UnO: Unsupervised Occupancy Fields for Perception and Forecasting

Ben Agro[*], Quinlan Sykora[*], Sergio Casas[*], Thomas Gilles, Raquel Urtasun
Waabi, University of Toronto
{bagro, qsykora, sergio, tgilles, urtasun}@waabi.ai

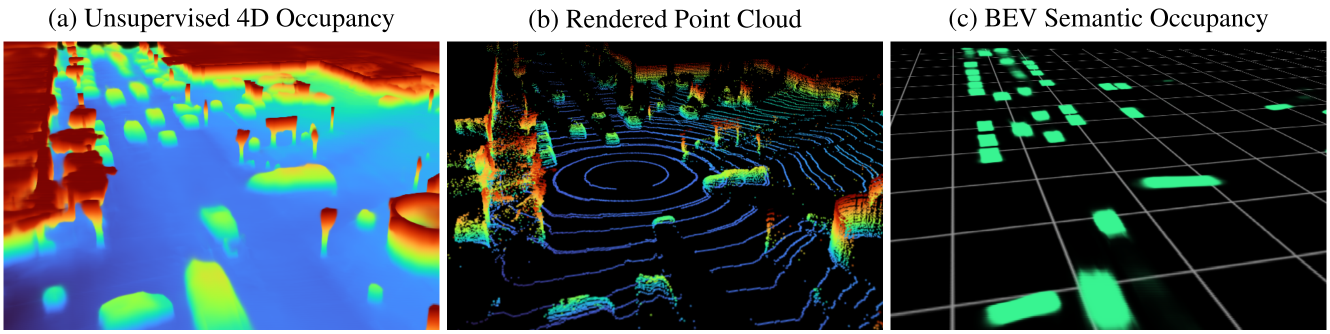
(a) Unsupervised 4D Occupancy
(b) Rendered Point Cloud
(c) BEV Semantic Occupancy

Figure 1. We present UNO, a world model that learns to predict 3D occupancy (a) over time from unlabeled data. This model can be easily and effectively transferred to downstream tasks like point cloud forecasting (b), and bird's-eye view semantic occupancy (c).

Abstract

Perceiving the world and forecasting its future state is a critical task for self-driving. Supervised approaches leverage annotated object labels to learn a model of the world — traditionally with object detections and trajectory predictions, or temporal bird's-eye-view (BEV) occupancy fields. However, these annotations are expensive and typically limited to a set of predefined categories that do not cover everything we might encounter on the road. Instead, we learn to perceive and forecast a continuous 4D (spatio-temporal) occupancy field with self-supervision from LiDAR data. This unsupervised world model can be easily and effectively transferred to downstream tasks. We tackle point cloud forecasting by adding a lightweight learned renderer and achieve state-of-the-art performance in Argoverse 2, nuScenes, and KITTI. To further showcase its transferability, we fine-tune our model for BEV semantic occupancy forecasting and show that it outperforms the fully supervised state-of-the-art, especially when labeled data is scarce. Finally, when compared to prior state-of-the-art on spatio-temporal geometric occupancy prediction, our 4D world model achieves a much higher recall of objects from classes relevant to self-driving.

1. Introduction

For a self-driving vehicle (SDV) to plan its actions effectively and safely, it must be able to perceive the environment and forecast how it will evolve in the future. Two paradigms have been developed in order to perform these two tasks. The most common approach is to detect a discrete set of objects in the scene, then forecast possible future trajectories of each object [5, 7, 10, 18, 19, 22, 30, 32, 34]. More recently, bird's-eye view (BEV) *semantic occupancy fields* [1, 6, 14, 23, 26, 29] have become popular as they avoid thresholding confidence scores and better represent uncertainty about future motion.

These approaches leverage supervision from human annotations to learn a model of the world. Unfortunately, their performance is bounded by the scale and expressiveness of the human annotations. Due to the high cost of these labels, the amount of available labeled data is orders of magnitude smaller than the amount of unlabeled data. Furthermore, these labels are typically restricted to a predefined set of object classes and the object shape is approximated with a 3D bounding box, which for many classes is a very crude approximation. Thus, rare events and infrequent objects are seldom included in labeled data, limiting the safety of current self-driving systems.

This motivates the development of methods that can leverage vast amounts of unlabeled sensor data to learn representations of the world. Prior works proposed to directly predict future point clouds from past point clouds [24, 31, 33, 35]. However, this makes the task unnecessarily difficult, as the model must learn not only a model of the world, but also the sensor extrinsics and intrinsics as well as LiDAR properties such as ray reflectivity, which is a complex function of materials and incidence angle. To address this issue, 4D-Occ [21] proposed to learn future *geometric occupancy voxel grids* exploiting the known sensor intrinsics and extrinsics. However, this method is limited by the use of a quantized voxel grid and a LiDAR depth-rendering objective that optimizes for optical density via regression. As shown in our experiments, this results in models that struggle learning the dynamics of the world. Furthermore, whether the learned representations are useful for downstream tasks other than point cloud forecasting remains unknown.

Our goal is to learn a model of the world that can exploit large-scale unlabeled LiDAR data and can be easily and effectively transferred to perform downstream perception and forecasting tasks. Towards this goal, we propose a novel unsupervised task: forecasting continuous 4D (3D space and time) occupancy fields (Fig. 1.a) from LiDAR observations. This objective is suitable for learning general representations because accurately predicting spatio-temporal occupancy fields requires an understanding of the world's *geometry* (e.g., to predict shapes of partially occluded objects), *dynamics* (e.g., to predict where moving objects will be in the future) and *semantics* (e.g., to understand the rules of the road). Importantly, we employ an implicit architecture to allow our model to be queried at any given continuous point (x, y, z, t) in space and future time. Our world model, which we dub UnO (UNsupervised Occupancy), learns common sense concepts such as the full extent of objects, even though the input LiDAR only sees a portion of the object. The ability to forecast multi-modal futures with associated uncertainty also emerges; e.g., UnO can predict that a vehicle may or may not lane change, and a pedestrian may stay on the sidewalk or enter the crosswalk.

