Calculate the FLOPs for V2X models

General Formula for FLOPs in a Convolutional Layer

For a convolutional layer, the number of FLOPs can be calculated as:

FLOPs=2×K^2×Cin×Cout×Hout×Wout

Where:

- K is the kernel size
- Cin and Cout are the number of input and output channels, respectively
- Hout and Wout are the height and width of the output feature map

Note: You can find the code for FLOPs Calculation is on GitHub

---

V2X-ViT

Table T2: **Detailed architectural specifications for V2X-ViT.**

	Output size	V2X-ViT framework
PointPillar Encoder	$M \times 352 \times 96 \times 256$	[Voxel samp. reso. 0.4m, Scatter, 64] [Conv3x3, 64, stride 2, BN, ReLU] $\times 3$ [Conv3x3, 128, stride 2, BN, ReLU] $\times 5$ [Conv3x3, 256, stride 2, BN, ReLU] $\times 8$ [ConvT3x3, 128, stride 1, BN, ReLU] $\times 1$ [ConvT3x3, 128, stride 2, BN, ReLU] $\times 1$ [ConvT3x3, 128, stride 4, BN, ReLU] $\times 1$
	$M \times 176 \times 48 \times 256$	[Concat3, 384] [Conv3x3, 256, stride 2, ReLU Conv3x3, 256, stride 1, ReLU] $\times 1$
Delay-aware Pos. Encoding	$M \times 176 \times 48 \times 256$	[sin-cos pos. encoding] [Linear, 256] $\times 1$
Transformer Backbone	$M \times 176 \times 48 \times 256$	[HSMA, dim 256, head 8 MSwin, dim 256, head $\{16, 8, 4\}$, ws. $\{4 \times 4, 8 \times 8, 16 \times 16\}$ MLP, dim 256] $\times 3$
Detection Head	$176 \times 48 \times 16$	Cls. head: [Conv1x1, 2, stride 1] Regr. head: [Conv1x1, 14, stride 1]

V2X-ViT has 4 modules, we will need to calculate for each module and sum them up for total FLOPs

PointPillar Encoder:

- [Conv3x3, 64, stride 2, BN, ReLU] x 3 :
  - Kernel size (K) = 3x3
  - Input channels (Cin) : 256
  - Output channels (Cout) = 64
  - Stride = 2

- Output feature map size (using stride 2 and output size)
  - H_{out} = 253/2 = 176
  - W_{out} = 96/2 = 48

=> FLOPs per layer = 2×9×256×64×176×48

=> Total FLOPs for 3 repetitions = 3×(FLOPs per layer)

=> With M agents, total FLOPs = M×3×(FLOPs per layer)

You can apply the same method to calculate FLOPs of the rest of PointPillar Encoder layers, note that the out put from previous layer is the input for current layer.

Delay-aware Pos. Encoding:

Negligible

Transformer Backbone:

- HSMA: Each attention head operation is assumed to have the following FLOPs:
  - Query, Key, Value Computation: N×(256/8)^2×2
  - Attention Score Calculation: N×N×(256/8)
  - Output Calculation: N×(256/8)×(256/8)
  - N is the number of tokens (176×48 = 8448 in this case)
  - Given that there are 8 heads, we multiply the FLOPs for one head by 8
- MSwin: The calculation for each scale of MSwin is similar to HSMA, with adjustments made for window sizes. Assume the FLOPs are the same as one HSMA head for each scale
- MLP: we calculate FLOPs as 2×256×256×176×48

Total FLOPs for Transformer Backbone is the sum of HSMA, MSwin, and MLP, the multiply be 3 to account for 3 repetitions in the transformer, then multiply by the number of agents M

Detection Head:

Negligible

My hand-written FLOPs calculation for V2X-ViT:

Assume we have 2 agents

$$
\left.\begin{aligned}
\text{FLOPs}:\ & M\times 2\times 9\times 256\times 64\times 176\times 48\times 3\ (\sim 7.5\,\text{B})\\
&+M\times 2\times 9\times 64\times 128\times 176\times 48\times 5\ (\sim 6.2\,\text{B})\\
&+M\times 2\times 9\times 128\times 256\times 176\times 48\times 8\ (\sim 39.9\,\text{B})\\
&+M\times 2\times 9\times 256\times 128\times 352\times 96\ (\sim 19.9\,\text{B})\\
&+M\times 2\times 9\times 128\times 128\times 176\times 48\ (\sim 2.5\,\text{B})\\
&+M\times 2\times 9\times 128\times 128\times 88\times 24\ (\sim 0.6\,\text{B})\\
&+M\times 2\times 9\times 256\times 256\times 88\times 24\ (\sim 2.5\,\text{B})\\
&+M\times 2\times 9\times 256\times 256\times 176\times 48\ (\sim 10.0\,\text{B})\\
&+M\times 2\times 1\times 256\times 256\times 176\times 48\ (\sim 1.1\,\text{B})
\end{aligned}\right\}\text{Point Pillar Encoder}
$$

