迈向更优的HTTP自适应流媒体体验质量

Babak Taraghi*Christian*
Doppler Laboratory ATHENA 信息技术研究所（*ITEC*）阿尔卑斯-亚得里亚大学克拉根福克拉根福，奥地利
babak.taraghi@aau.at

Selina Zo¨e Haack*Christian*
Doppler Laboratory ATHENA 信息技术研究所（*ITEC*）阿尔卑斯-亚得里亚大学克拉根福克拉根福，奥地利
zohaack@edu.aau.at

Christian Timmerer*Christian*
Doppler Laboratory ATHENA 信息技术研究所（*ITEC*）阿尔卑斯-亚得里亚大学克拉根福克拉根福，奥地利
christian.timmerer@aau.at

arXiv:2301.13523v2 [cs.MM] 1 Feb 2023

摘要——**如今，HTTP自适应流媒体（HAS）已成为一种广泛应用的多媒体传输方案。HAS的创新之处在于，借助自适应码率（ABR）算法，可以根据当前网络状况持续调整流媒体会话。Netflix、Amazon Prime Video和Twitch等多种热门流媒体及视频点播服务都采用了这种方法。鉴于其广泛的消费者群体，自适应码率算法不断改进，以提高用户满意度。这些改进所依据的见解，除其他来源外，还来自体验质量（QoE）研究领域。在这一领域，众多研究人员致力于识别潜在的视觉损伤，并测试其对观众体验质量的影响。影响体验质量的两种常被讨论的视觉损伤是卡顿事件和质量切换。迄今为止，人们通常认为卡顿事件对体验质量的影响最为严重。本文对这一看法提出了质疑，并通过将卡顿事件与多次质量切换以及高幅度质量切换进行比较，重新审视了这一假设。研究开展了两项主观研究。在第一项主观研究中，参与者获得了金钱激励；而第二项主观研究则由志愿者参与完成。统计分析表明，卡顿事件并不会导致体验质量最严重的下降。这些发现表明，有必要重新评估体验质量研究中卡顿事件的影响。因此，这些发现可用于进一步研究，并改进自适应码率算法中的现有自适应策略。**

索引词—**HTTP自适应流媒体、体验质量、主观评估、众包**

一、引言

自 Move Networks 首次发布以来，HTTP自适应流媒体便一直在颠覆行业。许多研究人员正致力于优化使HTTP自适应流媒体广受欢迎的自适应逻辑。HTTP自适应流媒体的发展很大程度上得益于体验质量（QoE）研究，这类研究旨在改善消费者的体验 [1]–[5]。这些研究旨在确定对用户体验产生积极和消极影响的参数。当播放器缓冲区为空、播放器无法继续播放时，就会发生卡顿事件，通常会以旋转图标提示。质量切换是播放器的一项功能，可通过更改媒体表示，以较低或较高的媒体质量继续播放。结合本文的重点，需要指出以下三项主要发现：

- 通常认为，卡顿事件对体验质量的影响最为严重 [6], [7]
- 多次画质切换会对体验质量产生负面影响 [1], [8]
- 高幅度画质切换会降低体验质量 [1], [3]

然而，目前尚无研究比较多次画质切换、高幅度画质切换和卡顿事件对体验质量的负面影响。本文通过评估和分析，进一步探讨了这些方面。

引言之后，本文分为以下几节：第二节概述本研究的背景。第三节重点介绍研究方法和研究设置。第四节分析并呈现本文的结果与发现。最后，第五节总结全文，并探讨未来可能的研究方向。

II. 背景概述

*A. HTTP*自适应流媒体

HTTP自适应流媒体使用HTTP协议通过网络传输数据。使用该协议的一大优势是可以利用围绕它构建的整个架构。这包括重定向互联网流量的代理服务器、提升性能的缓存，以及将内容存储在更靠近最终用户位置的内容分发网络（CDN） [2]。

动态图像专家组历时两年制定了基于HTTP的动态自适应流媒体（MPEG-DASH）。2011年，在提交了多个草案后，该标准最终作为国际标准（ISO/IEC 23009-1）发布；自发布以来，它已多次修订。

B. 体验质量

在“体验质量”（QoE）这一术语提出之前，许多研究都是在“服务质量”（QoS）的框架下开展的。然而，较高的服务质量评分并不一定意味着用户满意。因此，研究方向有必要从可量化的网络转向以用户为中心的主观方法 [9]。

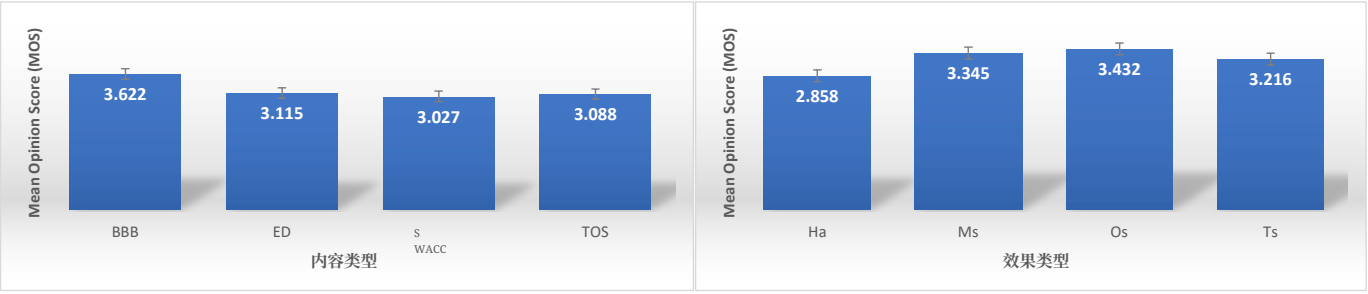


图 1：内容类型之间（左）和效果类型之间（右）的主观研究结果及标准误差线

多媒体系统与服务中的体验质量（Qualinet）对体验质量的定义如下：

“体验质量（QoE）是用户对某项应用程序或服务感到愉悦或恼怒的程度。它取决于该应用程序或服务是否满足用户对其效用和/或愉悦感的期望，同时也受到用户个性和当前状态的影响。” [10]

