Article | Received 19 December 2025; Revised 12 January 2026; Accepted 14 January 2026; Published 26 January 2026
https://doi.org/10.55092/rl20260004

InstaDrive: street view generation based on the unified instance segmentation input of vehicles and map elements

Qi Wang, Yizhou Wang and Hesheng Wang*

Department of Automation, Shanghai Jiao Tong University, Shanghai, China

* Correspondence author; E-mail: wanghesheng@sjtu.edu.cn.

Highlights:

- It is proposed to project the vehicle bounding box and map element annotations into 2D instance segmentation as the control condition, ensuring multi-view consistency from the input.
- Unify the instance segmentation input of the vehicle bounding box and map elements onto one image to express the occlusion relationship from the 2D perspective.
- Design an efficient instance segmentation encoder with instance segmentation invariance. Exchanging the order of different instance segmentation ids can also ensure that the output result remains unchanged.
- InstaDrive has achieved relatively good editing effects on the positions of vehicles and map elements on the public dataset.

Abstract: Aiming at the problems of cumbersome manual annotation and long-tail distribution of data in the training of autonomous driving detection models, this paper proposes the InstaDrive method. This method takes 3D bounding boxes of vehicles, vectorized annotations of map elements from the perspective of BEV, external parameters of the camera and text prompt as inputs, and encodes them as control conditions—generating 2D instance segmented images through projection Then control Stable Diffusion to generate a panoramic camera image of the autonomous driving scene. This method can efficiently generate a large amount of labeled training data and specifically edit scene elements to build corner Cases. To address the consistency issue in multi-view image generation, InstaDrive ensures multi-view consistency from the input layer and uniformly expresses the 2D view occlusion relationship. Experiments on the nuScenes dataset show that this method has achieved excellent results in the task of editing the positions of vehicles and map elements.

Keywords: driving scene generation; instance segmentation; Stable Diffusion; ControlNet

1. Introduction

The generative models [1–3] can generate images and other content based on control conditions such as text. The content generated by these models has been used for object detection [4] and semantic

segmentation [5]. 2D bounding boxes [6,7] or segmentation maps [8] All have control conditions that are adopted as input [9].

The manual annotation of data required for the training of autonomous driving detection models is cumbersome, and the actual collected data has the characteristic of long-tail distribution. InstaDrive projects the vehicle bounding box and map elements into instance segmented images as control conditions and generates images through Stable Diffusion. This not only provides a large amount of labeled training data for other models, but also enables targeted editing of vehicle and map elements in the scene to generate corner cases.

Specifically, as shown in Figure 1, InstaDrive takes the 3D bounding boxes of the vehicle, vectorized annotations of map elements from the perspective of BEV, camera parameters and text prompt as inputs, and encodes these inputs as control conditions. Control Stable Diffusion to generate panoramic camera images of the autonomous driving scene.

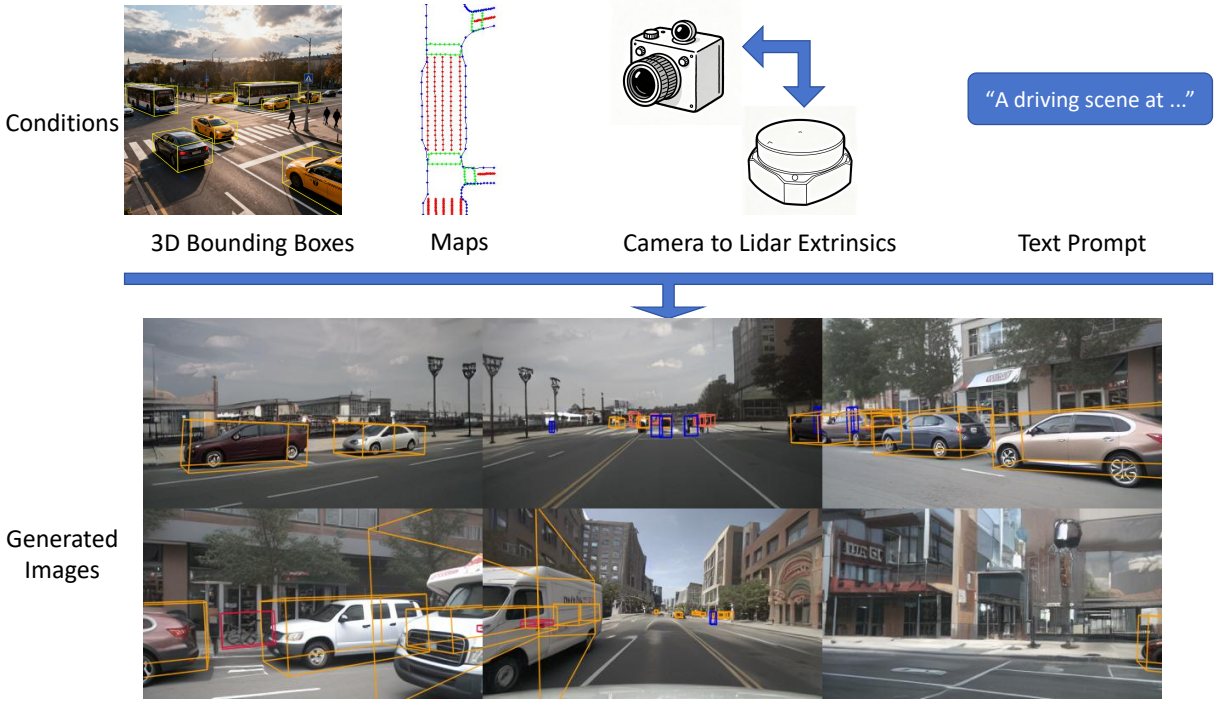

Figure 1. The street view generation task. Taking 3D bounding boxes, maps, camera extrinsics and text prompt as input, generate six panoramic camera images from different perspectives.

The challenge of this task lies in the consistency between different perspectives of the surround-view camera image. If the camera images from each perspective are generated as separate tasks, it is difficult to achieve precise stitching among the generated results from different perspectives.

The consistency of instances across multiple perspectives is also difficult to guarantee. As shown in Figure 2, in the demonstration of the Panacea [10] method, the same lane line in the front view and back view presents different forms: yellow line and white line.

The contributions of InstaDrive can be summarized in the following three points:

- It is proposed to project the vehicle bounding box and map element annotations into 2D instance segmentation as the control condition, ensuring multi-view consistency from the input.

