DriveFlow: Rectified Flow Adaptation for Robust 3D Object Detection in Autonomous Driving

Hongbin Lin[1,2], **Yiming Yang**[1,2], **Chaoda Zheng**[3], **Yifan Zhang**[4], **Shuaicheng Niu**[5], **Zilu Guo**[1,2], **Yafeng Li**[6], **Gui Gui**[7], **Shuguang Cui**[2,1], **Zhen Li**[2,1] †

[1] Shenzhen Future Network of Intelligence Institute (FNii), The Chinese University of Hong Kong, Shenzhen
[2] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
[3] Xpeng Motors
[4] MiroMind AI
[5] Nanyang Technological University
[6] Baoji University of Arts and Sciences
[7] Central South University

Abstract

In autonomous driving, vision-centric 3D object detection recognizes and localizes 3D objects from RGB images. However, due to high annotation costs and diverse outdoor scenes, training data often fails to cover all possible test scenarios, known as the out-of-distribution (OOD) issue. Training-free image editing offers a promising solution for improving model robustness by training data enhancement without any modifications to pre-trained diffusion models. Nevertheless, inversion-based methods often suffer from limited effectiveness and inherent inaccuracies, while recent rectified-flow-based approaches struggle to preserve objects with accurate 3D geometry. In this paper, we propose ***DriveFlow***, a Rectified Flow Adaptation method for training data enhancement in autonomous driving based on pre-trained Text-to-Image flow models. Based on frequency decomposition, **DriveFlow** introduces two strategies to adapt noise-free editing paths derived from text-conditioned velocities. 1) High-Frequency Foreground Preservation: DriveFlow incorporates a high-frequency alignment loss for foreground to maintain precise 3D object geometry. 2) Dual-Frequency Background Optimization: DriveFlow also conducts dual-frequency optimization for background, balancing editing flexibility and semantic consistency. Extensive experiments validate the effectiveness and efficiency of DriveFlow, demonstrating comprehensive performance improvements across OOD scenarios.

Code — https://github.com/Hongbin98/DriveFlow

Introduction

Three-dimensional (3D) Object Detection constitutes a critical computer vision challenge, involving the identification and localization of objects within three-dimensional space using various sensing modalities (Li et al. 2022; Chen et al. 2023). Due to the economic advantages, vision-centric 3D detection has emerged as a prominent paradigm that leverages solely RGB images from single or multiple cameras, complemented by calibration information (Xu et al. 2023; Wang et al. 2023; Yan et al. 2024; Pu et al. 2025). Given the inherent challenges in vision-centric detection, existing methods (Oh et al. 2025; Lin et al. 2025b; Zhang et al. 2025; Li, Yang, and Lei 2025) have still achieved remarkable progress over various benchmarks (Geiger, Lenz, and Urtasun 2012; Caesar et al. 2020; Sun et al. 2020).

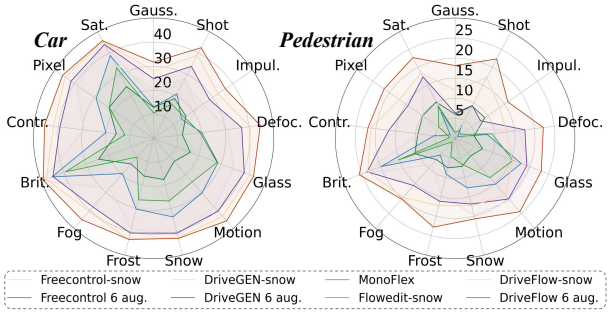

Figure 1: Comparison on KITTI-C based on MonoFlex. DriveFlow achieves 1) better performance with only Snow augmentation (orange) than DriveGEN with 6 aug. (purple) and 2) comprehensive gains on the minority class (Pedestrian) across OOD scenarios. Better viewed in color.

Such achievements mainly depend on one prerequisite: training data adequately covers all possible test scenarios. However, it is particularly challenging to satisfy this assumption since driving systems often operate continuously outdoors over extended periods. Once the system suffers from unexpected data changes, well-trained detectors often fail to maintain the performance due to the shifts between training and test data distributions, which is known as the out-of-distribution (OOD) issue (Wang et al. 2020). To illustrate this, we follow DriveGEN (Lin et al. 2025a) and visualize the performance degradation of a well-trained detector when deployed across different environmental conditions, as shown in Figure 2. The results clearly demonstrate that the detector achieves satisfactory performance under ideal conditions (daytime scenarios) while exhibiting significant performance deterioration in *unseen* scenes (e.g., fog). Therefore, it is essential to enhance the robustness of 3D Object