III. 方法论与研究设置

A. 众包方法

决定通过众包开展本研究，主要基于两个原因：通常所需资源较少，以及当时仍在持续的 COVID-19疫情。在实验室环境中选择、邀请并培训参与者，需要投入大量资源，例如时间、实验室空间和设备。通过众包，可以从工作者群体中快速招募参与者。此外，参与者可以使用自己的设备，在方便的时间开展研究，这也降低了参与者和研究人员的资源负担。此外，在开展本研究时，COVID-19疫情仍在持续。因此，邀请人们进入封闭的实验室环境并不负责任。由于众包平台在体验质量研究中的应用仍然有限，下面将简要说明这种方法。众包通常用于将任务外包给由全球各地、技能和背景各异的工作者组成的群体。其优势多种多样，例如测试对象更加多元、周转快速高效，并且如前所述，相较于实验室测试所需资源更少，包括时间、资金和技术资源。众包平台和任务有多种类型。无论平台类型如何，众包平台主要负责收集和分发任务。本研究中使用的序列评分等任务属于所谓的微任务，只需几分钟即可完成。为确保吸引合适的用户群体参与任务，大多数平台还可以根据不同参数进一步筛选，例如系统设备或特定技能。虽然众包有许多优势，但也给研究人员带来了各种挑战。一个常见问题是，由于说明不清晰且缺少测试主持人，任务可能会被误解。因此，评估说明应尽可能清晰详尽。众包中还需要考虑激励的影响。有些参与者可能会作弊，或不正确地完成任务，以便更快获得激励。为防止并筛除此类结果，Hossfeld 等人在有关众包最佳实践的论文中提出了多种技巧 [11]。其中一种是加入“金标准数据”，即设置答案已知的问题。这种方法可以核实参与者是否留意了初始说明，以及是否按照要求如实回答研究问题。关于如何实施金标准数据，已有多项研究。有些研究建议立即揭示答案是否正确 [12]，另一些则在后评估流程中使用这种方法 [13], [14]。此外，研究还表明，适当的任务设计也有助于防作弊。一项研究发现，如果作弊所需的时间与规范完成研究所需的时间相同，参与者就不太可能作弊 [15]。最佳实践还建议考虑技术实现，例如防止出现不必要的画质损伤。建议采取措施，在播放前缓存序列，以确保带宽可用性较低的参与者不会遇到任何意外损伤 [11]。此外，还建议考虑 Web服务器的容量，因为众包平台会吸引成千上万、甚至数百万名工作者，因此可能会有数百人同时访问研究。最后，在实现方面，建议使用 JavaScript 等标准脚本语言，以确保系统和设备兼容性，因为某些浏览器可能不支持不常见的脚本语言 [11]。

本研究使用了亚马逊土耳其机器人（MTurk）平台。MTurk 的具体功能将在第三节-F 中进一步说明。

B. 数据集选择

研究所用的测试序列选自四种不同的原始素材（表 I）。第一种是 Big Buck Bunny，由 Blender基金会制作的一部动画电影，以动物为主角，展现了适度的运动。此外，这部短片包含大面积的均匀背景和明亮的场景。第二种是 Elephants

表 I：测试序列

Name	来源	时间戳	顺序
Big Buck Bunny	Blender基金会	01:55 - 02:25	1
Big Buck Bunny	Blender基金会	05:45 - 06:15	2
Elephants Dream	Blender基金会	04:00 - 04:30	3
The Swiss Account	《Louder Than 11》	41:40 - 42:10	4
Tears of Steel	Blender基金会	07:25 - 07:55	5

Big Buck Bunny，取自 [16]Elephants Dream，取自 [17]The Swiss Account，取自 [18]Tears of Steel，取自 [19]

Dream，同样属于 Blender 的作品，讲述一男一女行走在一个奇异世界中的故事，展现了这个世界日益加深的疯狂。影片节奏缓慢，场景主要设定在黑暗环境中。第三个测试序列选自 The Swiss Account，这是一部关于抱石攀岩的自制纪录片，包含许多特写和细节镜头。最后，测试序列还选自 Tears of Steel，这是一部动作场面密集、包含各种特效的电影。影片中既有真人演员，也有动画机器。选择多样化的内容是为了减轻内容类型可能造成的影响：如果参与者普遍喜欢或不喜欢某一种特定类型的内容，这可能会影响评分。播放不同的序列也能减缓反复观看同一内容所导致的兴趣下降。每个准备好的序列长度均为 30 秒，因此只是原始内容的一个片段。主观研究中播放各序列的具体时间戳和顺序汇总于表 I。所有选定的视频均可依据知识共享许可协议获取。

