# SOccDPT: Semi-Supervised 3D Semantic Occupancy from Dense Predictive Transformers trained under memory constraints

Anonymous ICCV submission

Paper ID 42

## Abstract

*We present SOccDPT, a memory-efficient approach for 3D semantic occupancy prediction from monocular image input using dense predictive transformers. To address the limitations of existing methods trained on structured traffic datasets, we train our model on unstructured datasets including the Indian Driving Dataset and Bengaluru Driving Dataset. Our semi-supervised training pipeline allows SOccDPT to learn from datasets with limited labels by reducing the requirement for manual labelling and substituting it with pseudo-ground truth labels. This broader training enhances our model's ability to handle unstructured traffic scenarios effectively. To overcome memory limitations during training, we introduce patch-wise training where we select a subset of parameters to train each epoch, reducing memory usage during auto-grad graph construction. By considering unstructured traffic and introducing memory-constrained training, SOccDPT achieves a competitive performance as shown by semantic segmentation IoU score of 41.71% and monocular depth estimation RMSE score of 12.4075, even under limited memory constraints and operating at a competitive frequency of 47 Hz. We have made our code and dataset augmentations public*

## 1. Introduction

Autonomous navigation requires 3D semantic understanding of the environment at a high frequency with a limited compute budget. The field of autonomous driving has shown significant interest in vision-based 3D scene perception due to its exceptional efficiency and abundant semantic information. When it comes to choosing an architecture, works such as [1, 29, 28, 14, 21, 19] inspired from ViT [8] have the domain agnostic learning capabilities of the transformer. The transformer's versatility comes at the cost of having no good inductive priors for any domain, requiring large volumes of training data and a large volume of GPU memory to train. To apply such models on a new domain, we must be efficient in making use of transfer learning and pseudo-labelling to solve the ground truth data scale problem.

In the context of 3D semantic occupancy, ground truth data would refer to semantically labelled 3D point clouds with corresponding RGB images acquired from a calibrated sensor rig. While there exist datasets [24, 11, 9] which have labelled 3D semantic occupancy data in the context of structured traffic, the unstructured traffic scenarios remain largely underrepresented. It may not be feasible to gather large volumes of training data considering the fact that LiDAR sensors are expensive and labelling 3D semantic classes can be tedious. Hence, we make use of a set of teacher models and boosting techniques inspired from [35, 34, 4, 10] to produce labels for depth and semantic on driving video footage which we use to supervise the training of our model. We train our system on unstructured driving datasets such as the Indian Driving Dataset [44] and Bengaluru Driving Dataset [10] to ensure that our system generalizes well. Training such models requires large volume of GPU memory. We overcome this hurdle with our PatchWise training approach which keeps the GPU memory in check and this allowed us to explore higher batch sizes without altering the back-propagation algorithm.

With the goal of designing a model, which is efficient during both training and inference, we propose SOccDPT and our PatchWise training system. To ensure SOccDPT performs well in unstructured traffic scenarios, we introduce semi-supervision with our pseudo-labelling process for depth boosting and semantic auto-labelling. We use a common backbone for image feature extraction and dual heads to extract disparity and semantic information of the scene. Camera intrinsics are used along with disparity to project the semantic information into 3D space.

## 2. Related Work

**Multi-Task Learning.** The domain of visual perception has extensively studied the concept of Multi-Task Learning including tasks such as semantic segmentation [27, 52]

