华南理工大学硕士学位论文

基于渐进式策略的人体运动姿态预测算法

马铁铮

指导教师：聂勇伟 副教授

华南理工大学

2023 年 5 月 20 日

摘　要

3D 人体运动估计（3D Human motion prediction）指：在 3D 空间中，根据历史人体运动姿态序列，预测未来的人体姿态运动序列。随着人工智能化浪潮的到来，该技术被广泛应用于自动驾驶、监控视频异常检测、人体动作捕捉生成等领域中，有着良好的应用前景和研究价值。例如在自动驾驶算法中，需要根据行人当前运动轨迹来预测其未来运动趋势，进而指导自动驾驶程序做出相应处置。

本文提出了一种新颖且高效的 3D 人体运动估计算法，与现有方法相比，本方法在预测精确度和运行效率的综合指标上有较大的领先。具体地，我们在分析总结现有方法优劣的基础上提出了两个改进策略：（1）3D 空间中的人体运动存在高度的复杂性和不确定性，给预测过程带来了很大的难度。现有方法往往采用单个网络直接预测。在运动模式简单，周期性的样本上能达到较高的预测精度。但在处理较为复杂且无明显规律的动作类型时，往往出现模式坍塌、预测失准等情况。为此我们提出了一种渐进式的网络结构来降低整体的预测难度：网络由多个阶段构成，每个阶段不再直接预测最终结果，而是在上一阶段的基础上完善预测结果。浅层的阶段负责预测运动的大概趋势，深层阶段则在此基础上完善预测细节，使预测结果向真值逐渐靠拢，同时减少各个阶段的预测难度。（2）3D 人体运动数据同时具有时间和空间两维度，且人体姿态为无向不规则图结构，其内涵的空间先验结构信息极为重要。所以，网络中，特征提取模块的时空信息提取能力与网络性能密切相关，现有方法大多使用 CNN（卷积神经网络）、RNN（循环神经网络）、传统 GCN（图卷积网络），该类方法只适用于处理 2 维空间数据，难以捕捉时序联系。为此，我们提出了一种具有时空信息捕捉能力的 GCN 模块，该模块由空间和时间两部分构成，分别称为 $SD-GCN(Spatial\ Dense\ Graph\ Convolution)$ 和 $TD-GCN(Temporal\ Dense\ Graph\ Convolution)$, 两部分串行组合，当运动序列输入后，首先由 $SD-GCN$ 提取空间信息，随后送入 $TD-GCN$ 提取时间信息，由此网络间接地捕捉了时空信息，并具有全局感受野。

在渐进式结构和 $SD-GCN$、$TD-GCN$ 这两点改进措施的帮助下，本方法在 $Human3.6M$、$CMU-MoCap$、$3DPW$ 这三个公开数据集上使用公开度量指标，预测精度较现有方法均有较大提升，且运行效率无显著落后。

关键词： 3D 人体运动估计、渐进式策略、时空序列、图卷积网络

Abstract

Keywords: 3D Human Motion Prediction、Progressive Learning、Spatial Temporal Sequence、Graph Convolutional Networks

目　录

插图目录

表格目录

第一章　绪论

1.1　研究背景和意义

近年来随着人工智能技术和社会经济的高速发展，大量信息化、智能化的新技术渗透到了人们的大众生活中。其中理解和预测人体运动相关研究获得了显著的进展。该技术被广泛应用与自动驾驶、智能机器人、人机交互和多媒体领域。在自动驾驶领域，车载计算机需要预测其他交通参与成员的行动意向和未来位置，并以此来规划车辆未来运行路线。在智能机器人领域，特别是用于协助人类的机器人，如工业机器人、看护机器人等，需要准确地预测人的未来运动来采取对应行动。在人机交互领域，在人口稠密的空间中，机器应准确预测周围的人的动作以安全地穿过人群。在多媒体领域，特别是游戏和影视制作场景中，和通过昂贵的动作捕捉设备获得人体运动模型相比，基于软件的理解和预测人体运动方法更加廉价高效。综上，理解和预测人体运动算法在促进国民经济发展和数字化、智能化转型方面有较高的研究价值。

目前学术界和工业界对该课题进行了较为细致且充分的研究。人体运动预测问题被定义为：在某个三维场景下，已知某个个体的一段历史运动序列，需要根据该段历史运动中包含的趋势或规律，预测该个体在未来的运动序列。该问题的研究重点包含两部分，第一是通过对历史运动序列的理解，提取其中包含的运动信息。例如在观看任意一段运动序列后，人类可以轻易地判别出该序列的运动类型（如行走、拾取物品、舞蹈等）。但对计算机来说，如何理解运动序列中的时序信息是研究的重点。第二是基于对历史运动序列信息的提取，预测未来运动序列。由于人体运动的高度复杂性和不确定性，如何基于有限的运动信息尽可能降低预测过程的不确定性，从而输出准确的未来运动序列，是当前研究的一个主要难点。

在早期的研究中，由于循环神经网络（RNN[1]）可以利用其内部隐状态（Hidden State）来捕捉输入数据的时间依赖性，对于处理时间序列这类连续数据特别有效，所以RNN 被用来提取人体运动序列中的时序信息，预测未来的运动序列。

这类方法的主要思想是，每个 RNN Unit 有一个隐状态，可以在每个时间步骤中根据当前的输入的人体运动姿态和以前的隐状态进行更新。这个隐状态是对网络过去所见信息的总结，并被用来对未来进行预测。通过使用以前的隐状态来计算当前的隐状态，RNN 可以捕获输入序列的时间依赖性。这使得 RNN 可以提取关于序列结构和序列元素之间的依赖关系的信息，这对于时间序列预测任务特别有利。

然而，由于 RNN 中每一步输出只与当前输入和上一步隐状态有关，无法对每一步输出进行整体约束。这导致输入序列和预测序列的过渡部分出现不连续的情况。为此现有方法提出了一种有着编码器-解码器结构的序列到序列模型（Sequence-to-Sequence），编码器将输入数据整体映射到隐空间，随后由解码器一次性预测未来运动序列，由此可以对输出进行全局一致性约束。此外，隐状态容量有限，RNN 只能对短期依赖性进行建模，无法处理长距离时序依赖。这导致网络无法完全提取输入序列中的时序信息。为了解决这个问题，出现了长短期记忆（LSTM[2]）和门控循环单元（GRU[3]）网络，但它们更加复杂和计算量更大。最后，该类方法通常将一个人体姿态作为一个整体输入 RNN Unit，忽略了人体姿态的空间结构。然后，对于人体运动序列预测问题，人体姿态的空间结构是一个重要的先验信息。这导致基于 RNN 的方法在预测结果真实性和准确性方面有所欠缺。

近年来随着对图卷积网络（GCN[4]）的深入研究，部分现有方法引入图卷积网络对人体姿态空间结构进行建模。对于人体姿态这类不规则图状数据，图卷积网络有着天然的优势。在这类方法中，人体姿态被视作由一组顶点（或节点）和连接一对顶点的一组边组成的数学结构。通过图卷积网络对复杂的关节点对之间的联系进行建模。但传统图卷积网络主要应用于空间维度，如何设计高效的，具有时空信息提取能力的图卷积网络，对于人体运动序列预测这类涉及到时空序列数据的问题尤为重要，至今也依旧是学术界的一个难点问题。

除了需要尽可能提取输入序列中的时空信息，如何降低预测过程中的不确定性也是需要考虑的问题。在大多数情况下，由于人体运动序列的复杂性，输入序列和未来序列之间存在较大的差异，这导致预测过程存在较大的不确定性和预测歧义。现有方法大多采用单个前馈神经网络，直接接受输入序列，并预测未来运动序列。网络在预测过程中将承受较大的不确定性，预测结果可能出现模式坍塌（Mode collapse）问题。因此，如何设计更高效的预测策略来降低预测过程中的不确定性和歧义，是当前研究中需要重视的问题。

1.2 主要研究内容及贡献

针对上述研究存在的问题和人体运动姿态预测问题的特点，本文主要的研究内容和贡献被总结如下：

1. 基于渐进式策略的人体运动姿态预测算法框架。

- 与现有方法使用单阶段的网络结构不同，我们由浅至深地将预测过程拆分为多个阶段，除开位于网络入口的阶段，其他阶段均在上一步预测基础上进行预测，这将有利于降低每一阶段的预测难度。
- 我们遵循网络由浅至深，预测难度由易到难的原则。浅层阶段只负责预测大致的运动趋势，复杂的运动细节预测则由具有深层语义提取能力的深层阶段负责。
- 为了构建多阶段、渐进式的网络，我们为每个阶段构造对应的中级监督目标（Intermediate target）。具体的，我们设计了一种人体运动轨迹平滑方法，通过平滑关节点运动轨迹的方式，逐步去除运动细节，为每个阶段由深至浅提供不同平滑程度的预测目标。

2. 具有时空信息提取能力的 Spatial-temproal 图卷积网络模块（$SD\text{-}GCN$ 和 $TD\text{-}GCN$）。

- 我们提出了一种新颖的具有时空信息提取能力的图卷积模块，该模块由时间信息提取模块和空间信息提取模块两个独立的图卷积构成。
- 时间信息提取模块称为 $TD-GCN$，输入数据被视为多个关节点轨迹，$TD-GCN$ 提取时间维度上的信息。空间信息提取模块称为 $SD-GCN$，输入数据被视为多个人体姿态，$SD-GCN$ 提取空间维度上的信息。两个模式以串联的方式构成一个时空图卷积模块，当数据依次通过二者时，网络间接地提取到了时空信息。
- 由于时间和空间信息提取模块相互独立，因此随着输入数据的时间长度和空间复杂度提高，模型空间复杂度只随线性增长而非倍数增长，在保证模型信息提取能力的同时，降低了时间效率。

3. 我们在三个公开数据集上使用通用度量指标，与现有先进方法进行对比。在预测精确性方面大幅领先（Human3.6M 6%-7% , CMU-MoCap 5%-10%, and 3DPW 13%-16%）。并且在时间效率和内存占用指标上我们也处于靠前位置。

1.3 论文结构

本文包括七个主要章节，包含绪论、相关工作、基础知识、渐进式策略算法框架、Spatial-temproal 图卷积网络模块、对这两个模块的进行的实验分析，以及对全文进行总结与展望。其中各个章节的主要内容安排如下：

首先，在第一章，对 3D 人体运动估计问题的研究背景和研究意义进行详细阐述。

其次概述本文的主要研究内容和贡献。

第二章详细介绍了 3D 人体运动估计问题的研究现状和发展历程，对当前研究的参考价值。

第三章主要介绍本文所提出方法中使用到的相关技术和理论基础，为详细叙述新方案打下基础。

第四章提出了渐进式策略算法框架，本文将从实验和直观分析的角度来叙述该设计的合理性和有效性，并且我们还展示了该渐进式策略算法框架具有高度的灵活性，可以与任意网络模块进行组合以适应不同的任务种类。

第五章提出了 Spatial-temproal 图卷积网络模块，在这里我们将详细介绍该模块中的时间和空间部分，以及它们是如何整合为一个整体。

第六章对我们提出的基于渐进式策略的人体运动姿态预测算法在三个公开数据集（Human3.6M、CMU-MoCap、3DPW）上与现有先进方法进行定性和定量的对比。同时，通过消融实验对模型中的各个模块进行定量分析。另外对模型中的一些有趣的细节进行了充分讨论。

最后一章是对本文进行总结与展望，对本文提出的基于渐进式策略的人体运动姿态预测算法进行概略性的总结。同时分析本方法当前的不足，对后续研究提供指导性意见。

第二章　相关工作

近十年，3D 人体运动姿态预测算法受到了广泛的研究和探讨，涌现了一大批出色的工作。根据其对人体运动序列的建模方式不同，现有方法可以分为以下几类：基于循环神经网络的方法 [5-18]、基于卷积神经网络 (包含 CNN 与 GCN) 的方法 [19-35]、基于对抗生成网络的方法 [9,36-43]。在研究早期，由于人体运动的序列化特征，大部分方法使用循环神经网络对输入数据进行建模，然而循环神经网络的时序记忆能力受限于隐变量的大小，只能处理短期记忆，无法处理较长时间的序列。随后出现了一批由卷积神经网络构成的模型，其中包含 CNN 和 GCN 网络，前者与 RNN 相比拥有更大的感受野，这提高了网络的长时序依赖捕捉能力。而 GCN 则更适合处理人体姿态这类不规则的空间数据，能够感知人体结构先验信息。对抗生成网络近些年也被引入该领域，对抗生成的策略能够提供在真实性和多样性方面占优的结果，但网络的训练和最终结果的评估任然有待研究。另外随着近些年 Transformer 在计算机视觉领域的兴起，部分方法希望凭借其全局感受野的特性来捕捉全局的时序依赖。接下来本文将详细介绍以上四类方法中具有代表性的模型。

2.1　基于循环神经网络

人体运动序列预测问题通常被视为 $seq2seq$ 预测任务。RNN 因其在此类任务中的出色表现而得到广泛认可，这启发了许多研究人员利用基于 RNN 的方法来研究人类运动序列预测任务。EDR [5] 率先将 RNN 引入人体运动序列预测领域，其结构如图2-1所示。其中 x_t 代表第 t 个时刻的输入的人体姿态，而 y_t 则代表由 x_t 预测出的未来人体姿

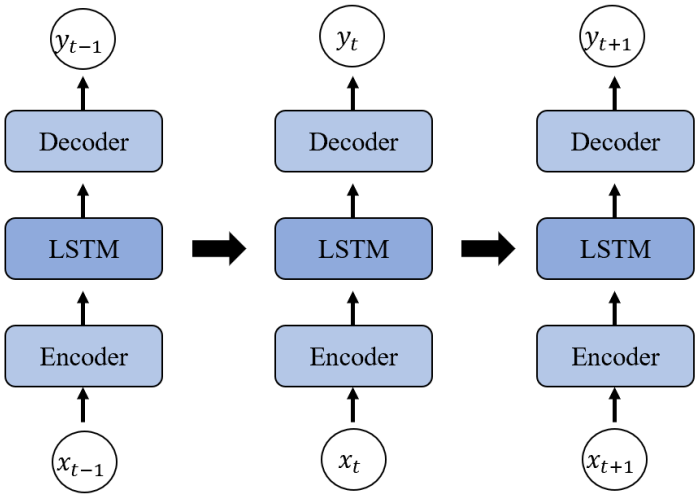

图 2-1 EDR 网络结构

态。网络接受 x 作为每个 RNN 节点的输入，首先输入姿态通过编码器（Encoder）编码到隐空间，随后送入 RNN 层，将时序信息提取并传递给下一个节点。同时通过解码器（Decoder）解码出对应的未来人体姿态作为当前节点的输出。该方法很好地利用了 RNN 的时序数据建模能力，有效提取了输入人体运动序列中的时序信息。但由于当前 RNN 节点是在上一个节点的基础上进行预测，因此容易出现误差累积问题。此外，由于在 EDR 中，未来运动序列被逐时刻、独立地预测，因此在输入序列和预测序列的过渡部分容易出现不连续的现象。Res. Sup.[8] 针对 EDR 中的问题提出了改进措施，如

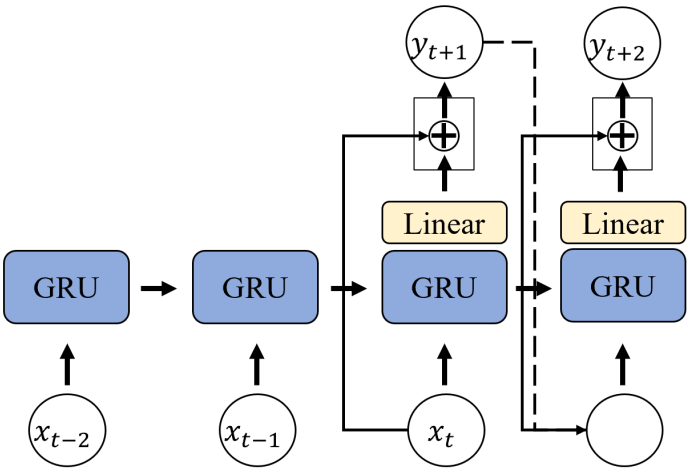

图 2-2 Res. Sup. 网络结构

图2-2所示，Res. Sup. 引入了在自然语言处理领域常用的 *seq2seq* 模型结构，与 EDR 相比，*seq2seq* 统一将输入序列编码到隐空间，此时隐空间包含所有的输入信息，这有助于模型从全局的角度考虑，而不是主要依靠当前时刻的输入。随后通过解码器结构，将隐空间中的信息解码为未来人体运动，当前时刻的输出将作为下一时刻的输入，这有助于保证时序上的连续性，也允许相邻时刻的运动通过残差连接的方式完成一致性约束。除网路结构外，另一些方法从人体运动学入手，通过分析人体运动模式来针对性地设计网络，例如 Tang *et al.*[10] 发现在人体运动中，并非所有关节点都处于运动状态。相反只有处于肢体末端的关节点位置才会较为频繁地改变。因此，他们提出了针对人体运动模型中频繁运动的关节点的方法，称为 HUM。具体的，他们设计了一个新颖的门控单元用来过滤运动幅度小的关节点。此外，注意力机制被用来关注具体的运动模式。AHMR[44] 为了捕获更多的长期相关性，在 RNN 单元中，可以同时对相邻关节和帧进行编码。此外，它不仅可以同时对本地和全局上下文进行建模，而且还使用了一个注意力模块来帮助更新全局上下文。

虽然上述方法在 EDR 的基础上提出了改进措施，提升了网络性能，但由于 RNN 网络的特性任然无法解决诸如误差累积、过渡部分不连续、训练困难和难以处理长时间依赖关系等问题，这将削弱网络预测的真实性。为此一些新的方法的将目光投向了效率更高，感受野更大的卷积神经网络。

2.2 基于卷积神经网络

人体运动序列数据包含时间和空间两个维度，而卷积神经网络（CNN）在处理空间数据上有天然优势，时序信息也可以由 1D 的 CNN（TCN[45]）进行处理，相比循环神经网络，TCN 更轻量化、推理速度更快、配合空洞卷积[46]感受野更大。在人体运动预测中，对于模型如何处理空间和时间的依赖关系是一个非常重要的问题。传统的 CNNs 只能捕捉静态图像的空间依赖性，但是在动态场景下，时间信息也是非常关键的。因此，研究人员提出了一些新的 CNN 架构，以处理人体运动预测的时空依赖关系。

在 Butepage *et al.*[33] 中，作者设计了一种新的卷积层来编码不同的时间尺度。这种卷积层可以有效地捕捉局部时间尺度的依赖关系，但是它无法处理长期的时间依赖性。为了解决这个问题，QuaterNet[47] 引入了扩张卷积，可以在网络中捕捉长期时间依赖关系。该方法在分层输入姿势的情况下表现良好，但仍然无法处理空间依赖性。

为了同时处理空间和时间的依赖性，一些研究人员采用了分层结构的 CNN Li *et al.*[34]。这种 CNN 架构利用卷积结构来捕捉长期隐藏状态，并将其送到解码器中以生成人体姿势。这种方法可以有效地处理时空依赖关系，但是它需要大量的计算资源和训练数据。为了进一步提高模型的性能，Li *et al.*[48] 提出了一种卷积分层自编码器框架，用于表示人体骨骼结构。在这种框架中，分层拓扑被用于表示骨骼结构，并且嵌入了 1D 卷积层来编码每个节点。该框架可以有效地捕捉空间和时间的依赖关系。最近，TrajectoryCNN[35] 被提出来处理人体运动预测的时空依赖关系。它引入了一种新型的轨迹空间，可以轻松地捕捉各种局部-全局和时空特征。这种框架在许多基准测试中取得了优异的性能。

虽然 CNN 能有效处理时间和空间数据，但 CNN 的规则卷积核决定它适合处理图像或视频这类规则数据。人体姿态属于不规则的无向图结构，人体关节点对应图中的顶点，骨骼对应顶点间的相互关系。这种拓扑结构是极其重要的先验空间信息，能有效辅助模型感知运动模式。而 CNN 的规则卷积核使得它很难利用这类先验信息, 因此，在最近的研究中，天然具有拓扑信息处理能力的图卷积网络（GCN）获得了越来越多的关

