References

[1] Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025. 3

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. 1, 2

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 1, 2, 3

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 1

[5] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, Angela P Schoellig, and Timothy D Barfoot. Boreas: A multi-season autonomous driving dataset. *The International Journal of Robotics Research*, 42(1-2):33–42, 2023. 1, 2, 3, 6

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 1, 2, 6

[7] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021. 6

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6

[9] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W. Cohen. Subject-driven text-to-image generation via apprenticeship learning, 2023. 3

[10] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pages 24185–24198, 2024. 3, 4, 1

[11] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 3

[12] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. VQGAN-CLIP: open domain image generation and editing with natural language guidance. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXVII*, pages 88–105. Springer, 2022. 3

[13] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025. 3, 7, 8, 9

[14] Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023. 1, 3, 7, 10

[15] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024. 3

[16] Zhi Gao, Yuntao Du, Xintong Zhang, Xiaojian Ma, Wenjuan Han, Song-Chun Zhu, and Qing Li. Clova: A closed-loop visual assistant with tool usage and update. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13258–13268, 2024. 3

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *CoRR*, abs/2208.01626, 2022. 2

[18] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[19] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 2

[20] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *CoRR*, abs/2210.09276, 2022. 3

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3, 1

[22] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation, 2024. 7

[23] Nupur Kumari, Sheng-Yu Wang, Nanxuan Zhao, Yotam Nitzan, Yuheng Li, Krishna Kumar Singh, Richard Zhang, Eli Shechtman, Jun-Yan Zhu, and Xun Huang. Learning an image editing model without image editing pairs, 2025. 2

[24] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, Shuchang Zhou, Li Zhang, Xiaojuan Qi, Hao Zhao,

Mu Yang, Wenjun Zeng, and Xin Jin. Uniscene: Unified occupancy-centric driving scene generation, 2025. 3

[25] Dongxu Li, Junnan Li, and Steven C. H. Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing, 2023. 3

[26] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 7

[27] Yiyuan Liang, Zhiying Yan, Liqun Chen, Jiahuan Zhou, Luxin Yan, Sheng Zhong, and Xu Zou. Driveeditor: A unified 3d information-guided framework for controllable object editing in driving scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5164–5172, 2025. 3

[28] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025. 8

[29] Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, pages 89–106. Springer, 2020. 3

[30] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 8, 7

[31] Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. *arXiv preprint arXiv:2412.03934*, 2024. 3, 10

[32] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. *arXiv preprint arXiv:2307.11410*, 2023. 3

[33] Marcel Aguirre Mehlhorn, Andreas Richter, and Yuri AW Shardt. Ruling the operational boundaries: A survey on operational design domains of autonomous driving systems. *IFAC-PapersOnLine*, 56(2):2202–2213, 2023. 1

[34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 3

[35] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection, 2024. 3, 1

[36] Sicheng Mo, Ziyang Leng, Leon Liu, Weizhen Wang, Honglin He, and Bolei Zhou. Dreamland: Controllable world creation with simulator and generative models, 2025. 10

[37] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 3

[38] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, pages 16784–16804. PMLR, 2022. 3

[39] NVIDIA, :, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2025. 7

[40] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models, 2024. 3

[41] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003. 4

[42] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8821–8831. PMLR, 2021. 2

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 1, 2

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022. 1, 2

[45] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, 2023. 3

[46] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 3

[47] Jan Philipp Schneider, Pratik Singh Bisht, Ilya Chugunov, Andreas Kolb, Michael Moeller, and Felix Heide. Neural atlas graphs for dynamic scene decomposition and editing, 2025. 3

[48] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 2

[49] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023. 3

[50] Yichun Shi, Peng Wang, and Weilin Huang. Seededit: Align image re-generation to image editing. *arXiv preprint arXiv:2411.06686*, 2024. 2

[51] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[52] Shanlin Sun, Bingbing Zhuang, Ziyu Jiang, Buyu Liu, Xiaohui Xie, and Manmohan Chandraker. Lidarf: Delving into lidar for neural radiance field on street scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19563–19572, 2024. 3

[53] Alexander Swerdlow, Runsheng Xu, and Bolei Zhou. Street-view image generation from a bird's-eye view layout. *IEEE Robotics and Automation Letters*, 2024. 3

[54] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 3

[55] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation, 2022. 3

[56] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18359–18369, 2023. 2

[57] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023. 3

[58] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[59] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 1, 2, 6

[60] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. 3, 4, 7, 8, 9

[61] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025. 7, 8, 9

[62] Kairui Yang, Enhui Ma, Jibin Peng, Qing Guo, Di Lin, and Kaicheng Yu. Bevcontrol: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout. *arXiv preprint arXiv:2308.01661*, 2023. 3

[63] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 3

[64] Ze Yang, Jingkang Wang, Haowei Zhang, Sivabalan Manivasagam, Yun Chen, and Raquel Urtasun. Genassets: Generating in-the-wild 3d assets in latent space. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22392–22403, 2025. 3

[65] Manyi Yao, Bingbing Zhuang, Sparsh Garg, Amit Roy-Chowdhury, Christian Shelton, Manmohan Chandraker, and Abhishek Aich. ifinder: Structured zero-shot vision-based

llm grounding for dash-cam video reasoning. *Advances in Neural Information Processing Systems*, 2025. 3, 1, 11

[66] Zongsheng Yue, Kang Liao, and Chen Change Loy. Arbitrary-steps image super-resolution via diffusion inversion, 2025. 4

[67] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[68] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing, 2024. 6

[69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[70] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 5

[71] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023. 3

[72] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale, 2024. 1, 2, 5, 6, 7, 8, 9

[73] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251. IEEE Computer Society, 2017. 3

HorizonWeaver: Generalizable Multi-Level Semantic Editing for Driving Scenes

Supplementary Material

6. Dataset Collection Details

6.1. Real-World Data Pairing Details

Given a multi-season driving dataset with repeated routes and calibrated camera poses, we convert unpaired recordings into pose-aligned image pairs using a simple geometric matching rule.