- Unify the instance segmentation input of the vehicle bounding box and map elements onto one image to express the occlusion relationship from the 2D perspective.
- Design an efficient instance segmentation encoder with instance segmentation invariance. Exchanging the order of different instance segmentation ids can also ensure that the output result remains unchanged.
- InstaDrive has achieved relatively good editing effects on the positions of vehicles and map elements on the nuScenes [11] dataset.

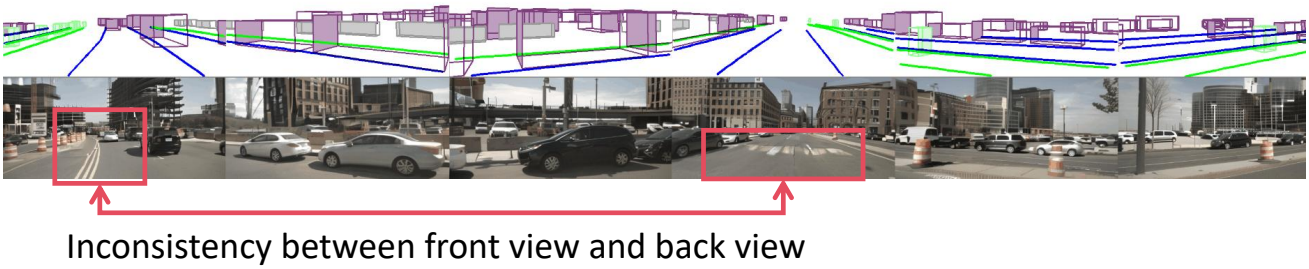

Figure 2. The main challenge is the consistency between different perspectives.

2. Related works

To achieve the editing of vehicle positions and ground elements on 2D images, it can be divided into two approaches: local editing and overall generation.

2.1. Local editing using Inpaint

The local editing method can mainly be implemented based on Inpaint [12]. This method first uses the Segment Anything [13] model on the image that needs to be edited to obtain the semantic segmentation result. When splitting, the prompt can be either a dot or a box. After Segment Anything obtains the semantic segmentation mask of the area to be edited, Inpaint Anything then calls methods such as Stable Diffusion [14] to generate locally. The network designs a loss function based on the purpose that the pixel values of the generated local region boundary are close to those of the original image boundary. This ensures that the local editing area can smoothly fit the original image.

The Inpaint Anything method will encounter the following three challenges when editing the lane lines in the autonomous driving scene:

- The lane line structure is slender, and it is difficult to obtain an accurate segmentation result of Segment Anything, as shown in Figure 3b, which further leads to a poor editing effect.
- Since this method requires that the pixels at the intersection of the generated area and the original image be as close as possible, the color of the lane lines generated in the slender area may be too close to the color of the road surface, thus making it impossible to achieve effective editing.
- This method can only be used for editing a single image and cannot solve the problem of multi-view consistency when generating surround-view camera images simultaneously. To sum up, a better solution is the overall generation based on Stable Diffusion.

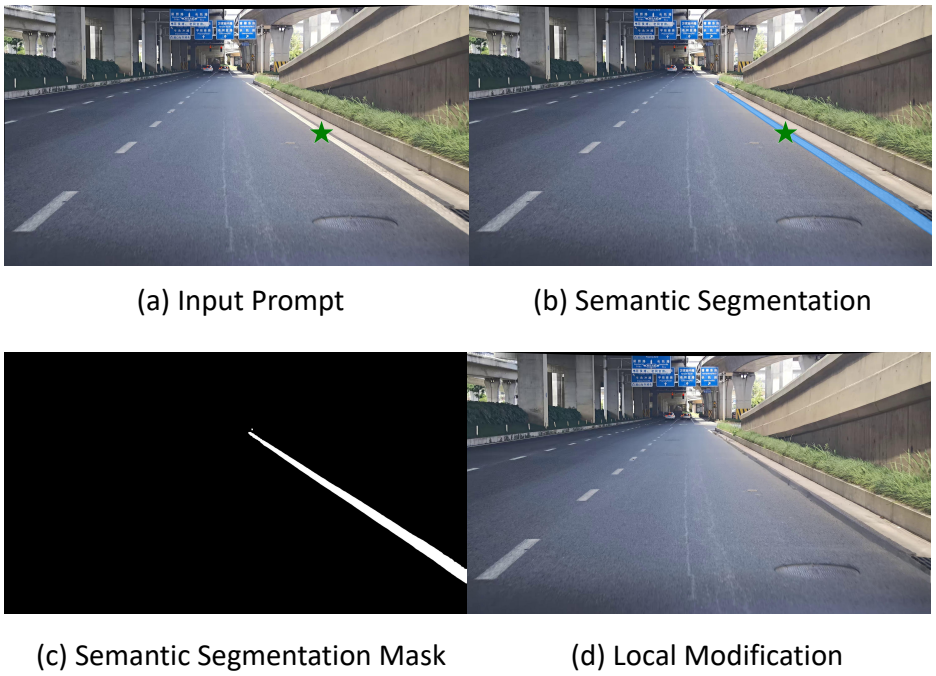

Figure 3. The process of local editing using Inpaint. **(a)** Input the prompts for the semantic segmentation model; **(b)** The semantic segmentation results for the edited lane lines; **(c)** The semantic segmentation mask; **(d)** The result of using Inpaint to restore the lane lines followed by further local modifications.

2.2. Global generation using Stable Diffusion

The overall generated scheme is mainly based on the structure of ControlNet [15], taking the content that needs to be edited as control conditions to affect the generation process of Stable Diffusion. Through training with a large amount of data, the generative model learns the correspondence between the input control conditions and the generated surround-view images. In the map editing problem studied in this paper, there can be four input methods as shown in Figure 4. The annotations for other traffic participants in the nuScenes [11] dataset are given in the form of 3D bounding boxes. The dataset is labeled for map elements in a vectorized form from the perspective of BEV.

2D semantic segmentation form, as shown in Figure 4a. Methods represented by [16], DrivingDiffusion [17], and Panacea [10] project map elements labeled in vectorized form from the original BEV perspective onto the 2D image perspective through external and internal parameters. These methods take mask images of different types of elements as control conditions, and encode the conditions into vectors through the image encoder and input them into Stable Diffusion. In addition, DriveDreamer-2 [18] has designed a tool for generating input maps, enabling users to obtain map control conditions through text prompts even without map annotations.

