

# PolyMaX: General Dense Prediction with Mask Transformer

Xuan Yang\* Liangzhe Yuan Kimberly Wilber Astuti Sharma Xiuye Gu  
Siyuan Qiao Stephanie Debats Huisheng Wang Hartwig Adam  
Mikhail Sirotenko Liang-Chieh Chen†  
Google Research

## Abstract

Dense prediction tasks, such as semantic segmentation, depth estimation, and surface normal prediction, can be easily formulated as per-pixel classification (discrete outputs) or regression (continuous outputs). This per-pixel prediction paradigm has remained popular due to the prevalence of fully convolutional networks. However, on the recent frontier of segmentation task, the community has been witnessing a shift of paradigm from per-pixel prediction to cluster-prediction with the emergence of transformer architectures, particularly the mask transformers, which directly predicts a label for a mask instead of a pixel. Despite this shift, methods based on the per-pixel prediction paradigm still dominate the benchmarks on the other dense prediction tasks that require continuous outputs, such as depth estimation and surface normal prediction. Motivated by the success of DORN and AdaBins in depth estimation, achieved by discretizing the continuous output space, we propose to generalize the cluster-prediction based method to general dense prediction tasks. This allows us to unify dense prediction tasks with the mask transformer framework. Remarkably, the resulting model PolyMaX demonstrates state-of-the-art performance on three benchmarks of NYUD-v2 dataset. We hope our simple yet effective design can inspire more research on exploiting mask transformers for more dense prediction tasks. Code and model will be made available<sup>1</sup>.

## 1 Introduction

Entering the deep learning era [37, 46, 82], enormous efforts have been made to tackle dense prediction problems, including but not limited to, image segmentation [15–17, 62, 112], depth estimation [28, 31, 48, 67, 76, 77], surface normal prediction [27, 44, 65, 91], and others [1, 23, 26, 56, 66, 108]. Early attempts formulate these problems as per-pixel prediction (*i.e.*, assigning a predicted value to every

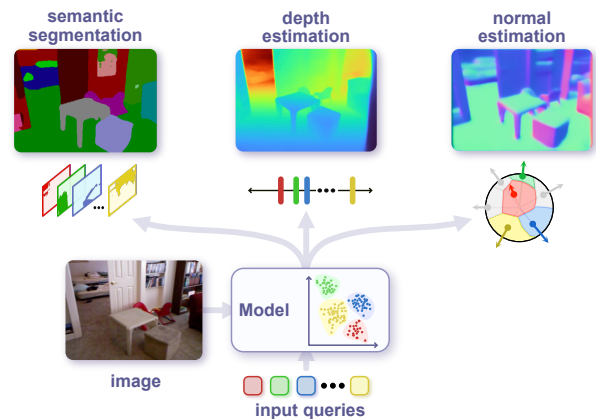


Figure 1. **Cluster-prediction forms the foundation of our unified Mask-Transformer-based framework PolyMaX for dense prediction tasks.** In the cluster-prediction paradigm, the model learns to transform input queries to cluster centers, where each cluster center learns to group similar pixels together. Pixels of the same group are assigned to the same label. This paradigm works well for discrete output space, such as segmentation. To extend it for continuous and even high-dimensional output space (*e.g.*, depth estimation and surface normal), we dynamically partition the output space into clusters, dependent on the input image.

pixel) via fully convolutional networks [62]. Specifically, when the desired prediction of each pixel is discrete, such as image segmentation, the task is constructed as per-pixel classification, while other tasks whose target outputs are continuous, such as depth and surface normal, are instead cast as per-pixel regression problems.

Recently, a new paradigm for segmentation tasks is drawing attention because of its superior performance compared to previous per-pixel classification approaches. Inspired by the object detection network DETR [13], MaX-DeepLab [87] and MaskFormer [22] propose to classify each segmentation mask as a whole instead of pixel-wise, by extending the concept of object queries in DETR to represent the clusters of pixels in segmentation masks. Specifically, with the help of pixel clustering via conditional convolutions [40, 85, 93], these *Mask-Transformer*-based works em-

\*Corresponding to xuanyang@google.com.

†Work done while at Google. Now at ByteDance.