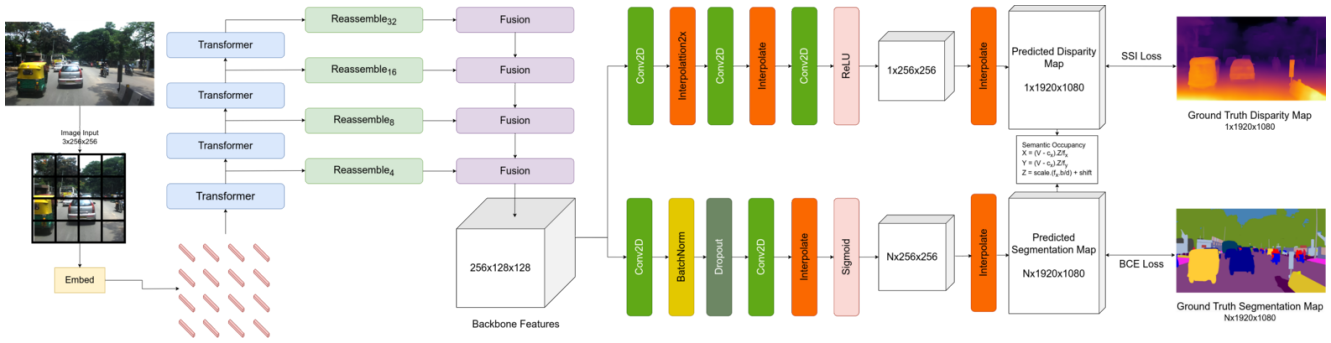


Figure 1. SOccDPT uses the ViT family [8, 1, 29, 28, 14, 21, 19] for backbone feature extraction which allows us to carefully balance accuracy and compute requirements. SOccDPT takes an RGB image input of shape 3x256x256 produces image features of shape 256x128x128. We then pass the extracted features to a disparity head and a segmentation head. We apply the Scale and Shift Invariant loss [38] and the Binary Cross Entropy loss for the disparity and segmentation outputs respectively. With the known camera intrinsic, we project the semantics into 3D space with the help of the disparity map, thus producing a 3D semantic map from one backbone

and jointly learning depth and semantics [45]. A comprehensive discussion of deep multi-task learning can be found in [43]. Although there have been some advancements in semi-supervised multi-task learning, as evidenced by previous studies [7, 31], these approaches do not specifically tackle the difficult problem of training models across diverse datasets in the absence of ground truth information. Our focus is to address this gap by developing a solution for semi-supervised multi-task learning in the domain of 3D semantic occupancy within unstructured traffic environments.

**Semi-Supervised Learning and Self-Supervised Learning.** In the context of disparity estimation, semi-supervised learning has become very important due to the challenges involved in obtaining accurate depth information in diverse real-world environments. Several self-supervised algorithms for perceiving depth have been suggested [12, 13, 46, 22, 33]. These algorithms offer the advantage of utilizing only a single camera, making them suitable for easy deployment in real-world scenarios. However, they still face numerous unresolved issues. One such problem is the generation of disparity maps that lack local and temporal consistency. Watson et al. [46] addressed the temporal inconsistency by incorporating multiple consecutive frames as input. Another line of research in semi-supervised learning looks into using the existing model to generate confident annotations on unlabelled data. Examples for such approaches include pseudo-labelling [18] and entropy minimization [15]. Since the degree of disparity is inversely related to depth, as demonstrated in equation 4, slight variations in disparity for distant objects lead to significant variations in depth. Consequently, the resulting point clouds exhibit non-uniform resolution, with closer objects represented by more points compared to those farther away. There are broadly two approaches tot he disparity estimation problem: monocular and stereoscopic.

**Monocular and Stereo Disparity Estimation.** Diverse neural network architectures, including variational autoencoders, convolutional neural networks, generative adversarial networks and recurrent neural networks, have demonstrated their efficacy in tackling the task of depth estimation. Within this framework, two methods are commonly employed: monocular, where depth is estimated from a single input image, and stereoscopic depth estimation, where depth is estimated from a pair of images provided as input to the system. Monocular approaches such as [12, 13, 46, 38] take advantage of depth cues such as occlusion boundaries, parallel lines and so on to understand the 3D scene. Techniques based on binocular learning [6, 26, 32, 47, 48], generate depth maps by leveraging the epipolar constraints associated with feasible disparity values. Although these approaches have enhanced accuracy, they come with a trade-off in terms of computational time and/or hardware demands. Running such systems in real-time becomes impractical on embedded devices that lack sufficient power, especially without specialized hardware like FPGAs.

**Bird's Eye View (BEV) Architectures.** Obtaining a top-down view of a scene offers a comprehensive understanding of the surrounding environment, effectively capturing both static and dynamic elements. BEV architectures, exemplified by [39, 23, 40], generate this top-down map, which can be utilized for path planning purposes. This top down map is essentially a segmentation map which would highlight the road, non-drivable space, parking areas, vehicles, pedestrians and so on. The concept of predicting BEV from multiple camera perspectives has demonstrated performance comparable to LiDAR-centric methods [54, 30]. However, a limitation of this approach is the absence of 3D information about the scene, such as unclassified objects, potholes, and overhanging obstacles.

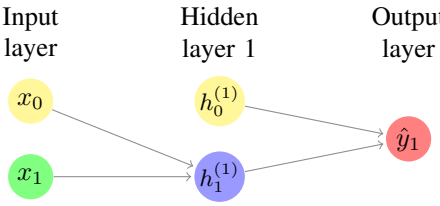


Figure 2. A perceptron which takes one input, produces one output and has a single hidden layer

| Parameter | Weight Update |
|---|---|
| $w_2$ | $\frac{\partial L}{\partial y}\frac{\partial y}{\partial w_2}$ |
| $h_0^{(1)}$ | $\frac{\partial L}{\partial y}\frac{\partial y}{\partial h_0^{(1)}}$ |
| $w_1$ | $\frac{\partial L}{\partial y}\frac{\partial y}{\partial h_1^{(1)}}\frac{\partial h_1^{(1)}}{\partial w_1}$ |
| $x_0$ | $\frac{\partial L}{\partial y}\frac{\partial y}{\partial h_1^{(1)}}\frac{\partial h_1^{(1)}}{\partial x_0}$ |

