X-Distill: Improving Self-Supervised Monocular Depth via Cross-Task Distillation

Hong Cai[1]
hongcai@qti.qualcomm.com
Janarbek Matai[1]
jmatai@qti.qualcomm.com
Shubhankar Borse[1]
sborse@qti.qualcomm.com
Yizhe Zhang[2]
yizhe.zhang.cs@gmail.com
Amin Ansari[3]
amina@qti.qualcomm.com
Fatih Porikli[1]
fporikli@qti.qualcomm.com

[1] Qualcomm AI Research
San Diego, CA, USA
Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.
Second and third authors contributed equally to this work

[2] Nanjing University of Science and Technology
Nanjing, China
Work done at Qualcomm AI Research

[3] Qualcomm Technologies, Inc.
San Diego, CA, USA

Abstract

In this paper, we propose a novel method, X-Distill, to improve the self-supervised training of monocular depth via cross-task knowledge distillation from semantic segmentation to depth estimation. More specifically, during training, we utilize a pretrained semantic segmentation teacher network and transfer its semantic knowledge to the depth network. In order to enable such knowledge distillation across two different visual tasks, we introduce a small, trainable network that translates the predicted depth map to a semantic segmentation map, which can then be supervised by the teacher network. In this way, this small network enables the backpropagation from the semantic segmentation teacher's supervision to the depth network during training. In addition, since the commonly used object classes in semantic segmentation are not directly transferable to depth, we study the visual and geometric characteristics of the objects and design a new way of grouping them that can be shared by both tasks. It is noteworthy that our approach only modifies the training process and does not incur additional computation during inference. We extensively evaluate the efficacy of our proposed approach on the standard KITTI benchmark and compare it with the latest state of the art. We further test the generalizability of our approach on Make3D. Overall, the results show that our approach significantly improves the depth estimation accuracy and outperforms the state of the art.

1 Introduction

Accurate monocular depth estimation plays a critical role in 3D visual scene understanding and is of great importance for a variety of application domains, such as self-driving, AR/VR, and robotics. Thanks to the advancement of deep learning algorithms, recent years have seen considerable progress in this area [25]. However, training accurate deep learning models in a supervised manner requires high-quality (e.g., dense and correctly aligned) ground-truth depth maps, which are difficult and costly to obtain.

In order to overcome this challenge, self-supervision has emerged as a new paradigm for training monocular depth estimation models [9, 10, 45]. Since the inception of such self-supervised training, researchers have looked at various directions in order to further improve the depth estimation accuracy, such as designing more complex architectures [13, 16, 23], improving the photometric matching [15, 33], handling dynamic objects [2, 6, 11, 18], utilizing edge information [28, 30, 46], multi-task learning [4, 22, 29, 35, 41], and exploiting temporal information [27].

Given the importance of visual scene understanding for depth estimation, researchers have recently started to study how to utilize semantic segmentation information to improve accuracy. In [14, 19, 44], the authors use pretrained or jointly trained semantic segmentation networks to assist the depth network during both training and test. While such approaches can considerably improve accuracy, they incur significant extra computation during inference as they require running a separate and usually heavy-weight segmentation network. Another route is to incorporate the semantic information into the loss function, which only requires the extra computation of semantic information during training. One possible way is to include semantic segmentation as an auxiliary task, by co-training a semantic network and a depth network that share a set of layers [35]. Other papers compare the semantic segmentations on both the warped and actual versions of a frame, and enforce a consistency regularization [3, 39]. However, this requires running the segmentation network in every training iteration, which still incurs considerable overhead. In [46], the authors use the segmentation masks to explicitly regularize the edges on the depth map, but their approach requires semantic labels on the same dataset and introduces many additional hyper-parameters.

In this paper, we propose a novel cross-task knowledge distillation approach, **X-Distill**, to utilize semantic information to improve self-supervised monocular depth estimation. Given a pretrained semantic segmentation teacher network, our goal is to transfer the semantic knowledge from this teacher network to the depth network during training, in order to enhance the depth network's visual scene understanding capability. Note that our setting is different from the conventional knowledge distillation where the teacher and student networks share the same visual task. In our case, the outputs of the depth network and the semantic segmentation network are not directly comparable. In order to enable such cross-task distillation, we utilize a small neural network to connect segmentation and depth, by generating semantic segmentation based on the predicted depth. The resulting depth-based semantic segmentation is then supervised by the teacher network. The small network is trained together with the depth network and as such, allows backpropagation from the semantic segmentation teacher's supervision to the depth network.

In addition to enabling gradient flow across the two tasks, it is necessary to redesign the semantic classes such that they are compatible with the visual information in the depth map. In particular, the classes commonly used in semantic segmentation are usually too fine-grained for depth. For instance, road and sidewalk are typically treated as two separate classes in semantic segmentation. However, it is not necessary to treat them differently on the depth map since both of them are on the ground surface and have highly similar depth variation patterns in the field of view. As such, we regroup the objects based on their visual and geometric characteristics. This allows the depth network to distill the key depth-relevant semantic information, without introducing unnecessary difficulties to the learning process.

