Data synthesis for downstream tasks in Autonomous Driving

Valentina Muşat

Green Templeton College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hillary 2025

Data synthesis for downstream tasks in Autonomous Driving
Candidate: Valentina Muşat
Supervisor: Professor Paul Newman
Examiners: Professor Victor Adrian Prisacariu and Professor Ian Reid
Date of examination: 22nd July 2025

University of Oxford
Mobile Robotics Group
Oxford Robotics Institute
Department of Engineering Science

Acknowledgements

First and foremost, I would like to sincerely thank my supervisor Prof. Paul Newman for taking the risk and offering me a chance. I am truly thankful for the absolute freedom you gave me in choosing my research direction, the guidance, support and words of encouragement. I also appreciate you didn't shy away from telling me what I did wrong when I got stuck.

Secondly, I would like to thank Dr. Daniele De Martini and Dr. Matthew Gadd for offering me academic and moral support, especially during the frantic late-night paper submission sessions. I enjoyed our discussions, the banter and the trip to Scotland for data collection. I would especially like to thank Dr. Daniele De Martini for the emotional support and for being so respectful and understanding in the last few months leading to the thesis submission.

I want to thank my MRG colleagues Efimia Panagiotaki, Georgi Pramatarov and Benjamin Ramtoula - whose intellects and resilience I very much admire - for their friendliness and moral support.

I would like to thank my husband and best friend Dr. Horia Porav, for his love and support - ever since we've met, for being my sounding board and for the perpetual push to try again tomorrow. The dreams we dreamed on the iron bridge have always served as my compass - *I love you too!*

I would also like to thank my father Muşat Valeriu-Dan, for supporting me financially during my first year at Brookes, making it easier to pivot to engineering - I wish you peace and good health.

Finally, I would like to thank Hollie Suzanna Cater for ensuring that all things run smoothly.

Institutional

I would like to thank Google DeepMind and the University of Oxford for offering me a DeepMind Engineering Science Scholarship and Prof. Paul Newman for extending funding for the rest of my DPhil. I would also like to thank the University of Oxford Advanced Research Computing (ARC) facility for the resources provided to carry out this work.

Abstract

Autonomous vehicles have complex data requirements, and data synthesis has emerged as a solution to alleviate data scarcity. This manuscript presents a set of approaches to tackling this scarcity when training supervised downstream autonomous visions tasks with 4 concepts in mind - realism, alignment to ground-truth, control and scalability.

The first contribution presented uses a combination between generative adversarial networks and cycle generative adversarial networks to multiply existing image data by adding weather effects and changes in levels of illumination. While the method is successful at improving diversity of data, it does not tackle changes in the structure of the generated images. To overcome this, the second contribution employs scene composition, while maintaining realism and quality of associated ground-truth, using a semi-parametric approach. Since the method operates only on 2D images, it is limited in its ability to reason about complex interactions. Thus, the third publication introduces a two-stage approach that splits the problem of data generation into two distinct steps: one that reasons about scene structure and geometry in a 3D representation, and another that is responsible for the appearance of the scene. The last publication finally tackles the issue of structure and stylistic consistency by extending the previous publication with multi-view data, style maps for appearance cues and auto-regressive training.

Approaches are evaluated in terms of visual quality and alignment of structure with ground-truth, along with select experiments that test the suitability of the data as training data in object detection, semantic segmentation and depth completion tasks.

Contents

List of Figures

List of Tables

List of Abbreviations

AP Average Precision. 5, 23, 30, 54, 114

AV Autonomous Vehicle. 2, 8, 9, 33, 38, 39, 41, 49

BEV Bird's Eye View. xi, xii, 6, 39, 44, 45, 50, 82, 92–94, 96, 97, 99, 110–112, 114

CLIP Contrastive Language-Image Pretraining. 26

CNN Convolutional Neural Network. 24, 25, 40, 98

Cycle-GAN Cycle Generative Adversarial Network. xi, 8, 12, 15–17, 24, 47, 48, 113

FID Fréchet Inception Distance. 22, 25–28, 66, 70

FVD Fréchet Video Distance. 22, 27, 109, 111, 115

GAN Generative Adversarial Network. xi, 8, 12–17, 19, 23–28, 41, 42, 49, 66, 69, 113

GDPR General Data Protection Regulation. 33

IMU Inertial Measurement Unit. 33

IS Inception Score. 22, 25, 26, 66

JEDi JEPA Embedding Distance. xiii, 22, 27, 111, 115

KID Kernel Inception Distance. 22, 26–28

LiDAR Light Detection and Ranging. 2, 33, 34, 37–39, 44, 47, 48, 117

LPIPS Learned Perceptual Image Patch Similarity. 23, 24, 26

mIoU mean Intersection Over Union. 23, 28, 29, 94

MLP Multi-Layer Perceptron. xi, 19, 21, 43, 48, 92, 97, 115–117

MMD Maximum Mean Discrepancy. 26, 27, 111

MSE Mean Squared Error. 23, 24

NeRF Neural Radiance Field. x, xi, 8, 19–22, 32, 43, 44, 48, 66, 115–117

PSNR Peak Signal-to-Noise Ratio. 23, 24, 26

RADAR Radio Detection And Ranging. 33

RMSE Root Mean Squared Error. 23, 29

SSIM Structural Similarity Index Measure. 23, 24, 26

V-JEPA Video Joint Embedding Predictive Architecture. 27, 111

1

Introduction

Contents

1.1 A brief summary

In recent years, considerable progress has been made in terms of machine learning model architecture and performance, with the last few years seeing a transition towards foundational models, large language models, and multi-modal models such as large vision models. Transformer-based

architectures have become the standard choice, while diffusion models have revolutionised generative models across image, video, text, speech and motion generation tasks. Concurrently, significant effort has been dedicated to optimising, distilling or compressing models, and enabling the fine-tuning of domain-specific models with considerably less amounts of data compared to the original datasets used to initially train them (e.g. low-rank adaptation).

Regardless of the model choice, data has also evolved to be an essential component, oftentimes playing a more crucial role than model architectures or training schemes. For autonomous driving especially, data has become both a **key requirement**, since high-fidelity data streams are needed to train these models, and a **key enabler**, since easily available, diverse and flexible data sources are key to reducing time to deployment and enhancing safety. As such, data is pivotal in both generalisation – as large and diverse datasets are needed to train models to generalise across a wide range of driving scenarios, but also in specialisation – to tackle specific operational design domains and use cases.

Moreover, data continues to be of critical importance post-deployment and should be treated as a **continuous process**, as even a simple change of onboard sensors can lead to a mismatch with the data accumulated so far due to domain shift. For example, acquiring data with one type of camera / Light Detection and Ranging (LiDAR) sensor and switching to a different type can introduce a mismatch between camera noise profiles, different number of beams, different range etc. However, collecting substantial datasets every time a domain shift occurs can become difficult or cost-prohibitive.

In terms of regulatory compliance, high-quality and high-fidelity data is fundamental in demonstrating safety to regulators, even before any deployment. The ability to perform a safety analysis is dependent on the availability of extensive data that can represent or simulate realistic scenarios of traffic, road conditions, weather and illumination. Furthermore, testing edge-case scenarios, such as abnormal pedestrian behaviour, can help to ensure that Autonomous Vehicles (AVs) anticipate and react properly to critical safety conditions and rare events.

Unfortunately, most of the time, such corner cases are difficult or even impossible to obtain via standard data collection methods. Consequently, simulators and data synthesis approaches have witnessed increased use in alleviating the demand for more training, validation and testing data, while domain adaptation has been used in adapting existing data in situations where domain shifts occur.

Although acquiring data from the real world is the ideal choice in terms of domain alignment, obtaining edge cases is challenging, and careful planning and logistics are required to capture specific

weathers or levels of illumination. Moreover, the captured data typically requires annotation to fit the needs of the downstream task.

On the other hand, 2D and 3D simulators provide the ability to model edge-cases and frequently offer the capability to automatically annotate the data. However, they suffer from relatively large domain gaps, and require high-quality assets (3D models, meshes, textures). Data synthesis using generative machine learning models can address some of these limitations, by reducing the domain gap compared to pure simulators, while retaining the ability to generate edge cases and associated ground-truth.

Moreover, generative models can be integrated with both existing sources of (real) data as well as with simulation. Already-labelled real data can be transformed or adapted to new weather conditions or levels of illumination, or decomposed and recomposed into new scenes. Similarly, simulation can be used to generate tail event scenarios, either as single or sequences of observations, and the resulting output and ground-truth can be used to condition or guide generative models. Numerous other combinations can be constructed by capitalizing on the strengths of each approach.

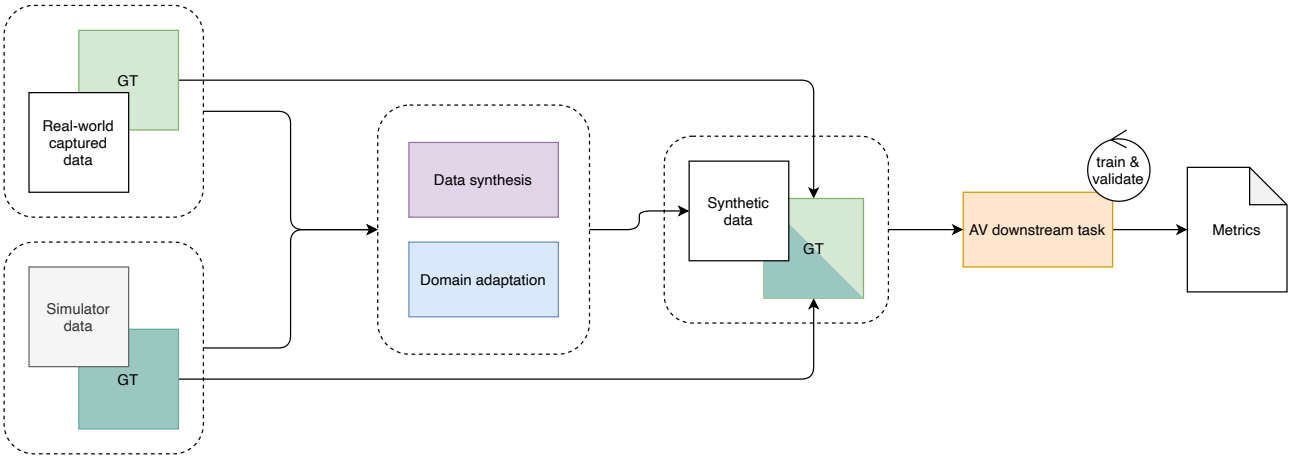


Figure 1.1: Data use: in order to train a downstream task, data from both simulators and real-capture can be used for synthesis and domain adaptation, while their ground-truth annotations can be transferred to the new synthetic set.

1.2 Motivation

In summary, data capture is often difficult, expensive, laborious and, for some edge cases, outright **hazardous** or impossible. However, since most state-of-the-art approaches to perception, planning or safety have transitioned to machine learning models, data remains essential for training, validating or certifying their performance.

While both simulation and generative models have vastly increased in popularity as a mechanism for data creation, there isn't a one-model-fits-all solution – a single model to generate all types of data distributions, modalities or scenarios – so data synthesis and adaptation thus need to be tailored for specific tasks or use cases.

For the application to autonomous driving downstream tasks, data synthesis must often simultaneously be accompanied by ground-truth, either pre-existing or co-generated, and needs to be synchronized between different modalities – for example images and point clouds need to represent the same scene, and maintain structure and appearance consistency across the temporal dimension.

Minimising the domain gap and ensuring diversity and control are all important aspects to consider. Additionally, it is crucial to train downstream tasks on data that is diverse, especially when developing model backbones and foundational models, but at the same time it is equally important to have data that is specific to the deployment domain for extensive validation before actual deployment. Waiting for the local weather to align with the deployment requirements is time-consuming – for example, deploying in December when there is a high chance of snow, but also not being able to collect snowy data prior to December. In many such other cases of unavailable environments, data synthesis is becoming a crucial tool for generating annotated data on demand.

Consequently, the following aspects are essential when considering data synthesis as a source of autonomous driving data:

- **realism**, which refers to the reduced domain gap between the generated data and the target real deployment data;
- **alignment**, which refers to how well the generated data complies with the associated ground-truth;
- **control**, which refers to the ability to compose and manipulate variables in the scene according to some input;
- **scalability**, which refers to the ability to generate data and its associated ground-truth at large scale efficiently.

1.3 Contribution

This thesis makes a number of contributions to address the aspects of autonomous driving data generation discussed above, improving the state-of-the-art across adverse weather generation, focusing on synthesising multiple modalities, and extending the applicability of synthetic data to additional downstream tasks by tackling frame-to-frame consistency.

In **Multi-weather-city** (Chapter 4), we address the challenges introduced by adverse weather conditions, especially when multiple types of weather may be present. The proposed method explores stacking multiple adverse weather effects, by combining models trained on individual weather conditions to generate composite weathers, thus allowing for more realistic and challenging testing scenes for autonomous driving perception models. We demonstrate enhanced performance and robustness in object detection and instance semantic segmentation when training with multiple synthetic weathers and testing on real weather, in some cases leading to an increase in mean Average Precision (AP) of more than 10 percentage points. We additionally provide the materials and instructions required to construct the dataset for further use.

In **Depth-SIMS** (Chapter 5), we address synthesis control in terms of compositionality and alignment of the output data with its ground-truth and realism. Domain adaptation approaches - such as Multi-weather-city - generate more data, covering more levels of illumination and weather types, without requiring additional annotation, as they do not change the structure of scenes, and without requiring additional data collection. But this does not address the lack of co-occurence of classes and types of structure needed to ensure adequate data coverage. This issue can be alleviated by making use of existing real-world data: while data on hand may not contain the required co-occurence of classes, it can be decomposed into its constituent semantic parts and re-combined to generate novel scenes with the desired elements. Additionally, this further reduces the domain gap by taking advantage of the realistic and natural appearance of elements extracted from existing data. However, changing the structure of the scene involves the generation of not only images, but also well-aligned associated depth and ground-truth annotations, for this data to be useful for training or testing downstream tasks specific to autonomous driving. The contributions of this approach include a semi-parametric model that combines the advantages of both parametric and non-parametric methods, retaining as much of the original blobs as possible, a novel scheme for retrieving candidate blobs based on Hu moments, and methods for generating both depth maps and updated semantic segmentation maps that match the synthesized images. We use the generated data to train a semantic segmentation and a depth

completion model to measure its suitability, and we demonstrate competitive perceptual quality but also an increase by 3.7 percentage points in terms of alignment to the conditioning semantic map.

In **NeuralFloors** (Chapter 6) we further address scalability and alignment, while further enhancing control through modern representations. Multi-weather-city requires RGB images at inference time, while Depth-SIMS requires a source of blobs, segmentation maps, and optionally depth maps. Scene manipulation requires 3D understanding, but this is difficult to tackle when inputs and outputs are ground-view perspective images. To address these shortcomings, NeuralFloors unlocks the scalability of data synthesis, generating ground-view images along with semantic segmentation, instance and depth maps by using 2D Bird's Eye View (BEV) representations as inputs. The advantage of using BEV inputs - such as semantic segmentation - to control the scene configuration is that they can be much more easily edited compared to ground-view perspective inputs. Additionally, they are simple, compact yet rich representations of scenes, enabling easy visualisation, inspection and a high degree of editability. The contributions include an approach to 2D-to-3D lifting of BEV maps using a neural field conditioned on BEV semantic segmentation maps, and a 2-stage approach that decouples the generation of structure from that of visual appearance. Our experiments on 3 representative urban driving datasets demonstrate both the perceptual quality of the generated images, and the high degree of spatial alignment of the synthesized ground-truth as opposed to prior art.

In **NeuralFloors++** (Chapter 7), we refocus on realism, as represented by multi-view and temporal consistency. While the previous approach can be used to generate scenes and associated ground-truth, its outputs cannot be used to train downstream tasks that require style and structural consistency between consecutive frames, as no mechanism is used to enforce temporal consistency.

To address this shortcoming, the proposed approach focuses on a number of improvements that are built on top of **NeuralFloors** . Firstly, it improves frame-to-frame consistency, by making use of multi-view training to encourage consistency of structure. Additionally, it introduces style maps that are generated by sampling style embeddings from a bank of style embeddings which are organised by class, and are consistent between BEV and ground-view outputs across camera poses and movement. In the second stage, the proposed method also employs autoregressive training which is responsible for coherent visual appearance, as the current generated frame is also conditional on a past generated frame. Additionally, by leveraging 3D bounding box annotations, the method is further improved by introducing dynamic objects in the scene. Finally, we provide an extensive comparison to the

performance of existing methods in terms of perceptual quality and alignment to ground-truth as well as an extensive ablation study for each stage.

While not in the scope of the publications presented in this manuscript, further details on how these approaches could be combined and improved in order to extend their applicability are presented in their respective chapters.

1.4 Publications

The publication timeline, as described in Fig. 1.2, consists in 4 manuscripts, each presented in their respective chapters:

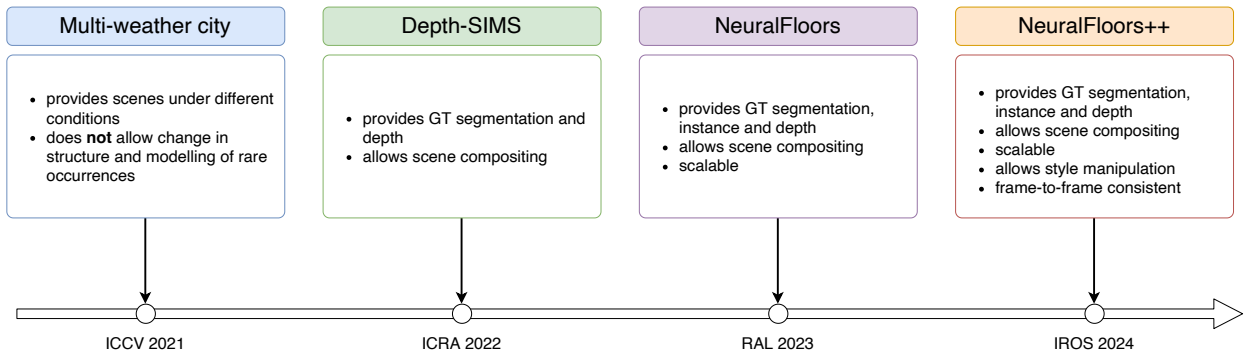


Figure 1.2: Publication timeline.

1. A method for synthesising and combining multiple adverse weather conditions is presented in *Multi-weather city: Adverse weather stacking for autonomous driving* (Musat et al., 2021) included in Chapter 4;

2. A method for generating images, depth and semantic maps is presented in *Depth-SIMS: Semi-parametric image and depth synthesis* (Musat et al., 2022) included in Chapter 5;

3. A method for scalable synthesis of images, depth, instance and segmentation maps from BEV maps is presened in *NeuralFloors: Conditional street-Level scene generation from BEV semantic maps via neural fields* (Muşat et al., 2024a) included in Chapter 6;

4. A method for frame-to-frame consistent and scalable synthesis of scenes is presented in *NeuralFloors++: Consistent street-level scene generation from BEV semantic maps* (Muşat et al., 2024b) included in Chapter 7.

1.5 Thesis outline

Chapter 2 provides a necessary brief overview of synthesis models and building blocks such as Generative Adversarial Networks (GANs), Cycle Generative Adversarial Networks (Cycle-GANs), Diffusion models, Neural Radiance Fields (NeRFs) and associated metrics. Understanding the advantages and disadvantages of such methods provides insights into the progression of the data synthesis domain as a whole, and the architectural choices made across the publications included in this manuscript. Chapter 3 presents a brief comparison between 3 data sources: existing datasets originating from standard capturing methods, simulator-derived data sources and data synthesis. Understanding the limitations of these approaches helps the reader in understanding the reasoning behind the choices made in this manuscript. Additionally, a chronological literature review of a selection of some of the most important publications within the field is provided, in order to easily place and contextualise the contribution of the manuscript. Chapters 4 to 7 describe the 4 publications and present brief further insights regarding the progress made within the field since the date of publication, along with various challenges. Chapter 8 presents a brief summary of contributions and explores possible future improvements.

1.6 Impact

The transportation sector is set to be transformed by the introduction of AVs, which need to be able to accurately interpret their surrounding environment, make context-aware decisions and navigate without the intervention of a human operator.

The UK government's investment and involvement in AV technology began in 2015 with the foundation of the Centre for Connected and Autonomous Vehicles (CCAV), whose aim is to connect different stakeholders to speed up the deployment of AVs which, according to the Connected and Automated Mobility 2025 report issued by the UK Department of Transport, are anticipated to bring improvements across multiple areas such as: increased road safety, enhanced access to transport and consequently a greener future due to reduced emissions (HM Government, 2022).

However, alongside technological challenges, the world-wide deployment of AVs has encountered issues such as public trust, ethical considerations and regulatory barriers. As a result, the government and the automotive sector have begun to collaborate on establishing guidelines and regulations that encourage a transparent assessment of the issues faced in their deployment. As such, the Automated

Vehicles Act 2024 (Department for Transport and Centre for Connected and Autonomous Vehicles, 2024) proposes safety standards for AVs across Great Britain, expecting that their performance is at least on par with that of human drivers, thus contributing to an improved overall road safety. Furthermore, they are expected to operate in specific pre-determined operational design domains, meaning the AV is allowed to operate solely in well-defined areas and under certain conditions e.g. downtown area, restricted to roads with a speed limit of 20 mph and during overcast weather conditions only. Additionally, AVs will likely require robust methods for identifying any deviations from the specified operational design domain, which will represent another distinct requirement for training and validation data.

Obstacle detection or pedestrian trajectory estimation in diverse environments requires training and validation on large datasets with enough representative cases to ensure safety. Depth estimation from monocular cameras requires training on datasets that are sufficiently diverse to help models to learn the appropriate prior knowledge. Visual localization across various times of the day or during challenging weather conditions may require custom feature detectors and descriptors that are invariant to weather effects. In most of the cases there is always a recurrent requirement - data with accurate ground-truth, which requires substantial resources to collect, curate and annotate. As this process represents a bottleneck, many tasks are trained or benchmarked on data from simulation, which by construction can provide readily available ground-truth. However, due to the sim to real domain gap, many approaches trained or tested on such data fail to perform well in real-world scenarios. As such, research efforts have been directed towards closing this domain gap by generating synthetic data or augmenting existing real data using techniques ranging from heuristics to the current use of large multi-modal models.

This thesis makes contributions in the field of domain adaptation and data synthesis for AVs, with a focus on providing associated ground-truth for training capabilities. Having access to synthetic data for training downstream tasks helps to overcome data biases in representation, and to meet safety and ethical standards, thus accelerating development cycles and reducing time to deployment. Additionally, the scope of synthetic data might evolve beyond just training data, with highly diverse validation and certification data needed to be made available to ensure both coverage and robust identification of out-of-domain conditions. For many unavailable domains, which have a low probability of being encountered, synthetic data will play a key role in enabling wide-scale deployment of AVs.

2

Preliminaries

Contents

2.1 Introduction

The manuscripts presented in Chapters 4 to 7 make use of model architectures and training techniques in combinations that depend on both their tasks but also on data and ground-truth availability. Firstly, Multi-weather-city, presented in Chapter 4, uses both GANs and Cycle-GANs to add weather effects and change levels of illumination in existing RGB data. Next, Depth-SIMS, presented in Chapter 5, makes use of RGB blobs (masked colored objects and textures extracted from RGB images) to compose new scenes. To do this, it employs both a GAN-based model to harmonize and blend RGB blobs on image canvases, but also strong supervision for depth completion. In contrast, NeuralFloors and NeuralFoors++, presented in Chapters 6 and 7, use a neural field approach in the first stage – which deals with structure and geometry, and a latent diffusion model in the second stage, which is responsible for the generation of realistic and diverse RGB images based on the outputs of the first stage.

Regardless of the architecture used, all outputs are evaluated using a set of metrics described in Section 2.6, which presents a detailed discussion on advantages and shortcomings of metrics that are commonly used in the literature.

2.2 Generative Adversarial Networks

Introduced by Goodfellow et al. (2014), GANs are a category of deep learning models that have been used to synthesize realistic data such as text, images, and videos. Their goal is to learn the distribution of the input data, such that once the model is trained, they can be used to create new data, by sampling from the learned distribution.

The training framework is a minimax game involving 2 players, where one player learns to fool the other player, while the other player learns to distinguish whether it is being fooled or not, i.e. whether the data is generated or real. The goal of each player is to minimize its own loss, while the Nash Equilibrium is the state in which no player can decrease its own loss, without the other player increasing theirs. Training a GAN thus involves training a generator and a discriminator in competition, where the generator is trained to generate data similar to the real distribution, while the discriminator is trained to distinguish between real and generated data.

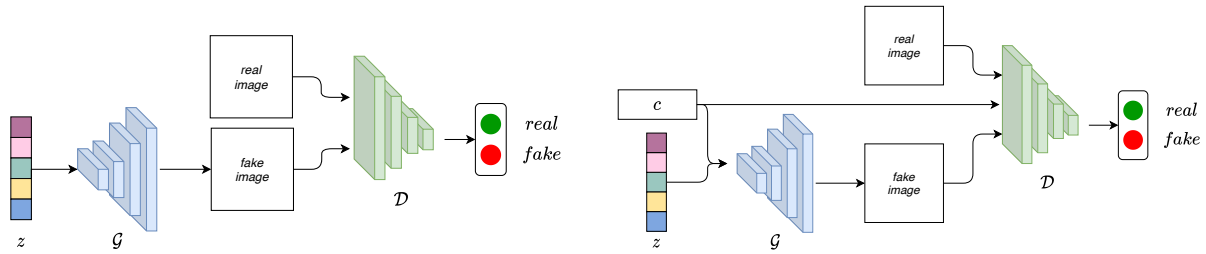
The fundamental, non-conditional configuration (Fig. 2.1a), in the context of image synthesis, consists of a generator $\mathcal{G}$ which takes as input a latent embedding z and learns to generate an image that is indistinguishable from one sampled from the real data distribution p_r.

On the other hand, the discriminator $\mathcal{D}$ takes as input either a real image x or an image generated by $\mathcal{G}$, and learns to distinguish whether the input image is sampled from the true data distribution p_r or from the distribution learned by the generator. Its output is typically in the range $[0, 1]$, indicating how confidently it classifies an input image as resembling a real image.

Following the notation from Goodfellow et al. (2014), when the generator and discriminator are trained together, they aim to optimize the following objective:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim p_r(x)}[\log \mathcal{D}(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))] \tag{2.1}$$

where $\mathcal{D}(x)$ denotes the discriminator's output when evaluating a real image x, and $\mathcal{D}(\mathcal{G}(z))$ its output when evaluating an image generated by the generator.



(a) In the unconditional GAN setting, the generator receives as input a latent embedding and generates an image. The discriminator then learns to distinguish between real and generated images.

(b) In the conditional GAN setting, both the generator and the discriminator receive additional information such as labels, text or sketches – this allows for more controlled and targeted synthesis.

Figure 2.1: GAN basic unconditional and conditional architectures.

For the discriminator to maximise its objective, both terms of equation 2.1 need to be maximized. Maximizing $\mathbb{E}_{x \sim p_r(x)}[\log \mathcal{D}(x)]$ means that the discriminator $\mathcal{D}$ needs to output $\mathcal{D}(x) = 1$ when it receives a real image as input, in order for the log operation to result in $\rightarrow 0$. However, if the discriminator outputs $\mathcal{D}(x) = 0$, the result will be a large negative value $\rightarrow -\infty$. To maximize $\mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))]$, the component $(1 - \mathcal{D}(\mathcal{G}(z)))$ should be close to 1 thus, $\mathcal{D}(\mathcal{G}(z))$ should be close to 0, which means that when given a generated image, the discriminator should output 0 i.e., the discriminator distinguishes the generated image as being fake. However, if $\mathcal{D}(\mathcal{G}(z)) = 1$ for a generated image, the log operation will result in $\rightarrow -\infty$, thus instead minimizing the second term.

For the generator to minimise its objective, the second term $\mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))]$ must be minimised. This will happen when $(1 - \mathcal{D}(\mathcal{G}(z))) = 0$, thus when $\mathcal{D}(G(z)) = 1$, which is when the discriminator is fooled (i.e. it outputs 1 for a generated image).

A GAN model can be extended (Fig. 2.1b) to be conditional on an additional input c, in order to provide guidance to the generation process, allowing more control over the synthesis process, enhancing diversity and promoting structured outputs. In this case, the generator $\mathcal{G}(z, c)$ receives both the latent embedding z and the condition c when generating an image, and the discriminator receives the condition c for both the real sample $\mathcal{D}(x, c)$ and the generated sample $\mathcal{D}(\mathcal{G}(z, c), c)$. The condition c could be for example: a one-hot encoding representing a class label, a depth map of the image to be generated, a text prompt describing the scene, a sketch etc.

Although powerful, GANs are challenging to train due to the adversarial nature of the learning process which requires an equilibrium of the two competing networks, while both needing to get better at defeating the other network. For example, if the discriminator $\mathcal{D}$ is over-parametrised, it can easily distinguish between real and generated samples, leaving the generator with no useful learning signal. On the other hand, if the generator $\mathcal{G}$ is over-parametrised, it will generate samples that will easily trick the discriminator $\mathcal{D}$. As a consequence, the feedback from the discriminator will fail to provide useful learning signals to the generator. In order to avoid such problems, various approaches have been proposed, such as: using networks that are balanced from an architectural point of view, regularization techniques (such as Spectral Normalisation in BigGAN (Brock et al., 2019)) which normalize the networks' gradients in order to prevent gradient explosion, or progressive growing of GANs (Karras et al., 2018) which involves gradually increasing the resolution of the images and/or the number of layers in the network during training. As a complement to these, learning rate scheduling and multi-scale discriminators can also lead to more stable training or higher quality results.

Another challenge that GANs face is mode collapse, which refers to the generator $\mathcal{G}$ producing output images with reduced diversity, thus failing to capture the full distribution of the data and leading to poor generalization. This occurs when the generator learns a parametrisation that minimally satisfies the discriminator and is typically due to either unbalanced training, or an architectural imbalance between the generator and the discriminator, with one overpowering the other. In order to overcome this, various approaches have been proposed such as mini-batch discrimination (Salimans et al., 2016) which compares samples within a mini-batch instead of a single sample, thus being able to detect similarity and encouraging diversity within a larger set of images. Additionally, feature matching (Salimans et al., 2016; T. Wang et al., 2018) can force the generator to match feature distributions of real data by minimizing the difference between the features extracted by the

discriminator from real images and generated images, thus encouraging the generator to learn the whole data distribution instead of a single or few modes.

The quality of GAN outputs can be improved by using a feature-based loss such as the VGG perceptual loss (J. Johnson et al., 2016), which leverages a pre-trained VGG network (Simonyan & Zisserman, 2015) to extract image features from both real and generated images, and computes the squared Euclidean distance between corresponding elements of the feature maps:

$$\mathcal{L}_{\text{VGG}}(x_g, x_r) = \sum_l \|\phi_l(x_g) - \phi_l(x_r)\|_2^2 \tag{2.2}$$

where x_g is the generated image, x_r is the real image and ϕ_l are feature maps from layer l of the VGG network.

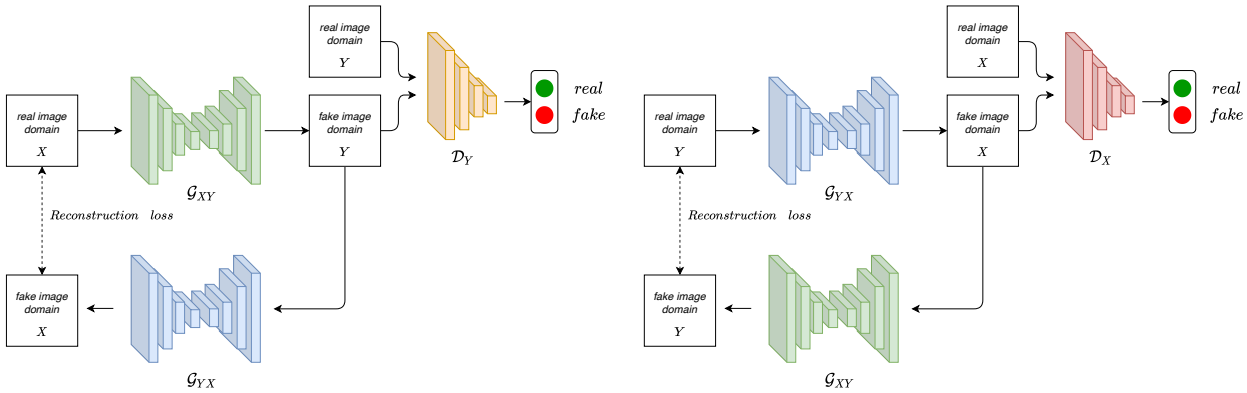
2.3 Cycle Generative Adversarial Networks

Cycle-GANs are an advanced GAN architecture designed for image-to-image translation tasks without requiring paired training examples. In their proposed architecture, Zhu et al. (2017) combine two generators and train them with two individual weakly supervised GAN adversarial losses and a strongly supervised cycle-consistency loss which ensures that an image transformed to another domain can be mapped back to its original form. Thus the first generator $\mathcal{G}_{XY}$ learns a transformation from X to Y, and the second generator $\mathcal{G}_{YX}$ learns the opposite transformation, from Y to X, such that $\mathcal{G}_{YX} \circ \mathcal{G}_{XY} = \mathbb{1}$, meaning the second generator applied on the output of the first generator should output the original, unchanged image.

For the mapping $X \to Y$, the generator $\mathcal{G}_{XY}$ translates images from domain X to domain Y, and the discriminator $\mathcal{D}_Y$ distinguishes between real images from domain Y and generated images $\mathcal{G}_{XY}(x)$. Following a notation close to Zhu et al. (2017), the adversarial loss for generator $\mathcal{G}_{XY}$ and discriminator $\mathcal{D}_Y$ is thus:

$$\mathcal{L}_{GAN}(\mathcal{G}_{XY}, \mathcal{D}_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log \mathcal{D}_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - \mathcal{D}_Y(\mathcal{G}_{XY}(x)))] \tag{2.3}$$

Similarly, for the reverse mapping $Y \to X$, the generator $\mathcal{G}_{YX}$ translates images from domain Y to domain X, and the discriminator $\mathcal{D}_X$ distinguishes between real images from domain X and generated



(a) The real image from domain X is translated into domain Y using generator $\mathcal{G}_{XY}$. The generated image from domain Y is further translated back into domain X using generator $\mathcal{G}_{YX}$.

(b) The real image from domain Y is translated into domain X using generator $\mathcal{G}_{YX}$. The generated image from domain X is further translated back into domain Y using generator $\mathcal{G}_{XY}$.

Figure 2.2: The Cycle-GAN basic architecture consists in two generators and two discriminators, one for each domain.

images $\mathcal{G}_{YX}(y)$. The adversarial loss for generator $\mathcal{G}_{YX}$ and discriminator $\mathcal{D}_X$ is thus:

$$\mathcal{L}_{GAN}(\mathcal{G}_{YX}, \mathcal{D}_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[\log \mathcal{D}_X(x)] + \mathbb{E}_{y \sim p_{data}(y)}[\log(1 - \mathcal{D}_X(\mathcal{G}_{YX}(y)))] \tag{2.4}$$

The cycle-consistency loss ensures that if an image is translated from one domain to another and back again, the result should be the image in the original domain. Thus in order to encourage the mappings $\mathcal{G}_{XY}$ and $\mathcal{G}_{YX}$ to be consistent with each other, a reconstruction loss is applied between the image in the first domain and the reconstructed image in the same domain:

$$\mathcal{L}_{cycle}(\mathcal{G}_{XY}, \mathcal{G}_{YX}) = \mathbb{E}_{x \sim p_{data}(x)}[||\mathcal{G}_{YX}(\mathcal{G}_{XY}(x)) - x||_1] + \mathbb{E}_{y \sim p_{data}(y)}[||\mathcal{G}_{XY}(\mathcal{G}_{YX}(y)) - y||_1] \tag{2.5}$$

Similarly to the original GAN loss, in the final objective, the generators $\mathcal{G}_{XY}$ and $\mathcal{G}_{YX}$ seek to minimise the total loss (fooling the discriminators), while the discriminators $\mathcal{D}_X$ and $\mathcal{D}_Y$ seek to maximise it by correctly distinguishing real from generated images:

$$\min_{\mathcal{G}_{XY}, \mathcal{G}_{YX}} \max_{\mathcal{D}_X, \mathcal{D}_Y} \mathcal{L}(\mathcal{G}_{XY}, \mathcal{G}_{YX}, \mathcal{D}_X, \mathcal{D}_Y) \tag{2.6}$$

Thus, the final loss becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{GAN}(\mathcal{G}_{XY}, \mathcal{D}_Y, X, Y) + \mathcal{L}_{GAN}(\mathcal{G}_{YX}, \mathcal{D}_X, Y, X) + \mathcal{L}_{cycle}(\mathcal{G}_{XY}, \mathcal{G}_{YX}) \tag{2.7}$$

The advantage of Cycle-GANs over GANs is the ability to be trained unsupervised, as opposed to GANs that would require paired datasets in order to be trained for image-to-image translation where structure is preserved. While training a GAN would require the same exact scene in two domains (e.g. a horse and a zebra in the same pose), this assumption is relaxed by the introduction of the cycle consistency loss which imposes constraints on the structure of the images produced by the two generators.

One drawback of early Cycle-GANs is their scalability, as a Cycle-GAN can only learn a one-to-one mapping between two domains which can become impractical as the number of domains grows. To this extent, latent embeddings that encode domain-specific information can enable one-to-many mappings by allowing the same generator to work across multiple domains. For example, Xun Huang et al. (2018) split the latent representations across a content embedding, which captures structure, and a style embedding, which captures features that are specific to a domain. To perform image translation, a content embedding is recombined with a style embedding corresponding to the desired domain, and the resulting latent embedding is decoded into an image representative of the target domain.

Finally, as opposed to GANs, Cycle-GANs also require at least double the computation requirements of GANs due the dual-network architecture and the extra loss constraints, leading to longer training times and higher memory usage.

2.4 Denoising Diffusion Models

Diffusion-based image models are a class of generative models that learn to sample data, by progressively refining random noise into structured data. The key idea is to model the reverse of a diffusion process, where an image is incrementally denoised to recover the original structure. These models have shown great promise in generating high-quality images and have been applied in various fields such as computer vision, art generation and more.

The process consists in 2 parts: a forward diffusion process and a reverse diffusion process, modeled as two Markov chains (where each step depends on the output of the previous one).

In the forward diffusion process, noise is gradually added to an image over a series of time steps. This can be mathematically represented as a Markov chain where each step adds a small amount of Gaussian noise to the image. The forward process can be described by equation $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}$, where x_t is the image at time step t, x_{t-1} is the image at time step $t-1$, β_t is a variance schedule at time step t, and ϵ_{t-1} is Gaussian noise at time step $t-1$.

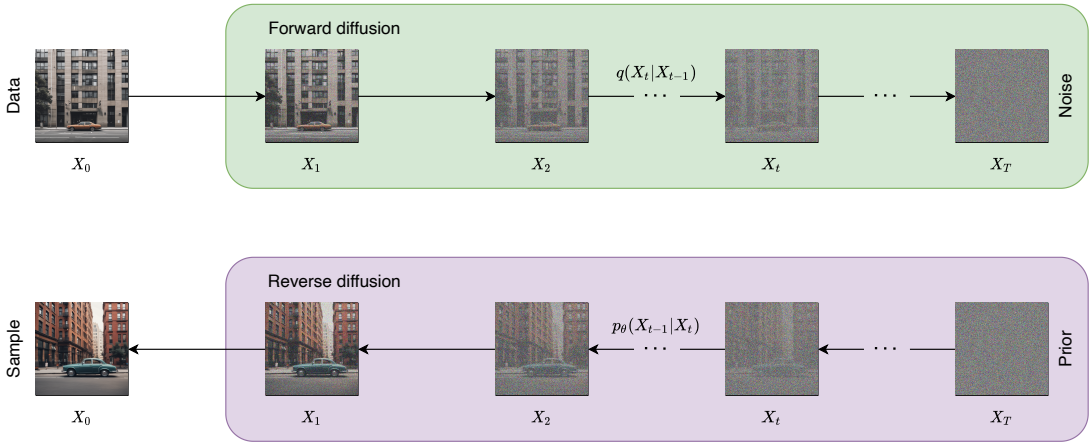


Figure 2.3: The diffusion process consists in a forward diffusion step were noise is gradually added to the input, and a reverse diffusion step, where noise is gradually subtracted from the noisy representation.

Using the addition property of Gaussian distributions, the forward diffusion process can be re-expressed as: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, which is useful, as it can be performed in a single step.

In the denoising process, the goal is to reverse the noise addition process in order to recover the original image. This is done by learning a denoising model that predicts the noise added at each step. The reverse process can be represented as $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\Big)$, where ϵ_θ is a neural network trained to predict the noise ϵ_t given the noisy image x_t and the time step t.

The training objective is to minimize the difference between the true noise ϵ_t and the noise $\epsilon_\theta(x_t, t)$ predicted by the neural network. This can be formulated as:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, \epsilon_t, t}\left[\|\epsilon_t - \epsilon_\theta(x_t, t)\|_2^2\right] \tag{2.8}$$

In order to synthesize novel images at inference time, the model starts from pure Gaussian noise and iteratively applies the learned reverse diffusion process, subtracting the predicted noise at every step.

Unfortunately, the iterative nature of the reverse diffusion process makes the image synthesis a costly method in terms of time and computational requirements, especially at higher resolutions. To this end, latent diffusion models have been designed to operate in a lower dimensional latent space, which increases computational efficiency while maintaining content fidelity. This is done through an encoder-decoder network, where the encoder is trained to transform an input image into a latent embedding z and a decoder is trained to reconstruct it back. During the inference phase, a latent embedding z_t is drawn from a Gaussian distribution, and ϵ_θ is applied to progressively denoise z_t. Finally, the decoder generates an RGB image from the denoised latent embedding.

Unlike GANs, where the generator and discriminator are trained adversarially, training diffusion models is stable and the synthesis process is performed step by step.

2.5 Neural Radiance Fields

Introduced by Mildenhall et al. (2020), NeRFs are a method for representing 3D scenes as a continuous function, generally parametrised by a neural network. This representation is useful, as it is learnable end-to-end from simple collections of 2D images with known poses, and is able to produce both RGB and depth images of scenes from novel viewpoints. Many areas of robotics require a mechanism for representing or storing 3D information about the environment (e.g. path planing in autonomous driving, robotic grasping and manipulation), where NeRFs can additionally help by encoding not just color but also features of the scene such as semantic segmentation.

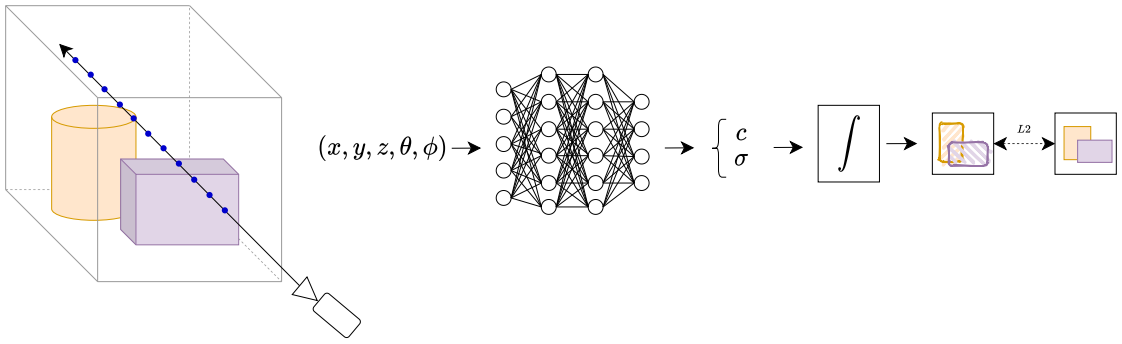


Figure 2.4: Basic NeRF process: given 3D point coordinates and a viewing direction, a Multi-Layer Perceptron (MLP) outputs colors and densities, which are accumulated to render a image.

Given a 3D point $\mathbf{p} \in \mathbb{R}^3$ in the world frame with coordinates (x, y, z), and a 2D viewing direction $\mathbf{d} \in \mathbb{R}^2$ with azimuth and elevation (θ, ϕ), a NeRF model outputs a color $\mathbf{c} \in \mathbb{R}^3$ with red, green and blue intensities (r, g, b), and a density $\sigma \in \mathbb{R}$. The function is parameterized by a neural network with parameters Θ and can be expressed as: $F_\Theta(\mathbf{p}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$.

In order to render an image, using a simple pinhole camera model, rays need to be cast from the camera into the scene. First, normalised unit direction vectors of 3D rays are obtained by unprojecting pixel coordinates (u, v) from the image space to the normalized camera space using the inverse of the intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$:

$$\mathbf{d}_c = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2.9}$$

where f_x and f_y are the focal lengths of the sensor on the X and Y axes (in pixels), c_x and c_y are the coordinates of the optical center (in pixels) and the intrinsic matrix $\mathbf{K}$ is given by:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

Then, the camera extrinsic matrix $[R|\mathbf{t}] \in \mathbb{R}^{3\times4}$ describing the position and orientation of the camera in the real world is used in order to transform the normalized 3D rays from the camera space to the world space. For each ray, its origin in world coordinates is given by $\mathbf{o}_w = \mathbf{t}$, and its direction by $\mathbf{d}_w = R\mathbf{d}_c$, where:

$$[R|\mathbf{t}] = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{array}\right] \tag{2.11}$$

Thus in NeRF, a ray is defined by its origin $\mathbf{o}_w$ and direction $\mathbf{d}_w$, with points along the ray being determined by their distance t from the camera pinhole: $\mathbf{r}(t) = \mathbf{o}_w + t\mathbf{d}_w$.

When forming the image, each ray will correspond to a pixel, and the expected color $C(\mathbf{r})$ of a pixel corresponding to a particular ray $\mathbf{r}$ is calculated by accumulating the product of the transmittance $T(t)$, the density $\sigma(\mathbf{r}(t))$ and the radiance $\mathbf{c}(\mathbf{r}(t), \mathbf{d})$ along the coordinates of points belonging to the ray $\mathbf{r}$, bounded by the near plane t_n and far plane t_f:

$$C(\mathbf{r}) = \int_{t=t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt \tag{2.12}$$

where t_n is the minimum distance along the ray from the camera origin starting from which points are sampled, t_f is the maximum distance along the ray from the camera origin up to which points are sampled and the transmittance $T(t)$ is the probability that light has not been occluded or absorbed up to point t, being expressed as $T(t) = \exp\left(-\int_{s=t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$.

In practice, the integral above is approximated as a discrete sum over points sampled along rays. For N points sampled uniformly along each ray, the rendered pixel color as a result of ray $\mathbf{r}$ becomes:

$$\hat{C}(\mathbf{r}) \approx \sum_{i=1}^{N} T_i \Big(1 - \exp(-\sigma_i \delta_i)\Big) c_i \tag{2.13}$$

where $T_i = \exp\Big(-\sum_{j=1}^{i-1} \sigma_j \delta_j\Big)$ is the cumulative transmittance up to point i, σ_j is the density at the j-th point and δ_j is the distance between the j-th and $(j+1)$-th points.

The radiance field is typically estimated using MLPs, which unfortunately have difficulty learning high frequency details such as textures and sharp edges. To overcome this issue, positional encoding is used in order to transform the input coordinates into higher-dimensional embeddings, allowing the network to learn high frequency details:

$$\gamma_L(\mathbf{p}) = \Big[\sin(2^0\pi\mathbf{p}), \cos(2^0\pi\mathbf{p}), \sin(2^1\pi\mathbf{p}), \cos(2^1\pi\mathbf{p}), \ldots, \sin(2^{L-1}\pi\mathbf{p}), \cos(2^{L-1}\pi\mathbf{p})\Big] \tag{2.14}$$

where $\gamma_L(\cdot)$ is applied to both the 3D point coordinates normalized between $[-1, 1]$ and viewing directions expressed as unit vectors.

The term L denotes the number of sine and cosine functions of progressively higher frequencies applied to the input, with higher L values leading to sharper details and lower values leading to smoother output. Since higher detail is needed to reconstruct sharp or detailed object boundaries, a higher number of functions need to be applied on the 3D point coordinates, whereas view-dependent effects are smoother, requiring a smaller L. Once the encoding is produced, the input to the MLP becomes: $F_\Theta(\gamma(\mathbf{p}), \gamma(\mathbf{d})) \rightarrow (\mathbf{c}, \sigma)$.

