CoBEVT: Cooperative Bird’s Eye View Semantic Segmentation with Sparse Transformers

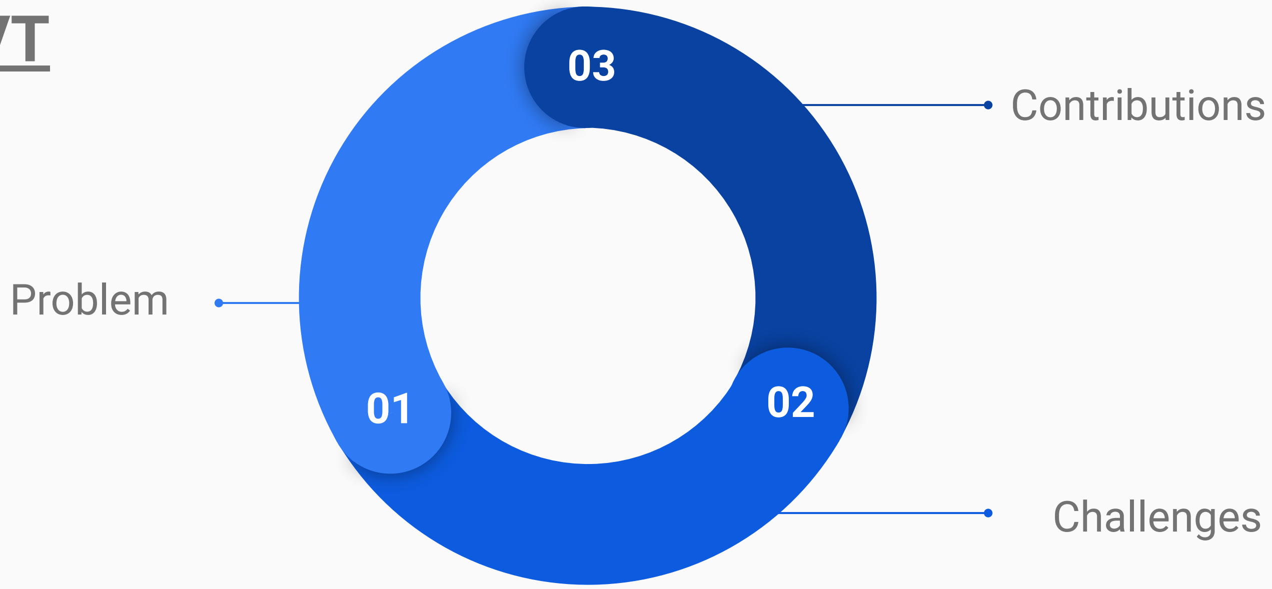
Authors: Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, Jiaqi Ma

Affiliations: University of California, Los Angeles; University of Texas at Austin; University of California, Davis

Presenter: Neil Zarghami, MSEE, UCR

Introduction

CoBEVT



Introduction

Problem

01

Addresses single-agent based systems:

- Limitations in handling occlusions
- Limitations in detecting distant objects
- Vulnerable to system malfunctions

Introduction

- Objects are hidden from single-agent systems that cannot detect beyond their sensor capabilities
- Real world traffic is highly variable and highly unpredictability
- Camera sensors often lack in distant object perception

02

Challenges

- **CoBEVT** is the first multi-agent multi-camera perception framework designed for BEV semantic labeling

03

Contributions

- Fused Axial Attention (**FAX**) Module is used for local and global spatial interactions across views and agents

- Demonstrated SOTA results on OPV2V with superior performance over existing models

