DiffusionPillars: Probabilistic Scene Generation for Object Detection and View Synthesis

Julian Ost
Princeton University
julian.ost@princeton.edu

Gene Chou
Cornell University
gc492@cornell.edu

Shruthi Santhanam
Princeton University
shruthisanth@princeton.edu

Felix Heide
Princeton University
fheide@cs.princeton.edu

Abstract

3D scene perception, namely object detection, and generation are closely related problems, but existing works typically address them in isolation. 3D object detection from monocular images relies on explicit priors and inductive biases about the scene structure, such as ground planes and depth distribution. Generative methods, such as probabilistic diffusion models, have shown to be effective learners of empirical priors. In this work, we join scene perception and generative tasks by re-framing object detection as a conditional generative process with a learned prior – closing the cycle by reconstructing and manipulating the observed image.

We model the features, poses, and size of each object instance explicitly on a ground plane which are produced as the output of a conditional generative process from an input observation. This underlying representation of each object models an elliptical feature representation on the ground plane that gets unprojected in the height dimension. The proposed diffusion-based neural rendering method simultaneously allows for 3D object detection, 3D multi-object scene generation, and scene reconstruction from a single image.

1 Introduction

3D perception is fundamental to many graphics and vision tasks. In particular, 3D object detection requires understanding the composition and semantics of a scene and is widely used for autonomous navigation [10, 1], robotic manipulation [3, 13], and augmented and virtual reality [52, 36]. 3D object detection from monocular images relies heavily on explicit priors and inductive biases about the scene structure, such as ground planes and depth distribution [6, 4, 51, 48, 28]. Existing works extract scene layouts using feed-forward networks, but we hypothesize that these layout priors can be better learned through generative methods that are trained on 3D layouts. By modeling the inherent structure of scenes, generative models capture complex spatial relationships and object interactions, providing a more robust and data-driven prior for perception. In this work, we propose a novel framework that combines generative and perception models, and creates a shared learned representation for 3D multi-object scene generation, 3D object detection, and scene reconstruction from a single image.

Generative methods have been extensively researched as generating three-dimensional scene content is fundamental to graphics and vision. In computer graphics, generating 3D assets and scenes [33, 56, 5] help create realistic virtual worlds. Computer vision for robotics [7, 25] and autonomous driving [62, 76] requires simulated scenes of high diversity and controllability that are rare or absent in real-world captures, e.g., disaster scenarios or unseen object poses for lost cargo. However, current works lack understanding of three-dimensional scene layouts. Previous compositional approaches for

Preprint. Under review.

multi-object scene generation [43, 42, 14] produced convincing outputs, but they do not exploit the organization of the scene. Blob-GAN [15] reconstructs scene layouts from image observations by representing each object with a feature blob and by applying GAN inversion techniques. However, this inversion requires an image encoder that is trained separately and cannot directly exploit the prior knowledge hidden in the generative process of a scene. Although their two-staged process allows for manipulation of the image composition, it lacks understanding of the underlying three-dimensional scene layout.

Detection can be performed by guiding the generation process for a specific input image. Therefore we propose to *reformulate 3D object detection as an inverse rendering problem*, where a conditional probabilistic diffusion process synthesizes an intermediate scene layout for a multi-object scene from a single input image. Formulating perceptive tasks as a probabilistic generation of world representations has been explored for years [32, 77, 26, 74] as analysis by synthesis, where models of the world help interpret observations. Our integration of perception and generation allows the models to complement each other in understanding and predicting object locations, orientations, and semantics.

To combine detection and generating layouts as one end-to-end training pipeline instead of requiring multi-stage training, we leverage neural rendering and view synthesis [40, 44, 58, 72, 47, 30, 57]. Specifically, we enforce 3D consistency by supervising with novel view synthesis. As shown in Figure 1, from an input image we eventually arrive at a novel view, and thus close the cycle.

We investigate a joint detection and generative diffusion approach that benefits from the mutual task learning paradigms. In the first step, we generate the scene layout on the BEV plane with a diffusion model, where each cell represents a background feature or an object and its size, location, orientation and appearance features. In the non-generative case, where a specific layout shall be reconstructed from an input view, we condition the diffusion process on a BEV feature map extracted from an image and are able to guide the layout generation towards reconstructing the scene layout, which comprises 3D object detection. The second step performs novel view synthesis. We first splat feature ellipses for each object on the ground plane, extrude those inside the objects bounding boxes, and project these elliptic pillars into the image plane. From the feature image, we render a novel view of the scene, using condition a diffusion-based rendering process. Both steps can be executed in a conditional case to perform 3D object detection and image reconstruction or unconditionally to generate novel scene layouts and views. By adding a conditional image input the proposed method is able to fulfill the perceptual task and predict each object's location, size, and features. This layout can later be manipulated to manipulate the layout and camera and render novel views of the same or unseen scene.

In summary, our contributions are:

- We introduce a novel inverse rendering method that joins object detection and scene generation with conditional diffusion. When conditioned on a single image, we generate plausible BEV layouts image, and without a condition we learn to generate novel unseen scene layouts.
- To close the cycle and select plausible layout given an image, we devise a novel efficient rendering method that operates on the projections of the features in the 3D scene located on the ground plane. Together with the layout generation model, the method disentangles layout and appearance in the scene.
- We validate the proposed method for 3D object detection and novel scene and view generation and find that the proposed inverse-rendering approach performs on par with existing feed-forward detectors.

2 Related Work

3D Scene Representations and Neural Rendering. Recent methods represent 3D scenes *implicitly*. This includes NeRF [40] and variants [76, 49, 39, 41] that have been extended to multi-object scenes [69, 47, 30, 71, 18]. A growing body of work addresses joint 3D reconstruction and detection from monocular cameras on a single scene or object. Existing methods have proposed different geometrical priors [38] for this task, including meshes [2], points [27], wire frames [21], voxels [67], CAD models or implicit functions [47, 30] and signed distance functions (SDFs) [78]. These methods focus on reconstructing specific scenes and either require explicit, pre-defined prior geometries, such as CAD models, or multi-view images/videos and geometry supervision from meshes or point clouds. We operate on a *single image* input and propose to learn the underlying prior with a generative scene layout representation embedded in a neural rendering pipeline.

