UniFusion: Unified Multi-view Fusion Transformer for Spatial-Temporal Representation in Bird's-Eye-View

Zequn Qin[1], Jingyu Chen[2], Chao Chen[2], Xiaozhi Chen[2]*, Xi Li[1,3,4]*
[1]College of Computer Science & Technology, Zhejiang University; [2]DJI
[3]Shanghai Institute for Advanced Study of Zhejiang University
[4]Zhejiang-Singapore Innovation and AI Joint Research Lab, Hangzhou
zequnqin@gmail.com, jeffery.chen@dji.com, huaijin.chen@dji.com
cxz.thu@gmail.com, xilizju@zju.edu.cn

Abstract

Bird's eye view (BEV) representation is a new perception formulation for autonomous driving, which is based on spatial fusion. Further, temporal fusion is also introduced in BEV representation and gains great success. In this work, we propose a new method that unifies both spatial and temporal fusion and merges them into a unified mathematical formulation. The unified fusion could not only provide a new perspective on BEV fusion but also brings new capabilities. With the proposed unified spatial-temporal fusion, our method could support long-range fusion, which is hard to achieve in conventional BEV methods. Moreover, the BEV fusion in our work is temporal-adaptive and the weights of temporal fusion are learnable. In contrast, conventional methods mainly use fixed and equal weights for temporal fusion. Besides, the proposed unified fusion could avoid information lost in conventional BEV fusion methods and make full use of features. Extensive experiments and ablation studies on the NuScenes dataset show the effectiveness of the proposed method and our method gains the state-of-the-art performance in the map and vehicle segmentation task.

1. Introducion

Recently, bird's-eye-view (BEV) representation [17, 20, 11] becomes an emerging perception formulation in the autonomous driving field. The main idea of BEV representation is to map the multi-camera features into the ego BEV space, *i.e.*, spatial fusion, as shown in Fig. 1. This kind of spatial fusion composes an integrated BEV space, and duplicate results from different cameras are uniquely represented in the BEV space, which greatly reduces the difficulty in fusing multi-camera features. Moreover, the BEV spatial fusion naturally shares the same 3D space as other modalities like LiDAR and radar, making multi-modality fusion simple.

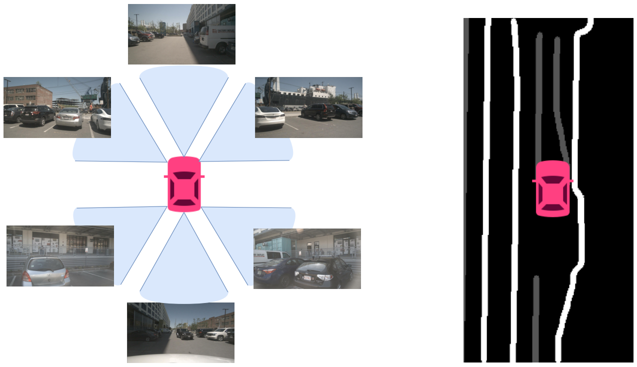

(a) Inputs with surrounding images.
(b) Map.

Figure 1: Illustration of the map segmentation task in BEV.

The integrated BEV representation based on spatial fusion provides the basis of temporal fusion. Temporal fusion is a cornerstone in BEV representation, which can be used in many aspects like 1) representing temporarily occluded objects; 2) accumulating observation in a long-range, which can be used for generating map; 3) stabilizing the perception results for standstill vehicles. There have been many methods [11, 9, 12] showing the importance and effectiveness of temporal fusion.

Despite the success of current progress, present methods usually use warp-based temporal fusion, *i.e.*, warping past BEV features to the current time according to the positions of BEV spaces at different time steps. Although this kind of design can well align temporal information, there are still some open problems. First, the warping is usually serial; that is to say, it is conducted only between ad-

*Corresponding authors: Xiaozhi Chen and Xi Li.

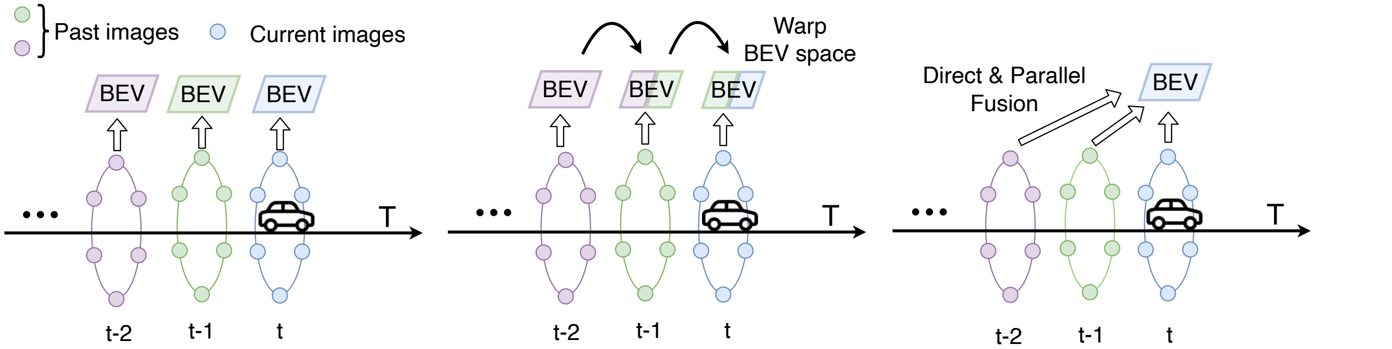


Figure 2: Different methods in BEV temporal fusion. From left to right, they are methods with no temporal fusion, warp-based temporal fusion, and our unified multi-view fusion. For the method with no temporal fusion, the BEV space is only predicted with surrounding images at the current time step. The warp-based temporal fusion would warp the previous BEV space to the current space. Each time only one previous step can be fused and the whole process is serial. In this work, we propose unified multi-view fusion, which directly fuses the surrounding images at all time steps parallelly. In this way, long-range and dynamic fusion is supported.