Let I_{source} be a frame with camera pose $(\vec{x}_s, \phi_s, \theta_s, \psi_s)$, where $\vec{x}_s \in \mathbb{R}^3$ is the camera position and $(\phi_s, \theta_s, \psi_s)$ denote roll, pitch, and yaw. We first define a local temporal neighborhood $\mathcal{I}$ of candidate frames from other traversals of the same route. The paired target frame I_{target} is selected by minimizing a pose-disparity distance:

$$\begin{aligned} I_{\text{target}} &= \arg\min_{I \in \mathcal{I}} \text{dist}(I, I_{\text{source}}), \\ \text{dist}(I_a, I_b) &= \left\|\vec{x}_a - \vec{x}_b\right\|_2 + \left|\phi_a - \phi_b\right| + \left|\theta_a - \theta_b\right| \\ &\quad + \left|\psi_a - \psi_b\right|, \end{aligned} \tag{6}$$

where $(\vec{x}_a, \phi_a, \theta_a, \psi_a)$ and $(\vec{x}_b, \phi_b, \theta_b, \psi_b)$ are the poses of frames I_a and I_b, respectively. In practice, we accept matches only if $\text{dist}(I_{\text{target}}, I_{\text{source}})$ falls below a fixed threshold, ensuring that the resulting pairs share nearly identical viewpoints. This pose-based pairing procedure is dataset-agnostic and can be applied to any driving dataset with sufficiently accurate pose estimates, including but not limited to Boreas.

6.2. Image Descriptor

Our "Image Descriptor", an image-based analogue of iFinder [65], is a **comprehensive annotation system**: it integrates vision-language and depth estimation models that can generate semantic-aligned scene descriptions at two levels:

Multimodal Environments Descriptions. We first extract global information about the scene:

1. We use an image-based vision-language model (VLM) [10] for a global interpretation of extremely fine-grained attributes. We show the VLM prompt in Appendix 8.1.
2. To estimate object distances, we apply a metric depth estimation model, Metric3d [18], to the full image, producing a depth map whose values correspond to real-world distances.

Instance-Level Semantic Decomposition. After preparing the global description, we then record objects present in the scene:

1. We run a 2D object detector (Owlv2 [35]) that returns, for each detected object, a bounding box, a class label (from the set 'ambulance', 'bicycle', 'traffic light', 'traffic cone', 'person', 'car', 'motorcycle', 'bus', 'building', 'fire truck'), and a unique object ID.
2. For each object, we crop the global depth map (from the 2nd step above) to its bounding box and then refine that region with a binary mask from the Segment Anything Model (SAM [21]), ensuring we exclude background pixels. The object's distance is taken as the mean depth over this masked area.
3. We invoke the VLM [10] on each object's bounding box to extract additional attributes, such as vehicle color or traffic-light state.

We show an example annotation in Sec 8.1.

6.3. Global Editing Details

We define three categories of global scene edits that capture the dominant real-world variations in driving environments:

- **Weather:** *Sunny, Cloudy, Foggy, Rainy, Snowy*
- **Time of day:** *Dawn, Day, Dusk, Night*
- **Season:** *Spring, Summer, Autumn, Winter*

Together, these attributes span a broad range of environmental conditions and naturally occurring appearance changes. Applying such global edits to driving scenes produces diverse variations of the same location, which is crucial for evaluating the robustness and generalization capability of autonomous driving systems.

6.4. Quality-control pipeline for local pseudo-edited samples.

We perform automated quality control using a combination of a Vision-Language Model (VLM) and image-based similarity metrics to remove local low-quality pseudo-edits.

Global sanity check. Given an edited image, we first prompt the VLM to flag samples that appear clearly synthetic or contain obvious artifacts (e.g., distorted geometry, inconsistent lighting). Samples flagged as unrealistic are discarded.

Pedestrian edits. For edits involving pedestrians, we crop and enlarge each modified person and query the VLM with a category-specific prompt asking whether the subject looks realistic after modification or replacement. Crops judged unrealistic are removed from the dataset.

Vehicle edits. For vehicles, we similarly crop each edited instance and prompt the VLM to predict the vehicle's orientation (e.g., facing forward, turning left). We then check whether this predicted orientation is consistent with the intended edit and the surrounding context. Samples with mismatched or ambiguous orientation are discarded.