Inverse Rendering. Inverse rendering methods conceptually "invert" the graphics rendering pipeline, which generates images from scene descriptions, and instead estimate scene properties, i.e., geometry, lighting, depth, and poses from input images. Recent works [64, 73, 34] jointly optimize a volumetric model and unknown camera poses with a set of images by back-propagating through a rendering pipeline. Other methods focus on material and lighting to find a physical representation that best models the observed image [45, 19, 46].

Scene and Object Generation. Generative models, in particular generative-adversarial networks (GANs) [17] and diffusion probabilistic models [22, 61], have been extensively explored for image generation [24, 53]. Text conditioning [53], style transfer [81], and image modalities [79] have added controllability to the generation process. In recent years these approaches have been transferred to 3D object [12, 8, 59, 55, 16] and scene generation [15, 14, 69, 43, 42].

We are primarily concerned with multi-object scenes. BlockGAN [42] disentangles each object with a latent identity and pose for scene composition but individual objects cannot be customized. BlobGAN [15] proposes to represent multi-object scenes with each object instance encoded by a feature blob in the projected image space, used as a condition for a GAN-based image generation network. A *separately* trained inversion network aims to translate images into a blob layout. We find that this direct prediction without end-to-end supervision fails to recover moderately complex scene layouts. For 3D scene generation, DisCoScene [69] follows a similar approach by rendering an image from blended features from disentangled per-object radiance fields. This approach does not generate or predict scene layouts and instead relies on scene descriptions extracted from the dataset.

3D Object Detection. Monocular 3D object detection [6, 4, 10, 51, 68, 48, 80, 28, 37] has been explored extensively in computer vision. Existing approaches have investigated purely two-dimensional convolutional architectures utilizing dense depth predictions in image [48] or the levering of inductive biases, such as BEV projections [48], or frustum segmentation [65]. Most existing methods can not be directly trained without a pre-trained depth prediction through depth supervision [48, 48, 37]. A large body of existing work exploits geometric priors. Encoding the scene layouts in the BEV space has been explored in image and LiDAR-based architectures [31, 51, 70], trading off height information for computational efficiency, which we also embrace in our work. Typically, a BEV feature extractor (combined with measured or predicted depth) is followed by a set of convolutional prediction heads that outputs one prediction such as location or bounding box, per head. However, the proposed method directly generates a BEV map of the scene instead through a jointly trained generative diffusion probabilistic model.

3 Diffusion Pillar Representation and Generation

The proposed method generates a scene layout in the 2D ground plane via a diffusion process that enforces this layout to be consistent with a rendered image observation, i.e., performing object detection by inverse rendering. An overview of this approach is shown in Figure 1. We represent a scene layout with a BEV grid **G**, where each cell contains either a background encoding or an object feature encoding and geometry information. Given the ground plane representation of the multi-object scene, novel views are rendered by 3D to 2D feature projections using a second diffusion-based process. Conditioning is optional, allowing the method to perform either 3D object detection, or to generate novel scene layouts and novel views.

The pipeline is end-to-end differentiable, which allows us to refine the ground plane layout when training the novel views. Doing so, we combine generation and detection into one integrated architecture, creating a cycle that is closed by image reconstruction and manipulation.

3.1 Scene Layout Representation

We represent a multi-object scene layout on the BEV ground plane as a prior for the natural distribution, without requiring costly 3D voxel discretization. The scene layout **G**, a $H_{BEV} \times W_{BEV}$

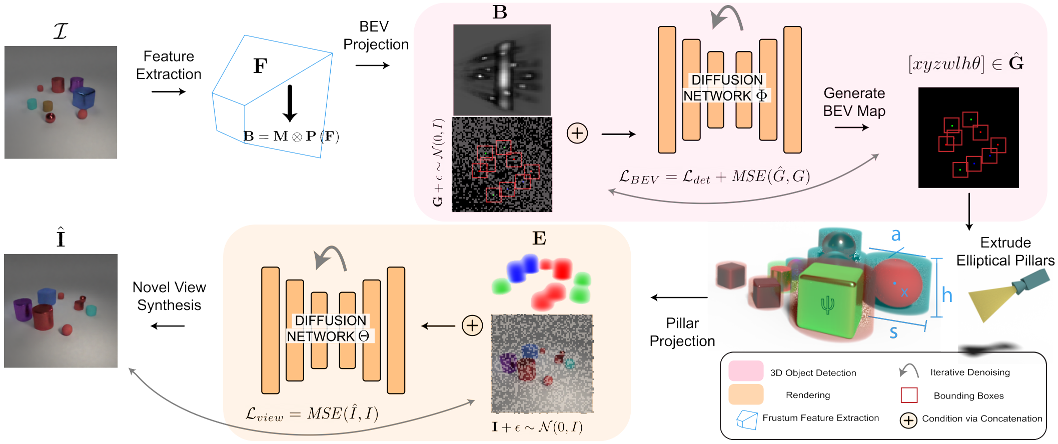

$\mathcal{I}$
Feature Extraction
F
$\mathbf{B} = \mathbf{M} \otimes \mathbf{P}(\mathbf{F})$
BEV Projection
B
$\mathbf{G} + \epsilon \sim \mathcal{N}(0, I)$
DIFFUSION NETWORK Φ
Generate BEV Map
$[xyzwlh\theta] \in \hat{\mathbf{G}}$
$\mathcal{L}_{BEV} = \mathcal{L}_{det} + MSE(\hat{G}, G)$
Extrude Elliptical Pillars
a
h
s
ψ
Pillar Projection
E
$\mathbf{I} + \epsilon \sim \mathcal{N}(0, I)$
DIFFUSION NETWORK Θ
$\mathcal{L}_{view} = MSE(\hat{I}, I)$
Novel View Synthesis
$\hat{\mathbf{I}}$
3D Object Detection
Rendering
Frustum Feature Extraction
Iterative Denoising
Bounding Boxes
Condition via Concatenation

