3个

Tesla AI Day过去已经4个多月，其介绍的很多前卫理念和超级详细的技术方案细节都成为全球自动驾驶从业者津津乐道的话题与专研的方向。这段时间以来我重复看了几遍AI Day的视频资料，也看了不少中英文分析解读的文章，一直希望能找机会把我对AI Day的理解和解读写成文章分享出来，可是因为拖延症一拖再拖，虽然拖了这么久，可至今AI Day上Tesla展现的技术创新仍旧走在自动驾驶视觉感知技术的最前沿，所以希望通过这篇文章能够让更多人了解如今自动驾驶车端感知技术的前沿发展动态。

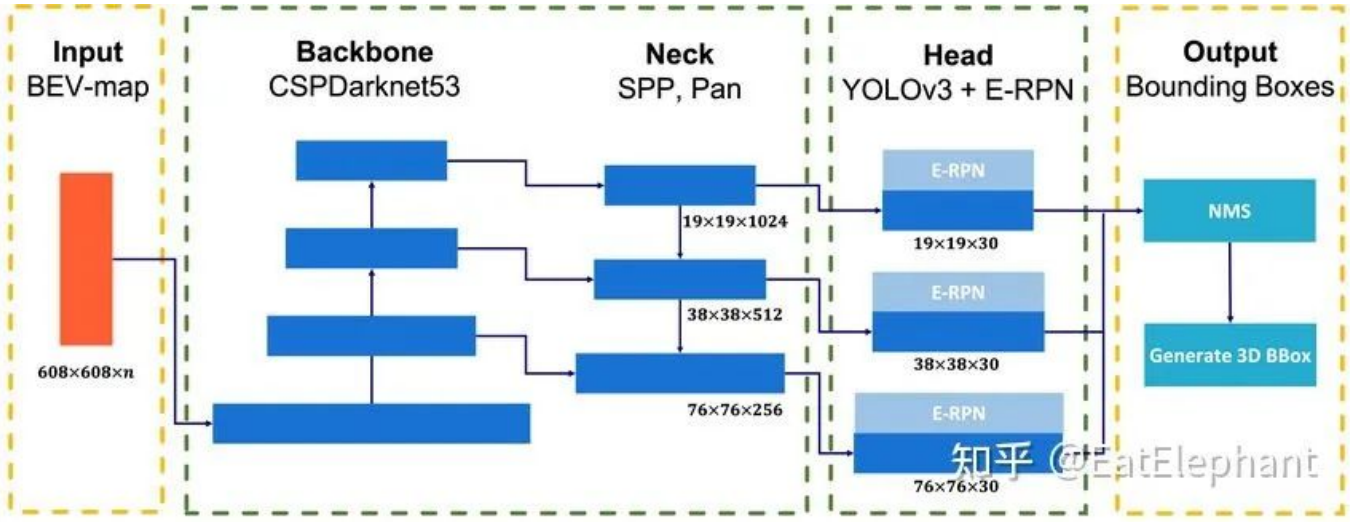
Tesla FSD Beta从2020年10月推送至今依然是全球唯一的用户能够独立使用的城区辅助驾驶系统

Tesla 感知技术的进化

通过Tesla不可谓不频繁的技术分享可以发现，Tesla的感知技术每隔一段时间都会对自己已有的技术栈进行一次重大革新，根据截至到文章撰写时刻所透漏的Tesla感知技术方案，我把这个技术不断迭代更新的过程分成三个重要阶段：以HydraNet为代表的第一次革新，以BEV Layer为代表的第二次革新，以及以时空序列Feature处理模块为代表第三次革新，下面将分别予以详细介绍。

HydraNet：共享特征的多任务网络

截至到目前，大多数学术研究中的网络模型为了便于与其他研究成果进行比较，且受限于有限的资源、时间，基本上都使用一套公开数据集，为了一个具体的目标（目标检测，语义分割，实例分割，深度估计等等）设计一套类似下图的网络架构，具有Backbone（主干）和Neck（颈部）两部分进行特征提取，以及Head（头部）部分根据任务具体类型给出任务需求的输出。



针对具体检测任务设计的具有Backbone，Neck，Head的网络结构

然而要想让车理解交通环境从而能自主地在人类的交通设施下完成自动驾驶，AI就必须同时掌握数量众多的感知任务（例如：车辆、行人、自行车检测，车道线分割，可通行区域分割，红绿灯、标志牌检测，视觉深度估计，异形障碍物检测等等），使用有限的车端计算资源不可能为每一项任务单独设计一路网络，而且人们发现位于网络低层的Backbone，Neck等主要提取一些具有通用性质的视觉特征，在不同任务中共用反而有助于最终感知任务的完成，因此自动驾驶行业从业者普遍采用共享用于特征提取的主干的多头（Multi-head）网络作为自动驾驶感知模块的网络结构，这样一方面能够利用共享主干节省大量重复近似的特征提取计算，同时可以在主干参数保持不变或者较少参与更新的情况下相对独立地优化各个Head的性能以满足自动驾驶的需求。

Tesla在为了研发FSD对代码进行彻底的重构之前使用的也是这样的技术方案，而这也是目前绝大多数一线自动驾驶公司所达到的技术水平了。Tesla与其他众多公司处于这一阶段的公司的差异主要体现在Head种类的多少上，根据2020年上半年Andrej Karparthy的技术分享，当时Tesla大概有1000多个Head来解决各种各样的自动驾驶感知任务类别，这1000多个Head又可归类于包括但不局限于下面分类的超过50个大的感知类别Task中：

- Moving Objects: Pedestrian, Cars, Bicycles, Animals, etc.
- Static Objects: Road signs, Lane lines, Road Markings, Traffic Lights, Overhead Signs, Cross-Walks, Curbs, etc.

- Environment Tags: School Zone, Residential Area, Tunnel, Toll booth, etc

正因为有着如此之多Heads，Tesla给自己的神经网络模型起名叫做HydraNet，寓意一个主干，多个头，不可谓不形象。然而最近在Lex的采访中，Elon Musk提到HydraNet时候表现出不是很喜欢这个名称，称总有一天会给FSD的神经网络取一个新名字，我们拭目以待。我在之前的一篇文章中曾经很详细的介绍过HydraNet的特点，以及Tesla训练HydraNet的一些最佳实践和挑战，感兴趣可以移步解读: Tesla Autopilot技术架构进一步研究，这里不再赘述。

Hydra即九头蛇，是一种存在于众多神话中的传说中的生物

BEV Layer：FSD感知的空间理解能力

FSD发布前的Autopilot的以及目前主流的自动驾驶感知神经网络结构都采用了类似HydraNet的共享主干多头结构，但是他们的问题在于整个感知都是在图像所在的Image Space(即图像空间）下进行的。相机图像所在的Image Space是一个2D像素世界，然而所有自动驾驶的决策和路径规划都是在车辆所在的3D世界下进行的，这样的维度不匹配就使得基于感知结果直接进行自动驾驶变得异常困难。

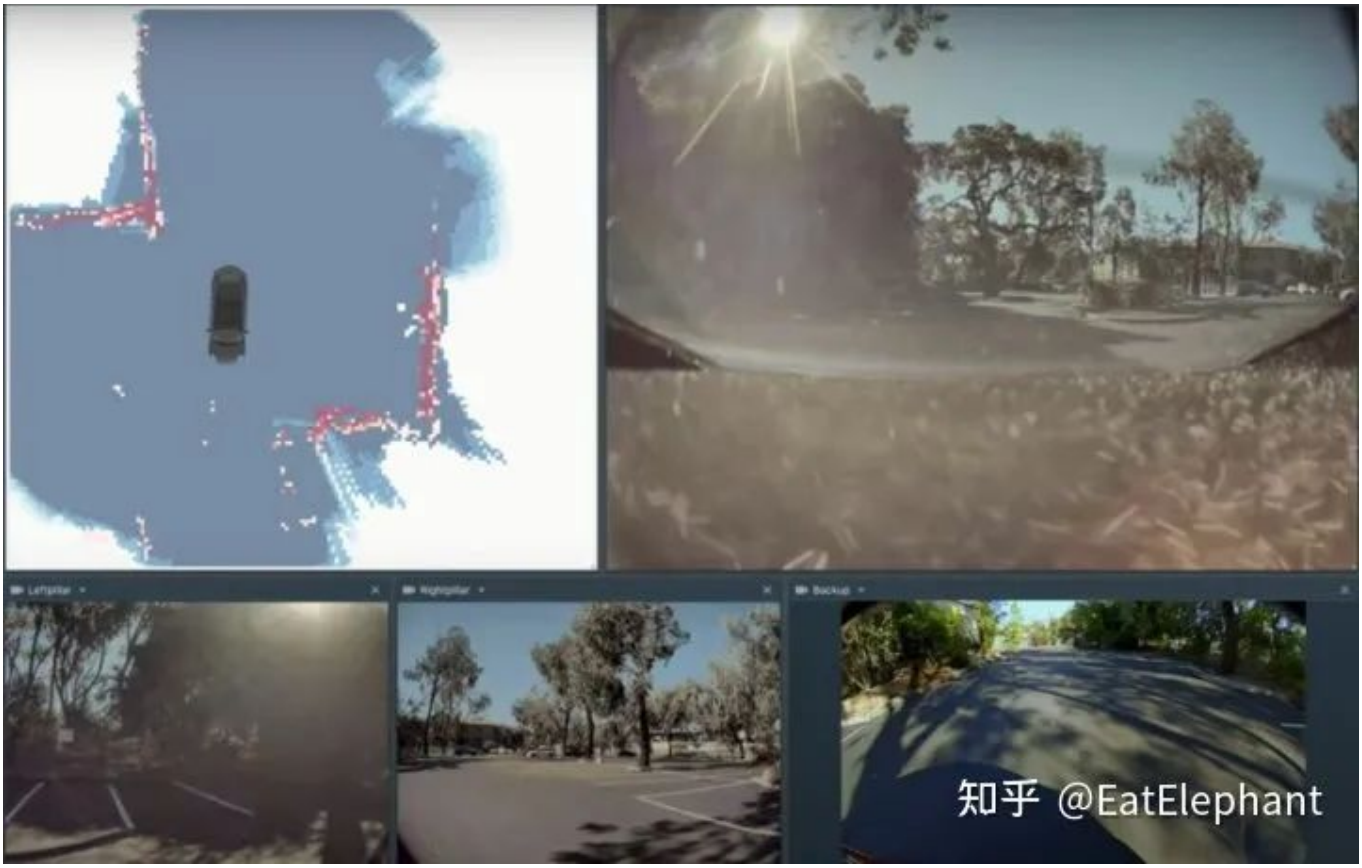
打个比方，开车的新手大多有体会，刚学倒车的时候利用后视镜观察到的场景很难直观的构建车子与周围环境的空间联系，因此新手靠倒车镜很容易发生误操作造成剐蹭，这其实就是缺乏从倒视镜图像平面到自车坐标系空间转换的空间理解能力造成的。