Object removals. For removal edits, we extract enlarged crops from both the original and edited images corresponding to the target region. The VLM is asked whether an instance of the removed class is still visible in the edited crop. If the VLM indicates successful removal, we additionally compute the Structural Similarity Index (SSIM) between the Canny edge maps of the original and edited crops. Only removals that pass both the semantic (VLM) and structural (edge-based SSIM) checks are retained, which is particularly important for densely packed objects such as vehicles.

Traffic light edits. For traffic lights, we restrict edits to color changes. A valid edit must satisfy three conditions: (i) the VLM confirms the presence of a plausible traffic light in the edited crop; (ii) the predicted signal color matches the target color specified by the edit; (iii) the SSIM between the edge maps of the original and edited crops remains above a fixed threshold, indicating preserved structure of the traffic light and its surroundings.

This quality-control procedure is dataset-agnostic and can be applied to any pseudo-edited driving dataset to systematically filter out unrealistic or structurally inconsistent edits.

7. Dataset Statistics

7.1. Compound Editing

In Tab. 7 we show the editing types in the compound editing dataset derived from the real-world dataset Boreas [5]. In Tab. 8, we detail how many edit types each example contains.

Table 7. **Compound Dataset Statistics (Edit Types).** Distribution of edit types across the compound editing dataset.

Edit Type	Count	Percentage
Road Conditions	146,159	15%
Time of Day	235,710	24%
Traffic	263,064	27%
Traffic Light	55,998	6%
Weather	269,715	28%

Table 8. **Compound Dataset Statistics (Subedit Counts).** Distribution of the number of edit types per example of the compound editing dataset.

Partition by Number of Subedits		
1 Edit Type	16,623	4.86%
2 Edit Types	95,295	27.88%
3 Edit Types	160,413	46.93%
4 Edit Types	64,385	18.84%
5 Edit Types	4,922	1.44%
Total Examples	341,796	–

7.2. Local Editing

We breakdown the local editing pseudo-pair datasets by generated editing type and targeted subject class for nuScenes (Tab. 9), Boreas (Tab. 10), Argoverse2 (Tab. 11), and nuPlan (Tab. 12).

Table 9. **nuScenes — Editing Statistics.** Editing types and editing actions for the nuScenes dataset.

Editing Type	Count	Action	Count
vehicle	3,584	modify	2,035
pedestrian	1,879	replace	1,559
trafficlight	41	delete/insert	1,910
Total	5,504		

Table 10. **Boreas — Editing Statistics.** Editing types and editing actions for the Boreas dataset.

Editing Type	Count	Action	Count
vehicle	5,729	modify	2,070
pedestrian	96	delete/insert	2,957
trafficlight	144	replace	942
Total	5,969		

Table 11. **Argoverse — Editing Statistics.** Editing types and editing actions for the Argoverse dataset.

Editing Type	Count	Action	Count
vehicle	2,558	modify	1,815
trafficlight	229	replace	1,503
pedestrian	2,031	delete/insert	1,500
Total	4,818		

Table 12. **nuPlan — Editing Statistics.** Editing types and editing actions for the nuPlan dataset.

Editing Type	Count	Action	Count
vehicle	50	delete/insert	33
trafficlight	23	replace	31
pedestrian	48	modify	57
Total	121		

7.3. Global Editing

We breakdown the global editing pseudo-pair datasets for nuScenes (Tab. 13), Argoverse2 (Tab. 14), and nuPlan (Tab. 15).

Table 13. **nuScenes Global Editing Type.** Distribution of global editing types in the nuScenes dataset.

Edit Type	Count	Percentage
Season	3,033	33%
Weather	3,116	33%
Time of Day	3,162	34%
Total	9,311	–

Table 14. **Argoverse2 Global Editing Types.** Distribution of global editing types in the Argoverse2 dataset.

Edit Type	Count	Percentage
Season	3,267	33%
Weather	3,288	33%
Time of Day	3,409	34%
Total	9,964	–

Table 15. **nuPlan Global Editing Types.** Distribution of global editing types in the nuPlan dataset.

Edit Type	Count	Percentage
Season	112	35%
Weather	100	31%
Time of Day	108	34%
Total	320	–

8. Annotation Pipeline

8.1. Example Annotation

The VLM is prompted with the instruction shown in Fig. 6. We show an annotated image (Fig. 7) and a truncated output caption from the annotation pipeline (Fig. 8).

8.2. LLM Generated Editing Instructions

We prompt chatGPT-4o mini with the following instruction (Fig. 9) to produce editing instructions for real-world images based on the captions of the target image.

9. Efficient LangMask Construction from User Instructions

To enable precise, instruction-aligned scene editing, we provide two efficient mechanisms for constructing LangMasks from user-specified edits.

Dataset-Driven LangMask Generation. First, following HorizonWeaver's dataset construction, a driving-scene image may be annotated using our "image descriptor." As shown in Fig. 8, this annotation provides, for each detected object, its bounding box, its distance from the ego vehicle, and a short appearance description. Given an annotated object (e.g. "car"), a user-specified editing action (e.g. "replace"), and optionally a target description (e.g. "green truck"), we form a simple sentence (i.e. "replace the car with a green truck") which is then encoded by CLIP. The resulting embedding is written into the mask for all pixels inside that object's bounding box. As multiple objects may overlap, masks are assembled in order of decreasing object distance, ensuring that nearer objects overwrite farther ones.