Figure 1: Our end-to-end training pipeline. (**Top row**) Given an image observation, we extract features and create a BEV ground plane. The diffusion model learns to generate a BEV map that contains background and object features. (**Bottom row**) We extrude the BEV map and perform projection to obtain elliptic pillars. The diffusion model learns to generate novel views.

is a grid of anchors $g_{i,j} \in \mathbb{R}^{C_{BEV}}$ with $i \in [0, H_{BEV}], j \in [0, W_{BEV}]$ and the center location $\mathbf{x}_{i,j} = [i + 0.5, j + 0.5]$ of each anchor in the scaled BEV space. Each anchor either represents a single background appearance encoding $\psi_{bckg} \in \mathbb{R}^{f_{bckg}}$ or one of the K objects $\mathbf{o}_k$ in a scene with $k = \{0, ..., K\}$. An object k is defined by an appearance feature $\psi_{fg} \in \mathbb{R}^{f_{fg}}$, a latent appearance encoding that represents the object style and class information, and a geometrical descriptor consisting of the pose with respect to the anchor center and the dimensions $\xi \in \mathbb{R}^7$

$$\xi_k = [\Delta x_k, \Delta y_k, \Delta z_k, w_k, l_k, h_k, \theta_k] \quad \text{and} \quad \mathbf{o}_k = (\mathbf{v}_k, \xi_k, \psi_k) \,. \tag{1}$$

An additional descriptor v at each anchor classifies the anchor either as one of the data-sets N_{cls} classes through one-hot encoding

$$\mathbf{v}_{i,j} = \textit{onehot}\,(cls_{i,j})\,, \text{with } \mathbf{v} \in \{0,1\}_{N_{cls}}, \tag{2}$$

or if zero for all entries, the respective anchor (i, j) is treated as background. The combined number of channels per anchor is $C_{BEV} = N_{cls} + 7 + f_{fg}$.

This explicit ground plane representation of objects effectively disentangles objects from the background. It allows for an arbitrary number of objects in a scene and separate manipulations in complex layouts of multiple objects.

While the BEV lives on a $H_{BEV} \times W_{BEV}$ grid, we can infer the object location in the reference world as

$$\mathbf{x}^{ref}_{i,j} = (\mathbf{x}_{i,j} + \mathbf{\Delta x}_{i,j}) \begin{bmatrix} \frac{d_{x,max} - d_{x,min}}{W_{BEV}} \\ \frac{d_{y,max} - d_{y,min}}{H_{BEV}} \\ 1 \end{bmatrix} + \mathbf{d}_{min}, \tag{3}$$

where $\mathbf{d}_{min}$ and $\mathbf{d}_{max}$ are the BEV bounds, and object dimension and rotation θ are in the unscaled reference space.

Elliptic Pillars. While we rely on the bounding box and location ξ_k as a high-level description for manipulation and 3D detection, a fixed 3D bounding box only defines the bounds of an object, and direct projection into an image plane overestimates the object extend. We, therefore, introduce an elliptic pillar representation with fading density η_k to transform ξ_k from the 2D ground plane to a 3D space. We adopt an elliptical mid-level object representation with a spatial falloff on the 2D BEV plane similar to BlobGAN [15] and introduce an extension into a height map. Each elliptical feature pillar for a given object is defined by $\eta_k = \left(\mathbf{x}^{ref}_k, s_k, a_k, \mathbf{R}_k, \psi_k, h_k\right)$ and can fully be inferred from the predicted objects of the layout generator. The ellipses on the ground plane are given by their

$$\text{center } \mathbf{x}_k^{ref} \in \mathbb{R}^3, \text{ rotation } \mathbf{R}\left(\theta_k\right) \in \mathbb{R}^2, \text{ scale } s_k = w_k\sqrt{\frac{l_k}{w_k}}, \text{ and aspect ratio } a_k = \sqrt{\frac{l_k}{w_k}}. \quad (4)$$

3.2 Probabilistic Layout Generation

To generate scene layouts, we design a diffusion probabilistic model based on [22]. Instead of generating images, our diffusion model Φ_ω learns to generate BEV maps $\mathbf{G}$. Given an input scene, we first obtain its corresponding BEV map. Then, we sample a timestep $t \in [1, 1000]$ that corresponds to some Gaussian noise level, which is added to the BEV map (forward step). The diffusion model learns to denoise the BEV map (reverse step) and we perform ancestral sampling during generation. We provide a complete formulation in the supplement.

We optionally concatenate a condition $\mathbf{B}$ to x_t. This is passed as input to our diffusion model, represented by a UNet [54]. Each block includes self-attention [63] to learn correspondences between the BEV map and its condition. The final output is the generated BEV map.

Conditional Generation as Object Detection. The probabilistic diffusion model not only allows for the generation of unconditional samples but also conditional generation, where generation is guided by $\mathbf{B}$. We condition the diffusion process on image features that are mapped from the image space onto the BEV map to align with the generated layout introducing an inductive model bias through the projection model. Here we follow a similar architecture for the BEV feature extractor from CaDDN [51] and DSGN [11], a monocular object detection and stereo detection approaches, which we formalize as

$$\mathbf{F} = \mathbf{J}\left(\mathcal{I}\right) \times \mathbf{D}\left(\mathcal{I}\right), \text{with } \mathbf{J} \in \mathbb{R}^{W_F \times H_F \times C}, \mathbf{D} \in \mathbb{R}^{W_F \times H_F \times D}. \quad (5)$$

Here, we project image plane information into 3D space; specifically, the image features $\mathbf{J}$ from a pre-trained ResNet18 [20] are projected into D discrete bins per pixel along the depth axis. Weighing each image feature $\mathbf{j}_{u,v}$ from the same image pixel with a predicted depth distribution $p(d)_{u,v}$, a bounded frustum feature grid $\mathbf{F} \in \mathbb{R}^{W_F \times H_F \times C \times D}$ is created where each bins feature is $\mathbf{f}_{u,v,d} = p(d)_{u,v}\mathbf{j}_{u,v}$. The predicted discrete depth distribution $p(d)_{u,v} = \mathbf{D}\left(u, v, d\right)$ for each pixel u, v follows the architecture of DeepLabV3 [9] and is not supervised, but jointly trained with the generative pipeline, which is in contrast to most monocular 3D detection methods that utilize sparse depth supervision from 3D point clouds. From the feature frustum, we can infer the BEV features

$$\mathbf{B} = \mathbf{M} \otimes \mathbf{P}\left(\mathbf{F}\right), \text{with } \mathbf{M} \in \mathbb{R}^{1 \times 1 \times}, \mathbf{B} \in \mathbb{R}^{H_{BEV} \times W_{BEV} \times Z * C}. \quad (6)$$

The feature frustum $\mathbf{F}$ is transformed back into the reference coordinate frame ref with the known camera calibration $\mathbf{K}$, pose $\mathbf{R}$, compressed into the BEV boundaries $[\mathbf{d}_{min}, \mathbf{d}_{max}]$, and finally down projected with $\mathbf{P}\left(\mathbf{F}\right)$. In the last step, the stacked features at each anchor location are accumulated and reduced into the BEV feature with $\mathbf{M}$.