To demonstrate the generalizability and effectiveness of our world model, we show that it can be transferred to two important downstream tasks: point cloud forecasting (Fig. 1.b) and supervised BEV semantic occupancy prediction (Fig. 1.c). For point cloud forecasting, UnO surpasses the state-of-the-art in Argoverse 2, nuScenes, and KITTI by learning a simple ray depth renderer on top of the occupancy predictions. For BEV semantic occupancy prediction, we show that fine-tuning UnO outperforms fully supervised approaches, and that the improvement is particularly large when limited labels are available for training, demonstrating impressive few-shot generalization.

2. Related Work

Point Cloud Forecasting: Predicting future point clouds from past observations provides a setting to learn world models from large unlabeled datasets. Prior works aim to predict point clouds directly [24, 31, 33, 35], which requires forecasting the sensor extrinsics, including where the ego vehicle will be in the future; the sensor intrinsics, e.g., the sampling pattern specific to the LiDAR sensor; and the shape and motion of objects in the scene. 4D-Occ [21] reformulates the point cloud prediction task to factor out the sensor extrinsics and intrinsics. Specifically, it first predicts occupancy in space and future time as a 4D voxel grid and then predicts the depth of LiDAR rays given by the sensor intrinsics and future extrinsics through a NERF-like rendering of the occupancy. However, 4D-Occ does not attain all the desired capabilities of a world model. For instance, it struggles to forecast moving objects. We believe this is because their regression loss emphasizes accurate depth predictions instead of occupancy classification, and their 4D voxel grids introduce quantization errors. While our work also builds on the idea of using an internal 4D geometric occupancy representation, we improve upon prior art by learning a continuous 4D occupancy field through a classification objective instead of a voxel grid through regression, and by decoupling the training of the occupancy (i.e., world model) and the point cloud renderer (i.e., downstream task).

Occupancy Forecasting from Sensors: This task consists of predicting the probability that a certain area of space will be occupied in the future, directly from sensory data like LiDAR [6, 29] and camera [14, 16, 17, 26]. The occupancy predictions are often *semantic*; they represent classes of objects (e.g., vehicle, cyclist) relevant to downstream tasks like motion planning for SDVs. Because most planners [6–8, 28] for autonomous driving reason in 2D BEV space, prior work has focused on forecasting accurate occupancy in BEV. P3 [29], MP3 [6] and FIERY [14] predict temporal BEV semantic occupancy grids with convolutional neural networks directly from sensor data. UniAD [17] predicts both trajectory predictions and BEV semantic occupancy forecasts. P3, MP3 and UniAD all demonstrate that utilizing BEV semantic occupancy can improve motion planning for self-driving. ImplicitO [1] introduces an attention-based architecture for predicting occupancy at any spatio-temporal continuous point (x, y, t) in BEV, improving efficiency by only producing occupancy at user specified query points. Other works have sought to predict BEV occupancy without semantic labels. For instance, [20] leverages self-supervision from visibility maps generated with future LiDAR point clouds, thereby combining the occupancy of semantic classes and the background, which we refer to as *geometric occupancy*.

Pre-Training for Perception: The ultimate goal of many representation learning methods is to improve performance in downstream vision tasks like 2D/3D object detection and semantic segmentation. Masked AutoEncoders (MAEs) [11] have recently gained traction: mask random image patches and use their pixel values as reconstruction targets. VoxelMAE [13] extends this concept to LiDAR data by voxelizing the point cloud data and demonstrates this is effective as pre-training for object detection in self-driving.

UniPAD [37] builds on these works by learning a NeRF-like world model to reconstruct color and depth data from masked multi-modal image and point cloud inputs. ALSO [3] shows that surface reconstruction from present-time LiDAR rays can be a strong pre-training task. In contrast to these prior works, we focus not only on pre-training for understanding the present-time world state from sensor data, but also to forecast the future.

3. Unsupervised Occupancy World Model

Time-of-flight LiDARs measure the distance of a surface from the sensor, which we refer to as *depth*, by emitting rays of light from a pulsed laser. A LiDAR point occurs when the ray hits an object and the laser returns. The depth is calculated based on the time it took for the ray to come back to the sensor. Thus, a LiDAR point indicates that the space along the ray between the sensor and the observed point is unoccupied, and that there is some occupied space directly after that point in the ray direction.

Although LiDAR point clouds are usually displayed as a 360° scan called a *sweep*, the points are actually acquired at different times. For instance, in mechanical spinning LiDARs, different parts of the scene are scanned as a set of beams emit rays into a polygon mirror while it rotates [27]. Accounting for the right time of emission is particularly important for objects moving at high speeds, and also if the data collection platform is equipped with multiple LiDAR sensors that scan the scene asynchronously, which is the case in many modern SDVs.

We take inspiration from these intuitions and propose an unsupervised task (Sec. 3.1) as well as a simple yet effective model (Sec. 3.2) to learn the world's geometry, dynamics and even semantics from future point clouds.

3.1. Unsupervised Task

We assume that we have access to a calibrated and localized vehicle as our data collection platform. Note that these assumptions are not restrictive as this is the norm for self-driving platforms. This implies that we know where the sensors are located on the vehicle, how the vehicle moves around the scene, and we can capture the observed LiDAR points over time. We can then make use of the known sensor intrinsics and extrinsics to construct occupancy pseudo-labels over space and time providing us with positive (occupied) and negative (unoccupied) supervision for every emitted ray that returned, as shown in Fig. 2. Note that we have partial supervision as there are regions where we do not know the occupancy due to occlusion or scene properties such non-reflective materials.