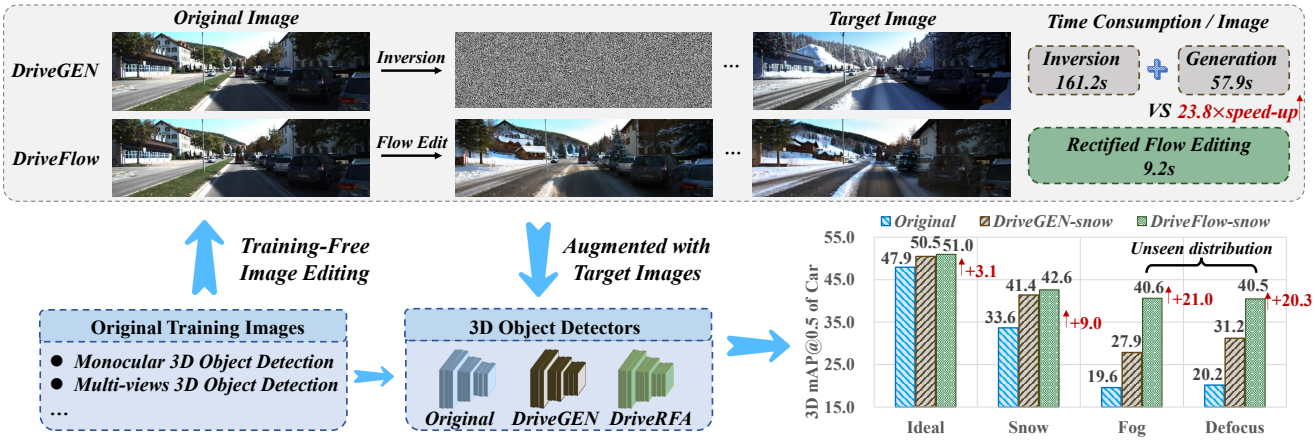

Figure 2: An illustration of DriveFlow for training data enhancement in vision-centric 3D object detection. In contrast to the inversion-based approach DriveGEN, DriveFlow conducts rectified flow adaptation based on pre-trained T2I flow models (*e.g.,* Stable Diffusion 3), thereby achieving comprehensive improvement and rapid generation for 3D detectors.

Detection models in systems, as unexpected performance degradation in OOD scenarios may pose severe safety risks.

To handle the OOD issues in autonomous driving, previous approaches either rely on test-time model adaptation (Lin et al. 2025b) or employ weather-adaptive diffusion models to transform adverse weather conditions to clear scenes (Oh et al. 2025), which introduces additional computational cost at test time. Prior work DriveGEN (Lin et al. 2025a) employs controllable T2I diffusion generation to augment training data, thereby enhancing the robustness of 3D detectors. However, DriveGEN requires image inversion (Song, Meng, and Ermon 2020) and relies on U-Net based pre-trained T2I diffusion models like Stable Diffusion 1.5 (Rombach et al. 2022). Previous methods (Kulikov et al. 2024; Wang et al. 2024) have shown that inversion-based editing produces unsatisfactory results regardless of whether ground-truth noise maps are available. Additionally, inversion-based approaches suffer from computational inefficiency (see Figure 2) since reverting to noise maps requires more time compared to rectified-flow-based editing methods (Kulikov et al. 2024). Recently, FlowEdit (Kulikov et al. 2024) shows that leveraging pre-trained Text-to-Image (T2I) flow models (*e.g.,* Stable Diffusion 3 (Esser et al. 2024) and FLUX (Labs 2024)) enables more powerful and efficient generation. However, FlowEdit may pose potential risks of object misalignment and omissions even if fine-grained text descriptions are available, as shown in Figure 3.

To address these challenges, we propose an image editing method termed ***DriveFlow***, which is training-free and controllable based on pre-trained T2I flow models. DriveFlow aims to enhance training images in autonomous driving via performing frequency-based decomposition and adaptation of noise-free editing paths derived from velocities. Specifically, DriveFlow consists of two strategies: 1) High-Frequency Foreground Preservation designs a foreground preservation loss for object regions to preserve accurate 3D geometry, while 2) Dual-Frequency Background Optimization introduces dual-frequency optimization to balance editing flexibility and semantic consistency of background regions. As shown in Figure 1, with only Snow augmentation, DriveFlow performs better than six augmentations of DriveGEN, demonstrating more comprehensive robustness improvement across both the majority (*i.e.,* Car) and minority class (*i.e.,* Pedestrian).

Contributions: 1) To the best of our knowledge, we are the first to apply rectified-flow-based editing for robust 3D object detection, offering novel perspectives on the usage of pre-trained T2I flow models in autonomous driving. 2) We propose DriveFlow which incorporates high-frequency foreground preservation and dual-frequency background optimization strategies, achieving rapid (*e.g.,* 23.8x faster on KITTI) and effective (*e.g.,* 14.54 mAP improvement on KITTI-C with only snow augmentation) training data enhancement. 3) Extensive experiments validate that DriveFlow brings comprehensive performance gains for both monocular and multi-view detectors. Moreover, DriveFlow enhances robustness even for temporal-based 3D detectors, demonstrating our broad applicability.

Related Work

We first review model robustness studies for 3D detectors and controllable T2I diffusion methods. Additional discussions on vision-centric 3D detection are in Appendix ***A***.

Robust 3D Object Detection. Visual detection serves as a fundamental component in autonomous driving perception systems, enabling essential understanding of surroundings like traffic sign recognition. Compared to LiDAR-based approaches, vision-centric 3D detectors offer lower hardware costs at the expense of model robustness. Recent approaches tackle this issue by: MonoWAD (Oh et al. 2025) adopts diffusion models to revert weather conditions to ideal situations, whereas MonoTTA (Lin et al. 2025b) improves model robustness via online test-time adaptation. Additionally, MagicDrive (Hong et al. 2021), Panacea (Sun et al. 2022), and GAIA (Hu et al. 2023; Russell et al. 2025) leverage generative models to synthesize multi-view 3D driving

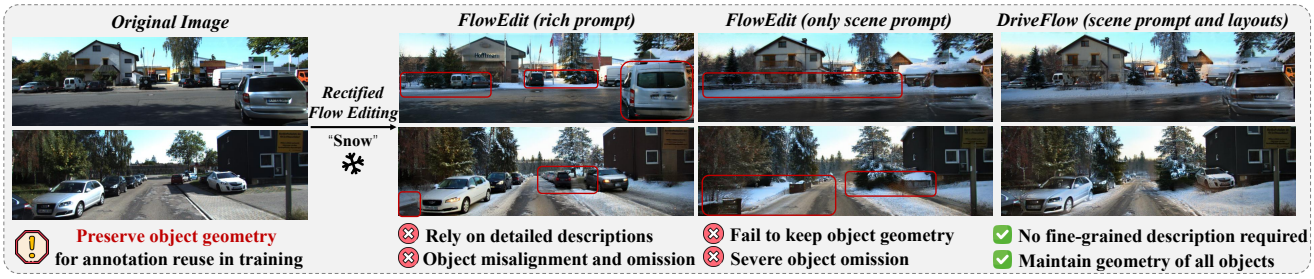

Figure 3: Due to the lack of foreground constraints, FlowEdit (Kulikov et al. 2024) often fails to maintain 3D objects even with text descriptions from Qwen2.5-VL (Bai et al. 2025), while DriveFlow only requires the target scene conditions and image layouts (*i.e.,* 2D bounding boxes). Note that foreground preservation enables annotation reuse for augmented training.

scenes, addressing data scarcity in autonomous driving. Despite their success, these methods introduce a considerable computational burden since they require substantial training data to train auxiliary modules or models.

