

基于因果历史上下文的轻量级睡眠分期方法

邓博恺*

河北地质大学信息工程学院, 河北 石家庄

收稿日期: 2026年8月1日; 录用日期: 2026年8月31日; 发布日期: 2026年9月8日

摘要

针对轻量级单片段睡眠分期难以充分利用跨片段历史信息, 以及部分序列模型在连续预测过程中依赖未来上下文的问题, 提出一种基于因果历史上下文的轻量级睡眠分期方法MSA-TC-Lite。该方法冻结MSA-CNN small的局部特征编码路径, 将当前片段与此前9个片段构成单向历史窗口, 并利用轻量时序模块建模历史依赖; 连续推理时通过缓存历史特征减少重复编码。在Sleep-EDF-20上采用受试者级十折交叉验证, MSA-TC-Lite在41,957个具有完整历史上下文的测试片段上取得83.89%的Accuracy和78.15%的Macro-F1, 较MSA-CNN small分别提高2.18和4.77个百分点, 其中N1和REM的F1分别提高14.88和6.50个百分点。No-History对照和历史上下文扰动结果进一步表明, 性能改善与真实历史内容及其时间排列的利用具有较一致的关联。缓存式连续推理将稳态计算量由12.164 M MACs降低至1.319 M MACs。结果表明, 该方法能够在不使用未来信息的条件下改善轻量级睡眠分期性能, 并减少连续推理过程中的重复计算。

关键词

睡眠分期, 单通道脑电, 因果历史上下文, 轻量级模型, 连续推理

A Lightweight Sleep Staging Method Based on Causal Historical Context

Bokai Deng*

School of Information Engineering, Hebei GEO University, Shijiazhuang Hebei

Received: August 1, 2026; accepted: August 31, 2026; published: September 8, 2026

Abstract

To address the difficulty of lightweight single-epoch sleep staging in fully exploiting cross-epoch

*通讯作者。

historical information and the reliance of some sequence-based models on future context during continuous prediction, a lightweight sleep staging method based on causal historical context, termed MSA-TC-Lite, is proposed. The method freezes the local feature encoding pathway of MSA-CNN small and constructs a unidirectional historical window consisting of the current epoch and the preceding nine epochs. A lightweight temporal module is then used to model dependencies across historical representations. During continuous inference, previously computed historical features are cached to avoid repeated encoding. Subject-wise ten-fold cross-validation was conducted on Sleep-EDF-20. On 41,957 test epochs with complete historical context, MSA-TC-Lite achieved an Accuracy of 83.89% and a Macro-F1 of 78.15%, outperforming MSA-CNN small by 2.18 and 4.77 percentage points, respectively. The F1 scores of N1 and REM increased by 14.88 and 6.50 percentage points. The No-History control and historical-context perturbation analyses further showed that the performance improvement was consistently associated with the use of real historical content and its temporal arrangement. Cached continuous inference reduced the steady-state computational cost from 12.164 M MACs to 1.319 M MACs. These results indicate that MSA-TC-Lite improves lightweight sleep staging without using future information while reducing redundant computation during continuous inference.

Keywords

Sleep Staging, Single-Channel EEG, Causal Historical Context, Lightweight Model, Continuous Inference

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

睡眠分期是睡眠结构分析的重要基础，其目标是依据多导睡眠监测(Polysomnography, PSG)信号将连续睡眠过程划分为清醒期(Wake)、N1、N2、N3 和快速眼动睡眠期(REM)等阶段。按照成人睡眠判读规则[1]，人工睡眠分期通常以 30s 片段作为基本单位，由专业人员综合脑电(Electroencephalogram, EEG)、眼电和肌电等信号完成阶段判断。然而，人工判读过程耗时较长，并且对专业经验具有较高依赖。随着公开睡眠数据库的发展以及深度学习技术的应用[2]，基于神经网络的自动睡眠分期方法[3]逐渐成为提高睡眠数据分析效率的重要研究方向。

EEG 信号能够直接反映睡眠过程中的脑电活动变化，因此基于单通道 EEG 的自动睡眠分期受到广泛关注。早期深度学习方法主要利用卷积神经网络自动学习局部睡眠特征，减少人工特征设计过程。随后，研究者进一步通过轻量网络结构、多尺度卷积和注意力机制提升模型的特征表示能力。例如，TinySleepNet [4]、DeepSleepNet-Lite [5]等方法从模型规模和计算效率角度探索轻量化睡眠分期，SleepSatelightFTC [6]和 EfficientSleepNet [7]进一步延续了面向单通道 EEG 的轻量化建模方向，而 MSA-CNN[8]则利用多尺度卷积与注意力机制增强不同尺度特征表达能力。上述方法能够有效利用单个 30s 片段中的局部信息，但由于睡眠阶段具有明显的时间连续性，仅依赖当前片段可能难以充分利用相邻阶段之间的演化关系。

为弥补单片段模型在跨片段信息利用方面的不足，已有研究逐渐将睡眠分期扩展至连续序列建模。DeepSleepNet [9]利用卷积网络与双向循环结构联合学习局部特征和阶段间时序关系，SeqSleepNet [10]进一步采用序列到序列框架对连续睡眠片段进行联合建模，L-SeqSleepNet [11]则将建模范围扩展至更长的

睡眠序列。此外, Chambon [12] 等通过引入目标片段邻近的时间上下文改善阶段判别。随着注意力机制的发展, AttnSleep [13] 将注意力结构应用于单通道 EEG 睡眠分期; Transformer [14] 提出的自注意机制为长距离序列依赖建模提供了新的结构基础, SleepTransformer [15] 进一步将其用于自动睡眠分期。近年来, TransSleep [16]、Parallel Frequency-Cut Attention [17] 和 FlexSleepTransformer [18] 又分别从阶段转移、时频注意和灵活输入等角度扩展了上下文建模方式。这些研究表明, 历史片段中包含的时序信息能够为睡眠阶段识别提供有效补充。然而, 现有序列模型仍存在两个值得进一步研究的问题: 一方面, 部分方法同时利用目标片段前后的上下文信息, 更适用于完整睡眠记录的离线分析, 而在按照时间顺序进行连续预测时, 未来片段在当前时刻并不可获得; 另一方面, 现有方法通常同时优化局部特征提取网络和时序建模结构, 使模型容量提升与历史信息利用相互耦合, 难以进一步区分性能变化来自新增时序结构还是历史上下文信息本身。

