

NeuralFloors: Conditional street-level scene generation from BEV semantic maps via Neural Fields

Valentina Muşat^{1†}, Daniele De Martini^{1‡}, Matthew Gadd^{1‡} and Paul Newman¹

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

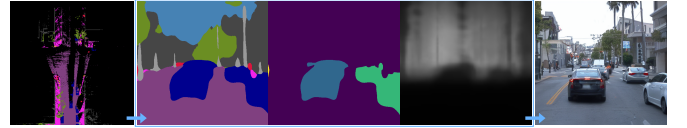


Fig. 1: *NeuralFloors* synthesises high-quality FV imagery (right) from a BEV segmentation map (left) in two steps: one generates FV semantic, instance and depth maps (centre), and the following synthesises an RGB image from the output of the first step.

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.

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 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

Manuscript received: August 4, 2023; Revised: November 13, 2023; Accepted: December 14, 2023

This paper was recommended for publication by Editor Cesar Cadena Lema upon evaluation of the Associate Editor and Reviewers’ comments.

This work was supported by a DeepMind Engineering Science Scholarship and the EPSRC Programme Grant “From Sensing to Collaboration” (EP/V000748/1). 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.

†Equal contribution ‡Corresponding author ¹Authors are with the Mobile Robotics Group (MRG), University of Oxford, OX13PJ Oxford, United Kingdom {valentina, danielle, mattgadd, pnnewman}@robots.ox.ac.uk

Digital Object Identifier (DOI): see top of this page.

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 [10], [9] 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 synthesise 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 synthesise 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.

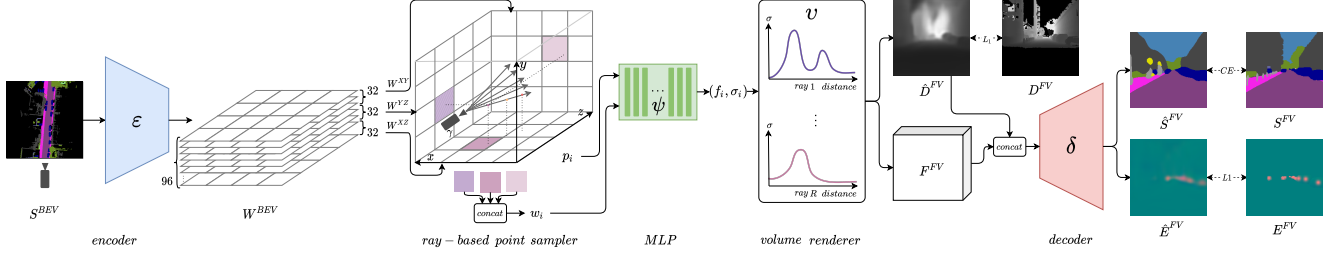


Fig. 2: **Stage 1 of our approach.** BEV segmentation S^{BEV} is encoded by ε into a 2D latent representation W^{BEV} from which features w_i are sampled. An MLP ψ produces lifted features f_i and densities σ_i corresponding to 3D coordinates p_i . Volume renderer v further produces FV feature and depth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

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 images. We discuss in the remainder of this section the two stages separately. Please refer to Tab. 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 v is used to integrate the features f and densities σ across each ray $r \in R$ to obtain final FV feature and depth images $(F^{FV}, \hat{D}^{FV}) = v(f, \sigma)$.

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

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

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:

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

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:

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

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

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

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:

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

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 :

$$\mathcal{L}_I = |I^{FV} - \hat{I}^{FV}| \quad (6)$$

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

where $\pi = \delta \circ v \circ \psi \circ \gamma \circ \varepsilon$ 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 :

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

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:

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

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 $\hat{\epsilon} = \epsilon_\theta(z_t, z_c, t)$.

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 [10], [9].

TABLE I: Notation. In the text, terms with the symbol $\hat{\cdot}$ represent the predicted counterpart of the GT term.

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
v	volume renderer
π, τ	generator and discriminator
ϕ	LDM extra layers

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 22 300 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 20 357 total pairs, out of which 16 674 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 39 791 total pairs, out of which 31 801 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 m \times 80 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 (Tab. II, Tab. III), we use KITTI-360 camera intrinsics. For experiments on nuScenes and Waymo (Tab. IV)

TABLE II: Baselines vs. NeuralFloors on KITTI-360 dataset

Model	FID↓		KID↓		mIoU↑	mIoU-align↑	RMSE↓
	@64	@512	@64	@512			
(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

TABLE III: Single stage base experiments on KITTI-360 dataset.

