**Real-Time, Accurate, and Consistent Video Semantic Segmentation**

**via Unsupervised Adaptation and Cross-Unit Deployment on Mobile Device**

地址：<https://openaccess.thecvf.com/content/CVPR2022/papers/Park_Real-Time_Accurate_and_Consistent_Video_Semantic_Segmentation_via_Unsupervised_Adaptation_CVPR_2022_paper.pdf>

## 摘要

该演示展示了我们在移动设备上高效、准确和时间一致的视频语义分割方面的创新。我们使用我们的测试时无监督方案 AuxAdapt 来使分割模型能够以在线方式适应给定的视频。更具体地说，我们**利用小型辅助网络来执行权重更新并保持大型的主要分割网络冻结**。与以前的方法（例如帐篷、DVP）相比，**这显着降低了适应的计算成本，同时防止了灾难性遗忘**。通过运行 AuxAdapt，我们可以显着提高视频分割的时间一致性，同时保持准确性。

我们展示了如何在搭载 Snapdragon® 移动平台的智能手机上高效地部署我们的自适应视频分割算法。我们不是简单地在 GPU 上运行整个算法，而是采用跨单元部署策略。将在测试期间冻结的主网络将在高度优化的 AI 加速器单元上执行推理，而将即时更新的小型辅助网络将在 GPU 上运行前向传播和反向传播。这种部署方案最好地利用了智能手机上的可用处理能力，并能够实时运行我们的自适应视频分割算法。我们在补充材料中提供示例视频。

## 介绍

语义分割是视觉场景理解的核心功能，对各种应用非常重要，例如 AR/VR、自动驾驶/ADAS、机器人和移动图像/视频处理。鉴于近年来视频数据的快速增长，对视频内容进行分割的能力变得越来越重要。然而，**直接将基于图像的分割模型应用于视频数据通常会导致时间上不一致（例如闪烁）的输出**，其示例如图 1 (a) 所示。这些工件会显着影响后台编辑等下游应用程序并降低用户体验。

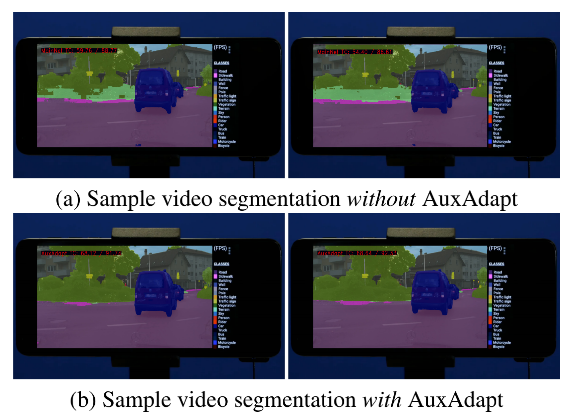


Fig1 我们的视频分割演示的快照。 (a) 不使用 AuxAdapt 的示例视频分割。 (b) 使用 AuxAdapt 进行视频分割示例。使用 AuxAdapt 时，我们看到更高的时间一致性和整体更好的视频分割质量。另一方面，如果没有 AuxAdapt，分割会受到闪烁伪影的影响，例如，在道路/人行道和植被之间的过渡区域。

在文献中，研究人员研究了各种提高时间一致性的方法。一项主要工作在训练和/或测试期间利用光流，因为它捕获跨帧的像素对应关系 [1-5]。然而，它们的性能**受到估计光流质量的限制**，并且它们还需要准确注释的视频训练集，而**这些训练集的收集成本可能很高**。其他一些工作应用测试时间适应来提高模型性能，但它们的**计算成本太高而无法实时运行** [6-8]。