Table 1. Weight updates for all the parameters of the network

**3D Occupancy Networks.** Achieving an effective representation of a 3D scene is a fundamental objective in perceiving 3D environments. One direct approach involves discretizing the 3D space into voxels within an occupancy grid [53, 54]. The voxel-based representation is advantageous for capturing intricate 3D structures, making it suitable for tasks like LiDAR segmentation [5, 25, 42, 50, 51, 54] and 3D scene completion [2, 3, 20, 41, 49]. A recent method, TPVFormer [17], addresses memory optimization by representing the 3D space as projections on three orthogonal planes. Despite the significant progress made by these approaches, they do not specifically tackle the issue of existing dataset biases towards structured traffic.

## 3. Proposed Work

### 3.1. SOccDPT Architecture

Modern vision based autonomous vehicles and robotics systems require fast and reliable methods to perceive the 3D environment. While cameras are cost effective and information-rich sensors, they are only able to capture 2D projections of our 3D world. Accurately undoing this projection while understanding 3D semantics in real time is crucial to be able to apply pure visual perception in safety critical systems. As described in figure 1, SOccDPT uses the Dense Predictive Transformer [37, 28] backbones to efficiently extract image features. We then use independent heads to produce the disparity and segmentation maps. Instead of penalizing the model for generating the output in an inaccurate scale, we address the issue of arbitrary scale in the disparity map by estimating the scale and shift relative to the ground truth for every frame. This estimation process involves aligning the prediction with the ground truth using a least-squares criterion as shown in equation 1 where $d$ is predicted disparity, $d^*$ is ground truth disparity, $\hat{d}$ is scaled and shifted prediction, $\hat{d}^*$ is scaled and shifted ground truth disparity (which is equivalent to $d^*$) and $M$ is the number of pixels. The terms $(s,t)$ refer to the estimated scale and shift respectively.

$$(s,t) = argmin_{s,t} \sum_{i=1}^{M} (sd_i + t - d_i^*)^2, \quad \hat{d} = sd + t, \hat{d}^* = d^* \tag{1}$$

$$\mathcal{L}_{ssi}(\hat{d}, \hat{d}^*) = \frac{1}{2M} \sum_{i=1}^{M} \rho(\hat{d}_i, \hat{d}_i^*) \tag{2}$$

Using the computed scale and translation $(s,t)$, we apply the Scale and Shift Invariant loss [38] as described by $\mathcal{L}_{ssi}(\hat{d}, \hat{d}^*)$ in equation 2 where $\rho$ is the specific loss function. We set $\rho$ to a sum of Mean Squared Error and Gradient Loss. Gradient Loss is computed as shown in equation 3 where $H$ and $W$ represent the height and width of the image respectively, $\delta$ is the difference between the predicted image and target image.

$$\mathcal{L}_{grad} = \sum_{i=1}^{N}\sum_{j=1}^{H} \left(|\delta_{ij} - \delta_{(i-1)j}|\right) + \sum_{i=1}^{H}\sum_{j=1}^{W} \left(|\delta_{ij} - \delta_{i(j-1)}|\right) \tag{3}$$

Once the segmentation and disparity maps are computed, we make use of the camera intrinsics to project the semantics into 3D space. Consider a point on the image plane at position $(u,v)$ with disparity $D(u,v)$ and 2D semantics $S_{2D}(u,v)$. This point corresponds to the 3D point $(x,y,z)$ as shown in equation 4 from which we can assert the 3D semantics correspondence to be $S_{3D}(x,y,z) \longleftarrow S_{2D}(u,v)$

$$(x,y,z) = \left(\frac{b\cdot(u-o_x)}{D(u,v)}, \frac{b\cdot f_x\cdot(v-o_y)}{f_y\cdot D(u,v)}, \frac{b\cdot f_x}{D(u,v)}\right) \tag{4}$$

In order to train our network, we started off by building a baseline model $V1$ which consists of 2 separate backbones, one for disparity and the other for segmentation. This informs us of the performance of the dense predictive transformer on unstructured traffic datasets. We improve upon $V1$ by having a common backbone in $V2$ which lead to optimizations in speed and memory consumption. This came at the cost of the accuracy of both the segmentation and disparity. This is due to the fact that the network would be

learning the features and intricacies of both the tasks from scratch simultaneously. To address this, $V3$ makes a minor modification to $V2$ which allows us to load in the disparity estimation backbone from $V1$. This allows $V3$ to have a backbone which is proficient in the disparity estimation task. When starting from this point, the backbone and segmentation head only have to learn the task of image segmentation, without making any major alterations to the existing disparity estimation. This provided an improvement in how much the model was able to learn with the same data.