The probability of a BEV anchor belonging to each possible object class is predicted with $\hat{\mathbf{v}}_{i,j} = \sigma\left(\mathbf{b}_{i,j}\right)$ at the first N_{cls} output values in the feature dimension. All object anchors with the predicted probability above the threshold τ_+ defined for the respective class, are positive object anchors. The remaining anchors are assumed to be part of the background. From the set of likely object anchors only K unique object anchors are further selected performing non-maximum suppression (NMS) on all 2D bounding boxes $\left[x_{i,j}^{ref}, y_{i,j}^{ref}, w_{i,j}, l_{i,j}\right] \forall (i, j) \in \arg\max_{cls}\left(\hat{\mathbf{v}}_{i,j}\left(cls\right)\right) > \tau_+$ on the ground plane with an IoU threshold τ_{NMS}.

3.3 Neural Layout Rendering

With the ground plane layout in hand, we render an RGB image $\mathbf{I} \in \mathbb{R}^{H_I, W_I, 3}$ of a scene through geometric projection and conditional image generation. The predicted anchors ξ_k and the respective appearance features ψ_k form feature pillar representation η_k with fading density to transform. Here, η_k is aligned with the 2D ground plane but defines the 3D space. We project and blend the representation of multiple objects in the image plane used to condition the generation of a novel view using diffusion.

Image Projection. Given a camera projection matrix $\mathbf{P}_c = \mathbf{K}_c\mathbf{R}_c$ we sample 3D rays $\mathbf{r}_{u,v}\left(t\right) = \mathbf{o}_{u,v} + \mathbf{d}_{u,v}t$ at each image pixel location (u, v). The rays are then projected into the BEV plane, where we analytically compute the closest point of each ray and ellipses. First each ray $r_{u,v}$ is transformed with respect to the location and rotated and scaled axis of the ellipses with

$$\mathbf{o}^{k,2D}_{u,v} = \begin{bmatrix} a & 0 \\ 0 & \frac{1}{a} \end{bmatrix} \mathbf{R}\left(\theta\right) \left(\mathbf{o}^{ref,2D}_{u,v} - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) \text{ and } \mathbf{d}^{k,2D}_{u,v} = \begin{bmatrix} a & 0 \\ 0 & \frac{1}{a} \end{bmatrix} \mathbf{R}\left(\theta\right) \mathbf{d}^{ref,2D}_{u,v}. \tag{7}$$

For each combination of ray $\mathbf{r}^{k,2D}_{u,v}$ and objects, the closest point $\mathbf{x}^{ref,2D}_{u,v,k}$ on the ray can be found as the perpendicular projection of the center of the ellipse onto the ray in the respective transformed form. All rays u, v where the third dimension of the shortest distance to the center of the ellipse is located below the BEV plane are clipped at the BEV plane and we use their BEV intersection point instead. While BlobGAN directly uses two-dimensional blobs and opacity, respectively density, based on the Mahalanobis distance for each pixel, our representation is based in 3D. We first follow the calculation for the density of an object with respect to x, y, notably representing the true 2D coordinates on the BEV plane and not pixel locations, as

$$b_{BEV}(\mathbf{x}^{ref,2D}_{u,v,k}) = \sigma\left(s_k - d\left(\mathbf{x}^{ref,2D}_{u,v,k}, x_k\right)\right), \text{ where} \tag{8}$$

$$\begin{aligned} d\left(\mathbf{x}^{ref,2D}_{u,v,k}, x_k\right) = \left(\mathbf{x}^{ref,2D}_{u,v,k} - x_k\right)^T \left(\mathbf{R}\left(\theta\right)\Sigma_k\mathbf{R}\left(\theta\right)^T\right)\left(\mathbf{x}^{ref,2D}_{u,v,k} - x_k\right), \\ \text{with } \Sigma_k = c_{BEV}\begin{bmatrix} a_k & 0 \\ 0 & \frac{1}{a_k} \end{bmatrix} \end{aligned} \tag{9}$$

is the Mahalanobis distance between the closest point and the center of the ellipse. The ellipses are extruded into elliptical pillars from which we compute a second sample from an objects density distribution for the intersection of each ray in the third dimension as

$$b_h(\mathbf{x}^{ref,2D}_{u,v,k}) = \sigma\left(c_h\left(z_k + h_k - z_{u,v,k}\right)\right), \tag{10}$$

assuming that the density of the pillar is consistent along the height except for the top, where it fades out. The combined density is their product $b_{3D} = b_h b_{BEV}$.

For all pixels, we perform alpha blending [50] along the corresponding ray with b_{3D} and ψ_k and ψ_{bckg}, the mean from all background anchors, at the last position, which computes the full pixel feature $e_{u,v} \in \mathbb{R}^{f_{fg}}$ and the feature pillar image $\mathbf{E} \in \mathbb{R}^{H_\mathcal{I}\times W_\mathcal{I}\times f_{fg}}$.

Diffusion-Based Rendering. For simplicity, we use the same training scheme and architecture as in Sec 3.2. The diffusion model Θ_κ takes input $x = \mathbf{I}$ and condition E to generate novel views.

