ICCV OCT 19-23, 2025 HONOLULU HAWAII

EVT: Efficient View Transformation for Multi-Modal 3D Object Detection

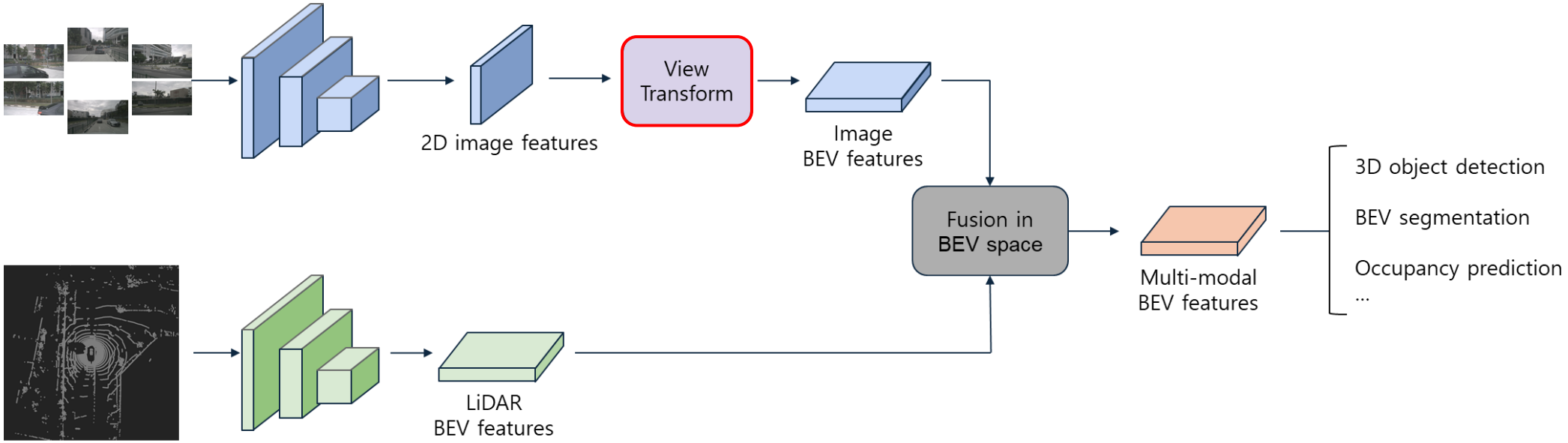
Yongjin Lee[1], Hyeon-Mun Jeong[1], Yurim Jeon[2], Sanghyun Kim[1], [2]

[1]ThorDrive Co., Ltd, South Korea

[2]Seoul National University, South Korea

Introduction

- View Transformation for BEV (bird's-eye-view) fusion



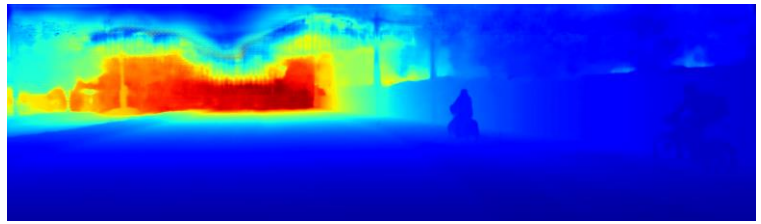
- Provides a unified bird's-eye-view representation of the scene
- Combines complementary strengths of cameras (rich semantics) and LiDAR (accurate geometry)
- Enables robust perception for downstream tasks (3D detection, tracking, planning)