jacent time steps. In this way, it is hard to model long-range temporal fusion. Long-range history information can only implicitly make an impact and would be forgotten and dispelled rapidly. Besides, excessive long temporal fusion would even harm the performance in the warp-based temporal fusion. Second, warping would cause information loss during temporal fusion, as shown in Figs. 3b and 3c. Third, since the warping is serial, the weights for all time steps are equal, and it is hard to adaptively fuse temporal information.

To solve the above problems, we propose a new perspective that combines both spatial and temporal fusion into a unified multi-view fusion, termed UniFusion. Specifically, spatial fusion is regarded as a multi-view fusion from multi-camera features. For the temporal fusion, since the temporal features are from the past and absent in the current time, we create “virtual views” for the temporal features as if they are present in the current time. The idea of “virtual views” is to treat past camera views as the current views and assign them virtual locations relative to the current BEV space based on the camera motion. In this way, the whole spatial-temporal representation in BEV can be simply treated as a unified multi-view fusion, which contains both current (spatial fusion) and past (temporal fusion) virtual views, as shown in Fig. 2.

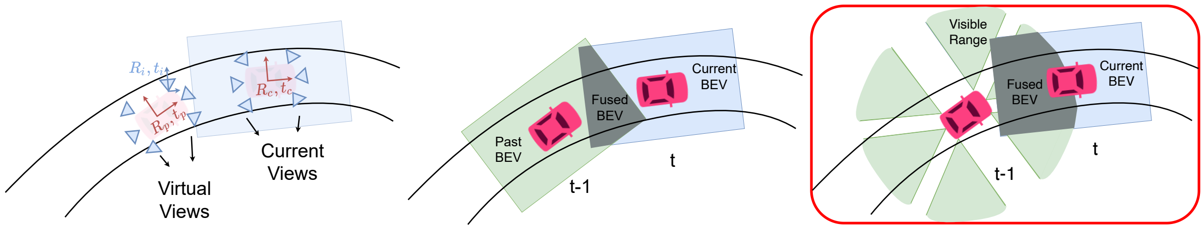
With the proposed unified fusion, both spatial and temporal fusions are conducted in parallel. We can directly access all useful features through space and time at once, which enables the long-range fusion. Another benefit is that we can realize adaptive temporal fusion since we can directly access all temporal features. Meanwhile, the parallel property guarantees that no information is lost during fusion. Furthermore, the multi-view unified fusion can even support different sensors, camera rigs, and camera types at different time steps. This will bridge higher-level and heterogeneous fusion like vehicle-side and road-side perceptions. For example, we can fuse information from a car’s camera and a surveillance camera on top of a traffic light, as long as they overlap in the BEV space.

The contributions of this work are as follows:

- We propose a new parallel multi-view perspective for BEV representation, which unifies the spatial and temporal fusion. The proposed unified parallel multi-view fusion can address the problem of long-range fusion and information loss. And we can realize adaptive temporal fusion based on the unified fusion. The proposed unified method can also support arbitrary camera rigs and bridge higher-level and heterogeneous fusion.
- We analyze the widely used evaluation settings in the map segmentation task on NuScenes [4] and propose a new setting for a more comprehensive comparison in Sec. 4.1.
- The proposed method achieves the state-of-the-art BEV map segmentation performance on the challenging benchmark NuScenes in all settings.

2. Related Work

Spatial fusion in BEV Spatial fusion is the basis of BEV representation, *i.e.*, how to transform and fuse information and features from surrounding multi-camera inputs into an ego BEV space to represent the surrounding 3D world. The earliest and most straightforward method is the inverse perspective mapping (IPM) [16, 2, 1, 7], which assumes the ground surface is flat and at a fixed height. In this way, the spatial fusion in BEV can be conducted with a homography transformation. Note that IPM is usually utilized in the image space. However, IPM is hard to cope with the



(a) Illustration of virtual views. (b) Warp-based BEV fusion. (c) Actual BEV space that can be fused.

Figure 3: Comparison between warp-based and actual fusion range. The fused area is marked in gray. With the warp-based fusion, the fused area is limited in the intersection between BEV rectangles and many already seen parts are wasted.

non-flat and unknown-height ground surface. Later, View Parsing Network (VPN) [17] uses a fully connected layer to transform the image features into the BEV features and directly supervise the features in the BEV space in an end-to-end manner. Similarly, BEVSegFormer [19] uses the deformable attention [27] mechanism to achieve end-to-end mapping. These methods avoid the explicit mapping between image and BEV spaces, but this property also makes them hard to adopt the geometry prior. Based on VPN, HDMapNet [11] proposes to only map the image space to camera-ego BEV space in an end-to-end manner, while the multi-camera BEV spaces are fused with the camera poses. In this way, part of the geometry prior, *i.e.*, the camera extrinsic information is utilized. To make full use of geometry prior in the spatial fusion of BEV space, Lift-splat-shoot [20] proposes a latent estimation network to predict depth for each pixel in the image space. Then all the pixels with depth can be directly mapped into the BEV space. Another kind of method OFT [21] does not make predictions of depth. OFT directly copy-and-paste the features in the image space to all locations that trace along the ray from the camera in the BEV space. Different from the spatial fusion perspective of geometric mapping, X-Align[3] aligns the semantics of camera and BEV spaces.

