Street-View Image Generation From a Bird's-Eye View Layout

Alexander Swerdlow, Runsheng Xu, and Bolei Zhou, *Member, IEEE*

***Abstract*—Bird's-Eye View (BEV) Perception has received increasing attention in recent years as it provides a concise and unified spatial representation across views and benefits a diverse set of downstream driving applications. At the same time, data-driven simulation for autonomous driving has been a focal point of recent research but with few approaches that are both fully data-driven and controllable. Instead of using perception data from real-life scenarios, an ideal model for simulation would generate realistic street-view images that align with a given HD map and traffic layout, a task that is critical for visualizing complex traffic scenarios and developing robust perception models for autonomous driving. In this letter, we propose BEVGen, a conditional generative model that synthesizes a set of realistic and spatially consistent surrounding images that match the BEV layout of a traffic scenario. BEVGen incorporates a novel cross-view transformation with spatial attention design which learns the relationship between cameras and map views to ensure their consistency. We evaluate the proposed model on the challenging NuScenes and Argoverse 2 datasets. After training, BEVGen can accurately render road and lane lines, as well as generate traffic scenes with diverse different weather conditions and times of day.**

***Index Terms*—Computer vision for automation, deep learning methods, simulation and animation.**

I. Introduction

BEV perception for autonomous driving is a fast-growing research area, with the goal of learning a cross-view representation that transforms information between a perspective and a bird's-eye view. Such a representation can be used in downstream tasks such as path planning and trajectory forecasting [1]. The recent successes in BEV perception, whether for monocular [2] or multi-view images [3], [4], focus on the predictive aspect of BEV perception with street-view images as input and a semantic BEV layout as output. However, the generative side of BEV perception–which aims at synthesizing realistic street-view images from a given BEV semantic layout–is far less explored. A BEV layout concisely describes a traffic scenario at the semantic level and thus it is a natural representation to use to generate corresponding street-view images. This is the first work we are aware of to introduce and tackle such a task. There are many applications for this new BEV-conditioned generative model. For example, we can create synthetic training data for perception models or visualize safety-critical situations.

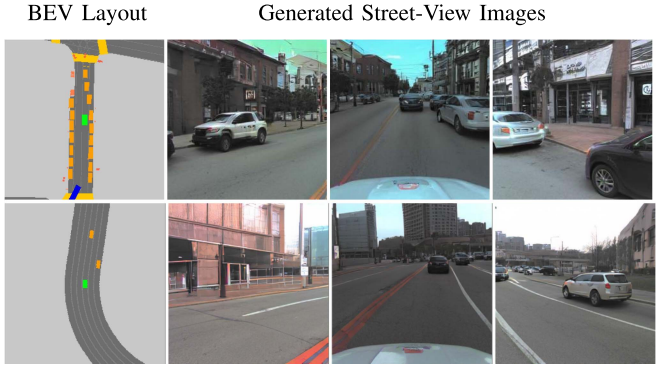

Fig. 1. Proposed BEVGen model generates realistic and spatially consistent street-view images from BEV layout. There are 3 camera views surrounding the ego vehicle as indicated by the green rectangle in the BEV layout.

Whereas most current approaches to synthetic training data involve a complex simulator or 3D reconstructed meshes, a controllable generative model is not only simpler but naturally provides diverse image generation. Another benefit provided by BEV generation is the ease of visualizing and editing traffic scenes. In the case of self-driving vehicles, we often care about a small set of rare scenarios where an accident is likely to happen. These corner-cases represent the long-tail distribution that is a key obstacle to robust autonomous driving [5], [6]. Human users can intuitively edit a BEV layout and then use a generative model to output a diverse set of corresponding street-view images for the stress test of the driving system. As shown in Fig. 1, our proposed **BEVGen** model can synthesize realistic street-view images from the BEV layout either collected from real world or provided by a driving simulator.

The fundamental question for BEV generation is: what is a plausible set of street-view images that correspond to a BEV layout? One could think of numerous images with varying vehicle types, backgrounds, and more. For a set of views to be realistic, we must consider several properties of the images. Similar to the problem of novel view synthesis, images must appear consistent–as if they were taken in the same physical location. For instance, cameras with an overlapping field-of-view (FoV) must ensure overlapping content is correctly shown, accounting for the shifted perspective. The visual styling of the scene also

Manuscript received 18 October 2023; accepted 31 January 2024. Date of publication 21 February 2024; date of current version 5 March 2024. This letter was recommended for publication by Associate Editor P. Gao and Editor A. Bera upon evaluation of the reviewers' comments. This work was supported by National Science Foundation under Grant 2235012. *(Corresponding author: Alexander Swerdlow.)*

The authors are with the Department of Computer Science, University of California, Los Angeles, Los Angeles, CA 90095 USA (e-mail: aswerdlow@ucla.edu; rxx3386@ucla.edu; bolei@cs.ucla.edu).