<sup>1</sup><https://github.com/google-research/deeplab2>

ploy transformer decoders [86] to convert object queries to a set of (mask embedding vector, class prediction) pairs, which finally yield a set of binary masks by multiplying the mask embedding vectors with the pixel features. Such methods are effective when the target domain is *discretely* quantized (e.g., one semantic label is encoded by one integer scalar, as in image segmentation). However, it is unclear how this framework can be generalized to other dense prediction tasks, whose outputs are *continuous* or even multi-dimensional, such as depth estimation and surface normal prediction.

On the contrary, in the field of depth estimation, DORN [31] demonstrates the potential of *discretizing* the continuous depth range into a set of fixed bins, and AdaBins [7] further adaptively estimates the bin centers, dependent on the input image. The continuous depth values are then estimated by linearly combining the bin centers. Promising results are achieved by jointly learning the bin centers and performing per-pixel *classification* on those bins. This insight – of using classification to perform dense prediction in a continuous domain – opens up the possibility of performing many other continuous dense prediction tasks within the Mask-Transformer-based framework, a powerful tool for discrete value predictions.

Consequently, a few natural questions emerge: *Can we extend these Mask-Transformer-based frameworks even further, to solve more continuous dense prediction tasks? Can the resulting framework generalize to other multi-dimensional continuous domain, e.g., surface normal estimation?* In this work, we provide affirmative answers to those questions by proposing a new general architecture for dense prediction tasks. Specifically, building on top of the insight from [7, 31], we generalize the mask transformer framework [105] to multiple dense prediction tasks by using the cluster centers (*i.e.*, object queries) as the intermediate representation. We evaluate the resulting model, called **PolyMaX**, on the challenging NYUD-v2 [67] and Taskonomy [108] datasets. Remarkably, our simple yet effective approach demonstrates new state-of-the-art performance on semantic segmentation, depth estimation, and surface normal prediction, without using any extra modality as inputs (e.g., multi-modal RGB-D inputs as in CMNeXt [111]), heavily pretrained backbones (e.g., Stable-Diffusion [78] as in VPD [113]) or complex pretraining schemes (e.g., a mix of 12 datasets as in ZoeDepth [9]).

Our contributions are summarized as follows:

- We propose PolyMaX, which generalizes dense prediction tasks with a unified *Mask-Transformer*-based framework. We take surface normal as a concrete example to demonstrate how general dense prediction tasks can be solved by the proposed framework.
- We evaluate PolyMaX on NYUD-v2 and Taskonomy datasets, and it sets new state-of-the-arts on multiple benchmarks on NYUD-v2, achieving 58.08 mIoU,

0.250 root-mean-square (RMS) error and 13.09 mean error on semantic segmentation, depth estimation and surface normal prediction, respectively.

- We further perform the scalability study, which demonstrates that PolyMaX scales significantly better than the conventional per-pixel regression based methods as the pretraining data increases.
- Lastly, we provide the high-quality pseudo-labels of semantic segmentation for Taskonomy dataset, aiming to compensate for the scarcity of existing large-scale multi-task datasets and facilitate the future research.

## 2 Related Work

**Dense Prediction in the Discrete Domain** Image segmentation partitions images into multiple segments by their semantic classes. For example, fully convolutional networks [63] train semantic segmentation in an end-to-end manner mapping pixels into their classes [50, 58, 59, 106]. Atrous convolution increases the network receptive field for semantic segmentation without additional learnable parameters [15–17, 19, 74, 90, 101]. Many state-of-the-art methods use an encoder-decoder meta architecture for stronger global and local information integration [3, 18, 21, 32, 35, 42, 69, 84, 88, 103, 107].

**Dense Prediction in the Continuous Domain** Unlike image segmentation that produces discrete predictions, depth and surface normal estimation expect the models to predict continuous values [28, 72]. Most early attempts [47, 57, 74, 98] solve it as a standard regression problem. To alleviate training instability, DORN [31] proposes treating the problem as classification by discretizing the continuous depth into a set of pre-defined intervals (bins). AdaBins [7] adaptively learns the depth bins, conditioned on the input samples. LocalBins [8] further learns the depth range partitions from local regions instead of global distribution of depth ranges. BinsFormer [53] views adaptive bins generation as a direct set prediction problem [14]. The current state-of-the-art on depth estimation is set by VPD [113], which employs the Stable-Diffusion [78] pretrained with the large-scale LAION-5B [81] dataset. Among the non-diffusion-based models, ZoeDepth [9] shows the strongest capability by pretraining on 12 datasets with relative depth, and then finetuning on two datasets with metric depth.