Finally, the network's parameters Θ are optimized during training by minimizing the difference between the colors of the rendered pixels and the ground-truth pixels. The most common loss used is:

$$\mathcal{L}(\Theta) = \sum_{\mathbf{r} \in \mathcal{R}} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2 \tag{2.15}$$

where $\mathcal{R}$ is the set of all rays cast for the camera poses corresponding to the images in the training set, and $C(\mathbf{r})$ is the ground-truth color of the pixel corresponding to ray $\mathbf{r}$.

One of the main advantages of NeRFs is their compact and memory-efficient representation. Since the structures are represented implicitly using continuous functions, rather than explicitly using data structures such as meshes, voxels or point clouds, the scene is often compressed in 5-10 MB (Mildenhall et al., 2020), representing the model weights. This has an impact on both the memory

footprint but also on scalability as we move towards reconstruction of larger scenes. For example, since voxel representations store data regarding color, density and other attributes of the scene in 3D arrays, reconstructing larger outdoor scenes requires more memory, which increases cubically. Also, finer details are lost because a fixed resolution is imposed. To overcome this, more specialized subdivision algorithms have been proposed, such as octrees and sparse octrees (Laine & Karras, 2010). On the other hand, since in NeRFs the 3D geometry is learned, not stored explicitly, they can be queried at arbitrary coordinates, and the discretization artifacts are reduced since the scene is interpolated smoothly. In point clouds, objects are represented as a set of discrete points in space (consisting in 3D coordinates, and RGB values or laser beam intensities), and although lightweight, their sparsity makes high-quality rendering difficult without extra interpolation methods, while NeRFs directly learn the 3D scene structure and appearance. In mesh-based representations, the surface of 3D objects is represented using a collection of vertices, edges and faces. Unfortunately, since their topology is variable, integration with machine learning methods is difficult and representing fine details is challenging - requiring shader programming, while NeRFs implicitly learn the reflections, shadows, and transparency from input images.

Finally, NeRF-based approaches have several drawbacks as well – they are computationally intensive, which makes them slow to train and slow at inference time, they are overfit to a single scene, requiring a new model to be trained for each new scene, and in their original formulation are limited to only representing static scenes.

2.6 Metrics

The criteria that synthesized data must meet varies based on its intended application. For domains such as entertainment and creative design, the data should be aesthetically pleasing and photorealistic however, when the data is intended to be used for training other models, it is crucial that it is **also** semantically accurate and well-aligned with the ground-truth, which is particularly important in autonomous driving. As such, in the following subchapter, metrics will be categorised based on the function they perform: either to assess the visual quality of the data, or to assess the goodness-of-fit as training data.

Out of the metrics presented in the sections below, we employ Inception Score (IS), Fréchet Inception Distance (FID), Kernel Inception Distance (KID), Fréchet Video Distance (FVD) and JEPA Embed-

ding Distance (JEDi) for evaluating visual quality, and mean Average Precision (AP), mean Intersection Over Union (mIoU) and Root Mean Squared Error (RMSE) when evaluating the downstream task.

2.6.1 Visual quality

Metrics for visual quality can be categorised based on many criteria, but for the purpose of the analyses performed in this study, they can be mainly split into two categories, depending on ground-truth availability - paired and unpaired. Additionally, based on the type of data on which they are applied, they can be categorised as image-based metrics and video-based metrics.

For cases where reference data or ground-truth exists, methods such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) or Learned Perceptual Image Patch Similarity (LPIPS) are often employed. On the other hand, when no reference data exists, or if higher degrees of diversity are either desired or simply a result of the models used, unpaired approaches are used.

From the paired category, MSE is a commonly employed metric in both image and video generation tasks, comparing images or frames on a pixel-for-pixel basis:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(I_i - \hat{I}_i\right)^2 \tag{2.16}$$

where I_i is a pixel value from the reference image and $\hat{I}_i$ is the corresponding pixel value from the reconstructed image. Used more often in practice, and building on MSE, PSNR is a metric that evaluates the quality of a processed output signal in comparison to a reference signal, after undergoing a transformation such as compression:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right) \tag{2.17}$$

where MAX_I is the maximum possible pixel value. While it is straightforward, quick and cost-efficient, it does not correlate very well with human perception of image quality (Z. Wang et al., 2004), as any deviation from the ground-truth image will lead to a reduced score, thus penalizing diversity. Since it compares pixel values directly, it is unable to capture structural or semantic similarities between images that are not a perfect match to each other, which is especially important in GAN evaluation, where both realism and diversity of outputs are more important than replicating pixel values. In the case of videos, the metric is computed on a frame by frame basis and averaged across frames, thus failing to

capture any temporal coherence. Given its drawbacks, it is considered more appropriate for tasks such as image reconstruction, denoising or super-resolution, where pixel-for-pixel reconstruction is the goal.

Alternatively, SSIM (Z. Wang et al., 2004) has been proposed to address the limitations of PSNR and MSE, and their lack of correlation with human visual perception. The metric is calculated on patches of local pixel intensities, typically on the luma channel, and averages resulting values across all image patches to obtain a global SSIM value which measures the degree of similarity between the ground-truth reference image and the reconstructed image:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.18}$$

where μ_x, μ_y are mean intensities of patches x and y, σ_x^2, σ_y^2 are variances of patches x and y, σ_{xy} is the covariance between patches x and y, while C_1 and C_2 are small constants that stabilize the division.

While the approach is more robust to small amounts of spatial misalignment as opposed to PSNR and MSE, it still penalises variations that might be semantically correct but are structurally different, and ignores chromatic consistency if only luma information is used. This makes it poorly suited for unsupervised image synthesis, especially Cycle-GANs, where reference images or paired datasets are not available.

In contrast to PSNR and SSIM, LPIPS (R. Zhang et al., 2018) is a method that evaluates both perceptual and semantic quality, and aligns more closely with human visual perception, frequently surpassing PSNR and SSIM in studies on image similarity. This makes it better suited for evaluating diverse GAN outputs that have multiple plausible variations, as it does not directly rely on pixel-to-pixel or patch-to-patch comparisons, instead comparing features extracted from deeper layers of Convolutional Neural Networks (CNNs), which are generally better at encoding shape, textures or style:

$$\text{LPIPS}(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\phi_l(x)_{h,w} - \phi_l(y)_{h,w})\|_2^2 \tag{2.19}$$

where $\phi_l(x)$ and $\phi_l(y)$ are the feature maps of patches x and y at layer l, H_l and W_l are the height and width of the feature map at layer l, w_l is a learned weight vector for channel importance at layer l, and $\odot$ is the Hadamard product.

In practice, since LPIPS uses features extracted from images using a pre-trained network such as VGG (Simonyan & Zisserman, 2015), it is more computationally demanding compared to PSNR and SSIM, as for each image it requires a forward pass through the network to obtain feature maps.

Moreover, the result is also dependent on the chosen network, making scores harder to compare across literature without standardization of CNN backbones (Goldblum et al., 2023).

From the unpaired category of metrics – when no aligned ground-truth data exists, IS (Salimans et al., 2016) was one of the first methods employed to evaluate the quality of GAN outputs, being based on the results of Inception-v3 image classifier pre-trained on ImageNet (Szegedy et al., 2016; Deng et al., 2009). It takes into account both classifier confidence, interpreted as a measure of how classifiable or "distinct" the generated images are, while also measuring coverage across all possible classes as a way to approximate diversity:

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{x\sim G}\left[D_{\mathrm{KL}}\left(p(y \mid x) \,\|\, p(y)\right)\right]\right) \tag{2.20}$$

where $x \sim G$ is a generated image, $p(y \mid x)$ is the predicted label distribution from a pretrained classifier, $p(y)$ is the marginal label distribution over all generated samples, and D_{KL} is the Kullback–Leibler divergence. A high IS score will be achieved when $p(y \mid x)$ has low entropy (i.e. the model is confident about its prediction with all probability focused on a single label) and $p(y)$ has high entropy (meaning that the entire set of generated images covers many individual classes).

The main drawback however is its reliance on the ImageNet-trained classifier which can lead to poor generalisation to other domains while also being susceptible to distortions, noise or adversarial attacks, which can lead to spurious high confidence classifications for images that are incoherent (Barratt & Sharma, 2018). Unlike more modern approaches, it does not compare distributions of real and generated images thus being limited to the domain captured by ImageNet.

Another metric that makes use of an ImageNet-trained Inception-v3 network is FID (Heusel et al., 2017), where the last pooling layer is used to extract features from both a set of generated images and a set of real images. The method assumes that the extracted features follow a multivariate Gaussian distribution, and computes the Fréchet Distance between the distributions of the two sets of features, parametrised by their means and covariance matrices:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{\frac{1}{2}}\right) \tag{2.21}$$

where μ_r and μ_g represent the mean of features from real and generated images, and Σ_r and Σ_g represent the covariance of features from real and generated images.

As opposed to PSNR, SSIM and LPIPS, the metric does not penalise GAN outputs with plausible variations and since it does not make use of the classifier output, it is less prone to spurious results compared to IS, making it a popular choice in literature. While it does not require paired data, the Gaussian assumption may not always hold, especially in complex datasets (G. Y. Luo et al., 2025), and the method requires a relatively large number of observations to accurately estimate the covariance matrices due to the high-dimensional feature vectors extracted from the Inception-v3 network.

An alternative to FID is KID (Bińkowski et al., 2018), which also uses features extracted from Inception-v3, but does not make any assumption about the distribution of the features. Instead, it calculates the distance between the distributions of the real and generated images using the squared Maximum Mean Discrepancy (MMD) with a polynomial kernel:

$$\text{KID} = \text{MMD}^2 = \mathbb{E}[k(x, x')] + \mathbb{E}[k(y, y')] - 2\,\mathbb{E}[k(x, y)] \tag{2.22}$$

where (x, x') are pairs of generated images, (y, y') are pairs of real images, $\mathbb{E}[k(x, x')]$ is the average kernel similarity between pairs of generated images, reflecting consistency within each distribution (i.e. how similar fake images are to each other), while $\mathbb{E}[k(y, y')]$ is the average kernel similarity between pairs of real images. On the other hand, $\mathbb{E}[k(x, y)]$ is the average kernel similarity between real and generated images, measuring the cross-distribution similarity. The kernel function, $k(x, y)$, is defined as:

$$k(x, y) = \left(\frac{1}{d} x^\top y + 1\right)^3 \tag{2.23}$$

where x and y are the feature vectors being compared and d is their length.

FID and KID are often reported together, as the latter provides an unbiased estimation through the use of MMD and can better capture higher order statistics, compared to FID, which only accounts for mean and covariance. On the other hand, KID is more computationally expensive than FID to calculate, with a runtime that has a quadratic complexity as a function of sample size, with many approaches instead computing KID on random subsets. Subsequent literature has also addressed the limitations of feature extractors that were trained on ImageNet, proposing to replace Inception-v3 with Contrastive Language-Image Pretraining (CLIP) embeddings, as they are richer and more expressive, being trained on a much larger dataset of 400 million image-text pairs (Radford et al., 2021; Jayasumana et al., 2024).

While metrics such as FID and KID can be used to reason about the quality or diversity of individual images, they are not suitable for doing the same for the case of video generation, where spatio-temporal

consistency plays a key role. FVD (Unterthiner et al., 2019) is a metric designed to assess the quality of videos generated by GANs and other models, and uses a formulation that is very similar to that of FID. It extends the concept of FID to the video domain, allowing to compare distributions of generated and real videos, and makes use of features extracted from a video classification model such as the I3D network trained on the Kinetics Human Action Video Dataset (Carreira & Zisserman, 2017; Kay et al., 2017). Besides the source of the features, the metric is computed identically to FID and has the same advantages and drawbacks, such as bias and a requirement for larger number of observations. Furthermore, no explicit mechanisms exist to measure temporal coherence – this is done implicitly by leveraging embeddings computed using a backbone trained on videos. Furthermore, recent literature has highlighted that FVD scores are affected more by spatial distortions than spatio-temporal ones, which suggests that FVD may not be adequate at capturing longer-term spatial or stylistic coherence (S. Ge et al., 2024).

To address some of the drawbacks of FVD, G. Y. Luo et al. (2025) introduced JEDi, which uses features extracted from a Video Joint Embedding Predictive Architecture (V-JEPA) model and employs MMD with a polynomial kernel to measure the distance between distributions of features extracted from real and generated videos. Similar to KID, this makes JEDi an unbiased estimator which makes no assumption about the feature space while also requiring significantly less observations. Additionally, it is also shown that the metric has better alignment with human evaluation of video quality and consistency. Furthermore, the advantage of using V-JEPA as a feature extractor is that, during training, it learns to predict latent representations of masked regions in videos, which encourages the model to learn both contextual relationships and spatio-temporal dynamics.

Introduced by J. Liu et al. (2024), Fréchet Video Motion Distance is another metric intended to help measure the temporal coherence and motion quality of videos. It relies on PIPs++, which is a keypoint tracking model trained on a large synthetic dataset, to track keypoints across frames and capture motion patterns (Zheng et al., 2023). It then computes velocity and acceleration vectors from the keypoints, and aggregates them into histograms to capture the distribution of both motion direction and magnitude. These histograms are treated as embeddings, and similar to FID, are used to compute the Fréchet Distance between the distribution of embeddings extracted from real and generated videos, with the same drawbacks as those specific to FID. As it focuses on the dynamics of motion, it can capture unnatural movement or temporal inconsistencies such as jitter, but it needs to be used as a complement to other video quality metrics, as appearance is ignored.

Finally, the Vendi score is a general purpose metric for measuring diversity, which can also be applied to GAN outputs to assess diversity independent from fidelity and without requiring a paired or reference dataset (Friedman & Dieng, 2023). Additionally, both the embedding type and the similarity function can be chosen by the user, making it highly adaptable cross domains. The process involves calculating a similarity matrix, normalising it, computing its eigenvalues, then the Shannon entropy of the eigenvalues, and finally exponentiating the entropy to obtain the Vendi score.

More specifically, given a feature extractor ϕ which produces embeddings of images x, a similarity function such as cosine-similarity can be used as a kernel that computes the similarity between two images $k(x, x')$:

$$k(x, x') = \exp\left(\frac{\phi(x) \cdot \phi(x')}{\|\phi(x)\| \, \|\phi(x')\|}\right) \tag{2.24}$$

where $\phi(x) \cdot \phi(x')$ is the dot product of the feature vectors, while $\|\phi(x)\| \, \|\phi(x')\|$ is the product of their norms.

A similarity matrix K is constructed for the dataset using the similarity function k, then normalised, and finally the eigenvalues of K are computed. Given the set of normalized eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, the Vendi score is calculated by exponentiating the Shannon entropy of the eigenvalues:

$$\text{Vendi Score} = \exp(\mathrm{H}) = \exp\left(-\sum_{i=1}^{n} \lambda_i \log \lambda_i\right) \tag{2.25}$$

The metric can be used in conjunction with other metrics such as FID or KID to give a more complete view of both model performance and output diversity.

2.6.2 Quality as training data

The usefulness of synthetic data as training data can be evaluated using a combination of task-specific metrics, domain-adaptation analyses and performance benchmarking. The choice of metric in evaluating the quality of the generative model output is dependent on the output type and the target task. If the model outputs image and semantic segmentation map pairs and the goal is to train a downstream semantic segmentation network, the mIoU can be computed in three distinct cases: 1) between what is now considered to be the ground-truth semantic map (i.e. the semantic map obtained from the generative model), and the semantic map predicted by a pre-trained segmenter network

from the generated image, as a way to validate the alignment of the data before training, 2) between segmenter outputs on both real and synthesized images (referred to in the manuscript as mIoU-align), and 3) between the predictions of the newly-trained segmentation network and ground-truth, when validating on real data after training on synthetic data.

In order to calculate mIoU, assume first there are N classes. For each class $i \in N$, TP_i (True Positive) will be the number of correctly predicted pixels as class i, FP_i (False Positive) will be the number of pixels incorrectly predicted as class i and FN_i (False Negative) will be the number of pixels of class i that have been misclassified. Thus mIoU is then given by the Intersection Over Union of each class i, averaged across all classes N:

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{2.26}$$

While mIoU is not a general-purpose metric for measuring the quality of generative models, it is useful when the output is a semantic segmentation map that should correspond to a particular ground-truth semantic segmentation map, or when the downstream task itself predicts a segmentation map.

Thus 2 types of mIoU are computed in this manuscript, which are further employed in Chapters 6 and 7. The first compares the ground-truth semantic segmentation map and the semantic segmentation map generated by the synthesis model, measuring the alignment of the synthesized segmentation output. The second version (which we denote as mIoU-align), compares the semantic segmentation map obtained by running the ground-truth image through an off-the-shelf segmenter with the semantic segmentation map obtained by running the synthesised image through the same off-the-shelf segmenter, in this case Deeplabv3+ (L.-C. Chen et al., 2018). This is used to measure the semantic alignment of the synthesised image with the ground-truth image, while the reason why both the ground-truth image and the synthesised image are passed through the segmenter is to control for the drop in segmentation quality inherent to the off-the-shelf segmentation model itself.

For the cases where the generative model outputs pairs of RGB images and depth maps, monocular depth estimation or completion models such as PEnet (M. Hu et al., 2021) can be trained or validated. To measure the quality of the generated depth, RMSE is commonly employed:

$$\text{RMSE} = \sqrt{\frac{1}{|N|}\sum_{i\in N}(\hat{d}_i - d_i)^2} \tag{2.27}$$

where $\hat{d}_i$ is the predicted depth value at pixel i, d_i is the ground-truth depth value at pixel i and N is the set of valid pixels for which a depth value is recorded.

For tasks such as object detection, where bounding box coordinates are used instead of segmentation masks, the generated segmentation or instance map can be used to determine box coordinates of objects in the image by finding the minimum and maximum of the X- and Y-axis coordinates of the object's pixels. In this case, the metrics will be computed between the extracted bounding boxes and the predicted bounding boxes of the detector network. Other common metrics used in evaluating this type of task include precision, recall, F1-score and mean AP.

3

Literature review

Contents

3.1 Introduction

High-quality data is essential for robotics and computer vision tasks. Historically, various approaches have been used to train and validate models, each with their own advantages and disadvantages. In this thesis, data sources will be generally divided into 3 main categories: real-world captured data, data generated using classical simulators such as 3D engines and data generated using methods that learn the distribution of the input data, referred to as synthetic data. The latter can itself be divided into domain adaptation (where existing data is transformed to match target domains) and data synthesis (where new data is created).

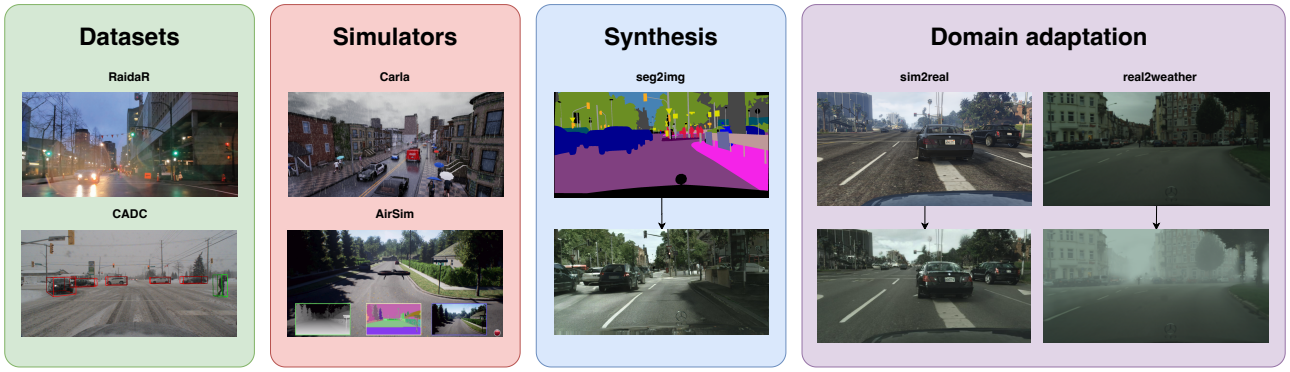


Figure 3.1: A visual example of sources of data – datasets such as RaidaR (Jin et al., 2021), CADC (Pitropov et al., 2021), simulators such as Carla (Dosovitskiy et al., 2017), AirSim (Shah et al., 2018), data synthesis methods such as Pix2pixHD (T. Wang et al., 2018) and domain adaptation methods such as sim2real (S. R. Richter et al., 2022), Foggy Cityscapes (Sakaridis et al., 2018).

3.1.1 Real-world captured data

In general, data captured in the real world will represent an upper bound on the amount of realism and domain alignment, since it represents a world that obeys the laws of physics, with accurate and natural dynamics and kinematics. Unfortunately, these desirable properties come with a set of notable disadvantages, as data capture is expensive, requires specialised equipment (sensors, mobile platforms), support from engineers, and especially in autonomous driving, the need for careful planning and

logistics. This becomes especially challenging because, unlike indoor data collection which is a more flexible process, one has to account for other traffic participants, dynamic agents and other constraints.

Additionally, the efficiency and usefulness of captured data is significantly influenced by the difficulty of capturing edge cases and rare events such as accidents, failures, near misses or adverse weather. Furthermore, manual annotation can be expensive, especially when performed on a per-pixel level, although pseudo ground-truth can be provided as long as any model bias is taken into account. Finally, the introduction of privacy laws such as General Data Protection Regulation (GDPR) has reduced the scalability of data collection but also imposed limitations on collaboration and sharing between EU and non-EU parties.

3.1.2 Data sourced from simulators

On the opposite side of the spectrum, data generated using classical simulators such as those based on 3D engines offers notable benefits compared to data collected in the real world, yet it also comes with a set of drawbacks.

Once a base simulator has been implemented (either 2D or 3D, with or without a physics engine) the process of generating data is more cost-efficient, highly scalable, and corresponding ground-truth such as bounding boxes, semantic and instance segmentation maps or depth maps is comparatively straightforward to obtain, and generally free from annotation mistakes.

Multiple sensors and modalities can be simulated, such as images, LiDAR, Radio Detection And Ranging (RADAR) or Inertial Measurement Unit (IMU) data, but also less conventional sensors such as event cameras, and usually ground-truth can be provided for each of these. Furthermore, the environment is fully controllable, allowing for modeling of edge cases or rare events, with accurate modeling of friction, gravity, collisions, and so on. Additionally, there are no issues with logistics, GDPR or sensor and vehicle availability, and even more importantly AV stacks or stack components can be integrated to run in simulation to aid development and reduce time-to-deployment.

Unfortunately, this approach also comes with its own shortcomings, the most concerning being the domain gap (the sim to real domain gap) between the distribution of data generated by simulators and that obtained from real-world observations. In the case of images, lighting and textures might be different due to the use of simplified graphics. In terms of dynamics, there might be inconsistencies because the simulated physics are only approximations of the forces acting in the real world. Similarly,

other simulated sensors such as LiDAR might be represented as a more idealized version than their real-world counterparts, which suffer from issues in adverse weather and even hardware faults.

Furthermore, the fidelity of simulation often depends on the available computational resources, with high fidelity simulations requiring increasingly more expensive hardware. Although scalable, the existence of a simulator alone is insufficient, as different techniques must be employed to produce scenarios, agent trajectories or to procedurally generate environments. As such, human involvement is still high, with many 3D assets requiring the expertise of 3D artists to reach an adequate level in closing the sim to real domain gap.

3.1.3 Synthetic data

Data generated using models that implicitly learn a distribution emerges as a method to address some of the inherent limitations of both data collected in the real world and data sourced from simulators. Once a model has been trained, it can serve as an efficient method to produce diverse data. Additionally, re-targeting a different domain usually involves fine-tuning, which can be faster and more cost-efficient than recreating assets or environments in a simulator, or than collecting the same amount of data in the real world.

Traditionally, training or fine-tuning a model involved access to an initial dataset, most of the time the choice being real-world datasets, however modern approaches such as diffusion models have been shown to extrapolate from their training data. For example, a model that is trained with images of cats and images of moons can generate an image of a cat on the moon given the prompt, even if this combination was not present in the training dataset. As a result, such models allow for rare events and edge cases to be generated more easily or safely.

Notably, these models can also be applied to address the primary limitations of each of the other two methods – real world data can be transformed and multiplied, thus increasing its diversity and usefulness, while data from simulation can benefit from a reduced domain gap via domain adaptation or from conditioning a model using the pixel-perfect ground-truth produced by simulators.

The primary disadvantage of data synthesis revolves around a domain gap that is lower than that of data from simulation, but still significant. Additionally, training data requirements increase with the complexity of conditioning. For example, models that are prompted with text only are easier, cheaper and faster to train compared to those that incorporate 2D information such as segmentation maps, or

3D information such as point clouds or meshes, but at the same time they are far more ambiguous since natural language as a conditioning input is comparatively less rich.

Models can also be set up to generate both the target modalities (images) and corresponding ground-truth (semantic segmentation, depth map), but the alignment and correspondence between the two is not guaranteed unless specific architectural choices are made and adequate data is available for training.

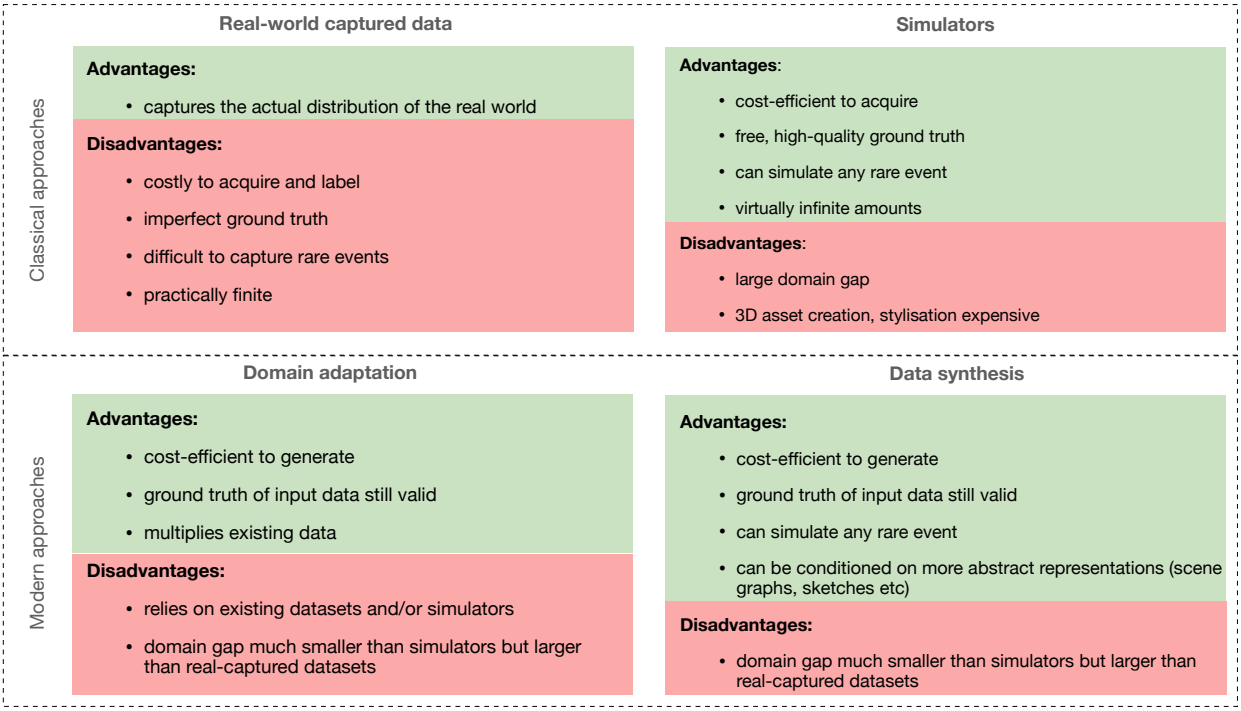


Figure 3.2: The advantages and disadvantages of classical and modern approaches to data.

Therefore, selecting the most suitable data source relies heavily on the domain and task, with the optimal outcomes being achieved through a combination of two or more of the above approaches, leading to synergistic effects. Some combinations include the following, as also visually explained in Fig. 3.3:

- Multiplying existing real-world data using domain adaptation techniques such as in Multi-weather city (Musat et al., 2021);
- Augmenting existing data with novel views such as Klinghoffer et al. (2023);
- Reducing the domain gap of data from simulation using domain adaptation such as S. R. Richter et al. (2022);
- Using ground-truth from real world (or simulator) data to condition models and obtain more data such as in Depth-SIMS (Musat et al., 2022), NeuralFloors (Muşat et al., 2024a), NeuralFloors++ (Muşat et al., 2024b), SimGen (Y. Zhou et al., 2024);

- Multiplying synthetic data using domain adaptation models such as a combination between Multi-weather city (Musat et al., 2021) by first transforming real world data and subsequently synthesizing fog using an analytical approach such as Foggy Cityscapes (Sakaridis et al., 2018);
- Multiplying synthetic data by swapping class labels and sampling style latent embeddings from diverse datasets.

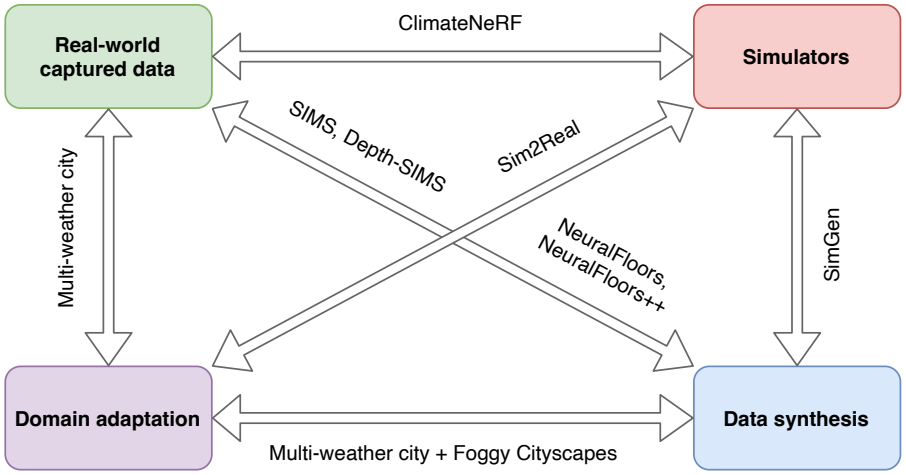


Figure 3.3: Combining approaches leads to synergistic effects – SimGen (Y. Zhou et al., 2024), Sim2Real (S. R. Richter et al., 2022), Foggy Cityscapes (Sakaridis et al., 2018), ClimateNeRF (Yuan Li et al., 2023) or SIMS (Qi et al., 2018), and as part of this manuscript – Multi-weather city (Musat et al., 2021), Depth-SIMS (Musat et al., 2022), NeuralFloors (Muşat et al., 2024a), NeuralFloors++ (Muşat et al., 2024b).

Data synthesis thus plays a critical role in robotics and autonomous driving as it alleviates the burden of data collection and limits reliance on real world data. It thus enables much safer and easier modeling of edge cases and rare scenarios, increases data usefulness, and provides a more scalable and controllable process for generating the data required.

3.2 Datasets

With the progression of machine learning methodologies, driven by advancements in both software and hardware, model architectures have grown increasingly sophisticated and computationally intensive, requiring larger volumes of data for effective training. Consequently, substantial efforts have been directed towards the development of more expansive datasets that encompass a broader range of geographical regions, multiple sensor modalities, diverse scenarios, and comprehensive ground-truth annotations.

Prominent examples of early vision-based datasets include: CamVid (Brostow et al., 2009) and Cityscapes (Cordts et al., 2016) in overcast daytime, followed by BDD100K (F. Yu et al., 2020)

and Mapillary (Neuhold et al., 2017) in diverse weather, offering a mix between 2D bounding boxes and pixel-wise semantic labels. However, due to their limited modality and ground-truth annotations, these datasets support a limited range of single-frame tasks such as 2D object detection and semantic segmentation.

As such, more modern datasets are designed to include multiple cameras, LiDAR and ground-truth types. Prominent examples include KITTI (Geiger et al., 2013), SemanticKITTI (Behley et al., 2019) , KITTI-360 (Liao et al., 2023), Argoverse (Chang et al., 2019), Apolloscape (Xinyu Huang et al., 2018), A2D2 (J. Geyer et al., 2020), A*3D (Pham et al., 2020), WOD (P. Sun et al., 2020) and Lyft (Houston et al., 2021), while nuScenes (Caesar et al., 2020) and Aduulm-360 (Schön et al., 2024) further include radar. This set offers a mix between pixel-wise semantic and instance annotations, 3D point-wise annotations, 3D bounding boxes and track IDs, being able to support the training of more complex tasks such as 3D object detection and tracking, 3D semantic segmentation and multi-sensor fusion.

While some of these datsets have been designed to capture diverse areas, weathers and illumination conditions, most of them are biased towards daytime overcast conditions, making them unsuitable for tasks that are supposed to perform in any weather condition, at any time. As such a lot of attention has been devoted towards capturing datasets that focus mostly on physical degradations due to fog, rain, snow and night-time. For this case, the group of datasets primarily dedicated to providing 2D annotations consists in IDD-AW (Shaik et al., 2024), DAWN (Kenk & Hassaballah, 2020), ACDC (Sakaridis et al., 2021), Brno (Ligocki et al., 2020), RaidaR (Jin et al., 2021), Dark Zurich (Sakaridis et al., 2019), NightOwls (Neumann et al., 2018) and NightCity (Tan et al., 2021), while a mix between 2D and 3D ground-truth is provided in DENSE (Bijelic et al., 2020), Ithaca365 (Diaz-Ruiz et al., 2022), WADS (Kurup & Bos, 2023), SemanticSpray (Piroli et al., 2023) and CADC (Pitropov et al., 2021). Adverse weathers have also been captured in Boreas (Burnett et al., 2023), OORD (Gadd et al., 2024), SID (El-Shair et al., 2024) and RADIATE (Sheeny et al., 2021), however ground-truth in these datasets is limited.

Of particular interest are Ithaca365 (Diaz-Ruiz et al., 2022), which provides amodal segmentation, capturing the full extent of the object despite occlusions, WADS (Kurup & Bos, 2023), which provides point-wise annotations for falling snow and accumulated snow particles, and SemanticSpray (Piroli et al., 2023), which focuses on the effects of vehicle-generated water spray and provides point-wise annotations for LiDAR points belonging to static elements, vehicles, water droplets and mist. Having fine labels that differentiate weather artifacts from actual scene content enables the development

of data-driven approaches that explicitly learn the characteristics of weather effects, de-weathering models and weather-induced outlier detection.

While some datasets, such as ACDC, try to capture the same scene in multiple conditions, it is impossible to guarantee pixel-level alignment between pairs, especially in urban complex datasets. As such, in Rainy Screens, Porav et al. (2020) propose to re-capture any existing dataset using a physical adherent rain droplet rig, while in Desoiling Dataset (Uricar et al., 2019), adherent mud and soil is added to 3 cameras while one is kept clear.

Additionally, frameworks such as Robo3D (Kong et al., 2023) offer the ability to apply multiple analytical weather models (e.g. fog, snowfall, rainfall) in the LiDAR point space to existing well-established datasets such as NuScenes, Waymo or A2D2.

While domain adaptation is a valuable tool for transforming existing datasets, some edge cases cannot be captured in the real-world, due to having a low probability of occurrence or a high risk profile. As such, traditional 2D and 3D simulators still play an important role in improving data coverage.

3.3 Simulators

To circumvent the limitations associated with data capture and manual annotation, methodologies that exploit simulated environments have been introduced, leveraging the ability to more easily extract all types of ground-truth that 2D or 3D simulator engines can offer. The most popular open-source AV-oriented simulators include Carla (Dosovitskiy et al., 2017) and AirSim (Shah et al., 2018), both built on Unreal Engine.

Their rise in popularity has been largely driven by a number of additional advantages associated with simulators, including the ability to capture dangerous scenarios that would otherwise be ethically infeasible to obtain, and the capacity for a large number of variations of corners cases through the manipulation of environment variables such as lighting, traffic density, agent trajectories, scene configurations or weather conditions. Similarly, scenarios can be replayed with full reproducibility, and novel sensors such as event cameras can be used to develop new perception algorithms without requiring access to expensive hardware.

In addition to standalone AV-oriented simulators, various synthetic datasets have been built around established 3D engines or 3D games. For example, S. Richter et al. (2016) propose the extraction of RGB images and corresponding semantic segmentation from Grand Theft Auto (RockstarNorth, 2015),

while in SYNTHIA (Ros et al., 2016), RGB images, semantic segmentation, and depth maps are rendered for street scenarios under diverse weather conditions. In Synscapes (Wrenninge & Unger, 2018), the authors focus on the controllability of scenario-generation in order to ensure a diverse combination of assets, placements, and types of appearance, and later Hahner et al. (2019) add synthetic fog.

MatrixCity (Y. Li et al., 2023) offers scenes under different simulated weather conditions and levels of illumination, along with varying levels of traffic participants and ground-truth that includes semantic segmentation, depth and surface normals. Urbansyn (Gómez et al., 2025) is later proposed as a photo-realistic alternative to simulator-derived data from S. Richter et al. (2016), while Wrenninge & Unger (2018) use OpenStreetMap for generating realistic urban layouts (OpenStreetMap contributors, 2004). Testolina et al. (2023) introduce SELMA, which is an extensive CARLA-based dataset comprising of a mobile rig with 24 diverse sensors capturing scenes under 27 environmental conditions, complete with semantic annotations for camera and LiDAR, representing 36 classes following the Cityscapes labeling standard. The Carla-derived dataset IDDA (Alberti et al., 2020) is proposed to deliver a wide range of weather conditions, while SHIFT (T. Sun et al., 2022) provides both discrete and continuous shifts in domain, to support the development of models for continuous domain adaptation.

Although 3D simulators offer high-quality ground-truth and virtually infinite variations, creating the virtual environments involves generating assets, designing scenarios, determining object paths, and managing appearance factors such as colors, textures, and lighting. The most significant drawback, however, is the gap between simulation and reality, making simulators suboptimal for data augmentation. J. Ge et al. (2025) address these limitations by proposing a data-driven AV simulator that employs gaussian splatting, intended for assessing end-to-end autonomous driving models.

Finally, Mehr & Eskandarian (2025) propose SimBEV, which is a comprehensive Carla-derived dataset, consisting in multiple sensor modalities, ground-truth and accurate BEV representations, facilitating the advancement of BEV segmentation and sensor fusion models.

3.4 Data synthesis

3.4.1 Image synthesis

Images are one of the most common sources of information about the environment, can be acquired cheaply using cameras, and are commonly used as input by numerous machine learning tasks such as object detection, semantic and instance segmentation, depth estimation, place recognition, visual

odometry and SLAM, trajectory prediction and affordance estimation. As such, image synthesis is an important capability for ensuring the highest possible coverage across visual information.

Classical image synthesis methods relied mostly on non-parametric models that do not make assumptions about the data distribution or the mapping between inputs and outputs. As a result, no network / model is trained when generating data and such approaches do not learn an internal representation of the data. Furthermore, they rely on either user input or externally stored data at inference time, such as memory banks, from which data is retrieved, and employ similarity-based algorithms such as search and patch matching for retrieval and interpolation for blending. Since they directly reference memory banks, the output data is generally highly photo-realistic, since pixels or patches are copied from real data, allowing for preservation of fine-grained details and textures. Although this aspect provides the greatest advantage, it simultaneously presents the most significant drawback, namely, limited capacity for generalisation i.e. the ability to generate data that is not present in the training set / reference set. While new data can be generated by interpolating existing data, swapping or re-shuffling and blending image patches, this generalisation is limited to the reference dataset. Notable works include approaches such as Sketch2Photo (T. Chen et al., 2009), CG2Real (M. K. Johnson et al., 2011), PhotoClipArt (Lalonde et al., 2007) and PatchMatch (Barnes et al., 2023).

With the rise in popularity of neural networks and CNNs, the focus has shifted towards parametric models, which are trained end-to-end and learn an internal representation of the data in their weights. This was especially due to the ability of Deep CNNs to learn and extract rich features, textures and shapes (Simonyan & Zisserman, 2015). As such, given enough diverse datasets on which to be trained, such models have a great generalisation power and the ability to synthesise true novel content. On the other hand, since these models learn an approximation of the data distribution, this can lead to loss of fine-details, bluriness and over-smoothing in their outputs. Additionally, they can suffer from mode collapse, being unable to represent long-tail and complex variations. Notable approaches include Pix2pixHD (T. Wang et al., 2018), CycleGAN (Zhu et al., 2017), StyleGAN (Karras et al., 2019), StyleGAN2 (Karras et al., 2020), SPADE (Park et al., 2019) and OASIS (Sushko et al., 2021).

Semi-parametric models integrate in a synergistic approach both a parametric component with learned weights, and a non-parametric component such as an external memory bank. In this setup, the generalisation power of the network combined with the flexibility and specificity of the retrieval process and external memory can ensure high-fidelity and novel data generation, as opposed to parametric models, where generalisation capability is a function of the trained weights, or non-parametric models,

where generalisation can only come from accessing other external memory banks. Recent works include SIMS (Qi et al., 2018), which explores a combination between a non-parametric component (retrieval from a bank of objects) and a learned, parametric component (in-painting encoder-decoder) showing realistic image generation, Depth-SIMS (Musat et al., 2022), which additionally generates depth and semantic segmentation ground-truth, while Blattmann et al. (2022) condition a diffusion model with exemplars retrieved from a memory bank. Flexibility is however limited if the method employs heuristics that break down under a domain shift.

However, fully parametric approaches have become widely adopted, and have since surpassed the performance of most previous methods, whether non-parametric or semi-parametric. The introduction of unconditional GANs by Goodfellow et al. (2014) marked a pivotal moment, as they took a game-theoretic approach with a generator and a discriminator being trained simultaneously, leading to the beginnings of unprecedented realism in the field of image synthesis. While the literature is vast, the following sections focus on only a few notable GAN models which are suitable for AV data.

In the unconditional setting, where image synthesis starts from a noisy latent code, Karras et al. (2018) propose to train on low-resolution images, then gradually increase the resolution by adding layers to both the generator and discriminator, in order to reduce training instability observed previously when training on high-resolution images, leading to results that improve image diversity and sharpness at a resolution of up to 1024×1024.

As opposed to unconditional GANs, conditional GANs are more practical and task-specific, as they can provide explicit control over the synthesis process via conditional input. Pix2pix (Isola et al., 2017) explores conditioning GANs using semantic maps, sketches, edges and Google Maps data. Subsequently, T. Wang et al. (2018) propose Pix2pixHD, a multi-scale generator and discriminator, with a feature matching loss and a VGG perceptual loss among others, to synthesise high-resolution images of up to 2048×1024 pixels. As models evolved, more types of conditioning data were used as input, apart form the original random noise vectors. This has led to sub-domains such as image-to-image synthesis, where guidance can come from semantic segmentation maps, depth maps, layout, sketch and edge maps or other RGB images, text-to-image synthesis where the input is a simple caption describing the scene contents, or other types of applications with inputs such as keypoint poses.

Nevertheless, a significant drawback of conditional GANs is the necessity for a curated or annotated dataset to be trained on. Regardless of their conditioning type - be it captions, embeddings, or 2D

data such as segmentation, instance or depth maps - it needs to be paired or associated with the desired output, for example RGB images.

CycleGAN (Zhu et al., 2017) was introduced to overcome the main limitation of conditional GANs – the requirement of paired training data – essentially enabling the domain of **unpaired image translation** and paving the road for **domain adaptation** via style transfer. The novelty consists in employing 2 conditional GANs together with a cycle-consistency loss, with a first GAN translating the original input into a target domain, and the second GAN translating the output of the first back into the original domain. Relaxing the strict requirement of paired data meant that a wider set of data sources could be used for training, thereby enabling a more diverse range of applications. For instance, although acquiring RGB-sketch pairs is straightforward, capturing corresponding daytime overcast images and nighttime clear images in an urban outdoors setting is unfeasible, since it is impossible to ensure that agents and poses remain unchanged.

More recently and due to its flexibility in describing visual concepts and relative ease of use, conditioning using natural language prompts (text) has become a key focus in image synthesis, as language can be used to describe a broad spectrum of details that may not be easily conveyed through alternative types of conditioning. However, language is also subject to variability in interpretation, which can lead to ambiguity in the generated data. As such, while generated images can display a large amount of diversity and richness of detail, the end results are oftentimes misaligned with the conditioning prompt.

Notable contemporary text-to-image models range from those that use a GAN backbone such as StackGAN++ (H. Zhang et al., 2017), to transformer-based approaches such as DALLE (Ramesh et al., 2021) and more recently, diffusion models such as Stable Diffusion (Rombach et al., 2021). Additionally, various approaches have focused on enabling domain adaptation through fine-tuning using small amounts of data, in comparison to the sizes of the datasets used to train the models initially, such as DreamBooth (Ruiz et al., 2023) or LoRA (E. J. Hu et al., 2022).

Finally, methods that enable further conditioning of text-to-image models using multiple additional modalities are highly relevant to the domain of data for autonomous driving, allowing for much finer control over the content placement and structure of the generated images, while maintaining the ability to control stylistic elements using natural language. A commonly used approach is ControlNet (L. Zhang et al., 2023), which adds the ability to condition the Stable Diffusion group of models using semantic segmentation and instance maps, depth maps, normal maps, canny edge maps or human pose keypoints by re-using the encoder portion of Stable Diffusion's UNet backbone as an additional

encoder. T2I-Adapter (Mou et al., 2024) achieves comparable results, but makes use of an encoder with a much smaller number of parameters, allowing for faster inference times. Notably, both approaches allow multiple independent encoders to be used together in parallel at inference time: for example, separate encoders can be trained, one for conditioning on segmentation maps and one on depth maps, but at runtime the two can be composed to enable enhanced controllability. These approaches have significantly narrowed the domain gap with respect to real-world data, and their ability to produce a diverse set of image styles has made them an essential source for data augmentation.

3.4.2 3D aware synthesis and NeRF

Models that only reason in 2D are a suboptimal approach for a whole range of common operations, as images are a lossy representation of the world since they are the result of a projection from 3D to 2D. Relighting, pose or view changes or object manipulation are difficult as the depth dimension is lost. Hence, various 3D representations have started being employed in image synthesis models to improve control.

Most 3D aware image synthesis models, which attempt to generate images that follow the underlying 3D structure of the scene, use various inductive biases to enforce this. Architectural biases are common, such as disentangling latent representations so that shape and appearance / texture are represented by separate latent vectors, or conditioning models on camera poses. Similarly, representation biases control how the 3D structure is encoded internally: implicitly, where 3D information is modeled as a continuous function such as MLPs in NeRFs, or explicitly via voxels, meshes or point clouds.

Several generative approaches explored mechanisms that rely on 3D inductive biases, such as PlatonicGAN (Henzler et al., 2019), which learns to lift an image into a 3D volume and renders novel views using several types of differentiable renderers, or HoloGAN (Nguyen-Phuoc et al., 2019), which disentangles pose from object identity (shape) without any explicit 3D supervision.

In 3D reconstruction, Mildenhall et al. (2020) propose to represent a scene via a fully-connected neural network, which takes as input a spatial location and viewing direction, and outputs a density and an RGB-color. The neural network is queried at coordinates of points along rays corresponding to a pinhole camera model with a known pose, and a volume renderer is used to then project all of the rays into a final image. While they present both high-quality synthesised views and fine-detailed depth maps, the base method is restricted to a single static scene for each trained network – since the scene is learned in the network weights, once a NeRF model is trained on a scene it cannot generalize to a different scene – and it is inherently slow for both training and inference. With this in mind, pixelNeRF (A. Yu et al.,

2021) enables NeRF to share prior knowledge between scenes, by conditioning the model on spatial image features, NeRF in the Wild (Martin-Brualla et al., 2021) tackles dynamic scenes by handling static and transient objects separately, while iNGP (Müller et al., 2022) improves the speed of training by relying on a multi-resolution hash table, and kiloNeRF (Reiser et al., 2021) improves the speed of rendering new views, by discretising the scene representation across many smaller neural networks.

With a focus on data for autonomous driving, NeuRAD (Tonderski et al., 2024) tackles novel view synthesis for scenes with dynamic actors such as vehicles and pedestrians by learning separate NeRF representations for the static part of the scene and for the dynamic parts, while also incorporating sensor-specific effects like rolling shutter and LiDAR beam divergence. Besides generating RGB images and depth, it also synthesizes LiDAR returns, including intensities and ray drop probablities, and allows manipulation of actors in the scene by changing the location of their bounding box coordinates. In SplatAD, Hess et al. (2025) offer the same functionality but use gaussian splatting (Kerbl et al., 2023), leading to significantly faster rendering compared to NeRF.