注。

2.3 基于图卷积网络

GCN 是一种可以处理图形结构数据的神经网络。在 GCN 中，卷积操作是基于邻居节点之间的连接进行计算的，这使得 GCN 可以有效地处理具有不规则连接的数据结构，例如人体关键点。此外，GCN 还可以利用拓扑信息来捕捉节点之间的关系，从而更好地理解图结构数据。该特性对人体运动序列数据处理非常有利。

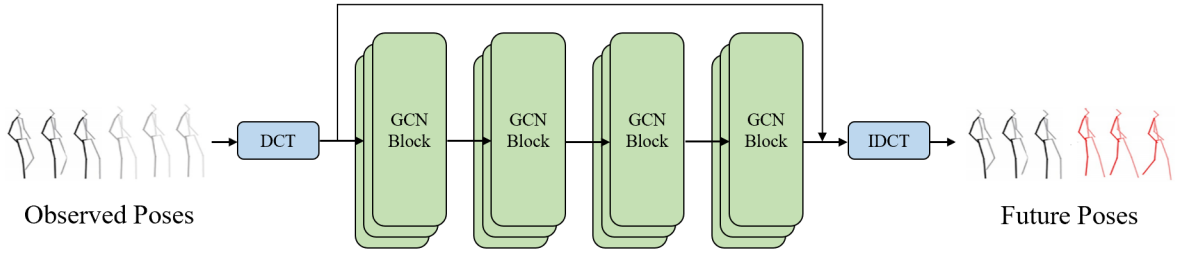

图 2-3 LTD 网络结构

LTD[20] 率先提出了一种代表性的 GCN 方法（图2-3），使用原始的 GCN 对人体运动序列进行建模。具体的，对于输入的人体运动序列，LTD 将其视作一个不规则的无向图。由于人体运动序列数据包含时间和空间两个维度，而原始的 GCN 只能处理二维平面数据。因此，LTD 将该运动序列中的关节点轨迹视作一个整体，将其放入图结构网络中。即图中的每个节点包含了某个关节点这段时间内的运动轨迹，由此 LTD 实现了使用一个描述平面节点联系的 GCN 来处理时空维度的人体运动序列。在网络结构方面，网络接受历史人体运动序列作为输入，为了保证输入数据和输出数据在时间维度上的一致性，LTD 提出用已知序列的最后一个人体姿态填充输入序列，使其与输出序列时序长度一致。此外，网络输入和输出数据之间的残差连接也得到保证，有助于提高网络的训练效率和预测精确性。完成填充步骤后，输入数据将经过离散余弦变换（DCT）从时域变换到频率域，通过过滤掉低频信息并保留高频信息，可以在降低数据维度的同时，减少噪声。随后，再被传入多个串联的 GCN 模块，将数据映射到隐空间后，提取时空信息，在填充数据的基础上预测未来运动。最后，经过离散余弦逆变换（IDCT）后，输出最终的预测结果。该方法的贡献在于，提出了一种使用原始 GCN 对时序数据进行建模的方式，在最终预测精度上大幅领先基于 RNN 的方法，通过全局的残差连接解决了输入序列和预测序列过渡部分的不连续性。但由于该方法忽略数据的时序特性，仅仅使用 GCN 提取人体姿态的空间结构信息，将关节点轨迹作为一个整体放入图节点中，这导

致该模型对时序运动的感知能力有所欠缺，未来仍然有提升空间。

用于人体姿态提取的方法 ST-GCN[49] 针对 LTD 存在的问题，提出了一种具有时空信息提取能力的 GCN。对于时空人体运动序列数据，一个直观的想法是建立一个跨越时空维度的图，囊括不同时间和空间上的关节点。但由于 GCN 复杂度随着时空维度的增加成倍数上升，这样的图结构数据的复杂度是难以接受的。因此 ST-GCN 提出将时间和空间维度的数据拆分, 分别用 TCN[45] 和 GCN 进行处理。具体的，1D 的卷积神经网络 TCN 负责提取各个关节点轨迹中的时序数据，GCN 负责处理人体姿态中的空间结构数据。通过将时空两个维度分为，ST-GCN 将网络的时间复杂度降为线性增长。并且通过实验证明网络时空信息提取能力优于现有方法。但 TCN 为局部算子，感受野被限制在卷积核范围内，导致 ST-GCN 在提取长时依赖上存在缺陷。

最近 MSR[28]，更进一步提出了空间层次化的 GCN 网络。它提出了一个类 Unet[50] 网络，编码器部分，逐渐简化人体姿态空间结构，只保留最简洁的空间信息。解码器部分，首先构造空间结构较简单的人体运动序列，随着网络的深入，人体运动序列的空间复杂程度逐渐增加，直到输出具有完整空间结构的数据。具体的 GCN 模块设计上它参考了 LTD，将关节点运动轨迹视作一个整体。该方法提出的空间层次化 Unet 网络，给网络一个渐进式的学习过程，这有利于降低网络的学习难度。但对空间结构进行简化的过程中，破坏了人体结构先验信息，导致网络预测效率相比 LTD 并没有明显提升，某些方面甚至出现了下滑。

由于 GCN 网络对图结构数据中节点关系的处理具有先天的优势，因此 GCN 能够更好地提取人体姿态数据中的结构先验信息。但现阶段的 GCN 对于时空跨维度信息的处理能力任然有待提高，它们或是忽略某一个维度来降低时间复杂度，或是在信息提取能力和时间效率上做出了妥协。因此，如何平衡模型复杂度和时空信息处理能力，将是未来的一个研究重点。

2.4 基于对抗生成网络

人体运动姿态预测算法的一个主要难点在于，预测过程中存在不确定性，这种不确定性是由于输入序列和预测目标序列之前的差异造成的。例如，如果输入序列与预测目标序列关联性强，则预测越简单，反之则越难。针对上述问题，一个解决思路是如上述方法，通过提高网络的时空信息提取能力，尽可能捕捉输入和预测序列间的关联性。另一个思路是引入生成式模型和随机性，生成更真实的运动序列。具体的，近年来由于对

抗生成网络 [51] 的深入研究，GAN 为生成人体运动姿态序列提供了更多新的可能性。

Barsoum *et al*.[36] 率先提出了一种基于 GAN 的 $seq2seq$ 人体运动序列预测方法，它使用改进版的 WGAN-GP 进行训练，与上述基于 RNN，CNN 或 GCN 的方法不同，它的网络输入表达为概率密度分布而非固定的人体运动序列。因此，在预测时可以通过为网络提供不同的随机噪声 z，来对同一个输入运动序列预测不同的未来运动序列。然而，虽然该方法在结果真实性方面有所提升，但由于输入噪声的引入，预测准确性有所下降。在此基础上，BiHMP-GAN[37] 同样通过在输入序列中添加从固定分布中采样的随机噪声来为预测过程添加随机性。不同的是，BiHMP-GAN 提出了一个双向对抗神经网络来解决预测过程中的模式坍塌问题。与此同时，受到上述工作的启发，AGED[9] 提出了一种新颖的对抗生成框架，它具有两个全局的循环鉴别器，一个鉴别器被用于促进生成序列的保真度，另一个鉴别器与网络进行联合训练，保证未来生成序列的连续性。STMI-GAN[38] 也沿用了该思路，用于处理长时依赖的人体运动序列。Adversarial Refinement Network (ARNet)[42] 设计了一种新的对抗式的误差调整策略，与上述方法不同的是。判别器不再直接判断生成结果的真实性而是用来估计预测误差，随后精修模块再根据误差调整预测结果。而 Lyu *et al*.[43] 则利用 GAN 模拟路径积分来解决随机微分方程并预测未来运动轨迹。值得注意的是，由于 GAN 的对抗训练特性，想要训练达到平衡状态是非常困难的，Cui *et al*.[41] 提出了一种新的 GAN，该 GAN 使用了 spectral 归一化，以避免模式坍塌。还有另一种称为 AMGAN[40] 的策略，它由复合 GAN 结构设计而成，包含用于不同低维身体部位的局部 GAN 和用于高维全身的全局 GAN 组成。该方法证明了降维可以有效地提高 GAN 的训练效率。

总而言之，利用 GAN 的策略主要可分为两类。（1）被用作学习算法以帮助网络生成更加真实的结果。（2）利用随机噪声向网络添加随机性，生成多样化的预测结果。而 GAN 作为一个具有明显优势和劣势的网络，也会给研究人员的工作带来一定的挑战。

2.5 基于 Transformer

近些年，Transformer 受到了学术界的广泛关注，它也从自然语言处理（NLP）领域被引入到计算机视觉领域，在诸如图片识别、图片分割等经典问题上大幅领先现有基于卷积神经网络的方法。对于人体运动序列预测问题，网络需要捕捉长时依赖关系的能力。而 Transformer 的全局感受野特点恰好可以解决该问题。因此出现了一批基于 Transformer 的方法 [52-53]。

Aksan *et al.*[52] 设计了一个包含时间和空间分支的 Transformer 网络，两个分支分别提取输入序列的空间结构信息和时序信息，最后再通过融合模块得到最后的预测结果。

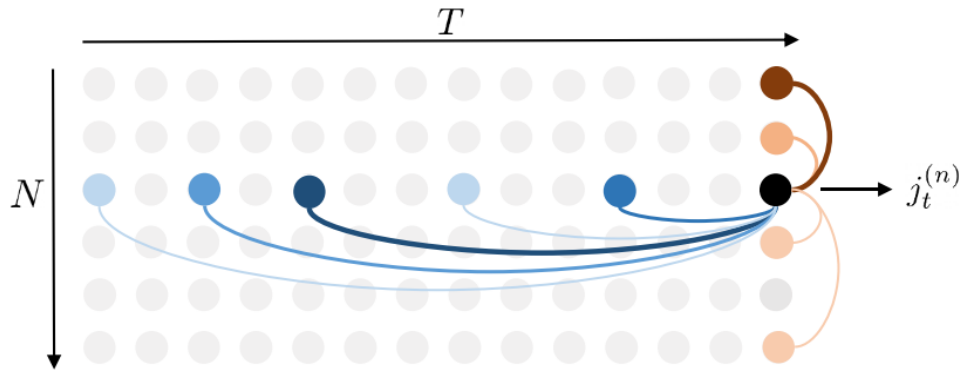

图 2-4 Spatial-temporal Transformer[52]

其中 spatial-temporal Transformer 原理如图2-4所示，$j_t^{(n)}$ 表示 t 时刻，第 n 个关节点。其中，$j_t^{(n)}$ 只和自己位于同一时间或空间的关节点进行注意力（attention）机制计算，图中颜色的深浅代表关节点之间的关联程度，颜色越深关联性越强，权值也越高，反之则越小。通过分离的时空 transformer，该方法间接地提取了时间和空间信息。

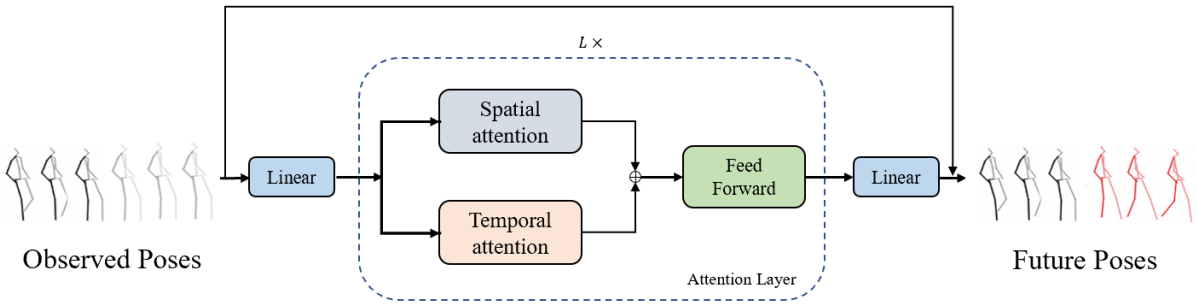

图 2-5 Spatial-temporal Transformer[52]

完整的网络结构如上图所示，网络由 L 个串联的注意力层构成，每个层包含一个空间维度和时间维度的注意力层，特征被传入注意力层后，分别送往两个分支，用于提取时间和空间信息。提取结束后，空间和时间信息相加，送入前馈神经网络进行特征融合，最终通过线性层输出预测的未来人体运动序列。该方法通过并行的方式分离时间和空间维度，减少了时间复杂度和模型参数量。但分支的方法使得时间和空间维度缺少信息通信手段，导致信息交流受阻，影响最终的模型质量。总的来说，Transformer 高效的全局注意力机制有利于模型捕捉长时序依赖，但 Transformer 的注意力计算模块也导致模型空间的上升和计算开销的增加。此外，时间和空间分支间的通信问题也是未来需要

研究的问题。

2.6 总结

本章，我们对人体运动姿态预测算法的发展做了一个简要的回顾。在初期，研究人员根据循环神经网络在处理时序数据上的优势，设计了 $seq2seq$ 的网络模型来对输入序列统一编码后预测未来运动序列。但由于循环神经网络对于时序记忆的能力依赖隐变量的大小，因此难以处理长时依赖。此外，梯度消失等训练问题也困扰着现有方法。随后，研究人员将目光转向了卷积神经网络，特别是图卷积神经网络，它对无结构不规则数据的处理有天然的优势。但如何对传统图卷积网络进行改进，使其拥有时序信息处理能力，任然是当前的研究热点。此外，GAN 机制的引入允许生成更多样化和真实的结果，但其训练过程的不稳定性和噪声对结果准确性的影响有待进一步解决。近些年，Transformer 的兴起给该问题带来了新的解决思路，其全局感受野的特性，允许其捕捉更长范围的全局依赖。但其注意力计算带来的额外计算开销和时空信息间的通信问题，还需要进一步探索。本文希望在现有基于图卷积的方法的基础上，提高模型的时空信息提取能力，并且控制模型的运行开销。

第三章　图卷积网络基础

3.1　图卷积网络简介

图卷积网络（Graph Convolutional Networks，GCN）是一种用于图像数据处理的神经网络模型。与传统的卷积神经网络（Convolutional Neural Networks，CNN）不同，GCN能够处理图像结构数据，并且能够通过学习节点之间的关系来推断节点的特征。GCN的发展是为了解决传统CNN在处理非欧几里德结构的数据时，面临的局限性和挑战性。

GCN的应用广泛，包括社交网络分析、推荐系统、化学和生物信息学等领域。例如，GCN可以用于社交网络中的节点分类任务，其中每个节点代表一个用户，而用户之间的关系可以通过社交网络的拓扑结构表示。通过学习这些节点之间的关系，GCN可以有效地预测新用户的属性，例如他们的兴趣爱好或职业等。

GCN的核心思想是利用卷积操作来对节点进行特征提取。由于图像数据的非欧几里德性质，传统的卷积操作无法直接应用于图像数据。因此，GCN引入了一种局部聚合的方式来计算每个节点的特征。具体来说，GCN利用节点的邻居信息来计算该节点的新特征，并通过非线性变换来实现特征提取。

GCN的结构可以看作是一个多层感知机（Multilayer Perceptron，MLP），其中每一层对邻居节点进行聚合，并通过激活函数进行非线性转换。每一层的输出都可以被视为该节点的新特征，这些新特征又被用于下一层的聚合。在图像分类和节点分类任务中，GCN通常通过全局平均池化来获得最终的输出。

最近，许多新的GCN变体已经被提出，例如Graph Attention Network（GAT）、GraphSAGE、ChebNet等等。这些变体使用不同的方法来聚合节点的邻居，并具有更好的性能和可扩展性。例如，GAT利用注意力机制来计算不同节点之间的权重，可以更加灵活地处理节点之间的关系。

总之，GCN是一种强大的图像处理方法，它可以有效地学习节点之间的关系，并利用这些关系来推断节点的特征。它已经被广泛应用于各种领域，成为了图像处理领域的重要工具。GCN的发展也为处理非欧几里德结构的数据提供了新的思路和方法。

3.2　图卷积模型定义

图 [4] 展示了一个标准的GCN，根据这个图，我们可以看到一个具有个 input channel 的图结构被输入，然后通过中间的隐藏层，产生了个 output channel 的输出。在这一过

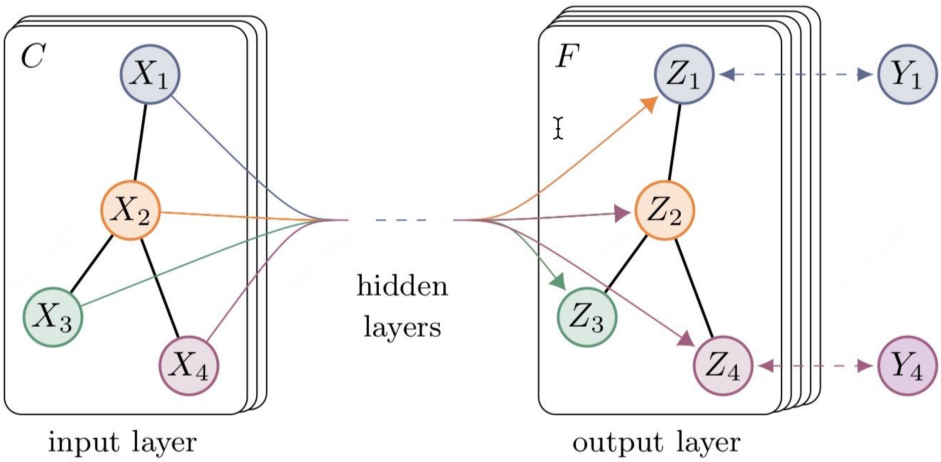

图 3-1 图卷积示意图[4]

程中，GCN 不改变图的空间结构，只是对节点特征进行变换，根据图中的信息流动方向，捕捉图的空间结构信息。我们将网络的输入 X 定义为一个 $V * F$ 的矩阵，其中 V 代表输入图中的节点数量，而 F 代表输入图中节点的特征维度。X 经过特征处理后即可得到输出 Z，一个 $V * F'$，其中 V 结构不变，F' 为变换后的特征维度。

一个图的空间结构可以用邻接矩阵描述。邻接矩阵是一种用于表示图形或网络的数据结构。它是一个方阵，其中行和列分别对应于图中的节点，矩阵中的每个元素表示两个节点之间是否存在一条边。具体来说，假设图有 V 个节点，那么邻接矩阵的大小为 $V \times V$。如果节点 i 和节点 j 之间存在一条边，则邻接矩阵中第 i 行第 j 列的元素为 1；如果它们之间不存在边，则该元素为 0。邻接矩阵还可以用于表示带权图，其中矩阵中的元素表示边的权重。如果节点 i 和节点 j 之间存在一条权重为 w 的边，则邻接矩阵中第 i 行第 j 列的元素为 w；如果它们之间不存在边，则该元素为 0 或一个特殊的表示不存在边的值（如-1）。邻接矩阵是一种简单且直观的图表示方法，适用于图比较稠密（边数接近节点数的平方）的情况。但是，对于稀疏图（边数远小于节点数的平方）来说，邻接矩阵会浪费大量的空间。此外，在一些图算法中，邻接矩阵的时间和空间复杂度可能较高，因此在应用时需要针对稀疏矩阵进行额外的优化。在定义邻接矩阵以后，可以很轻易重构一个图结构。我们用 $G = (V, E)$ 定义一个图，其中 V 代表节点，E 代表边。我们通过 $|V| \times |V|$ 得到 A，其中如果节点 i 和节点 j 之间存在联系，则 $A_{ij} = 1$ 否则则等于 0。此时即可通过一个图数据结构结构得到 A，同理也可以根据 A 得到一个图数据结构。如上图所示，我们定义了一个拥有 6 个节点的图结构数据。其中节点上的数字代表该节点的编号，节点间的连接代表两点间存在联系。根据上文，我们可以将该数据结