针对上述问题, 本文关注轻量睡眠分期模型中的历史上下文利用问题, 提出一种基于因果历史上下文的轻量级睡眠分期方法 MSA-TC-Lite。该方法以 MSA-CNN small 为基础, 将局部特征提取与历史关系建模进行解耦: 首先复用轻量局部编码路径提取单片段脑电表示, 并冻结该部分参数; 随后按照时间顺序构建由当前片段及此前 9 个历史片段组成的上下文窗口, 通过轻量时序模块学习跨片段关系。与依赖未来信息的双向上下文方法不同, 本文方法仅使用当前及历史片段进行预测, 从而满足连续睡眠记录中的时间方向约束。

此外, 为进一步分析历史信息对模型性能的实际贡献, 本文设计与 MSA-TC-Lite 结构和参数规模一致的 No-History 对照模型, 通过移除真实历史信息考察新增时序结构本身的作用; 同时结合历史内容替换和历史顺序扰动实验, 分析模型对历史信息及其时间排列的敏感性。在连续推理场景下, 本文进一步设计历史特征缓存机制, 仅对新到达片段进行特征编码, 并复用已有历史表示, 以降低滑动窗口预测过程中的重复计算。

本文主要工作如下:

- (1) 提出一种基于因果历史上下文的轻量级睡眠分期方法 MSA-TC-Lite, 在冻结局部特征提取路径的基础上引入历史片段建模能力, 在避免未来信息依赖的同时增强轻量模型对跨片段历史信息的利用能力。
- (2) 设计 No-History 训练级对照以及历史上下文扰动实验, 在控制模型结构差异的条件下分析真实历史信息对睡眠阶段预测的贡献。
- (3) 针对连续睡眠记录推理过程中的重复计算问题, 引入历史特征缓存策略, 并从模型复杂度和推理效率角度分析方法的计算特性。

2. MSA-TC-Lite 方法

2.1. 总体框架

为在轻量级睡眠分期模型中引入连续片段之间的历史依赖关系, 本文提出 MSA-TC-Lite, 其总体结构如图 1 所示。模型以连续到达的单通道 EEG 片段为输入, 将单片段局部特征提取与跨片段时序建模进行解耦, 整体由冻结局部编码器、因果历史窗口和轻量时序模块三个主要部分组成。

对于当前 EEG 片段, 首先利用冻结的 MSA-CNN small 局部编码路径提取 32 维片段级特征。随着睡眠记录按时间顺序输入, 当前特征与此前 9 个历史片段特征共同组成长度为 10 的历史窗口, 其中仅包含当前时刻及其之前已经获得的信息, 不使用任何未来片段。随后, 历史特征序列输入轻量时序模块进行跨片段关系建模, 并输出当前片段对应的 Wake、N1、N2、N3 或 REM 阶段。该结构在保留轻量局部表示能力的基础上, 使当前睡眠阶段的判别能够结合已经发生的睡眠状态演化信息。

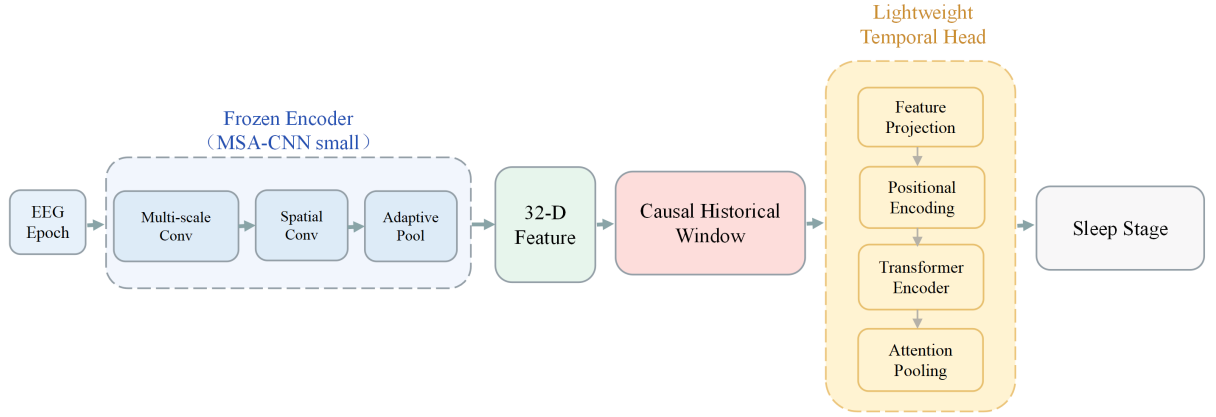


Figure 1. Overall architecture of MSA-TC-Lite

图 1. MSA-TC-Lite 模型总体框架

2.2. 冻结局部特征编码

如图 1 左侧所示, MSA-TC-Lite 首先对每个 30s EEG 片段进行独立局部编码。本文复用 MSA-CNN small 中的轻量特征提取路径, 包括多尺度卷积、空间特征提取和自适应平均池化, 并在历史时序模块训练过程中固定该部分参数。这样可以保持已学习的单片段特征表示, 同时避免新增历史建模过程改变原有局部编码能力。

设第 t 个 EEG 片段为 x_t , 冻结局部编码器记为 $E(\cdot)$, 则其片段级表示为:

$$z_t = E(x_t), z_t \in \mathbb{R}^{32}, \quad (1)$$

其中, z_t 为当前 EEG 片段对应的 32 维特征表示。对于连续睡眠记录中的不同片段, 均使用同一冻结编码器完成特征映射, 从而得到按照时间顺序排列的片段级特征序列, 该冻结局部编码路径包含 5680 个参数, 输出 32 维片段级表示。

2.3. 因果历史上下文建模

单个 30s EEG 片段主要描述当前局部脑电模式, 而连续睡眠过程中相邻阶段之间还具有明显的时间依赖关系。因此, 在获得片段级表示后, MSA-TC-Lite 进一步按照时间顺序组织历史特征。本文主实验将历史窗口长度设置为 10, 以在引入短历史信息的同时控制时序模块的计算规模。为进一步验证该设置的合理性, 实验部分另设置历史窗口长度为 5、10、15 和 20 的敏感性分析, 并比较不同历史范围下的分期性能与计算成本。

对于目标时刻 t , 将当前特征 z_t 与此前 9 个历史特征共同构成长度为 10 的历史窗口:

$$H_t = [z_{t-9}, z_{t-8}, \dots, z_{t-1}, z_t], H_t \in \mathbb{R}^{10 \times 32} \quad (2)$$