为了应对这些挑战，我们利用我们最近提出的适应方案 AuxAdapt [9] 来有效地执行在线模型更新以增强时间一致性。AuxAdapt 的设计基于以下关键见解**：不一致通常源于网络决策的不确定性**。因此，AuxAdapt **强制分割模型从自己的输出中学习并增强其预测信心**。此外，为了显着降低在线自适应的计算成本，AuxAdapt 采用了一个小型辅助网络（AuxNet）来执行更新，同时保持主分割网络（MainNet）不变。换句话说，**在处理每一帧时，MainNet 被冻结，只有 AuxNet 被更新**，而集成模型通过视频流式传输。**最终的分割由两个网络的聚合输出决定**。通过这样做，我们**避免了昂贵的模型更新和灾难性遗忘，并且可以增强时间一致性，而不需要光流或其他可能不可靠的时间特征**。

在这个演示中，我们通过一种新颖的**跨单元部署策略**在搭载 Snapdragon 的智能手机上部署我们的自适应视频分割算法 AuxAdapt。 虽然一种常见的方法是在手机 GPU 上简单地运行整个算法，但这确实没有充分利用手机上的可用处理能力。尽管与以前的自适应方法相比，AuxAdapt 对计算的要求要低得多，但在 GPU 上运行所有前向和后向操作可能不会导致实时视频分割。因此，我们还利用智能手机上高度优化的 AI 加速器并在其上运行主网的量化版本，因为它的权重不会改变。同时，AuxNet 在 GPU 上运行，**并在模型流过视频时调整其权重**。通过这样做，我们可以实时操作自适应视频分割模型，从而提供准确且时间一致的分割结果。

在评估视频分割的时间一致性时，通常使用估计的光流。但是，光流估计本身可能包含错误，从而导致评估结果不可靠。最近，我们**提出了一种基于感知一致性的新度量**[10]。正如我们在原始论文中所展示的，感知一致性可以更准确地捕捉视频分割的时间一致性。因此，除了现有的基于光流的度量外，我们还使用我们的感知一致性度量进行时间一致性评估。此外，在演示中，我们将提供具体示例来展示使用感知一致性对估计光流的优势。

通过这个演示，我们希望激发对时间一致和自适应视频分割算法的进一步研究，以及它们在移动设备上的有效部署。我们将这个演示的主要贡献总结如下：

• 我们采用我们最新的在线适应方案AuxAdapt，以获得准确且时间一致的视频语义分割。与以前的自适应框架（例如，Tent [7]、DVP [8]）相比，AuxAdapt 只需要它们的一小部分计算成本来进行模型更新。

• 我们提出了一种新颖的跨单元部署策略，以在由 Qualcomm AI Engine 提供支持的商用智能手机上有效运行我们的自适应视频分割算法。具体来说，模型中较大的且冻结的网络在高效的 AI 加速器上运行，而一个更小的自适应部件在手机 GPU 上运行。这使我们能够更好地利用手机上的处理能力并实现实时性能。

• 在评估时间一致性时，除了基于估计光流的传统方法外，我们还进一步利用了一种新的、基于感知一致性的方法。与使用估计的光流相比，这种新的评估方法可以更准确地捕捉时间一致性，我们还将在本演示中展示。

本文剩余部分的安排如下。在第二节，我们将给出我们自适应视频分割算法 AuxAdapt 的概览，以及新的评估时间一致性方法和样本评估结果。在第三节，我们将展示新的在手机上不同计算单元（即 GPU 和 AI 加速器）部署算法以实现实时性能的方案。在第四节，我们会描述实现细节和如何实现这个展示，最后会给出结论。

## AuxAdapt 和时间一致性评估

在本节中，我们将简要解释 AuxAdapt [9] 框架，该框架用作在线、无监督、测试时间适应方案，以增强视频语义分割模型的时间一致性。然后，我们分别基于估计的光流和感知一致性 [10] 描述了两种用于测量时间一致性的评估方案，并比较了不使用和使用 AuxAdapt 的评估结果。

### 2.1 AuxAdapt

考虑现在有一个预训练好的语义分割网络。它的输入是一张图片，，然后生成一个 2D 的预测图，，K 表示这是一个 K 个类别的分割任务。H 和 W 分别表示输入图片的高和宽。 表示对于位置(i,j)上的像素属于类别 k 的概率。对于分割决策，通常会在的最后一个维度上逐像素的使用 argmax 操作来将最有可能的类别分配到每个像素。我们将这种硬决策表示为。而为了获得一个视频序列帧 的语义分割，应用将生成结果，其中 T 表示视频帧的数量。