The advantage of the 2D semantic segmentation form input map is that the input conditions and the expected output are in the same coordinate system, and Stable Diffusion can easily directly learn the corresponding relationship between the two. However, the drawback of this input is that it cannot guarantee multi-perspective consistency from the fundamental input format, as semantic segmentation does not distinguish different instances.

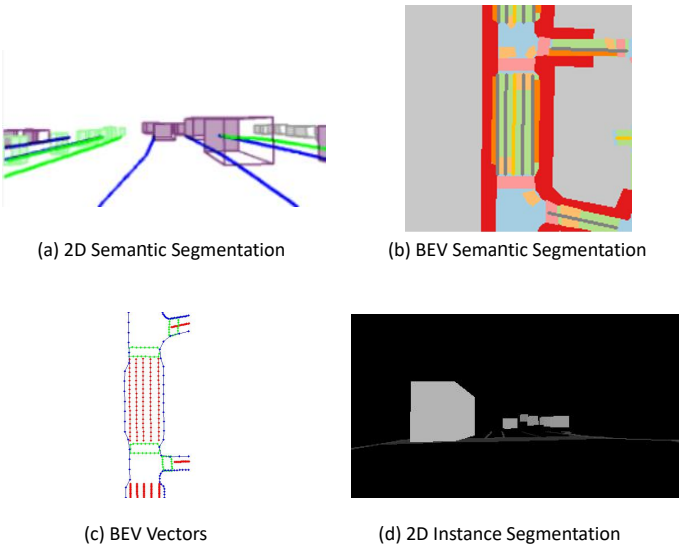

Figure 4. Four different input formats for map elements. **(a)** 2D semantic segmentation. Failed to distinguish different instances and lacks consistency across perspectives; **(b)** BEV semantic segmentation. The control conditions and the generated results are not in the same coordinate system, making it difficult to establish the corresponding relationship; **(c)** BEV vectors. Vector input is difficult to be encoded reasonably; **(d)** 2D instance segmentation. It can distinguish different instances, generate images in the same coordinate system, and is conducive to learning the corresponding relationship.

The semantic segmentation map form from the BEV perspective, as shown in Figure 4b. One of the most representative ways to use this format is MagicDrive [19]. To maintain consistency across multiple perspectives, MagicDrive inputs the same map control conditions when generating images from each perspective, that is, the semantic segmentation map from the BEV perspective. However, this also leads to a significant gap between the BEV coordinate system where the input map is located and the 2D image coordinate system where the final generated result is located. According to the experimental results, MagicDrive has a relatively weak editing ability for map elements. MagicDriveDiT [20] adopts the DiT architecture to enhance the quality of generation, but it does not fundamentally solve the multi-perspective consistency issue in map element editing.

In addition, MagicDrive3D [21] and DriveDreamer4D [22] also adopt the use of generated results for reconstruction, thereby optimizing the multi-view consistency of the generation process. This introduces a considerable amount of additional computation.

Input in the original vector form, as shown in Figure 4c. Such input computational load is very small and does not require an image encoder. However, currently, no work adopts this input method. On the one hand, it is difficult to encode vectors as control conditions. On the other hand, the relationship between the vectors in the BEV coordinate system and the final 2D image is too indirect, compared with the input method of projecting vectors onto 2D images.

The instance segmentation form input adopted in this paper is shown in Figure 4d. This format combines the advantages of both 2D semantic segmentation and semantic segmentation from the BEV perspective. The control conditions and the expected output are in the same coordinate system, so this one-to-one correspondence is relatively easy to learn. The difference between instance segmentation and semantic segmentation lies in that the information between different instances is retained during

the projection process. For instance, the same lane line area that appears from two perspectives has the same value. The values of different lane lines are different, rather than, as in semantic segmentation, different instances of the same category are not distinguished. This input method fundamentally ensures multi-perspective consistency.

3. Method

3.1. Overall architecture

The overall framework of InstaDrive is shown in Figure 5. To the left of the orange dotted line is the encoding of the input conditions. To the right of the orange dotted line is the generation process of Stable Diffusion after accepting input conditions. Four different input conditions, after passing through their respective encoders, act as conditions on the denoising process of the initial noise on the right. The annotations of other vehicles and map elements are first constructed into a unified instance segmentation input to describe the positions of other vehicles and map elements in the generated panoramic image, which is the focus of this article's editing. The camera external parameters are input to represent the geometric relationship between the surround-view cameras. Text prompts play an auxiliary role in editing the overall style of the scene, weather, *etc.*

