



上海交通大学硕士学位论文

基于面部表情、语音、头部姿态的在线实时数 字人手势驱动

申请学位：工学硕士

学科/专业：计算机科学与技术

院 系：计算机学院

2026 年 1 月 12 日

**A Dissertation Submitted to
Shanghai Jiao Tong University for the Degree of Master**

**REAL-TIME ONLINE DIGITAL HUMAN GESTURE
DRIVE BASED ON FACIAL EXPRESSION, SPEECH,
AND HEAD POSE**

School of Computer Science
Shanghai Jiao Tong University
Shanghai, P.R. China
January 12th, 2026

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全知晓本声明的法律后果由本人承担。

学位论文作者签名：

日期： 年 月 日

上海交通大学

学位论文使用授权书

本人同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于：

- 公开论文
 - 内部论文，保密 1年 / 2年 / 3年，过保密期后适用本授权书。
 - 秘密论文，保密 ____ 年（不超过 10 年），过保密期后适用本授权书。
 - 机密论文，保密 ____ 年（不超过 20 年），过保密期后适用本授权书。
- （请在以上方框内选择打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日 日期： 年 月 日

摘要

手势是人类交流中用于补充语义细节与传递情绪的重要非语言信号。在虚拟人/数字形象的实时控制中，传统动作捕捉往往依赖穿戴式设备，虽然精度高，但存在成本高、使用门槛高、便携性差等问题；相机动捕虽降低了硬件负担，却仍要求用户在镜头前持续表演手势，在远程交流、直播等长时间场景中会带来空间占用与体力消耗，并与键鼠操作产生冲突。近年来语音驱动手势生成技术为低门槛驱动提供了可能，但现有主流方法通常假设可获取完整语句或未来上下文，难以直接应用于用户逐字语音输入的在线实时交互环境，因此需要一种低延迟、可部署的实时手势生成方案。

为解决上述问题，本论文提出 FaceCapGes：一种面向实时数字人驱动的帧级手势生成方法，仅使用可在线采集的三类信号：语音、面部表情与头部姿态，在不依赖未来输入的条件下逐帧生成上半身 3D 骨骼动作，从而让用户无需实际做出手势即可驱动虚拟形象获得自然的随语表达。本文的主要研究内容包括：

(1) 提出 FaceCapGes 在线实时多模态手势生成框架，给出从信号采集、帧级推理到虚拟人驱动渲染的整体流程设计，并在严格因果约束下明确其可部署的实时生成机制。

(2) 在级联多模态架构中引入头部姿态作为新的实时输入模态，设计并实现姿态编码模块与弱耦合融合策略，为手势提供额外的节奏与朝向线索，从而增强动作节奏与空间指向一致性。

(3) 提出并实现在线实时手势生成的滑动窗口自回归训练策略，通过片段切割与窗口展开实现严格因果的逐帧生成，并结合片段级监督与对抗训练目标，提升实时生成的稳定性与时间连续性，缓解自回归漂移与抖动问题。

(4) 开展用户主观实验、客观指标测量与实时性能测试，并与代表性方法进行对比评估。实验结果表明，在严格因果约束下，本文方法 FaceCapGes 在真实性上表现良好，与现有扩散模型相当；同时在语调变化较快与头部朝向变化明显的片段中，能够生成更平滑且指向一致的动作，并具备满足实时交互场景需求的推理性能。

关键词：协同语音手势生成，数字人驱动，面部捕捉，多模态学习

Abstract

Gestures are essential nonverbal signals in human communication, enriching semantic details and conveying affective states. In real-time control of virtual humans or digital avatars, conventional motion capture systems typically rely on wearable devices. Although accurate, such systems are costly, inconvenient to use, and lack portability. Camera-based capture reduces hardware requirements but still forces users to continuously perform gestures in front of a camera, which occupies physical space, causes fatigue during long-term scenarios such as remote communication and live streaming, and conflicts with keyboard–mouse interaction. Recently, speech-driven gesture generation has emerged as a low-barrier alternative; however, most existing approaches assume access to complete utterances or future context, making them difficult to deploy in online interactive settings where speech is streamed word by word. Therefore, a low-latency and deployable real-time gesture generation solution is required.

To address these challenges, this thesis proposes FaceCapGes, a frame-level gesture generation method for real-time avatar driving. FaceCapGes uses only three types of online-available signals—speech, facial expressions, and head pose—to generate upper-body 3D skeletal motions frame by frame without relying on future inputs, enabling users to drive avatars with natural co-speech gestures without physically performing them. The main contributions of this thesis are summarized as follows:

(1) A real-time multimodal gesture generation framework, FaceCapGes, is proposed, providing an overall pipeline design from signal acquisition and frame-level inference to avatar driving and rendering, and clarifying a deployable real-time generation mechanism under strict causality constraints.

(2) We introduce head pose as a new real-time input modality in a cascaded multimodal architecture. A pose encoder and a weakly coupled fusion strategy are designed and implemented to provide additional rhythmic and orientation cues, improving gesture timing and spatial pointing consistency.

(3) A sliding-window autoregressive training strategy for online real-time gesture generation is proposed and implemented. By segmenting motion sequences and unfolding them with a sliding window, strictly causal frame-wise generation is achieved. Combined

with segment-level supervision and adversarial objectives, the proposed strategy improves temporal stability and continuity, alleviating autoregressive drift and jitter.

(4) Subjective user studies, objective metric evaluation, and real-time performance tests are conducted, with comparative assessments against representative methods. The results demonstrate that, under strict causality constraints, FaceCapGes achieves strong realism comparable to existing diffusion-based models; moreover, in segments with rapid prosodic variations and noticeable head orientation changes, it generates smoother motions with more consistent spatial pointing, while maintaining inference efficiency sufficient for real-time interactive applications.

Key words: co-speech gesture generation, virtual avatar driving, face-capture, multimodal learning

目 录

第 1 章 引言	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 手势分类体系与理论框架	2
1.2.2 规则驱动阶段	4
1.2.3 数据驱动阶段	4
1.2.4 多模态扩展阶段	5
1.2.5 手势生成方法的总结	5
1.3 本文研究目标与研究内容.....	6
1.4 论文组织架构.....	7
第 2 章 虚拟人驱动的多模态在线实时手势生成框架	9
2.1 需求分析.....	9
2.2 训练数据来源.....	10
2.2.1 骨架拓扑	10
2.2.2 头部姿态参数	11
2.2.3 面部参数	12
2.3 旋转参数的选取.....	13
2.4 在线实时手势生成系统整体框架.....	14
2.4.1 信号采集与系统配置层	15
2.4.2 手势生成模型层	16
2.4.3 渲染驱动层	16
2.5 本章小结.....	17
第 3 章 融合头部姿态的多模态级联手势生成模型架构	19
3.1 问题定义.....	19
3.1.1 任务描述	19
3.1.2 输入与输出模态	19

3.2 级联架构设计的继承.....	20
3.2.1 级联架构的原理与理论背景	20
3.2.2 说话人条件输入的简化	20
3.2.3 输入模态的编码	21
3.2.4 身体姿态的解码	21
3.3 头部姿态模态在级联架构中的引入.....	23
3.4 模型整体结构.....	24
3.5 本章小结.....	24
第 4 章 基于滑动窗口的实时手势生成自回归训练	25
4.1 单步因果 LSTM 预测器	25
4.1.1 LSTM 基本概念与状态传递机制.....	25
4.1.2 双时间尺度记忆结构	26
4.1.3 窗口内预测过程的形式化表达	27
4.2 训练片段切割.....	27
4.2.1 固定长度片段定义	27
4.2.2 重叠切割与样本覆盖	28
4.3 片段内部的滑动窗口展开策略.....	28
4.3.1 前置动作帧与预热阶段	28
4.3.2 滑动窗口展开与逐帧自回归生成	29
4.3.3 拼接生成序列与片段级输出	29
4.4 监督损失.....	30
4.4.1 片段级生成序列与损失计算范围	30
4.4.2 总体优化目标	30
4.4.3 姿态重构与时序平滑损失	30
4.4.4 损失权重设置	31
4.5 对抗训练.....	31
4.5.1 基于拼接片段的片段级判别	31
4.5.2 对抗损失定义	32
4.5.3 交替优化策略	32
4.6 本章小结.....	32

第 5 章 实验结果与分析	34
5.1 训练配置.....	34
5.2 实验配置.....	35
5.2.1 实验对比模型	35
5.2.2 跨模型评估设置	35
5.3 用户评估.....	36
5.3.1 用户评估系统与实验配置	36
5.3.2 结果与分析	39
5.4 定性分析.....	41
5.4.1 生成动作平滑性	41
5.4.2 头部朝向与手势空间指向一致性	41
5.5 客观评估指标与实现细节.....	44
5.5.1 手势分布相似度 (Fréchet Gesture Distance)	44
5.5.2 语义相关动作召回率 (Semantic Relevance Gesture Recall)	45
5.5.3 节奏对齐度 (Beat Alignment)	46
5.5.4 L1 范数	47
5.5.5 评估区域设定与公平性说明	48
5.6 定量评估结果.....	48
5.7 消融实验分析.....	49
5.8 性能评估.....	50
5.8.1 单帧推理性能	50
5.8.2 端到端计算链路延迟	51
5.8.3 系统更新率	52
5.8.4 帧率设定的可扩展性	52
5.9 本章小结.....	53
第 6 章 结论	54
6.1 本文工作总结.....	54
6.2 未来工作展望.....	55
6.2.1 高层语义信息	55
6.2.2 面向未来趋势的预测性训练目标	55

参考文献.....	57
附录 A 手势生成对比视频	60
附录 B 代码与实现资源	61
学术论文和科研成果目录.....	62

插 图

图 2.1 BEAT 数据集的骨架拓扑结构与驱动范围	11
图 2.2 FACS 中闭眼动作单元的定义 ^[22]	12
图 2.3 BlendShape 在闭眼形变上的线性插值效果.....	13
图 2.4 系统整体架构与数据流示意图.....	15
图 3.1 头部姿态编码器结构示意图.....	24
图 3.2 FaceCapGes 模型整体结构	24
图 5.1 用户评估工具实机界面.....	38
图 5.2 用户评估总体主观排名结果.....	39
图 5.3 平衡拉丁方设置下的主观排名结果.....	40
图 5.4 生成动作效果对比.....	42
图 5.5 头部朝向与手势指向一致性的动作对比.....	43

表 格

表 1.1 两种虚拟人手势生成任务目标的设定差异.....	5
表 3.1 输入输出模态符号与维度.....	20
表 5.1 对比模型的输入输出模态.....	35
表 5.2 定量评估结果.....	48
表 5.3 消融实验结果.....	49
表 5.4 推理速度评估结果.....	51

第1章 引言

1.1 研究背景和意义

近年来，随着元宇宙、虚拟社交与直播等领域的相关技术日趋成熟，用户已能够使用任意外观的虚拟人作为交互载体，在虚拟空间中与异地用户进行交流。虚拟人3D模型的姿态由其内部骨架关节的旋转参数（如欧拉角、四元数等）定义，最终通过蒙皮渲染技术完成可视化。得益于自动骨骼绑定技术，骨骼动画的生成可消除不同3D模型间的骨架拓扑差异，实现跨模型复用。

在虚拟人交互中，穿戴式动作捕捉设备是实时驱动手势的传统方案，贴合肢体的标记点可将肢体运动实时转换为相同的骨骼动画，提供直观、准确的操控。尽管其精度较高，但对于大多数用户而言，此类设备存在功能用途单一、硬件成本昂贵、便携性差等问题，限制了使用频率。因此，在当前的虚拟人交互应用中，仅少数专业用户会使用此类设备，而多数用户在设备限制下无法简单控制虚拟人肢体，导致了两种用户体验之间的不一致。

针对普通用户对低门槛虚拟人交互的需求，基于相机的动作捕捉技术^[1-2]成为主流替代方案。该技术无需额外硬件，仅通过手机或电脑的内置相机即可实时捕捉用户动作，用于转化为骨骼动画。然而，该方案仍存在三种局限：一是这种方法要求用户面向相机做出手势，过程中难以同步操作键盘、鼠标，造成操作冲突；二是手势活动范围受相机视野限制，在自然使用距离通过手机或电脑的内置相机拍摄用户，将严重限制手势的捕捉范围；三是持续的手势动作会产生体力消耗，在直播等长时间使用场景中，用户疲劳问题将变得显著。

基于上述问题，本文提出一种面向真实用户交互的需求：在无需用户实际做出手势的前提下，仅利用实时语音、面部捕捉与头部姿态信号，在线生成与语音匹配的上半身手势骨骼动画，从而降低使用门槛并缓解操作冲突与疲劳问题。

然而，现有研究尚难以直接满足该需求。其一，许多手势生成方法依赖完整的语音或文本输入，而真实交互中的语音通常以流式方式逐步输入；若等待未来输入以获得更完整语义，将不可避免地引入端到端延迟。其二，尽管头部姿态与手势节奏及空间指向存在明确关联，且可由常规相机实时捕捉，但将头部姿态作为输入特征以提升在线生成自然度的研究仍相对不足。

为此，本文将提出一种新颖的实时手势生成模型。该模型以帧为单位，输入语音、

面部表情与头部姿态数据，并逐帧输出对应的骨骼动画。此外，本文在实时手势生成中，尝试将头部姿态作为一种新的模态引入，利用其与自然手势节奏与朝向的高度相关性，结合语音和面部信息共同提升生成动作的自然度。本文采用级联多模态架构与自回归训练来融合这些模态，以学习其联合表征，从而在严格的实时约束下增强手势的表现力。

因此，下面将从语音驱动手势生成的发展脉络出发，总结现有方法，并指出在线实时手势生成仍待解决的问题。

1.2 国内外研究现状

1.2.1 手势分类体系与理论框架

早期语言学与手势研究对人类交流手势建立了较为系统的分类框架。一般而言，手势是与语音共同出现的身体动作信号，可承担语义补充、情感表达与互动调节等多重功能。这类动作通常具有明确的交际意图，从而区别于维持平衡、行走等以完成物理任务为主的功能性动作^[3]。在 Kendon 等人的理论框架下，手势与语言并非两个相互独立的系统，而是被视为共享认知与表达过程的协同产物。

需要说明的是，本文在广义层面使用“手势”一词时，其运动形式不局限于手部动作，也可扩展至头部运动与躯干姿态等与表达相关的上半身动作。

1.2.1.1 Kendon 连续体

Kendon 连续体^[3-4] 从语言化/符号化程度的连续变化角度，将交际性手势置于以下谱系中：

- 随语手势（gesticulation）
- 语言样手势（language-like gestures）
- 拟态/哑剧式动作（pantomime）
- 约定俗成的象征手势（emblems）
- 手语（sign language）

此连续体中，越靠左，通常更依赖当前的言语与语境、形式更即兴；越靠右，越接近离散的符号系统，规约化程度高，意义更稳定，可在缺少口语的情况下独立传达。

当前人机交互、虚拟人/数字人驱动等方向的手势生成，多数工作聚焦于 Kendon 连续体最左侧的随语手势，即说话过程中自然出现、与语音节奏与语义强关联的上肢动作。其含义往往依赖当前的口语与语境，脱离它们时通常难以传达清晰含义。

相对而言，连续体右侧的手势更接近文化中约定俗成的符号，可以脱离口语独立传达含义。例如，表达称赞的拍手动作，或表示“请保持安静”的嘴前竖起食指的动作。这些被视为象征手势，通常不在手势生成领域的研究对象中。

因此，本文将研究范围限定为随语手势的学习与生成。

1.2.1.2 随语手势的分类

McNeill^[4]将随语手势进一步划分为四种基本类型：

(1) 形象性手势 (Iconic gestures): 以具象方式描绘事物的外形、空间路径或动作特征。例如，用手势勾勒一个物体的轮廓，或划出一道线表示移动的轨迹。此类手势与语言内容直接对应，表达具体语义。

(2) 隐喻性手势 (Metaphoric gestures): 表达抽象概念或思维结构的手势，比如用双手做出捧起一个物体放到一边的动作，表示“先把这个话题放一边”。这种手势并不描绘实体，而是以具象化的方式呈现抽象语义。

(3) 指示性手势 (Deictic gestures): 指向空间中的对象、人物或方向，常用于对话焦点的指明与注意引导。

(4) 节奏型手势 (Beat gestures): 与语音重音、韵律或节奏同步的节奏性动作，通常不承载具体语义，但可用于强调语音节奏，引起听众对说话内容的注意。

这四类手势构成了随语手势在语义与语篇功能上的主要维度，并在自然交流中常以复合形式出现。在手势生成任务中，研究通常将其视为不同的可生成目标：其中节奏型手势由于与语音韵律高度同步、对齐与建模相对容易，长期以来在数据驱动方法中更常被优先刻画；而形象性与隐喻性等语义相关手势则对文本的语义推理有更高要求，因而是更具挑战性的方向。

鉴于本文以低延迟在线驱动为目标，本文优先建模与韵律强耦合的节奏型手势；语义一致的形象性、隐喻性手势留作后续工作。

1.2.1.3 头部手势的分类

除手部动作外，头部动作同样是手势的重要组成部分。头部的点动与摆动在时间结构上常与手势及语音节奏保持同步^[5]，在语用功能上既能辅助语音韵律的组织，也能表达态度与指向信息。

在不同研究中^[3-4,6]，头部动作被从多个维度加以分析，其主要功能可归纳为以下几方面：

(1) 韵律相关 (prosodic): 动作反映语音重音与句法节奏的对应关系；

(2) 语义或态度相关 (semantic/attitudinal): 动作表达说话者的情绪倾向与交际

意图；

(3) 指向相关 (deictic): 动作通过转头或注视方向建立叙事空间的参照。

此外，研究表明头部动作的启动时间往往早于发声^[7]。具体而言，头部动作存在启动与加速过程，若其峰值需与重读音节的时间对齐，则动作必须提前起势。因此，头部动作可能对即将到来的语音韵律具有前瞻性。

这一特征揭示了头部动作与语音之间的时序关系，说明视觉模态中的运动信号有时可先于声学事件出现。本文研究也因此关注头部动作，并将其纳入输入模态。

基于上述对随语手势的界定，研究者开始尝试构建能够自动生成此类动作的系统。

1.2.2 规则驱动阶段

早期的手势生成系统主要依赖语言学规则与专家知识构建^[5,8-10]。这类方法通过语义分类或韵律规则将语音片段映射为预定义的手势模板（如指示、肯定、节奏性动作），并以有限的动作库组合出手势序列。它们可在虚拟代理或机器人中实现基于语音的同步动作。然而，手势词典与语法规则的人工设计成本较高，难以覆盖自然语音中的多样变化，导致生成结果缺乏自然性与个体差异。