因此一个只能在相机图像平面进行感知的自动驾驶AI就好比缺乏空间理解力的驾驶新手，很难完成把车开好的难题，这时候很多公司就选择使用毫米波雷达，激光雷达这类具有深度测量能力的传感器，通过他们与相机结合来帮助相机把图像平面感知结果转换到自车所在的3D世界，这个3D世界用专业的术语叫做自车坐标系，如果忽略高程信息，那么很多人也管拍扁后的自车坐标系叫做BEV坐标系（即鸟瞰俯视图坐标系）。

不需要是超人或蝙蝠侠，普通人类依靠双眼就可以驾驶汽车

然而就像Elon Musk说的那样，人类不是超人，也不是蝙蝠侠，不能够眼放激光，也不装备有雷达，但是通过眼睛捕捉到的图像，人类依旧可以构建出对周围世界的3D空间理解能力从而很好地掌握驾驶这项能力，那么要像人一样单纯利用眼睛（相机）进行自动驾驶就必须具备从2D图像平面到3D自车空间的转换能力。

传统方案利用相机外参以及地面平面假设便可使用称作IPM（Inverse Perspective Mapping)的方法将图像平面的感知结果反投影回自车BEV坐标系，Tesla原本的方案也是这么做的，然而当地面不满足平面假说，且多相机视野关联受到各种复杂环境影响的时候，这类方法就会变得越来越难以维护。

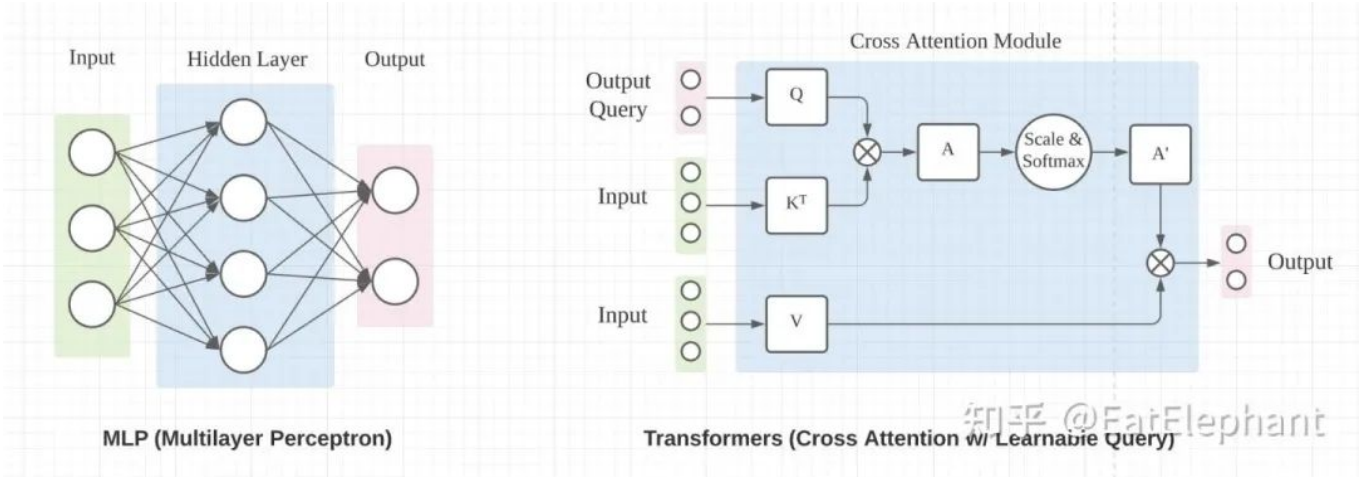


Tesla最早也曾使用IPM变换和帧间匹配进行多摄像头拼接和BEV的转换

出于IPM方法遇到的困难，Andrej Karparthy和他的团队开始尝试直接在神经网络中完成图像平面到BEV的空间变换，这一改变则成为了2020年10月发布的FSD Beta与之前的Autopilot产品最为显著的不同之处。

因为视觉感知输入的图片本就在相机2D平面内，而2D图像平面与自车BEV存在尺寸和维度上的不统一，要在神经网络模型内部进行图像平面到BEV的空间变换最重要的就是要找到方法进行神经网络内部的Feature Map的空间尺度上变换。

主流的能进行空间尺度上变换的神经网络操作主要有MLP中Fully Connected Layer和Transformer中Cross Attention两种办法，如下图所示（图片引用自towardsdatascience.com/），二者都可以改变输入量的空间尺度，从而达到进行尺度变换的目的。



在这个示意图中MLP和Transformer都可以将3D的输入改变成2D的输出

本来目标是尽量少摆公式把道理讲明白，但是BEV转换本身是一个尺度层面的数学变换，不摆公式很难讲的清楚，这里利用公式说明下：

Fully Connected Layer：

$$O = Act(W_{mlp} \cdot X + b)$$

忽略非线性激活函数和bias后

$$O = W_{mlp} \cdot X$$

Cross Attention:

