DreamForge: Motion-Aware Autoregressive Video Generation for Multi-View Driving Scenes

Jianbiao Mei[1,2], Yukai Ma[1,2], Xuemeng Yang[2], Licheng Wen[2], Tiantian Wei[2,3], Min Dou[2], Botian Shi[2,✉], Yong Liu[1,✉]

[1] Zhejiang University [2] Shanghai Artificial Intelligence Laboratory [3] Technical University of Munich

Abstract

Recent advances in diffusion models have significantly enhanced the cotrollable generation of streetscapes for and facilitated downstream perception and planning tasks. However, challenges such as maintaining temporal coherence, generating long videos, and accurately modeling driving scenes persist. Accordingly, we propose DreamForge, an advanced diffusion-based autoregressive video generation model designed for the long-term generation of 3D-controllable and extensible video. In terms of controllability, our DreamForge supports flexible conditions such as text descriptions, camera poses, 3D bounding boxes, and road layouts, while also providing perspective guidance to produce driving scenes that are both geometrically and contextually accurate. For consistency, we ensure inter-view consistency through cross-view attention and temporal coherence via an autoregressive architecture enhanced with motion cues. Codes will be available at https://github.com/PJLab-ADG/DriveArena.

1. Introduction

With the emergence of large-scale datasets [1–3] and growing demands for practical applications, autonomous driving (AD) algorithms have experienced remarkable advancements in recent decades. These advancements have driven a shift from traditional modular pipelines [4–6] to end-to-end models [7, 8], as well as the incorporation of knowledge-driven approaches [9–11]. Despite achieving impressive performance on various benchmarks, significant challenges such as generalization and handling corner cases remain, largely due to the limited data diversity in these datasets.

To enhance the diversity of driving scenes and facilitate downstream perception and planning tasks, recent approaches [12–14] have leveraged generative technologies such as NeRF [15], 3D GS [16], and diffusion models [17] to create multi-view driving scenes. Among these, diffusion-based methods [14, 18, 19] have gained particular attention due to their ability to produce high-fidelity, diverse scenarios. However, these methods still encounter challenges, such as maintaining temporal coherence across frames, generating long videos, and modeling geometrically and contextually accurate driving scenes, which can affect their effectiveness in real-world applications.

To alleviate the above issues, following [14, 19], we design a diffusion-based framework, named DreamForge for multi-view driving scene video generation. Specifically, our DreamForge leverages flexible control conditions, such as road layouts and 3D bounding boxes, along with textual inputs, to generate driving scenarios that are both geometrically and contextually accurate, maintaining cross-view and temporal consistency. By integrating perspective guidance and motion-aware autoregressive generation into conditional diffusion models [17, 20], our framework achieves significant improvements in several key aspects: (1) Better controllability. We can not only control the generation of scenes with varying weather conditions and styles through texts, layouts, and boxes, but also improve lane generation and foreground control by explicitly projecting road layouts and boxes into the camera view for perspective guidance. (2) Better scalability. By using road layouts, our framework can easily adapt to generating driving scenes for any city in the world by leveraging layouts from OpenStreetMap. (3) Better coherence. By injecting motion cues and generating long videos sequentially, our DreamForge ensures flexible video lengths while maintaining coherence and consistency, especially in extended sequences.

2. Methodology

2.1. Overview

We illustrate our proposed DreamForge in Figure 1. Built upon the stable diffusion pipeline [20], DreamForge incorporates an effective condition encoding module that handles various inputs, including road layouts, 3D bounding boxes, text descriptions, and camera parameters, to generate realistic surround-view images. To enhance lane generation and