1.2.3 数据驱动阶段

随着大规模语音与动作配对数据的出现，研究者开始采用统计学习和深度神经网络模型学习语音与手势的映射关系。在此阶段，语音通常作为唯一输入模态，模型通过长短期记忆网络（Long Short-Term Memory, LSTM）、多层感知机（Multilayer Perceptron, MLP）等结构预测连续手势序列。

为解决语音与手势间的多对多映射问题，研究者引入了向量量化变分自编码器（Vector Quantized Variational AutoEncoder, VQ-VAE）^[11-13]与扩散模型^[14-18]，在保持自然性的同时提升了生成多样性与表现力。

尽管上述深度生成模型在客观指标与视觉效果上优于传统模型，但通常依赖完整语句级上下文。在用户实时语音下的流式逐字输入场景中，为获取未来上下文以进行语义判别与韵律对齐，需引入缓冲机制，因此即使推理较快的模型^[15]，整体延迟也因上下文缓冲造成端到端的显著延迟。

1.2.4 多模态扩展阶段

为进一步提升动作表现力与语音理解能力,部分研究引入视觉模态或语言语义特征。例如, CaMN^[19]在语音输入的基础上融合面部捕捉信息以增强表现; EMAGE^[11]与 DiffSHEG^[15]同时生成手势与面部动作; DiffTED^[17]实现了端到端的视频合成。

这些方法在生成质量上取得了显著进展,但其多数仍依赖整句级输入与离线生成设定,为后续在线实时部署带来了新的挑战。

1.2.5 手势生成方法的总结

现有研究中,大量方法默认采用离线生成设定,即假设可访问完整语句级语音或文本上下文。这有利于模型获得更稳定的语义意图、能够在生成时进行全局时序安排,生成更连贯、语义一致性更好的动作序列。此外,在面向AI虚拟形象的对话系统或内容合成任务中,由于系统通常可以在AI开始发言前获得完整文本或语音,生成过程也不必严格满足实时性约束,因此,离线手势生成能够以较低的工程成本为AI虚拟角色提供高质量的动作表现。

相比之下,面向真实用户交互的在线实时场景中,系统只能获得流式语音输入并需逐帧输出动作,无法直接沿用依赖完整上下文的离线假设。我们在表1.1中,整理了两个任务的目标、可用条件与输入模态的差异。

表1.1 两种虚拟人手势生成任务目标的设定差异

Table 1.1 Setting Differences Between Two Virtual-Human Gesture Generation Objectives

对比维度	AI虚拟形象生成	用户虚拟人实时生成
输入信息	完整句级语音或文本(可使用未来信息)	实时语音流,仅使用过去与当前帧信息
输出目标	整句手势序列(离线生成)	连续流式手势(逐帧生成)
延迟影响	较低	较高,对用户交互体验影响大
应用场景	AI虚拟人、离线动画、内容合成	虚拟世界交互、视频会议、虚拟直播
语音模态	作为输入或由文本生成	作为实时输入特征
手部手势	生成目标(输出)	生成目标(输出)
面部表情	通常为生成目标(输出)	可通过设备实时采集
头部姿态	通常为生成目标(输出)	可通过设备实时采集

由于面向用户交互的在线实时手势生成中,用户的语音有逐字输入的特点,系统需根据当前输入流即时生成同步手势,不能查看未来信息,同时无法立即获得当前的语义。这意味着本文需要设计一个新的方法,让模型能够在在线实时的输入流中生成

当前的身体姿态。

此外，在此目标下，头部姿态不再是生成目标，而是可实时采集的输入特征，但目前还没有头部姿态作为输入特征的充分讨论。本文认为，引入头部姿态将对在线实时手势生成有重要贡献，原因如下：

时间前瞻性 之前的研究表明，头部动作在自然语音中常呈现出一定的时间前瞻性^[7]：其启动往往早于对应韵律词的发声，这意味着视觉模态可能比声学信号更早反映语音节奏的变化趋势。这种时序特性为实时生成任务提供了潜在的预测窗口，使系统能够在语音节奏变化尚未显现前，提前捕获相关的动态线索。因此，头部姿态在实时生成中不仅提供同步参考，也可能在时间上形成前驱信号，为手势节奏的自然启动提供时序优势。

反映互动焦点 头部姿态模态为实时语音驱动的手势生成提供了空间参考信号，即使在无语义信息的条件下，头部的转向与注视变化仍能反映说话者的注意焦点与手势朝向，从而帮助模型在动作生成中保持方向一致性。这一机制使用户在沉浸式虚拟环境中，更容易将虚拟人手势对齐叙事对象，让生成的手势在视觉上更具互动感与表达意图。

总体而言，头部姿态为实时生成提供了介于韵律与语义之间的关键中层约束。其时间上的前瞻性与空间上的指向性共同帮助模型在低延迟条件下保持自然、连贯且空间协调的动作表现，从而在因果生成框架内有效拓展了语音驱动手势的可表达范围，并为节奏型手势的实时生成提供了支持。

1.3 本文研究目标与研究内容

本文旨在面向实时数字人交互场景，研究一种可部署的在线随语手势生成方法。与离线生成任务不同，实时交互系统要求生成过程满足严格因果约束，即模型在任意时刻仅可利用当前及历史可观测信号进行推理，而无法依赖未来语音。同时，系统还需具备低延迟的性能要求，以支持逐帧驱动虚拟人并保持自然的随语手势表达效果。

在上述在线实时约束下，生成的随语手势不仅需要在形式上可行，还需在多个质量维度上满足实际交互需求，其中包括：

(1) 自然度与运动学合理性：生成动作在速度、加速度、关节角度范围与时序连续性等方面应符合人体运动规律，并在整体观感上保持自然流畅，避免高频抖动与突

变等不连续现象。

(2) 语音—动作同步性：生成动作的节奏变化需与语音韵律（如重音、停顿、能量变化）在时间轴上保持对齐，尤其体现在强调、停顿等处的动作起止与峰值对齐。

(3) 多样性：在不同输入条件（语音韵律/说话风格）下，生成结果应具备足够的变化性，避免动作模板化。

(4) 语义相关性与表达一致性：生成动作应与语音的语义内容、情感倾向及交互意图保持一致。

(5) 实时性：模型推理延迟需满足实时交互要求。

为满足上述多重约束，本文提出 FaceCapGes：一种仅依赖可在线采集信号的帧级多模态随语手势生成框架。该框架以语音、面部表情与头部姿态为输入，在严格因果条件下逐帧生成上半身 3D 骨骼动作，使用户无需真实做出手势即可驱动虚拟形象产生自然的随语手势。

围绕上述目标，本文的研究内容汇总如下：

(1) 提出面向实时交互的在线随语手势生成任务定义与系统框架，构建从多模态信号采集、流式推理到虚拟人驱动渲染的端到端流程，并实现可部署的实时数字人驱动系统。

(2) 设计基于语音、面部表情与头部姿态的在线多模态动作生成结构，引入适用于实时场景的头部姿态特征编码方法，并结合 CaMN 级联解码策略实现多模态融合建模。

(3) 提出满足严格因果约束的自回归训练策略，通过片段切割与滑动窗口展开实现帧级流式学习，并结合单向时序解码器与历史动作缓冲机制提升生成动作的连续性与平滑性。

(4) 构建统一的数据处理、推理与渲染评估平台，开展用户主观实验、客观指标测量与实时性能测试，并与代表性方法进行对比评估，以验证本文方法的有效性与性能。

1.4 论文组织架构

本文章节安排如下：

第一章陈述在线实时数字人驱动场景下，语音驱动手势生成的研究背景与意义，总结国内外研究现状并对比离线生成与在线生成任务的差异，在此基础上明确本文的研究目标与主要贡献；

第二章围绕在线实时手势生成系统展开，先进行需求分析，再说明多模态数据来源与参数化表示，并给出端到端系统框架与模块划分；

第三章介绍本文多模态级联手势生成模型 FaceCapGes，包括头部姿态模态的引入方式、编码器结构设计以及身体姿态的层次化解码过程；

第四章介绍本文模型的训练方法，提出滑动窗口自回归训练与推理一致性策略，给出片段切割、窗口展开、监督损失与对抗训练目标的定义；

第五章介绍本文模型的主观与客观评估，并从定性分析、消融实验与性能测试等角度验证本文方法在生成质量与实时性方面的优势；

第六章总结全文工作并讨论未来研究方向。

第2章 虚拟人驱动的多模态在线实时手势生成框架

本文面向的应用场景是低门槛的单设备虚拟人驱动：系统在用户侧仅依赖一台具备麦克风与前向摄像头的通用设备（如个人电脑或智能手机），即可实时获取语音信号以及由面部视频解析得到的面部参数与头部姿态，并进一步生成与语音表达一致的上半身手势序列用于虚拟人渲染。围绕这一场景，本章首先给出需求分析，明确端到端框架在线实时处理与多模态输入方面需要具备的能力，并据此推导训练数据应满足的基本信息条件；随后将进入数据集结构、表示与预处理，以及端到端架构设计的具体分析。

2.1 需求分析

本文拟设计的端到端框架需要在线环境中持续接收用户侧的多模态信号流，并在低延迟约束下输出可驱动虚拟人的手势运动。本文将输入侧约束为通用单设备条件，即系统仅依赖麦克风与前向摄像头即可完成所需信号的采集与解析。在该设定下，系统能够稳定获得以下输入信息：

(1) 语音信号 (Audio): 通过设备麦克风实时采集，为后续的语音特征提取提供原始波形输入。

(2) 面部参数 (Expression Parameters) 与头部姿态 (Head Pose): 通过设备前向摄像头获取面部视频序列，并借助面部捕捉工具对视频进行解析，输出结构化的面部表情参数以及头部三维姿态信息。

基于上述信号获取方式，本文端到端框架需要具备的核心能力可概括为：(1) 在线实时处理流：系统应面向连续输入流工作，支持随时间推进的增量式推理与输出，并满足交互场景所需的低延迟要求。

(2) 多模态条件生成：模型需要同时接受语音、面部参数与头部姿态三种输入模态，并学习它们与上半身手势运动之间的映射关系，从而生成与语音表达一致且具有非语言协同信息的动作序列。

(3) 实时虚拟人渲染：根据用户的面部参数与模型生成的实时动作，驱动虚拟人的姿态。

为训练满足上述能力的端到端模型，训练数据应在时间轴上提供与推理阶段一致的监督信号与条件信息。具体而言，数据集至少需要包含：

(1) 语音信号：用于提取语音内容与韵律特征，作为动作生成的主要时间驱动与语义线索。

(2) 面部参数：与推理阶段的面捕输出形式一致，用于向模型提供表情与语气相关的条件信息。

(3) 人体动捕：作为手势生成的监督目标，提供骨架运动。

此外，头部姿态不需要由数据集显式给出：在许多动捕数据中，头部相关骨骼或关节的旋转信息已经包含在人体运动序列内，因此头部姿态可以从动捕数据中计算得到，并与语音与面部参数对齐后作为模型输入条件。

综上，本节以单设备可获取信号为约束，明确了端到端实时框架的输入形式与能力要求，并给出了训练数据所需包含的基本信息条件。基于这些条件，后续小节将进一步分析训练数据的结构与表示方式。

2.2 训练数据来源

为在训练阶段引入面部模态监督，本文采用 BEAT 数据集^[19]。BEAT 是面向虚拟角色驱动的多模态动作数据集，核心优势在于同时提供语音、上半身动捕以及面部表情捕捉等信息，并包含丰富的演讲人数据。

通常，模型的骨架拓扑与面部参数化方式需要与训练数据保持一致。因此，本节将说明 BEAT 中使用的骨架拓扑结构与面部表情表示方式。

2.2.1 骨架拓扑

虚拟人身体姿态通常采用层次化骨骼结构表示，其中每个关节节点通过父子关系构成一棵运动学树，并通过关节的局部旋转描述其在三维空间中的姿态。骨架拓扑决定了系统中可驱动的关节集合与其运动学约束方式。

具体而言，设骨架包含 J 个关节节点，记关节集合为 $\mathcal{J} = \{1, 2, \dots, J\}$ ，每个关节 $j \in \mathcal{J}$ 具有唯一的父节点 $p(j)$ ，从而定义出一棵有根树结构。系统在每一帧 t 输出该骨架上所有关节的局部旋转 $\mathbf{r}_t^{(j)}$ ，从而构成虚拟人身体姿态的完整描述，形式化定义为：

$$\mathbf{v}_t^B = \left[\mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)}, \dots, \mathbf{r}_t^{(J)} \right] \in \mathbb{R}^{R \times J}. \quad (2.1)$$

其中， R 表示旋转表示的维度。

BEAT 数据集的标准骨架结构如图 2.1 所示，共包含人体中的 47 个关节节点，包

括上肢及躯干的三个主要控制点（蓝色区域所示），下肢关节则保持静态。



图 2.1 BEAT 数据集的骨架拓扑结构与驱动范围

Figure 2.1 Skeleton Topology and Actuated Joint Range in the BEAT Dataset

2.2.2 头部姿态参数

本文仅使用头部旋转信息作为头部姿态特征，不引入头部位置信息。

这是因为，头部位置更容易受到身体姿态变化的影响。以 BEAT^[19] 为例，该演讲数据集中的演讲者多为站姿录制，单次演讲时长较长（约 1 分钟），过程中可能出现轻微的重心移动等站姿调整；而在目标应用场景中，用户交互姿态可能包含坐姿，其身体活动方式与站姿存在差异，例如不易产生长时间站立带来的重心偏移。

因此，直接建模头部位置或位移可能引入额外的场景依赖性，从而削弱跨场景的泛化能力。基于上述考虑，本文仅采用头部旋转作为头部姿态输入特征。

BEAT 数据集没有单独的头部姿态参数。我们可以在预处理中利用骨架层级关系，沿骨架链路从根节点到头部关节的相对旋转进行旋转姿态的叠加，从而得到头部在全局坐标系下的绝对旋转表示。

设头部关节索引为 h ，根关节为 r 。记局部旋转 $\mathbf{r}_t^{(j)}$ 对应的旋转矩阵为 $\mathbf{R}_t^{(j)} = \Phi(\mathbf{r}_t^{(j)}) \in SO(3)$ ，其中 $\Phi(\cdot)$ 表示从所选旋转表示到旋转矩阵的映射。则关节 j 的全局旋转 $\mathbf{G}_t^{(j)} \in SO(3)$ 可递推定义为：

$$\mathbf{G}_t^{(r)} = \mathbf{R}_t^{(r)}, \quad \mathbf{G}_t^{(j)} = \mathbf{G}_t^{(p(j))} \mathbf{R}_t^{(j)}. \quad (2.2)$$

于是头部的绝对旋转为：

$$\mathbf{G}_t^{(h)} = \prod_{k \in \mathcal{P}(r \rightarrow h)} \mathbf{R}_t^{(k)}, \quad (2.3)$$

其中 $\mathcal{P}(r \rightarrow h)$ 表示从根关节到头部关节的有序关节序列（包含 r 与 h ）。

最后，将头部全局旋转再映射回与模型一致的旋转表示，得到头部姿态特征：

$$\mathbf{v}_t^H = \Psi(\mathbf{G}_t^{(h)}) \in \mathbb{R}^R, \quad (2.4)$$

其中 $\Psi(\cdot)$ 为从旋转矩阵到所选旋转表示的映射， R 为该表示的维度。

2.2.3 面部参数

为了在手势生成系统中实现对表情状态的采集、训练与渲染驱动，通常需要将面部形变定义为一组可控且可实时更新的参数向量，并使其能够稳定地映射到虚拟人网格模型的顶点形变上。

在 BEAT 数据集中，面部参数由 Apple ARKit^[20] 定义的多个形变基组合表示，在计算机图形与动画领域中，这个形变基系统被称为 BlendShape。

其思想来源于 Facial Action Coding System (FACS)^[21]，将复杂表情分解为若干可组合的基本动作单元 (Action Units, AU)，从而为将面部表情描述为一组可控参数。BlendShape 中，每个基形由一个标量权重控制局部表情变化。通过调节多个基形的权重并进行组合，即可生成丰富的复杂表情。BlendShape 没有固定的规范，动捕软件、三维人物模型可自定义自己的形变基组合，但通常只有基于相同规范的系统间才可以复用面部参数。

以闭眼动作为例，FACS 将闭上双眼设为一个动作单元，如图 2.2 所示。

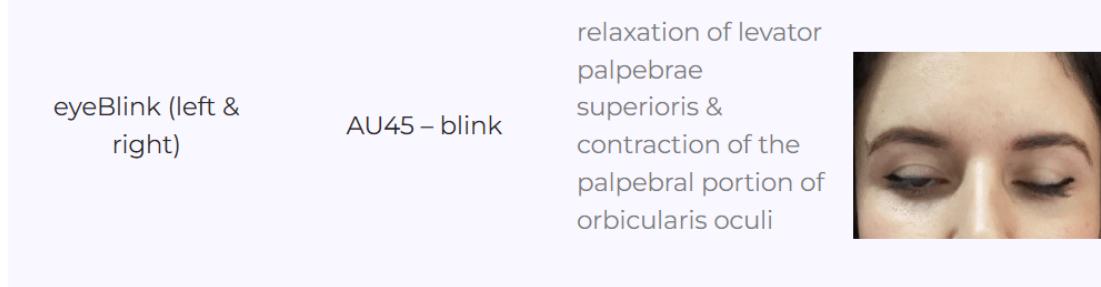


图 2.2 FACS 中闭眼动作单元的定义^[22]
Figure 2.2 Definition of Eye Closure Action Units in FACS^[22]

在 BEAT 数据集使用的 Apple ARkit 定义的 BlendShape 标准中，上述动作对应两种基形：闭左眼与闭右眼。图 2.3 展示了两种的 ARKit BlendShape 基形在权重 $w \in [0, 1]$ 下的线性插值效果（从张眼到闭眼），反映了 BlendShape 通过权重控制局部形变的基本机制。

设面部参数包含 K 个形变基，记形变基集合为 $\mathcal{K} = \{1, 2, \dots, K\}$ ，系统在每一帧



图 2.3 BlendShape 在闭眼形变上的线性插值效果

Figure 2.3 Linear Interpolation of BlendShape Deformation for Eye Closure

t 输出该所有的形变基权重 $\mathbf{w}_t^{(k)} \in [0, 1]$, 从而构成虚拟人面部的完整描述, 形式化定义为:

$$\mathbf{v}_t^F = \left[\mathbf{w}_t^{(1)}, \mathbf{w}_t^{(2)}, \dots, \mathbf{w}_t^{(K)} \right] \in \mathbb{R}^K. \quad (2.5)$$