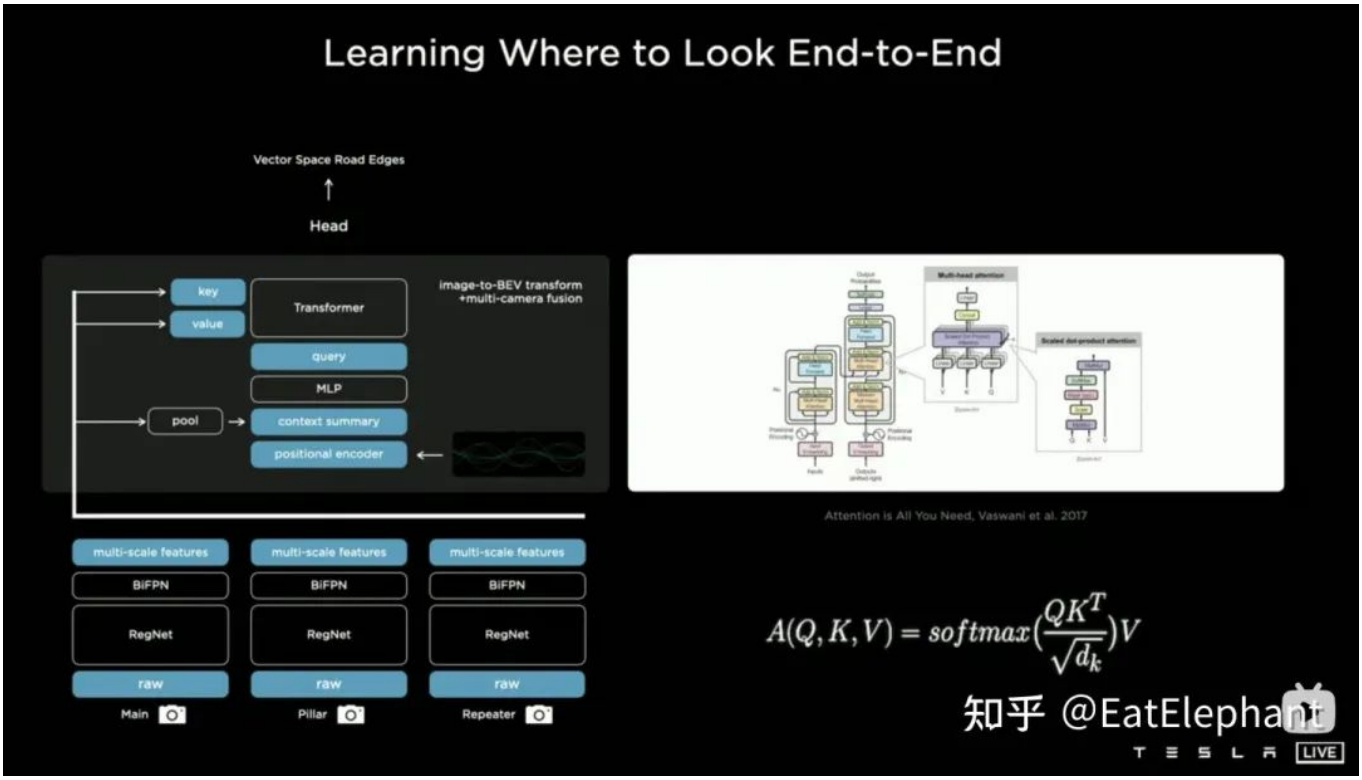
$$O = Softmax(Q \cdot K^T) \cdot V$$

这里K，V是输入X经线性变换后得到的，Q是输出空间上的索引量　　经线性变换得到的，对公式稍作修改可以得到：

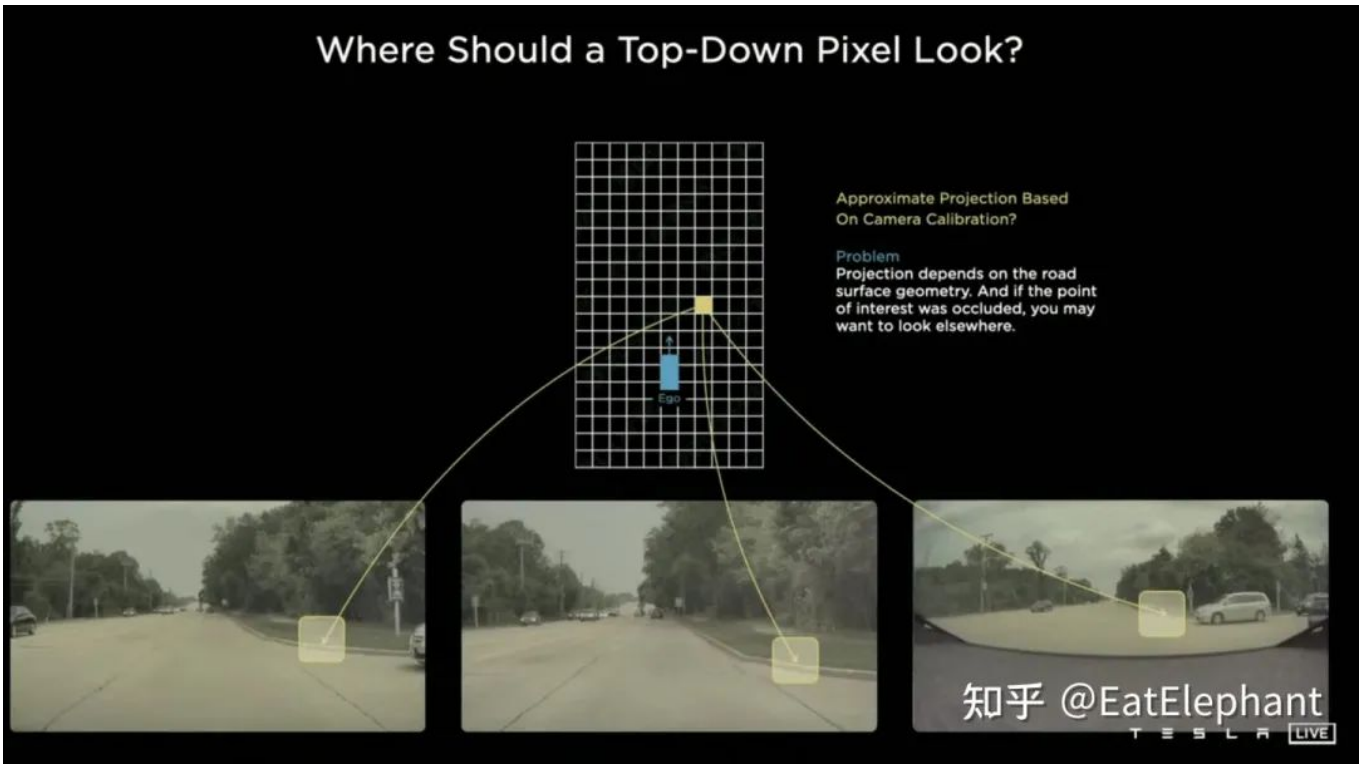
$$O = Softmax(\phi W_q \cdot (XW_K)^T) \cdot XW_V = W_{transformer}(X, \ \phi) \cdot XW_V$$

在进行BEV变换的过程实际上就是利用两个操作中的任意一个将输入的2D图像空间的特征图层转换到BEV自车坐标下的特征图层的过程。因此假设通过多层CNN特征提取，图像空间特征图层尺度即这里X的尺度，为图像的高乘宽　　，输出O所在的BEV空间尺度则是以自车位置为原点的前后左右各若干米范围内建立的栅格空间　　。

对比可见，Fully Connect Layer和Cross Attention的最大不同实际在于作用于输入量X的系数W，全联接层的W一旦训练结束后在Inference阶段是固定不变的，而采用Cross Attention的Transformer的X的系数W是一个输入量X和索引量　　的函数，在Inference阶段会根据输出量X和索引量的不同发生改变，从这个角度来讲使用Cross Attention来进行空间尺度变换可能使模型获得更强的表达能力。因为以上原因，Tesla在他们的方案中实际使用了Cross Attention的办法来实现BEV层。



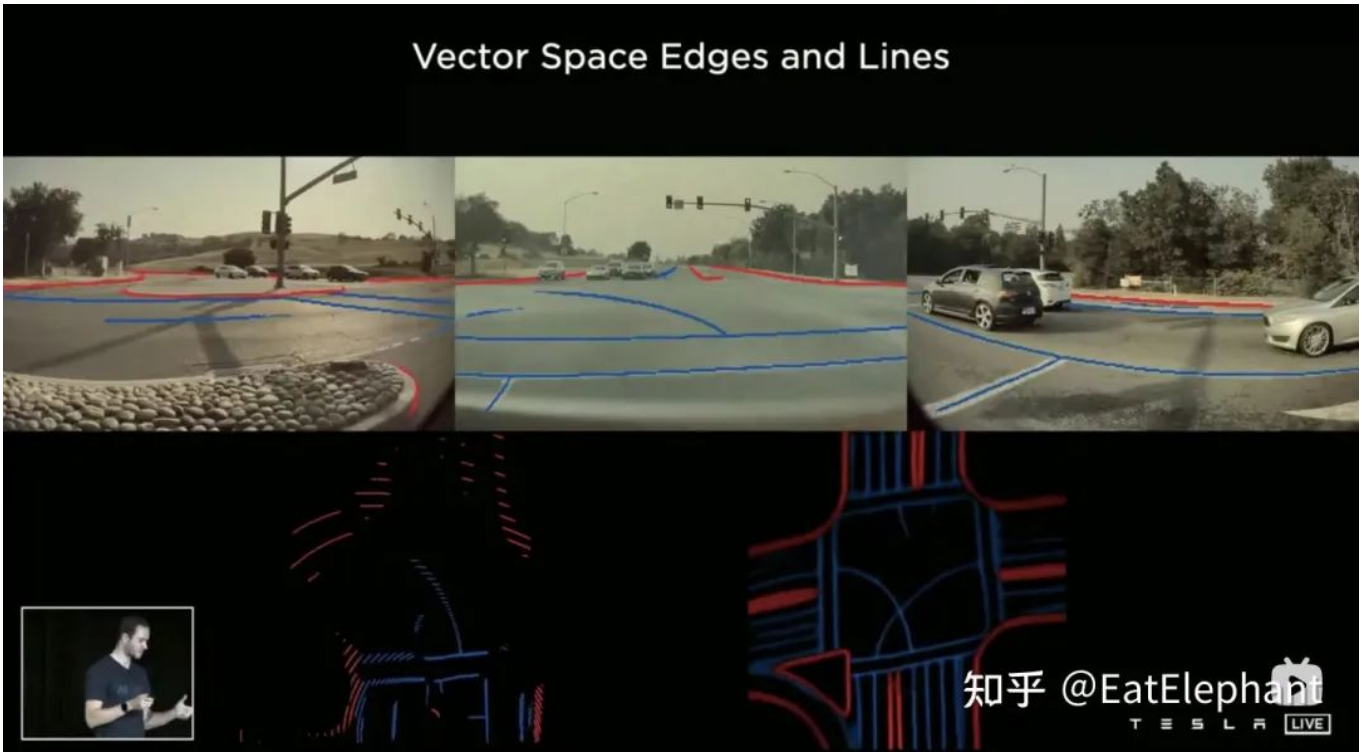
AI Day Karparthy给出的特斯拉基于Transformer的BEV网络结构