More formally, let $\mathbf{s}_i(t) = (s_i^x(t), s_i^y(t), s_i^z(t))$ be the 3D position at time t of the i-th LiDAR sensor mounted on the SDV. Let $\mathbf{p}_{ij} = (p_{ij}^x, p_{ij}^y, p_{ij}^z)$ be the j-th LiDAR point returned by sensor i, with emission time denoted t_{ij}. For a single LiDAR point, the line segment between $\mathbf{s}_i(t_{ij})$ and $\mathbf{p}_{ij}$ is unoccupied

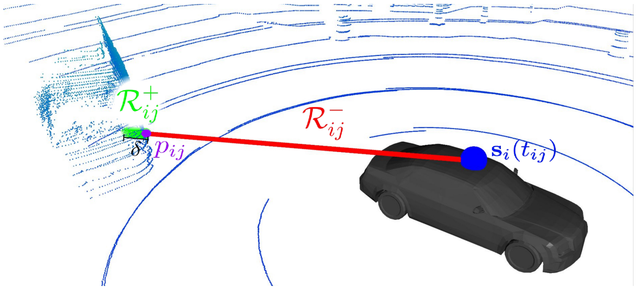

Figure 2. UNO's occupancy pseudo-labels: a laser beam emitted from sensor position s_i at time t_{ij} returns the point p_{ij}, meaning that the ray segment $\mathcal{R}_{ij}^-$ is unoccupied space and the segment within a buffer δ after the lidar return is occupied space $\mathcal{R}_{ij}^+$.

$$\mathcal{R}_{ij}^- = \{\mathbf{s}_i(t_{ij}) + (\mathbf{p}_{ij} - \mathbf{s}_i(t_{ij}))r \mid \forall r \in (0,1)\}. \quad (1)$$

We also produce a positive occupancy directly behind the LiDAR point $\mathbf{p}_{ij}$:

$$\mathcal{R}_{ij}^+ = \{\mathbf{p}_{ij} + \frac{(\mathbf{p}_{ij} - \mathbf{s}_i(t_{ij}))}{||\mathbf{p}_{ij} - \mathbf{s}_i(t_{ij})||_2} r \mid \forall r \in [0,\delta]\}, \quad (2)$$

where δ is a small buffer region, see Fig. 2 (e.g., we use $\delta = 0.1\,\mathrm{m}$ for all of our experiments). Note that δ should be small; if it were too large then the occupied regions would misrepresent the true shape of objects (especially thin structures), and multiple rays hitting the same object could conflict on their assignment of occupied / unoccupied space.

3.2. Unsupervised Occupancy Model (UNO)

UNO models 4D occupancy as a *continuous field*, which has many advantages. First, continuous temporal resolution allows the model to properly handle the continuous nature of the rays' emission time during training, and be queried at any time of interest during inference. Second, a very fine-grained spatial resolution is useful to accurately model complex shapes — especially for thin structures — without introducing quantization errors. Third, a fine spatial resolution allows us to make the best use of the self-supervision for positive occupancy for a small range interval δ, without exceeding memory limits when dealing with very fine quantization in voxel-based methods.

Architecture: Inspired by supervised implicit 3D occupancy [1], we design a *4D implicit occupancy forecasting architecture* that can predict occupancy at any arbitrary continuous query point $\mathbf{q} = (x, y, z, t)$ in 3D space and future time. We refer the reader to Fig. 3.

We first voxelize a stack of H past LiDAR sweeps in BEV [36] to represent the history, and then encode it through a ResNet [12] backbone to generate a 2D BEV feature map, $\mathbf{Z}$. This is preferable over a 3D feature volume

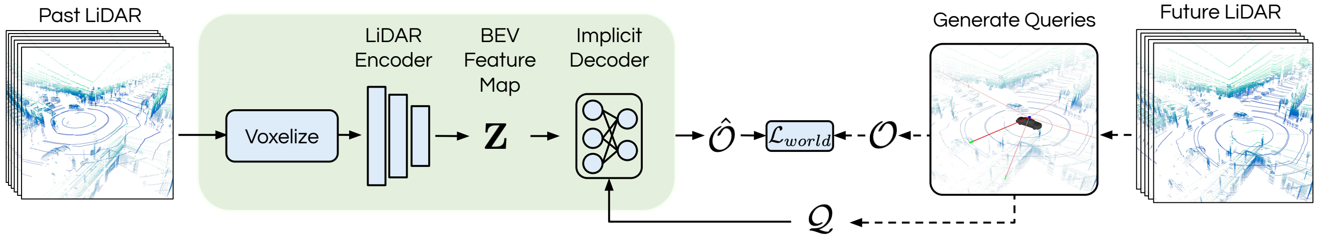

Figure 3. An overview of our method, UNO. The past LiDAR is voxelized and encoded into a BEV feature map which is used by an implicit occupancy decoder to predict occupancy $\hat{\mathcal{O}}$ at query points $\mathcal{Q}$. During training the query points and occupancy pseudo-labels are generated from future LiDAR data. At inference, the model can be queried at any (x, y, z, t) point. Refer to Fig. 2 for details on the query generation process.

since more representation and computing power can be allocated to the difficult task of learning dynamics in BEV, since objects primarily move along the relatively flat ground.

The BEV feature map $\mathbf{Z}$ is then used by an *implicit occupancy decoder*, where each 4D query point $\mathbf{q} = (x, y, z, t)$ is embedded with a positional embedding and outputs an occupancy logit by exploiting deformable attention [38] to relevant parts of the feature map. This implicit decoder is very lightweight, so it can rapidly produce occupancy at many query points in parallel, allowing for practical training and efficiency for downstream tasks that require many query points. More details are provided in the supplementary.

We note that the future sensor extrinsics are not an input to the model: while they are used to accurately supervise the occupancy field as described in Sec. 3.1, it is desirable that the model is uncertain about the raycasting origin. This encourages the model to learn the extent of objects such that when viewed from any perspective, the future occupancy can explain the future LiDAR observations accurately.

Training: A training example consists of LiDAR data from $t \in [t' - T_h, t' + T_f]$, where the history data $t \in [t' - T_h, t']$ is the model input and the future data $t \in [t', t' + T_f]$ is to be used as supervision by transforming it into 4D occupancy pseudo-labels as introduced in Sec. 3.1. Here, T_p is the past time horizon and T_f is the future time horizon.