但按上述做法，将网络应用到视频序列 X 将导致时间不一致的分割结果，正如图 1（a）所示。这个是因为网络对其输出的不确定性，也就是在不同帧里对于视觉上相似区域有不同的分割结果。因此，**AuxAdapt 通过自己的预测来强化网络，即根据自己的决策训练网络，以增强网络的预测置信度和一致性**。虽然可以根据自己的输出简单地更新整个网络以减少不确定性，但**通过整个网络执行反向传播的计算量很大，而且对于长视频，可能会使网络偏离其稳定的预训练权重**，导致精度下降。

为了执行自适应并避免天真地更新整个模型的缺点，AuxAdapt 使用辅助网络 (AuxNet) 与主分割网络 (MainNet) 一起工作。AuxNet 是一个单独训练的小型分割网络。对于时间 t 的视频帧，MainNet 和 AuxNet 产生各自的预测图 和 ，然后对其求和进行 argmax 操作，得到最终的分割决策。当流式传输视频时，MainNet 被冻结，而 AuxNet 基于更新。我们的 AuxAdapt 框架如图 2 所示，并在算法 1 中进行了总结。

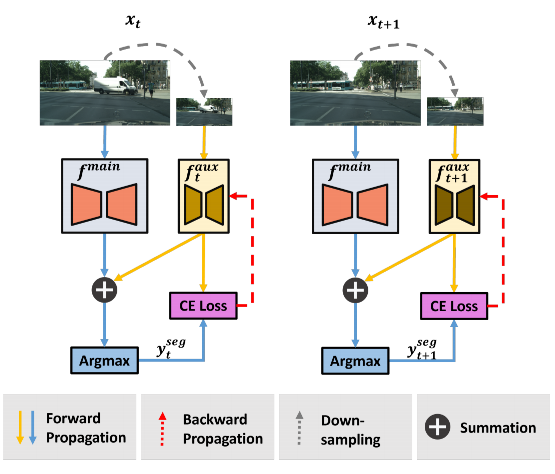
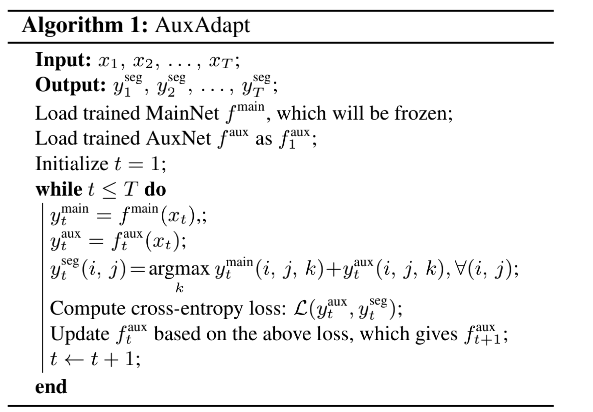


Fig2 AuxAdapt 框架的介绍。AuxAdapt 在比较低分辨率上使用了一个轻量辅助网络 AuxNet ，然后和主要的分割网络 一起生成分割的决策。对于每帧，它通过聚合分割结果 来更新 AuxNet，并且采用的是交叉熵损失。



在这个过程中，虽然 MainNet 没有改变，但 AuxNet 通过从最终的组合决策中学习来进行自我调整，以降低整体预测的不确定性。通过只更新小的 AuxNet，前向和后向操作将比更新 MainNet 便宜得多。此外，AuxAdapt 可以保持给定的特定测试视频和其他一般测试图像的准确性。具体来说，**由于主网没有改变，我们从根本上防止了灾难性遗忘，并且由于主网对整体输出的贡献，准确性的变化受到限制**。AuxNet 还通过组合输出从主网中提取知识，并学习匹配主网的性能。

**网络**：在本演示中，我们采用最先进的分割模型 HRNet-w18 [11]，并将其变体应用于 MainNet 和 AuxNet。特别是，对于 AuxNet，我们分析了 GPU 上 HRNet-w18 每一层的运行时间，并减少了观察到的bottlenecks的计算（例如通道数）。**我们还为 AuxNet 使用了较低的输入分辨率，以进一步减少计算量**。