Query来自于如上图一样对BEV空间进行栅格化，然后利用Positional Encoding对每个栅格进行差异化编码

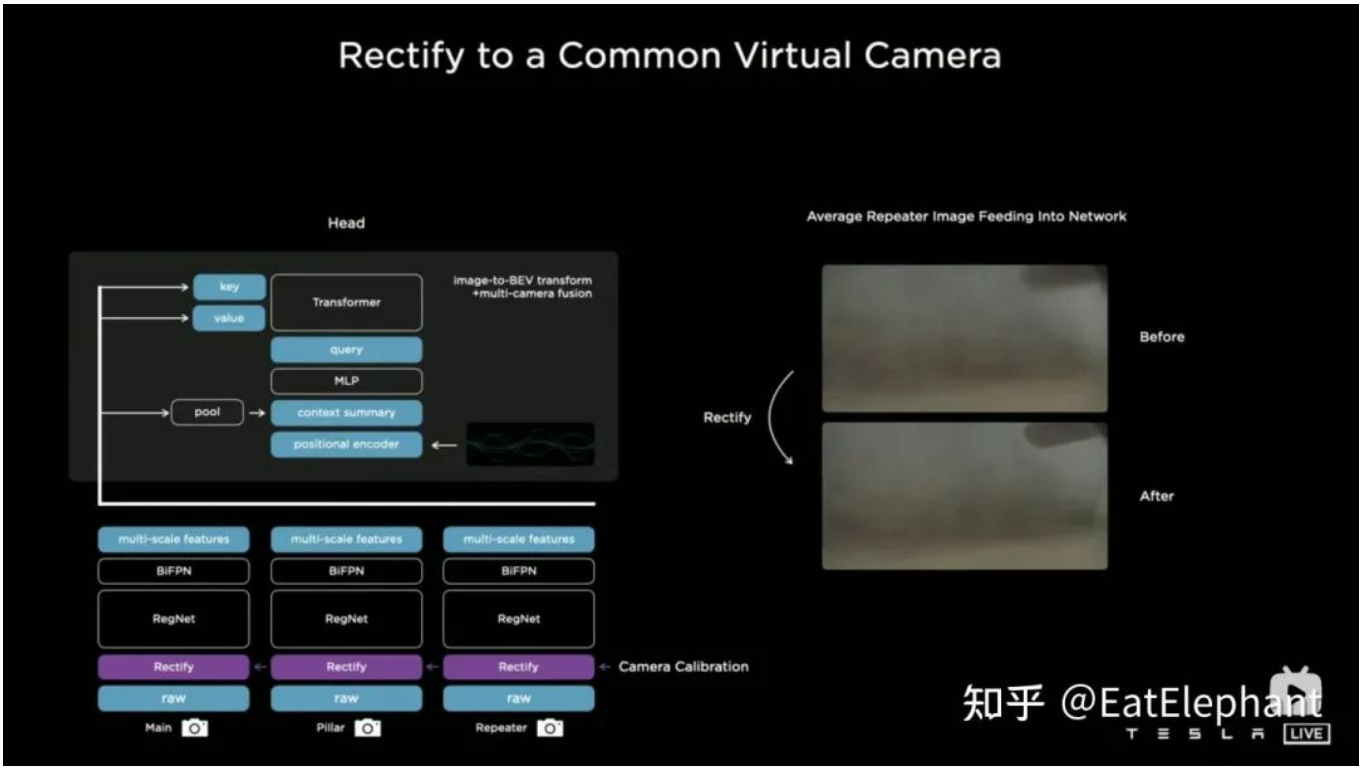
上面两张图展示了Tesla基于Transformer的BEV Layer的实现方案：首先在各个相机分别通过CNN主干网络和BiFPN提取多尺度特征图层，多尺度特征图层一方面通过MLP层生成Transformer的方法中所需的Key和Value，另一方面对多尺度Feature Map进行Global Pooling操作得到一个全局描述向量（即图中的Context Summary），同时通过对目标输出BEV空间进行栅格化，再对每个BEV栅格进行位置编码，将这些位置编码与全局描述向量进行

拼接（Concatenate）后再通过一层MLP层得到Transformer所需的Query。在Cross Attention操作中，Query的尺度决定最终BEV层之后的输出尺度（即BEV栅格的尺度），而Key和Value分别处于2D图像坐标空间下，按照Transformer的原理，通过Query和Key建立每个BEV栅格收到2D图像平面像素的影响权重，从而建立从BEV到输入图像之间的关联，再利用这些权重加权由图像平面下的特征得到的Value，最终得到BEV坐标系下的Feature Map，完成BEV坐标转换层的使命，后面就可以基于BEV下的Feature Map利用已经成熟的各个感知功能头来直接在BEV空间下进行感知了。BEV空间下的感知结果与决策规划所在的坐标系是统一的，因此感知与后续模块就通过BEV变换紧密地联系到了一起。



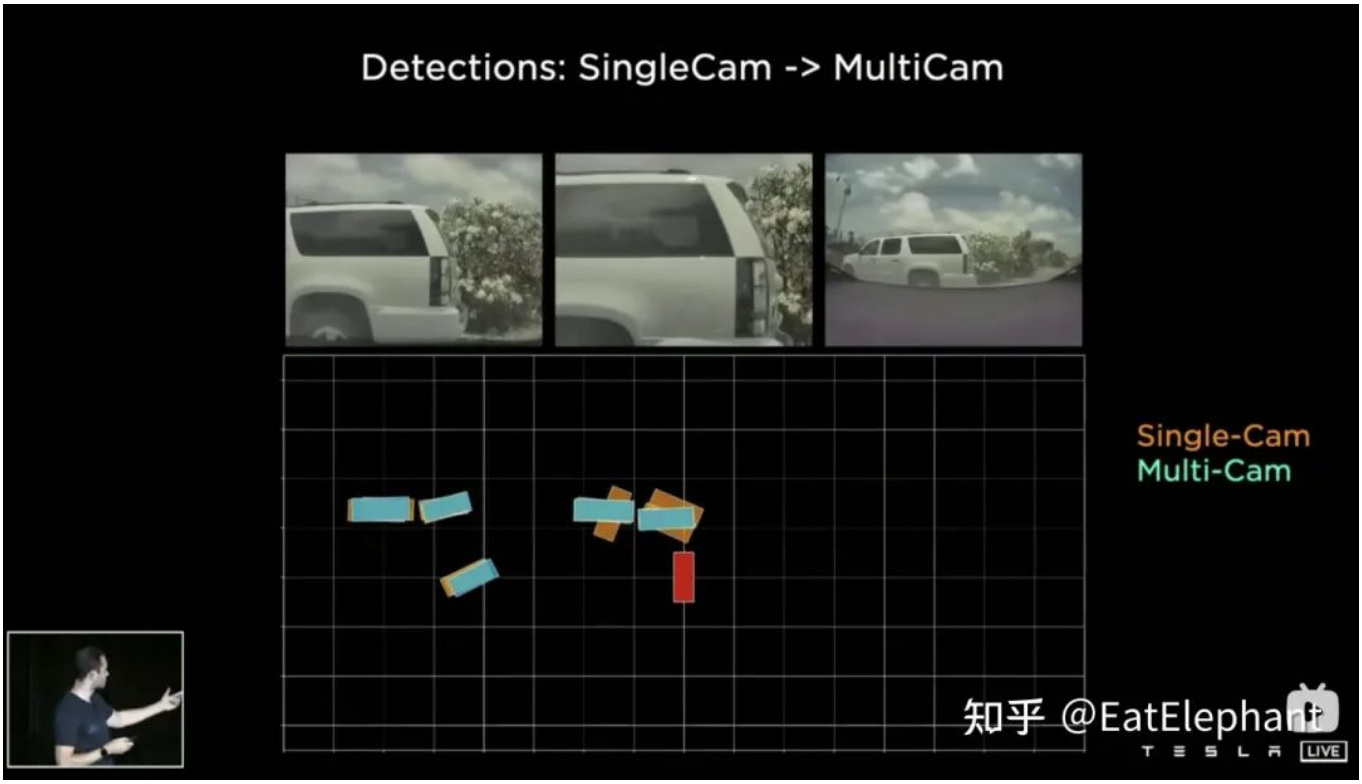
有了数据驱动的BEV变换，Tesla感知直接输出在BEV Vector Space下，如途中右下角所示

通过这种方法，实际上相机外参以及地面几何形状的变化都在训练过程中被神经网络模型内化在参数里边。这里存在的一个问题就是使用同一套模型参数的不同车子的相机外参存在微小的差异，Karparthy在AI Day上补充了一个Tesla应对外参差异的方法：他们利用标定出来的外参将每辆车采集到的图像通过去畸变，旋转，恢复畸变的办法统一转换到同样一套虚拟标准相机的布设位置，从而消除了不同车相机外参的微小差别。



特斯拉在原始图像进入神经网络模型前增加了Rectify的操作，利用相机标定参数去除相机安装的误差影响

BEV网络的优势不仅在于可以使用感知输出直接进行决策规划，BEV的方法还是一个非常有效的多相机融合框架，通过BEV的方案，原本很难进行正确关联的跨多个相机的近处的大目标的尺寸估计和追踪都变得更加准确、稳定，同时这种方案也使得算法对于某一个或几个相机短时间的遮挡，丢失有了更强的鲁棒性。

