

## 全景直播服务系统的挑战：高带宽、低延迟以及计算资源

急需：提高体验质量

### 一、相关工作：

- 第一类解决方案：通过先训练后预测的方法来提高 fov 预测的准确性

（用户一般只会看全景视频的一部分，如果只为 fov 内的分配高码率，会节省带宽。但这个性能取决于 fov 预测的准确性）

fov 是 pls 的热门：最开始的人们解决上述的方案主要是基于 fov 预测的准确性（不准确会导致带宽浪费，因为一般在系统中 fov 内会以高码率传输，fov 外会以低码率传输）

方法：深度学习、考虑头部运动以及显著性特征

缺点：1. 先训练来后预测，依赖前期的训练，需要大量的数据  
2. 引入了大量模型训练的延迟，但全景直播系统需要低延迟, 所以不能直接用于直播系统

- 第二类解决方案：新的在线解决方案（在线 FoV 预测模型、实时模型更新和实时 FoV 推理机制等）

方法：中央在线（livedeep）、强化学习、深度强化学习、多模型深度强化学习框架。高级的 pls 需要采用集中的解决方案，需要给

每一个人提供在线 fov 预测，非常昂贵。观看者多时，会出现扩展性问题，所以有人提出多代理深度强化学习，利用观看者行为相似性提出跨用户学习框架。

**缺点：**未考虑到实时全景视频是通过一个异步网络系统传输的，观众地理位置不同，流延迟不同，存在时间差。如果忽略时间差，可能会使相似性分析和协作模型训练在部署上变得困难，因为观看者在观看时会有一些对时间变化的个性化动作。

- 现在大多数协作式 360°视频传输解决方案是同步的

所以文章提出了异步的传输框架

## 二、贡献：

本文提出了一个新颖的异步成本效益的联合传输框架，称为 FedLive，用于 PLS。

FedLive 包括两个主要贡献：优化了 FoV 预测和内容交付

- 提出一个基于梯度的观看者聚类（GVC）方法，以提高模型训练和数据传输的通信效率；

详细来说，通过利用梯度损失的几何特性，根据观看者的视口和

头部动作（观看行为）的相似性，将其分成内容传输联盟。

- 提出一个基于强化变分推理（RVI）结构的异步传输方法，其应用是基于预测的异步传输（PAD）算法，以协助 FoV 预测模型的训练，同时加速多个多速率切片的交付。

RVI 的目标是 FoV 预测，这是一个典型的概率推理问题，因为每个视频切片都有一定的概率被观看者观看。实际上这里做的就是将 fov 预测转化为了在线推理问题。

RL 框架为问题解决中的概率推理提供了一种交互式的进化学习设计，并为设计一种具有推理技术的在线异步算法带来了潜力。在这种情况下，强化变分推理（RVI）是一种新颖的基于结构的联合预测和交付方法，它通过在 RVI 框架空间中制定联合优化问题，优化了分布式在线学习和基于切片的交付过程。

### 三、动机：

#### 1. 设计系统动机概述

不同用户观看行为不同，会根据自己感兴趣的内容切换视角，全景直播系统要在**观众差异化偏好**和**随机行为**的驱动下，为每个人提供个性化的 360°直播流。视频切片是最近很流行的解决上述问题的方式，通过这个行为，PLS 提供者也可以通过为观看者提供背景的低清晰度切片和感兴趣区域的高分辨率切片来节省带宽。基于视频切片的方案还使 PLS 系统能够在具有类似观看行为的用户之间重复使用切片。所以有必要设计这样一种数据传输方案，它采用先预测后

发送的方式，有效地识别和分配基于用户观看特征而即时生成的多个视频切片，以实现高质量的 PLS。

提供高质量的 PLS 需要两个关键技术的支持。

- 一种对时间敏感的、高精度的 FoV 预测方法，可以实时预测每个观众的需求切片；
- 一种基于切片的高效数据传输，可以灵活地调整切片的传输路径，以适应时间变化的网络条件和全球观众的观看行为。