We next summarize our main contributions as follows:

- We propose a novel method, X-Distill, to exploit semantic information to improve self-supervised monocular depth estimation. X-Distill enables the depth network to distill semantic knowledge in a cross-task manner from a segmentation teacher network during training. At inference time, the depth network then runs in a standalone manner, without requiring extra computation to process/generate semantic information.

- In order to make the semantic segmentation knowledge compatible with the visual information in depth, we regroup the semantic classes based on the visual and geometric characteristics of the objects. This allows the depth network to distill the key semantic knowledge while removing the unnecessary complexities in the learning.
- We evaluate our proposed approach on KITTI and Make3D, and compare it with the state of the art. We further conduct extensive ablation studies on our method. Overall, our proposed approach achieves considerably more accurate depth estimation, e.g., outperforming [9] by 14% on KITTI (in terms of squared relative error).

2 Related Work

Self-Supervised Monocular Depth Estimation: Due to the difficulty of collecting dense, high-quality ground-truth depth maps, researchers have proposed self-supervised training to obtain monocular depth estimation models. Such self-supervision leverages the geometric relationship among neighboring video frames [9, 45] or between the left and right cameras in a stereo setting [10]. While these methods provide a new way to train a depth network without labels, factors such as moving objects, occlusion, poor lighting, and low texture can considerably degrade their performance.

Utilizing Semantic Information for Depth Estimation: Given the high correlation between semantic and depth information, researchers have studied how to incorporate semantic information to improve depth accuracy. One way is to run an additional (sub)network to generate semantic information at inference time, which can be fed to the depth network [14, 19, 44]. While this approach can considerably improve the depth estimation performance, it incurs significantly more computation. Other works include new loss functions during training, either via multi-task training [35] or by enforcing segmentation consistency between the warped and real images [3, 24, 39]. These methods do not require extra semantic computation during test, but require running a semantic network at every training iteration, which still generates a considerable overhead.

Knowledge Distillation: Knowledge Distillation is usually used to transfer the knowledge from a more complex model to a smaller model, where both of them are designed for the same visual task [12]. Few papers have looked at knowledge distillation across two different visual tasks, e.g., classification tasks with non-overlapping classes [40], classification and text-to-image synthesis [42], RGB-based depth estimation and depth super resolution [34]. None of the existing works has studied cross-task distillation from semantic segmentation to depth and we show how to enable it in this paper.

3 Proposed Method

In this section, we present our novel take, X-Distill, on utilizing semantic segmentation to improve self-supervised monocular depth estimation, through cross-task distillation. In order to transfer the relevant knowledge from a semantic segmentation teacher network to the depth network during training, we use a small network to translate depth to segmentation, thus enabling gradient flow across the two visual tasks. In addition, we redesign the semantic classes to make them compatible with the visual information contained in depth.

3.1 Self-Supervised Monocular Depth Estimation

We utilize self-supervision to train a monocular depth estimation model, based on single-view video sequences [9, 45].

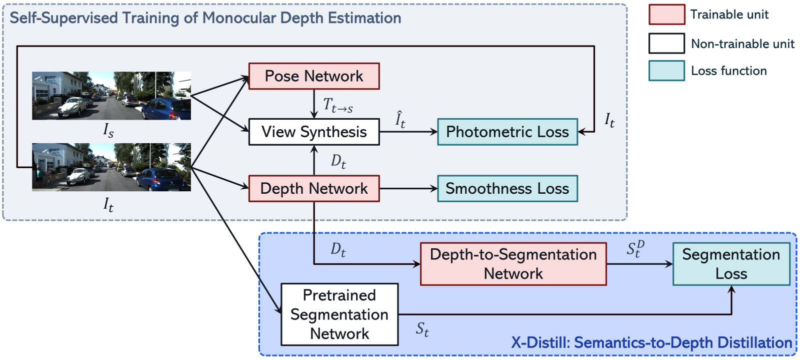

Figure 1: Overview of our proposed X-Distill approach. The gray block describes the self-supervised training of a monocular depth network based on single-view videos. The blue block illustrates our proposed cross-task semantics-to-depth distillation. By utilizing a trainable depth-to-segmentation network to translate predicted depth to segmentation, we enable cross-task knowledge transfer from the pretrained segmentation teacher network to the depth network during training. In addition, we regroup the semantic classes such that they become compatible with the visual information in depth.