C. 序列准备

遵循 Hossfeld等人提出的最佳实践框架，序列已提前准备好 [11]。在选定上一节介绍的原始内容后，通过 *ffprobe* 检查源内容。随后，使用*FFmpeg* 按照以下命令对其进行编码：

```
$ ffmpeg -i inputfile.mp4 \-map 0:v -map 0:v -map 0:v -map 0:a:0 \-c:aaac -c:vlibx264 -x264opts'keyint=24:min-keyint=24:no-scenecut' \-dashsegmenttypemp4 -segduration 5 -usetimeline 0- 1 - 24 - 24 \usetemplatekeyintminginitsegname \$Bandwidth\$/init.m4s-media segname \$Bandwidth\$/seg-\$Number%01d\$.m4s \-b:v:0 4300k -s:v:0 1920x1080 \-b:v:1 1050k -s:v:1 640x480 \-b:v:2 235k -s:v:2 320x240 \-b:a:3 128k -ar 48000 -ac 2 \-adaptationsets"id=0,streams=vid=1,streams=a" -fdash -r 24 manifest.mpd
```

最初，原始文件的路径作为输入，并映射了四条流。前三条代表视频流，用于实现比特率切换；最后一条代表音频流。音频流采用高级音频编码（AAC）进行编码，该编码已在术语部分（第?? 小节，第??节）中简要介绍。视频流使用*Libx264*编码器进行编码。该编码器与平台无关，可用于 H.264/MPEG-4 AVC 视频格式。此外，还自定义了设置，将图像组（GOP）的最大尺寸设为 24，并禁止场景切换。禁止场景切换可确保编码器不会使用自动 I 帧检测算法，在编码过程中添加额外的 I 帧。由于 I 帧的开销较大，因此建议在对流进行编码时限制插入的 I 帧数量 [20]。选项 "dash-segment-type mp4" 将文件格式设为 ISOBMFF，即 ISO/IEC 媒体分段标准格式 [21], [22]。此外，分段时长设为 5，这意味着算法每 5 秒可以切换到不同的画质。除此之外，启用了分段模板，并禁用了分段时间线。这些设置可确保先前设定的分段时长被视为实际时长，而非最短时长 [23]。此外，还设置了*dash* 中的两个模板变量。第 6 行指定初始化分段的命名规则，第 7 行则描述后续媒体分段的命名规则。媒体分段按降序命名。接下来的四行定义了所设流的比特率级别。流 0、1 和 2 代表视频流，并与相应的比特率和帧尺寸相对应。视频采用三种不同的分辨率进行编码：1080p、720p 和 240p。本研究采用的比特率阶梯基于 Netflix 的一项标准 [24]。流 3 对应音频，说明了其比特率以及另外两个参数：以赫兹为单位的音频采样率和音频声道数。每条流的具体规格，如比特率、分辨率、编解码器和片段长度，见表 II。最后两行侧重于 DASH 特定设置。首先，将映射后的流划分为两个自适应集：音频和视频。随后调用 DASH复用器，创建各个媒体分段以及*dash* 播放所需的清单文件 [23]。

表 I：码率编码

流	码率	分辨率	Type	编解码器	长度
流 0	4300kbps	1920x1080	视频	x264	5s
流1	1050kbps	854x480	视频	x264	5s
流2	235kbps	426x240	视频	x264	5s
流3	128千比特/秒	48000赫兹	音频	aac	5s

编码后，还使用了 *FFmpeg* 来处理选定的视频和音频片段，以创建测试序列。首先，必须按如下方式拼接每个测试序列中各质量级别所需的片段：

```
$ cat init.m4s seg-1.m4s seg-2.m4s >>firstVideoPart.m4s
```

使用 cat命令读取以下文件的数据，并按降序写入指定的输出文件。如果输出文件不存在，则会创建该文件。需要拼接的片段数量

取决于各个片段的长度，以及测试序列的长度和设计。在本研究中，测试序列设为 30 秒，编码时的片段长度设为 5 秒。因此，所有用于展示测试序列中段卡顿事件或切换的测试序列，都需要拼接四个片段。初始化分段后接三个片段，每个片段代表 5 秒内容，因此每次拼接的总长度为 15 秒。对于呈现不同时间事件模式的多次质量切换测试序列，每个拼接文件只需三个片段。同样，初始化分段后接两个片段，生成的输出内容时长为 10 秒。拼接后，必须使用以下 *.mp4*文件格式和*FFmpeg* 命令转换输出文件：

```
$ffmpeg –ifirstVideoPart.m4s –c copy fi
rstVideoPart.mp4
```

首先指定输入文件的路径，然后按照所述输出格式复制其数据；本例中，输出格式为*.mp4*。*-c copy* 选项可防止 *FFmpeg* 重新编码输入数据 [23]。

对于构建测试序列所需的每个独立片段，都必须重复拼接并转换为*.mp4* 的过程。高幅度质量切换和两个卡顿事件均由两个独立片段构成，每个片段代表 15 秒的内容。由于多次质量切换展示了三个不同的质量级别，因此由三个独立片段构成，每个片段代表 10 秒的内容。此外，还必须以相同的方式拼接和转换音频片段。完成此过程后，使用*FFmpeg* 合并各个视频和音频部分：

```
$FFmpeg –y –ifirstVideoPart.mp4 –ifirstAu
dioPart.mp4 –c:v copy –c:a copy firstPart.
mp4
```