Model	FID↓		KID↓		mIoU↑	mIoU-align↑	RMSE↓
	@64	@512	@64	@512			
(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

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.

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

TABLE IV: NeuralFloors on nuScenes and Waymo dataset.

Model	FID↓		KID↓		mIoU↑	mIoU-align↑	RMSE↓
	@64	@512	@64	@512			
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

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 m, 80 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 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:

- (LDM+LDM): outputs \hat{S}^{FV} in 1st stage, outputs \hat{I}^{FV} in 2nd; trained with \mathcal{L}_{LDM} in both stages.
- (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.
- (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 (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 v on the architectures proposed in [3].

E. Implementation and training details

The integrals in Eqs. (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

Tabs. 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 (GSN) 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 Tab. 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

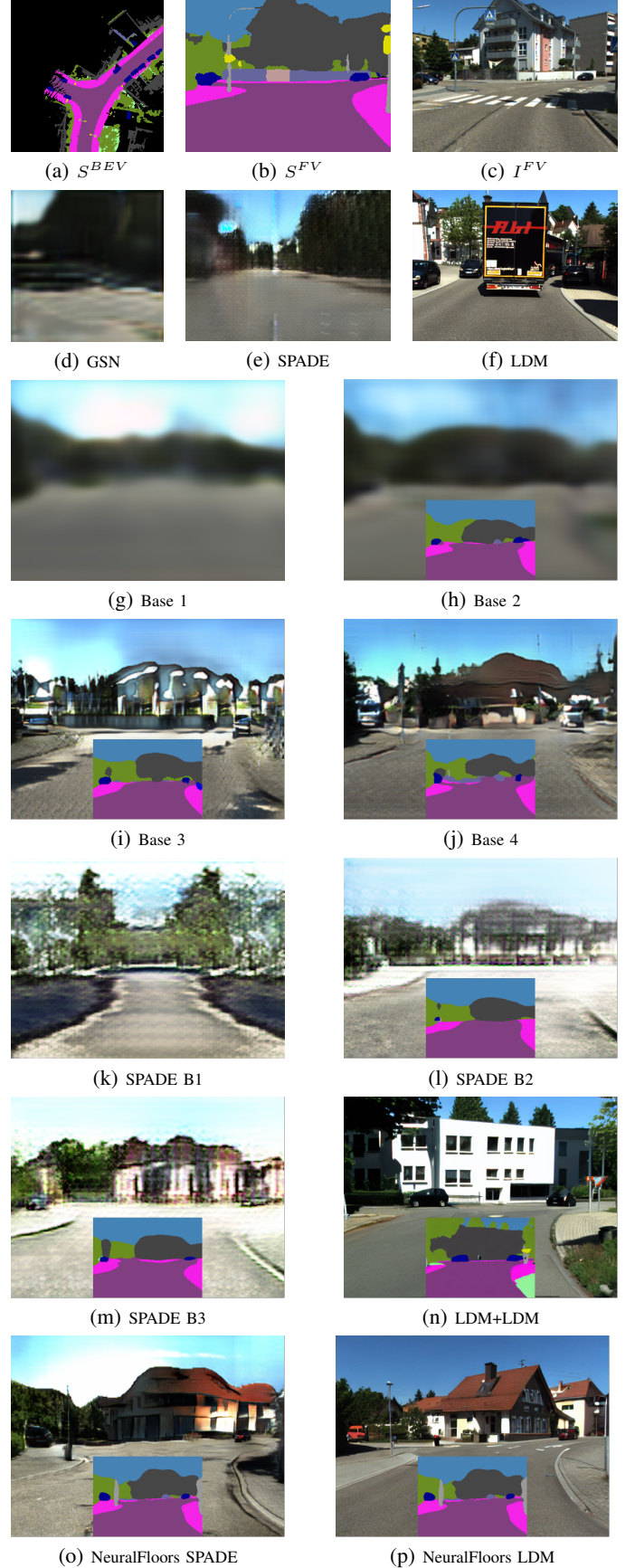


Fig. 3: Qualitative examples of FV output across all methods. GT illustrated for reference (1st row).

