# Realistically distributing object placements in synthetic training data improves the performance of vision-based object detection models

**Setareh Dabiri**[1] **Vasileios Lioutas**[1,2] **Berend Zwartsenberg**[1] **Yunpeng Liu**[1,2]
**Matthew Niedoba**[1,2] **Xiaoxuan Liang**[1,2] **Dylan Green**[1,2] **Justice Sefas**[1,2]
**Jonathan Wilder Lavington**[1,2] **Frank Wood**[1,2,3] **Adam Ścibior**[1,2]
[1]Inverted AI [2]University of British Columbia [3]Mila

## Abstract

*When training object detection models on synthetic data, it is important to make the distribution of synthetic data as close as possible to the distribution of real data. We investigate specifically the impact of object placement distribution, keeping all other aspects of synthetic data fixed. Our experiment, training a 3D vehicle detection model in CARLA and testing on KITTI, demonstrates a substantial improvement resulting from improving the object placement distribution.*

| Metric | Dataset | AP11/AP40 Easy | Moderate | Hard |
|---|---|---|---|---|
| 2D BBox | Baseline | 56.7/56.9 | 42.1/40.6 | 35.2/32.8 |
| | INITIALIZE | **67.3/67.7** | **51.3/49.9** | **43.7/40.9** |

Table 1. Average precision of 2D bounding boxes on KITTI validation set, predicted by models trained on synthetic datasets with baseline and realistic vehicle placements respectively. Predicted bounding boxes with IoU larger than 0.7 with ground truth are considered successful detections.