首先，使用 y 标志命令授予覆盖输出文件的全局权限。然后定义并映射两个输入，第一个代表视频，第二个代表音频源。与之前一样，使用 copy 命令确保输入不会被重新编码。最后，定义输出文件名和格式 [23]。这个过程也必须重复进行，直到将设计测试序列所需的所有独立音频和视频片段合并起来。为了模拟卡顿事件，还需要额外的缓冲序列。为了营造逼真的场景，*FFmpeg* 被用于剪切卡顿前一个 GOP 的最后一帧，再将其插入作为缓冲转圈视频的背景。通过以下两个命令即可实现：

```
$ffmpeg –sseof –3 –ifirstStallPart.mp4 –
update 1 –q:v 1 lastFrame.png
```

```
$ffmpeg –y –framerate 24 –loop 1 –i lastFra
me.png –i loading Wheel.mp4 –filter_complex’[1]
format=argb,colorchannelmixer=aa=0.5[ol];
[0][ol]overlay’ –t 1 finalBufferEffect.mp4
```

第一条命令会创建最后一帧的屏幕捕获。sseof 选项指示 *FFmpeg* 根据相对于输入文件末尾的位置进行查找，并输出该位置指定的所有帧；在本例中，该位置为三秒。为确保仅保存最后一帧，update 选项从默认值 0 更改为 1。此设置会持续覆盖指定输出文件中的数据，最终只保留最后一帧。编码质量设置为 1，表示最高质量级别。捕获的帧将以 *.png*文件格式保存 [23]。第二条命令会将之前捕获的帧插入缓冲转圈视频中。输出视频的帧率设置为每秒 24 帧，并指定图像文件在整个输出文件时长内循环播放。此外，还定义了用于创建复杂滤镜图的选项。首先，将称为索引 1 的第二个输入的颜色模型设置为 Alpha 红绿蓝（ARGB）。然后将其 Alpha 通道值设置为 0.5，使输入源呈半透明状态 [**?**]。接着，将标记为索引 0 的第一个输入设置为主输入，并在其上叠加第二个输入。随后将输出时长设置为一秒，并定义输出名称和格式 [23]。为所有内容类型创建缓冲序列后，使用 *FFmpeg*将之前准备的所有部分拼接起来，以创建最终的测试序列。因此，使用了以下命令：

```
$ffmpeg –y –f concat –safe 0 –i list.txt –c
copy finalTestSequence.mp4
```

首先，定义了全局覆盖权限。f 标志选项会强制输出文件采用拼接序列的文件格式。将 safe 选项设为 0，以禁用安全模式，从而允许输入所有文件。输入项指定了一个.*txt* 文件的路径，该文件包含测试序列所需文件的相对路径，并按拼接顺序排列。下面以两秒卡顿测试序列为例，给出所用.*txt* 文件的结构：

```
文件“卡顿事件/第一部分.mp4”文件“卡顿事件/最终缓冲效
应.mp4”文件“卡顿事件/最终缓冲效应.mp4”文件“卡顿事
件/第二部分.mp4”
```

最后，输入数据未经进一步编码，直接复制到指定的输出.*mp4* 格式。对于本研究中定义的每个测试序列，都必须重复这一拼接过程 [23]。

D. 效应设计

为回答研究问题，并衡量卡顿事件、多次质量切换和高幅度质量切换对体验质量的不同影响，提出以下情形：1 秒卡顿事件（图 2(a)）、2 秒卡顿事件（图 2(b)）、从 1080p 降至 240p 的高幅度质量切换（图 2(c)），以及从 1080p 降至 240p 再升至 720p 的多次质量切换（图 2(d)）。引入 1 秒卡顿的决定基于 De Pessemier 等人的一项研究，该研究量化了卡顿事件的影响[25]。在这项研究中，单次卡顿事件的中位数

约为一秒。选择 2 秒卡顿事件，是为了分析 1 秒卡顿与 2 秒卡顿之间的影响差异。高幅度质量切换和多次质量切换所采用的质量级别，则基于一项特别研究不同切换模式对体验质量影响的研究 [26]。研究表明，与其他事件相比，这两种事件都会导致更差的体验质量。为避免首因效应或近因效应等因素造成额外影响（这些效应认为，视频的开头或结尾几秒对观众体验质量的影响更大），卡顿事件被安排在每个测试序列的中间位置 [27]。