其中, z_t 位于窗口末端, 表示当前待预测片段。窗口中不存在 t 时刻之后的特征, 因此模型在预测当前阶段时仅使用已经获得的历史信息。本文将这种满足单向时间信息约束的输入方式称为因果历史上下文; 这里的“因果”表示信息按照历史到当前的时间方向使用, 而不涉及统计意义上的因果推断。

历史特征首先经过特征投影、Layer Normalization 和 PReLU 激活, 并加入固定正弦位置编码。随后, 采用单层 Transformer 进行跨片段关系建模, 其中注意力头数设置为 4, 前馈网络维度为 64, dropout 为 0.1。Transformer 输出经注意力池化获得窗口级表示, 并进一步完成当前睡眠阶段预测:

$$p_t = \text{softmax}(G(H_t)), p_t \in \mathbb{R}^5 \quad (3)$$

其中, $G(\cdot)$ 表示轻量时序建模与分类过程, p_t 为 Wake、N1、N2、N3 和 REM 五个睡眠阶段的预测概率。

2.4. 缓存式连续推理

图 1 展示的是 MSA-TC-Lite 在连续睡眠片段上的总体处理过程。由于相邻历史窗口具有较大的特征重叠, 当模型由 t 时刻推进至 $t+1$ 时, 原窗口中已有 9 个片段特征仍可直接复用。因此, 在连续推理过程中无需重新编码完整的 10 个 EEG 片段。

当新片段 x_{t+1} 到达后, 仅通过冻结局部编码器计算其特征 z_{t+1} , 同时移除原窗口中最早的 z_{t-9} , 保留其余 9 个历史特征, 并将窗口更新为:

$$H_{t+1} = [z_{t-8}, z_{t-7}, \dots, z_t, z_{t+1}] \quad (4)$$

更新后的历史窗口继续输入相同的轻量时序模块完成 $t+1$ 时刻的阶段预测。缓存机制不改变模型的输入定义和分类过程, 而是通过复用已计算的历史特征减少滑动窗口中的重复局部编码, 从而适应按照时间顺序进行的连续睡眠分期。

3. 实验设计

3.1. 数据集与预处理

本文采用 PhysioNet 发布的 Sleep-EDF Database Expanded v1.0.0 [19] 中的 Sleep Cassette 数据[20]进行实验, 选取常用于 Sleep-EDF-20 设置的 20 名受试者, 共包含 39 条整夜睡眠记录。实验使用单通道 Fpz-Cz EEG 信号, 采样率为 100 Hz, 并按照 30s 划分为互不重叠的睡眠片段, 每个片段包含 3000 个采样点。

原始睡眠阶段依据 Rechtschaffen and Kales (R&K) 标准[21]进行标注。本文将标签统一映射为 Wake、N1、N2、N3 和 REM 五个阶段, 其中 Stage 3 和 Stage 4 合并为 N3, Unknown 和 Movement 片段不参与训练与评价。为减少整夜记录首尾长时间清醒片段造成的类别偏斜, 以每条记录中首个和末个非 Wake 片段为边界, 在其前后分别保留不超过 30 min 的 Wake 片段。EEG 信号采用 4 阶 Butterworth 低通滤波器进行处理, 截止频率为 40 Hz, 并采用零相位滤波方式; 除此之外不额外进行独立的输入标准化处理。

完成上述处理后, 共获得 42,308 个有效 EEG 片段。由于 MSA-TC-Lite 需要由当前片段及其前 9 个历史片段构成长为 10 的历史窗口, 因此每条记录开始的 9 个片段用于历史特征积累, 39 条记录共包含 351 个此类片段; 其余 41,957 个片段均具有完整历史上下文, 并作为主要模型性能比较中的配对评价样本。历史窗口仅在同一条睡眠记录内部构造, 不跨记录连接。

3.2. 训练与对照设置

实验采用固定的受试者级十折交叉验证, 以保证同一受试者的全部睡眠记录始终位于同一训练集或测试集中。对于每一折, 首先仅使用该折训练受试者训练 MSA-CNN small, 随后提取对应的轻量局部编码路径并固定其参数, 在相同训练划分上训练 MSA-TC-Lite 的时序模块。不同交叉验证折之间不共享模型参数, 也不使用全部受试者训练得到的统一编码器。

两种模型均采用 Adam 优化器和多分类交叉熵损失。MSA-CNN small 的学习率设为 0.001, 权重衰减为 1×10^{-4} ; MSA-TC-Lite 时序模块的学习率设为 1×10^{-4} , 权重衰减为 1×10^{-6} 。batch size 均设为 64, 训练过程中未采用学习率调度和早停策略, 所有模型均使用第 100 个 epoch 的参数进行测试, 从而保持各折训练协议一致。

为区分真实历史信息与新增时序结构对分类性能的影响, 本文设置 No-History 训练级对照。该模型与 MSA-TC-Lite 采用相同的冻结局部编码器、时序模块结构及训练协议, 但在训练和测试阶段均以当前目标特征的重复输入代替真实历史窗口, 即:

$$H_t^{NH} = [z_t, z_t, \dots, z_t] \in \mathbb{R}^{10 \times 32} \quad (5)$$

通过该设置，可以在保持时序模块结构和参数规模一致的条件下，对比真实历史信息带来的附加作用。

在 No-History 对照之外，本文进一步对训练完成的 MSA-TC-Lite 开展测试时历史上下文扰动分析。Current-epoch repetition 条件将当前特征重复至窗口全部 10 个位置；Historical-order shuffling 条件保留前 9 个历史特征的内容，但改变其时间排列顺序，同时保持当前目标特征位于窗口末端。上述扰动均不重新训练模型，用于分析模型对历史内容及其时间顺序的输入敏感性。

为评价连续推理过程中的计算特性，进一步统计模型前向参与参数量、乘加运算量(MACs)以及 batch size 为 1 时的 CPU 和 GPU 推理延迟。其中 CPU 测试采用单线程执行，GPU 测试在 RTX 5090 上完成；计时过程中输入数据预先加载，仅统计模型前向过程，不包含磁盘读取和完整信号预处理时间。

对于窗口长度敏感性分析，L = 5、15、20 均在各交叉验证折内重新训练对应的轻量时序模块，L = 10 沿用主实验设置；不同窗口长度之间保持冻结局部编码器、模型结构及训练超参数一致。

针对缓存式推理计算量降低但 GPU 延迟未同步改善的现象，进一步采用 PyTorch Profiler 对 RTX 5090、batch size 为 1 条件下的 GPU 执行过程进行性能剖析，分析局部编码、缓存更新、时序模块、CUDA kernel 及主机 - 设备数据传输等开销。