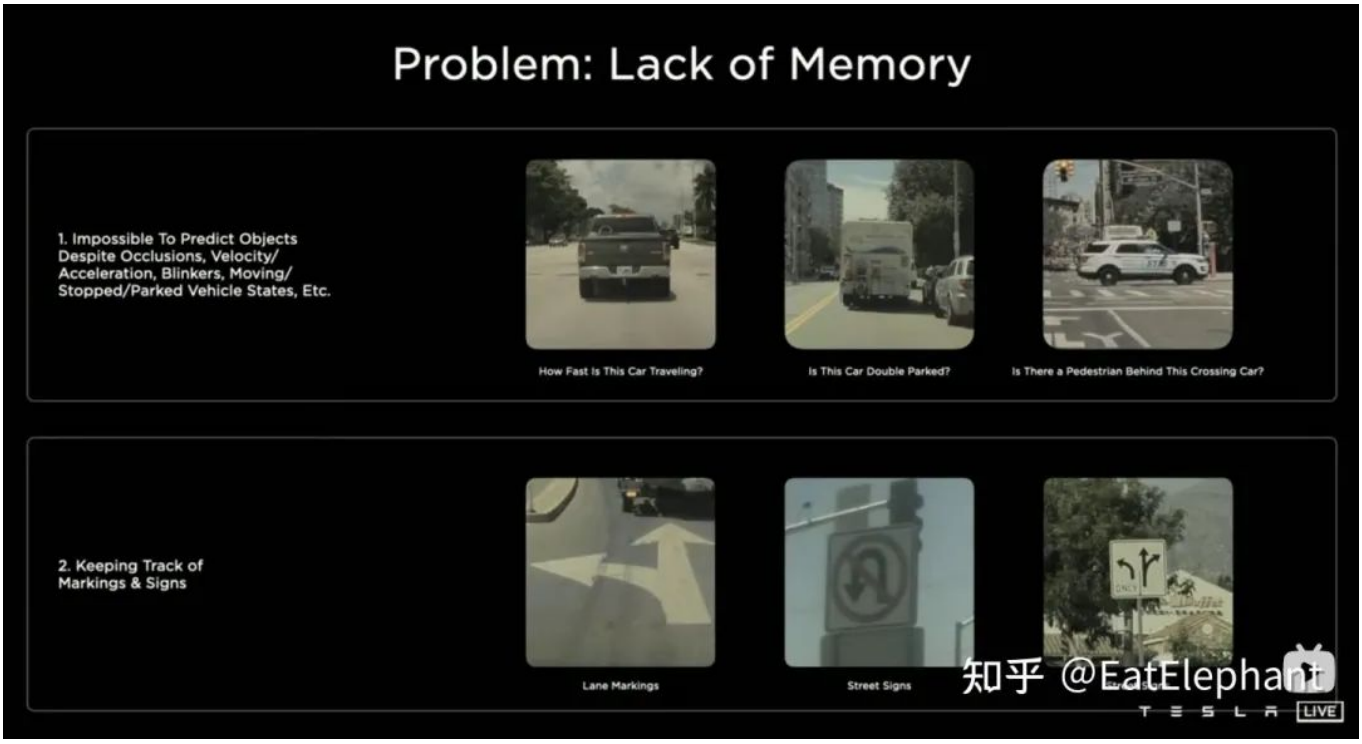


BEV融合多相机特征，显著提高了近处物体检测的准确性

时空序列Feature：为自动驾驶智能增添短期记忆

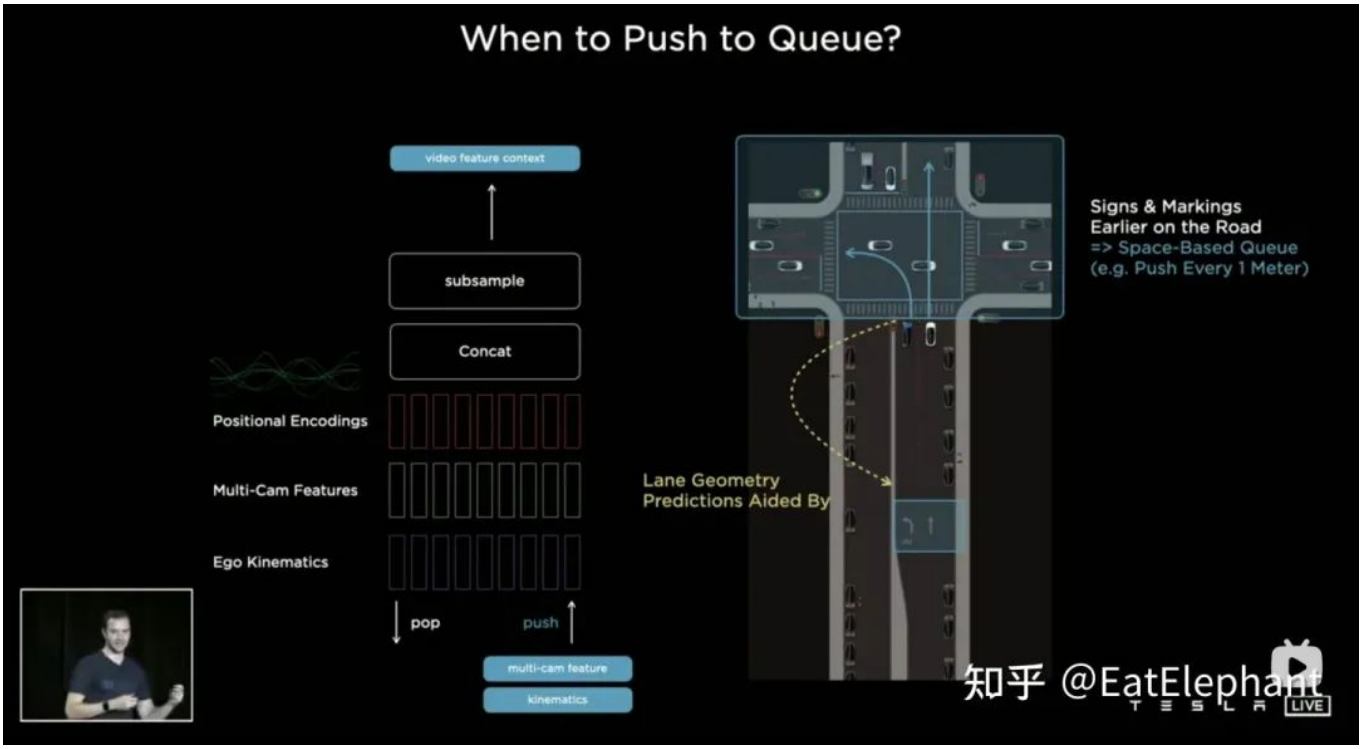
BEV的使用奠定了Tesla FSD在视觉感知的领先地位，然而单纯依靠HydraNet和BEV仍会因为只是用了单一时刻的多张图像作为感知输入而存在连续信息丢失的问题。人对于速度，对于空间的感知很多来自时间维度，好比一个老司机可以在一个熟悉的场景里的很好的完成驾驶任务，但是如果把这个场景中录下的视频拆成乱序的单帧图片，再问这个老司机在其中一张图片里应该怎么打方向盘，怎么踩油门刹车，那恐怕老司机也要被难倒了。老司机无法对单帧图像给出驾驶操作的原因在于熟悉连续数据处理的人脑在单帧图像里无法形成很好的空间概念也无法处理周围物体的速度，轨迹等跟时间相关的信息，因此我们的驾驶技能就没法正常工作。

同样的道理，FSD也需要具备处理连续的时空序列数据的能力，才能正确处理如城市环境下常见的闪烁的交通灯，分辨参与交通的临时停车和路边的静止车辆，预测周围物体与自车的相对速度，根据历史信息预测参与交通的物体可能的运行轨迹，解决段时间遮挡问题，记忆刚刚开过的速度标识，车道行驶方向等等。换句话说，FSD需要被赋予短期记忆。