Geometric Modeling: Consider two neighboring video frames, I_t and I_s. Suppose that pixel $p_t \in I_t$ and pixel $p_s \in I_s$ are two different views of the same point of an object, then p_t and p_s are related geometrically as follows:

$$d(p_s)h(p_s) = \mathbf{K}(\mathbf{R}_{t\to s}\mathbf{K}^{-1}d(p_t)h(p_t) + \mathbf{t}_{t\to s}), \tag{1}$$

where $h(p) = [h, w, 1]$ denotes the homogeneous coordinates of a pixel p with h and w being its vertical and horizontal positions on the image, $d(p)$ is the depth at p, $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the camera intrinsic matrix, and $\mathbf{T}_{t\to s} = [\mathbf{R}_{t\to s}|\mathbf{t}_{t\to s}] \in \mathbb{R}^{3\times 4}$ is the 6DoF relative camera motion/pose from t to s, with $\mathbf{R}_{t\to s} \in \mathbb{R}^{3\times 3}$ and $\mathbf{t}_{t\to s} \in \mathbb{R}^{3\times 1}$ being the rotation matrix and translation vector.

Given the depth map of I_t, denoted by D_t, and the relative camera pose from I_t to I_s, we can synthesize I_t from I_s based on Eq. 1, assuming that the 3D points captured in I_t are also present in I_s. We denote the synthesized/warped version of I_t as $\widehat{I_t}$.

Self-Supervised Training: Suppose that the depth map and the relative camera pose are provided by a depth network and a pose network, respectively. By minimizing the difference between the warped and actual versions of I_t, we can train these two networks. A common photometric loss function for comparing I_t and $\widehat{I_t}$ is given by

$$\mathcal{L}_{\text{PH}}(I_t, \widehat{I_t}) = \alpha\|I_t - \widehat{I_t}\|_1 + (1-\alpha)\frac{1-\text{SSIM}(I_t, \widehat{I_t})}{2}, \tag{2}$$

where $\|\cdot\|_1$ denotes the $\mathcal{L}_1$ norm and SSIM is the Structural Similarity Index Measure [38]. Note that $\mathcal{L}_{\text{PH}}$ is computed in a per-pixel manner.

It is common to further include a smoothness regularization to prevent drastic variations in the predicted depth map. Furthermore, in practice, not all the 3D points in I_t can be found in I_s, due to occlusion and objects (partially) moving out of the frame. Some objects can also be moving (e.g., cars), which is not considered in the geometric model of Eq. 1. In order to correctly measure the photometric loss and train the networks, it is a common practice to mask out the pixel points that violate the geometric model (see [9] for more details on the masking techniques). Fig. 1 (gray block) illustrates the self-supervised training scheme of a monocular depth network.

3.2 Cross-Task Distillation from Semantics to Depth

Consider a depth network, f_D, and a pretrained semantic segmentation network, f_S. Our goal is to transfer the knowledge of the teacher network, f_S, to the depth network, f_D. However, unlike conventional knowledge distillation where teacher and student networks are used for the same visual task, f_D and f_S are used for two different tasks, and their outputs are not directly comparable. In other words, given an input, we cannot directly measure the difference between the outputs of f_D and f_S in order to generate a loss to train f_D.

As such, we utilize a Depth-to-Segmentation (D2S) neural network, h_{D2S}, to translate depth to semantic segmentation. Given the segmentation map generated from the predicted depth map, we are now able to construct a segmentation loss to distill semantic knowledge from f_S to f_D. More formally, the new loss term is given as follows:

$$\mathcal{L}_{D2S}(S_t^{\mathrm{D}}, S_t) = \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{\mathcal{L}_{\mathrm{CE}}(S_t^{\mathrm{D}}(i,j), S_t(i,j))}{HW}, \tag{3}$$

where $S_t^D = h_{D2S}(f_D(I_t))$ is the semantic segmentation map generated by h_{D2S} based on the predicted depth map $D_t = f_D(I_t)$, S_t is the semantic segmentation output generated by the semantic segmentation teacher network, $\mathcal{L}_{\mathrm{CE}}$ denotes the cross-entropy loss, and H and W are the height and width of the input image.[1]

The total loss is then given by

$$\mathcal{L}_{\mathrm{Total}} = \sum_{k=1}^{N_s} \mathcal{L}_{PH,k} + \sum_{k=1}^{N_s} \lambda_{SM,k} \mathcal{L}_{SM,k} + \lambda_{D2S} \mathcal{L}_{D2S}, \tag{4}$$

where the self-supervised depth loss is computed over N_s scales, $\mathcal{L}_{PH,k}$ is the photometric loss at the k^{th} scale, $\lambda_{SM,k}$ and $\mathcal{L}_{SM,k}$ are the weight and loss for the smoothness regularization at the k^{th} scale, and λ_{D2S} is the weight of the cross-task distillation loss, $\mathcal{L}_{D2S}$.

It can be seen that during training, h_{D2S} is jointly trained with the depth network. This makes it possible for the pretrained teacher network to provide semantic supervision to the depth network, by backpropagating through h_{D2S}. Our proposed approach is illustrated in Fig. 1, with the semantics-to-depth distillation module highlighted in the blue block.