Controllable T2I Diffusion. Pre-trained models such as Stable Diffusion (Rombach et al. 2022) and other large-scale architectures (Labs 2024) enable high-fidelity image synthesis. This capability has improved controllable T2I diffusion to serve as a valuable paradigm for generating diverse synthetic data with fine-grained control. Recent methods such as ControlNet (Zhao et al. 2023) and Layoutdiffusion (Zheng et al. 2023) offer users spatial control based on trainable auxiliary modules. Alternatively, training-free methods like PnP (Tumanyan et al. 2023) and FreeControl (Mo et al. 2024) manipulate self-attention features for semantic and spatial control. Besides, FlowEdit (Kulikov et al. 2024) achieves the same goal in an inversion-free manner by constructing an ODE that directly maps source and target distributions. However, even if fine-grained text descriptions are available (c.f. Figure 3), general-purpose editing methods still pose potential risks of object misalignment and omissions. To solve it, DriveGEN (Lin et al. 2025a) extracts self-prototypes to guide the diffusion process for object preservation in autonomous driving. Unfortunately, previous studies (Kulikov et al. 2024; Wang et al. 2024) have shown that inversion-based editing methods often suffer from unsatisfactory results and computational inefficiency.

Preliminary

Rectified Flow models. Flow-based generative models aim to construct a transportation between two distributions X_0 and X_1 through an ordinary differential equation (ODE):

$$dZ_t = V(Z_t, t)\, dt, \tag{1}$$

where time $t \in [0, 1]$ and V is a time-dependent velocity field which is typically parameterized by a learnable neural network. The learned velocity field V satisfies the boundary condition that if the vector $Z_1 \sim X_1$ at $t = 1$, then $Z_0 \sim X_0$ at $t = 0$. Generally, we choose $X_1 = \mathcal{N}(0, I)$ which allows to easily draw samples from the distribution X_0. To generate target samples, we get the initial Gaussian noise at $t = 1$ and solve the ODE backward to $t = 0$.

Rectified Flow (Liu, Gong, and Liu 2022) is a particular paradigm of flow models, which learns a straight path to transport the Gaussian Noise distribution X_1 to the real data distribution X_0. Thus, the marginal distribution X_t at time t corresponds to a linear interpolation between X_0 and X_1:

$$X_t \sim (1 - t)X_0 + tX_1. \tag{2}$$

With the text prompt C, T2I flow models adapt their velocity field V to $V(X_t, t, C)$. Then, such models are trained on the image-text paired data (X_0, C), which allows models to generate images via conditional sampling from $X_0|C$.

FlowEdit. Inversion-based editing involves two stages: 1) Invert the source image to noise space via the forward trajectory Z_t^{src}, then 2) generate the target image from the noise latent via the reverse trajectory Z_t^{tar}. FlowEdit (Kulikov et al. 2024) shows this process can be reformulated as a direct path $Z_t^{inv} = Z_0^{src} + Z_t^{tar} - Z_t^{src}$. This equation can be further expressed as an ODE:

$$dZ_t^{inv} = V_t^{\triangle}(Z_t^{src}, Z_t^{inv} + Z_t^{src} - Z_0^{src})dt. \tag{3}$$

Since a fixed Z_t^{src} often create mismatched pairings, FlowEdit solves it by averaging across multiple random pairings: $\hat{Z}_t^{\text{src}} = (1-t)Z_0^{\text{src}} + tN_t$ where $N_t \sim \mathcal{N}(0, 1)$. Therefore, substituting back into the Eqn. (3), we obtain:

$$dZ_t^{\text{FE}} = \mathbb{E}\left[V_t^{\Delta}(\hat{Z}_t^{\text{src}}, Z_t^{\text{FE}} + \hat{Z}_t^{\text{src}} - Z_0^{\text{src}}) \,\middle|\, Z_0^{\text{src}}\right] dt. \tag{4}$$

This path achieves *noise-free* editing since the velocity difference vector $V_t^{\triangle}(\hat{Z}_t^{src}, \hat{Z}_t^{tar})$ cancels out the same noise (c.f. the light green arrow in Figure 4). In this way, noise-free trajectories enhance editing stability by preventing stochastic disturbances during the generation process.

Rectified Flow Adaptation

Problem Statement. Based on the labeled training images $\mathcal{D}=\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, we can obtain the well-trained 3D visual detector $f_{\Theta_d}(\cdot)$ where Θ_d represents the learnable parameters. A total of N training images are drawn from the training distribution $P(\mathbf{x})$ (*i.e.,* , $\mathbf{x} \sim P(\mathbf{x})$). During deployment, the model accesses unlabeled test images $\mathcal{D}_t=\{\mathbf{x}_i^t\}_{i=1}^{M}$ from distribution $Q(\mathbf{x})$ (*i.e.,* $\mathbf{x}^t \sim Q(\mathbf{x})$), which often differs from the training distribution $P(\mathbf{x})$ due to diverse environmental conditions and weather variations, *i.e.,* $P(\mathbf{x}) \neq Q(\mathbf{x})$. Once data distribution shifts exist,