### 3.2. Patchwise Training

Consider the single-layered perceptron shown in figure 2 with no activation functions. The perceptron takes a single input $x$ and produces a single output $y$. This network is being supervised by the MSE Loss.

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

The perceptron's forward pass is computed as shown in the equations 6 and 7

$$h_1^{(1)} = w_1.x_1 + x_0 \tag{6}$$

$$y = w_2.h_1^{(1)} + h_0^{(1)} \tag{7}$$

Given the forward pass equations and the loss function, we can calculate the weight update functions for all the weights and biases of the network as shown in table 1. The weight update for any given weight is not affected by the other weights. Hence we can compute the weight updates for each parameter individually before performing a batch update on all the parameters. While this is a well-known fact, modern Deep Learning tensor libraries such as PyTorch [36] compute the weight updates for all the parameters at once on the GPU. Such an operation will attempt to allocate a large volume of GPU memory and can result in "out of memory" errors. We solve this issue by implementing our "PatchWise" module that allows us to only compute the weight updates for a given percentage of the model at a time and we update all the weights at once for each epoch. This allows us to make the speed-time trade-off where we can train larger neural networks on our systems with limited GPU memory while increasing training time. We also benefit from being able to experiment with larger batch sizes that would otherwise not fit into GPU memory. The algorithm 1 describes the implementation of our PatchWise method on a PyTorch model.

### 3.3. Pseudo-Ground Truth Labels for Semi-Supervision

The ability of vision based networks to learn and accurately predict based on image input is limited by their receptive field and their overall learning capacity (number of parameters). Most of these networks are trained on a predefined low input resolution. While it is possible to feed in larger input resolutions at test time, the system's performance will suffer due to the loss of features that would have otherwise been visible at higher resolutions. We use existing disparity estimation and image segmentation models to generate labels for existing datasets which don't already have the desired labels. The broad concept is to trade away compute time for higher accuracy as this allows us to generate pseudo-labels. The Indian Driving Dataset [44] has 2D

**Algorithm 1:** PatchWise

PatchWise (net, train_percentage, train_step);
**Input :** PyTorch Module net, training percentage train_percentage, training function train_step
**Output:** Trained neural network

```
N ← length(net.parameters);
M ← round(N × train_percentage);
num_iterations ← ⌈N/M⌉;
updated_weights ← {};
saved_weights ← {};
for index, param in net.parameters do
    saved_weights[index] ← param;
end
for net_patch_index in range(0, num_iterations) do
    start_index ← net_patch_index × M;
    end_index ← min(start_index + M, N);
    train_indices ← range(start_index, end_index)
    for index, param in net.parameters do
        param ← saved_weights[index];
        param.requires_grad ← bool(
            index ∈ train_indices
        );
    end
    train_step(net);
    save_indices ← range(start_index, end_index);
    for index, param in net.parameters do
        if index ∈ save_indices then
            updated_weights[index] ← param;
        end
    end
end
for index, param in net.parameters do
    param ← updated_weights[index];
end
return net;
```

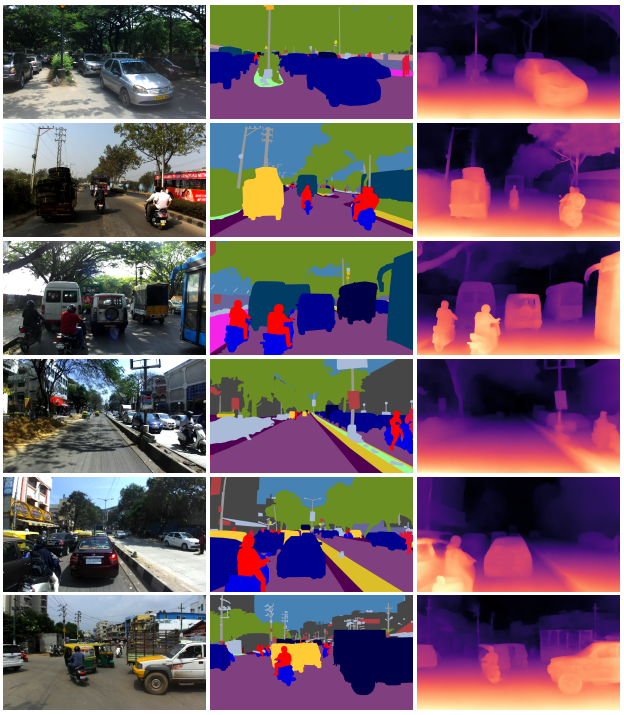

Figure 3. We use Depth Boosting to generate depth labels for the Indian Driving Dataset. We have the RGB frames on the left, segmentation map in the middle and our depth labels on the right

