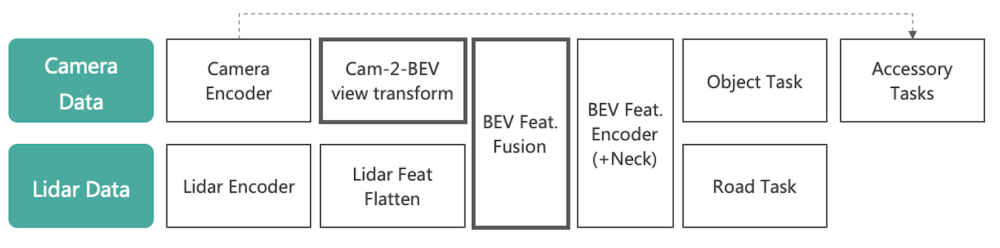
# BEV模型实验和优化Summary

此文档主要维护BEV Model Research的实验结果和优化讨论，不包含传感器适配、工程集成、鲁棒性测试等BEV模型落地任务。

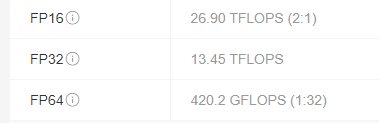
# Unified BEV Perception Pipeline



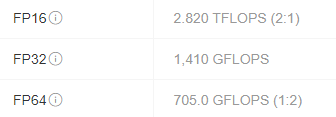
# 实验结果汇总（组件｜精度｜性能）

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 模块 | 实验模型 | BM-BEV(Camera Only) | | | BM-BEV(Camera+Lidar) | | FCOS3D | | | BEVFusion(MIT)（Camera Only） | | |
| 测试平台 | 2080Ti | | | Xaiver | | 2080Ti | | | 2080Ti | | |
|  | 组件 | GFLOPs | Params(M) | 组件 | Timing（s） | 组件 | GFLOPs | Params(M) | 组件 | GFLOPs | Params(M) |
| Image Encoder | Input Size | [256, 704] | / | / | [256, 704] |  | [256, 704] |  |  | [256, 704] | / | / |
| Image Backbone | EfficientNet-B3 | 0.556 | / | EfficientNet-B3 | 0.1447 | EfficientNet-B3 | 0.678 |  | SwinTransformer | 108.238 | 27.497 |
| Image Neck | Up | 20.1 | / | Up | FPN | 4.602 |  | GeneralizedLSSFPN | 15.673 | 1.591 |
| Camera->BEV | LSS | 0 | / | LSS | 0.3294 | / | / |  | LSSTransform | 6.546 | 0.224 |
| Lidar Encoder | Lidar Backbone | / | / | / | PointPillars | 1.0661 | / | / |  | / | / | / |
| BEV Encoder | BEV Input Size | [1, 64, 512, 512] | / | / | [1,128, 512, 512] | / | / | / |  | [1, 80, 128, 128] | / | / |
| BEV Backbone | SECOND | 36.346 | / | SECOND | 0.0152 | / | / |  | GeneralizedResNet | 12.987 | 11.039 |
| BEV Neck | SECOND FPN | 1.361 | / | SECOND FPN | 0.0042 | / | / |  | LSSFPN | 12.774 | 1.345 |
| Task | 3D Detection | CenterHead | 26.145 | / | CenterHead | 0.1769 | FCOSMono3DHead | 37.356 |  | CenterHead | 24.937 | 1.52 |
| MAP | / | 27.281 | | / | 46.216 | / |  | | / | 33.25 | |
| NDS | / | 33.572 | | / | 55.023 | / |  | | / | 40.15 | |
| Segmentation |  |  |  |  |  |  |  |  |  |  |  |
| GFLOPs(all) | | / | 84,74 | 23.03 | / |  |  | 42.66 | 20.95 | / | 181.16 | 43.24 |
| FPS（img/s） | | / | 6.1 | | / | 0.57 |  | 5.5 | | / | 7.4 | |

**2080Ti算力:** <https://www.xincanshu.com/gpu/NVIDIA_GeForce_RTX_2080_Ti/canshu.html>



**AGX Xaiver算力:** <https://www.xincanshu.com/gpu/NVIDIA_Jetson_AGX_Xavier_GPU/canshu.html>



# 对比结论/猜想

* ~~Lidar对精度上限的明显提升~~

* 更强的Image Encoder和更大的input size对于识别精度提升的重要性

* BEV特征的尺寸（信息量）对于最终识别精度的影响有限

* 性能瓶颈（暂只针对BEV底座）：点云处理、Cam2BEV变换、Image Encoder

# 后续优化点

#### 识别精度

1. View Transformation（LSS方式&Transformer方式）

* 1. BM-BEV的image backbone相对于其他组件来说有点过于弱，需要替换性能更优的backbone进行重新实验——**替换Image Backbone（如Swin Transform）**

* 1. BM-BEV现有的图像分辨率为256\*704，对于某些远处的目标检测不是很好，将输入图像恢复到1600\*900——**提升图像分辨率（依赖服务器GPU显存）**

* 1. 使用在深度估计模型预训练好的Backbone与Neck来替换现有的图像的Backbone与Neck，会有不错的提点，可以参考BEVDet——**在深度估计数据上预训练**

* 1. BM-BEV的Camera->BEV时，并未对BEV视角下的特征分层，直接生成了[Batch\_Size, 64, 512, 512]大小的特征图，这种情况会丢失一些高度信息，可以生成[Batch\_Size, 4, 64, 512, 512]大小的特征图，4代表的是将高度分为4个层次，后续处理可以将高度信息在channel维度进行concat处理——**Camera->BEV时，生成分层的BEV特征**

* 1. 应用Transformer：做Cam2BEV时通过LUT方式对BEV Queries做优化，避免全局做Cross Attention -- 参考GKT

1. BEV Feature Fusion

* 1. BEV特征图的大小于图像的输入尺寸有着较大关系，越高的图像分辨率输入，对应生成的伪点云就越多，因此在拍平到BEV特征时，BEV特征图越大损失的信息越少，需要有个平衡——**平衡BEV特征图大小与分辨率尺寸之间的关系**

* 1. 不应用Transformer：BEV特征融合的组件设计 -- **参考BEVFusion（Alibaba）和M2BEV（Nvidia）**

* 1. 添加合适的时序模型进BM-BEV框架，待前述模型优化完成或着替换Backbone为较重的模型后，对时序融合框架进行调整优化，可以参照之前设计好的方案二——**时序特征融合研究**

1. Lidar feature encoding (TBD)

1. Task head design (TBD)

#### 算力层面

* PointPillars模型在端侧需要进行加速，现阶段的点云处理方法存在较多的矩阵乘法计算

* Camera Feature从2D -> 3D时，存在升维操作，计算量较大，需要有一些更加高效但不损失性能的方法，可参考论文如下所示

* + M2BEV，不进行升维操作，直接将2D特征平铺到三维空间中

* + BEVFusion(MIT)

* 涉及到较多的相机的内外参变换，端侧加速性能不是很好

* 现有模型的各个组件需要针对性的优化，更换某些Backbone为端侧支持更优的模型，如RepVGG

* 如果使用Transformer作为时序融合模型，可以考虑将BEV特征划分Patch后先进行下采样再经过Transformer进行跨模态的信息融合与时序特征的融合，具体可以参考时序设计方案二

* 针对特定硬件的加速，如转换到TRT@Nvidia devices