TimeSoccer：用于足球解说生成的端到端多模态大型语言模型

Ling You[1]
China East China Normal University

Wenxuan Huang[1]
China East China Normal University

Xinni Xie
China East China Normal University

Xiangyi Wei
China East China Normal University

Bangyan Li
China East China Normal University

Shaohui Lin*
China East China Normal University

Yang Li*
China East China Normal University

Changbo Wang*
China East China Normal University

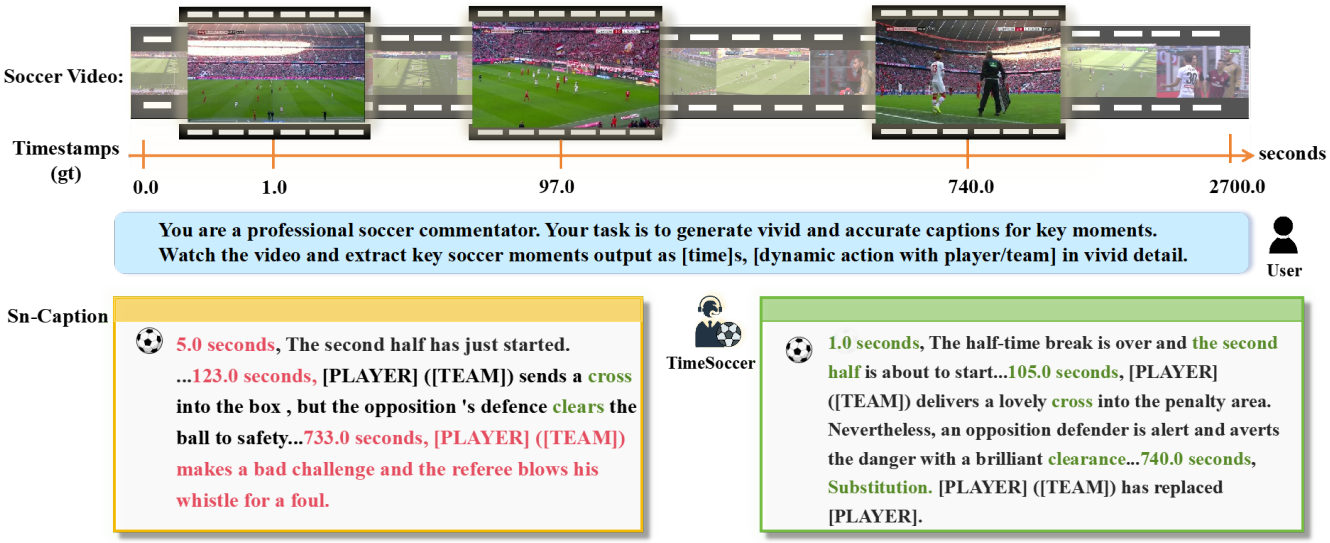

Figure 1: SoccerNet-Caption [?] 和我们提出的 TimeSoccer 在足球解说性能上的比较。传统方法（左）在时间对齐和字幕质量上表现出有限的准确性，而 TimeSoccer（右）则能生成基于上下文的描述，与真实事件的对齐更好。绿色（红色）文本表示时间戳或内容预测正确（错误）。

Abstract

为了解决上述问题，我们提出了 TimeSoccer，这是第一个用于全场比赛足球视频单锚密集视频描述（SDVC）的端到端足球 MLLM。

CCS Concepts

• **Computing methodologies** → *Computer vision.*

Keywords

Video Captioning, Multimodal Model, Temporal Localization

*Corresponding authors. Email: {shlin, yli, cbwang}@cs.ecnu.edu.cn

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Ling You[1], Wenxuan Huang[1], Xinni Xie, Xiangyi Wei, Bangyan Li, Shaohui Lin, Yang Li, and Changbo Wang. 2018. TimeSoccer：用于足球解说生成的端到端多模态大型语言模型. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 介绍

足球是一项具有全球影响力的运动，仍然是世界上最受欢迎的运动之一。这种解耦设计使得解说生成紧密依赖于时间标记模块，导致一个不仅限制性而且次优的流程，因为每个解说生成都仅基于一个短片段，从而限制了模型捕捉全球背景的能力，如图所示。

为解决上述问题，我们提出了 TimeSoccer，这是一种针对 SDVC 任务的端到端模型，能够捕捉全局上下文并以单次通行的端到端形式联合预测时间戳和评论，用于生成足球评论。图 1 展示了我们的方法与传统方法在评论能力上的比较。为了应对足球比赛中长视频理解的问题，我们首先通过引入 MoFA-Select，一个运动感知、由细到粗自适应帧选择模块，扩展了时间感知的大型语言模型 TimeChat [?] 。该模块将类似帧进