Previous methods for view transformation

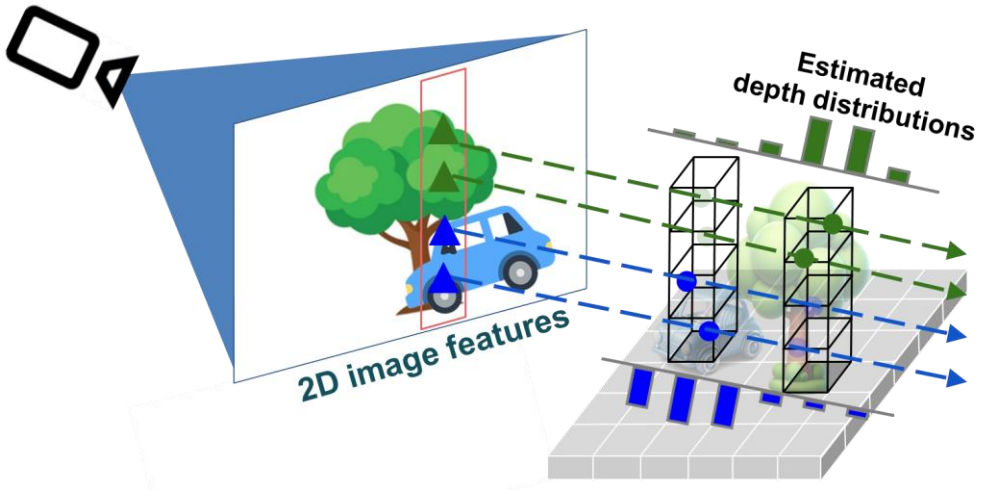
- Depth-based method

RGB images

Depth estimation networks



Depth maps



Lift 2D image features into 3D space

Rasterize into 3D grids & pool along z-axis

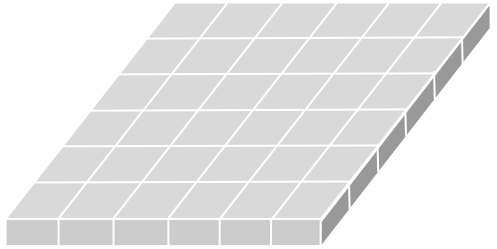
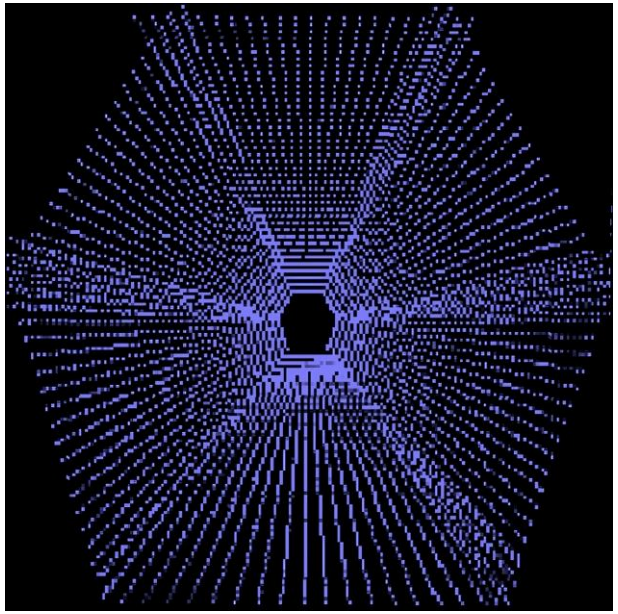


Image BEV features

Previous methods for view transformation

- Depth-based method

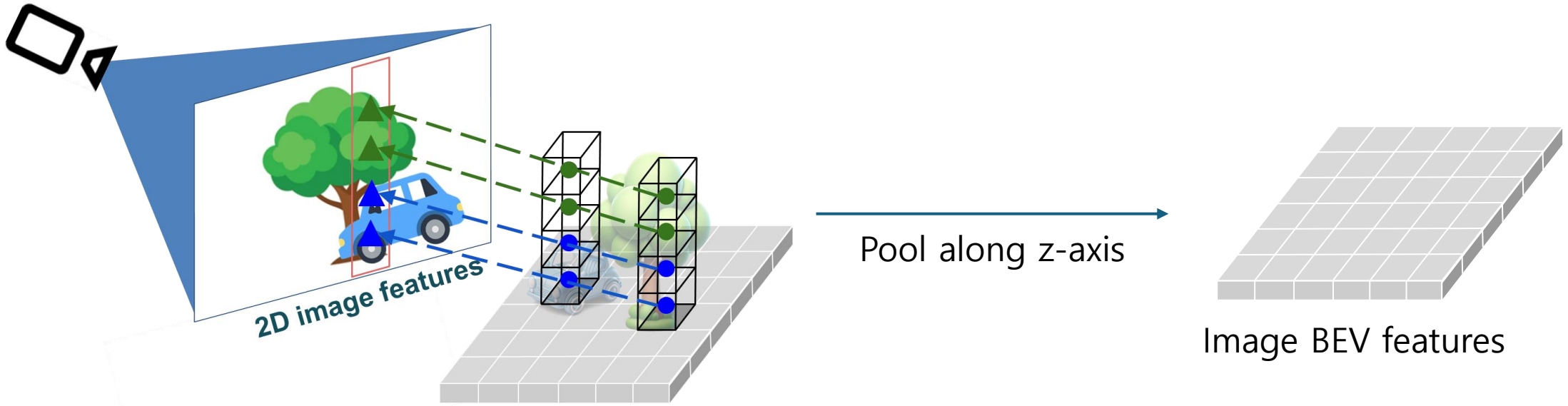


Result BEV feature map

1. Heavily dependent on depth accuracy
2. Sparse BEV representation
3. Quantization error due to 3D grid rasterization

Previous methods for view transformation

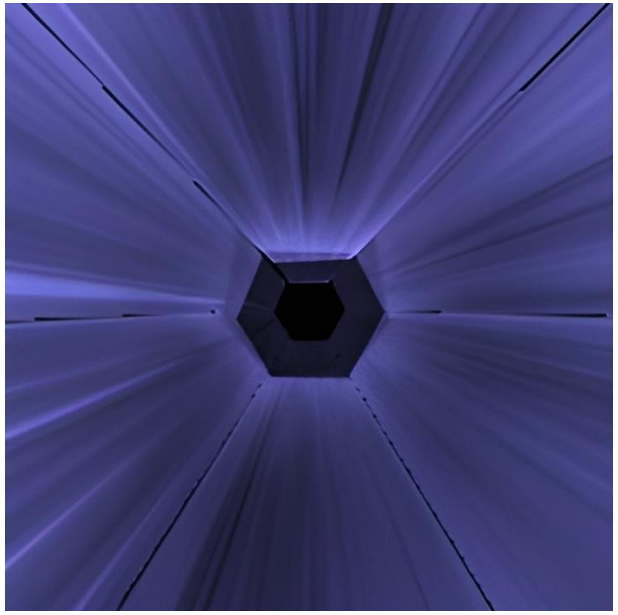
- Query-based method



Project predefined 3D grid points into 2D image plane
& sample the image features at those projected positions