Code is open-source and available at github.com/alexanderswerdlow/BEVGen.

Digital Object Identifier 10.1109/LRA.2024.3368234

needs to be consistent such that all virtual views appear to be created in the same geographical area (e.g., urban vs. rural), at the same time of day, with the same weather conditions, and so on. In addition to this consistency, the images must correspond to the HD map, faithfully reproducing the specified road layout, lane lines, and vehicle locations. Unlike image-to-image translation with a semantic mask–which has significant amount of prior work–a BEV generation model must infer the image layout to account for occlusions between objects and the relative heights of objects in a scene.

In this work, we tackle this new task of generating street-view images from a BEV layout and propose a generative model to address the underlying challenges. We develop **BEVGen**, an autoregressive neural model that generates a set of n realistic and spatially consistent images as seen in Fig. 1. **BEVGen** has two technical novelties: (i) it incorporates spatial embeddings using camera intrinsics and extrinsics to allow the model to attend to relevant portions of the images and HD map, and (ii) it contains a novel attention bias and decoding scheme that maintains both image consistency and correspondence. Thus the model can generate high-quality scenes with spatial awareness and scene consistency across camera views. Compared to baselines, the proposed model obtains substantial improvement in terms of image synthesis quality and semantic consistency. The model can also render realistic scene images from out-of-domain BEV maps, such as those provided by a driving simulator or edited by a user. We summarize our contributions as follows:

- We tackle the new task of multi-view image generation from BEV layout. It is the first attempt to explore the generative side of BEV perception for driving scenes.
- We develop a novel generative model, **BEVGen**, that can synthesize spatially consistent street-view images by incorporating spatial embeddings and a pairwise camera bias.
- The model achieves high-quality synthesis results and shows promise for applications such as data augmentation and simulated driving scene rendering.

II. Related Work

Synthetic data-generation for autonomous driving has been widely studied, with early approaches relying on handcrafted assets in a game engine, and more recent data-driven approaches that augment or entirely replace specified handcrafted assets with learned generative models [7]. Prior works have demonstrated purely data-driven simulation [8], [9] but have not tackled multi-view generation and lack the ability to arbitrarily control the position of actors and objects.

Outside of autonomous driving applications, there has also been significant work on image generation models. One large area of research has been on image generation conditioned on modalities such as text and audio [10], [11] or more directly through semantic masks [12] or bounding boxes [13]. Our task has spatial constraints as in the latter tasks, but is distinct in that the constraints must be inferred from an alternate perspective which requires a 3D understanding of the scene.

Next, our task is closely related to–but distinct from, novel view synthesis (NVS), where the goal is to generate an image of a scene from a virtual perspective given a true image from a different perspective. This has been studied in the context of transforming satellite images into a corresponding street-view image, which is commonly referred to as cross-view synthesis [14]. Other approaches to NVS focus on smaller viewpoint transformations and use warping [15] or simply implicitly learn the transformation [16], [17]. BEV Generation requires a similar 3D understanding between different viewpoints as in NVS, but lacks the conditioning information provided by a source view(s) and requires consistency not only between frames but also with an HD map.

III. Method

In this section, we introduce the framework of the proposed BEVGen. The problem definition of BEV generation is that given a semantic layout in Birds-Eye View (BEV), $B \in \mathbb{R}^{H_b \times W_b \times c_b}$ with the ego at the center and c_b channels describing the locations of vehicles, roads, lane lines, and more (see Section IV-A), we would like to generate n images $I_k \in \mathbb{R}^{H_c \times W_c \times 3}$ under a set of n virtual camera views $(K_k, R_k, t_k)_{k=1}^{n}$, where K_k, R_k, t_k are the intrinsics, extrinsic rotation, and translation of the kth camera.

Fig. 2 illustrates the framework of the proposed BEVGen. BEVGen consists of two autoencoders modeled by VQ-VAE, one for images and one for the BEV representation, that allow the causal transformer to model scenes at a high level. The key novelty lies in how the transformer can relate information between modalities and across different views. The cross-view transformation encodes a cross-modal inductive 3D bias, allowing the model to attend to relevant portions of the HD map and nearby image tokens. We explain each part in more detail below.

A. Model Structure

Image Encoder: To generate a globally coherent image, we model our distribution in a discrete latent space instead of pixel space. We use the VQ-VAE model introduced by Oord et al. [18] as an alternative generative architecture to GANs.[1] We replace the L_2 with a perceptual loss and incorporate a patch-wise adversarial loss as in [19]. The VQ-VAE architecture consists of an encoder E^c, a decoder D^c, and a codebook $\mathcal{Z}^c = \{z_f\}_{f=1}^{F_c} \subset \mathbb{R}^{n_c}$ where F_c is the number of code vectors and n_c is the embedding dimension of each code. Given a source image, $x_k \in \mathbb{R}^{H_c \times W_c \times 3}$, we encode $\hat{z}_k^c = E^c(x_k) \in \mathbb{R}^{h_c \times w_c \times n_c}$. To obtain a discrete, tokenized representation, we find the nearest codebook vector for each feature vector $\hat{z}_{k,ij}^c \in \mathbb{R}^{n_c}$ where i, j are the row, column indices in the discrete latent representation with size $h_c \times w_c$:

$$z_{k,ij}^c = \arg\min_f \left\| \hat{z}_{k,ij}^c - z_f \right\| \in \mathbb{N}. \tag{1}$$