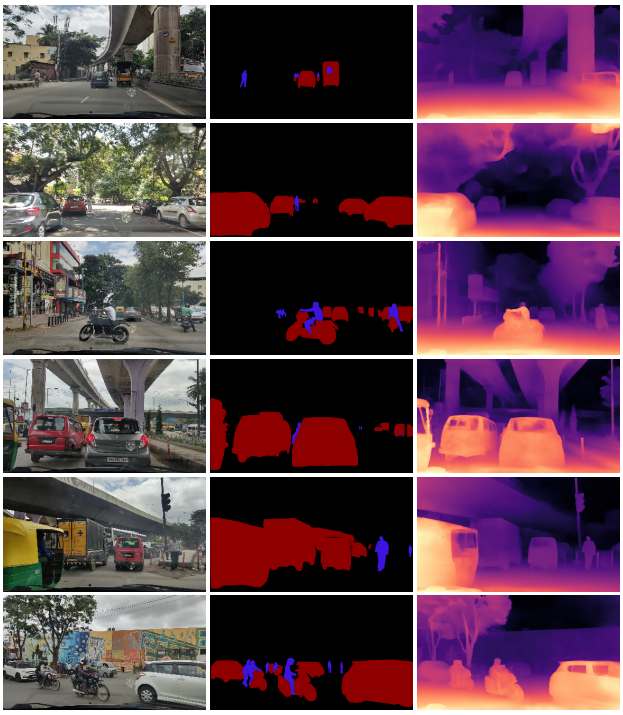

Figure 4. We use Semantic Segmentation auto-labelling to generate semantic labels for the Bengaluru Driving Dataset. We have the RGB frames on the left, our segmentation maps in the middle and depth labels on the right

semantic labels and we augment this dataset with depth labels. The Bengaluru Driving Dataset [10] has depth labels and we augment this dataset with 2D semantic labels.

**Depth Boosting.** Monocular depth estimation systems use a lot of the depth cues used by humans including occlusion boundaries, parallel lines, edges, vanishing points and the shape and size of objects. Altering the resolution of the image affects the clarity of these depth cues. While increasing the resolution can produce sharper results, feeding in smaller patches of the image fails. This happens when the window size shrinks to the point where there are no depth cues, which generates an inconsistent overall structure and may introduce low frequency artefacts. Taking inspiration from the depth boosting techniques [35, 34, 10], we select a content adaptive resolution $R_x$, beyond which the low frequency artefacts begin to hurt the overall structure of the image. By merging the disparity maps from the various resolutions, we are able to generate a high resolution disparity map with global consistency. As suggested by [35], we select $R_{20}$ as the high-resolution upper bound $R_x$, as their work shows that using resolutions higher than $R_{30}$ results in a decrease in performance due to the aforementioned artefacts. We use this method to generate disparity labels for the Indian Driving Dataset as shown in figure 3. The depth images on the left are colored by inverse depth (or disparity), such that pixels representing objects closer to the camera are brighter and those representing objects further away are darker.

**Semantic Segmentation auto-labelling.** To produce high resolution 2D semantic labels, we take inspiration from PointRend [4]. We take an image as input and produce a coarse intermediate segmentation map using an existing segmentation approach MaskRCNN [16]. This coarse map is gradually up-sampled using bi-linear interpolation and only the regions of the resized map with high uncertainty are refined. The uncertain regions typically include the boundaries of objects. The uncertain region is refined by a lightweight multi-layered perceptron. Its input is a feature vector which is extracted through interpolation from the feature maps, which intern has been computed by the base model. As shown in figure 4 we have auto-labelled vehicles in red and humans in blue.

## 4. Experiments

### 4.1. Experimental Setup

We train SOccDPT on a computer with an Intel i7-12700H (20 threads) and NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB VRAM and utilize PyTorch 2.1.0 [36]. With the goal of focusing performance in unstructured traffic, our network has been trained on the Indian Driving