Previous methods for view transformation

- Query-based method

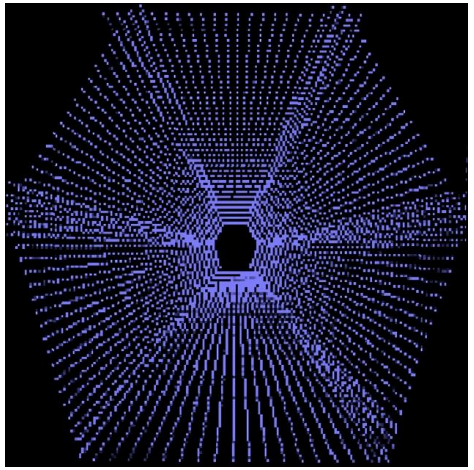


Result BEV feature map

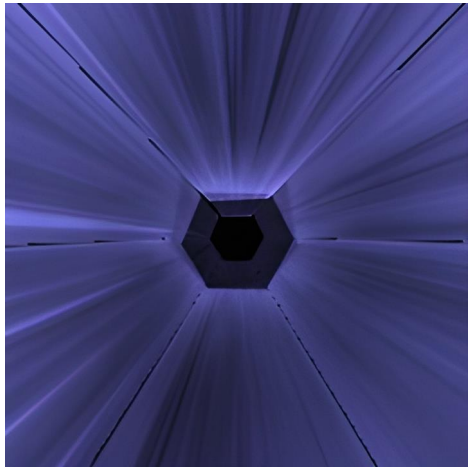
- Visually dense but with redundant features due to the lack of spatial cues
- Copying irrelevant features introduces noise and confuses the network.
- Attention-based methods to mitigate this issue introduce a computational bottleneck.

Previous methods for view transformation

- Problems



Result of
depth-based method



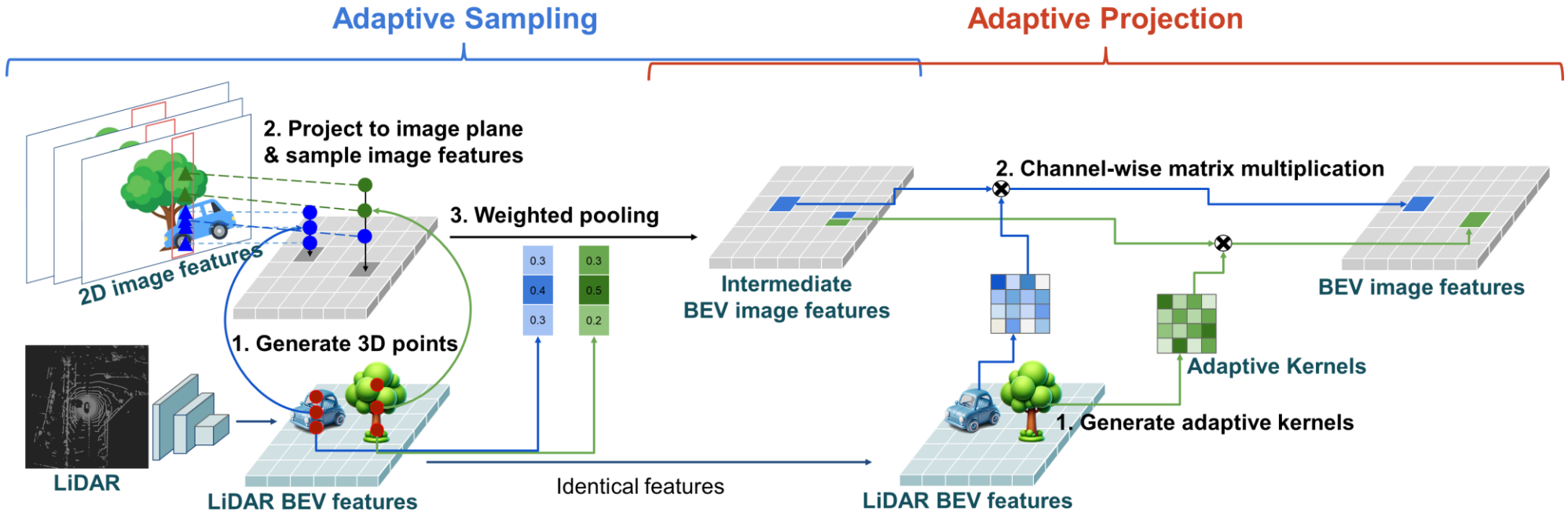
Result of
query-based method

- Both approaches have their own limitations.
- It is also difficult to interpret what scene the transformed BEV features are actually capturing.

➢ We propose a LiDAR-guided view transformation that produces a noise-free, dense BEV representation with accurate spatial placement, without computational bottlenecks!!