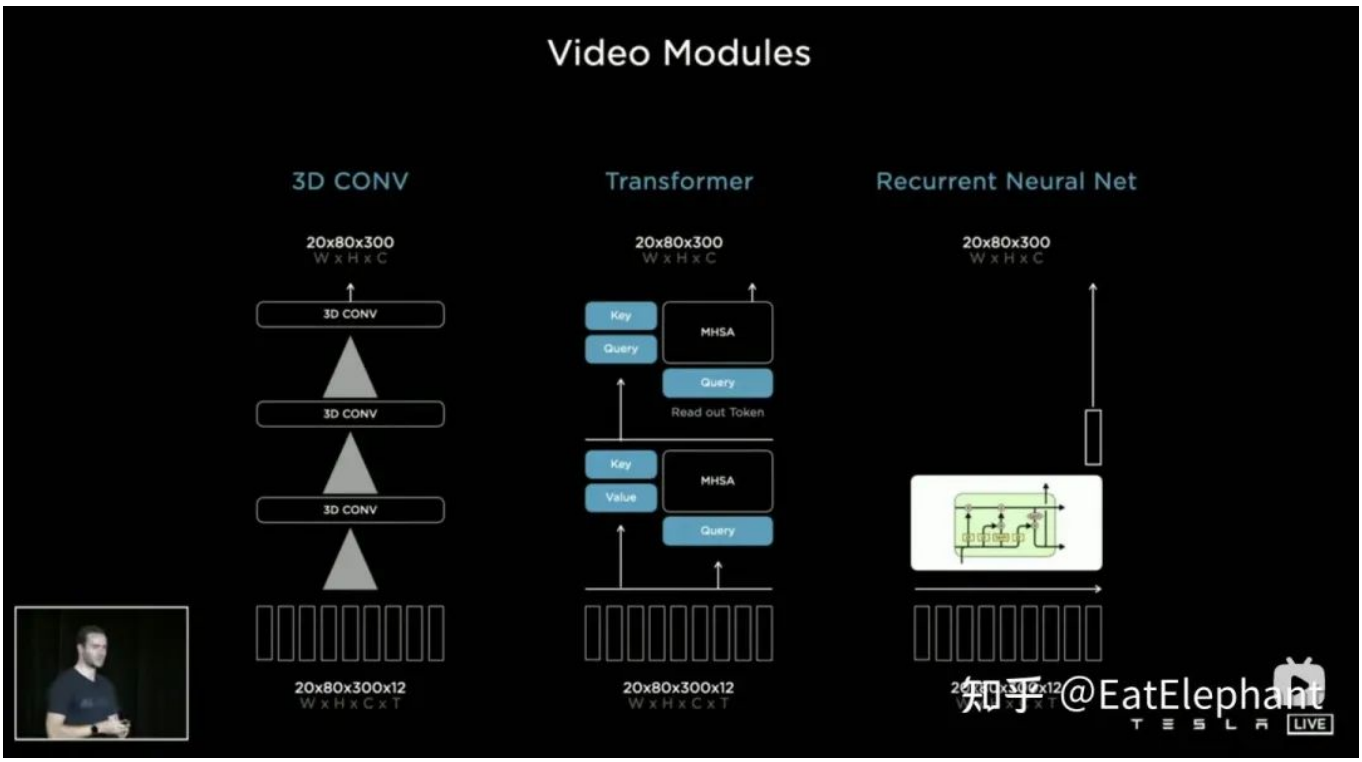
缺乏短期记忆能力的自动驾驶AI所不能解决的问题

人类的记忆究竟是如何工作的仍旧是脑科学中未解的谜题之一，因此AI算法至今仍旧无法重构人类的长期记忆能力。不过Tesla希望通过使用具有时序信息的视频片段替代图像对神经网络进行训练，从而使感知模型具有短时间的记忆的能力，实现这个功能的方法是分别引入时间维度和空间维度上的特征队列进入神经网络模型。针对为什么需要空间维度的特征队列Karparthy举了一个例子，如果车子行驶到红绿灯路口停止，车子在到达路口前观察到了各车道允许行驶方向的箭头，然而如果单纯依靠时间队列，那么当红灯非常长的时候，前面时刻观察到的车道方向终究会被遗忘，但如果引入空间队列，那么由于红灯下车子没动，无论在红绿灯路口停止多久，空间队列仍能保留对前面观察到的车道行驶方向的记忆。

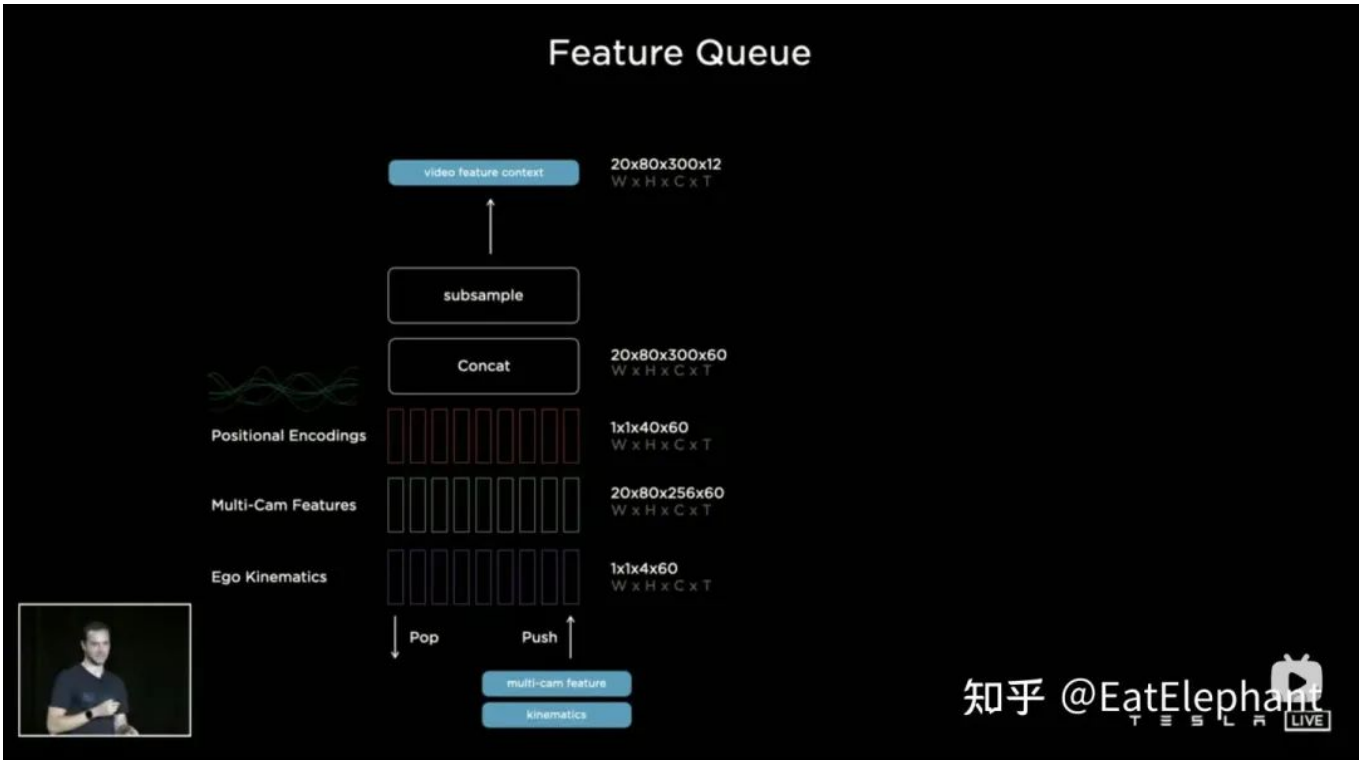


时序队列在路口停车的case中无法胜任，需要引入空间队列

对于如何融合时序信息，Tesla尝试了三种主流的方案：3D卷积，Transformer以及RNN。这三种方法都需要把自车运动信息与单帧感知结合起来，Karparthy表示自车运动信息只使用了包括速度和加速度的四维信息，这些运动信息可以从IMU中获取，然后与BEV空间下的Feature Map(20x80x256)还有Positional Encoding相结合(Concatenate)，形成20x80x300x12维的特征向量队列，这里第三维由256维视觉特征 + 4维运动学特征（vx, vy, ax, ay)以及40维位置编码（Positional Encoding）构成，因此300 = 256 + 4 + 40，最后一维是降采样过后的12帧时间/空间维度。