For the depth-to-segmentation network, h_{D2S}, we adopt a small architecture. More specifically, h_{D2S} consists of two 3×3 convolutional layers, each followed by a BatchNorm layer and a ReLu layer, as well as a pointwise convolutional layer at the end which outputs the segmentation. Note that the h_{D2S} should not be too complex, since a deeper network would take over too much of the learning load and weaken the knowledge flow to the depth network. As we shall see in our experiments in Sec. 4, while using a deeper h_{D2S} can still increase the accuracy of the depth network, the improvement is not as significant as that by using our proposed smaller h_{D2S}.

Once the training is finished, the depth network can then run in a standalone manner, without requiring any extra computation of semantic information during inference. Furthermore, our proposed distillation approach only adds a small amount of computation to training. More specifically, the segmentation maps from the teacher network only need to be computed once and the additional forward/backward passes are cheap since h_{D2S} is small.

3.3 Depth-Compatible Grouping of Semantic Classes

Semantic segmentation usually contains much more fine-grained visual recognition information that is not present in the depth map. For instance, road and sidewalk are typically treated

[1] Note that we can include a "background" class for the pretrained segmentation model (which is a common practice). This will allow us to ignore pixels that are not of interest when computing the distillation loss of Eq. 3.

Object Groups	Foreground vs. Background	Shape of Bounding Box	Location in 3D Space
Thin objects	Foreground	Thin rectangular	Above ground plane
People and vehicles	Foreground	Rectangular	Above ground plane
Background objects	Background	Not boundable	Above ground plane
Ground	Background	Not boundable	On ground plane

Table 1: Depth-compatible semantic class grouping for outdoor scenes.

as two different semantic classes. However, the depth map does not contain such classification information as both road and sidewalk are on the ground plane and have similar depth variations. As a result, it is not necessary to differentiate them on the depth map. On the other hand, the depth map does contain the information for differentiating certain classes. For instance, a road participant (e.g., pedestrian, vehicle) can be easily separated from the background (e.g., road, building) given the different patterns of their depth values.

As such, is it necessary to reconsider the grouping of semantic classes, such that the key semantic information is preserved while the unnecessary complexity is removed from the distillation. Table 1 summarizes our new grouping, which results in four groups.[2] In the first two groups, we have objects in the foreground. The respective foreground objects in these two groups are then further differentiated based on their shapes, where the first group contains thin structures, e.g., traffic lights/signs (including the poles), and the second group consists of people and vehicles which are of more general shapes. The third and fourth groups then contain the background objects, such as buildings, vegetation, road, and sidewalk. We further separate the ground plane (e.g., road and sidewalk) from the other background objects.

4 Experiments

In this section (and also in the supplementary file), we present a comprehensive performance analysis on our proposed X-Distill approach and compare with the current state of the art. We furthermore conduct in-depth ablation studies on various aspects of our method.

4.1 Experiment Setup

Datasets: We evaluate depth estimation on KITTI [8] using the standard Eigen split [7], with two input resolutions, 640×192 and 1024×320. Following [45], we remove the static frames in the training set. There are 39,810, 4,424, and 697 samples for training, validation, and test.

[2] We focus on outdoor scenes in this paper and will consider an extension to indoor scenes as part of future work.

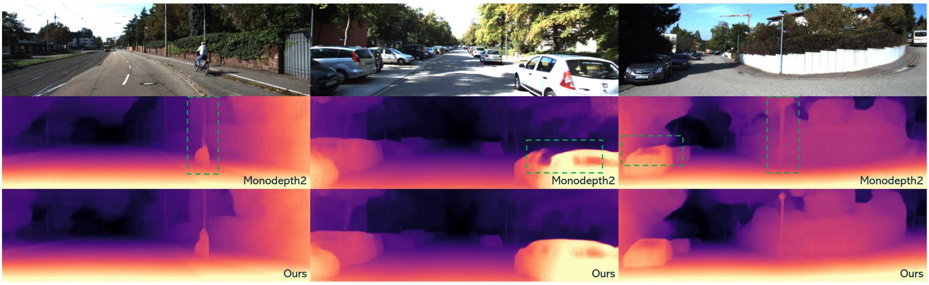

Figure 2: Depth estimation on three sample images. The second row shows the estimated depth maps by Monodepth2 [9] and the third row shows the depth maps by our proposed X-Distill approach. It can be seen that our method provides more accurate depth estimation. The green boxes indicate sample regions where our method considerably improves the estimation quality.

We use Cityscapes [5] training and validation sets to train the segmentation teacher network. We further use Make3D [31, 32] to evaluate the generalizability of our KITTI-trained model.

Grouping of Semantic Classes: We group the Cityscapes classes according to our proposed scheme in Table 1, such that they are compatible with the depth information. More specifically, we have 1) thin objects: poles and traffic lights/signs, 2) people and vehicles: persons, riders, cars, trucks, buses, motorcycles, bicycles, and trains, 3) background objects: buildings, walls, fences, vegetation, terrain, and sky, and 4) ground: road and sidewalk.