### 2.2 时间一致性评估

在本节中，我们首先描述了两种评估时间一致性的方法，然后在 Cityscapes 数据集 [12] 上评估使用和不使用 AuxAdapt 的分割结果。在测量时间一致性时，第一种方法是基于估计的光流，并已被普遍使用。第二种方法基于我们新提出的感知一致性，它利用视觉相似性并且不需要光流。正如我们将看到的，我们基于感知一致性的方案提供了一种更可靠的方法来测量时间一致性。

**基于光流的方法**：许多现有论文使用估计的光流来评估视频分割的时间一致性，例如 [4,5,13,14]。在 [5] 之后，我们采用 FlowNet2 [15] 来计算两个相邻帧之间的光流，并将帧 t 的分割warp到帧 t-1。 然后，我们使用标准平均交并比 (mIoU) 度量比较每帧 (t < T) 的扭曲和估计分割掩码。然后，整体 mIoU 用作基于流的时间一致性度量，我们将其表示为 F -TC。请注意，这与 [5] 中使用的度量标准相同，可以在其中找到更多详细信息。

尽管基于流量的度量已被广泛使用，但也存在一些缺点。首先，生成高度准确和可概括的光流估计具有挑战性。此外，它容易受到遮挡和物体移出框架的影响。

**基于感知一致性的方法**：最近，我们提出利用感知一致性来捕捉视频分割的时间一致性[10]。给定两个相邻的视频帧和相应的预测分割图，我们评估建立在这两帧的感知特征图上的分割和跨帧的像素对应关系。

为了量化这一点，对于一帧中的每个像素，我们首先**通过匹配感知特征从另一帧中找到最相关的像素**。这两个像素应该属于同一类。接下来，我们找到分割图里也是最相关的像素。如果分割与感知对应一致，则通过无约束特征匹配和分割一致的特征匹配找到的相关性将相等。否则，受约束的会更小。因此，我们可以使用这两个相关性之间的比率来量化分割和感知对应之间的像素一致性。然后可以将其聚合到像素上，以测量两帧上的分割图之间的感知一致性，这自然地捕获了它们的时间一致性。

与光流不同，感知一致性不会在两个图像之间寻找精确的像素对应关系。相反，**它会找到最大相关的像素对。这使得感知一致性不受遮挡和物体移出框架等情况的影响**，这些情况不再存在精确的对应关系。

**在 Cityscapes 上评估结果**：我们使用 Cityscapes 数据集来展示我们使用 AuxAdapt 进行视频分割的效果。如表 1 所示，我们的方法在两个指标方面都实现了更高的时间一致性。具体而言，AuxAdapt 基于光流的测量 (F-TC) 获得了 74.46 的时间一致性得分，而基于感知一致性的测量（PC）则获得 82.85 的得分。另一方面，当不使用 AuxAdapt 时，即 MainNet 单独应用于每个视频帧而不进行任何自适应时，各自的分数较低：73.06 和 81.02。就分割精度（以 mIoU 为单位）而言，与不使用 AuxAdapt 的情况相比，我们实现了相似（略高）的精度。 我们还报告了 GMAC 测量的计算成本，可以看出由于 AuxAdapt 的适配而增加的计算量仅为 4.2G。通过使用提议的 AuxAdapt，我们有效地提高了时间一致性，同时保持了原始准确性。

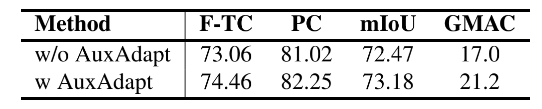


表 1 在 Cityscapes 验证集上的性能评估结果。F-TC 表示基于光流的时间一致性评估（采用 FlowNet2），而 PC 表示使用基于感知一致性的时间一致性评估标准。GMAC 是基于输入为 1024x2048 的计算。