Interactive LangMask Construction. Alternatively, through our user interface a user may select a bounding box in the scene and specify an editing action, and optionally a target description. Then following our "image descriptor" a VLM would describe the selected subject appearance that we combine with the user's inputs into a simple sentence. After encoding this instruction with CLIP, we populate the LangMask within the chosen bounding box. When multiple user selections are made, the masks are applied in order of descending object distance.

9.1. Compound Dataset Development

LangMask instructions are simple sentences. Editing actions are one of {insert, delete, modify, replace}. For modifications, a target appearance is required (e.g. "change the car to green"). For replacements and insertions both a target subject and appearance are sampled (e.g. "insert a middle-aged person", "replace the car with a blue truck"). The LangMasks are systematically derived by comparing corresponding frame annotations via three rules:

- Distance-based filtering: Objects beyond 50 meters from the ego vehicle are excluded unless they occupy a significant image area.
- Truncation detection for undersized 2D bounding boxes near image boundaries.
- Occlusion handling: In complex traffic scenarios, overlapping vehicle bounding boxes are each preserved to maintain scene coherence.

For compound editing pairs, the traffic usually completely changes between images. For this reason, we prepend in the global text prompt "remove all pedestrians and all vehicles from the scene." Meanwhile the LangMask specifies the vehicles and pedestrians to insert. Traffic lights use the "modify" editing action.

9.2. Pseudo-Dataset Development

LangMask instructions are simple sentences. Editing actions are one of {insert, delete, modify, replace}. For modifications, a target appearance is required (e.g. "change the car to green"). For replacements and insertions both a target subject and appearance are sampled (e.g. "insert a middle-aged person", "replace the car with a blue truck").

For pseudo-pair development, we perform one edit per LangMask. Given an unpaired image, we annotate it with our "image descriptor" and randomly select one subject, and sample an editing action from {insert, delete, modify, replace}. If required, we randomly sample a target object

```
You are an expert in autonomous driving, specializing in analyzing traffic scenes. You
    ↪ receive a series of traffic images from the perspective of the ego car. Your task is
    ↪ to describe the driving environment, focusing on weather, lighting, road layout, and
    ↪ environment.

It is essential that you strictly follow the rules and instructions below. Any deviation
    ↪ from the specified structure or format will result in an invalid output.

STRICTLY follow Rules:
 - You must strictly follow the dictionary structure provided above.
 - Only use the specified terms for weather, light, road layout, and environment. Do not
     ↪ create your own terms.
 - No additional information or categories should be added.
 - You should strictly follow these instructions. If an object or element is not visible or
     ↪ does not exist in the scene, set the value to 'None'. Ensure every field is filled
     ↪ with the appropriate value or 'None'.
 - STRICTLY ignore any text written on the image.


Output the result in the following dictionary format:

{
  "surrounding_info": {
    "weather": "[e.g., 'cloudy', 'sunny', 'rainy', 'fog', 'snowy']",
    "road_layout": "[Choose from: 'straight road', 'curved road', 'intersection',
        ↪ 'T-junction', 'ramp']",
    "environment": "[Choose from: 'city street', 'country road', 'highway', 'residential
        ↪ area']",
    "sun_visibility_conditions": "[Choose from: 'clear', 'foggy', 'low visibility', 'hazy']",
    "road_condition": "[Choose from: 'wet', 'icy', 'normal', 'debris', 'potholes']",
    "surface_type": "[Choose from: 'asphalt', 'gravel', 'dirt', 'concrete']",
    "surface_color": "[Choose from: 'light grey', 'dark grey', 'black', 'brown']",
    "time_of_the_day": "[Choose from: 'morning', 'midday', 'afternoon', 'night', 'dawn',
        ↪ 'dusk'.]",
    "precipitation_intensity": "[Choose from: 'none', 'light', 'moderate', 'heavy',
        ↪ 'torrential'.]",
    "precipitation_visibility_impact": "[Choose from: 'none', 'low', 'moderate', 'high']",
    "cloud_cover": "[Choose from: 'clear', 'light', 'moderate', 'heavy'.]
    }
}
```

Figure 6. Annotation pipeline prompt used for the VLM.

and appearance from the following word banks.

Traffic Lights. The action is always `modify`. The target appearance is sampled from {`green`, `red`, `yellow`}.

Vehicles. For modifications, a target appearance is sampled. For replacements and insertions both a target subject and appearance are sampled.

- Target colors: `[red, blue, green, yellow, black, white, silver, grey]`
- Target objects: `[car, truck, bus, motorcycle with its rider, bicycle with its rider, ambulance, fire truck]`

Pedestrians. For insertions and replacements, we sample from both clothing types and age. For modifications, we change only the clothing type.

- Target clothing adjectives: `[red, blue, green, yellow, black, white, casual, formal, businesslike, vibrant, summer, winter, sporty]`
- Target clothing articles: `[shirt, jacket, coat, sneakers, boots, hat, dress, skirt, trousers, pants, clothes]`
- Target ages (for modifications): `[young, middle-aged, elderly]`

During training, the pseudo images are used as conditioning except for object removals which with 0.5 probability are used as ground-truth to have equal representation between deletions and insertions.