NeRF-based metods, along with volume rendering have also become the backbone for 3D aware generative models, with GRAF (Schwarz et al., 2020) demonstrating consistent novel view generation for single objects or human faces, EG3D (Chan et al., 2022) combining a StyleGAN generator (Karras et al., 2020) with a neural renderer and a StyleGAN discriminator to generate novel views of faces and animals, while GSN (DeVries et al., 2021) and GAUDI (Bautista et al., 2022) use a conditional or unconditional encoder to output a 2D floorplan or tri-plane representation respectively, that is sampled using points along rays and finally rendered into an image and a depth map using a volume renderer.

Kim et al. (2023) propose NeuralField-LDM, which is a 3-stage generative pipeline for synthesizing photorealistic 3D scenes, where the first stage learns to lift posed images into a 3D representation, the second stage learns to compress this representation into a set of 1D, 2D (representing a BEV) and 3D latent embeddings, while the third stage learns to sample these embeddings, either unconditionally or conditional on a semantic BEV. At inference time, the third stage is used to sample latent embeddings, the second stage decodes these embeddings into a 3D representation, and the first stage uses this 3D representation to render novel views.

K. Yang et al. (2023) propose BEVControl, where sketch-style BEV layouts along with natural language prompts are used to generate street view RGB images with multi-view consistency. The method lifts the BEV sketch along with bounding boxes into 3D and projects it into each camera to obtain foreground and background conditioning inputs, which are then used to generate a set of

geometry-consistent and appearance-consistent features that are decoded into images. However, the depth information is lost during the 2D projection step, making it harder to reason about occlusions or scale. Similarly, BEVGen (Swerdlow et al., 2024) uses semantic BEV layouts as inputs to an autoregressive transformer to generate multi-view images, but it relies on an implicit spatial attention mechanism rather than on any explicit 3D representations.

MagicDRIVE (Gao et al., 2024b) uses a diffusion model conditioned on semantic BEVs, object bounding boxes, camera pose information and natural language prompts to generate street-view images with multi-view consistency and weather conditions. MagicDRIVE3D (Gao et al., 2024a) builds on MagicDRIVE, splitting the street-view synthesis task into 2 steps. Firstly, by extending the conditioning with a temporal sequence of camera poses, the method generates conditional multi-view videos. Next, deformable gaussian splatting is used to generate a 3D reconstruction of the scene using the generated video frames, allowing for novel-view synthesis from different camera poses.

In contrast to methods that mainly output RGB images, Nunes et al. (2025) propose 3DiSS, which focuses on generating 3D semantic data without additionally producing or relying on images. The method first trains a VAE to encode 3D data into latent representations, then trains a diffusion model to sample these embeddings. At inference time, novel dense 3D scenes can be generated, with each point being assigned a semantic class.

3.4.3 Video synthesis

The ability to generate realistic videos without the necessity of manually modeling materials, lighting, weather conditions, scene geometry, and dynamics saves a significant amount of effort and time that would otherwise be required for manual generation and rendering.

Similar to image synthesis, video synthesis aims to generate sequences of images (or frames), typically representing some degree of movement within a static or dynamic scene. This requires a high degree of correlation between consecutive frames, both in terms of stylistic consistency but also with respect to the physical plausibility of movements and interactions between objects.

Earlier approaches such as video-to-video (T.-C. Wang et al., 2018; Mallya et al., 2020) showcased transforming the style of existing videos, while Tulyakov et al. (2018) and Clark et al. (2019) demonstrated video generation from scratch, although at lower resolutions and often with significant artefacts. More recently, text-to-image model backbones have been adapted for video generation by Blattmann et al. (2023), X. Wang et al. (2023), and Bar-Tal et al. (2024) for the case of generation from

scratch, or M. Geyer et al. (2023) for transforming existing videos. Similarly, multi-modal conditioning adapters such as Ctrl-Adapter (H. Lin et al., 2025) can transform sequences of conditioning frames such as semantic segmentation into videos with good stylistical consistency, while Kondratyuk et al. (2024) model the task of video generation as a sequence of tokens obtained from multi-modal inputs and predicted by a large language model.

However, these methods lack explicit mechanisms for 3D understanding, relying instead on learning plausible composition, generation, and movement from the training data, thus leading to artifacts and temporal inconsistency in longer videos.

3.5 Adverse weather synthesis and weather effect removal

The domain of weather synthesis and removal is vast, spanning analytical models and conventional rendering using 3D engines, data driven approaches, and physical data augmentation. Therefore, this section highlights a limited yet relevant part of this domain, with a stronger focus on weather synthesis, while briefly describing approaches that disentangle the effect of weather from scenes – a topic explored further in Section 8.2.

In general, weather synthesis methods can be grouped into three categories: analytical – which use a set of formulas to represent or approximate the physics of weather effects, data-driven – which try to learn the distribution of data, and hybrid – which combine the previous two.

Analytical methods rely on physical principles such as light scattering or particle attenuation models, and can offer high realism and physical accuracy. While no training or reference data is needed, the methods are often difficult to develop and computationally intensive and may not generalize well to all real-world variations, which may contain complex interactions. Due to this, these approaches are less commonly used today.

On the other hand, data-driven methods aim to synthesize weather conditions by learning from data affected by weather, and are typically framed as either a synthesis or a domain adaptation task. Both strongly and weakly supervised approaches have been investigated, with applicability to various weather conditions provided that adequate training data is available. However, the primary limitation of such approaches is that, depending on formulation, physically meaningful parameters (e.g. millimeters of rain per minute) are often difficult to control.

Hybrid approaches typically integrate an analytical model with a data-driven approach that serves to parameterise the analytical model, leveraging the ability to learn complex interactions or weather patterns and the ability to retain control of physically meaningful parameters. Alternatively, a data-driven approach can be used to provide a representation of a scene that makes reasoning about weather effects easier, for example, lifting images into a 3D representation makes it possible to approximate water and snow accumulation or depth-based degradation of visibility.

3.5.1 Image

A very common approach that best approximates real weather is to attempt to replicate the processes that cause a particular type of weather within a controlled environment. Examples include the DENSE (Bijelic et al., 2020) dataset, which provides images and associated ground-truth captured inside a chamber where the level of illumination, fog or rain can be tightly controlled, but also datasets proposed by Uricar et al. (2019), Qian et al. (2018), and Porav et al. (2020) which simulate weather or soiling adherent to a camera lens. The drawback of these methods however is that they are both resource intensive, time consuming, expensive and also cannot scale to generate very large and diverse datasets required for training large models.

A second class of approaches makes use of analytical models of weather to augment existing datasets or to synthesize data that follows the distribution of real world weather data. Such methods range from simple rain streak generation as in W. Yang et al. (2017), H. Wang et al. (2021), and Tremblay et al. (2020), to fog generation based on a model of light transmittance as in Sakaridis et al. (2018), Hahner et al. (2019), and Yiming Xie et al. (2024) or snow flake generation as in Y.-F. Liu et al. (2018).

Data-driven generative models have also been employed to augment existing clean images, such as Cycle-GAN-based domain adaptation models to add rain and snow (Porav et al., 2018), conditional image translation and a simple analytical model to add rain and rain streaks (Jeon et al., 2025) or unpaired image translation techniques to add fog, snow or wet surfaces (Rothmeier & Huber, 2021).

3.5.2 Lidar

For the case of LiDAR weather effects, Robo3D (Kong et al., 2023) offers a framework for augmenting existing laser data with fog, snow, or distortions while also simulating the effect of wet ground on LiDAR attentuation. In Hahner et al. (2021), the authors implement a physically valid simulation

for fog, while Teufel et al. (2022) explores analytical models for rain, snow and fog. In D. Yang et al. (2024), the CARLA simulator is used to develop an analytical model for the effect of water splashes and water spray on LiDAR returns, with the resulting augmented data used to improve the robustness of a real-world 3D object detector. Finally, Lee et al., 2022 propose a modified Cycle-GAN architecture in order to transform range and intensity images obtained from LiDAR data, for adding rain and fog effects, with the results being benchmarked against a dataset of rain and fog data collected under laboratory conditions.

3.5.3 NeRF-based

A special class of approaches is represented by models that learn to both reconstruct a scene and simultaneously eliminate the weather effects that degrade it. In ScatterNeRF (Ramazzina et al., 2023) and DehazeNeRF (W.-T. Chen et al., 2024) the authors use an explicit model of how fog affects light by scattering and implictly parametrise it using an additional MLP, while in WaterNeRF (Sethuraman et al., 2023) and SeaThruNeRF (Levy et al., 2023), additional models estimate coefficients for backscattering and light attenuation effects. In DerainNeRF (Yunhao Li et al., 2024), a mask indicating the location of adherent droplets is used to generate a clean reconstruction of the scene from collections of images affected by droplets.

In ClimateNeRF (Yuan Li et al., 2023), a pre-trained NeRFs is combined with physical simulation to render accumulated snow, smog and water flooding, while ClimateGS (Yuezhen Xie et al., 2025) adopts a very similar approach using gaussian splatting as a replacement for NeRF, and RainyGS (Dai et al., 2025) augments scenes with rain streaks and puddles by combining analytical methods with a height map obtained from a gaussian splatting representation.

Although Clean-NeRF (X. Liu et al., 2023) is not specifically designed to tackle weather effects, it makes use of a mechanism for manipulating the density peaks along rays, which can have valid applications for weather synthesis or weather removal. The drawback of contemporary NeRF or gaussian splatting-based methods is that they are overfit to a particular scene and cannot be applied to a new scene without re-training. However, the same ideas used here can be repurposed to train a more generalizable model - the final experiments proposed in Section 8.2 in this document offer more details on how this could be achieved.

3.6 Training and validating with synthetic data

For modular self-driving stacks which employ sequential perceptual, predictive, planning, and control modules, a perception module will typically provide information about the state of the world, often in the form of object bounding boxes, semantic segmentation, depth maps, occupancy grids and so on. A more accurate understanding of the world surrounding the ego-vehicle — such as reduced perception errors, more accurate detection of small objects, or more temporally consistent outputs – typically results in a reduction of unsafe or overly cautious decisions taken by downstream tasks such as planners, which consume the output of perception. The quality of the outputs of such downstream tasks is often a direct function of the quality of perception, since no other component can provide up-to-date information about the world, especially with respect to dynamic actors or obstacles.

Similarly, for end-to-end or pixel-to-pedal approaches, which map raw sensor inputs directly to vehicle control signals, training on extended or more diverse datasets has the same effect of increasing both the performance of the model and the robustness to edge cases. For example, in Goel et al. (2024), synthetic data is used to improve two end-to-end models in terms of metrics such as route completion, infraction, and driving score, leading to an improvement in driving performance of up to 20% across a diverse range of weather and illumination conditions.

The work undertaken in this manuscript is part of a broader category of approaches that focus on the generation of synthetic data for the purpose of training modular downstream tasks – such as perception, localisation or planning - with this generated data.

Historically, throughout the stages of generative model exploration, approaches were aimed primarily at enhancing image realism, visual quality, and ensuring a wide range of diversity. Subsequently, a stronger emphasis was placed on enforcing structure and semantic control, in order to facilitate label-consistent synthesis. For example, the auxiliary classifier GAN was built upon the classical conditional GAN framework by incorporating a discriminator with dual purpose - to predict both the source of the input (real or fake) and its class probability (Odena et al., 2017). As such, the generator learned to output images that are not only visually plausible, but also comply with their semantic constraints. Later, more advanced architectures were proposed, such as OASIS (Sushko et al., 2021), which outputs a pixel-wise semantic segmentation map, instead of a single class label.

In the meantime, Porav et al. (2018) and Anoosheh et al. (2019) laid the groundwork for using image synthesis as a source of training and validation data for AV downstream tasks in the context of visual localisation under varying weather and illumination conditions, while Xu et al. (2021) and

Teeti et al. (2022) later explored object detection. This established a new direction, as synthesized or domain-adapted data must not only possess high visual quality but should also improve the performance of the downstream tasks that are trained and validated on it. Bai et al. (2024) further evaluate the difference in performance between models trained on simulation-derived data and those trained on real data and further adapt depth estimation and odometry models to improve their effectiveness. Goel et al. (2024) use ControlNet (L. Zhang et al., 2023) to generate Waymo (P. Sun et al., 2020) and CARLA-derived scenes in rare conditions such as night-time or night-time rain with lens glare, using the data to improve Mask2Former (Cheng et al., 2022) and SegFormer (E. Xie et al., 2021) predictions. Rothmeier et al. (2024) use ChaptGPT (OpenAI, 2024) to craft 300 textual narratives depicting driving scenes and further employ Midjourney (Midjourney, 2024) to generate images of underrepresented weather conditions such as rain, snow and fog, obtaining $18,000$ images. After providing labels, they fine-tune and validate FasterRCNN (Ren et al., 2015), FCOS (Tian et al., 2019), RetinaNet (T.-Y. Lin et al., 2017) and SSD (W. Liu et al., 2016) and further show improvements in their performance as opposed to only training on real data.

The robustness of simulator-derived data for 2D and 3D object detection is further tested by Özeren & Bhowmick (2025), when training in 3 configurations: with real data only, with simulator-derived data only and on a combination of both. After assessing the generalisation capabilities of detectors by testing them on out of sample data, it is concluded that training on a combination between real and synthetic data outperforms training on a single source alone.

An extensive study by Silva et al. (2025) explores the impact of diverse rendering parameters in simulator-derived data on the performance of semantic segmentation models. They vary factors such as rendering noise, color transformations and material realism in the UrbanSyn simulator (Gómez et al., 2025) and further train DeepLabv3+ (L.-C. Chen et al., 2018) and SegFormer (E. Xie et al., 2021) on the generated data. Experiments conducted on real-world datasets like Cityscapes (Cordts et al., 2016), BDD100K (F. Yu et al., 2020), and Mapillary Vistas (Neuhold et al., 2017) reveal that using simple materials leads to a drop in model accuracy, while employing suitable color transfer methods can narrow the domain gap, increasing the model's generalisation power, and furthermore training on a combination of both noisy and denoised images can improve robustness.

As opposed to studies so far that have focused on improving tasks from the ego-view, MagicDRIVE (Gao et al., 2024b) uses generated data to augment training for both a BEV segmentation and a 3D object detection model, showing improved performance on BEVFusion (Z. Liu et al., 2023) and

CVT (B. Zhou & Krähenbühl, 2022), while in BEVControl (K. Yang et al., 2023), the authors train Bevformer (Z. Li et al., 2025) on synthetic data, showing improved results. Similarly, to evaluate the quality of generated data in 3DiSS (Nunes et al., 2025), a 3D semantic segmentation model is trained with varying mixtures of real and synthetic data, demonstrating that the inclusion of generated data leads to significant improvements in performance.

Additionally, Klinghoffer et al. (2023) argue that variability in camera rigs resulting from diverse sensors and mounting configurations across different vehicle platforms can lead to distribution shifts which in turn affect the accuracy of downstream tasks. Since data collection and annotation for every possible rig and configuration is an impractical avenue, they instead improve the performance of downstream models such as LSS (Philion & Fidler, 2020) and CVT (B. Zhou & Krähenbühl, 2022) using an augmented dataset generated using novel-view synthesis.

An alternative approach to training downstream tasks is to train an adapter that converts raw sensor data into a representation that maximises the performance of downstream tasks, such as in Porav et al. (2019) or Clement & Kelly (2017), but this category of methods also relies on diverse, multi-modal data to train the adapter.

Finally, while all the data generated using the methods described so far can be used for training models in a strongly-supervised fashion, data from NeuralField-LDM (Kim et al., 2023), NeuRAD (Tonderski et al., 2024) or SplatAD (Hess et al., 2025), among others, can similarly be used for training self-supervised tasks such as novel view synthesis, simultaneous localisation and mapping or 3D reconstruction from 2D images.

4

Multi-weather city

Contents

4.1 Contribution

This paper contributes to the field of perception for autonomous driving by addressing the challenges introduced by adverse weather conditions, especially when multiple types of weather may be present. The key contributions of the paper include:

1. The development of a comprehensive weather dataset that includes a diverse set of adverse weather conditions such as night-time, rain, snow, fog and their combinations, designed to test and improve the robustness of autonomous driving systems. Beginning with a dataset that

includes ground-truth annotations in the form of semantic segmentation and bounding boxes, we create 7 different weather-degraded versions. Additionally, we suggest an extension to enhance the generated conditions by incorporating an analytical fog model from existing literature, leading to a total of 14 adverse weather conditions, all based on a single set of ground-truth annotations;

2. The introduction of a novel method for stacking multiple adverse weather models, allowing for more realistic and challenging testing scenarios for the perception modules. This stacking is advantageous as it combines data-driven model effects from individual weather conditions to generate composite weathers;

3. The demonstration of the effectiveness of the proposed dataset and methods in terms of both perceptual quality and also through extensive perception experiments on multiple in- and out-of-domain real-world datasets. Our results demonstrate enhanced performance and robustness in object detection and instance semantic segmentation when trained with multiple synthetic weathers, in some cases leading to an increase in mean AP of more than 10 percentage points;

4. We additionally provide the resources and instructions required to reconstruct the multi-weather dataset, which is derived from the original Cityscapes dataset (Cordts et al., 2016).

4.2 Integrated manuscript

The manuscript was published at the 2nd Autonomous Vehicle Vision (AVVision) Workshop, International Conference on Computer Vision (ICCV) 2021, (Musat et al., 2021)

Multi-weather city: Adverse weather stacking for autonomous driving

Valentina Muşat* Ivan Fursa† Paul Newman* Fabio Cuzzolin‡ Andrew Bradley†

valentina@robots.ox.ac.uk,17076662@brookes.ac.uk,pnewman@robots.ox.ac.uk,fabio.cuzzolin@brookes.ac.uk,abradley@brookes.ac.uk

Abstract

Autonomous vehicles make use of sensors to perceive the world around them, with heavy reliance on vision-based sensors such as RGB cameras. Unfortunately, since these sensors are affected by adverse weather, perception pipelines require extensive training on visual data under harsh conditions in order to improve the robustness of downstream tasks - data that is difficult and expensive to acquire. Based on GAN and CycleGAN architectures, we propose an overall (modular) architecture for constructing datasets, which allows one to add, swap out and combine components in order to generate images with diverse weather conditions. Starting from a single dataset with ground-truth, we generate 7 versions of the same data in diverse weather, and propose an extension to augment the generated conditions, thus resulting in a total of 14 adverse weather conditions, requiring a single ground truth. We test the quality of the generated conditions both in terms of perceptual quality and suitability for training downstream tasks, using real world, out-of-distribution adverse weather extracted from various datasets. We show improvements in both object detection and instance segmentation across all conditions, in many cases exceeding 10 percentage points increase in AP, and provide the materials and instructions needed to re-construct the multi-weather dataset, based upon the original Cityscapes dataset.

1. Introduction

Autonomous vehicles rely on a set of sensory information in order to correctly perceive the environment and ensure a safe journey. Unfortunately, adverse weather and lighting conditions can affect how the environment is perceived, thus impacting the performance of downstream tasks and, ultimately, the safety of the traffic participants. Cameras, which are one of the most cost-effective modalities in autonomous vehicles, are also among the most affected by adverse weather and illumination conditions [52], with matters made worse by the overlap with some of the causes of LIDAR performance degradation [17].

*Oxford Robotics Institute, University of Oxford
†Autonomous Driving Group, Oxford Brookes University
‡Visual Artificial Intelligence Laboratory, Oxford Brookes University

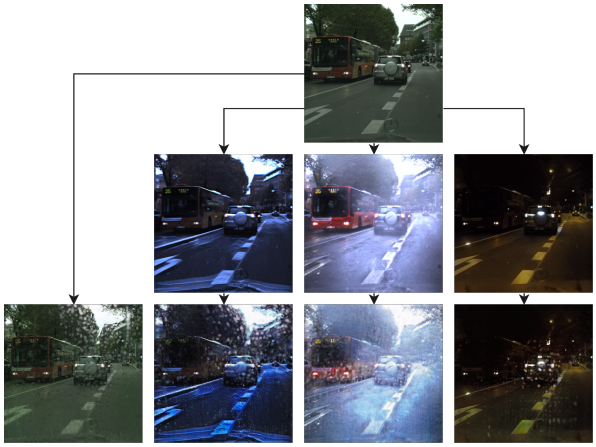


Figure 1: Concept of weather stacking: generated weather appearance, starting from real overcast.

Due to the increase in popularity of the autonomous driving industry, a lot of effort has been devoted to tackling these issues. While hardware solutions are being developed using the latest technology in order to ensure more robustness to adverse weather at the data acquisition stage [5], a large body of research focuses on improving the robustness of downstream tasks via domain adaptation, de-noising, de-weathering and sensor fusion, amongst others.

Unfortunately, both the aforementioned methods and the relevant AV-related tasks (such as semantic segmentation, object detection and depth estimation) require large training datasets, both in general and in each of the specific weather conditions the vehicle might encounter. Data availability, however, has become a serious bottleneck due to the cost, time and difficulty of obtaining it. To overcome this issue, significant work has recently been directed at the synthetic generation of weather conditions [13, 38] and the photo-realistic style-transfer of weather appearance [34, 33, 32, 2, 42]. For the purpose of testing downstream tasks in the wild, datasets have been designed to include scenes with diverse weather and the corresponding ground truth [45, 21]. Others have attempted to provide both clear weather and weather condition pairs for both static [10] and dynamic scenes [33, 35].

While these approaches are commonly benchmarked in isolation, rather than in combination, here we aim to show that combining these techniques can yield much more visu-

ally diverse outputs in a controlled and stackable way. In this work, in particular, we generate augmented imagery under 7 distinct various weather and illumination settings starting from a *single* dataset with ground truth (Cityscapes [8]), and test if the generated data is a good proxy for real weather. We do this by using the generated data as training data in the context of autonomous driving-related downstream tasks. As an extension, we propose as future work 7 other conditions based on the work of [38], and further present visual results.

The contributions of the paper are as follows:

1. We generate a number of weather conditions using a unified generator architecture for image translation, for both paired and unpaired settings, based on the work of [30] and [54], which results in imagery that not only is of increased realism and has fewer
2. We use the above-generated weather appearance as input to an additional network designed to add adherent droplets, thus resulting in a combination of more diverse weather appearances, again starting from only a single dataset with paired ground truth.
3. For a more extensive evaluation, we use multiple publicly available datasets comprising real adverse weather for validating the suitability of the data for instance segmentation and object detection, while also evaluating the quality of the images using the Inception Score and Fréchet Inception Distance.
4. We release the relevant materials and steps needed to recreate and use the multi-weather Cityscapes dataset, which can be found at https://github.com/vnmusat/multi-weather-city.
   Due to licensing restrictions, the dataset itself is distributed as a set of additive transformations that can be applied to the original Cityscapes dataset [8].

We would like to stress the facts that the purpose of this study is not to present an entirely novel image-to-image translation architecture, but to demonstrate a methodology for creating diverse data by starting from a single dataset with paired ground truth, using cascaded image translation models.

2. Related work

Adverse weather can affect the performance of computer vision tasks in multiple ways: temperature and temperature variations affect the optical, electronic and mechanical components used in capturing visual data, while ambient conditions affect light propagation and the appearance of the environment [7]. For example, cold temperatures or foggy conditions can result in condensation on the lens, blurring the view; raindrops on the windshield can act as a double lens or generate glares; static snow on roads may cover the lane markings, affecting detection of driveable areas, while wet road surfaces might result in reflections and artefacts due to water puddles, and deteriorated contrast between road features. As the success of autonomous vehicles depends on the ability to overcome the effects of these conditions, some studies have developed hardware solutions to tackle these problems. For example, [5] studies the performance of gated cameras, while [4] extends the study to combine stereo, gated and thermal cameras with Radar and LiDAR scanners, showing significant improvements for car detection at various levels of fog, rain and snow. Other studies use domain adaptation to 'change' the weather conditions as a post data-acquisition process. For example, [29] explores the effect of generated night-time and generated day-time rain images on road segmentation and traffic object detection, whereas [34] shows an improvement in localisation by using generated night-time imagery and [33] develops a de-raining model to improve semantic segmentation.

2.1. Real adverse weather capture

Among the first to provide a dataset with clear and weather-affected image pairs were the authors of [10], who used a transparent pane to add dirt and droplets to real-world scenery. Unfortunately, the dataset focuses only on static scenes. In the same category fall the works of [46], which uses 4 cameras attached to a vehicle to capture pairs of clear and images affected by soil; [33], which uses a stereo camera behind a bi-partite chamber with one clear lens and one lens affected by adherent droplets; and finally, [35] which uses a similar setup to [10], but captures outdoor images in an indoor environment.

A related but different category is represented by efforts to collect and annotate data in a series of target conditions such as: night-time, rainy night, heavy snow and other variations, such as [39, 28, 45, 31, 21, 44]. Whist these provide some of the most extensive datasets so far, the data is limited to specific road conditions in specific areas of the world, and the data collection process is heavily influenced by weather forecast. To facilitate the development of a truly weather-proof system using real data would require the collection of training imagery in all conditions, in all usage areas and at all times - which is a time-consuming and expensive undertaking. To overcome this difficulty, efforts have been made to provide a cost-effective and more scalable alternative, such as augmented visual data that is based upon physics models, synthesis or appearance style transfer.

2.2. Synthetic adverse weather generation

Physics-based approaches are often employed in generating synthetic weather, especially for fog and droplets. For example, [36] proposes a pipeline that uses a stereo pair and depth information to add synthetic fog on clear images,

while [13] creates a purely synthetic fog dataset based on Synscapes [49] (synthetic fog to synthetic images). Similarly, [14] uses a physics simulator to add rain streaks and fog on clear images and further tests object detection and semantic segmentation on real rainy imagery, while [33] uses a physics-based pipeline to add synthetic adherent raindrops to clear images, and further tests a lane-marking segmentation model.

Furthermore, [1] uses a computer graphics engine to render photo-realistic in-focus and defocused raindrops, and [22] develops a model for restoring images affected by heavy rain. Neither, however, tests the viability of restored images as training data, focusing instead on reconstruction metrics. On the other hand, [23] develops a decomposition network to split rain-affected images into a clean image and a rain layer and further trains the model on synthetic generated rain, but only tests it on 20 real-world images.

2.3. Weather appearance transfer

Due to the recent developments in GAN [12] and CycleGAN architectures [54], an increasing body of research has been devoted to applying these models for autonomous driving tasks. In the case of unpaired data, the first to use appearance transfer were the authors of [34], who trained a model to generate images with snow and diverse illumination in order to optimise feature matching for localisation. Later research includes that of [2], which generates day-time images from night-time images in order to improve retrieval-based localisation and [53], which learns to de-haze synthetic hazy images. The authors of [24] generate night-time images from day-time images, whereas [47] generates the soiled counterpart of a clear image and [11] adds synthetic fog to clear images. Similarly, [29] generates night-time images from day-time images and shows qualitative results on a day-time image with adherent droplets, while [9] is among the few papers that test semantic segmentation under rainy night conditions, with the drawback that their model requires paired data. While the aforementioned works provide multiple weather appearance pairs, they do not combine or stack conditions, and provision of a dataset is outside the scope of their work.

Other approaches involve direct image synthesis using paired data, with prominent examples being [20] and [48] which synthesize images from semantic or instance maps. These models, however, assume that a semantic segmentation ground truth exist in order to ensure higher-quality image generation. Later extensions that aim to improve realism include AdaIN [19] and SPADE [30], which propose improved normalization techniques to encourage the intermediate convolutional layers to make a better use of the input data. We describe how our work derives from existing methods in the following section.

3. Methodology

Our overall weather stacking methodology (Fig. 2) consists of two stages. In the first stage, a set of N models receives as input real overcast imagery, and outputs the same data but in N different weather styles. In the second stage, a single model receives the generated weather images and adds adherent raindrops.

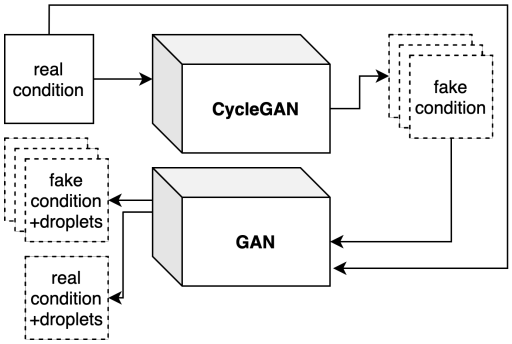


Figure 2: Overall methodology for weather stacking. First, an image translation CycleGAN model (trained using unpaired data) is used to create N weather and illumination conditions from a reference real condition. Then, a second image translation GAN model (trained using paired data) is used to apply adherent droplets to the N conditions. The current setup is *one* example of such a model stack, with both models being freely electable.

3.1. Datasets

We chose Cityscapes [8] as our source dataset, on which we transfer weather appearance (snowy, rainy/wet, night-time), using Oxford RobotCar [26] as a source of style for rainy/wet and snowy and the train set of Dark Zurich [39] for night-time appearances.

Cityscapes was chosen as it is a widely used dataset for training downstream tasks, with high-quality instance annotations and additional sources of ground-truth such as disparity. Additionally, many of the methods adapted in this work have been either trained or tested on Cityscapes or its derivatives. The choice for a source dataset is however open and free and should be consistent with the target applications. Since RainyScreens [35] contains imagery captured through a transparent pane with added droplets, it makes a good source of paired data for training a droplet generation model. Finally, to evaluate the night and night+droplets generated data, we extract diverse real images with adverse weather from Mapillary [27], BDD100K [51], DAWN [21] and ACDC [37] to cover the conditions of interest.

3.2. Models

Generative Adversarial Networks [12] are a class of generative models where a generator and a discriminator compete against each other: the generator G learns to generate data from a particular distribution $p_{data}(x)$, whereas the discriminator D learns to detect which data comes from the same distribution. The learning setting can thus be formulated as a minimax game, where each of the models tries to

minimise its own losses:

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{y\sim p_y(y)}[\log(1-D(G(y)))]. \quad (1)$$

The generator seeks to minimise its own loss by generating images with high fidelity. Thus, its loss will be minimal when the discriminator is fooled, i.e., $D(G(y)) = 1$. On the other hand, the loss of the discriminator will be minimal when it is able to correctly identify real images ($D(x) = 1$) from generated images ($D(G(y)) = 0$).

Cycle-consistency GANs are an extension of GAN models, developed in order to allow image translation between unpaired datasets. Training a CycleGAN involves optimizing simultaneously two generators and two discriminators, where one generator learns the mapping function from a domain A to a domain B, while the other learns the mapping from domain B to domain A. Since the supervision of the two discriminators is not enough to ensure transfer, an additional reconstruction loss is used in order to enforce cycle consistency, by forcing the two generators to reconstruct each other's output back into the original domain.

We use N CycleGANs ($N = 3$ in our case) to train image translation from $(real, overcast, daytime) \rightarrow (fake, night)$, $(real, overcast, daytime) \rightarrow (fake, wet)$ and $(real, overcast, daytime) \rightarrow (fake, snow)$, using the official architecture [54] but with a SPADE-based generator, as it was shown to generate images with higher fidelity due to improved normalization layers [30].

For the paired image translation task (in this case, applying adherent droplets to the generated conditions) we use a pix2pix-like architecture [20, 48], again with a SPADE-based generator [30]. As we have pairs of clear and droplet-affected images of the Cityscapes dataset (from the Rainyscreen dataset [35]), we employ one single GAN to learn the $(real, overcast, daytime) \rightarrow (fake\text{-}droplet, overcast, daytime)$ mapping, and at inference time we run it on the N conditions generated as previously explained.

3.3. Evaluation

3.3.1 Perceptual quality evaluation

Following image quality assessment methods as used in [30] and [41], we evaluate the perceptual quality of the generated styles using Fréchet Inception Distance (FID) [18], but also Inception Score (IS) [40], both based on the Inception-v3 network [43], and shown to be in line with human judgements [6]. Whereas IS is computed by taking into account the predicted class probabilities of generated images via [43], the FID score analyses the last pooling layer (prior to classification) and models the activations of real and generated images as two multi-variate Gaussian distributions, calculating the distance between the two distributions using the Fréchet distance. An image with high diversity and high quality would have a high IS, whereas an image with a low FID would correlate with high quality.

Dataset	O	D	W	WD	S	SD	N	ND
BDD100K	100	37	70	68	63	12	46	14
Mapillary	65	-	99	9	385	-	20	-
DAWN	-	-	-	200	-	204	-	-
ACDC	-	-	400	-	400	-	400	-

Table 1: Number of images used in out-of-sample testing of Mask-RCNN

3.3.2 Quantitative evaluation

The suitability of the generated images as training data for relevant downstream tasks is extensively tested on various real weather conditions, in terms of (i) object detection, (ii) semantic segmentation and (iii) instance segmentation performance. Due to the large number of condition-and-dataset combinations, we chose to use and fine-tune Mask-RCNN [15], as it performs all tasks from the same backbone, while training relatively efficiently. We would like to stress that our goal is not to produce state-of-the-art results, but instead to assess the suitability of our generated training data while keeping all other variables constant. Any other recent or state-of-the-art model could be a substitute for MaskRCNN, yielding potentially better results overall. Table 1 contains a summary of the datasets, weather conditions and number of images extracted and used for testing.

In order to evaluate the suitability of each generated condition, we start with a Mask-RCNN model pre-trained on Cityscapes real overcast images, which is then further fine-tuned for each generated condition (7 different instances of the same initial pre-trained model). After fine-tuning each model, we test it out-of-sample and out-of-distribution on the real conditions extracted previously, and note the changes in results. Finally, to test the performance in the case of a monolithic model instead of individual models, we fine-tune one model on all the generated conditions at once, and test again out-of-sample on all real conditions.

The image-translation models were trained on images that have been resized to 512×1024 and randomly cropped to 512×512. In this way we enforce a uniform standard across all analyses and ensure that the ground truth is processed to reflect the changes.

We use the Detectron2 [50] implementation of Mask-RCNN with a ResNet+FPN backbone [16, 25]), which outputs both predicted masks and bounding boxes. We start with the official pre-trained model on ImageNet, COCO and Cityscapes for instance segmentation and bounding box detection.

We report our results in terms of mean Average Precision (AP), AP@50 and AP@75[1], for Object detection, Semantic segmentation and Instance segmentation, depending on the ground truth availability of the test dataset.

[1] AP at IoU=.50/IoU=.75, where only candidates with an area at least 50%/70% compared to GT area are considered.

4. Results

4.1. Qualitative results

Using the IS and FID metrics described in section 3.3.1, the results are reported in Table 2. The Inception Score is provided as a means for performing a rough comparison with other approaches, but needs to be used carefully when comparing models, as outlined in [3]. On the other hand the FID score may be used to check the degree of alignment (how similar the conditions or their distributions might be) between datasets or conditions, and in our case has a surprising amount of correlation with the Quantitative results reported in Table 3, under "Improvement on individual fine-tuned models". Cityscapes' synthetic conditions that have a comparatively lower (better) FID score with respect to BDD (such as *wet*, *snow* and *night*) also perform much better on their corresponding BDD conditions, with the ranking predicted by the FID score being a good indicator of object detection and instance segmentation performance across various conditions.

Dataset	IS	FID
Real overcast (**O**)	3.75	69.74
Fake droplets on real overcast (**D**)	4.04	124.66
Fake wet (**W**)	4.13	87.10
Fake droplets on fake wet (**WD**)	3.21	182.24
Fake snow (**S**)	4.12	116.40
Fake droplets on fake snow (**SD**)	3.72	227.00
Fake night (**N**)	3.27	86.56
Fake droplets on fake night (**ND**)	3.45	152.48

Table 2: Qualitative results for Inception Score and Fréchet Inception Distance. FID is computed wrt. the selected BDD100K train set.

4.2. Quantitative results

We split our quantitative analysis into four parts: testing against the BDD, Mapillary, DAWN and ACDC datasets, respectively. We would like to point out that all our testing, except for the initial baseline, is done on out-of-distribution data in order to strengthen the validity of the results and to act as a better proxy for real world performance.

4.2.1 BDD

Table 3 shows our results on various conditions extracted from the BDD dataset. We begin by benchmarking the performance of the model fine-tuned on half-resolution, center-cropped Cityscapes overcast images against the Cityscapes validation set, in order to establish a baseline (1st row).

First, we note that the performance is slightly lower that the official baseline in [50] due to the use of half-resolution images. We then assess the loss of performance due to domain shift by testing the same model on BDD real *overcast* imagery. We note that the drop in performance is less pronounced for Object detection as compared to Instance segmentation, but still significant. After establishing these two **overcast baselines**, we then assess the performance of the model (fine-tuned on *overcast* images) on the 7 representative conditions extracted from BDD, establishing our **condition baselines**. We notice particularly low performance for *real droplets on real night* and unusually high performance for *real droplets on real snow*. This is potentially due to the low number of samples used for these two conditions (see Table 1), and should be assessed with care.

We then test models trained on the 7 individual synthetic Cityscapes conditions on their respective BDD conditions. On first analysis, the mixed results (potentially discounting the two aforementioned conditions with low number of samples) might be surprising, with *fake night* and *fake wet* showing large increases, *fake snow* and *fake droplets on fake wet* remaining largely the same, and *fake droplets on real overcast* showing much lower performance. However, the FID scores in Table 2 may contain an indication for this behavior: we notice that *fake night* and *fake wet* have relatively good (low) FID scores, appearing to be aligned with their respective BDD conditions, while the other 5 conditions seem much more unaligned with their respective BDD conditions.

We analyze this claim in the next block of rows, where we show results for testing a model trained on all the Cityscapes conditions against the individual BDD conditions. Because the model now has to learn to generalise across a much wider set of distributions of conditions instead of only one potentially misaligned distribution, we would expect to see significant gains against both the results on individual models and against the baselines. And indeed, we observe gains across all conditions, with large improvements (discounting the two conditions with reduced samples) for *fake droplets on real overcast*, *fake night*, *fake snow*, *fake wet* and *fake droplets on fake wet*. Additionally, we test this model on the Cityscapes overcast validation set and show that it outperforms the original baseline, by up to 3.3 percentage points.

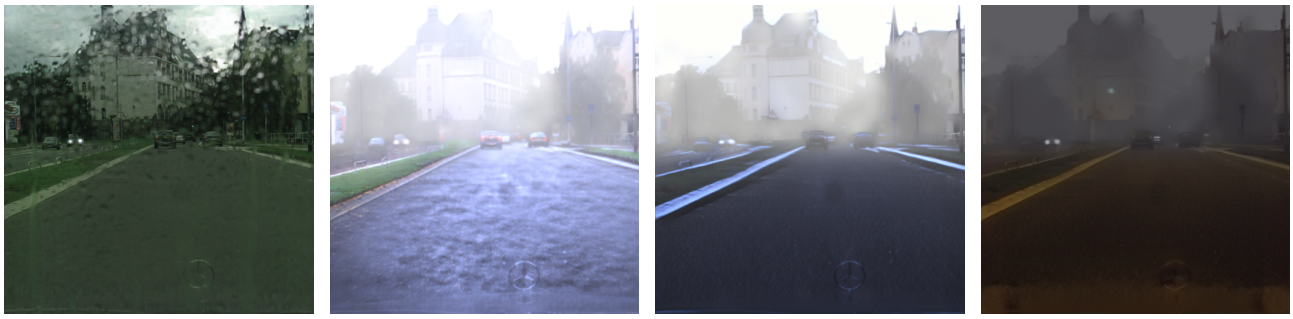
4.2.2 Mapillary, DAWN and ACDC

To make up for the reduced number of samples for certain conditions in BDD, we also test on conditions extracted from the Mappillary dataset, with results presented in Table 4. We follow the same procedure as for BDD, establishing baselines, observing mixed results for individual models, and finally obtaining significant increases for all conditions when using the model trained on all synthetic Cityscapes conditions (for example an almost 11 percentage points increase in AP@50 when testing on snow). The DAWN [21] dataset contains examples of harsh weather conditions, and specifically covers *real droplets with snow* (which was underrepresented in BDD) and *real droplets with wet*. Our results are reported in Table 5. Again, we begin by establishing baselines for the model trained on overcast data. We then obtain mixed results for the individually trained models, with *fake droplets on fake wet* improving considerably. Finally, we show significant improvements on both conditions when using the model trained on all synthetic

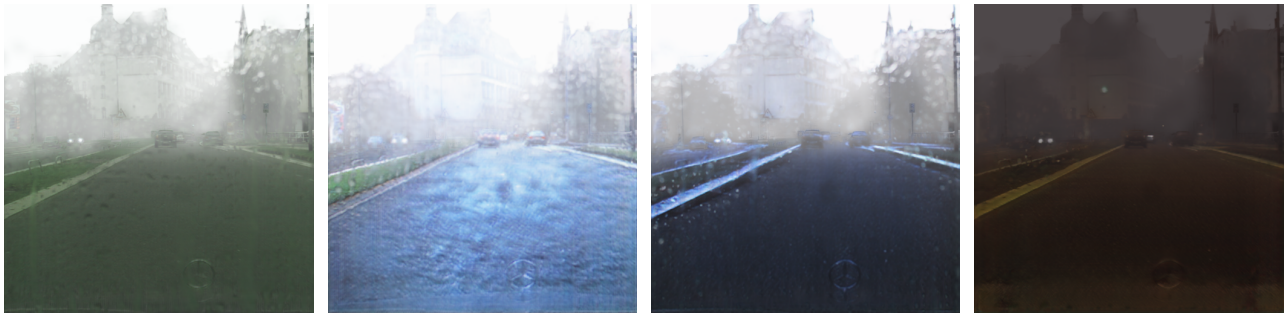
(a) real overcast **(b)** fake snow **(c)** fake rain/wet **(d)** fake night

(e) fake droplets (real overcast) **(f)** fake droplets (fake snow) **(g)** fake droplets (fake rain/wet) **(h)** fake droplets (fake night)

Figure 3: Generated conditions



(a) drop(real) **(b)** fog(wet(real)) **(c)** fog(snow(real)) **(d)** fog(night(real))



(e) fog(drop(real)) **(f)** fog(drop(wet(real))) **(g)** fog(drop(snow(real))) **(h)** fog(drop(night(real)))

Figure 4: Extension: fog applied on generated conditions, with a fog coefficient of 0.01

Cityscapes conditions, with *fake droplets on fake wet* gaining more than 10 percentage points over the real *overcast* baseline, and *fake droplets on fake snow* more than 6 percentage points.

The ACDC dataset [37] contains examples of night, snow and wet conditions with semantic segmentation ground truth. We report results in Table 6. Similarly to previous experiments, we observe mixed results for individual models and an increase in performance across the board for the monolithic model, reinforcing the trend observed in previous experiments.

Description	Fine-tune set Cityscapes	Test condition	Object detection AP	AP@50	AP@75	Instance segmentation AP	AP@50	AP@75
Domain shift to BDD	Real **O** train	City real **O** val	31.83	52.73	30.77	27.14	48.15	24.95
	Real **O** train	BDD **O**	29.22	51.92	25.28	20.99	37.52	19.32
Weather baselines	Real **O** train	BDD **D**	26.00	47.82	16.06	19.57	43.12	7.49
	Real **O** train	BDD **N**	20.60	29.90	23.67	15.11	27.69	21.50
	Real **O** train	BDD **ND**	7.75	19.12	3.61	4.02	14.74	1.23
	Real **O** train	BDD **S**	24.95	40.40	27.08	20.50	34.37	19.48
	Real **O** train	BDD **SD**	39.38	57.60	51.67	37.40	52.02	45.98
	Real **O** train	BDD **W**	22.09	39.99	20.95	16.58	34.54	15.55
	Real **O** train	BDD **WD**	17.57	37.65	17.04	13.76	32.10	10.99
Results on individual fine-tuned models	Synth **D** train	BDD **D**	21.03	35.92	22.97	15.86	32.51	8.88
	Synth **N** train	BDD **N**	26.45	36.48	25.82	22.03	33.56	22.70
	Synth **ND** train	BDD **ND**	16.86	34.21	9.48	8.35	22.41	3.07
	Synth **S** train	BDD **S**	25.07	42.62	26.46	17.33	27.35	19.40
	Synth **SD** train	BDD **SD**	8.77	15.68	9.66	4.44	8.05	3.58
	Synth **W** train	BDD **W**	25.12	43.30	25.35	18.73	37.13	17.52
	Synth **WD** train	BDD **WD**	17.10	35.64	15.91	15.22	30.26	9.13
Results on all-weathers fine-tuned models	Synth all train	BDD **D**	31.46	52.71	42.89	21.43	48.13	13.01
	Synth all train	BDD **N**	26.73	46.08	32.75	25.43	42.48	24.09
	Synth all train	BDD **ND**	26.27	45.32	26.78	13.66	38.36	4.63
	Synth all train	BDD **S**	36.59	62.38	33.28	28.29	56.72	28.50
	Synth all train	BDD **SD**	36.33	49.62	44.47	29.88	43.65	36.40
	Synth all train	BDD **W**	35.13	58.62	35.14	28.94	52.37	27.45
	Synth all train	BDD **WD**	23.98	49.39	23.56	24.88	42.54	30.06
Re-test	Synth all train	City real **O** val	35.18	56.42	35.37	25.68	44.63	24.66

Table 3: Object detection and instance segmentation results on BDD conditions

Description	Fine-tune set Cityscapes	Test condition	Object detection AP	AP@50	AP@75	Instance segmentation AP	AP@50	AP@75
Domain shift to Mapillary	Real **O** train	City real **O** val	31.83	52.73	30.77	27.14	48.15	24.95
	Real **O** train	Mapillary **O**	24.02	38.44	25.48	20.20	35.49	19.05
Weather baselines	Real **O** train	Mapillary **N**	10.40	19.08	11.07	7.85	19.34	6.52
	Real **O** train	Mapillary **S**	11.12	18.18	11.17	10.51	16.91	11.14
	Real **O** train	Mapillary **W**	17.67	29.03	17.64	15.06	29.11	13.07
	Real **O** train	Mapillary **WD**	12.90	17.94	14.37	13.90	27.13	13.98
Results on individual fine tuned models	Synth **N** train	Mapillary **N**	8.44	17.34	6.55	6.15	11.66	5.74
	Synth **S** train	Mapillary **S**	7.75	12.27	8.59	7.04	11.51	7.09
	Synth **W** train	Mapillary **W**	16.09	27.78	15.54	15.10	26.24	15.04
	Synth **WD** train	Mapillary **WD**	13.16	19.96	12.95	15.18	29.53	16.89
Results on all-weathers fine tuned models	Synth all train	Mapillary **N**	11.14	19.94	9.11	9.80	16.62	10.25
	Synth all train	Mapillary **S**	13.69	23.22	13.28	13.22	21.10	14.40
	Synth all train	Mapillary **W**	18.22	30.89	18.69	16.80	32.41	12.73
	Synth all train	Mapillary **WD**	16.52	25.60	15.04	17.12	23.73	18.64

Table 4: Object detection and instance segmentation results on Mapillary conditions

Description	Fine-tune set Cityscapes	Test condition	AP	AP @50	AP @75
Weather baselines	Real **O** train set	DAWN **SD**	9.27	25.19	6.84
	Real **O** train set	DAWN **WD**	10.39	18.24	11.11
Results on individual fine tuned models	Synth **SD** train set	DAWN **SD**	8.13	16.04	7.74
	Synth **WD** train set	DAWN **WD**	14.49	24.96	15.76
Results on all-weathers fine tuned models	Synth all train set	DAWN **SD**	16.55	37.21	14.46
	Synth all train set	DAWN **WD**	21.20	39.19	21.62

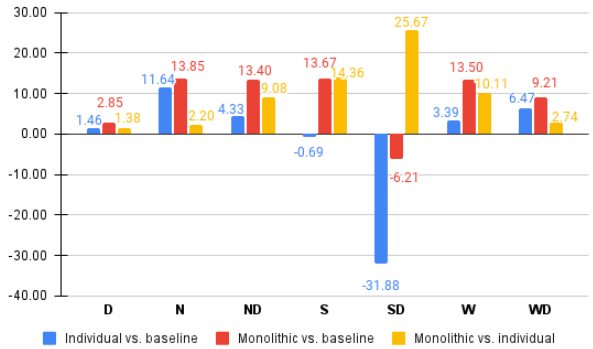
Table 5: Object detection results on DAWN conditions

Description	Fine-tune set Cityscapes	Test condition	AP	AP @50	AP @75
Weather baselines	Real **O** train set	ACDC **N**	1.63	4.68	1.12
	Real **O** train set	ACDC **S**	9.29	20.96	6.31
	Real **O** train set	ACDC **W**	10.60	21.74	8.04
Results on individual fine tuned models	Synth **N** train set	ACDC **N**	3.03	8.24	1.81
	Synth **S** train set	ACDC **S**	5.95	14.15	4.32
	Synth **W** train set	ACDC **W**	9.76	20.16	7.22
Results on all-weathers fine tuned models	Synth all train set	ACDC **N**	3.95	10.51	2.10
	Synth all train set	ACDC **S**	12.07	27.09	9.41
	Synth all train set	ACDC **W**	11.69	25.48	8.10

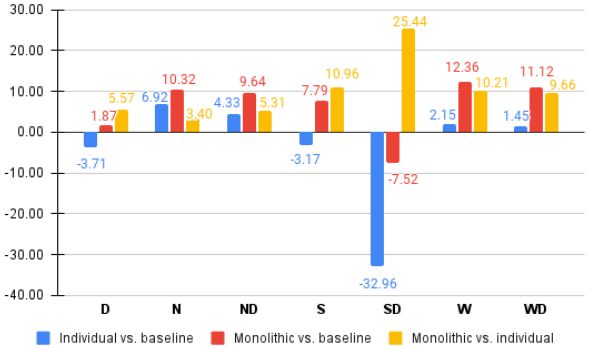
Table 6: Semantic segmentation results on ACDC conditions

5. Conclusions and proposed work

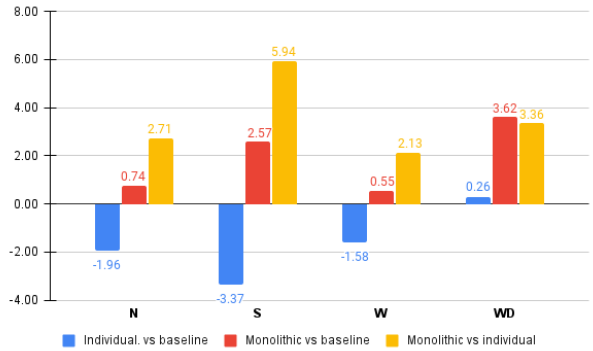
In this work we propose a modular architecture aimed to unlock diverse and stackable weather conditions with the purpose of weather appearance synthesis for improving perception downstream tasks. We generate 7 different weather styles starting from a *single* dataset with ground truth and show significant improvements in AP in both object detection and instance segmentation, in many cases exceeding 10 percentage points increase in AP. Due to the difficulty of finding aligned real-world conditions with existing ground truth from available datasets, we also train a monolithic model and show significant improvements not only over the weather baselines, but also over the original real overcast Cityscapes baseline. Finally, we publish instructions to reconstruct the dataset. As future work, we propose an ex-



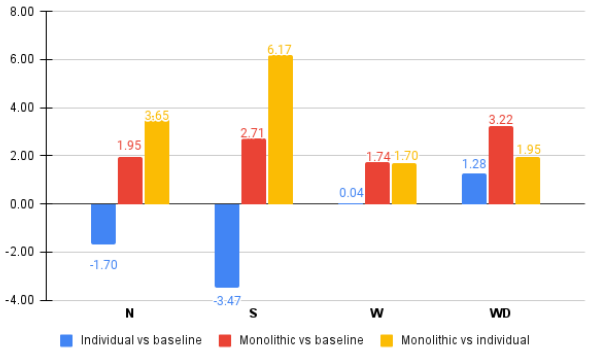
(a) Percentage points improvements in AP for Object detection, evaluated on BDD weather conditions



(b) Percentage points improvements in AP for Instance segmentation, evaluated on BDD weather conditions

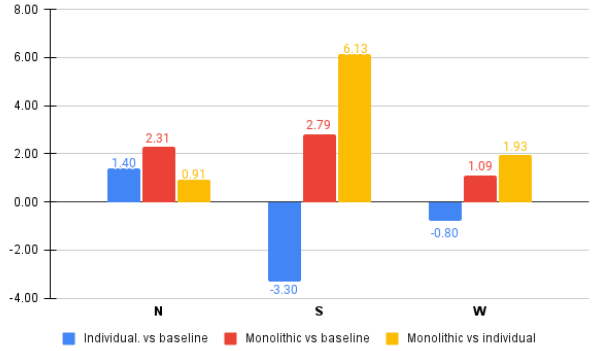


(c) Percentage points improvements in AP for Object detection, evaluated on Mapillary weather conditions

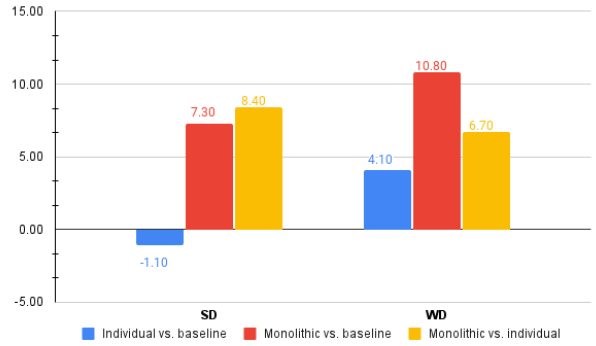


(d) Percentage points improvements in AP for Instance segmentation, evaluated on Mapillary weather conditions

Figure 5: Due to misalignment between the train conditions and test conditions, models trained on individual conditions only may exhibit loss of performance (Blue). However, training the model on all weathers leads to large improvements in performance across all conditions, compared to the baseline model trained on the original Cityscapes overcast imagery.