This creates a set of tokens $z_k^c \in \mathbb{N}^{h_c \times w_c}$ that we refer to as our image tokens. To generate an image from a set of tokens,

[1]Note that switching to the recently developed class of diffusion models can potentially improve the image synthesis quality, but such models require an order of magnitude of additional data and computational resources for training and thus we leave it for future works.

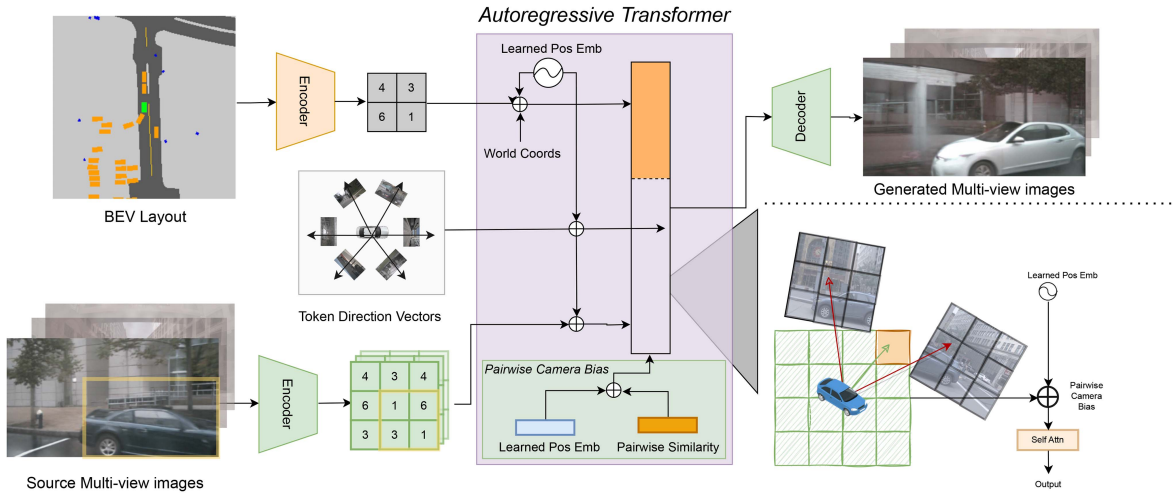

Fig. 2. BEVGen framework. A BEV layout and source multi-view images are encoded to a discrete representation and are flattened before passed to the autoregressive transformer. Spatial embeddings are added to both camera and BEV tokens inside each transformed bloc. We construct a pairwise matrix that encodes relationship between a given image token and another BEV/image token (bottom right). This learned pairwise camera bias is added to the attention weights. We pass the generated tokens to the decoder to obtain generated images during inference.

we first obtain $\tilde{z}_k^c \in \mathbb{R}^{h_c \times w_c \times n_c}$, by using z_k^c as an index to our codebook, $\mathcal{Z}_c$. We can later decode with a convolutional decoder, $D^c(\tilde{z}_k^c) \in \mathbb{R}^{H_c \times W_c \times 3}$, using the same architecture as [19].

BEV Encoder: To condition our model on a BEV layout, we use a similar discrete representation as for camera images, except we replace the perceptual and adversarial losses with a binary cross entropy loss for binary channels and an L_2 loss for continuous channels. We encode our BEV map h as before with $E^b(h) \in \mathbb{R}^{h_b \times w_b \times n_b}$ and $\mathcal{Z}^b = \{z_f\}_{f=1}^{F_b} \subset \mathbb{R}^{n_b}$ to obtain a set of tokens, $z^b \in \mathbb{N}^{h_b \times w_b}$. We discard the decoder stage, D^b, after training the 1st stage as it is not needed for our second stage or for inference.

Autoregressive Modeling: Given a BEV layout and a set of n camera parameters, we seek to generate n images by learning the prior distribution of a set of discrete tokens, z^c conditioned on $z^b, (K_k, R_k, t_k)_{k=1}^n$.

$$p(z^c|z^b, K, R, t) = \prod_{i=0}^{h_c \times w_c \times n} p(z_i^c|z_{<i}^c, z^b, K, R, t). \quad (2)$$

We model $p(.)$ by training a transformer τ that iteratively predicts a probability distribution over all possible tokens based on prior image tokens, discretized BEV features, and their respective camera parameters. We choose a transformer architecture as it provides global attention which aids in cross-view consistency. To implement this, we perform causal self-attention between image tokens such that image tokens cannot look at future tokens in the attention process. We allow global attention between image tokens and all BEV tokens. This serves as an extension to the prior modeling proposed in [18], [19] which we refer to for further details.