To generate the query points used during training, we sample N^+ positive (occupied) points randomly along all positive rays $\mathcal{R}_{ij}^+$ that belong to $t_{ij} \in [t', t' + T_f]$ to form a set of points $\mathcal{Q}^+$ in occupied regions. Similarly, we sample N_- negative (unoccupied) points uniformly along all current and future negative rays $\mathcal{R}_{ij}^-$ to form $\mathcal{Q}^-$. As shown in Fig. 2, for most LiDAR points the $\mathcal{R}_{ij}^+$ positive ray segment is much shorter than the negative ($\mathcal{R}_{ij}^-$), but setting $N^+ = N^-$ allows us to balance the positive and negative supervision during training. This generates a set of query points $\mathcal{Q} = \mathcal{Q}^+ \cup \mathcal{Q}^- = \{\mathbf{q}\}$ of cardinality $|\mathcal{Q}| = N^+ + N^-$.

We train our world model, with parameters θ, using a simple binary cross-entropy loss summed across all query points. Let $f_\theta(\mathbf{q})$ denote UNO's occupancy probability output at a query point $\mathbf{q}$. Our self-supervised loss is:

$$\mathcal{L}_{\text{world}} = -\frac{1}{|\mathcal{Q}|}\left(\sum_{\mathbf{q}\in\mathcal{Q}^-} \log(1 - f_\theta(\mathbf{q})) + \sum_{\mathbf{q}\in\mathcal{Q}^+} \log(f_\theta(\mathbf{q}))\right). \tag{3}$$

4. Transferring UNO to downstream tasks

To showcase the representational power and transferability of UNO, we propose simple ways to transfer it to effectively perform the downstream tasks of point cloud forecasting and BEV semantic occupancy prediction.

4.1. Point Cloud Forecasting

Point cloud forecasting has emerged as a natural task to evaluate the 4D understanding of self-driving world models [21, 24, 31, 33]. The task consists of predicting future LiDAR point clouds given past LiDAR point clouds and known sensor intrinsics and extrinsics into the future. Given *query rays* (sensor origin, direction, and timestamp) from the future LiDAR returns, the goal is to predict the depth d the rays will travel before hitting any surface.

We can simply transfer UNO to perform this task by learning a lightweight neural network $g_\gamma(\cdot)$ that acts as a renderer and predicts ray depth d from a vector of UnO-estimated occupancy values along a query ray. Specifically, to predict depth given a query ray, we first generate $N_r = 2000$ query points at a fixed interval $\epsilon_r = 0.1\,\text{m}$ along the ray, with the time of each query point equal to the timestamp of the query ray. UNO is queried at each of these points, to generate N_r occupancy predictions, which we concatenate into a vector $\hat{\mathbf{o}} = [\hat{o}_1, \ldots, \hat{o}_{N_r}]$. During training, we sample a *batch* composed of a random subset of rays from the future point clouds in order to avoid going out of memory. For the objective, we use a simple ℓ_1 loss against the ground truth (GT) depth:

$$\mathcal{L}_{render} = \sum_{(d_{GT}, \hat{\mathbf{o}}) \in \text{batch}} ||d_{GT} - g_\gamma(\hat{\mathbf{o}})||_1. \tag{4}$$

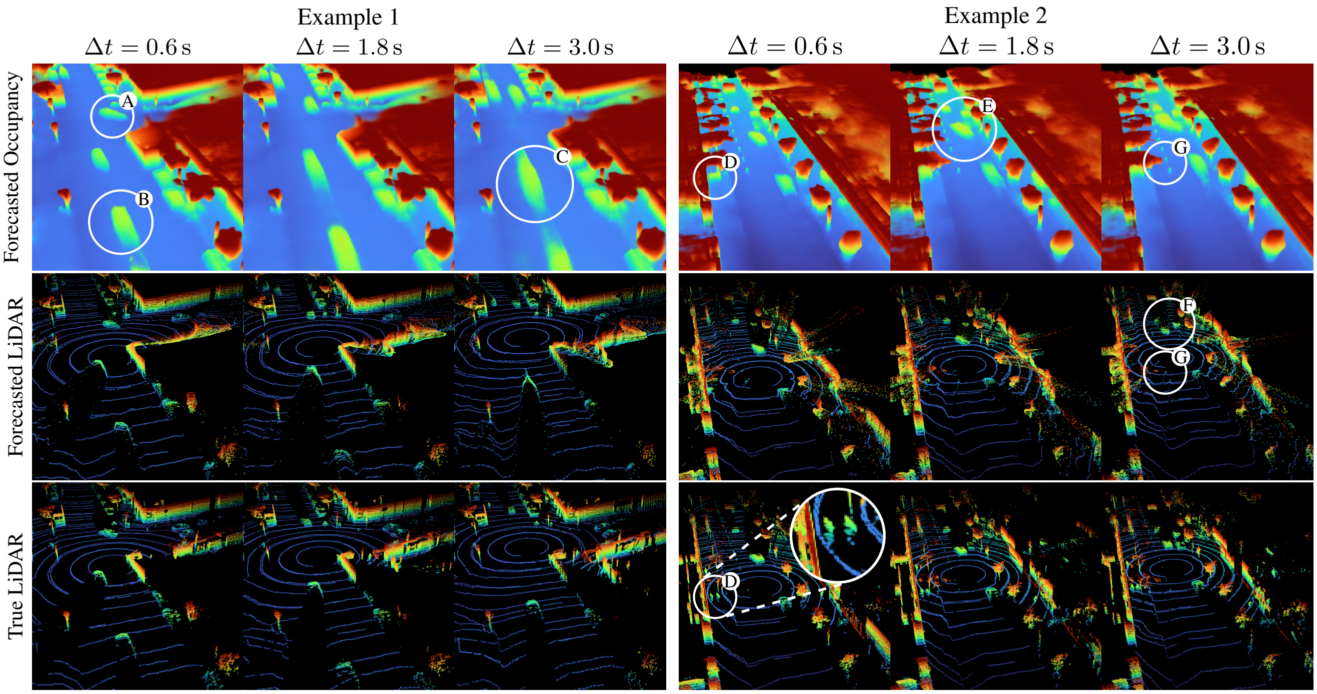
Example 1
$\Delta t = 0.6\,\text{s}$
$\Delta t = 1.8\,\text{s}$
$\Delta t = 3.0\,\text{s}$
Example 2
$\Delta t = 0.6\,\text{s}$
$\Delta t = 1.8\,\text{s}$
$\Delta t = 3.0\,\text{s}$
Forecasted Occupancy
Forecasted LiDAR
True LiDAR
A
B
C
D
E
G
F
G
D

Figure 4. A visualization of UNO on two different examples. We have labeled observations of note: **(A)** prediction of a right-turning vehicle, **(B)** object extent with only a partial viewpoint from the LiDAR data, **(C)** prediction of moving vehicle where spreading occupancy represents uncertainty in future acceleration, **(D)** prediction of walking pedestrians on the sidewalk, **(E)** prediction of a vehicle lane changing around a parked car, **(F)** persistent point cloud predictions on the lane-changing vehicle, **(G)** perceiving small objects like cones.