Temporal fusion in BEV With the basis of spatial fusion, temporal fusion could further boost the representation in BEV space. The mainstream methods of temporal fusion are the warp-based method [26, 9, 12]. The main idea of the warp-based method is to warp and align BEV spaces at different time steps based on the ego motions of vehicles. The major differences reflect in the way of using wrapped BEV spaces. BEVFormer [12] uses deformable self-attention to fuse wrapped BEV spaces while BEVDet4D directly concatenates the wrapped BEV spaces. BEVFusion proposes [14] a unified multi-task and multi-sensor fusion method that can fuse camera and LIDAR.

3. Method

In this section, we elaborate on the design of our method from two aspects. First, we show the derivation of the unified multi-view fusion. Then we demonstrate the network architecture with unified multi-view fusion.

3.1. Unified Fusion with Virtual Views

As discussed in the introduction, spatial fusion is the foundation of BEV representation, while temporal fusion reveals a new direction for better BEV representation.

Conventional BEV temporal fusion is warp-based fusion, as shown in Fig. 3b. The warp-based fusion warps past BEV features and information based on the ego-motion of different time steps. Since all features are already organized in a pre-defined ego BEV space at a certain time step before warping, this process would lose information.

The actual visible range of a camera is much bigger than the one of ego BEV space. For example, 100m is a very humble visible range for typical cameras, while most BEV ranges are defined as no more than 52m [12, 20]. In this way, it is possible to obtain better BEV temporal fusion than simply warping BEV spaces, as shown in Fig. 3c.

To achieve better temporal fusion, we propose a new concept, *i.e.*, virtual view, as shown in Fig. 3a. Virtual views are defined as the views of sensors that do not present in the current time step, and these past views are rotated and translated according to the ego BEV space as if they are present in the current time step. Denote $R_c \in \mathbb{R}^{3\times 3}, t_c \in \mathbb{R}^{3\times 1}$ and $R_p \in \mathbb{R}^{3\times 3}, t_p \in \mathbb{R}^{3\times 1}$ as the rotations and translations matrices of current and past ego BEV spaces, respectively. Suppose $R_i \in \mathbb{R}^{3\times 3}$, $t_i \in \mathbb{R}^{3\times 1}$, and $K_i \in \mathbb{R}^{3\times 3}$ are the rotation, translation and intrinsic matrices of a certain view V_i. The rotation and translation matrices of virtual views can be written as:

$$\begin{aligned} R_i^v &= R_i^{-1} R_p^{-1} R_c \\ t_i^v &= R_i^{-1} R_p^{-1} t_c - R_i^{-1} R_p^{-1} t_p - R_i^{-1} t_i, \end{aligned} \tag{1}$$

in which $R_i^v \in \mathbb{R}^{3\times 3}$ and $t_i^v \in \mathbb{R}^{3\times 1}$ are the unified virtual rotation and translation matrices for any view V_i. It can be examined that Eq. (1) also holds for the current views. In this way, all views can be mapped and utilized in the same way, no matter they are past or current views. Suppose $P_{bev} \in \mathbb{R}^{N\times 3}$ represents the coordinates in the BEV space,

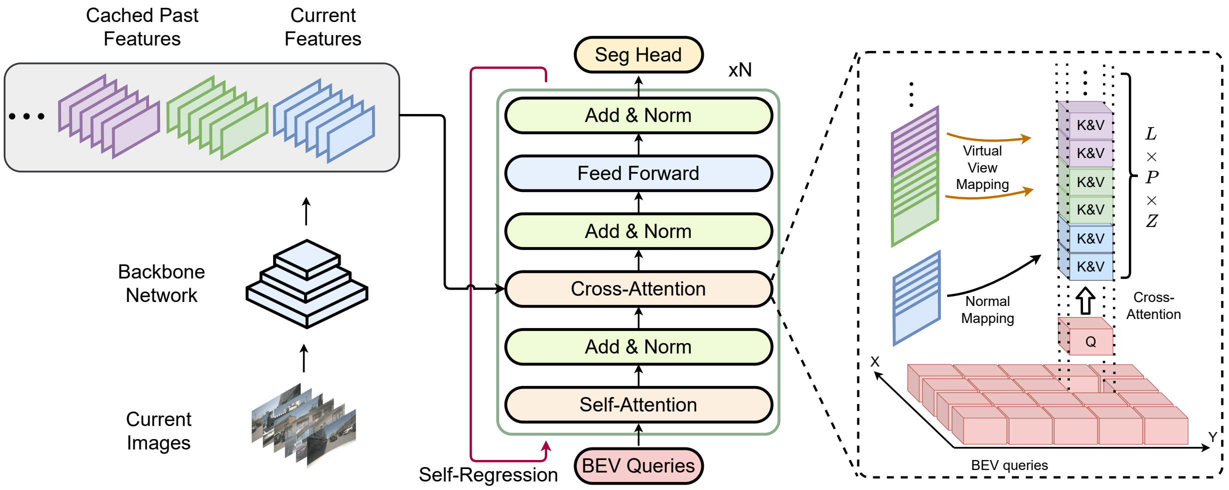


Figure 4: Network architecture.