3.4 Training and Inference

The proposed joint model for 3D detection and scene reconstruction is trained in an end-to-end fashion. Diffusion probabilistic models are typically trained by iteratively denoising an input. While this probabilistic approach can result in diverse outputs we want to guide our model towards accurate predictions on the BEV plane given a condition. To this end, we propose a loss function combining BEV reconstruction and accurate 3D predictions on the extracted object locations alongside a view reconstruction loss.

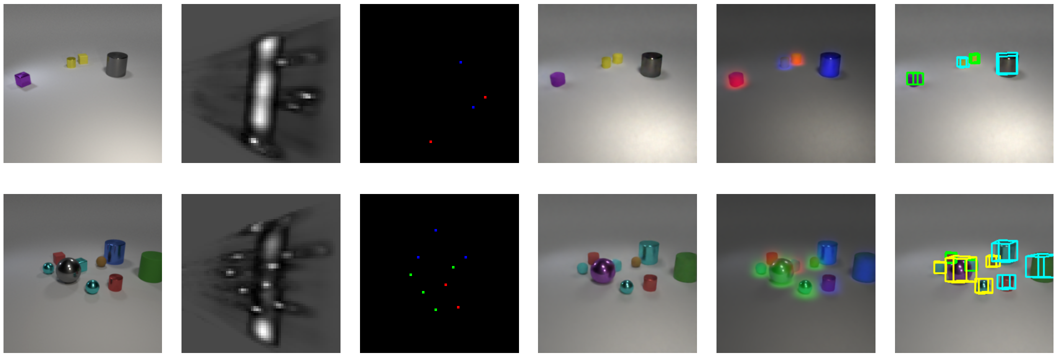
The loss on the BEV map $\hat{\mathbf{B}}$ generation is formulated with the ground-truth target anchors $\hat{\xi}$. We use a ℓ_2 loss function $\mathcal{L}_2(\hat{\mathbf{B}}, \mathbf{B})$. From $\hat{\mathbf{B}}$ we only select N_{det} anchors $g_{i,j}\setminus\psi_{i,j}$ with the highest probability $\max\left(\mathbf{v}_i, j\right)$ for one class. From all $\hat{\xi}_n \forall n \in [0, N_{det}]$ we calculate the intersection-over-union (IoU) with ξ and select all $N_{IoU} = N_{+IoU} + N_{-IoU}$ anchor proposals with an IoU above the thresholds $\tau_{+IoU} = 0.45$ and $\tau_{-IoU} = 0.25$ respectively. We supervise the predicted $\hat{\xi}_m \forall m \in [0, N_{IoU}]$ using a standard 3D detection loss function known from SECOND [70], which combines a regression bounding box loss L_{bbox}, a direction classification loss L_{dir} and the object class loss L_{cls}, resulting in the detection loss

$$\mathcal{L}_{det} = \lambda_{reg}\mathcal{L}_{bbox} + \lambda_{dir}\mathcal{L}_{dir} + \lambda_{fg}\mathcal{L}_{cls} \tag{11}$$

For L_{cls}, we choose the focal loss [35] to imbalances in the amount of background and foreground anchors. In this way, our model employs an "analysis-by-synthesis" paradigm, using the generation of scene layouts to detect objects in existing scenes.

The generated image $\hat{I}$ is solely supervised by the known image applying the mean-squared error to all rendered pixels. There is no direct supervision of the generation of the appearance feature ψ for each rendered BEV anchor, this is implicitly learned because we can update the weights ω of the BEV diffusion process Φ with respect to the gradients $dMSE(\hat{I}, I)/d\psi$ and $d\psi/d\omega$ through the fully differentiable pipeline.

$$\mathcal{L}_{sup} = \mathcal{L}_{det} + \lambda_{img} MSE\left(\hat{I}, I\right) \tag{12}$$



(a) Input images

(b) BEV features condition

(c) Reconstructed BEV map

(d) Reconstructed image

(e) Pillars overlaid on image

(f) 3D Bounding Box overlaid

Figure 2: Visualization of the outputs of each step of our scene reconstruction and detection pipeline.

4 Evaluation

We assess the proposed method on monocular 3D object detection, view synthesis, and scene manipulation tasks separately on scenes generated with the public code from the CLEVR dataset [23] with 3 to 10 objects from the three different classes. While this dataset does not include in-the-wild scenes, it perfectly highlights the pros and cons of repurposed multi-object scene reconstruction and generative methods on perceptual layout reconstruction tasks. Moreover, given the training ressources available to us, this dataset allowed for the investigations discussed in the following. We modified the procedural Blender scene generation to export camera calibrations for detection. All detection and generative methods are trained on the same set of posed images from 50k scenes and evaluated on another set of 5k scenes. Camera and light source positions are randomly perturbed.

4.1 Layout Reconstruction.

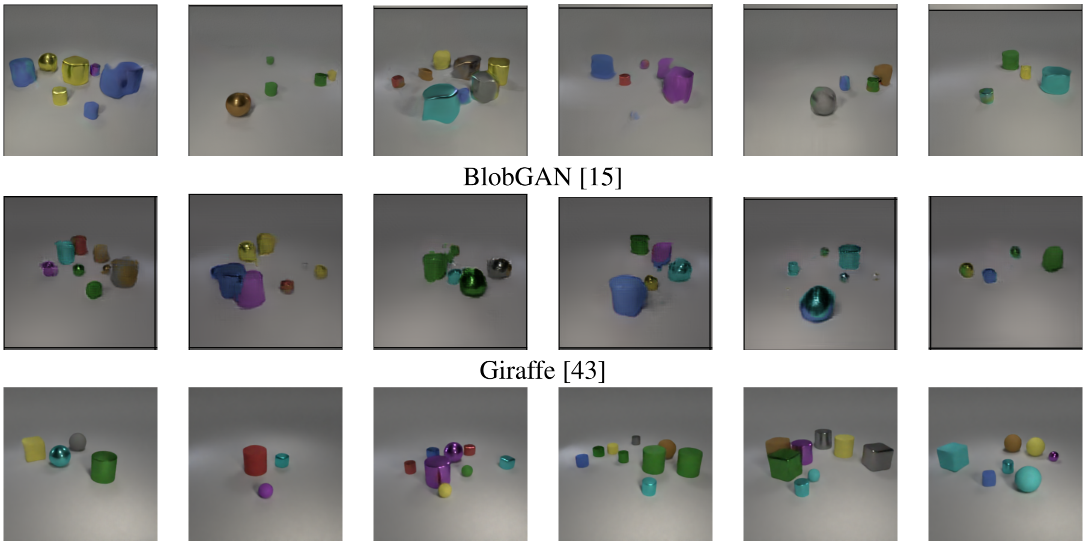
The proposed approach is able to reconstruct layouts from different numbers of objects in an image, which includes occluded objects. In Fig. 2 we show a reconstructions from a variety of test scenes from 3 to 10 objects and the generated underlying BEV map. Each of the three colors (RGB) for the object centers matches one of the object classes. Additionally, we visualize the output of the BEV feature extractor, which is jointly trained with the object detector without leveraging any form of pre-trained or supervised dense depth prediction. The spatial alignment of high activation with the reconstructed object locations in the BEV indicates, that a BEV feature extractor can be learned by backpropagating through the conditioning of the diffusion probabilistic generative process. Additionally, we overlay feature pillars projected onto the image plane and the 3D bounding boxes.