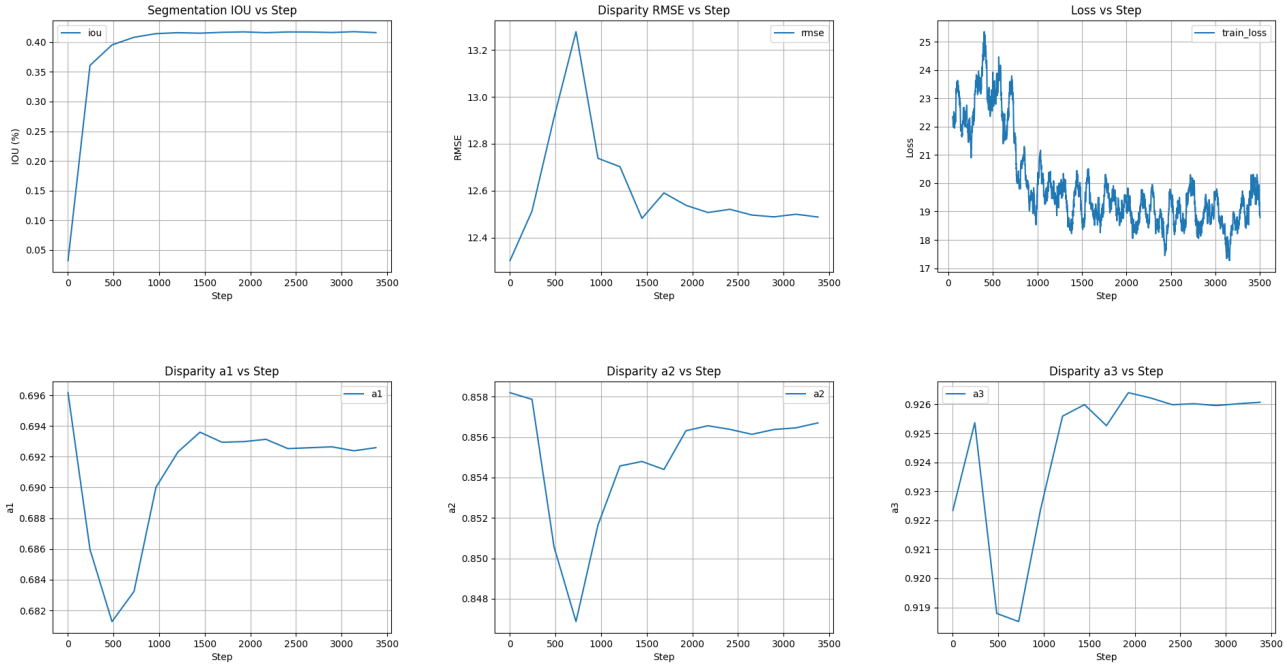


Figure 5. While training $SOccDPT_{V3}$ we start with the pre-trained depth backbone. As a result, the initial disparity metrics are good while the initial IoU score is under 5%. Within the first few epochs the IoU score starts growing steadily, and we observe a small spike in the RMSE as the depth head is adjusting to the changes made to accommodate the segmentation head. Note that these graphs show the training only up to 3,500 steps and we have trained our model to 10,500 steps

| Model | Dataset | $RMSE^{\downarrow}$ | $a1^{\uparrow}$ | $a2^{\uparrow}$ | $a3^{\uparrow}$ | $FPS^{\uparrow}$ (Hz) | $Parameters^{\downarrow}$ |
|---|---|---|---|---|---|---|---|
| $MiDaS_{v2.1}\ Small_{256}$[38] | BDD | 25.783 | 0.4742 | 0.7200 | 0.8298 | **327.7055** | **21.3M** |
| $MiDaS_{v3.1}\ LeViT_{224}$[14] | BDD | 24.387 | 0.5331 | 0.7325 | 0.8383 | 242.3232 | 50.6M |
| $MiDaS_{v3.1}\ Swin2_{T-256}$[28] | BDD | 23.325 | 0.5944 | 0.7816 | 0.8585 | 82.2656 | 41.7M |
| $MiDaS_{v3.0}\ DPT_{H-384}$[37] | BDD | 21.861 | 0.5527 | 0.7712 | 0.8558 | 13.4394 | 123.1M |
| $MiDaS_{v3.0}\ DPT_{L-384}$[37] | BDD | 13.36 | 0.6888 | 0.8514 | 0.9192 | 6.3142 | 344.1M |
| $SOccDPT_{V1}$ | BDD | 13.3782 | 0.6854 | 0.8442 | 0.9172 | 39.1141 | 84.3M |
| $SOccDPT_{V2}$ | BDD | 26.2383 | 0.4879 | 0.7181 | 0.8309 | 69.6503 | 42.3M |
| $SOccDPT_{V3}$ | BDD | **12.4075** | **0.6935** | **0.8588** | **0.9265** | 69.4733 | 42.3M |

Table 2. We compare SOccDPT's disparity metrics on the Bengaluru Driving Dataset, FPS and number of parameters with existing approaches

Dataset [44] and the Bengaluru Driving Dataset [10]. In table 3, we present the set of hyper-parameters which produce optimal results.