$$
\left.\begin{aligned}
+3\times\{\ & 8\times[(8448\times(256/8)^2\times 2)+(8448\times 8448\times 256/8)+(8448\times(256/8)^2)]\\
&+16\times[(8448\times(256/8)^2\times 2)+(8448\times 8448\times 256/8)+(8448\times(256/8)^2)]\\
&+8\times[(8448\times(256/8)^2\times 2)+(8448\times 8448\times 256/8)+(8448\times(256/8)^2)]\\
&+4\times[(8448\times(256/8)^2\times 2)+(8448\times 8448\times 256/8)+(8448\times(256/8)^2)]\\
&+2\times 256\times 256\times 176\times 48\}\ (\sim 253.2\,\text{B})\times M
\end{aligned}\right\}\text{Transformer Backbone}
$$

$$
\left.\begin{aligned}
&+2\times 1\times 16\times 2\times 48\times 176\\
&+2\times 1\times 2\times 14\times 176\times 48
\end{aligned}\ (\text{small})\right\}\text{Detection Head}
$$

$$
M=2 \Rightarrow \text{Total FLOPs} = \boxed{686.8\ \text{B}}
$$

CoBEVT

Table A2: Detailed architectural specifications of CoBEVT for OPV2V camera track. M represents the number of cameras and N is the number of agents.

	Output size	CoBEVT framework
ResNet34 Encoder	$N\times M\times 64\times 64\times 128$	[ResNet34-layer1]
	$N\times M\times 32\times 32\times 256$	[ResNet34-layer2]
	$N\times M\times 16\times 16\times 512$	[ResNet34-layer3]
SinBEVT Backbone	$N\times 128\times 128\times 128$	[FAX-CA, dim 128, head 4, bev win. sz.$\{16\times 16\}$ feat win. sz.$\{8\times 8\}$ MLP, dim 256 Res-Bottleneck-block $\times 2$] $\times 1$
	$N\times 64\times 64\times 128$	[FAX-CA, dim 128, head 4, bev win. sz.$\{16\times 16\}$ feat win. sz.$\{8\times 8\}$ MLP, dim 256 Res-Bottleneck-block $\times 2$] $\times 1$
	$N\times 32\times 32\times 128$	[FAX-CA, dim 128, head 4, bev win. sz.$\{32\times 32\}$ feat win. sz.$\{16\times 16\}$ MLP, dim 256 Res-Bottleneck-block $\times 2$] $\times 1$
FuseBEVT Backbone	$N\times 32\times 32\times 128$	[FAX-SA, dim 128, head 4, win. sz.$\{8\times 8\}$ MLP, dim 256] $\times 3$
Decoder	$64\times 64\times 128$	[Bilinear-upsample, Conv3x3, BN]
	$128\times 128\times 64$	[Bilinear-upsample, Conv3x3, BN]
	$256\times 256\times 32$	[Bilinear-upsample, Conv3x3, BN]
	$256\times 256\times k$	Dyna. Obj. head: [Conv1x1, 2, stride 1] Stat. Obj. head: [Conv1x1, 3, stride 1]

CoBEVT has 4 modules, we will calculate the FLOPs for each and add them up

ResNet34 Encoder:

Base on my research, ResNet34 has 3.6 GFLOPs for input image of 224x224

CoBEVT input is 512x512 => ResNet34 GFLOPs are 4 times of 3.6

With 4 camera, and 3 ResNet34 layers => Total FLOPs = 3.6 x 4 x 4 x 3

SinBEVT Backbone:

- FAX-CA:
  - FLOPs = N x H x W x Number of heads x Output channels x (bev win + fear win)
- MLP:
  - FLOPs = N x H x W x dimension x Output channels x 2
- Res-Bottleneck:
  - FLOPs = N x H x W x Kernel size x Input channels x Output channels

FuseBEVT Backbone:

Same as SinBEVT backbone

Decoder:

Follow the general formula for FLOPs calculation

My hand-written FLOPs calculation for CoBEVT:

Assume we have 2 agents, each agent has 2 cameras