9</td>
<td>53.24</td>
</tr>
</tbody>
</table>

Table 9. Evaluation of dynamic routing (DR) mechanism.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Action Recognition↑</th>
<th>Pose Estimation↓</th>
<th>Early Action Prediction↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPS w/o DR</td>
<td>87.0</td>
<td>42.8</td>
<td>53.02</td>
</tr>
<tr>
<td>UPS</td>
<td>89.3</td>
<td>40.8</td>
<td>53.25</td>
</tr>
</tbody>
</table>

Figure 4. Qualitative visualization of block selection. Overall, there are three layers in the UPS encoder, and each layer has two blocks. Here, for each task, we indicate the selected blocks in each layer with a red box. We can see that the action recognition and the action prediction tend to share blocks in the later layers, i.e., the last two layers, while the pose estimation and the action prediction tend to share blocks in the earlier two layers.
References


[82] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9878–9888, 2021. 2

[83] Li Xu, Mark He Huang, Xindi Shang, Zehuan Yuan, Ying Sun, and Jun Liu. Meta compositional referring expression segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023. 2


[85] Tao Xu, Pengchuan Zhang, Qiyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attnetg: Fine-grained text to image generation with attentional generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1316–1324, 2018. 2