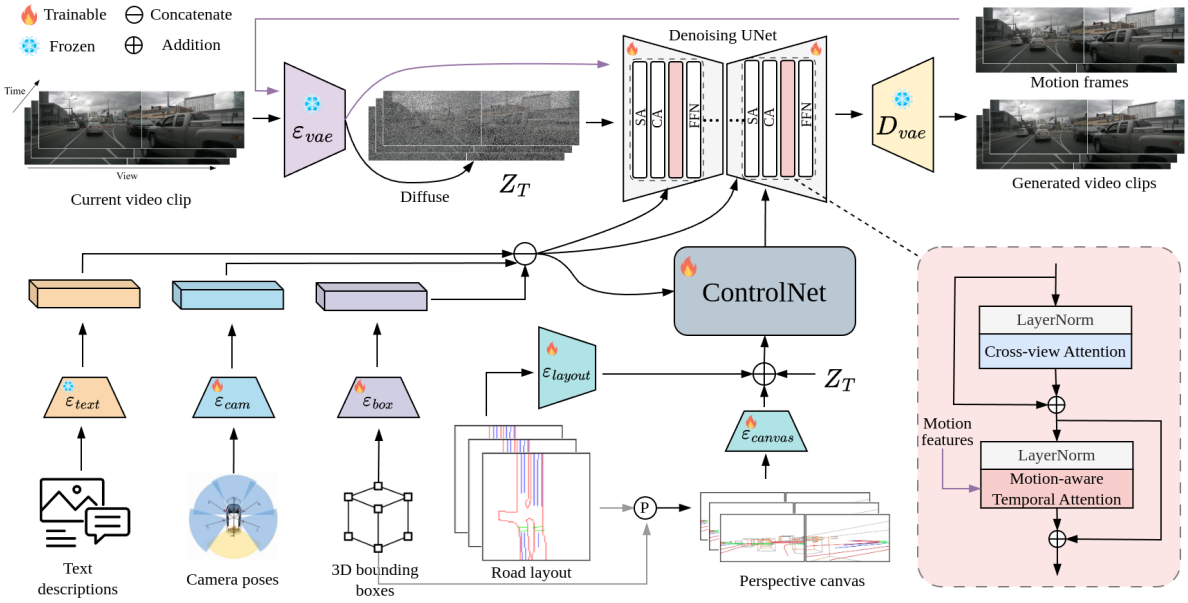

Figure 1. The overall framework of our DreamForge. During the denoising process, DreamForge leverages various conditions, including text descriptions, camera poses, 3D bounding boxes, road layouts, and perspective canvases, to enhance the modeling of driving scenes. Additionally, we incorporate cross-view and motion-aware attention mechanisms to achieve both view and temporal consistency, supporting long-term video generation through the autoregression mechanism. "P" denotes the perspective projection.

foreground control, we explicitly project road layouts and boxes into the camera view for perspective guidance. Recognizing the importance of maintaining scene consistency across different views, we integrate a cross-view attention module inspired by [14] to ensure coherence across multiple perspectives. Additionally, we have designed a motion-aware temporal attention module and an autoregressive generation paradigm that facilitate seamless video generation with flexible lengths while preserving coherence and consistency, particularly in extended sequences, thereby providing robust support for autonomous driving simulations.

2.2. Condition Encoding with ControlNet

Similar to MagicDrive [14], we perform scene-level encoding, 3D bounding box encoding, and road layout encoding for various conditions before feeding into ControlNet. Specifically, for scene-level encoding, we first enrich the text descriptions using GPT-4 then utilize the CLIP text encoder [21] ($\mathcal{E}_{text}$) to extract the text embeddings e_{text} from these descriptions. The camera poses $\mathbf{P} = \{\mathbf{K} \in \mathbb{R}^{3\times3}, \mathbf{R} \in \mathbb{R}^{3\times3}, \mathbf{T} \in \mathbb{R}^{3\times1}\}$ of each camera are encoded to e_{cam} by Fourier Embedding [15] and MLP ($\mathcal{E}_{cam}$), where $\mathbf{K}$, $\mathbf{R}$, $\mathbf{T}$ represent camera intrinsic, rotations and translations respectively. For 3D bounding box encoding, label embeddings are first extracted from the class labels using a text encoder. Coordinate embeddings are derived from the eight vertices of the 3D bounding box through Fourier Embedding and MLP. Finally, both label and coordinate embeddings are combined and compressed into the final box embeddings e_{box} using MLP. These embeddings, e_{text}, e_{cam}, and e_{box}, all have the same dimensions and are concatenated before being fed into the ControlNet and the denoising UNet, as illustrated in Figure 1. As for the road layout encoding, the 2D grid-formatted road layouts are processed through a ConvNet ($\mathcal{E}_{layout}$) to produce layout embeddings e_{layout}, which are then combined with the noised latents and fed into the ControlNet.