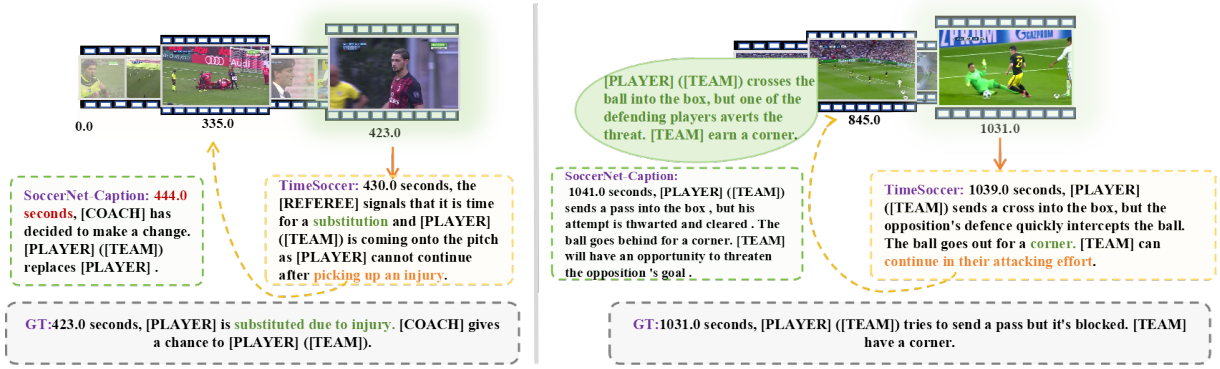

Figure 2: 一个例子展示了 TimeSoccer 与 SN-Caption [?] 相比的全局语境理解。左侧面板：TimeSoccer 正确地将球员替换归因于之前的伤病事件；右侧面板：TimeSoccer 识别了比赛中较早发生的角球事件，并适当地继续解说 "[TEAM] 可以继续他们的进攻努力。" 绿色（红色）文本表示准确（不准确）模型生成的时间戳或描述。

行聚类，并迭代合并冗余，以形成固定长度的表示，旨在保留语义上重要和对运动敏感的帧，同时降低推理资源成本，并确保与下游网络模块的输入限制兼容。此外，我们结合了两种互补策略来支持长距离时间建模：(i) 逐步学习计划，该计划在训练过程中逐渐增加视频长度，帮助模型适应更长的时间上下文；(ii) 位置嵌入外推策略，使模型能够在其原始时间范围之外进行泛化。通过联合利用这两种策略，模型能够更好地捕捉长距离时间依赖关系，并在整个 45 分钟的足球比赛中保持语义连贯性。大量实验表明，我们的方法在 SDVC 任务上达到了 SoTA 性能，表现出高时间精度并生成语义高质量的字幕。

我们的主要贡献如下：

- 我们提出了第一个用于全场足球视频评论生成的端到端足球多模态语言模型，为 SDVC 任务引入了一种新的范式，具有增强的全局上下文建模能力。
- 我们提出了 MoFA-Select，这是一种无训练的帧压缩模块，通过粗到细的策略通过特征相似性选择代表性的帧，保留关键信息以促进长视频理解，并进一步从学习和架构角度引入渐进式训练策略，以保持完整的 45 分钟足球比赛的性能。
- 广泛的实验和消融研究证明，我们的模型通过捕捉全局上下文，以实现精确定位和语义丰富的评注，达到 SDVC 任务上的 SoTA 表现。

2 相关工作

2.1 视频理解的多模态大模型

视觉-语言建模的最新进展促成了强大架构的开发 [? ? ? ? ?]，在一系列广泛的任务中取得了令人印象深刻的成果，包括分类、图像字幕生成和图文检索。在这些坚实基础之上，视频理解领域近年来取得了快速进展，多模态语言模型（MLLMs）在多模态对话 [? ?]、事件预测 [?]、时间定位 [? ? ?] 和密集视频字幕生成 [? ? ?] 等任务上表现出色。在密集视频字幕生成领域，Video-LLaMA [?] 引入了将视觉表示与语言提示对齐的技术，利用 LLaMA [? ?] 来生成视频描述。TimeChat [?] 通过将时间信息注入图像 Q-Former [?] 来对齐视频帧与其相应的时间戳，使得时间段的预测更加准确。然而，目前大多数的视频字幕生成工作着眼于开放世界场景 [? ? ?]，忽视了足球特定解说生成的独特挑战，这不仅要求事件发生的精确定位，还需要产生准确且真实的足球特定描述的能力。

长视频理解在计算机视觉中仍然是一个具有挑战性的任务，因为在数以千计的帧中保持时间一致性和捕捉关键信息存在困难。为了应对这一挑战，几项工作 [? ?] 提出了基于内存的机制，以在时间上保存重要内容。例如，XMem [?] 使用多槽内存架构，其中互连的特征存储模块有效处理长视频，而 MovieChat [?] 采用结合短期和长期记忆的混合内存设计进行时间推理。同时，许多令牌缩减策略 [? ? ?] 为长视频建模提供了洞见：ToMe [?] 将相似令牌合并到共享注意力簇中，而 G-Prune [?] 通过基于图的语义推理来修剪冗余令牌。这些方法提高了 MLLMs 在长视频形式上的效率和扩展性。然而，它们通常忽略了运动模式或依赖复杂的内存设计，限制了其对现实世界体育的适用性。为此，我们引入了 MoFA-Select，这是一种帧级选择策略，能够自适应地保留运动敏感和语义重要的帧，以实现长时间足球视频的理解。