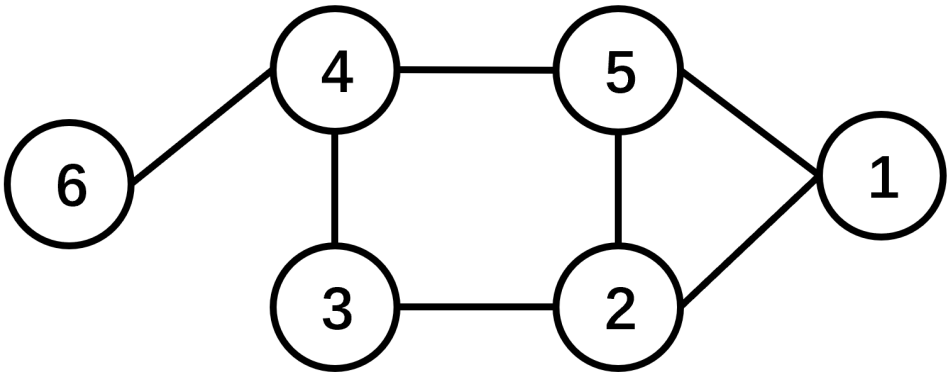

图 3-2 图结构数据示意图

构以邻接矩阵的形式表示如下。

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{3-1}$$

其中有联系的关节点则被置为 1，且因为3-2为无向图无环图，因此邻接矩阵为沿着对角线对称，对角线上的元素均为 0。例如，由于 1 号节点和 5 号节点之间存在连接，则 A_{15} 和 A_{51} 均被置为 1。所以可以将图卷积网络中的一层定义为：

$$H^{(l+1)} = f(H^{(l)}, A) \tag{3-2}$$

其中 H 每一层的隐变量，其中输入层 $H(0) = X$，输出层 $H(L) = Z$，L 代表最后一层，Z 为最终输出网络的结果。不同的 GCN 函数，使用不同的 GCN 模型 $f(\cdot,\cdot)$。被广泛传播的图卷积层定义如下：

$$f(H^{(l)}, A) = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3-3}$$

其中 D 代表图结构的度矩阵，W 为每一个卷积层的权重，该公式的物理含义为，当前层隐变量是由上一层隐变量以邻接矩阵为权重加权求和得到的。此外，邻接矩阵有一个重要性质，邻接矩阵 A 代表某个节点在一跳（沿两个节点之间的联系移动一次被称为一跳）的范围内到达邻居节点的路线数量，由于要求步数为一跳，因此到邻居节点路线数量只能是 0 或 1。而对于邻接矩阵的 n 次方，A^n。则代表某个节点在 n 跳的距离上到

达邻居节点的路线。物理意义上，一阶的邻接矩阵定义了物理相连节点间的关系，而高阶的邻接矩阵可以定义非物理相连的节点之间的关系，表述节点间的高阶语义信息。而该种特性可以通过神经网络中的 GCN 层叠加实现。接下来，我们将给出3-3的推导过程以及数学含义，从直观的角度解释图卷积网络的设计逻辑。

3.3 拉普拉斯算子

为了推导图卷积网络，我们将首先介绍拉普拉算子（Laplace Operator）作为铺垫，进而给出图卷积网络的物理定义。拉普拉斯算子被定义为欧几里得空间中，一个二阶可微函数 f 梯度的散度，即对该函数求二阶微分，结果表示为 Δf 其中 Δ 代表拉普拉斯算子。如果 f 是图上定义的一组高维向量，则 Lf 是在图空间求二阶微分，其中 L 为拉普拉斯矩阵。为了叙述方便，我们首先在连续欧式空间中，定义 $f(x,y,z)$w 为一个二阶可微的三元函数，在该函数上分别给梯度和散度的定义。

3.3.1 连续空间中的拉普拉斯算子

梯度是一个矢量，表示某一函数在沿着该点的方向导数上可以取得局部极值。即函数在该方向上沿着该方向下降速度最快，变化率最大。对于位于欧式空间中，一阶可微的函数 $f(x,y,z)$。在该函数上一点 $P(x,y,z)$，称向量：

$$\left\{\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial f}{\partial z}\right\}=\frac{\partial f}{\partial x}\vec{i}+\frac{\partial f}{\partial y}\vec{j}+\frac{\partial f}{\partial z}\vec{k} \tag{3-4}$$

为 $f(x,y,z)$ 在点 P 处的梯度，记为 $gradf(x,y,z)$ 或 $\nabla f(x,y,z)$，其中 $\nabla=\frac{\partial}{\partial x}\vec{i}+\frac{\partial}{\partial y}\vec{j}+\frac{\partial}{\partial z}\vec{k}$。

散度 $(\nabla\cdot)$ 被定义为空间中各矢量场的发散程度，某点的散度被定义为 $div(F)$，散度越高的点代表该点向周围散播能量的能力越强，而散度越低的点代表该点吸收能量的能力越强。当某点散度为 0 时，则代表该点即不发散也不吸收能量，为无源。

在了解散度和梯度在连续函数空间中的定义后，即可给出连续二阶可微函数空间中，拉普拉斯算子的具体定义。在 n 维欧几里得空间中，拉普拉斯算子被定义为梯度（∇f）的散度 $(\nabla\cdot)$。$\Delta f=\nabla^2 f=\nabla\cdot\nabla f=div(gradf)$，在笛卡尔坐标系下被表示为：

$$\begin{aligned}\Delta f&=\frac{\partial^2 f}{\partial x^2},\frac{\partial^2 f}{\partial y^2},\frac{\partial^2 f}{\partial z^2}\\ \Delta&=\sum_i\frac{\partial^2}{\partial x_i^2}\end{aligned} \tag{3-5}$$

3.3.2 离散空间中的拉普拉斯算子

在公式3-5，我们给出了在连续二阶可微函数上的拉普拉斯算子定义。然而，图卷积网络所处理的图结构数据位于离散空间中，需要进一步给出离散空间中的拉普拉斯算子。我们首先从一维离散空间中进行推导。对于位于一维离散空间中的函数 $f(x)$，设离散函数空间中的最小间隔为 h，则 $f(x)$ 在 x_i 处的一阶微分为：

$$f(x_i)' = \frac{f(x_{i+1}) - f(x_{i+1})}{2h} \tag{3-6}$$

使用同样的方法计算 $f(x)$ 在 x_i 处的二阶微分

$$f(x_i)'' = \frac{f(x_{i+1}) + f(x_{i+1}) - 2f(x_i)}{h^2} \tag{3-7}$$

在一维基础上进行推广，可以获得拉普拉斯算子在离散二维空间上的表达式，公式3-8展示了离散二维空间中的函数 $f(x,y)$ 的拉普拉斯算子表达式。

$$\begin{aligned}\Delta f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= f(x+1,y) + f(x-1,y) - 2f(x,y) + \\ &\quad f(x,y+1) + f(x,y-1) + 2f(x,y) \\ &= f(x+1,y) + f(x-1,y) + \\ &\quad f(x,y+1) + f(x,y-1) - 4f(x,y)\end{aligned} \tag{3-8}$$

由公式3-8可以直观的看到，对于离散二维空间中的点 (i,j), 由拉普拉斯算子值由该点和四个方向上的相邻点的特征值之差得到。这代表拉普拉斯算子可以描述相邻离散点之间的信息差异，反应了信息的流动方向。当 $\Delta f > 0$ 表示该点的信息将会流向相邻节点，当 $\Delta f < 0$ 时，四周节点的信息流向该节点，当 $\Delta f = 0$ 时，则关节点间不发生信息流动。由于拉普拉斯算子可以很好的描述离散节点间的信息流动方向和强度。因此，在图卷积网络中被用来描述关节点之间的联系，具体方式我们将在下一章详细叙述图卷积网络公式推导过程。我们首先将拉普拉斯算子推广到图结构数据中。

对具有 V 个节点的图 G，我们将其定义为一个 V 维的向量 $G = (x_1, \cdots, x_V \square E)$，$x_i$ 表示第 i 个节点的特征值，E 为比的集合，我们通过公式3-8将图结构数据中的拉普

拉斯算子定义为：

$$\begin{aligned}\Delta x_i &= \sum_{j\in V} A_{ij}(x_i - x_j) \\ &= \sum_{j\in V} A_{ij}x_i - \sum_{j\in V} A_{ij}x_j \\ &= d_i x_i - A_{i:}H\end{aligned} \tag{3-9}$$

其中 d_i 是节点 x_i 的度。$A_{i:} = (A_{i1}, \cdots, A_{iV})$, $H = (x_1, \cdots, x_V)^T$，$A_{i:}H$ 为这两个向量的内积。

公式3-9仅仅给出了单个节点拉普拉斯算子的计算方法，我们将其推广到所有节点，以矩阵形式表示拉普拉斯算子。

$$\begin{aligned}\Delta G &= \begin{pmatrix} \Delta x_1 \\ \vdots \\ \Delta x_V \end{pmatrix} = \begin{pmatrix} d_1 x_1 - A_{1:}H \\ \vdots \\ d_V x_V - A_{V:}H \end{pmatrix} \\ &= diag(d_i)H - AH \\ &= (D - A)H \\ &= L(G)\end{aligned} \tag{3-10}$$

公式3-10给出了图结构数据中的拉普拉斯矩阵 $L(G)$。其中 D 为度矩阵，A 为邻接矩阵，它将作为图卷积网络中的重要组成部分。

3.4 图卷积网络推导

在3.3.2节中，我们得到了图结构数据 G 的拉普拉斯矩阵 $L(G) = (D - A)H$，对该拉普拉斯矩阵进行对称归一化后可得：

$$\begin{aligned}L^{norm} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \\ &= D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} \\ &= I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\end{aligned} \tag{3-11}$$

该归一化矩阵，对角线为 1，非对角线为 $-\frac{1}{\sqrt{dev(x_i)dev(x_j)}}$。

我们希望通过傅里叶级数处理图卷积过程，可两个函数的卷积定义如下：

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\} \tag{3-12}$$

其中 $\mathcal{F}$w 为傅里叶级数。假设 LG 的特征值为 Λ，特征向量为 U。在图卷积网络中，傅里叶变换对应 $\mathcal{GF}\{x\}=U^Tx$，对应的其逆变换为 $\mathcal{IGF}\{x\}=Ux$，将二者代入公式3-12中即可得到对应的图卷积公式。

$$\begin{aligned} g*x &= \mathcal{IGF}\{\mathcal{GF}\{g\}\cdot\mathcal{GF}\{x\}\} \\ &= U(U^Tg\cdot U^Tx) \end{aligned} \tag{3-13}$$

我们将 g 定义为一个以拉普拉斯矩阵为参数的函数 $g(L)$，其作用是传播或接受周围一次周围邻居的信息。有因为 $U^TL=U^TU\Lambda U^T=\Lambda U^T$，所以我们进一步将 U^Tg 看作是以拉普拉斯矩阵特征值为参数的函数 $g_\theta(\Lambda)=diag(\theta)$，其中 θ 为参数。因此在频率域上，公式3-13可以简化表示为：

$$g_\theta x = Ug_\theta U^Tx \tag{3-14}$$

为了计算方面，我们将 g_θ 近似为：

$$g_{\theta'}*x \approx \sum_K^{k=0}\theta'_kT_k(\widetilde{\Lambda}) \tag{3-15}$$

其中 $\widetilde{\Lambda}=\Lambda-I$，$K$ 表示经过了 K 阶拉普拉斯矩阵的处理。$T_k(x)=2xT_{k-1}(x)-T_{k-2}(x)$，同时 $T_0=1$，$T_1=x$。上述近似计算代入公式3-14可得：

$$\begin{aligned} &g_\theta x = Ug_\theta U^Tx \\ &\approx U\sum_K^{k=0}\theta'_kT_k(\widetilde{\Lambda})U^Tx \quad = \sum_K^{k=0}\theta'_kT_k(U\widetilde{\Lambda}U^T)x=\sum_K^{k=0}\theta'_kT_k(\widetilde{L})x \end{aligned} \tag{3-16}$$

其中 $\widetilde{L}=L-I$，由于一个 GCN 只包含一次信息传递过程，因此拥有一个一阶的拉普拉斯矩阵，取 $k=1$ 代入公式3-16，可得到如下所示的图卷积计算公式：

$$\begin{aligned} g_\theta x &= \sum_1^{k=0}\theta'_kT_k(\widetilde{L})x \\ &= \theta'_0T_0(L-I)x+\theta'_1T_1(L-I)x \\ &= \theta'_0x+\theta'_1(L-I)x \\ &= \theta'_0x+\theta'_1(I-D^{-\frac{1}{2}}AD^{-\frac{1}{2}}-I)x \\ &= \theta'_0x-\theta'_1D^{-\frac{1}{2}}AD^{-\frac{1}{2}}x \end{aligned} \tag{3-17}$$

为了减少参数和降低模型复杂度，设 $\theta=\theta'_0=-\theta'_1$，同时令 $\widetilde{A}=I+A$ 和

$\widetilde{D}_i i = \sum_{j \in V} \widetilde{A}_i j$，可得：

$$g_\theta * x = \theta \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} x \tag{3-18}$$

再加上激活函数 σ，即可得到3.2节中提到的图卷积模型计算公式：

$$f(H^{(l)}, A) = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \tag{3-19}$$

其中 H 为输入的特征向量，W 代表参数 θ。在实际使用中，为了降低模型的复杂度，通常不进行对邻接矩阵进行归一化，因此模型被简化为：

$$f(H^{(l)}, A) = \sigma\left(\widetilde{A} H^{(l)} W^{(l)}\right) \tag{3-20}$$

3.4.1 总结

在本章节中，我们介绍了图卷积的基本概念，包括图结构、拉普拉斯算子、谱理论和特征表示等。拉普拉斯算子是图卷积的核心概念之一，通过计算它可以得到图的特征表示。接着，我们又推导了图卷积公式，介绍了如何使用卷积神经网络在图上进行卷积操作，进一步拓展了图卷积的应用。

首先，图卷积是一种用于处理图数据的卷积操作，可以用于学习图的特征表示。其次，拉普拉斯算子是计算图卷积的核心算子之一，它可以将图的结构信息转化为数学表达形式，为图卷积提供了基础。第三，谱理论是图卷积的重要理论基础之一，通过它我们可以了解拉普拉斯算子的性质和作用。最后，特征表示是图卷积的重要应用之一，它可以将图的信息转化为向量表示，方便进行机器学习任务的处理。

在对基本的图卷积网络知识进行一定的一定的铺垫后，在后续的内容中，我们将详细介绍提出的基于渐进式策略的人体运动姿态预测算法，主要包含训练策略和网络结构设计两部分。

第四章　基于渐进式策略的多阶段人体运动姿态预测框架

本章节将围绕本文的两个主要贡献点：渐进式的网络学习框架和集成 $SD-GCN$ 和 $TD-GCN$ 的图卷积模块。分别从动机、方案、实现框架和算法细节几个方面对本方法进行详细的阐述。在此之前，我们首先通过数学语言定义人体运动姿态预测问题，并介绍在此过程中使用的相关数据结构，以方便在本文后续章节中进行准确的叙述。

4.1　数据描述与问题定义

4.1.1　人体运动姿态数据结构

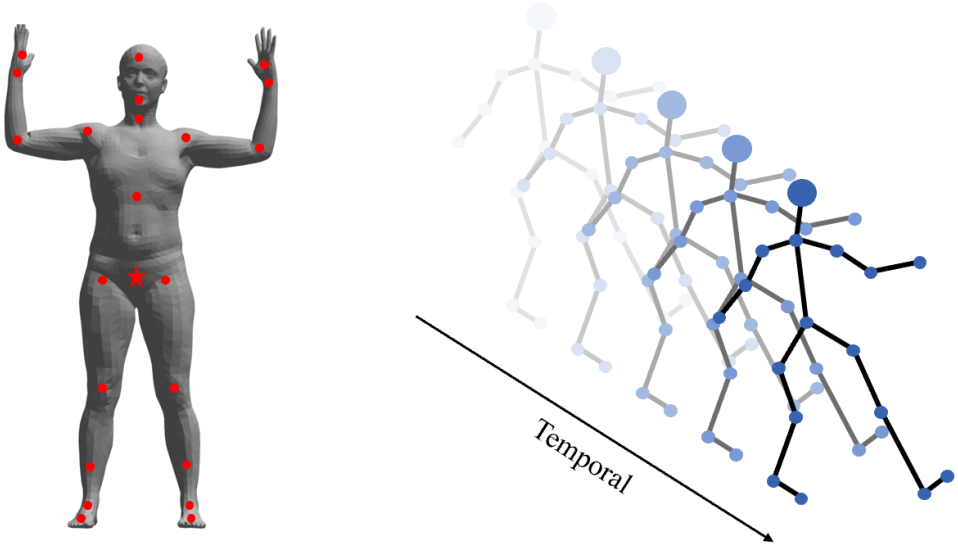

图 4-1 人体运动姿态数据结构

首先我们介绍人体运动姿态预测问题所使用的数据结构。如图4-1左所示，人体运动数据是通过动作捕捉设备，在封闭室内或开放室外场景提取到的人体关键点运动数据，这些数据以 SKT（Skeletal Kinematic Tree）的形式表示和存储。在实际动作捕捉过程中，通常只关注在运动过程中起决定性作用的关节点，例如手肘、肩部、膝盖等。这些关节点在4-1左中以红色标记的形式展现。其中位于胯部的五角星节点被称为根节点，其余节点通过递归的形式计算自身对于根节点的相对位置来得到自身位置。由于本文仅关注三维欧式空间中的人体运动，因此，我们用与根节点的相对 3D 坐标来描述每个关节点空间位置。将独立的关节点按照人体结构连接后，即可得到抽象后人体姿态。由于我们处理的是序列数据，同时包含时间和空间两个维度的关节点，因此在4-1右中，空间维度上描述人体结构信息，时间维度上描述关节点序列运动信息。

4.1.2 人体运动姿态预测问题定义

从数学上，对于一个长度为 T 的人体运动序列，我们将其定义为 $S_{1:T} = \{P_1, P_2, \cdots, P_T\}$，其中 P_i 当前运动序列中位于 i 时刻的人体姿态。每个人体姿态 P_i 又由若干个关节点组成其中 $P \in \mathbb{R}^V$，V 为该人体姿态包含的关节点数量，每个关节点又由一个 D 维的向量描述。

对于人体运动姿态预测问题，网络 Φ 接收已知输入序列 $S_{1:T_h} = \{P_1, P_2, \cdots, P_{T_h}\}$ 作为输入，预测未来运动序列 $S_{T_h+1:T_h+T_f} = \{P_{T_h+1}, P_{T_h+2}, \cdots, P_{T_h+T_f}\}$，这一过程的数学描述如公式4-1所示。其中 θ 为可训练的网络参数。

$$S_{T_h+1:T_h+T_f} = \Phi(S_{1:T_h}, \theta) \tag{4-1}$$

4.2 渐进式人体运动序列预测框架

正如1.1节所提到的，预测过程中的不确定性是影响预测精确度进一步提升的关键因素，而这种不确定性来自输入运动序列和待预测序列之间的差异。简而言之，由于输入运动序列和预测运动序列之间存在较大差异（例如，输入运动和待预测运动的运动模式有较大差异），网络无法根据输入序列中的信息来准确推测未来运动，导致预测结果脱离真实情况。因此，如何降低预测过程中的不确定性成为了当务之急。

在调研过程中，我们注意到 LTD[20] 提出的一项改进使得预测精度相较于现有方法得到了极大提升。在早期的方法中，如公式4-1所示，输入运动序列长度 $1:T_h$ 与预测序列长度 $T_h+1:T_h+T_f$ 通常存在差异，通常预测序列的长度要远远长于输入序列（例如，输入 10 帧预测 25 帧），这使得网络需要在毫无参考基础的情况下去构造未来运动序列。这通常会导致预测结果不连续，与真值出现较大偏差。针对该现象，LTD[20] 提出公式4-2, 通过使用输入序列的最后一帧来填充输入序列，使得输入序列的长度和预测序列保持一致。

$$\begin{aligned} \widetilde{S}_{1:T_h} &= [S_{1:T_h}, \{P_{T_h}^{T_h+1}, \cdots, P_{T_h}^{T_f}\}] \\ S_{T_h+1:T_h+T_f} &= \Phi(\widetilde{S}_{1:T_h}, \theta) \end{aligned} \tag{4-2}$$