More details on the network $g_\gamma(\cdot)$ are provided in the supplementary. Note that we freeze UNO and train the renderer separately during this transfer, so the gradients do not affect the world model.

4.2. BEV Semantic Occupancy Forecasting

BEV semantic occupancy forecasting, which consists of predicting the occupancy probability for a set of predefined classes (e.g., vehicles, cyclists, pedestrians) for multiple time steps into the future, has become a popular task in recent years [1, 6, 14, 17, 23, 29]. The main reasons are that planning for self-driving is usually performed in BEV as the vast majority of objects of interest move along this plane, and that vulnerable road users such as pedestrians are more difficult to perceive and forecast due to irregular shapes and non-rigid motion, challenging the assumptions of traditional object detection and trajectory prediction.

In this section, we describe an efficient and simple fine-tuning method to transfer UNO to this task by leveraging supervision from semantic object annotations. To be able to query BEV points $\mathbf{q}_{BEV} = (x, y, t)$, we take a pre-trained UNO and replace the input z with a fixed learnable parameter ψ. Then, the original parameters from UNO (θ) along with the new parameter (ψ) are jointly fine-tuned in a second training stage on a small set of labeled data. Following ImplicitO [1], BEV query points are randomly sampled continuously in space and future time, and are supervised with a binary cross-entropy loss per class indicating whether the BEV point lies within an object of that category or not.

5. Experiments

In this section, we evaluate the performance of UNO across multiple tasks and datasets. Point cloud forecasting (Sec. 5.1) provides an increasingly popular and active benchmark where we can compare against prior self-driving world models. BEV semantic occupancy prediction is another very interesting task because of its direct applicability to motion planning for self-driving [6, 14, 17, 29] (Sec. 5.2). Finally, we also evaluate the quality of the geometric 4D occupancy the world model predicts (before any transfer) compared to other occupancy-based world models, with an emphasis on how well different methods can predict occupancy for object classes relevant to self-driving (Sec. 5.3).

Implementation details: We conduct our experiments on Argoverse 2 [35], nuScenes [4], and KITTI Odometry [2, 9], three popular large-scale datasets for autonomous driving. On KITTI and Argoverse2, UNO and the baselines receive $H = 5$ past frames at an interval of 0.6 s. For nuScenes, the input is $H = 6$ at an interval of 0.5 s. These settings follow prior work in point cloud forecasting [21]. For all datasets, we use a learning rate of 8.0×10^{-4} with a 1000 iteration warmup from a learning rate of 8.0×10^{-5} and a cosine learning rate schedule. We train for a total of 50,000 iterations with a batch size of 16 on the training split

		NFCD (m^2) ↓	CD (m^2) ↓	AbsRel (%) ↓	L1 (m) ↓
AV2	RayTracing [21]	2.50	11.59	25.24	3.72
	4D-Occ[21]	2.20	69.81	14.62	2.25
	UnO	**0.71**	**7.02**	**8.88**	**1.86**
NuScenes	SPFNet [33]	2.50	4.14	32.74	5.11
	S2Net [31]	2.06	3.47	30.15	4.78
	RayTracing [21]	1.66	3.59	26.86	2.44
	4D-Occ[21]	1.40	4.31	13.48	1.71
	UnO	**0.89**	**1.80**	**8.78**	**1.15**
KITTI	ST3DCNN [24]	4.19	4.83	28.58	3.25
	4D-Occ[21]	0.96	1.50	12.23	1.45
	UnO	**0.72**	**0.90**	**9.13**	**1.09**

Table 1. Point cloud prediction results on Argoverse 2 LiDAR, NuScenes, and KITTI.

of each dataset with the AdamW optimizer. On each batch sample, we train UnO on $N^+ = N^- = 900,000$ positive and negative query points. The implicit occupancy decoder is very lightweight, so it can run efficiently on many queries in parallel: UnO has roughly 17.4M parameters, only 0.06M of which are in the implicit occupancy decoder. Since some baselines are task-specific we introduce them in the corresponding sections. More details for model hyperparameters, training process for each dataset, and baselines can be found in the supplementary.

5.1. Point Cloud Forecasting

Following 4D-Occ [21], we utilize Chamfer distance (CD), Near Field Chamfer Distance (NFCD), depth L1 error (L1), depth relative L1 error (AbsRel) as our metrics. NFCD computes CD on points inside the region of interest (ROI), which is defined to be $[-70, 70]$ m in both x-axis and y-axis, as well as $[-4.5, 4.5]$ m in z-axis around the ego vehicle. AbsRel is just the L1 error divided by the ground-truth depth. Models are evaluated on 3 s long point cloud forecasts. For the Argoverse 2 LiDAR and KITTI datasets, the target point clouds are evaluated at $\{0.6\,\mathrm{s}, 1.2\,\mathrm{s}, \ldots, 3.0\,\mathrm{s}\}$, and for nuScenes they are at $\{0.5\,\mathrm{s}, 1.0\,\mathrm{s}, \ldots, 3.0\,\mathrm{s}\}$. While these parameters are chosen to match prior evaluation protocols [21], UnO can produce occupancy and point cloud predictions at any continuous point in space and future time without re-training thanks to its implicit architecture. This is important for downstream tasks such as motion planning which may require occupancy at arbitrary times.