3D Object detection - Conditional Generation. Next, we evaluate DiffusionPillars on an object detection task on the CLEVR dataset. We also compare our method to two state-of-the-art monocular 3D object detection methods, DD3D [48] and DEVIANT [29]. DD3D [48] is a single-stage 3D object detector that leverages pseudo-lidar methods such as monocular depth estimation [75]. DEVIANT [29] uses a depth equivariant network [66] which enforces consistency to depth translations, to learn depth estimates before detection.

Table 1: Monocular 3D Object Detection on CLEVR. **Bold** and underline denote the best and second-best result for each metric. Our method performs on par with state-of-the art monocular object detection baselines.

Method	$AP_{BEV\|R40\|IoU\geq0.7}\uparrow$	$AP_{3D\|R40\|IoU\geq0.7}\uparrow$
DD3D	**98.20**	<u>83.23</u>
DEVIANT	94.86	**86.08**
DiffusionPillars (ours)	<u>96.41</u>	80.57

In Tab. 1 we report monocular 3D object detection results on the multi-object CLEVR dataset. DiffusionPillars' generative approach is comparable to state-of-the art methods, measured by 3D and BEV AP, although not following conventional feed-forward prediction approaches.



BlobGAN [15]

Giraffe [43]

DiffusionPillars

Figure 3: We compare the rendering quality on multi-object scenes with varying amount of objects from CLEVR [23] with GIRAFFE [43] and BlobGAN [15].

4.2 Scene Manipulation and Reconstruction with Generative Models.

For the task of scene generation and reconstruction, we compare against GIRAFFE [43] for manipulating scenes through object addition, removal, translation, and camera movement and show a qualitative comparison with the 2D layout-based method BlobGAN [15] on the task of scene layout reconstruction. We train GIRAFFE, which was only trained on less complex CLEVR [23] scenes with up to 6 objects, on our data and provide additional camera information. During training, we experimented with different hyperparameter settings for the size and amount of blobs to better accommodate the sparse scene structure and image space rendering. Note that the used dataset is different from the dense room datasets presented in [15] and has some impact on the output quality.

View Reconstruction Fig. 3 presents rendered scenes from all three methods. GIRAFFE and BlobGAN apply a GAN-based up-sampling and reconstruction pipeline on a feature image, either from a discrete layout or a feature map from volumetric rendering of multiple feature fields. Both methods tend to blur object features from occluding and close objects. The proposed pillar projection from a sparse 2D BEV map followed by a diffusion-based rendering step in contrast is only adding a local instead of larger splatted features or volumes to the generation process, which results in sharp corners and only minor blurring of object.

Using BlobGAN inversion, we show reconstructions of a possible scene layout and compare with our BEV reconstruction in Fig. 4. We explicitly changed the number of objects to the input images'

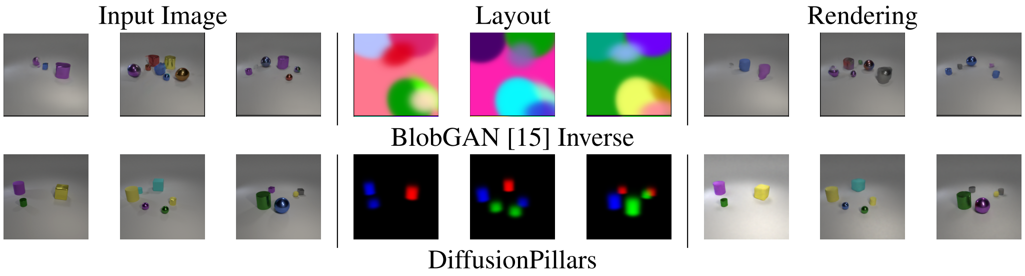


Figure 4: **Layout Reconstruction and Rendering.** The Layout is reconstructed from the input image and a re-rendered on the right side. In our case colors are explicitly matched to the corresponding class of cube, sphere or cylinder.

object amount, but were only able to reconstruct less explainable layouts on those sparse multi-object scenes. Leveraging DiffusionPillars explicit underlying 3D detection, layouts are reasonable in sparse and dense multi-object scenes.

4.3 Manipulation - Controlled View Synthesis.

We compare manipulation capabilities between GIRAFFE [43] and our method via removal and translation of objects in Fig. 5. BlobGAN [15] presents manipulation capabilities, but they are not accessible. For GIRAFFE, blended objects already present in the reconstruction become more dominant when adding more objects to the scene. Our method presents more natural manipulations. These edits, including object transformation, deletion or insertion, and camera transformation, allows our method to reconstruct scene layouts as well as generate novel views and scenes based on the underlying layout from an input image.