大多数关于足球视频理解的现有研究都是基于 SoccerNet [? ?] 数据集系列，涵盖了诸如动作识别 [?]、球员跟踪 [?] 和评论生成 [? ? ?] 等各种任务。随着 MLLMs 的出现和快速发展，近期的工作 [? ?] 开始探索将 MLLMs 应用于足球评论生成。这些方法通常将短视频片段（大约 30 秒）传入 MLLMs，以为每个剪辑生成一个相应的字幕。然而，它们通常忽视了时间定位方面，依赖于真实的时间戳作为输入。即使在包含时间建模的模型中 [?]，主要的范式仍然是一个两阶段的流程：首先使用一个识别模块来识别与关键事件对应的时间戳，然后使用一个字幕生成模块基于修剪过的视频片段生成字幕。这种分离的设计使得评论生成紧密依赖于时间戳模块，并阻止模型捕获全局上下文。相比之下，我们提出的框架支持在单次处理中端到端地预测时间戳和评论，从而实现更连贯的全局理解，并更好地满足现实世界足球评论生成的需求。

3 方法

在本节中，我们介绍我们提出的模型 TimeSoccer，这是一种为足球解说生成设计的端到端 MLLM。我们首先在第 ?? 节中对问题进行公式化。第 3.1 节概述了 TimeSoccer 的整体流程及其所依赖的基线模型。第 ?? 节详细描述了提出的 MoFA-Select 模

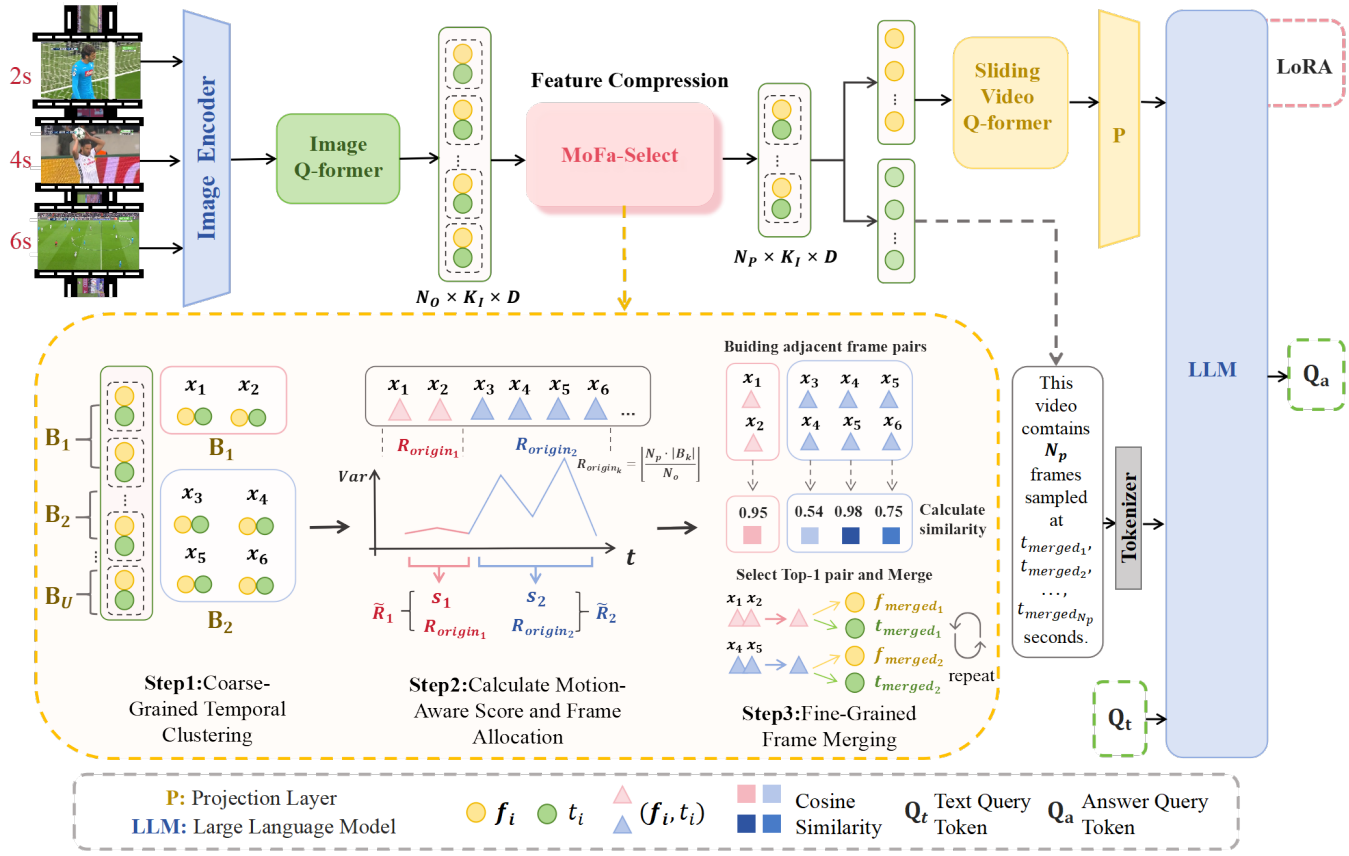

Figure 3: TimeSoccer 概览。对于一段完整的 45 分钟足球视频，首先使用图像编码器和 Image Q-Former 提取帧特征，而时间戳则从原始帧序列中获取。接着，这些特征和时间戳由 MoFA-Select 模块处理，该模块：(a) 应用时间限制的 K-Means 聚类对帧进行分段，(b) 计算运动感知分数以分配帧预算 $\tilde{R}_k$，(c) 合并每个段内的冗余帧。经过压缩的特征通过滑动的 Video Q-Former 生成视频标记，这些标记在输入到 LLM 进行最终预测之前，与基于时间戳的文本标记和用户查询标记连接在一起。