Comparison against state-of-the-art: Tab. 1 shows quantitative comparisons of UnO and the state-of-the-art unsupervised point cloud forecasting methods. UnO, which is transferred to this task as explained in Sec. 4.1, demonstrates significant improvements in all metrics and all datasets. We observe that UnO captures dynamic objects better than the baselines, which we investigate further in Sec. 5.3. This explains the largest relative improvement on Argoverse 2, where there are more dynamics objects [35]. Additionally, we submitted UnO to the public Argoverse 2 leaderboard[1] for point cloud forecasting, where we achieved first place. Fig. 4 visualizes the point cloud predictions of our model, where we see it accurately forecasts moving vehicles with diverse future behaviors, and it captures small objects. See the supplementary for a visual comparison of our model to baselines and ablations, and more details about the public leaderboard.

	Point Cloud Forecasting				**BEV Semantic Occupancy**	
Pre-training Procedure	NFCD (m^2) ↓	CD (m^2) ↓	AbsRel (%) ↓	L1 (m^2) ↓	mAP ↑	Soft-IoU ↑
Depth-Rendering	1.79	14.1	17.0	2.80	20.7	8.6
Free-Space-Rendering	1.16	10.0	13.0	2.28	39.6	14.6
Unbalanced UnO	0.84	8.90	11.8	2.04	43.6	17.0
UnO	**0.83**	**8.10**	**10.1**	**2.03**	**52.3**	**22.3**

Table 2. Ablating the effect of occupancy pre-training procedures on the downstream tasks of point cloud forecasting and BEV semantic occupancy forecasting in Argoverse 2 Sensor dataset.

Effect of the pre-training method: We ablate multiple pre-training procedures to better understand where our improvements over state-of-the-art come from, keeping the architecture constant (as described in Sec. 3). The Free-Space-Rendering objective used in [20] learns to forecast occupancy by leveraging visibility maps of future point clouds. Concretely, the model computes free-space as the complement of the cumulative maximum occupancy along each LiDAR ray. The free-space predictions are trained via cross entropy to match the visibility map. The Depth-Rendering objective is the differentiable depth rendering loss used in [21]. This objective is similar to the one used in NeRF [25], but for estimating the expected ray depth instead of the color, based on a sequence of occupancy predictions along the ray at a fixed depth interval of ϵ_r. The Unbalanced UnO objective is the same as the UnO objective described in Sec. 3, but the query points are sampled uniformly along the ray $\{\mathcal{R}^-_{ij}, \mathcal{R}^+_{ij}\}$, instead of an equal number of query points with positive and negative labels.

While Depth-Rendering directly estimates ray depth, UnO and Free-Space-Rendering do not. Thus, we train the learned renderer described in Sec. 4 for each of these occupancy forecasting models. Tab. 2 presents the results of this comparison, evaluated on the Argoverse 2 Sensor dataset. We notice that UnO greatly outperforms the baselines on all metrics. Although the gains of balancing the loss are small in point cloud forecasting metrics, we can see the importance of it for BEV semantic occupancy, which we discuss further in Sec. 5.2.

5.2. Semantic 3D Occupancy Forecasting (BEV)

In this section we show that pre-training with unsupervised occupancy can be useful for the downstream task of 3D se-

[1] https://eval.ai/web/challenges/challenge-page/1977/leaderboard/4662

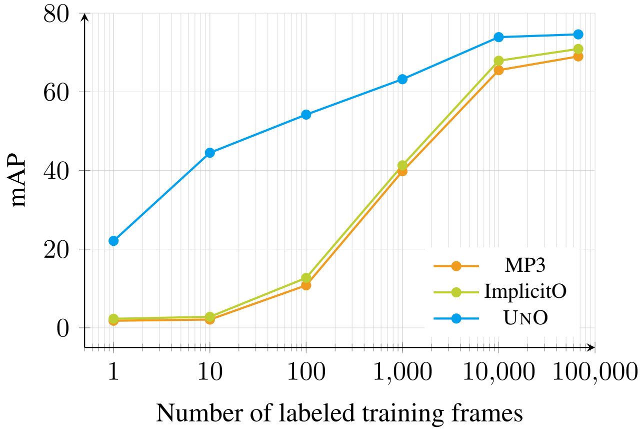

Figure 5. BEV semantic occupancy results. Fine-tuning UNO vs. SOTA supervised methods across different scales of supervision.

mantic occupancy forecasting in BEV (2D + time), especially when the supervision available is limited. We follow the evaluation protocol of [1], which evaluates occupancy forecasts on annotated vehicles from the Argoverse 2 Sensor dataset. Unlike Argoverse 2 LiDAR, this dataset has object labels annotations. For BEV semantic occupancy supervision and evaluation, we consider a ROI of $80\,\mathrm{m}$ by $80\,\mathrm{m}$ centered on the ego vehicle, and a future time horizon of $3\,\mathrm{s}$. The UNO model that serves as pre-trained weights for this experiment is the same as the one used in Sec. 5.1. During supervised fine-tuning (explained in Sec. 4.2), the query-points are sampled uniformly at random from space and future time, and during evaluation the query points are sampled on a uniform grid of spatial resolution $0.4\,\mathrm{m}$ at times $\{0.0\,\mathrm{s}, 0.5\,\mathrm{s}, \ldots, 3.0\,\mathrm{s}\}$. We follow [1, 23] in the use of mean Average Precision (mAP) and Soft-IoU as metrics to evaluate BEV semantic occupancy prediction.

Benchmark results: Fig. 5 compares UNO to SOTA BEV semantic occupancy prediction models MP3 [6] and ImplicitO [1] across varying amounts of labeled data, spanning from just 1 labeled frame to the entire Argoverse 2 Sensor `train` split. We train all models until their validation loss plateaus. When finetuning UNO, it outperforms the baselines at all levels of supervision, including when all labels are available. In the few-shot learning setting (1-10 frames), the results are particularly impressive, showing the representational power and transferability of UNO. Even with a single frame of semantic supervision, our model can learn to separate vehicles from other classes and background to a certain degree. We also call out that with roughly an order of magnitude less labeled data, UNO outperforms MP3 and ImplicitO trained with all the available training data. This highlights the efficacy of our pre-training procedure for understanding the geometry, dynamics and semantics of the scene, and the applicability of this understanding to this important task in the self-driving stack.