## 2. 系统设计的参考

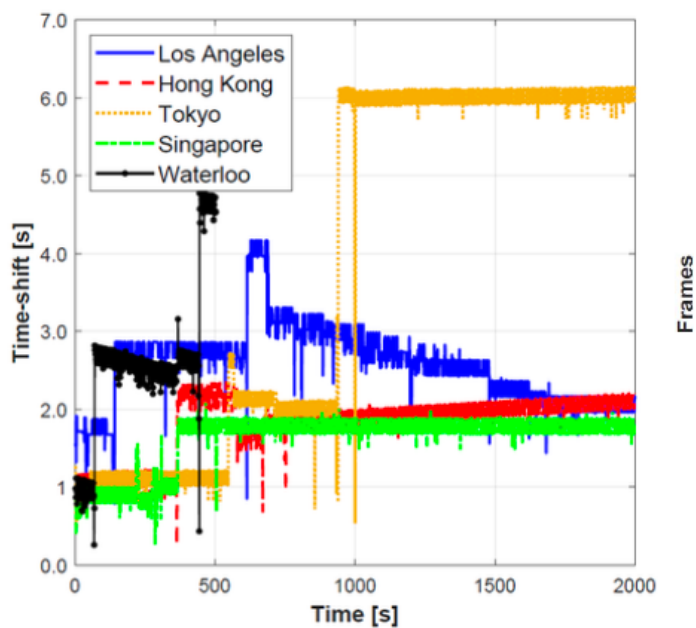
作者收集位于不同地区的内容提供商和其观众之间的延迟。然后，根据收集到的数据和数据集，进行追踪驱动的研究，对用户进行剖析，并为 FedLive 的设计提供参考。

- 时移现象（不同的观众在请求和接收同一帧视频之间存在一个时间差。我们称这方面为时间偏移现象。）

—**造成原因：**视频传输（内容交付）需要时间，比如 PLS 系统将收集全景视频内容发送到区域数据中心等

—**影响时移时间的因素：**地理位置、异质网络条件和服务器的工作量

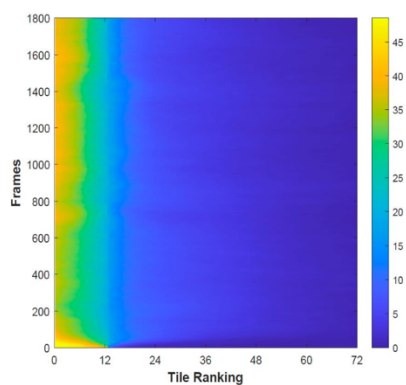
—**总结：**如图所示，时移现象很普遍，由于地理分布的原因，时移时间甚至可以达到 4 秒以上

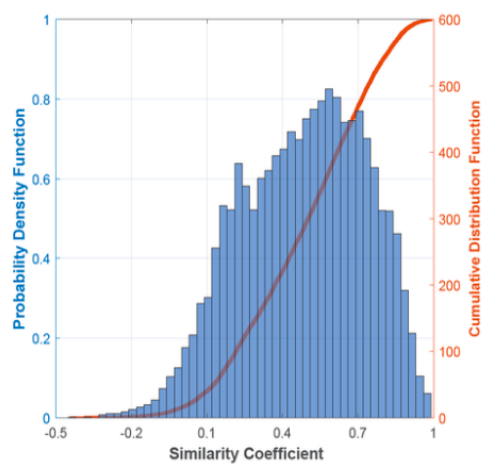
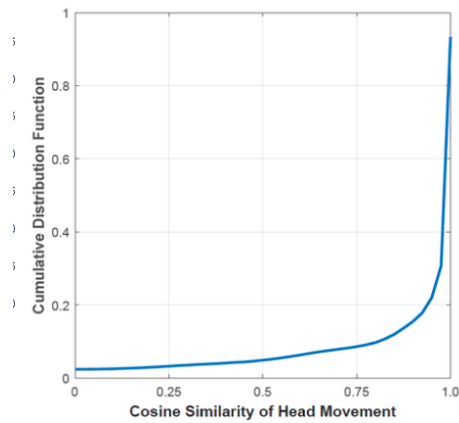


(a) The shifted time of different regions

- 相似行为

对观看者观看行为的相似性分析是基于一个公共数据集[32]，它包含了 48 个用户对 18 个不同视频的头部运动轨迹。分析了不同观看者之间的 FoV 覆盖率、视点运动的相似性