Design Details

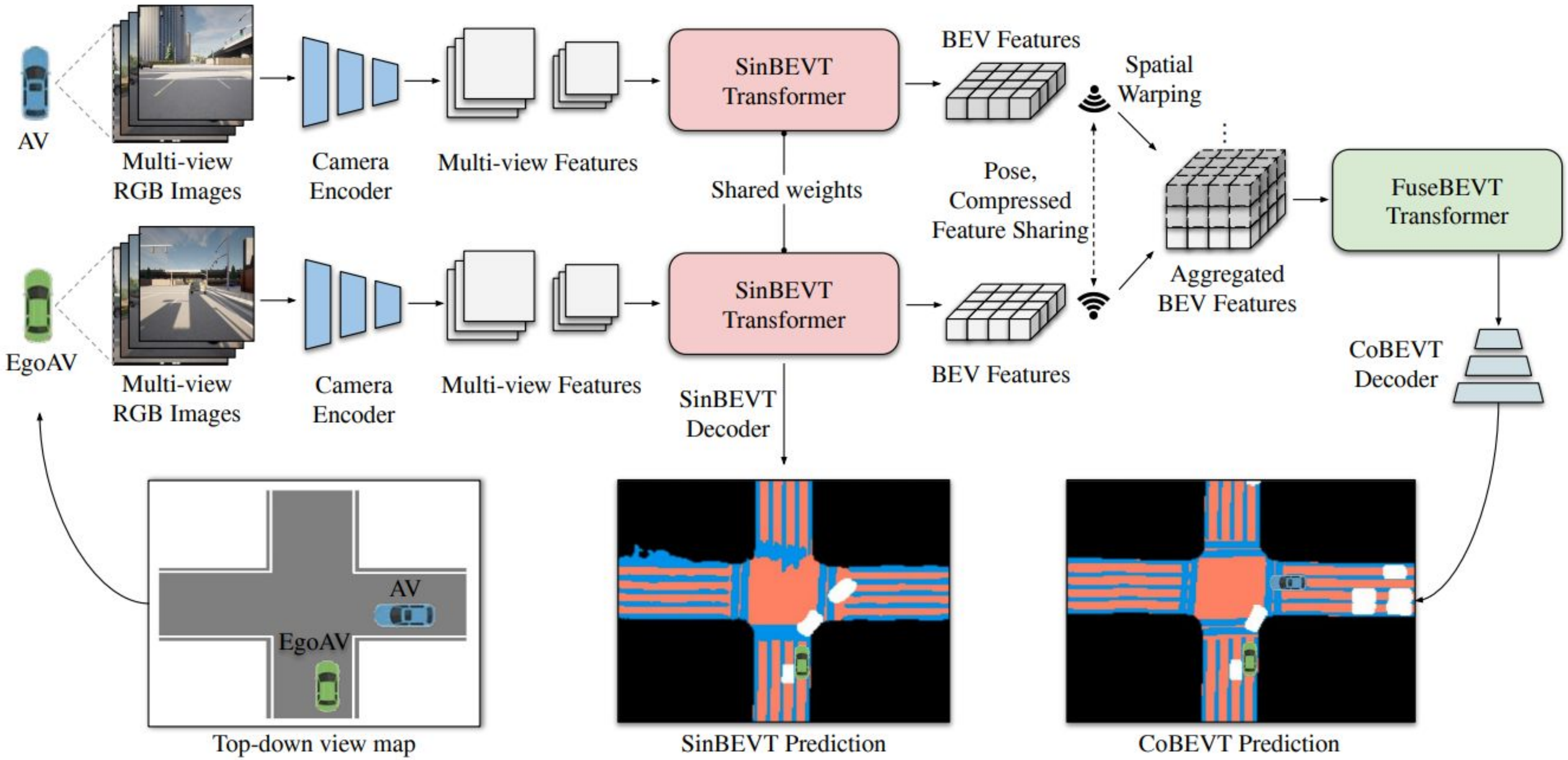


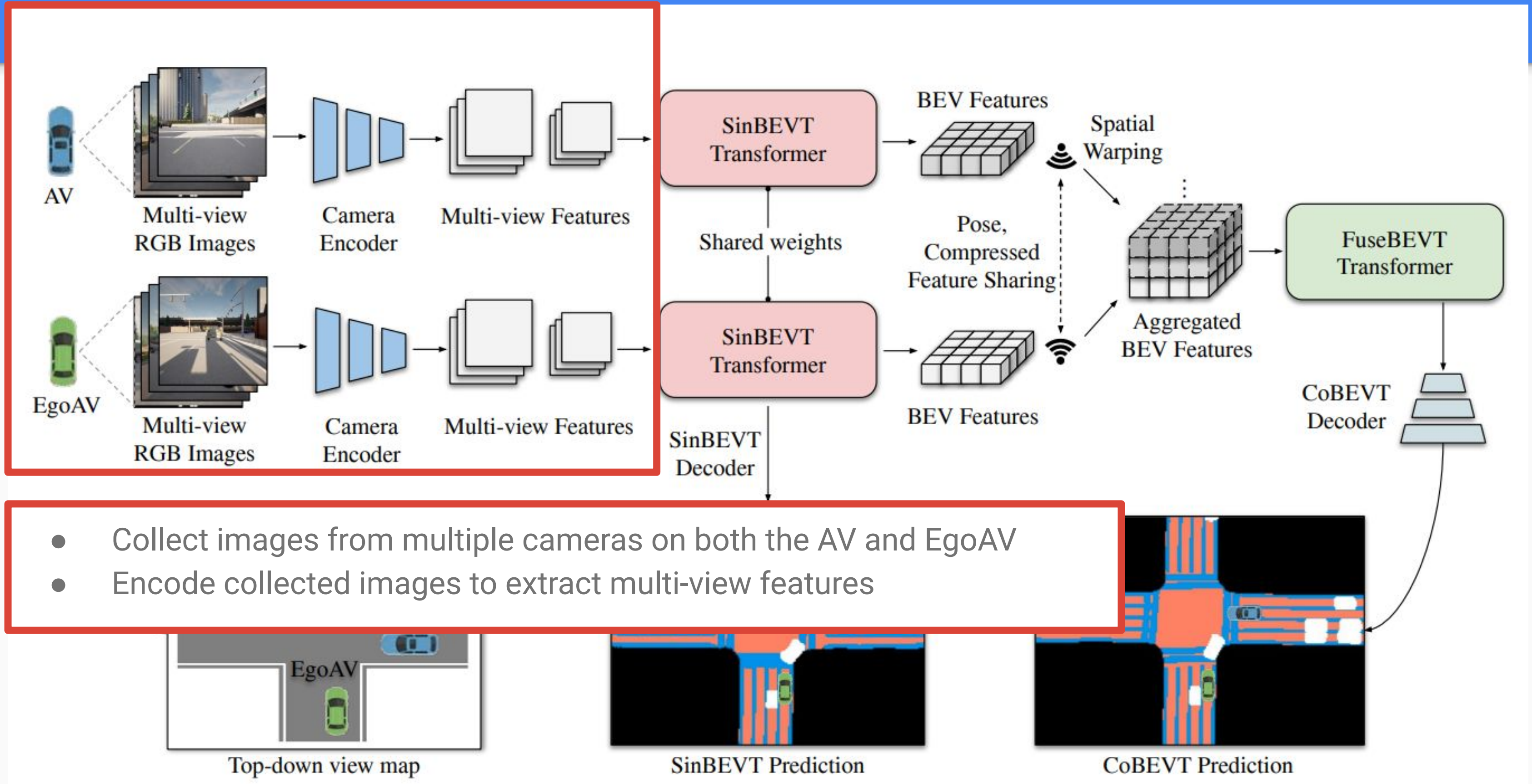
Figure 1: The overall framework of CoBEVT.

Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	**SinBEVT** for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	**FuseBEVT** for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding

Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	SinBEVT for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	FuseBEVT for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding



- Collect images from multiple cameras on both the AV and EgoAV
- Encode collected images to extract multi-view features

Figure 1: The overall framework of CoBEVT.

Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	**SinBEVT** for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	FuseBEVT for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding

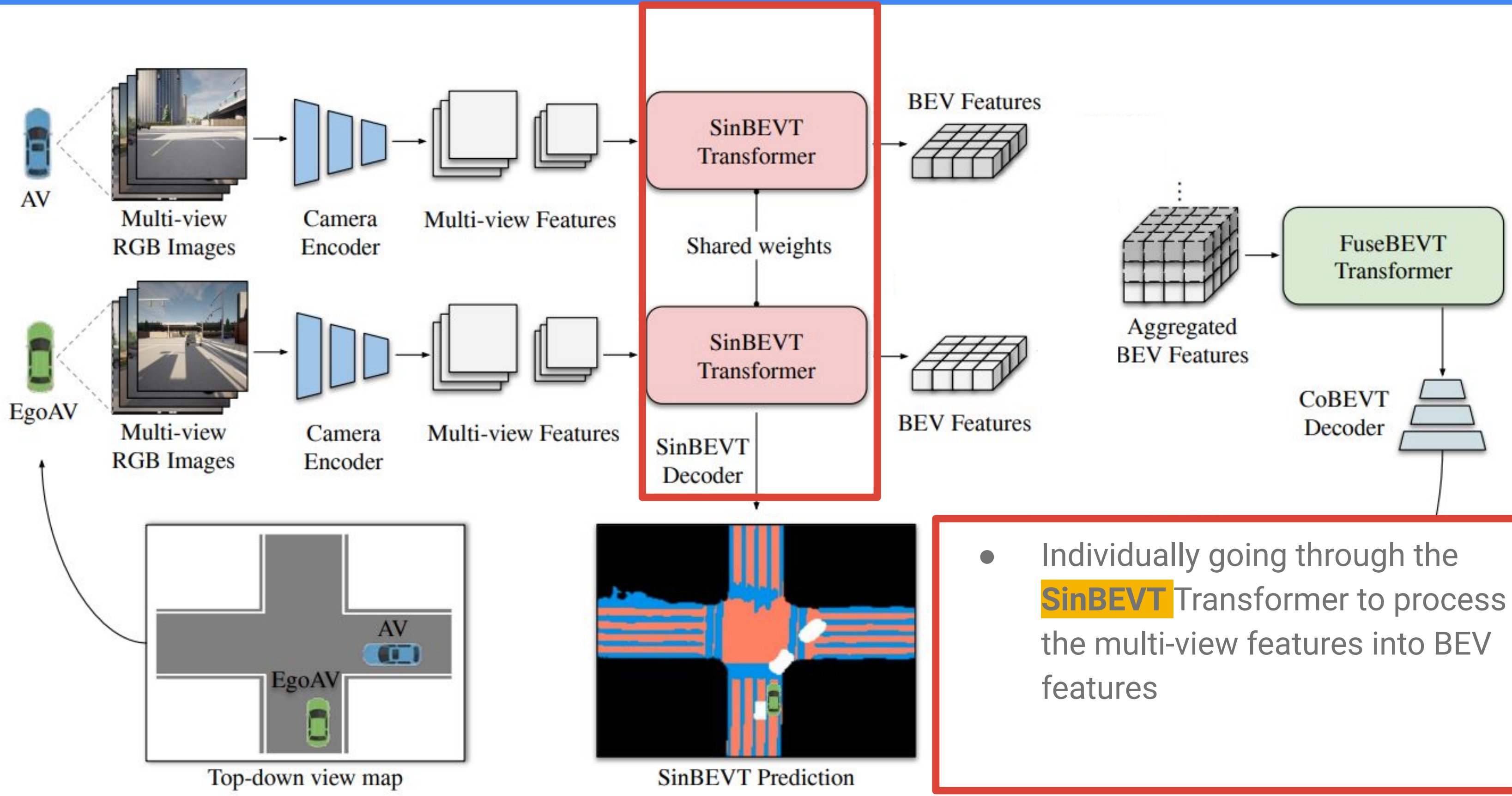
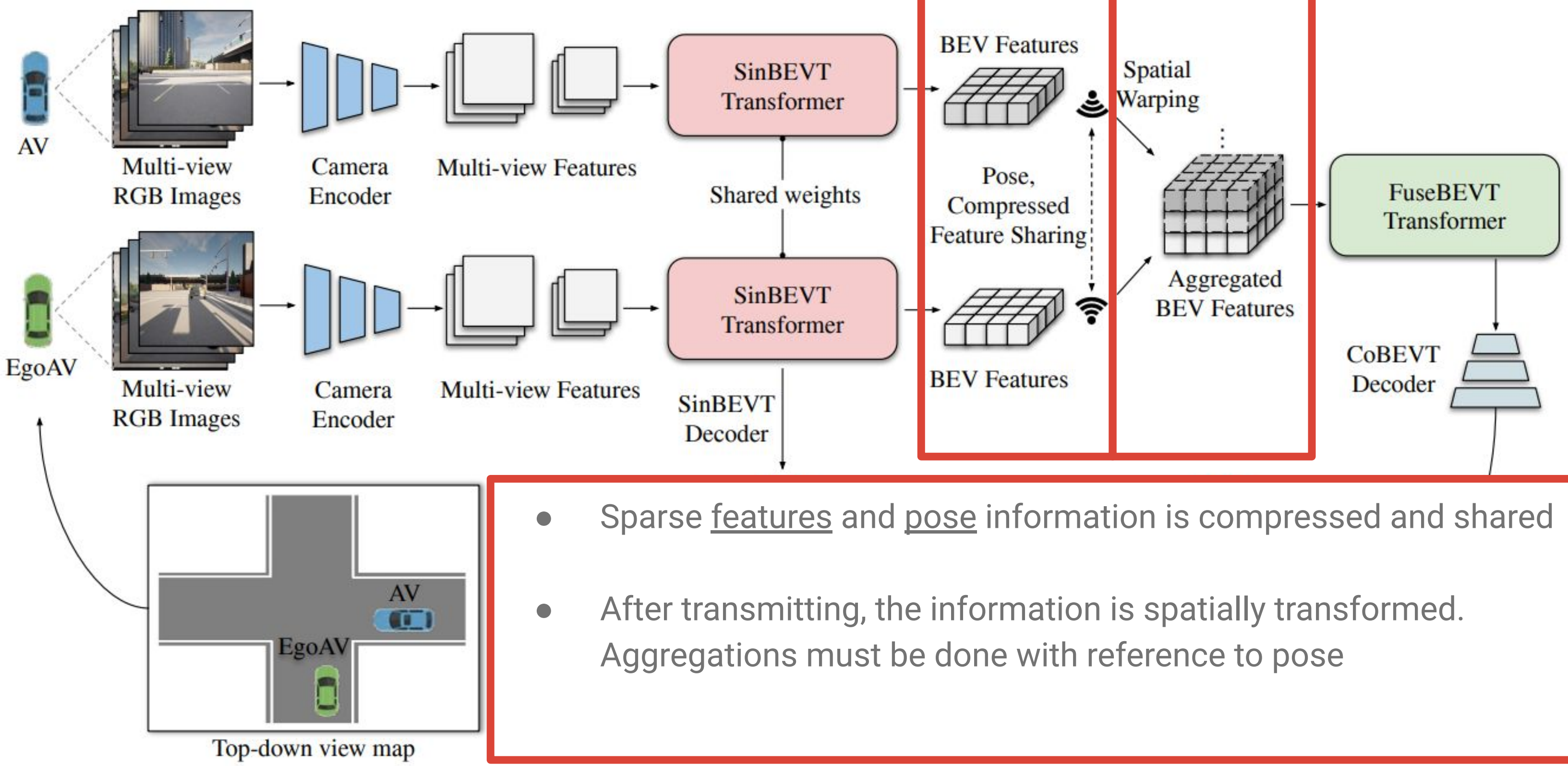


Figure 1: The overall framework of CoBEVT.