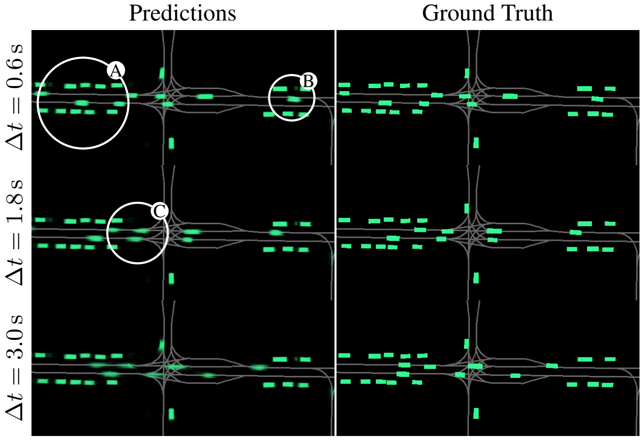
Predictions
Ground Truth
$\Delta t = 0.6\,\mathrm{s}$
$\Delta t = 1.8\,\mathrm{s}$
$\Delta t = 3.0\,\mathrm{s}$
A
B
C

Figure 6. BEV semantic occupancy predictions of fine-tuned UNO. We visualize the map for context, but this is not an input to the model. **A**: accurately perceiving crowded regions of the scene, **B**: predicting the end of a left turn, showing implicit map understanding, **C**: accurate future predictions for moving objects.

UNO pre-training is key to understanding vehicles: For each pre-trained model in Tab. 2, we fine-tune it on 100 frames of labeled data from the Argoverse 2 Sensor dataset using the procedure described in Sec. 4, with the encoder frozen for faster convergence. The occupancy forecasting metrics are in Tab. 2. We observe that UNO provides pre-trained weights that are best for fine-tuning for BEV semantic occupancy, outperforming other pre-training procedures by a large margin. While UNBALANCED UNO achieves similar point cloud forecasting performance to UNO, it is significantly worse when transferred to BEV semantic occupancy forecasting. The large ratio of free to occupied space means that the UNBALANCED UNO gets less supervision on difficult moving objects, which makes learning accurate dynamic occupancy and object extent more difficult (see the supplementary for visualizations). Point cloud forecasting metrics are largely dominated by the static background and occupancy extent does not matter for the learned raycasting model, explaining the similarity in performance between UNO and UNBALANCED UNO in these metrics. However, UNO has learned a richer representation of moving objects and extent, which are both important for BEV semantic occupancy prediction.

Qualitative results: Fig. 6 illustrates BEV semantic occupancy predictions of fine-tuned UNO on all available labeled data of vehicles from the training split of Argoverse 2 Sensor. UNO perceives all vehicles in the scene, and accurately predicts their behavior into the future. We notice that UNO implicitly understands the map, although it is not used as input or supervision. This is likely because our self-supervision of geometric occupancy forces the model to learn the roadway in order to predict accurate future behavior.

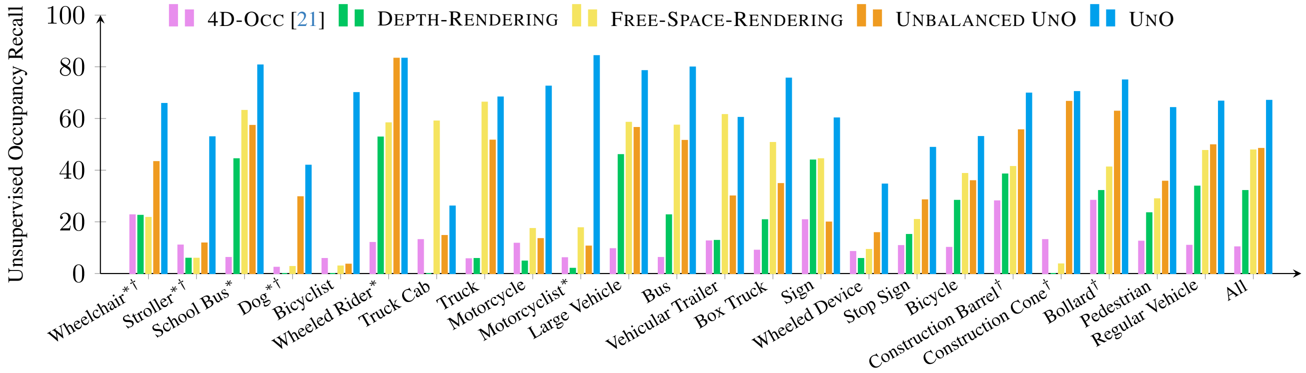

Figure 7. Unsupervised occupancy recall comparison on the Argoverse 2 Sensor dataset, averaged across the prediction horizon. Recall is computed at a precision of 0.7. * denotes the rarest 25% of classes, and † denotes the smallest (by bounding box volume) 25% of classes.

5.3. Geometric 4D Occupancy Forecasting

Sec. 5.1 showed that UnO brings significant advantages to point cloud forecasting. However, point cloud forecasting metrics are dominated by background areas where most of the LiDAR points are. In practice, these areas are the least relevant to downstream self-driving tasks like motion planning, where accurately modelling foreground dynamic road agents is vital. While Sec. 5.2 already shed some light onto UnO's understanding of dynamic objects, in this section we seek to evaluate the unsupervised geometric occupancy predictions directly, without any transfer.