**(d) Correlation of FoV coverage and head move**

——时移现象——FOV 预测模型重复使用（一些用户看的 fov 几乎一样，只是看到时间先后问题）

——相似行为——切片重复使用（相似行为的用户有相似的切片需求）

这些给我们带来了设计一个新的传输框架的灵感，该框架允许模型和切片被联合重用，以促进 PLS 中的相互作用。为此，我们选择聚类联合学习作为标准，将有可能重复使用数据的观众分组，并进一步

步提供一个强化的变异推理方案，以加速预测模型的训练和数据传输。

四、fedlive 框架和系统模型

- 网络模型
- 切片 PLS 的 FoV 预测
- FedLive，即 PLS 的联合传输框架

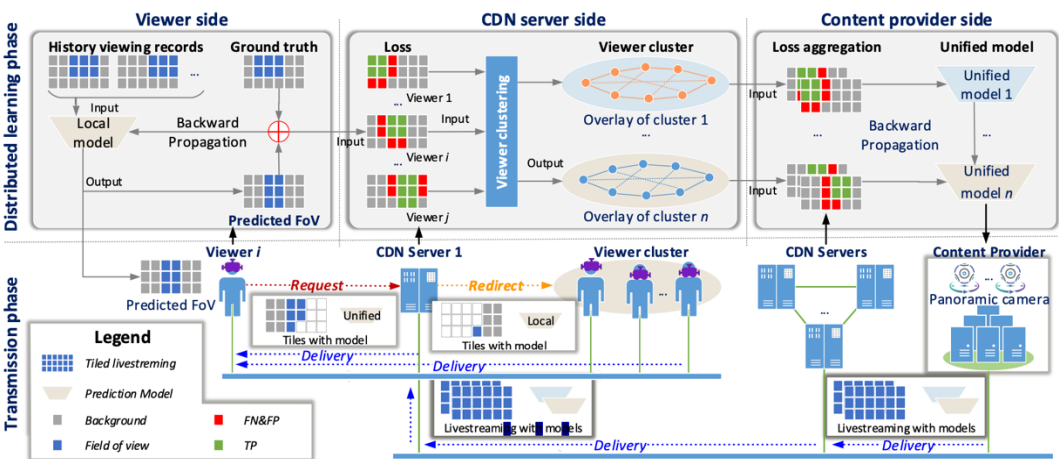


Fig. 2. Diagram of Federated Transmission Framework for PLS.

六、性能评估

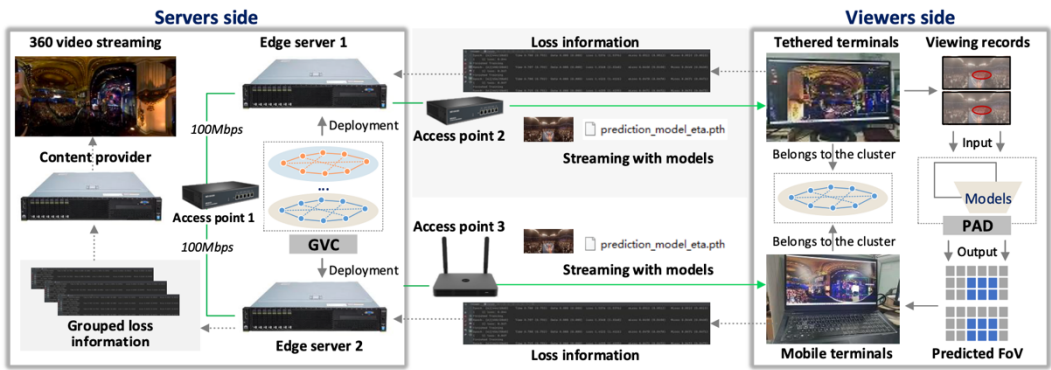


Fig. 3. The topology diagram of the prototype system.