在 Apple ARKit 中, 存在 52 个形变基, 即 $K = 52$ 。

2.3 旋转参数的选取

身体姿态的表示在训练时需要满足可学习性, 具体为 (1) 没有数值跳动 (2) 不需要额外规范化, 约束简单

因此, 我们需要将数据集的原始旋转参数格式 (欧拉角) 在预处理时进行转换。

常见的关节旋转表示包括: 欧拉角 (Euler Angles)、轴角表示 (Axis-Angle)、四元数 (Quaternion)、旋转矩阵 (Rotation Matrix), 以及近年来在神经网络回归任务中广泛使用的连续旋转表示, 如 6 维旋转表示 (Rot6D)。本文主要比较以下几种形式在工程系统中的适用性:

- 欧拉角: 参数维度低且直观, 没有冗余信息, 适合存储。但存在万向节锁 (gimbal lock) 问题, 且角度空间不连续, 在训练使用容易导致预测输出在临界角附近出现跳变, 从而破坏实时动画的平滑性。
- 轴角表示: 可用三维向量描述旋转, 但在接近零旋转时存在不稳定性, 且角度

周期性带来的不连续同样会对网络回归造成影响。

- 四元数：能避免万向节锁并具有较好的数值性质，但其单位范数约束使得网络直接回归时需要额外规范化步骤；同时四元数具有符号二义性（ \mathbf{q} 与 $-\mathbf{q}$ 表示同一旋转），可能导致学习目标不一致。
- 旋转矩阵：表达完整且无二义性，但需要满足正交约束与行列式为 1 的条件（ $SO(3)$ 群约束），直接回归时需额外投影或正交化处理，否则难以保证输出有效性。

综合考虑网络学习的稳定性、输出连续性与工程部署便利性，本文采用 Rot6D 作为关节局部旋转的统一表示形式。Rot6D 通过取旋转矩阵的前两列向量（记为 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ ）组成一个 6 维向量，并通过 Gram-Schmidt 正交化恢复完整的旋转矩阵，从而避免了显式的正交约束回归问题。具体而言，设模型输出的 Rot6D 为 $\mathbf{r} = [\mathbf{a}; \mathbf{b}] \in \mathbb{R}^6$ ，则可以按如下方式恢复旋转矩阵 $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ ：

$$\mathbf{u}_1 = \frac{\mathbf{a}}{\|\mathbf{a}\|}, \quad (2.6)$$

$$\mathbf{u}_2 = \frac{\mathbf{b} - (\mathbf{u}_1^\top \mathbf{b})\mathbf{u}_1}{\|\mathbf{b} - (\mathbf{u}_1^\top \mathbf{b})\mathbf{u}_1\|}, \quad (2.7)$$

$$\mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2, \quad (2.8)$$

$$\mathbf{R} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]. \quad (2.9)$$

该表示的主要优势体现在：

- (1) 连续性强，避免角度周期性造成的跳变；
- (2) 无需显式单位范数或正交约束回归，优化过程更稳定；
- (3) 可在推理后确定性地恢复旋转矩阵或转换为四元数，便于下游渲染引擎消费。

因此，本文采用 Rot6D 作为动作生成模型的训练目标。

根据式(2.1)，训练集中的身体姿态表示为： $\mathbf{v}_t^B \in \mathbb{R}^{6 \times 47}$ 。

又根据式(2.4)，训练集中的头部姿态表示为： $\mathbf{v}_t^H \in \mathbb{R}^6$ 。

2.4 在线实时手势生成系统整体框架

本节介绍整个系统的端到端驱动流程及模块职责划分。如图 2.4 所示，系统整体架构由五个层级组成：用户配置层、设备层、中间件层、手势生成模型层以及渲染与驱动层。各层之间通过多模态信号接口进行连接，实现从信号采集到虚拟人动作生成的端到端实时处理。

FaceCapGes 模型位于中间层，承担多模态输入到上半身姿态输出的核心推理任务，而输入采集与渲染模块分别负责信号获取与结果展示。

为实现基于语音、面部捕捉与头部姿态的实时数字人驱动系统，本文构建了完整的信号采集、动作生成与渲染展示的处理管线。FaceCapGes 模型作为该系统的核心计算模块，负责在实时约束下从多模态输入推理出当前帧的上半身骨骼姿态。

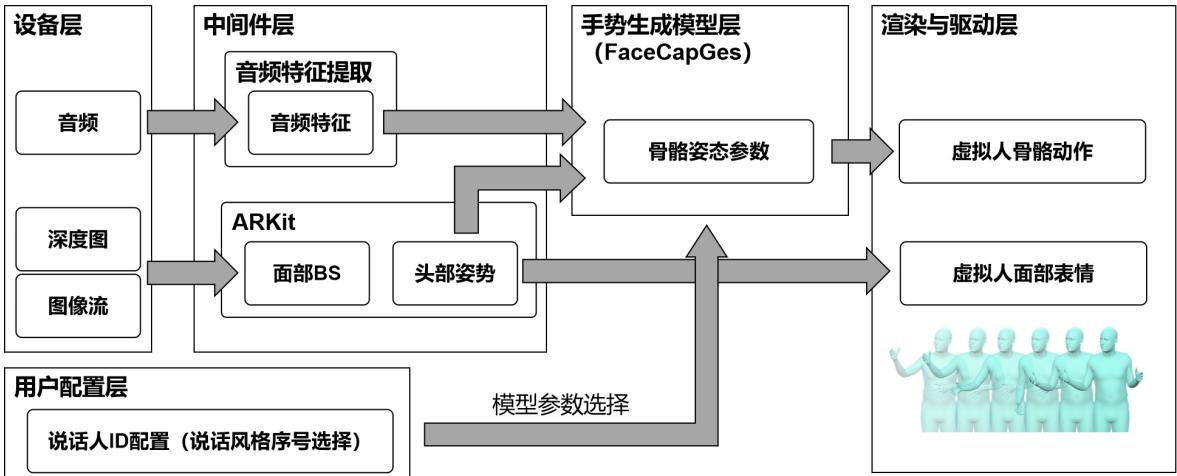


图 2.4 系统整体架构与数据流示意图

Figure 2.4 System Overview and Data Flow Diagram

2.4.1 信号采集与系统配置层

该层位于系统整体架构的输入端，用于从用户端设备实时获取多模态信号，并在系统初始化阶段完成运行参数的配置。整体结构可划分为设备层、中间件层与用户配置层三个部分，如图 2.4 所示。

设备层 设备层负责采集语音与视觉模态信号。语音信号由麦克风实时录制，采样率与帧移可根据运行设备性能调整；视觉信号由前置深度相机摄像头获取面部深度图与视频流，并作为 ARKit 面部追踪模块的输入。

中间件层 中间件层通过 Apple 提供的 ARKit 框架^[20]，将设备层的原始图像流与深度图转化为结构化特征。ARKit 输出两类主要数据：(1) 面部表情特征 ARKit 提供 52 维 BlendShape 系数向量，用于描述关键肌肉群的局部形变状态。该特征能够反映用户的表情、口型与情感变化，并以帧级形式同步输出。(2) 头部姿态特征 ARKit 在 ARFaceAnchor 中提供一个齐次变换矩阵 $\mathbf{T} \in \mathbb{R}^{4 \times 4}$ ，用于描述人脸锚点相对会话世界

坐标系的位姿：

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \mathbf{R} \in \mathbb{R}^{3 \times 3}, \mathbf{t} \in \mathbb{R}^{3 \times 1}. \quad (2.10)$$

其中左上角的 \mathbf{R} 为旋转矩阵，右上角的 \mathbf{t} 为平移向量。本研究从矩阵中提取旋转部分 \mathbf{R} ，并将其转换为 Rot6D^[23]表示形式，以提升旋转空间的连续性与模型训练的稳定性。

同时，音频流在中间件层中被传入特征提取模块以生成时间序列特征。模型训练阶段使用 Librosa 库离线提取 Mel 频谱、短时能量与基频 F_0 等声学特征，以保证特征精度与一致性。系统运行阶段可由等价的实时特征提取模块（如 torchaudio 或 TensorFlow Audio）逐帧生成对应特征，以实现端到端的低延迟运行。

用户配置层 用户配置层负责系统初始化阶段的模型与参数设定。用户可在应用中选择说话风格，对应加载不同说话人 ID 配置下的模型权重。该配置仅在系统启动时生效，不参与实时推理过程。

本层提供的多模态信号经中间件处理后，以统一的数据接口传递至手势生成模型，实现语音、表情与头部姿态的实时融合输入。

2.4.2 手势生成模型层

FaceCapGes 模块作为系统的核心推理单元，接收来自信号采集与系统配置层的三类输入特征：语音特征、面部 BlendShape 系数以及头部姿态参数，并在不依赖未来帧的条件下，逐帧预测用户当前时刻的上半身骨骼姿态。

生成的骨骼姿态采用 Rot6D^[23]连续旋转表示形式，覆盖上半身 47 个关节的旋转参数。模型内部通过级联多模态编码结构提取时序相关特征，并利用单向 LSTM 解码器完成时间依赖建模，从而在保持实时性的同时，生成与语音节奏、表情变化及头部朝向高度一致的自然手势。

FaceCapGes 输出的姿态数据通过统一接口传递至渲染与驱动模块，与实时面部捕捉信号共同驱动虚拟角色的整体动作。由于模型仅依赖当前与历史帧输入，可与输入层以固定帧率并行运行，实现端到端的低延迟推理。

2.4.3 渲染驱动层

该模块位于系统输出端，负责将手势生成模型与面部捕捉结果共同转化为虚拟人的实时动作表现。首先，场景内有一个虚拟人 3D 模型，其顶点网络数据具有骨骼绑定与面部 BlendShape 绑定。系统将 FaceCapGes 模型输出的上半身骨骼姿态与 ARKit

实时检测的 52 维面部 BlendShape 系数传递至渲染引擎，由引擎内的模块解析并映射至目标虚拟人的骨骼与表情控制接口，从而实现多模态动作驱动。

2.4.3.1 虚拟人骨骼蒙皮渲染

虚拟人角色一般由可渲染的三维网格 \mathcal{M} 与其绑定的骨骼结构 \mathcal{S} 组成。骨骼结构定义了关节的层级与运动学关系，而网格则提供可见的表面几何形态。在骨骼蒙皮渲染中，每个网格顶点 \mathbf{x} 会被赋予若干骨骼关节的影响权重，从而使得关节运动能够以连续方式驱动角色表面产生形变。

需要指出的是，由于不同虚拟人模型在骨骼命名与比例结构上可能存在差异，本文在 Unity^[24] 渲染阶段使用 Mecanim 提供的人形骨架映射机制^[25]，将本文输出骨架与目标角色骨架进行自动匹配，从而实现动作的重定向与稳定播放。该过程确保了本文定义的统一骨架拓扑能够在不同角色模型间保持一致的驱动效果，并将动作生成模块与渲染资产解耦，提高了系统的可扩展性。

2.4.3.2 虚拟人面部表情驱动

与骨骼蒙皮渲染类似，虚拟人角色有与三维网格绑定的各形变基的顶点位移。设虚拟人面部基础网格的顶点集合为 $\mathbf{v}_0 \in \mathbb{R}^{P \times 3}$ ，其中 P 为顶点数；给定 K 个 BlendShape 基形，其对应的顶点偏移为 $\Delta\mathbf{v}_i \in \mathbb{R}^{P \times 3}$ ($i = 1, \dots, K$)。则时间帧 t 下的面部表情网格可表示为：

$$\mathbf{v}(t) = \mathbf{v}_0 + \sum_{i=1}^K w_t^{(i)} \Delta\mathbf{v}_i, \quad (2.11)$$

其中 $w_t^{(i)} \in [0, 1]$ 为第 i 个基形在帧 t 的权重系数。式(2.11)表明，BlendShape 可以视为对基础网格的一组线性形变叠加，因此其具有良好的实时可驱动性与渲染兼容性，非常适合用于表情控制。

最终，系统能够在实时流式输入条件下稳定运行，同步呈现语音、表情与身体动作，以自然流畅的数字人形象实现从多模态信号输入到可视化输出的完整驱动流程。

2.5 本章小结

本章围绕在线实时数字人驱动场景的多模态手势生成任务，从系统约束出发给出了输入信号形式、数据来源与参数表示方式，并提出端到端实时系统的总体框架。首先，本章在单设备条件下明确系统需要持续接收语音信号与视觉解析得到的面部参数、头部姿态，并在低延迟约束下输出可驱动虚拟人的上半身骨骼动作。随后，本

章基于 BEAT 数据集说明训练数据的模态构成与同步关系，并分别给出骨架拓扑与面部 BlendShape 参数化表示；同时补充头部姿态数据的获取方式。在动作表示方面，本章选择 Rot6D 作为身体关节旋转的统一表示，以提升训练与推理的数值稳定性并便于下游渲染。最后，本章给出了端到端在线实时系统的分层架构与数据流组织方式，说明从多模态采集、特征转换到 FaceCapGes 推理与渲染驱动的完整链路。

下一章将进一步详细介绍 FaceCapGes 的多模态级联模型架构，包括语音、面部与头部姿态特征的编码方式、融合策略以及面向上半身姿态的层次化解码过程。

第3章 融合头部姿态的多模态级联手势生成模型架构

本章将介绍多模态编码、特征融合以及身体姿态解码的整体结构。本文参考基线模型 CaMN^[19] 的级联解码设计，在保留其核心思想的同时对离线时序建模进行在线实时适配，使模型能够在严格因果约束下仅依赖当前与历史信息稳定生成。与此同时，本文额外引入头部姿态模态以补充空间方向线索与节奏一致性，并构建对应编码器以学习补充表征。

3.1 问题定义

在整体系统中，FaceCapGes 模块承担着从多模态输入信号到上半身骨骼姿态预测的核心任务。为了明确模型的输入输出结构与学习目标，本节对该问题进行形式化定义。

3.1.1 任务描述

目标是在实时条件下，根据用户当前时刻的语音、面部表情与头部姿态信息，预测其对应的上半身骨骼姿态。模型需能够逐帧生成与语音节奏、面部动态和头部转动方向协调的自然手势动作，而不依赖未来的输入帧或整句语音信息。

形式上，可以将该任务定义为一个多模态时序映射函数：

$$\hat{\mathbf{v}}_t^B = f_{\theta}(\mathbf{v}_{t-N:t}^A, \mathbf{v}_{t-N:t}^F, \mathbf{v}_{t-N:t}^H), \quad (3.1)$$

其中 f_{θ} 表示由参数 θ 控制的生成模型， N 为历史窗口长度。各模态输入定义如下： \mathbf{v}_t^A 表示语音模态在时刻 t 的特征向量，由麦克风信号经音频特征提取得到，采样率为 16KHZ，与 CaMN^[19] 的提取方式一致； \mathbf{v}_t^F 表示面部模态的输入，为 ARKit 输出的 52 维标准化 BlendShape 系数； \mathbf{v}_t^H 表示头部模态的输入，为 ARKit 得出的头部旋转；而 $\hat{\mathbf{v}}_t^B$ 为生成模型在当前时刻预测的上半身骨骼姿态向量。模型仅利用当前及过去 N 帧的输入信息估计 $\hat{\mathbf{v}}_t^B$ ，从而满足严格的实时推理约束。

3.1.2 输入与输出模态

FaceCapGes 模型的输入由三种可同时实时获取的模态组成：语音特征、面部 BlendShape 权重及头部姿态参数；输出为当前帧的上半身骨骼旋转状态。各模态的符号与维度如表 3.1 所示。

表 3.1 输入输出模态符号与维度
Table 3.1 Notations and Dimensions of Input/Output Modalities

模态	符号	维度	描述
语音特征	v_t^A	\mathbb{R}^{1067}	由原始音频信号提取的时序特征
面部 BlendShape	v_t^F	\mathbb{R}^{52}	ARKit 输出的标准化表情权重向量
头部姿态	v_t^H	\mathbb{R}^6	头部关节旋转状态
骨骼姿态 (输出)	\hat{v}_t^B	$\mathbb{R}^{6 \times 47}$	上半身 47 个关节的旋转状态

输入序列 $(v_{t-N:t}^A, v_{t-N:t}^F, v_{t-N:t}^H)$ 描述了用户在当前即过去 N 帧内的语音与表情动态信息。模型通过学习其时序变化规律，逐帧生成对应的骨骼姿态输出 \hat{v}_t^B 。在推理阶段，模型仅访问至时刻 t 的输入序列，无法访问任何未来帧信息，保证了生成过程的因果性与实时性。

3.2 级联架构设计的继承

3.2.1 级联架构的原理与理论背景

现有语音驱动手势生成模型多采用多模态融合结构，其中以 CaMN^[19]为代表的级联架构在设计理念上具有代表性。其核心思想是将语音、面部表情与身体动作视为语义表达的不同层级：语音模态承担语义与节奏驱动作用，面部模态反映情感与意图，身体动作则是语言与情绪的外化呈现。CaMN 采用自上而下的处理顺序，即依次对语音、面部和动作模态进行建模，从而以层次化结构保持模态间的语义依存关系。

这种设计符合人类交流中“语言、表情、动作”一体化的认知规律^[3-4]。语音先规划语义与节奏，面部表情作为情绪强化信号随后产生，最终通过身体动作完成完整的非语言表达。模型中，语音编码器输出的时间嵌入被输入至面部编码器，再与面部特征融合后驱动动作解码器，从而保持语义一致性并增强表现力。

然而，CaMN 的原始设计面向离线整句生成任务，需要访问未来上下文以维持全局连贯性。在实时场景下，这种依赖将引入显著延迟并破坏因果性。FaceCapGes 在继承其层次思想的同时，对输入模式、训练方式与模态选择进行了系统性重构，以满足帧级实时约束。

3.2.2 说话人条件输入的简化

第2.4节提到，本文设计的框架中，系统配置层支持按说话人选择模型参数，但在本文任务设定中，说话人 ID 不参与网络输入。这是因为，在实时交互的单次运行

过程中，用户身份通常保持不变，系统无需在帧级推理过程中频繁切换说话风格。因此，模型无需显式感知并分离属于说话人风格的成分；相反，将风格选择上移至系统层即可满足使用需求。基于上述考虑，FaceCapGes 不引入说话人 ID 输入模态，采用针对每个说话人独立训练并部署模型参数的方式，从而减少网络输入与分支结构带来的建模负担与耦合复杂度，同时保持实时推理流程的简洁性。

3.2.3 输入模态的编码

语音特征通过时间卷积网络（Temporal Convolutional Network, TCN）和多层感知机（Multilayer Perceptron, MLP）编码，以捕捉短时节奏模式；面部模态采用相似结构，并在中间层融合语音嵌入，从而增强语音与表情之间的语义关联。两者都延续 CaMN^[19]的结构设计。

具体而言，语音编码器 E_A ，采用 12 层 TCN 建模局部时间依赖，并通过跳跃连接（skip connection）增强深层特征的传递稳定性；在第 12 层提取的语音时序特征后接入两层 MLP，用于进一步特征精炼与维度压缩，最终输出在时刻 t 下的语音潜在表示 $\mathbf{z}_t^A \in \mathbb{R}^{128}$ 。面部编码器 E_F 采用较浅的 8 层 TCN 结构以捕捉面部表情的短时动态变化，并在第 8 层将语音嵌入与面部特征进行通道维拼接融合，以增强面部表情变化与语音节奏之间的对应关系；编码器末端同样使用两层 MLP 进行特征映射与输出压缩，最终得到面部潜在表示 $\mathbf{z}_t^F \in \mathbb{R}^{32}$ 。