从图4-2可以看到，经过填充后的输入数据对网络有两点促进作用，第一，输入和输出维度一致，避免了数据维度变换过程中的不确定性。第二，网络在预测未来序列时可以在已知部分最后一帧基础上进行预测，降低了预测的难度。然而这种粗糙的填补方法也有其固有缺陷。首先整个填充过程不区分时序距离，全部使用最后一帧进行填充，

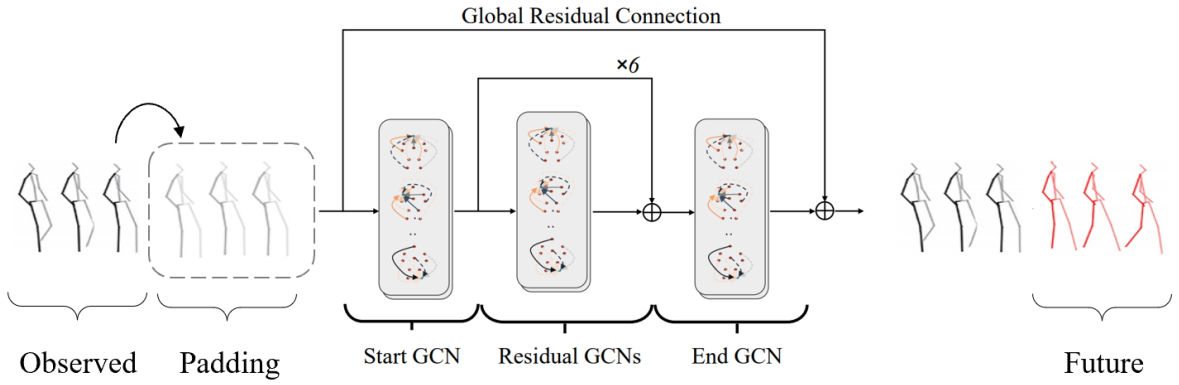

图 4-2 LTD 数据填充过程

对于离 P_{T_h} 较近的未来运动，P_{T_h} 还能提供一定的参考。随着时间向前，未来帧与 P_{T_h} 的关联越来越弱，其提供的参考价值也越来越低，预测的不确定性也逐渐增加。因此该填充方法无法缓解较远距离的预测不确定问题。

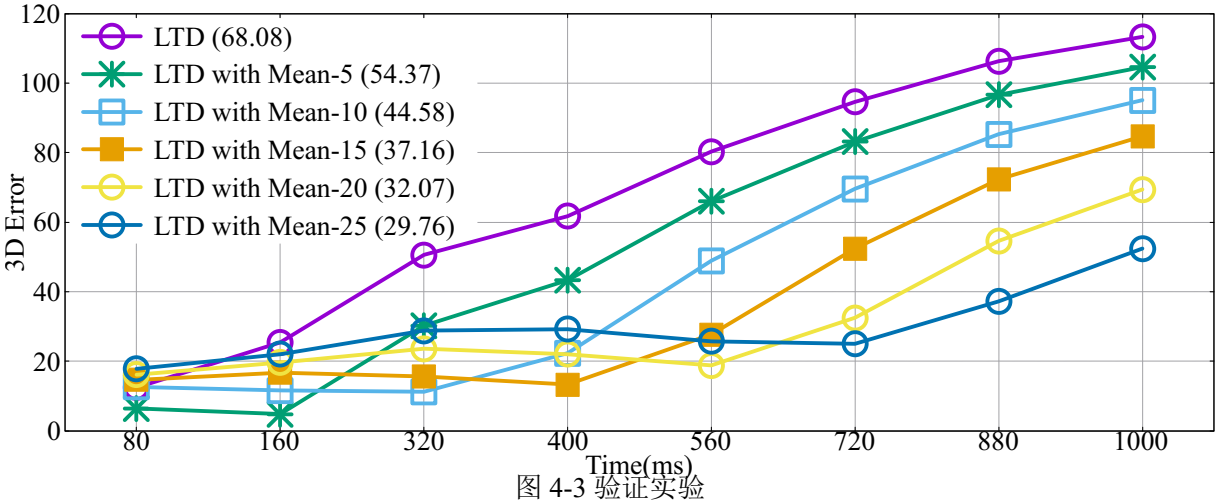

图 4-3 验证实验

LTD 证明了通过缩小输入数据和预测目标之间的维度差异，可以有效降低预测不确定性。而我们希望验证，缩小二者在内容上的差异也能达到同样的效果。因此我们以 LTD 为基准，设计了图4-3中所示的验证实验。在实验中，预测目标保持不变，输入数据的填充内容被替换为部分预测目标的均值，例如图中所表示的前 L 帧的均值（Mean-L）。从直观感觉上，填入均值相当于向输入数据泄露了部分预测目标的信息，间接缩短了二者在内容上的差距，预测难度也会随之降低。图4-3展示的结果也验证了我们的猜想，混入的平均预测目标信息有效缩短了输入数据和预测目标之间的差距，整体预测精度相较于 LTD 有了明显提升，且混入的信息越多对应位置和整体上预测精度提升越多。

在分析上述方法得失后，我们认为，缩小输入序列和待预测序列的差异（维度差异、内容差异等）可以有效降低预测过程中的不确定性，使得预测结果向我们期望的方

向趋近。但单纯地填充输入序列最后一帧所带来的性能提升还有一定的上涨空间，验证实验为我们提供了一个思路，可以通过缩小输入数据和预测目标内容上的差异来进一步提升预测精度。

由于在实际预测场景中，无法将预测目标信息泄露给输入数据，因此我们考虑通过降低预测目标的难度来缩小二者之间的内容差异。受到最近被广泛应用的由粗糙到精细（Coarse To Fine）策略的启发，我们设计了一个简易实验来验证我们的想法。Coarse to fine 策略与大多数一步到位的方法不同，预测被分为了两个阶段。位于网络浅层的阶段被称为粗糙（Coarse）网络，它接收原始的输入，并输出一个较为粗糙的结果，虽然该结果离最终的目标存在一定的偏差，但与最初的输入信息相比，它已经包含了目标的绝大部分信息。随后该粗糙结果被送入精修（Fine）网络，精修网络将在粗糙网络的基础上进一步完善预测细节。该策略被广泛应用于图片修复（Image Inpainting）[54-55] 领域，原始缺失图片通常由粗糙网络生成一个低分辨率较模糊的修复版本，此次修复的目的是修复图片内容的结构。随后，粗修版本被送入精修网络提高分辨率并进一步完善细节，最后输出高质量的修复结果。参考该思路，我们设计了一个 Coarse To Fine 人体运动序列预测网络。

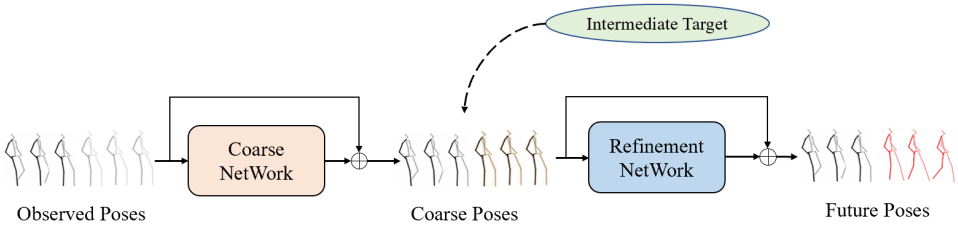

图 4-4 Coarse To Fine 预测网络

在最初的粗修阶段，我们任然保留了 LTD 中基于最后一帧的填充步骤，因为通过此步骤可以保证网络输入输出维度一致，减少预测维度上的不确定性。经过填充后的输入序列经过粗修网络，预测得到一个较为粗糙的结果，该结果受到中级监督目标的监督。该中级监督目标相比较最终的预测目标，去除了部分运动细节，只保留了运动大致趋势，通过缩小输入信息与预测目标之间的差距，降低了预测过程的不确定性。随后，粗糙的预测结果被送入后续的精修网络，进一步丰富动作的细节。此时，预测结果受到原始真值的监督，以期望获得与真值一致的结果。该模型通过两阶段的结构，将预测过程拆分为两个部分，每个部分的预测不可确定性得到了减少。预测精确性与单阶段的 LTD 网络相比有显著提升，这一结果在实验部分有所证明。

4.2.1 渐进式多阶段预测网络框架

随后我们进一步拓展了该过程，将一个两阶段的 Coarse To Fine 网络拓展为多阶段的网络。通过对预测过程的进一步细分，每个阶段的预测难度被进一步降低，网络也容易做出准确的预测。我们将多阶段的网络模型定义如下。

$$\begin{aligned}\hat{S}^1_{1:L} &= \Phi^1([S_{1:T_h}; P_{T_h}, \cdots, P_{T_h}]),\\ \hat{S}^i_{1:L} &= \Phi^i([S_{1:T_h}; \hat{S}^{i-1}_{T_h+1:L}]), i = 2, 3, \cdots, T,\end{aligned} \tag{4-3}$$

公式4-3中，现有方法中单阶段的预测过程 Φ 被拆分为多个阶段 $\Phi = \{\Phi_1, \cdots, \Phi_T\}$，每个阶段在上一个阶段的预测基础上，不断完善预测精度，使得预测精确度不断稳步提升，相比较现有的单阶段网络，多阶段网络预测过程更加可控，每阶段的预测结构都受到与之对应的中级监督目标的监督，此外预测任务的细分也使得每阶段网络的预测不确定性进一步降低，预测难度也进一步降低。为此，我们设计了一个多阶段的网络，其网络结构图如图4-5所示。

多阶段的网络设计包含两点，第一是每个阶段（Stage）的子网络设计，在这里我们最初使用了 LTD 中提出的 GCN 模块（在文章后续内容中我们提出了具有更强时空信息提取能力的 GCN 模块用以替换）。每个子网络内部为一个解码器编码器网络（Encoder-Decoder），编码器解码器结构对称。内部由多个 GCB（GCN Block）构成，每个 GCB 又由两个 GCL（GCN Layer）构成，GCL 是网络的最基本的构成结构，其具体结构如图4-3右下角所示，GCL 由一个 GCN、BatchNorm、Tanh 和 DropOut 组成，可以完成最基本的人体运动数据时空信息提取功能。值得注意的是，网络的规模并不随着阶段数量的增加而提升，在我们的设计中，网络中的 GCB 数量固定，随着阶段的增加，每个阶段包含的 GCB 数量成比例下降。网络参数量的增加在一定范围下提高了网络的容量和表达能力，但当突破某一阈值的时候网络复杂度增加带来的计算开销负担抵消了这一优势。因此我们的多阶段网络，在提升网络性能的同时，并没有提高网络的复杂程度和计算开销。第二是，中级监督目标的设计，中级监督目标必须具有层次化特点，遵循由简单到复杂的原则，浅层网络负责较为简单的处理大致框架的任务，而深层网络负责较为复杂的细节完善任务。在其他类型任务（图像等规则数据）的中级监督目标设计中，可以简单地通过降低分辨率、模糊处理和提取边缘特征等方式降低图片的细节和提取内容的结构信息来构造中级监督目标。但人体运动姿态数据是不规则的图状数据，并且已经高度抽象。无法通过寻常的降维手段提取结构信息或降低运动复杂度。虽然当前

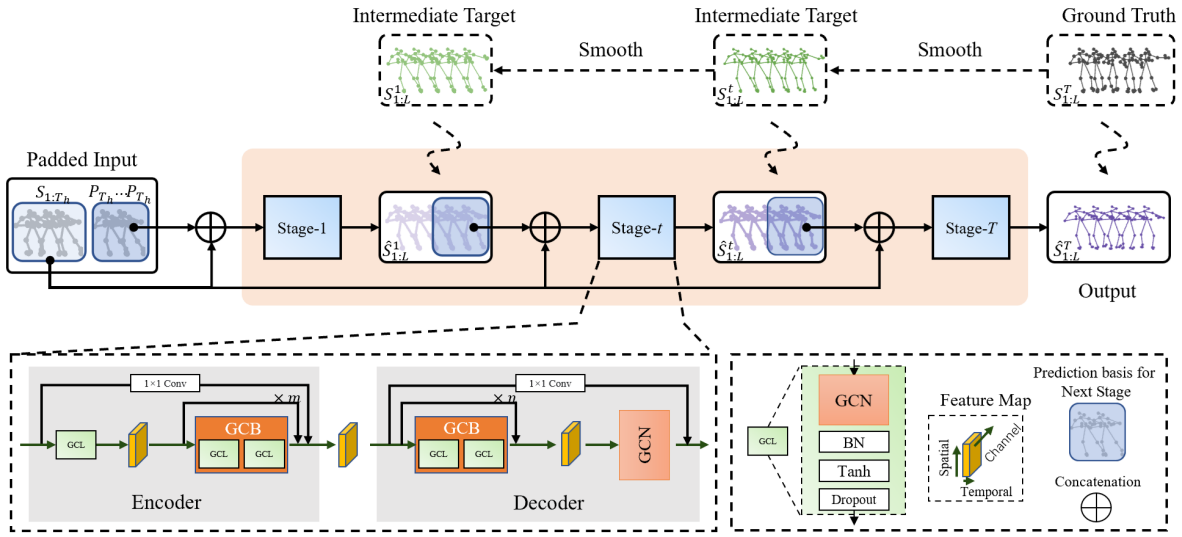

图 4-5 渐进式多阶段的预测网络

也有一些方法，如 MSR[28] 提出在空间维度上对人体结构进行降维，合并运动模型类似的相邻关节点来减少图中的关节点数量。该方法建立了空间层次化的中级监督目标，但该方法破坏了重要的人体结构先验信息，导致渐进式的策略发挥了极其有效的作用。

4.2.2 基于累积均值平滑的中级监督目标

因此我们设计了一种适用于人体运动姿态时空数据的中级监督目标构建方法，该方法可以在时间维度上降低数据的运动复杂度，同时保持原有的空间结构，以便后续网络提取其中的人体结构先验信息。具体的，我们在不同阶段对关节点轨迹施加不同强度的平滑算法。网络浅层因为表达能力不足，因此其对应的中级监督目标被施加了更强的平滑来降低运动的复杂度和预测难度。随着网络深入，网络的表达能力增加，能够承担更复杂的预测任务，此时，对于中级监督目标的平滑程度就会被削弱，以帮助其在之前阶段的预测基础上，丰富结果的细节。为此，我们设计了一种名为累积均值平滑（Accumulated Average Smoothing）的方法构造用于人体运动姿态序列中级监督目标。

在介绍累积均值平滑算法之前，为了方便叙述，我们首先给出人体运动模型中关节点轨迹的数学定义。假设每个姿态包含 V 个关节点，每个关节点由 D 维的向量描述。一个人体运动姿态序列 $S_{1:L}$ 包含 $V \times D$ 条轨迹：$\{T_j | j \in [1, M \times D]\}$，每条轨迹 T_j 由同一个关节点的某位维度上的运动组成：$T_j = \{x_j^i | i \in [1, L]\}$。由于所有轨迹都由同样的平滑方法处理，因此在下面的叙述中我们忽略了不同轨迹的区别，统一用 T 代称 T_j。

T 由两部分组成：已知的运动序列 $\{x^i | i \in [1, T_h]\}$ 和待预测的运动序列 $\{x^i | i \in [T_h + 1, T_h + T_f]\}$。由于已知的运动序列属于模型的输入数据，不需要预测，所以不需

要经过累积平滑算法处理。待预测的运动序列是网络需要预测的部分，需要用累积平滑算法调节该部分数据的预测难度。目前被广泛使用的平滑方式是基于高斯卷积核的滤波器。高斯滤波器被广泛应用于图像平滑领域，它是一种常见的线性滤波器。它的原理是将一个二维高斯分布函数应用于图像的每一个像素，使得该像素周围的像素加权平均起来，从而达到平滑图像的目的。高斯滤波器的核心是高斯核（Gaussian kernel），也称为卷积核（Convolution Kernel）或滤波器（Filter）。高斯核是一个二维高斯分布函数，它的中心是图像上的当前像素点。高斯核中的每个元素表示该位置的权重，越靠近中心位置的像素权重越高，越远离中心位置的像素权重越低。通常情况下，高斯核是一个奇数 × 奇数的矩阵，这样可以保证中心像素的位置。在轨迹平滑算法中，2D 的高斯卷积核退化为一维，其卷积核权重计算公式为：

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{4-4}$$

其中 x 表示当前位置相对于卷积核中心的距离，σ 表示标准差，标准差越大越靠近卷积核中心的权重越高，反之权重分布越平均。

$$G = \frac{1}{\sqrt{2\pi}\sigma} \begin{bmatrix} e^{-\frac{-((N-1)/2)^2}{2\sigma^2}} & \cdots & e^{-\frac{-((N-1)/2)^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{0^2}{2\sigma^2}} & \cdots & e^{-\frac{0^2}{2\sigma^2}} \\ \vdots & \ddots & \vdots \\ e^{-\frac{((N-1)/2)^2}{2\sigma^2}} & \cdots & e^{-\frac{((N-1)/2)^2}{2\sigma^2}} \end{bmatrix} \tag{4-5}$$

如果需要生成一个 $N \times 1$ 的高斯卷积核，可以将上述公式4-4代入矩阵形式，得到卷积核矩阵4-5，其中，G 是 $N \times 1$ 的高斯卷积核矩阵，N 表示卷积核的大小。

由于在本问题中，只需要对一条轨迹待预测部分进行平滑，已知的运动序列保持不变即可。而高斯滤波器在对轨迹进行处理时，不可必要的需要计算卷积核范围内的所有阶段数据，因此在已知运动序列和待预测部分的过渡部分会出现跳跃（Jump）现象，这是由于高斯滤波器在计算过渡部分的平滑值时，将卷积核范围内已知运动序列纳入计算。而已知运动序列并没有进行平滑操作，所以导致计算结果在该处出现了跳跃现象。此外，由于高斯滤波器在计算时，其节点原始值的权重较高，导致平滑力度不足，难以构造层次化的中级监督目标。

因此，我们提出了累积均值平滑算法来解决以上两个问题。我们将该算法定义如下：

$$\bar{x}^i = \frac{1}{i - T_h} \sum_{k=T_h+1}^{i} x^k, \forall i \in [T_h + 1, T_h + T_f]. \tag{4-6}$$

待预测部分的某个节点的平滑值由其之前所有节点的累计平均值得到。首先平滑值计算过程只涉及待预测部分，不涉及已知部分，这就避免了出现过渡部分的跳跃问题。其次，该节点的平滑值是由前面所有节点的平均值计算得出的，这种方法消除了原始节点权重过高影响平滑结果的问题。这意味着每个节点在平滑过程中都具有相等的影响力，没有任何一个节点能够主导结果。随着离已知部分的距离增加，该节点参与计算的节点数量增加，平滑力度也随之增强。这是因为距离已知部分较远的节点预测难度更大，需要更强的平滑力度来降低预测难度。

相反，靠近已知部分的节点保留了更多的原始信息，这是因为这些节点的不确定性较低，预测难度也相对较低，因此不需要进行过度平滑。因此，该算法能够基于每个节点的不确定性程度和预测难度来适当地平衡平滑力度和原始信息保留。这使得该算法可以在处理各种复杂的预测问题时，提供精确的结果，并减少预测误差。总之，累积均值平滑算法相比传统的高斯滤波算法能够生成过渡更平滑，难度梯度更合理的中级监督目标，能有效地帮助网络建立一个渐进式的学习框架，降低预测难度，提高网络学习效率。