Perspective guidance. As mentioned above, ControlNet encodes rich 3D information and camera poses, which could theoretically allow it to perform view transformation implicitly, however, our experiments found that this implicit learning struggles to generate surround-view images that accurately align with the road layout and 3D bounding boxes, particularly in distant and complex areas, as illustrated in Figure 2. Therefore, we further project the road layout and 3D bounding boxes into the camera view using the camera poses to explicitly provide perspective guidance, which decreases the difficulty of the network in learning to generate geometrically and contextually accurate driving scenes. Specifically, the contents of each category in the road layouts and in the 3D bounding boxes are projected onto the image plane of each camera to obtain the road canvas and box canvas, respectively. These canvases are concatenated to create the perspective canvas, which is encoded with the ConvNet ($\mathcal{E}_{canvas}$) to form the canvas embeddings e_{convas}. The canvas embeddings are merged with

the noised latents, and then input into ControlNet, enhancing the accuracy of lane and foreground generation.

2.3. Motion-Aware Autoregressive Generation

Most recent works [14] focus on fixed-length video clip generation, often struggling with extended video generation due to GPU memory limitations and insufficient temporal consistency between different clips. Some methods [18, 19] propose using keyframes (e.g., the first frame) as control conditions and employing sliding windows to enhance temporal coherence during inference. However, due to the lack of motion cues and inadequate temporal modeling between video clips, the coherence of extended videos remains unsatisfactory. In this report, we introduce motion frames and design a motion-aware temporal attention module to incorporate ego-motion cues and enhance temporal consistency between adjacent clips. By sampling historical frames as motion frames, our DreamForge can easily achieve autoregressive video generation.

Motion-aware temporal attention. Let $\mathbf{I}_{i-M}^{-1}$ represent the M motion frames sampled from the previous video clip. As shown in Figure 1, these motion frames are processed by the VAE encoder to extract motion latents, which are then fed into the UNet using shared parameters with the denoising UNet to generate multi-resolution motion features. During the denoising process, these motion features are concatenated with the corresponding noised latents to compute temporal attention. Additionally, we encode the relative poses between adjacent frames into the motion embedding to further incorporate motion cues. Specifically, given the motion features $\mathbf{F}_M = \{\mathbf{F}_{-M}, ..., \mathbf{F}_{-1}\} \in \mathbb{R}^{HW \times M \times C}$ and the noised latents $\mathbf{Z}_T = \{\mathbf{Z}_0, ..., \mathbf{Z}_{T-1}\} \in \mathbb{R}^{HW \times T \times C}$ before being fed into the temporal attention layer, where M, T, H, W, and C denote the motion length, video length, spatial height, width, and number of channels, respectively, the motion-aware temporal attention can be formulated as follows:

$$\mathbf{Z}_{MT} = [\phi(\mathbf{F}_M), \mathbf{Z}_T] \tag{1}$$

$$\overline{\mathbf{Z}}_{MT} = \mathbf{Z}_{MT} + \text{ZeroConv}(\text{SelfAttn}(\mathbf{Z}_{MT} + \delta(\mathbf{P}_{rel}))) \tag{2}$$

$$\overline{\mathbf{Z}}_T = \overline{\mathbf{Z}}_{MT}[M:] \tag{3}$$

where ϕ is a linear adapter, δ denotes the MLP used for motion encoding, and $\mathbf{P}_{rel}$ represents the relative poses between adjacent frames. Note that the relative pose is set to the identity matrix for the initial motion frame.

Autoregressive video generation. To support online inference and streaming video generation while maintaining temporal coherence, we employ an autoregressive generation pipeline. During inference, we randomly sample previously generated images as motion frames and calculate the corresponding relative ego poses to provide motion cues. This method allows the diffusion model to generate the current video clip with enhanced consistency, ensuring smoother transitions and better coherence with the previously generated frames. By utilizing motion frames, our method eliminates the need for a sliding window and avoids redundant generation. However, our experiments showed that overlapping frames within the sliding window can improve generation stability. Therefore, we also provide an optional post-processing strategy to further enhance temporal coherence between adjacent video clips. Specifically, at the t-step of the denoising process for the current video clip, we replace the noised latents $\mathbf{Z}_T^t[:N]$ with the noised latents $\sqrt{\overline{\alpha}_t} \cdot \widehat{\mathbf{Z}}_T^0[-N:] + \sqrt{1-\overline{\alpha}_t} \cdot \epsilon_t$ from the previous video clip before inputting them into the denoising UNet, where, $\widehat{\mathbf{Z}}_T^0$ denotes the latents extracted using the VAE encoder, and ϵ_t represents the Gaussian noise at the t-step. By this means, we force the first N frames of the current video to be consistent with the last N frames of the previous clip as possible for better coherence.