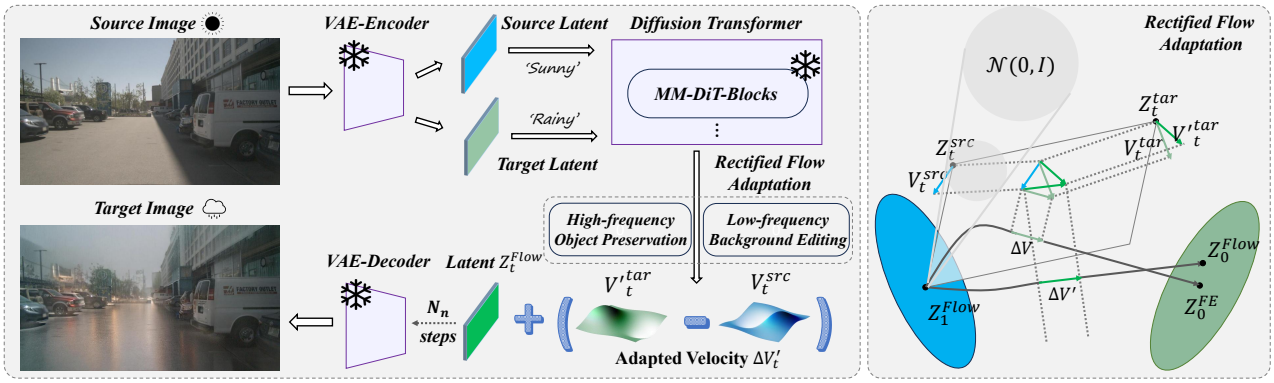

Figure 4: An illustration of DriveFlow. Without modification of the pre-trained model, DriveFlow employs frequency-based decomposition for both velocity fields V_t^{src} and V_t^{tar}, and then applies: 1) High-Frequency Foreground Preservation, applying a L2 alignment loss to align high-frequency contents between velocity fields explicitly. 2) Dual-Frequency Background Optimization, introducing dual-frequency optimization for background areas to ensure editing flexibility and semantic consistency.

the well-trained detector encounters the Out-of-distribution (OOD) issue, leading to unexpected performance drop.

Prior works suffer from several key challenges in addressing the OOD issue. General image editing methods (Mo et al. 2024; Kulikov et al. 2024) fail to maintain all objects with precise geometry, while test-time approaches (Lin et al. 2025b; Oh et al. 2025) require additional computation costs during inference. Prior method (Lin et al. 2025a) relies on inversion-based techniques, which may lead to suboptimal results and a significant computational burden.

Overall Scheme. We introduce DriveFlow, a rectified flow adaptation method for training data enhancement in vision-centric 3D Object Detection, which builds upon pre-trained T2I flow models as illustrated in Figure 4. The editing process is driven by a set of timesteps $\{t_i\}_{i=0}^{T}$ where T represents the total number of intervals. The objective of DriveFlow is to learn a suitable target velocity V'^{tar}_t through N_n inner iterations at each of the N_{max} diffusion steps, subject to the constraint $N_{max} \leq T$.

Without loss of generality, given a source image X_0^{src}, we first encode it by the encoder of the Variational AutoEncoder (VAE) to obtain the initial latent Z_0^{src}. Then, we prepare two latent-prompt pairs for the diffusion transformer. For the source pair, the source latent at time t_i is equal to:

$$\hat{Z}^{src}_{t_i} = (1 - t_i)Z_0^{src} + t_i N_t. \tag{5}$$

The source text prompt c_{src} is generated by simply describing the scene of X_0^{src} (*e.g.,* 'An urban scene on a sunny day'). As for the target pair, the target latent is obtained by:

$$\hat{Z}^{tar}_{t_i} = Z^{Flow}_{t_i} + \hat{Z}^{src}_{t_i} - Z_0^{src}, \tag{6}$$

where $Z^{Flow}_{t_{max}}$ is initialized by Z_0^{src} and $t_{max} = \max_i\{t_i\}$. Similarly, the target prompt c_{tar} contains the description for the desired scene (*e.g.,* rainy). Given the source and target velocity fields V_t^{src} and V_t^{tar}, DriveFlow performs frequency-based decomposition, parameterizing V_t^{tar} as a learnable vector. To learn an appropriate V'^{tar}_t, the key challenge lies in achieving the desired scene-level editing without compromising the integrity of the 3D object geometry. To this end, DriveFlow first applies the foreground preservation loss $\mathcal{L}_{obj}$ between the high-frequency components of the foreground to maintain 3D object geometry. Meanwhile, DriveFlow derives a spatial cosine-similarity map between the low-frequency components for background regions, utilizing it to compute the diversity loss $\mathcal{L}_{div}$ for sufficient editing intensity. To prevent unexpected collapse on background regions, DriveFlow also enables the high-frequency background regularization term $\mathcal{L}_{bg}$ (see Algorithm 1).

Overall, the total scheme of DriveFlow is as follows:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{obj} + \lambda_2\mathcal{L}_{div} + \lambda_3\mathcal{L}_{bg}, \tag{7}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters. Subsequently, we obtain the updated target velocity V'^{tar}_t which guides the velocity difference $\Delta V'_t$ via:

$$V_t^{\triangle} \leftarrow V'^{tar}_t(\hat{Z}^{tar}_{t_i}, t_i) - V^{src}(\hat{Z}^{src}_{t_i}, t_i). \tag{8}$$

Eventually, we update the edited latent $Z^{Flow}_{t_{i-1}}$ via:

$$Z^{Flow}_{t_{i-1}} \leftarrow Z^{Flow}_{t_i} + (t_{i-1} - t_i)V^{\triangle}_{t_i}. \tag{9}$$