- Individually going through the SinBEVT Transformer to process the multi-view features into BEV features

Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	SinBEVT for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	FuseBEVT for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding

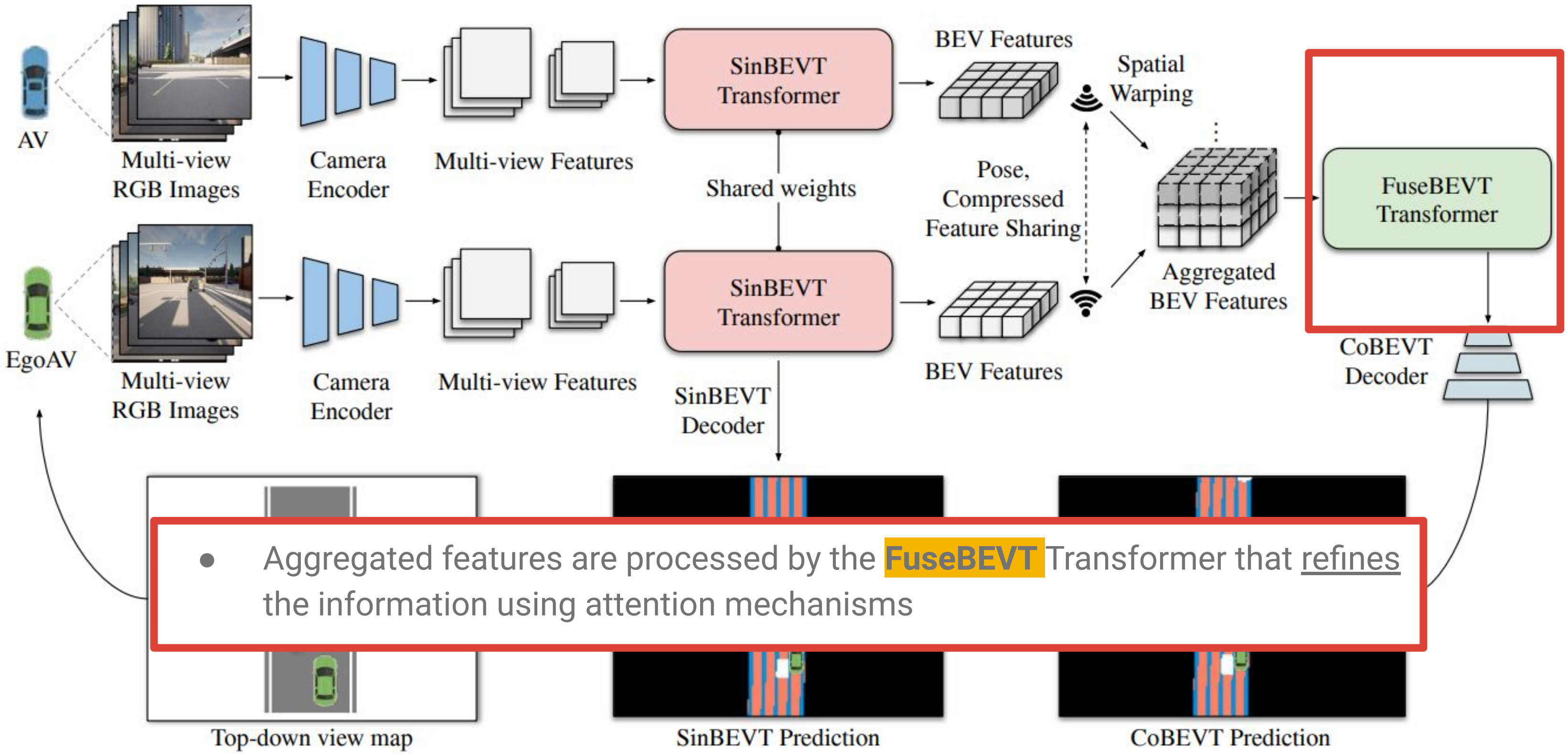


- Sparse features and pose information is compressed and shared
- After transmitting, the information is spatially transformed. Aggregations must be done with reference to pose

Figure 1: The overall framework of CoBEVT.

Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	SinBEVT for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	**FuseBEVT** for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding

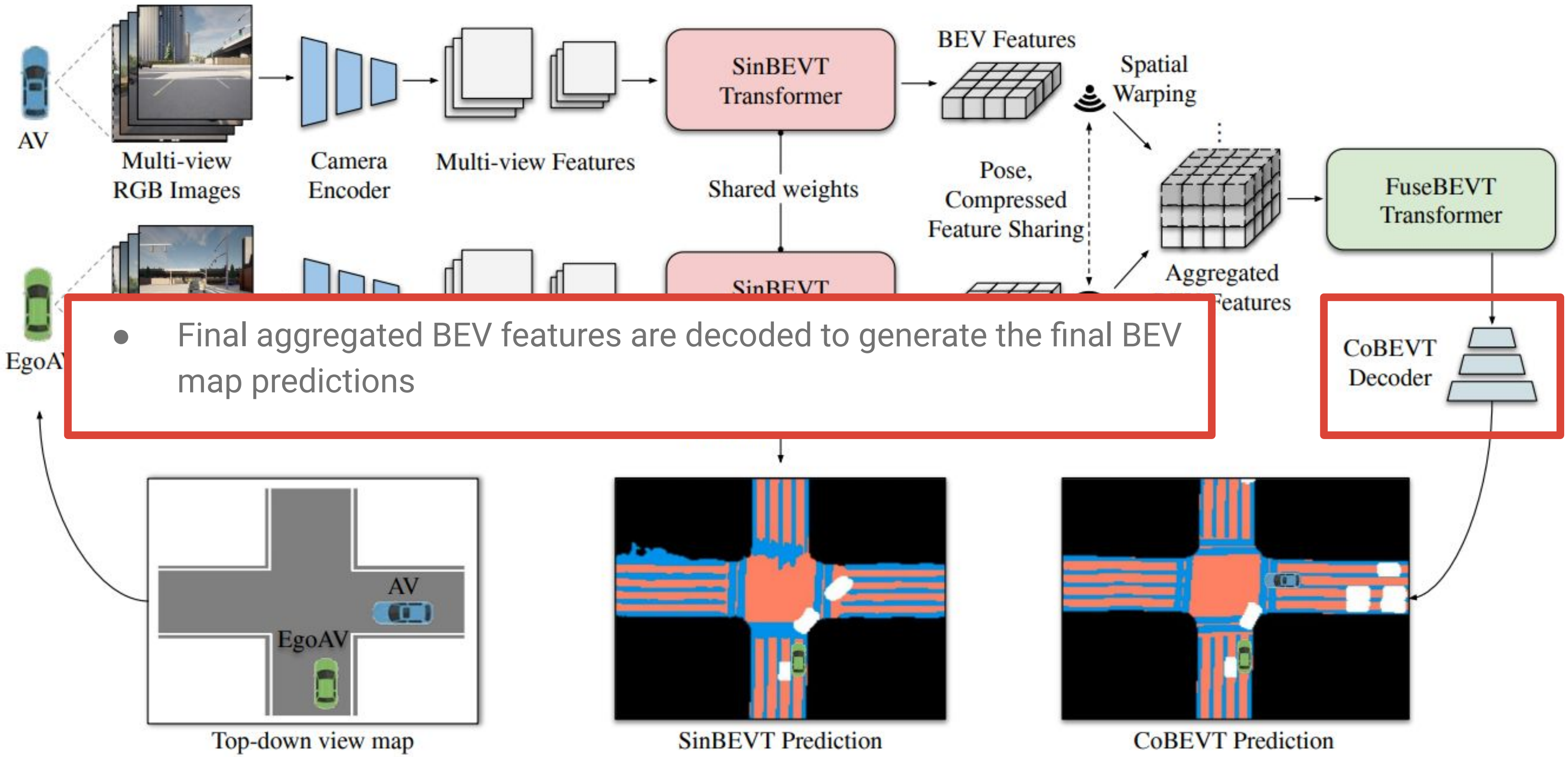


- Aggregated features are processed by the **FuseBEVT** Transformer that refines the information using attention mechanisms