3.3. 评价指标与统计分析

本文采用 Accuracy、Macro-F1、Balanced Accuracy 和 Cohen's kappa 评价模型的总体睡眠分期性能。其中，Accuracy 反映整体分类正确率；Macro-F1 对五个睡眠阶段的 F1 值进行等权平均，用于评价类别不均衡条件下的综合识别能力；Balanced Accuracy 反映各类别召回率的平均水平；Cohen's kappa 用于评价预测结果与真实标签在校正随机一致性后的整体一致程度。此外，分别报告 Wake、N1、N2、N3 和 REM 五个睡眠阶段的类别 F1 值，以分析模型对不同阶段的识别差异。

十折测试完成后，将各折样本外预测合并计算总体指标，并保留逐折结果用于分析模型性能的一致性。进一步分别以受试者和整夜记录为重采样单位进行 10,000 次配对聚类 Bootstrap，并采用百分位法计算模型性能差异的 95%置信区间。

4. 实验结果与分析

4.1. 总体分期性能

为评价因果历史上下文建模对睡眠分期性能的实际影响，本文在具有完整历史信息的测试片段上，对单片段基线 MSA-CNN small 与所提出的 MSA-TC-Lite 进行统一比较。考虑到 MSA-TC-Lite 需要当前片段及其前 9 个历史片段共同构成长度为 10 的输入窗口，因此仅选取能够形成完整历史上下文的 41,957 个测试片段作为配对评价样本。两种模型采用相同的受试者级十折数据划分，并在对应测试片段上计算 Accuracy、Macro-F1、Balanced Accuracy 和 Cohen's kappa，以保证性能差异主要反映历史上下文建模带来的影响。总体分期结果如表 1 所示。

Table 1. Overall performance comparison between MSA-CNN small and MSA-TC-Lite

表 1. MSA-CNN small 与 MSA-TC-Lite 总体性能比较

模型	Accuracy/%	Macro-F1/%	Balanced Accuracy/%	Kappa
MSA-CNN small	81.70	73.38	73.40	0.7477
MSA-TC-Lite	83.89	78.15	78.22	0.7785

由表 1 可见, 与 MSA-CNN small 相比, MSA-TC-Lite 的 Accuracy 由 81.70% 提高至 83.89%, Macro-F1 和 Balanced Accuracy 分别由 73.38% 和 73.40% 提高至 78.15% 和 78.22%, Cohen's kappa 由 0.7477 提高至 0.7785。其中 Macro-F1 和 Balanced Accuracy 分别提升 4.77 和 4.82 个百分点, 增幅明显高于总体 Accuracy, 说明引入历史上下文后, 模型在保持整体分类性能提升的同时, 对不同睡眠阶段的识别更加均衡。

MSA-TC-Lite 的 Macro-F1 和 Balanced Accuracy 在全部 10 个测试折中均高于基线, Accuracy 和 Cohen's kappa 则在 9 个测试折中取得提升; 受试者级配对聚类 Bootstrap 所得四项指标差异的 95% 置信区间均位于 0 以上, Accuracy 和 Macro-F1 增益的 95% 置信区间分别为 [1.12, 3.17] 和 [3.46, 6.18] 个百分点; 记录层级分析同样得到完全位于 0 以上的区间。

4.2. 不同睡眠阶段识别结果

总体评价指标能够反映模型在全部测试样本上的综合性能, 但不同睡眠阶段的信号特征、样本数量及识别难度存在差异, 因此仅依赖总体指标难以判断性能改善主要来源于哪些类别。为进一步分析历史上下文对各阶段判别能力的影响, 本文分别统计 MSA-CNN small 与 MSA-TC-Lite 在 Wake、N1、N2、N3 和 REM 五个睡眠阶段上的类别 F1 值, 并比较不同阶段的性能变化幅度。各睡眠阶段的具体结果如表 2 所示。

Table 2. F1-score comparison across different sleep stages

表 2. 不同睡眠阶段 F1 值比较

睡眠阶段	MSA-CNN small/%	MSA-TC-Lite/%
Wake	88.73	89.36
N1	29.07	43.95
N2	86.94	87.89
N3	87.02	87.88
REM	75.15	81.65

由表 2 可见, MSA-TC-Lite 在五个睡眠阶段上的 F1 值均高于单片段基线, 但不同类别的改善幅度存在明显差异。其中, N1 的 F1 由 29.07% 提高至 43.95%, 提升 14.88 个百分点, 是五个阶段中增幅最大的类别; REM 的 F1 由 75.15% 提高至 81.65%, 提升 6.50 个百分点。Wake、N2 和 N3 的提升幅度相对较小。

这一结果说明, 历史上下文带来的性能改善主要体现在 N1 和 REM 等类别, 而不是对所有阶段产生相同程度的影响。尤其对于基线模型识别能力较弱的 N1, 结合历史片段特征中所包含的时序演化信息后获得了更明显的改善, 这也是 Macro-F1 提升幅度高于总体 Accuracy 的重要原因之一。

为了进一步观察不同模型在五个睡眠阶段之间的具体混淆情况, 图 2 给出了 41,957 个严格配对测试片段上的行归一化混淆矩阵。横轴表示模型预测阶段, 纵轴表示参考标注阶段, 每一行的数值表示对应真实类别被预测为各阶段的比例。

从图 2 可以进一步看到, MSA-TC-Lite 对 N1 的正确识别比例由 MSA-CNN small 的 25.1% 提高至 47.4%, 同时 N1 被误判为 REM 的比例由 33.6% 下降至 11.8%, 表明历史上下文建模明显改变了基线模型在 N1 上的主要混淆结构。N2 和 REM 的正确识别比例也分别由 87.1% 和 77.8% 提高至 88.4% 和 78.9%。对于 Wake 和 N3, 两种模型均保持了较高的识别水平, 其中 N3 的对角线召回比例略有变化, 但其类别 F1 总体仍保持稳定。结合表 2 可以看出, 类别 F1 和混淆矩阵所得结果具有较好的一致性, 进一步说明模型总体性能提高主要与 N1 及 REM 等阶段识别能力的改善有关。