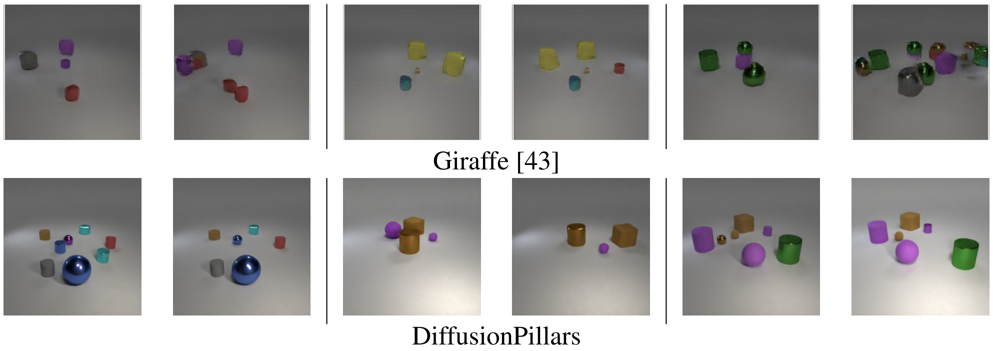


Figure 5: **Manipulation of objects via removal and translation.** Explicitly modeling object locations and instances allows us, similar to Giraffe, to manipulate objects through removal or translation.

5 Conclusion

We investigate joining scene perception and generation by re-framing object detection as a conditional generative process with a learned prior. The method learns inverse rendering and view synthesis in an end-to-end fashion. We evaluate the approach in supervised, self-supervised, and unconditional generative settings. We find the approach performs favorably against feed-forward detectors, and excels in scene understanding and interpretability. We are optimistic that future work can further improve results. Next steps include testing in-the-wild datasets and extending our self-supervised layout generation to alleviate the need for annotations for non-synthetic data.

Broader Impact Our work joins 3D perception and generation, two extensively researched fields. DiffusionPillars fosters mutual learning that could enhance each individual task in future research. The experiments are conducted on the CLEVR dataset and do not pose immediate ethical concerns.

However, as DiffusionPillars includes generative and detection methods, we are aware of potential malicious applications such as deepfakes and surveillance, respectively. Thus, caution should be applied when dealing with sensitive applications. We do not recommend using DiffusionPillars in cases where privacy or erroneous recognition could be an issue. Instead, we encourage practitioners to carefully evaluate the setting under which this method will be applied before proceeding.

Limitations We have only validated DiffusionPillars on CLEVR, a synthetic dataset with limited classes of objects. Sampling time is slow due to iterative denoising of diffusion models. This may be further improved with techniques such as DDIM [60] sampling. Finally, due to probabilistic behavior, object color can be incorrect during rendering as it is not explicitly disentangled.

References

sorting

[1] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. “A survey on 3d object detection methods for autonomous driving applications”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3782–3795.

[2] Deniz Beker, Hiroharu Kato, Mihai Adrian Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. “Monocular differentiable rendering for self-supervised 3d object detection”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 514–529.

[3] Aude Billard and Danica Kragic. “Trends and challenges in robot manipulation”. In: *Science* 364.6446 (2019), eaat8414.

[4] Garrick Brazil and Xiaoming Liu. “M3d-rpn: Monocular 3d region proposal network for object detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9287–9296.

[5] M Lo Brutto and Paola Meli. “Computer vision tools for 3D modelling in archaeology”. In: *International Journal of Heritage in the Digital Era* 1.1_suppl (2012), pp. 1–6.

[6] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. “Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 10478–10485.

[7] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. “The ycb object and model set: Towards common benchmarks for manipulation research”. In: *2015 international conference on advanced robotics (ICAR)*. IEEE. 2015, pp. 510–517.

[8] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. “Efficient Geometry-aware 3D Generative Adversarial Networks”. In: *arXiv*. 2021.

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. “Rethinking Atrous Convolution for Semantic Image Segmentation”. In: *CoRR* abs/1706.05587 (2017). arXiv: 1706.05587. URL: http://arxiv.org/abs/1706.05587.

[10] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. “Monocular 3d object detection for autonomous driving”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2147–2156.

[11] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. “Dsgn: Deep stereo geometry network for 3d object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12536–12545.

[12] Gene Chou, Yuval Bahat, and Felix Heide. “Diffusion-SDF: Conditional Generative Modeling of Signed Distance Functions”. In: *arXiv preprint arXiv:2211.13757* (2022).

[13] Alvaro Collet, Dmitry Berenson, Siddhartha S Srinivasa, and Dave Ferguson. “Object recognition and full pose registration from a single image for robotic manipulation”. In: *2009 IEEE International Conference on Robotics and Automation*. IEEE. 2009, pp. 48–55.

[14] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. “Unconstrained Scene Generation with Locally Conditioned Radiance Fields”. In: *arXiv preprint arXiv:2104.00670* (2021).

[15] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A. Efros. *BlobGAN: Spatially Disentangled Scene Representations*. 2022. arXiv: 2205.02837 [cs.CV].

[16] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. "GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images". In: *Advances In Neural Information Processing Systems*. 2022.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[18] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. "Object-centric neural scene rendering". In: *arXiv preprint arXiv:2012.08503* (2020).

[19] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. "Nerfren: Neural radiance fields with reflections". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18409–18418.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[21] Tong He and Stefano Soatto. "Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8409–8416.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6840–6851.

[23] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2901–2910.

[24] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Training Generative Adversarial Networks with Limited Data". In: *Proc. NeurIPS*. 2020.

[25] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. "A survey on learning-based robotic grasping". In: *Current Robotics Reports* 1.4 (2020), pp. 239–249.

[26] Alina Kloss, Maria Bauza, Jiajun Wu, Joshua B Tenenbaum, Alberto Rodriguez, and Jeannette Bohg. "Accurate vision-based manipulation through contact reasoning". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 6738–6744.

[27] Jason Ku, Alex D Pon, and Steven L Waslander. "Monocular 3d object detection leveraging accurate proposals and shape reconstruction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 11867–11876.

[28] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. "DEVIANT: Depth EquiVarIAnt NeTwork for Monocular 3D Object Detection". In: *ECCV*. 2022.

[29] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. "DEVIANT: Depth EquiVarIAnt NeTwork for Monocular 3D Object Detection". In: *In Proceeding of European Conference on Computer Vision*. Tel-Aviv, Israel, Oct. 2022.

[30] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. "Panoptic neural fields: A semantic object-aware neural scene representation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12871–12881.

[31] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. "Pointpillars: Fast encoders for object detection from point clouds". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 12697–12705.