(a) Percentage points improvements in AP for Semantic segmentation, evaluated on ACDC weather conditions



(b) Percentage points improvements in AP for Object detection, evaluated on DAWN weather conditions

Figure 6: Reinforcing the trend observed in previous experiments, we observe that models trained with all conditions perform significantly better than either the baselines or models trained with individual conditions.

tension to our current overall methodology based on Foggy Cityscapes [38], which applies synthetic fog on real images with good weather conditions. Since their fog pipeline is based on stereo pairs from Cityscapes, we are able to use the authors' provided demo in order to add synthetic fog to our generated weathers. While quantitative analysis of the extended foggy conditions is out of the scope of this paper, we present visual results in Fig.4.

ACKNOWLEDGMENT

The authors wish to thank Alexander Rast, Peter Ball and Matthias Rolf for fruitful discussions and support throughout this work, and Izzeddin Teeti and Valeriu Plamadeala for helping out with test data management. This project has received funding from the European Union's Horizon 2020 research and innovation programme, under grant agreement No. 964505 (E-pi).

References

[1] S. Alletto, C. Carlin, L. Rigazio, Y. Ishii, and S. Tsukizawa. Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2329–2338, 2019. 3

[2] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. pages 5958–5964, 05 2019. 1, 3

[3] Shane Barratt and Rishi Sharma. A Note on the Inception Score. *arXiv e-prints*, page arXiv:1801.01973, Jan. 2018. 5

[4] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. pages 11679–11689, 06 2020. 2

[5] Mario Bijelic, Tobias Gruber, and W. Ritter. Benchmarking image sensors under adverse weather conditions for autonomous driving. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1773–1779, 2018. 1, 2

[6] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179, 02 2018. 4

[7] Pak Chan, Gunwant Dhadyalla, and Valentina Donzella. A framework to analyze noise factors of automotive perception sensors. *IEEE Sensors Letters*, PP:1–1, 05 2020. 2

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[9] Shuai Di, Qi Feng, Chun-Guang Li, Mei Zhang, Honggang Zhang, Semir Elezovikj, Chiu Tan, and Haibin Ling. Rainy night scene understanding with near scene semantic adaptation. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–9, 02 2020. 3

[10] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. pages 633–640, 12 2013. 1, 2

[11] Rui Gong, Dengxin Dai, Yu-Hua Chen, Wen Li, and L. Gool. Analogical image translation for fog generation. *ArXiv*, abs/2006.15618, 2020. 3

[12] Ian J. Goodfellow, Jean Pouget-Abadie, M. Mirza, B. Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *ArXiv*, abs/1406.2661, 2014. 3

[13] M. Hahner, D. Dai, C. Sakaridis, J. Zaech, and L. V. Gool. Semantic understanding of foggy scenes with purely synthetic data. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3675–3681, 2019. 1, 3

[14] Shirsendu Sukanta Halder, Jean-Franccois Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10202–10211, 2019. 3

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 4

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[17] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork. Weather influence and classification with automotive lidar sensors. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1527–1534, 2019. 1

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 12 2017. 4

[19] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017. 3

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei Efros. Image-to-image translation with conditional adversarial networks. pages 5967–5976, 07 2017. 3, 4

[21] Mourad A. Kenk and M. Hassaballah. Dawn: Vehicle detection in adverse weather nature dataset. *ArXiv*, abs/2008.05402, 2020. 1, 2, 3, 5

[22] Ruoteng Li, L. Cheong, and R. Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1633–1642, 2019. 3

[23] Siyuan Li, Wenqi Ren, Jiawan Zhang, Jinke Yu, and Xiaojie Guo. Single image rain removal via a deep decomposition–composition network. *Computer Vision and Image Understanding*, 186:48–57, 2019. 3

[24] C. Lin, S. Huang, Y. Wu, and S. Lai. Gan-based day-to-night image style transfer for nighttime vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, 2020. 3

[25] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 4

[26] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 3

[27] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[28] Lukáš Neumann, Michelle Karg, Shanshan Zhang, Christian Scharfenberger, Eric Piegert, Sarah Mistr, Olga Prokofyeva, Robert Thiel, Andrea Vedaldi, Andrew Zisserman, and Bernt Schiele. *NightOwls: A Pedestrians at Night Dataset*, pages 691–705. 05 2019. 2

[29] Vladislav Ostankovich, R. Yagfarov, Maksim Rassabin, and S. Gafurov. Application of cyclegan-based augmentation for

autonomous driving at night. *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, pages 1–5, 2020. 2, 3

[30] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019. 2, 3, 4

[31] Matthew Pitropov, Danson Evan Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, 0(0):0278364920979368, 0. 2

[32] Horia Porav, Tom Bruls, and P. Newman. Don't worry about the weather: Unsupervised condition-dependent domain adaptation. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 33–40, 2019. 1

[33] H. Porav, T. Bruls, and P. Newman. I can see clearly now: Image restoration via de-raining. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7087–7093, 2019. 1, 2, 3

[34] H. Porav, W. Maddern, and P. Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1011–1018, 2018. 1, 2, 3

[35] Horia Porav, Valentina-Nicoleta Musat, Tom Bruls, and P. Newman. Rainy screens: Collecting rainy datasets, indoors. *ArXiv*, abs/2003.04742, 2020. 1, 2, 3, 4

[36] Christos Sakaridis, Dengxin Dai, and L. Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018. 2

[37] Christos Sakaridis, Dengxin Dai, and L. Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. *ArXiv*, abs/2104.13395, 2021. 3, 6

[38] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018. 1, 2, 8

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 3

[40] Tim Salimans, Ian J. Goodfellow, W. Zaremba, Vicki Cheung, A. Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 4

[41] Edgar Schönfeld, Vadim Sushko, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *International Conference on Learning Representations*, 2021. 4

[42] Lei Sun, Kaiwei Wang, Kailun Yang, and Kaite Xiang. See clearer at night: towards robust nighttime semantic segmentation through day-night image conversion. page 8, 09 2019. 1

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and ZB Wojna. Rethinking the inception architecture for computer vision. 06 2016. 4

[44] X. Tan, Yiheng Zhang, Ying Cao, Lizhuang Ma, and R. Lau. Night-time semantic segmentation with a large real dataset. *ArXiv*, abs/2003.06883, 2020. 2

[45] F. Tung, J. Chen, L. Meng, and J. J. Little. The raincouver scene parsing benchmark for self-driving in adverse weather and at night. *IEEE Robotics and Automation Letters*, 2(4):2188–2193, 2017. 1, 2

[46] Michal Uricar, Jan Ulicny, Ganesh Sistu, Hazem Rashed, Pavel Krizek, David Hurych, Antonin Vobecky, and Senthil Yogamani. Desoiling dataset: Restoring soiled areas on automotive fisheye cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

[47] Michal Uřičář, Pavel Křížek, Ganesh Sistu, and Senthil Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, page 67–72. IEEE Press, 2019. 3

[48] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3, 4

[49] Magnus Wrenninge and J. Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *ArXiv*, abs/1810.08705, 2018. 3

[50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4, 5

[51] F. Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, V. Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *ArXiv*, abs/1805.04687, 2018. 3

[52] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, 14(2):103–111, 2019. 1

[53] Jingming Zhao, Juan Zhang, Z. Li, J. Hwang, Yongbin Gao, Zhijun Fang, Xiaoyan Jiang, and Bo Huang. Dd-cyclegan: Unpaired image dehazing via double-discriminator cycle-consistent generative adversarial network. *Eng. Appl. Artif. Intell.*, 82:263–271, 2019. 3

[54] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 2, 3, 4

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Multi-weather city: Adverse weather stacking for autonomous driving
Publication Status	Published
Publication Details	**V. Mușat**, I. Fursa, P. Newman, F. Cuzzolin and A. Bradley, "Multi-weather city: Adverse weather stacking for autonomous driving," *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, 2021, pp. 2906-2915, doi: 10.1109/ICCVW54120.2021.00325.

Student Confirmation

Student Name:	Valentina Musat		
Contribution to the Paper	- developed the idea supporting the paper under the guidance of Mr. Bradley and Prof. Cuzzolin - implemented the architecture - compiled the data for training - ran all experiments and interpreted data - wrote the manuscript - prepared presentation materials and presented the paper at ICCV remotely		
Signature 		Date	21.04.2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman		
Supervisor comments I certify that the candidate made a substantial and leading contribution to the publication, and that the description described above is accurate		
Signature 	Date	24/04/25

4.3 Further insights

Table 2 from the integrated manuscript, which presents qualitative results in terms of both IS and FID scores, has a good degree of correlation with the quantitative results presented in Table 3, but also a considerable amount of fluctuation in the FID scores, especially for *fake droplets on fake snow*. This is possibly due to the addition of droplets, which could introduce a large domain gap with respect to the BDD100K dataset (F. Yu et al., 2020), which contains examples of images with droplets but which may not be well aligned with the style of droplets found in the RainyScreens (Porav et al., 2020) dataset. This is evident in Table 3 concerning individual fine-tuned models, where performance with respect to baselines is degraded for some conditions, which might again highlight a degree of domain misalignment. However, for monolithic models trained on all synthetic weather conditions at the same time, performance is almost always improved compared to baselines, across all datasets, sometimes by a large amount. This could indicate that even in the absence of a robust mechanism for aligning domains, training on data with increased diversity results in an improvement compared to baselines.

The models employed in Multi-weather city to perform domain adaptation and image translation were based on GANs, with a SPADE (Park et al., 2019) backbone, and the state-of-the-art has progressed significantly ever since, having largely switched over to latent diffusion models. Both the paired and unpaired image translation tasks can be easily improved using approaches such as ControlNet (L. Zhang et al., 2023), which extends latent diffusion models with extra conditioning, or CycleGAN-Turbo (Parmar et al., 2024), which uses latent diffusion models in a cycle-consistent configuration.

More recently, various 3D-based methods have emerged to enhance synthetic data by incorporating weather effects. ClimateNeRF (Yuan Li et al., 2023) integrates 3D reconstruction with analytical methods and neural stylisation, where a NeRF model first learns a representation of the scene parametrised by densities and color estimates. Then, various heuristics - one for each weather type - are used to modify the densities and colors, with the resulting image being obtained using the typical NeRF volume rendering approach. Neural stylisation (Yijun Li et al., 2018) is used to further change the style of images, with the underlying NeRF model being subsequently fine-tuned on the updated images to improve multi-view consistency. ClimateGS (Yuezhen Xie et al., 2025) adopts a very similar approach using gaussian splatting instead of NeRF, while RainyGS (Dai et al., 2025) explores rain streaks and puddles by combining analytical methods with a height map derived from a gaussian splatting representation.

These subsequent approaches highlight that working solely in a 2D image space without any understanding of weather placement guided by the 3D geometry of the scene is suboptimal, and methods that incorporate 3D information are more suitable for this endeavor. However, the disadvantage of many such models is that they rely on heuristics that are specifically tailored to each weather type. Furthermore, when tackling multiple types of weather conditions, a data-driven mechanism is more appropriate as it offers flexibility, especially when dealing with weather combinations where multiple heuristics-based components would need to be combined. As such, future work is proposed in Section 8.2.

5

Depth-SIMS

Contents

5.1 Contribution

Domain adaptation approaches such as Multi-weather city (Musat et al., 2021) commonly leverage an existing dataset of real-world annotated data to generate more data. While this is a powerful approach, the initial real-world dataset may not contain the necessary placement and co-occurence of class objects to ensure adequate data coverage. At the time of publication, the field of image synthesis had emerged as a possible solution to this, with GANs demonstrating good results, however, they were still suffering from either a domain gap, or a difficulty in controlling the style of the individual objects or scenes in synthesized images. Both of these issues can be alleviated by making use of existing real-world data: while data on hand may not contain the required co-occurence of classes,

it can be decomposed into its constituent semantic parts and re-combined to generate novel scenes with the desired elements. Additionally, this approach also enhances photo-realism as it leverages already-captured components. As such, this paper contributes to the field of perception for autonomous driving by enabling scene composition using an approach for synthesizing pairs of images, dense depth and semantic segmentation maps. The key contributions of the paper include:

1. A semi-parametric model that combines the advantages of both parametric and non-parametric methods for image and depth synthesis. Components (blobs) obtained from real-world data have a low domain gap, but re-combining them into a novel scene is a difficult task, requiring inpainting and image harmonisation. The method includes an updated model for image harmonisation that retains as much of the original blobs as possible, and also a novel scheme for retrieving candidate blobs based on Hu moments (M.-K. Hu, 1962), thus avoiding expensive pair-wise comparisons;

2. Besides images, the method also produces a corresponding depth map, conditional on an initial sparse depth map but following the structure of the blobs contained in the newly synthesized image. This extends the applicability to training and validating depth prediction models as well;

3. While the synthesis process is conditioned on an initial semantic segmentation map, the resulting image and depth maps are constructed from blobs that may not perfectly match the shapes indicated by the segmentation map. To address this, the method also outputs an updated semantic segmentation map that matches the synthesized result;

4. Experiments demonstrating the effectiveness of the proposed method, both in terms of perceptual quality metrics such as FID, but also in terms of alignment with the conditioning semantic segmentation data. Additionally, the data is used to train both a semantic segmentation and a depth completion downstream task to measure its effectiveness as training data. At the time of publication, Depth-SIMS outperformed existing state-of-the-art methods, matching the perceptual quality of the best performing previous method but surpassing it by 3.7 percentage points in terms of alignment to the conditioning semantic map.

5.2 Integrated manuscript

The manuscript was published at the International Conference on Robotics and Automation (ICRA) 2022, (Musat et al., 2022)

2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Depth-SIMS: Semi-Parametric Image and Depth Synthesis

Valentina Muşat[1], Daniele De Martini*, Matthew Gadd*, Paul Newman
Mobile Robotics Group (MRG), University of Oxford
{valentina,daniele,mattgadd,pnewman}@robots.ox.ac.uk

***Abstract*— In this paper we present a compositing image synthesis method that generates RGB canvases with well aligned segmentation maps and sparse depth maps, coupled with an in-painting network that transforms the RGB canvases into high quality RGB images and the sparse depth maps into pixel-wise dense depth maps. We benchmark our method in terms of structural alignment and image quality, showing an increase in mIoU over SOTA by 3.7 percentage points and a highly competitive FID. Furthermore, we analyse the quality of the generated data as training data for semantic segmentation and depth completion, and show that our approach is more suited for this purpose than other methods.**

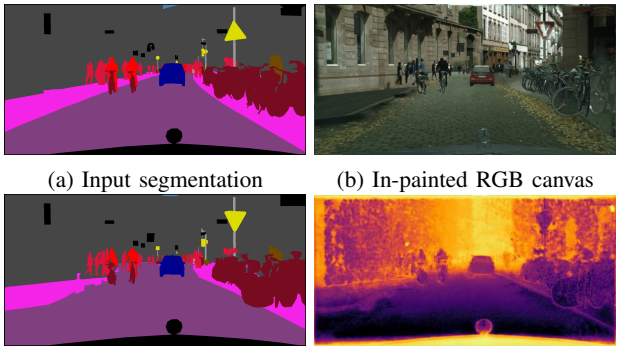
I. Introduction

Vision for autonomous driving has come a long way due to the availability of high-quality datasets with manual ground-truth (GT) annotations [1]–[3]. Still, providing GT annotation for real datasets is cumbersome, expensive and slow, especially at pixel-level. To overcome this bottleneck, several methods have emerged, enabled by the rapid development of deep-learning, two of which have received significant attention - domain adaptation and image synthesis.

While domain adaptation seeks to adapt data from a source domain to resemble the characteristics of a target domain (in particular day-to-night and sim-to-real appearance style transfer), image synthesis aims to generate data from abstract representations instead, such as semantic segmentation or latent embeddings. In the context of image synthesis, end-to-end parametric approaches involving Generative Adversarial Networks (GANs) have been particularly successful at reproducing the spatial structure of segmentation layouts, thus scoring high on metrics such as mean Intersection over Union (mIoU). On the other hand, semi-parametric approaches that combine both compositing-based and learned methods [4], achieve state-of-the-art (SOTA) results in terms of perceptual image quality, as measured by the Frechét Inception Distance (FID).

In this paper, we look at two strong and well-known examples of the above methods in the context of training data synthesis. We investigate the effect of perceptual image quality and structural alignment (measured via FID and mIoU respectively) of the synthesised images on the performance of semantic segmentation and depth completion. In doing so, we propose a set of improvements to a compositing framework derived from SIMS [4], yielding SOTA results in terms structural alignment of the synthesised images with their GT, highly-competitive results in terms of perceptual quality, *as well as* surpassing previous methods in terms of performance on downstream tasks trained with the generated data. Moreover, since our synthesised images are well aligned with their segmentation maps, we extend the model to also synthesise dense depth from sparse depth information, such that datasets with LiDAR information can be leveraged.

[1]Corresponding author. *Equal contribution. Thanks to the Assuring Autonomy International Programme, a partnership between Lloyd's Register Foundation and the University of York, as well as EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1).



(a) Input segmentation (b) In-painted RGB canvas

(c) Output segmentation (d) Synthesised depth

Fig. 1: Depth-SIMS produces in-painted RGB canvasses *alongside* synthesised dense depth maps.

II. Overview and Contributions

Our main contribution is a reformulation of the SIMS [4] compositing framework, for the purpose of generating well-aligned RGB images with complementary dense depth (Fig. 2). Therefore, our model enhances SIMS [4] by: 1) producing GT segmentation maps which are better aligned with the RGB image; 2) synthesizing dense depth from sparse depth alongside RGB images; 3) using Hu moments as blob descriptors instead of pair-wise Intersection over Union (IoU) comparisons, in order to speed up blob retrieval. As our goal is to build a simple and general framework with minimal handcrafted heuristics, we opt for a simple ordering of object blobs instead of the more complex ordering network that is employed in [4].

Our work is accompanied by an application of the generated data and GT to train semantic segmentation and monocular depth completion tasks.

III. Related Work

Image synthesis On one side of the image generation spectrum lies the image compositing process, through which a foreground image or object blob is overlaid on top of a background image or blank canvas. Although this method is straight-forward and maintains object realism, it often results in artefacts due to differences in appearance of objects coming from different sources (i.e. lighting, shading), or differences in object poses that lead to geometric inconsistencies. To solve these drawbacks, [6] presents a classic method

978-1-7281-9680-0/22/$31.00 ©2022 IEEE

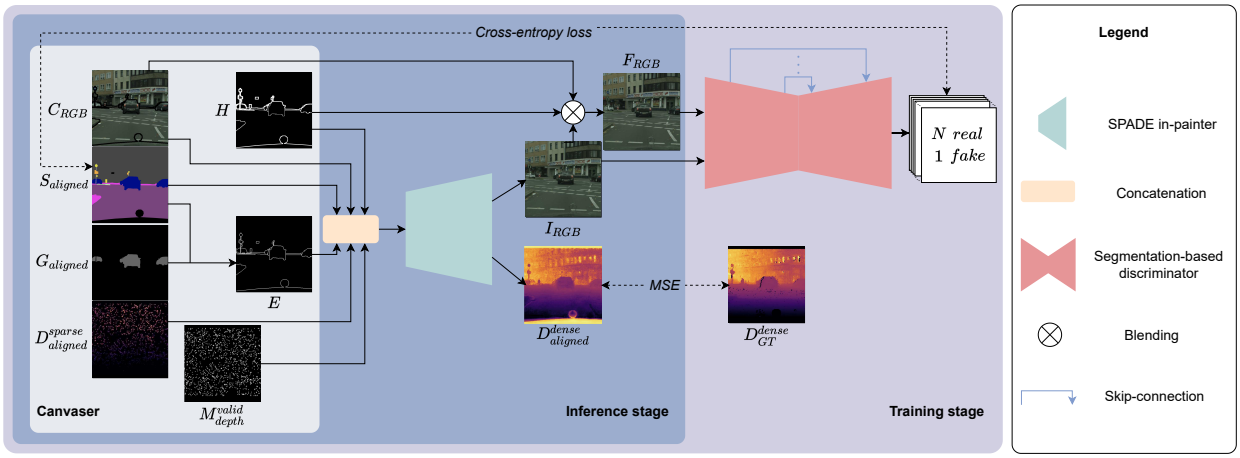


Fig. 2: GAN-based in-painting of RGB canvas holes and densification of sparse canvas depth maps; the blue area indicates the processes active at inference time, while the purple area shows the components used for the training procedure. We use a generator based on SPADE [5] to produce in-painted RGB images, I_{RGB}, *as well as* a dense depth map by giving it access to an RGB canvas, hole-and-boundary map, the semantic canvas, the edges of blob instances and finally sparse depth maps and masks – which are shown and described further in Sec. IV-B and Fig. 3.

of pixel-value interpolation while later, [7] proposes an end-to-end image harmonisation architecture based on CNNs.

On the opposite side, lies end-to-end image synthesis, which has been enabled by the rapid advancement of GANs [8]. Initially unconditional and further conditional on latent representations, segmentation information and style encodings, GANs have been successfully applied to image-to-image translation and manipulation, with impressive qualitative results [5], [9]–[12]. One of the most common architectures is pix2pixHD [11] which allows photo-realistic synthesis of 2048×1024 images based on semantic maps, and enables changing of label classes in order to create new scenes. The architecture is further improved in terms of image quality by introducing techniques such as spatially-adaptive normalisation layers [5], segmenting discriminators and 3D-noise sampling to ensure multi-modality [12]. Although GANs are trained to match the distribution of natural images, the realism is still not ideal as the images contain artefacts or lack physical correctness, thus their applicability in robotic downstream tasks is limited.

A different approach to improve robotic tasks makes use of simulators. As a result of the improvements in specialised software and hardware for Computer Graphics (CG), a number of simulators such as GTA [13], Carla [14], Airsim [15], and Synthia [16] have been employed to simulate data from different modalities. While this approach offers access to virtually unlimited data, the process is tedious since it requires creation of environments, assets and scenarios, as well as careful control of parameters. Most disadvantageous, however, is the sim-to-real domain gap, making simulators less than ideal for data generation.

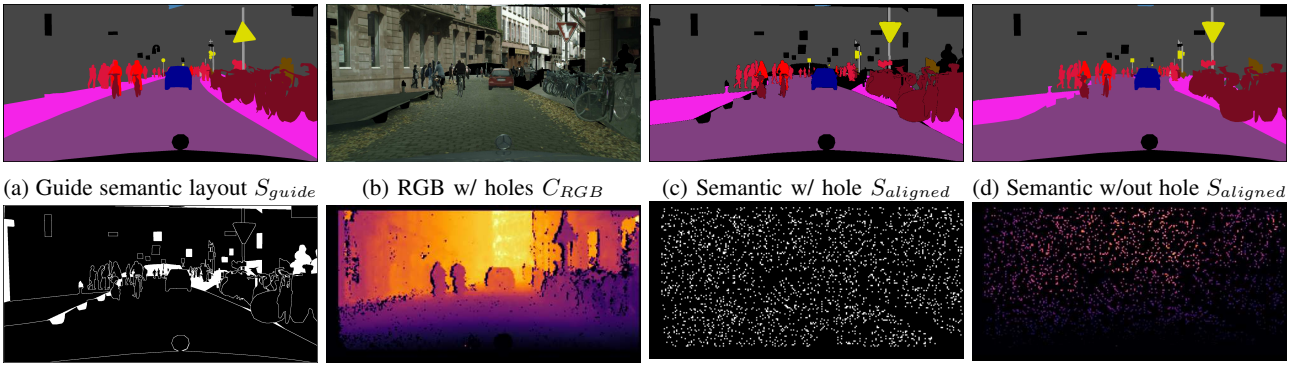
To this end, semi-parametric approaches combine the advantages of model-based and data-driven approaches by taking advantage of the existing data to enforce realism in appearance, and the situational diversity and controllability of simulation. In the context of 2D images, [17] employs two generative models that learn the shape of an object and a plausible, context-aware location in the segmentation map. For videos, GeoSim [18] performs geometry-aware image composition in which new urban scenarios are synthesised by adding dynamic objects (from a pre-built asset bank) on top of existing images, and further blending them in.

In-painting and Harmonization Although image compositing is a powerful synthetic data generation tool, the composed canvases are generally affected by missing regions or rough edges. To this end, image editing tasks such as image in-painting and completion have been used to fill in holes and smooth edges in a perceptually pleasing manner. For example, the architecture in EdgeConnect [19] successfully fills in missing regions with fine details, using two modules: a generator that first hallucinates missing edges of objects in the image, and a subsequent generator that fills in pixel values based on the output from the edge network.

In an autonomous-driving context, [20] makes use of depth information in order to guide a video in-painting task. The authors build a 3D map of frame-wise point clouds which are then projected onto frames, to generate a dense depth map that is further used to sample pixel colors. When removing a dynamic object from the scene, the area is then in-painted using pixel colors from adjacent frames. In contrast, [18] adds dynamic images on top of videos and further smooth the canvas using an in-painting GAN [21].

Depth estimation Geometry-based methods include Structure from Motion (SfM) where a sequence of 2D images is used to estimate 3D structures via feature matching. This approach is heavily reliant on feature matches which themselves rely on high quality image sequences [22]. On the other hand, stereo vision matching requires two viewpoints of the same scene to estimate the 3D structure via disparity maps. Both approaches include either image pairs or image sequences and are thus not applicable for our current setup where we compose a canvas from a single viewpoint.

Sensor-based methods (RGB-D cameras, LiDAR) are capable of retrieving pixel-level depth information but have a limited range and are affected by weather conditions, whereas monocular camera depth estimation has become



(a) Guide semantic layout S_{guide} (b) RGB w/ holes C_{RGB} (c) Semantic w/ hole $S_{aligned}$ (d) Semantic w/out hole $S_{aligned}$

(e) Hole-and-boundary map H (f) Guiding depth D_{guide} (g) Sparse depth mask M_{depth}^{valid} (h) Aligned valid depth $D_{aligned}^{sparse}$

Fig. 3: An overview of our canvasing procedure, described in more detail in Sec. IV-B. The semantic layout S_{guide}, shown in (a) is used to guide the retrieval and placement of blobs onto the RGB canvas C_{RGB}, shown in (b). The associated semantic footprints of the blobs are simultaneously pasted onto a corresponding semantic canvas $S_{aligned}$, shown in (c), which is further corrected by repairing holes due to mismatched target and original shapes, shown in (d). The missing areas in C_{RGB} are reflected in a hole-and-boundary map in (e), which is used to guide the in-painting network later on. The guiding depth in (f) (which corresponds to S_{guide}) is corrected by checking the consistency in class contents between $S_{aligned}$ and S_{guide}. Based on valid depth locations and a sampling probability, a sparse depth mask (shown in (g)) is created, based on which D_{guide} is sparsified, resulting in an aligned sparse depth map, shown in (h).

more popular due to lower weight and cost requirements.

Finally, due to the rapid advancement of deep learning, CNN-based methods have dramatically improved the accuracy of monocular depth estimation [23]–[26]. SOTA architectures such as PENet [27] make use of both image and sparse depth information in a two-branch approach. One branch is designed to exploit color information from the image and sparse depth via an encoder-decoder network with skip connections, while a second depth-dominant branch is designed to output dense depth from a sparse depth map and color information that is processed by the color-dominant branch.

IV. Methodology

We aim to synthesise an image and its corresponding pixel-wise segmentation and dense depth map, from initial guiding semantic layout and sparse depth, and a database of objects. The process is split into four steps: 1) the creation of an object blob database from a source set, 2) a canvasing step in which a series of compositing rules are used to position objects retrieved from the database onto a blank image canvas and a corresponding segmentation canvas, based on a guiding segmentation layout, 3) a sparse depth map composition step, where any existing depth cues are used to create a sparse depth map that corresponds to the newly created image canvas, 4) an in-painting step in which the image canvas is harmonised and the sparse depth map is densified. A graphical representation of the system can be found in Fig. 2.

A. Object blob database

Let N be an image dataset where for each image $I_i \in N$, object instances have been labelled through segmentation and instance masks S_i and classified as one of L classes. Our first objective is to decompose N into a dataset B of object blobs, where each class of objects is stored separately. More specifically, for each blob b_j in each image and segmentation pair $(I_i, S_i) \in N_{train}$ from the train set, we extract its corresponding binary mask m_j and label l_j; where $b_j \in \mathbb{R}^{h_j \times w_j \times 3}$, $m_j \in \{0,1\}^{h_j \times w_j \times 1}$ and $l_j \in L$, where $h_j \times w_j$ represent the dimension of the tight bounding rectangle inscribing the blob. Empirically, we ignore object blobs whose areas are smaller than 1000 pixels, and prefer larger, high-quality blobs that can then be resized. For object classes lacking instance segmentation, we attempt to obtain them by performing *connected components* [28].

Alongside each blob b_j, mask m_j, and label l_j from image I_i, we store its corresponding shape descriptor q_j. The original framework of [4] uses mIoU as a metric for blob matching however, this is difficult to scale to large numbers of blobs as any retrieval requires pair-wise comparisons over the entire dataset that belong to a particular class. To overcome this limitation, we use Hu moments [29] shape descriptors, which – beyond being invariant to translation, scale, rotation, and reflections – can be easily stored in a database and quickly compared, allowing faster retrieval.

To keep our assessment in line with previous works, we focus on the Cityscapes [2] dataset as a source of object blobs. However, any other dataset which provides instance segmentation can be used.

B. Canvasing

Elements of our canvasing procedure are shown in Fig. 3. In this second stage we compose both an image canvas $C_{RGB} \in \mathbb{R}^{w \times h \times 3}$ based on a guiding semantic layout S_{guide}, and a corresponding segmentation map $S_{aligned} \in \mathbb{R}^{w \times h \times 1}$ that is aligned to C_{RGB}.

The canvases C_{RGB} and $S_{aligned}$ are composed starting from blank canvases onto which objects from database B and their masks are pasted. Based on a guiding semantic layout S_{guide} that the compositing should follow, we retrieve blobs that have a similar shape to that of the object's footprints from the guiding segmentation layout S_{guide}. More specifically, for a segmentation map $S_{guide}^{i} \in N_{val}$ from the validation set, shown in Fig. 3(a), for each instance label with corresponding mask $(l_j, m_j) \in S_{guide}^{i}$, we compute a Hu moments shape descriptor and use it to find the closest blob $b_j \in B$ and paste it on the canvas C_{RGB}, shown in Fig. 3(b). Similarly, we paste the mask of the blob b_j onto the semantic

canvas $S_{aligned}$, shown in Fig. 3(c). Due to invariance to reflection of the shape descriptor based on Hu moments, we test the suitability of both orientations of the retrieved blob by flipping it horizontally and comparing the IoUs with the target footprint. Vertical flipping can also be used, but for most classes it is not applicable. As a general rule, we first place sky, buildings, road and pavement blobs, followed by the rest of the static objects and finally, the dynamic ones.

As there is often a mismatch between the target and original shapes, the RGB canvas C_{RGB} and the semantic canvas $S_{aligned}$ may contain holes. The RGB canvas holes will be in-painted by the in-painting model in a subsequent step using a hole-and-boundary map described below. The semantic canvas holes are repaired using a simple strategy:

1) For any holes that correspond to static classes in the original guiding semantic map S^i_{guide}, we simply copy over the semantic information from S^i_{guide};
2) For any remaining holes that do not correspond to static classes in the original guiding semantic map S^i_{guide}, we create a copy of the semantic canvas $S^{copy}_{aligned}$, dilate the static class footprints in this copy until no more holes exist, and finally transfer information corresponding to the location of the initial holes into $S_{aligned}$.

This strategy allows us to keep blobs corresponding to dynamic classes unmodified, but also assumes that static classes (road, sky, pavement, vegetation etc) are mainly texture-based and can be easily in-painted by the in-painting model. An example of the results of the hole-filling strategy can be seen in Fig. 3(d).

Finally, a hole-and-boundary map H (Fig. 3(e)) that encodes the location of any remaining holes in the RGB canvas along with the boundaries of all pasted blobs will be used as guide in the in-painting phase.

C. Sparse depth

In the third stage, we optionally create a sparse depth map. For a guiding semantic layout S_{guide} that has an available depth map D_{guide} (in Fig. 3(f)), we can also create a sparse depth map $D^{sparse}_{aligned}$ that corresponds to the newly generated RGB canvas C_{RGB} and its segmentation map $S_{aligned}$, as shown in Fig. 3(g).

In order to produce the sparse depth map $D^{sparse}_{aligned}$, we first compute the intersection between the original segmentation guiding layout S_{guide} and the segmentation map of the RGB canvas $S_{aligned}$, yielding a validity mask M^{valid}_{depth}, with the assumption being that only points from $S_{aligned}$ that have kept the same class as in S_{guide} are valid as a source of depth information. We then sample locations from the validity mask M^{valid}_{depth}, and produce a sparse depth map $D^{sparse}_{aligned}$ with depth information taken from the depth map D_{guide} of the guiding semantic layout $S_{aligned}$. The choice of a sparse depth map implies that both dense and sparse sources of depth (e.g. LIDAR) can thus be used with our method.

D. Image in-painting and depth synthesis

In the fourth and final stage we train a GAN-based in-painter to both synthesise a final RGB image by in-painting holes in the RGB canvas C_{RGB} and to densify the associated sparse depth map $D^{sparse}_{aligned}$. For this, we choose SOTA components for the pipeline, such as a SPADE-based in-painting generator [5], and a segmenting discriminator based on OASIS [12]. The overall architecture is shown in Fig. 2.

The in-painting generator takes as input the RGB canvas C_{RGB}, hole-and-boundary map H, semantic canvas $S_{aligned}$, edge map E (encoding the edges of blob instances, generated from the instance map), sparse depth map $D^{sparse}_{aligned}$ and sparse depth mask $M^{sparse}_{aligned}$ that indicates which depth values are valid. The in-painter then outputs both an in-painted RGB image I_{RGB} with no holes and a dense depth map $D^{dense}_{aligned}$ corresponding to the in-painted RGB image.

An intuitive explanation of the low FID scores (indicative of high image quality) of compositing approaches such as [4] is the fact that most of the areas in the generated images come from natural or real blobs. As such, our implementation also attempts to use as much information from the original RGB canvas as possible. Specifically, in our case, this means replacing blobs from the in-painted RGB output with blobs from the input RGB canvas using the hole-and-boundary mask to create a final RGB in-painted image. However, initial training experiments failed to converge, resulting in blurry in-painted regions, a possible explanation for this being the sparse gradients. The framework was therefore modified to alternatively pass both the raw output from the generator and the final in-painted and blended image to the discriminator.

The segmentation-based discriminator outputs a $L+1$ one-hot encoded prediction of the class of each pixel in the image. Similar to [12], we have L real classes and 1 fake class. We use a cross-entropy loss between the one-hot encoded prediction and the GT one-hot encoded segmentation map. Given real images x_R and canvas images x_C, the discriminator loss is:

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{(x_R)}\left[\sum_{c=1}^{L}\alpha_c\sum_{i,j}^{H\times W} t_{i,j,c}\log D(x_R)_{i,j,c}\right]\\ &-\mathbb{E}_{(x_C)}\left[\sum_{i,j}^{H\times W}\log D(G(x_C))_{i,j,c=L+1}\right]\end{aligned} \tag{1}$$

where i,j represent the pixel coordinates and c represents the channel. In this equation, $t_{i,j,c}=1$ if the pixel (i,j) belongs to the class c and 0 otherwise. The goal is for the discriminator to output a value of 1 at pixel location (i,j) in the corresponding channel of a particular class when the input is a real image, such that the first term $\rightarrow 0$. On the contrary, when the input to the discriminator is an in-painted canvas, the goal is for the discriminator to output a value of 1 in the $(N+1)$-th channel, thus signalling that the pixel is fake, minimising the second term.

The generator tries to minimize the following loss:

$$\mathcal{L}_G = -\mathbb{E}_{(x_C)}\left[\sum_{c=1}^{L}\alpha_c\sum_{i,j}^{H\times W} t_{i,j,c}\log D(G(x_C))_{i,j,c}\right] \tag{2}$$

From the point of view of the generator, the goal is to fool the discriminator, i.e. to output a value of 1 in the corresponding channel of a particular class when the input is a fake image, such that the term $\rightarrow 0$.

The weights $\alpha_c \in \mathbb{R}$ in Eqs. (1) and (2) are used to weigh

the loss according to the class balance of the example, with

$$\alpha_c = \frac{H \times W}{\sum_{i,j}^{H \times W} t_{i,j,c}} \tag{3}$$

Finally, for the depth densification task, we use a Mean Squared Error (MSE) loss between the predicted dense depth $D^{dense}_{aligned}$ and the GT depth D^{dense}_{GT}:

$$\mathcal{L}_{depth} = \|(D^{dense}_{aligned} - D^{dense}_{GT})^2\| \tag{4}$$

The final combined loss is: $\mathcal{L}_{total} = \mathcal{L}_D + \mathcal{L}_G + \lambda_d \mathcal{L}_{depth}$ where λ_d controls the strength of the MSE loss in relation to the other two terms.

E. Training details

In order to train the in-painting generator, we cannot directly use the outputs of our Canvaser, as a corresponding GT image and depth do not exist. Instead, starting from the original Cityscapes train set [2], we create a dataset of eroded RGB images and corresponding segmentation and sparse depth maps to emulate a canvas. For each input RGB image, we make use of its instance map to generate a boundary map, to which we add randomly-shaped masking polygons to simulate missing areas. This hole-and-boundary map H is further used to mask the RGB image. Such a step is necessary to mimic the canvasing process that would take place at inference time. To increase the generalisation ability, we dilate sections of map H using randomly-sized kernels.

The depth map is generated from the disparity image and then uniformly sampled with probability p_{sample} to generate a sparse depth map and a corresponding sparse depth mask. Finally, the edge map is created from the segmentation and instance maps.

V. Evaluation

All image synthesis models are trained with the Adam [30] optimizer for 50 epochs and a learning rate of 0.01. For our model, we set λ_d to 100.

A. Model evaluation

In line with previous work, we compare our model in terms of perceptual quality and alignment of synthesised images with their GT [12]. We choose strong SOTA baselines, such as pix2pixHD [11], SPADE [5], SPADE+, and OASIS [12] for the parametric approach and SIMS [4] for the semi-parametric approach. In measuring perceptual quality, we use FID [31], as it has been shown to be in line with human judgements. In measuring semantic alignment between generated data and GT labels, we employ a DRN-D-105 [32] network – pretrained for the task of semantic segmentation – and apply it to the generated images to measure the mIoU between the resulting segmentation and the GT segmentation masks, as in [12]. The model evaluation results are reported on the validation set of Cityscapes [2].

B. Image data evaluation

To test the usefulness of the synthesised image data as training data, we employ DRN-D-22 [32] as it is faster to train. We train our model, SIMS and OASIS on the Cityscapes train set, and evaluate them on the validation set to produce training data at resolution 512×256 [1]. We then train 3 individual instances of DRN-D-22 on the output of each of the models in the same training regime as above, and test them on the out-of-domain A2D2 dataset [3]. We report mIoU results.

C. Depth evaluation

To test the quality of our estimated dense depth associated with the in-painted images, we employ a light-weight depth-completion model, based on the architecture from Fig. 2. The network only takes as input the concatenation of an RGB image, sparse depth map and corresponding depth mask and outputs predicted dense depth. We make use of MSE loss between the aligned predicted dense depth $D^{dense}_{aligned}$ and GT dense depth D^{dense}_{GT}, as illustrated in Fig. 4.

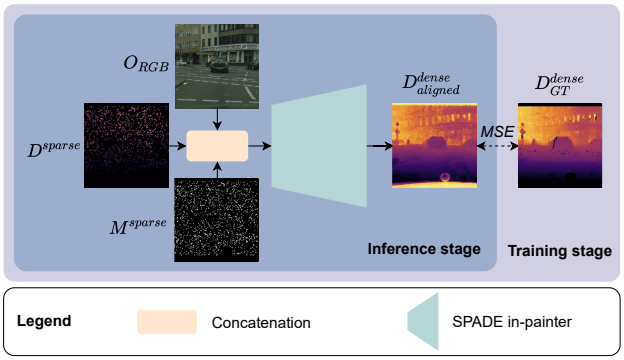


Fig. 4: We employ a light-weight depth-completion model based on a SPADE generator and use MSE loss for training.

We train 3 individual instances of our monocular depth network on the generated RGB images of SIMS, SPADE and our model, and evaluate them out-of-domain, on the validation split of the KITTI dataset [33]. Since the depth estimator is supervised with dense depth, we use the synthesised depth for our model as target, and the GT depth of Cityscapes for SIMS and SPADE, as they only synthesise RGB images. We report Root Mean Squared Error (RMSE) results.

VI. Results

Tab. I presents results from benchmarking the data generated by the model itself. Images produced by our method are better aligned with the GT segmentation, yielding a significantly higher mIoU score than all previous methods, approaching the performance of the real Cityscapes validation split. Additionally, it should be noted that, as with other methods that make use of a pretrained segmentation network, the resulting mIoU is also dependent on how closely the distribution of generated images follows the distribution of the dataset on which the segmentation network was trained. Although both SIMS and our method aim to preserve as much real blob components as possible, the higher mIoU indicates that our method benefits from better alignment, at the cost of a slight increase in FID, which nevertheless remains highly competitive, indicating that our method achieves a higher perceptual quality compared to parametric methods. We further conduct an ablation study to

[1] experiment equivalent to those reported in [4] "Cityscapes-fine"

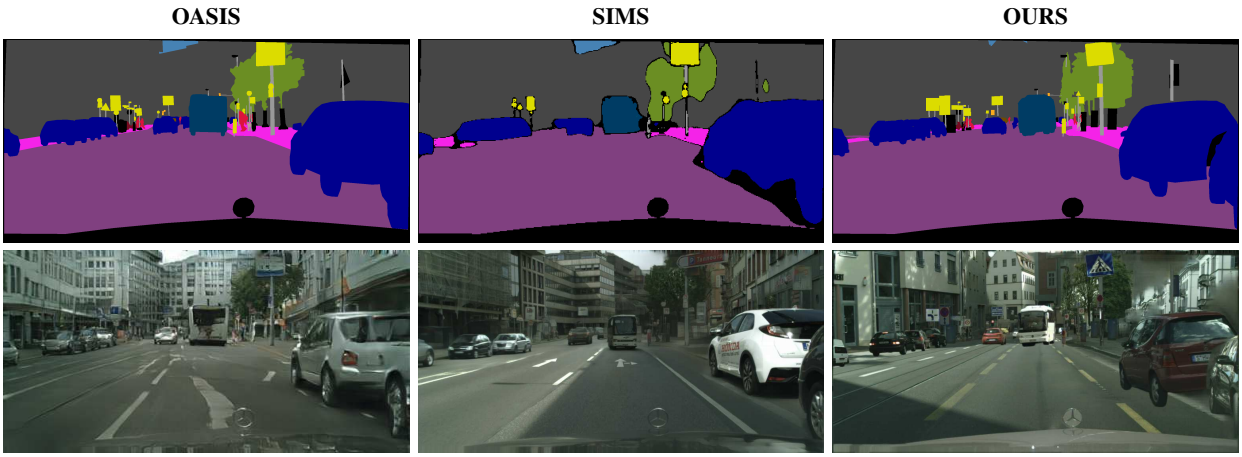


Fig. 5: RGB images and corresponding GT segmentation for OASIS, SIMS and our method. We note that OASIS has aligned semantic segmentation but lower image quality, SIMS has high image quality but imperfectly-aligned segmentation, while our method has both good image quality and well-aligned segmentation.