10. Training & Inference Details

10.1. Training

To incorporate the guidance from the LangMasks, we expand the VAE input channels to accept the concatenation of the input image and conditioning masks and train end-

Figure 7. Example annotation produced by the annotation pipeline.

to-end. The weights of any existing convolutions are maintained and new weights are initialized as zero. The RGB-image VAE is frozen. We train at a resolution of 512×512 and a learning rate of 1×10^{-5}.

We evaluate the UltraEdit model with the following settings:

- **UltraEdit**. The UltraEdit model supports a single binary mask as conditioning therefore we project our LangMask into a binary mask. The LangMask editing prompts are appended to the global text prompt.
- **UltraEdit-Text**. We train an UltraEdit model using only supervised objectives without LangMasks. To do this, following the process described in Sec. 3.2.1, we construct the editing prompts by asking chatGPT-4o to describe, in addition to global changes, all objects to remove from left to right, and all objects to add from left to right. Similar to the base model we project our LangMask into a binary mask. This model was trained only on the compound editing dataset from Sec. 3.2.1. This model is trained across 4×48 GB NVIDIA A6000 GPUs with a total batch size of 256 for 10000 steps. We set $\lambda_{\text{sft}} = 1.0$ during supervised training.
- **UltraEdit-Mask**. We train an UltraEdit model using only supervised training objectives and LangMasks. This model was trained only on the compound editing dataset from Sec. 3.2.1. We train this model across 4×80GB NVIDIA H100 GPUs with a total batch size of 4 for 25000 steps. We set $\lambda_{\text{sft}} = 1.0$ during supervised training.
- **UltraEdit-Mask-UFT**. We train an UltraEdit model using both supervised and unsupervised training objectives and LangMasks. This model was trained only on the compound editing dataset from Sec. 3.2.1. We finetune this model across 4×80GB NVIDIA H100 GPUs with a total batch size of 4 for 1000 steps starting at the *UltraEdit-Mask* pretrained checkpoint described above. We use the same loss hyperparameters as below.
- **Ours**. We train an UltraEdit model using supervised training for 25000 steps on both the compound editing dataset from Sec. 3.2.1 and pseudo-paired dataset (both local and global) from Sec. 3.2.2. We set $\lambda_{\text{sft}} = 1.0$ during supervised training. Then we finetune this model for 1000 steps using all the objectives described in Sec. 3.4. To adapt these objectives to multi-step diffusion, we apply gradient checkpointing and perform end-to-end training with our unsupervised losses. We train this model across 4×80GB NVIDIA H100 GPUs with a total batch size of 4 for 25000 steps. Our training parameters are:

$$\begin{aligned} \lambda_{\text{gan}} &= 0.5, \\ \lambda_{\text{id}} &= 0.05, \quad \lambda_{\text{id-lpips}} = 0.05, \\ \lambda_{\text{cycle}} &= 0.05, \quad \lambda_{\text{cycle-lpips}} = 0.05, \\ \lambda_{\text{sft}} &= 3.0, \quad \lambda_{\text{sft-lpips}} = 0.5, \\ \lambda_{\text{clip}} &= 3. \end{aligned} \tag{7}$$

```
{
    "surrounding_info": {
        "weather": "cloudy",
        "road_layout": "intersection",
        "environment": "city street",
        "sun_visibility_conditions": "low visibility",
        "road_condition": "normal",
        "surface_type": "asphalt",
        "surface_color": "dark grey",
        "time_of_the_day": "morning",
        "precipitation_intensity": "none",
        "precipitation_visibility_impact": "none",
        "cloud_cover": "heavy"
    },
    "object_info": [
        {
            "class": "building",
            "bbox": [
                1234,
                745,
                2060,
                1051
            ],
            "object_id": 1,
            "distance_from_ego_vehicle": "67.42 meters",
            "attributes": "Brown color, Brick texture, Apartment type"
        },
        {
            "class": "traffic light",
            "bbox": [
                779,
                963,
                790,
                986
            ],
            "object_id": 2,
            "distance_from_ego_vehicle": "78.61 meters",
            "attributes": "Red light"
        },
        {
            "class": "car",
            "bbox": [
                79,
                1025,
                147,
                1062
            ],
            "object_id": 10,
            "distance_from_ego_vehicle": "69.81 meters",
            "attributes": "White color, Not a police car."
        },
        ...
    ]
}
```


Figure 8. Truncated annotation output corresponding to Fig. 7.

Supervised training requires 16 hours to complete 25,000 steps. When all the objectives described in Sec. 3.4 are enabled, the model trains at approximately 15 seconds per iteration. Consequently, the 1,000-iteration post-training stage takes roughly 4 hours.