语音编码器 E_A 与面部编码器 E_F 的输出定义为：

$$\mathbf{z}_t^A = E_A(\mathbf{v}_{t-N:t}^A), \quad \mathbf{z}_t^F = E_F(\mathbf{v}_{t-N:t}^F; \mathbf{z}_t^A). \quad (3.2)$$

以上两个编码器负责提取低层次语音节奏与表情动态信息，为后续模态融合提供稳定上下文表征。

此外，系统在此基础上引入头部姿态模态 \mathbf{v}_t^H ，用于补充空间方向与节奏信号。由于本文实现的头部姿态编码器 E_H 采用逐帧前馈映射，其在因果窗口 $t-N:t$ 内对每一帧独立提取潜在表征：

$$\mathbf{z}_t^H = E_H(\mathbf{v}_t^H). \quad (3.3)$$

其中， $\mathbf{z}_t^H \in \mathbb{R}^{12}$ 。头部姿态编码器 E_H 的结构将在第 3.3 节详细说明。

3.2.4 身体姿态的解码

在输入模态经过编码与融合后，模型需将多模态特征映射至对应的身体姿态空间。为实现层次化的动作生成与结构协调，本文将上半身的输出区域划分为两个互补

分支：躯干（Torso, T）与上肢（Upper limbs, U）。躯干部分包含脊椎的三个主要控制关节，用于确定身体的姿态基准与运动节奏；上肢部分包含双臂及手部关节，负责生成与语音节奏及情绪表达相呼应的细节动作。最终上半身姿态表示为两者的组合：

$$\mathbf{v}_t^B = [\mathbf{v}_t^T; \mathbf{v}_t^U], \quad (3.4)$$

其中 $[::]$ 表示在特征维度上的拼接。

该分层设计继承了 CaMN 的层次预测思路：模型首先生成相对稳定的躯干姿态以确定整体方向，再以此为条件预测上肢动作，从而在实时生成中保持整体协调性与自然度。

融合输入的序列化表示 在时刻 t ，来自语音、面部与头部编码器的特征 \mathbf{z}_t^A 、 \mathbf{z}_t^F 、 \mathbf{z}_t^H 与历史动作上下文共同构成 LSTM 解码器的输入。为保持严格因果性，我们显式提供最近 N 帧的历史上半身姿态 $\mathbf{v}_{t-N:t-1}^B$ 作为条件信息，并对当前时刻的动作输入使用占位符进行对齐。具体而言，定义历史动作对齐序列 \mathbf{s}_τ 为：

$$\mathbf{s}_\tau = \begin{cases} \mathbf{v}_\tau^B, & \tau \leq t-1, \\ \mathbf{0}, & \tau = t, \end{cases} \quad \tau \in \{t-N, \dots, t\}, \quad (3.5)$$

则窗口内每一帧的融合输入向量可写为：

$$\mathbf{z}_\tau^{fuse} = [\mathbf{z}_\tau^A; \mathbf{z}_\tau^F; \mathbf{z}_\tau^H; \mathbf{s}_\tau], \quad \tau \in \{t-N, \dots, t\}, \quad (3.6)$$

其中 $[::]$ 表示在特征维度上的拼接。由此可得到长度为 $N+1$ 的因果上下文输入序列：

$$\mathbf{Z}_t^{fuse} = (\mathbf{z}_{t-N}^{fuse}, \dots, \mathbf{z}_t^{fuse}), \quad (3.7)$$

其中 (\cdot) 表示按时间顺序堆叠得到的序列。

躯干与上肢的级联解码 将输入序列 \mathbf{Z}_t^{fuse} 分别送入躯干与上肢两路单向 LSTM 解码器，得到窗口内的输出序列：

$$\mathbf{O}_t^T = \text{LSTM}_T(\mathbf{Z}_t^{fuse}), \quad \mathbf{O}_t^U = \text{LSTM}_U(\mathbf{Z}_t^{fuse}; \mathbf{O}_t^T), \quad (3.8)$$

其中 $\mathbf{O}_t^T = (\mathbf{o}_{t-N}^T, \dots, \mathbf{o}_t^T)$ ， $\mathbf{O}_t^U = (\mathbf{o}_{t-N}^U, \dots, \mathbf{o}_t^U)$ 。由于本文在时刻 t 的目标是预测当前帧动作，我们仅取序列末端输出作为当前帧的潜在表征：

$$\mathbf{z}_t^T = \mathbf{o}_t^T, \quad \mathbf{z}_t^U = \mathbf{o}_t^U. \quad (3.9)$$

最后，通过两路独立的两层 MLP 模块将潜在表征还原为旋转参数：

$$\hat{v}_t^T = \text{MLP}_T(z_t^T), \quad \hat{v}_t^U = \text{MLP}_U(z_t^U), \quad (3.10)$$

并根据式(3.4)通过特征维度上的拼接得到当前帧的完整上半身动作预测：

$$\hat{v}_t^B = [\hat{v}_t^T; \hat{v}_t^U]. \quad (3.11)$$

上述内容描述了模型在一个因果上下文输入序列内完成对当前帧动作的解码过程。

需要说明的是，LSTM 在实现中维护隐藏状态与记忆单元状态以编码时间依赖，其形式化表达见第4.1节。

3.3 头部姿态模态在级联架构中的引入

在级联架构设计中，我们考察了头部姿态特征与其他模态的多种组合方式。具体而言，分别尝试了：(1) 将头部姿态特征在编码阶段与语音或面部特征进行早期融合；(2) 在解码阶段以前两者的嵌入结果为条件，预测头部姿态特征作为辅助信号。实验结果显示，这两种交互方式均未带来显著性能提升，观察到了训练收敛速度的下降。

本文推测，头部动作虽然与语音韵律在时间上存在同步性，但头部动作与语音或表情之间的驱动关系不强，因为头部动作可以包含演讲人自然面向不同方向的听众等，与语音韵律或情感难以找到关联性的信息。

基于此观察，本文在最终架构中采用了弱耦合的设计：头部姿态特征在语音与面部特征编码完成后，以独立通道的形式拼接至多模态隐向量 z_t^{fuse} ，同时头部姿态特征不参与其它输入模态的编码过程。我们在此配置下得到了更快的训练收敛。

编码器结构 图 3.1所示为头部姿态编码器结构。该编码器由两层前馈网络组成，输入为 Rot6D^[23] 表示的 6 维向量。式(3.3)中的 E_H 的具体形式为：

$$\mathbf{h}_1 = \text{ReLU}(W_1 v_t^H + b_1), \quad (3.12)$$

$$z_t^H = W_2 \mathbf{h}_1 + b_2. \quad (3.13)$$

网络维度设置为：输入 6，中间层 36，输出 12。在特征层面，其输出将与语音、面部嵌入在通道维拼接后输入解码器，时间依赖由后续单向时序解码器在因果窗口内进行建模。



图 3.1 头部姿态编码器结构示意图

Figure 3.1 Architecture of the Head Pose Encoder

3.4 模型整体结构

图 3.2展示了 FaceCapGes 从音频、面部、头部编码器分别提取模态特征后拼接，输入至 LSTM 解码器生成躯干与手部姿态的过程。图中， \otimes 符号表示在特征维度上的拼接。

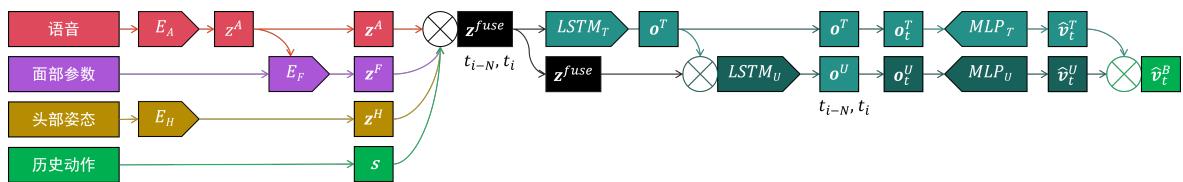


图 3.2 FaceCapGes 模型整体结构

Figure 3.2 Overall Architecture of FaceCapGes

3.5 本章小结

本章围绕 FaceCapGes 的模型结构设计，系统介绍了面向严格实时交互场景的多模态级联架构与关键输入输出模态设定。首先，本文在任务设定上明确了逐帧在线生成与严格因果约束：模型在任意时刻仅利用当前及过去帧的语音、面部与头部信息进行动作预测，从而区别于依赖未来上下文的一次性离线生成方法，并为后续结构设计提供了统一前提。

在模型结构方面，本文继承 CaMN^[19]的级联框架，将语音、面部表情与身体动作视为具有层次关联的表达信号，并进一步系统性地引入头部姿态作为独立输入模态，以补充空间方向与节奏信息。同时，本文仅使用头部旋转信息进行建模，从而提升跨场景泛化能力，并给出了分别适用于训练与推理阶段的特征获取方式。

在解码端，本章说明了面向上半身骨骼姿态的层次化输出设计：将动作划分为躯干与上肢两路分支，并通过级联解码保持整体协调性与自然度。上述模型结构为后续章节提出的滑动窗口自回归展开策略与片段级优化目标提供了清晰的结构基础。

下一章将进一步介绍模型在训练与推理阶段的统一展开过程，并证明该策略在严格因果约束下能够实现稳定的实时动作生成。

第4章 基于滑动窗口的实时手势生成自回归训练

本章讨论在严格因果约束下，如何将第3章定义的窗口内单步预测模型展开为逐帧自回归生成过程，并构建与之对应的训练目标。由于实时场景无法观测未来的多模态输入，模型必须在每个时间步仅依赖历史信息预测下一帧动作，从而保证在线推理的一致性与可部署性。为此，本文提出片段切割与滑动窗口展开的流程：通过前置动作帧缓冲历史上下文，并在生成阶段逐帧预测与写回动作缓存，形成连续的片段级输出序列。在此基础上，本章进一步给出片段级监督损失与对抗训练目标的定义。

4.1 单步因果 LSTM 预测器

为实现严格因果的实时手势生成，本章首先建立一个用于“下一帧动作预测”的基本单元：单步因果 LSTM 预测器。该预测器在每个时间步利用当前可用的多模态输入特征，并结合历史动作上下文与循环状态，对下一帧动作进行估计。在后续章节中，我们将该单步预测器以滑动窗口方式展开，从而形成对一个动作片段的自回归生成过程，并据此定义片段级别的监督损失与对抗训练目标。

4.1.1 LSTM 基本概念与状态传递机制

手势动作序列具有显著的时间依赖性：当前姿态不仅受当前输入模态（如语音、面部表情、头部姿态等）影响，也与过去的动作状态密切相关。循环神经网络（Recurrent Neural Network, RNN）通过引入随时间递推的隐状态来建模序列依赖，而 LSTM 进一步通过门控机制缓解长序列训练中的梯度消失问题，从而更适合用于动作序列建模。

在标准的 LSTM 结构中，网络在每个时间步接收当前输入特征 \mathbf{x}_t ，并维护两类递推状态：隐藏状态 \mathbf{h}_t 与记忆单元状态 \mathbf{c}_t 。其中， \mathbf{h}_t 可视为与当前输出相关的短期表示，而 \mathbf{c}_t 作为更稳定的记忆轨道，用于在更长时间尺度上保留信息。其递推过程可形式化表示为：

$$(\mathbf{h}_t, \mathbf{c}_t) = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}). \quad (4.1)$$

根据隐状态传播方向的不同，LSTM 主要分为单向 LSTM（Unidirectional LSTM）与双向 LSTM（Bidirectional LSTM）。

双向 LSTM 通过同时构建前向与反向递推路径，在输出时刻 t 的表示时会融合来自未来时间步的反向信息。该结构在离线分析或全序列可见的任务中能够更充分

利用上下文，从而提升预测性能；然而在实时生成任务中，未来输入在时刻 t 尚不可用，反向路径的依赖无法满足。

相比之下，单向 LSTM 沿时间正向递推，其状态更新仅依赖于过去与当前输入，从结构上满足因果约束。对于下一帧动作生成问题，UniLSTM 能够自然地被解释为一个逐步更新的预测器：在每个时间步基于当前可观测输入与历史状态输出下一帧动作估计，并将生成结果反馈至下一步，从而形成自回归（autoregressive）的实时生成流程。此外，单向递推的状态传递机制也使得模型能够在有限的输入上下文之外保留更长程的动态信息，为后续滑动窗口展开提供了必要的记忆能力。

基于以上原因，本文采用单向 LSTM 作为动作生成的时序建模骨干网络，以满足严格因果的实时生成需求。在后续的滑动窗口展开策略中（见第4.3节），该单步预测器将被逐帧迭代调用，从而在片段级别完成动作序列的自回归生成，并用于定义监督损失与对抗训练目标。

4.1.2 双时间尺度记忆结构

尽管单向 LSTM 通过递推状态 $(\mathbf{h}_t, \mathbf{c}_t)$ 在理论上具备建模长程依赖的能力，但在自回归动作生成任务中，若仅依赖循环状态作为唯一的历史信息通道，将迫使该状态同时承担短期细节与长期规律的表征。由于循环状态本质上是对历史信息的压缩表示，其容量有限且受 LSTM 的门控遗忘机制影响，短期运动细节与局部连续性约束可能难以稳定保留，从而使生成过程更易出现抖动、漂移或长期一致性下降等现象。因此，本文在跨步循环状态之外显式提供固定长度的历史动作上下文，作为短期约束信号，以在保持长期记忆能力的同时增强逐帧预测的稳定性与可控性。

为提升逐帧生成的稳定性，本文在循环状态之外显式引入固定长度的历史动作上下文作为短期条件信息。具体而言，设 N 为历史动作上下文长度，在预测时刻 t 的动作时，我们显式提供最近 N 帧的动作历史序列 $\mathbf{v}_{t-N:t-1}^B$ 作为可观测输入，使模型在每一步预测中均能获得短期运动细节与局部连续性约束。为保持因果性，该历史动作序列不包含当前待预测时刻 t 的真实动作，而采用占位符进行对齐（例如以零向量或掩码符号表示），从而形成长度为 $N+1$ 的因果上下文序列。

在该设计下，模型的历史信息将通过两条互补路径传递：一方面，循环状态 (\mathbf{h}, \mathbf{c}) 作为长期记忆通道，用于编码超过 N 帧范围的更长程动态趋势、说话风格与运动节奏；另一方面，显式历史动作上下文作为短期精确条件，在每一步预测中直接提供最近 N 帧的局部运动信息，从而缓解状态漂移带来的不确定性，并提升自回归生成的鲁棒性。本文采用“短期显式上下文 + 长期隐式状态”的双时间尺度记忆结构，以在

严格因果约束下平衡表达能力与生成稳定性。

4.1.3 窗口内预测过程的形式化表达

本节将第3章定义的窗口内解码结构形式化为可递推的单步因果预测器，以明确在线生成时循环状态的跨步传递方式。在时刻 t ，模型以长度为 $N+1$ 的因果上下文输入序列 \mathbf{Z}_t^{fuse} 作为输入（其定义见式(3.7)），并分别维护躯干与上肢的 LSTM 隐藏状态与记忆单元状态： $(\mathbf{h}_{t-1}^T, \mathbf{c}_{t-1}^T)$ 与 $(\mathbf{h}_{t-1}^U, \mathbf{c}_{t-1}^U)$ 。

在线运行时，我们将跨步传递的状态作为 LSTM 解码器的初始状态，从而得到窗口内输出序列并更新状态：

$$\mathbf{O}_t^T, (\mathbf{h}_t^T, \mathbf{c}_t^T) = \text{LSTM}_T(\mathbf{Z}_t^{fuse}, \mathbf{h}_{t-1}^T, \mathbf{c}_{t-1}^T), \quad (4.2)$$

$$\mathbf{O}_t^U, (\mathbf{h}_t^U, \mathbf{c}_t^U) = \text{LSTM}_U(\mathbf{Z}_t^{fuse}, \mathbf{h}_{t-1}^U, \mathbf{c}_{t-1}^U). \quad (4.3)$$

由于本文在时刻 t 的目标是预测当前帧动作，我们使用窗口末端输出作为当前帧潜在表征，并通过 MLP 解码得到 $\hat{\mathbf{v}}_t^T$ 与 $\hat{\mathbf{v}}_t^U$ ，最终拼接为 $\hat{\mathbf{v}}_t^B$ 。该过程已在第3.2.4节给出，此处不再赘述。

因此，可以将单步因果预测器抽象为如下递推形式：

$$\hat{\mathbf{v}}_t^B, (\mathbf{h}_t, \mathbf{c}_t) = f_\theta(\mathbf{Z}_t^{fuse}, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}), \quad (4.4)$$

其中 $(\mathbf{h}_t, \mathbf{c}_t)$ 表示躯干、上肢 LSTM 状态的集合。在第4.3节中，我们将进一步描述该单步预测器如何随时间滑动展开，从而在片段尺度上生成长度为 M 的动作序列，并用于定义监督损失与对抗训练目标。

4.2 训练片段切割

本工作沿用 CaMN 采用的训练样本构造方式^[19]，将长序列动作数据切割为固定长度的短片段作为训练样本。

4.2.1 固定长度片段定义

设原始动作序列为 $\{\mathbf{v}_t^B\}_{t=1}^T$ ，以及对应的多模态输入特征序列 $\{\mathbf{z}_t^A, \mathbf{z}_t^F, \mathbf{z}_t^H\}_{t=1}^T$ 。我们从长序列中截取长度为 L 的连续片段作为训练样本，其中片段长度由历史上下文长度 N 与片段内生成步数 M 共同决定：

$$L = N + M. \quad (4.5)$$

在本文设定中，沿用 CaMN 的片段长度配置取 $L = 34$ 帧，同时选取历史上下文长度 $N = 16$ 帧，因此对应的片段内自回归生成步数为 $M = 18$ 帧。

对于第 k 个训练片段，其时间范围为 $[s_k, s_k + L - 1]$ ，片段内的动作与多模态输入分别表示为：

$$\mathbf{V}_k^B = (\mathbf{v}_{s_k}^B, \mathbf{v}_{s_k+1}^B, \dots, \mathbf{v}_{s_k+L-1}^B), \quad (4.6)$$

$$\mathbf{Z}_k = \left(\{\mathbf{z}_{s_k:t}^A\}_{t=s_k}^{s_k+L-1}, \{\mathbf{z}_{s_k:t}^F\}_{t=s_k}^{s_k+L-1}, \{\mathbf{z}_{s_k:t}^H\}_{t=s_k}^{s_k+L-1} \right). \quad (4.7)$$