The proposed view transformation

- Overview

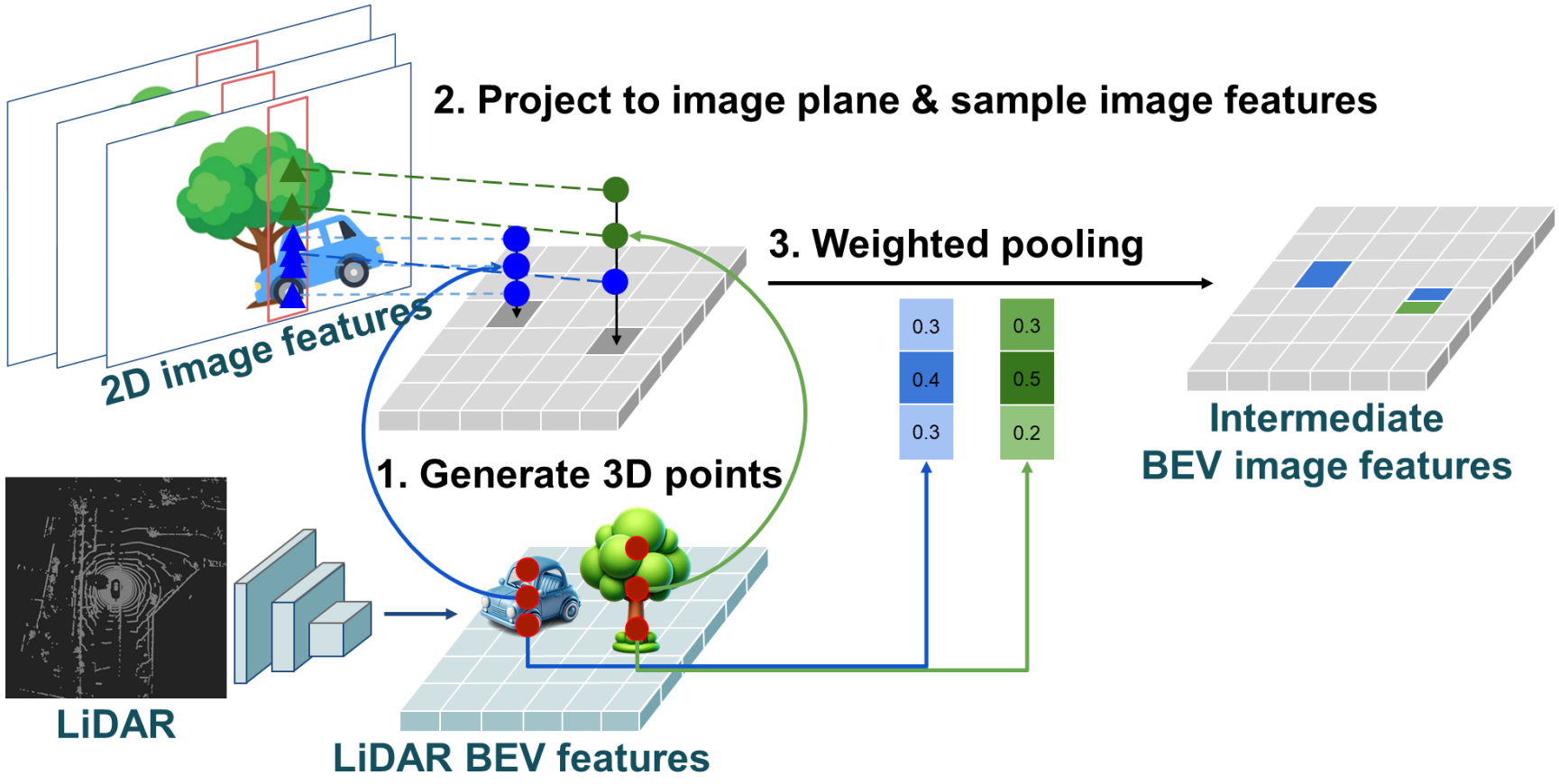


ASAP: two-stage view-transformation method using LiDAR guidance

- Adaptive Sampling (AS): **generates** intermediate image BEV features
- Adaptive Projection (AP): **refines** the intermediate image BEV features

Adaptive Sampling (AS) – generate stage

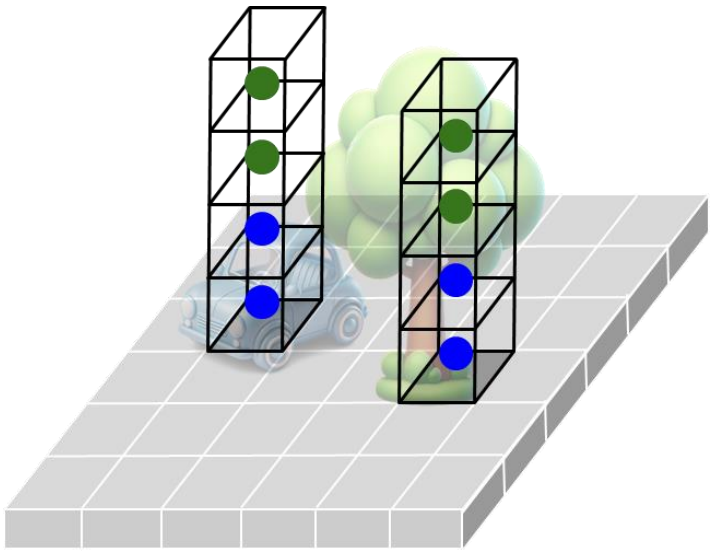
- Overview of Adaptive Sampling (AS)