Figure 1: The overall framework of CoBEVT.

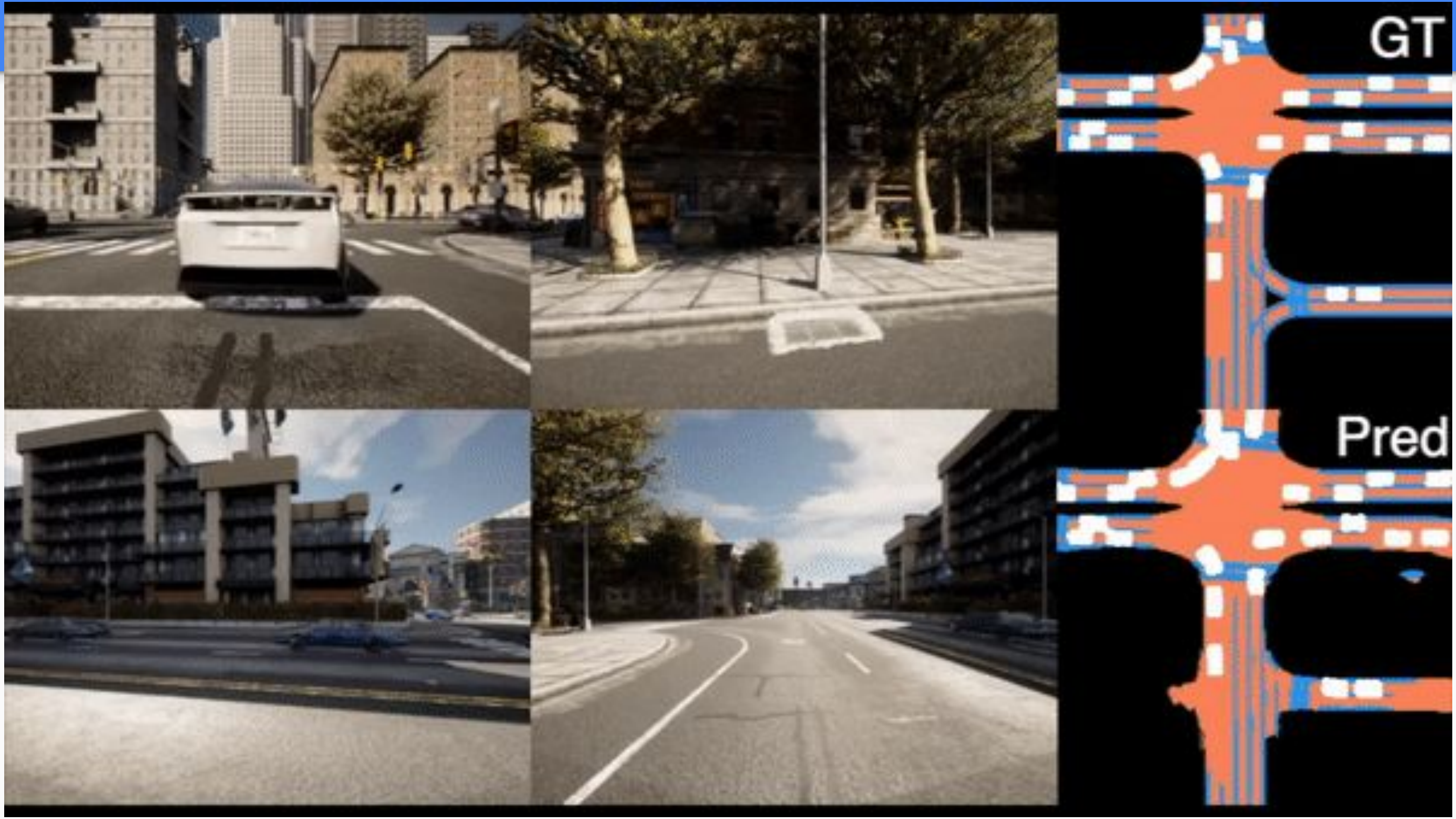
Design Details

Step 1	Step 2	Step 3	Step 4	Step 5
Multi-View Camera Inputs and Encoding	SinBEVT for Single-Agent BEV Feature Computation	Feature Sharing and Transformation	FuseBEVT for Multi-Agent BEV Feature Fusion	Final Aggregation and Decoding



- Final aggregated BEV features are decoded to generate the final BEV map predictions

Figure 1: The overall framework of CoBEVT.



OPV2V: Multi-agent BEV map prediction.