B. Spatial Embeddings

To help the model attend to relevant tokens both in the camera and BEV feature space, we introduce positional embeddings. We take inspiration from work on BEV segmentation [3] on alignment between the BEV and first-person view (FPV) perspectives.

Camera Embedding: In order to align image tokens with BEV tokens, we use the known intrinsics and extrinsics to reproject from image coordinates to world coordinates. Given a token in image space, $z_{k,ij}^c \in \mathbb{R}^2$, we convert to homogeneous coordinates, $\bar{z}_{k,ij}^c$ and obtain a direction vector in the ego frame as follows:

$$d_{k,ij} = R_k^{-1} K_k^{-1} \bar{z}_{i,jk}^c - t_k \quad (3)$$

We use a 1D convolution, $\theta_c(d) \in \mathbb{R}^{n \times h_c \times w_c \times n_{emb}}$, to encode our direction vector in the latent space of the transformer, n_{emb}. We encode our image tokens using a shared learnable embedding $\lambda_c(z_{k,ij}^c) \in \mathbb{R}^{n_{emb}}$, and add a per-token learnable parameter, $\Lambda_{k,ij}^c \in \mathbb{R}^{n_{emb}}$, across image tokens:

$$l_{k,ij} = \lambda(z_{k,ij}^c) + \theta(d_{k,ij}) + \Lambda_{k,ij}. \quad (4)$$

BEV Embedding: To align our BEV tokens with our image tokens, we perform a similar operation as in (4) and use the known BEV layout dimensions to obtain coordinates in the ego frame. We obtain $t_{yx} \in \mathbb{R}^2$, where y, x correspond to the row and column indices of discrete BEV representation, for each token and encode these into our transformer latent space, with $\theta_b(t) \in \mathbb{R}^{h_b \times w_b \times n_{emb}}$. We similarly use a shared learnable embedding for our discrete tokens, $\lambda(z_{yx}^b) \in \mathbb{R}^{n_{emb}}$, and a per-token learnable parameter, Λ_{yx}:

$$l_{yx} = \lambda(z_{yx}^b) + \theta_b(t_{yx}) + \Lambda_{yx}. \quad (5)$$

where l represents the final input embeddings for the transformer decoder block.

C. Camera Bias

In addition to providing the model with aligned embeddings, we add a bias to our self-attention layers that provides both an

intramodal (image to image) and intermodal (image to BEV) similarity constraint. This draws inspiration from [17], but instead of providing a blockwise similarity matrix that is composed of encoded poses between frames, we provide a per-token similarity based on their relative direction vectors. Our approach also encodes the relationship between image and BEV tokens. For self-attention in any layer between some query $q_r \in \mathbb{R}^{n_{emd}}$ and key/value $k_c/v_c \in \mathbb{R}^{n_{emd}}$, where r, c are the row, column of the pairwise attention matrix, we have:

$$\text{Attention}(q_r, k_c, v_c) = v_c \,\text{softmax}\left(\frac{a_{rc}}{\sqrt{d}}\right), \tag{6}$$

$$a_{rc} = q_r k_c + \beta_{rc}. \tag{7}$$

The transformer sequence is composed of $h_b w_b$ conditional BEV tokens followed by camera tokens. If $r, c > h^b w^b$, both positions correspond to image tokens and thus we have two direction vectors, d_r, d_c, computed as in (3). As discussed in Section III-A, we have a mapping between the sequence index and image token (i, j) in camera k. If $r > h^b w^b > c$, we have a query for some image token and a key/value pair corresponding to BEV token. Thus, we again construct two direction vectors. In this case our BEV direction vector consists of the 2D World coordinates (in the ego-center frame) and our image direction vector is the same as in (3) except with the row value as the center of the image. Given these two direction vectors, d_r, d_c, we add the cosine similarity and a learnable parameter, θ_{rc}, as shown in the bottom right of Fig. 2:

$$\beta_{rc} = \frac{d_r \cdot d_c}{\|d_r\|\|d_c\|} + \theta_{rc}. \tag{8}$$

IV. Experiments

A. Datasets

We evaluate the proposed method using the nuScenes dataset [20] and Argoverse 2 dataset [21], which are commonly used for BEV segmentation and detection. Each instance in both of our datasets contain ground-truth 3D bounding boxes, mutli-view camera images, calibrated camera intrinsics and extrinsics, and LiDAR. We project these 3D bounding boxes onto a BEV layout following standard practice used in BEV segmentation [3], [22].

NuScenes: The nuScenes dataset consists of 1000, 20-second scenes captured at 12 Hz by 6 cameras, which provide full 360° camera coverage. However, synchronization between cameras occurs only at 2 Hz so we increase our dataset size by resampling at 20 Hz, pruning instances where any camera is more than 100 ms outside of the frame. We linearly interpolate from the nearest ground-truth annotations to create paired annotation data. This provides roughly 9x the number of instances as the original training set, for a total of 260 k instances. For validation, we use the standard set containing 6 k instances. For all visualizations, we flip the back left and right cameras along the vertical axis to highlight the image consistency of our model.