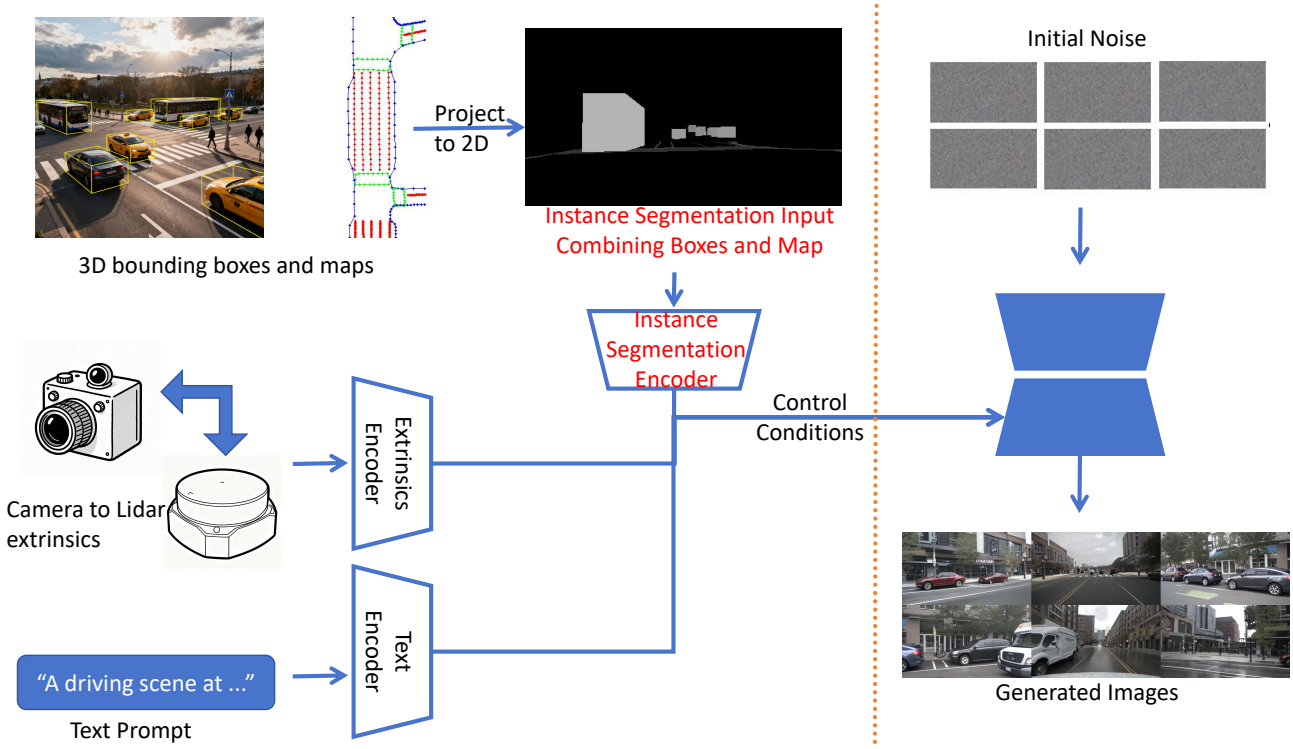

Figure 5. The overall framework of InstaDrive. To the left of the orange dotted line is the encoding of the input conditions. To the right of the orange dotted line is the generation process of Stable Diffusion after accepting the input conditions. We constructed a unified instance segmentation input for map elements and bounding boxes, and designed a control condition encoder with instance segmentation invariance, which affects the image generation process on the right side.

3.2. *Construction of instance segmentation input*

Considering the computational overhead and accuracy issues of invoking the instance segmentation network, in this paper, vehicles and map annotations are directly projected onto 2D images to construct the instance segmentation input. Since the generation process does not require pixel-level fine control, for instance, once the position of the vehicle is specified, the specific category and outline of the vehicle are freely determined by InstaDrive.

According to whether it is a closed area or not, the vectorized annotation of ground elements from the perspective of BEV can be divided into two categories: broken lines with lane lines as the main body and polygonal areas such as pedestrian crossings. For the former, the lane line width is fixed at 0.15 m, and the original polygonal marking is expanded to the same polygonal marking as the latter. In this way, it can be projected onto the corresponding area of the 2D image together with the latter, and then the mask of each map element instance can be generated.

For the annotation of the 3D bounding box of the vehicle, the vertices of the box are directly projected onto the 2D image through the internal and external parameters of the camera, and the corresponding instance segmentation mask can be obtained, as shown in Figure 6.

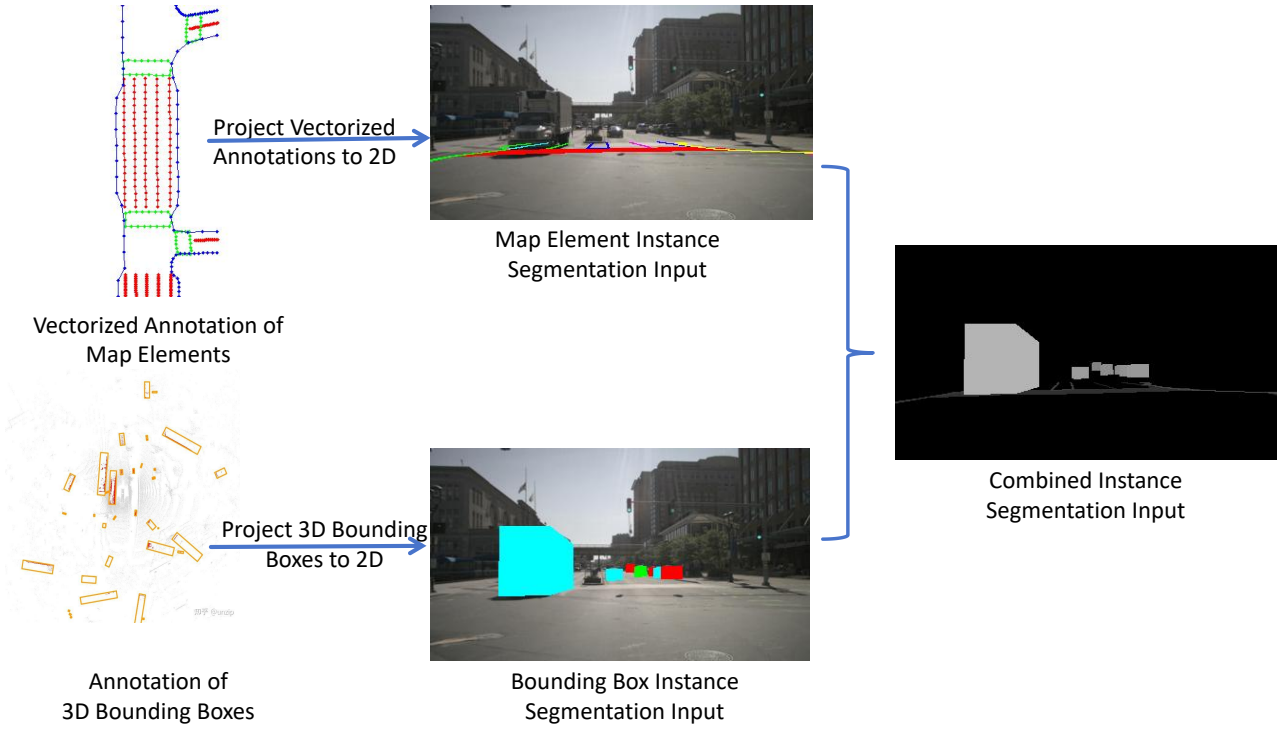

Figure 6. The construction process of instance segmentation input. The Vectorized Annotation of Map Elements is projected onto the 2D image to form the input for the segmentation of map element instances. The 3D bounding box inputs of other traffic participants are also projected onto the 2D image. To handle occlusion relationships, we combine the instance segmentation inputs of these two types of elements into one.

Since both the map and the vehicle need to be projected onto a 2D image to obtain the corresponding mask, these two inputs can be unified into a single image input encoder. Because the objects seen on 2D images often have mutual occlusion relationships, when constructing a unified input, it is necessary to

project the ground elements first, and then project the bounding box of each vehicle in the order from far to near. This way, the masks of vehicles closer to one's own can cover those in the distance. Thus, this method unifies the map elements and the vehicle bounding box onto a single instance segmentation image.

