感谢审稿人和编辑们为本文指出的问题，这些意见能帮助我们提高这篇文章的质量，尤其关于其中关于公式部分提出的问题让我们认识到了在公式解析过程中存在表达上的问题，而关于创新点的疑问则帮助我们梳理并完善了文章关于贡献和创新性的。我们仔细阅读后对本文做出了修改，具体的回复和修改如下，其中原意见用蓝色字体标出，文中修改或者添加的部分用红色字体标出：

**审稿人一：**

感谢审稿人提出的宝贵意见，我们针对每一条意见做出如下答复：

**意见1：**在英文缩写第一次出现时，建议给出其中文名称和英文全称，例如“DQN”等

**答复：**

我们在以下部分添加了英文全称（按照出现顺序排序）：（1）摘要处第一次出现的DQN添加了Deep Q-Learning （2）在引言处第一次出现DASH处添加了“Dynamic Adaptive Streaming over HTTP，基于HTTP传输的动态自适应流”（3）在相关工作处第一次出现“边缘数据缓存”处添加了英文全称“Mobile Edge Data Caching

”（4）在相关工作处第一次出现“潜在语义索引”和“协同过滤”处添加了英文全称“Latent Semantic Indexing”和“Collaborative Filtering”(4)在引言处第一次出现“用户体验质量”和“组合多臂老虎机”处添加了英文全称“Quality of Experience”和“Combinatorial Multi-Armed Bandit”

意见2-4：

（1）将3.2 问题形式化第二段中“传输时延定义为”改为“传输时延定义为” 。

（2）将4.1 状态空间第二段中改为，以与后文一致。

（3）将4.2 动作空间第一段中“动作和缓存状态间存在(2)中所示关系”改为“动作和缓存状态间存在(4)中所示关系”

**答复：**

我们按照要求修改了对应部分的笔误，并且加以检查，确认更正部分没有问题。

意见5：5.1 基于DQN的边缘缓存策略中loss函数的公式不正确

**答复：**

此处应当是均方差公式，在公式输入过程中出现了笔误，我们按照正确的格式修改了第七页的公式（7），如下所示:



意见6：前文考虑了边缘服务器从云服务器下载高比特率版本的视频情况下的总服务时延，其中包括转化时延，而后面设置奖励函数，时，未考虑转化时延，请解释原因或进行修改。

**答复：**

此处我们实际上在奖励函数中考虑到了转化时延，但是此处的表述缺少详细的介绍，因此公式的含义可能出现偏差，我们在4.3部分添加了更详细的解释，对公式中的每一个表达式进行了更详细的介绍。

对应的，在3.2中我们对需要从云服务器获取资源的场景做了修改，之前的表述为下载对应资源高比特率版本，我们更正为了“从云服务下载所请求的视频k的指定版本”，以避免出现认为一定要选用高比特率版本进行转码的误解。

特别的，对于的情况，此时边缘服务器的动作是从云端服务器获取到了用户所请求的内功的指定比特率版本，因此无需再进行额外的转码操作，相应的在公式中也不会出现转化时延。在后面的a=-1的情况中我们对转化时延进行了考虑，这一点在修改后的公式说明里进行了详细的介绍，修改部分如下所示（修改点用红色标出）：

当时，边缘服务器的动作为先从远端服务器下载内容再进行传输，总时延为，而云计算的时延为，则总节约时间为；当时，边缘服务器移除视频的比特率等级内容，也就意味着边缘服务器中没有缓存内容的比特率等级内容，此时若存在视频的比特率等级内容，即当时，则可以通过计算和转码进行转化，此时由边缘服务器提供服务的时延为，即转码时间和边缘服务器传输给用户时间的总和，总节约时间为。对应的，当此时边缘服务器中没有缓存视频的任一比特率等级内容，即时，总节约时间则为0；当时，表明边缘服务器不对视频的比特率等级内容做任何操作，此时总节约时间则与时刻的缓存状态有关，若时刻缓存了内容的比特率等级内容，即，边缘服务器可以直接向用户传输所请求内容，则总节约时间为，否则如果若时刻没有缓存视频的比特率等级，边缘服务器需要从云服务器请求内容的比特率等级内容，此时总节约时间为0，综合两种情况即为。