Argoverse 2: Argoverse 2 comprises 1000, 15-second scenes annotated at 10 hz, with a synchronized camera captured at 20 hz. Compared to Argoverse 1 [23], it covers much broader and more diverse image content and is comparable to the size of nuScenes dataset. This provides 210 k and 30 k instances for train and validation, respectively. As the front camera is rotated 90°, we extract a square crop from the front three cameras for all experiments.

Preprocessing: The BEV layout representation used in training and testing covers 80 m × 80 m around the ego center. On nuScenes, there are 21 channels, with 14 channels being binary masks representing map information (lane lines, dividers, etc.) and actor annotations (cars, trucks, pedestrians, etc.). The remaining 7 channels provide instance information, including the visibility and size of the annotation. On nuScenes, we resize all images to $H \times W = 224 \times 400$, resulting in a discrete latent representation of $h_c \times w_c = 14 \times 25$. On Argoverse 2, we crop to $H \times W = 256 \times 256$, resulting in $h_c \times w_c = 16 \times 16$. Our BEV layout has a discrete latent representation of $h_b \times w_b = 16 \times 16$. We appropriately modify intrinsics after cropping and resizing. To enable our weighted cross-entropy loss, we project the provided 3D annotations onto the camera frame and weight the corresponding tokens in our discrete camera frame representation, $z_k \in \mathbb{N}^{h_c \times w_c}$.

B. Training Details

VQ-VAE: We train the 1st stage camera VQ-VAE with aggressive augmentation consisting of flips, rotations, color shifts, and crops. Similarly, we train our 1st stage BEV VQ-VAE with flips and rotations. For the 2nd stage, we add minimal rotations and scaling, as well as cropping.

Transformer: Our transformer is GPT-like [24] with 16-heads and 24-layers. We use DeepSpeed to facilitate sparse self-attention and 16-bit training. We clip gradients at 50 and use the AdamW optimizer [25] with $\beta_1, \beta_2 = 0.9, 0.95$ and a learning rate of λ = 5e-7. Both the BEV and image codebooks have $|\mathcal{Z}_c| = |\mathcal{Z}_b| = 1024$ codes with an embedding dimension, $n_c = n_b = 256$.

Additionally, we develop a lighter-weight model that uses sparse attention as in [26]. As opposed to a uniform random attention mask, we unmask regions of the image near the token we attend. Using the same formulation as in (8), we create a pairwise similarity matrix for image tokens only. As sparse attention groups the input sequence into discrete blocks, we perform average pooling on these blocks and use these values as weights for sampling. Additionally, we have a sliding window in which we always attend to the last r tokens, and we attend to all BEV tokens. For our sparse models, we have an attention mask density of 35% with a sliding window length of $r = 96$. Except as described in Section IV-D, our sparse model is derived from fine-tuning our full-attention model for 10 epochs.

C. Results

Qualitative result: Fig. 3 exhibits the generation examples from **BEVGen** on nuScenes. Our model is able to generate a diverse set of scenes including intersections, parking lots, and boulevards. We observe that each camera view not only correctly displays the surrounding of the same location, but