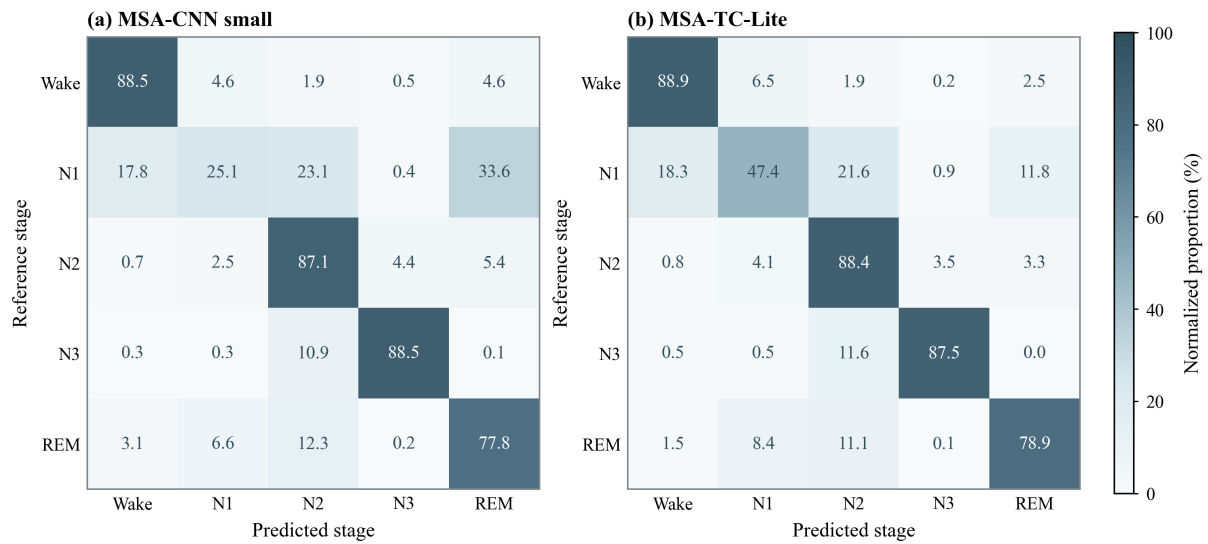


Figure 2. Normalized confusion matrices of MSA-CNN small and MSA-TC-Lite

图 2. MSA-CNN small 与 MSA-TC-Lite 的归一化混淆矩阵

4.3. 历史上下文作用分析

前述结果能够说明 MSA-TC-Lite 在总体性能和部分睡眠阶段上优于单片段基线，但模型在引入真实历史信息的同时，也增加了新的时序建模结构。因此，仅比较 MSA-CNN small 与 MSA-TC-Lite 尚不足以区分性能变化是来自时序模块本身，还是来自历史特征所包含的信息。针对这一问题，本文进一步从训练级对照和测试时输入扰动两个角度展开分析，以考察真实历史内容及其时间排列在模型预测中的作用。

4.3.1. No-History 对照实验

为控制新增时序结构和参数规模可能带来的影响，本文构建 No-History 训练级对照模型。该模型与 MSA-TC-Lite 采用相同的冻结局部编码器、时序模块结构、可训练参数规模和训练协议，唯一差异在于训练与测试过程中均不输入真实历史特征，而是将当前目标片段的特征重复至长度为 10 的窗口。通过在模型结构基本一致的条件下改变历史信息来源，可以进一步比较真实历史上下文相对于新增时序结构本身所提供的附加作用。MSA-CNN small、No-History 和 MSA-TC-Lite 的总体性能比较如表 3 所示。

Table 3. Performance comparison in the No-History control experiment

表 3. No-History 对照实验性能比较

模型	真实历史	Accuracy/%	Macro-F1/%	Balanced Accuracy/%	Kappa
MSA-CNN small	—	81.70	73.38	73.40	0.7477
No-History	×	81.11	72.42	72.29	0.7389
MSA-TC-Lite	√	83.89	78.15	78.22	0.7785

由表 3 可见，No-History 在 Accuracy、Macro-F1、Balanced Accuracy 和 Cohen’s kappa 四项指标上均未超过 MSA-CNN small，说明在没有真实历史输入的情况下，仅增加相同的时序结构并未获得与 MSA-TC-Lite 相同的性能改善。与 No-History 相比，MSA-TC-Lite 的 Accuracy、Macro-F1 和 Balanced Accuracy 分别提高 2.78、5.73 和 5.93 个百分点，Cohen’s kappa 提高 0.0396，并且四项指标在全部 10 个测试折中均呈正向变化。上述结果说明，在保持时序结构和训练协议一致的条件下，真实历史输入能够为当前睡眠阶段判别提供额外信息。

4.3.2. 历史上下文扰动分析

No-History 实验主要从训练层面控制了新增时序结构的影响, 但仍有必要进一步考察已经训练完成的 MSA-TC-Lite 是否对历史窗口中的具体内容及其时间排列具有敏感性。为此, 本文保持模型参数完全不变, 仅在测试阶段改变历史窗口输入形式。其中, Current-epoch repetition 条件以当前目标特征替代窗口中的历史位置, 用于削弱真实历史内容; Historical-order shuffling 则保留原有历史特征集合, 但改变前 9 个历史特征的时间顺序, 同时保持当前目标特征位于窗口末端。通过与原始历史上下文条件进行比较, 可以从输入扰动角度进一步分析模型对历史内容和时间顺序的利用情况, 结果如表 4 所示。

Table 4. Macro-F1 changes under historical-context perturbations

表 4. 历史上下文扰动条件下的 Macro-F1 变化

历史上下文条件	Macro-F1 差值/百分点	正向折数
Original historical context—Current-epoch repetition	+6.89 ± 1.30	10/10
Original historical context—Historical-order shuffling	+1.71 ± 1.09	9/10

当历史窗口中的真实历史内容被当前目标特征重复替代后, 原始历史上下文条件的 Macro-F1 平均高 6.89 个百分点, 并在全部 10 个测试折中保持正向差异, 说明训练完成的模型对历史窗口中实际包含的信息具有明显敏感性。进一步改变历史特征的时间排列后, 原始历史上下文条件的 Macro-F1 平均高 1.71 个百分点, 并在 9 个测试折中取得更高结果。由于该条件仍保留了相同的历史特征集合, 仅改变其排列方式, 因此这一结果进一步表明历史信息的时间组织方式也会影响模型预测。结合 No-History 训练级对照和测试时上下文扰动结果可以看出, MSA-TC-Lite 相对于单片段基线获得的性能改善并非仅表现为增加时序模块后的模型容量变化, 而与真实历史内容及其时间排列的利用具有较一致的关联。