意见7：深度强化学习算法包括A2C,A3C，DDPG等等，请作者解释为什么使用DQN来解决所提出的问题？DQN的优点是什么？

**答复：**

我们对此的解释如下：首先深度强化学习按照Agent选取动作的策略不同可以分为基于策略的强化学习(Policy-Based RL)和基于价值的强化学习(Value-Based RL)，在基于策略的强化学习中所有可能动作都有一定概率被选中，只是不同动作有不同的概率。而在基于价值的强化学习中则会选用某一时间反馈值最高的动作，因此基于价值的强化学习适合用于离散的动作空间。

审稿人提到的三种强化学习算法的确也是主流的深度强化学习算法（还有PPO等），其中A2C和A3C[1]是基于Actor-Critic架构，其中包括了两个神经网络，其中的Actor使用的是基于策略的强化学习，而DDPG[2]则是直接使用了基于策略的强化学习。考虑到这些使用基于策略的强化学习算法主要针对的问题往往有连续的动作空间，而在本文所解决的问题中，动作空间是离散的，边缘服务器对于缓存内容的操作只存在添加缓存，移除缓存和保持不变三种动作（对应着文中所述的a=1，a=-1和a=0三种情况）。虽然前面三种方法同样也可以改成离散的动作空间，但是会对准确率造成一些影响。

此外，考虑到文中所设计的算法应用的场景是视频传输，对时间延迟的要求非常高，因此选用更为简单的DQN算法在解决问题的基础上更容易部署，时间延迟也更低。所以综合考虑后我们决定使用DQN作为解决文中所提到的问题的算法。我们在文章中第5节同样添加了关于这一部分的描述。添加部分如下：

“深度强化学习算法根据选取动作的策略不同可以分为基于策略的强化学习(Policy-Based RL)和基于价值的强化学习(Value-Based RL)，在基于策略的强化学习中所有可能动作都有一定概率被选中，只是不同动作有不同的概率。而在基于价值的强化学习中则会选用某一时间反馈值最高的动作，因此基于价值的强化学习适合用于离散的动作空间。除了本文选用的DQN，其他的深度强化学习算法还包括A2C、A3C、DDPG等，然而这些算法是基于策略的强化学习，本身针对的问题往往有连续的动作空间，本文提出的动作空间是离散的，因为边缘服务器只会进行有限类型的操作，即添加缓存、清除缓存和保持不变三种动作。尽管前面提到的三种算法可以更改成离散动作空间的算法，但是这会对其准确性造成影响。另外，选用更为简单的DQN算法更容易部署，时间延迟也更低。综合考虑后本文选用了DQN作为解决建模问题的强化学习算法以支持离散的动作空间。”

**审稿人二：**

感谢审稿人提出的细节方面以及创新性方面的疑问，我们对每条意见做出如下答复：

意见1：论文格式需要认真修改，比如中文论文引用格式，字母斜体等等

**答复：**

我们仔细核对了论文中的引用格式，同时对所有用到公式和字母的地方统一更换成了斜体表示，保证了全文公式格式的一致性。

意见2：论文强调全景视频场景，但全文并没有体现全景视频的特色，绪论中所说的视频数据包大这一特点在实际建模中也没有很好体现，码率选择也是较为常见的建模思路。因此，论文建模和方法上并无创新。

**答复：**

关于全景视频的特色，文中提到全景视频传输场景往往存在多比特率的视频资源，这同样也和全景视频的传输数据量大的特点息息相关，正是因为其数据量大，带宽需求也响应的变大了，而用户在异构网络下不同的网络状况决定了他们所能提供的带宽状况是多样化的。针对这种多样化，全景视频传输网络为同一视频提供了不同比特率等级的内容，可以根据用户的状态和需求动态调控。总的来说，本文并不将数据包大这一特点作为建模中的核心特点，而是将全景视频的多比特率内容结构作为其核心特点。

我们的模型创新性的加入了对于多比特率视频内容的考虑，在缓存策略中特别考虑了同一视频的不同比特率等级内容之间的转码问题，我们认为这一点是全景视频传输场景的特色之一。但是考虑到创新性可能没有突出描述的问题，我们对引言部分做出了小的修改以更突出针对全景视频场景做出的创新。目前本文在此方面的创新性在摘要、引言和相关工作的最后部分都有更明确的体现。摘要部分修改如下（修改部分用红色标出）：