## 4.2. Ablation Study

In table 3, we present the set of hyper-parameters for SOccDPT's $V1$, $V2$ and $V3$. We observe that $V1$ produces good disparity and segmentation metrics while also being the largest network in terms of number of parameters and the slowest to run as shown in table 2. $SOccDPT_{V1}$ has two independent backbones which explains the larger number of parameters and increased inference time. While $SOccDPT_{V2}$ shows an improvement in speed and reduction in number of parameters, it takes a performance hit in terms of disparity and segmentation accuracy, as this network is being trained from scratch. $SOccDPT_{V2}$ introduces the common backbone which reduces the compute requirements, but since this entire network is being trained from scratch, it has no priors regarding either semantics or disparity estimation. We introduce this prior into $SOccDPT_{V3}$ by changing the architecture of $V2$ to allow us to load in pre-trained weights from the disparity backbone. As seen in figure 5, $SOccDPT_{V3}$ starts off with good RMSE scores for disparity estimation and poor

| Method | Dataset | Hyperparameters | | | | $RMSE^{\downarrow}$ | $a1^{\uparrow}$ $\delta < 1.25^1$ | $a2^{\uparrow}$ $\delta < 1.25^2$ | $a3^{\uparrow}$ $\delta < 1.25^3$ | $IoU^{\uparrow}$ (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | BS | PP | EP | LR | | | | | |
| $SOccDPT_{V1}$ | IDD | 12* | 0.5 | 0.5 | 0.00001 | 11.2353 | 0.7717 | 0.8991 | 0.9211 | **42.48** |
| | BDD | 12* | 0.5 | 0.5 | 0.00001 | 13.3782 | 0.6854 | 0.8442 | 0.9172 | 41.73 |
| $SOccDPT_{V2}$ | IDD | 6 | 0.5 | 0.95 | 0.00001 | 27.6473 | 0.5302 | 0.7084 | 0.8134 | 26.29 |
| | BDD | 6 | 0.5 | 0.95 | 0.00001 | 26.2383 | 0.4879 | 0.7181 | 0.8309 | 34.75 |
| $SOccDPT_{V3}$ | IDD | 6 | 0.5 | 0.95 | 0.0001 | **9.1473** | **0.7807** | **0.9009** | **0.9416** | 40.50 |
| | BDD | 6 | 0.5 | 0.95 | 0.0001 | 12.4075 | 0.6935 | 0.8588 | 0.9265 | 41.71 |

Table 3. **Ablation Study** SOccDPT's hyper-parameters and the metrics achieved. RMSE, a1, a2, a3 are disparity metrics and IoU is the segmentation metric. The hyper-parameters are batch size (BS), patch-wise percentage (PP), Encoder Percentage (EP) and learning rate (LR). Models with a * have had their two heads and backbone trained separately

IoU for segmentation, which is as expected. Through the course of training, the IoU steadily climbs. Initially, we see a spike in RMSE which comes back down over several epochs. $SOccDPT_{V3}$ has similar timing and memory characteristics when compared to $SOccDPT_{V2}$ as it is only a minor modification that allows us to load in the disparity backbone. But this small change allows $SOccDPT_{V3}$ to vastly outperform $SOccDPT_{V2}$ without requiring additional training data.

### 4.3. Comparison with Existing Methods

$SOccDPT_{V3}$'s performance exceeds existing disparity estimation approaches on unstructured traffic scenarios presented from the Bengaluru Driving Dataset. As shown in table 2, $SOccDPT_{V3}$ balances accuracy in disparity estimation, image segmentation while also maintaining a high FPS and keeping compute requirements low. While we observe that $MiDaS_{v2.1}$ $Small_{256}$[38] is much smaller and faster, this comes at the cost of accuracy. We show that our model is able to balance accuracy in terms of disparity estimation and semantic segmentation. We do this while performing in real time and keeping memory requirements low.

## 5. Conclusion

Existing disparity and segmentation approaches have come far, but do not specifically address the challenge in the autonomous vehicle context in unstructured traffic scenarios. We use depth boosting and semantic auto-labelling to build a self-supervised training pipeline, which can take videos as input and train a 3D semantic occupancy network. $SOccDPT$ uses a multi-headed Dense Transformer based architecture to take advantage of this self-supervised pipeline, to learn 3D semantic occupancy in the context of autonomous navigation in unstructured traffic. Our Patch-Wise training system allowed us to explore training with larger batch sizes which would not have been possible with memory constrained hardware. These models show potential in their ability to learn 3D semantic occupancy from monocular vision and operate at real time.

## References


[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[2] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022.

[3] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4192–4201, 2020.

[4] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. Pointly-supervised instance segmentation. 2021.