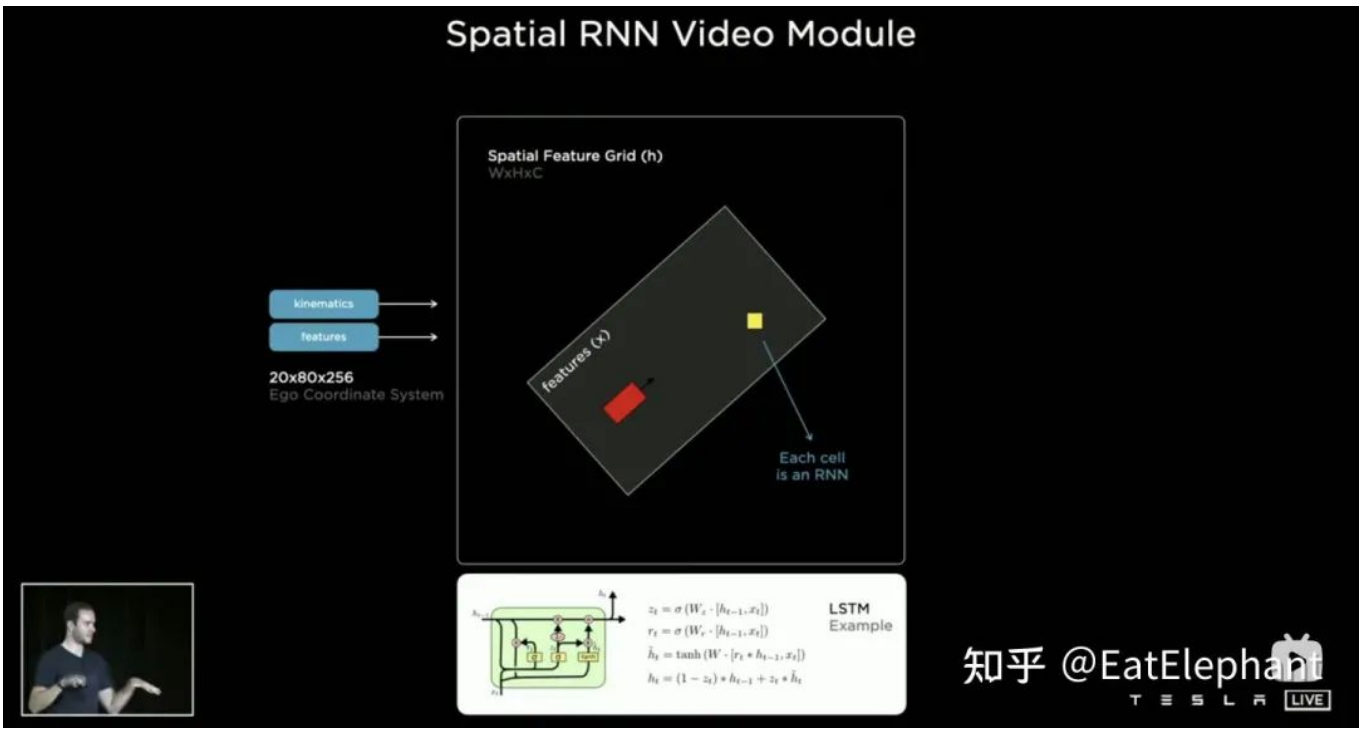


Tesla同时尝试了三种时序融合方案：3D CNN， Transformer， RNN



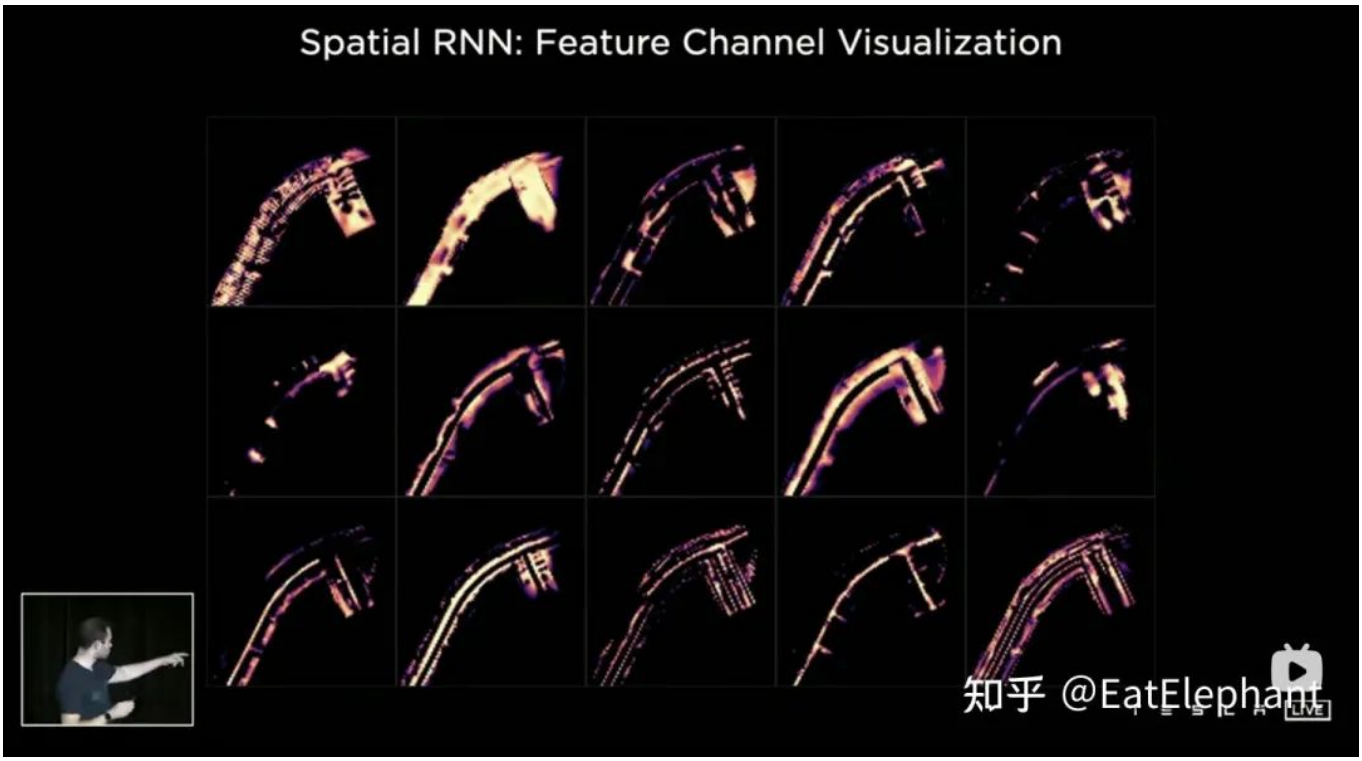
特征队列在BEV视觉特征的基础上叠加了自车运动学信息和位置编码信息

3D Conv， Transformer，RNN都能处理序列信息，三者在不同任务上各有长短，但大部分时间采用哪个方案其实区别不大，然而AI Day上Karparthy另外分享了一个简单有效，而且效果十分有趣可解释的方案叫做Spatial RNN。与上面三个方法有所不同，Spatial RNN由于RNN原本就是串行处理序列信息，帧间前后顺序得以保留，因此无需将BEV视觉特征进行位置编码就可以直接给进RNN网络，因此可以看到这里输入信息就只包括20x80x256的BEV视觉Feature Map和1x1x4的自车运动信息。

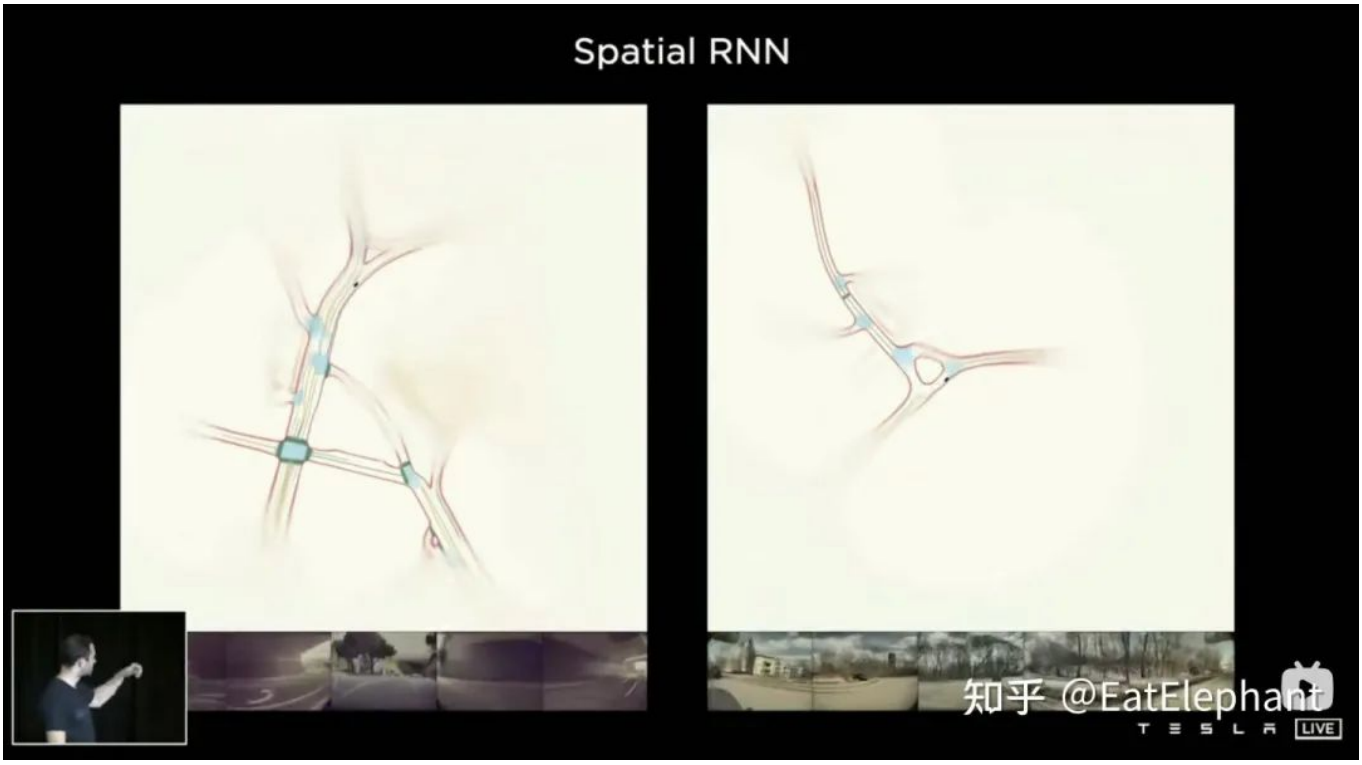


以LSTM为例实现的Spatial RNN模块

Spatial特征在CNN中常指图像平面上的宽高维度上的特征，这里Spatial RNN中的Spatial则指的是类似以某时刻的BEV坐标为基准的一个局部坐标系里的两个维度。这里为了进行说明使用了LSTM的RNN层，LSTM的优势在于其可解释性强，这里作为例子进行理解再合适不过了。LSTM特点在于Hidden State里面可以保留前面长度可变的N个时刻的状态的编码（也即短时记忆），然后当前时刻可以通过输入和Hidden State决定哪一部分记忆的状态需要被使用，哪一部分需要被遗忘等等。在Spatial RNN中，Hidden State是一个比BEV栅格空间更大的矩形栅格区域，尺寸为（WxHxC）（见上图，WxH大于20x80的BEV尺寸），自车运动学信息决定前后BEV特征分别影响的是Hidden State的哪一部分栅格，这样连续的BEV数据就会不断对Hidden State的大矩形区域进行更新，且每次更新的位置与自车运动相符合。这样经过连续的更新后，就形成了一个类似局部地图一样的Hidden State Feature Map如下图所示。