4.4. 历史窗口长度敏感性分析

为进一步分析历史上下文范围对睡眠分期性能及计算成本的影响, 本文在主实验采用 $L = 10$ 的基础上, 设置 $L = 5, 10, 15, 20$ 四种历史窗口长度开展敏感性分析。不同窗口长度分别对应使用当前片段及其之前不同数量的历史片段, 从而考察模型在获得更多历史信息后, 其分类性能是否能够持续改善, 以及由此带来的额外计算开销。为保证比较结果主要反映历史窗口长度变化本身的影响, 四种设置均采用相同的受试者级十折交叉验证划分, 并保持冻结局部编码器、轻量时序模块结构、优化器、训练轮数及其他训练超参数一致, 仅改变输入历史序列的长度。由于不同窗口长度对完整历史上下文的要求不同, 其自然形成的有效测试片段数量并不完全一致, 若直接在各自有效样本上进行比较, 可能引入评价样本差异带来的影响。因此, 本文进一步构建四种窗口长度均具有完整历史信息共同测试子集, 并统一在 41,567 个目标片段上计算各项性能指标, 同时统计完整窗口重算和缓存式连续推理条件下的计算量, 以综合分析不同历史窗口长度在分期性能与计算成本之间的权衡关系, 具体结果如表 5 所示。

Table 5. Performance and computational cost under different historical window lengths

表 5. 不同历史窗口长度下的分期性能与计算成本

L	Accuracy/%	Macro-F1/%	Balanced Accuracy/%	Kappa	Full MACs/M	Cached MACs/M
5	83.43	77.54	77.68	0.7718	6.081	1.261
10	83.75	78.05	78.12	0.7761	12.164	1.319
15	83.87	78.54	78.89	0.7784	18.251	1.382
20	83.97	78.43	78.45	0.7789	24.342	1.447

随着窗口长度由 5 增加至 10, Macro-F1 由 77.54%提高至 78.05%; 继续增加至 15 和 20 后, Macro-F1 分别为 78.54%和 78.43%, 较 L=10 仅提高 0.49 和 0.38 个百分点。配对聚类 Bootstrap 结果显示, L=15 和 L=20 相对于 L=10 的 Macro-F1 差异在受试者级和记录级 95%置信区间均跨越 0, 表明较长窗口的附加增益尚未表现出稳定优势。与 L=10 相比, L=15 和 L=20 的完整窗口计算量分别增加约 50%和 100%, 而缓存式推理计算量仅增加约 4.7%和 9.7%。综合性能和计算成本, L=10 并非绝对性能最优窗口, 但已较 L=5 获得更好的分期性能, 继续增加历史范围仅带来有限且不稳定的 Macro-F1 改善, 因此本文保留 L=10 作为兼顾性能与轻量计算需求的默认设置。

4.5. 模型复杂度与连续推理效率

历史上下文的引入扩展了模型在当前睡眠阶段判别过程中可利用的时间信息, 但与此同时, 也会带来额外的计算开销。对于长度为 10 的历史窗口, 若在连续预测过程中每到达一个新的 EEG 片段, 都重新对当前片段及其前 9 个历史片段执行完整的特征编码, 则相邻窗口之间高度重叠的历史片段将被重复计算。例如, 相邻两个时间步仅新增一个片段, 其余 9 个历史片段均已在上一时刻完成特征提取, 若仍采用完整窗口重算方式, 将造成较明显的冗余计算。因此, 除评价模型的分类性能外, 还需要进一步分析历史上下文建模在连续推理场景下的计算代价, 以及历史特征复用能否有效降低这种重复开销。为此, 本文分别比较 MSA-CNN small 单片段推理、MSA-TC-Lite 完整窗口重算和 MSA-TC-Lite 缓存式连续推理三种方式。其中, MSA-CNN small 用于反映原始轻量单片段模型的计算基准; 完整窗口重算用于衡量在不进行特征复用时, 引入长度为 10 的历史上下文所带来的计算负担; 缓存式连续推理则在保持相同历史窗口和预测过程的基础上, 复用此前已经获得的历史特征, 仅对新到达片段执行局部编码, 从而减少相邻窗口间的重复计算。本文从前向参与参数量、可训练参数量、乘加运算量(MACs)以及 batch size 为 1 条件下的 CPU 和 GPU 实测推理延迟等方面进行综合评价, 以分析不同推理方式在理论计算复杂度和实际运行效率上的差异, 具体结果如表 6 所示。

Table 6. Comparison of model complexity and inference efficiency
表 6. 模型复杂度与推理效率比较

模型	前向路径参数量	可训练参数	MACs (M)	CPU/ms	GPU/ms
MSA-CNN small	8.52 K	8.52 K	1.568	1.009	0.812
MSA-TC-Lite 全窗口重算	16.60 K	10.92 K	12.164	7.125	0.903
MSA-TC-Lite 缓存推理	16.60 K	10.92 K	1.319	0.993	0.921

MSA-TC-Lite 前向计算共涉及 16,599 个参数, 其中 10,919 个为轻量时序模块的可训练参数。完整窗口推理需在每个时间步重复编码 10 个连续片段, 计算量为 12.164 M MACs; 采用历史特征缓存后, 仅对新到达片段进行编码, 其余 9 个历史特征直接复用, 使稳态计算量降至 1.319 M MACs, 较完整窗口重算减少约 89.2%, 并低于 MSA-CNN small 单片段推理的 1.568 M MACs。在 CPU 单线程条件下, 缓存式推理延迟为 0.993 ms, 与 MSA-CNN small 的 1.009 ms 接近, 明显低于完整窗口重算的 7.125 ms。GPU 条件下, 三种方式的延迟分别为 0.812、0.903 和 0.921 ms, 缓存式推理虽显著降低 MACs, 但未获得相应的延迟优势, 表明理论运算量的减少并未直接转化为实际推理延迟的下降。

为进一步分析这一现象, 在 RTX 5090、batch size 为 1 的条件下进行了独立 GPU 性能剖析。每种推理方式采用 3 个轮换顺序区块, 每区块预热 100 次并使用 CUDA Events 测量 1000 次。完整窗口和缓存式推理的平均延迟分别为 0.8816 ms 和 0.9017 ms, 与前述结果趋势一致。Profiler 结果显示, 缓存式推理将 MACs 降低 89.2%, 但 GPU kernel 活跃时间仅由 0.08165 ms/步下降至 0.07810 ms/步, 降幅约 4.35%;