不使用和使用 AuxAdapt 的示例视频分割结果如图 3 所示。可以看出，Aux Adapt 显着提高了视频分割的时间一致性。 此外，我们展示了基于光流（左）和感知一致性（右）的每一帧（在当前帧和前一帧之间计算）的时间一致性分数。 分数还表明基于 AuxAdapt 的视频分割实现了更好的时间一致性。 在演示中，我们将展示在智能手机上实时运行的完整视频的一致分割。

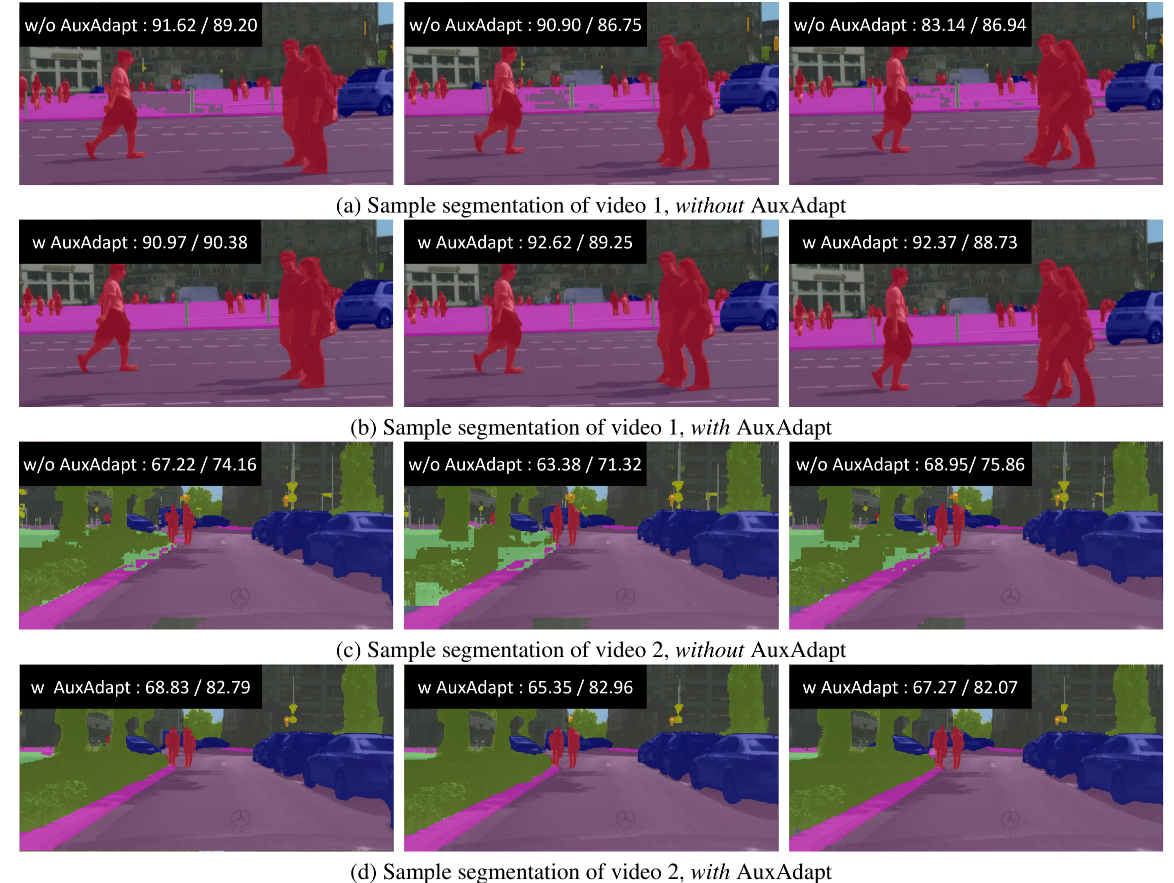


Fig3 两个视频帧序列的分割结果，没有和使用 AuxAdapt。第 1 行和第 3 行显示没有 AuxAdapt 的结果，第 2 行和第 4 行是通过应用 AuxAdapt 获得的。对于每一帧，我们还显示了基于光流（左）和感知一致性（右）的时间一致性分数（当前帧和前一帧之间）。可见，AuxAdapt 大大增强了视频分割的时间一致性，这也反映在分数上。