其中 \mathbf{V}_k^B 将进一步划分为两部分：前 N 帧作为片段的历史上下文（亦即前置动作帧，用于缓冲与提供因果条件），后 M 帧为片段内需要逐帧生成并参与训练目标计算的部分。该划分将于第4.3节中用于描述滑动窗口展开过程与片段级损失计算策略。

4.2.2 重叠切割与样本覆盖

为提高训练样本覆盖率并增强模型对不同对齐位置的鲁棒性，我们采用带重叠的滑动切割方式从长序列中提取片段。具体而言，片段起始位置 s_k 按固定步长 Δ 滑动：

$$s_k = 1 + (k - 1)\Delta, \quad (4.8)$$

从而得到一组相互重叠的训练片段 $\{\mathbf{V}_k^B, \mathbf{Z}_k\}$ 。在本文实现中，沿用 CaMN 的设置取 $\Delta = 10$ 帧，以在样本数量与数据冗余之间取得平衡。

4.3 片段内部的滑动窗口展开策略

上一节定义了训练样本以固定长度片段组织：每个片段长度为 $L = N + M$ ，其中前 N 帧为历史上下文，后 M 帧为需要逐帧生成的目标区间。本节进一步描述片段内部的滑动窗口展开策略：该策略在训练与推理阶段保持一致，以单步因果预测器（第4.1节）为基本计算单元，逐帧生成长度为 M 的动作序列并拼接得到片段级输出。在本文设定中， $L = 34$ ， $N = 16$ ，因此 $M = 18$ ，窗口长度为 $N+1 = 17$ 。

4.3.1 前置动作帧与预热阶段

在严格因果的实时生成中，模型在时刻 t 预测动作时不能访问未来输入，并且其输出需要在片段之间保持连续。为此，我们将每个片段的前 N 帧动作视为前置动作帧，其作用是为后续生成提供短期运动上下文，并避免片段边界处出现断裂。

在片段开始时刻 s_k ，我们首先执行预热阶段：将片段的前 N 帧真实动作 $\{\mathbf{v}_{s_k}, \dots, \mathbf{v}_{s_k+N-1}\}$ 写入历史动作缓存 \mathcal{H} ，并同步读取对应的多模态输入特征。本文

在预热阶段不执行任何前向计算，生成阶段的初始 LSTM 状态由预设初始化值给出。本文采用零初始化作为 $(\mathbf{h}_{t-1}, \mathbf{c}_{t-1})$ 的初值。

4.3.2 滑动窗口展开与逐帧自回归生成

在预热阶段结束后，我们进入生成阶段：模型在片段内部进行 M 步滑动窗口展开，每一步生成 1 帧动作，并将预测结果写回历史动作缓存以供下一步使用。设片段内生成阶段的第 m 步对应全局时间 $t = s_k + N - 1 + m$ （其中 $m = 1, 2, \dots, M$ ），历史动作缓存 \mathcal{H}_{t-1} 包含最近 N 帧可用动作序列：

$$\mathcal{H}_{t-1} = (\tilde{\mathbf{v}}_{t-N}^B, \dots, \tilde{\mathbf{v}}_{t-1}^B), \quad (4.9)$$

其中 $\tilde{\mathbf{v}}$ 表示当前可用的动作帧：在生成开始时它包含真实前置动作帧，在生成推进过程中则逐渐由模型预测结果覆盖。

在时间步 t ，我们构造长度为 $N+1$ 的因果上下文窗口，组合当前可观测的多模态输入特征与历史动作缓存，形成融合输入序列 \mathbf{Z}_t^{fuse} （其定义见式(3.7)）。其中历史动作序列部分采用 $(\tilde{\mathbf{v}}_{t-N}^B, \dots, \tilde{\mathbf{v}}_{t-1}^B, \mathbf{0})$ 进行对齐，末帧使用占位符以保证严格因果性。

随后，单步因果预测器以 \mathbf{Z}_t^{fuse} 与跨步 LSTM 状态作为输入，输出当前帧动作预测 $\hat{\mathbf{v}}_t^B$ 并更新状态（窗口内形式化表达见第4.1.3节）。生成结果 $\hat{\mathbf{v}}_t^B$ 会被写回历史动作缓存，从而更新 \mathcal{H}_t 并用于下一时间步预测：

$$\mathcal{H}_t = (\tilde{\mathbf{v}}_{t-N+1}^B, \dots, \tilde{\mathbf{v}}_{t-1}^B, \hat{\mathbf{v}}_t^B). \quad (4.10)$$

通过上述逐帧自回归展开，我们得到片段内生成区间的预测序列：

$$\hat{\mathbf{V}}_k^{gen} = (\hat{\mathbf{v}}_{s_k+N}^B, \hat{\mathbf{v}}_{s_k+N+1}^B, \dots, \hat{\mathbf{v}}_{s_k+L-1}^B), \quad (4.11)$$

其长度为 M 。

4.3.3 拼接生成序列与片段级输出

由于滑动窗口展开在每一步仅生成 1 帧动作，片段级输出通过对 M 步预测结果进行时间维拼接获得。片段级生成结果 $\hat{\mathbf{V}}_k^{gen}$ 与真实动作片段 \mathbf{V}_k^B 在时间上对齐，其中前 N 帧为可观测上下文，后 M 帧为模型生成输出。因此，后续训练目标的定义将以 $\hat{\mathbf{V}}_k^{gen}$ 为核心对象，并与真实片段中对应的生成区间进行比较。

4.4 监督损失

在训练阶段，给定来自多模态语音动作数据集的配对样本序列：

$$(z_t^A, z_t^F, z_t^H, v_t^B), \quad (4.12)$$

模型的学习目标是在严格因果约束下生成与输入相匹配的上半身动作序列。与传统离线序列预测不同，本文采用第4.3节所述的滑动窗口展开策略：模型在每个片段内部自回归地逐帧生成 M 帧动作，并将其拼接为片段级生成序列。

4.4.1 片段级生成序列与损失计算范围

对于第 k 个训练片段，其长度为 $L = N + M$ （第4.2节），其中前 N 帧为前置动作帧，后 M 帧为模型需要生成的目标区间。根据滑动窗口展开过程（式(4.11)），模型得到片段生成区间的预测序列：

$$\hat{\mathbf{g}}_k = (\hat{v}_{s_k+N}^B, \hat{v}_{s_k+N+1}^B, \dots, \hat{v}_{s_k+L-1}^B) \in \mathbb{R}^{M \times d}, \quad (4.13)$$

其中 d 表示动作表示的维度（本文中使用 Rot6D^[23]，即 $d = 6$ ）。对应的真实动作序列为：

$$\mathbf{g}_k = (v_{s_k+N}^B, v_{s_k+N+1}^B, \dots, v_{s_k+L-1}^B) \in \mathbb{R}^{M \times d}. \quad (4.14)$$

需要强调的是，片段前 N 帧前置动作仅用于提供因果历史上下文与片段平滑过渡，其本身并非生成目标。因此，与 CaMN^[19]不同的是，本文的监督损失仅在片段生成区间 $\{s_k + N, \dots, s_k + L - 1\}$ 上计算，不对前置动作帧计算任何损失项。这是因为前置动作帧作为已知条件用于初始化历史缓存，而模型输出仅对应后续的生成区间。

4.4.2 总体优化目标

综合考虑空间重构精度、时序平滑性以及动作分布一致性，本文的总体优化目标定义为：

$$\mathcal{L}_{total} = \lambda_r \mathcal{L}_{rec} + \lambda_v \mathcal{L}_{vel} + \lambda_a \mathcal{L}_{acc} + \lambda_{adv} \mathcal{L}_{adv}, \quad (4.15)$$

其中 \mathcal{L}_{rec} 衡量片段生成区间的姿态重构误差， \mathcal{L}_{vel} 与 \mathcal{L}_{acc} 分别约束速度与加速度的连续性， \mathcal{L}_{adv} 表示对抗训练损失，将在第4.5节中进一步介绍。

4.4.3 姿态重构与时序平滑损失

为同时保证空间重构精度与时间连续性，我们采用基于 Huber 误差的重构损失形式，并分别作用于姿态、速度与加速度信号。给定任意预测序列 $\hat{\mathbf{x}}$ 及其对应的真实

序列 \mathbf{x} , 基础误差项定义为:

$$\mathcal{L}_{Huber}(\mathbf{x}, \hat{\mathbf{x}}) = \beta \cdot \text{SmoothL1}\left(\frac{\mathbf{x}}{\beta}, \frac{\hat{\mathbf{x}}}{\beta}\right), \quad (4.16)$$

其中 SmoothL1(\cdot) 表示平滑 L1 误差, β 为平滑系数, 本文中设为 0.1。

在此基础上, 片段生成区间的姿态、速度与加速度损失分别定义为:

$$\mathcal{L}_{rec} = \mathcal{L}_{Huber}(\mathbf{g}_k, \hat{\mathbf{g}}_k), \quad (4.17)$$

$$\mathcal{L}_{vel} = \mathcal{L}_{Huber}(\mathbf{g}'_k, \hat{\mathbf{g}}'_k), \quad (4.18)$$

$$\mathcal{L}_{acc} = \mathcal{L}_{Huber}(\mathbf{g}''_k, \hat{\mathbf{g}}''_k), \quad (4.19)$$

其中一阶与二阶时间差分 \mathbf{g}'_k 、 \mathbf{g}''_k 定义为:

$$\mathbf{g}'_{k,t} = \mathbf{g}_{k,t} - \mathbf{g}_{k,t-1}, \quad \mathbf{g}''_{k,t} = \mathbf{g}'_{k,t} - \mathbf{g}'_{k,t-1}, \quad (4.20)$$

预测序列 $\hat{\mathbf{g}}'_k$ 、 $\hat{\mathbf{g}}''_k$ 同理定义。

上述多尺度重构约束在自回归预测过程中能够有效缓解高频抖动与速度漂移问题, 在保证运动学精度的同时提升生成序列的时间稳定性。

4.4.4 损失权重设置

各损失项的权重系数在实验中设定为 $\lambda_r = 5 \times 10^2$, $\lambda_v = 10^3$, $\lambda_a = 10^3$, $\lambda_{adv} = 10^{-1}$ 。

4.5 对抗训练

尽管第4.4节的监督损失能够约束生成序列在逐帧空间误差与局部平滑性上的一致性, 但仅依赖点对点重构目标往往难以完全刻画真实动作序列的整体动力学分布。为进一步提升生成动作的自然度与分布一致性, 本文引入片段级判别器, 在片段尺度上约束生成序列与真实序列的统计特性一致。

4.5.1 基于拼接片段的片段级判别

与窗口内部的中间输出不同, 本文的判别器直接以片段生成区间的拼接序列为输入。对于第 k 个片段, 生成器输出的预测序列 $\hat{\mathbf{g}}_k$ 及其对应的真实序列 \mathbf{g}_k 定义见式(4.13)与式(4.14), 二者均为长度 M 的序列。判别器 $Dis(\cdot)$ 接收一段动作序列, 并输出其来自真实数据分布的概率:

$$Dis(\mathbf{g}) \in (0, 1). \quad (4.21)$$

通过片段级输入, 判别器能够从整体动力学角度判断生成动作的真实感, 从而在节奏、能量变化与运动统计特性等层面提供补充监督信号。

前置动作帧的掩码策略 需要强调的是，与 CaMN^[19]不同，本文在对抗训练中同样不将片段前 N 帧前置动作输入判别器。前置动作帧属于可观测上下文条件，其内容在训练阶段为真实动作，在推理阶段为历史缓存或上一片段输出；它们并非模型需要生成的目标。因此，本文仅对生成区间 $\hat{\mathbf{g}}_k$ 与 \mathbf{g}_k 进行对抗判别，使对抗目标严格作用于模型实际生成的部分，并与第4.4节的监督损失范围保持一致。

4.5.2 对抗损失定义

判别器的训练目标是区分真实序列与生成序列，其损失定义为：

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{g}_k} [\log Dis(\mathbf{g}_k)] - \mathbb{E}_{\hat{\mathbf{g}}_k} [\log (1 - Dis(\hat{\mathbf{g}}_k))]. \quad (4.22)$$

生成器则希望其输出被判别器判定为真实，从而对应的对抗损失为：

$$\mathcal{L}_{adv} = -\mathbb{E}_{\hat{\mathbf{g}}_k} [\log Dis(\hat{\mathbf{g}}_k)]. \quad (4.23)$$

其中 $\mathbb{E}_{\mathbf{g}_k}$ 与 $\mathbb{E}_{\hat{\mathbf{g}}_k}$ 分别表示对真实序列与生成序列的采样期望。该对抗目标从分布层面鼓励生成序列在整体运动统计特性上接近真实数据，从而补充监督损失在局部误差上的约束。

4.5.3 交替优化策略

在训练过程中，本文采用交替优化方式更新生成器与判别器。具体而言，对于每个训练批次，我们首先固定生成器参数 θ ，最小化式(4.22)更新判别器参数；随后固定判别器参数，最小化总体损失 \mathcal{L}_{total} （见式(4.15)）更新生成器参数，其中对抗项 \mathcal{L}_{adv} 由式(4.23)给出。通过上述训练方式，判别器不断提升对真实与生成序列的区分能力，而生成器则在监督约束与分布约束的共同作用下，逐步生成更加自然且时间一致的动作序列。

4.6 本章小结

本章围绕严格因果的实时手势生成任务，系统阐述了本文的滑动窗口训练与推理策略，并给出了与之配套的片段级优化目标。首先，我们建立了单步因果预测器作为基本计算单元：采用单向 LSTM 以满足严格因果约束，并在每个预测步中结合跨步循环状态与显式历史动作上下文，采用“短期显式条件 + 长期隐式记忆”的双时间尺度建模，从而提升自回归生成过程的稳定性与长期规律记忆能力。随后，我们沿用 CaMN^[19]的固定长度片段切割方法将长序列组织为训练样本，并在片段内部执行

滑动窗口展开：通过预热阶段缓冲前置动作帧，再在生成阶段逐帧自回归预测并将输出写回历史缓存，最终拼接生成的所有动作帧得到生成序列。

在训练目标方面，本章将监督损失定义在片段生成区间的拼接输出上，采用姿态、速度与加速度的多尺度 Huber 约束以缓解抖动与速度漂移，并进一步引入片段级判别器以提供分布一致性的对抗监督。本文将前置动作帧视为纯条件上下文而非生成目标，在监督损失与对抗训练中均显式移除其影响，使训练目标严格作用于模型实际生成的区间，从而与实时推理阶段的因果生成流程保持一致。

第 5 章 实验结果与分析

为评估 FaceCapGes 的生成质量与实时性能，本文从用户评估、客观指标测量与推理效率三个方面展开实验，并在生成质量上与代表性方法进行对比分析。

5.1 训练配置

本文模型 FaceCapGes 基于 PyTorch 实现，所有实验在单张 NVIDIA RTX 4090 GPU 上进行。

本文基于 BEAT 数据集^[19] 进行训练与评估。该数据集包含多模态同步的语音、面部 blendshape 与全身动作信息，以 15 fps 记录多位专业表演者的演讲片段，覆盖多种语义与情绪场景。

本文选取表演者 ID 2、4、6、8 的数据进行训练与测试，其中 2、4 为男性，6、8 为女性，确保在性别与说话风格上的分布均衡。训练集与测试集均包含相同的表演者，但使用不同的演讲片段，在预处理阶段已进行严格划分以避免片段交叉。

训练配置 训练时输入窗口的前序帧数设为 $N = 16$ ，预测长度为 $M = 34$ ，训练片段的切割步长为 10 帧。相邻片段因此存在部分重叠，从而在保证充分上下文信息的同时提升数据覆盖率与时间连续性。批大小设为 256。

优化器采用随机梯度下降 (Stochastic Gradient Descent, SGD)^[26]。基础学习率设置为 $\text{lr}_{base} = 2.5 \times 10^{-4}$ ，并根据批大小按线性规则缩放为

$$\text{lr}_g = \text{lr}_{base} \cdot \frac{\text{batch_size}}{128}, \quad (5.1)$$

其中 lr_g 为生成器（手势生成网络）的实际学习率。

在对抗训练阶段，判别器使用相同类型的 SGD 优化器，其学习率按权重系数 w_d 缩放为

$$\text{lr}_d = w_d \cdot \text{lr}_g, \quad (5.2)$$

本文中设定 $w_d = 0.2$ 。

为防止早期训练阶段的不稳定，对抗项在第 10 个 epoch 后引入，即前 10 个 epoch 仅优化重构与时序平滑损失，从第 11 个 epoch 起加入判别器并交替优化生成器与判别器参数。整体训练共 374 个 epoch。

姿态表示 所有身体动作均转换为连续可微的 Rot6D 表示，使用 EMAGE^[11] 中的实现方法，以避免欧拉角奇异性与四元数的符号不确定性。

运行性能 在实时推理阶段，FaceCapGes 以 15 FPS 的速度驱动虚拟角色。

5.2 实验配置

5.2.1 实验对比模型

我们选取 FaceCapGes 的基线模型 CaMN^[19]，以及扩散模型方法中具有代表性的 DiffSHEG^[15] 作为对比模型。

表 5.1 总结了各模型的输入输出模态特征。其中，* 表示模型结构未显式输入说话人 ID，采用每位说话人独立训练的设置，说话人 ID 通过模型参数指定。

值得注意的是，FaceCapGes 是唯一满足严格因果约束的在线模型，因此在评估时采用逐帧推理并将输出拼接为完整序列，以模拟实时输入流。

表 5.1 对比模型的输入输出模态
Table 5.1 Comparison of Input/Output Modalities of Evaluated Methods

模型	输入模态					未来信息	输出
	音频	面部捕捉	头部姿态	说话人 ID	情绪		
CaMN	✓	✓	✗	✓	✓	✓	身体
DiffSHEG	✓	✗	✗	✓	✗	✓	身体 + 面部
本模型	✓	✓	✓	*	✗	✗	身体

5.2.2 跨模型评估设置

本章所有实验均基于 BEAT 数据集进行，各模型使用一致的骨架拓扑与姿态表示，从而保证输出格式可直接对齐与比较。其中 CaMN 与 DiffSHEG 使用其官方公开的预训练模型；FaceCapGes 则在相同的数据预处理与骨架设定下由本文训练得到。

对于离线模型（CaMN 与 DiffSHEG），我们将整段演讲作为输入并一次性生成完整动作序列；对于在线模型（FaceCapGes），我们在关闭批处理（Batch=1）的条件下模拟逐帧输入流，并将逐帧输出按时间顺序拼接得到最终序列，以还原实时交互场景下的运行状态。

5.3 用户评估

为验证模型在交互环境中的表现，本文进行了用户主观评估实验，比较 FaceCapGes、CaMN^[19]、DiffSHEG^[15]三个模型在动作自然性、同步性与多样性方面的主观质量。本节首先介绍用户评估系统与实验配置，随后报告主观评价结果与分析。