“为了解决全景视频服务中云服务器和边缘服务器的联合边缘缓存问题，优化边缘缓存机制以降低用户获取视频资源的时延，提出用DQN(Deep Q-Learning)进行深度强化学习以生成视频资源缓存策略的方法。该方法针对当前全景视频中多比特率版本的情况，不仅仅考虑边缘缓存内容，同时将不同比特率版本间的转码时延考虑在内。首先，以总节约时间为目标将问题建模成马尔可夫决策过程；其次，结合模型中数据离散的特性，使用DQN算法对模型进行训练，在迭代中获取最优缓存策略。仿真结果表明，所提算法具有较高的收敛速度和最优的性能，且在约束条件改变时能主动变换调整边缘缓存策略使得算法性能稳定上升，为全景视频边缘缓存机制提供了切实可行的方案。”

另外意见中提到了“码率选择”，也就是在多比特率资源结构中需要使用到的ABR（Adaptive Bit-rate）算法，ABR算法运行在客户端，通过不同的信息如吞吐量、缓冲区状态以及基于学习的决策做出比特率选择，其目的是控制客户端请求的比特率等级以提高用户体验质量。而本文提出的算法运行在边缘服务器，这与运行在客户端的码率选择算法有本质上的区别，我们的算法目的是优化边缘服务器的缓存决策而非客户端的比特率决策，这一点上也存在较大的区别。

意见3：论文对比算法较为简单，没有和最新的方法进行对比，DQN解决此类问题的思路也没有学术创新

**答复：**

我们调研并实现了[1]中所提出的多更新强化学习算法(Multi Update Reinforcement Learning, MURL)，该算法同样是针对边缘计算场景下的边缘服务卸载问题，使用一种创新的强化学习算法解决。我们在实验部分添加了我们提出的算法于MURL算法的对比实验，并且更新了结果图。结果表明本文所提出的算法性能略优于MURL算法，这是因为本文提出的算法额外考虑到了不同比特率等级资源之间的转码过程，这是MURL算法所没有考虑的。此外，本文的创新性主要体现在建模过程中考虑到了全景视频场景下的多比特率资源结构以及转码过程，DQN是我们选用的解决建模问题的方法，而不是文章的创新点所在。对于新添加的对比，在引言的贡献总结处也有更新：

“我们模拟了变化的网络环境进行了实验，并使用了当今实际场景中部署最多的Popular算法和Random算法进行对比，同时，对比了2021年CCF A类会议ACM MM 中所提的关于该问题的多更新强化学习算法MURL算法。仿真结果表明，DQN算法相较于其他算法有着最高的总节约时间和收敛速度，同时在改变计算能力这一限制条件时，尽管算法性能的提升会遇到瓶颈，但DQN可以主动调整自己的缓存策略使得算法性能稳定上升。”

同时在后面的实验部分，我们也增加了对比试验的文字说明：

“……而所提出的DQN算法由于可以根据约束条件主动变换调整边缘计算策略，如随着计算能力增加，缓存空间逐渐成为约束时，将原某些计算需求小缓存大的服务更换为计算需求大缓存小的服务，从而使得算法性能稳定上升，这一点与MURL算法相同。另外，由于随机算法的随机性，当缓存空间成为瓶颈前，其性能基本随计算能力的增加而线性增加。与MURL相比，DQN算法的性能略高于MURL，这是由于本文提出的DQN算法额外考虑到了不同比特率版本之间的转码过程，在这种特殊场景下，我们提出的DQN算法更贴合于实验设计的全景视频传输模型，而MURL算法由于没有考虑边缘服务器的转码功能，在这种细化场景下的决策不占优势。图6-4情况类似。综合两张图可以看出，单个约束条件对算法性能的影响会逐渐减小，只有同时增加两方面能力才能实现算法性能持续增加。”