To evaluate the 4D occupancy quality on relevant objects, we compute the occupancy recall within the labeled 3D bounding boxes (each with a semantic class) provided by the Argoverse 2 Sensor `val` split. To ensure fair comparison between models, we evaluate the recall of all models at the same precision level. For each model, we determine the occupancy confidence threshold that achieves a precision of 70%. To compute precision, we consider all points inside the ground truth bounding boxes (of any class) as positives, and the remainder of the space as negatives (excluding non-visible regions, which are ignored). The visible regions at a given time step are determined using ray tracing to compute a 3D visibility voxel grid from the LiDAR point cloud [15]. We evaluate points at a grid of size $80\,\mathrm{m}$ by $80\,\mathrm{m}$ centered on the ego in x and y and $[0.0\,\mathrm{m}, 3.0\,\mathrm{m}]$ in z, using a spatial discretization of $0.2\,\mathrm{m}$ at future timesteps $\{0.6\,\mathrm{s}, 1.2\,\mathrm{s}, \dots, 3.0\,\mathrm{s}\}$.

Class-wise 4D occupancy recall results: Fig. 7 shows the results of this experiment. We first note that UnO succeeds in predicting occupancy for small and rare objects. Furthermore, UnO attains significantly higher recall than state-of-the-art [21] and the other unsupervised occupancy ablations for almost all object classes, with impressive improvements on many vulnerable road users such as Stroller, Bicyclist, and Motorcycle/Motorcyclist.

Multiple aspects of UnO's geometric occupancy predictions contribute to the significant improvements over the prior art. We observe an improved forecasting of the behavior of dynamic objects. UnO's occupancy captures the multi-modal behavior of pedestrians (who may either cross the road or continue along the sidewalk), vehicles (which may lane change or change speeds), and even bicyclists (who move inside and outside of traffic). See Fig. 4 and the supplementary for some examples. On the other hand, 4D-Occ, Depth-Rendering and Free-Space-Rendering struggle to accurately model dynamic objects. Unbalanced UnO captures dynamic objects better than the other baselines, but not as well as UnO; which makes sense as its supervision has less weight on moving foreground objects. We note that Depth-Rendering uses the same objective function as 4D-Occ but has better recall in most categories. We attribute this to Depth-Rendering's implicit architecture (same as UnO), which helps mitigate discretization error and improve effective receptive field. Moreover, UnO better captures the extent of traffic participants, going beyond their visible parts (see Fig. 4). The other baselines, particularly 4D-Occ and Free-Space-Rendering, only capture the visible surfaces in the LiDAR point clouds, as shown in the supplementary.

6. Conclusion

In this paper, we propose UnO, a powerful unsupervised occupancy world model that forecasts a 4D geometric occupancy field from past LiDAR data. To tackle this problem, we leverage the occupancy implied by future point clouds as supervision to train an implicit architecture that can be queried at any continuous (x, y, z, t) point. Not only does UnO achieve an impressive understanding of the geometry, dynamics and semantics of the world from unlabeled data, but it can also be effectively and easily transferred to perform downstream tasks. To demonstrate this ability, we show that UnO outperforms the state-of-the-art on the tasks of point cloud forecasting and supervised BEV semantic occupancy prediction. We hope that UnO and future work in unsupervised world models will unlock greater safety for self-driving, notably for vulnerable and rare road users.

References

[1] Ben Agro, Quinlan Sykora, Sergio Casas, and Raquel Urtasun. Implicit occupancy flow fields for perception and prediction in self-driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2023. 1, 2, 3, 5, 7

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 5

[3] Alexandre Boulch, Corentin Sautier, Björn Michele, Gilles Puy, and Renaud Marlet. Also: Automotive lidar self-supervision by occupancy estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13455–13465, 2023. 3

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[5] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *CoRL*, 2018. 1

[6] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021. 1, 2, 5, 7

[7] Alexander Cui, Sergio Casas, Abbas Sadat, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *ICCV*, 2021. 1

[8] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint*, 2018. 2

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[10] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021. 1

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[12] 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*, pages 770–778, 2016. 3

[13] Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked autoencoder for self-supervised pre-training on lidar point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 350–359, 2023. 2

[14] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 1, 2, 5

[15] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020. 8

[16] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 2

[17] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17853–17862, June 2023. 2, 5

[18] Boris Ivanovic, Karen Leung, Edward Schmerling, and Marco Pavone. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *RA-L*, 2020. 1

[19] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, 2019. 1

[20] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *European Conference on Computer Vision*, pages 353–369. Springer, 2022. 2, 6

[21] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. Point cloud forecasting as a proxy for 4d occupancy forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1116–1124, 2023. 2, 4, 5, 6, 8

[22] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *CVPR*, 2020. 1

[23] Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):5639–5646, 2022. 1, 5, 7

[24] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022. 1, 2, 4, 6

[25] 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. *Communications of the ACM*, 65(1):99–106, 2021. 6

[26] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding

images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1, 2

[27] Thinal Raj, Fazida Hanim Hashim, Aqilah Baseri Huddin, Mohd Faisal Ibrahim, and Aini Hussain. A survey on lidar scanning mechanisms. *Electronics*, 9(5):741, 2020. 3

[28] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. *arXiv preprint arXiv:2210.14222*, 2022. 2

[29] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *ECCV*, 2020. 1, 2, 5

[30] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 1

[31] Xinshuo Weng, Junyu Nan, Kuan-Hui Lee, Rowan McAllister, Adrien Gaidon, Nicholas Rhinehart, and Kris M Kitani. S2net: Stochastic sequential pointcloud forecasting. In *European Conference on Computer Vision*, pages 549–564. Springer, 2022. 1, 2, 4, 6

[32] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IROS*, 2020. 1

[33] Xinshuo Weng, Jianren Wang, Sergey Levine, Kris Kitani, and Nicholas Rhinehart. Inverting the pose forecasting pipeline with spf2: Sequential pointcloud forecasting for sequential pose forecasting. In *Conference on robot learning*, pages 11–20. PMLR, 2021. 1, 2, 4, 6

[34] Xinshuo Weng, Ye Yuan, and Kris Kitani. Ptp: Parallelized tracking and prediction with graph neural networks and diversity sampling. *RA-L*, 2021. 1

[35] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 1, 2, 5, 6

[36] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 3

[37] Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. Unipad: A universal pre-training paradigm for autonomous driving. *arXiv preprint arXiv:2310.08370*, 2023. 3

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 4