E. 图形用户界面

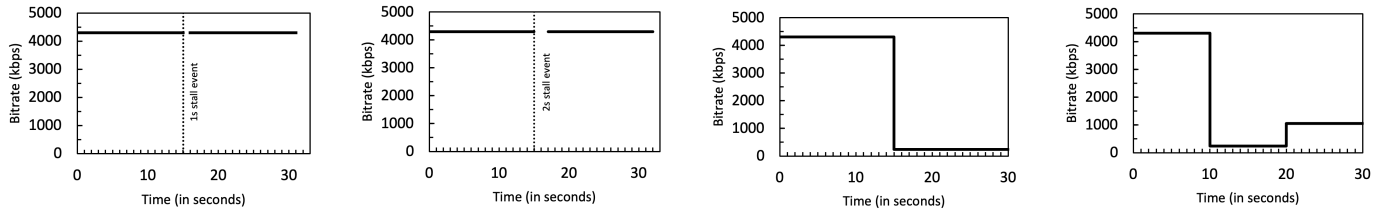
为了向参与者展示研究内容，开发了一个基于网站的图形用户界面（GUI）。该网站基于 HTML、CSS 和 JavaScript 构建。研究使用了 IndexedDB，这是一种可在客户端预缓存数据的低级应用程序接口，以防止出现不必要的损 impairments [28]。参与者打开网站后，测试序列的缓存过程会立即开始，并在后台持续进行，不会造成任何干扰。这样做是为了减少参与者在研究后续阶段可能遇到的等待时间。在后端，使用 Node.js Express 服务器从托管服务器请求准备好的测试序列，并处理与数据库的连接。测试序列存储在 Amazon S3 中，并从那里请求至网站。部署时选择了 Amazon 的 EC2 实例。根据最佳实践框架，研究人员考虑了实例的规模。最终选择了一种小型实例类型，预计足以满足本研究的需求。打开网站后，参与者会看到一段介绍性文字，用于说明研究流程，同时还会看到一个要求填写 MTurk ID 的输入字段。提交后，网站会显示一个示例视频序列，以帮助参与者熟悉播放器设计。使用 HTML 视频标签实现的视频播放器被限制在观看区域内，并停用了所有其他自定义选项，例如全屏、开始/停止和音频选项。此外，视频播放器的宽度固定为 1280px。采取这一措施是为了确保每位参与者从技术角度获得相同的体验。每个测试序列上都会叠加一个定制的覆盖层，以便参与者在必要时暂停。测试序列开始播放后，视频播放器会移至屏幕中央，背景则按照 ITU-T 标准 P. 910 [29]中的建议变为灰色。测试序列结束后，背景会变为白色，随后问卷才会出现。实施这一功能是为了确保参与者不能在未观看测试序列的情况下简单地点击完成整个研究。正如 [15] 中所建议的，这一措施还可以确保正常完成研究所需的时间与作弊尝试所需的时间相同。每份问卷都包含一道与内容相关的问题，该问题被作为金标准数据，同时还收集五点评分质量量表的评分。

由于本研究的主要目标是比较所选效果类型产生的质量，因此采用了 ACR 方法。测试序列按以下顺序展示：Big Buck Bunny、Elephants Dream、The Swiss Account 和 Tears of Steel（图 3）。虽然参与者之间的内容顺序保持不变，但效果由使用 JavaScript 实现的算法随机选择。该算法会改变效果与内容类型的组合，同时确保每位参与者都会观看每种效果一次。完成后，结果会发送至 Express 服务器，并存储在 MongoDB 数据库中。存储的数据包括：MTurk ID、提交时间戳、问卷答案，以及为每位用户自动生成的 UUID。UUID 还会被发送回网站并显示给参与者。手机和平板设备被排除在研究之外。如果参与者尝试通过这些设备访问网站，页面会显示错误消息，建议其切换至台式机或笔记本电脑。同样，除 Google Chrome 之外的其他浏览器也被排除在研究之外，因为试验表明，这些浏览器不支持在经过操纵的序列中进行手动比特率切换。在这种情况下，页面也会显示类似的错误消息。

F. 亚马逊土耳其机器人

亚马逊土耳其机器人（MTurk）是一个众包市场，使公司和个人能够在线外包任务 [30]。该平台允许无限创建活动并自定义设置。每个活动都必须包含简短的描述文本、设定的时间范围以及分配的报酬。其他可自定义的内容包括地理位置和最低用户评分。为测试功能和用户可靠性，我们创建了一项包含 5 名测试对象的试运行活动。时间范围设为 15 分钟，报酬设为 0,50 $。要获得报酬，用户必须在研究界面中输入其 MTurk ID，并将完成后收到的 UUID 粘贴回 MTurk 平台。此举旨在确保用户确实完成了研究。同时，这也使我们能够交叉核对数据库与 MTurk 平台的信息，以筛除重复参与。初步测试确认了活动功能正常。尽管界面和 MTurk 设置均具备相应功能，但提供的五份数据集没有一份能给出与金标准数据一致的准确答案。为进一步测试，我们启动了第二项包含 10 名参与者的活动。除沿用此前的设置外，该活动仅面向接受率达到 95% 及以上的用户开放，且时限为 10 分钟。初步分析显示，第二项活动似乎取得了成功，因为金标准数据得到了正确回答。因此，我们推测第二项活动的参与者认真观看了内容。于是，我们启动了第三项也是最后一项活动，新增 60 名参与者，并沿用之前的设置。