$P_{img} \in \mathbb{R}^{N\times 3}$ is the homogeneous coordinates in the image space, and N is the number of coordinates. The mapping between BEV space and all views can be written as:

$$P_{img} = K_i(R_i^v P_{bev} + t_i^v). \tag{2}$$

Then we can map the image features to the BEV features F.

3.2. Network Design with Unified Fusion

With the help of the unified multi-view fusion, we show the network architecture in this part. The network is composed of three parts, which are the backbone network, unified multi-view fusion Transformer, and segmentation head, as shown in Fig. 4.

Backbone We use three kinds of widely used backbones ResNet50 [8], Swin-Tiny [13] and VoVNet [10] to extract L multi-scale features ($L = 4$) from multi-camera images. For the ResNet50 and VoVNet models, only features from stages 2, 3, and 4 are used. Following Deformable-DETR [27], an extra 3x3 convolution with a stride of 2 is used to generate the last feature. The backbone is shared between all views' images. It is worth mentioning that the features of past images can be maintained and reused in a feature queue without extra computational cost.

Fusion Transformer We use a Transformer [23] encoder to fusion features from all views. There are four major parts in the Transformer encoder, which are the BEV queries, the self-attention module, the cross-attention model, and the self-regression mechanism.

In order to represent the BEV space, we use $X \times Y$ queries $\{Q_{x,y} \in \mathbb{R}^C | x \in \{1, \cdots, X\}, y \in \{1, \cdots, Y\}\}$ in a 2D grid to represent the whole BEV space, where X and Y are the spatial sizes of the BEV grid.

The second major part is the self-attention module. It is used to interact with all BEV queries and exchange information in the BEV space. Since the time complexity of the vanilla self-attention interaction is $O(X^2Y^2)$, we use deformable self-attention [27] to reduce the computational cost.

The most important module of this work is the cross-attention used for unified multi-view spatial-temporal fusion. With the help of the unified multi-view fusion, all spatial-temporal features can be mapped to the same ego BEV space. The goal of the cross-attention module is to fuse and integrate the mapped spatial-temporal BEV space features F.

Denote $(\hat{x}, \hat{y}, \hat{z})$ are the real-world coordinates in the 2D BEV grid (x, y), and $\hat{z}$ is the real-world height for sampling. Suppose the number of sampling in height in each BEV grid is Z, then each BEV query $Q_{x,y}$ corresponds to Z points, and the total coordinates in the BEV space is $P_{bev} \in \mathbb{R}^{XYZ\times 3}$. Then we can obtain the mapped BEV features F according to Eq. (2) with P_{bev}. Suppose the number of time steps in temporal fusion is P, then the cross-attention (CA) module can be written as:

$$\text{CA}(Q_{x,y}, F) = \sum_{p,l,z} \frac{e^{att^{p,l,z}_{x,y}}}{\sum_{p,l,z} e^{att^{p,l,z}_{x,y}}} F^{p,l,z}_{x,y}, \tag{3}$$

where $F^{p,l,z}_{x,y}$ is the sampled value at the point of $(\hat{x}, \hat{y}, \hat{z})$ from the BEV features F of l-th multi-scale level and p-th time step. $\sum_{p,l,z}$ is the summation over P time steps, L scales, and Z heights. The attention value of $att^{p,l,z}_{x,y}$ is:

$$att^{p,l,z}_{x,y} = \frac{Q_{x,y}K^{p,l,z}_{x,y}}{\sqrt{C}}, \tag{4}$$

in which C is the dimension of each BEV query, and $K^{p,l,z}_{x,y}$

Table 1: Comparison of different map segmentation settings on NuScenes.

Setting	Front/rear range	Left/right range	BEV grid size	Map element type	Line width	Split
100m × 100m	50m	50m	0.5m × 0.5m	Line, polygon	1-pixel	Vanilla
60m × 30m	30m	15m	0.15m × 0.15m	Line	5-pixel	Vanilla
160m × 100m	100m/60m	50m	0.25m × 0.25m	Line	3-pixel	City-based

is the attention key composed of input $F_{x,y}^{p,l,z}$ and positional embedding.

In this way, we can use BEV queries Q to iterate over features from different places in the BEV space, time steps, multi-scale levels, and sampling heights. **The information from all over the places and all over the time can be directly retrieved without any loss in a unified manner**. This kind of design also makes long-range fusion possible since all features are directly accessed no matter how long before, which also enables adaptive temporal fusion.

The last major part of our method is the self-regression mechanism. Inspired by BEVFormer [12], which concatenates the warped previous BEV features with the BEV queries before the self-attention module to realize the temporal fusion, we use a self-regression mechanism that concatenates the output of Transformer with the BEV queries as the new inputs and rerun the Transformer to get the final features. For the first running of the Transformer, we simply double and concatenate the BEV queries as the inputs.

In BEVFormer, it is believed that the concatenation of warped BEV features and BEV queries brings temporal fusion, and it is the root cause of performance gain. In this work, we propose another explanation for this phenomenon, that is, the concatenation of BEV features and queries is to implicitly deepen and double the number of the Transformer's layers. Because the warped BEV features are already processed by the Transformer at previous time steps, the concatenation can be viewed as the grafting of two successive Transformers. In this way, a simple self-regression without warping can achieve a similar performance gain as BEVFormer. The detailed ablation study can be found in Sec. 4.3.

Segmentation head We use a lightweight, fully convolutional model ERFNet [22] as our segmentation head, which will upsample the output of the Transformer to the given BEV space resolution.