块。最后，在第 ?? 节中，我们介绍了我们的训练策略，以进一步增强模型的长视频理解能力。

给定一个足球比赛视频 即 ，$\boldsymbol{V} \in \mathbb{R}^{R\times 3\times H\times W}$ ，我们旨在构建一个端到端模型 $\Phi_{\text{TimeSoccer}}$ ，用以识别视频中所有需要进行解说的时刻 $\hat{\boldsymbol{T}}$ ，并为每个检测到的时刻生成相应的足球解说 $\boldsymbol{C}$ 。每个解说应描述和解释在视频中该特定时刻周围发生的事件，这可以公式化为：

$$(\hat{T}, C) = \Phi_{\text{TimeSoccer}}(V) \tag{1}$$

，其中 $(\hat{T}, C) = \{(\hat{t_1}, C_1), (\hat{t_2}, C_2), \ldots, (\hat{t_n}, C_n)\}$ 表示预测的时间戳及其对应的解说标题的集合。

3.1 架构

我们提出的 TimeSoccer 包含几个关键组件，包括时间感知帧视觉特征提取器、相似性感知帧压缩模块（MoFa-Select）、滑动视频 Q-former 和大型语言模型（LLM），如图 3 所示。给定一个输入视频，时间感知帧编码器首先从视频帧中提取时间特征。然后，这些特征由 MoFA-Select 模块处理，该模块比较帧级表示，仅保留最具代表性的关键帧特征。选定的关键帧特征随后通过滑动视频 Q-former 传递，以生成时间关联的视频标记。最后，这些视频标记被输入到 LLM 中以产生解说输出。基线。我们采用 TimeChat [?] 作为我们的基线，因为它在长视频的时间理解建模方面表现强劲。对于给定的输入视频 $\boldsymbol{V}$ ，模型首先采用时间感知帧编码器从每个帧中提取时间特征，即 ，$\boldsymbol{F} \in \mathbb{R}^{R\times K_I\times D}$ 。具体而言，这个编码器附加帧的时间戳，例如 “This frame is sampled at 2s.”，作为 Q-Former 的条件，使其能够融合视觉和时间信息。

为了捕捉帧间的时间关系，TimeChat 使用了滑动视频 Q-former。它引入了一个长度为 M_w 的滑动窗口，该窗口以步长 S 依次跨越已采样的帧特征移动。每个窗口内的特征被传递到视频 Q-former 中，生成形状为 $(R/S)\times k_v \times D$ 的视频标记序列。此机制缓解了长视频理解中过度压缩的问题。

结果视频标记 $\boldsymbol{Q}_v$ 通过视频投影层进行投影，并与相应的文本标记 $\boldsymbol{Q}_t$ 连接，然后输入到大型语言模型中以生成最终输出 $\boldsymbol{Q}_a$ 。在训练期间，给定长度为 M_a 的答案序列 Q_a ，计算相应的损失 $\mathcal{L}$ 如下：

$$\mathcal{L} = -\sum_{i=1}^{\mathrm{M_a}} \log P(\boldsymbol{Q}_a^{(i)} \mid \boldsymbol{Q}_a^{(<i)}, \boldsymbol{Q}_v, \boldsymbol{Q}_t) \tag{2}$$

尽管 TimeChat 展现了对长视频的强大时间定位能力，但在单次处理极长输入（例如 45 分钟的比赛）时仍然存在困难。为了在保留必要的运动信息的同时有效处理此类长视频输入，

我们提出了 MoFA-Select，这是一种运动感知的、由粗到细自适应帧选择方法。该方法旨在从大量采样帧中提取语义重要和运动敏感的帧，同时减少内存占用并确保与原始滑动视频 Q-former 的输入长度限制兼容。

视觉特征提取。给定输入视频 $\boldsymbol{V} \in \mathbb{R}^{R\times 3\times H\times W}$，我们首先对 N_o 帧进行采样，并使用时间感知帧特征提取器 $\Phi_{\text{extractor}}$ 提取其视觉特征。此过程可以被如下公式描述：

其中 $\mathcal{X}_{N_o}$ 表示提取的特征 $\boldsymbol{f}_i$ 及其对应的时间戳 t_i 的集合。我们的目标是将初始 N_o 帧压缩成具有固定长度 N_p 的更短序列。

粗粒度时间聚类。随后，我们采用由粗到细的压缩策略，以从全局和局部视角捕捉关键信息。首先，我们对获得的集合 $\mathcal{X}_{N_o}$ 进行粗粒度分段。为了识别语义和视觉上相似的时间段，我们计算集合 $\mathcal{X}_{N_o}$ 中每对特征之间的余弦相似度，如下所示：

$$\text{sim}(\boldsymbol{f}_i, \boldsymbol{f}_j) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\|_2 \, \|\boldsymbol{f}_j\|_2}, \quad \forall i, j \in \{1, \ldots, N_o\} \tag{3}$$