在对 MTurk 结果进行深入分析时，发现了以下问题：首先，所有效果都获得了



(a) 15秒时发生1秒卡顿事件　(b) 15秒时发生2秒卡顿事件　(c) 高幅度质量切换　(d) 多次质量切换

图2：1秒和2秒卡顿事件、高幅度质量切换（1080p-240p）以及多次画质切换（1080p-240p-480p）的表示

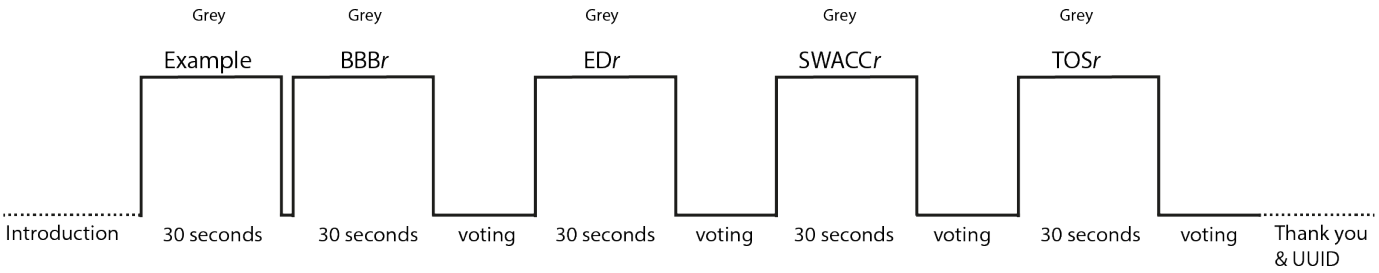


图3：主观研究中ACR刺激呈现的图示——BBB*r*，Big Buck Bunny测试序列的随机效应*r* - ED*r*，Elephants Dream测试序列的随机效应*r* - SWACC*r*，The Swiss Account测试序列的随机效应*r* - TOS*r*，Tears of Steel测试序列的随机效应*r*

相对较高的评分。其次，许多情况下，参与者对所有效果类型给出了相同的评分，这表明他们认为不同效果类型之间没有差异。基于这些发现，我们作出以下假设：5点评分量表未能提供足够多样的选项，使参与者能够以有意义的差异来评判这些效果。一些参与者只想尽快完成问卷以获得激励。因此，他们没有花时间认真观看和评估测试序列，因而未能注意到效果类型之间的差异。

G. 后续研究

为回应上述观察结果，我们设计了一项后续研究。为考虑先前的发现和假设，我们做出了三项重大调整。首先，将5点评分量表改为推荐的9点评分量表，以提高参与者评分的区分能力。其次，为回应不同效果类型之间评分相同的情况，我们增加了四个测试序列。因此，每位参与者将对8个序列进行评分，并体验每种效果类型两次。这项调整还引入了一项额外的可靠性测量：对于在同一种效果类型下评分差异显著的参与者，可以进一步调查。新增序列取自相同的四部电影，以避免产生其他潜在的内容影响。最后一项调整涉及参与者。后续研究中的参与者没有获得报酬，而是从志愿者群体中招募，以减少因激励而导致结果不可靠的可能性。为了取代 MTurk 的时间范围设置，我们又引入了两个变量，并将其保存在数据库中：观看第一个测试序列时的时间戳，以及提交时的时间戳。相应地调整了介绍页面上的评分量表，以及每个测试序列下方的量表，见图5。

IV. 统计分析与结果

以下章节将总结两项研究的统计结果：首先是获得激励的参与者，其次是志愿者。统计分析采用单因素方差分析，以验证不同效果组之间是否存在显著差异。针对效果类型内部的分析，则采用双样本t检验。所有统计检验的显著性水平均设为 0.05。我们采用了两种事后检验来验证显著差异：其一是使用校正后显著性水平的 Bonferetti检验，其二是 Tukey B检验。除效果类型之间的显著差异外，结果还检验了不同内容类型之间的差异。为简洁起见，以下章节中的图表将使用以下缩写来表示各种效果：1秒卡顿记为*Os*，2秒卡顿记为*Ts*，多次质量切换记为*Ms*，高幅度质量切换记为*Ha*。

A. MTurk 评估

总共有 72 条记录提交到了数据库。可以推测，另外两名参与者完成了研究，但未能将 UUID 输入回 MTurk。因此，MTurk 未将他们计为参与者，但他们仍将结果提交到了数据库。基于此，这两份数据集以及另外 10 份数据集被排除

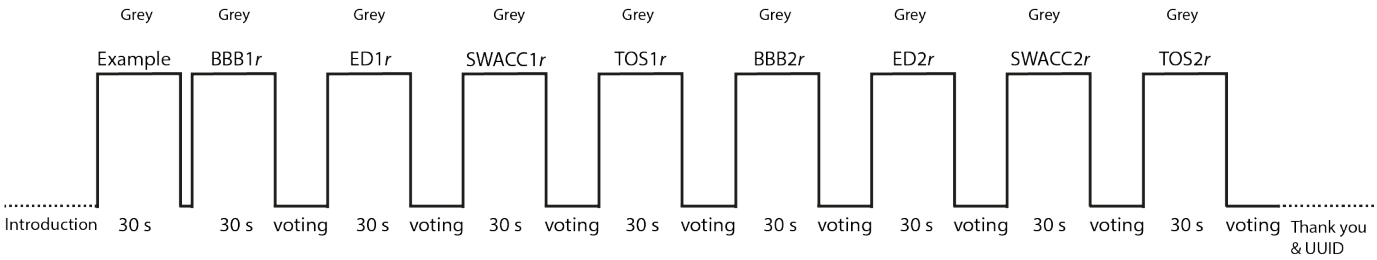


图4：后续研究中 ACR刺激呈现的图示——BBB1r & BBB2r，随机效应下《Big Buck Bunny》的第一和第二个测试序列r - ED1r &ED2r，随机效应下《Elephants Dream》的第一和第二个测试序列r - SWACC1r & SWACC2r，随机效应下《The Swiss Account》的第一和第二个测试序列r - TOS1r& TOS2r，随机效应下《Tears of Steel》的第一和第二个测试序列r



图5：测试序列播放后显示问卷的图形用户界面

在分析中，原因是金标准数据有误或重复参与。随后使用单因素方差分析检验其余 60 名参与者的评分是否具有显著性。评估结果如表 III 所示，各效果组之间仅存在较弱的显著性。因此进行了两项事后检验，将所有效果相互比较（表 IV）。根据 Bonferroni方法，计算得出的校正后 alpha 值为 $\alpha = 0,$ 008333，这表明各效果组之间不存在显著差异。Tukey B事后检验也得出了相同结果，其 Q 值为 $Q = 3,$ 66，所得 T 值为 $T = 0,$ 41583。