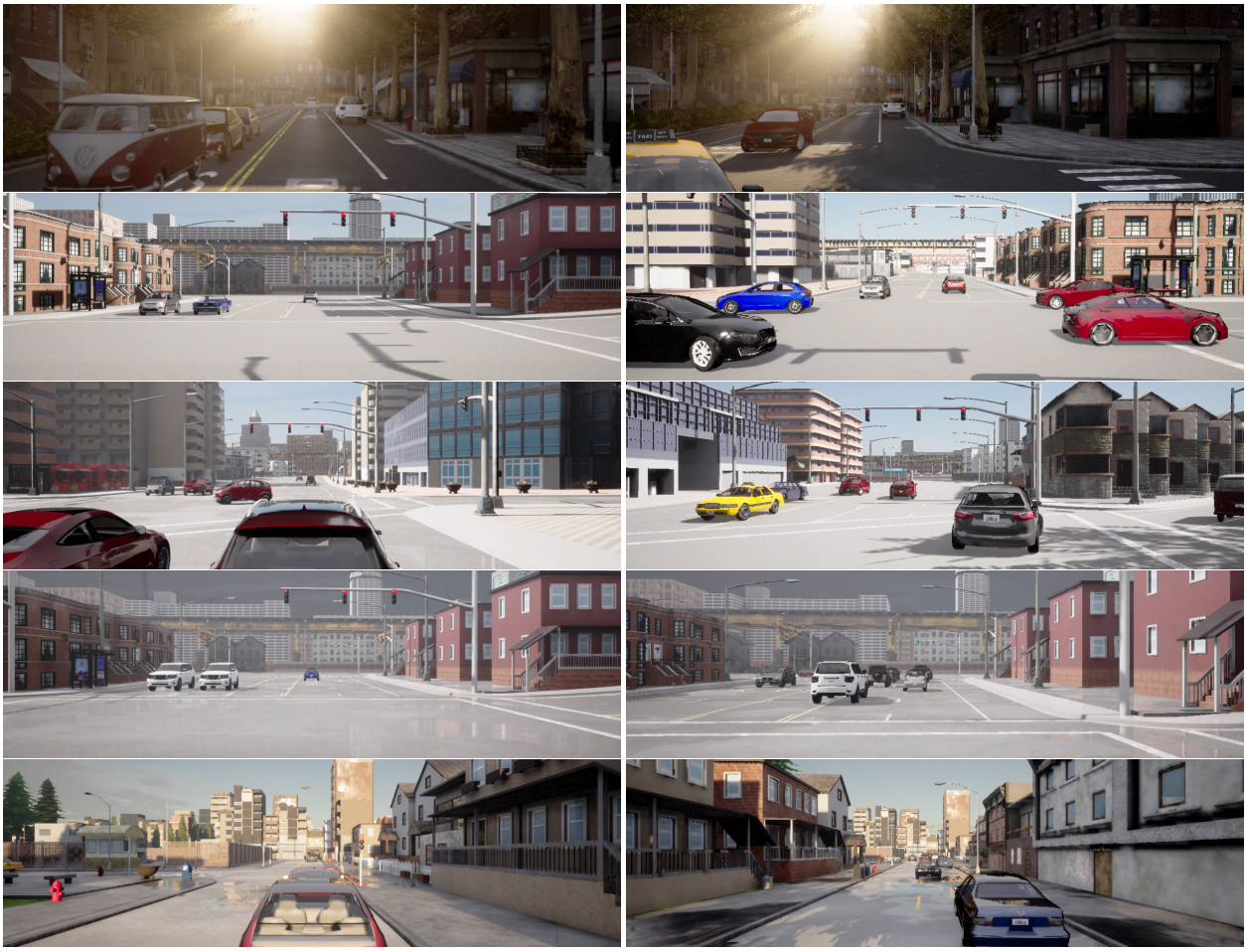
## 1. Introduction

It is well known that any domain gap between training and test data hurts the performance of machine learning models in general, and object detectors in particular. When training with synthetic data obtained from simulations, the bulk of attention in the existing literature has been on the domain gap that has to do with the visuals, such as textures, lighting, weather, etc. (also referred to as the *appearance gap*), while the impact of different types, numbers, and placements of objects (called the *content gap*) has not been the primary area of research. In Sec. 3 we review existing work that addresses the content gap generally and the placement distribution in particular, but we believe the literature is lacking a clear demonstration of how much of an impact the placement distribution in synthetic data can have on the performance of vision-based object detectors in driving contexts. In this paper we test the hypothesis that the realism of physical object placement distribution in synthetic data has a significant impact on the performance of vision models trained on said data.

We use a carefully controlled experimental setup, where we generate training data using the CARLA driving simulator [3] and we use real validation data provided in KITTI [5] as test data. We compare a *baseline* object placement model, where we allow the CARLA Traffic Manager to freely move vehicles and take a snapshot of their positions at a particular time, with our commercial model[1], called INITIALIZE, that jointly samples realistic vehicle placements. We specifically isolate the object placement distribution as the independent variable, fixing the object types, appearances, counts, as well as weather conditions and locations so that they match exactly between the two versions of the training dataset. Our results show a large improvement in test set performance arising from that single intervention.

We use a PGD [21] model for object detection. This choice is mostly orthogonal to the claims of the paper and we do not expect it to have a significant bearing on the results. We use the publicly available source code for PGD [1], and will release our training datasets and specific configurations for reproducibility purposes. We report standard quantitative performance metrics and provide some qualitative illustrations for the differences between the baseline and INITIALIZE training sets, as defined in the paragraph above, as well as the different test set predictions made by PGD trained on each of those datasets respectively. It is worth noting that we do not attempt to achieve competitive performance on KITTI, which would require addressing the appearance gap, as well as increasing the variety of vehicle models and locations, and perhaps additional training on real data. Instead, our focus is solely on isolating the

[1] https://docs.inverted.ai



(a) Baseline (b) INITIALIZE

Figure 1. Sample training set images generated using CARLA. We compare the baseline placement (left) with a realistic one (right).

impact of vehicle placement distribution in training data on test set performance.

## 2. Experiments

### 2.1. Data generation

We generate a KITTI-like dataset for 3D object detection from a forward-facing camera angle using CARLA. We manually designate regions of interest in Town 1, Town 2, Town 3, Town 4, and Town 10 to cover different road geometries. The baseline dataset uses the available "Traffic Manager" in CARLA to drive vehicles from predefined spawning locations, while the dataset with realistic vehicle placements samples vehicle positions directly based on traffic patterns learned from data. In an attempt to obtain a dataset that is diverse in appearance, we generate scenes by varying the sun angle to simulate different times of day, as well as various weather conditions, including clear, cloudy, and rainy settings. To ensure fairness in the generation process for both datasets, we specify the same number of agents and episodes for each generated scene. Specifically, we place 15 vehicles within a circular region of interest (ROI) with a radius of 50 meteres. If the number of available CARLA spawn points within the ROI is less than 15, we lower the number of vehicles to match the number of spawn points available, in both versions of the dataset. This ensures that each image has the same number of vehicles within the same ROI in the same weather in both datasets. The number of vehicles visible in the image can vary between the datasets, since the camera does not capture all agents within the ROI.

We randomly assign one of the vehicles in the ROI as the ego vehicle and record snapshots from the camera located on the top of the ego agent for both datasets. To maintain consistency with the KITTI dataset, we save the images in $1242 \times 375$ resolution. Each dataset includes 1844 images. Figure 1 depicts some samples from the baseline dataset and the INITIALIZE dataset. It is easy to see that the realistic

placement produces more variability, in particular regarding the positioning of vehicles relative to the centerline.