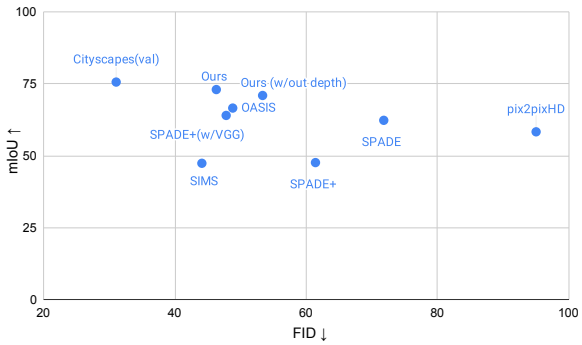


Fig. 6: Our model scores the highest in terms of mIoU, with a highly competitive FID. The performance of the real Cityscapes validation split is shown as an upper bound.

determine the usefulness of the depth completion component of our in-painter. Our approach benefits from the addition of depth completion, which both increases the mIoU by 2.1 percentage points and decreases FID by 7 points. Qualitative examples of image synthesis can be seen in Fig. 5 and a visual comparison of SOTA methods in Fig. 6.

Tab. II presents results from benchmarking the 2 downstream tasks trained using the data generated by the 3 models. We emphasise that this benchmarking is meant to rank the performance of synthetic training data rather than outperform SOTA segmentation and depth completion architectures. The segmentation task trained using data generated by our model produces better results when tested on an out-of-domain test set (A2D2) as compared to training using data produced by SIMS and OASIS. We interpret these results as a reflection of the importance of good alignment between generated images and segmentation maps when training a new task, as both our method (explicit segmentation alignment) and OASIS (implicit segmentation alignment) outperform SIMS in the segmentation task. In terms of depth completion, our method outperforms competing methods, when the depth synthesised by our in-painter is used to train the depth completion task. Conversely, and as a mini-ablation study, using the original guiding GT depth instead of our synthesised depth leads to worse results, possibly indicating that the synthesised depth is better matched to the generated images, compared to the original guiding GT depth.

Model	VGG	FID ↓	mIoU ↑
SIMS[2]	✓	49.7	47.2
SIMS[3]	✓	**44.1**	47.4
pix2pixHD[2]	✓	95.0	58.3
SPADE[2]	✓	71.8	62.3
SPADE+[2]	✓	47.8	64.0
	✗	61.4	47.6
OASIS[2]	✗	47.7	69.3
OASIS[3]	✗	48.8	66.6
Ours w/out depth	✓	53.3	70.9
Ours	✓	46.3	**73.0**
Cityscapes (val set)	-	31.05	75.6

TABLE I: Model benchmark by alignment and perceptual quality.

Model	mIoU ↑	RMSE (m) ↓
OASIS w/ GT depth	28.31	9.34
SIMS w/ GT depth	24.36	4.48
Ours w/ synthesised depth	**29.87**	**3.90**
Ours w/ GT depth	-	5.99

TABLE II: Train data benchmark on semantic segmentation and depth completion.

VII. Conclusions

In this paper we have presented a compositing image synthesis method that makes use of a set of simple heuristics to compose not only an RGB image canvas but also a corresponding well-aligned semantic segmentation map and a sparse depth map, followed by an in-painting stage that yields a high quality RGB image and a dense depth map. We successfully benchmark the quality of the data produced by the model, showing an increase of 3.7 percentage points for mIoU over other SOTA parametric and semi-parametric approaches and a highly competitive FID score. Additionally, we benchmark the suitability of our synthesised data as training data for semantic segmentation and depth completion, showing that the data produced by our model is more suitable for this purpose than the other methods.

[2]Results quoted from [12] (Table 1, page 8).
[3]Results of our own experiments, with the official implementation

REFERENCES

[1] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi Autonomous Driving Dataset," 2020. [Online]. Available: https://www.a2d2.audi

[4] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8808–8816, 2018.

[5] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[6] J. Martino, G. Facciolo, and E. Meinhardt-Llopis, "Poisson image editing," *Image Processing On Line*, vol. 5, pp. 300–325, 11 2016.

[7] Y.-H. Tsai, X. Shen, Z. L. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2799–2807, 2017.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, Oct. 2020. [Online]. Available: https://doi.org/10.1145/3422622

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

[10] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251.

[11] T. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018.

[12] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, "You only need adversarial supervision for semantic image synthesis," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=yvQKLaqNE6M

[13] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[15] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[16] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Context-aware synthesis and placement of object instances," in *NeurIPS*, 2018.

[18] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *CVPR*, 2021.

[19] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edge-connect: Generative image inpainting with adversarial edge learning," 2019.

[20] M. Liao, F. Lu, D. Zhou, S. Zhang, W. Li, and R. Yang, "Dvi: Depth guided video inpainting for autonomous driving," in *ECCV*, 2020.

[21] J. Yu, Z. L. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4470–4479, 2019.

[22] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Science China Technological Sciences*, pp. 1–16, 2020.

[23] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[24] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv e-prints*, vol. abs/1812.11941, 2018. [Online]. Available: https://arxiv.org/abs/1812.11941

[25] S. Pillai, R. Ambrus, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," 05 2019, pp. 9250–9256.

[26] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[27] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Penet: Towards precise and efficient image guided depth completion," 2021.

[28] C. Fiorio and J. Gustedt, "Two linear time union-find strategies for image processing," *Theoretical Computer Science*, vol. 154, no. 2, pp. 165–181, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304397594002622

[29] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[31] 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," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6629–6640.

[32] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Depth-SIMS: Semi-Parametric Image and Depth Synthesis
Publication Status	Published
Publication Details	**V. Musat**, D. De Martini, M. Gadd and P. Newman, "Depth-SIMS: Semi-Parametric Image and Depth Synthesis," *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 2022, pp. 2388-2394, doi: 10.1109/ICRA46639.2022.9811569.

Student Confirmation

Student Name:	Valentina Musat		
Contribution to the Paper	- developed the idea supporting the paper under the guidance of Dr. De Martini and Dr. Gadd, and supervision of Prof. Newman - implemented the architecture - compiled the data for training - ran all experiments and interpreted data - wrote the manuscript - prepared presentation materials and a video presentation for the conference		
Signature 		Date	21.04.2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman		
Supervisor comments I certify that the candidate made a substantial and leading contribution to the publication, and that the description described above is accurate		
Signature 	Date	24/04/25

5.3 Further insights

While the proposed method offers the ability to compose and synthesise multi-modal outputs, the model employed is a rather shallow generative model as most of the complexity associated with the RGB output consists in blending or harmonising edges. Since the year of publication, the field of image synthesis has progressed at a rapid pace, with diffusion models and latent diffusion models taking center stage, enabling both stable training and improved results. Later studies such as (Blattmann et al., 2022) use exemplars retrieved from a database to further condition a text-to-image model during training. However, this method does not generate any associated ground-truth or depth maps, making it less suitable for producing training data for supervised autonomous driving-specific tasks.

The approaches through which the method presented in this chapter could be improved have both advantages and disadvantages. For example, the main architecture could be swapped with a ControlNet (L. Zhang et al., 2023), T2I-Adapter (Mou et al., 2024) or Ctrl-Adapter (H. Lin et al., 2025), however these architectures in turn will need to be adapted to support multi-modal output (depth, in this case) with pixel-wise alignment.

The current retrieval mechanism could further be improved by considering appearance embeddings when matching candidate blobs to query blobs, not just shape as described by Hu moments. This could subsequently help when performing post-hoc domain adaptation by changing the database of blobs to one that best matches the target domain.

On the other hand, performance will still be limited by the heuristics used to compose the input canvas, which is unfortunately a characteristic of all methods that employ heuristics. Alternatively, a latent diffusion model could be used to generate ground-truth (semantic segmentation, depth) using text-prompt conditioning, but granular control of object placement would be difficult. More descriptive conditioning such as 2D bounding boxes or object centroids could also be used in conjunction with text prompting to improve control.

Finally, the two approaches presented in chapters 4 and 5 could further be combined to improve coverage, by first synthesizing images, depth maps and segmentation maps using Depth-SIMS, and subsequently making use of Multi-weather-city to add various levels of illumination and weathers to the synthesised data.

6

NeuralFloors

Contents

6.1 Contribution

Many approaches, including Depth-SIMS (Musat et al., 2022), that synthesize images - with or without ground-truth annotations - rely on relatively detailed conditional inputs such as semantic segmentation maps. While simulation is a good alternative for sourcing this data, it still requires that environments are built with some amount of fidelity, and populated with assets, which is often expensive, time-consuming and suffers from the sim-to-real gap. To unlock the scalability of data synthesis, as models grow larger and require more data, alternative conditional inputs can be used,

such as topological maps or 2D BEV representations, which have gained traction as they offer a more comprehensive view of the environment surrounding the ego-vehicle.

This method presents a framework for generating ground-view images, along with semantic segmentation and depth maps, from BEV semantic representations of environments. The key contributions of the paper include:

1. A two stage approach to data synthesis, where the first stage learns priors over the 3D structure of the world, while the second stage learns priors over the visual appearance of the environment;

2. An approach to 2D-to-3D lifting of BEV maps using a neural field model conditioned on a BEV semantic segmentation map. An encoder transforms the BEV map into a tri-plane representation, while an MLP learns to lift the features from the three planes into a 3D volume, from which ground-view data under varying poses can be generated using a volumetric renderer;

3. The additional synthesis of ground-view semantic segmentation, instance and depth maps that are well-aligned with the generated images;

4. Experiments on 3 representative urban driving datasets that demonstrate both the perceptual quality of the generated images, and the high degree of spatial alignment of the synthesized ground-truth as opposed to prior art.

6.2 Integrated manuscript

The manuscript was published in IEEE Robotics and Automation Letters, 2023 (Muşat et al., 2024a)

NeuralFloors: Conditional Street-Level Scene Generation From BEV Semantic Maps via Neural Fields

Valentina Muşat, Daniele De Martini, *Member, IEEE*, Matthew Gadd, *Member, IEEE*, and Paul Newman, *Fellow, IEEE*

***Abstract*—Semantic Bird's Eye View (BEV) representations are a popular format, being easily interpretable and editable. However, synthesising ground-view images from BEVs is a difficult task as the system would need to learn both the mapping from BEV to Front View (FV) structure as well as to synthesise highly photo-realistic imagery, thus having to simultaneously consider both the geometry and appearance of the scene. We therefore present a factorised approach that tackles the problem in two stages: a first stage that learns a BEV to FV transformation in the semantic space through a Neural Field, and a second stage that leverages a Latent Diffusion Model (LDM) to synthesise images conditional on the output of the first stage. Our experiments show that this approach produces RGB images with a high perceptual quality that are also well aligned with their corresponding FV ground-truth.**

***Index Terms*—Deep learning for visual perception, computer vision for transportation, neural rendering, cross-view transformation, data-driven simulation.**

I. Introduction

SOFTWARE-STACK test and validation in Autonomous Driving (AD) is paramount for the safety of passengers and other traffic participants. The ability to generate and test traffic scenarios with a high degree of control, complexity, variability and fidelity is essential for identifying potential points of failure before deployment. As a result, research in this area has gained popularity in both academia [1] and industry, where there has been much focus on building digital twins endowed with the diversity and high fidelity of the real world they are emulating.

Synthetic 2D and 3D worlds have been simulated through either data- or model-based approaches. Model-based approaches such as 3D simulators support high complexity and diversity but are computationally expensive and require detailed knowledge of the environment being simulated [2]. Conversely, data-driven approaches allow high realism but suffer from poor scalability and diversity since they rely on real-world data acquisition, which does not capture the full range of possible scenes or situations.

Manuscript received 4 August 2023; accepted 14 December 2023. Date of publication 22 January 2024; date of current version 31 January 2024. This letter was recommended for publication by Associate Editor M. Liu and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported by the DeepMind Engineering Science Scholarship and the EPSRC Programme "From Sensing to Collaboration" under Grant EP/V000748/1. *(Daniele De Martini and Matthew Gadd contributed equally to this work.) (Corresponding author: Valentina Muşat.)*

The authors are with the Mobile Robotics Group (MRG), University of Oxford, OX13PJ Oxford, U.K. (e-mail: valentina@robots.ox.ac.uk; daniele@robots.ox.ac.uk; mattgadd@robots.ox.ac.uk; pnewman@robots.ox.ac.uk).

Digital Object Identifier 10.1109/LRA.2024.3356793

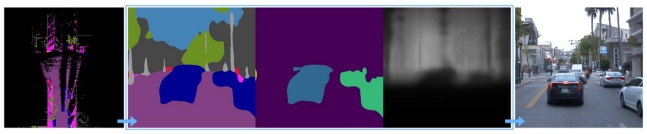


Fig. 1. *NeuralFloors* synthesises high-quality FV imagery (right) from a BEV segmentation map (left) in two steps: one generates FV semantic, instance and depth maps (centre), and the following synthesises an RGB image from the output of the first step.

Concurrently, Bird's Eye View (BEV) semantic maps have become popular because they are simple, compact yet rich representations of traffic scenes, enabling easy visualisation, inspection and a high degree of editability, characteristics that make them ideal inputs for an AD simulator. As such, we present a system able to generate realistic Front View (FV) RGB images of real-world urban traffic scenes, coupled with well-aligned FV semantic, instance, and depth maps, starting from their semantic Bird's Eye View (BEV) representations – i.e. 2D top-down semantic maps, as depicted in Fig. 1. Nevertheless, to synthesise an FV RGB image directly from BEV semantic maps, a model would need to learn the mapping from BEV to FV structure *and* to synthesise highly photo-realistic images, accounting for both geometric structure and appearance simultaneously.

In this context, Neural Radiance Fields (NeRFs) are attractive as they leverage the intrinsic geometry of a ray-based sampler following a simple camera model. In contrast to a vanilla NeRF approach, which optimises a Multi-Layer Perceptron (MLP) to render a particular scene, we take inspiration from Generative Scene Networks (GSN) [3], which uses the points along rays to sample features from a latent floorplan generated unconditionally; however, as opposed to their unconditional model based on Gaussian noise inputs, we condition our model on BEV segmentation maps to enable visualisation, interpretability, and improved control over the scene contents. The advantage of this approach is that once the model is trained, certain scene editing, synthesis abilities and scalability are enabled by manipulating the 2D BEV floorplan.

While the world could be represented using 3D information, e.g. voxels in GANcraft [4], we opt for a flat 2D representation of our input for simplicity and scalability. The rationale is that incorporating 3D information requires extra effort and tools in modelling semantic scenarios, whilst our goal is to alleviate such

requirements while preserving output quality and usefulness. Similarly, a contemporary approach such as InfiniCity [5] uses an intermediate 3D voxel representation – this is flexible at inference time but necessitates a separate training procedure with expensive-to-acquire information, e.g. CAD models.

The closest work to ours is BEVGen [6], an autoregressive model based on a Visual Transformer able to directly generate consistent panoramic images from a BEV segmentation map, source input views and camera parameters. The main difference is that they learn an implicit geometric transformation using an attention mechanism, whereas we use an explicit geometric mechanism based on the ray sampler and volumetric renderer. We factorise the RGB-image generation in two steps, one that learns the scene structure - generating FV semantic, instance and depth maps, and one that learns to generate color and appearance from the generated maps. We exploit separate architectures suitable for these two distinct purposes – Neural Fields and Diffusion Models. The maps generated in stage 1 are thus aligned with the RGB images, which makes them suitable to be used as training data for AD downstream tasks.

To summarise, we tackle the cross-view and cross-modality image synthesis task in an urban traffic scene context. Our contributions are:

- A two-stage approach to generate FV RGB images of complex driving scenes from BEV semantic maps, using Neural Fields for 2D-to-3D lifting of semantic information and geometric projections, and Diffusion for high-quality image synthesis;
- The additional generation of FV segmentation, depth, and instance maps well-aligned with generated images.

We test our system on the KITTI-360 [7], nuScenes [8], [9], and Waymo [9], [10] datasets against a set of baselines and analyse the different components of our approach through ablation studies. The method has applicability both as a system for generating diverse training and testing/validating data for perception, prediction, path planning etc., but also as a data-driven simulator that can generate a wide distribution of structures and styles without requiring expensive and often manual 3D asset generation.

II. Related Work

A. Model-Based vs Data-Driven Simulators

Creating visual content is important, especially in the context of AD, where software-stack validation can be done without exposure to risks in the wild. Two major paradigms are model/physics-based and data-driven simulators. The first, such as Carla [11], have the advantage of providing perfect Ground-Truth (GT) signals for a variety of sensors. At the same time, high visual consistency and control make them desirable for scene manipulation. However, assets and scenes require tedious and time-consuming manual creation and the simulated world suffers from the sim-to-real gap [12]. On the other hand, data-driven simulators favour realistic sensor readings, typically involving generative models that learn the data distribution. In the 2D setting, a pivotal example is pix2pixHD [13], in which a Generative Adversarial Network (GAN) is trained to generate urban street-level photo-realistic images from semantic segmentation maps.

Semi-parametric approaches combine the advantages of model- and data-based approaches, such as SIMS [14] and Depth-SIMS [15], which use pre-extracted image segments to compose new images and depth from segmentation masks. In the 3D setting, GeoSim [16] embeds learnt elements in a geometry-aware framework that allows vehicles to be added through composition, while [17] generate new data by geometrically reprojecting different sensors to new viewpoints.

B. Neural Radiance Fields

The seminal work on NeRFs [18] proposed to encode a (static) scene as a continuous volumetric function parametrised by a MLP, yielding per-point colours and densities, rendered into images using a differentiable renderer, conditioned on camera poses. While the approach is highly photo-realistic, it cannot render unseen environments or rearrange objects, as each scene is optimized individually. To overcome this limitation, PixelNeRF [19] additionally conditions the model on features extracted from the input views using an off-the-shelf pre-trained network; EG3D [20] additionally encodes a 3D object into a "tri-plane" representation while GSN [3] conditions the radiance field with a latent "floor plan" obtained from Gaussian noise. GIRAFFE-HD [21] recreates scenes by incorporating compositional 3D scene structure into the generative model, where each object's synthesis can be controlled in shape, appearance and 3D position. Finally, other methods [22] incorporate semantic information, either as an input to the model, or output.

C. Cross-View and Cross–modality Image Synthesis

The reprojection of sensor inputs onto a 2D ground plane, aka BEV, and back to FV is of great interest in the AD community due to its advantages for planning and navigation problems. For instance, [23] use a satellite RGB image to condition the in-painting process of a forward-facing semantic map given as input. In contrast, our goal is to synthesize forward-facing RGB images based on BEV semantic maps, which is both a task of generating a feasible structure, and an RGB image synthesis task.

Most similar to our work, BEVGen [6] learns to synthesize spatially-consistent FV images through a spatial-attention design that encodes the relationships between the map and cameras. While their inputs are BEV segmentation maps, source view images, and spatial embeddings derived from camera parameters, our method is instead conditional only on a BEV segmentation map, while camera parameters are directly used to define the image formation process, including pose, resolution, aspect ratio and field of view. Another related work, InfiniCity [5], can synthesise street-view images of a city environment from a 3D voxelised representation trained using semantic, depth, normals and RGB images, and relies on CAD models. However, this method presents a limitation as street assets are not tackled; thus, editability is limited. In contrast, our approach can model both street furniture, vehicles and pedestrians.

III. Method

We formulate our setup as a two-stage approach, with the first learning a mapping from BEV segmentation maps to FV segmentation maps ($BEV \rightarrow FV$) and the second synthesising FV photo-realistic images from FV segmentation maps ($FV \rightarrow FV$). This modular approach has the added benefit of allowing training each stage with different data sources, reducing the difficult requirement for paired BEV segmentation and FV RGB

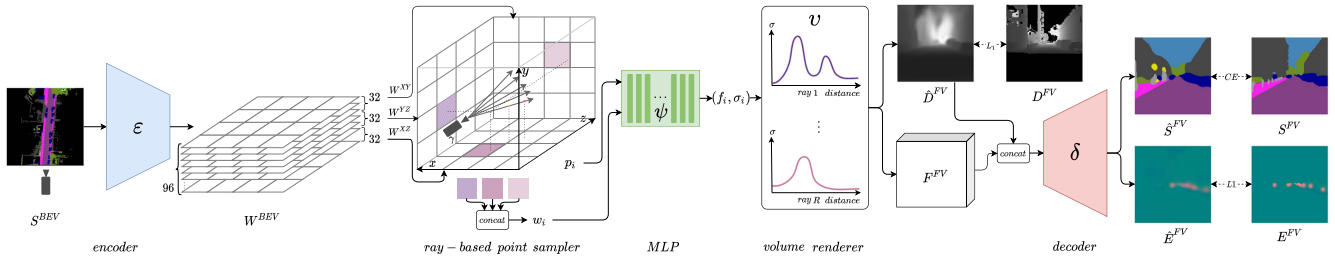


Fig. 2. **Stage 1 of our approach.** BEV segmentation S^{BEV} is encoded by ε into a 2D latent representation W^{BEV} from which features w_i are sampled. An MLP ψ produces lifted features f_i and densities σ_i corresponding to 3D coordinates p_i. Volume renderer υ further produces FV feature and depth images: F^{FV} and $\hat{D}^{FV}$. Finally, FV segmentation and instance maps are produced by decoder δ.

TABLE I
NOTATION

Symbol	Description
$N \in \mathbb{N}$	total number of classes
α_n	weight of class n
$R \in \mathbb{N}$	total number of rays
S^{BEV}	GT BEV semantic map
S^{FV}	GT FV semantic map
D^{FV}	GT FV depth map
I^{FV}	GT FV RGB image
E^{FV}	GT FV instance map
C^{FV}	GT FV X & Y instance pixel center map
O^{FV}	GT FV X & Y instance pixel offset map
W^{BEV}	BEV latent feature map
$[t_n, t_f]$	near and far planes for point sampling
$p_i \in \mathbb{R}^3$	point along ray
$w_i \in W^{BEV}$	latent feature of projected point p_i
(f_i, σ_i)	non-integrated output feature and density of p_i
F^{FV}	integrated front view feature map
ε, δ	encoder, decoder networks (stage 1)
κ, μ	encoder, decoder networks (stage 2)
γ	ray-based volume sampler
ψ	MLP neural field function
υ	volume renderer
π, τ	generator and discriminator
ϕ	LDM extra layers

In the text, terms with the symbol ˆ represent the predicted counterpart of the GT term.

images. We discuss in the remainder of this section the two stages separately. Please refer to Table I as it summarises the notation used.

A. $BEV \rightarrow FV$ Semantic Lifting

As the first stage tackles a geometric problem, we chose a Neural Fields approach for their geometrically-correct representation of the image formation process using a camera model.

This stage thus leverages a ray-based point sampler and an MLP to lift features from a 2D latent floorplan into a 3D space and a volumetric renderer to render them onto a FV feature image, which is then decoded into instance and segmentation maps. The latent floorplan feature map W^{BEV} is conditional on an encoded BEV segmentation map S^{BEV}: $W^{BEV} = \varepsilon(S^{BEV})$.

A ray-based point sampler γ following a simple pinhole camera model is then used to sample 3D points p_i along rays within a 3D volume above the latent floorplan. The 3D points are projected onto the floorplan to yield 2D coordinates, which are then used to sample a per-point feature $w_i = \gamma(p_i, W^{BEV})$ via bilinear interpolation, as in GSN [3].

A neural field function ψ (MLP) is then used to transform each latent floorplan feature w_i, conditioned on its 3D coordinate p_i (with positional encoding $\rho(p_i)$ applied), yielding a lifted feature and a corresponding density $(f_i, \sigma_i) = \psi(p_i \oplus \rho(p_i), w_i)$, where the symbol $\oplus$ refers to concatenation.

A key aspect is that the neural field function learns to lift 2D features into a 3D volume of transformed features. In Fig. 2, for points that project onto the same (x, z) coordinate of the latent floorplan, the ray sampler will pick the same feature w_i irrespective of the points' height coordinate y. Thus, for each group of points that project to the same location, the neural field ψ learns to map latent feature w_i and height y to a transformed feature f_i associated with a 3D coordinate.

While the basic method proposed treats W^{BEV} as the scene floorplan only, we extend this to a tri-plane representation similar to EG3D [20], where the feature map is reshaped into 3 orthogonal planes: $W^{BEV} = \{W^{XZ}, W^{XY}, W^{YZ}\}$, where W^{XZ} represents the floorplan, while W^{XY} and W^{YZ} are two extra feature maps. For this case, the point sampler γ will return 3 corresponding features which are then concatenated to form $w_i = w_i^{XZ} \oplus w_i^{XY} \oplus w_i^{YZ}$.

A volume renderer υ is used to integrate the features f and densities σ across each ray $r \in R$ to obtain final FV feature and depth images $(F^{FV}, \hat{D}^{FV}) = \upsilon(f, \sigma)$.

The feature image F^{FV} is obtained by integrating features weighted by their corresponding densities along each ray:

$$F^{FV} = \int_{t_n}^{t_f} T(t)\sigma(t)f(t)\,dt \tag{1}$$

Similar to GSN [3] the depth image $\hat{D}^{FV}$ is obtained by integrating point depths t (along the ray) weighted by their corresponding densities:

$$\hat{D}^{FV} = \int_{t_n}^{t_f} T(t)\sigma(t)t\,dt \tag{2}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(s)\,ds)$ is the accumulated transmittance from the near-plane t_n to a point t, as a function of densities along each ray. The densities are used to weigh the delta distances between depth planes, which are then summed into a depth map, on which depth loss is applied.

Finally, a decoder δ decodes the concatenation of the feature image F^{FV} and predicted depth map $\hat{D}^{FV}$ into a FV panoptic segmentation map, as a one-hot segmentation map and an instance map $(\hat{S}^{FV}, \hat{E}^{FV}) = \delta(F^{FV})$, where, similar to [24], the instance map E^{FV} is given by instance centres map C^{FV} (a one-hot encoding that indicates the centres of mass of each

instance blob) and instance pixel offsets map O^{FV} (for each instance blob, the offset of each blob pixel with respect to its centre) on which L1 loss is applied:

$$\mathcal{L}_C = |C^{FV} - \hat{C}^{FV}|;\ \mathcal{L}_O = |O^{FV} - \hat{O}^{FV}| \tag{3}$$

To speed up convergence and stabilize training [3], [25], we apply a masked L1 loss on the predicted depth:

$$\mathcal{L}_D = M^D \odot |D^{FV} - \hat{D}^{FV}| \tag{4}$$

where $M^D\odot$ represents an elementwise masking operation which masks out the loss if D_{FV} exceeds $(t_n, t_f]$.

For the segmentation map, we use cross-entropy loss:

$$\mathcal{L}_S = -\mathbb{E}\left[\sum_{n=1}^{N} \alpha_n \sum_{i,j}^{H\times W} h_{i,j,n} \log \hat{S}^{FV}_{i,j,n}\right] \tag{5}$$

where $h_{i,j,n} = 1$ if the pixel (i, j) belongs to class n in the GT segmentation map S^{FV} else 0, and α_n a weight balancing each class, with higher weight for rare classes.

We train the encoder ε, decoder δ and neural radiance field ψ end-to-end, combining the losses above.

Additionally, for the single-stage ablations, we use a combination of image reconstruction (L1) loss $\mathcal{L}_I$ and adversarial loss $\mathcal{L}_A$:

$$\mathcal{L}_I = |I^{FV} - \hat{I}^{FV}| \tag{6}$$

$$\mathcal{L}_A = \mathbb{E}[\log \tau\left(I^{FV}\right)] + \mathbb{E}[\log\left(1 - \tau\left(\pi\left(S^{BEV}\right)\right)\right)] \tag{7}$$

where $\pi = \delta \circ \upsilon \circ \psi \circ \gamma \circ \varepsilon$ represents the generator.

B. $FV \rightarrow FV$ Image Synthesis

Inspired by the recent success of Latent Diffusion Models (LDMs) [26], the second stage synthesises FV photo-realistic images conditional on the FV segmentation, instance and depth maps generated by the first stage.

Diffusion Models (DMs) work in two steps: first, a forward diffusion process adds t steps of Gaussian noise ϵ to an input x to obtain a noisy version of the input x_t. Secondly, a reverse diffusion process takes the noisy input x_t and learns to predict back the added noise $\hat{\epsilon}$, which is then subtracted from the noisy input, to recover the original input, with a self-supervised loss between the predicted noise $\hat{\epsilon}$ and the GT noise ϵ:

$$\mathcal{L}_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \hat{\epsilon}||_2^2\right] \tag{8}$$

where a neural network ϵ_θ predicts the noise $\hat{\epsilon} = \epsilon_\theta(x_t, t)$.

While image DMs are applied to the original, high-dimensional pixel space, LDMs work in a more computationally-efficient lower-dimensional latent space z. A widely used approach is to train a VAE [26], composed of an encoder κ and a decoder μ, to encode images into latent representations z, with $z = \kappa(I)$ and $\hat{I} = \mu(z)$. The training time objective of [26] is:

$$\mathcal{L}_{LDM} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \hat{\epsilon}||_2^2\right] \tag{9}$$

where $\hat{\epsilon} = \epsilon_\theta(z_t, t)$ is the predicted noise, the latent $z = \kappa(x)$ is obtained by encoding the original input x through the latent encoder κ, and z_t is the noisy latent. We extend the network ϵ_θ to additionally be conditional on $z_c = \phi(S^{FV}, E^{FV}, D^{FV})$, a learned latent encoding of the maps generated during our first stage, with $\hat{\epsilon} = \epsilon_\theta(z_t, z_c, t)$.

TABLE II
BASELINES VS. NEURALFLOORS ON KITTI-360 DATASET

Model	FID↓ @64	FID↓ @512	KID↓ @64	KID↓ @512	mIoU↑	mIoU-align↑	RMSE↓
(GSN)	187.39	-	0.2150±0.0037	-	-	-	-
(SPADE)	203.66	325.31	0.2351±0.0023	0.4137±0.0034	-	8.14	-
(LDM)	**41.10**	68.75	**0.0241±0.0009**	0.0343±0.0010	-	20.55	-
(LDM+LDM)	48.01	77.52	0.0475±0.0018	0.0684±0.0027	14.12	13.03	-
(Neural+SPADE)	76.31	96.80	0.0532±0.0013	0.0631±0.0015	**32.06**	20.33	6.897
(Neural+LDM)	42.23	**65.81**	0.0266±0.0011	**0.0322±0.0012**	**32.06**	**28.11**	6.897

At inference time, z_t is sampled from a normal distribution, the neural network ϵ_θ is applied to remove noise from z_t, and the VAE decoder μ is used to decode the denoised latent representation into an image $\hat{I} = \mu(z_t - \hat{\epsilon})$.

IV. EXPERIMENTAL SETUP

A. Data

Since our work focuses on an application in autonomous urban driving, we train and test 3 independent experiments on KITTI-360 [7], nuScenes [8], [9] and Waymo Open Dataset [9], [10].

1) KITTI-360: In order to generate semantic BEVs, we make use of the provided accumulated point clouds, and pair each constructed BEV with its corresponding FV data. However, since objects in motion do not appear in the accumulated point clouds but do in the FV segmentation map and RGB image (e.g. walking pedestrians and moving cars, while parked static cars remain valid), we curate the dataset by removing inconsistent BEV-FV pairs to prevent the model from "hallucinating" nonexistent objects. Out of the 9 scenes featured in KITTI-360, we employ the first 8 for training, and reserve the last for model selection and testing. This results in approximately 22300 train observations, 500 observations for model selection and 2200 testing observations.

2) Nuscenes: To generate BEVs, we make use of the Occ3D Large Scale Dataset [9], which provides a voxelised and accumulated representation on the original nuScenes [8] dataset. As opposed to KITTI-360, objects in motion are present in both the BEV and FV data. We use a similar splitting strategy as above, yielding 20357 total pairs, out of which 16674 are used for training and 3683 are reserved for model selection and testing.

3) Waymo: Similar to nuScenes, we make use of the same Occ3D Large Scale Dataset [9] to generate BEVs. We use a similar splitting strategy as above, yielding 39791 total pairs, out of which 31801 are used for training and 7990 are reserved for model selection and testing.

As neither nuScenes nor Waymo provide pixel-wise semantic and instance segmentation for FV data, we use off-the-shelf Panoptic-DeepLab [24] to provide panoptic pseudo-GT.

For all 3 datasets, the BEV map covers an area of $80\,\mathrm{m} \times 80\,\mathrm{m}$, representing the scene in front of the camera, thus, we assume the camera origin to be in the middle of the bottom edge of the BEV and latent floorplan. For KITTI-360 experiments (Tables II, III), we use KITTI-360 camera intrinsics. For experiments on nuScenes and Waymo (Table IV) we use their respective camera parameters.

For stage 2 of our factorised approach, we provide RGB images from each of the 3 datasets and FV segmentation maps (GT or pseudo-GT) and, additionally, Cityscapes's [27] train set (2975 images). Our two stages are trained separately, this setup allowing to relax the assumption of synchronisation between BEV segmentation and FV RGB images.

TABLE III
SINGLE STAGE BASE EXPERIMENTS ON KITTI-360 DATASET

Model	FID↓ @64	FID↓ @512	KID↓ @64	KID↓ @512	mIoU↑	mIoU-align↑	RMSE↓
(Base 1)	300.35	285.12	0.3204±0.0027	0.2783±0.0027	-	6.25	6.782
(Base 2)	309.30	296.10	0.3427±0.0029	0.3045±0.0032	28.90	7.12	6.616
(Base 3)	95.32	157.27	0.0714±0.0012	0.1428±0.0023	30.22	10.93	6.356
(Base 4)	85.52	125.24	0.0540±0.0012	0.1031±0.0019	29.64	14.72	6.544
(SPADE B1)	115.58	177.89	0.0795±0.0008	0.0991±0.0009	-	10.84	6.751
(SPADE B2)	101.78	141.77	0.0645±0.0008	0.0968±0.0014	28.85	13.25	6.058
(SPADE B3)	96.06	143.37	0.0558±0.0006	0.0977±0.0010	29.20	11.83	6.339

TABLE IV
NEURALFLOORS ON NUSCENES AND WAYMO DATASET

Model	FID↓ @64	FID↓ @512	KID↓ @64	KID↓ @512	mIoU↑	mIoU-align↑	RMSE↓
nuScenes	15.02	21.48	0.0036±0.0003	0.0061±0.0005	36.02	25.84	8.421
Waymo	19.10	22.27	0.0091±0.0009	0.0102±0.0010	28.14	20.01	6.193

B. Metrics

In order to measure the quality of the generated output, we follow prior art and compare the perceptual quality of the generated images using (1) the Fréchet Inception Distance (FID) [28] and (2) Kernel Inception Distance (KID) [29] at resolutions of 64×64 (the output size of GSN [3]) and at 512×512 (the output size of LDM [26]). Secondly, to measure the ability to reproduce scene structure, we compare the predicted FV segmentation with the GT segmentation map in terms of Mean Intersection Over Union (mIoU) to understand how accurately the scene structure has been reconstructed. Similarly, we report the Root Mean Squared Error (RMSE) (in meters) between predicted and GT depths.

Finally, we report mIoU-align, which we define as the mIoU between the segmentation maps extracted by an off-the-self segmentation model (Deeplabv3+ [30]) – from both the GT and predicted FV RGB images. We choose to apply the segmentation model on both GT and predicted outputs because we want to test the images' alignment without introducing undesired variance due to the performance of the off-the-shelf segmentation model itself.

We report mIoU and mIoU-align to allow comparison to prior art, but we highlight that these metrics are not ideal for our case where one BEV topology may have multiple geometrically correct and plausible FV outputs, as they constrain the output to the configuration of the GT. BEVGen [6] instead reprojects the FV outputs onto the BEV plane using an off-the-shelf BEV segmentation network, but this incurs a significant drop in performance due to the network itself. We believe that, going forward, a promising alternative is to use the synthesized data to train downstream tasks (e.g. object detection, semantic segmentation), and present validation results on real data.

C. Benchmarks

We first test the unconditional (GSN) baseline where scene reconstruction starts from normally distributed noise input. For this experiment, we split our dataset into sub-sequences of 100 frames each, similar to the original methodology [3], and train the model using the associated poses as input and the RGB images as targets. We keep the size of W^{BEV} to 32×32 with 32 channels as per original methodology and set the depth clipping planes $(t_n, t_f]$ at $(0\,\mathrm{m}, 80\,\mathrm{m}]$.

To study the quality of conditional FV output in the absence of a dedicated BEV to FV transformation step, we train two models to output $\hat{I}^{FV}$ directly from input S^{BEV}. We choose SPADE [31] (SPADE), trained with $\mathcal{L}_A$ but also state-of-the-art Latent Diffusion Model (LDM) [26] (LDM), trained with $\mathcal{L}_{LDM}$, as single-stage models.

To check the contribution of a conditional BEV to FV transformation step in the single-stage models, we design the following experiments and set W^{BEV} size to 100×100:

1) (Base 1): outputs $(\hat{I}^{FV}, \hat{D}^{FV})$; trained with $\mathcal{L}_I$, $\mathcal{L}_D$;
2) (Base 2): outputs $(\hat{I}^{FV}, \hat{D}^{FV}, \hat{S}^{FV})$; trained with $\mathcal{L}_I$, $\mathcal{L}_D$ and $\mathcal{L}_S$, where the additional segmentation CE loss supervision acts as a guide for producing better structure;
3) (Base 3): outputs $(\hat{I}^{FV}, \hat{D}^{FV}, \hat{S}^{FV})$; trained with losses $\mathcal{L}_D$, $\mathcal{L}_S$ and $\mathcal{L}_A$, as it has been argued that reconstruction loss leads to blurry average images [32], [33];
4) (Base 4): outputs $(\hat{I}^{FV}, \hat{D}^{FV}, \hat{S}^{FV})$; trained with losses $\mathcal{L}_I$, $\mathcal{L}_A$, $\mathcal{L}_D$, $\mathcal{L}_S$ and as this mixture could lead to better results [32], [33];
5) (SPADE B1): outputs $(\hat{I}^{FV}, \hat{D}^{FV})$; trained with losses $\mathcal{L}_I$, $\mathcal{L}_A$ and $\mathcal{L}_D$;
6) (SPADE B2): outputs $(\hat{I}^{FV}, \hat{D}^{FV}, \hat{S}^{FV})$; trained with losses $\mathcal{L}_A$, $\mathcal{L}_D$ and $\mathcal{L}_S$;
7) (SPADE B3): outputs $(\hat{I}^{FV}, \hat{D}^{FV}, \hat{S}^{FV})$; trained with losses $\mathcal{L}_I$, $\mathcal{L}_A$, $\mathcal{L}_D$ and $\mathcal{L}_S$.

While single-stage models are tasked to learn both structural lifting and image synthesis simultaneously, our two-stage modular approach has each stage specialised on one task: the first stage performs the BEV to FV semantic map transformation $(S^{BEV} \rightarrow S^{FV})$, while at inference time the second stage synthesises the RGB image from the resulting output of the first stage $(S^{FV} \rightarrow I^{FV})$.

We perform three experiments for the 2-stage models:

1) (LDM+LDM): outputs $\hat{S}^{FV}$ in 1st stage, outputs $\hat{I}^{FV}$ in 2nd; trained with $\mathcal{L}_{LDM}$ in both stages.
2) (Neural+SPADE): outputs $(\hat{S}^{FV}, \hat{C}^{FV}, \hat{O}^{FV}, \hat{D}^{FV})$ in 1st stage, $\hat{I}^{FV}$ in 2nd; trained with $\mathcal{L}_S$, $\mathcal{L}_C$, $\mathcal{L}_O$, $\mathcal{L}_D$ in stage 1 and $\mathcal{L}_A$ in stage 2.
3) (Neural+LDM): outputs $(\hat{S}^{FV}, \hat{C}^{FV}, \hat{O}^{FV}, \hat{D}^{FV})$ in 1st stage, and $\hat{I}^{FV}$ in 2nd; trained with $\mathcal{L}_S$, $\mathcal{L}_C$, $\mathcal{L}_O$, $\mathcal{L}_D$ in stage 1 and $\mathcal{L}_{LDM}$ in stage 2.

D. Model Components

We repurpose the encoder and decoder from [26] for our encoder-decoder pair (ε, δ). We initialise (ε, δ) and the entire LDM (LDM and stage 2 of (Neural+LDM)) from publicly available pre-trained parameters (SD-v-1-4) [26] but extend the weights of the first layer to support the one-hot encoded input.

For experiments (Base 1) to (Base 4) we use the decoder δ as backbone to directly output combinations of $\hat{I}^{FV}$ and $\hat{S}^{FV}$, while for for (SPADE B1) to (SPADE B3), the output from δ (in this case, $\hat{S}^{FV}$) is input to a SPADE [31] backbone. For the second stage of (Neural+LDM), we extend the model from [26] with two extra convolutional layers (ϕ) that embed the conditional inputs. The embedded inputs are concatenated with the standard noisy latents and given as input to the LDM denoising network. For adversarial supervision in experiments (Base 1) to (Base 4), we use the OASIS discriminator backbone [34]. We base our ray-based point sampler γ, neural radiance field ψ and volume renderer υ on the architectures proposed in [3].

E. Implementation and Training Details

The integrals in (1) and (2) are approximated across a discrete set of samples following equation (3) in the NeRF [18] formulation, and similar to the public implementation of GSN [3], with color values replaced by features f.

The resolution of the input BEV is 1024×1024. We use Adam optimiser with a learning rate of 10^{-4} and a batch size of 1. We sample 100 points per each ray, except in (+ double points per ray) where we sample 200 points. We empirically set the weights of rare classes *[bicycle, person, rider, pole, traffic sign]* to *[3.0, 5.0, 3.0, 3.0, 3.0]*. We use $t = 50$ diffusion steps in (LDM), (LDM+LDM) and (Neural+LDM). We train the models on an NVIDIA V100 GPU with 32 GB of VRAM.

V. Results

Tables II and III present the results of our method and the selected baselines on KITTI-360. In terms of perceptual quality, both the single-stage (LDM) approach and our factorised model (Neural+LDM) have a significantly lower (better) FID score than the rest of the models, with $41.10/68.75$ and $42.23/65.81$, respectively. In contrast, the unconditional model (Neural+LDM) has a much lower perceptual quality, with an FID of 187.39. The 2 stage model where both stages are tackled via an LDM (LDM+LDM) produces FID and KID scores comparable to the single stage (LDM) model.

The conditional single-stage models ((Base 1) to (SPADE B3)) in Table III, generally produce low-quality images as opposed to (LDM), with FID ranging from 309.30 to 85.52 and KID following a similar ranking, as these models are tasked to deal with both BEV to FV transformation and image synthesis. However, in the (SPADE) experiment where the BEV to FV semantic map transformation is not employed, the perceptual quality is much lower than the experiments where the transformation is ((SPADE B1) to (SPADE B3)), although they all have the same backbone for image synthesis, highlighting the benefit of the semantic transformation step. Moreover, both the mIoU alignment and perceptual quality is improved in this group of models, as they are forced to reproduce the FV semantic structure S^{FV}.

In terms of how well the produced RGB images align with the GT RGB images, the mIoU-alignment metric shows that our factorised approach (Neural+LDM) has significantly better alignment (28.11) than both the single-stage (LDM) model (20.55) and all other approaches. The (LDM) model reproduces the general layout of scenes in terms of road and buildings but cannot accurately synthesise smaller classes in the right location. In contrast, our two-stage approach reproduces the general layout *and* the local details. The success of the (LDM) model in terms of perceptual quality can be attributed to the ability to leverage the learnt prior (being trained on a large dataset) however, when compared to (Neural+LDM) that also employs the BEV to FV transformation, the mIoU alignment is lower.

In terms of mIoU alignment between predicted and GT FV segmentation, our model also performs best (32.06) compared to all other baselines that output $\hat{S}^{FV}$. In terms of RMSE, all models perform approximately the same, which is however unsurprising, as we did not design extra experiments to optimise the output depth in particular. While being able to generate naturally-looking semantic shapes, the (LDM+LDM) model has both a low mIoU and low mIoU-alignment, indicating that the model is less capable of correctly spatially mapping the BEV structure to the FV outputs.

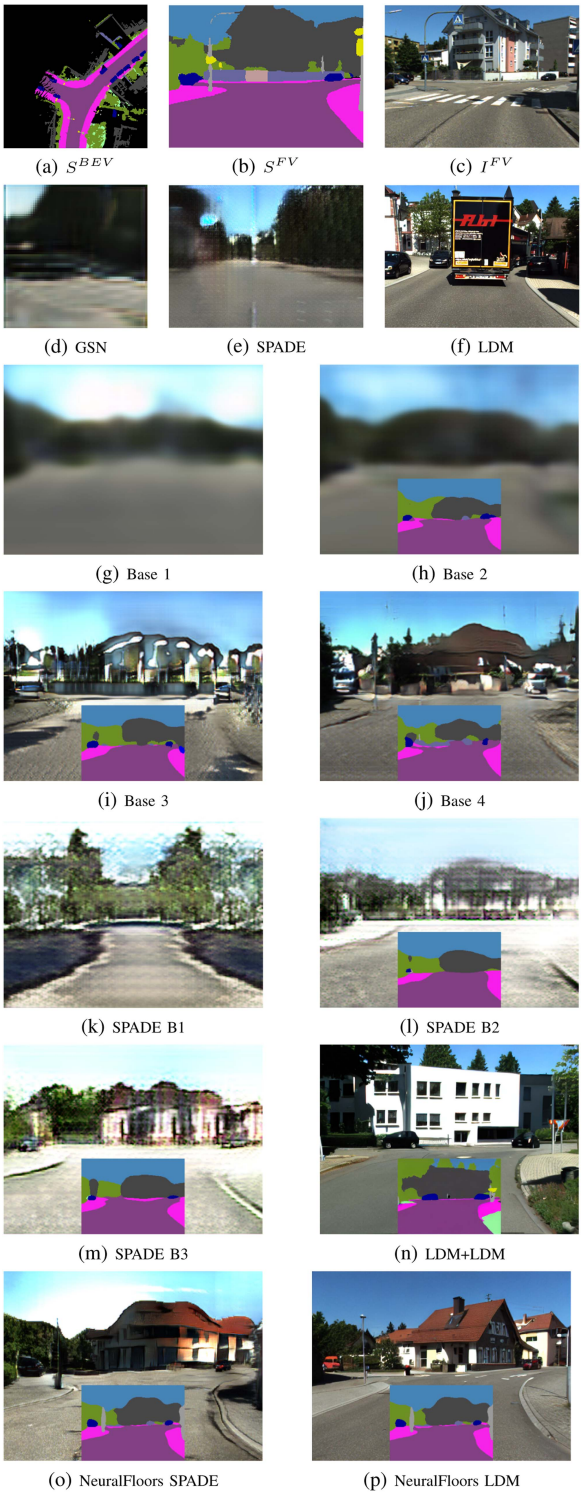


Fig. 3. Qualitative examples of FV output across all methods. GT illustrated for reference (1st row).

From a qualitative point of view, we see that (GSN) Fig. 3(d) is unable to generate high-quality samples when trained on urban scenes, as also confirmed by [5]. On the other hand, we see in Fig. 3(f) that (LDM) produces realistic looking images,

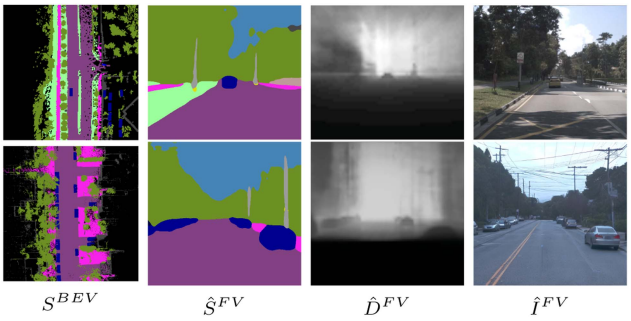


Fig. 4. FV output: nuScenes (1st row) and Waymo (2nd row).

TABLE V
ABLATION FIRST STAGE $BEV \rightarrow FV$ SEMANTIC LIFTING

Model	mIoU ↑	RMSE ↓
(Basic)	28.86	7.683
(+ tri-plane representation)	29.64	6.901
(+ panoptic segmentation)	29.79	6.609
(+ small class weighting)	30.48	6.632
(+ depth-conditioned decoder)	31.24	6.584
(+ double points per ray)	*32.06*	*6.897*

TABLE VI
ABLATION SECOND STAGE: $FV \rightarrow FV$ IMAGE SYNTHESIS

Model	FID ↓	KID ↓
(Basic)	45.52	0.0213±0.0009
(+ instance-conditioned)	54.18	0.0192±0.0008
(+ depth-conditioned)	*32.89*	*0.0095±0.0005*

but which are misaligned with the GT semantic layout. In this example, the centre of the frame is dominated by a truck, which is however not present in the BEV segmentation. In contrast, our method in Fig. 3(p) produces imagery much better aligned with the input semantic contents and FV semantic map, while also being highly realistic-looking.