测试中未检测到 H2D 或 D2H 数据传输, 缓存更新耗时仅为 0.01875 ms, 说明数据传输和缓存维护并非主要瓶颈。进一步观察发现, 完整窗口一次编码 10 个片段与缓存式仅编码 1 个片段的 GPU 耗时分别为 0.38644 ms 和 0.38631 ms, 两者基本相当, 同时两条路径每步分别包含 50 和 51 个小于 10 μ s 的 CUDA kernel。上述结果表明, 在当前高并行 GPU 及轻量 batch size = 1 推理条件下, 理论计算量下降并未同比降低 GPU 实际执行时间, 上述结果与细粒度算子及固定调度开销在轻量 batch size = 1 推理中占据较高相对比重的解释一致。总体而言, 缓存机制的主要优势在于减少重复编码和降低稳态计算量, 其实际延迟收益仍受具体硬件执行环境影响。

4.6. 讨论

综合上述实验结果, MSA-TC-Lite 的性能改善并非均匀分布于各睡眠阶段, 而是主要集中于 N1 和 REM。其中, N1 是 MSA-CNN small 识别能力最低的类别, 引入历史上下文后其 F1 由 29.07% 提高至 43.95%, 提升 14.88 个百分点; REM 的 F1 也提高 6.50 个百分点。与此同时, Macro-F1 和 Balanced Accuracy 的提升幅度均高于总体 Accuracy, 说明历史上下文带来的收益更多体现在类别均衡识别能力的改善上。从本文实验结果看, 单片段局部特征对于部分边界较模糊或局部判别信息不足的片段可能仍存在表达局限, 而将目标片段置于已经发生的睡眠历史中进行判别, 可以为当前阶段提供额外的时序参照。相比之下, Wake、N2 和 N3 原本具有较高的识别水平, 其性能提升相对有限, 表明历史上下文对不同睡眠阶段的作用具有一定差异性, 而非简单地对所有类别产生一致增益。

No-History 对照和历史上下文扰动进一步从不同角度验证了真实历史信息的附加价值。No-History 与 MSA-TC-Lite 具有相同的冻结局部编码器、时序结构和训练协议, 但在移除真实历史输入后, 其总体性能未超过单片段基线, 说明仅增加时序模块和可训练参数本身不足以获得与 MSA-TC-Lite 相同的性能改善。另一方面, 在保持模型参数不变的情况下, 以当前特征重复替代真实历史内容后, Macro-F1 平均下降 6.89 个百分点; 打乱历史特征的时间顺序后也出现 1.71 个百分点的平均下降。前者说明模型对历史窗口中实际包含的信息具有较明显的敏感性, 后者则表明历史信息的时间组织方式同样会影响预测结果。两类实验相互补充, 支持 MSA-TC-Lite 的性能改善与真实历史内容及其时间排列的利用具有一致的正向关联, 但这些结果主要反映当前实验设置下的输入依赖关系, 并不等同于统计意义上的因果效应证明。

从计算特性看, 因果历史上下文虽然增加了时序建模过程, 但冻结局部编码路径使新增训练主要集中于轻量时序模块, 同时历史特征缓存能够避免连续滑动窗口中的重复编码。缓存式推理将稳态计算量由 12.164 M MACs 降低至 1.319 M MACs, 并在 CPU 条件下获得与单片段基线接近的推理延迟, 说明特征复用能够有效降低连续预测中的计算冗余。然而, GPU 条件下计算量降低并未转化为相应的延迟优势, 表明理论计算复杂度与实际硬件执行效率并非完全一致。窗口长度敏感性分析进一步表明, 适当增加历史范围能够改善模型性能, 但这种收益并不随窗口长度持续增加。相较 $L = 10$, $L = 15$ 和 $L = 20$ 仅取得有限的 Macro-F1 增益, 且配对 Bootstrap 区间均跨越 0, 同时完整窗口计算成本进一步增加。因此, 在当前数据集和完整窗口计算成本约束下, $L = 10$ 可作为较合理的轻量化折中设置。另一方面, GPU 性能剖析表明, 缓存式推理虽然显著降低 MACs, 但在当前 RTX 5090 和 batch size 为 1 的条件下, GPU 实际执行时间并未同比下降, 说明理论计算复杂度与实际硬件执行效率之间仍存在差异。本文目前仍主要基于 Sleep-EDF-20 单通道 Fpz-Cz 数据进行验证, 后续可进一步在其他公开数据集、EEG 通道及硬件平台上验证上述结果。

5. 结论

针对轻量级单片段睡眠分期方法难以显式利用跨片段历史信息, 以及部分序列模型在连续预测过程

中依赖未来上下文的问题，本文提出了基于因果历史上下文的轻量级睡眠分期模型 MSA-TC-Lite。该方法复用并冻结 MSA-CNN small 的轻量局部特征编码路径，将当前片段与此前 9 个历史片段构成长度为 10 的单向历史窗口，并利用轻量时序模块完成当前睡眠阶段预测。在连续推理过程中，通过缓存并复用已计算的历史特征，减少相邻窗口之间的重复局部编码，从而兼顾历史信息建模与连续推理计算开销。

在 Sleep-EDF-20 上的受试者级十折实验结果表明，MSA-TC-Lite 在 41,957 个具有完整历史上下文的配对测试片段上取得 83.89% 的 Accuracy 和 78.15% 的 Macro-F1，相较 MSA-CNN small 分别提高 2.18 和 4.77 个百分点，Balanced Accuracy 提高 4.82 个百分点。其中，N1 和 REM 的 F1 分别提高 14.88 和 6.50 个百分点。No-History 训练级对照显示，在保持时序结构和训练协议基本一致但移除真实历史信息后，模型未获得相同的性能改善；进一步的历史上下文扰动结果表明，替换历史内容或改变历史特征的时间顺序均会降低 Macro-F1。上述结果说明，MSA-TC-Lite 的性能改善与真实历史信息及其时间组织方式的利用具有较一致的关联。

在计算效率方面，缓存式连续推理通过复用 9 个历史特征，将稳态计算量由完整窗口重算的 12.164 M MACs 降低至 1.319 M MACs，并在 CPU 单线程条件下获得与单片段基线接近的推理延迟。需要指出的是，计算量降低并未在 GPU 条件下体现为相应的延迟优势，因此模型实际运行效率仍受到具体硬件和算子执行方式的影响。目前实验主要基于 Sleep-EDF-20 单通道 Fpz-Cz EEG 数据，后续可进一步在不同公开睡眠数据集、不同 EEG 通道及其他硬件平台上验证模型的泛化能力以及窗口长度相关规律的可重复性。此外，窗口长度敏感性分析表明，较长历史窗口仅带来有限且不稳定的 Macro-F1 附加收益，而完整窗口计算成本持续增加，综合性能与计算成本，L = 10 可作为当前实验设置下较合理的轻量化折中选择。进一步的 GPU 性能剖析表明，缓存机制带来的 MACs 下降在高并行 GPU 的 batch size 为 1 条件下未转化为相应的延迟改善，其主要价值仍在于减少重复编码和理论计算负担。