10.2. Inference

We perform inference on Qwen-Image-Edit, BAGEL, Omnigen2, and UltraEdit following their default hyperparameters. Generating a single image with 100 inference steps using Qwen-Image-Edit takes roughly 60 seconds on an 80GB NVIDIA H100 GPU.

```
You are an expert in autonomous driving, specializing in analyzing traffic scenes. You
    ↪ receive a text description of a traffic image from the perspective of an autonomous
    ↪ vehicle's camera.

Your task is to produce FOUR VERSIONS of the SAME PROMPT, each with DIFFERENT WORDING BUT
    ↪ IDENTICAL CONTENT, that describes the driving scene depicted in the image.

IMPORTANT:
- Each version should contain the EXACT SAME DESCRIPTION, just phrased differently.
- ALL prompts should describe EXACTLY THE SAME SCENE with no variation in what is being
    ↪ described.
- Only use adjectives and descriptors that are explicitly provided in the caption. Do NOT
    ↪ add your own subjective descriptors like "moody," "tranquil," "charming," etc. Stick
    ↪ strictly to the attributes and descriptors that appear in the input caption.

At the end of each prompt version, append the following line:
    "There may be minor additional changes in time or weather (such as lighting, clouds, or
        ↪ rain) between the images that are not fully captured by the descriptions, but
        ↪ these are expected to be subtle."

The prompt should be in the format below where each version describes the same contents but
    ↪ with different wording.

### Scene Description:

version_1: {{description_1}}
version_2: {{description_2}}
version_3: {{description_3}}
version_4: {{description_4}}

Image Caption: {caption_0}
```

Figure 9. Prompt used to generate real-world image editing instructions.

For *UltraEdit-Text* and *UltraEdit-Mask* we follow the default hyperparameters of UltraEdit. For *UltraEdit-Mask-UFT* and *Ours*, we use 8 inference steps and disable classifier-free-guidance mirroring training. Generating a single image takes less than 5 seconds on an 80GB NVIDIA H100 GPU.

11. Cycle Consistency Details

Global text prompts are either obtained by a VLM for pseudo-paired data, or by applying chatGPT-4o to our image annotations for compound data. In Section 3.4, t_t corresponds to the caption of the target image and t_s corresponds to the caption of the source image. Each image is annotated with up to four LLM-paraphrased variations of the same caption and one is randomly sampled during training. In the case of local edits in pseudo-paired data, t_t and t_s match, while M_t describes the transformation to obtain the target and M_s describes the transformation to obtain the source.

12. Additional Results

We show additional results for local editing (Fig. 10), global editing (Fig. 11) and compound editing (Fig. 12).

13. Driving Specific Editor Baselines & Long-tail Editing

As shown in Fig. 13, MagicDrive does not take RGB images as conditions, and thus cannot preserve critical driving scene components (appearance of vehicles, construction infrastructure, road signs, etc.). Fig 13 demonstrates editing on rare scenarios, e.g., crosswalks, road signs, and specialized vehicles (cement mixer).

14. Additional Safety-Critical Challenges in Autonomous Driving

The application of BEV map segmentation is important and studied in multiple work ([14, 26, 30]). Our results demonstrate that our synthesized weather improves BEV map segmentation in real-world conditions. To show that HorizonWeaver can help core AD challenges, such as temporal and geometric consistency, we use our edits to synthesize driving scene video edits using [39]. We show qualitative results in Fig 14.

Following the evaluation in [22] we quantify 1) the instruction alignment 2) the temporal consistency, and 3) the perceptual quality of the video edit. Tab. 16 shows that our method is competitive with Qwen in semantic consis-

Figure 10. **HorizonWeaver Editing. Local edits:** the masks (projected as binary images and stated in text for reference) enable modifications to traffic. We compare to Qwen [60], OmniGen2 [61], UltraEdit [72], and BAGEL [13].

tency and outperforms baselines in temporal consistency and visual quality. Our model has fewer than 8B parameters whereas Qwen has 20B.

We further evaluate geometric consistency by comparing the depth prediction of edited images from Boreas with the ground truth LiDAR. To do this we run the depth-prediction model from Depth-AnythingV3 [28] and follow its evaluation metric. HorizonWeaver yields δ_1 of 67.7, which is larger by 8.9 and 7.6 than Qwen and UltraEdit, respectively, showing that our edits are geometry-aligned. Many works