Adaptive Sampling (AS) – generate stage

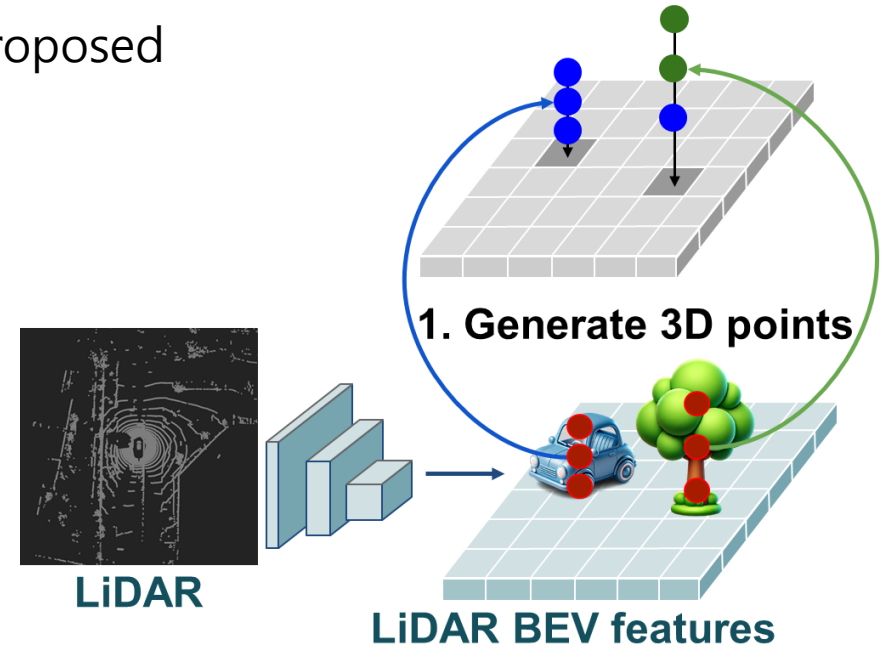
- Defining 3D sampling points

Previous (query-based method)



Input-invariantly predefined 3D grid points

Proposed

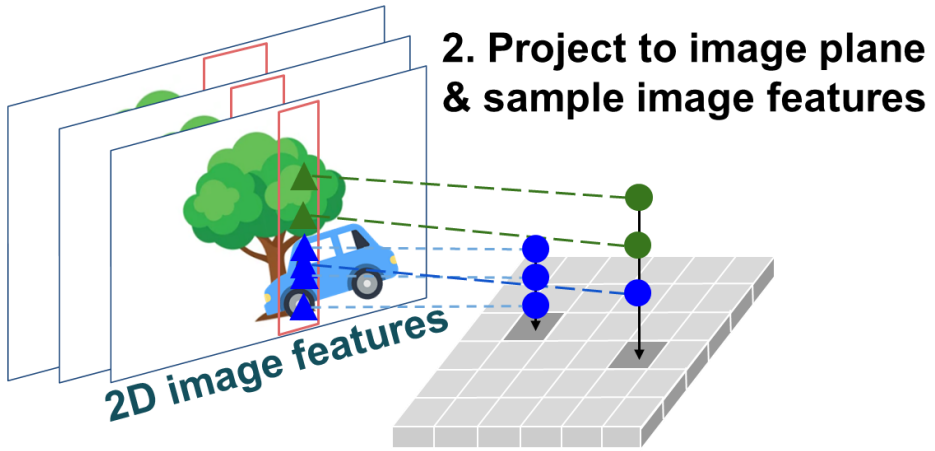


Adaptive 3D points in continuous space

➢ Previous methods sample uniformly even in empty space, whereas our approach uses LiDAR guidance to adaptively generate sampling points within object regions.

Adaptive Sampling (AS) – generate stage

- Point projection & feature sampling



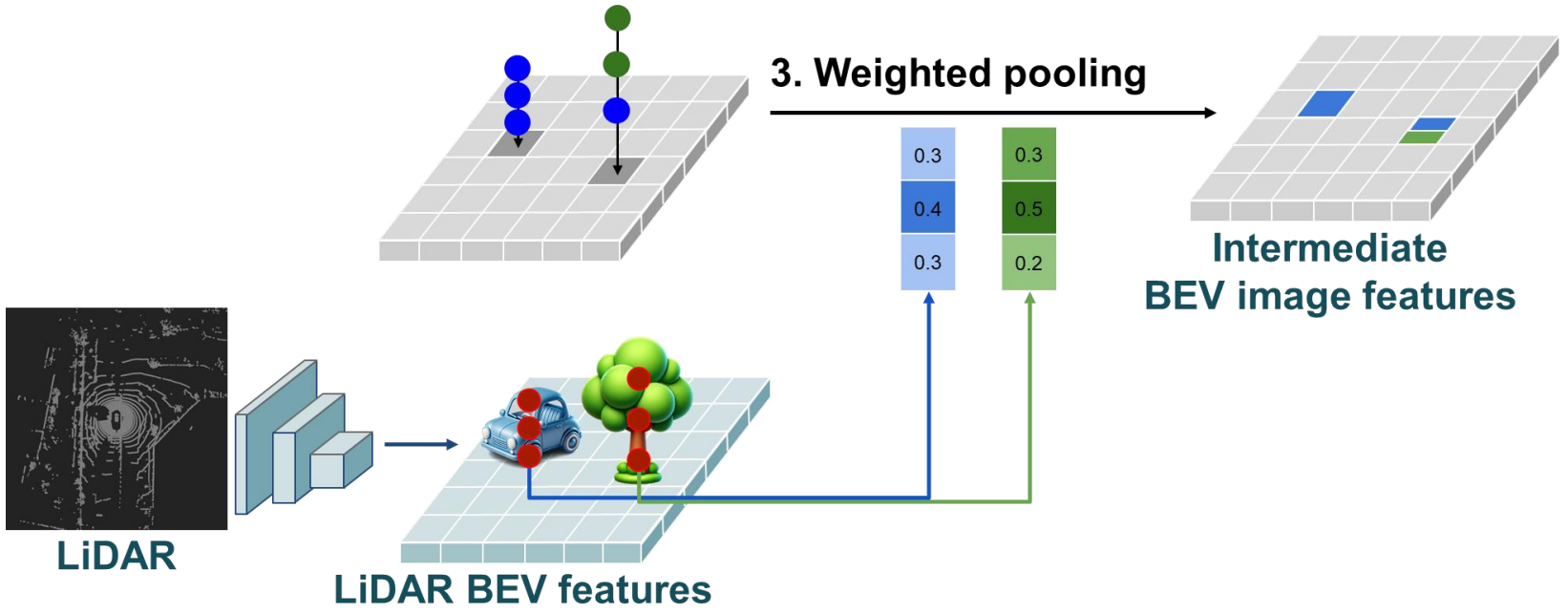
- Project 3D points onto the image plane using known camera parameters
- Sample image features at projected locations (bilinear interpolation)