3.3. Design of instance segmentation encoder

Because the instance segmentation image distinguishes each instance with different ids, and the encoding results obtained by arranging different instances in different ID orders should be the same. Therefore, the encoder should not learn the size relationship between instance ids. The instance segmentation encoder used by InstaDrive is shown in Figure 7. After extracting the one-hot mask of each instance, it is sent to the basic feature extraction network respectively to obtain their respective feature maps. The basic feature extraction network structure is quite simple, consisting of only two convolution layers, with ReLU as the activation function after the convolution layers.

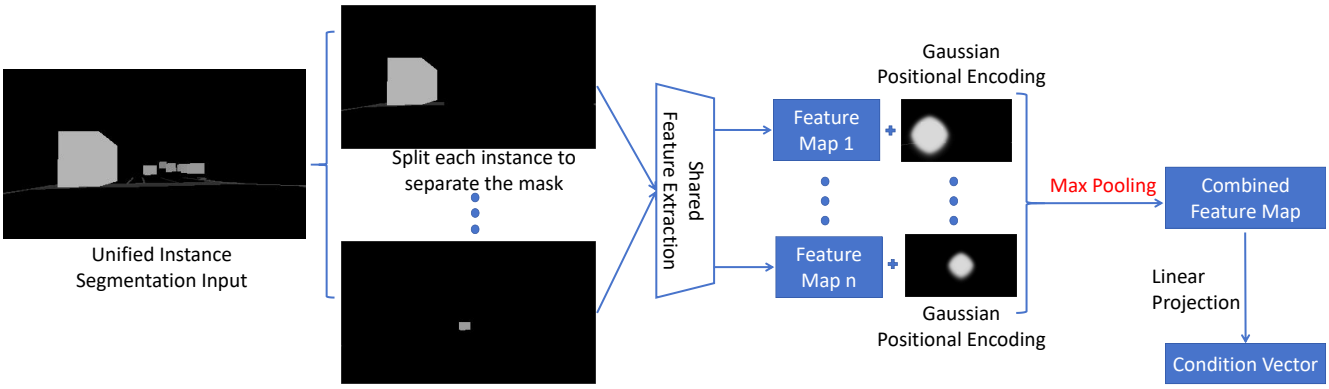

Figure 7. Design of instance segmentation encoder with invariance.

Then, based on the centroid coordinates of each instance's location, a Gaussian heat map is generated as the position code and added to the feature map. The Gaussian position encoding module calculates the centroid coordinates of the input instance based on its binary mask and maps these positions to the feature map scale. It then generates a heatmap using a two-dimensional Gaussian function. The core parameter settings are as follows: The standard deviation σ of the Gaussian kernel is an adaptive value, and the calculation formula is

$$\sigma = \frac{min(H_f, W_f)}{8}, \tag{1}$$

where H_f and W_f are the height and width of the feature map. This ensures that the diffusion range of the Gaussian distribution is proportional to the scale of the feature map. The value of the generated heatmap is calculated by the formula

$$G(x,y) = exp(-\frac{d^2}{2\sigma^2}), \tag{2}$$

where d is the Euclidean distance from each point on the feature map to the centroid. This heatmap does not undergo additional maximum value normalization because the value range of the Gaussian function is already limited to the $[0,1]$ interval, and its maximum value 1 is located at the centroid, thus ensuring the stability of the position encoding amplitude. The key step is to refer to the idea of PointNet [23,24], max pooling is performed on all feature maps to obtain the fused feature map, and finally the control vector

can be obtained through linear mapping. max pooling can ensure that the result obtained by encoding remains unchanged when the order of input instances changes.

Furthermore, to input the category information of the instances, we retained the semantic segmentation input. The semantic segmentation input is similar to the instance segmentation image, but the values in it represent the instance categories. Since the categories do not need to maintain invariance as in instance segmentation, the semantic segmentation input is simply fused with the instance segmentation feature map through four layers of convolution after feature extraction by six layers of convolution. The final linear mapping unifies the output dimension to 32 dimensions for concatenation with other control conditions.

4. Eperiments and discussions

4.1. Experimental setups

Dataset and Baselines. Experiments were conducted using the nuScenes [11] public autonomous driving dataset. It contains 850 training set scenarios and 150 validation set scenarios. A generative network was trained using a total of 28,130 samples from the training set. The trained network generates 6,019 sets of surround-view camera images for the validation set and compares them with the real images. For the visualization results, we attempted experiments such as removing all vehicles and removing some lane lines to verify the editing ability of this method for vehicles and lane lines. In addition, in order to measure the superiority of this method in lane line editing more directly and quantitatively, the pre-trained model of MapTR [25], a classic method for detecting high-precision maps by inputting surround-view camera images, is selected. High-precision map detection results were obtained by reasoning respectively on real images, MagicDrive [19] generated images and InstaDrive generated images using this model. The differences in results reflect the high-precision map editing and control capabilities. BEVGen [26], BEVControl [27], MagicDrive [19], DriveDreamer [18], DrivingDiffusion [17], Panacea [10], and Panacea+ [10] were selected as baselines for quantitative result comparison.

Evaluation Metrics. Like other methods, we choose Frechet Inception Distance (FID) as the main measurement indicator. FID reflects the authenticity of the generated data by measuring the differences between the generated image and the real image sent into a specific backbone—Google Net Inception V3 [28] encoding to obtain the feature map. For the reasoning task of MapTR in generating results, we choose the mAP of MapTR as the measurement metric. This metric takes into account the combination of precision and recall under different confidence levels and can measure the accuracy of high-precision map detection.

Model Setup. To be consistent with MagicDrive, the InstaDrive model also uses Stable Diffusion v1.5 [14] as the generative model and UniPC [29] as the sampler within it. The resolution of the generated result is set to 224*400. Train for 100 epochs. For the MapTR model, the pre-training weights use ResNet50 [30] as the backbone to train the MapTR tiny model for 110 epochs.

4.2. Generate and edit results

As shown in Figure 8, when the instance segmentation projection of the complete ground elements and vehicle bounding boxes is input, a panoramic camera image containing all elements can be generated accordingly, and the positions of the vehicle and ground elements in it are consistent with the positions specified in the input.

When the input conditions are modified, the output image also changes accordingly. As shown in Figure 9, if all the vehicle bounding boxes in the input are removed, all the vehicles in the output image are also removed, while the ground elements remain unchanged.