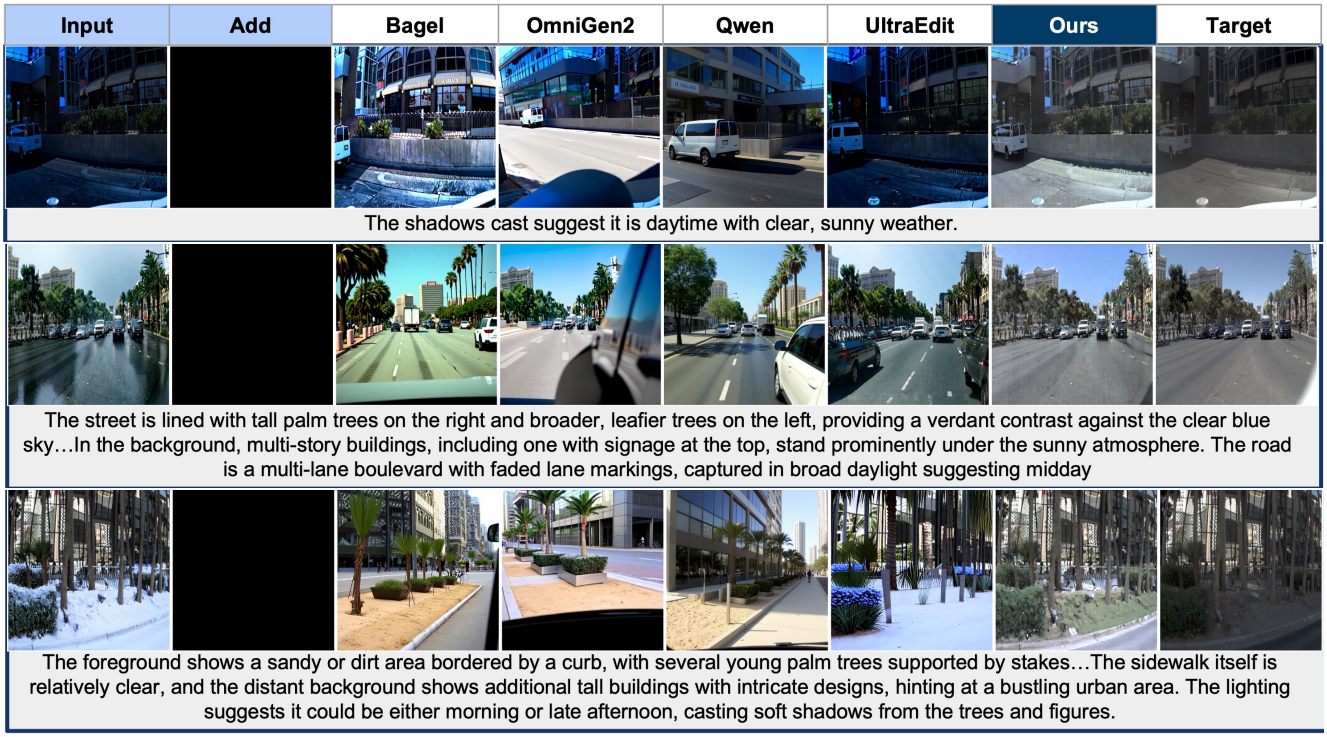

Figure 11. **HorizonWeaver Editing. Global edits:** the text prompt informs the appearance of the scene. For brevity, only the portions relevant to the shown edits are displayed. We compare to Qwen [60], OmniGen2 [61], UltraEdit [72], and BAGEL [13].

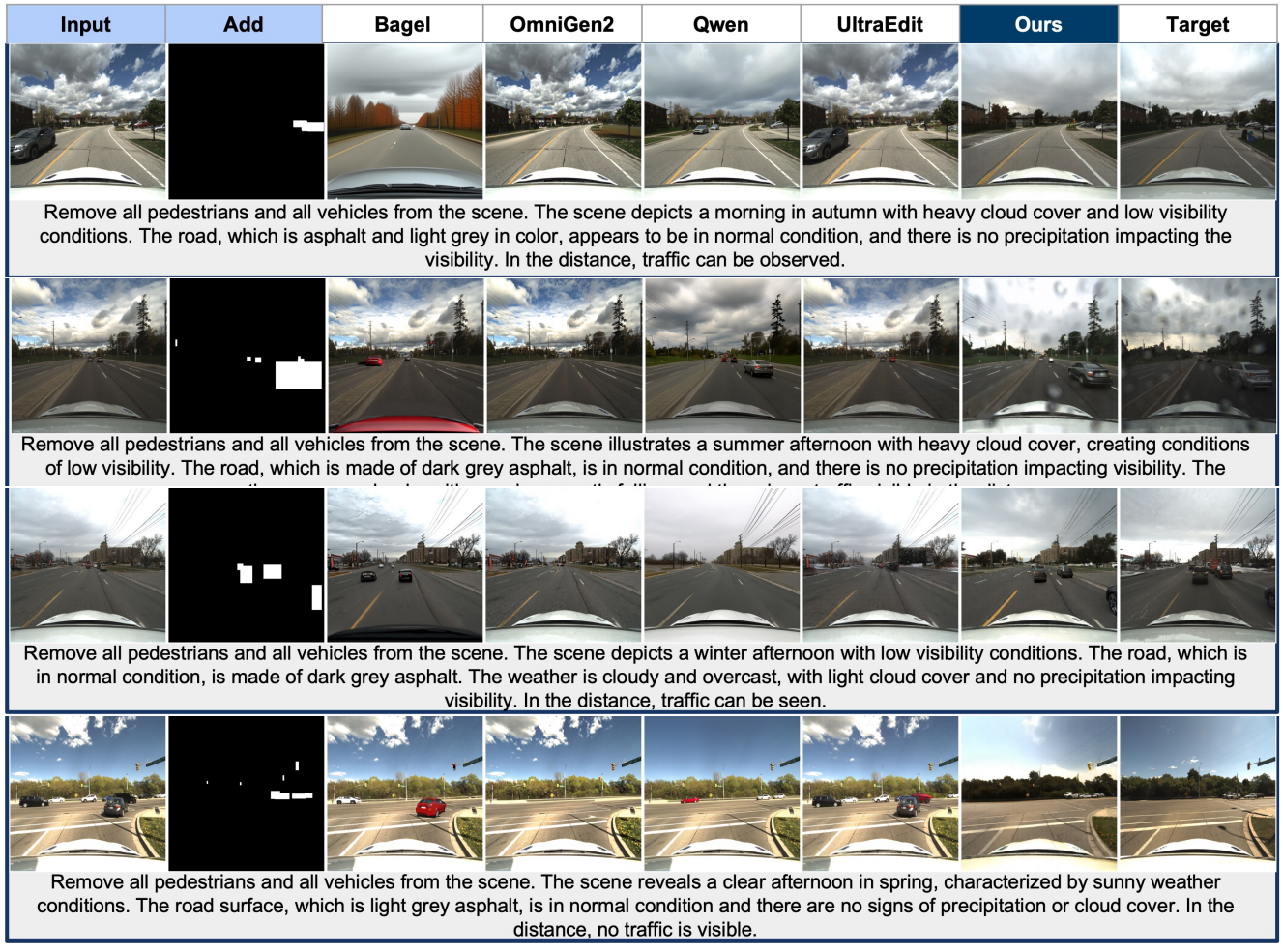

Figure 12. **HorizonWeaver Editing. Compound edits:** The masks (projected as binary images) enable modifications to traffic while the text prompt informs the desired global appearance. We compare to Qwen [60], OmniGen2 [61], UltraEdit [72], and BAGEL [13].