➢ Several methods apply deformable attention at this stage (adding multiple offsets around each sampling point)

➢ Ours only requires simple bilinear interpolation — sampling points are already well aligned

Adaptive Sampling (AS) – generate stage

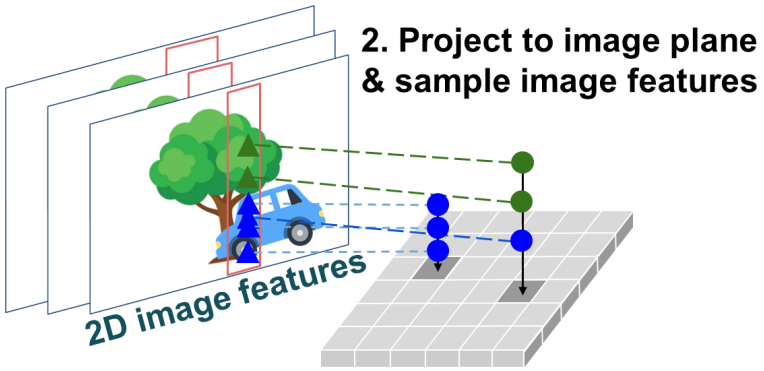
- Pooling features along z-axis



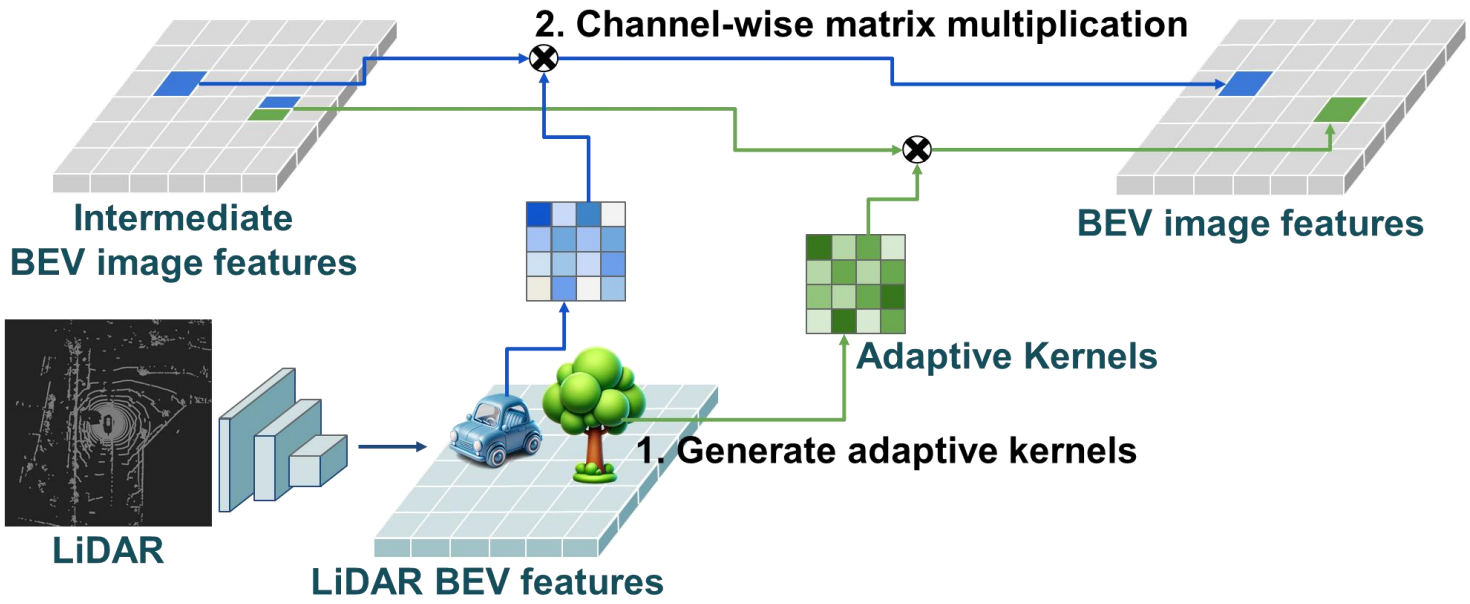
➢ LiDAR guidance assigns weights to the points along the z-axis, enabling weighted pooling.