意见4：论文结果图中，节省时间的单位有时候为毫秒，有时候为秒，请确认其正确性

**答复：**

感谢审稿人提出的细节问题，我们仔细核对了绘图代码，发现的确存在坐标轴单位输入的笔误，在图6.4中纵坐标轴正确的单位应该是秒(s)而不是毫秒（ms），我们现已将图片更改为正确单位。

**审稿人三：**

感谢审稿人指出的问题，我们对每条意见给出了如下答复：

意见1：若边缘服务器没有缓存任何版本视频，终端设备能否直接从云端下载相应版本视频？若能，此时的服务时延应该定义为？

**答复：**

该问题与审稿人一的意见6中的问题相似，我们在文章的3.2部分对于此问题存在的误解进行了更清晰的解释，即若边缘服务器没有缓存视频的任何版本，则需要先从云服务下载所请求的视频的指定版本，下载时延为，此时总的服务时延为。因此，与从云服务器请求内容相比，从边缘服务器提供服务节省的时间为

。由于此时的假设是边缘服务器没有缓存任何版本视频，因此从云端下载会优先选择相应的版本视频，这一点在文章之前的版本中没有强调，因此存在表达上的问题，我们详细更改了这部分的表达方式，可以参考审稿人一的意见6回复中的更改内容（红色字体标出）

意见2: 公式中部分变量缺乏解释，定义不明确；

意见3：部分函数定义错误，如公式（7）中的loss函数；

**答复：**

对于意见2的内容，我们在意见1中做出了相应的修改，对公式的每一个部分进行了详细的介绍和推导，内容如下所示：

当时，边缘服务器的动作为先从远端服务器下载内容再进行传输，总时延为，而云计算的时延为，则总节约时间为；当时，边缘服务器移除视频的比特率等级内容，也就意味着边缘服务器中没有缓存内容的比特率等级内容，此时若存在视频的比特率等级内容，即当时，则可以通过计算和转码进行转化，此时由边缘服务器提供服务的时延为，即转码时间和边缘服务器传输给用户时间的总和，总节约时间为。对应的，当此时边缘服务器中没有缓存视频的任一比特率等级内容，即时，总节约时间则为0；当时，表明边缘服务器不对视频的比特率等级内容做任何操作，此时总节约时间则与时刻的缓存状态有关，若时刻缓存了内容的比特率等级内容，即，边缘服务器可以直接向用户传输所请求内容，则总节约时间为，否则如果若时刻没有缓存视频的比特率等级，边缘服务器需要从云服务器请求内容的比特率等级内容，此时总节约时间为0，综合两种情况即为。