1. 实验模拟设置：
- 代码环境：Python3.8

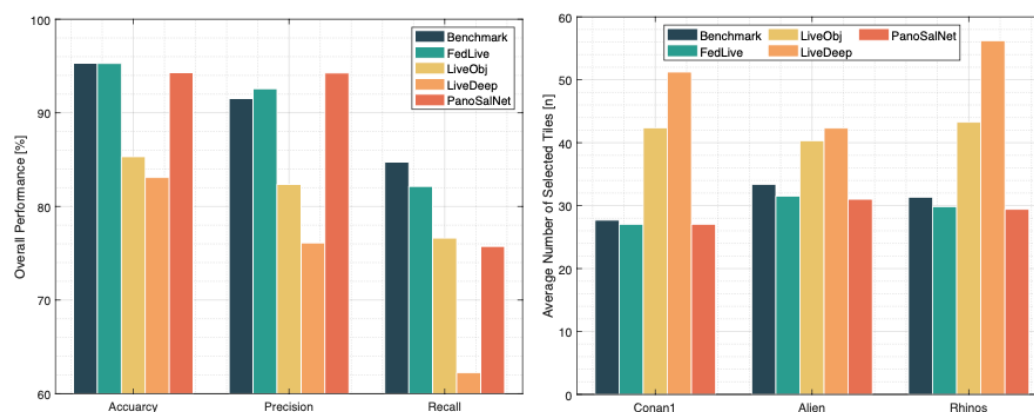
- 模拟场景：一个内容提供者、两个边缘服务器和 48 个虚拟观众的场景
- 模拟时移现象：配置各种启动延迟来实现时间偏移现象
- 模拟全景视频的切片机制：开发了一个基于（OpenAI Gym 是一款用于研发和比较强化学习算法的工具包）OpenAI-Gym6 库的 FoV 预测的平台
- 数据细节：
  - 48 个用户（24 个男性和 24 个女性）
  - 观看 18 个不同类别（包括演讲、体育和比赛）的全景视频时的头部轨迹
  - 用户在每个会话中的头部运动
  - 用户在全景视频中的兴趣目标
  - 用户在全景视频中的单位四元数（X、Y、Z、W）
  - HMD 位置（X、Y、Z）的头部方向

## 2. 实验结果

- 基准 **Benchmark**：集中式同步 RVI-结构算法（PLS 系统只包含内容提供者和观看者，没有时间偏移，而算法被部署在具有全局信息的内容提供者一方）
- 比较对象：PanoSalNet[7]、LiveDeep[13]和 LiveObj[14]  
 PanoSalNet 是一个离线解决方案，通过 360 特征提取和多模式学习来预测视口  
 LiveDeep 是一个在线训练解决方案，在服务器端部署预测模型。使用 CNN 来捕捉图像特征，并使用 LSTM 来分析观众的观看轨迹。它预测观众的视口基于这两个结果。  
 LiveObj 在部署方面与 LiveDeep 相似。它在 RL 结构中制定了 FoV 预测问题，其中状态由物体和视口在切片中的覆盖率表示，动作是 FoV 预测结果。
- 比较内容：准确性、精确性、召回率、服务性能、资源效率



(1) 准确性、精确性、召回率和带宽消耗（预取切片的数量）优于其他方案



左边图：

--FedLive 和 Benchmark 比其他三个解决方案提供了**最佳的平均性能**，与 PanoSalNet 相比，其性能提高了近 1 个百分点。

--与 PanoSalNet 相比，FedLive 和 Benchmark 提供了最佳的平均性能。然而，PanoSalNet 在精确度方面优于其他所有解决方案，比 FedLive 高出约 1 个百分点

--Fedlive 在准确性、精确度和召回率方面都比 LiveDeep 和 LiveObj 有优势。在异步 PLS 系统中，FedLive 优于 LiveDeep 的一个重要原因是，我们的解决方案可以通过利用多个观察者合作训练预测模型来提高预测性能。

--在召回率方面，fedlive 比 PanoSalNet 高约 5%。尽管 PanoSalNet 与我们提出的解决方案相比，取得了相当的预测性能，但 PanoSalNet 需要一个离线训练过程，很难应用于全景直播服务。