4. Experiments

4.1. Dataset and Evaluation Settings

Dataset In this work, we use NuScenes [4] as the evaluation dataset for the map and vehicle segmentation task, which contains 1,000 driving scenes collected in Boston and Singapore. There are 28,130 and 6,019 keyframes for the training and validation set. Each keyframe contains six surrounding images.

Evaluation settings There are two widely used settings for the map segmentation task on NuScenes. The first one is the 100m × 100m setting [20, 12, 25] with two classes `road` and `lane`. The other one is the 60m × 30m setting [11, 19, 26] with three classes `boundary`, `divider`, and `ped crossing`. In this work, we also propose a new 160m × 100m setting for a more comprehensive evaluation, as shown in Tab. 1. The key motivations of the new setting are: 1) the evaluation range should be as large as the visible limit. 2) the evaluation criterion should be discriminative for both bad and good predictions. 3) the evaluation should avoid overfitting and show the ability of generalization[1]. In the new setting, we also use two difficulty levels "easy" and "hard". For the "easy" level, the evaluation is conducted with the front, rear, left, and right ranges of 50m, 30m, 30m, and 30m, respectively. The "hard" level is onducted with the left areas in the 160m × 100m range. For all settings, mean intersection-over-union (mIoU) is used as the evaluation metric.

For the vehicle segmentation task, we follow the setting in [20, 12], which contains "car" and "vehicles" classes.

4.2. Implementation Details

To evaluate the results of our method, we use ResNet50 [8], Swin-Tiny [13], and VoVNet [10] as our backbones. The ResNet50 and Swin backbones are initialized from ImageNet [6] pretraining, and VoVNet backbone is initialized from DD3D checkpoint [18]. The default number of layers of the Transformer is set to 12. The input image resolutions are set to 1600 × 900 for ResNet50 and Swin. For VoVNet, we use 1408 × 512 image size. We use AdamW [15] optimizer with a learning rate of 2e-4 and a weight decay of 1e-4. The learning rate is decreased by a factor of 10 for the backbone. The batch size is set to 1 per GPU, and models are trained with eight GPUs for 24 epochs. At the 20th epoch, the learning rate is decreased by a factor of 10. The number of multi-scale features is set to $L = 4$, the default number of previous time steps is set to $P = 6$, and the number of sampling heights is set to $Z = 4$.

[1] The detailed information, motivation, and derivation of the new setting can be found in the supplementary materials.

The height range is $(-5\text{m}, 3\text{m}]$ with a stride of 2m.

For the 100m × 100m setting, we use 50 × 50 BEV queries to represent the whole BEV space, then the results are upsampled by a factor of 4 to match the BEV resolution. For the 60m × 30m setting, we use 100 × 50 BEV queries with a similar upsampling as the 100m × 100m setting. For the 160m × 100m setting, we use 80 × 50 BEV queries and then upsample 8x to match the ground truth resolution. We use cross entropy loss to train on both settings. The loss weight for the background class is set to 0.4 by default for the class imbalance problem. Since the `road` class in the 100m × 100m setting is polygon area without the class imbalance problem, the loss weight of the `road` background class is set to 1.0.

4.3. Ablation Study

Ability of long-range fusion As discussed in the Introduction, the proposed unified multi-view fusion has the ability of long-range fusion since it can directly access both spatial and temporal information. In this part, We show the results of different fusion time steps to examine the ability of long-range fusion.

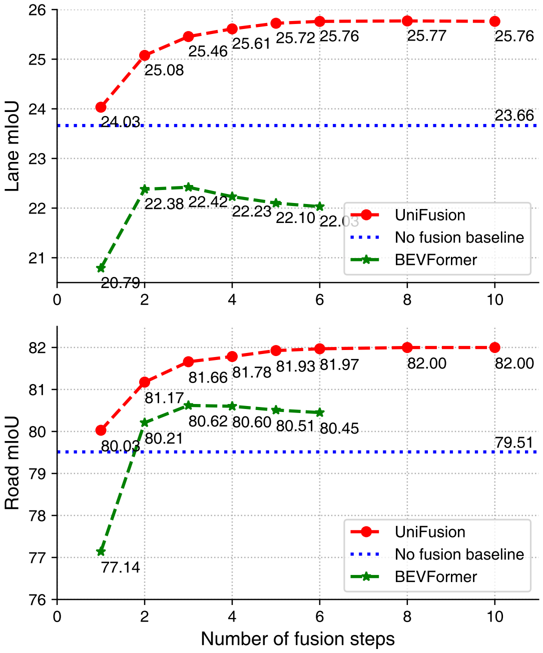


Figure 5: Ability of long-range temporal fusion.

From Fig. 5, we can see that our method could consistently benefit from the long temporal fusion even up to 10 steps. And the fusion duration for the 10 steps is 2 seconds. However, the warp-based BEVFormer's performance would drop after 3 fusion steps. This is also in accord with the results in BEVFormer [12] that the performance of warp-based temporal fusion would decrease with longer fusion than 4 contiguous steps. This shows the effectiveness of the proposed multi-view unified temporal fusion and the ability of long-range fusion.

Since the performance gradually converges after 6 fusion steps, we set the number of temporal fusion steps P to 6 in this work.

Disentangled training and inference fusion Although the proposed unified fusion has the ability of long-range fusion, this also brings another problem of computational complexity, especially during training. Longer fusion steps demand more memory and computational cost. We find a phenomenon that can alleviate this problem, *i.e.*, the number of temporal fusion steps during training does not need to be the same as the one during inference. And a model trained with a short-range fusion setting still has the ability of long-range fusion during inference. We call this phenomenon disentangled training and inference fusion. The results are shown in Tab. 2.