## 2.2. Results

To demonstrate the importance of vehicle placement on realism in synthetically generated training data, we train a monocular 3D detection model named PGD [21], using the source code provided by its original authors [1]. We train two versions of this model, on baseline dataset and the dataset with realistic vehicle placements respectively, but otherwise identical, and evaluate the performance of both versions on the KITTI validation dataset, consisting of 3769 images. We use the same hyperparameters for both versions, the exact values of which will be released with the source code. The average precision of the 2D bounding box (2DBBox), bird's eye view (BEV), 3D bounding box (3D BBox) and average orientation similarity (AOS) [5] of two trained models tested on the KITTI validation set are reported in tables Tab. 3, Tab. 2 and Tab. 1.

The tables display the performance on three different difficulty levels defined in the KITTI dataset: Easy, Moderate, and Hard. Table 3 presents results where bounding boxes with an overlap of more than 70% are counted as positive, and Tab. 2 displays the same for overlaps greater than 50%. The criteria for determining the difficulty are the minimum bounding box height, maximum occlusion level, and maximum truncation percentage as described in [5]. The results include both object detection and orientation estimation, which are evaluated using the average precision (AP) and average orientation similarity (AOS) metrics, respectively.

As evident from the data presented in Tab. 2 and Tab. 3, using realistic object placements drastically improves average precision of 3D bounding box and BEV of cars across all dataset difficulty categories. Moreover, as indicated in Tab. 3, training the model on the dataset with realistic vehicle placements results in a considerable gain in the average orientation similarity of the predicted bounding boxes. Table 1 illustrates a substantial improvement in the average precision of 2D bounding boxes. Figure 2 illustrates the predicted 3D bounding boxes on images from the KITTI validation set, once again showing that the realistic vehicle placement from INITIALIZE results in better performance on real data.

| | | AP11/AP40 | | |
|---|---|---|---|---|
| Metric | Dataset | Easy | Moderate | Hard |
| BEV | Baseline | 0.68/0.35 | 0.56/0.2 | 0.53/0.19 |
| | INITIALIZE | **9.1/5.8** | **7.4/4.4** | **6.5/3.8** |
| 3D BBox | Baseline | 0.32/0.13 | 0.32/0.10 | 0.32/0.04 |
| | INITIALIZE | **6.8/2.8** | **5.8/2.2** | **5.6/1.9** |

Table 2. Average precision of BEV and 3D bounding boxes on KITTI validation set, predicted by models trained on synthetic datasets with baseline and realistic vehicle placements respectively. Predicted bounding boxes with IoU larger than 0.5 with ground truth are considered successful detections.

| | | AP11/AP40 | | |
|---|---|---|---|---|
| Metric | Dataset | Easy | Moderate | Hard |
| BEV | Baseline | 0.05/0.02 | 0.06/0.01 | 0.057/0.01 |
| | INITIALIZE | **0.27/0.11** | **1.51/0.09** | **1.51/0.03** |
| 3D BBox | Baseline | 0.02/0.01 | 0.04/0.01 | 0.04/0.0 |
| | INITIALIZE | **0.07/0.01** | **1.51/0.01** | **1.51/0.01** |
| AOS | Baseline | 17.7/17.9 | 13.7/13.2 | 14.1/11.2 |
| | INITIALIZE | **21.0/20.9** | **17.3/16.5** | **15.2/13.8** |