Additionally, we report quantitative results on the nuScenes and Waymo datasets. For nuScenes, we obtain a significantly improved mIoU for stage 1, which we attribute to higher quality and more diverse labels. Similarly, we notice a large improvement in FID and KID, indicating that perceptual quality is also better. The model trained on Waymo data shows a lower mIoU, but a much better FID and KID compared to KITTI-360. We also show qualitative results in Fig. 4, with examples of input BEVs and output FV segmentation, depth and color images.

VI. ABLATION STUDY

To further support our design choices, we perform ablations on the first and second stage of the factorised approach, with results in Tables V and VI respectively, where we iteratively add new features to a base case, with + indicating incremental additions to the setting of the previous row.

A. $BEV \rightarrow FV$ Semantic Lifting

As our main method, we test a simple 1-plane representation (Basic) of the W^{BEV} latent floorplan where all 96 feature maps are used to sample features using the (x, z) coordinates via bilinear interpolation. We include a tri-plane representation (+ tri-plane representation) where W^{BEV} is reshaped into 3 planes W^{XZ}, W^{XY} and W^{YZ}, each with 32 feature maps. Coordinate pairs (x, z), (x, y) and (y, z) are then used to sample from these 3 maps via bilinear interpolation. In (+ panoptic segmentation), we introduce panoptic supervision, as the model might benefit from information related to object boundaries.

Since we use real-world datasets, rare/low-area classes are under-represented thus, they are often ignored in the predicted $\hat{S}^{FV}$. To overcome this limitation, we introduce class weighting (Section IV-E) to improve accuracy for small classes such as poles and traffic signs (+ small class weighting).

To improve segmentation output, we modify the final decoder to also be conditional on the predicted depth (+ depth-conditioned decoder). As we sample large scenes, we increase the number of samples per ray from 100 to 200, at a cost of approximately $2\times$ more memory use and increase in training and inference time (+ double points per ray).

Results in Table V show that the mIoU has benefited from each additional component. However, depth was only slightly advantaged, with the most improvement being brought by a tri-plane representation and panoptic segmentation.

B. $FV \rightarrow FV$ Image Synthesis

We also investigate whether additional conditional inputs to the LDM-based image synthesis improves performance. We start with a basic model that is conditional on semantic segmentation only: $\phi(S^{FV})$ (Basic); we add instance segmentation $\phi(S^{FV}, E^{FV})$ (+ instance-conditioned); and finally depth maps $\phi(S^{FV}, E^{FV}, D^{FV})$ (+ depth-conditioned). To isolate stage 2 ablation results from the effects of stage 1 predictions, we use GT segmentation, instance and depth data as inputs rather than predictions from stage 1. The results are shown in Table VI, where the inclusion of instances did not have a net beneficial effect but the inclusion of depth significantly improved FID and KID.

VII. CONCLUSION

We have presented a novel system for synthesising RGB FV imagery from a BEV segmentation map. Our contribution in this area is twofold, addressing the difficult problem of transforming the geometry of the BEV to that of the Ground-View *and* the generation of realistic imagery from FV semantics. Our novelty in the first area is using conditional Neural Fields to model and improve the geometric transformation from the BEV to the FV. In the second area, we extend LDMs to be conditional on segmentation, instances, and depth information to achieve high-quality image synthesis.

Extensive experiments demonstrate that our factorised approach produces more realistic imagery than prior methods and baselines and that, critically, this realism is accompanied by good alignment of output and GT semantic layout.

An advantage of this factorised approach is that it relaxes the requirement of paired BEV segmentation and FV images, thus enabling the usage of diverse sources of input data, such as simple/proto-simulators for the 1st stage. Since our model is conditional on BEV segmentation, it offers higher interpretability and level of control than previous methods.

Future Work

We plan to extend the existing architecture to incorporate viewpoint and temporal consistency, in order to produce structure- and appearance-consistent scenes, which would enable a wider range of downstream tasks to be trained.

Acknowledgment

The authors would like to acknowledge the use of Hartree Centre resources and the University of Oxford Advanced Research Computing facility in carrying out this work.

References

[1] S. Tan et al., "Scenegen: Learning to generate realistic traffic scenes," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 892–901.
[2] C. Richter, N. Roy, and V. Koltun, "Playing for benchmarks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.
[3] T. DeVries, M. Á. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind, "Unconstrained scene generation with locally conditioned radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.,*2021, pp. 14284–14293.
[4] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "GANcraft: Unsupervised 3D neural rendering of minecraft worlds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14072–14082.
[5] C. H. Lin et al., "Infinicity: Infinite-scale city synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 22751–22761. [Online]. Available: https://api.semanticscholar.org/CorpusID:256105305
[6] A. Swerdlow, R. Xu, and B. Zhou, "Street-view image generation from a bird's-eye view layout," 2023, *arXiv:2301.04634*.
[7] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023.
[8] H. Caesar et al., "nuscenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.
[9] X. Tian et al., "Occ3D: A large-scale 3D occupancy prediction benchmark for autonomous driving," in *Proc. 37th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2023.
[10] T. W. Team., "Simulation city: Introducing waymo's most advanced simulation system yet for autonomous driving," 2021. [Online]. Available: https://blog.waymo.com/2021/06/SimulationCity.html
[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.
[12] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning," *IEEE Access*, vol. 9, pp. 153171–153187, 2021.
[13] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[14] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8808–8816.
[15] V. Musat, D. D. Martini, M. Gadd, and P. Newman, "Depth-sims: Semi-parametric image and depth synthesis," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2388–2394.
[16] Y. Chen et al., "Geosim: Realistic video simulation via geometry-aware composition for self-driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7230–7240.
[17] A. Amini et al., "VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *Proc. Int. Conf. Robot. Automat.*, 2021, pp. 2419–2426.
[18] 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," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Dec. 2021, doi: 10.1145/3503250.
[19] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixeLNERF: Neural radiance fields from one or few images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4578–4587.
[20] E. R. Chan et al., "Efficient geometry-aware 3D generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16123–16133.
[21] Y. Xue, Y. Li, K. Singh, and Y. Lee, "Giraffe HD: A high-resolution 3D-aware generative model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18419–18428.
[22] X. Fu et al., "Panoptic NeRF: 3D-to-2D label transfer for panoptic urban scene segmentation," in *Proc. IEEE Int. Conf. 3D Vis.*, 2022, pp. 1–11.
[23] B. Ren, H. Tang, Y. Wang, X. Li, W. Wang, and N. Sebe, "PI-Trans: Parallel-convmlp and implicit-transformation based Gan for cross-view image translation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
[24] B. Cheng et al., "Panoptic-Deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12475–12485.
[25] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," Jun. 2022, doi: 10.1109/CVPR52688.2022.01254.
[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10684–10695.
[27] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[28] 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," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.
[29] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1lUOzWCW
[30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.
[31] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
[32] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[33] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
[34] V. Sushko, E. Schönfeld, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, "OASIS: Only adversarial supervision for semantic image synthesis," *Int. J. Comput. Vis.*, vol. 130, no. 12, pp. 2903–2923, Dec. 2022, doi: 10.1007/s11263-022-01673-x.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	NeuralFloors: Conditional Street-Level Scene Generation From BEV Semantic Maps via Neural Fields
Publication Status	Published
Publication Details	**V. Muşat**, D. De Martini, M. Gadd and P. Newman, "NeuralFloors: Conditional Street-Level Scene Generation From BEV Semantic Maps via Neural Fields," in *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2431-2438, March 2024, doi: 10.1109/LRA.2024.3356793.

Student Confirmation

Student Name:	Valentina Musat		
Contribution to the Paper	- developed the idea supporting the paper under the guidance of Dr. De Martini and Dr. Gadd, and supervision of Prof. Newman - implemented the architecture - compiled the data for training - ran all experiments and interpreted data - wrote manuscript - prepared presentation materials - worked towards IP registration and patent application		
Signature 		Date	21.04.2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman		
Supervisor comments I certify that the candidate made a substantial and leading contribution to the publication, and that the description described above is accurate		
Signature 	Date	24/04/25

6.3 Further insights

The tri-plane representation is based on the intuitive idea that 3D space can be efficiently encoded using 3 spatially-aligned orthogonal 2D maps, each representing a projection of the 3D volume of features, with the XZ plane encoding information across the floorplan, the YZ plane encoding information along the depth dimension and the XY plane encoding information along the height dimension. While not a complete 3D volumetric representation, the tri-plane representation is a learned approximation that preserves 3D context, with 2D features sampled from the coordinates of the 3D point projected onto each plane being concatenated before further processing by the MLP.

Additionally, two further aspects are worth highlighting in relationship to this particular representation. Firstly, while features from the XZ plane can be directly extracted from the BEV input, features from the YZ and XY planes represent information that needs to be inferred by the encoder, since both are a function of the height dimension along the Y axis, which does not exist in the input 2D BEV. Secondly, multiple points in the 3D volume will project onto the same coordinate of a plane – for example, all 3D points with coordinates (x_i, y_i, z_i) will project onto the same (x_i, z_i) coordinate of the XZ plane. This means that the MLP, which takes as input the 3D coordinates (x_i, y_i, z_i), their positional encoding, and the feature sampled from the (x_i, z_i) coordinate will need to output a transformed feature that is a function of the y_i coordinate, i.e. that changes with height, even if the input feature remains constant. Similarly, while the features sampled from the YZ and XY planes may vary with height, they are still inferred by the BEV encoder as projections of one of many possible plausible interpretations of the 3D scene described by the input 2D BEV.

The decoupling between the model stages allows for separate datasets to be used. For example, semantic segmentation BEVs and ground-view semantic segmentation and depth maps obtained from a 3D engine simulator such as CARLA (Dosovitskiy et al., 2017) can be used to either train the first stage, or at inference time, as shown in Figs. 6.1 and 6.2. In comparison to BEV maps obtained or constructed from real world datasets, data from a simulator has a number of advantages, including perfect ground-truth annotations and very good alignment between BEV and ground-view data. Moreover, a simpler 2D BEV-based simulator can be used as a source of input data during inference, resulting in a pipeline that leverages the best of both worlds: easy editability or controlability in 2D and generation of diverse ground-view data, including ground-truth.

However, the sim to real domain gap can be problematic even in the case of semantic segmentation maps, where ground-truth produced by simulators is more fine-grained compared to

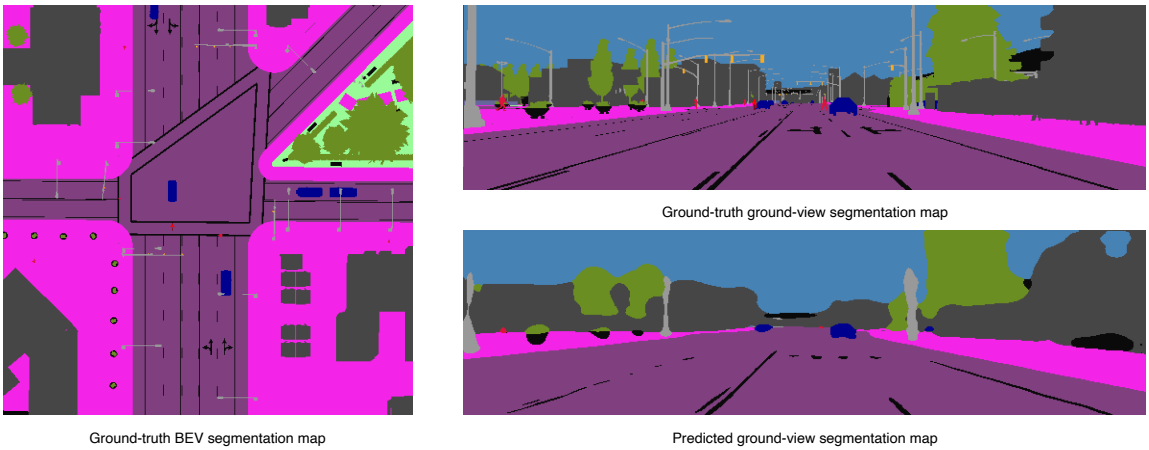


Figure 6.1: An example from the first stage of NeuralFloors, where the input is a BEV semantic segmentation of an intersection from the Carla simulator.

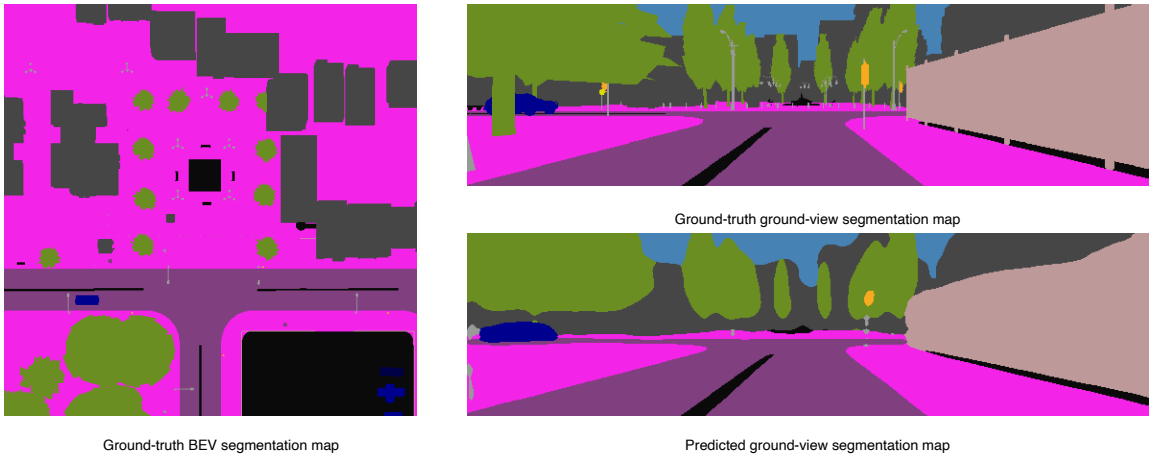


Figure 6.2: An example from the first stage of NeuralFloors, where the input is a BEV semantic segmentation of an intersection from the Carla simulator.

ground-truth obtained through human- or machine-labeling. Additionally, objects that are partially transparent such as vehicles with windows, and differences in taxonomies might also introduce issues in terms of domain gaps.

Secondly, the method has no explicit mechanism for ensuring frame-to-frame (temporal) or multi-view consistency. This problem becomes more pronounced in the second stage, where each pixel represents a distinct part of a texture, whereas in the first stage, pixels largely form contiguous blocks of identical values representing the semantic class. However, in the second stage, especially due to the use of a multi-step diffusion model, small changes in the conditioning can lead to large variations in the RGB outputs across consecutive frames. The approach presented in Chapter 7 addresses some of these limitations.

Furthermore, weather effects might be incorporated by conditioning the second stage with text

prompts, exemplar conditioning, or both. With text-based conditioning, the existing configuration which optimises the entire UNet would require pairing the image data with relevant text captions. With exemplar-based conditioning, an extra input in the form of an RGB image depicting the desired style (but which does not necessarily need to be structurally correlated with the semantic segmentation, instance segmentation or depth conditioning) could be used to encourage style injection, similar to the method proposed in Chapter 7. Alternatively, the NeuralFloors approach could be combined with Multi-weather-city (Chapter 4), by first synthesising data at scale and subsequently adding various levels of illumination and weathers.

Finally, while the method outputs ground-view ground-truth data (in contrast to other methods such as CC3D (Bahmani et al., 2023)), metrics such as mIoU and mIoU-align are not an ideal choice for capturing the one-to-many nature of the BEV to ground-view mapping, neither in terms of plausibility nor in terms of diversity, instead penalising any deviation from the training ground-truth. A common approach involves re-projecting the outputs back onto the BEV and comparing to the original conditioning BEV, but this ultimately discards the height information and is not particularly informative.

7

NeuralFloors++

Contents

7.1 Contribution

Many visual perception tasks can be trained, tested or validated on datasets comprised of individual, uncorrelated images, but not all. Tasks such as object tracking or visual localisation require structural and stylistical consistency between frames. While the previous work, NeuralFloors (Muşat et al., 2024a), has shown good results when generating individual frames, it suffers from a lack of temporal coherence due to the inherent nature of the latent diffusion model used in the second stage, and the absence of mechanisms that enforce consistency.

This work represents an extension of NeuralFloors (Chapter 6), focusing on techniques that improve frame-to-frame consistency, for both structure and style. The key contributions of the paper include:

1. Multi-view training for the first stage to encourage consistency of structure. Semantic segmentation, instance and depth maps from multiple camera poses are generated from the accumulated point cloud data of KITTI-360 to supplement the original single-view data used in the previous approach;

2. The first stage is also trained to output ground-view feature maps from BEV instance-averaged features, which are consistent across camera poses and movement. The second stage is additionally trained to make use of these features (together with the initial input) to synthesize consistent RGB images. This mechanism offers controlability capabilities, as changes to instance-averaged features in the BEV are reflected in the ground-view RGB images;

3. Additional auto-regressive training and inference for the second stage, where the RGB image generated at the previous timestep is used as an additional input at the current timestep, further improving temporal consistency;

4. An extensive comparison to the performance of existing methods in terms of perceptual quality and alignment to ground-truth, and qualitative experiments to illustrate style control and compositionality being achieved by changing the BEV input.

The main difference between the previous work NeuralFloors (Chapter 6) and NeuralFloors++ is the frame-to-frame consistency, which is enabled by the use of style maps (further discussed in Section 7.3) in the 2 stages, auto-regressive training in stage 2 and the use of multi-view data. Additionally, dynamic objects are re-introduced in the dataset by leveraging the 3D bounding box annotations from KITTI-360 (Liao et al., 2023). In order to support these additions, several architectural changes have been implemented, as shown in Fig. 7.1 and Fig. 7.2.

The BEV encoder ϵ^{BEV} receives as input the concatenation of the BEV semantic segmentation map S^{BEV}, instance map Q^{BEV}, and style map H^{BEV}, as opposed to NeuralFloors which only received as input the semantic segmentation map. The encoder further outputs a latent representation that is reshaped to 3 orthogonal planes $W^{BEV} = \{W^{XZ}, W^{XY}, W^{YZ}\}$. In order to enforce layout consistency, the latent representation W^{BEV} is additionally given as input to the BEV decoder δ^{BEV}, which reconstructs the BEV semantic segmentation $\hat{S}^{BEV}$ and style map $\hat{H}^{BEV}$ – this is in addition to

the previous architecture, where such a component was not present. A ray-based point sampler is further used to define 3D points along camera rays between the bounds of the near plane t_n and far plane t_f, where t_n is the minimum distance along the ray from the camera origin from which points are sampled and t_f is the maximum distance along the ray from the camera origin up to which points are sampled.

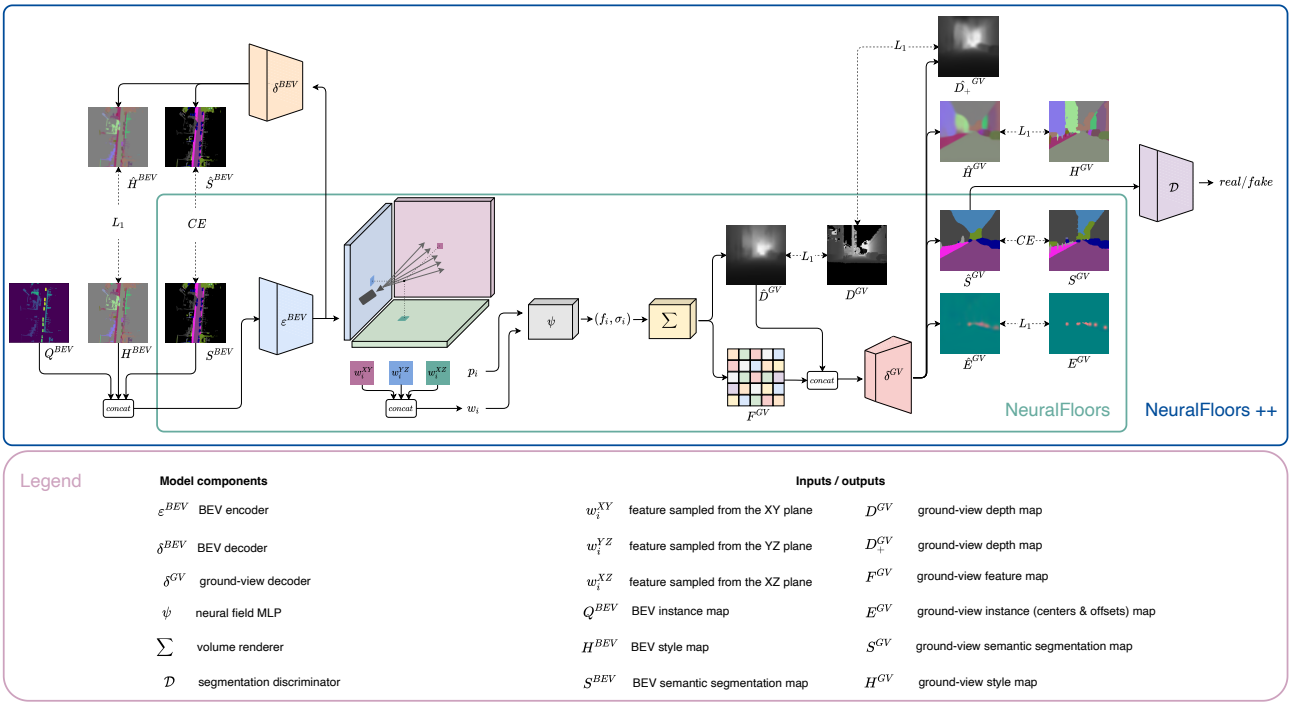


Figure 7.1: The first stage receives as input semantic, instance and style BEV maps representing the top-down scene, and outputs ground-view maps representing semantic, instance, style and depth information. Components with ˆ notation represent the predicted counterpart e.g. H^{BEV} is the input style map whereas $\hat{H}^{BEV}$ is the predicted style map.

For each 3D point, 3 latent features are sampled from the 3 orthogonal planes using bi-linear interpolation and further concatenated to obtain $w_i = w_i^{XZ} \oplus w_i^{XY} \oplus w_i^{YZ}$, which is given as input to a neural field MLP ψ together with the positional encoding $\rho(p_i)$ of the coordinates of the point. The neural field then predicts for each 3D point a feature f_i and a density value σ_i. The volume renderer then accumulates all the predicted features and densities for all 3D points across all rays, and further outputs a ground-view feature image and expected depth, which are then concatenated and given as input to a ground-view decoder δ^{GV}. The decoder further predicts the ground-view semantic segmentation map $\hat{S}^{GV}$, instance segmentation map $\hat{E}^{GV}$, style map $\hat{H}^{GV}$ and an up-sampled depth map $\hat{D}_+^{GV}$ – this is in contrast to NeuralFloors, where the decoder only outputs the semantic and instance segmentation maps. Finally, in addition to the previous architecture, the predicted

ground-view semantic segmentation map $\hat{S}^{GV}$ is consumed by a segmentation discriminator $\mathcal{D}$, which assesses whether the segmentation is real or generated.

The second stage consists in a latent diffusion model that is adapted to receive as input the ground-view maps produced in the first stage and a previously generated image. More specifically, during training, the model receives as input at current time t, the concatenation of the ground-view semantic segmentation map S_t^{GV}, instance segmentation map E_t^{GV}, style map H_t^{GV}, depth map D_t^{GV} and a previously generated image I_{past}^{GV}. This is different from the architecture in NeuralFloors where the model did not receive as input the style map or a previously generated image. Further, a shallow CNN is used to encode the ground-view input which is further concatenated with a noisy latent embedding and de-noised by the model. The final ground-view image I_t^{GV} at time t is generated by decoding the denoised latent embedding.

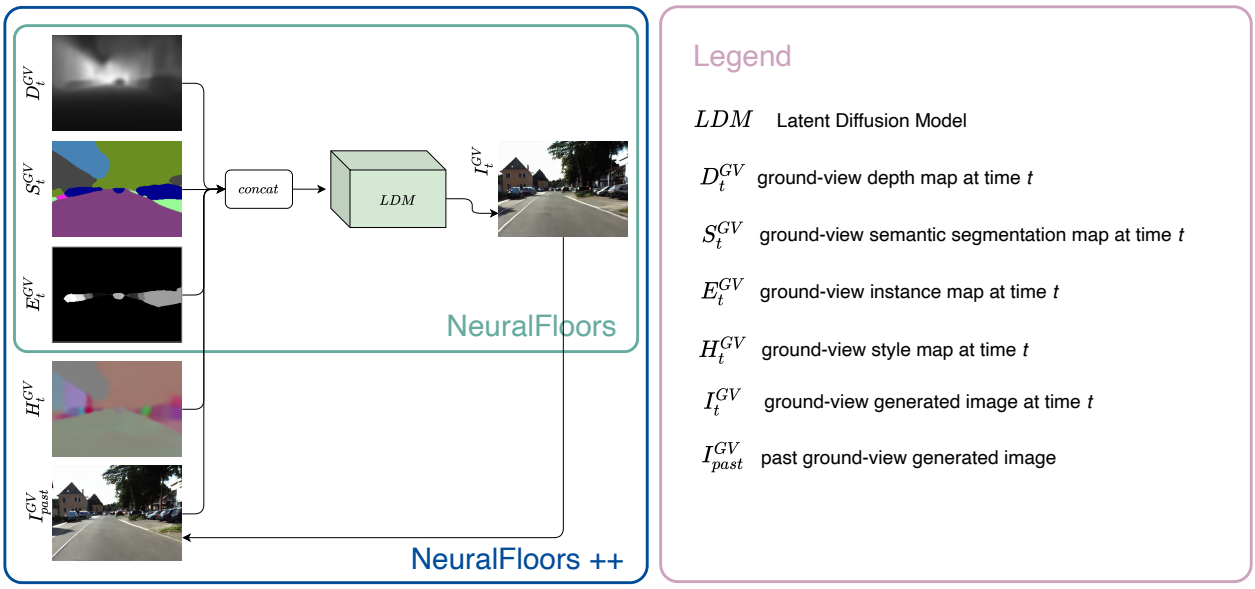


Figure 7.2: The second stage is a conditional latent diffusion model that receives as input ground-view depth, semantic segmentation, instance segmentation and style maps, and additionally an RGB generated image from a past timestamp.

With respect to the second stage, two widely-known implementations are used. The first one is the original Stable Diffusion (Rombach et al., 2021) architecture that is extended with a shallow CNN similar to NeuralFloors – referred to as NeuralFloors++(SD) in the integrated publication manuscript. The second, and more up-to-date method is the T2I-Adapter architecture proposed in Mou et al. (2024), which extends the capability of Stable Diffusion to accept multi-modal input – referred to as NeuralFloors++(TI-Adapter) in the integrated publication manuscript. While the two architectures use the same main components (UNet, image encoder, image decoder), the encoded

conditioning in T2I-Adapter is injected into the UNet residual blocks at multiple resolutions, whereas in NeuralFloors++(SD), it is concatenated with the noisy latent embedding as described above.

In order to improve multi-view consistency, the dataset is extended by randomising the camera poses and rendering both the BEV and ground-view semantic, instance and depth maps using Open3D (Q.-Y. Zhou et al., 2018). Additionally, the dataset is extended to contain dynamic objects by loading their 3D bounding boxes together with the accumulated static point clouds.

7.2 Integrated manuscript

The manuscript was published at the International Conference on Intelligent Robots and Systems (IROS), 2024 (Muşat et al., 2024b)

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802002

NeuralFloors++: Consistent Street-Level Scene Generation From BEV Semantic Maps

Valentina Muşat†, Daniele De Martini‡, Matthew Gadd‡ and Paul Newman
Mobile Robotics Group (MRG), University of Oxford, † Corresponding author ‡*Equal contribution*
{valentina,daniele,mattgadd,pnewman}@robots.ox.ac.uk

***Abstract*— Learning autonomous driving capabilities requires diverse and realistic training data. This has led to exploring generative techniques as an alternative to real-world data collection. In this paper we propose a method for synthesising photo-realistic urban driving scenes, along with semantic, instance and depth ground-truth. Our model relies on Bird's Eye View (BEV) representations due to their compositionality and scene content control capabilities, reducing the need for traditional simulators. We employ a two-stage process: first, a 3D scene representation is extracted from BEV semantic, instance and style maps using a neural field. After rendering the semantic, instance, depth and style maps from a ground-view perspective, a second stage based on a diffusion model is used to generate the photo-realistic scene. We extend our prior work - NeuralFloors, to include multiple-view outputs, style manipulation for finer control at the object level through instance-wise style maps and cross-frame consistency via auto-regressive training. The proposed system is evaluated extensively on the KITTI-360 dataset, showing improved realism and semantic alignment for generated images.**

I. Introduction

Generative AI has made exceptional progress in the last few years, especially in image and video generation, where scene synthesis has popular applications in creative industries, gaming, and robotics. Synthetic data generation has become increasingly crucial for training computer vision models, particularly in safety-critical applications like Autonomous Driving (AD). Here, generated data paired with its source Ground Truth (GT) offers a cheap, scalable, and easier-to-obtain data source that complements real-world data collection. Additionally, it eliminates exposure to real-world risks, thus allowing for the generation of challenging and rare-case scenarios that would otherwise be unattainable.

Traditional simulators, while widely employed, often lack scalability and realism due to their reliance on manually created 3D assets and artists for appearance modelling. Recent advancements in Neural Radiance Fields (NeRFs) and diffusion-based models have shown remarkable potential in synthesizing realistic scenes, making them promising candidates for data-driven simulators. Inspired by this, we present a method for synthesising outdoor urban driving photo-realistic ground-view scenes paired with GT semantic, instance, depth and style maps, starting from a Bird's Eye View (BEV) representation – shown in Fig. 1. Our approach is factorized into two stages: 1) learning a 3D representation of the scene conditional on a BEV representation through a neural field, and 2) generating ground-view photo-realistic scenes based on the output of the first stage, using a diffusion-based model. Our work, NeuralFloors++, extends NeuralFloors [1] to multi-view-consistent output and incorporates scenes containing objects in motion. We introduce style manipulation by employing an instance-wise average style BEV map inspired by pix2pixHD [2] and enhance cross-frame consistency by introducing auto-regressive training in the second stage. In contrast to CC3D [3], which employs a global latent style code, our approach allows for finer style control at the object level, and provides a wider range of ground-view modalities. Finally, by leveraging a BEV semantic map that is isometric and inherently compositional in 2D, we enhance compositionality and control of the scene contents bypassing the need for a traditional simulator.

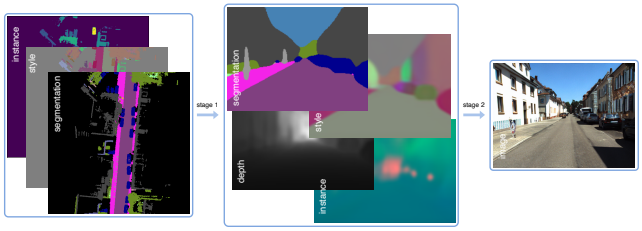


Fig. 1: NeuralFloors++ generates photo-realistic ground-view scenes (*Right*) paired with ground-truth semantic, instance, depth and style maps (*Middle*), starting from a BEV representation (*Left*).

To summarise, our key contributions are as follows: 1) a system that allows ground-view scene synthesis from a BEV representation that can 2) output a consistent style which is 3) controllable and offers object-level manipulation. Our approach is 4) able to synthesise a diverse set of ground-view modalities that can be further used to train downstream tasks. We conduct thorough experiments on the KITTI-360 dataset [4], evaluating our method using a variety of metrics, including quantitative measures for depth, segmentation, perceptual quality, and we present a set of qualitative analyses and ablation studies.

II. Related Work

Traditional data acquisition, whilst offering the most realistic data for training and testing, suffers from limited diversity, complexity and scarce pixel-level GT, especially

*Supported by a DeepMind Engineering Science Scholarship and EPSRC Programme Grant "From Sensing to Collaboration" (EP/V000748/1). The authors also acknowledge the use of Hartree Centre resources and the University of Oxford Advanced Research Computing facility in this work.

when moving from 2D to 3D. Asset-based, classic 3D simulators [5] have been a common choice for gathering data due to their physically-grounded outputs, virtually unlimited GT data and high level of control; however, they require laborious asset building, making the process difficult to scale, while the inherent gap between the complexities of the real world and the approximations made by the simulation engines limits their applicability. To address this, researchers have proposed data-driven simulators, with the latest approaches leveraging Large Language Models (LLMs) to generate environments [6]. Since the success of Generative Adversarial Networks (GANs) [7], works such as pix2pixHD [2] have shown great capabilities of synthesizing high-resolution, realistic images from segmentation maps, with geometric [8] and temporal consistency [9] being enforced by extra conditioning. Finally, semi-parametric methods combine the benefits of both learnt and data-driven approaches by relying on existing data to encourage realism [10], [11].

A. 3D-aware scene generation

Whereas photorealism has been the initial priority, interpolation, disentanglement and compositionality of attributes and contents are paramount to gaining control over the synthesis process. However, as image synthesis takes place in 2D, capturing effects such as shading, occlusions, objects' poses, and appearance interactions requires an understanding of the 3D world. Various approaches have tackled 3D geometry-aware synthesis by employing inductive biases in combination with differentiable rendering [12], [13], [14].

GIRAFFE [15] proposes to tackle compositionality in 3D via compositional generative feature fields, where the scene is composed of background and foreground objects, generated by separate Multi-Layer Perceptrons (MLPs). Subsequently, GIRAFFE HD [16] improved the architecture by employing a StyleGAN2-like renderer [17], enabling synthesis of high-quality scenes. While most 3D generative models have focused on object-centric scenes, GSN [18] generates unbounded indoor scenes but does not tackle compositionality, making it difficult to control the scene content. CC3D [3] and NeuralFloors [1], instead, tackle outdoor scenery and enable scene-content control via BEV semantic map conditioning.

B. Cross-view and cross-modality scene generation

GSN [18] trains an unconditional generative model to render scenes from a freely-moving camera via a NeRF and a latent floorplan representation. In their setup, a 1D Gaussian noise latent code representing the whole scene is mapped to a 2D grid of latent codes from which features are sampled using 2D coordinates. However, NeuralFloors [1] and CC3D [3] argue that this representation is not expressive enough to offer the level of control, compositionality, and interpretability generally required for scene generation, especially from a robotics application point of view. To overcome this limitation, they propose to condition the model on a top-down semantic-segmentation map representing the scene.

NeuralFloors [1] relies on a triplane representation from a single floorplan projection plane as in EG3D [19], whereas CC3D [3] extrudes the floorplan representation into a 3D grid of features. BerfScene [20] aims to tackle the same task but uses a 2D Fourier feature map of grid coordinates as input, while the BEV map and a latent code are used to modulate the encoder-decoder network. InfiniCity [21] also relies on a neural renderer to generate the ground-view but requires hard-to-acquire CAD models to represent the scene.

Outdoor scene generation in a BEV to ground-view cross-view setup has also been tackled by non-NeRF approaches such as BEVGen [22] and BEVControl [23]. BevGen learns the image formation process implicitly with an autoregressive transformer fed with multi-view images, BEV semantic layout and token direction vectors. BevControl conditions a diffusion model with a BEV sketch geometrically projected in the camera space and relies on cross-view attention to generate geometry- and appearance-consistent images.

C. Diffusion-based models

Latent Diffusion Models (LDMs) have become popular alternatives to GANs, which generally suffer from mode collapse and inherent training instability. Popular works such as ControlNet [24], T2I-Adapter [25] and InstanceDiffusion [26] unlock flexible user control of the synthesis process by conditioning a diffusion model with, e.g., edges, poses, depth, or normal maps. While this approach works well for still images, controllable video generation is more challenging as objects must follow spatial and temporal constraints. VideoComposer [27] addresses this by incorporating motion vectors and a spatiotemporal encoder employing cross-frame attention, while Lumiere [28] introduces a Space-Time UNet architecture that generates the full video sequence at once.

D. Contemporary works

Our work is based on NeuralFloors [1], which generates ground-view images conditioned on the BEV semantic map of the scene contents. To relax the assumption of paired semantic BEV maps and ground-view RGB images, the model is trained in two stages, each leveraging an architecture specialised for the task. The first stage involves a neural field to lift and extract a 3D representation of the scene, while the second stage relies on an LDM for high-quality image synthesis. However, NeuralFloors is trained on single-view images only, possibly leading to geometric ambiguity, while the second stage does not model frame-to-frame consistency. NeuralFloors++ builds on top of this work by training on multi-view data in the first stage and enhancing frame-to-frame consistency in the second stage by a style map inspired by pix2pixHD [2] and auto-regressive training.

The closest work to ours is CC3D [3], which is also conditional on the top-down semantic layout of the scene. However, there are key differences: while NeuralFloors reshapes the 2D feature maps into a tri-plane representation as in EG3D [19], CC3D reshapes them into a 3D feature grid. Additionally, CC3D is trained end-to-end, synthesising the ground-view RGB images directly with a dual discriminator,

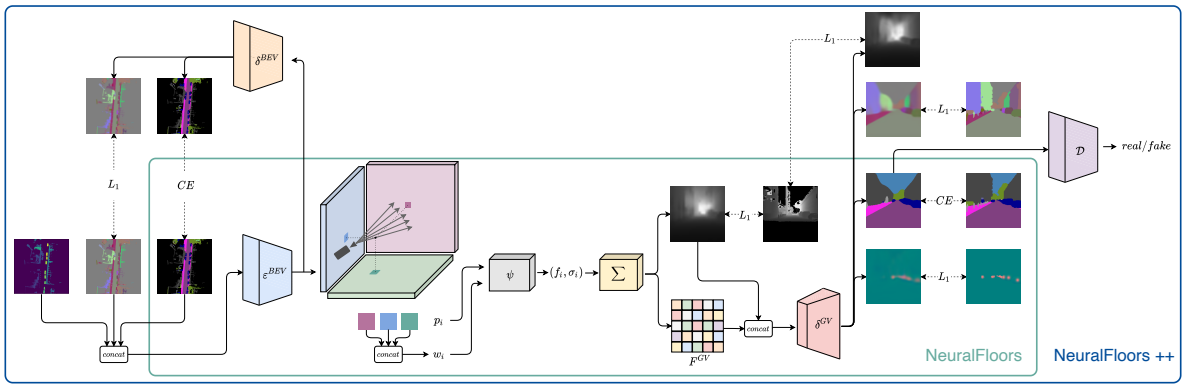


Fig. 2: **Stage 1 architecture.** BEV semantic, instance and style maps are encoded into a latent representation from which per-point features w_i are sampled via bi-linear interpolation. A neural field ψ learns a 3D representation of the scene and outputs per-point features and densities that are further aggregated by a volume renderer to obtain expected depth and feature maps. A decoder predicts the corresponding ground-view semantic, instance, style and upscaled depth maps. Additionally, a BEV decoder reconstructs the semantic and style maps from the latent representation, and a segmentation discriminator judges whether the ground-view segmentation is real.

in contrast with our 2-stage method. Moreover, CC3D manipulates the style of the generated scene through a 1D global latent noise vector, while we employ a 2D instance-wise style map spatially aligned with the BEV semantic map, allowing for finer control of style at the object level. Finally, we introduce a semantic-map discriminator to discern between ground-view generated and GT semantic maps.

III. Method

Our methodology borrows from NeuralFloors [1], where the setup is a two-stage approach, with each stage employing architectures specialised for its task. The first stage learns a mapping from BEV to ground-view representations of the scene, while the second synthesises highly photo-realistic ground-view images using the output of the first stage.

A. Semantic lifting

We leverage a ray-based point sampler following a pinhole camera model, coupled with a neural field to learn a 3D representation of the scene, conditional on 2D maps, and employ a volumetric renderer to produce ground-view maps. The system diagram for the first stage is depicted in Fig. 2.

We start from a BEV semantic segmentation map S^{BEV}, an instance ID map Q^{BEV}, and an instance average style map H^{BEV}. These inputs are representative of the local scene we want to render from ground-view, encompassing semantic and appearance information. An encoder ε^{BEV} learns to encode them into a latent representation W^{BEV}, which is reshaped into 3 orthogonal planes $W^{BEV} = \{W^{XZ}, W^{XY}, W^{YZ}\}$, similar to EG3D [19] tri-plane representation. Additionally, we enforce BEV layout consistency by reconstructing S^{BEV} and H^{BEV} via a decoder δ^{BEV}.

A ray-based sampler is used to cast R rays, with P 3D points $\{p_i | p_i = (x_i, y_i, z_i)\, \forall i = 1, ..., P\}$ along each ray r. For each point, we sample 3 latent features via bi-linear interpolation from the 3 orthogonal planes, which are further concatenated to obtain an aggregated per-point latent feature $w_i = w_i^{XZ} \oplus w_i^{XY} \oplus w_i^{YZ}$. The latent feature w_i and the corresponding 3D coordinate p_i – concatenated with its positional-encoding $\rho(p_i)$ – are given as input to a neural field ψ parametrised by an MLP, to produce a final corresponding feature and density $(f_i, \sigma_i) = \psi(p_i \oplus \rho(p_i), w_i)$.

Given the volume of densities and features, the ground-view feature image is obtained by aggregating the weighted features f_i of all the points p_i across each ray $r \in R$:

$$\hat{F}(r) = \sum_{i=1}^{P} \alpha_i T_i f_i \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \quad (1)$$

with T_i denoting the transmittance along the ray, $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$ the opacity of point i, and δ_j the distance from point j to point $j+1$. Similarly, to approximate the ground-view expected depth, the distances t_i to points p_i are weighted and aggregated along each ray $r \in R$:

$$\hat{D}(r) = \sum_{i=1}^{P} \alpha_i T_i t_i \quad (2)$$

By sampling rays from the ego-perspective of the vehicle, we obtain the expected ground-view depth $\hat{D}^{GV}$ and ground-view feature $\hat{F}^{GV}$ images, which are further concatenated as input to the decoder δ^{GV} to predict the corresponding ground-view segmentation, instance, style and upscaled depth maps $\hat{S}^{GV}$, $\hat{E}^{GV}$, $\hat{H}^{GV}$ and $\hat{D}_{+}^{GV}$. For the ground-view instance map we follow a formulation similar to [29], and define $E^{GV} = (C^{GV}, O^{GV})$, where C^{GV} is the instance centres map (centres of mass), and O^{GV} is the instance pixels offsets map (each blob pixel's offset from its corresponding centre along the image axes).

Finally, to encourage the model to produce segmentation with natural shapes, we pass the predicted segmentation map $\hat{S}^{GV}$ to a segmentation discriminator $\mathcal{D}$, which predicts whether $\hat{S}^{GV}$ comes from a distribution of real or synthesised segmentation maps. This component is added to balance the effects of a strict loss criteria applied to outputs with multiple plausible representations. More specifically, given the arrangement of scene contents in a BEV map, there are many configurations of shapes in the ground-view map that satisfy its layout. However, while applying segmentation supervision on the ground-view output (Tab. V), the model

is forced to reconstruct a particular configuration (out of many possible configurations), without knowing which one it should. Although a discriminator does not directly solve the one-to-many mapping, it can be used to balance the effects of strong supervision.

B. Image synthesis

Given the remarkable results of denoising diffusion models [30], we follow [1] and employ a latent diffusion model for the task of ground-view map to RGB image synthesis.

A diffusion model consists of two processes: a forward diffusion process adds noise ϵ drawn from a Gaussian distribution iteratively to an input x to obtain a diffused version x_t, and a reverse diffusion process recovers the original input gradually, by predicting the added noise $\hat{\epsilon}$ using a model ϵ_θ and subtracting it from x_t. Unlike diffusion models applied directly in the pixel space, latent diffusion models are applied in a lower-dimensional latent space that is more computationally efficient but still preserves content richness. In this case, an encoder and decoder learn to encode images in a latent representation z and decode them back. During inference, a diffused latent embedding z_t is sampled from a Gaussian distribution and ϵ_θ is used to incrementally denoise z_t, from which the decoder produces an RGB image.

Our method extends the denoising network ϵ_θ to additionally condition on $z_c = \phi(S^{GV}, E^{GV}, D^{GV}, H^{GV}, I^{GV}_{past})$, where I^{GV}_{past} is a previous RGB image and ϕ is an additional convolutional network used to embed the ground-view data, thus the predicted noise becoming $\hat{\epsilon} = \epsilon_\theta\big(z_t, z_c, t\big)$.

C. Instance-wise average style encoding

To encode style, we draw inspiration from pix2pixHD [2] and construct instance-wise averaged style maps. The purpose of using a style map that has the same spatial resolution as the other BEV inputs is that, as opposed to using a global style code [3], this method offers precise delimitation between objects styles and thus, finer control. Ground-view colour images are given as input to an encoder ε_{style} to produce a latent style map which is upsampled via bi-linear interpolation to the original size of the image. Using the corresponding GT segmentation and instance maps, we select the averaged style features corresponding to unique objects within an image and construct a bank of latent style codes. For classes that do not have individual instances, the average style is calculated across the semantic class in the image. Thus we obtain BEV and ground-view instance-wise averaged style maps H^{BEV} and H^{GV} that encode instance and class-specific style information.

To build the BEV and corresponding ground-view GT style maps, we first construct an empty BEV and ground-view map of the same spatial resolution as the input for each observation set. For a particular instance of an object in the BEV, we randomly pick a style code from the bank of latent style codes particular to that class. Then for all pixel locations that belong to the instance, we broadcast the style code in both the BEV and ground-view maps. Similarly, we broadcast a style code for all segmentation pixels that belong to a particular class but do not have instance information.

D. Losses

For the BEV and ground-view segmentation supervision, we apply Cross-Entropy loss[1]:

$$\mathcal{L}_S = -\mathbb{E}\left[\sum_{n=1}^{N} \alpha_n \sum_{i,j}^{H\times W} h_{i,j,n} \log \hat{S}^*_{i,j,n}\right] \tag{3}$$

where α_n weights each class. Indicator $h_{i,j,n} = 1$ if pixel $(i,j) \in n$ in the GT segmentation map otherwise $h_{i,j,n} = 0$.

For the BEV and ground-view style supervision, we apply an L1 loss between the GT and predicted style maps:

$$\mathcal{L}_H = |H^* - \hat{H}^*| \tag{4}$$

For ground-view instance map reconstruction, we apply L1 loss between GT and predicted centers and offsets:

$$\mathcal{L}_C = |C^{GV} - \hat{C}^{GV}|;\ \mathcal{L}_O = |O^{GV} - \hat{O}^{GV}| \tag{5}$$

For depth reconstruction and training stability, we apply an L1 loss between GT and predicted depth maps:

$$\mathcal{L}_D = M^D \odot |D^{GV} - \hat{D}^{GV}| \tag{6}$$

where mask M^D is used to perform pixel-wise masking if D^{GV} is not within $(t_n, t_f]$ bounds.

Additionally, an L1 loss is applied between the GT depth map and the upscaled depth predicted by decoder δ^{GV}:

$$\mathcal{L}_{D_+} = M^D \odot |D^{GV} - \hat{D}^{GV}_+| \tag{7}$$

All losses of stage 1 are additionally masked in the sparse random view, to account for pixels that do not belong to a particular class (void).

Additionally, an adversarial loss is used to train the generator $\mathcal{G}$ and discriminator $\mathcal{D}$:

$$\mathcal{L}_A = \mathbb{E}[\log \mathcal{D}\big(S^{GV}\big)] + \mathbb{E}[\log\big(1 - \mathcal{D}\big(\hat{S}^{GV}\big)\big)] \tag{8}$$

where the generator is $\mathcal{G} = \delta^{GV} \circ \sum \circ\ \psi \circ \varepsilon^{BEV}$ and its output is $\hat{S}^{GV} = \mathcal{G}(S^{BEV}, Q^{BEV}, H^{BEV})$.

The final loss $\mathcal{L}_{total}$ for stage 1 is a sum of the losses above weighted by their λs, while in stage 2, the training objective of the latent diffusion model is:

$$\mathcal{L}_{LDM} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t}\big[||\epsilon - \hat{\epsilon}||_2^2\big] \tag{9}$$

where ϵ_θ parametrised by a convolutional neural network is trained to predict the noise $\hat{\epsilon} = \epsilon_\theta(z_t, z_c, t)$.