**对比两个时间一致性指标**：图 4 显示了基于光流和基于感知一致性的时间一致性分数示例视频分割结果。可以看出，在图 4（a）中，当不使用 AuxAdapt 时，分割结果有闪烁的伪影（由红色框突出显示）。另一方面，使用 AuxAdapt 的视频分割在帧之间显示出更加一致的结果，如图 4 (b) 所示。我们根据每帧的两个时间一致性度量（在当前帧和先前帧之间计算）显示各自的分数，左侧是基于光流的分数，右侧是基于感知一致性的分数。我们看到，**基于我们新的感知一致性的度量将更高的分数分配给更一致的分割（底行），而基于流的度量错误地为不太一致的结果（顶行）分配了更高的分数**。在演示过程中，我们将提供更多示例和分析，以更好地说明使用感知一致性作为时间一致性度量的优势。

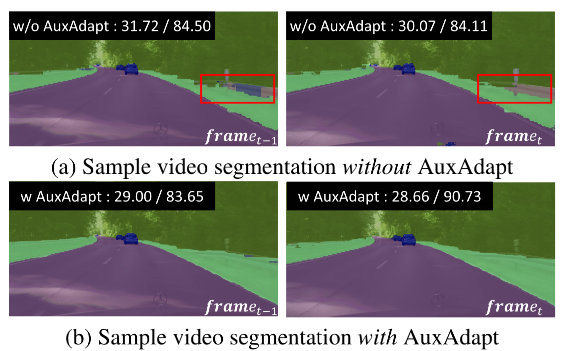


Fig4 连续视频帧样本对的两个时间一致性指标的比较。 在每一帧上，基于流的度量和基于感知一致性的度量的时间一致性分数分别显示在左侧和右侧。 可以看出，感知一致性正确地为底部行中明显更一致的结果分配了更高的分数，而基于流的度量错误地表明没有 AuxAdapt（顶部）的视频分割更加一致

## 3. 在手机上跨单元部署

我们采用了一种新颖的跨单元部署策略，以最好地利用由 Qualcomm AI 引擎驱动的智能手机的处理能力。该平台由多个硬件和软件组件组成，可加速设备上支持 AI 的用户体验。AI 引擎支持的硬件架构包括 Qualcomm® Hexagon™ 处理器、Qualcomm® Adreno™ GPU 和 Qualcomm® Kryo™ CPU——所有这些都旨在在设备上快速高效地运行 AI 应用程序。Hexagon 处理器具有融合的 AI 加速器架构，将标量、向量和张量加速器融合在一起。这使得能够提供快速的定点整数运算，同时具有低功耗。

由于与 AuxNet 相比，MainNet更重且需要更多计算操作，因此我们利用 Hexagon 处理器并在其上运行MainNet的量化版本。这可以实现对MainNet的非常快速的推理。同时，较小且计算要求较低的 AuxNet 运行在 GPU 上，在 GPU 上执行前向推理和反向传播以适应模型。 通过采用这种跨单元部署的方式，我们适当地利用了智能手机上 Qualcomm AI Engine 提供的可用 AI 计算资源。相反，在 GPU 上同时运行 MainNet 和 AuxNet 的简单方法会导致功能强大的 Hexagon 处理器未被使用，并导致性能大大降低。正如我们将在演示中展示的那样，我们的跨单元方法可以实现实时、自适应、准确和一致的视频语义分割。

图 5 描述了我们的跨单元部署策略。特别需要注意的是，给定一个视频帧，量化的 MainNet 在 Hexagon 处理器上运行并生成一个分割预测。 会转为浮点类型并传给 GPU，这里用表示。在 GPU 上，输入帧手续进行下采样然后再传入 AuxNet，这可以减小计算量。AuxNet 会生成一个分割预测。然后，和相加，并采用一个 argmax 函数得到对于输入帧的最终分割结果。为了对 AuxNet 自适应，接着会计算和的交叉熵，并采用反向传播来更新 AuxNet 的权重。