Adaptive Projection (AP) – refine stage

- Overview



Noisy features from **AS**

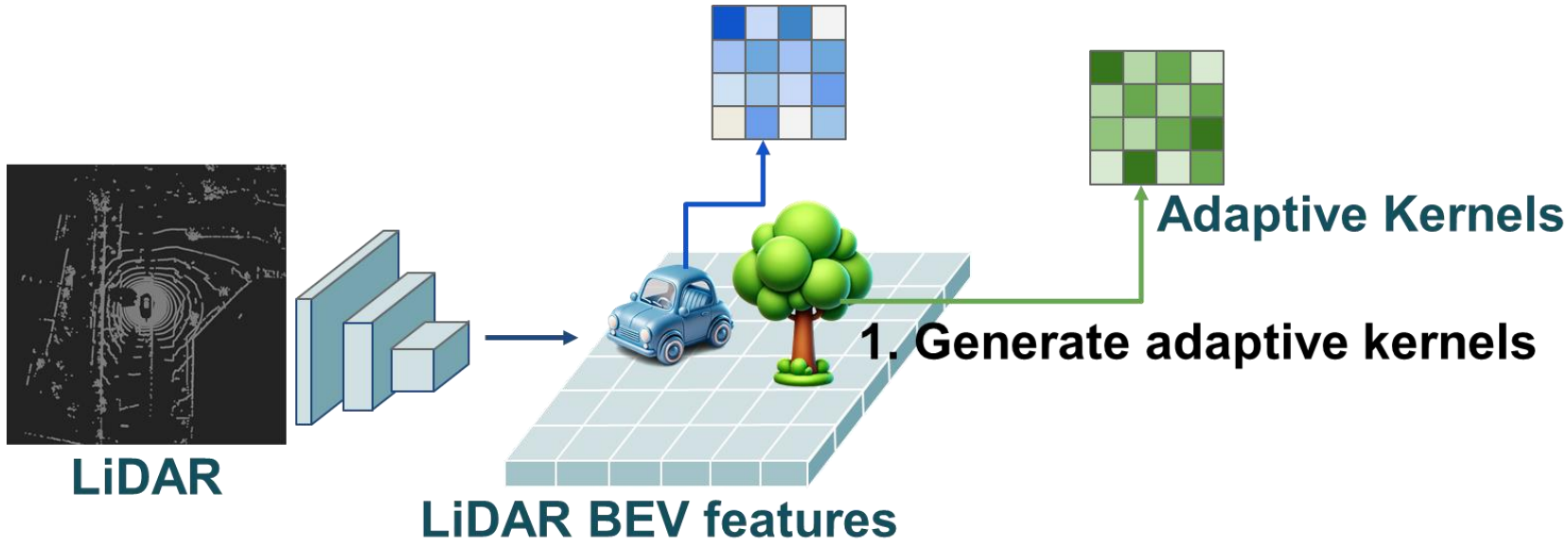


AP refines the intermediate BEV features

- Feature sampling stage inevitably produce misaligned features
- The proposed method refine directly in 3D space with LiDAR guidance.

Adaptive Projection (AP) – refine stage

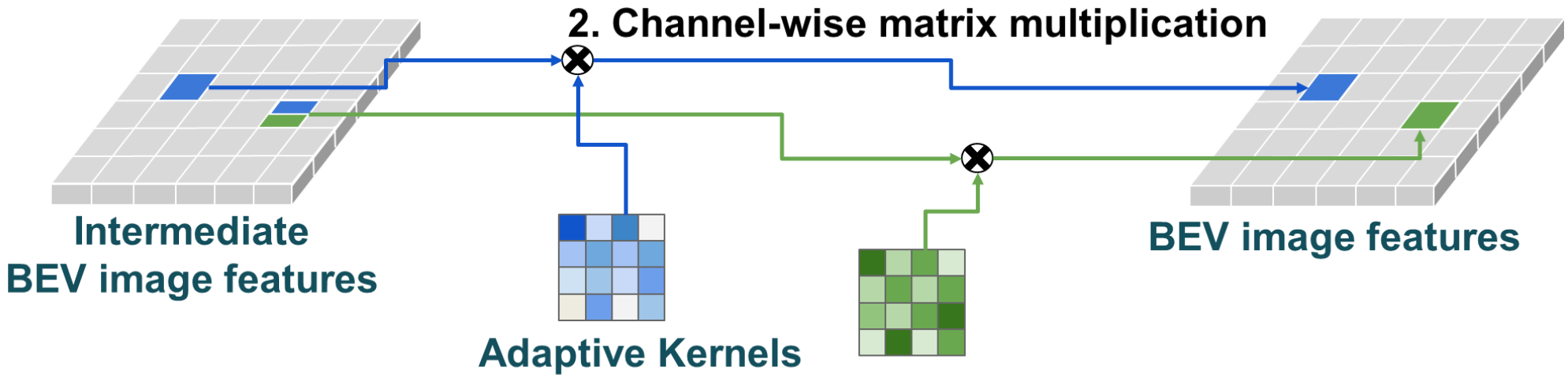
- Generate adaptive kernels for each grid cell



➢ LiDAR guided input-dependent refinement kernels

Adaptive Projection (AP) – refine stage

- Feature refinement using adaptive kernels



➢ Suppresses ambiguous or redundant assignments

➢ Enhances features more consistent with the actual 3D geometry.