High-Frequency Foreground Preservation

General-purpose editing methods (Kulikov et al. 2024; Mo et al. 2024) often fail to maintain 3D object geometry (see figure 3) even when guided by detailed text descriptions from Qwen2.5-VL (Bai et al. 2025). To handle this, we decompose the velocity fields V_t^{src} and V_t^{tar} at timestep t by applying the Gaussian blur $G_\sigma^{(k)}$ to achieve the low-frequency components $V_{L,t}$:

$$V_{L,t} = V * G_\sigma^{(k)}, \tag{10}$$

$$G_\sigma^{(k)}(i,j) = \frac{1}{K}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), \tag{11}$$

where $i, j \in \left\{-\frac{k-1}{2}, ..., \frac{k-1}{2}\right\}$ depends on the kernel size k, K is a constant and σ controls the blur strength. Therefore, we can achieve the high-frequency component $V_{H,t}$ by:

$$V_{H,t} = V - V_{L,t}. \tag{12}$$

Since $G_\sigma^{(k)}$ acts as a low-pass filter to preserve slowly varying components $V_{L,t}$, the high-frequency residual $V_{H,t}$ in Eqn. 12 captures rapidly varying components that typically correspond to objects within 2D bounding boxes. With parameterization of V^{tar} as a learnable vector, we calculate the foreground preservation loss $\mathcal{L}_{obj}$ between $V_{H,t}^{src}$ and $V_{H,t}^{tar}$ within all object regions:

$$\mathcal{L}_{obj} = \frac{1}{|\mathbf{M}|}\|\mathbf{M} \odot (V_{H,t}^{tar} - V_{H,t}^{src})\|_2^2, \tag{13}$$

where $\mathbf{M}$ is the binary mask derived from the coordinate transformation of image layouts $\mathbf{L}$ through downsampling, with object regions marked as 1 and background as 0.

Dual-Frequency Background Optimization

To fully exploit the pre-trained T2I flow model, we aim for sufficient editing intensity in background regions. To this end, we first compute the diversity loss $\mathcal{L}_{div}$ between the low-frequency components $V_{L,t}^{src}$ and $V_{L,t}^{tar}$ within background regions by:

$$\mathcal{L}_{div} = \frac{1}{|\bar{\mathbf{M}}|}\sum_{\bar{\mathbf{M}}} \cos(V_{L,t}^{tar}, V_{L,t}^{src}), \tag{14}$$

where $\bar{\mathbf{M}} = 1 - \mathbf{M}$ and $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} \in [-1, 1]$ denotes the cosine-similarity calculation. Specifically, the objective of the diversity loss $\mathcal{L}_{div}$ is to maximize the discrepancy between the source and target low-frequency components of background regions, which encourages the optimized velocity field $V_t'^{tar}$ to exhibit sufficient variations. By emphasizing regions with higher similarity, $\mathcal{L}_{div}$ guides this process to pay more attention to the regions which are more similar to the original ones. Such a design encourages more comprehensive background editing.

However, exclusive reliance on $\mathcal{L}_{div}$ for the velocity filed adaptation may result in trivial solutions within the background, *i.e.,* the optimized velocity field indiscriminately seeks to maximize the differences from the source velocity field $V_{L,t}^{src}$. To prevent the unexpected collapse, we further enforce semantic consistency constraints by applying the background regularization term:

$$\mathcal{L}_{bg} = \frac{1}{|\bar{\mathbf{M}}|}\|\bar{\mathbf{M}} \odot (V_{H,t}^{tar} - V_{H,t}^{src})\|_2^2. \tag{15}$$

With the introduction of $\mathcal{L}_{bg}$, the background editing process achieves a trade-off between diversity and semantic consistency. This dual-frequency background optimization mechanism ensures the simultaneous achievement of background diversity and semantic consistency, thereby effectively mitigating potential semantic drift. Prior approach (Lin et al. 2025a) often emphasizes explicit foreground constraints while overlooking the need for semantic consistency in background regions during the editing process. However, it is essential for temporal multi-view 3D object detection (Huang and Huang 2022) to apply reasonable constraints to background regions since it controls whether the augmented training data retains adequate temporal consistency (c.f. Temporal-Based Section in Experiments). We summarize the Pseudo-code of DriveFlow in Algorithm 1.

Algorithm 1: The pipeline of the proposed DriveFlow

Require: Training data $\{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^N$; Hyper-parameters $\lambda_1, \lambda_2, \lambda_3, N_n, N_{max}$; Target scene; Pre-trained model.

for each training image $\mathbf{x}_i$ **do**
  for diffusion step $i = N_{max} \to 1$ **do**
    Get c_{src}, c_{tar} based on the source and target scene;
    Extract the source latent Z_0^{src} and initialize $Z_{t_i}^{Flow}$;
    Get $\hat{Z}_{t_i}^{\text{src}}$ and $\hat{Z}_{t_i}^{tar}$ based on Eqn. 5 and Eqn. 6;
    Undergo the transformer to get V^{src} and V^{tar};
    Decomposition based on Eqn. 10 and Eqn. 12;
    for inner loop $n = 1 \to N_n$ **do**
      Calculate $\mathcal{L}_{obj}, \mathcal{L}_{div}, \mathcal{L}_{bg}$ by Eqn. 13, 14 and 15;
      Update $V_t'^{tar}$ based on Eqn. 7;
    end for
    Update $V_t^{\triangle}$ based on Eqn. 8;
    Update $Z_{t_{i-1}}^{Flow}$ based on Eqn. 9;
  end for
end for
return Output images for all $\mathbf{x}_i$ of the target scene.

Experiments

We validate the effectiveness for DriveFlow on both monocular and multi-view 3D object detection. Following DriveGEN (Lin et al. 2025a), we set three different training settings with various enhanced scenarios: 1) Traditional techniques (*i.e.,* Color Jitter and Brightness); 2) Scenes with Snow augmentation; 3) Scenes with Snow, Rain, Fog, Night, Defocus and Sandstorm augmentation (*6 × Aug.*). More implementation details are put in Appendix ***B***.