Design Details

Step 2

SinBEVT for Single-Agent BEV Feature Computation

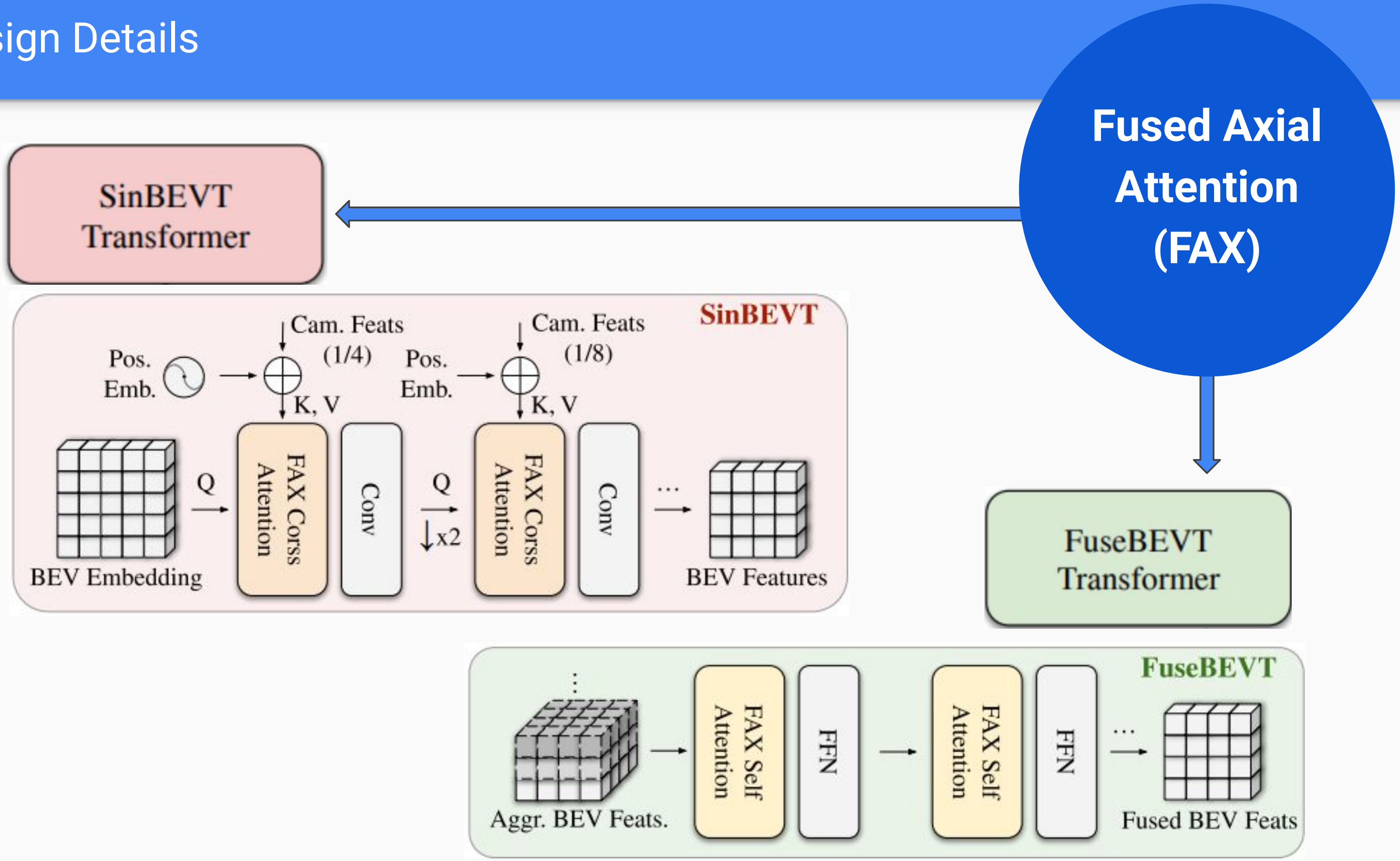
SinBEVT Transformer

Step 4

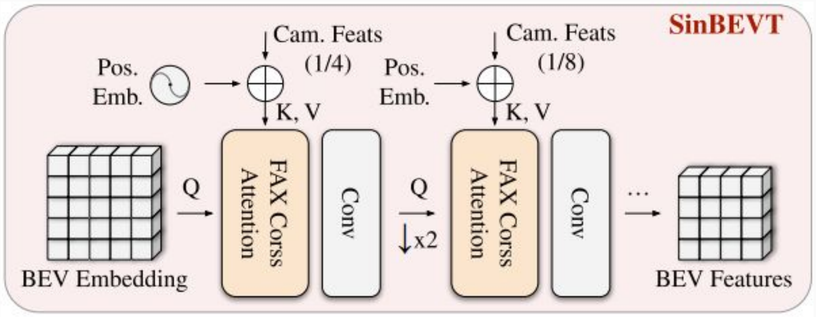
FuseBEVT for Multi-Agent BEV Feature Fusion

FuseBEVT Transformer

Design Details



Design Details



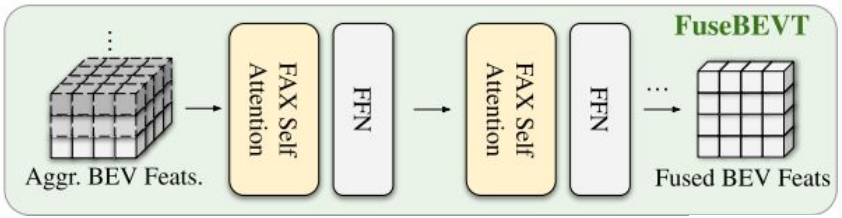
$$\text{3D-Rel-Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{B})\mathbf{V},$$

Q, K, V = query, key, value matrices projected from the input tensor

B = relative position bias

$$\hat{\mathbf{B}} \in \mathbb{R}^{(2N-1)\times(2H-1)\times(2W-1)}$$

1. Initialize BEV embedding as Q
2. Positional info and camera features are added (K&V)
3. Q*KV go through FAX to process hi-res features
4. Second pass focuses on low-res features
5. BEV Features



Preprocessing:

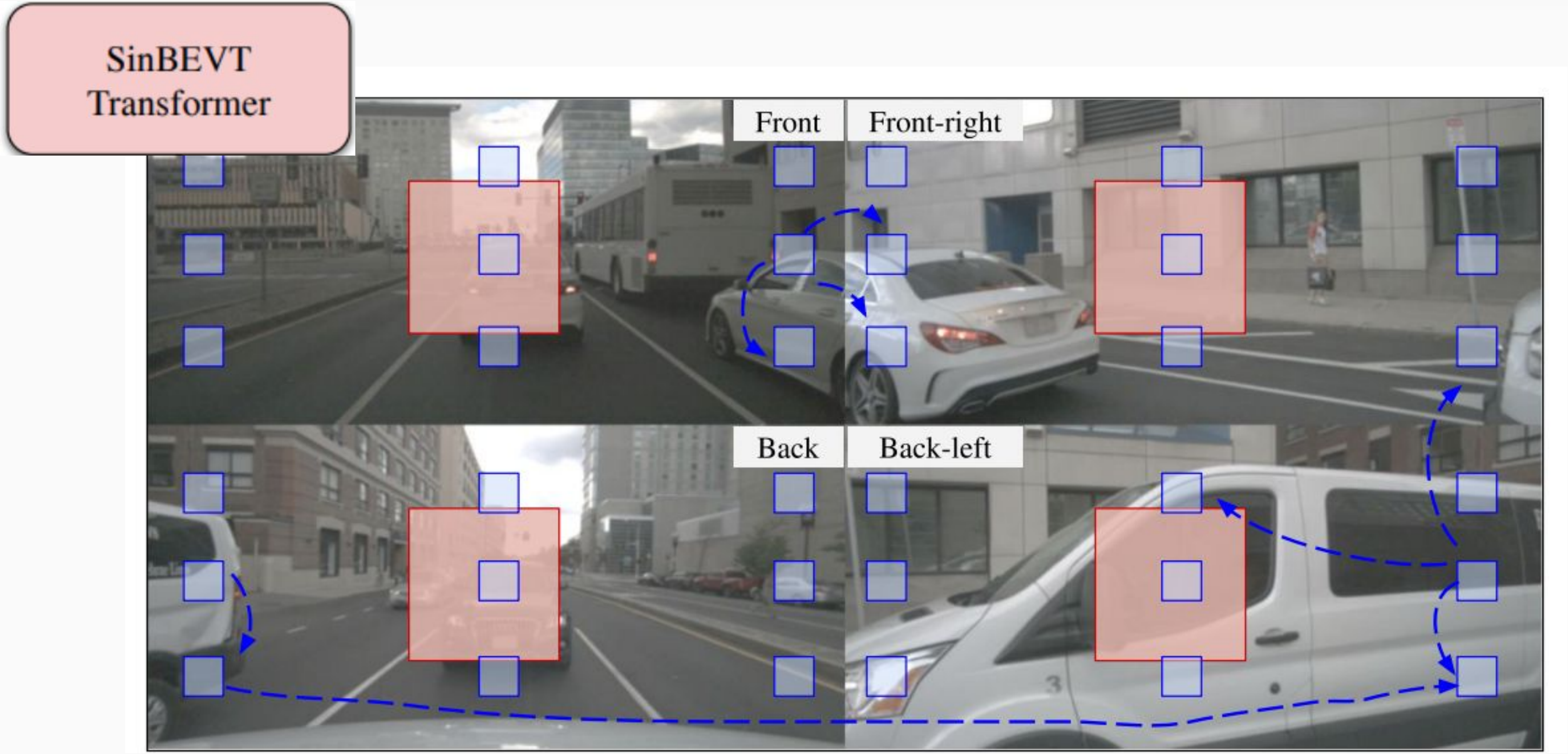
$$\text{Fused-Block}: (N,H,W,C) \to (N, \frac{H}{P}\times P, \frac{W}{P}\times P, C) \to (\frac{HW}{P^2}, \underbrace{N\times P^2}_{\text{“spatial axis”}}, C)$$

$$\text{Fused-Grid}: (N,H,W,C)\to(N, G\times\frac{H}{G}, G\times\frac{W}{G}, C)\to\underbrace{(N\times G^2, \frac{HW}{G^2}, C)\to(\frac{HW}{G^2}, N\times G^2, C)}_{\text{swapaxes(axis1=-2,axis2=-3)}}$$

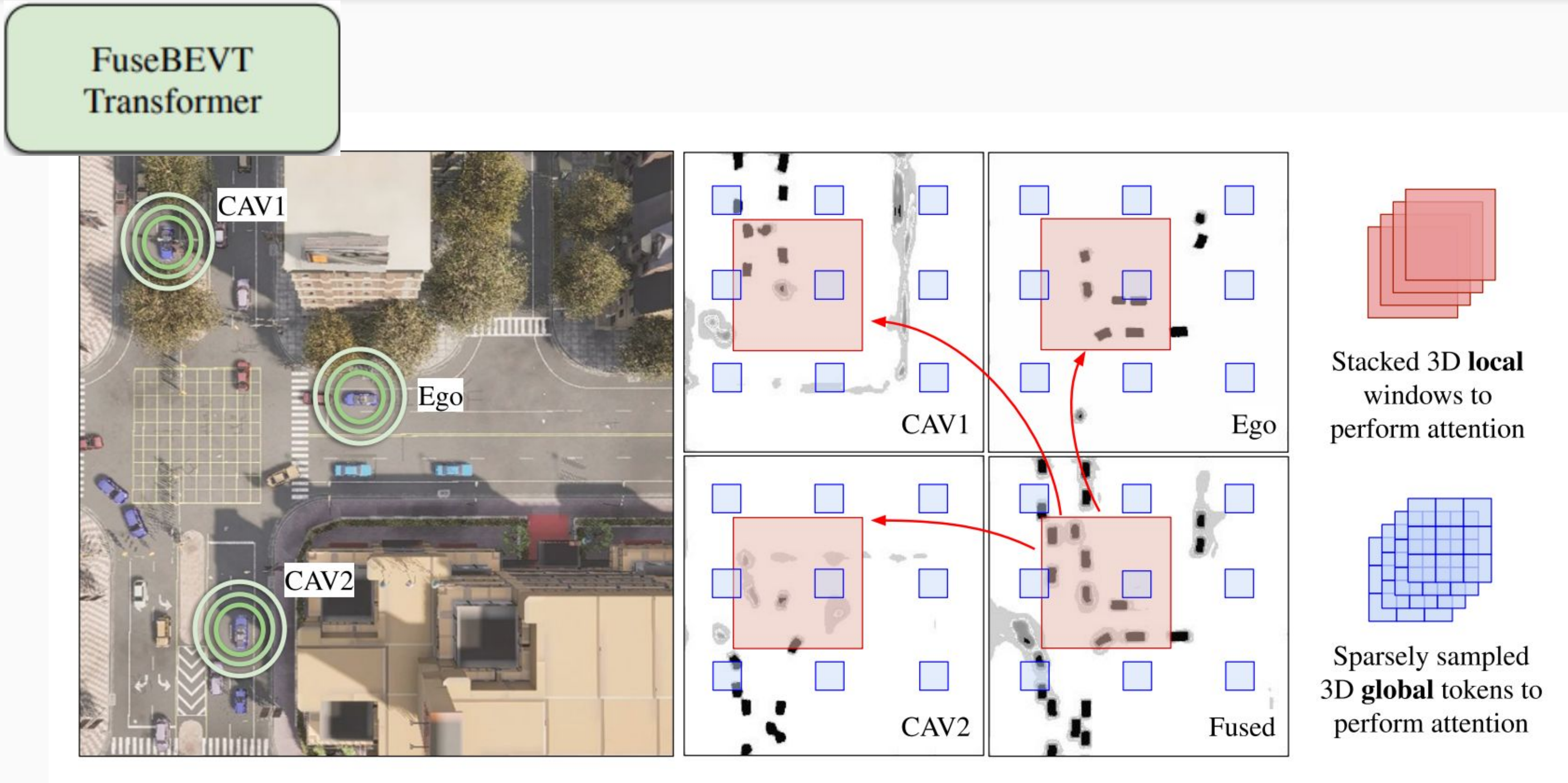
0. Preprocessing " … define the Fused-Block(·) operator with parameter P as partitioning the input 3D feature x ∈ R^{N×H×W×C} into non-overlapping 3D windows … N × P × P."