Table 2: Comparison of different numbers of temporal fusion steps. Note that the number of steps does not include current step.

#Fusion steps (training)	#Fusion steps (inference)	Road mIoU	Lane mIoU
0	0	79.04	22.64
1	1	79.48	23.03
1	6	81.12	24.24
2	6	80.91	24.99
3	6	81.02	24.48
4	6	81.25	24.75

From Tab. 2, we can see that no matter how many temporal fusion steps we use during training, the performance is very close when using 6 inference fusion steps. Moreover, even if we use only one previous step during training, the model still gains good performance with 6 temporal steps during inference. That is to say, the model still has the ability of long-range fusion when trained with a short-range fusion setting. By default, we use 2 temporal fusion steps during training.

Effectiveness of self-regression mechanism In Sec. 3.2, we propose a self-regression mechanism to further boost the performance. In this part, we examine the effectiveness of the self-regression mechanism. As shown in Tab. 5, we can see that the model with self-regression always gains better performance. Interestingly, the performance of the 12-layer non-regression model is close to the one of the 6-layer self-regression model. This verifies the analysis in Sec. 3.2.

Table 3: Experiments on NuScenes with the **100m** × **100m** setting. * means the results are reported from BEVFormer [12]. † indicates that M2BEV uses a different setting, in which the BEV resolution is 2x larger. So the "Lane mIoU" is high.

Method	Years	Backbone	Parameters	FPS	mIoU (Vanilla / City-based)		
					Road mIoU	Lane mIoU	All
LSS [20]	ECCV20	EffNetb0	-	-	72.9 / -	20.0 / -	46.5 / -
VPN* [17]	IROS20	Res101DCN	-	-	76.9 / -	19.4 / -	48.2 / -
LSS* [20]	ECCV20	Res101DCN	-	-	77.7 / -	20.0 / -	48.9 / -
M2BEV [25]	-	ResNeXt101	112.5	1.4	77.2 / -	40.5 / -†	58.9 / -†
BEVFormer [12]	ECCV22	Res101DCN	68.7	1.7	80.1 / -	25.7 / -	52.9 / -
UniFusion	-	ResNet50	42.4	2.6	**82.0 / 42.6**	**25.8 / 11.2**	**53.9 / 26.9**
UniFusion		VoVNet99	84.0	2.7	**85.4 / 47.9**	**31.0 / 11.6**	**58.2 / 29.8**

Table 4: Experiments on NuScenes with the **60m** × **30m** setting. * means the results are reported from HDMapNet [11]. ** means the BEVFormer is reimplemented in this work.

Method	Years	Backbone	mIoU (Vanilla / City-based)			
			Divider	Ped Crossing	Boundary	All
VPN* [17]	IROS20	EffNetb0	36.5 / -	15.8 / -	35.6 / -	29.3 / -
LSS* [20]	ECCV20	EffNetb0	38.3 / -	14.9 / -	39.3 / -	30.8 / -
HDMapNet [11]	ICRA22	EffNetb0	40.6 / -	18.7 / -	39.5 / -	32.9 / -
BEVSegFormer [19]	-	ResNet101	51.1 / -	32.6 / -	50.0 / -	44.6 / -
BEVerse [26]	-	Swin-tiny	56.1 / -	**44.9** / -	58.7 / -	53.2 / -
BEVFormer** [12]	ECCV22	ResNet50	53.0 / 20.4	36.6 / 8.9	54.1 / 24.3	47.9 / 17.9
UniFusion	-	Swin-tiny	**58.6 / 32.4**	43.3 / 17.2	**59.0 / 29.8**	**53.6 / 26.5**
UniFusion	-	VoVNet99	**60.6 / 32.5**	**49.0 / 11.5**	**62.5 / 32.9**	**57.4 / 25.6**

Moreover, we can see that the number of layers is also important for the final performance.

Table 5: Comparison with different number of Transformer layers and self-regression.

#Layers	Self-Reg	Road mIoU	Lane mIoU
6		80.42	24.26
6	✓	80.91	24.99
12		81.13	25.29
12	✓	81.97	25.76

Unified cross attention brings adaptive temporal fusion In Eq. (3), we show the core design of the unified multi-view spatial-temporal fusion is the unified cross attention module based on virtual views. The cross attention module can iterate over features from different time steps, which brings another important property, *i.e.*, adaptive temporal fusion. To verify this, we directly average the P temporal features before feeding them into the Transformer as the counterpart for comparison, which can be viewed as a fixed equal-weighted fusion. The results are shown in Tab. 6.

We can see that our method outperforms the equal-weighted temporal fusion counterpart in all settings. This shows that our method could adaptively fuse information from different time steps.

Table 6: Effectiveness of adaptive temporal fusion with different fusion steps. "Avg." is the equal-weighted fusion.

Fusion Steps	1	2	3	4	5	6
UniFusion	24.03	25.08	25.46	25.61	25.72	25.76
Avg.	23.26	24.47	24.82	24.95	25.03	25.08

4.4. Results

To validate the performance of our method, we use VPN [17], Lift-Splat-Shoot [20], M2BEV [25], and BEVFormer [12] for comparsion in the 100m × 100m setting, as shown in Tab. 3. The FPS of our method is measured on the RTX 3090 GPU.