Datasets. In monocular 3D object detection, we follow the existing protocol (Zhang, Lu, and Zhou 2021) to split the images of KITTI (Geiger, Lenz, and Urtasun 2012) into a training set (3712 images) and a validation set (3769 images), including three classes: Car, Pedestrian, and Cyclist. To validate the model robustness, well-trained detectors are evaluated on KITTI-C (Lin et al. 2025b), including 13 corrupted scenarios for validation across four categories: Noise, Blur, Weather, and Digital (Hendrycks and Dietterich 2018).

For multi-view 3D object detection, we conduct experiments on the nuScenes (Caesar et al. 2020) dataset. Following DriveGEN (Lin et al. 2025a), we augment 500 daytime training scenes under the *snow* condition to enhance multi-view 3D detectors. Then, they are evaluated on the widely used Robo3D benchmark (Xie et al. 2025). Moreover, we also validate DriveFlow for enhancing temporal-based methods on real-world scenarios following (Liu et al. 2023). More dataset details are provided in Appendix ***C***.

Compared Methods. All the experiments are based on well-known or state-of-the-art baselines (Zhang, Lu, and Zhou 2021; Qin and Li 2022; Yan et al. 2024; Li et al. 2022; Huang and Huang 2022). We compare DriveFlow with: 1) Well-trained model, *i.e.,* fully trained on original data and apply the model to corrupted test data; 2) Traditional data augmentation techniques, *i.e.,* Color Jitter and Brightness; 3) Training-based T2I diffusion: ControlNet (Zhang, Rao, and Agrawala 2023) with additional masks (Ravi et al.

Car, IoU @ 0.7, 0.5, 0.5														
Method	Noise			Blur			Weather				Digital			Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Snow	Frost	Fog	Brit.	Contr.	Pixel	Sat.	
MonoGround	13.05	21.77	18.87	20.79	30.74	32.02	34.43	27.02	14.15	46.21	14.63	33.41	35.60	26.36
• Color Jitter (*Trad.*)	12.88	24.31	18.95	23.07	30.44	31.42	35.94	30.43	19.89	44.66	20.61	29.75	36.65	26.36
• Brightness (*Trad.*)	14.02	23.52	20.14	23.95	31.78	28.79	35.08	31.87	18.87	42.94	17.75	25.55	37.18	27.03
• ControlNet (*Snow*)	1.76	3.23	4.63	5.20	12.95	14.11	17.70	11.58	3.04	35.21	2.98	7.29	13.98	10.28
• ControlNet (*6 × Aug.*)	0.00	0.00	0.00	1.68	1.26	0.35	1.13	0.52	0.44	4.08	0.38	2.22	1.77	1.06
• FreeControl (*Snow*)	11.75	21.89	15.76	17.70	21.45	21.69	32.08	20.60	13.57	36.05	14.03	26.75	38.35	22.43
• FreeControl (*6 × Aug.*)	15.20	22.59	15.35	22.00	21.18	18.95	17.69	14.85	14.82	24.02	16.97	22.99	26.12	19.44
• DriveGEN (*Snow*)	17.07	26.78	23.78	32.89	37.52	39.06	40.61	34.91	25.29	46.21	27.12	38.25	44.45	33.38
• DriveGEN (*6 × Aug.*)	23.84	32.59	30.34	38.57	41.20	40.19	38.16	38.40	32.53	43.95	34.80	44.10	45.13	37.21
• FlowEdit (*Snow*)	4.38	8.54	6.98	24.57	30.98	27.19	27.84	28.36	24.32	38.31	28.84	28.00	31.98	23.87
• DriveFlow (*Snow*)	26.73	35.70	26.59	38.22	41.73	42.16	**43.43**	40.73	41.20	47.16	43.72	44.15	45.18	39.75
• DriveFlow (*6 × Aug.*)	**29.64**	**39.45**	**30.56**	**43.95**	**45.02**	**45.49**	42.63	**42.51**	**44.18**	**47.73**	**45.61**	**46.59**	**46.22**	**42.27**
MonoCD	8.88	15.60	13.22	23.44	32.83	33.93	30.18	27.94	22.52	46.07	23.20	29.87	37.31	26.54
• Color Jitter (*Trad.*)	8.61	14.28	12.79	21.13	32.22	33.81	32.14	30.63	24.03	45.09	25.68	30.57	38.78	26.90
• Brightness (*Trad.*)	11.76	19.38	16.09	21.60	31.01	32.36	32.32	29.87	22.56	45.69	24.56	34.70	39.18	27.78
• ControlNet (*Snow*)	0.00	0.00	0.00	1.00	1.59	4.35	5.06	5.99	2.67	18.24	3.28	0.64	1.57	3.41
• ControlNet (*6 × Aug.*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
• FreeControl (*Snow*)	11.30	20.10	13.00	16.10	23.70	24.20	27.70	22.60	19.60	32.20	20.90	30.00	34.50	22.80
• FreeControl (*6 × Aug.*)	12.90	20.00	13.00	13.60	16.70	14.60	15.70	13.50	15.60	21.30	15.60	21.60	23.10	16.70
• DriveGEN (*Snow*)	19.91	28.93	24.87	35.06	38.61	38.81	37.00	37.32	37.26	43.74	38.37	41.86	43.56	35.79
• DriveGEN (*6 × Aug.*)	23.35	34.49	30.36	40.47	41.15	42.67	40.08	39.61	41.51	46.15	42.99	44.35	45.57	39.44
• FlowEdit (*Snow*)	8.43	14.94	7.14	28.44	33.26	32.53	30.95	30.63	35.66	40.40	36.82	32.82	39.26	28.56
• DriveFlow (*Snow*)	27.84	39.42	30.02	38.92	42.18	43.82	**41.64**	**42.43**	43.36	**47.53**	44.16	45.22	**46.38**	40.99
• DriveFlow (*6 × Aug.*)	**29.26**	**40.64**	**32.04**	**44.55**	**44.90**	**44.81**	39.76	39.53	**45.50**	47.30	**46.00**	**45.83**	45.84	**42.00**

Table 1: Comparison on KITTI-C, severity level 1 regarding Mean $AP_{3D|R_{40}}$. The **bold** number indicates the best result.