其中 sim(·) 计算两个特征向量之间的余弦相似度。具体来说，我们实现了一种连续的 K-Means 聚类变体，该变体施加时间连续性约束。通过最小化累积余弦距离，帧最初被聚类为 U 段：

$$\mathcal{L}_{\text{clust}} = \sum_{k=1}^{U} \sum_{t \in \text{cluster}_k} \left(1 - \text{sim}(\boldsymbol{f}_t, \boldsymbol{c}_k)\right) \tag{4}$$

，其中 $\boldsymbol{c}_k$ 表示簇 k 的质心，$\mathcal{L}_{\text{clust}}$ 表示总的聚类损失。因此，集合 $\mathcal{X}_{N_o}$ 被划分为 B_k 的子集用于 $k = 1, \ldots, U$ 。运动感知自适应帧分配。为了防止由于帧间高相似性而错误合并重要动态事件，我们为每个簇 B_k 分配一个运动感知的重要性权重，以确保具有显著运动的片段得以适当保留。具体而言，我们为每个时间簇 B_k 引入了一个运动感知分数 s_k ，该分数是基于每个簇内的方差计算的：

$$s_k = \frac{\text{Var}(\boldsymbol{f}_t \mid t \in B_k)}{\max_j \text{Var}(\boldsymbol{f}_t \mid t \in B_j)}. \tag{5}$$

在为每个簇 B_k 获得运动感知分数 s_k 后，我们根据其相对运动贡献比例地为每个分段分配帧。分配给每个簇 B_k 的帧数，记为 R_k ，具体确定如下：

$$R_k = \max\left(1, \min\left(\left\lfloor \frac{N_P \cdot |B_k|}{N_o} \right\rfloor + s_k \cdot \left\lfloor \frac{N_P \cdot |B_k|}{N_o} \right\rfloor, |B_k|\right)\right) \tag{6}$$

，其中 $R_{\text{origin}} = \left\lfloor \frac{N_P \cdot |B_k|}{N_o} \right\rfloor$ 表示在应用运动感知缩放之前，根据其相对大小分配给簇 B_k 的初始帧数。基于方程 6 ，一旦确定了每个簇的目标帧数 R_k ，我们会按比例缩放它们，以确保分配的帧总数等于最终压缩长度 N_p ，并将缩放后的目标记为 $\tilde{R}_k$ 。这样的缩放保证了平衡且运动感知的分配，优先考虑动态片段，同时严格匹配所需的压缩长度。精细粒度帧合并。在每个簇 B_k 中，受 Moviechat [?] 的启发，我们通过选择具有最大相似性的帧对，迭代地合并相邻帧中的特征。为了保留运动动态，我们基于合并帧之间的差异计算运动惩罚：

$$\Delta_{\text{motion}}(i) = \text{Var}(\{\boldsymbol{f}_i, \boldsymbol{f}_{i+1}\}) \tag{7}$$

，其中 $\Delta_{\text{motion}}(i) = \text{Var}(\{\boldsymbol{f}_i, \boldsymbol{f}_{i+1}\})$ 表示相邻帧 i 和 $i+1$ 之间的运动惩罚函数，计算为它们特征表示的方差。如果 Δ_{motion} 超过预定义的阈值 δ ，我们会丢弃冗余帧，而不是合并它们，以保留重要的运动信息。阈值 δ 根据验证性能经验设定为 0.3。否则，我们通过如下方法平均相邻帧的特征表示来合并它们：

$$\boldsymbol{f}_{\text{merged}} = \frac{\boldsymbol{f}_i + \boldsymbol{f}_{i+1}}{2}, \quad t_{\text{merged}} = \frac{t_i + t_{i+1}}{2}. \tag{8}$$

合并过程被反复迭代，直到每个簇 B_k 达到其分配的帧数 $\tilde{R}_k$ 。通过按时间顺序连接所有簇中保留的特征，我们获得最终压缩表示 $\mathcal{X}_{N_p}$ 。

结论。在应用 MoFA-Select 后，压缩表示 $\mathcal{X}_{N_p} = \{(\boldsymbol{f}_1, t_1), \ldots, (\boldsymbol{f}_{\boldsymbol{N_p}}, t_{N_p})\}$ 保留关键运动片段的同时保持时间上的连贯性，结果得到一个平衡而信息丰富的固定长度序列，用于下游处理。具体而言，视觉特征 $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{\boldsymbol{N_p}}\}$ 通过滑动视频 Q-Former 产生视频令牌，而相应的时间戳 $\{t_1, \ldots, t_{N_p}\}$ 被格式化为如下形式的文本摘要：“`This video contains` N_p `frames sampled at` t_1 , t_2 , ..., t_{N_p} `seconds.`”。这些基于时间戳的文本令牌然后与用户查询和视频令牌连接，并共同输入到 LLM 中以生成最终响应。