参考文献

- [1] Silber, M.H., Ancoli-Israel, S., Bonnet, M.H., Chokroverty, S., Grigg-Damberger, M.M., Hirshkowitz, M., *et al.* (2007) The Visual Scoring of Sleep in Adults. *Journal of Clinical Sleep Medicine*, **3**, 121-131. <https://doi.org/10.5664/jcsm.26814>
- [2] Yue, H., Chen, Z., Guo, W., Sun, L., Dai, Y., Wang, Y., *et al.* (2024) Research and Application of Deep Learning-Based Sleep Staging: Data, Modeling, Validation, and Clinical Practice. *Sleep Medicine Reviews*, **74**, Article ID: 101897. <https://doi.org/10.1016/j.smr.2024.101897>
- [3] Liu, P., Qian, W., Zhang, H., Zhu, Y., Hong, Q., Li, Q., *et al.* (2024) Automatic Sleep Stage Classification Using Deep Learning: Signals, Data Representation, and Neural Networks. *Artificial Intelligence Review*, **57**, Article No. 301. <https://doi.org/10.1007/s10462-024-10926-9>
- [4] Supratak, A. and Guo, Y. (2020) TinySleepNet: An Efficient Deep Learning Model for Sleep Stage Scoring Based on Raw Single-Channel EEG. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, 20-24 July 2020, 641-644. <https://doi.org/10.1109/embc44109.2020.9176741>
- [5] Fiorillo, L., Favaro, P. and Faraci, F.D. (2021) DeepSleepNet-Lite: A Simplified Automatic Sleep Stage Scoring Model with Uncertainty Estimates. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **29**, 2076-2085. <https://doi.org/10.1109/tnsre.2021.3117970>
- [6] Ito, A. and Tanaka, T. (2025) SleepSatelightFTC: A Lightweight and Interpretable Deep Learning Model for Single-Channel EEG-Based Sleep Stage Classification. *IEEE Access*, **13**, 46263-46272. <https://doi.org/10.1109/access.2025.3549436>
- [7] Wang, F., Zheng, Z., Hu, B., Yang, X., Tang, M. and Huang, H. (2025) EfficientSleepNet: A Novel Lightweight End-To-End Model for Automated Sleep Staging on Single-Channel EEG. *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, 6-11 April 2025, 641-644. <https://doi.org/10.1109/icassp49660.2025.10889937>
- [8] Goerttler, S., Wang, Y., Eldele, E., He, F. and Wu, M. (2026) MSA-CNN: A Lightweight Multi-Scale CNN with Attention for Sleep Stage Classification. *Biomedical Signal Processing and Control*, **120**, Article ID: 110141. <https://doi.org/10.1016/j.bspc.2026.110141>

-
- [9] Supratak, A., Dong, H., Wu, C. and Guo, Y. (2017) DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **25**, 1998-2008. <https://doi.org/10.1109/tnsre.2017.2721116>
 - [10] Phan, H., Andreotti, F., Cooray, N., Chén, O.Y. and De Vos, M. (2019) SeqSleepNet: End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **27**, 400-410. <https://doi.org/10.1109/tnsre.2019.2896659>
 - [11] Phan, H., Lorenzen, K.P., Heremans, E., Chén, O.Y., Tran, M.C., Koch, P., *et al.* (2023) L-SeqSleepNet: Whole-Cycle Long Sequence Modeling for Automatic Sleep Staging. *IEEE Journal of Biomedical and Health Informatics*, **27**, 4748-4757. <https://doi.org/10.1109/jbhi.2023.3303197>
 - [12] Chambon, S., Galtier, M.N., Arnal, P.J., Wainrib, G. and Gramfort, A. (2018) A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **26**, 758-769. <https://doi.org/10.1109/tnsre.2018.2813138>
 - [13] Eldele, E., Chen, Z., Liu, C., Wu, M., Kwoh, C., Li, X., *et al.* (2021) An Attention-Based Deep Learning Approach for Sleep Stage Classification with Single-Channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **29**, 809-818. <https://doi.org/10.1109/tnsre.2021.3076234>
 - [14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 6000-6010.
 - [15] Phan, H., Mikkelsen, K., Chén, O.Y., Koch, P., Mertins, A. and De Vos, M. (2022) SleepTransformer: Automatic Sleep Staging with Interpretability and Uncertainty Quantification. *IEEE Transactions on Biomedical Engineering*, **69**, 2456-2467. <https://doi.org/10.1109/tbme.2022.3147187>
 - [16] Phyto, J., Ko, W., Jeon, E. and Suk, H.I. (2023) TransSleep: Transitioning-Aware Attention-Based Deep Neural Network for Sleep Staging. *IEEE Transactions on Cybernetics*, **53**, 4500-4510. <https://doi.org/10.1109/tycb.2022.3198997>
 - [17] Chen, Z., Yang, Z., Zhu, L., Chen, W., Tamura, T., Ono, N., *et al.* (2023) Automated Sleep Staging via Parallel Frequency-Cut Attention. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **31**, 1974-1985. <https://doi.org/10.1109/tnsre.2023.3243589>
 - [18] Guo, Y., Nowakowski, M. and Dai, W. (2024) FlexSleepTransformer: A Transformer-Based Sleep Staging Model with Flexible Input Channel Configurations. *Scientific Reports*, **14**, Article No. 26312. <https://doi.org/10.1038/s41598-024-76197-0>
 - [19] Kemp, B., Zwinderman, A.H., Tuk, B., Kamphuisen, H.A.C. and Oberyé, J.J.L. (2000) Analysis of a Sleep-Dependent Neuronal Feedback Loop: The Slow-Wave Microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, **47**, 1185-1194. <https://doi.org/10.1109/10.867928>
 - [20] Kemp, B. (2013) Sleep-EDF Database Expanded, Version 1.0.0. <https://physionet.org/content/sleep-edfx/1.0.0/>
 - [21] Rechtschaffen, A. and Kales, A. (1968) A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects. U.S. Government Printing Office.