2024) and prompts (Chen et al. 2024); 4) Training-free T2I diffusion (inversion-based): FreeControl (Mo et al. 2024) and DriveGEN (Lin et al. 2025a) enables zero-shot control of pretrained diffusion models. 5) Rectified-flow editing (inversion-free): FlowEdit (Kulikov et al. 2024) enables powerful generation based on pre-trained T2I flow models.

Evaluation Protocols. For monocular 3D object detection, we primarily report experimental results using Average Precision (AP) for 3D bounding boxes, denoted as $AP_{3D|R_{40}}$. On the KITTI-C dataset, results on the KITTI-C dataset are averaged across three difficulty levels, with Intersection over Union (IoU) thresholds set to 0.7, 0.5, 0.5 for Cars and 0.5, 0.25, 0.25 for Pedestrians and Cyclists, respectively. As for multi-view 3D object detection, we report the mean average precision (mAP) and nuScenes detection score (NDS).

Comparisons with Previous Methods

In monocular 3D object detection, the results of Figure 1 and Table 1 reveal that: 1) Well-trained detectors exhibit substantial performance degradation when deployed under corrupted scenarios, and conventional augmentation techniques fail to mitigate the data distribution shifts. 2) Due to the absence of foreground constraints, ControlNet (Zhang, Rao, and Agrawala 2023), FreeControl (Mo et al. 2024) and FlowEdit (Kulikov et al. 2024) yield marginal improvements for the single snow augmentation. As more augmented scenes are incorporated, they show progressively declining performance. 3) DriveFlow consistently outperforms DriveGEN (Lin et al. 2025a) within all baselines across 13 OOD scenarios. Remarkably, with only a single snow augmentation, our method outperforms DriveGEN with six augmented scenes across both majority and minority classes. In addition, we provide the object-region SSIM between generated and original images of KITTI in Appendix ***D***.

For multi-view 3D object detection, we follow the DriveGEN protocol by selecting 3,000 daytime training images and applying the snow augmentation (3k Snow) for enhancement. Table 2 shows that DriveFlow enhances BEVFormer-tiny (Li et al. 2022) to achieve better performance, outperforming DriveGEN across all 8 OOD scenarios in nuScene-C (Xie et al. 2025). Considering the substantial computational efficiency of DriveFlow, these results further validate our effectiveness. More results such as the class-wise analysis on nuScenes are put in Appendix ***D***.

Ablation Studies and Visualizations

To examine DriveFlow, we provide qualitative results generated by various settings as shown in Figure 5. Compared

Figure 5: Ablation studies on the loss terms $\mathcal{L}_{obj}$, $\mathcal{L}_{div}$ and $\mathcal{L}_{bg}$. More results are available in Appendix ***E***.

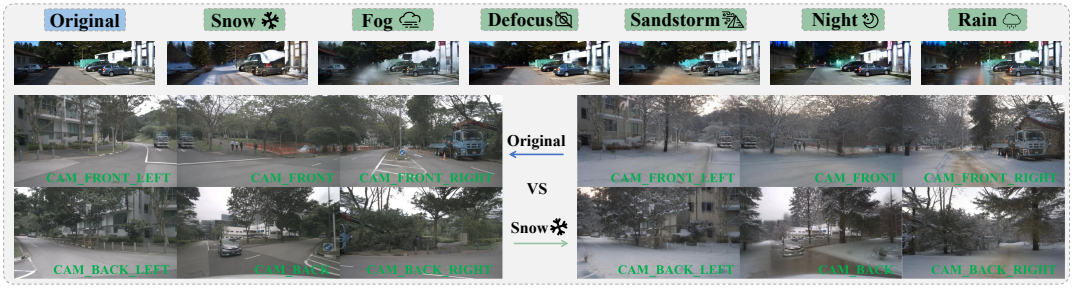

Figure 6: Qualitative visualizations of DriveFlow based on KITTI with various scenes and nuScenes with different views.

Metric	Metric	Brightness	CameraCrash	ColorQuant	Fog	FrameLost	LowLight	MotionBlur	Snow	Avg.
mAP	BEVFormer-tiny	24.26	14.89	23.91	21.98	20.43	16.11	20.78	10.83	19.15
	• DriveGEN (*3k Snow*)	26.04	15.79	25.78	24.13	21.13	17.20	22.39	11.72	20.52
	• DriveFlow (*3k Snow*)	**26.26**	**16.76**	**25.82**	**24.45**	**21.72**	**17.87**	**22.53**	**12.62**	**21.00**

Table 2: Detection results on nuScenes-C, regarding mAP. Due to page limitations, results in terms of NDS are in Appendix ***D***.

with *no velocity adaptation*, applying the foreground preservation loss $\mathcal{L}_{obj}$ preserves all annotated objects, while applying the dual-frequency background optimization terms $\mathcal{L}_{div}$ and $\mathcal{L}_{bg}$ improve editing intensity and enforce semantic consistency. Eventually, introducing all loss terms achieves the best results. Detailed results of hyper-parameter selection are put in Appendix ***E***. In addition, we provide qualitative visualizations of monocular (top) and multi-view (bottom) object detection as shown in Figure 6. More visualizations and the results of another powerful flow model, *i.e.,* FLUX (Labs 2024), are available in Appendix ***F***.

Validation on Temporal-Based Methods

An intuitive concern is whether DriveFlow can still improve temporal-based 3D object detection since DriveFlow has considered semantic consistency in background regions. More detailed results are available in Appendix ***G***.