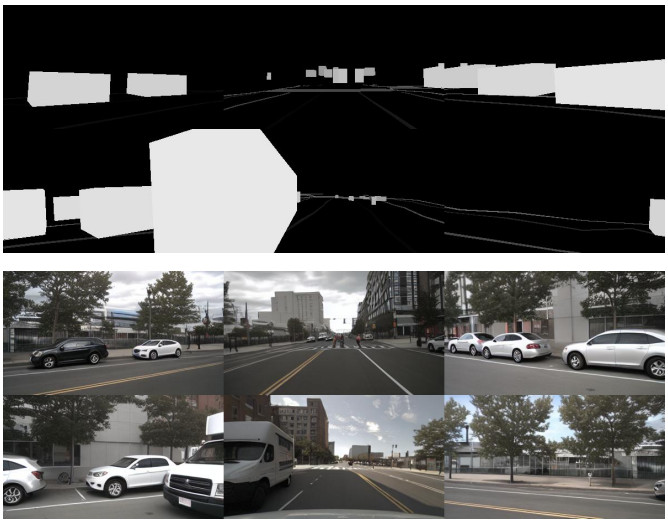

Figure 8. Generating result with complete input.

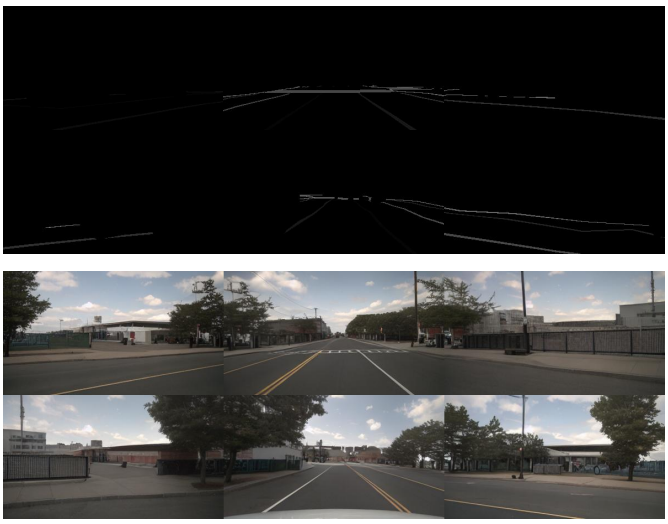

Figure 9. Generating result with no bounding boxes.

Next, verify the modifications made by InstaDrive to individual map element instances. As shown in Figure 10, after removing the vecquantized lane line input at the very beginning, the mask of this lane line in the front view and back view where the instance segmentation is projected is removed. In the final generated image, this lane line was also removed simultaneously in the front view and back view, while the positions of other elements remained unchanged.

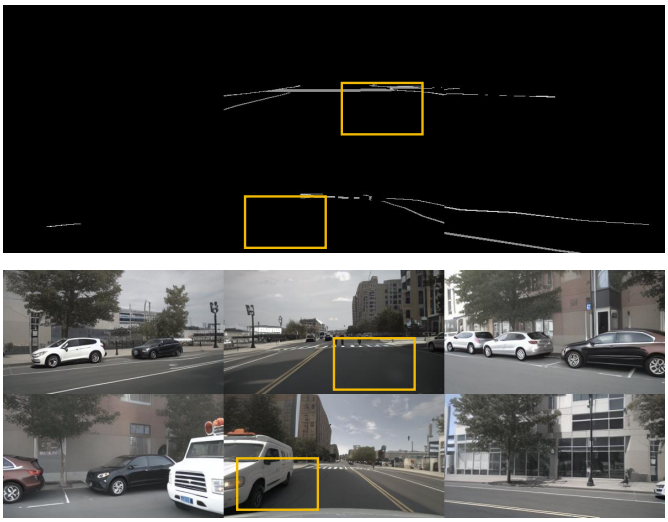

Figure 10. Generating result after removing a white line.

We have selected another rainy T-shaped intersection for the visual demonstration. The weather is controlled through text. Figure 11 shows the initial annotations and the corresponding generated results. Figure 12 adds a truck in the left-front camera view by modifying the conditions. Figure 13 removes the lane center line in the rear-view camera.

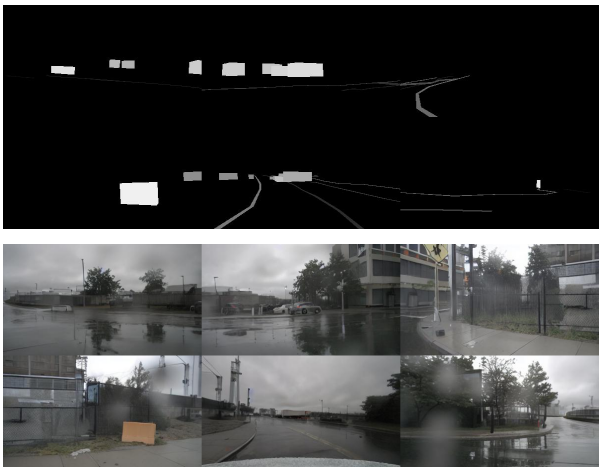

Figure 11. Generating result of a rainy day with complete input.

Figure 12. Generating result of a rainy day after adding a truck in the left-front camera.

Figure 13. Generating result of a rainy day after removing the center line in the rear camera.

The FID metric comparison of the results generated by InstaDrive on the nuScenes validation set with other baseline results is shown in Figure 14.