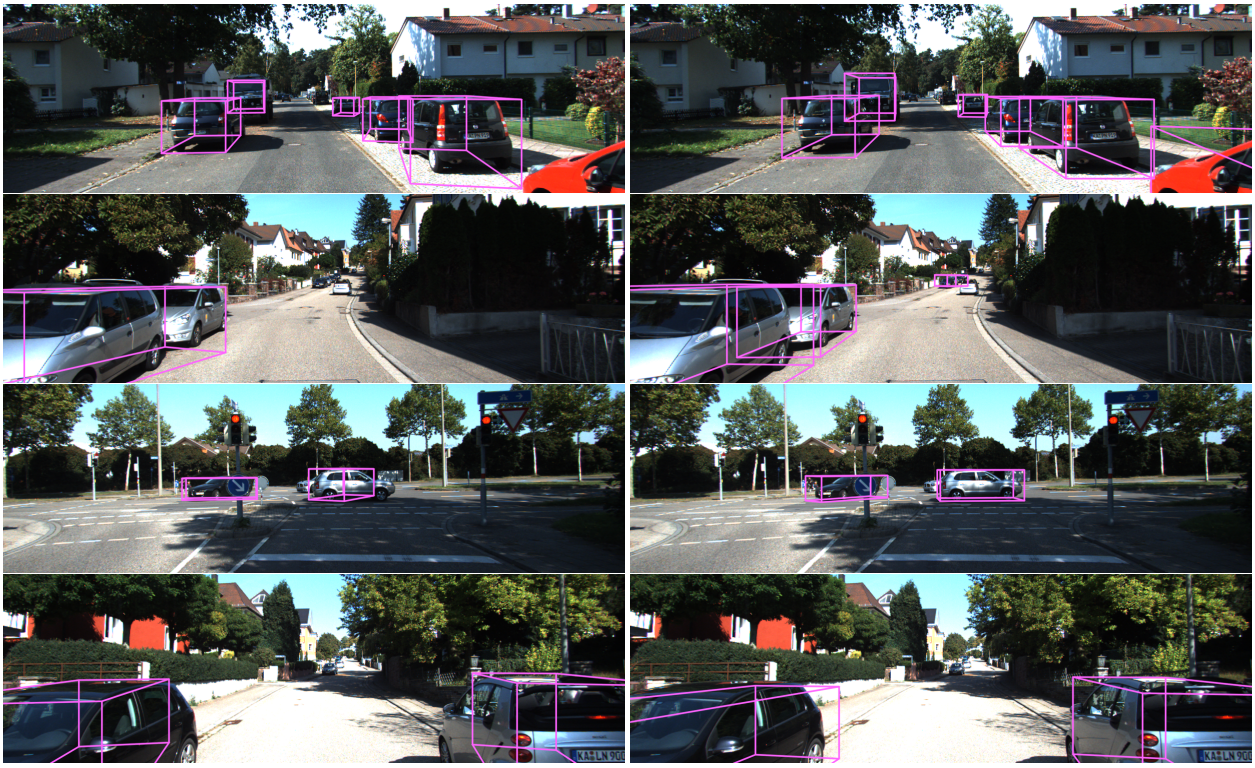
Table 3. Average precision of BEV, 3D bounding boxes and AOS on KITTI validation set, predicted by models trained on synthetic datasets with baseline and realistic vehicle placements respectively. Predicted bounding boxes with IoU larger than 0.7 with ground truth are considered successful detections.

# 3. Related work

Training object detection models with synthetic data is a well-established approach, which is attractive mostly due to its relative low cost for both recording and annotating compared to data obtained in the real world. The literature on the subject is vast and the topic of many articles [4, 11–13, 19]. The domain gap between real and synthetic data is recognized as a key problem and substantial effort has been dedicated to reducing the appearance gap [4, 10], with few notable papers addressing the content gap [6, 7, 14]. In most applications, the content gap is addressed through Domain Randomization [18, 20], where object placements are sampled from some broad, uninformative distribution, but we also discuss some more elaborate approaches below. Overall, it has been demonstrated that training on synthetic data can produce state-of-the-art performance on real data [12, 19], although it is typically advantageous to use a mix of synthetic and real data [12].

There is a major line of work addressing the content gap, and in particular the placement distribution, originating with Structured Domain Randomization (SDR) [14], which used a highly structured, but hand-coded distribution to generate content for the scene. This approach was then extended to learn both the parameters [9] and the structure [2] of this distribution, eventually being able to learn the full distribution of both content and appearance from unlabelled real data [15]. While some of those papers contain experiments similar to ours they do not isolate the impact of placement distribution, and their code and data are not available.

The basic approach for placing vehicles in a driving simulator, which we use as the baseline in this paper, is to spawn vehicles at designated locations and then allow the built-in behavioral models to drive them around and take



(a) Baseline (b) INITIALIZE

Figure 2. 3D bounding box predictions on the KITTI dataset. The left column depicts predictions of the model trained on synthetic data with baseline vehicle placements, while the right column shows predictions from the model trained on synthetic data with realistic vehicle placements.

a snapshot of their positions at some point in time. This produces limited variability due to the simplicity of the behavioral models and is often supplemented by Domain Randomization [18, 20], where the objects are placed in the scene at random according to some simple distribution. Scenes generated this way are often unrealistic, in particular in driving scenes many vehicles would be placed off-road. It is therefore common to manually engineer more complex distributions with domain-specific heuristics [16], which can perform well but require a lot of human effort. Another approach is to use ground truth object placements from real data and synthetically generate a variety of appearances [4].