[5] Ran Cheng, Ryan Razani, Ehsan Taghavi, Thomas Li, and Bingbing Liu. (af) 2 -s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. pages 12542–12551, 06 2021.

[6] Xuelian Cheng, Yiran Zhong, Mehrtash Harandi, Yuchao Dai, Xiaojun Chang, Hongdong Li, Tom Drummond, and Zongyuan Ge. Hierarchical neural architecture search for deep stereo matching. *Advances in Neural Information Processing Systems*, 33, 2020.

[7] Yanhua Cheng, Xin Zhao, Rui Cai, Zhiwei Li, Kaiqi Huang, and Yong Rui. Semi-supervised multimodal deep learning for rgb-d object recognition. In *International Joint Conference on Artificial Intelligence*, 2016.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *arXiv preprint arXiv:2109.03805*, 2021.

[10] Aditya N Ganesh, Dhruval Pobbathi Badrinath, Harshith Mohan Kumar, Priya S, and Surabhi Narayan.

Octran: 3d occupancy convolutional transformer network in unstructured traffic scenarios. Spotlight Presentation at the Transformers for Vision Workshop, CVPR, 2023.

[11] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 09 2013.

[12] Clement Godard, Oisin Aodha, and Gabriel Brostow. Unsupervised monocular depth estimation with left-right consistency. 07 2017.

[13] Clément Godard, Oisin Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. 11 2019.

[14] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Herve Jegou, and Matthijs Douze. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12259–12269, October 2021.

[15] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. pages 2980–2988, 10 2017.

[17] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2302.07817*, 2023.

[18] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation, 2018.

[19] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs outputs, 2021.

[20] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. pages 3348–3356, 06 2020.

[21] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022.

[22] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers.

[24] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *CoRR*, abs/2109.13410, 2021.

[25] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation, 2020.

[26] Biyang Liu, Huimin Yu, and Yangqi Long. Local similarity pattern and cost self-reassembling for deep stereo matching networks, 2021.

[27] Shuo Liu, Wenrui Ding, Chunhui Liu, Yu Liu, and Hongguang Li. Ern: Edge loss reinforced semantic segmentation network for remote sensing images. *Remote Sensing*, 10(9):1339, 2018.

[28] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[30] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[31] Xiaoqiang Lu, Xuelong Li, and Lichao Mou. Semi-supervised multitask learning for scene recognition. *IEEE Transactions on Cybernetics*, 45(9):1967–1976, 2015.

[32] Yamin Mao, Zhihua Liu, Weiming Li, Yuchao Dai, Qiang Wang, Yun-Tae Kim, and Hong-Seok Lee. Uasnet: Uncertainty adaptive sampling network for deep stereo matching. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6291–6299, 2021.

[33] Armin Masoumian, Hatem A Rashwan, Saddam Abdulwahab, Julian Cristiano, M Salman Asif, and Domenec Puig. Gcndepth: Self-supervised monocular depth estimation based on graph convolutional network. *Neurocomputing*, 2022.

[34] Seyed Mahdi Hosseini Miangoleh. Boosting monocular depth estimation to high resolution. Master's thesis, Simon Fraser University, 2022.

[35] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9680–9689, 2021.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[37] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[38] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[39] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, 2020.

[40] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. pages 11135–11144, 06 2020.

[41] Luis Roldão, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 111–119, 2020.

[42] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pages 685–702. Springer, 2020.

[43] Simon Vandenhende, Stamatios Georgoulis, Wouter Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 01 2021.

[44] Girish Varma, Anbumani Subramanian, Anoop M. Namboodiri, Manmohan Chandraker, and C. V. Jawahar. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. *CoRR*, abs/1811.10200, 2018.

[45] Yufeng Wang, Yi-Hsuan Tsai, Wei-Chih Hung, Wenrui Ding, Shuo Liu, and Ming-Hsuan Yang. Semi-supervised multi-task learning for semantics and depth. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2663–2672, 2022.

[46] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1174, Los Alamitos, CA, USA, June 2021. IEEE Computer Society.

[47] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7483–7492, 2019.

[48] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023.

[49] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI Conference on Artificial Intelligence*, 2020.

[50] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *CoRR*, abs/2209.09385, 2022.

[51] Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Drinet++: Efficient voxel-as-point point cloud segmentation, 2021.

[52] Mingmin Zhen, Jinglu Wang, Lei Zhou, Shiwei Li, Tianwei Shen, Shang Jiaxiang, Tian Fang, and Long Quan. Joint semantic segmentation and boundary detection using iterative pyramid contexts. pages 13663–13672, 06 2020.

[53] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[54] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9934–9943, Los Alamitos, CA, USA, June 2021. IEEE Computer Society.