IV. Experimental Setup

A. Data

We train and evaluate the experiments on KITTI-360 [4] – a widely-known dataset that contains complex real-world urban driving scenes, with 20 semantic classes following the Cityscapes [31] convention. Whereas NeuralFloors [1], being based solely on the accumulated point cloud for BEV creation, discards the data with objects in motion, we overcome this problem by using 3D bounding boxes for objects in motion, allowing us to use the full dataset. To create

[1]With the notation $(\cdot)^*$, we denote that a variable $(\cdot)$ can either be $(\cdot)^{BEV}$ or $(\cdot)^{GV}$

the paired data, we load the point cloud and the bounding boxes in Open3D [32], and at every k-th original pose in the KITTI-360 dataset, we generate a set of observations from different view-points. Thus, we sample 4 additional random poses with up to $1.5\,\mathrm{m}$ of lateral, $0.5\,\mathrm{m}$ forward and 40° yaw displacements, resulting in observation sets each with 1 dense and 4 sparse ground-view GT maps. For each observation, we render both BEV and ground-view semantic, instance and depth maps. Since the 3D bounding boxes in the ground-view perspective do not have natural shapes, we randomly sample points around the centre of each object's box. We use the first 8 sequences for training and reserve the last sequence for model selection and testing. We chose a step size $k = 5$ for the train set and of $k = 1$ for the test set, resulting in 73 285 frames for training, 1800 for model selection and 13 330 for testing. The last sequence does not overlap with any of the training sequence and model selection and testing partitions are spatially distinct.

B. Metrics

We follow the literature and evaluate the perceptual quality of the generated images (at resolutions 64^2, 256^2, 512^2) using Fréchet Inception Distance (FID) [33] and Kernel Inception Distance (KID) [34], and Fréchet Video Distance (FVD) [35] for the sequence counterpart. To measure the ability of the model to reconstruct the semantic structure of the scene, we follow [1] and report the Mean Intersection Over Union (mIoU) between the predicted ground-view and GT ground-view semantic maps. Similarly, we report mIoU-alignment, which evaluates the mIoU between ground-view semantic maps extracted from the predicted ground-view images and GT ground-view RGB images, using a pre-trained segmentation model (Deeplabv3+ [36]). The goal is to measure how well the generated ground-view images perform on a segmentation downstream task as compared to the real images, but without introducing undesired variance from the performance of the pre-trained segmentation model itself. Similarly to [1], we report Root Mean Squared Error (RMSE) (metres) between the predicted upscaled ground-view and GT ground-view depth maps to check the accuracy of the synthesised depth. Both single-view and multi-view experiments are evaluated on multi-view data.

C. Baselines

We compare our model with recent work such as NeuralFloors [1] and competing method CC3D [3], but also prior art GSN [18]. The GSN model is trained unconditionally based on normally-distributed noise following the original training scheme, using the extended dataset containing dynamic objects. For CC3D [3], we report two sets of results, one after running the official pre-trained model on our validation data (CC3D pre-trained) and one after training the model from scratch (CC3D trained) – both are either evaluated or trained and evaluated on the extended dataset. Finally, we report our new proposed approaches NeuralFloors++ (SD) and NeuralFloors++ (T2I-adapter) in comparison to NeuralFloors [1], which is trained and evaluated on the previous static dataset.

D. Implementation details

Stage 1 encoders (ε^{BEV}, ε^{style}), decoders (δ^{BEV}, δ^{GV}) and stage 2 VAE and LDM of NeuralFloors++ (SD) are initialised from publicly available weights (SD-v-1-4) [30]. The NeuralFloors++ (SD) LDM is extended with two convolutional layers (ϕ) that embed the conditional inputs, which are concatenated with the standard noisy latent variable and fed into the LDM denoising network. Stage 2 of NeuralFloors++ (T2I-adapter) is the implementation of [25], where we train the Adapter (ϕ) but also fine-tune the UNet, which is initialised from SD-XL-v1.0. For the segmentation-discriminator, we use the OASIS discriminator [37] backbone, which now receives as input a one-hot encoding of the segmentation map and outputs a real/fake signal.

We set $t_n = 0$ and $t_f = 80$ meters. The BEV map has a size of 512×512, covering an area of 80×80 meters. Encoder ε^{BEV} receives input of shape $25 \times 512 \times 512$ and outputs W^{BEV} of shape $96 \times 128 \times 128$. Decoder δ^{BEV} receives W^{BEV} and outputs $24 \times 512 \times 512$. We sample 2136 rays with 200 points per ray. D^{GV} shape is $1 \times 24 \times 89$ and F^{GV} is $16 \times 24 \times 89$. Decoder δ^{GV} receives as input 17 channels and outputs $28 \times 192 \times 712$. Discriminator $\mathcal{D}$ receives input of $20 \times 192 \times 712$ and outputs $1 \times 192 \times 712$. Stage 1 output is resized, center-cropped and zero-padded to 512×512. Stage 2 NeuralFloors++ (SD) receives input of shape $29 \times 512 \times 512$, downscaled to $29 \times 256 \times 256$ where ϕ encodes to $4 \times 64 \times 64$ before it is concatenated with the noise, and finally outputs an image of size $3 \times 512 \times 512$. Stage 2 NeuralFloors++ (T2I-adapter) receives inputs at the native 192×712 output resolution of Stage 1 and outputs an image of size 192×712. Style encoder ε^{style} receives the original KITTI-360 image of shape $3 \times 376 \times 1408$ and outputs $4 \times 47 \times 176$ which is upscaled bi-linearly to $4 \times 376 \times 1408$. The bank of style embeddings consists of $269,843$ latent codes organised by class.

E. Training details

We empirically set the weights α of rare classes *[bicycle, person, rider, pole, traffic sign]* to *[4.0, 5.0, 4.0, 4.0, 4.0]*. We use batch size 1 and Adam optimiser. Each stage model is trained individually on 4 NVIDIA V100 GPUs with 32 GB of VRAM. Inference takes 0.33s, 29s and 5.5s per observation for Stage 1, Stage 2 NeuralFloors++ (SD) and Stage 2 NeuralFloors++ (T2I-adapter) respectively, on a single GPU. Stage 1 is trained with a learning rate of 10^{-4}, uses the DeepSpeed [38] library and has 117 M trainable parameters. Stage 2 of NeuralFloors++ (SD) is trained with a learning rate of 10^{-4}, $t = 50$ diffusion steps at inference and has 943.2 M params, while stage 2 NeuralFloors++ (T2I-adapter) uses $t = 20$ steps and has 2.737 B params. We use the following weights to scale losses: $\lambda_{\mathcal{L}_S} = 10$, $\lambda_{\mathcal{L}_J} = 10$, $\lambda_{\mathcal{L}_C} = \lambda_{\mathcal{L}_O} = 100$, $\lambda_{\mathcal{L}_D} = \lambda_{\mathcal{L}_{D_+}} = 1$, $\lambda_{\mathcal{L}_A} = 1$.

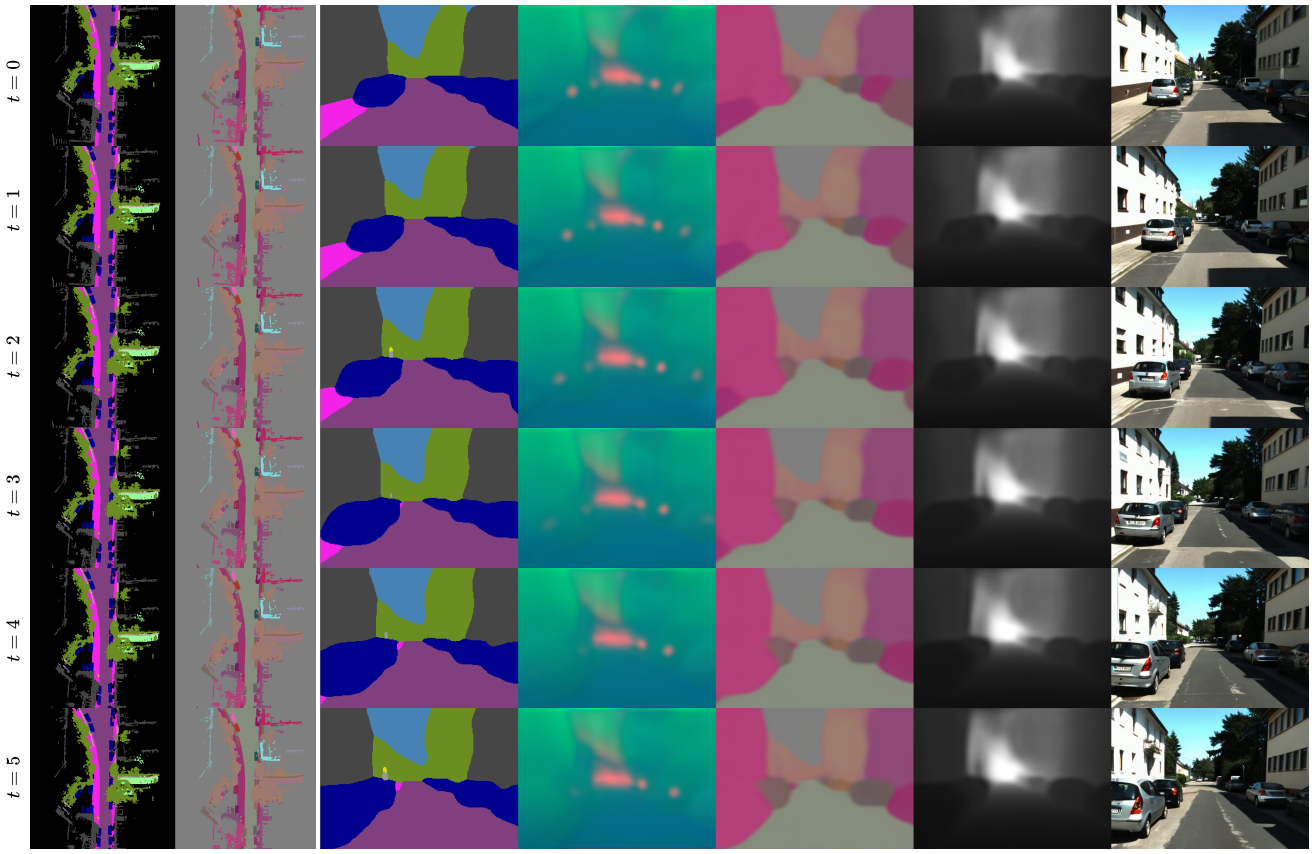


Fig. 3: Consecutive frame predictions from BEV inputs with NeuralFloors++ (T2I-adapter) model.

Model	mIoU ↑	mIoU-align ↑	RMSE ↓	FID ↓ 64^2	FID ↓ 256^2	FID ↓ 512^2	KID ↓ 64^2	KID ↓ 256^2	KID ↓ 512^2	FVD ↓ 64^2	FVD ↓ 256^2	FVD ↓ 512^2
GSN	-	-	-	203.42	-	-	0.2512(46)	-	-	1254.77	-	-
CC3D pre-trained	-	10.12	-	55.24	104.59	-	0.0396(11)	0.0905(18)	-	486.88	1043.94	-
CC3D trained	-	11.47	-	97.05	125.85	-	0.0503(28)	0.0853(25)	-	491.80	1004.41	-
NeuralFloors	32.06	28.11	6.897	42.23	66.50	65.81	0.0266(11)	0.0351(12)	0.0322(12)	454.95	834.95	868.29
NeuralFloors++ (SD)	**35.82**	29.15	**6.730**	25.49	36.51	48.63	0.0157(11)	0.0166(8)	0.0229(9)	**206.90**	603.81	600.29
NeuralFloors++ (T2I-adapter)	**35.82**	**29.33**	**6.730**	**20.60**	**29.41**	**42.55**	**0.0149(11)**	**0.0162(10)**	**0.0219(11)**	208.57	**386.07**	**397.27**

TABLE I: Baselines comparison regarding segmentation, depth and perceptual quality at different image sizes. Parentheses indicate smallest digit uncertainty – e.g. 0.0396 ± 0.0011 in the second row for KID.

V. Results

In Tab. I, as evaluated by FID (203.42), KID (0.2512) and FVD (1254.77), GSN seems to be the least performing model, as also emphasized in [3] and [1]. Compared to NeuralFloors, our proposed model performs better in terms of semantic correctness, as both mIoU and mIoU-alignment have improved from 32.06 to 35.82 and from 28.11 to 29.33 respectively, while RMSE dropped from 6.897 to 6.730, highlighting the benefit of improved complex data and multi-view training. In terms of perceptual quality of both the generated images and video – FID, KID and FVD are improved at all resolutions. In terms of semantic structure, since CC3D [3] does not output ground-view segmentation map nor it is trained for the output to follow a particular ground-view semantic structure, we can only compare the segmentation of the images from the off-the-shelf segmenter. In this case, both CC3D pre-trained and CC3D trained models have close mIoU-alignment of 10.12 and 11.47 respectively, which are much lower than our 29.33. Regarding perceptual quality, metrics also significantly improve, especially at the 256^2 resolution. While these results are reported for our data, which makes use of KITTI-360 to the full extent, CC3D [3] also report an FID of 65.6 at the 256^2 resolution on their curated dataset, where turning cases are removed.

As it can be noted in Fig. 3, NeuralFloors++ (T2I-adapter) can synthesise consecutive frames of high quality and diverse modalities. Additionally, in Figs. 4 and 5 we present qualitative results of style control and compositionality.

Fig. 4: We demonstrate style control by randomly sampling 2 sets of latent codes (*2nd* and *4th* columns) for the objects, using the same BEV semantic map. In the generated ground-view images, the layout of objects is consistent, while the style is distinct.

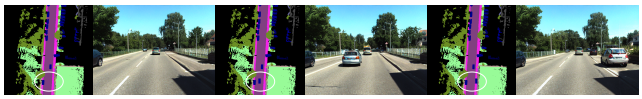


Fig. 5: We demonstrate compositionality and editability capabilities by manipulating the input map. We add and move a car from left to right in the input BEV semantic map re-rendered the scene.

VI. Ablation Study

We conduct an ablation study to evaluate and justify the choice of our final model, by iteratively adding new features to a base case, with + indicating incremental additions to the settings of the previous row.

Since we use a real-world dataset, some classes are less common. Thus, in Tab. II, we analyse the effects of assigning a larger weight to under-represented classes (small-class weighting), compared to the baseline with equal weighting (equal weighting) and observe an improvement in mIoU from 32.69 to 33.74. We additionally investigate whether a model with larger capacity (F^{GV} from $4 \times 24 \times 89$ to $16 \times 24 \times 89$) leads to improvements in mIoU (+ larger model).

Model	mIoU ↑
equal weighting	32.69
small-class weighting	33.74
+ larger model	33.89

TABLE II: Single-view ablations group 1.

In Tab. III, starting from small-class weighting as a base setup, we find that both enforcing BEV semantic reconstruction (+ BEV segmentation reconstruction) and adding BEV instance information (+ input BEV instance map) lead to an increase in mIoU. We additionally generate an upscaled depth map from decoder δ^{GV} (+ depth upscaled). The addition of this output leads to a reduction in mIoU, but represents an important component of the set of generated ground-view data, so we include it in the final model setup.

Model	mIoU ↑
small-class weighting	33.74
+ BEV segmentation reconstruction	34.20
+ input BEV instance map	34.76
+ depth upscaled	33.65

TABLE III: Single-view ablations group 2.

In Tab. IV, starting from small-class weighting, we check the effects of adding BEV style maps as input, and style map reconstruction (+ style input & reconstruction). We observe a drop in performance (from 33.74 to 25.97) and investigate whether this is due to a model bottleneck by employing a larger model (F^{GV} from $4 \times 24 \times 89$ to $16 \times 24 \times 89$) (+ larger model). We observe a recovery in mIoU (34.49) and notice that the initial base setup did not suffer from the same drawback (+ larger model in Tab. II, 33.89). We conclude that this is an indication of a bottleneck when diversifying the output signal. Finally, we analyse the performance of a single-view model with all components introduced so far (+ all components): small-class weighting, BEV segmentation reconstruction, input BEV instance map, upscaled depth prediction, segmentation discriminator, BEV style map input and ground-view style map reconstruction on a larger model, leading to the best performance in this ablation group (34.55).

Model	mIoU ↑
small-class weighting	33.74
+ style input & reconstruction	25.97
+ larger model	34.49
+ all components	34.55

TABLE IV: Single-view ablations group 3.

In Tab. V, starting from small-class weighting setup, we first train a simple multi-view model (+ multi-view) and set a new baseline of mIoU 35.36. As outlined in Sec. IV-A, our multi-view data is comprised of observations from the original poses, which are dense and contain the class sky, and random observations around these, which come from sparse LiDAR point clouds, with sky segmentation being absent. We empirically observe that our model learns to output dense segmentation maps with sky present for the original poses but fails to synthesize sky class in any of the random observations. To mitigate this, for each observation set, we randomly select one of the camera poses and the BEV associated with it, and express the poses of all other cameras with respect to this new reference camera (+ randomised). As mIoU cannot be computed for class sky in the random views, we conduct a qualitative inspection. We then add all of the single view components (+ all components), which leads to an mIoU of 34.93. Finally, we add the segmentation discriminator (+ segmentation discriminator), leading to a final mIoU of 35.82. We select this as our final model.

Model	mIoU ↑
small-class weighting	33.74
+ multi-view	35.36
+ randomised	35.46
+ all components	34.93
+ segmentation discriminator	35.82

TABLE V: Multi-view ablations.

In Tab. VI, we start with a baseline of FVD 868.29 based on the previous approach (NeuralFloors). We test the benefit of additionally conditioning the synthesis model via instance-wise style maps (NeuralFloors + style) and note an improvement in FVD. As consecutive frames are highly correlated, we take advantage of this characteristic by conditioning the synthesis of the current frame $\hat{I}_t^{GV}$ on the previous generated frame $I_{past}^{GV} = \hat{I}_{t-1}^{GV}$ (NeuralFloors + AR(1)). However, as this ablation is evaluated on the output data from stage 1, style and content drifts that are present in initial frames possibly affect the synthesis of subsequent frames, leading to less perceptually-pleasing results. We test a similar setup where the extra conditional input frame is randomly chosen between $t-1$ and $t-8$ (NeuralFloors + AR(*)) and note an improvement in FVD to 692.36. Finally, we evaluate both a past generated frame and the style map conditioning (NeuralFloors + style + AR(*)), and observe an

improved FVD of 600.29. We select this as our final model.

Model	FVD ↓
NeuralFloors	868.29
NeuralFloors + style	668.48
NeuralFloors + AR(1)	1167.64
NeuralFloors + AR(*)	692.36
NeuralFloors + style + AR(*)	600.29

TABLE VI: Stage 2 ablations evaluated at 512^2, on the output of stage 1.

VII. Conclusions

We have presented improvements in multiple-view consistency, style manipulation, and fine object-level control when synthesising ground-view images from BEV representations. These contributions have lead to improved realism and alignment for the generated images and associated modalities. With the addition of scene compositionality and style control, the proposed system has important applications in training AD downstream tasks without the need of a simulator. It is worth noting that our system may encounter challenges when dealing with tunnels or other structures that obscure the scene entirely. Although in this study we focus on 2D representations for simplicity and useability, other BEV inputs (depth or height) can be used to improve performance.

References

[1] V. Muşat, D. D. Martini, M. Gadd, and P. Newman, "Neuralfloors: Conditional street-level scene generation from bev semantic maps via neural fields," *IEEE Robotics and Automation Letters*, 2024.

[2] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018.

[3] S. Bahmani, J. J. Park, D. Paschalidou, X. Yan, G. Wetzstein, L. J. Guibas, and A. Tagliasacchi, "Cc3d: Layout-conditioned generation of compositional 3d scenes," *ArXiv*, vol. abs/2303.12074, 2023.

[4] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *ArXiv*, vol. abs/2109.13410, 2021.

[5] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *CoRL*, 2017.

[6] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. Liu, *et al.*, "Holodeck: Language guided generation of 3d embodied ai environments," in *CVPR*, 2024.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[8] H. Alhaija, S. Mustikovela, A. Geiger, and C. Rother, *Geometric Image Synthesis*, 2019.

[9] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[10] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *CVPR*, 2018.

[11] V. Musat, D. D. Martini, M. Gadd, and P. Newman, "Depth-sims: Semi-parametric image and depth synthesis," *International Conference on Robotics and Automation*, 2022.

[12] P. Henzler, N. J. Mitra, and T. Ritschel, "Escaping plato's cave: 3d shape from adversarial rendering," in *The IEEE International Conference on Computer Vision*, October 2019.

[13] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "Hologan: Unsupervised learning of 3d representations from natural images," in *ICCV*, 2019.

[14] T. H. Nguyen-Phuoc, C. Richardt, L. Mai, Y. Yang, and N. Mitra, "Blockgan: Learning 3d object-aware scene representations from unlabelled images," *Advances in neural information processing systems*, 2020.

[15] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2021.

[16] Y. Xue, Y. Li, K. Singh, and Y. Lee, "Giraffe hd: A high-resolution 3d-aware generative model," in *CVPR*, 2022.

[17] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. CVPR*, 2020.

[18] T. DeVries, M. Á. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind, "Unconstrained scene generation with locally conditioned radiance fields," *CoRR*, vol. abs/2104.00670, 2021.

[19] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein, "Efficient geometry-aware 3D generative adversarial networks," in *arXiv*, 2021.

[20] Q. Zhang, Y. Xu, Y. Shen, B. Dai, B. Zhou, and C. Yang, "BerfScene: Generative novel view synthesis with 3D-aware diffusion models," in *arXiv*, 2023.

[21] C. Lin, H.-Y. Lee, W. Menapace, M. Chai, A. Siarohin, M.-H. Yang, and S. Tulyakov, "Infinicity: Infinite-scale city synthesis," 2023.

[22] A. Swerdlow, R. Xu, and B. Zhou, "Street-view image generation from a bird's-eye view layout," *ArXiv*, vol. abs/2301.04634, 2023.

[23] K. Yang, E. Ma, J. Peng, Q. Guo, D. Lin, and K. Yu, "Bevcontrol: Accurately controlling street-view elements with multi-perspective consistency via bev sketch layout," *ArXiv*, vol. abs/2308.01661, 2023.

[24] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *IEEE International Conference on Computer Vision*, 2023.

[25] C. Mou, X. Wang, L. Xie, J. Zhang, Z. Qi, Y. Shan, and X. Qie, "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," *ArXiv*, vol. abs/2302.08453, 2023.

[26] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra, "Instancediffusion: Instance-level control for image generation," *arXiv preprint arXiv:2402.03290*, 2024.

[27] X. Wang, H. Yuan, S. Zhang, D. Chen, J. Wang, Y. Zhang, Y. Shen, D. Zhao, and J. Zhou, "Videocomposer: Compositional video synthesis with motion controllability," in *International Conference on Neural Information Processing Systems*, 2023.

[28] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, Y. Li, T. Michaeli, O. Wang, D. Sun, T. Dekel, and I. Mosseri, "Lumiere: A space-time diffusion model for video generation," *ArXiv*, vol. abs/2401.12945, 2024.

[29] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *CVPR*, 2020.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021.

[31] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[32] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[33] 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," in *International Conference on Neural Information Processing Systems*, 2017.

[34] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *International Conference on Learning Representations*, 2018.

[35] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Fvd: A new metric for video generation," in *DGS@ICLR*, 2019.

[36] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[37] E. Schönfeld, V. Sushko, D. Zhang, J. Gall, B. Schiele, and A. Khoreva, "You only need adversarial supervision for semantic image synthesis," in *International Conference on Learning Representations*, 2021.

[38] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	NeuralFloors++: Consistent Street-Level Scene Generation From BEV Semantic Maps
Publication Status	Published
Publication Details	**V. Muşat**, D. De Martini, M. Gadd and P. Newman, "NeuralFloors++: Consistent Street-Level Scene Generation From BEV Semantic Maps," *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, United Arab Emirates, 2024, pp. 12872-12879, doi: 10.1109/IROS58592.2024.10802002.

Student Confirmation

Student Name:	Valentina Musat		
Contribution to the Paper	- developed the idea supporting the paper under the guidance of Dr. De Martini and Dr. Gadd, and supervision of Prof. Newman - implemented the architecture - compiled the data for training - ran all experiments and interpreted data - wrote the manuscript - prepared presentation materials and a presentation video for the conference - presented the paper at IROS remotely		
Signature		Date	21.04.2025

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman		
Supervisor comments I certify that the candidate made a substantial and leading contribution to the publication, and that the description described above is accurate		
Signature		Date 24/04/25

7.3 Further insights

Similar to the previous approach presented in Chapter 6, the proposed method, NeuralFloors++, did not leverage the ability of the latent diffusion model to be conditioned on text prompts, since the UNet was not fine-tuned with meaningful text captions. This is because no such captions existed for the datasets at the time - instead having to choose a constant, generic prompt.

The capability of the the family of Stable Diffusion models to be prompted with text can be retained, by choosing to bypass fine-tuning the UNet for the version that uses the T2I-Adapter (Mou et al., 2024). This would enable prompted weather addition, and could be used along with the auto-regressive formulation. Additionally, if training data with weather conditions and text captions associated with the weather condition is available, the UNet can be fine-tuned. Alternatively, the T2I-Adapter can receive an additional RGB input acting as a style cue.

One of the drawbacks of using an auto-regressive model in the second stage is style drift, where each subsequent generated frame introduces a small amount of error in terms of stylistical consistency, which is then further amplified when the generated frame is used for conditioning the next step. As shown in Table VI in the integrated paper manuscript, introducing additional conditioning (in the presence of an auto-regressive model) in the form of instance-wise average style features improves temporal consistency by approximately 13 percent, as measured by FVD. Intuitively, the addition of style features leads to improvements by providing coarse (being instance-averaged) and consistent style cues.

The principle of data requirements separation between stages is continued and applied to the style features as well. While the first stage is trained with features that are extracted from RGB data that would normally be used to train the second stage, two distinctions should be clarified. Firstly, at train time, for each iteration, the features are randomly sampled from the bank of styles, ensuring that the first stage model does not learn a correlation between specific objects, their location and any corresponding feature, intuitively encouraging the model to instead learn to lift and transfer arbitrary input style features into their correct location in the output ground-view feature map. Secondly, at inference time, the features can be re-extracted from a larger dataset for more diversity, or from a specialised dataset for post-hoc style alignment.

To enhance style consistency, we thus design a process of extracting features, creating a bank of style embeddings from which to further sample. This process does not employ a structure and style disentanglement architecture for extracting style embeddings, instead relying on the variational auto-encoder from Stable Diffusion (Rombach et al., 2021) in order to extract meaningful embeddings

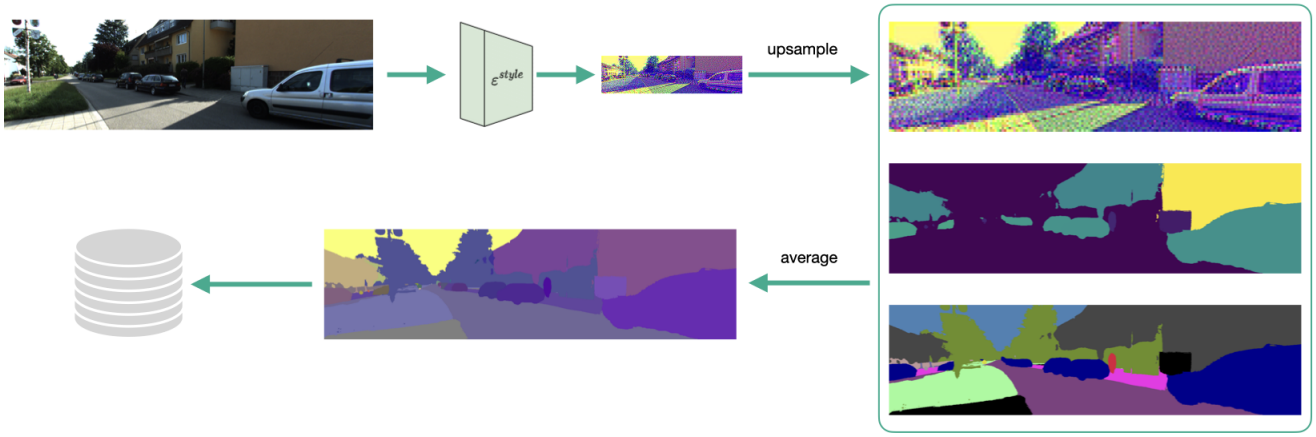


Figure 7.3: Style encoding process: an RGB image is first encoded to obtain style embeddings and then upsampled back to the original resolution. Using its corresponding semantic and instance segmentation maps, the style embedding of each object is averaged and saved in a database.

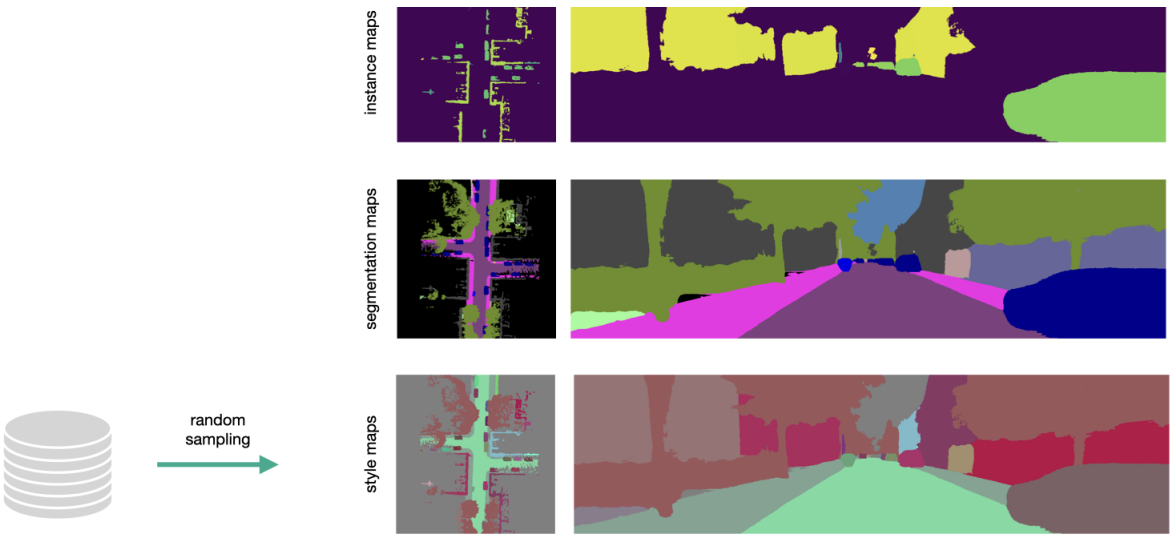


Figure 7.4: Style embeddings are sampled from the style bank, then BEV and ground-view style maps are constructed according to their semantic and instance information.

from RGB images. More specifically, features are extracted using the encoder, then upsampled to the original image resolution. After being averaged across each instance using the corresponding instance map associated with the input RGB image, they are finally saved in a feature bank, categorised by class, as visually explained in Figure 7.3. At train time, a feature is randomly sampled for each instance ID and each class belonging to an observation, and both the associated BEV and the ground-view feature maps are populated with corresponding features based on either the instance segmentation map (for objects with an instance ID) or the semantic segmentation map (for background classes), as visually explained in Figure 7.4. At inference time, the features are sampled only once, and kept

Figure 7.5: An example of a BEV semantic map with different ground-view RGB styles that have been randomly sampled using the BEV style map.

constant throughout the entire sequence or traversal to ensure consistency.

Furthermore, the use of instance- or class-wise features allows the style of generated images to be controlled with a better level of granularity compared to global style embeddings, as shown in the examples in Fig.7.5.

In terms of temporal coherence, the approach presented in this chapter was evaluated using FVD. However, recent literature such as a study by G. Y. Luo et al. (2025) has highlighted that FVD may be more strongly correlated with individual frame fidelity rather than cross-frame consistency. As such, JEDi has been proposed as a new metric, as it is better correlated with both temporal consistency and human visual perception, since it uses both a new feature extractor based on V-JEPA (Bardes et al., 2024) and replaces the Fréchet Distance with MMD. Similar to FVD, a lower JEDi score indicates that the synthesised videos are more closely aligned with real video distributions, reflecting a high-quality generative model. We thus employ JEDi on the synthesised test sequence, where the metric is applied on snippets of length of 10 frames. The results reported in Tab. 7.1 indicate that both versions of the NeuralFloors++ model outperform the previous architecture NeuralFloors, and competing model CC3D (Bahmani et al., 2023).

Model	JEDi ↓
CC3D pre-trained	16.54
CC3D trained	13.96
NeuralFloors	10.54
NeuralFloors++ (SD)	4.28
NeuralFloors++ (T2I-adapter)	4.36

Table 7.1: Results using the JEDi metric.

Finally, an alternative (and contemporary) approach to the proposed method is MagicDRIVE (Gao et al., 2024b), which models cross-frame temporal consistency as a sequence of embeddings

obtained from a BEV map, 3D annotations and camera poses, which are then used to condition a UNet for multi-view image or video generation. However, this method does not generate any additional dense ground-truth beyond what is used as an input.

8

Discussion

Contents

8.1 Summary of contributions

The work presented in this manuscript focuses on methods that generate data for various use cases across a wide spectrum of autonomous driving applications, starting with a method for multiplying existing image data by adding weather effects and changes in levels of illumination in **Multi-weather city**.

While the initial publication focuses on a specific set of weather domains, the method presented is applicable to a larger spectrum, as long as a minimum amount of paired or unpaired data is available. Employing both GANs and Cycle-GANs provides a high degree of flexibility in terms of data sources, while their outputs can be further combined with analytical methods in order to extend to domains

beyond what is available through the input training data alone.

The data generated with this method was thoroughly evaluated, by using it to train an object detection model and an instance segmentation model, and testing them on real-weather datasets, in some cases exceeding 10 percentage points increase in mean AP compared to baselines. This approach, on the other hand, is not able to change the structure of the scenes represented in the input images, but this is further tackled in the next publication.

As such, in **Depth-SIMS**, the focus is placed on scene compositing, while maintaining realism and quality of associated ground-truth. The approach builds on semi-parametric prior art, by making use of a bank of blobs, while being conditioned on semantic and instance segmentation maps. However, as opposed to the prior method, if offers the ability to generate depth and semantic maps that are aligned with the RGB output. The suitability of the generated data as training data was also evaluated, showing improvements in semantic segmentation and depth completion models.

Unfortunately, since the method operates directly on 2D images, it is limited in its ability to reason about complex interactions such as occlusion, and requires a guiding ground-view semantic segmentation map as input, making scalable synthesis difficult. This limitation is further tackled in the next publication.

In **Neuralfloors**, a two-stage approach is proposed to split the problem of image generation into two distinct steps. The first stage reasons about scene structure and geometry by lifting the structure of a BEV representation into a 3D representation using a neural field approach, while the second stage employs a latent diffusion model responsible for the appearance of the generated scene, given the outputs from the first stage. The input to this pipeline is a BEV semantic segmentation map, which is a popular representation adopted by many modern perception and planning approaches, and which is easier to generate and edit than its ground-view alternative.

Similar to the previous method, the proposed approach is designed to output ground-view RGB images and aligned ground-truth in the form of depth, semantic and instance segmentation maps. It is further evaluated against prior art and competing methods, showing significant improvement in both visual quality and structural alignment.

Unfortunately, the method lacks an explicit mechanism to ensure temporal consistency, making it less applicable to downstream tasks that rely on frame sequences, this being further addressed in the next publication.

In **Neuralfloors++**, the previous method is improved by tackling both structural and stylistic consistency: initially, in the first stage, through training on multi-view data and via the introduction of a style map for appearance cues, followed by auto-regressive training in the second stage, where the past generated RGB image is used as an additional input for the next timestep. The output data is further evaluated for both individual frame quality using the same metrics employed in the previous publication, but also for temporal consistency using FVD and JEDi, showing improved quality.

Finally, similar to other examples from prior art, these approaches can also be combined, in order to further extend their applicability.

8.2 Future work

As explored by previous 3D-based methods (and discussed in Section 4.3), a suitable choice for 3D-informed weather synthesis are NeRFs, which make use of a light transmittance model and learn the structure of the scene by estimating a volume of densities and colors. The method approximates the image formation process by sampling rays of light and computing accumulated transmittance along the rays as a function of densities, and can deal with occlusions, which is especially important for modeling weather particles such as, for example, snowflakes. Although it isn't a comprehensive physics particle simulator, it could exceed 2D weather image synthesis models in terms of expresiveness and control, without introducing too much complexity.

The proposal is that a special-purpose NeRF-based architecture can be coupled with a pre-trained NeRF-MLP to produce 3D reconstructions of a scene under various weather conditions, without requiring any training on weather images of that specific scene. Prior methods for this exist in recent NeRF literature, with ScatterNeRF (Ramazzina et al., 2023) tackling the reconstruction of foggy scenes using a combination of two MLPs, one that learns the clean, non-foggy scene, and one that learns to reconstruct the foggy medium. While approaches such as ScatterNeRF (Ramazzina et al., 2023) or SeaThru-NeRF (Levy et al., 2023) are focused on 3D reconstruction and weather **removal**, the proposed method instead focuses on **adding** weather effects to existing 3D scenes parametrized as NeRF-MLPs.

For the purpose of representing a scene that is degraded by weather, a simplifying assumption is that everything happens between the camera and the surfaces that make up the scene. This can be organised into three categories:

1. Atmospheric weather (e.g. fog, rain, snowflakes, dust particles) can roughly be represented as 2 types: those that act through scattering, such as fog, and those that act as opaque occluders, such as snowflakes, dust or rain, depending on the distance to the sensor;

2. Adherent weather on object surfaces (e.g. accumulated snow, puddles) can roughly be modeled as accumulation of material on surfaces. Thus accumulated weather can be considered as a new surface on top of the original surface, with a different feature (i.e. color);

3. Adherent weather on lens (e.g. droplets, snowflakes) can be roughly represented as 2 types: those that act as a secondary lens, where the rays will change direction, and those that act as occluders, where the opacity is increased.

For all weather effects, the goal is to avoid explicit analytical models (physics simulation of individual weather effects at particle level) and instead learn from the data, which can be hopefully done through NeRF models.

In the proposed architecture, a NeRF is first trained on overcast data (*overcast MLP*). The pre-trained *overcast MLP* is then **frozen**, combined with a second NeRF (*weather MLP*), and the architecture is trained on adverse weather data, with only the *weather MLP* being optimised. More specifically, the outputs of the 2 MLPs can be combined residually to obtain the final per-point ray outputs: colors and densities. A volumetric renderer is then used to produce final (weather-affected) images, on which weak supervision via an adversarial loss is applied. The advantage of using weak supervision is that it alleviates the strict requirements of needing paired, pixel-aligned overcast and weather-affected images, which are very difficult to obtain in the real world.

Following the simplifying assumptions from above, the intuition is that the frozen *overcast MLP* will output densities and colors specific to the original scene, while the *weather MLP* will generate densities and colors corresponding to weather effect being added.

In order to ensure that the *weather MLP* learns meaningful properties of the scene under a specific weather condition, an orthogonality loss (direct orthogonality loss and gram matrix orthogonality loss) can be applied between densities (and colors) predicted by the 2 MLPs. Inspired by L. Liu et al. (2021), the coupling of an orthogonality loss between the density outputs of the two MLPs with the residual combination of their outputs could lead to the MLPs learning complementary but different signals. The goal here is to have the *overcast MLP* output densities as a function of the location of scene

surfaces in absence of any weather effects, while having the *weather MLP* output densities indicating the placement of weather effects in between the scene surfaces and the camera lens.

An added benefit of employing a NeRF-based architecture is that similar to the approach in NeuRAD (Tonderski et al., 2024), LiDAR can be incorporated as well through the addition of a second renderer or second feature decoder, enabling multi-sensor modeling.

Finally, the possible disadvantage is that weak supervision enforces a distribution of the images, rather than pixel-wise accuracy, and may not be a strong enough signal to learn plausible or correct weather effects. The mitigation for this is to use stronger supervision, but this comes with a stricter data requirement of having paired overcast and weather images. In this case, datasets such as SemanticSpray (Piroli et al., 2023) or WADS (Kurup & Bos, 2023) may be particularly useful.

References

Alberti, Emanuele; Tavera, Antonio; Masone, Carlo, and Caputo, Barbara, 2020. IDDA: A Large-Scale Multi-Domain Dataset for Autonomous Driving. *IEEE Robotics and Automation Letters*. Vol. 5, no. 4, pp. 5526–5533. Available also from: https://doi.org/10.1109/LRA.2020.3009075.

Anoosheh, Asha; Sattler, Torsten; Timofte, Radu; Pollefeys, Marc, and Van Gool, Luc, 2019. Night-to-day image translation for retrieval-based localization. In: *2019 International conference on robotics and automation (ICRA)*. IEEE, pp. 5958–5964. Available also from: https://doi.org/10.1109/ICRA.2019.8794387.

Bahmani, Sherwin; Park, Jeong Joon; Paschalidou, Despoina; Yan, Xingguang; Wetzstein, Gordon; Guibas, Leonidas, and Tagliasacchi, Andrea, 2023. CC3D: Layout-Conditioned Generation of Compositional 3D Scenes. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 7137–7147. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00659.

Bai, Xiangyu; Luo, Yedi; Jiang, Le; Gupta, Aniket; Kaveti, Pushyami; Singh, Hanumant, and Ostadabbas, Sarah, 2024. Bridging the Domain Gap between Synthetic and Real-World Data for Autonomous Driving. *ACM J. Auton. Transport. Syst.* Vol. 1, no. 2. Available also from: https://doi.org/10.1145/3633463.

Bar-Tal, Omer; Chefer, Hila; Tov, Omer; Herrmann, Charles; Paiss, Roni; Zada, Shiran; Ephrat, Ariel; Hur, Junhwa; Liu, Guanghui; Raj, Amit; Li, Yuanzhen; Rubinstein, Michael; Michaeli, Tomer; Wang, Oliver; Sun, Deqing; Dekel, Tali, and Mosseri, Inbar, 2024. Lumiere: A Space-Time Diffusion Model for Video Generation. In: *SIGGRAPH Asia 2024 Conference Papers*. Tokyo, Japan: Association for Computing Machinery. SA '24. Available also from: https://doi.org/10.1145/3680528.3687614.

Bardes, Adrien; Garrido, Quentin; Ponce, Jean; Chen, Xinlei; Rabbat, Michael; LeCun, Yann; Assran, Mido, and Ballas, Nicolas, 2024. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*. Available also from: https://doi.org/10.48550/arXiv.2404.08471.

Barnes, Connelly; Shechtman, Eli; Finkelstein, Adam, and Goldman, Dan B., 2023. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 1st ed. New York, NY, USA: Association for Computing Machinery. Available also from: https://doi.org/10.1145/3596711.3596777.

Barratt, Shane and Sharma, Rishi, 2018. A note on the inception score. *ArXiv*. Vol. abs/1801.01973. Available also from: https://doi.org/10.48550/arXiv.1801.01973.

Bautista, Miguel Angel; Guo, Pengsheng; Abnar, Samira; Talbott, Walter; Toshev, Alexander; Chen, Zhuoyuan; Dinh, Laurent; Zhai, Shuangfei; Goh, Hanlin; Ulbricht, Daniel; Dehghan, Afshin, and Susskind, Josh, 2022. GAUDI: a neural architect for immersive 3D scene generation. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc. NIPS '22. Available also from: https://dl.acm.org/doi/10.5555/3600270.3602090.

Behley, Jens; Garbade, Martin; Milioto, Andres; Quenzel, Jan; Behnke, Sven; Stachniss, Cyrill, and Gall, Jurgen, 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9297–9307. Available also from: https://arxiv.org/abs/1904.01416.

Bijelic, Mario; Gruber, Tobias; Mannan, Fahim; Kraus, Florian; Ritter, Werner; Dietmayer, Klaus, and Heide, Felix, 2020. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 11679–11689. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01170.

Bińkowski, Mikołaj; Sutherland, Danica J; Arbel, Michael, and Gretton, Arthur, 2018. Demystifying MMD GANs. In: *International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.1801.01401.

Blattmann, Andreas; Dockhorn, Tim; Kulal, Sumith; Mendelevitch, Daniel; Kilian, Maciej; Lorenz, Dominik; Levi, Yam; English, Zion; Voleti, Vikram; Letts, Adam; Jampani, Varun, and Rombach, Robin, 2023. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv preprint arXiv:2311.15127*. Available also from: https://doi.org/10.48550/arXiv.2311.15127.

Blattmann, Andreas; Rombach, Robin; Oktay, Kaan; Müller, Jonas, and Ommer, Björn, 2022. Semi-Parametric Neural Image Synthesis. *arXiv preprint arXiv:2204.11824*. Available also from: https://doi.org/10.48550/arXiv.2204.11824.

Brock, Andrew; Donahue, Jeff, and Simonyan, Karen, 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In: *International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.1809.11096.

Brostow, Gabriel J.; Fauqueur, Julien, and Cipolla, Roberto, 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*. Vol. 30, no. 2, pp. 88–97. Available also from: https://doi.org/10.1016/j.patrec.2008.04.005.

Burnett, Keenan; Yoon, David J; Wu, Yuchen; Li, Andrew Z; Zhang, Haowei; Lu, Shichen; Qian, Jingxing; Tseng, Wei-Kang; Lambert, Andrew; Leung, Keith YK; Schoellig, Angela P, and Barfoot, Timothy D, 2023. Boreas: A multi-season autonomous driving dataset. *The International Journal of Robotics Research*. Vol. 42, no. 1-2, pp. 33–42. Available also from: https://doi.org/10.1177/02783649231160195.

Caesar, Holger; Bankiti, Varun; Lang, Alex H.; Vora, Sourabh; Liong, Venice Erin; Xu, Qiang; Krishnan, Anush; Pan, Yu; Baldan, Giancarlo, and Beijbom, Oscar, 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 11618–11628. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01164.

Carreira, João and Zisserman, Andrew, 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733. Available also from: https://doi.org/10.1109/CVPR.2017.502.

Chan, Eric R.; Lin, Connor Z.; Chan, Matthew A.; Nagano, Koki; Pan, Boxiao; de Mello, Shalini; Gallo, Orazio; Guibas, Leonidas; Tremblay, Jonathan; Khamis, Sameh; Karras, Tero, and Wetzstein, Gordon, 2022. Efficient Geometry-aware 3D Generative Adversarial Networks. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16102–16112. Available also from: https://doi.org/10.1109/CVPR52688.2022.01565.

Chang, Ming-Fang; Lambert, John; Sangkloy, Patsorn; Singh, Jagjeet; Bąk, Sławomir; Hartnett, Andrew; Wang, De; Carr, Peter; Lucey, Simon; Ramanan, Deva, and Hays, James, 2019. Argoverse: 3D Tracking and Forecasting With Rich Maps. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8740–8749. Available also from: https://doi.org/10.1109/CVPR.2019.00895.

Chen, Liang-Chieh; Zhu, Yukun; Papandreou, George; Schroff, Florian, and Adam, Hartwig, 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*. Munich, Germany: Springer-Verlag, pp. 833–851. Available also from: https://doi.org/10.1007/978-3-030-01234-2_49.

Chen, Tao; Cheng, Ming-Ming; Tan, Ping; Shamir, Ariel, and Hu, Shi-Min, 2009. Sketch2Photo: internet image montage. *ACM Transactions Graphics*. Vol. 28, no. 5, pp. 1–10. Available also from: https://doi.org/10.1145/1618452.1618470.

Chen, Wei-Ting; Yifan, Wang; Kuo, Sy-Yen, and Wetzstein, Gordon, 2024. DehazeNeRF: Multi-image Haze Removal and 3D Shape Reconstruction using Neural Radiance Fields. In: *2024 International Conference on 3D Vision (3DV)*, pp. 247–256. Available also from: https://doi.org/10.1109/3DV62453.2024.00039.

Cheng, Bowen; Misra, Ishan; Schwing, Alexander G.; Kirillov, Alexander, and Girdhar, Rohit, 2022. Masked-attention Mask Transformer for Universal Image Segmentation. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1280–1289. Available also from: https://doi.org/10.1109/CVPR52688.2022.00135.

Clark, Aidan; Donahue, Jeff, and Simonyan, Karen, 2019. Adversarial Video Generation on Complex Datasets. *arXiv preprint arXiv:1907.06571*. Available also from: https://doi.org/10.48550/arXiv.1907.06571.

Clement, Lee and Kelly, Jonathan, 2017. How to Train a CAT: Learning Canonical Appearance Transformations for Direct Visual Localization Under Illumination Change. *IEEE Robotics and Automation Letters*. Vol. 3, pp. 2447–2454. Available also from: https://doi.org/10.1109/LRA.2018.2799741.

Cordts, Marius; Omran, Mohamed; Ramos, Sebastian; Rehfeld, Timo; Enzweiler, Markus; Benenson, Rodrigo; Franke, Uwe; Roth, Stefan, and Schiele, Bernt, 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223. Available also from: https://doi.org/10.1109/CVPR.2016.350.

Dai, Qiyu; Ni, Xingyu; Shen, Qianfan; Chen, Wenzheng; Chen, Baoquan, and Chu, Mengyu, 2025. RainyGS: Efficient Rain Synthesis with Physically-Based Gaussian Splatting. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16153–16162. Available also from: https://doi.org/10.48550/arXiv.2503.21442.