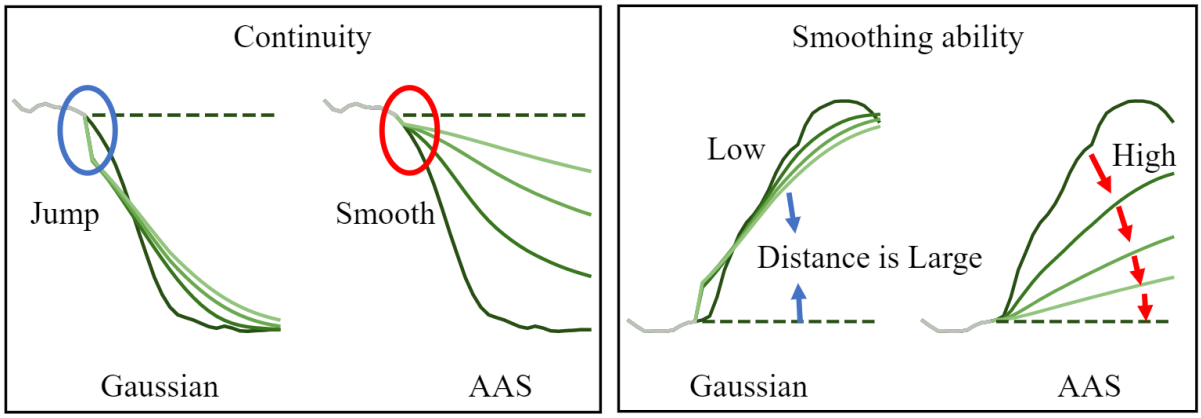

图 4-6 累积均值平滑对比高斯滤波

上图展示了累积均值平滑算法（AAS）和高斯滤波算法（Gaussian）的结果对比。其中灰色实线是输入模型的已知运动序列，深绿色虚线是使用已知运动序列最后一帧构成的填充序列。接下来由深到浅的绿色实现是不同阶段的平滑结果。拥有最深颜色的曲线是待预测的原始运动序列，稍浅一点的为经过一次平滑的结果，后续逐渐变浅的线段是经过多次平滑的结果，颜色越浅则经历的平滑次数越多。

图4-6左展示了高斯滤波出现的过渡部分跳跃问题。从图中我们可以看到，待预测的原始轨迹与已知序列的过渡部分是平滑的。然而在经过高斯滤波处理后，平滑后的待预测轨迹与已知部分的连接处，出现了明显的跳跃现象，这是由于在计算该部分平滑值时，卷积核窗口包含了已知序列和待预测序列两部分的信息。而我们提出的累积均值平滑算法解决了这一问题，在对某个节点进行平滑处理时，纳入计算的运动数据只涉及当前节点之前的待预测运动序列，不包含已知运动序列，平滑的强度随时间增加而线性增长。例如，当计算待预测序列的第一个节点的平滑值时，由于该节点已经位于待预测序列的顶端，因此，计算该节点的平滑值只需将该节点代入公式4-6，得到的结果也既是它本身。由于平滑节点数据未发生突变，因此可以和已知部分平滑过渡。

图4-6右，展示了累积均值平滑算法和高斯滤波算法的平滑力度对比。高斯滤波算法的计算过程中，在卷积核窗口中给予了原始节点过高的权值，导致平滑结果与原始节点的差异较小。最终，即使在经过迭代后的多次平滑步骤。最终结果的平滑程度仍然较低，无法满足构造层次化的中级监督目标的要求。而在累积均值平滑算法中，采用了累积的策略，越远离已知部分的节点受到的平滑力度越大，其次，使用均值的计算方法而不是加权平均的算法，平滑结果受到原始数据的影响更小，平滑的结果也更加明显。累积均值平滑算法的平滑结果表明，该算法能够在去除信号噪声的同时，保留数据序列的趋势特征。与其他常见的平滑算法相比，累积均值平滑算法在平滑强度方面表现更加优秀。这种层次化的特点使得累积均值平滑算法的平滑结果更加符合渐进式策略中预测目标由易到难的变化趋势，为多阶段预测网络模型提供了更好的中间监督目标。这种中间监督目标可以有效地降低预测难度，并提高预测的准确性和稳定性。因此，基于累积均值平滑算法的平滑结果是一种有价值的中间监督目标，它可以帮助多阶段预测网络模型实现更加准确、稳定和可靠的预测结果。

4.2.3 总结

本章介绍了基于渐进式策略的多阶段人体运动姿态预测网络，以及对应渐进式多阶段网络结构的中级监督目标构造方法。首先我们从对现有方法的分析入手，发现现阶段人体运动姿态预测问题的难点在于输入数据和待预测数据之间的差异过大，导致网络预测过程存在不确定性。LTD 提出使用已知部分来填充空白维度，使得输出输出数据达成维度上的统一，消除由于维度差异带来的不确定性。受此启发，我们希望更近一步，通过降低二者在内容上的差异性来消除预测不确定性。为此我们首先设计了一个 Coarse to fine 二阶段实验网络，人体运动姿态预测被分为两个部分，第一个阶段网络的预测目

标是较简单的大致的运动趋势，第二阶段的网络在上一步的基础上完善复杂的运动细节，使最终的网络输出与真值一致。在两阶段的网络中，我们在不增加网络参数量的前提下，通过分解任务的方式将缩小了各个阶段中，输入与输出间内容上的差距，从而降低了预测的不确定性。在最终的设计中，我们将两阶段的网络推广到多阶段的网络，进一步体现渐进式策略的优势。

我们的另一个贡献点是提出了一种名为累积均值平滑的中级监督目标构造方法，该方法相比较常用的高斯滤波平滑方法，可以避免平滑后运动序列的过渡部分出现跳跃现象。此外由于其累积平滑的特性，拥有更强的平滑能力，相比高斯滤波平滑能够生成更具层次化的中级监督目标，辅助多阶段网络构建一个从难到易的网络预测框架。

除了通过基于渐进式策略的多阶段网络结构来降低输入数据和预测目标内容上的差距，我们还期望提高网络中的基础图卷积模块的时空信息提取能力，来降低预测过程的不确定性。现有图卷积模块或缺失了对时序信息的建模，或感受野范围受限于传统算子的卷积核大小，又或是带来了不可接受的网络规模膨胀。而我们希望提出一种在不显著提升时空开销的前提下，拥有对时空信息高效建模能力的 GCN 模块。我们将在下一节详细阐述该方案。

第五章　基于时空分离策略的 Non-Local 时空图卷积模块

在第5.2章中提到，设计合理的网络架构和学习策略，进而减少预测过程中的不确定性，是提高人体运动姿态预测精度的一个有效方法。但更多的方法，着眼于提升网络特征提取能力，通过从输入数据中获得更多的信息来降低预测过程中的不确定性。早期的方法借助循环神经网络在处理序列化数据上的优势，设计了基于 RNN 的人体运动姿态运动预测算法。但这类方法只考虑了数据的序列化特性，忽略了对重要的人体空间结构信息进行建模。随后的方法，注意到了图卷积网络在对不规则数据进行结构建模的能力。设计了使用 GCN 对人体结构信息进行建模的方法。但由于人体运动姿态数据包含时间和空间两部分，使用传统的图卷积网络进行建模时，邻接矩阵的规模将随着时空维度倍数增加。因此，有部分方法提出将时空信息的处理步骤分离，空间维度由 GCN 处理，时间维度则由 TCN[45]处理。但 TCN 的卷积核大小限制了其感受野范围，不利于捕捉时间序列中长时依赖。鉴于现有方法存在的种种不足，我们提出了一种基于时空分离策略的 Non-Local 时空图卷积模块。该方法通过分离时空维度降低了网络的时间复杂度。在时间和空间维度均使用 Non-Local 的 GCN 算子，赋予网络全局感知能力。接下来，我们将首先介绍现有时空图卷积模块和本方案的设计思路以及优劣对比，随后再详细叙述基于时空分离策略的 Non-Local 时空图卷积模块的构造细节。

5.1　时空图卷积模块设计思路对比

再展示各个时空图卷积模块设计思路之前，我们首先展示人体运动姿态数据的时空结构特点。

如图5-1所示，图中的彩色节点代表关节点，同一种颜色的节点代表某个关节点在时序上的运动轨迹（Trajectory）。同一时间点所有不同颜色的节点构成一个人体姿态。在数学上，我们将上述人体运动姿态序列定义为一个时空无向图（spatialtemporal graph）$\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$。其中，$\mathcal{V}$ 是对应人体运动姿态序列中所有关节点的节点集合。而 $\mathcal{E}$ 是描述这些关节点对间联系的边集合。图5-1展示了一个长度为 4（T=4），每个姿态包含 3 个关节点 (V=3) 的人体运动序列。而时空图卷积模块的任务就是借助邻接矩阵 $\mathbf{A}^{ST}\in\mathbb{R}^{VT\times VT}$ 对包含时间和空间两个维度的人体运动姿态序列进行建模。通过如下递归的时空图卷积网络可以提取到运动序列中的高阶语义信息。

$$\mathbf{H}^{(l+1)}=\mathbf{A}_{ST}^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}, \tag{5-1}$$

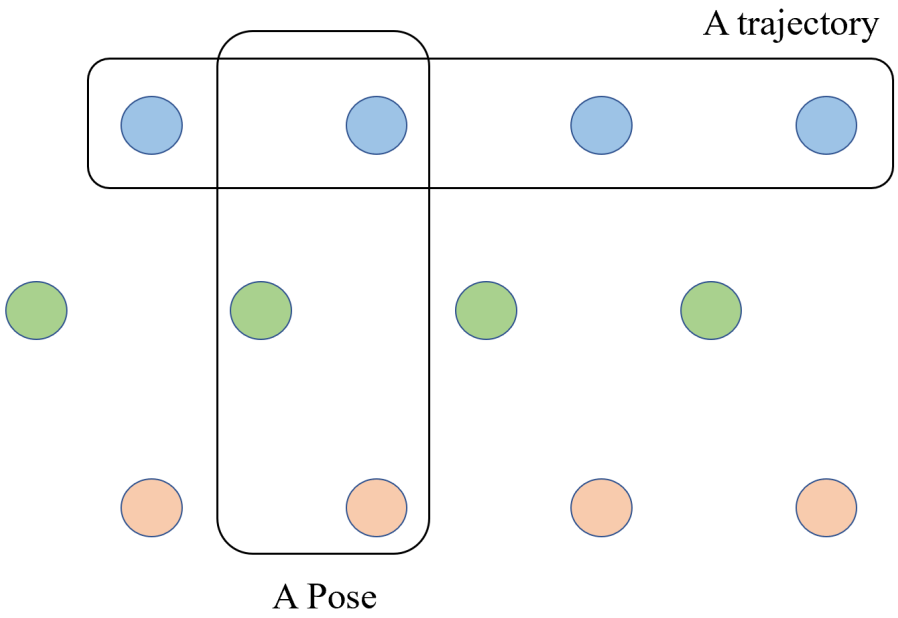

图 5-1 人体运动姿态数据时空结构

其中，l 指第 l 时空图卷积层，$\mathbf{A}_{ST}^{(l)} \in \mathbb{R}^{VT\times VT}$ 则是该层描述时空依赖关系的邻接矩阵。$\mathbf{H}^{(l)} \in \mathbb{R}^{VT\times F^l}$ 是该层待处理的特征图，特征空间等于时空维度。$\mathbf{W}^{(l)} \in \mathbb{R}^{F^l\times F^{l+1}}$ 是特征映射权值矩阵，负责将关节点特征在特征空间上进行变换。最终，$\mathbf{H}^{(l+1)}$ 作为 $\mathbb{R}^{VT\times F^{l+1}}$ 空间中的特征图被送往下一个阶段进行进一步处理。

在处理过程中，由于数据包含时间和空间两个维度，每个维度上的数据增加都会导致节点数量成倍膨胀，因此要求时空图卷积模块具有高效率。同时，为了对连续的时序数据和不规则的空间结构进行建模，时空图卷积模块需要具备较强的时空信息提取能力。而现有方法很难同时兼顾着两点要求。

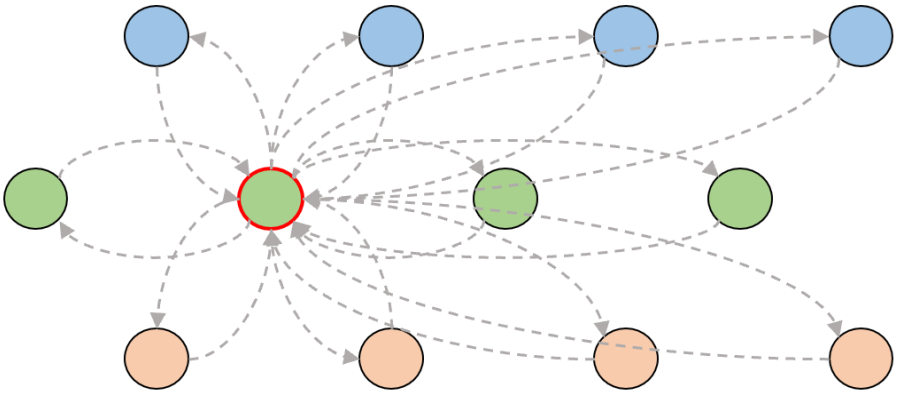

图 5-2 使用 GCN 同时对时空维度建模

最直接的思路是如图5-2所示使用一个 GCN 同时对时空两个维度进行建模，图中的

虚线表示由邻接矩阵描述的关节点对间关系，图中红圈关节点的邻接关系规模为 $\mathbb{R}^{VT}$。虽然该 GCN 可以通过邻接矩阵学习，不同空间位置、不同时间点的关节点对的联系。但这意味着在时空卷积层中，邻接矩阵的规模与数据时空维度相关 $\mathbf{A}^{ST} \in \mathbb{R}^{VT \times VT}$，这将导致模型规模剧烈膨胀，严重降低模型的运行效率。同时，如此规模的网络也使得训练过程更加困难，容易出现欠拟合现象。因此该这设计并没有被实际应用。

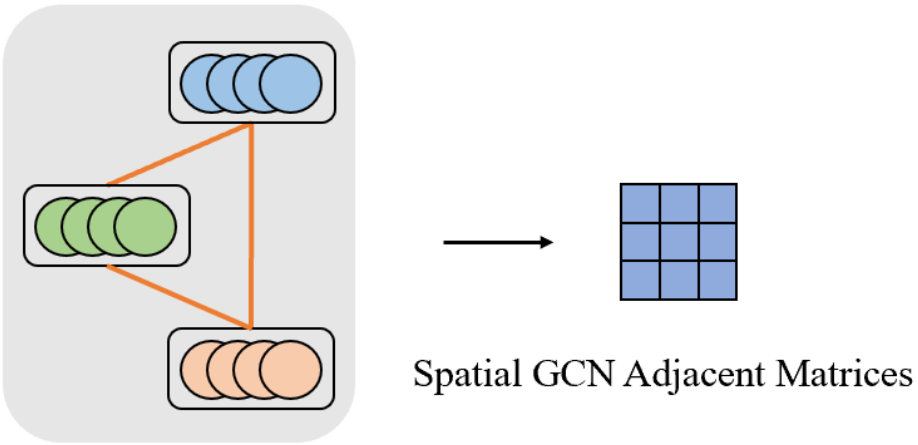

图 5-3 LTD 中的图卷积模块

作为率先使用 GCN 的方法，LTD 提出：忽略在时序空间上对输入数据的时序结构进行建模，而是将关节点运动序列看作图卷积网络中的节点值，在特征空间上通过权值矩阵 W 对齐进行变换。具体 GCN 结构由图5-3可见，其中由多个圆形组成的序列代表关节点运动序列，而它们被当作节点放入无向图中。该设计中，由于将关节点运动序列看作节点值 $\mathcal{V}_S = \{v|v \in \mathbb{R}^T\}$，时序数据被组织为一个节点值为关节点运动序列的空间无向图。避免了对时间节点间的邻接关系进行建模，因此如公式5-2所示，只需要一个空间上的 GCN 对人体空间结构进行处理，大大降低模型的空间复杂度。

$$\mathbf{H}^{(l+1)} = \mathbf{A}_S^{(l)} \mathbf{H}^{(l)} \mathbf{W}^{(l)}, \tag{5-2}$$

然而，该方法仅仅对人体空间结构进行建模，忽略了数据时序上的联系，时空信息提取能力并不完备，制约了模型对输入信息的提取与利用。

和 LTD 同期的工作 ST-GCN[49]提出了不同的思路，它使用了分离输入数据时间维度和空间维度的策略，分别设计算子对时间和空间信息进行提取。如图5-4所示。图左部分，为空间维度特征提取步骤，ST-GCN 根据人体空间结构特征使用 GCN 进行建模，且通过参数共享机制降低模型的参数量，同一个运动序列，位于不同时间点的人体姿态

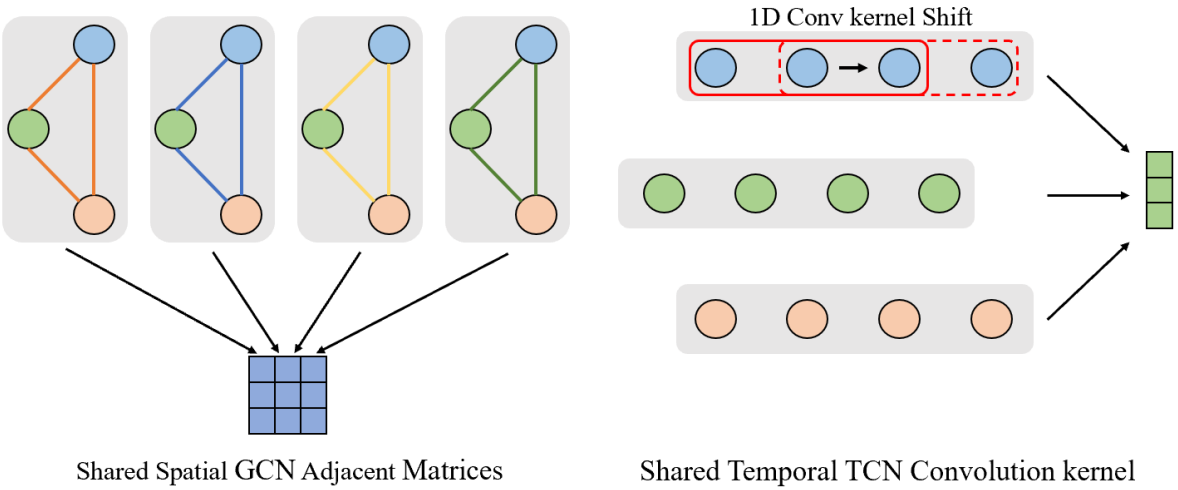

图 5-4 ST-GCN 中的图卷积模块

共享一个图卷积网络，因为人体空间结构先验信息是普适通用的，不会随着时间位置的不同而变化，因此对位于不同时刻的人体姿态，可以使用同一个 GCN 处理。图右展示了时间维度特征提取步骤，不同于人体空间结构属于不规则无向图，时序上的关节点轨迹可以看作一条 1D 的规则数据，因此可以使用适用于 1D 结构数据建模的 TCN[45]算子，TCN 是一种继承自 CNN 的模型，其空间不变性特点得以保留。在进行空间特征提取时，TCN 采用了参数共享的思路，以便将不同空间位置的关节点运动轨迹共享一个 TCN。与 CNN 类似，TCN 的卷积核也沿时间维度滑动，以提取时间信息。然而，这种设计导致模型在时间维度上的感受野受限于卷积核的尺度，从而限制了其长时依赖捕捉能力。综上，其卷积模块以公式5-3形式表示, 输入数据在经过 GCN 提取空间信息后，再由 TCN 提取时间信息，最终间接地捕捉到了数据中的时空信息。虽然 ST-GCN 较 LTD，补全了模型在时序信息提取能力上的缺陷，但受限于感受野的局部性，在长时依赖捕捉能力上任然有所欠缺。

$$\mathbf{H}^{(l+1)} = TCN(\mathbf{A}_s^{(l)}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{5-3}$$

近来，STS-GCN[56]提出了一种在时空维度均使用图卷积网络的方法，它扩展了图卷积网络（GCN）并添加了捕捉时间序列上人体姿势演变的时间卷积。与 ST-GCN 类似，它同样将时间和空间维度拆分。在空间维度上，它使用传统的图卷积网络对人体结构信息进行建模。在时间维度，与 ST-GCN 将关节点运动序列看作规则的 1D 数据不同，它选择用图结构来定义时序上关节点对的连接关系。具体的，它将关节点运动序列也视作一个无向不规则的图结构，图中的关节点不光被允许与邻接节点产生联系，还可以和