5.3.1 用户评估系统与实验配置

实验材料与呈现方式 用户评估所使用的手势动画均基于 BEAT 数据集中的测试集语音片段生成，并以 Biovision Hierarchy (BVH) 文件形式保存。BVH 是一种通用的动作捕捉数据格式，通过层级化定义骨骼结构与帧级旋转参数，可直接导入 3D 动画与虚拟人系统。

本文使用的 BVH 文件采用欧拉角旋转表示。由于三种对比模型的输出旋转参数形式不同，因此在导出 BVH 之前，需将输出姿态统一转换为欧拉角表示，以便在后续动画播放中使用统一渲染流程。

本系统当前支持的虚拟人三维模型，需同时具备骨骼绑定，和与 ARKit^[20]兼容的 BlendShape 参数。基于此约束，本实验采用 BEAT 数据集提供的公开的男女两名演讲者三维模型，二者均满足兼容要求，可实现身体与面部的联合驱动。此外，经过 Unity^[24]的 Mecanim^[25]自动骨骼绑定系统匹配，可自动配对模型生成的 BVH 文件中定义的骨骼层级与虚拟人模型的骨骼节点，从而在不依赖手动权重绘制的情况下完成动作重定向。

播放系统实现 我们基于 Unity 自行编写播放脚本，将各模型生成的 BVH 动画用于驱动虚拟人身体骨骼，同时以面部捕捉序列驱动 BlendShape 表情参数，并同步播放原始语音音频。系统支持同时呈现三种模型生成的动画结果：用户可在同一画面中（左、中、右）并行观察三种手势表现，所有语音与面部表情完全一致，唯一变量为身体动作。该设计使参与者能够直接比较不同模型在动作风格、节奏响应与语音同步性方面的差异。

为确保主观评价的公正性与可重复性，系统在每次实验开始前会随机分配三种模型的位置（左、中、右），界面上不会显示模型名称，从而避免潜在偏向。各测试片段的播放顺序在实验前统一设定，以保证不同参与者之间的样本顺序均衡。实验员在播放系统后台记录当前序列与模型对应关系，以便后续结果统计。

实验界面与设备 用户评估系统提供桌面端与虚拟现实（VR）端两种版本，功能完全一致。VR 版本基于 PICO 设备^[27]实现；桌面版支持多窗口并行播放，方便用户同时对比。如图 5.1 所示，播放界面在两种设备上保持统一布局，播放完成后参与者需通过交互界面对三个模型进行排序打分。

VR 用户在沉浸式环境中逐一观看三段动画；桌面端用户则可在单屏上同时观察全部模型。因此前者注重细节感知与临场性，后者更有利于整体风格与节奏的一致性对比。

实验流程与指导 实验正式开始前，研究人员向参与者说明了三项主观评价标准的含义，确保所有被试对评分维度理解一致：

- 真实性：整体动作是否自然流畅，是否存在明显的违和感，如朝向异常或突然抖动；
- 同步性：手势动作与语调、语音节奏是否协调一致；
- 多样性：手势是否丰富多变，避免长时间静止或重复单一动作。

在实验过程中，VR 版本于线下环境进行，桌面版通过线上远程环境执行。两种形式均保持实时交流通道，研究人员可在参与者提问时即时解释操作或澄清评分标准。在正式评估阶段，参与者可多次重播当前片段，但不能返回查看先前内容，以减少记忆偏差。所有播放条件（Unity 场景内的相机角度、光照参数、音量与分辨率设置）在全部被试环境中保持一致，以确保渲染输出的可比性。

需要说明的是，对于 VR 实验，所有测试均在相同的线下实验室环境中进行，使用同一套 PICO 设备与照明条件；而桌面端实验通过远程方式执行，参与者在各自电脑上运行实验程序。研究人员可通过实时屏幕共享观察其操作流程并保持语音沟通，但无法严格控制其所在房间的光照或环境噪声条件。因此，桌面端实验在观看环境上存在一定差异，但由于任务内容与播放系统完全相同，且实验员在测试中持续指导，可认为该差异对结果的总体影响有限。

实验材料与任务设计 评估样本来自 BEAT 数据集中四位演讲者（ID: 2、4、6、8），其中 2、4 为男性，6、8 为女性。每位演讲者各选取两段平均长度约 1 分钟的语音片段，演讲话题互不重复，共组成 8 段固定视频样本。所有实验均使用相同的 8 段样本，但其呈现顺序在不同被试间经过随机化或平衡化处理，以避免顺序效应。每段视频均包含三种模型生成的动作版本（FaceCapGes、CaMN、DiffSHEG），并在播放时随机分配每个模型的动画在屏幕中的排序。参与者在观看每一片演讲音频后，根据三

项主观标准（真实性、同步性、多样性）对三个模型的手势动画表现进行排名评估。

图 5.1 中为用户评估工具的实机界面。画面中共有 3 个虚拟人模型水平分布，在每一片演讲音频播放时，将 3 个生成模型的动画随机分配给 3 个虚拟人的身体骨骼。



图 5.1 用户评估工具实机界面
Figure 5.1 User Study Interface on the Evaluation Device

实验参与者 本实验共邀请 16 名参与者（12 名使用 VR 设备，4 名使用桌面端），涵盖不同性别。所有参与者在实验前均接受了操作说明与校准，并在系统指导下完成评分练习。

为避免呈现顺序对主观印象造成偏差，另设计了采用平衡拉丁方（Balanced Latin Square）顺序的实验版本，使不同参与者观看样本的顺序均衡分布。该版本实验共招募 8 位 VR 用户（4 男 4 女），排序顺序由 HCI 用户评估工具包^[28]自动生成，确保模型与演讲者组合的呈现顺序在全体被试间均匀分布。所有条件保持一致，唯一变量为视频播放顺序。

实验环境说明 为全面验证模型在不同交互场景下的表现稳定性，本次主观评估设置了桌面端与 VR 端两种实验环境，确保覆盖常规屏幕交互与沉浸式交互两类典型应用场景，具体环境配置如下：

- 桌面端环境：参与者通过个人电脑或实验室台式机进行评估，实验界面采用三窗口并行布局，参与者可同时观察左、中、右三个区域的虚拟人动作，聚焦于整体动作风格、节奏同步性的直观对比，注意力分布于整个屏幕的动作全局表现。
- VR 端环境：基于 PICO VR 设备^[27]搭建沉浸式评估场景，参与者佩戴 VR 头显后进入虚拟观测空间，虚拟人以 1:1 比例呈现在眼前，观看距离模拟真人际交流（约 1.2m）。该环境下参与者注意力更易聚焦于虚拟人上半身细节动作，对空间一致性、动作协调感的感知更敏锐。

两种环境的实验流程、评估指标定义及测试样本完全一致，仅通过设备差异构建不同的观察视角与注意力聚焦模式，以验证模型表现的跨设备适配性。

5.3.2 结果与分析

本节综合分析两轮用户评估的统计结果与参与者反馈。所有结果基于 BEAT 测试集中 4 位演讲者（ID:2、4、6、8；2 男 2 女）各 2 段语音片段，共 8 段固定样本。

总体测评结果 如图 5.2 所示，在 16 名参与者的总体评价中，FaceCapGes 在三个维度（真实性、同步性、多样性）上均优于基线模型 CaMN，并在“真实性”维度上略优于离线扩散模型 DiffSHEG。这一结果表明，FaceCapGes 虽在严格的实时因果约束下运行，但仍能保持与非实时生成模型相近的动作自然度与流畅性。

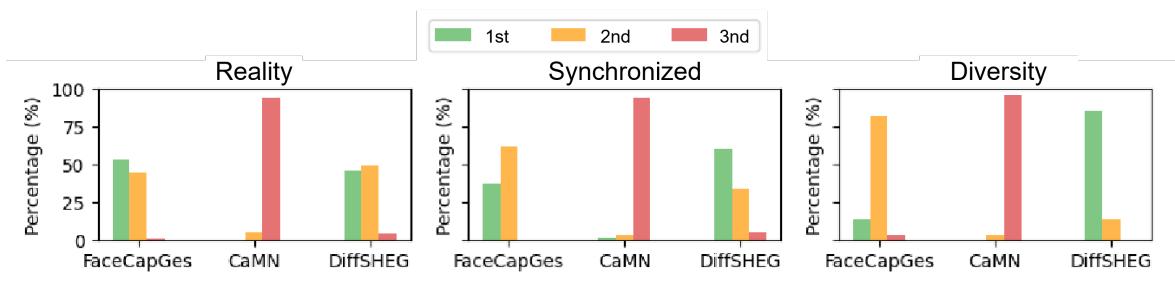


图 5.2 用户评估总体主观排名结果
Figure 5.2 Overall Subjective Ranking Results

平衡拉丁方实验结果 图 5.3 展示了平衡拉丁方设置下 8 位 VR 用户的独立结果，该版本严格控制了模型与演讲者组合的呈现顺序。结果与总体趋势一致，且 FaceCapGes 在“真实性”与“同步性”得到了更好的评价。这表明实验结论在不同顺序条件下保持稳定，进一步验证了模型主观评价结果的鲁棒性。

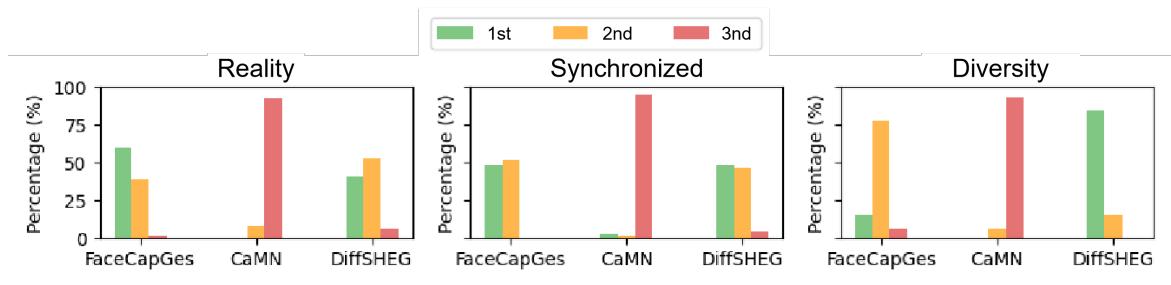


图 5.3 平衡拉丁方设置下的主观排名结果
Figure 5.3 Subjective Ranking Results under Balanced Latin Square Design

用户反馈分析 根据实验后访谈汇总，参与者普遍认为 FaceCapGes 的动作过渡自然、节奏感强，手势响应与语音重音、语调变化更为一致。我们认为，FaceCapGes 融入的头部姿态信息使手势与身体朝向更贴近真实动作，可能是其获得较高真实性评价的重要因素之一。

相比之下，CaMN 在头部与上身动作衔接处常出现僵硬或转向延迟的现象，且头部朝向容易偏离听众方向，从而影响了整体自然度与同步性评分。

对于 DiffSHEG，多数参与者提到其动作丰富度较高，并倾向于将其视为最具多样性的模型。这可能与 DiffSHEG 在生成过程中依赖完整文本输入有关：文本语义边界为动作变化提供了更明确的触发信号，使其更容易生成幅度更大、变化更频繁的动作模式。相比之下，FaceCapGes 主要基于实时语音流与感知模态驱动，缺乏显式语义解析，因此更难实现细粒度的语义级动作响应。

不过，不少参与者也提到 DiffSHEG 在部分片段中存在短暂的手部摆动过快或突然抖动的问题，从而降低了其真实性和同步性的评分。该现象可能与 Axis-Angle 旋转表示在数值空间中存在非连续点或奇异性有关，从而在训练或推理过程中更容易诱发关节角度的突变与抖动。相比之下，本文模型采用 Rot6D^[23] 旋转表示，该表示具有更好的连续性与数值稳定性，有助于缓解由旋转表示引入的突变问题。在本次用户研究中，FaceCapGes 未出现明显的抖动相关反馈，生成动作在关节转动与姿态过渡上保持较为连贯自然。

结果讨论 FaceCapGes 的因果式时间建模与头部姿态融合策略有效提升了局部动作的平滑性与节奏协调，在不依赖未来输入信息的条件下完成逐帧推理，更契合在线实时交互场景的因果性约束。此外，平衡拉丁方版本进一步证明主观结论在不同呈现顺序下的一致性，排除了顺序偏差对结果的显著影响。

综上，用户研究表明 FaceCapGes 在在线实时生成条件下仍能维持与离线模型相

近的主观表现，验证了本文提出的多模态融合与时间建模策略的有效性。

5.4 定性分析

我们强烈建议观看附录中的演示视频（附录 A），内容包含 Ground Truth（GT）、本模型（FaceCapGes）、CaMN 与 DiffSHEG 在不同演讲数据上生成的手势的并排展示，能够直观体现时间对齐性、手势响应性以及头部-身体协调性方面的差异。

为进一步理解模型在实时因果约束下的动态响应行为，我们对每个模型的生成手势中进行了逐帧观察，并选取具有代表性的片段进行案例分析。

5.4.1 生成动作平滑性

如图 5.4 所示，FaceCapGes 在多个片段中均能平滑地响应说话人的语调变化。当语调出现明显上升或下降趋势时，本模型生成的双手高度能够随之连续变化，且动作幅度保持自然，整体运动轨迹连续、关节过渡平稳。该现象与主观用户评估中参与者对 FaceCapGes “动作过渡自然、节奏感强”的反馈一致。

相比之下，CaMN 的生成动作在语调变化较快的片段中表现出一定的迟滞性：双手高度调整通常较为缓慢，且动作幅度变化更趋于保守，导致整体动作轨迹的动态范围较小，视觉上更容易产生“僵硬感”。这一观察与用户反馈中提到的 CaMN “转向延迟与衔接僵硬”现象相一致。

DiffSHEG 的生成动作整体更活跃，在部分片段中能够在更细粒度的时间尺度上产生频繁的手势变化。然而，我们也观察到其在少数时间步出现局部关节的突然加速或短暂抖动，表现为手部轨迹在相邻帧间产生明显跳变或方向快速反转。该现象与主观用户反馈中提到的“偶发抖动”一致，可能与其旋转表示在数值空间中存在非连续点有关，从而使推理阶段更容易产生局部突变。

总体而言，FaceCapGes 通过采用 Rot6D 表示并结合帧级因果时间建模，在保证实时性的同时维持了更高的动作连续性与稳定性，生成结果在视觉平滑性上更接近真实动作序列。

5.4.2 头部朝向与手势空间指向一致性

为分析头部姿态输入对生成手势空间指向的影响，我们选取测试集中 GT 头部朝向发生明显偏转的片段，并对比不同模型在该时刻生成动作的朝向一致性表现。这里的“空间指向一致性”指生成手势的主要运动方向是否与角色的头部/身体朝向保持匹配，从而反映模型是否能够利用非语言姿态线索生成更符合交互场景的空间表达。

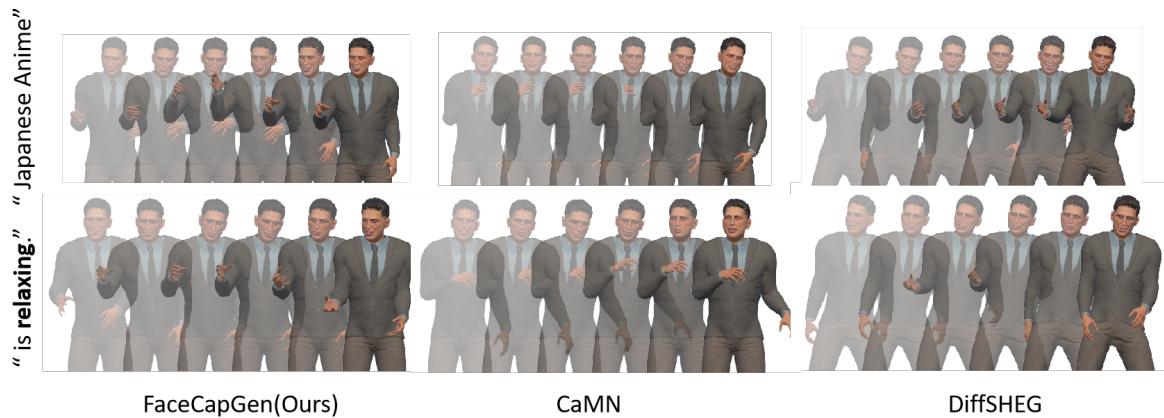


图 5.4 生成动作效果对比
Figure 5.4 Qualitative Comparison of Generated Gestures

值得注意的是，CaMN 与 DiffSHEG 在输入端均不显式包含头部姿态信息，因此其生成的手势方向主要依赖语音或文本内容，难以体现真实头部转向带来的空间指向变化。

图 5.5 展示了一个包含明显头部转向变化的片段截图，从左到右依次为 GT、本模型（FaceCapGes）、CaMN 与 DiffSHEG 的生成结果。在该片段中，GT 的身体与头部朝向呈现出明确的空间指向，并且手势多沿着当前头部/躯干朝向展开，体现出面向不同听众时的自然交流习惯。

我们观察到，本模型在多数帧中能够保持与 GT 相近的身体朝向，并使双手动作的空间指向与头部朝向保持一致，例如在头部向右侧转动的阶段，生成的手势也倾向于朝向相同方向展开。相比之下，CaMN 的动作表现更为僵硬，身体朝向变化较弱且缺乏稳定的空间锚定，手势方向在多帧间呈现随机波动。DiffSHEG 虽能在其可获取的语音信息下生成较连贯的手势节奏与整体动作幅度，但由于其输入不包含头部姿态信号，生成结果难以反映 GT 中由头部转向引起的空间指向变化。

该现象表明，在严格因果与实时输入条件下，头部姿态作为额外的非语言模态能够为手势生成提供空间指向上的中层约束，使动作更自然地与说话者的注意方向和交互对象匹配。这种头-手空间一致性在元宇宙等虚拟交互场景中尤为重要：当用户面向不同方向的听众或对象进行交流时，头部转向与与之匹配的手势能够增强空间合理性与沉浸感，从而提升多方位交流中的可理解性与聆听体验。



图 5.5 头部朝向与手势指向一致性的动作对比

Figure 5.5 Comparison of Head Orientation and Gesture Direction Consistency

5.5 客观评估指标与实现细节

为客观层面评价模型在动作自然性、节奏同步性与多样性等方面的表现，本文在客观评估中采用四项度量：手势分布相似度（Fréchet Gesture Distance, FGD）^[29]、语义相关动作召回率（Semantic Relevance Gesture Recall, SRGR）^[19]、节奏对齐度（Beat Alignment, BA）^[19,30]以及 L1 范数（L1DIV）。这些指标分别对应生成动作在分布一致性、语音同步性与变化丰富性等不同维度，共同构成对模型质量的综合评估体系。