Lalonde, Jean-François; Hoiem, Derek; Efros, Alexei A.; Rother, Carsten; Winn, John M., and Criminisi, Antonio, 2007. Photo clip art. *ACM Transactions on Graphics*. Vol. 26, no. 3, p. 3. Available also from: http://doi.acm.org/10.1145/1276377.1276381.

Deng, Jia; Dong, Wei; Socher, Richard; Li, Li-Jia; Li, Kai, and Fei-Fei, Li, 2009. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Available also from: https://doi.org/10.1109/CVPR.2009.5206848.

Department for Transport and Centre for Connected and Autonomous Vehicles, 2024. *Self-driving vehicles set to be on roads by 2026 as Automated Vehicles Act becomes law*

[https://www.gov.uk/government/news/self-driving-vehicles-set-to-be-on-roads-by-2026-as-automated-vehicles-act-becomes-law]. Accessed: April 21, 2025.

DeVries, Terrance; Bautista, Miguel Angel; Srivastava, Nitish; Taylor, Graham W., and Susskind, Joshua M., 2021. Unconstrained Scene Generation with Locally Conditioned Radiance Fields. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14284–14293. Available also from: https://doi.org/10.1109/ICCV48922.2021.01404.

Diaz-Ruiz, Carlos A; Xia, Youya; You, Yurong; Nino, Jose; Chen, Junan; Monica, Josephine; Chen, Xiangyu; Luo, Katie; Wang, Yan; Emond, Marc, et al., 2022. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21383–21392. Available also from: https://doi.org/10.48550/arXiv.2208.01166.

Dosovitskiy, Alexey; Ros, German; Codevilla, Felipe; Lopez, Antonio, and Koltun, Vladlen, 2017. CARLA: An Open Urban Driving Simulator. In: Levine, Sergey; Vanhoucke, Vincent, and Goldberg, Ken (eds.). *Proceedings of the 1st Annual Conference on Robot Learning*. PMLR. Vol. 78, pp. 1–16. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v78/dosovitskiy17a.html.

Friedman, Dan and Dieng, Adji Bousso, 2023. The Vendi Score: A Diversity Evaluation Metric for Machine Learning. *Transactions on Machine Learning Research*. Available also from: https://doi.org/10.48550/arXiv.2210.02410.

Gadd, Matthew; De Martini, Daniele; Bartlett, Oliver; Murcutt, Paul; Towlson, Matt; Widojo, Matthew; Muşat, Valentina; Robinson, Luke; Panagiotaki, Efimia; Pramatarov, Georgi; Alexander Kühn, Marc; Marchegiani, Letizia; Newman, Paul, and Kunze, Lars, 2024. OORD: The Oxford Offroad Radar Dataset. *IEEE Transactions on Intelligent Transportation Systems*. Vol. 25, no. 11, pp. 18779–18790. Available also from: https://doi.org/10.1109/TITS.2024.3424984.

Gao, Ruiyuan; Chen, Kai; Li, Zhihao; Hong, Lanqing; Li, Zhenguo, and Xu, Qiang, 2024a. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*. Available also from: https://doi.org/10.48550/arXiv.2405.14475.

Gao, Ruiyuan; Chen, Kai; Xie, Enze; Hong, Lanqing; Li, Zhenguo; Yeung, Dit-Yan, and Xu, Qiang, 2024b. MagicDrive: Street View Generation with Diverse 3D Geometry Control. In: *The Twelfth International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.2310.02601.

Ge, Junhao; Liu, Zuhong; Fan, Longteng; Jiang, Yifan; Su, Jiaqi; Li, Yiming; Zhang, Zhejun, and Chen, Siheng, 2025. Unraveling the Effects of Synthetic Data on End-to-End Autonomous Driving. *arXiv preprint arXiv:2503.18108*. Available also from: https://doi.org/10.48550/arXiv.2503.18108.

Ge, Songwei; Mahapatra, Aniruddha; Parmar, Gaurav; Zhu, Jun-Yan, and Huang, Jia-Bin, 2024. On the Content Bias in Fréchet Video Distance. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 7277–7288. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.00695.

Geiger, Andreas; Lenz, Philip; Stiller, Christoph, and Urtasun, Raquel, 2013. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*. Vol. 32, no. 11, pp. 1231–1237. Available also from: https://doi.org/10.1177/0278364913491297.

Geyer, Jakob; Kassahun, Yohannes; Mahmudi, Mentar; Ricou, Xavier; Durgesh, Rupesh; Chung, Andrew S.; Hauswald, Lorenz; Pham, Viet Hoang; Mühlegg, Maximilian; Dorn, Sebastian; Fernandez, Tiffany; Jänicke, Martin; Mirashi, Sudesh; Savani, Chiragkumar; Sturm, Martin; Vorobiov, Oleksandr; Oelker, Martin; Garreis, Sebastian, and Schuberth, Peter, 2020. A2D2: Audi Autonomous Driving Dataset. *arXiv preprint arXiv:2004.06320*. Available also from: https://doi.org/10.48550/arXiv.2004.06320.

Geyer, Michal; Bar-Tal, Omer; Bagon, Shai, and Dekel, Tali, 2023. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. *arXiv preprint arXiv:2307.10373*. Available also from: https://doi.org/10.48550/arXiv.2307.10373.

Goel, Harsh; Narasimhan, Sai Shankar; Akcin, Oguzhan, and Chinchali, Sandeep, 2024. Syndiff-ad: Improving semantic segmentation and end-to-end autonomous driving with synthetic data from latent diffusion models. *arXiv preprint arXiv:2411.16776*. Available also from: https://doi.org/10.48550/arXiv.2411.16776.

Goldblum, Micah; Souri, Hossein; Ni, Renkun; Shu, Manli; Prabhu, Viraj; Somepalli, Gowthami; Chattopadhyay, Prithvijit; Ibrahim, Mark; Bardes, Adrien; Hoffman, Judy; Chellappa, Rama; Wilson, Andrew Gordon, and Goldstein, Tom, 2023. Battle of the backbones: a large-scale comparison of pretrained models across computer vision tasks. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc. NIPS '23. Available also from: https://dl.acm.org/doi/10.5555/3666122.3667399.

Gómez, Jose L.; Silva, Manuel; Seoane, Antonio; Borràs, Agnés; Noriega, Mario; Ros, German; Iglesias-Guitian, Jose A., and López, Antonio M., 2025. All for one, and one for all: UrbanSyn Dataset, the third Musketeer of synthetic driving scenes. *Neurocomputing*. Vol. 637, p. 130038. Available also from: https://doi.org/10.1016/j.neucom.2025.130038.

Goodfellow, Ian J.; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron, and Bengio, Yoshua, 2014. Generative adversarial nets. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. Montreal, Canada: MIT Press, pp. 2672–2680. NIPS'14. Available also from: https://dl.acm.org/doi/10.5555/2969033.2969125.

Hahner, Martin; Dai, Dengxin; Sakaridis, Christos; Zaech, Jan-Nico, and Gool, Luc Van, 2019. Semantic Understanding of Foggy Scenes with Purely Synthetic Data. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3675–3681. Available also from: https://doi.org/10.1109/ITSC.2019.8917518.

Hahner, Martin; Sakaridis, Christos; Dai, Dengxin, and Van Gool, Luc, 2021. Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15263–15272. Available also from: https://doi.org/10.1109/ICCV48922.2021.01500.

Henzler, Philipp; Mitra, Niloy, and Ritschel, Tobias, 2019. Escaping Plato's Cave: 3D Shape From Adversarial Rendering. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 9983–9992. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.01008.

Hess, Georg; Lindstrom, Carl; Fatemi, Maryam; Petersson, Christoffer, and Svensson, Lennart, 2025. SplatAD: Real-Time Lidar and Camera Rendering with 3D Gaussian Splatting for Autonomous Driving. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA:

IEEE Computer Society, pp. 11982–11992. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.01119.

Heusel, Martin; Ramsauer, Hubert; Unterthiner, Thomas; Nessler, Bernhard, and Hochreiter, Sepp, 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA: Curran Associates Inc., pp. 6629–6640. NIPS'17. Available also from: https://dl.acm.org/doi/10.5555/3295222.3295408.

HM Government, 2022. *Connected & Automated Mobility 2025: Realising the Benefits of Self-Driving Vehicles in the UK* [https://www.gov.uk/government/publications/connected-and-automated-mobility-2025-realising-the-benefits-of-self-driving-vehicles]. Accessed: April 21, 2025.

Houston, John; Zuidhof, Guido; Bergamini, Luca; Ye, Yawei; Chen, Long; Jain, Ashesh; Omari, Sammy; Iglovikov, Vladimir, and Ondruska, Peter, 2021. One Thousand and One Hours: Self-driving Motion Prediction Dataset. In: *Proceedings of the 2020 Conference on Robot Learning*. PMLR. Vol. 155, pp. 409–418. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v155/houston21a.html.

Hu, Edward J; yelong shen; Wallis, Phillip; Allen-Zhu, Zeyuan; Li, Yuanzhi; Wang, Shean; Wang, Lu, and Chen, Weizhu, 2022. LoRA: Low-Rank Adaptation of Large Language Models. In: *International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.2106.09685.

Hu, Ming-Kuei, 1962. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*. Vol. 8, no. 2, pp. 179–187. Available also from: https://doi.org/10.1109/TIT.1962.1057692.

Hu, Mu; Wang, Shuling; Li, Bin; Ning, Shiyu; Fan, Li, and Gong, Xiaojin, 2021. Penet: Towards precise and efficient image guided depth completion. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 13656–13662. Available also from: https://doi.org/10.1109/ICRA48506.2021.9561035.

Huang, Xinyu; Cheng, Xinjing; Geng, Qichuan; Cao, Binbin; Zhou, Dingfu; Wang, Peng; Lin, Yuanqing, and Yang, Ruigang, 2018. The ApolloScape Dataset for Autonomous Driving. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1067–10676. Available also from: https://doi.org/10.1109/CVPRW.2018.00141.

Huang, Xun; Liu, Ming-Yu; Belongie, Serge, and Kautz, Jan, 2018. Multimodal Unsupervised Image-to-Image Translation. In: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*. Munich, Germany: Springer-Verlag, pp. 179–196. Available also from: https://doi.org/10.1007/978-3-030-01219-9_11.

Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui, and Efros, Alexei A., 2017. Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976. Available also from: https://doi.org/10.1109/CVPR.2017.632.

Jayasumana, Sadeep; Ramalingam, Srikumar; Veit, Andreas; Glasner, Daniel; Chakrabarti, Ayan, and Kumar, Sanjiv, 2024. Rethinking FID: Towards a Better Evaluation Metric for Image Generation. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA:

IEEE Computer Society, pp. 9307–9315. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.00889.

Jeon, Hyeonjae; Seo, Junghyun; Kim, Taesoo; Son, Sungho; Lee, Jungki; Choi, Gyeungho, and Lim, Yongseob, 2025. RainSD: Rain style diversification module for image synthesis enhancement using feature-level style distribution. *Robotics and Autonomous Systems*. Vol. 186, p. 104922. Available also from: https://doi.org/10.1016/j.robot.2025.104922.

Jin, Jiongchao; Fatemi, Arezou; Pinto Lira, Wallace Michel; Yu, Fenggen; Leng, Biao; Ma, Rui; Mahdavi-Amiri, Ali, and Zhang, Hao, 2021. RaidaR: A Rich Annotated Image Dataset of Rainy Street Scenes. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 2951–2961. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCVW54120.2021.00330.

Johnson, Justin; Alahi, Alexandre, and Fei-Fei, Li, 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, pp. 694–711. Available also from: https://doi.org/10.1007/978-3-319-46475-6_43.

Johnson, Micah K.; Dale, Kevin; Avidan, Shai; Pfister, Hanspeter; Freeman, William T., and Matusik, Wojciech, 2011. CG2Real: Improving the Realism of Computer Generated Images Using a Large Collection of Photographs. *IEEE Transactions on Visualization and Computer Graphics*. Vol. 17, no. 9, pp. 1273–1285. Available also from: https://doi.org/10.1109/TVCG.2010.233.

Karras, Tero; Aila, Timo; Laine, Samuli, and Lehtinen, Jaakko, 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In: *International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.1710.10196.

Karras, Tero; Laine, Samuli, and Aila, Timo, 2019. A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410. Available also from: https://doi.org/10.48550/arXiv.1812.04948.

Karras, Tero; Laine, Samuli; Aittala, Miika; Hellsten, Janne; Lehtinen, Jaakko, and Aila, Timo, 2020. Analyzing and improving the image quality of stylegan. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119. Available also from: https://doi.org/10.48550/arXiv.1912.04958.

Kay, Will; Carreira, Joao; Simonyan, Karen; Zhang, Brian; Hillier, Chloe; Vijayanarasimhan, Sudheendra; Viola, Fabio; Green, Tim; Back, Trevor; Natsev, Paul; Suleyman, Mustafa, and Zisserman, Andrew, 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*. Available also from: https://doi.org/10.48550/arXiv.1705.06950.

Kenk, Mourad A. and Hassaballah, M., 2020. DAWN: Vehicle Detection in Adverse Weather Nature Dataset. *arXiv preprint arXiv:2008.05402*. Available also from: https://doi.org/10.48550/arXiv.2008.05402.

Kerbl, Bernhard; Kopanas, Georgios; Leimkühler, Thomas, and Drettakis, George, 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*. Vol. 42, no. 4. Available also from: https://doi.org/10.1145/3592433.

Kim, Seung Wook; Brown, Bradley; Yin, Kangxue; Kreis, Karsten; Schwarz, Katja; Li, Daiqing; Rombach, Robin; Torralba, Antonio, and Fidler, Sanja, 2023. NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR), pp. 8496–8506. Available also from: https://doi.org/10.1109/CVPR52729.2023.00821.

Klinghoffer, Tzofi; Philion, Jonah; Chen, Wenzheng; Litany, Or; Gojcic, Zan; Joo, Jungseock; Raskar, Ramesh; Fidler, Sanja, and Alvarez, Jose M., 2023. Towards Viewpoint Robustness in Bird's Eye View Segmentation. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 8481–8490. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00782.

Kondratyuk, Dan; Yu, Lijun; Gu, Xiuye; Lezama, Jose; Huang, Jonathan; Schindler, Grant; Hornung, Rachel; Birodkar, Vighnesh; Yan, Jimmy; Chiu, Ming-Chang; Somandepalli, Krishna; Akbari, Hassan; Alon, Yair; Cheng, Yong; Dillon, Joshua V.; Gupta, Agrim; Hahn, Meera; Hauth, Anja; Hendon, David; Martinez, Alonso; Minnen, David; Sirotenko, Mikhail; Sohn, Kihyuk; Yang, Xuan; Adam, Hartwig; Yang, Ming-Hsuan; Essa, Irfan; Wang, Huisheng; Ross, David A; Seybold, Bryan, and Jiang, Lu, 2024. VideoPoet: A Large Language Model for Zero-Shot Video Generation. In: *Proceedings of the 41st International Conference on Machine Learning*. PMLR. Vol. 235, pp. 25105–25124. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v235/kondratyuk24a.html.

Kong, Lingdong; Liu, Youquan; Li, Xin; Chen, Runnan; Zhang, Wenwei; Ren, Jiawei; Pan, Liang; Chen, Kai, and Liu, Ziwei, 2023. Robo3D: Towards Robust and Reliable 3D Perception against Corruptions. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 19937–19949. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01830.

Kurup, Akhil M. and Bos, Jeremy P., 2023. Winter adverse driving dataset for autonomy in inclement winter weather. *Optical Engineering*. Vol. 62, no. 3, p. 031207. Available also from: https://doi.org/10.1117/1.OE.62.3.031207.

Laine, Samuli and Karras, Tero, 2010. Efficient sparse voxel octrees. In: *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. Washington, D.C.: Association for Computing Machinery, pp. 55–63. I3D '10. Available also from: https://doi.org/10.1145/1730804.1730814.

Lee, Jinho; Shiotsuka, Daiki; Nishimori, Toshiaki; Nakao, Kenta, and Kamijo, Shunsuke, 2022. GAN-Based LiDAR Translation between Sunny and Adverse Weather for Autonomous Driving and Driving Simulation. *Sensors*. Vol. 22, no. 14. Available also from: https://www.mdpi.com/1424-8220/22/14/5287.

Levy, Deborah; Peleg, Amit; Pearl, Naama; Rosenbaum, Dan; Akkaynak, Derya; Korman, Simon, and Treibitz, Tali, 2023. SeaThru-NeRF: Neural Radiance Fields in Scattering Media. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 56–65. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00014.

Li, Y.; Jiang, L.; Xu, L.; Xiangli, Y.; Wang, Z.; Lin, D., and Dai, B., 2023. MatrixCity: A Large-scale City Dataset for City-scale Neural Rendering and Beyond. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 3182–3192. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00297.

Li, Yijun; Liu, Ming-Yu; Li, Xueting; Yang, Ming-Hsuan, and Kautz, Jan, 2018. A Closed-Form Solution to Photorealistic Image Stylization. In: *Computer Vision – ECCV 2018: 15th European Conference, Munich,*

Germany, September 8–14, 2018, Proceedings, Part III. Munich, Germany: Springer-Verlag, pp. 468–483. Available also from: https://doi.org/10.1007/978-3-030-01219-9_28.

Li, Yuan; Lin, Zhi-Hao; Forsyth, David; Huang, Jia-Bin, and Wang, Shenlong, 2023. ClimateNeRF: Extreme Weather Synthesis in Neural Radiance Field. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 3204–3215. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00299.

Li, Yunhao; Wu, Jing; Zhao, Lingzhe, and Liu, Peidong, 2024. DerainNeRF: 3D Scene Estimation with Adhesive Waterdrop Removal. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2787–2793. Available also from: https://doi.org/10.1109/ICRA57147.2024.10609981.

Li, Zhiqi; Wang, Wenhai; Li, Hongyang; Xie, Enze; Sima, Chonghao; Lu, Tong; Yu, Qiao, and Dai, Jifeng, 2025. BEVFormer: Learning Bird's-Eye-View Representation From LiDAR-Camera via Spatiotemporal Transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 47, no. 3, pp. 2020–2036. Available also from: https://doi.org/10.1109/TPAMI.2024.3515454.

Liao, Yiyi; Xie, Jun, and Geiger, Andreas, 2023. KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis & Machine Intelligence*. Vol. 45, no. 03, pp. 3292–3310. Available also from: https://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3179507.

Ligocki, Adam; Jelinek, Ales, and Zalud, Ludek, 2020. Brno Urban Dataset-The New Data for Self-Driving Agents and Mapping Tasks. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3284–3290. Available also from: https://doi.org/10.1109/ICRA40945.2020.9197277.

Lin, Han; Cho, Jaemin; Zala, Abhay, and Bansal, Mohit, 2025. Ctrl-Adapter: An Efficient and Versatile Framework for Adapting Diverse Controls to Any Diffusion Model. In: *The Thirteenth International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.2404.09967.

Lin, Tsung-Yi; Goyal, Priya; Girshick, Ross; He, Kaiming, and Dollár, Piotr, 2017. Focal Loss for Dense Object Detection. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007. Available also from: https://doi.org/10.1109/ICCV.2017.324.

Liu, Jiahe; Qu, Youran; Yan, Qi; Zeng, Xiaohui; Wang, Lele, and Liao, Renjie, 2024. Fréchet Video Motion Distance: A Metric for Evaluating Motion Consistency in Videos. In: *First Workshop on Controllable Video Generation @ICML24*. Available also from: https://doi.org/10.48550/arXiv.2407.16124.

Liu, Lina; Song, Xibin; Wang, Mengmeng; Liu, Yong, and Zhang, Liangjun, 2021. Self-supervised Monocular Depth Estimation for All Day Images using Domain Separation. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12717–12726. Available also from: https://doi.org/10.48550/arXiv.2108.07628.

Liu, Wei; Anguelov, Dragomir; Erhan, Dumitru; Szegedy, Christian; Reed, Scott; Fu, Cheng-Yang, and Berg, Alexander C, 2016. Ssd: Single shot multibox detector. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, pp. 21–37. Available also from: https://doi.org/10.1007/978-3-319-46448-0_2.

Liu, Xinhang; Tai, Yu-Wing, and Tang, Chi-Keung, 2023. Clean-NeRF: Reformulating NeRF to account for View-Dependent Observations. *arXiv preprint arXiv:2303.14707*. Available also from: https://doi.org/10.48550/arXiv.2303.14707.

Liu, Yun-Fu; Jaw, Da-Wei; Huang, Shih-Chia, and Hwang, Jenq-Neng, 2018. DesnowNet: Context-Aware Deep Network for Snow Removal. *IEEE Transactions on Image Processing*. Vol. 27, no. 6, pp. 3064–3073. Available also from: https://doi.org/10.1109/TIP.2018.2806202.

Liu, Zhijian; Tang, Haotian; Amini, Alexander; Yang, Xinyu; Mao, Huizi; Rus, Daniela L, and Han, Song, 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 2774–2781. Available also from: https://doi.org/10.1109/ICRA48891.2023.10160968.

Luo, Ge Ya; Favero, Gian Mario; Luo, ZhiHao; Jolicoeur-Martineau, Alexia, and Pal, Christopher, 2025. Beyond FVD: An Enhanced Evaluation Metrics for Video Generation Distribution Quality. In: *The Thirteenth International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.2410.05203.

Mallya, Arun; Wang, Ting-Chun; Sapra, Karan, and Liu, Ming-Yu, 2020. World-Consistent Video-to-Video Synthesis. In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII*. Glasgow, United Kingdom: Springer-Verlag, pp. 359–378. Available also from: https://doi.org/10.1007/978-3-030-58598-3_22.

Martin-Brualla, Ricardo; Radwan, Noha; Sajjadi, Mehdi S. M.; Barron, Jonathan T.; Dosovitskiy, Alexey, and Duckworth, Daniel, 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7206–7215. Available also from: https://doi.org/10.1109/CVPR46437.2021.00713.

Mehr, Goodarz and Eskandarian, Azim, 2025. SimBEV: A Synthetic Multi-Task Multi-Sensor Driving Data Generation Tool and Dataset. *arXiv preprint arXiv:2502.01894*. Available also from: https://doi.org/10.48550/arXiv.2502.01894.

Midjourney, 2024. *Midjourney: AI-Powered Image Generation Platform* [https://www.midjourney.com]. Accessed: April 21, 2025.

Mildenhall, Ben; Srinivasan, Pratul P.; Tancik, Matthew; Barron, Jonathan T.; Ramamoorthi, Ravi, and Ng, Ren, 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: *Computer Vision – ECCV 2020*. Springer International Publishing, pp. 405–421. Available also from: https://doi.org/10.1007/978-3-030-58452-8_24.

Mou, Chong; Wang, Xintao; Xie, Liangbin; Wu, Yanze; Zhang, Jian; Qi, Zhongang, and Shan, Ying, 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38, pp. 4296–4304. No. 5. Available also from: https://doi.org/10.1609/aaai.v38i5.28226.

Müller, Thomas; Evans, Alex; Schied, Christoph, and Keller, Alexander, 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*. Vol. 41, no. 4, 102:1–102:15. Available also from: https://doi.org/10.1145/3528223.3530127.

Musat, Valentina; De Martini, Daniele; Gadd, Matthew, and Newman, Paul, 2022. Depth-SIMS: Semi-Parametric Image and Depth Synthesis. In: *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2388–2394. Available also from: https://doi.org/10.1109/ICRA46639.2022.9811569.

Musat, Valentina; Fursa, Ivan; Newman, Paul; Cuzzolin, Fabio, and Bradley, Andrew, 2021. Multi-Weather City: Adverse Weather Stacking for Autonomous Driving. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2906–2915. Available also from: https://doi.org/10.1109/ICCVW54120.2021.00325.

Muşat, Valentina; De Martini, Daniele; Gadd, Matthew, and Newman, Paul, 2024a. NeuralFloors: Conditional Street-Level Scene Generation From BEV Semantic Maps via Neural Fields. *IEEE Robotics and Automation Letters*. Vol. 9, no. 3, pp. 2431–2438. Available also from: https://doi.org/10.1109/LRA.2024.3356793.

Muşat, Valentina; De Martini, Daniele; Gadd, Matthew, and Newman, Paul, 2024b. NeuralFloors++: Consistent Street-Level Scene Generation From BEV Semantic Maps. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12872–12879. Available also from: https://doi.org/10.1109/IROS58592.2024.10802002.

Neuhold, Gerhard; Ollmann, Tobias; Bulò, Samuel Rota, and Kontschieder, Peter, 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5000–5009. Available also from: https://doi.org/10.1109/ICCV.2017.534.

Neumann, Lukáš; Karg, Michelle; Zhang, Shanshan; Scharfenberger, Christian; Piegert, Eric; Mistr, Sarah; Prokofyeva, Olga; Thiel, Robert; Vedaldi, Andrea; Zisserman, Andrew, and Schiele, Bernt, 2018. NightOwls: A pedestrians at night dataset. In: *Asian Conference on Computer Vision*. Springer, pp. 691–705. Available also from: https://doi.org/10.1007/978-3-030-20887-5_43.

Nguyen-Phuoc, Thu; Li, Chuan; Theis, Lucas; Richardt, Christian, and Yang, Yongliang, 2019. HoloGAN: Unsupervised Learning of 3D Representations From Natural Images. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7587–7596. Available also from: https://doi.org/10.1109/ICCV.2019.00768.

Nunes, Lucas; Marcuzzi, Rodrigo; Behley, Jens, and Stachniss, Cyrill, 2025. Towards Generating Realistic 3D Semantic Training Data for Autonomous Driving. *arXiv preprint arXiv:2503.21449*. Available also from: https://doi.org/10.48550/arXiv.2503.21449.

Odena, Augustus; Olah, Christopher, and Shlens, Jonathon, 2017. Conditional image synthesis with auxiliary classifier GANs. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. Sydney, NSW, Australia: JMLR.org, pp. 2642–2651. ICML'17. Available also from: https://dl.acm.org/doi/10.5555/3305890.3305954.

OpenAI, 2024. *ChatGPT: AI Language Model by OpenAI* [https://chatgpt.com]. Accessed: April 21, 2025.

OpenStreetMap contributors, 2004. *OpenStreetMap* [https://www.openstreetmap.org]. Accessed: September 4, 2025.

Özeren, Enes and Bhowmick, Arka, 2025. Evaluating the Impact of Synthetic Data on Object Detection Tasks in Autonomous Driving. *arXiv preprint arXiv:2503.09803*. Available also from: https://doi.org/10.48550/arXiv.2503.09803.

Park, Taesung; Liu, Ming-Yu; Wang, Ting-Chun, and Zhu, Jun-Yan, 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2332–2341. Available also from: https://doi.org/10.1109/CVPR.2019.00244.

Parmar, Gaurav; Park, Taesung; Narasimhan, Srinivasa, and Zhu, Jun-Yan, 2024. One-Step Image Translation with Text-to-Image Models. *arXiv preprint arXiv:2403.12036*. Available also from: https://doi.org/10.48550/arXiv.2403.12036.

Pham, Quang-Hieu; Sevestre, Pierre; Pahwa, Ramanpreet Singh; Zhan, Huijing; Pang, Chun Ho; Chen, Yuda; Mustafa, Armin; Chandrasekhar, Vijay, and Lin, Jie, 2020. A*3D Dataset: Towards Autonomous Driving in Challenging Environments. In: *Proc. of The International Conference in Robotics and Automation (ICRA)*. Available also from: https://doi.org/10.1109/ICRA40945.2020.9197385.

Philion, Jonah and Fidler, Sanja, 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, pp. 194–210. Available also from: https://doi.org/10.1007/978-3-030-58568-6_12.

Piroli, Aldi; Dallabetta, Vinzenz; Kopp, Johannes; Walessa, Marc; Meissner, Daniel, and Dietmayer, Klaus, 2023. Energy-Based Detection of Adverse Weather Effects in LiDAR Data. *IEEE Robotics and Automation Letters*. Vol. 8, no. 7, pp. 4322–4329. Available also from: https://doi.org/10.1109/LRA.2023.3282382.

Pitropov, Matthew; Garcia, Danson Evan; Rebello, Jason; Smart, Michael; Wang, Carlos; Czarnecki, Krzysztof, and Waslander, Steven, 2021. Canadian Adverse Driving Conditions dataset. *The International Journal of Robotics Research*. Vol. 40, no. 4–5, pp. 681–690. Available also from: https://doi.org/10.1177/0278364920979368.

Porav, Horia; Bruls, Tom, and Newman, Paul, 2019. Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 33–40. Available also from: https://doi.org/10.1109/ITSC.2019.8917073.

Porav, Horia; Maddern, William P., and Newman, Paul, 2018. Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1011–1018. Available also from: https://doi.org/10.1109/ICRA.2018.8462894.

Porav, Horia; Muşat, Valentina; Bruls, Tom, and Newman, Paul, 2020. Rainy screens: Collecting rainy datasets, indoors. *arXiv preprint arXiv:2003.04742*. Available also from: https://doi.org/10.48550/arXiv.2003.04742.

Qi, Xiaojuan; Chen, Qifeng; Jia, Jiaya, and Koltun, V., 2018. Semi-Parametric Image Synthesis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8808–8816. Available also from: https://doi.org/10.1109/CVPR.2018.00918.

Qian, Rui; Tan, Robby T.; Yang, Wenhan; Su, Jiajun, and Liu, Jiaying, 2018. Attentive Generative Adversarial Network for Raindrop Removal from A Single Image. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2482–2491. Available also from: https://doi.org/10.1109/CVPR.2018.00263.

Radford, Alec; Kim, Jong Wook; Hallacy, Chris; Ramesh, Aditya; Goh, Gabriel; Agarwal, Sandhini; Sastry, Girish; Askell, Amanda; Mishkin, Pamela; Clark, Jack; Krueger, Gretchen, and Sutskever, Ilya, 2021. Learning Transferable Visual Models From Natural Language Supervision. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR. Vol. 139, pp. 8748–8763. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v139/radford21a.html.

Ramazzina, Andrea; Bijelic, Mario; Walz, Stefanie; Sanvito, Alessandro; Scheuble, Dominik, and Heide, Felix, 2023. ScatterNeRF: Seeing Through Fog with Physically-Based Inverse Neural Rendering. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 17911–17922. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.01646.

Ramesh, Aditya; Pavlov, Mikhail; Goh, Gabriel; Gray, Scott; Voss, Chelsea; Radford, Alec; Chen, Mark, and Sutskever, Ilya, 2021. Zero-Shot Text-to-Image Generation. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR. Vol. 139, pp. 8821–8831. Proceedings of Machine Learning Research. Available also from: https://proceedings.mlr.press/v139/ramesh21a.html.

Reiser, Christian; Peng, Songyou; Liao, Yiyi, and Geiger, Andreas, 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 14315–14325. Available also from: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.01407.

Ren, Shaoqing; He, Kaiming; Girshick, Ross, and Sun, Jian, 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Vol. 28. Available also from: https://doi.org/10.48550/arXiv.1506.01497.

Richter, Stephan R; AlHaija, Hassan Abu, and Koltun, Vladlen, 2022. Enhancing photorealism enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 45, no. 2, pp. 1700–1715. Available also from: https://doi.org/10.1109/TPAMI.2022.3166687.

Richter, Stephan R.; Vineet, Vibhav; Roth, Stefan, and Koltun, Vladlen, 2016. Playing for Data: Ground Truth from Computer Games. In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, pp. 102–118. Available also from: https://doi.org/10.1007/978-3-319-46475-6_7.

RockstarNorth, 2015. *Grand Theft Auto V Video game* [https://www.rockstargames.com/gta-v]. Accessed: September 4, 2025.

Rombach, Robin; Blattmann, A.; Lorenz, Dominik; Esser, Patrick, and Ommer, Björn, 2021. High-Resolution Image Synthesis with Latent Diffusion Models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685. Available also from: https://doi.org/10.1109/CVPR52688.2022.01042.

Ros, German; Sellart, Laura; Materzynska, Joanna; Vazquez, David, and Lopez, Antonio M., 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3234–3243. Available also from: https://doi.org/10.1109/CVPR.2016.352.

Rothmeier, Thomas and Huber, Werner, 2021. Let it Snow: On the Synthesis of Adverse Weather Image Data. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3300–3306. Available also from: https://doi.org/10.1109/ITSC48978.2021.9565008.

Rothmeier, Thomas; Huber, Werner, and Knoll, Alois C., 2024. Time to Shine: Fine-Tuning Object Detection Models with Synthetic Adverse Weather Images. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4435–4444. Available also from: https://doi.org/10.1109/WACV57701.2024.00439.

Ruiz, Nataniel; Li, Yuanzhen; Jampani, Varun; Pritch, Yael; Rubinstein, Michael, and Aberman, Kfir, 2023. 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. Los Alamitos, CA, USA: IEEE Computer Society, pp. 22500–22510. Available also from: https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.02155.

Sakaridis, Christos; Dai, Dengxin, and Van Gool, Luc, 2018. Semantic Foggy Scene Understanding with Synthetic Data. *International Journal of Computer Vision*. Vol. 126, no. 9, pp. 973–992. Available also from: https://doi.org/10.1007/s11263-018-1072-8.

Sakaridis, Christos; Dai, Dengxin, and Van Gool, Luc, 2019. Guided Curriculum Model Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7373–7382. Available also from: https://doi.org/10.1109/ICCV.2019.00747.

Sakaridis, Christos; Dai, Dengxin, and Van Gool, Luc, 2021. ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10745–10755. Available also from: https://doi.org/10.1109/ICCV48922.2021.01059.

Salimans, Tim; Goodfellow, Ian; Zaremba, Wojciech; Cheung, Vicki; Radford, Alec, and Chen, Xi, 2016. Improved techniques for training GANs. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain: Curran Associates Inc., pp. 2234–2242. NIPS'16. Available also from: https://dl.acm.org/doi/10.5555/3157096.3157346.

Schön, Markus; Ruof, Jona; Wodtko, Thomas; Buchholz, Michael, and Dietmayer, Klaus, 2024. The ADUULM-360 Dataset-A Multi-Modal Dataset for Depth Estimation in Adverse Weather. In: *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1403–1409. Available also from: https://doi.org/10.1109/ITSC58415.2024.10920201.

Schwarz, Katja; Liao, Yiyi; Niemeyer, Michael, and Geiger, Andreas, 2020. GRAF: generative radiance fields for 3D-aware image synthesis. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Vancouver, BC, Canada: Curran Associates Inc. NIPS '20. Available also from: https://dl.acm.org/doi/abs/10.5555/3495724.3497416.

Sethuraman, Advaith Venkatramanan; Ramanagopal, Manikandasriram Srinivasan, and Skinner, Katherine A., 2023. WaterNeRF: Neural Radiance Fields for Underwater Scenes. In: *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, pp. 1–7. Available also from: https://doi.org/10.23919/OCEANS52994.2023.10336972.

Shah, Shital; Dey, Debadeepta; Lovett, Chris, and Kapoor, Ashish, 2018. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In: *Field and Service Robotics*. Cham: Springer International Publishing, pp. 621–635. Available also from: https://doi.org/10.1007/978-3-319-67361-5_40.

Shaik, Furqan Ahmed; Reddy Malreddy, Abhishek; Billa, Nikhil Reddy; Chaudhary, Kunal; Manchanda, Sunny, and Varma, Girish, 2024. IDD-AW: A Benchmark for Safe and Robust Segmentation of Drive Scenes in Unstructured Traffic and Adverse Weather. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4602–4611. Available also from: https://doi.org/10.1109/WACV57701.2024.00455.

El-Shair, Zaid A; Abu-raddaha, Abdalmalek; Cofield, Aaron; Alawneh, Hisham; Aladem, Mohamed; Hamzeh, Yazan, and Rawashdeh, Samir A, 2024. SID: Stereo Image Dataset for Autonomous Driving in Adverse Conditions. In: *NAECON 2024-IEEE National Aerospace and Electronics Conference*. IEEE,

pp. 403–408. Available also from: https://doi.org/10.1109/NAECON61878.2024.10670659.

Sheeny, Marcel; De Pellegrin, Emanuele; Mukherjee, Saptarshi; Ahrabian, Alireza; Wang, Sen, and Wallace, Andrew, 2021. RADIATE: A Radar Dataset for Automotive Perception in Bad Weather. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE Press, pp. 1–7. Available also from: https://doi.org/10.1109/ICRA48506.2021.9562089.

Silva, Manuel; Seoane, Antonio; Alvarez Mures, Luis; López, Antonio, and Iglesias Guitián, José, 2025. Exploring the effects of synthetic data generation: a case study on autonomous driving for semantic segmentation. *The Visual Computer*. Vol. 41, pp. 7379–7397. Available also from: https://doi.org/10.1007/s00371-025-03811-1.

Simonyan, Karen and Zisserman, Andrew, 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations*, pp. 1–14. Available also from: https://doi.org/10.48550/arXiv.1409.1556.

Sun, Pei; Kretzschmar, Henrik; Dotiwalla, Xerxes; Chouard, Aurélien; Patnaik, Vijaysai; Tsui, Paul; Guo, James; Zhou, Yin; Chai, Yuning; Caine, Benjamin; Vasudevan, Vijay; Han, Wei; Ngiam, Jiquan; Zhao, Hang; Timofeev, Aleksei; Ettinger, Scott; Krivokon, Maxim; Gao, Amy; Joshi, Aditya; Zhang, Yu; Shlens, Jonathon; Chen, Zhifeng, and Anguelov, Dragomir, 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. Available also from: https://doi.org/10.1109/CVPR42600.2020.00252.

Sun, Tao; Segu, Mattia; Postels, Janis; Wang, Yuxuan; Van Gool, Luc; Schiele, Bernt; Tombari, Federico, and Yu, Fisher, 2022. SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21339–21350. Available also from: https://doi.org/10.1109/CVPR52688.2022.02068.

Sushko, Vadim; Schönfeld, Edgar; Zhang, Dan; Gall, Juergen; Schiele, Bernt, and Khoreva, Anna, 2021. You Only Need Adversarial Supervision for Semantic Image Synthesis. In: *International Conference on Learning Representations*. Available also from: https://doi.org/10.48550/arXiv.2012.04781.

Swerdlow, Alexander; Xu, Runsheng, and Zhou, Bolei, 2024. Street-View Image Generation From a Bird's-Eye View Layout. *IEEE Robotics and Automation Letters*. Vol. 9, no. 4, pp. 3578–3585. Available also from: https://doi.org/10.1109/LRA.2024.3368234.

Szegedy, Christian; Vanhoucke, Vincent; Ioffe, Sergey; Shlens, Jon, and Wojna, Zbigniew, 2016. Rethinking the Inception Architecture for Computer Vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. Available also from: https://doi.org/10.1109/CVPR.2016.308.

Tan, Xin; Xu, Ke; Cao, Ying; Zhang, Yiheng; Ma, Lizhuang, and Lau, Rynson W. H., 2021. Night-Time Scene Parsing With a Large Real Dataset. *IEEE Transactions on Image Processing*. Vol. 30, pp. 9085–9098. Available also from: https://doi.org/10.1109/TIP.2021.3122004.

Teeti, Izzeddin; Muşat, Valentina; Khan, Salman Hameed; Rast, Alexander; Cuzzolin, Fabio, and Bradley, Andrew, 2022. Vision in adverse weather: Augmentation using CycleGANs with various object detectors for robust perception in autonomous racing. In: *1st ICML Workshop on Safe Learning for Autonomous Driving (SL4AD)*. Baltimore, MD, USA. Available also from: https://doi.org/10.48550/arXiv.2201.03246.

Testolina, Paolo; Barbato, Francesco; Michieli, Umberto; Giordani, Marco; Zanuttigh, Pietro, and Zorzi, Michele, 2023. SELMA: SEmantic Large-Scale Multimodal Acquisitions in Variable Weather, Daytime and Viewpoints. *Transactions on Intelligent Transportation Systems*. Vol. 24, no. 7, pp. 7012–7024. Available also from: https://doi.org/10.1109/TITS.2023.3257086.

Teufel, Sven; Volk, Georg; Von Bernuth, Alexander, and Bringmann, Oliver, 2022. Simulating Realistic Rain, Snow, and Fog Variations For Comprehensive Performance Characterization of LiDAR Perception. In: *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–7. Available also from: https://doi.org/10.1109/VTC2022-Spring54318.2022.9860868.

Tian, Zhi; Shen, Chunhua; Chen, Hao, and He, Tong, 2019. FCOS: Fully Convolutional One-Stage Object Detection. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9626–9635. Available also from: https://doi.org/10.1109/ICCV.2019.00972.

Tonderski, Adam; Lindström, Carl; Hess, Georg; Ljungbergh, William; Svensson, Lennart, and Petersson, Christoffer, 2024. NeuRAD: Neural Rendering for Autonomous Driving. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904. Available also from: https://doi.org/10.1109/CVPR52733.2024.01411.

Tremblay, Maxime; Halder, Shirsendu Sukanta; de Charette, Raoul, and Lalonde, Jean-François, 2020. Rain Rendering for Evaluating and Improving Robustness to Bad Weather. *International Journal of Computer Vision*. Vol. 129, pp. 341–360. Available also from: https://doi.org/10.1007/s11263-020-01366-3.

Tulyakov, Sergey; Liu, Ming-Yu; Yang, Xiaodong, and Kautz, Jan, 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1526–1535. Available also from: https://doi.org/10.1109/CVPR.2018.00165.

Unterthiner, Thomas; van Steenkiste, Sjoerd; Kurach, Karol; Marinier, Raphaël; Michalski, Marcin, and Gelly, Sylvain, 2019. FVD: A new Metric for Video Generation. In: *DGS@ICLR*. Available also from: https://doi.org/10.48550/arXiv.1812.01717.

Uricar, Michal; Ulicny, Jan; Sistu, Ganesh; Rashed, Hazem; Krizek, Pavel; Hurych, David; Vobecky, Antonín, and Yogamani, Senthil, 2019. Desoiling Dataset: Restoring Soiled Areas on Automotive Fisheye Cameras. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4273–4279. Available also from: https://doi.org/10.1109/ICCVW.2019.00526.

Wang, Hong; Yue, Zongsheng; Xie, Qi; Zhao, Qian; Zheng, Yefeng, and Meng, Deyu, 2021. From Rain Generation to Rain Removal. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14786–14796. Available also from: https://doi.org/10.1109/CVPR46437.2021.01455.

Wang, Ting-Chun; Liu, Ming-Yu; Zhu, Jun-Yan; Liu, Guilin; Tao, Andrew; Kautz, Jan, and Catanzaro, Bryan, 2018. Video-to-Video Synthesis. In: *Conference on Neural Information Processing Systems (NeurIPS)*. Available also from: https://doi.org/10.48550/arXiv.1808.06601.

Wang, Ting-Chun; Liu, Ming-Yu; Zhu, Jun-Yan; Tao, Andrew; Kautz, Jan, and Catanzaro, Bryan, 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807. Available also from: https://doi.org/10.1109/CVPR.2018.00917.

Wang, Xiang; Yuan, Hangjie; Zhang, Shiwei; Chen, Dayou; Wang, Jiuniu; Zhang, Yingya; Shen, Yujun; Zhao, Deli, and Zhou, Jingren, 2023. VideoComposer: compositional video synthesis with motion

controllability. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc. NIPS '23. Available also from: https://dl.acm.org/doi/abs/10.5555/3666122.3666456.

Wang, Zhou; Bovik, Alan C.; Sheikh, Hamid R., and Simoncelli, Eero P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*. Vol. 13, no. 4, pp. 600–612. Available also from: https://doi.org/10.1109/TIP.2003.819861.

Wrenninge, Magnus and Unger, Jonas, 2018. Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing. *arXiv preprint arXiv:1810.08705*. Available also from: https://doi.org/10.48550/arXiv.1810.08705.

Xie, Enze; Wang, Wenhai; Yu, Zhiding; Anandkumar, Anima; Alvarez, Jose M., and Luo, Ping, 2021. SegFormer: simple and efficient design for semantic segmentation with transformers. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc. NIPS '21. Available also from: https://dl.acm.org/doi/10.5555/3540261.3541185.

Xie, Yiming; Wei, Henglu; Liu, Zhenyi; Wang, Xiaoyu, and Ji, Xiangyang, 2024. SynFog: A Photorealistic Synthetic Fog Dataset Based on End-to-End Imaging Simulation for Advancing Real-World Defogging in Autonomous Driving. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21763–21772. Available also from: https://doi.org/10.1109/CVPR52733.2024.02056.

Xie, Yuezhen; Zhang, Meiying, and Hao, Qi, 2025. ClimateGS: Real-Time Climate Simulation with 3D Gaussian Style Transfer. *arXiv preprint arXiv:2503.14845*. Available also from: https://doi.org/10.48550/arXiv.2503.14845.

Xu, Weihuang; Souly, Nasim, and Brahma, Pratik Prabhanjan, 2021. Reliability of GAN Generated Data to Train and Validate Perception Systems for Autonomous Vehicles. In: *2021 IEEE Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 171–180. Available also from: https://doi.org/10.1109/WACVW52041.2021.00023.

Yang, Donglin; Cai, Xinyu; Liu, Zhenfeng; Jiang, Wentao; Zhang, Bo; Yan, Guohang; Gao, Xing; Liu, Si, and Shi, Botian, 2024. Realistic Rainy Weather Simulation for LiDARs in CARLA Simulator. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 951–957. Available also from: https://doi.org/10.1109/IROS58592.2024.10802036.

Yang, Kairui; Ma, Enhui; Peng, Jibing; Guo, Qing; Lin, Di, and Yu, Kaicheng, 2023. BEVControl: Accurately Controlling Street-view Elements with Multi-perspective Consistency via BEV Sketch Layout. *arXiv preprint arXiv:2308.01661*. Available also from: https://doi.org/10.48550/arXiv.2308.01661.

Yang, Wenhan; Tan, Robby T.; Feng, Jiashi; Liu, Jiaying; Guo, Zongming, and Yan, Shuicheng, 2017. Deep Joint Rain Detection and Removal from a Single Image. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1685–1694. Available also from: https://doi.org/10.1109/CVPR.2017.183.

Yu, Alex; Ye, Vickie; Tancik, Matthew, and Kanazawa, Angjoo, 2021. pixelNeRF: Neural Radiance Fields from One or Few Images. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4576–4585. Available also from: https://doi.org/10.1109/CVPR46437.2021.00455.

Yu, Fisher; Chen, Haofeng; Wang, Xin; Xian, Wenqi; Chen, Yingying; Liu, Fangchen; Madhavan, Vashisht, and Darrell, Trevor, 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In:

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2633–2642. Available also from: https://doi.org/10.1109/CVPR42600.2020.00271.

Zhang, Han; Xu, Tao; Li, Hongsheng; Zhang, Shaoting; Wang, Xiaogang; Huang, Xiaolei, and Metaxas, Dimitris N., 2017. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 41, pp. 1947–1962. Available also from: https://doi.org/10.48550/arXiv.1710.10916.

Zhang, Lvmin; Rao, Anyi, and Agrawala, Maneesh, 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. Available also from: https://doi.org/10.1109/ICCV51070.2023.00355.

Zhang, Richard; Isola, Phillip; Efros, Alexei A.; Shechtman, Eli, and Wang, Oliver, 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Available also from: https://doi.org/10.48550/arXiv.1801.03924.

Zheng, Yang; Harley, Adam W.; Shen, Bokui; Wetzstein, Gordon, and Guibas, Leonidas J., 2023. PointOdyssey: A Large-Scale Synthetic Dataset for Long-Term Point Tracking. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19798–19808. Available also from: https://doi.org/10.1109/ICCV51070.2023.01818.

Zhou, Brady and Krähenbühl, Philipp, 2022. Cross-view Transformers for real-time Map-view Semantic Segmentation. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13750–13759. Available also from: https://doi.org/10.1109/CVPR52688.2022.01339.

Zhou, Qian-Yi; Park, Jaesik, and Koltun, Vladlen, 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv preprint arXiv:1801.09847*. Available also from: https://doi.org/10.48550/arXiv.1801.09847.

Zhou, Yunsong; Simon, Michael; Peng, Zhenghao Mark; Mo, Sicheng; Zhu, Hongzi; Guo, Minyi, and Zhou, Bolei, 2024. Simgen: Simulator-conditioned driving scene generation. *Advances in Neural Information Processing Systems*. Vol. 37, pp. 48838–48874. Available also from: https://doi.org/10.48550/arXiv.2406.09386.

Zhu, Jun-Yan; Park, Taesung; Isola, Phillip, and Efros, Alexei A., 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251. Available also from: https://doi.org/10.1109/ICCV.2017.244.