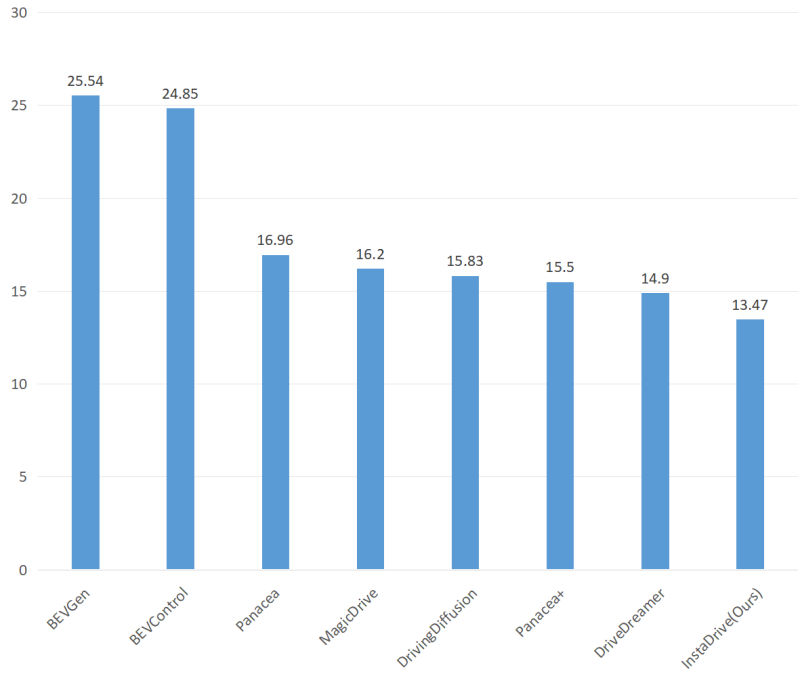

Figure 14. Comparison of FID↓ indicators by different methods

4.3. Performance in the high-precision map detection task

We respectively tested the inference results of the MapTR model on the data generated by Panacea+, MagicDrive and InstaDrive, and compared these results with the ground truth annotations (Table 1). These quantitative results demonstrate the superiority of InstaDrive in generating map elements. However, since the detection accuracy of the MapTR model itself cannot reach one hundred percent, and what we need to measure is the difference between the generated data and the real data, a fairer approach is to take the result inferred by MapTR on the real data as the ground truth and then measure the difference between the inference result of MapTR on the two types of generated data and this ground truth (Table 2).

Table 1. The matching degree with the ground truth annotation (mAP).

Method	Divider	Ped Crossing	Boundary	Total
Panacea+	0.006	0.000	0.006	0.004
MagicDrive	0.194	0.120	0.181	0.165
InstaDrive (Ours)	0.206	0.122	0.240	0.189

Table 2. The matching degree with the predicted results on real data (mAP).

Method	Divider	Ped Crossing	Boundary	Total
Panacea+	0.015	0.004	0.013	0.011
MagicDrive	0.145	0.071	0.108	0.108
InstaDrive (Ours)	0.157	0.077	0.132	0.122

Select one of the scenes as shown in Figure 15. In this scene, since MagicDrive did not correctly generate the structure of the right turn road based on the input, it led to a significant deviation between the inference results of the MapTR model on the data generated by MagicDrive and those of the MapTR model on the real data. The data generated by InstaDrive, compared with the real data, retains the road structure while removing the obstruction of the right fence. Therefore, the results detected by MapTR will be better.

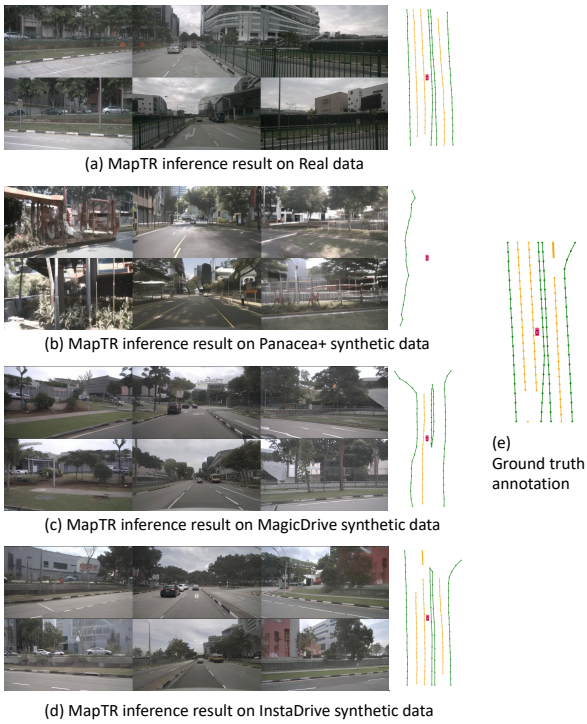

Figure 15. Comparison of inference results of MapTR on different data. **(a)** MapTR inference result on real data. There is a fence on the right side of the actual data, which prevented the correct detection of the right-side fork road; **(b)** MapTR inference result on Panacea+ synthetic data. The synthetic data failed to meet the input control conditions, thus the detection results were not satisfactory; **(c)** MapTR inference result on MagicDrive synthetic data. The synthetic effect still needs to be improved; **(d)** MapTR inference result on InstaDrive synthetic data. Restore to the original road structure, and the detection results have a high consistency with the ground truth; **(e)** Ground truth annotation.

As shown in the first row of the Table 3 and Table 4, if there is only semantic segmentation input but no instance segmentation input, which is similar to the Panacea method, it will lead to a deterioration in the cross-view consistency of map elements, thereby affecting the FID metric as well as the mAP metric in the high-precision map detection task. This proves that compared to only inputting semantic segmentation, simultaneously inputting semantic categories and the instance segmentation constructed in InstaDrive will enhance the cross-view consistency of the generated results, thereby enhancing the network's controllable editing ability for map elements in the generated results.

Table 3. The results of ablation study compared with ground truth annotation.

Method	FID	Comparison With Ground Truth			
		Divider	Ped Crossing	Boundary	Average
only semantic input	16.47	0.170	0.119	0.213	0.168
convolutional encoding	13.31	0.177	0.105	0.214	0.165
InstaDrive	13.47	0.206	0.122	0.240	0.189

Table 4. The results of ablation study compared with predicted results.

Method	FID	Comparison With The Predicted Results On Real Data			
		Divider	Ped Crossing	Boundary	Average
only semantic input	16.47	0.136	0.068	0.114	0.106
convolutional encoding	13.31	0.140	0.065	0.118	0.108
InstaDrive	13.47	0.157	0.077	0.132	0.122

The result in the second row of the Table 3 and Table 4 is the result obtained by using instance segmentation input but using 10 convolution layers as the encoder. Since no encoder with instance segmentation consistency as described in InstaDrive was used, the network learned the relationship between different instance values. However, this relationship is for generating only noise and is not conducive to learning. After this modification, the FID metric and the results of the complete method are very close, but in the high-precision map detection task, the mAP metric significantly decreases. This further indicates that merely looking at the FID metric is not sufficient, and the mAP in the high-precision map detection task is a very important metric for evaluating the generation effect of map elements.

In conclusion, the above ablation experiments respectively prove the positive effects of the instance segmentation input proposed in InstaDrive and the encoder with instance segmentation invariance.