Networks: For the depth network and the pose network, we use the ResNet-50 (RN50)-based models in [9]. The semantic segmentation teacher network is an HRNet [37] with OCR [43] and InverseForm [1]. It has an mIoU of 85.6% on Cityscapes test set. During the self-supervised training of the depth network, this segmentation network is frozen.

Hyperparameters: For the self-supervised part, we follow the hyperparameter setting in [9]. For the semantics-to-depth distillation loss, $\mathcal{L}_{D2S}$, we linearly increase its weight from 0 to 0.005 during training. As we shall see in the ablation studies, this linear schedule can improve the training as compared to using a constant weight.

Evaluation Metrics: We use the commonly used error metrics to evaluate the depth estimation performance, including the Absolute Relative Error (Abs Rel), Squared Relative Error (Sq Rel), Root Mean Squared Error (RMSE), and the RMSE of the log of the depth values. In addition, we use the classification metrics, δ_1, δ_2, and δ_3, which measure whether the ratio between the ground-truth and predicted depth values is within a certain interval around 1. Mathematical definitions of these metrics can be found in the supplementary file.

4.2 Results

We extensively compare our proposed approach with the latest state of the art (SOTA) on KITTI, including methods that 1) use more complex architectures [13, 16, 23], 2) require additional computation of semantic information during inference [14], 3) utilize semantic information during training and do not incur extra computation during test [18], 4) propose better photometric matching [15, 33], 5) utilize multiple frames [27], and 6) perform multi-task learning [35]. Note that we do not consider pretraining/online finetuning of the depth network or applying post-processing on the predicted depth maps. We also analyze both the depth estimation accuracy and computation efficiency of the methods. Furthermore, we test our KITTI-trained model on Make3D and compare it with the related SOTA to evaluate generalizability. Finally, we perform extensive ablation studies on our proposed approach.

4.2.1 Performance Evaluation

Evaluation on KITTI: Table 2 shows the evaluation results on KITTI and comparison with the latest SOTA methods. It can be seen that our proposed X-Distill approach performs the best for most of the metrics. When our approach does not achieve the top-1 result, it is very close to the best number. Fig. 2 shows sample prediction results of our proposed approach as compared to those by Monodepth2. It can be seen that Monodepth2 can predict inconsistent depth values on an object, which visually appear as missing parts on the depth map (e.g., see the missing upper part of the car in Fig. 2 (middle)). On the other hand, our approach provides more accurate and semantically more structured depth maps, thanks to its ability to better understand the semantics of the scene. For instance, it generates more structurally complete depth estimations for the biker in Fig. 2 (left) and for the cars in the middle and right figures (as indicated by the green boxes). Moreover, our approach is also able to capture the thin structures better. For instance, in Fig. 2 (right), our model is able to generate a more clear depth estimation over the lamp post. This is because the thin objects are grouped as a

Method	Resolution	*Lower is Better*				*Higher is Better*		
		Abs Rel	**Sq Rel**	**RMSE**	**RMSE$_{\text{Log}}$**	δ_1	δ_2	δ_3
Monodepth2 [9] (RN18)	640 × 192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Monodepth2 [9] (RN50)	640 × 192	0.110	0.831	4.642	0.187	0.883	0.962	0.982
Tosi et al. [35]†	640 × 192	0.120	0.792	4.750	0.191	0.856	0.958	**0.984**
PackNet-SfM [13]	640 × 192	0.111	$\underline{0.785}$	$\underline{4.601}$	0.189	0.878	0.960	0.982
Johnston et al. [16] (RN18)	640 × 192	0.111	0.941	4.817	$\underline{0.185}$	0.885	0.961	0.981
Johnston et al. [16] (RN101)	640 × 192	**0.106**	0.861	4.699	$\underline{0.185}$	**0.889**	0.962	0.982
HR-Depth [23]	640 × 192	$\underline{0.109}$	0.792	4.632	$\underline{0.185}$	0.884	0.962	$\underline{0.983}$
Guizilini et al. [14]† (RN18)	640 × 192	0.117	0.854	4.714	0.191	0.877	0.959	0.981
Guizilini et al. [14]† (RN50)	640 × 192	0.113	0.831	4.663	0.189	0.878	**0.971**	0.982
Klingner et al. [18]† (RN18)	640 × 192	0.113	0.835	4.693	0.191	0.879	0.961	0.981
Klingner et al. [18]† (RN50)	640 × 192	0.112	0.833	4.688	0.190	0.884	0.961	0.981
DiPE [15]	640 × 192	0.112	0.875	4.795	0.190	0.880	0.960	0.981
Patil et al. [27]	640 × 192	0.111	0.821	4.650	0.187	0.883	0.961	0.982
X-Distill (ours)†	640 × 192	**0.106**	**0.777**	**4.580**	**0.184**	$\underline{0.888}$	$\underline{0.963}$	0.982
Monodepth2 [9] (RN18)	1024 × 320	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Tosi et al. [35]†	1024 × 320	0.118	0.748	4.608	0.186	0.865	0.961	$\underline{0.985}$
PackNet-SfM [13]	1280 × 384	0.107	0.802	4.538	0.186	0.886	0.962	0.981
HR-Depth [23]	1024 × 320	0.106	0.755	4.472	0.181	$\underline{0.892}$	**0.966**	0.984
Klingner et al. [18]† (RN18)	1280 × 384	0.107	0.768	$\underline{4.468}$	0.186	0.891	0.963	0.982
Shu et al. [33]	1024 × 320	$\underline{0.104}$	$\underline{0.729}$	4.481	**0.179**	0.893	$\underline{0.965}$	**0.987**
X-Distill (ours)†	1024 × 320	**0.102**	**0.698**	**4.439**	$\underline{0.180}$	**0.895**	$\underline{0.965}$	0.983