5.5.1 手势分布相似度（Fréchet Gesture Distance）

手势分布相似度（Fréchet Gesture Distance, FGD）^[29]用于衡量生成手势分布与真实手势分布之间的统计距离，灵感源自图像生成领域的 Fréchet Inception Distance (FID)。不同于图像任务直接利用 Inception 网络特征，在动作生成领域，特征空间需由单独训练的动作自编码器定义。该自编码器通过重构任务学习手势的潜在表示，使潜在空间具备对运动模式的压缩与区分能力。在该潜在空间中，假设真实分布与生成分布的高维嵌入向量分别为 $\mathcal{N}(\mu_r, \Sigma_r)$ 与 $\mathcal{N}(\mu_g, \Sigma_g)$ ，其中， $\mathcal{N}(\cdot)$ 为高斯分布， μ 与 Σ 分别为嵌入特征的均值向量与协方差矩阵。

此时，FGD 定义为：

$$\text{FGD} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}). \quad (5.3)$$

其中， $\text{Tr}(\cdot)$ 为矩阵迹运算。

较小的 FGD 值表示生成动作的统计分布更接近真实数据，可反映动作的整体自然度与风格一致性。

FGD 的模型结构与训练配置 本文在每位说话人的训练集上分别训练一组评估用自编码器，以避免跨说话人分布差异对指标的干扰。自编码器输入为以 Rot6D 表示的上半身骨架序列，训练目标为最小化位置与速度的重构误差：

$$\mathcal{L}_{AE} = \lambda_r \|\hat{\mathbf{g}} - \mathbf{g}\|_2^2 + \lambda_v \|\hat{\mathbf{g}}' - \mathbf{g}'\|_2^2, \quad (5.4)$$

其中 $\lambda_r = 1$, $\lambda_v = 0.1$, \mathbf{g}' 为速度序列。

训练配置如下：输入片段长度设为 32 帧，训练片段切割步长设为 10 帧，批大小设为 256，编码器与解码器的隐藏层维度均为 128。优化器采用 SGD^[26]，基础学习率设为 $\text{lr}_{base} = 1.2 \times 10^{-4}$ ，并按批大小线性缩放为

$$\text{lr} = \text{lr}_{base} \cdot \frac{\text{batch_size}}{128}. \quad (5.5)$$

训练过程中仅使用位置与速度重构损失（式（5.4）），不包含加速度或对抗项。自编码器共训练 400 个 epoch。

用于 FGD 评估的骨架感知自编码器 在基线模型 CaMN 的 FGD 评估中，采用了一种基于时间卷积的嵌入式自编码器（embedding-based autoencoder）进行手势特征提取。该结构将每一帧姿态平铺为高维向量输入，并对各关节分量进行独立建模。在实际使用中我们观察到，这类嵌入空间对旋转表示的数值尺度较为敏感。当采用 Rot6D 表示时，不同关节与分量之间的不均衡方差可能在潜在空间中被进一步放大，从而导致协方差估计条件较差，并引发 FGD 数值不稳定的问题。

为提高 FGD 评估的鲁棒性，本文采用了一种骨架拓扑感知的自编码器作为特征提取器。该模型在编码阶段显式引入骨架邻接矩阵 A ，并通过对相邻关节进行局部卷积实现结构约束。具体而言，第 l 层中关节 i 的特征向量 $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_l}$ ，其中 d_l 表示第 l 层中每个关节节点的特征维度。通过聚合其邻域 $N(i)$ 内相邻关节 j 的特征 $\mathbf{h}_j^{(l)}$ 进行更新，其中 A_{ij} 表示骨架邻接矩阵中节点 j 到节点 i 的连接权重，即

$$A_{ij} = \begin{cases} 1, & \text{若关节 } i \text{ 与 } j \text{ 在骨架拓扑中直接相连, 或 } i = j; \\ 0, & \text{否则.} \end{cases} \quad (5.6)$$

此外，设 $W^{(l)} \in \mathbb{R}^{d_{l+1} \times d_l}$ 与 $b^{(l)} \in \mathbb{R}^{d_{l+1}}$ 分别为第 l 层的可学习权重矩阵与偏置项。设 $\sigma(\cdot)$ 为非线性激活函数，在本文实现中取为双曲正切函数 $\tanh(\cdot)$ 。上述局部邻域聚合过程可形式化表示为：

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{j=1}^J A_{ij} W^{(l)} \mathbf{h}_j^{(l)} + b^{(l)} \right), \quad (5.7)$$

这种基于局部邻域的权重共享机制有助于保持人体运动的空间结构一致性，并在特征提取过程中缓解由旋转表示差异带来的数值尺度放大问题。

实验结果表明，该骨架感知自编码器在 Rot6D 下产生更加稳定的潜在分布统计量。在本文的实验设置中，该设计有效提升了 FGD 评估的数值稳定性，并有效避免了在 Rot6D 表示条件下使用嵌入式自编码器时出现的极端 FGD 数值现象。

5.5.2 语义相关动作召回率（Semantic Relevance Gesture Recall）

语义相关动作召回率（Semantic Relevance Gesture Recall, SRGR）^[19] 用于衡量生成手势在语义相关时间段内与真实手势在数值层面的匹配程度。该指标关注的是语

音语义触发的关键手势是否被正确生成，而非整体分布一致性或语音-手势节奏相关性。

与基于分布的评估指标（如 FGD）不同，SRGR 属于基于阈值的逐帧召回率度量，通过统计生成手势在允许误差范围内命中的比例，反映模型在语义相关动作重现方面的准确性。

定义与原理 在本文实现中，SRGR 作用于关节的旋转表示。设真实手势序列与生成手势序列在第 t 帧第 j 个关节的旋转表示分别为 $\mathbf{r}_{t,j}$ 与 $\hat{\mathbf{r}}_{t,j}$ ，其中 $\mathbf{r}_{t,j} \in \mathbb{R}^6$ 为关节的 Rot6D 表示， T 为序列总帧数， J 为关节数量。

在给定旋转表示误差阈值 δ 的条件下，若生成关节旋转与真实关节旋转之间的表示差异满足 $\|\mathbf{r}_{t,j} - \hat{\mathbf{r}}_{t,j}\|_1 < \delta$ ，则认为该关节在该时刻被成功召回。在本文实验中，阈值固定设为 $\delta = 0.5$ 。

SRGR 通过对所有时间帧与关节进行统计，并引入语义相关性权重 λ_t ，定义为：

$$\text{SRGR} = \frac{1}{TJ} \sum_{t=1}^T \sum_{j=1}^J \lambda_t \mathbb{I}(\|\mathbf{r}_{t,j} - \hat{\mathbf{r}}_{t,j}\| < \delta), \quad (5.8)$$

其中 $\mathbb{I}(\cdot)$ 为指示函数。

语义相关性权重 λ_t 用于强调语音中语义显著时间片段（如强调词或语义触发点）对应的手势匹配程度，由 BEAT 数据集提供的语义标注确定，从而使 SRGR 更加关注语义相关手势的召回情况，而非对所有时间片段进行均匀统计。

5.5.3 节奏对齐度 (Beat Alignment)

节奏对齐度 (Beat Alignment, BA) 指标用于衡量语音节拍事件与手势关键动作事件在时间轴上的对齐程度，反映模型在语音 - 手势时序同步性 (temporal synchronization) 方面的性能。本文采用 BEAT/CaMN 中使用的 BeatAlign 指标^[19]，其原始形式由 Li 等人提出于 AI Choreographer 工作中^[30]，并可视为音频节拍与动作节拍集合之间的单向 Chamfer 相似度度量。

与 SRGR 基于阈值的逐帧数值匹配不同，BA 关注的是离散事件层面的时间对齐关系，即手势关键动作是否在时间上合理地响应了语音中的重读节拍。

定义与原理 设语音节拍事件集合为 $\mathcal{B}_s = \{t_k^s\}$ ，通过对语音信号的能量相关特征进行起始点检测 (onset detection) 得到。具体而言，首先基于谱能量变化计算 onset strength 曲线，并在该曲线上进行峰值检测以获得候选语音事件位置。随后，引入 Root

Mean Square (RMS) 特征作为短时能量幅度的描述，并在 RMS 曲线上对检测到的起始点进行 backtracking 校正，从而将语音节拍事件定位至能量实际开始上升的位置，以提高时间定位的稳定性与准确性。

手势关键动作事件集合为 $\mathcal{B}_g = \{t_m^g\}$ ，定义为关节运动速度的局部极小值点，对应动作中的停顿或方向变化等显著运动事件。

对于每一个语音节拍事件 t_k^s ，计算其与所有手势关键动作事件之间的最小时间偏差：

$$\Delta t_k = \min_m |t_k^s - t_m^g|. \quad (5.9)$$

随后采用高斯核函数将时间偏差映射为相似度分数，从而得到单个语音节拍的对齐得分。最终 BA 指标定义为：

$$BA = \frac{1}{|\mathcal{B}_s|} \sum_k \exp\left(-\frac{(\Delta t_k)^2}{2\sigma^2}\right), \quad (5.10)$$

其中 σ 为时间尺度参数，本文中取 $\sigma = 0.3$ ，用于控制对齐容忍范围。

该定义可视为从语音节拍集合到手势节拍集合的单向 Chamfer 相似度，当语音节拍与手势关键动作在时间上高度对齐时，BA 值接近 1；反之，当二者时间偏差较大时，BA 值趋近于 0。

本文使用 BA 指标评估模型在语音重读节拍与手势关键动作之间的时间同步能力，该指标与主观观察到的语音-手势同步自然度通常具有较好一致性。

5.5.4 L1 范数

L1 范数 (L1DIV)^[19] 用于衡量模型生成手势序列的多样性，即不同生成样本之间在动作空间中的差异程度。该指标反映模型在避免生成结果收敛到平均动作模式 (mode collapse) 的同时，是否能够保持足够丰富的动作变化。

定义与原理 设模型在评估过程中生成 N 个手势序列样本，第 i 个生成样本在第 t 帧第 j 个关节的旋转表示为 $\hat{\mathbf{r}}_{t,j}^{(i)}$ ，其中 $\hat{\mathbf{r}}_{t,j}^{(i)} \in \mathbb{R}^6$ 为关节的 Rot6D 表示， T 为序列总帧数， J 为关节数量。

L1DIV 通过计算不同生成样本之间在所有时间帧与关节上的平均 L1 距离，来刻画生成动作分布的离散程度。其数学形式定义为：

$$L1DIV = \frac{1}{N(N-1)} \sum_{i < k} \frac{1}{TJ} \sum_{t=1}^T \sum_{j=1}^J \left\| \hat{\mathbf{r}}_{t,j}^{(i)} - \hat{\mathbf{r}}_{t,j}^{(k)} \right\|_1. \quad (5.11)$$

较高的 L1 范数表明模型生成具有较强的多样性，但过高可能意味着动作不稳定或噪声放大。因此，L1 范数通常与 FGD 联合分析：FGD 反映真实度，L1 范数反映丰富度，两者共同平衡模型在自然性—多样性维度上的表现。

5.5.5 评估区域设定与公平性说明

由于本文方法 FaceCapGes 在输入端显式引入了真实的头部姿态作为额外模态，并在输出中生成包含头部旋转的上半身骨骼序列，若直接将头部旋转也计入整体误差，可能会对不使用头部姿态输入的对比方法（如 CaMN 与 DiffSHEG）造成不公平的优势。为保证跨方法的可比性，本文在所有定量评估指标中分别报告两种评估区域：

- (1) 上半身（含头部）：包含头部与上半身全部关节旋转；
- (2) 上半身（不含头部）：在计算指标时移除头部关节的旋转分量，仅统计身体与手臂部分。

具体而言，在计算 FGD、SRGR、BA 与 L1DIV 时，我们将头部关节的旋转维度从指标计算中排除，从而消除头部旋转差异对指标的直接影响。

5.6 定量评估结果

表 5.2 定量评估结果
Table 5.2 Quantitative Evaluation Results

区域	方法	FGD↓	SRGR↑	BA↑	L1DIV↑
上半身（含头部）	CaMN	47.732	0.098	0.845	7.591
	DiffSHEG	<u>26.846</u>	0.109	<u>0.883</u>	<u>11.028</u>
	本模型	23.385	<u>0.107</u>	0.913	13.284
上半身（不含头部）	CaMN	49.437	0.103	0.845	7.327
	DiffSHEG	<u>26.847</u>	0.110	<u>0.883</u>	<u>10.990</u>
	本模型	23.384	<u>0.107</u>	0.913	13.330

表 5.2 显示，FaceCapGes 在所有指标上均优于 CaMN。值得注意的是，“上半身（含头部）”与“上半身（不含头部）”两种评估区域下的结果趋势基本一致，表明本文方法的性能优势并非仅来源于头部姿态输入或头部输出的额外信息。

此外，表 5.2 还显示，与 DiffSHEG 相比，本模型 FGD 更低，SRGR 相近，且在 L1 范数上表现最优，表明其生成动作具备良好多样性。但这一结论与用户主观评分存在一定出入：DiffSHEG 在多样性上主观排名更高。

这个现象可能来自于 L1 范数的局限性：它主要衡量空间偏离程度，不能直接体现动作的颗粒度，或用户感知上的丰富性。虽然我们的模型在动作结构上更丰富，但用户普遍认为 DiffSHEG 的动作更活跃，在合理的时机做出了更多吸引注意的动作。

5.7 消融实验分析

我们围绕两个核心设计展开消融：

- (1) 头部姿态输入是否能提供有效空间与时序线索；
- (2) 帧级自回归生成与滑动窗口训练是否优于片段级一次性解码。

为此，我们构造了三种变体：移除头部姿态输入、移除帧级生成策略，以及基线 CaMN。

所有消融实验均基于 BEAT 数据集的第 2 位说话人进行，训练与测试划分与主实验保持一致。

表 5.3 展示了各模块在四个指标上的结果。

**表 5.3 消融实验结果
Table 5.3 Ablation Study Results**

区域	变体	FGD↓	SRGR↑	BA↑	L1DIV↑
上半身（含头部）	CaMN	32.870	0.111	0.858	7.214
	移除头部姿态	19.591	0.123	0.916	10.642
	移除帧级生成	21.592	0.125	0.892	10.456
	FaceCapGes（本模型）	19.290	0.123	0.918	10.871

头部姿态输入的贡献 对比“移除头部姿态”与完整模型可以发现，尽管两者在 SRGR 上差异不大（均为 0.123），但完整模型在 FGD、BA 与 L1DIV 上均取得更优表现，尤其在动作分布一致性（FGD）与多样性（L1DIV）上呈现出稳定收益。这表明头部姿态作为非语言模态提供了额外的空间方向与交互焦点线索，能够帮助模型在生成手势时保持更一致的身体朝向与动作指向，从而提升动作自然度与表达丰富性。

需要注意的是，该提升幅度相对温和，原因可能在于语音与面部表情已包含较强的节奏与情绪提示，头部姿态主要补充空间层面的约束，因此其收益更集中地反映在分布与多样性相关指标上。

帧级自回归生成的优势 值得注意的是，“移除帧级生成”版本允许利用未来上下文并采用双向 LSTM，但其整体表现仍不及帧级版本，尤其在 FGD 上出现较为明显的

退化 (FGD 从 19.290 上升至 21.592)。

一种合理解释是, 该变体在训练阶段以独立窗口为单位进行优化, 而在测试阶段采用整段演讲作为长序列输入并一次性生成完整动作序列。这种训练与推理的序列长度范围不一致, 可能导致模型在长序列推理时的隐状态传播方式偏离训练分布, 从而使生成动作在嵌入空间中的统计特性偏离真实数据分布, 表现为生成分布与真实动作分布的距离增大 (FGD 上升)。

尽管本模型同样采用窗口化训练, 但在每个窗口内通过滑动窗口展开与纯自回归预测进行多步生成, 并对每步输出的误差累计取平均作为优化目标, 使模型在训练阶段即暴露于自身预测分布并学习局部动力学的稳定性。因此训练目标与在线推理过程保持一致, 对推理阶段序列长度变化所引入的分布偏移具有更强鲁棒性。

与基线模型 CaMN 的差异 相比基线 CaMN, 本模型在所有指标上均显著提升, 其中 FGD 从 32.870 降至 19.290, 表明生成分布更接近真实动作; 同时 BA 与 L1DIV 的提高说明动作更平衡且更具表达多样性。这进一步验证了本文引入的因果时序建模、头部模态补充与滑动窗口训练策略对于实时交互场景下的手势生成具有有效作用。

此外, 基线 CaMN 采用欧拉角而本模型采用 Rot6D 表示, 该表示差异亦可能部分解释性能提升幅度较大的原因。

5.8 性能评估

5.8.1 单帧推理性能

FaceCapGes 作为端到端实时手势生成框架的核心计算模块, 其性能评估聚焦于单帧输入, 单帧输出的核心推理流程, 即模型接收当前帧的语音、面部表情与头部姿态多模态输入, 实时输出对应帧的上半身手势骨骼动画。

为模拟真实应用中的实时输入流场景, 性能测试基于 BEAT 数据集的测试集展开, 关闭批处理机制, 确保每帧数据均独立输入模型进行推理, 还原逐帧处理的实际运行状态。测试过程中, 我们将测试集中总计 93015 帧的多模态输入数据传入模型, 记录从首帧输入到末帧输出的总推理耗时, 通过总时长与测试帧数的比值计算平均单帧处理时间。测试时使用的硬件配置为单张 RTX4090 硬件。

测试结果如表 5.4 所示, 模型平均单帧处理时间为 6.07 毫秒, 具备良好的实时响应能力。

表 5.4 推理速度评估结果
Table 5.4 Inference Speed Evaluation Results

指标	时间
测试帧数	93015 (f)
推理总时长	504 (s)
平均单帧时间	6.07E-03 (s/f)

5.8.2 端到端计算链路延迟

结合第 2.4 节所述的端到端框架流程，系统端到端延迟可按时间顺序拆解为数据采集、特征提取、模型推理与结果返回四个核心阶段。各阶段性能消耗及瓶颈分析如下：

- 数据采集阶段：依赖设备端传感器实时捕获多模态信号，主要耗时来源于 ARKit 面部与头部姿态追踪。根据官方文档标注，在 iOS 设备上 ARKit 的目标追踪帧率为 60 FPS^[2]，对应输入更新周期约为 16.7 ms。该阶段耗时由设备端硬件算力与系统负载决定。
- 特征提取阶段：将原始传感器数据转换为模型可识别的结构化特征（语音 Mel 频谱、面部 BlendShape 系数、头部 Rot6D 旋转参数）。测试原型中采用 Librosa 对测试集音频进行离线特征提取，93015 帧数据的总计算耗时约 61 秒，对应的平均单帧计算成本约为 0.66 ms。需要注意的是，该统计不包含实时 I/O 与缓冲管理等系统开销，但可用于估计音频特征提取的计算量级。实际部署时可替换为 PyAudio 等实时流式提取工具，因此该阶段预计不构成主要性能瓶颈。
- 模型推理阶段：为端到端流程的核心计算环节，结合第 5.8.1 节的模型推理性能测试，在单张 RTX4090 硬件、无批处理（Batch=1）配置下，单帧推理耗时为 6.07 ms。该阶段耗时与硬件算力强相关，是由神经网络结构设计决定的主要性能变量。
- 结果返回阶段：将模型输出的骨骼姿态参数传输至渲染引擎，耗时可忽略（通常低于 0.1 ms），不构成性能瓶颈。