为了进一步使我们的模型能够理解全长的足球视频（通常为 45 分钟），我们在训练过程中从学习和结构两个角度增强其长视频建模能力。从学习的角度来看，我们采用渐进式训练策略，即模型在越来越长的视频上逐步进行微调，使其能够逐步构建理解扩展时间上下文的能力。从结构的角度来看，为了通过减少压缩保留更多的时间细节，我们采用基于周期性复制的位置嵌入外推策略，其中预训练的位置嵌入被周期性地重复以覆盖所需的长度。这种方法确保了与预训练模型的无缝兼容，同时能够灵活适应不同的输入长度。

4 实验

4.1 实施细节

我们采用来自 EVA-CLIP [?] 的 ViT-G/14 作为视觉编码器，并将 LLaMA-2 (7B) [?] 作为语言骨干网。所有剩余模块从 TimeChat [?] 初始化。我们使用原始的 Soccernet-Caption 数据集分别微调图像 Q-Former 和视频 Q-Former，同时保持视觉变换器（ViT）和大语言模型（LLM）不变，如图 3 所示。为了更好地适应 LLM 到我们的视频理解任务，我们采用参数高效的微调方法 LoRA，秩为 32。MoFA-Select 模块中的聚类数设置为 6。δ 为 0.3。所有实验均在 4 × NVIDIA H20 GPU（96 GB）上进行。

4.2 评估设置

数据集。我们在 Soccernet-Caption 数据集上微调和评估我们的模型。具体来说，我们根据 MatchTime [?] 中定义的训练/测试分割，将数据集分为 422 个训练视频和 49 个测试视频。此外，我们用 MatchTime 提供的精细化时间标签替换了原始时间戳注释，以确保更高的对齐精度。评估指标。在时间评估中，我们将地面真值和预测的时间戳两侧分别扩展 5 秒，并在 0.3、0.5、0.7 和 0.9 的阈值下计算 IoU，以衡量对齐准确性。我们还报告 F1 分数以反映精准度和召回率之间的平衡。对于字幕质量，我们采用 CIDEr [?] ，METEOR [?] ，以及 SODA_c [?] 。之前对评论质量的评估主要集中在预测和参考字幕之间的句子级别对齐，这忽视了生成评论的整体连贯性和信息性。为了弥补这一不足，我们使用 Qwen2.5-VL-72B-Instruct [?] 通过两个指标进行更全面的评估：匹配级语义对齐（M-S）和综合评论质量（C-S）。每个维度的评分范围为 1 到 10。