B. 后续研究评估

在后续研究中，共收集了43组数据。根据 ITU-T P.910建议，样本量超过40时，结果差异很小 [29]。因此，认为这43组数据足以用于统计分析。后续研究的参与者均没有视频或图像评分的既往经验。数据集分析显示，所有参与者都正确回答了金标准数据。不过，有6名参与者被排除在外，因为他们完成研究所用的时间与平均值相差两个标准差，表明他们没有认真关注测试。

针对每种效果类型均进行了双样本t检验，以检查该效果的首次评分与第二次评分是否存在显著差异。结果未发现显著差异。根据所得的平均主观评分，各效果按降序排列如下：1秒卡顿、多次质量切换、2秒卡顿和高幅度质量切换（图1）。根据以下公式，将9点评分量表的结果换算为5点评分平均主观评分，其中 *NP*表示9点评分量表上的评分值：

$$MOS = \frac{NP}{2} + 0,5$$

与前一项研究相同，我们采用单因素方差分析检验不同效果类型之间的方差，结果显示存在显著差异（表 III）。对于 Bonferroni 事后检验，校正后的 alpha 值仍为 $\alpha = 0,$008333。在 1秒卡顿与高幅度质量切换，以及多次质量切换与高幅度质量切换的比较中，$P <$校正后 $\alpha = 0,$00833，表明差异显著。Tukey B事后检验以 $T >$ 绝对均值差 （表 IV）确认了这些结果。因此可以认为，高幅度质量切换与 1秒卡顿事件之间，以及高幅度质量切换与多次质量切换事件之间存在显著差异。在这两种情况下，高幅度质量切换的表现都显著更差。此外，我们还检验了不同内容类型之间是否存在显著差异。单因素方差分析结果显示差异显著（表 III）。Tukey B事后检验的结果（$T = 0,$764）证实，Big Buck Bunny 与其他影片序列之间存在显著差异。在所有效果类型中，Big Buck Bunny 片段的平均主观评分最高（图1）。

V. 结论与未来工作

本文研究了与多次或高幅度画质切换相比，卡顿事件是否对降低观众的体验质量影响最大。结果证实，与多次及高幅度画质切换相比，卡顿事件对体验质量的影响并非最差

表 III：MTurk 和实验室参与者主观结果中不同效果和内容类型的方差分析（单因素方差分析，组间）

评估	df [1]	均方	F [2]	*P* [3]	F [4]
效果（MTurk）	3	2.070	2.673	0.048	2.642
内容（MTurk）	3	1.348	1.720	0.163	2.642
效果（实验室）	3	4.721	5.758	0.00077	2.635
目录（实验室）	3	22.382	6.900	0.00016	2.635

[1] 自由度，即总类别数减 1 2 F 值，即两个总体之间方差的显著性 [3] P 值，若 $P < 0.05$（所选 α 水平）则显著 [4] 临界 F 值，若 $F > Fcrit$ 则显著

表 IV：MTurk 和实验室参与者对不同效果类型的事后检验（Bonferroni，校正后 $\alpha = 0{,}0083$）和 Tukey B（$T = 0{,}415$）

环境	I [1]	J [1]	P [2]	绝对均值差 [3]
MTurk	Ha	Ms	0.649	0.083
MTurk	Ha	Os	0.066	0.3
MTurk	Ha	Ts	0.018	0.4
MTurk	Ms	Os	0.161	0.216
MTurk	Ms	Ts	0.049	0.316
MTurk	Os	Ts	0.458	0.1
实验室	Ha	Ms	0.00099	0.486
实验室	Ha	Os	0.00012	0.574
实验室	Ha	Ts	0.0253	0.358
实验室	Ms	Os	0.526	0.087
实验室	Ms	Ts	0.399	0.128
实验室	Os	Ts	0.159	0.216

[1]I 和 J，不同的效果类型 [2]双尾 t 检验[3] $|I - J|$

。结果表明，自适应比特率算法应考虑引入短暂卡顿事件，而非高幅度画质切换。不过，建议针对不同的卡顿时长、切换模式和画质切换幅度开展进一步研究。

致谢

谨此感谢奥地利联邦数字与经济事务部、国家研究、技术与发展基金会以及克里斯蒂安·多普勒研究协会提供的资金支持。https://athena.itec.aau.at/。

参考文献

[1] P. Ni、R. Eg、A. Eichhorn、C. Griwodz 和 P. Halvorsen，《面向手持设备的自适应视频流中的闪烁效应》，载于第 *19* 届 *ACM* 国际多媒体会议论文集，2011年，第 463–472 页。[2] C. M¨uller、S. Lederer 和 C. Timmerer，《车载环境中基于 HTTP 的动态自适应流媒体评估》，载于第 *4* 届手机视频研讨会论文集，2012 年，第 37–42 页。[3] D. Z. Rodr´ıguez、Z. Wang、R. L. Rosa 和 G. Bressan，《视频质量等级切换对基于 HTTP 的动态自适应流媒体中用户体验质量的影响》，第 2014 卷，第 1 期，2014 年，第 1–15 页。[4] B. Taraghi、M. Nguyen、H. Amirpour 和 C. Timmerer，《Intense：深入研究卡顿事件和画质切换及其对 HTTP 自适应流媒体体验质量的影响》，*IEEE Access*，第 9 卷，第 118 087–118 098 页，2021 年。[5] B. Taraghi、A. Bentaleb、C. Timmerer、R. Zimmermann 和 H. Hellwagner，《理解启发式 HTTP 自适应比特率算法的体验质量》，载于数字音频与视频的网络和操作系统支持（*NOSSDAV*）研讨会，2021 年。 [在线]。网址：https://doi.org/10.1145/3458306.3458875