Table 2: Performance evaluation on KITTI Eigen split. For methods that report performance for multiple models, we use the encoder to differentiate them (e.g., RN18 vs. RN50). Note that two architectures can be very different even if they use the same encoder (e.g., Monodepth2 [9] vs. Johnston et al. [16]). For each metric, the best (second best) results are in bold (underlined). We use † to indicate methods that utilize semantic information during training.

class in the semantics-to-depth distillation, which encourages the depth network to learn to recognize these structures.

Accuracy vs. Computation Efficiency: Fig. 3 shows the accuracy, in terms of squared relative error, and the efficiency, in terms of GMAC (Multiply-Accumulate Operations in 10^9), of our proposed approach and the SOTA methods on KITTI.[3] It can be seen that our trained model is able to achieve smaller depth estimation errors while using the same or less computation. We further show the performance of applying our cross-task distillation to an RN18-based model from [9]. It can be seen that our method allows this smaller network to achieve an accuracy similar to PackNet (which uses 20× more computation).

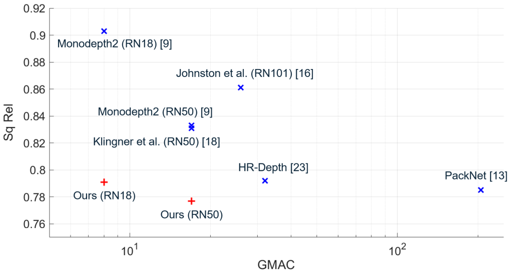

Figure 3: Accuracy (in squared relative error) vs. computation efficiency (in GMAC log-scale).

Depth Estimation on Center and Surrounding Areas: Since KITTI images are acquired with a wide-angle lens, we further evaluate the depth estimation performance on center and surrounding areas in the image. Specifically, we horizontally divide each image into 3 equal sections. The middle part is considered the center area and the left and right parts are surrounding areas. It can be seen in Table 3 that the depth estimation is much more accurate

[3] For Johnston et al. [16], the GMAC shown in Fig. 3 is a lower bound which only includes the RN101 encoder's computation since their self-attention and discrete disparity volume implementation is not publicly available.

in the center area, for both Monodepth2 and X-Distill, since surrounding areas suffer from lens distortion/rectification artifacts. We note that for both areas, our proposed X-Distill consistently provides more accurate depth estimation as compared to Monodepth2.

Method	*Lower is Better*				*Higher is Better*		
	Abs Rel	**Sq Rel**	**RMSE**	**RMSE$_{\text{Log}}$**	δ_1	δ_2	δ_3
Over Center Areas							
Monodepth2 (RN50)	0.061	0.059	0.568	0.077	0.978	0.996	0.999
X-Distill (ours)	0.056	0.048	0.526	0.072	0.982	0.997	0.999
Over Surrounding Areas							
Monodepth2 (RN50)	0.135	1.228	5.700	0.220	0.839	0.945	0.974
X-Distill (ours)	0.125	1.054	5.368	0.210	0.852	0.950	0.976

Table 3: Performance evaluation on center and surrounding image areas.

Generalizability on Make3D: We evaluate the generalizability of our KITTI-trained model on Make3D (following the test setup in [9]). It can be seen in Table 4 that our model significantly outperforms other SOTA self-supervised methods on this dataset.