Conclusion

In this paper, we propose a novel rectified flow adaptation method, namely DriveFlow, aiming to improve model robustness via training data enhancement in vision-centric 3D object detection. Specifically, our method performs frequency-based decomposition for the velocity fields of pre-trained T2I flow models. Then, DriveFlow devises two strategies: 1) High-frequency foreground preservation aims to maintain all 3D object geometry via a foreground preservation loss. 2) Dual-frequency background optimization introduces the diversity loss to fully exploit pre-trained T2I flow models and the background regularization term to prevent unexpected collapse in background regions. Experiments on monocular, multi-view and temporal-based multi-view 3D object detection demonstrate our effectiveness.

Acknowledgements

By NSFC with Grant No. 62573371, by the Basic Research Project No. HZQBKCZYZ2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, by Guangdong S&T Program with Grant No. 2024B0101030002, by the Shenzhen General Program No. JCYJ20220530143600001, by the Shenzhen Outstanding Talents Training Fund 202002, by the Guangdong Research Project No. 2017ZT07X152 and No. 2019CX01X104, by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), by the Guangdong Provincial Key Laboratory of BigData Computing CHUK-Shenzhen, by the NSFC 61931024&12326610&62293482, by the Key Area R&D Program of Guangdong Province (Grant No. 2018B030338001), by the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. SYSPG-20241211173853027), by China Association for Science and Technology Youth Care Program, by the Shenzhen-Hong Kong Joint Funding No. SGDX20211123112401002, and by Tencent & Huawei Open Fund.

References

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Caesar, H.; Bankiti, V.; Lang, A. H.; and et.al. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; and Jia, J. 2023. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21674–21683.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24185–24198.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Hendrycks, D.; and Dietterich, T. 2018. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hong, Y.; Xu, W.; Yang, Y.; Zhou, H.; and Wang, X. 2021. MagicDrive: 3D Controllable Driving Video Synthesis Using Layout and 3D Representations. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Hu, A.; Russell, L.; Yeo, H.; Murez, Z.; Fedoseev, G.; Kendall, A.; Shotton, J.; and Corrado, G. 2023. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*.

Huang, J.; and Huang, G. 2022. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*.

Kulikov, V.; Kleiner, M.; Huberman-Spiegelglas, I.; and Michaeli, T. 2024. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*.

Labs, B. F. 2024. FLUX. https://github.com/black-forest-labs/flux.

Li, Y.; Yang, Y.; and Lei, Z. 2025. Rctrans: Radar-camera transformer via radar densifier and sequential decoder for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 5048–5056.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, 1–18. Springer.

Lin, H.; Guo, Z.; Zhang, Y.; Niu, S.; Li, Y.; Zhang, R.; Cui, S.; and Li, Z. 2025a. Drivegen: Generalized and robust 3d detection in driving via controllable text-to-image diffusion generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27497–27507.

Lin, H.; Zhang, Y.; Niu, S.; Cui, S.; and Li, Z. 2025b. MonoTTA: Fully Test-Time Adaptation for Monocular 3D Object Detection. In *European Conference on Computer Vision*, 96–114. Springer.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, 2774–2781. IEEE.

Mo, S.; Mu, F.; Lin, K. H.; Liu, Y.; Guan, B.; Li, Y.; and Zhou, B. 2024. Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7465–7475.

Oh, Y.; Kim, H.-I.; Kim, S. T.; and Kim, J. U. 2025. MonoWAD: Weather-Adaptive Diffusion Model for Robust Monocular 3D Object Detection. In *European Conference on Computer Vision*.

Pu, F.; Wang, Y.; Deng, J.; and Yang, W. 2025. Monodgp: Monocular 3D object detection with decoupled-query and geometry-error priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 6520–6530.

Qin, Z.; and Li, X. 2022. Monoground: Detecting monocular 3d objects from the ground. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3793–3802.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Russell, L.; Hu, A.; Bertoni, L.; Fedoseev, G.; Shotton, J.; Arani, E.; and Corrado, G. 2025. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Sun, L.; Cao, L.; Zheng, Y.; Wang, B.; Li, Z.; Xie, W.; and Sun, X. 2022. Panacea: A framework for diverse, controllable generation of urban scenes. *arXiv preprint arXiv:2207.10701*.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Tumanyan, N.; Geyer, M.; Bagon, S.; and Dekel, T. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1921–1930.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wang, J.; Pu, J.; Qi, Z.; Guo, J.; Ma, Y.; Huang, N.; Chen, Y.; Li, X.; and Shan, Y. 2024. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*.

Wang, S.; Liu, Y.; Wang, T.; Li, Y.; and Zhang, X. 2023. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3621–3631.

Xie, S.; Kong, L.; Zhang, W.; Ren, J.; Pan, L.; Chen, K.; and Liu, Z. 2025. Benchmarking and Improving Bird's Eye View Perception Robustness in Autonomous Driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu, J.; Peng, L.; Cheng, H.; Li, H.; Qian, W.; Li, K.; Wang, W.; and Cai, D. 2023. Mononerd: Nerf-like representations for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6814–6824.

Yan, L.; Yan, P.; Xiong, S.; Xiang, X.; and Tan, Y. 2024. MonoCD: Monocular 3D Object Detection with Complementary Depths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10248–10257.

Zhang, J.; Zhang, Y.; Qi, Y.; Fu, Z.; Liu, Q.; and Wang, Y. 2025. Geobev: Learning geometric bev representation for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 9960–9968.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

Zhang, Y.; Lu, J.; and Zhou, J. 2021. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3289–3298.

Zhao, S.; Chen, D.; Chen, Y.-C.; Bao, J.; Zhang, D.; Yuan, L.; Zhang, H.; Li, D.; Chen, B.; and Zhang, L. 2023. Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models. *arXiv preprint arXiv:2305.16322*.

Zheng, G.; Zhou, X.; Li, X.; Qi, Z.; Shan, Y.; and Li, X. 2023. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22490–22499.