Results

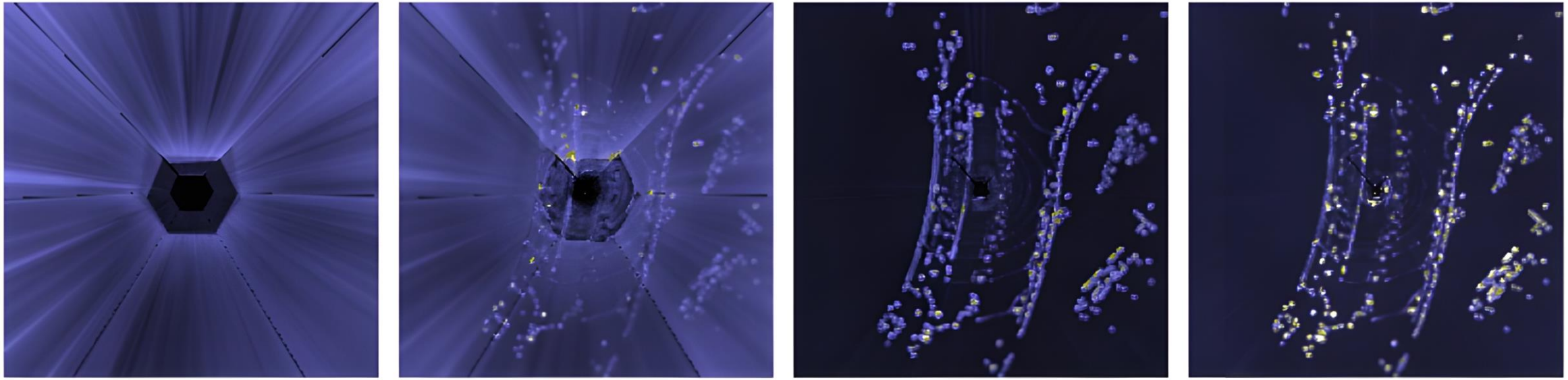
- Comparison of sampling points

Existing methods : input-invariantly predefined 3D grid points

Adaptive Sampling: adaptive 3D points in continuous space

Results

- Visualization of image BEV feature maps



(a) vanilla (b) only AS (c) only AP (d) ASAP

Results

- Performance comparison on nuScenes validation and test sets

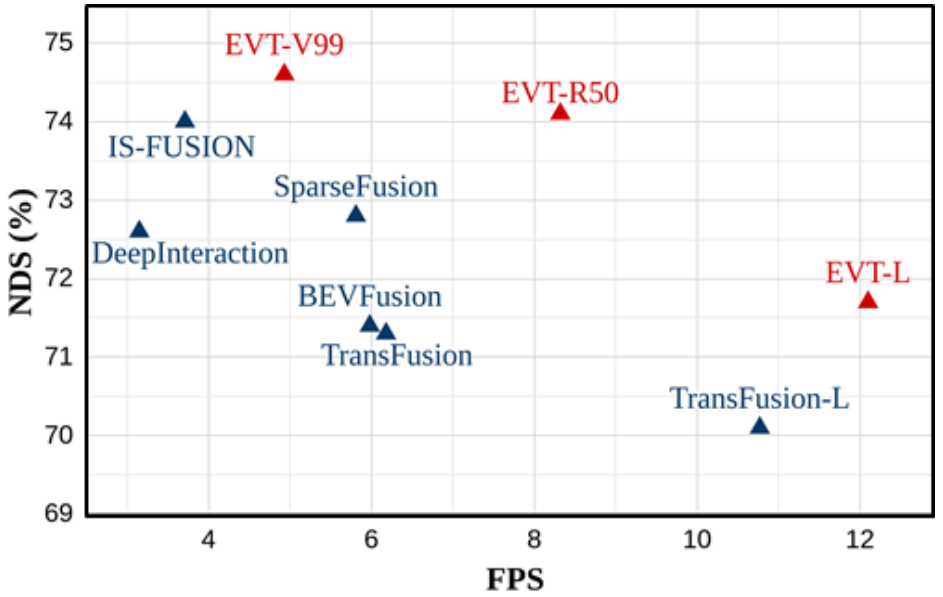
Method	Modality	NDS (*val*)	mAP (*val*)	NDS (*test*)	mAP (*test*)
TransFusion [1]	LC	71.3	67.5	71.7	68.9
DeepInteraction [48]	LC	72.6	69.9	73.4	70.8
BEVFusion [30]	LC	71.4	68.5	72.9	70.2
FocalFormer3D [5]	LC	71.1	66.5	73.9	71.6
CMT [45]	LC	72.9	70.3	74.1	72.0
BEVFusion4D-S [3]	LC	72.9	70.9	73.7	71.9
SparseFusion [44]	LC	72.8	70.4	73.8	72.0
UniTR [39]	LC	73.3	70.5	74.5	70.9
FusionFormer [11]	LCT	74.1	71.4	75.1	72.6
EVT (Ours)	LC	**74.6**	**72.1**	**75.3**	**72.6**

➢ Our method ranks first on the official nuScenes leaderboard under the same conditions (no ensembling, no TTA).

Results

- Efficiency of the proposed view transformation method

	LiDAR	Camera	AS	AP	NDS	mAP	FPS
(a)	✓				71.7	66.4	12.1
(b)	✓	✓			72.7	69.1	8.5
(c)	✓	✓	✓		73.5	70.6	8.5
(d)	✓	✓	✓	✓	74.1	71.1	8.3



- ASAP takes only 3 ms (≈2% overhead)
- EVT is both faster and more accurate than other methods.

Thank You