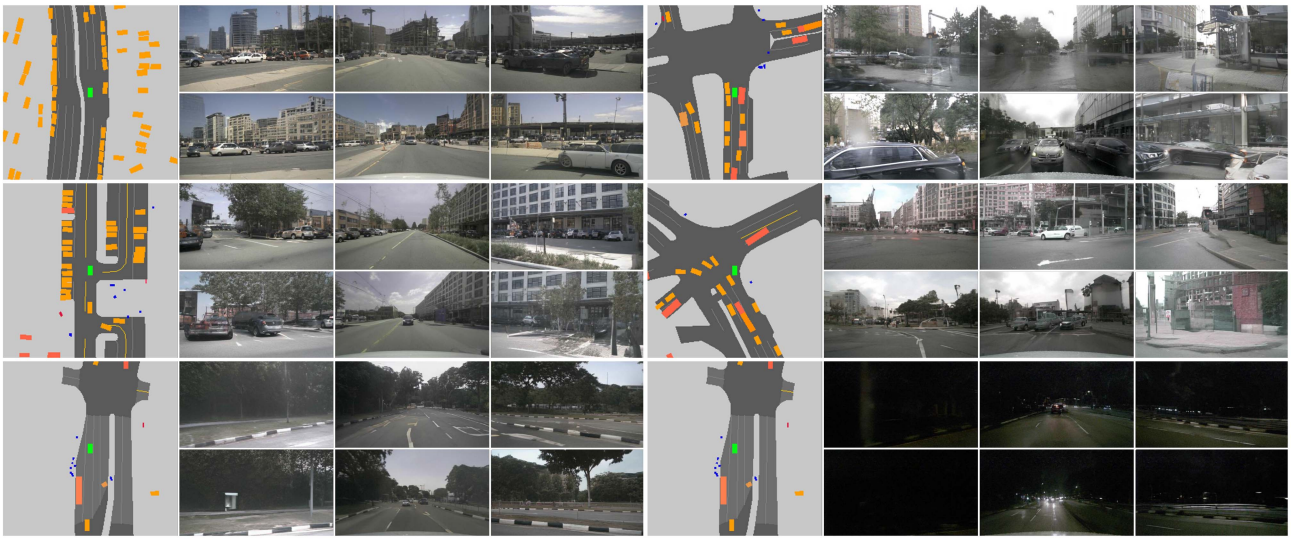

Fig. 3. Synthesized multi-view images from BEVGen on nuScenes. Image contents are diverse and realistic. The two instances in the bottom row use the same BEV layout for synthesizing the same location in day and night.

Fig. 4. Comparison to real-views on Argoverse 2. Images on the left are synthesized, images on the right are corresponding real ones.

also preserves the spatial perspective. **BEVGen** synthesizes images under various weather conditions, with the same weather apparent in all images, including physical artifacts such as rain. We also demonstrate that our model is capable of generating diverse scenes corresponding to the same BEV layout. We see at the bottom of Fig. 3 the same location rendered in the day and at night by the model.

In Fig. 4, we compare synthesized images to real ones for the same BEV layouts on Argoverse 2. We observe that our model places vehicles in the same location as the real one, even when a vehicle is in multiple views, demonstrating that the model learns to synthesize coherent content across views. *Quantitative result:* We use the Fréchet Inception Distance (FID) to evaluate our synthesized quality compared to the source images. Metrics are calculated on the validation set for each respective dataset. We employ no post-generation filtering. For calculating FID scores, we use clean-fid [28].

To differentiate between the performance of our 1st and 2nd stage, we compare our results to the results obtained by feeding the encoded tokens directly to the decoder, as is done when training the 1st stage. This represents the theoretical upper bound of our model's performance, largely removing the effect of the 1st stage which is not the focus of this letter.

As seen in Table I, our **BEVGen** model achieved an FID score of 25.54 on nuScenes, compared to the baseline score of 138.30. This is in comparison to our reference upper-bound FID score of 9.37. Our model utilizing our sparse masking design from Section 3.4 achieved an FID score of 28.67. This sparse variant

TABLE I
BASELINE COMPARISON OVER ALL 6 VIEWS ON NUSCENES VALIDATION

Method	FID↓	Road mIoU↑	Vehicle mIoU↑
X-Seq[27]	138.30	-	-
BEVGen	25.54	50.20	5.89
BEVGen Sparse	28.67	50.92	6.69

TABLE II
ABLATION OF THE KEY MODEL COMPONENTS

Method	FID↓	VSC↑
Row-major Decoding	43.18	4.33
Center-out Decoding	42.32	4.53
+ Camera Bias	41.20	4.74
+ Camera Bias, Spatial Embedding	**40.48**	**6.24**
+ Sparse, Camera Bias, Spatial Embedding	48.31	5.44

is approximately 48% faster during inference and 40% faster for training. On Argoverse 2, our model achieves an FID score of 25.51.

While FID is a common metric to measure image synthesis quality, it fails to entirely capture the design goals of our task and cannot reflect the synthesis quality of different semantic categories. Since we seek to generate multi-view images consistent with a BEV layout, we wish to measure our performance on this consistency. To do this, we leverage a pre-trained BEV segmentation network CVT from [3]. We apply the CVT to the generated images and then compare the predicted layout with the ground-truth BEV layout. We report both the road and vehicle class mean intersection-over-union (mIoU) scores. As shown in Table I, we beat our baseline by 4.4 and 1.45 for road and vehicle classes respectively. Note that the performance of the BEV segmentation model on the validation set is 66.31 and 27.51 for road and vehicle mIoU respectively. This reveals that though the model can generate road regions in the image in a reasonable manner, it still has a limited capability of generating high-quality individual vehicles that can be recognized correctly by the segmentation network. This is a common problem for scene generation where it remains challenging to synthesize the small objects entirely. Our work is a starting point and we plan to improve small object synthesis in future work.