在本地推理设置下，若将一次响应链路定义为单帧采集完成后立即进入后续计算，则计算链路延迟可近似表示为：

$$t_{e2e} \approx t_{\text{ARKit}} + t_{\text{feat}} + t_{\text{infer}} + t_{\text{return}}, \quad (5.12)$$

其中 $t_{\text{ARKit}} \approx 16.7 \text{ ms}$, $t_{\text{feat}} < 1 \text{ ms}$, $t_{\text{infer}} = 6.07 \text{ ms}$, $t_{\text{return}} < 0.1 \text{ ms}$ 。因此在该设置下，计算链路理论延迟可视为 25 ms 以下。

需要注意的是，该估计未计入音频特征提取的窗口缓存与渲染端同步可能引入的额外等待，实际交互延迟还将受渲染刷新周期影响。

远程推理部署的额外开销 在移动端等算力受限的平台上，模型推理可部署于远程服务器并通过网络进行输入输出传输。此时端到端延迟需进一步加上网络往返与序列化开销：

$$t_{\text{e2e}}^{\text{remote}} \approx t_{\text{ARKit}} + t_{\text{feat}} + t_{\text{net}} + t_{\text{infer}} + t_{\text{return}}, \quad (5.13)$$

其中 t_{net} 表示网络传输与通信开销，其大小由网络条件与系统实现决定，本文不在此展开测量。

5.8.3 系统更新率

在本实验设置下，模型在单张 RTX4090 上的单帧推理时间为 6.07 ms，对应的理论推理吞吐约为 165 FPS，说明模型推理阶段在当前硬件条件下不会成为实时交互中的性能瓶颈。需要强调的是，该数值仅反映神经网络推理吞吐能力，并不等同于系统端到端更新率。

然而，端到端系统的实际更新率仍将受到输入采集频率与渲染刷新率的共同约束。根据官方文档，ARKit 面部与头部追踪通常以 60 FPS 为目标更新率^[2]，因此在采用 ARKit 作为面部与头部输入来源的应用场景中，系统可获得的相关输入更新频率理论上可视为 60 Hz，相应地，系统的有效输出更新率亦不会超过 60 FPS。在远程推理部署下，网络传输与同步开销可能进一步降低实际更新率。

5.8.4 帧率设定的可扩展性

本文实现中采用 15 FPS 的动作时间采样率进行训练与输出，主要原因在于与 BEAT 数据集预处理及基线模型的设置保持一致，以便进行公平对比。该采样率选择并不构成本模型的固有限制。

但需要注意的是，本文模型的输入模态不包含帧间时间间隔 Δt 的显式信息，因此训练过程中隐式假设固定的离散时间步长（即 $\Delta t = 1/15 \text{ s}$ ）。当部署环境的输入采样率或输出刷新率发生变化（例如 ARKit 为 60 FPS）时，若直接使用 15 FPS 训练的模型进行逐帧推理，模型可能会对运动速度与节奏尺度产生偏差，从而影响手势动态表现。

因此，在目标运行帧率与训练帧率不一致的情况下，更稳妥的做法是对数据集进行对应帧率的重采样并重新训练模型，以确保训练时域与实际系统时域一致，从而

使模型学习到正确的动力学时间尺度。未来也可进一步探索将 Δt 作为显式条件输入，提高模型对不同采样率输入的鲁棒性。

在实际部署中，系统可依据端到端管线中的主要瓶颈选择合适的目标动作帧率：当推理计算为瓶颈（如移动端或低算力设备）时，可维持较低帧率以确保每帧按时生成；当输入采样率与计算资源允许（例如更高刷新率的追踪）时，可采用更高帧率训练版本以获得更细粒度的时间响应与动作细节表达。

5.9 本章小结

本章围绕 FaceCapGes 的生成质量与实时性能开展了系统评估。

在主观评估方面，本章结合两轮用户评估对模型在真实感、同步性与多样性三个维度进行了分析，结果显示本方法整体优于基线模型 CaMN，并在真实感维度上与离线模型表现相近。结合用户反馈与访谈，本章进一步讨论了头部姿态信息在提升动作朝向一致性与真实感方面的潜在贡献。此外，通过对生成结果的定性分析，我们观察到本方法能够随头部朝向变化生成方向一致的手势动作，从而增强交互场景下的空间合理性。

在客观评估方面，本文采用 FGD、SRGR、BA 与 L1DIV 等指标，分别从动作分布一致性、语音节奏对齐、动作平衡性与多样性等角度对比了 CaMN 与 DiffSHEG 等方法，并通过“含头部/不含头部”两种评估区域设定保证指标对比的公平性。在此基础上，本章进一步通过消融实验分析了头部姿态输入与帧级自回归生成策略对性能提升的贡献，验证了各模块设计的有效性。

最后，本章对模型推理速度、端到端计算链路延迟与系统更新率进行了评估，表明 FaceCapGes 在保持较高生成质量的同时具备实时交互所需的低延迟响应能力。

第6章 结论

6.1 本文工作总结

本论文围绕“面向用户交互的在线实时数字人驱动”这一核心目标，针对现有语音驱动手势生成方法普遍依赖整句输入、难以满足严格实时约束的问题，提出了一种仅使用在线可获取信号（语音、面部表情与头部姿态）即可逐帧生成上半身骨骼动作的帧级生成方法 FaceCapGes，实现了无需用户实际做出手势的低门槛数字人自然表达能力。论文的主要工作与贡献可总结如下：

(1) 构建了面向在线实时数字人驱动的多模态帧级手势生成系统框架。针对在线交互场景下语音逐字输入带来的未来信息不可用问题，本文从系统层面提出了严格因果的实时生成机制，明确了从信号采集、特征同步、帧级推理到虚拟人驱动渲染的完整流水线结构，并在实时约束下制定了输入模态选择、姿态表示方式以及端到端数据流组织策略。该框架保证了模型能够在仅依赖当前与历史信息的情况下持续输出动作流，为后续的可部署实现提供了系统基础与设计规范。

(2) 在级联多模态架构中引入头部姿态作为新的实时输入模态，并提出弱耦合融合策略以增强节奏与指向一致性。本文在继承 CaMN 级联设计思想的基础上，将头部姿态视为实时可获取的辅助输入信号，用于提供节奏前瞻与空间锚定线索，从而弥补语音模态在严格因果条件下对方向一致性与互动焦点建模不足的问题。为此，论文设计了头部姿态编码器，并通过弱耦合方式将其作为独立通道拼接进多模态隐向量，避免其与语音/表情编码产生过强耦合导致收敛困难。该设计在保持在线实时推理能力的同时，使模型能够显式利用用户头部朝向变化，从而增强生成动作的空间协调性与指向一致性。

(3) 提出并实现了基于滑动窗口展开的自回归训练策略，使训练过程与在线推理严格一致，从而提升实时生成的稳定性与连续性。为解决帧级自回归生成中常见的漂移与抖动问题，本文提出了结合片段切割与滑动窗口展开的训练流程，将单步因果预测器在片段内部逐帧展开，并采用历史动作缓存写回机制以模拟真实在线推理过程，使训练与部署阶段保持一致。此外，论文在片段生成区间内引入姿态、速度与加速度的多尺度监督约束，并结合片段级判别器进行对抗训练，从局部运动统计分布层面进一步提升生成动作的自然性与时间连续性。该训练策略在严格因果约束下有效缓解了自回归误差累积带来的稳定性下降问题。

(4) 构建了完整的评估平台并进行了主客观与性能实验验证, 证明 FaceCapGes 在严格因果条件下仍具备良好的生成质量与实时推理能力。论文搭建了统一的跨模型评估与渲染平台, 使不同方法可在相同输入条件与渲染设置下公平对比, 并在 BEAT 数据集上对 FaceCapGes、CaMN 与 DiffSHEG 等代表性方法开展了用户主观评估、客观指标测量, 并进一步对 FaceCapGes 进行了消融分析与实时性能测试。实验结果表明, 在严格因果约束下, FaceCapGes 的生成真实性可达到与扩散模型方法相当的水平; 同时在韵律变化较强的语音片段及头部方向变化明显的场景中, 其动作表现更平滑且空间指向与真实数据更一致, 体现出该方法在虚拟交互场景中保持空间表达一致性优势。此外, 模型帧级推理效率与端到端链路延迟满足实时交互应用的运行需求, 验证了其作为可部署在线数字人驱动方案的可行性。

综上所述, 本论文在严格因果约束下, 提出了基于语音、面部捕捉与头部姿态的在线帧级手势生成方法 FaceCapGes, 并从系统设计、模型结构、训练策略与实验验证四个方面证明了在无需未来信息的条件下实现自然随语手势生成的可行性与应用价值。FaceCapGes 在严格实时约束下实现了无需手势采集的自然随语动作生成, 可支持实时虚拟人直播、沉浸式虚拟社交互动等交互式数字人应用场景, 提升了数字人表达的自然性与互动一致性。与此同时, 本研究也为未来进一步融合高层语义信息与预测性控制机制的实时数字人驱动方法提供了技术基础与参考。

6.2 未来工作展望

6.2.1 高层语义信息

当前模型主要关注语音声学特征与运动感知模态对手势生成的影响, 尚未显式引入语言层面的语义理解与表达意图建模。未来可结合实时语音识别与增量式语义解析技术, 引入语篇结构、强调意图或对话功能等高层语义信息, 以丰富手势在交互场景中的表达能力。在不破坏实时性的前提下, 探索对有延迟但可修正的语义假设的鲁棒利用方式, 将有助于提升生成手势在语义层面的准确性与一致性。

6.2.2 面向未来趋势的预测性训练目标

从建模目标的角度来看, 当前 FaceCapGes 的训练过程主要以当前时间步手势姿态的重建误差为优化目标, 即在给定历史与当前多模态输入的条件下, 监督模型对当前手势的预测精度。然而, 该学习目标并未对未来时间段内手势节奏与结构变化施加显式约束, 使模型对历史信息的利用更多服务于当前帧生成, 而非对即将发生动作

变化进行前瞻性建模。

未来的研究可在现有框架基础上，引入针对手势未来趋势的预测性监督信号，尤其是充分挖掘头部与面部动态中所蕴含的准备性线索。与直接预测未来具体手势姿态不同，该方向更侧重于对抽象化时序属性的建模，例如未来短时间窗口内的手势起始概率、运动能量变化或强调强度等。这类趋势性变量具有时间平滑、语义明确且可提前出现的特点，适合作为实时系统中的前瞻性约束。

通过在训练阶段同时优化当前手势生成与未来趋势预测两个目标，模型有望学习到更具时间结构性的中间表示，从而在不引入额外模态或显著增加系统延迟的前提下，实现对手势节奏的提前准备与更稳定的时序对齐。

参考文献

- [1] KARTYNNIK Y, ABLAVATSKI A, GRISHCHENKO I, et al. Real-time facial surface geometry from monocular video on mobile gpus[J]. arXiv:1907.06724, 2019.
- [2] NYISZTOR K. Introduction to Augmented Reality with ARKit[EB/OL]. 2019. <https://www.pluralsight.com/resources/blog/guides/introduction-to-augmented-reality-with-arkit>.
- [3] KENDON A. Gesture: Visible Action as Utterance[M]. Cambridge, UK: Cambridge University Press, 2004.
- [4] MCNEILL D. Hand and Mind: What Gestures Reveal about Thought[M]. Chicago, IL: University of Chicago Press, 1992.
- [5] WAGNER P, MALISZ Z, KOPP S. Gesture and speech in interaction: An overview [J/OL]. Speech Communication, 2014, 57: 209-232. DOI: 10.1016/j.specom.2013.09.008.
- [6] HADAR U, BUTTERWORTH B. Iconic gestures, imagery, and word retrieval in speech[J]. Semiotica, 1989, 75(1/2): 63-83.
- [7] ESTEVE-GIBERT N, PRIETO P, PONS X, et al. The timing of head movements: The role of prosodic heads and edges[J/OL]. The Journal of the Acoustical Society of America, 2017, 141(6): 4727-4739. DOI: 10.1121/1.4986649.
- [8] CASSELL J, VILHJÁLMSSON H H, BICKMORE T. BEAT: the Behavior Expression Animation Toolkit[C/OL]//SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. 2001: 477-486. <http://doi.org/10.1145/383259.383315>. DOI: 10.1145/383259.383315.
- [9] HUANG C M, MUTLU B. Robot behavior toolkit: generating effective social behaviors for robots[C/OL]//HRI '12: Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction. 2012: 25-32. <https://doi.org/10.1145/2157689.2157694>. DOI: 10.1145/2157689.2157694.
- [10] KIPP M. Gesture generation by imitation: from human behavior to computer character animation[C/OL]//. 2005. <https://api.semanticscholar.org/CorpusID:26271318>.
- [11] LIU H, ZHU Z, BECHERINI G, et al. Emage: Towards unified holistic co-speech ges-

- ture generation via expressive masked audio gesture modeling[J]. arXiv:2401.00374, 2024.
- [12] ZHANG Z, AO T, ZHANG Y, et al. Semantic Gesticulator: Semantics-Aware Co-Speech Gesture Synthesis[J]. ACM Transactions on Graphics (TOG), 2024, 43(4): 1-17.
- [13] 万浩聪, 刘长红, 杨海, 等. 基于细粒度时空建模的语音驱动手势生成模型[C/OL] //第二十四届中国计算语言学大会论文集 (CCL 2025). 济南, 中国: 中国中文信息学会计算语言学专业委员会, 2025: 684-695. <https://aclanthology.org/2025.ccl-1.51.pdf>.
- [14] ZHU L, LIU X, LIU X, et al. Taming Diffusion Models for Audio-Driven Co-Speech Gesture Generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 10544-10553.
- [15] CHEN J, LIU Y, WANG J, et al. DiffSHEG: A Diffusion-Based Approach for Real-Time Speech-driven Holistic 3D Expression and Gesture Generation[C]//CVPR. 2024.
- [16] YANG S, WU Z, LI M, et al. DiffuseStyleGesture: Stylized Audio-Driven Co-Speech Gesture Generation with Diffusion Models[C/OL]//Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. 2023: 5860-5868. <https://doi.org/10.24963/ijcai.2023/650>. DOI: 10.24963/ijcai.2023/650.
- [17] HOGUE S, ZHANG C, DARUGER H, et al. DiffTED: One-shot Audio-driven TED Talk Video Generation with Diffusion-based Co-speech Gestures[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. 2024: 1922-1931.
- [18] DEICHLER A, MEHTA S, ALEXANDERSON S, et al. Diffusion-Based Co-Speech Gesture Generation Using Joint Text and Audio Representation[C/OL]//ICMI '23: Proceedings of the 25th International Conference on Multimodal Interaction. 2023: 755-762. <https://doi.org/10.1145/3577190.3616117>. DOI: 10.1145/3577190.3616117.
- [19] LIU H, ZHU Z, IWAMOTO N, et al. BEAT: A Large-Scale Semantic and Emotional Multi-Modal Dataset for Conversational Gestures Synthesis[J]. arXiv preprint arXiv:2203.05297, 2022.

-
- [20] ARKit in iOS - Tracking and Visualizing Faces. 2024. https://developer.apple.com/documentation/arkit/arkit_in_ios/content_anchors/tracking_and_visualizing_faces.
 - [21] EKMAN P, FRIESEN W V. Facial Action Coding System: A Technique for the Measurement of Facial Movement[M]. Palo Alto, California: Consulting Psychologists Press, 1978.
 - [22] OZEL M. ARKit to FACS Cheat Sheet[EB/OL]. 2022. <https://melindaozel.com/arkit-to-facs-cheat-sheet/>.
 - [23] ZHOU Y, BARNES C, LU J, et al. On the Continuity of Rotation Representations in Neural Networks[J]. arXiv preprint arXiv:1812.07035v4, 2020.
 - [24] Unity Editor [Computer Software][A/OL]. Unity Technologies. <https://unity.com/>.
 - [25] Mechanim Animation System [Computer Software Module][A/OL]. Unity Technologies. <https://docs.unity3d.com/Manual/AnimationOverview.html>.
 - [26] BOTTOU L. Large-Scale Machine Learning with Stochastic Gradient Descent[C/OL] //International Conference on Computational Statistics. 2010. <https://api.semanticscholar.org/CorpusID:115963355>.
 - [27] PICO 4 All-in-One VR Headset [Hardware Device][A/OL]. ByteDance Inc. <https://www.pico-interactive.com/>.
 - [28] SCHWIND V, RESCH S, SEHRT J. The HCI User Studies Toolkit: Supporting Study Designing and Planning for Undergraduates and Novice Researchers in Human-Computer Interaction[C/OL]//Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23). ACM, 2023: 7. DOI: 10.1145/3544549.3585890.
 - [29] YOON Y, CHA B, LEE J H, et al. Speech gesture generation from the TRIMODAL context of text, audio, and speaker identity[J]. arXiv:2009.02119, 2020.
 - [30] LI R, YANG S, ROSS D A, et al. AI Choreographer: Music Conditioned 3D Dance Generation With AIST++[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021: 13401-13412.

附录 A 手势生成对比视频

为直观展示不同模型在语音驱动手势生成任务中的表现，本文提供了基于 BEAT 数据集的可视化对比结果。具体而言，在说话人 2、4、6、8 的测试语音上，分别对本文模型 FaceCapGes、以及 CaMN^[19] 与 DiffSHEG^[15] 生成的手势序列进行了对比展示。

对应的生成结果视频可通过以下链接访问：

- **Gesture Generation Comparison Videos:**

[https://drive.google.com/drive/folders/1TmPqF-e6eHwSsMCQ2JNyPgGdMzKllDbW?
usp=sharing](https://drive.google.com/drive/folders/1TmPqF-e6eHwSsMCQ2JNyPgGdMzKllDbW?usp=sharing)

附录 B 代码与实现资源

本文模型 FaceCapGes 的完整训练与推理代码已开源，以便于复现本文中的实验结果与评估指标。代码仓库地址如下：

- **FaceCapGes GitHub Repository:**

<https://github.com/IORGestureTeam/FaceCapGes>

学术论文和科研成果目录

学术论文

- [1] First Author. FaceCapGes: Real-Time Frame-by-Frame Gesture Generation from Audio, Facial Capture, and Head Pose[C]. Computer Graphics International (CGI 2025), 2025.