3. Experiments

3.1. Implementation Details

Our DreamForge is built on the pretrained Stable Diffusion V1.5 [20]. The input resolution of the six camera views is set to 224×400. The video length T and length M of motion frames are set to 7 and 2.

Training details. We train the newly added modules on eight A100 GPUs using the AdamW optimizer [22] with a learning rate of 8e-5. The training process consists of two stages. In the first stage, we train the single-frame version without the motion-aware temporal attention module for 100,000 iterations with a total batch size of 24. The training objective and hype-parameters are consistent with [14]. In the second stage, we focus solely on training the temporal module for another 100,000 iterations, using a total batch size of 8. The motion frames are randomly sampled from the previous 5 frames with GT values.

Inference stage. Following the approach outlined in MagicDrive [14], we utilize the UniPC [23] scheduler for 20 steps, applying a CFG of 2.0 to generate the multi-view videos. The motion frames are randomly sampled from previously generated video clips. When generating extended videos, for the first video clip, we use the single-frame model to generate the initial frame as the motion frames. By default, the length of the overlapping frames in the post-processing strategy described in Section 2.3 is set to 2. Please note that we do not train a new model for different video lengths; all videos of varying lengths are generated using our 7-frame model.

3.2. Dataset and Metrics

Dataset. We utilize the nuScenes dataset [1] to train our controllable multi-view street view video generation model

Source of test data	FID ↓	mAP ↑	NDS ↑	Divider	Pred crossing	Boundary	mIoU ↑
Ori nuScenes	-	41.86	51.32	48.81	33.92	49.55	44.09
MagicDrive [14] (Baseline)	19.05	15.15	29.37	24.48	7.79	22.92	18.40
+ Perspective guidance	**16.03**	**16.57**	**29.50**	**33.03**	**20.99**	**36.62**	**30.21**

Table 1. Comparison of generation fidelity on generate images. The data synthesis conditions are from the nuScenes validation set. All results are computed by using the official implementation and checkpoints of BEVFormer. **Bold** represents the best results.

Source of test data	FVD ↓	mAP ↑	mIoU ↑
MagicDrive-t [14] (Baseline)	**218.12**	11.86	18.34
DreamForge (ours)	224.76	**13.80**	**29.05**

Table 2. Comparison of generation fidelity on generated 16-frame video clips. The data synthesis conditions are from the nuScenes validation set. All results are computed by using the official implementation and checkpoints of BEVFormer.

DreamForge. The nuScenes dataset provides 6 camera views at 12 Hz, offering a 360-degree perspective of the scenes. It includes 750 scenes for training and 150 scenes for validation, encompassing different cities and a variety of lighting and weather conditions, such as daytime, nighttime, sunny, cloudy, and rainy scenarios. Since the nuScenes dataset only provides annotations at 2 Hz, we employ ASAP [24] to generate interpolated annotations at 12 Hz. Additionally, we annotated each scene using GPT-4, providing detailed descriptions that include elements like time, weather, street style, road structure, and appearance. These descriptions serve as conditions for text input.

Metircs. We use FID [25] and FVD [26] to assess the quality of the generated images and videos. Additionally, we evaluate the sim-to-real gap using BEVFormer [27] to measure performance on the generated images and videos in downstream tasks, including 3D object detection (mAP and NDS) and BEV segmentation (mIoU).

3.3. Quantitative Comparison

We use MagicDrive [14] as our baseline to evaluate performance and demonstrate the effectiveness of our proposed modules. As shown in Table 1, projecting road layouts and 3D bounding boxes onto the camera view for perspective guidance enhances performance across all metrics, including FID, accuracy in 3D object detection, and map segmentation. We have observed that perspective guidance significantly improves the quality of map segmentation (a 64.2% improvement), demonstrating its effectiveness in generating geometrically and contextually accurate driving scenes.

We provide the quantitative comparison of video generation in Table 2. The score evaluation aligns with W-CODA Track 2[1], assessing the quality of the generated video with a length of 16. Compared with the temporal MagicDrive-t trained with 16-frame clips, our DreamForge achieves comparable FVD and surpasses it by a large margin in terms of object mAP and map mIoU. Note that we generate the required clips through motion-aware autoregressive generation using our 7-frame model only. Additionally, there is no need to retrain the model to generate longer videos, making it more applicable and resource-friendly.