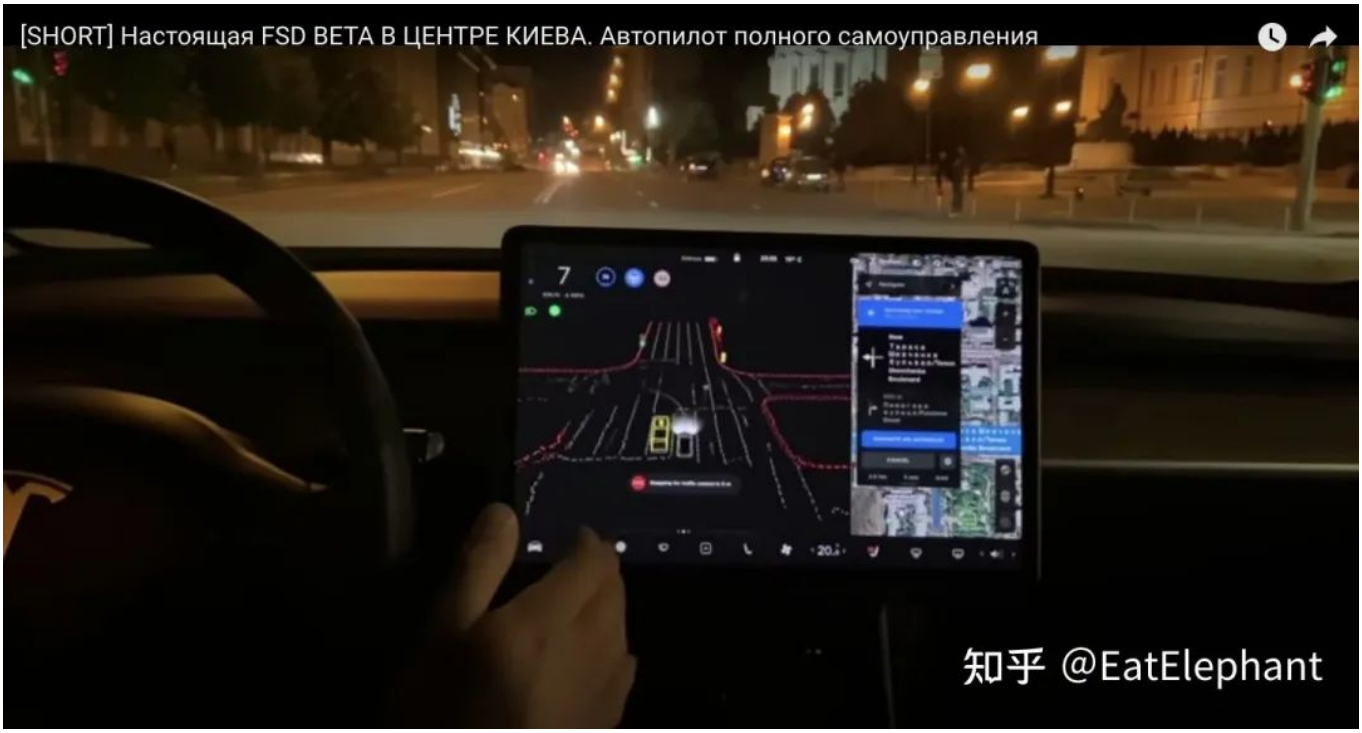


Spatial RNN中提取的同一Hidden State的部分特征图层的动态更新过程



利用BEV感知和时序队列进行处理可以End to End进行局部地图的生成

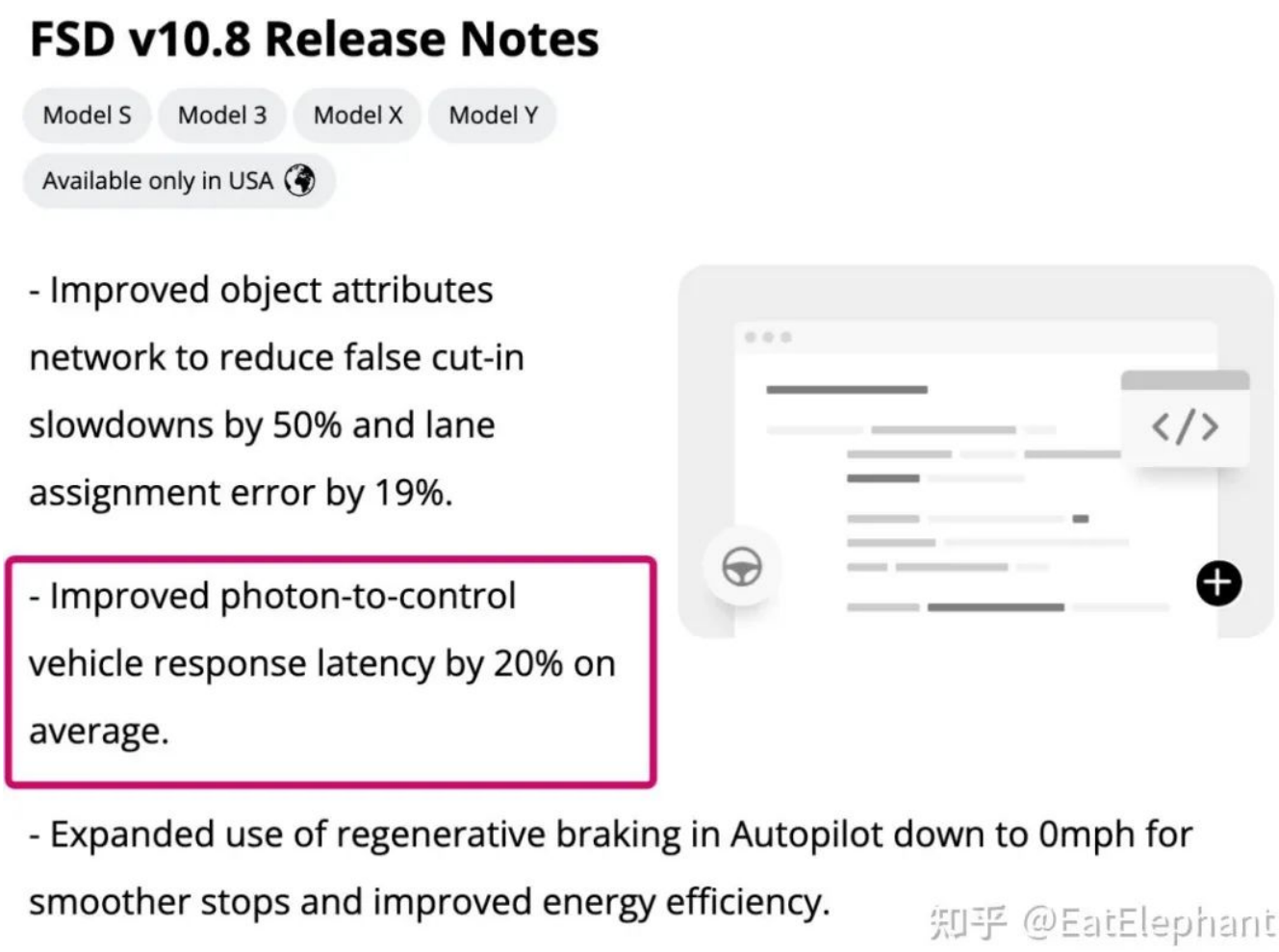
时序队列的使用赋予了神经网络获得帧间连续的感知结果的能力，与BEV结合后则使FSD获得了应对视野盲区和遮挡，选择性地对局部地图进行读写的能力，正因为有了这样的实时的局部地图构建的能力，FSD才能不依赖高精地图进行城市中的自动驾驶。



乌克兰黑客通过破解FSD功能在基辅市中心开启了FSD功能，证明了FSD可以在首次到达一个新的城市的情况进行驾驶

One More Thing：Photon to Control，Tesla用以替代ISP的秘密武器

AI Day非常详细地介绍了Tesla FSD的感知技术栈，然而AI Day的时间有限，必定不可能覆盖到FSD所使用的所有感知技术。如果有关注FSD Beta测试的同学会在FSD Release Note中发现每个版本都有这么一条不是很好理解的特征说明，如下图所示：



FSD Release Note中出现了photon-to-control一项

这里边photon-to-control字面意思就是从光子到车辆控制，但是其具体含义即使是像我这样的自动驾驶行业的从业者也是一时摸不到头脑，似乎并没有任何一家自动驾驶公司使用过类似的技术，甚至这个技术的具体含义也无从得知。

好在前一段时间我非常喜欢的Podcast主播Lex Fridman在他的节目中与Elon Musk进行了一次长达两个半小时的访谈，在这个访谈里面Elon谈到了photon-to-control才使得这一Tesla独有的视觉处理技术浮出水面。

原来大多数自动驾驶公司使用ISP（Image Signal Processing）模块在传感器获取原始图像数据后对图像进行如白平衡，动态范围调整，滤波等操作以获得最佳质量的图像，再将调整后的图像送入感知模块进行处理。这一步对于自动驾驶感知模块十分重要，因为我们的人眼是一个复杂无比的视觉传感器，能够根据周围环境的实际情况对接收的图像进行动态调节，例如调节曝光以适应明暗变化，调节聚焦以专注在不同距离的物体等等，而自动驾驶系统则通过ISP模块来尽量的让相机性能接近人眼。然而ISP最初的目的是为了在多变的外部环境下

获得一张好看的图片，但什么是自动驾驶最需要的图片形式，目前仍没有定论，这就使得ISP在处理图像数据的时候有时候无从下手，更多时候只能依据人的判断来调节ISP的处理效果。

然而Musk认为相机采集的数据最原始的形式是通过CMOS传感器对通过镜头到达传感器的可见光光子（photon）进行计数，无论ISP采用何种处理方法，本质上会丢掉部分原始图像信息。因此不知道从什么时候开始，Tesla开始尝试使用原始photon计数获得的图像用作神经网络输入，这样不仅节省了ISP所需消耗的算力，减少了延迟，还最大程度的保留了传感器获取的信息。根据Musk所说，在原始的photon计数图像上，即使在光线很弱的黑夜，行人反光也会造成微小的photon变化，最终效果是使用photon作为输入的感知系统不仅响应更快，延迟更低，还能获得远超人眼的超长夜间视距，如科幻小说一般实力强大。

这一切最初听起来有点科幻，但是在Tesla FSD的正式Release Note上找到photon-to-control的相关更新内容无疑证明了这就是Tesla已经应用在其产品功能中的技术。当然可以想像这样的photon原始数据可能很难用传统方法进行标注，因此短时间内可能还不能完全取代ISP，但是Tesla在这些前沿方向的尝试对于行业里其他从业者也能起到巨大的启发作用。