1. Aggregated BEV features
2. FAX captures local and global interactions
3. Second pass is for FAX feature enhancement and refinement
4. Fused BEV Features

Design Details



Design Details



Evaluations

Evaluations

Notes:

- CoBEVT is insensitive to compression and still beats other fusion methods @64cpr
- SinBEVT approach achieves SOTA performance with real-time inference speed
- Not limited to cameras i.e. radar, LiDAR

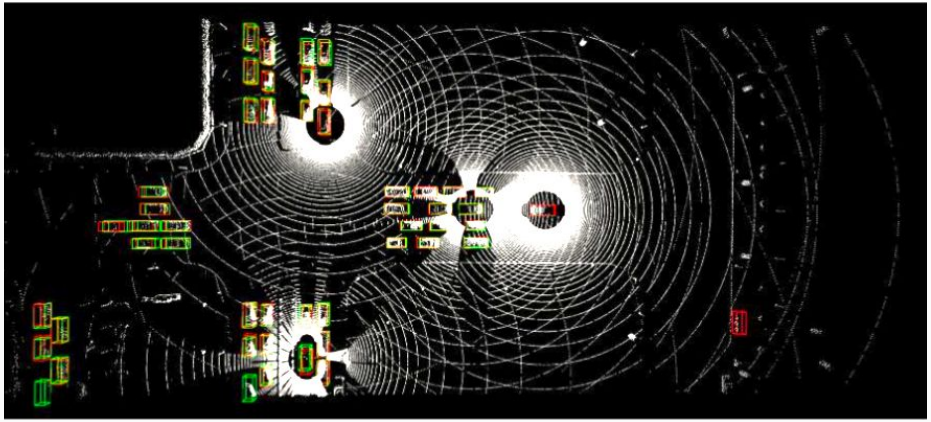


Table 4: Compression effect on OPV2V Camera.

CPR-rate	Size (KB)	IoU
0x	524	60.4
8x	66	60.1
16x	33	58.9
32x	16	56.2
64x	8	54.8

Evaluations

Experiment Setup

Datasets:

- OPV2V
- nuScenes

Evaluation Metrics:

- IoU
- AP

Table 1: **Map-view segmentation on OPV2V camera-track.** We report IoU for all classes. All fusion methods employs CVT [3] backbone, except for CoBEVT which uses SinBEVT backbone.

Method	Veh.	Dr.Area	Lane
No Fusion	37.7	57.8	43.7
Map Fusion	45.1	60.0	44.1
F-Cooper [21]	52.5	60.4	46.5
AttFuse [12]	51.9	60.5	46.2
V2VNet [20]	53.5	60.2	47.5
DiscoNet [14]	52.9	60.7	45.8
FuseBEVT	**59.0**	**62.1**	**49.2**
CoBEVT	**60.4**	**63.0**	**53.0**

Table 2: **3D detection results on the OPV2V LiDAR-track.** All methods employ PointPillar [61] backbone. (C) denotes using 64× feature compression.

Method	AP0.7	AP0.7(C)
No Fusion	60.2	60.2
Late Fusion	78.1	78.1
Early Fusion	80.0	-
F-Cooper	79.0	78.8
AttFuse	81.5	81.0
V2VNet	82.2	81.4
DiscoNet	83.6	83.1
FuseBEVT	**85.2**	**84.9**

Table 3: **Vehicle map-view segmentation on nuScenes.** All models use only a single time-stamp. $*$ denotes our reproduced result with EfficientNet-b4 backbone.

Method	Veh.	Par(M)	FPS
VPN* [27]	29.3	4.	31
OFT [59]	30.1	-	-
Lift-Splat	32.1	14	25
FIERY [4]	35.8	7	8
CVT [3]	36.0	1.2	35
SinBEVT	**37.1**	1.6	35

Evaluations

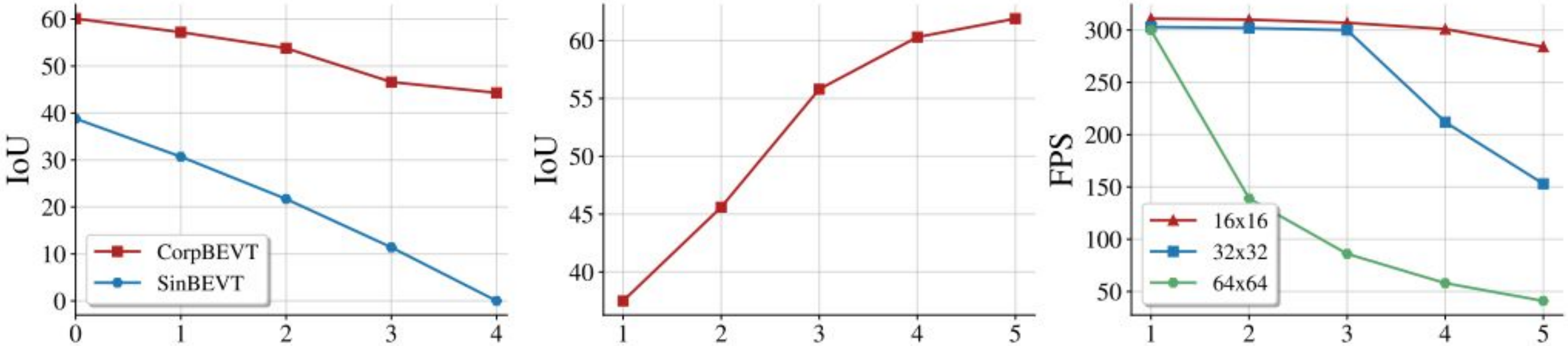


Figure 5: **Ablation studies.** (a) IoU *vs.* number of dropped cameras (b) IoU *vs.* number of agents. (c) FPS *vs.* number of agents. The channel dimension of BEV feature map is fixed as 128 for (c).

IoU

FPS @ higher feature map resolutions

Comparisons

Comparisons

The V2X Communication approach essentially utilizes multi-agent, multi-sensors data to enhance perception. However the following methods differ in the following ways:

AVR: Augmented Vehicular Reality

- Focuses on extending visual range by sharing 3D sensor data between agents

V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction

- Uses Graph Neural Network (GNN) to improve detection and motion forecasting

COOPERNAUT: End-to-End Driving with Cooperative Perception for Networked Vehicles

- Uses V2V to share encoded LiDAR data using Point Transformers for better decision making

BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation

- Focuses on fast and efficient processing of converting camera data to a BEV representation

Conclusion

- CoBEVT introduces a **multi-agent, multi-camera** perception framework designed for BEV semantic labeling using sparse transformers and data fusion

- Future work aims to address common V2V challenges like asynchronization, positional errors, and noise

Questions?