We can see that the proposed method with a ResNet50 backbone even outperforms the BEVFormer model with a ResNet101DCN [5, 24] backbone. In the `road` class, our method outperforms the previous SOTA BEVFormer

Table 7: Comparison on NuScenes with the **160m** × **100m** setting. We reimplement other methods with the same setting for comparison. All results are reported with the format of Vanilla split / City-based split.

Method	Years	Backbone	mIoU (Easy)				mIoU (Hard)			
			Divider	Crossing	Boundary	All	Divider	Crossing	Boundary	All
VPN	IROS20	ResNet50	25.4 / 8.3	6.7 / 0.5	25.3 / 14.6	19.1 / 7.8	13.4 / 2.9	4.3 / 0.0	13.1 / 6.5	10.3 / 3.1
LSS	ECCV20	ResNet50	11.3 / 6.4	0.3 / 0.2	10.8 / 4.4	7.5 / 3.7	6.0 / 1.2	0.4 / 0.2	6.2 / 1.1	4.2 / 0.8
BEVFormer	ECCV22	ResNet50	42.2 / 16.1	26.9 / 7.6	42.1 / 18.6	37.1 / 14.1	27.3 / 7.8	17.5 / 2.3	26.3 / 10.0	23.7 / 6.7
UniFusion	-	ResNet50	**46.3 / 18.5**	**30.5 / 10.5**	**45.8 / 21.0**	**40.9 / 16.7**	**28.1 / 8.8**	**17.6 / 2.7**	**26.9 / 10.2**	**24.2 / 7.2**

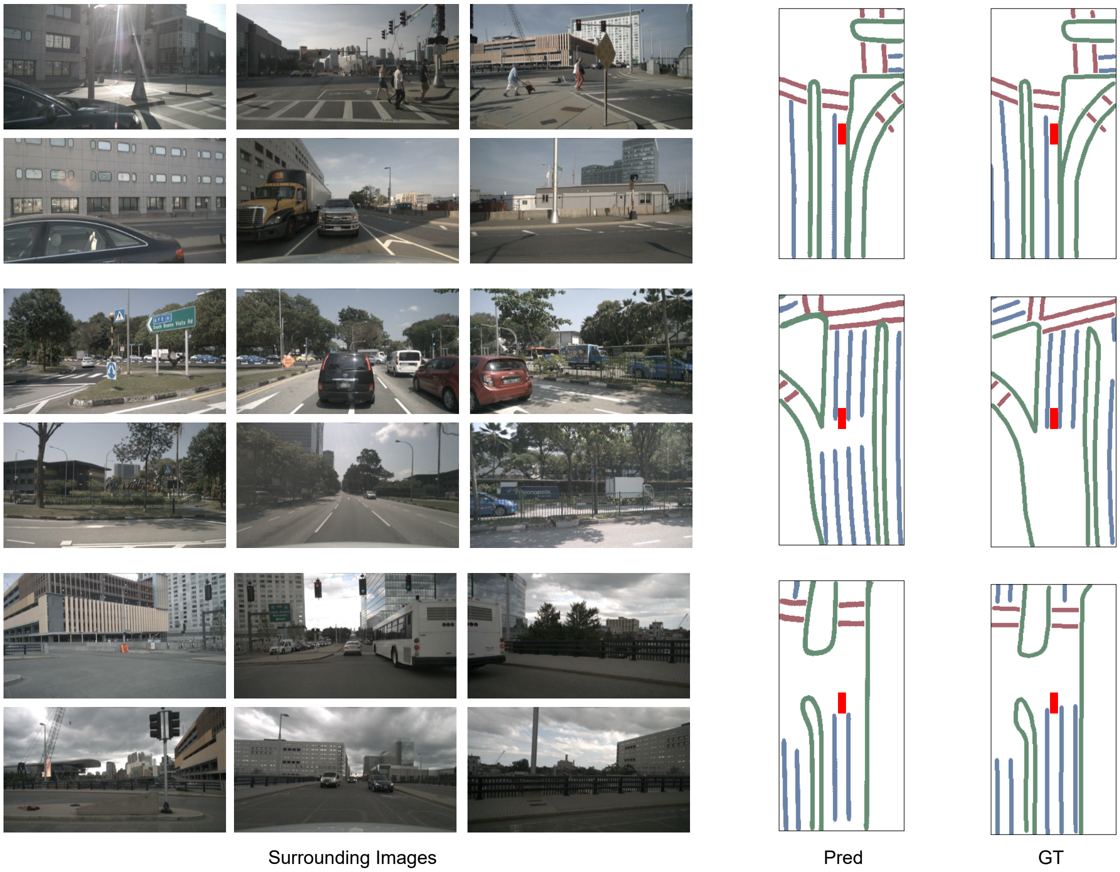


Figure 6: Visualization of our method on NuScenes *val* set under complex road structures with the 60m × 30m setting. From left to right, there are surrounding images, predictions, and ground truth. The red rectangle represents the ego car.

by 1.9 points with the vanilla split. It is worth mentioning that BEVFormer uses much more BEV queries than ours (200 × 200 vs. 50 × 50), which could benefit the segmentation of thin lane lines. But our method still outperforms BEVFormer in the `lane` class with a smaller backbone and fewer BEV queries, which shows the effectiveness of the proposed UniFusion. Besides, our method also achieves the fastest speed compared with BEVFormer and M2BEV. Finally, our method with a larger VoVNet99 backbone outperforms BEVFormer by more than 5 points in all classes.