为了验证我们方法的有效性，我们将其与现有的最先进的 SDVC 模型，SoccerNet-Caption [?] ，以及其他近期的足球解说

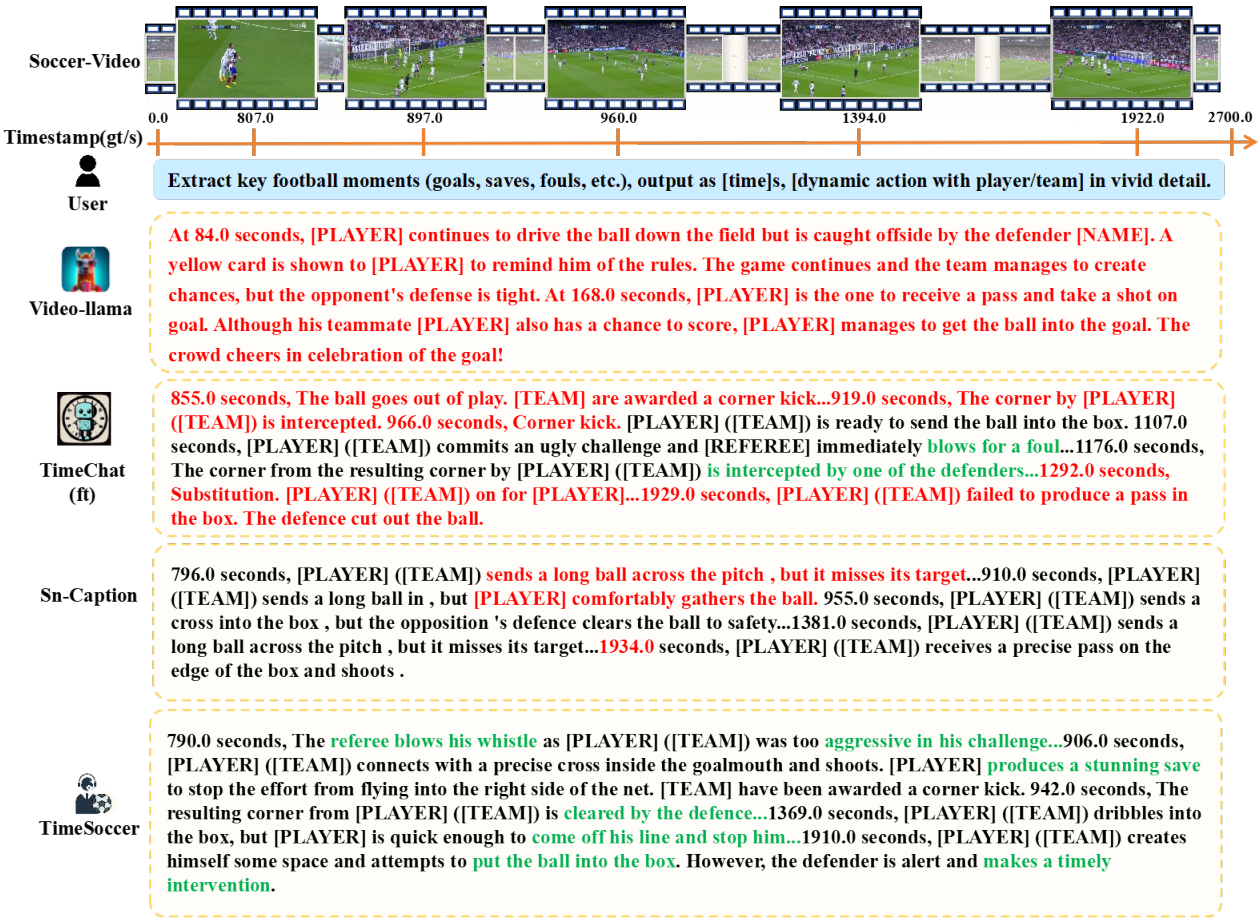

Figure 4: 不同方法的质量比较结果。TimeSoccer 从多个角度展示了其优势：(i) 更准确的时间戳对齐；(ii) 改进的事件描述；(iii) 更丰富、更逼真的评论，类似于专业广播。黑色文本表示在时间或语义上较为接近真实情况的输出，其中绿色突出显示语义准确的描述，红色标记错误或不相关的内容。

生成方法 [? ?] 进行比较。由于这些方法通常依赖于真实时间戳，我们使用 SoccerNet-Caption 预测的时间戳来提供数据，以确保公平比较。我们在 3 分钟和完整的 45 分钟环境下进行评估。虽然 45 分钟环境下整体指标分数略低于 3 分钟版本，但它能够对整个比赛视频进行端到端的推断。相反，3 分钟设置需要 15 次单独的前向传递，因此不能被视为完全的端到端方法。因此，我们采用 45 分钟设置作为代表性的评估协议，并与其他基准进行比较以评估模型的整体性能。此外，我们还包括与现有强大的基于 MLLMs [? ?] 的端到端视频理解模型的比较，使用与我们模型相同的提示设置，以进一步展示我们方法在时间定位和解说生成方面的优势。具体的定量结果呈现在表中 **??** 。时间定位准确性。如表 **??** 所示，在足球特定基准中，SN-Caption 仅依赖特征提取器、聚合器和分类头来预测每帧的字幕概率，这限制了其在精准时间定位上的鲁棒性。基于 SN-Caption，MatchTime 和 UniSoccer 继承了其时间戳预测，而未引入明确的时间建模，从而限制了它们的泛化性和在不同场景中的有效性。对于通用的视频 MLLMs，大多数方法将帧特征压缩成固定长度的表示，使得它们在实现准确的时间定位时存在困难。相比之下，我们的方法在 45 分钟的时间定位中，在 Precision@0.3、0.5、0.7 和 0.9 上分别超越第二名 SN-Caption +4.5、+3.8、+3.0 和 +2.3。这凸显了我们模型选择突显特征并准确定位关键时刻的能力，显著提升了 MLLMs 在长视频时间理解方面的能力。

标题质量评估。如表 **??** 所示，时间定位不准确显著影响解说质量，导致 SN-Caption、MatchTime 和 UniSoccer 的字幕生成性能较低。其中，SN-Caption 的结果略好，可能是由于其字幕生成和检测模块之间除了分类头之外还共享参数，这可能提供了更好的时间兼容性。对于通用视频 MLLMs，缺乏特定于足球的适应以及对长时间结构模型的不充分，极大地限制了其生成精确且具上下文丰富的足球字幕的有效性。相比之下，我们的方法取得了显著的性能提升，在 3 分钟视频中超过之前的前沿技术 +2.8 CIDEr，+0.6 METEOR，以及 +0.4 SODA_c。此外，它在完整的 45 分钟视频中仍然保持竞争力，展示了强大的长时间理解能力。

模型的多功能性。我们使用 Qwen2.5-VL-72B-Instruct 评估生成评论的质量，重点关注两个方面：匹配级别的语义对齐（M-S）和综合评论质量（C-S）。如图表 **??** 所示，在 45 分钟设置下，我们的方法在 M-S 和 C-S 上分别优于 SN-Caption +1.62 和 +0.84，这得益于语言模型强大的叙述能力。此外，45 分钟设置产生