5. Conclusions

This paper focuses on the annotation efficiency of training data for autonomous driving models and the pain points of long-tail distribution, and proposes the InstaDrive generative data construction scheme. The core innovation lies in converting the multi-dimensional physical information of the autonomous driving scene (such as 3D bounding boxes of vehicles, BEV map elements, *etc.*) into a unified control condition of 2D instance segmentation. Combined with Stable Diffusion, it realizes the controllable generation of the surround-view camera image, which not only solves the cumbersome problem of manual annotation, It can also proactively build key corner case data. To address the consistency challenge of multi-view generation,

InstaDrive effectively avoids the problems of view splicing deviation and inconsistent cross-view instance forms through the condition design of the input layer. Ultimately, experiments on the nuScenes dataset verified the effectiveness of this method in editing the positions of vehicles and map elements, providing a feasible path for the large-scale and high-quality construction of autonomous driving training data.

Acknowledgments

We are grateful to every reviewer and the editor-in-chief for their efforts. At the same time, we are thankful for the efficient and professional coordination work of the editorial department, which ensured the smooth progress of the review process. Here, we sincerely express our most heartfelt gratitude to all the teachers and professors who have contributed their efforts to this article.

Data availability statement

The data or datasets generated or analyzed in this study are available in https://www.nuscenes.org/.

Authors' contribution

Conceptualization, Qi Wang; resources, Qi Wang; data curation, Qi Wang; software, Qi Wang; formal analysis, Qi Wang; supervision, Hesheng Wang; validation, Qi Wang; investigation, Qi Wang; visualization, Qi Wang; methodology, Qi Wang; Writing—original draft, Qi Wang; project administration, Hesheng Wang; writing—review & editing, Yizhou Wang. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

Hesheng Wang holds the position of Editor-in-Chief for *Robot Learning* and has not peer reviewed or made any editorial decisions for this paper.

References

[1] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* 2020, 33:6840–6851.

[2] Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, *et al.* Score-based generative modeling through stochastic differential equations. *arXiv* 2020, arXiv:2011.13456.

[3] Zheng Z, Gao R, Xu Q. Non-cross diffusion for semantic consistency. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Tucson, USA, February 26–March 6, 2025, pp. 3897–3906.

[4] Chen K, Xie E, Chen Z, Wang Y, Hong L, *et al.* Geodiffusion: text-prompted geometric control for object detection data generation. *arXiv* 2023, arXiv:2306.04607.

[5] Wu W, Zhao Y, Chen H, Gu Y, Zhao R, *et al.* Datasetdm: synthesizing data with perception annotations using diffusion models. *Adv. Neural Inf. Process. Syst.* 2023, 36:54683–54695.

[6] Lin TY, Maire M, Belongie S, Hays J, Perona P, *et al.* Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, Zurich, Switzerland, September 6–12, 2014, pp. 740–755.

[7] Han J, Liang X, Xu H, Chen K, Hong L, *et al.* SODA10M: a large-scale 2D self/semi-supervised object detection dataset for autonomous driving. *arXiv* 2021, arXiv:2106.11118.

[8] Zhou B, Zhao H, Puig X, Xiao T, Fidler S, *et al.* Semantic understanding of scenes through the ade20k dataset. *Int. J. Comput. Vision* 2019, 127(3):302–321.

[9] Li Y, Liu H, Wu Q, Mu F, Yang J, *et al.* Gligen: open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, June 18–22, 2023, pp. 22511–22521.

[10] Wen Y, Zhao Y, Liu Y, Jia F, Wang Y, *et al.* Panacea: panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 17–21, 2024, pp. 6902–6912.

[11] Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, *et al.* Nuscenes: a multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 13–19, 2020, pp. 11621–11631.

[12] Yu T, Feng R, Feng R, Liu J, Jin X, *et al.* Inpaint anything: segment anything meets image inpainting. *arXiv* 2023, arXiv:2304.06790.

[13] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, *et al.* Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France, October 2–6, 2023, pp. 4015–4026.

[14] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, June 18–24, 2022, pp. 10684–10695.

[15] Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France, October 2–6, 2023, pp. 3836–3847.

[16] Wang X, Zhu Z, Huang G, Chen X, Zhu J, *et al.* Drivedreamer: towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, Milan, Italy, September 29–October 4, 2024, pp. 55–72.

[17] Li X, Zhang Y, Ye X. DrivingDiffusion: layout-guided multi-view driving scenarios video generation with latent diffusion model. In *European Conference on Computer Vision*, Milan, Italy, September 29–October 4, 2024, pp. 469–485.

[18] Zhao G, Wang X, Zhu Z, Chen X, Huang G, *et al.* Drivedreamer-2: LLM-enhanced world models for diverse driving video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Philadelphia, USA, February 25–March 4, 2025, pp. 10412–10420.

[19] Gao R, Chen K, Xie E, Hong L, Li Z, *et al.* Magicdrive: street view generation with diverse 3D geometry control. *arXiv* 2023, arXiv:2310.02601.

[20] Gao R, Chen K, Xiao B, Hong L, Li Z, *et al.* Magicdrivedit: high-resolution long video generation for autonomous driving with adaptive control. *arXiv* 2024, arXiv:2411.13807v1.

[21] Gao R, Chen K, Li Z, Hong L, Li Z, *et al.* Magicdrive3D: controllable 3D generation for any-view rendering in street scenes. *arXiv* 2024, arXiv:2405.14475.

[22] Zhao G, Ni C, Wang X, Zhu Z, Zhang X, *et al.* Drivedreamer4D: world models are effective data machines for 4D driving scene representation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, Nashville, USA, June 11–15, 2025, pp. 12015–12026.

[23] Qi CR, Su H, Mo K, Guibas LJ. Pointnet: deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, July 21–27, 2017, pp. 652–660.

[24] Qi CR, Yi L, Su H, Guibas LJ. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, USA, December 4–9, 2017.

[25] Liao B, Chen S, Wang X, Cheng T, Zhang Q, *et al.* MapTR: structured modeling and learning for online vectorized HD map construction. *arXiv* 2022, arXiv:2208.14437.

[26] Swerdlow A, Xu R, Zhou B. Street-view image generation from a bird's-eye view layout. *IEEE Rob. Autom. Lett.* 2024, 9(4):3578–3585.

[27] Yang K, Ma E, Peng J, Guo Q, Lin D, *et al.* Bevcontrol: accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv* 2023, arXiv:2308.01661.

[28] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 27–July 1, 2016, pp. 2818–2826.

[29] Zhao W, Bai L, Rao Y, Zhou J, Lu J. UniPC: a unified predictor-corrector framework for fast sampling of diffusion models. *Adv. Neural Inf. Process. Syst.* 2023, 36:49842–49869.

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 27–July 1, 2016, pp. 770–778.