Finally, we seek to quantitatively verify the cross-view consistency of our model. To do this, we introduce a consistency metric that looks at the similarity between overlapping portions of images. For example, for the front-left and front-center camera on nuScenes, we extract a window on the right and left edges of the images, respectively. We then attempt to find keypoint correspondences between these image patches by performing feature matching with [29]. As each correspondence has a confidence, we simply find the sum of these confidences and average across all windows to obtain a metric for cross-view consistency. We call this metric the View Consistency Score (VSC). On nuScenes, synthesized view images from our model obtain a VSC of 5.65, with the corresponding real images obtaining a result of 12.86. On Argoverse 2, synthesized view images obtain a VSC of 13.00, with the corresponding real images obtaining a result of 42.90. We see further results in Table II. We show

Fig. 5. We display point correspondences (blue) obtained with LoFTR for the VSC metric. Green lines indicate inliers from RANSAC. Top: Generated images, Bottom: Real images.

matched keypoints in Fig. 5, with both images corresponding to the same BEV layout. We note that this metric does not evaluate whether there is a proper perspective shift between the viewpoints, only that there are point correspondences in overlapping regions. Qualitatively, BEVGen consistently generates a realistic shift in perspective between views.

As far as we know, our work is the first attempt at synthesizing street views conditioned on a BEV layout. To establish a comparison baseline, we reproduce one of the cross-view synthesis methods proposed in [27]. Firstly, we project the BEV semantic mask onto the camera perspective view using extrinsics and intrinsics. Next, we utilize a conditional GAN to generate a street view semantic mask from this warped BEV layout. Finally, we use another conditional GAN to perform the semantic mask-to-image translation. As neither the nuScenes nor the Argoverse datasets provide 2D segmentation labels, we use a pre-trained segmentation model [30] to generate pseudo ground-truth labels for the first conditional GAN. Following [27], we use the Pix2Pix [12] model for the first and second stage. However, at their core, existing cross-view synthesis methods such as this one are not designed to handle the BEV generation task as, at their core, they translate existing rich RGB information rather than generate new images from a sparse semantic layout.

D. Ablation Study

To verify the effectiveness of our design choices, we run an ablation study on key features of our model. We run these experiments on the same subset of the nuScenes validation set as in Section IV-C, but only consider the 3 front-facing views to reduce training time. The 3 front-facing views have a larger FoV overlap and capture more relevant scene features compared to the side-facing rear views. This area is more relevant to our task and still allows us verify the model design objectives.

We test four variants of our model, one with only center-out decoding, one with our camera bias, one with the camera bias and spatial embeddings, and a final model that we train from scratch using our sparse masking, instead of fine-tuning. Table II shows

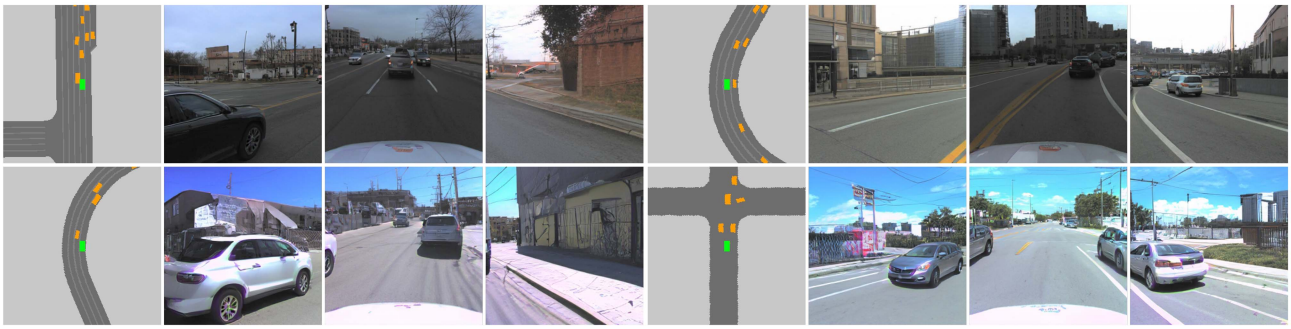

Fig. 6. Generating images based on the BEV layouts provided by the MetaDrive driving simulator.

a steady decrease in FID scores and increase in VSC, as we add the camera bias, and spatial embeddings.

V. Applications

Generating realistic images from BEV layout has many applications. In this section we explore the applications of data augmentation for BEV segmentation and image generation from simulated BEV.

Image generation from simulated BEV: Since one motivation for our task definition lies in the simplicity of the BEV layout, we wish to determine whether this enables our model to generate new scenes from out-of-domain (OOD) HD maps. We use MetaDrive simulator [31] to procedurally generate traffic scenarios and input these BEV layouts into **BEVGen**. Generated images are shown in Fig. 6. Scenario generation that addresses the long-tail problem is itself an active area of research [32], [33] and outside the scope of our work, but we clearly demonstrate that we can synthesize such synthetic scenarios into real images using the BEV layout as a bridge. This has potential to address the sim2real gap.

Additionally, we demonstrate that we can generate a safety-critical scenario by editing an existing scene. This functionality is an important component of the safety testing for many autonomous driving systems as it allows for both permutation testing for real-life scenarios but also for a human developer to construct difficult scenarios would otherwise be dangerous and observe the behavior of a system. We demonstrate this editing ability in Fig. 7 where we translate cars in two different scenes.