[32] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel L.K. Yamins, Jiajun Wu, Joshua B. Tenenbaum, and Antonio Torralba. "Visual Grounding of Learned Physical Models". In: *ICML*. 2020.

[33] Liang Liang, Fanwei Kong, Caitlin Martin, Thuy Pham, Qian Wang, James Duncan, and Wei Sun. "Machine learning–based 3-D geometry reconstruction and modeling of aortic valve deformation using 3-D computed tomography images". In: *International journal for numerical methods in biomedical engineering* 33.5 (2017), e2827.

[34] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. “Barf: Bundle-adjusting neural radiance fields”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5741–5751.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[36] Luyang Liu, Hongyu Li, and Marco Gruteser. “Edge assisted real-time object detection for mobile augmented reality”. In: *The 25th annual international conference on mobile computing and networking*. 2019, pp. 1–16.

[37] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. “Rethinking pseudo-lidar representation”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer. 2020, pp. 311–327.

[38] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. “3d object detection for autonomous driving: A review and new outlooks”. In: *arXiv preprint arXiv:2206.09474* (2022).

[39] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. “NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections”. In: *arXiv*. 2020.

[40] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. arXiv: `2003.08934 [cs.CV]`.

[41] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. “Instant neural graphics primitives with a multiresolution hash encoding”. In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15.

[42] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. “BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images”. In: *Advances in Neural Information Processing Systems 33*. Nov. 2020.

[43] Michael Niemeyer and Andreas Geiger. “GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[44] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. “Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[45] Merlin Nimier-David, Zhao Dong, Wenzel Jakob, and Anton Kaplanyan. “Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering”. In: *Eurographics Symposium on Rendering - DL-only Track*. Ed. by Adrien Bousseau and Morgan McGuire. The Eurographics Association, 2021. ISBN: 978-3-03868-157-1. DOI: `10.2312/sr.20211292`.

[46] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. “Mitsuba 2: A Retargetable Forward and Inverse Renderer”. In: *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38.6 (Dec. 2019). DOI: `10.1145/3355089.3356498`.

[47] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. “Neural scene graphs for dynamic scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2856–2865.

[48] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. “Is Pseudo-Lidar needed for Monocular 3D Object detection?” In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[49] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. “Nerfies: Deformable Neural Radiance Fields”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2021).

[50] Thomas Porter and Tom Duff. “Compositing digital images”. In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. 1984, pp. 253–259.

[51] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. “Categorical depth distribution network for monocular 3d object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8555–8564.

[52] Jun Rekimoto. “Matrix: A realtime object identification and registration method for augmented reality”. In: *Proceedings. 3rd Asia Pacific Computer Human Interaction (Cat. No. 98EX110)*. IEEE. 1998, pp. 63–68.

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: `2112.10752 [cs.CV]`.

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[55] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. “GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[56] Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues. “Towards machine-learning assisted asset generation for games: a study on pixel art sprite sheets”. In: *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. IEEE. 2019, pp. 182–191.

[57] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Ambrus, Adrien Gaidon, William T Freeman, Fredo Durand, Joshua B Tenenbaum, and Vincent Sitzmann. “Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement”. In: *arXiv preprint arXiv:2207.11232* (2022).

[58] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. “Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations”. In: *Advances in Neural Information Processing Systems*. 2019.

[59] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. “EpiGRAF: Rethinking training of 3D GANs”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022. URL: `https://openreview.net/forum?id=TTM7iEFOTzJ`.

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=St1giarCHLP`.

[61] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[62] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. “Scalability in perception for autonomous driving: Waymo open dataset”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2446–2454.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[64] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. “NeRF–: Neural radiance fields without known camera parameters”. In: *arXiv preprint arXiv:2102.07064* (2021).

[65] Xinshuo Weng and Kris Kitani. “Monocular 3d object detection with pseudo-lidar point cloud”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[66] Daniel E. Worrall and Gabriel J. Brostow. “CubeNet: Equivariance to 3D Rotation and Translation”. In: *CoRR* abs/1804.04458 (2018). arXiv: `1804.04458`. URL: `http://arxiv.org/abs/1804.04458`.

[67] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. “Data-driven 3d voxel patterns for object category recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1903–1911.

[68] Tianfu Wu Xianpeng Liu Nan Xue. “Learning Auxiliary Monocular Contexts Helps Monocular 3D Object Detection”. In: *36th AAAI Conference on Artifical Intelligence (AAAI)*. Feb. 2022.

[69] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Skorokhodov Ivan, Siarohin Aliaksandr, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, and Tulyakov Sergy. “DiscoScene: Spatially Disentangled Generative Radiance Field for Controllable 3D-aware Scene Synthesis”. In: *arxiv: 2212.11984* (2022).

[70] Yan Yan, Yuxing Mao, and Bo Li. “Second: Sparsely embedded convolutional detection”. In: *Sensors* 18.10 (2018), p. 3337.

[71] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. “Learning object-compositional neural radiance field for editable scene rendering”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 13779–13788.

[72] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. “Multiview neural surface reconstruction by disentangling geometry and appearance”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2492–2502.

[73] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. “iNeRF: Inverting neural radiance fields for pose estimation”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1323–1330.

[74] Ilker Yildirim, Mario Belledonne, Winrich Freiwald, and Josh Tenenbaum. “Efficient inverse graphics in biological face processing”. In: *Science advances* 6.10 (2020), eaax5979.

[75] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. “Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving”. In: *CoRR* abs/1906.06310 (2019). arXiv: `1906.06310`. URL: `http://arxiv.org/abs/1906.06310`.

[76] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. “Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13144–13152.

[77] Alan Yuille and Daniel Kersten. “Vision as Bayesian inference: analysis by synthesis?” In: *Trends in cognitive sciences* 10.7 (2006), pp. 301–308.

[78] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. “Autolabeling 3d objects with differentiable rendering of sdf shape priors”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12224–12233.

[79] Lvmin Zhang and Maneesh Agrawala. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023. arXiv: `2302.05543 [cs.CV]`.

[80] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. “Objects Are Different: Flexible Monocular 3D Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 3289–3298.

[81] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.