另外对于意见3，公式的错误在审稿人一的意见5的答复中也进行了修改说明，公式修改如下：



意见15：伪代码中动作是基于UCB算法选取的，而下文相应解释中动作是基于贪婪算法选取的，二者之间是否存在相应的转换？

**答复：**

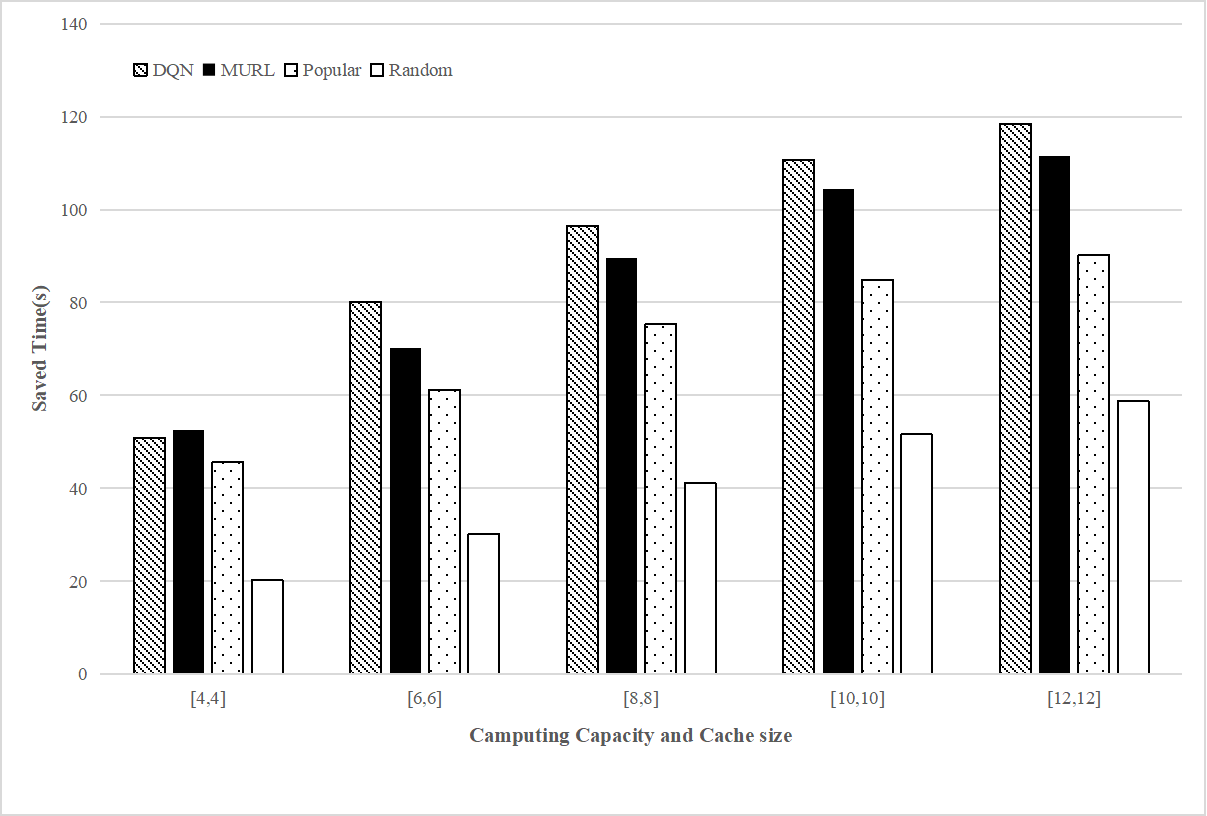
这部分的算法在文章编写过程中进行过修改，此处存在笔误，在后面选取动作时依据应该是选取使得UCB算法结果最大的行为，我们修改了文章5.1节处的描述，具体如下所示：

“之后算法将运行多个循环，在每个循环中选取每个时刻的缓存行为，依据是选取使得最大的行为。之后执行选取的行为并获取回报函数的结果…”

意见16：.本文结论指出，只有增加两方面的能力才能实现算法性能的持续增加，而文中实验仅能证明增加单方面能力无法实现算法性能的持续增加，需要补充额外的实验证明本文观点

**答复：**

我们的确认识到了该部分实验的不足，并且补充了新的实验以证明增加两方面的能力可以实现算法性能的持续增加，其中实验结果可见实验部分的图6.5，如下所示：



此外，我们添加了该部分的文字说明于实验部分（红字标出）：

为了证明同时增加计算能力和缓存能力可以实现算法性能的持续增加，我们分别控制计算能力和缓存能力为【4GHz，4GB】，【6GHz，6GB】，【8GHz，8GB】，【10GHz，10GB】和【12GHz，12GB】共五组情况，同时分别测试四种算法在每种情况下的总节约时间，结果如图6.5中所示，当同时增加两种能力时，四种算法节约时间的增加基本随两种能力线性增加，并且对比图6.3和图6.4可以发现其性能优于同样情况下限制其中一种能力的场景，例如当限制缓存能力为8GB而将边缘服务器的计算能力提升到12GHz的时候，DQN算法节约的时间是106.9秒，而当同时增加计算能力和缓存能力至【12GHz，12GB】时，DQN算法节约的时间是118.5秒。其中我们还发现DQN算法的性能在高性能的情况中表现略优于MURL算法，这进一步证明了在特殊的问题场景下，即考虑不同比特率等级视频的互相转码时，我们提出的算法的性能优于MURL算法。

意见17：文中存在多处笔误，需加以校正

**答复：**感谢审稿人指出的细节问题，我们认真审阅了文章并且对笔误处进行了修改。

参考文献：

1. Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1928-1937.
2. Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
3. Hao H, Xu C, Zhong L, et al. A multi-update deep reinforcement learning algorithm for edge computing service offloading[C]//Proceedings of the 28th ACM International Conference on Multimedia. 2020: 3256-3264.