的匹配级别对齐要高于 3 分钟设置，表明全长生成有助于更有效地捕捉全局上下文。

4.3 消融研究

我们进行了系列消融实验，以评估我们框架中关键组件的有效性。具体来说，我们评估了 MoFA-Select 模块和互补训练策略的影响。我们进一步在附录中考察了关键超参数变化的影响。这些研究提供了关于每个部分如何为我们方法的整体有效性做出贡献的见解。MoFA-Select 的影响。如表 1 所示，当不使用 MoFA-Select 并且模型仅在 SN-Caption 数据集上进行微调时，所采样帧的数量有限，导致关键时刻识别效果不佳。这在更严格的时间 IoU 阈值（0.7 和 0.9）下导致性能下降，以及较低的 CIDEr 和 METEOR 分数（-1.8, -0.6, -2.8, -1.1）。G-Prune [?] 和 MoFA-Select 都使模型能够采样更多帧并选择信息量最大的帧，从而改进关键时间戳和语义内容的定位。然而，MoFA-Select 始终优于 G-Prune，因为后者仅依赖于全局相似度传播且倾向于保留冗余片段。此外，我们对 MoFA-Select 的四个组件进行了详细的消融。结果显示，去除任何一个组件都会导致性能下降，这突显了每个组件在实现有效长视频理解中的必要性。
渐进式训练策略的影响。如表 2 所示，我们的渐进式训练策略逐步增强了模型理解长篇视频的能力，在 CIDEr 和 METEOR, SODA_c 方面相较于直接训练取得了显著的改进，这证明了其在促进长时间范围的时间建模中具有效性。此外，扩展位置编码在直接和渐进训练设置下均可提升表现。我们进一步比较了两种扩展方法：插值和周期性复制。结果显示，周期性扩展优于插值，这可能归因于大幅超出原始长度进行插值时引入的过度平滑。这种平滑扭曲了位置语义，而周期性复制则保留了原始嵌入的分布，从而带来更稳健的性能。

4.4 定性比较

如图 4 所示，我们比较了多种模型在 SDVC 任务上的性能，包括 VideoLLaMA、在 45 分钟视频上微调的 TimeChat、SN-Caption 和我们提出的 TimeSoccer。结果突出了 TimeSoccer 的以下优势：(i) 改进的时间锚定能力：配备了 MoFA-Select 模块，TimeSoccer 能有效捕捉关键的视觉和时间线索，使生成的评论与关键事件之间的对齐更加准确。(ii) 增强的上下文连贯性：TimeSoccer 改进的时间锚定能力使其能更好地将评论与事件进程对齐，产生更准确、逻辑一致的描述。(iii) 更丰富的语义内容：MLLM 的显著能力帮助 TimeSoccer 生成更详细且具有上下文意识的描述，提升了可读性和领域相关性。更多的分析结果可以在附录中找到。

在本文中，我们提出了 TimeSoccer，这是第一个用于完整足球比赛的单锚密集视频描述 (SDVC) 的端到端框架。与依赖于真实时间戳或基于短视频片段的两阶段范式的先前方法不同，TimeSoccer 在一次前向传递中同时预测时间段并生成描述。该端到端设计能够整体建模长时间段的上下文，并允许直接推断完整的 45 分钟足球比赛。为了支持高效的长视频理解，我们提出了 MoFA-Select，一个无需训练、运动感知的帧压缩模块，使用从粗到细的策略自适应地选择代表帧。最后，我们采用渐进式训练策略以进一步加强模型的时间推理能力。大量实验证明，TimeSoccer 在时间定位和评论质量上都优于现有的基线。

Table 1: 在 45 分钟视频设置下，MoFA-Select 与标准 SFT 和基线压缩方法在 Soccernet-Caption 数据集上的比较。

	Temporal Metrics ↑					Caption Quality ↑		
Method	P@0.3	P@0.5	P@0.7	P@0.9	F1	CIDEr	METEOR	SODA_c
TimeChat [?]	0.2	0.2	0.2	0.0	0.0	0.0	0.0	0.0
TimeChat (ft) [?]	13.4	9.0	4.1	1.6	8.2	5.5	5.1	2.6
Ours (G-Prune [?])	15.5	9.9	4.7	2.4	8.5	7.7	5.8	2.7
Ours w/o (Fine Stage [?])	13.8	9.4	4.7	2.4	8.0	7.3	5.4	2.6
Ours w/o Coarse Stage	15.0	9.9	4.6	2.2	8.5	7.5	5.6	2.8
Ours w/o Motion-Aware	15.8	10.5	5.1	2.6	8.5	6.1	2.9	4.1
Ours w/o time-merge	12.9	9.0	4.8	2.6	7.3	7.1	4.8	2.2
Ours (Full MoFA-Select)	17.0	11.0	6.0	3.4	8.8	8.3	6.2	2.7

Table 2: 在 Soccernet-Caption 数据集上关于全匹配视频理解的训练范式和位置编码扩展的消融研究。

	Temporal Metrics ↑				Caption Quality ↑		
Method	P@0.3	P@0.5	P@0.7	P@0.9	CIDEr	METEOR	SODA_c
Direct Training							
Direct (45-min Only)	13.4	9.0	4.1	1.6	5.5	5.1	2.6
Direct (45-min + Interpolated PosEnc)	13.2	8.9	5.0	2.6	6.7	4.5	1.2
Direct (45-min + Repeated PosEnc)	15.4	10.8	5.2	3.0	8.7	4.6	0.9
Progressive Training							
Progressive (3⟶15⟶45-min)	13.8	9.0	4.7	2.7	8.6	4.6	1.2
Progressive (3⟶15⟶45-min + Repeated PosEnc)	15.3	10.3	4.9	2.2	9.2	5.7	2.6
Progressive + MoFA-Select (Full Model)	17.0	11.0	6.0	3.4	8.3	6.2	2.7