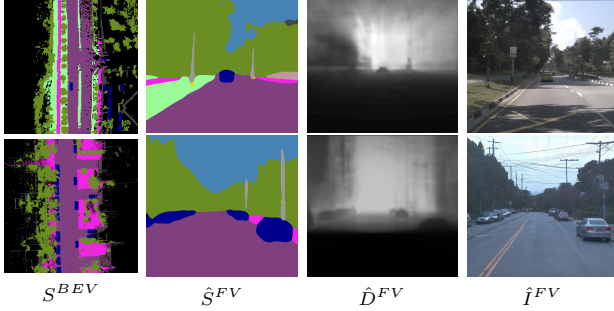


Fig. 4: FV output: nuScenes (1st row) and Waymo (2nd row).

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.

From a qualitative point of view, we see that (**GSN**) Fig. 3d 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. 3f that (**LDM**) produces realistic looking images, 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. 3p 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 Tabs. 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 (Sec. 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 Tab. 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.

TABLE V: Ablation first stage BEV \rightarrow FV semantic lifting.

Model	mIoU \uparrow	RMSE \downarrow
(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

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 Tab. VI, where the inclusion of instances did not have a net beneficial effect but the inclusion of depth significantly improved FID and KID.

VII. CONCLUSIONS

We have presented a novel system for synthesising RGB FV imagery from a BEV segmentation map. Our contribution in

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

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.

VIII. 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.

REFERENCES

- [1] S. Tan, Y. Zhang, X. Piao, J. Liu, C. Chen, B. Chen, Y. Chen, W. Wang, and C. Feng, "Scenegen: Learning to generate realistic traffic scenes," in *NeurIPS*, 2020. 1
- [2] C. Richter, N. Roy, and V. Koltun, "Playing for benchmarks," in *ICCV*, 2017, pp. 2223–2232. 1
- [3] 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. 1, 2, 3, 5, 6
- [4] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "GANcraft: Unsupervised 3D Neural Rendering of Minecraft Worlds," in *ICCV*, 2021. 1
- [5] C. Lin, H.-Y. Lee, W. Menapace, M. Chai, A. Siorohin, M.-H. Yang, and S. Tulyakov, "Infinicity: Infinite-scale city synthesis," 01 2023. 2, 7
- [6] A. Swerdlow, R. Xu, and B. Zhou, "Street-view image generation from a bird's-eye view layout," *ArXiv*, vol. abs/2301.04634, 2023. 2, 5
- [7] 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. 2, 4
- [8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020. 2, 4
- [9] X. Tian, T. Jiang, L. Yun, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *arXiv preprint arXiv:2304.14365*, 2023. 2, 4
- [10] T. W. Team. (2021) Simulation city: Introducing waymo's most advanced simulation system yet for autonomous driving. [Online]. Available: <https://blog.waymo.com/2021/06/SimulationCity.html> 2, 4
- [11] 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. 2
- [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. 153 171–153 187, 2021. 2
- [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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2
- [14] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," 06 2018, pp. 8808–8816. 2
- [15] V. Musat, D. D. Martini, M. Gadd, and P. Newman, "Depth-sims: Semi-parametric image and depth synthesis," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2388–2394, 2022. 2
- [16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7230–7240. 2
- [17] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus, "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," *arXiv preprint arXiv:2111.12083*, 2021. 2
- [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," in *ECCV*, 2020. 2, 6
- [19] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2
- [20] 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. 2, 3
- [21] Y. Xue, Y. Li, K. Singh, and Y. Lee, "Giraffe hd: A high-resolution 3d-aware generative model," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 18 419–18 428. 2
- [22] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, and Y. Liao, "Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation," in *International Conference on 3D Vision (3DV)*, 2022. 2
- [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," 07 2022. 2
- [24] 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. 3, 4
- [25] K. Deng, A. Liu, J. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," *CoRR*, vol. abs/2107.02791, 2021. 3
- [26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021. 4, 5, 6
- [27] 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. 5
- [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 *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. 5
- [29] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *International Conference on Learning Representations*, 2018. 5
- [30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818. 5
- [31] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346. 5, 6
- [32] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. 5
- [33] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017. 5
- [34] E. Schöenfeld, 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. 6