[1] https://coda-dataset.github.io/w-coda2024/track2/

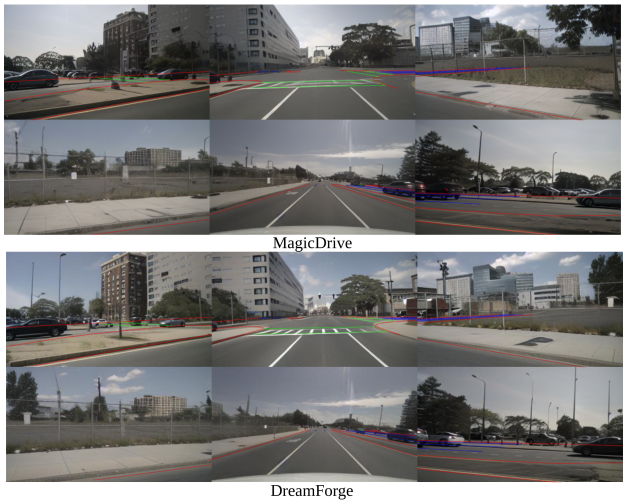

Figure 2. Visual comparison between MagicDrive and our DreamForge. We can see that our method generates more geometrically and contextually accurate surrounding view images.

4. Conclusion

This report presents DreamForge, which integrates perspective guidance and motion-aware autoregressive generation into conditional diffusion models to enhance the data diversity. We improve lane generation and foreground control by explicitly projecting road layouts and boxes onto the camera view for perspective guidance. Also, the proposed motion-aware autoregressive generation leverages motion cues and sequential generation, ensuring flexible video lengths while maintaining coherence.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving,"

in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020. 1, 3

[2] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.

[3] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020. 1

[4] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021. 1

[5] Z. Guo, X. Gao, J. Zhou, X. Cai, and B. Shi, "Scenedm: Scene-level multi-agent trajectory generation with consistent diffusion models," *arXiv preprint arXiv:2311.15736*, 2023.

[6] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao, *et al.*, "Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17524–17534, 2023. 1

[7] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023. 1

[8] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023. 1

[9] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "Dilu: A knowledge-driven approach to autonomous driving with large language models," *arXiv preprint arXiv:2309.16292*, 2023. 1

[10] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 910–919, 2024.

[11] J. Mei, Y. Ma, X. Yang, L. Wen, X. Cai, X. Li, D. Fu, B. Zhang, P. Cai, M. Dou, *et al.*, "Continuously learning, adapting, and improving: A dual-process approach to autonomous driving," *arXiv preprint arXiv:2405.15324*, 2024. 1

[12] G. Yan, J. Pi, J. Guo, Z. Luo, M. Dou, N. Deng, Q. Huang, D. Fu, L. Wen, P. Cai, *et al.*, "Oasim: an open and adaptive simulator based on neural rendering for autonomous driving," *arXiv preprint arXiv:2402.03830*, 2024. 1

[13] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng, "Street gaussians for modeling dynamic urban scenes," *arXiv preprint arXiv:2401.01339*, 2024.

[14] R. Gao, K. Chen, E. Xie, L. Hong, Z. Li, D.-Y. Yeung, and Q. Xu, "Magicdrive: Street view generation with diverse 3d geometry control," *arXiv preprint arXiv:2310.02601*, 2023. 1, 2, 3, 4

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021. 1, 2

[16] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering.," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023. 1

[17] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021. 1

[18] Y. Wen, Y. Zhao, Y. Liu, F. Jia, Y. Wang, C. Luo, C. Zhang, T. Wang, X. Sun, and X. Zhang, "Panacea: Panoramic and controllable video generation for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6902–6912, 2024. 1, 3

[19] X. Li, Y. Zhang, and X. Ye, "Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model," *arXiv preprint arXiv:2310.07771*, 2023. 1, 3

[20] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023. 1, 3

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021. 2

[22] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 3

[23] W. Zhao, L. Bai, Y. Rao, J. Zhou, and J. Lu, "Unipc: A unified predictor-corrector framework for fast sampling of diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 3

[24] X. Wang, Z. Zhu, Y. Zhang, G. Huang, Y. Ye, W. Xu, Z. Chen, and X. Wang, "Are we ready for vision-centric driving streaming perception? the asap benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9600–9610, 2023. 4

[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017. 4

[26] T. Unterthiner, S. Van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018. 4

[27] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*, pp. 1–18, Springer, 2022. 4