[6] M. Seufert、P. Casas、N. Wehner、L. Gang 和 K. Li，《基于流的机器学习：加密视频流量的实时体验质量分析》，载于《2019 年第 22 届云、互联网与网络创新会议及研讨会（ICIN）》，2019 年，第 76–81 页。[7] A. Bokani、M. Hassan、S. Kanhere 和 X. Zhu，《利用马尔可夫决策过程优化车载环境中的基于 HTTP 的自适应流媒体》，《IEEE 多媒体汇刊》，第 17 卷，第 12 期，第 2297–2309 页，2015 年。[8] M. Zink 、 J. Schmitt 和 R. Steinmetz，《可扩展自适应流中的分层编码视频》，《IEEE 多媒体汇刊》，第 7 卷，第 1 期，第 75–84 页，2005 年。[9] S. M¨oller 和 A. Raake，《体验质量：高级概念、应用与方法》。Springer，2014 年。[10] K. Brunnstr¨om、S. A. Beker、K. De Moor、A. Dooms、S. Egger、M.-N. Garcia、T. Hossfeld、S. Jumisko-Pyykk¨o、C. Keimel、M.-C. Larabi 等，《Qualinet 体验质量定义白皮书》，2013 年。[11] T. Hossfeld、C. Keimel、M. Hirth、B. Gardlo、J. Habigt、K. Diepold 和 P. Tran-Gia，《体验质量众包测试的最佳实践：基于众包的体验质量评估》，《IEEE 多媒体汇刊》，第 16 卷，第 2 期，第 541–558 页，2013 年。[12] J. Le、A. Edmonds、V. Hester 和 L. Biewald，《确保众包搜索相关性评估的质量：训练问题分布的影响》，载于 SIGIR 2010 年搜索评估众包研讨会，第 2126 卷，2010 年，第 22–32 页。[13] P. G. Ipeirotis、F. Provost 和 J. Wang，《亚马逊土耳其机器人上的质量管理》，载于 ACM SIGKDD 人类计算研讨会论文集，2010 年，第 64–67 页。[14] R. Snow、B. O'connor、D. Jurafsky 和 A. Y. Ng，《廉价又快速——但质量好吗？自然语言任务中非专家标注的评估》，载于 2008 年自然语言处理实证方法会议论文集，2008 年，第 254–263 页。[15] A. Kittur、E. H. Chi 和 B. Suh，《利用 Mechanical Turk 开展众包用户研究》，载于 SIGCHI 人因计算系统会议论文集，2008 年，第 453–456 页。[16] 《Big Buck Bunny》，http://bbb3d.renderfarming.net/explore.html，最后访问日期：17.03.2022。[17] 《Elephants Dream》，https://orange.blender.org，最后访问日期：17.03.2022。[18] 《The Swiss Account》，https://vimeo.com/26475347，最后访问日期：17.03.2022。[19] 《Tears of Steel》，https://mango.blender.org，最后访问日期：17.03.2022。[20] B. Furht 和 O. Marques，《视频数据库手册：设计与应用》。CRC 出版社，2003 年。[21] 《ISO BMFF 字节流格式》，https://www.w3.org/TR/mse-byte-stream-format-isobmff/，最后访问日期：17.03.2022。[22] 《信息技术——视听对象编码——第 12 部分：ISO 基础媒体文件格式》，ISO/IEC JTC 1/SC 29 音频、图像、多媒体和超媒体信息编码，标准，2020 年 12 月。[23] 《FFmpeg 文档》，https://ffmpeg.org/ffmpeg-all.html，最后访问日期：21.03.2022。[24] A. Aaron、Z. Li、M. Manohara、J. D. Cock 和 D. Ronca，《逐标题编码优化》，https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2，最后访问日期：17.03.2022。[25] T. De Pessemier、K. De Moor、W. Joseph、L. De Marez 和 L. Martens，《量化重新缓冲中断对用户在移动视频观看期间体验质量的影响》，《IEEE 广播汇刊》，第 59 卷，第 1 期，第 47–61 页，2013 年。[26] A. Asan、W. Robitza、I.-h. Mkwawa、L. Sun、E. Ifeachor 和 A. Raake，《视频分辨率变化对自适应视频流体验质量的影响》，载于 2017 年 IEEE 多媒体与博览会国际会议（ICME），2017 年，第 499–504 页。[27] Y. Shen、Y. Liu、Q. Liu 和 D. Yang，《一种基于比特率分布的自适应流体验质量评估方法》，载于 2014 年 IEEE 国际通信会议研讨会（ICC），2014 年，第 551–556 页。[28] 《IndexedDB》，https://www.w3.org/TR/IndexedDB/，最后访问日期：17.03.2022。[29] 《P.910 建议书：多媒体应用的主观视频质量评估方法》，国际电信联盟，标准。[30]《亚马逊土耳其机器人》，https://www.mturk.com/，最后访问日期：17.03.2022。