For the 60m × 30m setting, we adopt VPN [17], Lift-Splat-Shoot [20], HDMapNet [11], BEVSegFormer [19], and BEVerse [26] for comparsion. The comparison results are shown in Tab. 4. From Tab. 4, we can see that our method still obtains the best results in all settings.

In order to better evaluate different models and provide a scenario that is closer to real-world autonomous driving, we also introduce a new 160m × 100m setting. We use VPN [17], LSS [20], BEVFormer [12], and our method with the same training setting for comparison, as shown in Tab. 7.

From Tab. 7 we can see that visible range is crucial for the map segmentation task. And the relatively low perfor-

Table 8: Experiments on dynamic objects. TS means adding timestamp to the network to indicate the time information.

Method	Backbone	TS	Car	Vehicles	Road	Lane
LSS	EffNetb0	-	32.1	32.1	72.9	20.0
LSS	Res101DCN	-	42.1	41.7	77.7	20.0
VPN	Res101DCN	-	31.0	31.8	76.9	19.4
BEVFormer	Res101DCN	-	44.8	44.8	80.1	25.7
UniFusion	Res101DCN		42.1	42.4	81.2	25.8
UniFusion	Res101DCN	✓	44.3	44.7	82.2	26.3
UniFusion	VoVNet99		44.9	46.4	84.8	29.8
UniFusion	VoVNet99	✓	47.3	48.3	85.3	30.0

mance suggests that large-range real-world map segmentation is still an open problem. Finally, we can see our method still obtains the best performance.

It should be noted that the vanilla NuScenes *train*/*val* sets contain many similar samples, and it is likely to be influenced by overfitting. In this way, we introduce the new city-based split for NuScenes, the results can be seen in Tabs. 3, 4 and 7. We can see that with the city-based split, all methods' performance drops significantly, and the poor improvement of VoVNet in Tab. 4 with the city-based split also indicates the problem of overfitting. This could be an important direction for future works.

In order to evaluate the method's performance on dynamic objects, we show the vehicle segmentation comparison in Tab. 8. Since dynamic objects can move and the alignment between temporal features is not as ideal as static fusion, we add an extra timestamp to help the segmentation. To import timestamps, we divide 12 ticks per second (same as the collecting frequency of NuScenes [4]), and use a set of learnable embeddings to indicate the delta T. The timestamp embeddings are then fed into the network as part of the positional embeddings of features.

from Tab. 8 we can see that our method still gets comparable performance with other methods and is still effective in dynamic objects. This is because although the features of moving objects might not be aligned temporally, the proposed deformable self-attention could learn the offsets and gather the moving features to the current step. With the added timestamp, the performance of vehicle segmentation can be further boosted. Since the static elements like road and lane are already aligned temporally, the performance gain of adding timestamp is relatively low compared with dynamic objects.

At last, we show the visualization results of our method, as shown in Fig. 6. We can see that our method gains good results under complex road structures. Our method could even segment the parts that are missing in the ground truth, as shown in the second row. Moreover, for the irregular road boundary, our method still gains good results.

5. Conclusion

In this work, we propose a unified spatial-temporal fusion method for BEV representation, termed UniFusion. Different from previous methods that use warpping, we propose a new concept, *i.e.*, virtual views that merge both spatial and temporal fusion in a unified formulation. With this design, we can realize long-range and adaptive temporal fusion with no information loss. The experiments and visualizations validate the effectiveness of our method.

Acknowledgements

This work is supported in part by National Key Research and Development Program of China under Grant 2020AAA0107400, National Natural Science Foundation of China under Grant U20A20222,National Science Foundation for Distinguished Young Scholars under Grant 62225605, The Ng Teng Fong Charitable Foundation in the form of ZJU-SUTD IDEA Grant, 188170-11102, Zhejiang Key Research and Development Program under Grant 2023C03196, and sponsored by CCF-AFSG Research Fund, CAAI-HUAWEI MindSpore Open Fund as well as CCF-Zhipu AI Large Model Fund (CCF-Zhipu202302).

References

[1] Mohamed Aly. Real time detection of lane markers in urban streets. In *IV*, pages 7–12. IEEE, 2008. 2

[2] Massimo Bertozzi and Alberto Broggi. Real-time lane and obstacle detection on the gold system. In *IV*, pages 213–218. IEEE, 1996. 2

[3] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *WACV*, pages 3287–3297, 2023. 3

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 2, 5, 9

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 7

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[7] Liuyuan Deng, Ming Yang, Hao Li, Tianyi Li, Bing Hu, and Chunxiang Wang. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Trans. Intell. Transp. Syst.*, 21(10):4350–4362, 2019. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 5

[9] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 1, 3

[10] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *CVPRW*, 2019. 4, 5

[11] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, pages 1–7, 2022. 1, 3, 5, 7, 8

[12] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1, 3, 5, 6, 7, 8

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 4, 5

[14] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 3

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, pages 1–18, 2019. 5

[16] Larry Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *IJCV*, 8(1):71–91, 1992. 2

[17] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE robot. autom. lett*, 5(3):4867–4873, 2020. 1, 3, 7, 8

[18] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, pages 3142–3152, 2021. 5

[19] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. *arXiv preprint arXiv:2203.04050*, 2022. 3, 5, 7, 8

[20] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210, 2020. 1, 3, 5, 7, 8

[21] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *BMVC*, 2019. 3

[22] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst.*, 19(1):263–272, 2017. 5

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, pages 1–15, 2017. 4

[24] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, pages 913–922, 2021. 7

[25] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 5, 7

[26] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 3, 5, 7, 8

[27] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, pages 1–16, 2020. 3, 4