1.规避计算能力不足及漏报风险

（1）计算能力不足：利用“算力共享+丢帧策略”解决

（2）漏报风险：利用“算力替代+丢帧策略”解决

2.实现模块化的功能处理：分为三个芯片（存储、计算和通讯）

（1）计算芯片：需要增强各个终端节点的计算性能以解决计算量大的问题。除此以外，能够在每一块区域之间，拥有一个具有超乎周围节点计算能力的节点，并且要求这样的节点不仅计算能力强，而且一定能连接电源，保证持续供电。

（2）存储芯片：对于接收的数据，本地进行存储；如果需要别的节点帮助计算，别的节点可以做临时的数据存储，然后处理完成后的一段事件后可以进行删除（不用害怕丢失，因为本地会有存储）；并且如果运行了一段时间以后（比如几个月），每个边缘节点将视频数据进行压缩，分别传输到一个具体的存储节点进行统一存储（这个节点可以不同于一般的边缘节点，但是一定具备强大存储能力，像IDC中做存储数据的那种），对边缘节点进行定期的数据清理。

（3）通讯芯片：能够利用通讯协议，将每个边缘设备看作一个站点，不同站点之间相互告知路径、节点用电状态等信息，当某个节点电力不足时，切换到临近的最优节点帮助计算（可能是最优、次优或者再次优），通过这种方式能起到计算节点的冗余和备份。

3.增强智能边缘计算节点的可靠性：”物理安全“和“数据传输安全”两个方面

通信方面的解决方案：

1.不同站点之间互相通信路由信息，电量信息以经处理状态（相当于对节点对自身信息的结构化）；

2.当前节点需要进行算力共享和算力替代的话，查看路由表，充分考虑最近节点的电量以及其处理能力（如果该节点已经处理了90%的数据，就不要选择该节点进行处理了，所以是对最近节点功能的综合考虑，从中选择最优的），选择合适的节点进行处理；

3.如果利用了路由信息，那么可以起到线路冗余的效果，并且如果最近的两台设备的综合性能一致，那么可以利用负载均衡，将数据发送给这两台设备，并且要保证发送给每台设备的数据具有连续性（因为传输的是视频数据，所以如果具有离散性，就难以捕捉到视频的连续特征，没办法起到时序预测的功能等），两台设备处理完成以后，将数据结果返回给源边缘节点。

两个问题：

1.我想了下，对于本地的视频数据最好还是用丢帧策略，不要用结构化把视频变成数据，因为硕哥他们的时序建模是对视频数据进行处理，否则发送到远端就没办法进行操作了。

2.数据存储的话，要考虑多重备份问题么，但要备份，这数据量也不小