Method	Supervision	*Lower is Better*			
		Abs Rel	**Sq Rel**	**RMSE**	**RMSE$_{\text{Log}}$**
Karsch [17]	GT	0.428	5.079	8.389	0.149
Liu [21]	GT	0.475	6.562	10.05	0.165
Laina [20]	GT	0.204	1.840	5.683	0.084
Monodepth [10]	S	0.544	10.94	11.760	0.193
Zhou et al. [45]	M	0.383	5.321	10.470	0.478
DDVO [36]	M	0.387	4.720	8.090	0.204
Monodepth2 [9] (RN18)	M	0.322	3.589	7.417	0.163
X-Distill (ours)†	M	0.308	3.122	7.015	0.158

Table 4: Performance evaluation on Make3D. GT indicates that the method is trained with ground-truth Make3D depth annotations, S indicates self-supervised training using KITTI stereo data, and M indicates self-supervised training using KITTI single-view videos. We use † to indicate methods that utilize semantic information during training.

4.2.2 Ablation Studies

Grouping Semantic Classes: In addition to our proposed grouping shown in Table 1, we further test the baseline of using the original 19 Cityscapes classes without regrouping, as well as a more aggressive grouping method that only considers foreground and background objects. As shown in Table 5, while the other two grouping baselines can also improve the depth estimation, the improvements are not as large as compared to our proposed method.

Categorization Scheme	*Lower is Better*				*Higher is Better*		
	Abs Rel	**Sq Rel**	**RMSE**	**RMSE$_{\text{Log}}$**	δ_1	δ_2	δ_3
Monodepth2 (RN50)	0.110	0.831	4.642	0.187	0.883	0.962	0.982
Fore/Back-ground (2)	0.108	0.798	4.663	0.187	0.886	0.962	0.982
Proposed scheme (4)	0.106	0.777	4.580	0.184	0.888	0.963	0.982
Cityscapes classes (19)	0.110	0.806	4.619	0.184	0.882	0.963	0.983

Table 5: Performance of different ways of grouping the semantic classes.

Complexity of Depth-to-Segmentation Network: As discussed in Sec. 3.2, the D2S network should be of a proper complexity such that it does not take away the learning from the depth network. As shown in Table 6, by using a more complex D2S network (about 2× larger), the depth network gains a smaller improvement. We further test a baseline using a simple D2S network with one-layer pointwise convolution. This baseline does not perform well as the corresponding D2S network is too simple to translate depth to segmentation.

Weighting Segmentation Loss: In our proposed approach, we adopt a linear weighting schedule to combine the segmentation distillation loss with the self-supervised depth loss. It can be seen in Table 7, the linearly scheduled weight allows the depth network to achieve a higher depth estimation accuracy as compared to using a constant weight. We further vary

Depth-to-Segmentation	*Lower is Better*				*Higher is Better*		
Network	**Abs Rel**	**Sq Rel**	**RMSE**	**RMSE_{Log}**	δ_1	δ_2	δ_3
Monodepth2 (RN50)	0.110	0.831	4.642	0.187	0.883	0.962	0.982
2 3×3-Conv + 1 Pointwise Conv	0.106	0.777	4.580	0.184	0.888	0.963	0.982
4 3×3-Conv + 1 Pointwise Conv	0.108	0.786	4.615	0.185	0.886	0.963	0.982
1 Pointwise Conv	0.110	0.840	4.683	0.188	0.885	0.961	0.981

Table 6: Depth-to-Segmentation network.

the final weight by ±20% and the results show that our proposed method is not very sensitive to the exact value of the weight.

Weighting of	*Lower is Better*				*Higher is Better*		
Distillation Loss	**Abs Rel**	**Sq Rel**	**RMSE**	**RMSE_{Log}**	δ_1	δ_2	δ_3
Monodepth2 (RN50)	0.110	0.831	4.642	0.187	0.883	0.962	0.982
Constant: 0.0050	0.109	0.810	4.637	0.185	0.887	0.963	0.983
Linear: 0 - 0.0040	0.107	0.779	4.632	0.185	0.887	0.962	0.982
Linear: 0 - 0.0045	0.107	0.751	4.553	0.185	0.884	0.963	0.983
Linear: 0 - 0.0050	0.106	0.777	4.580	0.184	0.888	0.963	0.982
Linear: 0 - 0.0055	0.108	0.795	4.606	0.183	0.887	0.963	0.983
Linear: 0 - 0.0060	0.107	0.775	4.580	0.184	0.888	0.963	0.983

Table 7: Weighting of segmentation loss.

Applying X-Distill to Different Architectures: We apply our proposed approach to different depth networks, e.g., Monodepth2 [9] with different encoders and HR-Depth [23]. Specifically, for the encoder of Monodepth2, in addition to RN18 and RN50 that are used in the original paper, we also employ a recent backbone, DONNA, which is optimized for mobile processors via neural architecture search [26]. This will demonstrate the efficacy of our method for practical mobile use cases. As can be seen in Table 8, our proposed X-Distill considerably improves the depth estimation accuracy for all these different depth networks.