Figure 13. **Driving specific editor.** MagicDrive [14], cannot preserve critical driving scene components (appearance of vehicles, construction infrastructure, road signs, etc.) **Long-Trail Editing** our edits can extend to rare scenarios, e.g., crosswalks, road signs, and specialized vehicles (cement mixer) which are not preserved by MagicDrive.

Table 16. Quantitative comparison of semantic consistency, temporal consistency, and perceptual quality.

	UltraEdit	Qwen	Ours
Sem. Cons.	5.603	8.250	8.191
Temp. Cons.	6.588	6.471	6.765
Perceptual	2.603	6.191	6.265

such as [14, 31, 36] which target these AD data generation challenges are limited for editing as they do not enforce content preservation which includes safety-critical elements. Our editing can preserve scene critical elements Fig 13 such as the cement mixer and crosswalk.

15. Computational Cost & Scalability

We are comparable with previous models. We train our models in two stages. Stage 1 is supervised finetuning for 25K steps (21 GPU hours), comparable to UltraEdit. Stage 2, adds our unsupervised training objectives for 2K steps (8 GPU hours) improving performance with minimal additional cost, demonstrating the scalability of our approach. Inference is efficient: our model generates an image in 8

Figure 14. **Temporal Consistency.** Videos produced from an edited initial image by HorizonWeaver, UltraEdit, and Qwen (top to bottom). UltraEdit leaves the initial image unchanged. Prompt: *Change the season to summer.*

steps (taking 0.021s with 17GB of GPU memory), similar to UltraEdit (50 steps in 0.025s using 17GB of GPU memory), but much faster and more memory-efficient than Qwen (taking 68s using 61GB of GPU memory). Although UltraEdit has lower per-step latency, its multi-step performance is worse than ours.

16. LLM & VLM Evaluation

To analyse annotation/construction noise, we retrain our model on pre- and post-filtered data for 10K steps. Tab. 17 shows the performance of the editing increases with filtering, especially for fine-grained editing. 56% of all our edited images are retained post-filter. For more analysis, please see [65] Sec. 4, which forms part of our LLM-based pipeline.

Table 17. **Effect of VLM and algorithmic filtering.**

	Fine-Grained (Full)				Fine-Grained (Crop)			
Setting	L1↓	L2↓	CLIP↑	DINO↑	L1↓	L2↓	CLIP↑	DINO↑
Prefilter	0.0300	0.0032	0.9423	0.9263	0.0720	0.0157	0.8214	0.6646
Postfilter	0.0321	0.0040	0.9450	0.9267	0.0755	0.0176	0.8206	0.6536
	Global				**Compound**			
Prefilter	0.1010	0.0196	0.9193	0.9112	0.2030	0.0740	0.8708	0.8247
Postfilter	0.0983	0.0184	0.9174	0.9074	0.2026	0.0768	0.8683	0.8374

17. LangMasks with Localized Semantics

Unlike general purpose image editing, LangMasks allow for localized semantic information to be provided to the editor. Fig. 15 shows the complexity of describing dense driving scenes without LangMasks. HorizonWeaver can apply insertion edits to empty spaces in crowded regions as shown in Fig. 16.

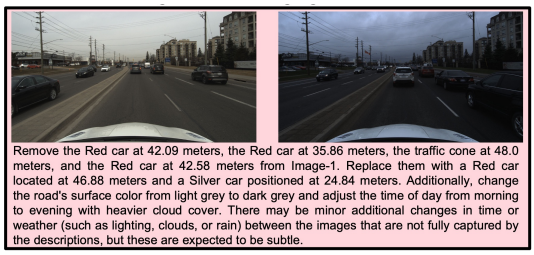

Figure 15. Driving scenes are dense; thus it is difficult to describe rich semantics for multiple object-oriented changes solely with natural language.

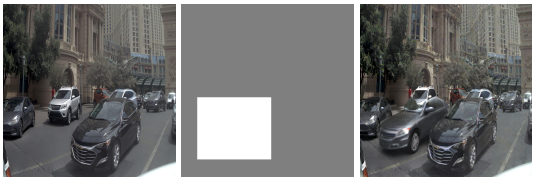

Figure 16. **Overlapping Edits**, HorizonWeaver will apply insertion edits to empty spaces in crowded regions

18. Limitations

A failure case appears in Fig. 6 (row 1, col 8), where the ground truth inaccurately labels unclear lane markings as "well-maintained." However, other major changes like time-of-day are generally correct. While our filtering partially addresses the issue of residual artifacts in pseudo-editing (Tab. 17), we plan to further improve it in future work.