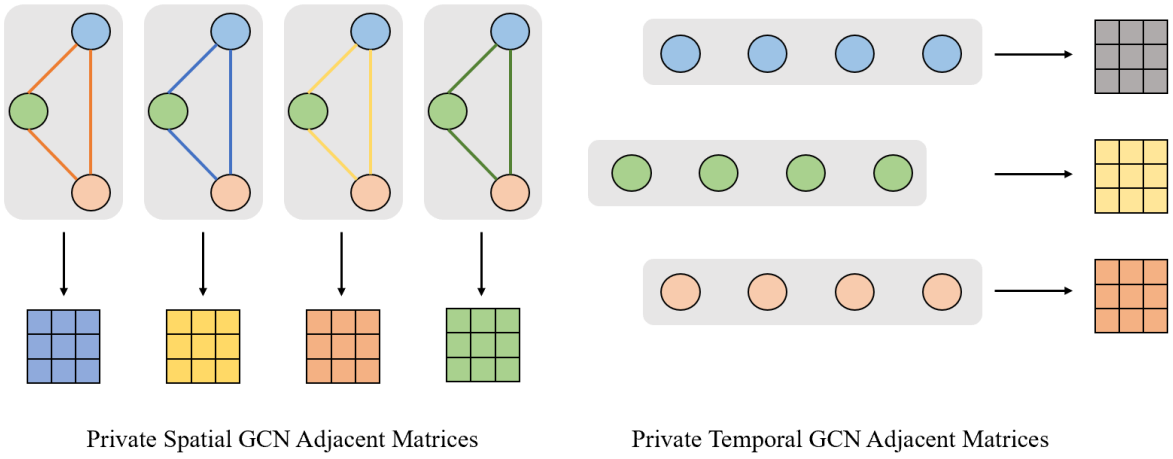

图 5-5 STS-GCN 中的图卷积模块

序列中任意一个关节点建立邻接关系。这使得网络拥有了全局的时序信息感知能力，能捕捉更长时间范围内的运动信息。这将有助于网络从整体上理解输入运动序列，从而对未来运动序列做出更准确的预测。然而，与 ST-GCN 中同一时间维度或空间维度中的数据共享同一个特征提取模块不同，STS-GCN 为不同时刻的人体姿态和不同空间位置的关节点运动轨迹设计了私有的特征提取模块。如图5-5中所示。对于不同时刻的人体姿态，STS-GCN 为每一个都分配了一个不共享的空间图卷积。对于不同空间位置的关节点序列，则由对应的、私有的时间图卷积提取特征。这虽然在一定程度上提高了网络的冗余度。但由于不同的人体姿态和关节点轨迹在空间或时间上都具有相同的空间结构或时序位置，它们对网络来说是平等的，这种平等性的存在使得网络可以更加普适地处理不同的人体姿态和关节点轨迹，而无需为每种情况设计特定的特征提取模块。

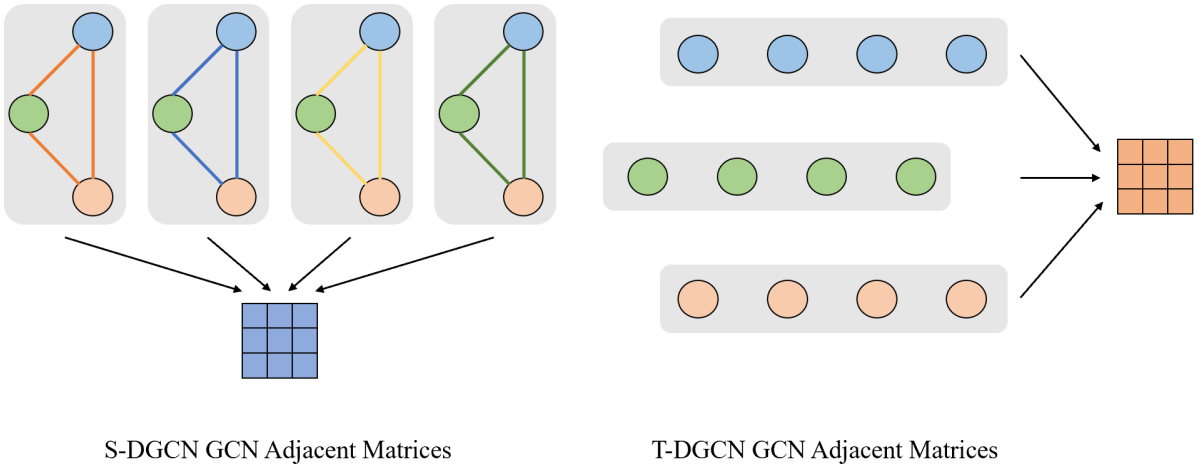

图 5-6 基于时空分离策略的 Non-Local 时空图卷积模块

我们则基于对以上方法的分析提出了一种基于时空分离策略的 Non-Local 时空图卷积模块。如图5-6所示，首先，我们将数据的时空处理步骤进行解耦。分别分配 S-DGCN（Spatial Dense GCN）和 T-DGCN（Temporal Dense GCN）用于提取空间信息和时序信息，二者均为标准的 GCN，可以捕捉时域和空域中的全局依赖，因此具有 Non-Local 的特性。值得注意的是，我们称二者为稠密（Dense）的 GCN，因此部分现有方法 [22] 中的邻接矩阵是以人为设计的方式定义，该类方法仅仅允许物理上有联系的关节点间建立联系，导致最终的邻接矩阵是稀疏的，只能处理低阶的连接关系。而真实环境中的由人体运动产生的关节互动是十分复杂的，例如，即使手肘与膝盖部位关节点没有物理上的连接，但在行走动作中，二者仍然会产生规律性的联系。我们认为人为地固定邻接矩阵将会削弱网络对高阶联系的捕捉能力。因此，我们将邻接矩阵设置为完全可学习的网络参数，赋予网络更高的灵活性。由网络根据训练数据自行定义关节点对间的邻接关系。为了降低网络规模，我们也使用了上文中提到的参数共享策略，同一个时空维度下的数据共享一个 GCN。

5.2 基于 ST-DGCN 的多阶段网络结构

具体的，在实际网络中，假设第 l 层的特征图为 $\mathbf{H}^{(l)} \in \mathbb{R}^{T\times V\times F^l}$，其中 F^l 指该层人体关节点特征维度的大小。我们通过公式5-4中的步骤更新特征图 $\mathbf{H}^{(l)}$。首先数据经过 S-DGCN 提取空间信息，随后得到的中间结果被送入 T-DGCN 提取时间信息并输出到下一层。S-DGCN 和 T-DGCN 以串联的形式结合。

$$\begin{aligned} \widetilde{H}^{(l)} &= \text{S-DGCN}(H^{(l)}) = A^s H^{(l)} W^s, (1) \\ H^{(l+1)} &= \text{T-DGCN}\widetilde{H}^{(l)} = A^t \widetilde{H}^{(l)} W^t, (2) \end{aligned} \tag{5-4}$$

将公式5-4两个公式合并后，即可得到 ST-DGCN（Spatial-temporal-DGCN）的表达式，其中 $\mathcal{T}$ 代表对输入特征图的维度顺序进行交换，以使其满足矩阵相乘的维度要求，例如，提供一个 $\mathbb{R}^{T\times V\times F^l}$ 的特征矩阵，$\mathcal{T}$ 维度变换操作，将其前两个维度顺序交换，最终输出的矩阵为 $\mathbb{R}^{V\times T\times F^l}$。注意，$\mathcal{T}$ 仅仅改变维度顺序，不改变具体数据，这使得 $\mathcal{T}$ 是一个可逆操作。

$$\mathbf{H}^{(l+1)} = \mathcal{T}\left(\mathbf{A}_t^{(l)} t\left(\mathcal{T}\left(\mathbf{A}_s^{(l)}\mathbf{H}^{(l)}\mathbf{W}_s^{(l)}\right)\right)\mathbf{W}_t^{(l)}\right), \tag{5-5}$$

图5-7展示了单个阶段的详细网络结构，对应多阶段网络模型中的每个子阶段（Stage）。在第章中曾提到，我们所设计的多阶段网络具有较灵活的通用性。具体的，

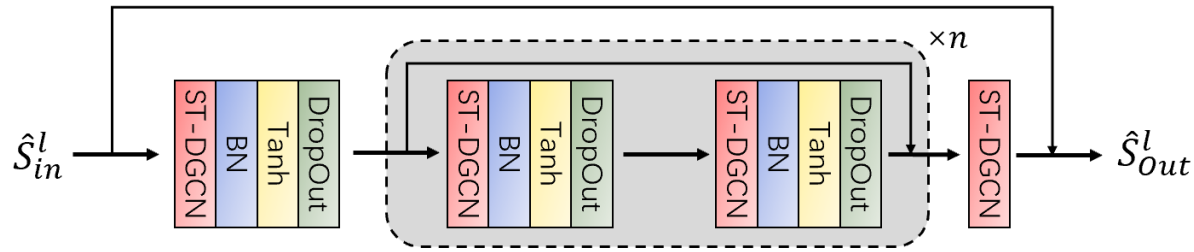

图 5-7 基于 ST-DGCN 的单阶段网络结构

多阶段网络框架中的特征提取模块可以根据具体任务灵活更换，做到即插即用，与网络框架解耦。因此，在将 ST-DGCN 应用到图**??**中时，不需要对网络结构做出过多修改，只需要用 ST-DGCN 替换网络中的原始 GCN。特征提取模块中的最基本构成部分为 GCL（Graph Convolution Layer）, 由 ST-DGCN、Batchnorm、Tanh、DropOut 构成。其中 ST-DGCN 又包含一对串联的 S-DGCN、T-DGCN。两个 GCL 构成一个 GCB（Graph Convolution Block），一个局部的残差连接跨越整个 GCB，如图中的灰色矩形部分所示。每个阶段包含 n 个 GCB。为了控制参数规模，整个网络包含的 GCB 数量固定不变，若阶段数量增多，则每个阶段包含 GCB 数量下降，我们将在实验部分探讨最佳的阶段数。最后，一个大的残差连接建立在该网络的输入与输出之间，被用来提高训练效率，对结果施加一致性约束。

在训练模式下，输入数据 $\hat{S}_i^l n$ 首先经过一个位于网络入口处的 GCL，它对输入特征的特征空间进行映射，保证与后续网络隐空间维度一致。后续的 n 个 GCB 对输入特征进行进一步的特征提取。最后得到的特征经过一个 ST-DGCN 映射后得到该阶段的预测结果。注意，最后一部分由一个独立的 ST-DGCN 构成，不包含额外的 Batchnorm、Tanh 和 DropOut。因为该部分需要输出直接的结果，不应对特征施加额外的修改。随后该输出结果接受中级监督目标的约束（如果该阶段是最后一个阶段则直接接受真值的约束。）并传递给下一阶段。最终，将5-7中所述网络复制多次，以串联的形式连接，最终即可得到图4-5中的多阶段网络。

5.3 总结

在本章节中，我们介绍了基于时空分离策略的 Non-Local 时空图卷积模块，称为 ST-DGCN。首先，我们从具有代表性的现有方法入手，分析了当前几种被广泛使用的

具有时空信息提取能力的图卷积模块。经过详细调研，我们发现现有方法在时序数据处理能力、网络效率、长时依赖捕捉能力方面有所欠缺。具体的，LTD选择牺牲网络的时间依赖提取能力，换来网络效率的提升。它放弃直接对时序信息建模，将关节点运动轨迹视作一个整体，在特征空间中对其进行变换。因此可以使用传统的 GCN 仅仅对空间接口进行建模，将网络时间复杂度控制在较低的水平。同期的工作，ST-GCN[49]则使用轻量化的 TCN 对时序数据进行建模，但受限于卷积核的大小，模型难以捕捉长时的时序依赖。STS-GCN 更近一步，将 GCN 推广到了时间维度，建立一统一的时空 GCN 模型，但其设计存在冗余，时空效率不高。

在总结以上方法的优劣势后，我们提出了 ST-DGCN，该模块的设计思路为：（1）通过时空分离的策略，将图卷积网络推广到时间和空间维度，以间接的方式建立一个高效的时空 GCN 模块，赋予网络 Non-Local 的特性。（2）通过人体姿态和关节点序列间的参数共享，控制网络参数规模，保证网络效率。最终我们实现了预测精度和时空效率的均衡，具体我们将在实验部分进一步讨论。另外，得益于第4-5章中，我们的多阶段模型的高度灵活性和拓展性，ST-DGCN 可以轻易地应用于该多阶段模型中，我们将在随后的实验证明，ST-DGCN 较强的时空信息提取能力将帮助提升网络的预测性能。

第六章　实验

6.1　前言

在上述内容中，我们分两个章节分别介绍了渐进式学习策略、基于累积均值平滑的中级监督目标构造方法和基于时空分离策略的 Non-Local 时空图卷积模块。在上述部分中，我们详细阐述了现有方法的不足与可取之处，进而引出了针对其缺点提出的改进思路，并在最终给出了我们所设计的解决方案。在本章节希望对我们提出的方法进行定性和定量的分析。具体的，我们将首先介绍模型设计中的细节，展示本方法中的具体网络设计。随后我们将叙述在实验过程中涉及到的数据集，针对各个数据集的特点以及在测试过程中使用的具体设置进行详细的讨论。随后是对实验设置的介绍，包括简述参与对比实验的各个现有方法，实验过程中所使用的对比指标，实验运行的物理环境等等。在完成对实验环境的介绍后，我们将正式进行预测结果对比步骤，首先是在各个数据集中，不同方法在不同时刻的预测精确性对比，其次是对可视化预测结果进行定性的分析。通过预测结果对比证明本方法的高效性后，我们将通过烧蚀分析对网络各部分设计的有效性进行验证。最后是对本方法进行时间效率和空间效率上的分析。

6.2　模型实现细节

由于本模型为多阶段模型，为了简便，我们在表6-1中只展示了一个阶段的网络模型。若要得到完整的模型只需要对表6-1中的结构进行简单的复制即可。除去网络入口的 $InGCN$ 和出口处的 $OutGCN$ 部分，整个网络共包含 3 个 GCB（Graph Convolution Block）模块。每个 GCB 模块又包含两个 GCL（Graph Convolution Layer），每个 GCL 由串联的 $S-DGCN$ 和 $T-DGCN$，$BatchNorm$ 以及 $DropOut$ 构成。每个 GCB 内部包含一个横跨输入与输出的残差连接，整个网络又被一个横跨网络输入输出的大的残差连接覆盖，具体的残差连接设计方式请参考表6-1。

注意，表6-1中的输入和输出数据维度，以及每一层的超参数都是基于模型在 Human3.6M 数据集上的设置得到的。例如，第三列第二行中输入特征向量的维度为（35，22，3）表示，输入特征是一个长度为 35 帧的序列（其中前 10 帧为历史已知序列，由网络输入提供。后 25 帧为待预测的未来序列，由上一阶段的预测得到（如果当前处于第一个预测阶段，则这部分数据通过复制历史已知序列最后一帧 25 次得到。））。其中的 22 表示每个人体姿态由 22 个关节点构成，每个关节点又由维度为 3 的特征向量描

表 6-1 单阶段网络实现细节

Module	Layer		Input Size	Operation	Output Size
Stage	In GCN		(3,22,35) ❶	S-DGCN:A(22,22), W(3,16)	(16,22,35)
			(16,22,35)	T-DGCN:A(35,35),W(16,16)	(16,22,35)
			(16,22,35)	BatchNorm	(16,22,35)
			(16,22,35)	Tanh	(16,22,35)
			(16,22,35)	DropOut:0.3	(16,22,35)
	GCB × 3	GCL	(16,22,35) ❷	S-DGCN:A(22.22), W(16,16)	(16,22,35)
			(16,22,35)	T-DGCN:A(35,35), W(16,16)	(16,22,35)
			(16,22,35)	BatchNorm	(16,22,35)
			(16,22,35)	Tanh	(16,22,35)
			(16,22,35)	DropOut	(16,22,35)
		GCL	(16,22,35)	S-DGCN:A(22.22), W(16,16)	(16,22,35)
			(16,22,35)	T-DGCN:A(35,35), W(16,16)	(16,22,35)
			(16,22,35)	BatchNorm	(16,22,35)
			(16,22,35)	Tanh	(16,22,35)
			(16,22,35)	DropOut	(16,22,35)❸
		Residual connection (❷ + ❸)			
	Out GCN		(16,22,35)	S-DGCN:A(22.22), W(16,3)	(3,22,35)
			(3,22,35)	T-DGCN:A(35,35), W(3,3)	(3,22,35) ❹
	Residual connection (❶ + ❹)				

述。为了满足残差连接中的维度一致性的要求，一个 $1*1 Convolution layer$ 被用来对输入特征进行特征映射，将原始特征映射到 $\mathbb{R}^{35\times22\times16}$ 空间中。

此外 $W(x,y)$，$A^S(x,y)$，$A^T(x,y)$，$W^S(x,y)$ 和 $W^T(x,y)$ 中的 x 和 y 代表网络中对

应层的权重维度。例如，在第四列第二行中，$S-DGCN$ 空间上邻接矩阵的维度为 $\mathbb{R}^{22\times22}$，权重矩阵的维度为 $\mathbb{R}^{3\times16}$。注意网络中的 DropOut 的遗忘参数被统一设置为 0.3。

6.3 数据集

我们分别在 Human3.6M[57]，CMU-MoCap，3DPW[58] 这三个数据集上进行测试。

Human3.6M 是一个大规模的人体运动捕捉数据集，通常用于与人体运动分析相关的计算机视觉和机器学习研究。它包含超过 360 万张人体主体执行各种活动的图像，例如行走、慢跑和坐着。该数据集是由 Max Planck 智能系统研究所和英属哥伦比亚大学的研究人员创建的。它使用运动捕捉系统捕捉了附着在主体身体上的标记的运动。该数据集包括 15 个不同的人体主体，每个人都执行了 17 种不同的活动。Human3.6M 数据集特别适用于人体姿势估计、动作识别和 3D 重建等领域的研究。它已被广泛用于各种研究论文，并推动了这些领域的最新技术发展。

我们参考现有工作，挑选出了由 7 个不同的采样对象贡献的包含 15 个不同种类的运动数据，其中的 7 个采集对象的代号为：S1，S5，S6，S7，S8，S9 和 S11。每个人体姿态由 32 个以 $Exponcntialmap$ 格式存储的关节点构成。由于本文主要研究 3D 空间中的人体运动，因此我们将 $Exponcntialmap$ 格式转化为 3D 格式。此外由于原始数据中存在冗余关节点，即不同编号的关节点位置重合。因此我们去除了 10 个冗余的关节点。同时，与现有方法一样，我们去除了运动姿态的整体旋转和移动，将视频帧率从 50FPS 均匀下采样到 25FPS。在数据集划分上，采样对象 S5 贡献的数据作为测试集，S11 被用于验证集，其余的数据全部作为训练集。

CMU-Mocap 数据集，也称为卡内基梅隆大学运动捕捉数据集，是由卡内基梅隆大学的运动捕捉实验室创建的一组大型和全面的运动捕捉数据。该数据集包含超过 2600 个运动捕捉序列，这些序列使用光学运动捕捉系统记录，包括各种人类运动。CMU-Mocap 数据集包括各种活动的运动捕捉数据，例如步行、奔跑、跳跃和跳舞。该数据集还包括不同主体的运动捕捉数据，包括男性和女性成年人以及儿童。CMU-Mocap 数据集中的运动捕捉数据以多种格式提供，包括 C3D、BVH 和 ASF/AMC。此外，该数据集还包括有关运动捕捉设置和校准的详细信息，以及有关主体和执行的活动的信息。CMU-Mocap 数据集已被广泛用于计算机图形学、动画、机器人和生物力学等领域的研究和开发。