Architectures	*Lower is Better*				*Higher is Better*		
	Abs Rel	**Sq Rel**	**RMSE**	**RMSE_{Log}**	δ_1	δ_2	δ_3
Monodepth2 [9] (RN18)	0.115	0.903	4.863	0.193	0.877	0.959	0.981
+ X-Distill	**0.111**	**0.791**	**4.772**	**0.188**	0.874	**0.960**	**0.983**
Monodepth2 [9] (RN50)	0.110	0.831	4.642	0.187	0.883	0.962	0.982
+ X-Distill	**0.106**	**0.777**	**4.580**	**0.184**	**0.888**	**0.963**	0.982
Monodepth2 [9] (DONNA)	0.115	0.916	4.827	0.193	0.879	0.960	0.981
+ X-Distill	**0.109**	**0.772**	**4.678**	**0.188**	**0.884**	**0.962**	**0.982**
HR-Depth [23]	0.109	0.792	4.632	0.185	0.884	0.962	0.983
+ X-Distill	**0.108**	**0.755**	**4.579**	**0.184**	0.884	**0.963**	0.983

Table 8: Applying our semantics-to-depth distillation to different depth networks. For each model, improved numbers by using X-Distill are highlighted in bold.

5 Conclusions

In this paper, we presented a novel cross-task distillation approach, X-Distill, to improve the self-supervised training of monocular depth by transferring semantic knowledge from a teacher segmentation network to the depth network. In order to enable such cross-task distillation, we utilized a small, trainable network that translates the predicted depth map to a semantic segmentation map, which the semantic teacher network can then supervise. This enables the backpropagation from the semantic teacher's supervision to the depth network during training. We further studied the visual and geometric characteristics of the objects and designed a new way of grouping them that can be shared by both tasks. We evaluated our proposed approach on KITTI and Make3D, and conducted extensive ablation studies. The results show that by training with the proposed cross-task distillation, we can significantly improve the depth estimation accuracy and outperform the state of the art without incurring additional computation during inference.

References

[1] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[2] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[3] Po-Yi Chen, Alexander H Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2624–2632, 2019.

[4] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1004–1005, 2020.

[7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[9] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[11] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019.

[12] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, pages 1–31, 2021.

[13] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3D packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[14] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *Proceedings of the International Conference on Learning Representations*, 2020.

[15] Hualie Jiang, Laiyan Ding, Zhenglong Sun, and Rui Huang. Dipe: Deeper into photometric errors for unsupervised learning of depth and ego-motion from monocular videos. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10061–10067, 2020.

[16] A. Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[17] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[18] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *Proceedings of the European Conference on Computer Vision*, 2020.

[19] Varun Ravi Kumar, Marvin Klingner, Senthil Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mader. SynDistNet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[20] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the International Conference on 3D Vision*, pages 239–248, 2016.

[21] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014.

[22] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3D holistic understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2624–2641, 2019.

[23] X. Lyu, L. Liu, M. Wang, X. Kong, L.a Liu, Y. Liu, X. Chen, and Y. Yuan. HR-Depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[24] Yue Meng, Yongxi Lu, Aman Raj, Samuel Sunarjo, Rui Guo, Tara Javidi, Gaurav Bansal, and Dinesh Bharadia. Signet: Semantic instance aided unsupervised 3D geometry perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[25] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 2021.

[26] Bert Moons, Parham Noorzad, Andrii Skliar, Giovanni Mariani, Dushyant Mehta, Chris Lott, and Tijmen Blankevoort. Distilling optimal neural networks: Rapid search in diverse spaces. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[27] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020.

[28] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *Asian Conference on Computer Vision*, pages 298–313. Springer, 2018.

[29] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019.

[30] Faraz Saeedan and Stefan Roth. Boosting monocular depth with panoptic segmentation maps. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3853–3862, 2021.

[31] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances on Neural Information Processing Systems*, pages 1161–1168, 2005.

[32] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

[33] C. Shu, K. Yu, Z. Duan, and K. Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *Proceedings of the European Conference on Computer Vision*, 2020.

[34] Baoli Sun, Xinchen Ye, Baopu Li, Haojie Li, Zhihui Wang, and Rui Xu. Learning scene structure guidance via cross-task knowledge transfer for single depth super-resolution. *arXiv preprint arXiv:2103.12955*, 2021.

[35] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[36] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[37] J. Wang, K. Sun, T. Cheng, B.i Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[39] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

[40] Han-Jia Ye, Su Lu, and De-Chuan Zhan. Distilling cross-task knowledge via relationship matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12396–12405, 2020.

[41] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Mingkuan Yuan and Yuxin Peng. Ckd: Cross-task knowledge distillation for text-to-image synthesis. *IEEE Transactions on Multimedia*, 22(8):1955–1968, 2019.

[43] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Proceedings of the European Computer Vision Conference*, 2020.

[44] Min Yue, Guangyuan Fu, Ming Wu, and Hongqiao Wang. Semi-supervised monocular depth estimation based on semantic supervision. *Journal of Intelligent & Robotic Systems*, 100:455–463, 2020.

[45] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[46] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13116–13125, 2020.