Data augmentation for BEV segmentation: An important application of our BEV conditional generative model is generating synthetic data to improve prediction models. Thus, we seek to verify the effectiveness of our model by incorporating our generated images as augmented samples during the training of multiple camera-only BEV segmentation and 3D detection models. For BEV Segmentation, we use CVT [3], which is also used in Section IV-C, and compare our results to training without any synthetic samples. We generate 6,000 unique instances using the BEV layout from the *train* set on nuScenes and use these to augment the original training set. These synthetic instances are associated with the ground truth BEV layout for training, with no relation to results from Section IV-C. For the detection task, we verify on the state-of-the-art detector BEVFormer [34]. We select two model sizes for testing and apply the same augmentation regime used for CVT. As seen in Table III, our synthetic data improves validation mIoU for BEV Segmentation by 0.6 for both the road category and the vehicle category. Additionally, we see more substantial gains in BEV Object Detection with improvements of 1.6 and 1.3 for mAP and NDS respectively for the `Tiny` model and 2.9 and 2.7 respectively for the `Small` model.

Fig. 7. We modify two existing BEV layouts to demonstrate spatial disentanglement of the model.

TABLE III
Data Augmentation Results for BEV Segmentation and Detection on the Validation Set of Nuscenes

	CVT		BEVFormer T		BEVFormer S	
	Road mIoU↑	Vehicle mIoU↑	mAP↑	NDS↑	mAP↑	NDS↑
w/o aug	71.3	36.0	25.7	35.9	37.0	47.9
w/ aug	**71.9**	**36.6**	**27.3**	**37.2**	**39.9**	**50.6**

VI. Conclusion

In this letter we introduce this new BEV generation task and propose and approach in the form of a generative model we call **BEVGen**. After training on real-world driving datasets, the proposed model can generate spatially consistent multi-view images from a given BEV layout. We further show its application in data augmentation and simulated BEV generation.

References

[1] Y. Zhang et al., "BEVerse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," 2022, *arXiv:2205.09743*.

[2] N. Gosala and A. Valada, "Bird's-eye-view panoptic segmentation using monocular frontal view images," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1968–1975, Apr. 2022.

[3] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13750–13759.

[4] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "CoBEVT: Cooperative Bird's eye view semantic segmentation with sparse transformers," in *Proc. Conf. Robot Learn.*, 2022, pp. 989–1000.

[5] O. Makansi, O. Cicek, Y. Marrakchi, and T. Brox, "On exposing the challenging long tail in future prediction of traffic actors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13127–13137.

[6] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, "Corner cases for visual perception in automated driving: Some guidance on detection approaches," 2021, *arXiv:2102.05897*.

[7] F. Mütsch, H. Gremmelmaier, N. Becker, D. Bogdoll, M. R. Zofka, and J. M. Zöllner, "From model-based to data-driven simulation: Challenges and trends in autonomous driving," 2023, *arXiv:2305.13960*.

[8] S. W. Kim, J. Philion, A. Torralba, and S. Fidler, "DriveGAN: Towards a controllable high-quality neural simulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5820–5829.

[9] A. Amini et al., "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1143–1150, Apr. 2020.

[10] A. Ramesh et al., "Zero-shot text-to-image generation," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8821–8831.

[11] J. Li et al., "Direct speech-to-image translation," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 3, pp. 517–529, Mar. 2020.

[12] 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.

[13] Z. Li, J. Wu, I. Koh, Y. Tang, and L. Sun, "Image synthesis from layout with locality-aware mask adaption," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 13799–13808.

[14] A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixe, "Coming down to earth: Satellite-to-street view synthesis for Geo-localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6484–6493.

[15] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "SynSin: End-to-end view synthesis from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7465–7475.

[16] R. Rombach, P. Esser, and B. Ommer, "Geometry-free view synthesis: Transformers and no 3D priors," in *Proc IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 14336–14346.

[17] X. Ren and X. Wang, "Look outside the room: Synthesizing a consistent long-term 3D scene video from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3553–3563.

[18] A. V. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6309–6318.

[19] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12868–12878.

[20] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11618–11628.

[21] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021.

[22] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 194–210.

[23] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8748–8757.

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog.*, vol. 1, no. 8, 2019, Art. no. 9.

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019, *arXiv:1711.05101*.

[26] M. Zaheer et al., "Big bird: Transformers for longer sequences," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 17283–17297.

[27] K. Regmi and A. Borji, "Cross-view image synthesis using conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3501–3510.

[28] G. Parmar, R. Zhang, and J.-Y. Zhu, "On aliased resizing and surprising subtleties in GAN evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11410–11420.

[29] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8922–8931.

[30] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, vol. 12351, pp. 173–190.

[31] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3461–3475, Mar. 2023.

[32] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou, "TrafficGen: Learning to generate diverse and realistic traffic scenarios," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3567–3575.

[33] Q. Li et al., "ScenarioNet: Open-source platform for large-scale traffic scenario simulation and modeling," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2024.

[34] Z. Li et al., "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–18.