CMU-Mocap 数据集包含 8 个人体运动类型，每个人体姿态包含 38 个关节点，每

个关节点的数据格式与 Human3.6M 类似，为 $Exponcntialmap$。同样的我们将其转化为 3D 格式。运动序列中的整体旋转和移动也被同时去除。在冗余关节点清除上，参考[20,28]，最终每个人体姿态只保留了 25 个关节点，其余关节点则被去除。数据集分割策略同样参考 [20,28]，数据集被分割为训练集，测试集和验证集。

3D People in the Wild（3DPW）数据集是一个大规模的自然环境下的 3D 姿态估计和跟踪基准数据集。该数据集由 60 个序列组成，包含超过 51,000 个帧，使用高质量的多摄像头设置在室内外环境中进行捕捉。数据集包括多种活动，例如走路、跳舞和运动，由不同身材、着装和光照条件的多个被试者进行演练。3DPW 数据集提供了 2D 和 3D 地面实况标注，包括 2D 关节位置、相机参数和世界坐标系下的 3D 人体姿态。使用多视图立体重建来重建 3D 姿态，通过将参数化人体模型拟合到 3D 重建中获得地面实况标注。该数据集被广泛应用于评估和开发 3D 人体姿态估计、3D 跟踪和其他相关计算机视觉任务。

对于人体运动姿态预测问题，3DPW 是一个十分具有挑战性的数据集。因为它同时包含相对稳定，环境因素简单的室内场景和情况较为复杂的室外场景。与前两个数据集不同，该数据集中的人体运动姿态序列以 3D 的形式存储，每个人体姿态由 26 个关节点构成，参考现有方法，我们去掉了前 3 个冗余的关节点，最终只保留了 23 个关节点。

6.4 实验设置

6.4.1 参与实验的现有方法

在接下来的实验部分中，我们将与 Res. Sup.[8]，DMGNN[23]，LTD[20]，STS-GCN[56] 和 MSR[28]。Res. Sup. 是一个纯 RNN 结构的方法，DMGNN 使用 GCN 对人体姿态空间结构进行建模，使用 RNN 进行序列建模。LTD 则完全依赖 GCN，且在频率域对数据进行建模。STS-GCN 提出了一种时空分离的时空 GCN，但在细节思路上与我们有所不同。MSR 则使用了一种空间维度上的渐进式预测策略。以上所有方法均提供公开代码且预测结果达到 SOTA。为了公平对比，我们使用它们的预训练模型或在它们的公开代码上使用默认的超参数进行训练。

6.4.2 定量对比指标

我们使用 MPJPE（Mean Per-Joint Position Error）作为评价模型预测精确程度的指标。MPJPE 代表平均关节点位置误差，用于评估 3D 人体姿态预测模型性能。它测量预测的 3D 关节点位置与真实的 3D 关节点位置之间的平均距离，覆盖数据集中的所有关

节点和所有帧。MPJPE 是流行的度量标准，因为它易于计算和解释，并提供简单的定量测量 3D 姿态估计的准确性。较低的 MPJPE 值表示姿态估计模型的性能更好，而较高的值表示性能更差。MPJPE 被定义为公式6-1。

$$MPJPE = \frac{1}{VT}\sum_{i=1}^{V}\sum_{j=1}^{T}|\mathbf{p}_{ij} - \mathbf{p}_{ij}^{GT}|^2 \tag{6-1}$$

其中 V 是关节点的数量，T 是帧数，$\mathbf{p}_{ij}$ 是第 i 个关节点在第 j 帧中的预测 3D 位置，$\mathbf{p}_{ij}^{GT}$ 是第 i 个关节点在第 j 帧中的真实 3D 位置，$|\cdot|^2$ 表示欧几里得距离。VT 表示数据集中的关节点和帧数总数。MPJPE 通过统计每个关节点预测值和真实值之间的平均误差来度量预测精确性。注意我们只在未知待预测运动序列上计算 MPJPE，而非对整个输出序列。例如，假如输入网络的数据长度为 35 帧，前 10 帧为已知历史运动序列，后 25 帧为待预测的未来序列，我们只在后 25 帧上计算 MPJPE。

6.4.3 超参数设置和实验环境

我们的多阶段网络包含 4 个阶段，每个阶段共包含 3 个 GCB 以及入口和出口处的 ST-DGCN 网络。网络中共计包含 12 个 GCB 模块。我们使用 $Adam$ 作为求解器，学习率被初始化为 0.005，随后在训练过程中以每个 epoch 0.96 的递减率下降。训练过程总共包含 50 个 epoch，训练时 batchsize 为 16。我们在 NVIDIA RTX 2060 GPU 和 AMD Ryzen 5 3600 CPU 的设备上进行测试。

在网络输入输出格式上，我们参考现有方法，在 Human3.6M 和 CMU-Mocap 数据集上采用输入 10 帧预测 25 帧，在 3DPW 上，输入 10 帧预测未来 30 帧。

需要注意的是，对于 Human3.6M 数据集，我们发现现有方法采用了独特的测试方式。由于 Human3.6M 数据集中各运动样本数量有细微差距，部分方法选择在每个运动中采样固定数据量的样本作为测试数据集。具体的，[8,20,23] 从每个动作中采样 8 个样本，而 [21] 从每个动作中采样 256 个样本。我们认为上述采样方式虽然可以确保每个动作提供的样本数量一致，但大大减小的测试数据集的规模，导致模型在该数据集上的表现不稳定。同时，Human3.6M 中各运动样本数量差距并不悬殊，因此我们沿用了 MSR[28] 的做法，不进行采样操作，使用全部的测试数据集。为了公平对比，我们也提供了采样数据集上的预测对比，结果显示，我们的优势并没有被削弱。

表 6-2 Human3.6M 上的短时预测误差对比

scenarios	walking				eating				smoking			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	29.4	50.8	76.0	81.5	16.8	30.6	56.9	68.7	23.0	42.6	70.1	82.7
DMGNN	17.3	30.7	54.6	65.2	11.0	21.4	36.2	43.9	9.0	17.6	32.1	40.3
LTD	12.3	23.0	39.8	46.1	8.4	16.9	33.2	40.7	7.9	16.2	31.9	38.9
STS-GCN	17.3	31.7	44.7	51.5	12.7	23.9	38.4	46.1	12.7	23.7	35.6	42.7
MSR	12.2	22.7	38.6	45.2	8.4	17.1	33.0	40.4	8.0	16.3	31.3	38.2
Our	**10.2**	**19.8**	**34.5**	**40.3**	**7.0**	**15.1**	**30.6**	**38.1**	**6.6**	**14.1**	**28.2**	**34.7**
scenarios	discussion				directions				greeting			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	32.9	61.2	90.9	96.2	35.4	57.3	76.3	87.7	34.5	63.4	124.6	142.5
DMGNN	17.3	34.8	61.0	69.8	13.1	24.6	64.7	81.9	23.3	50.3	107.3	132.1
LTD	12.5	27.4	58.5	71.7	9.0	19.9	43.4	53.7	18.7	38.7	77.7	93.4
STS-GCN	17.0	34.5	61.9	74.7	13.7	28.0	48.6	58.9	23.0	45.2	81.5	96.9
MSR	12.0	26.8	57.1	69.7	8.6	19.7	43.3	53.8	16.5	37.0	77.3	93.4
Our	**10.0**	**23.8**	**53.6**	**66.7**	**7.2**	**17.6**	**40.9**	**51.5**	**15.2**	**34.1**	**71.6**	**87.1**
scenarios	phoning				posing				purchases			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	38.0	69.3	115.0	126.7	36.1	69.1	130.5	157.1	36.3	60.3	86.5	95.9
DMGNN	12.5	25.8	48.1	58.3	15.3	29.3	71.5	96.7	21.4	38.7	75.7	92.7
LTD	10.2	21.0	42.5	52.3	13.7	29.9	66.6	84.1	15.6	32.8	65.7	79.3
STS-GCN	14.5	27.8	46.6	56.6	20.2	40.9	75.5	93.3	20.6	40.4	71.3	84.8
MSR	10.1	20.7	41.5	51.3	12.8	29.4	67.0	85.0	14.8	32.4	66.1	79.6
Our	**8.3**	**18.3**	**38.7**	**48.4**	**10.7**	**25.7**	**60.0**	**76.6**	**12.5**	**28.7**	**60.1**	**73.3**
scenarios	sitting				sittingdown				takingphoto			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	42.6	81.4	134.7	151.8	47.3	86.0	145.8	168.9	26.1	47.6	81.4	94.7
DMGNN	11.9	25.1	44.6	**50.2**	15.0	32.9	77.1	93.0	13.6	29.0	46.0	58.8
LTD	10.6	21.9	46.3	57.9	16.1	31.1	61.5	75.5	9.9	20.9	45.0	56.6
STS-GCN	16.2	30.2	52.7	65.1	23.3	41.5	69.6	83.5	16.7	31.4	52.8	64.3
MSR	10.5	22.0	46.3	57.8	16.1	31.6	62.5	76.8	9.9	21.0	44.6	56.3
Our	**8.8**	**19.2**	**42.4**	53.8	**13.9**	**27.9**	**57.4**	**71.5**	**8.4**	**18.9**	**42.0**	**53.3**

scenarios	waiting				walkingdog				walkingtogether				avg			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	30.6	57.8	106.2	121.5	64.2	102.1	141.1	164.4	26.8	50.1	80.2	92.2	34.7	62.0	101.1	115.5
DMGNN	12.2	24.2	59.6	77.5	47.1	93.3	160.1	171.2	14.3	26.7	50.1	63.2	17.0	33.6	65.9	79.7
LTD	11.4	24.0	50.1	61.5	23.4	46.2	83.5	96.0	10.5	21.0	38.5	45.2	12.7	26.1	52.3	63.5
STS-GCN	16.1	31.9	54.0	65.2	27.8	51.5	85.0	97.4	15.3	27.8	41.2	48.0	17.8	34.0	57.3	68.6
MSR	10.7	23.1	48.3	59.2	20.7	42.9	80.4	93.3	10.6	20.9	37.4	43.9	12.1	25.6	51.6	62.9
Our	**8.9**	**20.1**	**43.6**	**54.3**	**18.8**	**39.3**	**73.7**	**86.4**	**8.7**	**18.6**	**34.4**	**41.0**	**10.3**	**22.7**	**47.4**	**58.5**

表 6-3 Human3.6M 上的长时预测误差对比。

scenarios	walking		eating		smoking		discussion		directions	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	81.7	100.7	79.9	100.2	94.8	137.4	121.3	161.7	110.1	152.5
DMGNN	73.4	95.8	58.1	86.7	50.9	72.2	**81.9**	138.3	110.1	115.8
LTD	54.1	59.8	53.4	77.8	50.7	72.6	91.6	121.5	71.0	101.8
STS-GCN	56.3	69.1	59.3	84.3	53.3	75.4	93.0	121.8	75.5	104.8
MSR	52.7	63.0	52.5	77.1	49.5	71.6	88.6	**117.6**	71.2	100.6
Our	**48.1**	**56.4**	**51.1**	**76.0**	**46.5**	**69.5**	87.1	118.2	**69.3**	**100.4**
scenarios	greeting		phoning		posing		purchases		sitting	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	156.1	166.5	141.2	131.5	194.7	240.2	122.7	160.3	167.4	201.5
DMGNN	152.5	157.7	78.9	**98.6**	163.9	310.1	118.6	153.8	**60.1**	**104.9**
LTD	115.4	148.8	69.2	103.1	114.5	173.0	102.0	143.5	78.3	119.7
STS-GCN	117.6	145.6	73.3	108.5	122.4	175.1	104.8	140.8	86.0	124.3
MSR	116.3	147.2	68.3	104.4	116.3	174.3	101.6	139.2	78.2	120.0
Our	**110.2**	**143.5**	**65.9**	102.7	**106.1**	**164.8**	**95.3**	**133.3**	74.4	116.1

scenarios	sittingdown		takingphoto		waiting		walkingdog		walkingtogether		average	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	205.3	277.6	117.0	143.2	146.2	196.2	191.3	209.0	107.6	131.1	97.6	130.5
DMGNN	122.1	168.8	91.6	120.7	106.0	136.7	194.0	182.3	83.4	115.9	103.0	137.2
LTD	100.0	150.2	77.4	119.8	79.4	108.1	111.9	148.9	55.0	65.6	81.6	114.3
STS-GCN	107.8	157.3	85.0	125.8	81.7	112.4	114.3	146.9	56.1	69.8	85.7	117.5
MSR	102.8	155.5	77.9	121.9	76.3	106.3	111.9	148.2	52.9	65.9	81.1	114.2
Our	**96.7**	**147.8**	**74.3**	**118.6**	**72.2**	**103.4**	**104.7**	**139.8**	**51.9**	**64.3**	**76.9**	**110.3**

6.5 预测误差对比

6.5.1 Human3.6

表6-2展示了在 Human3.6M 数据集上的短时预测结果（输入 10 帧预测 10 帧，在 25FPS 的视频中即输入 400ms 的运动预测 400ms 的运动。），我们统计了 15 个运动类别上，80ms，160ms，320ms，400ms 四个时刻的平均预测误差，图中预测误差最低的结果被加粗，而第二好的结果被加上了下划线。从表6-2可以看到，在全部 15 个运动类别共 60 个时刻中，本方法在其中的 59 个时刻预测误差均为六个方法中的最低，唯一

一个落后的时刻也处于第二低的位置。且本方法相较于现有方法在 sitting，sittingdown，purchases，walkingdog 这些较为复杂的运动上取得了较为明显的领先，这表明了本方法的有效性，在所有运动的平均误差上，我们的结果（10.3，22.7，47.4，58.5）也远远高于第二的 MSR（12.1，25.6，51.6，62.9）。

表6-3展示了在 Human3.6M 数据集上长时预测结果（输入 10 帧预测 25 帧，即输入 400ms 预测 1000ms。），与表6-2一样，我们统计了 15 个运动类别上，560ms，1000ms 两个时刻的平均预测误差，不同方法间预测误差最低的结果被加粗，第二低的结果添加下划线。在长时预测中，本方法依然在 30 个时刻中的 25 个中的得到了最低的预测误差，其余 5 个非最优结果中，也位于第二优的位置。且由于长时预测中不确定性增加，本方法相较于现有 SOTA 方法优势更明显，体现了本方法在提取长时依赖能力上的强大。

除了在完整的测试数据集上进行测试，我们还参考 [8,20-21,23] 分别在每个动作采样 8 或 256 个样本的的测试数据集上进行实验。表6-4和表6-5分别表示模型在每个动作采样 256 个样本的测试数据集上的短时和长时预测误差。表6-6和表6-7分别表示模型在每个动作采样 8 个样本的测试数据集上的短时和长时预测误差。注意在这两个表中，我们添加与 Transformer[52] 对比的内容，因为该文章只提供了采样 8 个样本的预测结果，且在长时预测上存在部分数据缺失的情况。从上述对比表格来看，本方法依然延续了在预测精度上的优势。

表 6-4 Human3.6M 上每个动作随机采样 256 的短时误差对比

scenarios	walking				eating				smoking				discussion			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	23.2	40.9	61	66.7	16.8	31.5	53.5	61.7	18.9	34.7	57.5	65.4	25.7	47.8	80	91.3
DMGNN	18.4	33.6	56.8	65.1	10.1	19.7	38.3	46.7	11.4	22.0	41.5	50.1	18.0	36.2	71.9	85.2
LTD	11.1	21.4	37.3	42.9	7	14.8	29.8	37.3	7.5	15.5	30.7	37.5	10.8	24	52.7	65.8
STS-GCN	18.4	33.6	56.8	65.1	10.1	19.7	38.3	46.7	11.4	22.0	41.5	50.1	18.0	36.2	71.9	85.2
MSR	10.8	20.9	36.9	42.4	6.9	14.6	29.0	36.0	7.5	15.4	30.6	37.5	10.4	23.5	51.9	65.0
Ours	**9.4**	**19.0**	**34.3**	**40.4**	**6.0**	**13.4**	**27.8**	**35.3**	**6.5**	**14.2**	**28.8**	**35.5**	**9.0**	**21.8**	**49.9**	**62.9**
scenarios	directions				greeting				phoning				posing			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	21.6	41.3	72.1	84.1	31.2	58.4	96.3	108.8	21.1	38.9	66	76.4	29.3	56.1	98.3	114.3
DMGNN	13.8	27.7	55.3	67.2	22.6	45.1	89.0	106.6	14.3	28.0	52.4	63.3	18.6	37.6	80.1	100.0
LTD	8	18.8	43.7	54.9	14.8	31.4	65.3	79.7	9.3	19.1	39.8	49.7	10.9	25.1	59.1	75.9
STS-GCN	13.8	27.7	55.3	67.2	22.6	45.1	89.0	106.6	14.3	28.0	52.4	63.3	18.6	37.6	80.1	100.0
MSR	7.7	18.9	44.7	56.2	15.1	33.1	70.9	85.4	9.1	18.9	39.9	49.8	10.3	24.6	59.2	75.9
Ours	**6.4**	**16.8**	**41.5**	**52.7**	**12.4**	**28.3**	**61.2**	**76.0**	**7.8**	**17.2**	**37.3**	**47.3**	**8.7**	**22.2**	**53.9**	**70.4**
scenarios	purchases				sitting				sittingdown				takingphoto			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	28.7	52.4	86.9	100.7	23.8	44.7	78	91.2	31.7	58.3	96.7	112	21.9	41.4	74	87.6
DMGNN	21.7	42.4	77.3	91.6	14.7	30.0	61.5	74.5	20.7	39.9	81.0	97.4	14.4	29.2	59.4	74.6
LTD	13.9	30.3	62.2	75.9	9.8	20.5	44.2	55.9	15.6	31.4	59.1	71.7	8.9	18.9	41	51.7
STS-GCN	21.7	42.4	77.3	91.6	14.7	30.0	61.5	74.5	20.7	39.9	81.0	97.4	14.4	29.2	59.4	74.6
MSR	13.3	30.1	63.6	77.8	9.8	20.6	44.2	55.5	15.4	32.0	60.7	73.8	8.9	19.5	43.1	54.4
Ours	**11.7**	**27.8**	**59.4**	**73.5**	**8.5**	**18.8**	**41.8**	**53.2**	**13.7**	**29.3**	**57.2**	**69.7**	**7.6**	**17.2**	**38.5**	**49.2**
scenarios	waiting				walkingdog				walkingtogether				average			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	23.8	44.2	75.8	87.7	36.4	64.8	99.1	110.6	20.4	37.1	59.4	67.3	25	46.2	77	88.3
DMGNN	15.5	30.7	61.5	74.4	31.7	62.1	109.8	125.3	15.7	29.2	51.1	60.7	17.4	34.2	65.8	78.9
LTD	9.2	19.5	43.3	54.4	20.9	40.7	73.6	86.6	9.6	19.4	36.5	44	11.2	23.4	47.9	58.9
STS-GCN	15.5	30.7	61.5	74.4	31.7	62.1	109.8	125.3	15.7	29.2	51.1	60.7	17.4	34.2	65.8	78.9
MSR	10.4	22.4	50.7	62.4	24.9	51.5	100.3	112.9	9.2	18.7	35.7	43.2	11.3	24.3	50.8	61.9
Ours	**7.4**	**17.3**	**39.6**	**50.8**	**18.4**	**38.1**	**71.8**	**85.1**	**8.1**	**17.4**	**34.0**	**41.5**	**9.4**	**21.3**	**45.1**	**56.2**