Among the learning-based approaches to object placement, SceneGen [17] is a method specifically designed to learn the placement distribution of vehicles on the road and the paper contains an experiment that demonstrates how the realism of this distribution in synthetic training data impacts object detection on real data. The experiment is limited to LiDAR-based, rather than vision-based models, and the results are reported on a private dataset only. Various other models for placement distribution have been proposed, such as LayoutVAE [8] and Permutation Invariant Flows [22], but those papers do not study how using such models for synthetic data generation impacts downstream object detection performance. Since, to the best of our knowledge, none of those models are publicly available, we obtain our realistic placement samples by calling INITIALIZE, a public commercial API, which is learning-based but the details of the underlying model were not disclosed.

## 4. Conclusion

We have conducted a simple experiment that unambiguously shows that a realistic object placement distribution can have a dramatic impact on real-world performance when training object detectors on synthetic data in driving contexts. We believe that this placement distribution is a critical consideration when assembling synthetic datasets and that our paper will convince practitioners to pay close attention to this issue when working with synthetic data. To allow better reproducibility and comparisons with other placement generation methods, we make our code and datasets publicly available[2].

[2] https://github.com/inverted-ai/object-detection

## References


[1] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 1, 3

[2] Jeevan Devaranjan, Amlan Kar, and Sanja Fidler. Meta-Sim2: Unsupervised Learning of Scene Structure for Synthetic Data Generation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12362, pages 715–733. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science. 3

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Conference on Robot Learning*, pages 1–16. PMLR, Oct. 2017. ISSN: 2640-3498. 1

[4] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual Worlds as Proxy for Multi-Object Tracking Analysis. *arXiv:1605.06457 [cs, stat]*, May 2016. arXiv: 1605.06457. 3, 4

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 3

[6] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017. 3

[7] Josef Haddad. Data synthesis in deep learning for object detection, 2021. 3

[8] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. LayoutVAE: Stochastic Scene Layout Generation From a Label Set. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9894–9903, Seoul, Korea (South), Oct. 2019. IEEE. 4

[9] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-Sim: Learning to Generate Synthetic Datasets. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4550–4559, Seoul, Korea (South), Oct. 2019. IEEE. 3

[10] Rawal Khirodkar, Donghyun Yoo, and Kris Kitani. Domain Randomization for Scene-Specific Car Detection and Pose Estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1932–1940, Jan. 2019. ISSN: 1550-5790. 3

[11] Qinghai Miao, Yisheng Lv, Min Huang, Xiao Wang, and Fei-Yue Wang. Parallel Learning: Overview and Perspective for Computational Learning Across Syn2Real and Sim2Real. *IEEE/CAA Journal of Automatica Sinica*, 10(3):603–631, Mar. 2023. Conference Name: IEEE/CAA Journal of Automatica Sinica. 3

[12] Farzan Erlik Nowruzi, Prince Kapoor, Dhanvin Kolhatkar, Fahed Al Hassanat, Robert Laganiere, and Julien Rebut. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *arXiv preprint arXiv:1907.07061*, 2019. 3

[13] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning Deep Object Detectors From 3D Models. pages 1278–1286, 2015. 3

[14] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255, Montreal, QC, Canada, May 2019. IEEE. 3

[15] Aayush Prakash, Shoubhik Debnath, Jean-Francois Lafleche, Eric Cameracci, Gavriel State, Stan Birchfield, and Marc T. Law. Self-Supervised Real-to-Sim Scene Generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16024–16034, Montreal, QC, Canada, Oct. 2021. IEEE. 3

[16] Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight without a Single Real Image, June 2017. arXiv:1611.04201 [cs]. 4

[17] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. SceneGen: Learning to Generate Realistic Traffic Scenes. *arXiv:2101.06541 [cs]*, Jan. 2021. arXiv: 2101.06541. 4

[18] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, Sept. 2017. ISSN: 2153-0866. 3, 4

[19] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to Track with Object Permanence. *arXiv:2103.14258 [cs]*, Sept. 2021. arXiv: 2103.14258. 3

[20] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018. 3, 4

[21] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting objects in perspective. In *Conference on Robot Learning (CoRL) 2021*, 2021. 1, 3

[22] Berend Zwartsenberg, Adam Ścibior, Matthew Niedoba, Vasileios Lioutas, Yunpeng Liu, Justice Sefas, Setareh Dabiri, Jonathan Wilder Lavington, Trevor Campbell, and Frank Wood. Conditional Permutation Invariant Flows. *Transactions on Machine Learning Research*, 2023. 4