右边图：

--Fedlive 和 PanoSalNet 都取得了很好的预测结果，这两个解决方案通过预取较少的高码率切片给观看者来节省带宽资源。

--Livedeep 的性能最差，因为它取了最多的高码率切片，而这些切片是在观看者 fov 之外的

（2）不仅在准确率、精确度和召回率方面优于其他解决方案，而且对于不同类型的视频具有更稳定的性能。

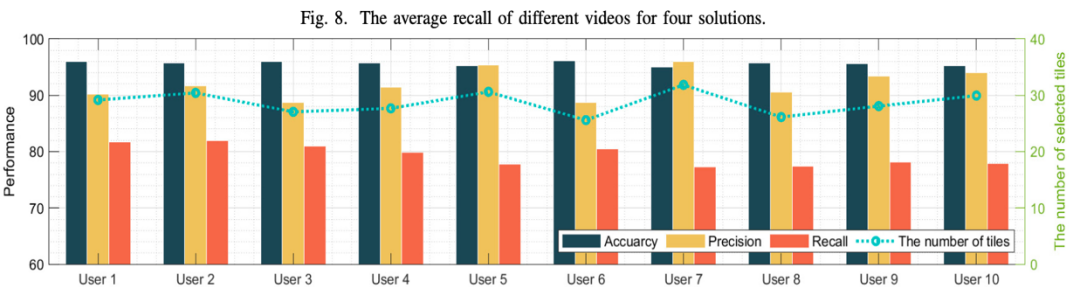


Fig. 9. Overall performance (accuracy, precision, recall, and the number of selected tiles) of different viewers for *Conan 1* with *FedLive*.

不同用户的预测性能有很小的差距，表明 FedLive 的预测性能在不同观众之间是稳定的。

（3）服务性能

在模拟中测试了所选切片的平均分辨率[Mbps]、播放实时视频时的平均缓冲区大小[秒]、带宽节省[Mbps]和比特率变化率[%]。

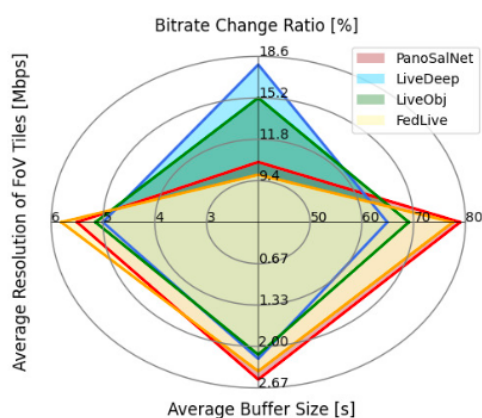


Fig. 12. The results of service performance and bandwidth efficiency

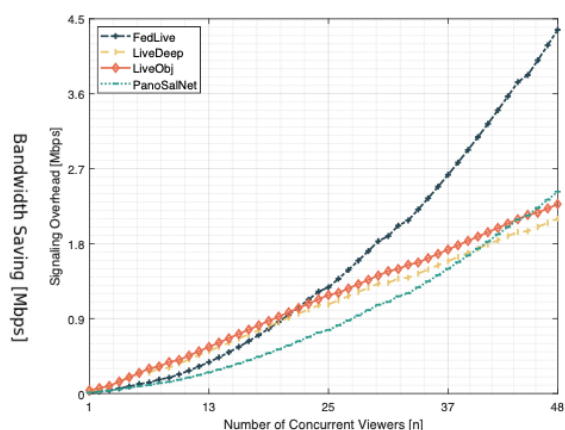


Fig. 13. Signalling overhead of control message for four solutions.

左图

--FedLive 保持了最高的所选切片的平均分辨率和最低的比特率变化率，与其他三个解决方案相比，在节省带宽和平均缓冲区大小的性能上有竞争力。

--Fedlive 在服务性能方面仍然优于 LiveDeep 和 LiveObj，并且与离线解决方案 PanoSalNet 相比具有相当的性能。

右图

--比较了 FedLive 与 LiveDeep、LiveObj 和 PanoSalNet 的信号成本  
 --FedLive 比其他解决方案需要更多的消息交换，但该方法的两层传递结构限制了信令成本，它随着观众数量的增加而线性增加，因此是可扩展的。

## 五、 基于梯度的观众聚类 and 强化变异推理的联合传输优化

- 首先提供了关于强化变异推理（RVI）的必要背景
- 简要介绍了聚类联邦学习和基于梯度的用户聚类
- 在 RVI 结构中提出了 FoV 预测和切片传输的联合问题

## 1、背景

### 2、基于梯度的观众聚类（GVC）

在 FedLive 框架中采用聚类联合学习时，我们需要考虑用户聚类的工作流程。

#### 第一个观众：

- PLS 系统在第一个观众从内容提供商那里请求一个特定的视频时开始。
- 该请求将被附近的 CDN 服务器捕获。
- 然后，CDN 服务器向观众提供所需的实时流媒体和一个统一的模型。
- 观看者将使用统一的模型作为本地模型来预测 FoV，并将梯度损失上传到 PLS 系统。
- 同时，观看者可以选择是否训练本地模型，或者只是等待内容提供商制作的下一个统一模型。