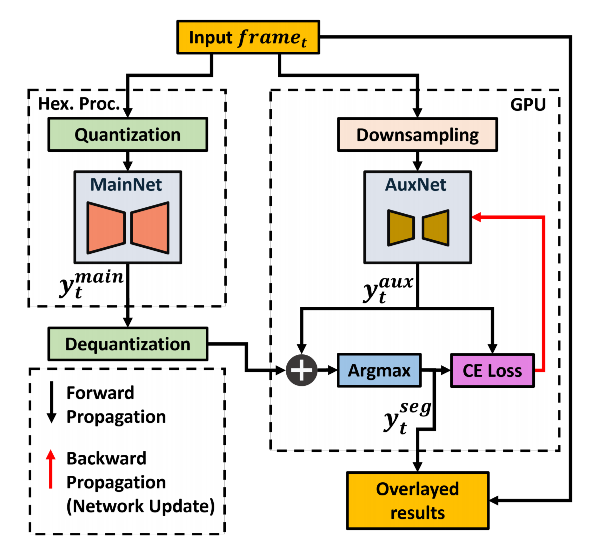


Fig5 AuxAdapt 的跨单元部署，用于移动设备上的实时应用程序。MainNet 的量化版本部署在 Hexagon 处理器 (Hex. Proc.) 上，而 AuxNet 部署在移动 GPU 上。MainNet 的输出在发送到 GPU 之前被转换回浮点数。在 GPU 上，MainNet 和 AuxNet 的输出被聚合以提供给定帧的最终分割。之后，AuxNet 基于这个组合的分割决策通过反向传播进行更新。

## 实现和示范细节

该演示将作为 Android 应用程序在搭载 Qualcomm AI 引擎的手机上运行。该应用程序将视频序列作为输入，并运行我们部署的自适应视频分割模型以实时生成相应的分割图。

为了在 Hexagon 处理器上部署，将使用 Qualcomm 创新中心的开源 AI 模型效率工具包 (AIMET) [16] 对MainNet进行量化。对于在 GPU 上在线更新 AuxNet，我们使用 OpenCL ML SDK [17, 18] . 我们进一步利用多种技术来加速 AuxNet 在 GPU 上的操作。 首先，使用 16 位浮点数 (FP16) 进行前向传递，使用 32 位浮点数 (FP32) 进行反向传递。 与使用 FP32 相比，使用 FP16 将前向推理时间减少了一半。 此外，**我们冻结了网络中的早期层以减少可训练层的数量，从而加速了反向传播**。

在演示期间，我们将首先简要概述我们要演示的技术和创新。之后，我们将展示我们在手机上实时运行的在线自适应视频分割模型。更具体地说，我们将强调我们方法的两个方面：**1）更好的时间一致性和 2）满足实时性能的帧速率要求**。这将通过一个 Android 应用程序来完成，该应用程序将在手机屏幕上显示输入 RGB 视频帧序列和基于我们部署的模型以高帧速率生成的一致分割结果，以及另一个将显示基线分割网络的应用程序在另一部手机上。在运行应用程序时，我们将在特定时间暂停视频以突出显示示例区域，以展示我们改进的分割。还将显示其他信息，例如准确度数、时间一致性分数和实时帧速率。将向观众提供屏幕录像，并举行互动问答环节以回答问题。

## 结论

在这项工作中，我们概述了我们的方法来展示我们在移动设备上高效、准确和时间一致的视频语义分割方面的创新。我们应用了我们的测试时间适应方案 AuxAdapt，以使图像分割模型能够在线适应给定的视频。通过运行 AuxAdapt，我们可以以经济高效的方式显着提高视频分割的时间一致性，同时保持准确性。为了测量时间一致性，我们使用了我们的新指标，感知一致性。我们还展示了该指标比广泛使用的基于光流的指标更有效。最后，我们展示了一个跨单元部署方案，该方案充分利用了搭载 Snapdragon 的智能手机上可用的 AI 处理能力，并实现了我们的自适应视频分割算法的实时操作。