表 6-5 Human3.6M 上每个动作随机采样 256 的长时误差对比

scenarios	walking		eating		smoking		discussion		directions		greeting		phoning		posing	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	71.6	79.1	74.9	98	78.1	102.1	109.5	131.8	101.1	129.1	126.1	153.9	94	126.4	140.3	183.2
DMGNN	75.4	96.8	61.9	91.0	64.1	93.2	107.1	138.6	88.4	121.4	132.5	165.2	80.0	112.9	136.6	210.4
LTD	51.8	60.9	50	**74.1**	51.3	73.6	87.6	118.6	76.1	108.8	104.3	140.2	68.7	105.1	109.9	171.7
STS-GCN	75.4	96.8	61.9	91.0	64.1	93.2	107.1	138.6	88.4	121.4	132.5	165.2	80.0	112.9	136.6	210.4
MSR	53.3	63.7	50.8	75.4	50.5	72.1	87.0	**116.8**	75.8	105.9	106.3	**136.3**	67.9	104.7	112.5	176.5
Ours	**49.6**	**58.9**	**50.0**	74.9	**48.8**	**69.9**	**86.1**	116.9	**73.3**	**105.9**	**100.2**	136.4	**66.5**	**102.7**	**102.8**	**167.0**
scenarios	purchases		sitting		sittingdown		takingphoto		waiting		walkingdog		walkingtogether		average	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	122.1	154	113.7	152.6	138.8	187.4	110.6	153.9	105.4	135.4	128.7	164.5	80.2	98.2	106.3	136.6
DMGNN	115.5	155.9	95.7	138.7	130.4	188.1	100.3	146.8	97.1	141.5	147.2	184.9	74.7	97.5	100.5	138.9
LTD	99.4	135.9	78.5	118.8	99.5	144.1	76.8	120.2	75.1	106.9	105.8	**142.2**	58	69.6	79.5	112.7
STS-GCN	115.5	155.9	95.7	138.7	130.4	188.1	100.3	146.8	97.1	141.5	147.2	184.9	74.7	97.5	100.5	138.9
MSR	99.2	134.5	77.6	115.9	102.4	149.4	77.7	121.9	74.8	105.5	107.7	145.7	56.2	69.5	80.0	112.9
Ours	**95.7**	**132.1**	**75.1**	**114.8**	**94.4**	**139.0**	**70.5**	**112.9**	**71.6**	**103.7**	**105.7**	145.9	**54.4**	**64.6**	**76.3**	**109.7**

表 6-6 Human3.6M 上每个动作随机采样 8 的短时误差对比

scenarios	walking				eating				smoking				discussion			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	23.8	40.4	62.9	70.9	17.6	34.7	71.9	87.7	19.7	36.6	61.8	73.9	31.7	61.3	96	103.5
DMGNN	17.2	30.6	54.4	65.0	11.0	21.4	35.9	43.5	8.9	17.3	31.7	40.0	17.4	34.6	60.8	69.5
LTD	8.9	15.7	29.2	33.4	8.8	18.9	39.4	47.2	7.8	14.9	25.3	28.7	9.8	22.1	39.6	**44.1**
STS-GCN	17.2	30.6	54.4	65.0	11.0	21.4	35.9	43.5	8.9	17.3	31.7	40.0	17.4	34.6	60.8	69.5
MSR	8.7	15.5	28.4	32.4	8.3	**17.7**	**36.3**	**43.7**	7.5	15.4	27.4	31.5	9.3	22.1	40.5	45.5
Transformer	7.9	**14.5**	29.1	34.5	8.4	18.1	37.4	45.3	6.8	**13.2**	**24.1**	**27.5**	8.3	21.7	43.9	48.0
Ours	**7.6**	14.6	**24.9**	**28.3**	**8.0**	17.9	38.0	45.7	**6.3**	13.4	25.2	30.3	**7.3**	**19.3**	**38.1**	45.2
scenarios	directions				greeting				phoning				posing			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	36.5	56.4	81.5	97.3	37.9	74.1	1390	158.8	25.6	44.4	74	84.2	27.9	54.7	131.3	160.8
DMGNN	13.2	24.9	64.8	81.9	23.4	50.3	107.2	131.9	12.7	26.0	48.4	58.4	15.3	29.2	71.5	96.6
LTD	12.6	24.4	48.2	58.4	14.5	30.5	74.2	89	11.5	20.2	37.9	43.2	9.4	23.9	66.2	82.9
ST-GCN	13.2	24.9	64.8	81.9	23.4	50.3	107.2	131.9	12.7	26.0	48.4	58.4	15.3	29.2	71.5	96.6
MSR	11.4	21.9	**45.8**	**56.1**	13.5	26.5	68.8	86.1	11.8	20.6	37.5	41.7	8.5	21.8	**61.2**	**76.4**
Transformer	11.1	22.7	48.0	58.4	13.2	28.0	64.5	**77.9**	10.8	19.6	37.6	46.8	8.3	22.8	65.6	81.8
Ours	**10.1**	**21.7**	48.1	59.5	**11.2**	**24.1**	**63.6**	80.0	**10.6**	**18.8**	**34.1**	**39.7**	**6.6**	**20.1**	61.6	78.1
scenarios	purchases				sitting				sittingdown				takingphoto			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	40.8	71.8	104.2	109.8	34.5	69.9	126.3	141.6	28.6	55.3	101.6	118.9	23.6	47.4	94	112.7
DMGNN	21.4	38.8	75.9	93.0	11.9	25.2	**44.6**	**50.1**	15.0	32.8	77.1	93.1	13.5	28.7	45.6	58.4
LTD	19.6	38.5	64.4	72.2	10.7	24.6	50.6	62	11.4	27.6	56.4	67.6	6.8	15.2	38.2	**49.6**
STS-GCN	21.4	38.8	75.9	93.0	11.9	25.2	**44.6**	**50.1**	15.0	32.8	77.1	93.1	13.5	28.7	45.6	58.4
MSR	19	38.7	64.5	72.6	11.3	26.5	56.1	69.2	11.1	28.2	56.1	66.8	6.6	15.8	40.8	53.1
Transformer	18.5	38.1	**61.8**	**69.6**	9.5	23.9	49.8	61.8	11.2	29.9	59.8	68.4	6.3	14.5	38.8	49.4
Ours	**17.2**	**36.5**	63.4	72.2	**8.3**	**22.1**	49.3	61.4	**9.8**	**26.3**	**53.5**	**63.2**	**5.8**	**14.1**	**38.0**	49.8
scenarios	waiting				walkingdog				walkingtogether				average			
millisecond	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms	80ms	160ms	320ms	400ms
Res. Sup.	29.5	60.5	119.9	140.6	60.5	101.9	160.8	188.3	23.5	45	71.3	82.8	30.8	57	99.8	115.5
DMGNN	12.1	23.8	59.5	77.5	47.1	93.3	160.3	171.4	14.4	26.7	50.1	63.2	17	33.6	65.9	79.6
LTD	9.5	22	57.5	73.9	32.2	58	102.2	122.7	8.9	18.4	35.3	44.3	12.1	25	51	61.3
ST-GCN	12.1	23.8	59.5	77.5	47.1	93.3	160.3	171.4	14.4	26.7	50.1	63.2	17	33.6	65.9	79.6
MSR	8.9	20.9	53.6	69.8	24.4	53.6	**95.6**	**110.4**	8.7	18.5	35.4	45.6	11.3	24.3	49.9	60.1
Transformer	8.4	21.5	53.9	69.8	**22.9**	**50.4**	100.8	119.8	8.7	18.3	34.2	44.1	10.7	23.8	50.0	60.2
Ours	**7.4**	**18.2**	**50.4**	**66.7**	27.3	53.6	97.6	119.0	**7.2**	**16.7**	**33.8**	**42.8**	**10.1**	**22.5**	**48.0**	**58.8**

表 6-7 Human3.6M 上每个动作随机采样 8 的长时误差对比

scenarios	walking		eating		smoking		discussion		directions		greeting		phoning		posing	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	86.3	107.6	87.7	99.4	96.1	141.4	120.7	161.6	110.2	150.5	162.2	174.227	139.098	127.029	192.096	230.697
DMGNN	73.4	95.8	57.8	86.5	50.4	71.6	81.9	138.2	110.1	115.6	152.2	157.6	78.8	98.8	164.0	310.3
LTD	42.3	51.3	56.5	68.6	32.3	60.5	70.5	103.5	85.8	109.3	91.8	87.4	65.0	**113.6**	113.4	220.6
ST-GCN	73.4	95.8	57.8	86.5	50.4	71.6	81.9	138.2	110.1	115.6	152.2	157.6	78.8	98.8	164.0	310.3
MSR	42.1	43.5	57.0	71.5	35.2	62.5	75.4	113.5	**78.5**	**101.7**	100.1	95.1	63.7	113.9	**103.0**	219.9
Transformer	36.8	**41.2**	58.4	**67.9**	**29.2**	58.3	74.0	103.1	-	-	-	-	-	-	-	-
Ours	**35.9**	43.9	**55.7**	69.5	33.1	**58.1**	**69.9**	**99.9**	83.7	105.3	**90.7**	**87.1**	**62.1**	115.6	104.3	**209.3**
scenarios	purchases		sitting		sittingdown		takingphoto		waiting		walkingdog		walkingtogether		average	
millisecond	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms	560ms	1000ms
Res. Sup.	115.8	159.4	161.6	195.3	214.5	285.2	117.9	141.1	152.9	199.1	196.8	213.3	107.8	136.5	137.5	168.2
DMGNN	118.8	154.5	59.7	104.3	122.0	168.8	91.2	120.6	106.1	136.6	194.1	182.2	83.5	115.8	102.9	137.1
LTD	94.3	130.4	**79.6**	114.9	82.6	140.1	**68.9**	**87.1**	100.9	167.6	136.6	**174.3**	57.0	85.0	78.5	114.3
ST-GCN	118.8	154.5	59.7	104.3	122.0	168.8	91.2	120.6	106.1	136.6	194.1	182.2	83.5	115.8	102.9	137.1
MSR	**86.5**	125.5	83.1	**103.9**	83.1	145.8	72.6	95.9	100.7	**164.3**	144.4	193.5	55.8	84.5	78.7	115.7
Transformer	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ours	89.7	**122.9**	81.0	115.8	**80.2**	**130.8**	70.3	90.5	**94.5**	168.1	137.8	180.8	**54.6**	**80.3**	**76.2**	**111.9**

结　论

本文主要是展示如何使用修改“祖传模板”得到的新模板，在使用时直接替换成自己的论文内容即可。总结下来最最最麻烦的是科学上网，只有科学上网才能获取文献信息生成 bib 文件，后面就好办了。

本模板难免有不足之处，主要是我本人的论文涉及的格式有限，有些地方没探索到自然就没去设置。比如附录，附录的图文并茂等等，我本人是没有研究的，这里仅仅做了一些初步的工作，不过对很多同学来说本模板是够用的。希望有能帮助到华工的同学们，有不足之处请多多理解，可以通过邮件联系我，上班之余我会尽量回复。

本模板会一直更新——2022-5-28

参考文献

[1] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. ArXiv preprint arXiv:1409.2329, 2014.

[2] Shi X, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. Advances in neural information processing systems, 2015, 28.

[3] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. ArXiv preprint arXiv:1406.1078, 2014.

[4] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. ArXiv preprint arXiv:1609.02907, 2016.

[5] Fragkiadaki K, Levine S, Felsen P, et al. Recurrent network models for human dynamics[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 4346-4354.

[6] Jain A, Zamir A R, Savarese S, et al. Structural-rnn: Deep learning on spatio-temporal graphs[C]//Proceedings of the ieee conference on computer vision and pattern recognition. 2016: 5308-5317.

[7] Ghosh P, Song J, Aksan E, et al. Learning human motion models for long-term predictions[C]//2017 International Conference on 3D Vision (3DV). 2017: 458-466.

[8] Martinez J, Black M J, Romero J. On human motion prediction using recurrent neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2891-2900.

[9] Gui L Y, Wang Y X, Liang X, et al. Adversarial geometry-aware human motion prediction[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 786-803.

[10] Tang Y, Ma L, Liu W, et al. Long-term human motion prediction by modeling motion context and enhancing motion dynamic[J]. ArXiv preprint arXiv:1805.02513, 2018.

[11] Gui L Y, Wang Y X, Ramanan D, et al. Few-shot human motion prediction via meta-learning[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 432-450.

[12] Guo X, Choi J. Human motion prediction via learning local structure representations and temporal dependencies[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 2580-2587.

[13] Liu Z, Wu S, Jin S, et al. Towards natural and accurate future motion prediction of humans and animals[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 10004-10012.

[14] Chiu H k, Adeli E, Wang B, et al. Action-agnostic human pose forecasting[C]//2019 IEEE Winter Conference on Applications of Computer Vision (WACV). 2019: 1423-1432.

[15] Gopalakrishnan A, Mali A, Kifer D, et al. A neural temporal model for human motion prediction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 12116-12125.

[16] Sang H F, Chen Z Z, He D K. Human Motion prediction based on attention mechanism[J]. Multimedia Tools and Applications, 2020, 79(9): 5529-5544.

[17] Corona E, Pumarola A, Alenya G, et al. Context-aware human motion prediction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 6992-7001.

[18] Pavllo D, Feichtenhofer C, Auli M, et al. Modeling human motion with quaternion-based neural networks[J]. International Journal of Computer Vision, 2020, 128(4): 855-872.

[19] Aksan E, Kaufmann M, Hilliges O. Structured prediction helps 3d human motion modelling[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7144-7153.

[20] Mao W, Liu M, Salzmann M, et al. Learning trajectory dependencies for human motion prediction[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 9489-9497.

[21] Mao W, Liu M, Salzmann M. History repeats itself: Human motion prediction via motion attention[C]//European Conference on Computer Vision. 2020: 474-489.

[22] Cui Q, Sun H, Yang F. Learning dynamic relationships for 3d human motion prediction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 6519-6527.

[23] Li M, Chen S, Zhao Y, et al. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 214-223.

[24] Li M, Chen S, Chen X, et al. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[25] Li B, Tian J, Zhang Z, et al. Multitask Non-Autoregressive Model for Human Motion Prediction[J]. IEEE Transactions on Image Processing, 2020, 30: 2562-2574.

[26] Liu J, Yin J. Multi-grained Trajectory Graph Convolutional Networks for Habit-unrelated Human Motion Prediction[J]. ArXiv preprint arXiv:2012.12558, 2020.

[27] Lebailly T, Kiciroglu S, Salzmann M, et al. Motion Prediction Using Temporal Inception Module[C]//Proceedings of the Asian Conference on Computer Vision. 2020.

[28] Dang L, Nie Y, Long C, et al. MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11467-11476.

[29] Cui Q, Sun H. Towards Accurate 3D Human Motion Prediction From Incomplete Observations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 4801-4810.

[30] Shi L, Wang L, Long C, et al. Social Interpretable Tree for Pedestrian Trajectory Prediction[C]//AAAI Conference on Artificial Intelligence. 2022.

[31] Shi L, Wang L, Long C, et al. SGCN: Sparse Graph Convolution for Pedestrian Trajectory Prediction[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2021.

[32] Duan J, Wang L, Long C, et al. Complementary Attention Gated Network for Pedestrian Trajectory Prediction[C]//AAAI Conference on Artificial Intelligence. 2022.

[33] Butepage J, Black M J, Kragic D, et al. Deep representation learning for human motion prediction and classification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 6158-6166.

[34] Li C, Zhang Z, Lee W S, et al. Convolutional sequence to sequence model for human dynamics[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 5226-5234.

[35] Liu X, Yin J, Liu J, et al. TrajectoryCNN: a new spatio-temporal feature learning network for human motion prediction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020.

[36] Barsoum E, Kender J, Liu Z. Hp-gan: Probabilistic 3d human motion prediction via gan[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 1418-1427.

[37] Kundu J N, Gor M, Babu R V. Bihmp-gan: Bidirectional 3d human motion prediction gan[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 8553-8560.

[38] Hernandez A, Gall J, Moreno-Noguer F. Human motion prediction via spatio-temporal inpainting[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7134-7143.

[39] Jain D K, Zareapoor M, Jain R, et al. GAN-Poser: an improvised bidirectional GAN model for human motion prediction[J]. Neural Computing and Applications, 2020, 32(18): 14579-14591.

[40] Liu Z, Lyu K, Wu S, et al. Aggregated multi-gans for controlled 3d human motion prediction[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 35: 3. 2021: 2225-2232.

[41] Cui Q, Sun H, Kong Y, et al. Efficient human motion prediction using temporal convolutional generative adversarial network[J]. Information Sciences, 2021, 545: 427-447.

[42] Chao X, Bin Y, Chu W, et al. Adversarial refinement network for human motion prediction[C]//Proceedings of the Asian Conference on Computer Vision. 2020.

[43] Lyu K, Liu Z, Wu S, et al. Learning human motion prediction via stochastic differential equations[C]//Proceedings of the 29th ACM International Conference on Multimedia. 2021: 4976-4984.

[44] Liu Z, Wu S, Jin S, et al. Investigating pose representations and motion contexts modeling for 3D motion prediction[J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 45(1): 681-697.

[45] Oord A v d, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J]. ArXiv preprint arXiv:1609.03499, 2016.

[46] Yu F, Koltun V, Funkhouser T. Dilated residual networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 472-480.

[47] Pavllo D, Grangier D, Auli M. Quaternet: A quaternion-based recurrent model for human motion[J]. ArXiv preprint arXiv:1805.06485, 2018.

[48] Li Y, Wang Z, Yang X, et al. Efficient convolutional hierarchical autoencoder for human motion prediction[J]. The Visual Computer, 2019, 35: 1143-1156.

[49] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition[C]//Thirty-second AAAI conference on artificial intelligence. 2018.

[50] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. 2015: 234-241.

[51] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[52] Aksan E, Kaufmann M, Cao P, et al. A spatio-temporal transformer for 3d human motion prediction[C]//2021 International Conference on 3D Vision (3DV). 2021: 565-574.

[53] Cai Y, Huang L, Wang Y, et al. Learning progressive joint propagation for human motion prediction[C]//European Conference on Computer Vision. 2020: 226-242.

[54] Yu J, Lin Z, Yang J, et al. Generative image inpainting with contextual attention[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 5505-5514.

[55] Zamir S W, Arora A, Khan S, et al. Multi-stage progressive image restoration[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 14821-14831.

[56] Sofianos T, Sampieri A, Franco L, et al. Space-Time-Separable Graph Convolutional Network for Pose Forecasting[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11209-11218.

[57] Ionescu C, Papava D, Olaru V, et al. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 36(7): 1325-1339.

[58] Von Marcard T, Henschel R, Black M J, et al. Recovering accurate 3d human pose in the wild using imus and a moving camera[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 601-617.

攻读博士/硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况**（只填写与学位论文内容相关的部分）：**

序号	作者（全体作者，按顺序排列）	题目	发表或投稿刊物名称、级别	发表的卷期、年月、页码	与学位论文哪一部分（章、节）相关	被索引收录情况
1	蒙超恒、裴海龙、程子欢	涵道风扇式无人机的优先级控制分配	航空学报	已录用，2020年5月	2.1、2.2、3.4、4.1、4.2、5.1和5.3节	EI
2	蒙超恒、裴海龙、程子欢	Dynamic Control Allocation for A Twin Ducted Fan UAV	2020 International Conference on Guidance, Navigation and Control	已录用，2020年8月	2.3、4.3和5.2节	EI

注：在“发表的卷期、年月、页码”栏：

1. 如果论文已发表，请填写发表的卷期、年月、页码；

2. 如果论文已被接受，填写将要发表的卷期、年月；

3. 以上都不是，请据实填写“已投稿”，“拟投稿”。

不够请另加页。

二、与学位内容相关的其它成果（包括专利、著作、获奖项目等）

致　谢

这次你离开了没有像以前那样说再见, 再见也他妈的只是再见
我们之间从来没有想象的那么接近, 只是两棵树的距离
你是否还记得山阴路我八楼的房间, 房间里唱歌的日日夜夜
那么热的夏天你看着外面, 看着你在消逝的容颜
我多么想念你走在我身边的样子, 想起来我的爱就不能停止
南京的雨不停地下不停地下, 就像你沉默的委屈
一转眼, 我们的城市又到了夏天, 对面走来的人都眯着眼
人们不敢说话不敢停下脚步, 因为心动常常带来危险
我多么想念你走在我身边的样子, 想起来我的爱就不能停止
南京的雨不停地下不停地下, 有些人却注定要相遇
你是一片光荣的叶子，落在我卑贱的心
像往常一样我为自己生气并且歌唱
那么乏力, 爱也吹不动的叶子

蒙超恒

2020 年 7 月 10 日

于华南理工大学