#### 随着观众增多：

- CDN 服务器将收集由访问观众更新的经验风险
- 计算分离标准。如果符合分离标准，CDN 服务器将首先对观看者进行分组，并将损失梯度分批传送给内容提供商。

（任何两个具有相同统一模型的观众之间的相似性表示为其损失梯度的余弦相似度，如下所示

$$\beta_{i,j}(\nabla_{\eta} E_i(\eta^*), \nabla_{\eta} E_j(\eta^*)) \triangleq \frac{\langle \nabla_{\eta} E_i(\eta^*) \nabla_{\eta} E_j(\eta^*) \rangle}{\| \nabla_{\eta} E_i(\eta^*) \| \| \nabla_{\eta} E_j(\eta^*) \|}$$

)

- 当内容提供商收到来自多个 CDN 服务器的分组梯度损失时，它分别为每个服务器的每个组训练一个统一模型。然后，内容提供者通过比较不同统一模型的参数  $\eta$  的余弦相似度来决定是否合并某些组。这里，我们设置了一个阈值，当余弦相似度大于阈值时，两个不同的组将被合并。
- 这时，用户聚类完成，内容提供商根据分组情况将基于瓦片的视频内容和统一模型交付给 CDN 服务器。

### 3、基于强化变分推理的问题提法

本文目标：为观众提供高精度的预测模型，同时优化数据传输

重点讨论：数据传输、如何制定联合优化

## 七、算法

### 1、GVC（基于梯度的观众聚类算法）

部署位置：**CDN 服务器**

运行周期：以定期的时间间隔  $T_i$  运行

输入：观众访问集合  $VI$ 、收集的梯度损失、余弦相似度

- 余弦相似度按照降序排序，储存观众的索引向量
- 依次取出元素，判断是否符合分离标准
- 如果满足标准，算法继续执行，算法对观众访问集合进行分割，首先将两个最相似的元素分割成一个新的观看者集合，并依次迭代该过程。

否则就退出直接向内容提供商更新数据观众访问集合  $VI$ 、梯度损失。

- 该算法会一直上传数据直到  $VI$  中没有一个观众集满足分离标准。算法最终以二分法结束。如果两分区仍然满足标准，我们将两分区作为新的输入，并分别执行算法。

## 2、PAD（基于预测的异步传输算法）

异步体现在：观看者可以异步更新数据到 CDN 服务器，同样，服务器也可以异步更新到内容提供商。

算法三个主要部分：观众、内容提供商、**CDN**

**观众：**用本地策略来预测观看者视野。可以自由地更新本地策略，或者只是将

$$E_u(\eta_t^u) \text{ and } \nabla_{\eta} E_u(s_t, a_t),$$

等信息上传到 CDN 服务器，并获得内容提供商提供的新策略  $\pi_{\eta, t}$

**CDN 服务器：**负责观看者的聚类。服务器使用梯度损失来计算  $\beta$  VI，然后采用 GVC 来获得分组观众，最后按聚类上传数据 VI、

$$E_u(\eta_t^u) \text{ and } \nabla_{\eta} E_u(s_t, a_t),$$

**内容提供者：**按标准政策梯度更新。在收到聚类信息

$$E_u(\eta_t^u) \text{ and } \nabla_{\eta} E_u(s_t, a_t),$$

后，算法通过首先计算历史回报的期望值执行更新。我们可以进一步用  $R'(\pi_{\eta, t})$  减去  $u \in V_i E_u/|V_i|$  来计算 VI 的联合行动值  $Q_t$ 。

由于  $u \in V_i E_u/|V_i|$  是时间  $t$  的平均奖励，我们认为它是政策  $\pi_{\eta, t}$  不变时预期收益的近似值。我们的目标是让政策接近效用 (11)，因此算法的第 17 步可以看成是得分  $\nabla_{\eta} \log \pi_{\eta}(s, a)$  [38]。同样，优势值是行动值和状态值之间的差异 (13b)。有了优势值，

政策参数  $\eta_{t+1}$  被更新，算法将新的统一政策  $\pi_{\eta_{t+1}}$  分发给观看者。