**Surface Normal Prediction** Despite having a continuous output domain like depth estimation, the surface normal problem remains under-explored for discretization approaches. Most prior works on surface normal estimation focus on improving the loss [4, 24, 55], reducing the distribution bias [4] and shift [24], and resolving the artifacts in ground-truth by leveraging other modalities [39, 73]. Until recently, iDisc [71], a method based on discretizing internal representations, shows promise for depth estimation and

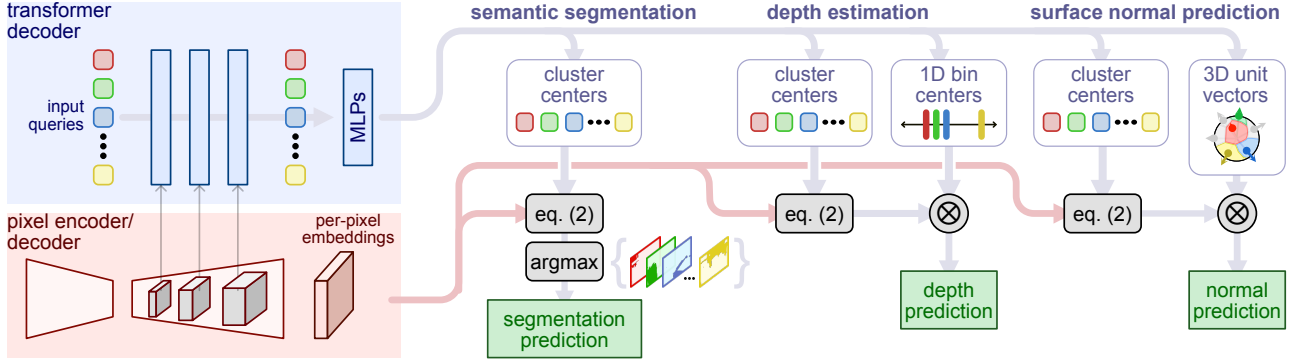


Figure 2. **The proposed model PolyMaX unifies dense prediction tasks with the mask transformer framework, where cluster centers are used as an intermediate representation.** The mask transformer framework contains two paths: (1) pixel encoder/decoder to generate per-pixel embeddings, and (2) transformer decoder to generate cluster centers (*i.e.*, object queries) from input queries. As shown in Eq. (2), the probability distribution map is generated by first multiplying the per-pixel embeddings with the cluster centers, followed by a softmax operation. For semantic segmentation, the final segmentation map is obtained by applying the argmax operation to find the maximum probability semantic label. For depth estimation, we further apply another MLP (omitted in the Figure) to estimate the bin center values, which are then linearly combined with the probability distribution map to form the final depth map. Similarly, for surface normal prediction, the estimated 3D unit vectors are linearly combined with the probability distribution map to form the final surface normal map.  $\otimes$  denotes the linear combination operation.

surface normal prediction. However, it differs from depth estimation discretization methods by enforcing discretization in the *feature* space rather than the target output space. We are more interested in the latter approach, as it is more generalizable to other dense prediction tasks.

**Other Efforts to Unify Dense Prediction Tasks** Realizing segmentation, depth and surface normal are all pixel-wise mapping problem, previous works [10, 49, 64, 68, 92] have deployed them to the same framework. Some works [11, 12, 99, 100, 102] improve performance by exploring the relations among dense prediction tasks. UViM [45] and Painter [92] unify segmentation and depth estimation, but neither of them is based on discretizing the continuous output space, and neither considers surface normal estimation. By contrast, we focus on a complementary perspective: a unified architecture for dense prediction that models both discrete and continuous tasks, covering image segmentation, depth estimation, and surface normal prediction.

**Mask Transformer** Witnessing the success of transformers [86] in NLP, the vision community has begun exploring them for more computer vision tasks [25]. Inspired by DETR [14] and conditional convolutions [40], MaX-DeepLab [87] and MaskFormer [22] switch the segmentation paradigm from per-pixel classification to mask classification, by introducing the *mask transformer*, where each input query learns to correspond to a mask prediction together with a class prediction. Several works [83, 96, 114] also introduce transformer into semantic segmentation. Further improvements are made on attention mechanism to enhance the performance of mask transformers [20, 52, 116]. CMT-DeepLab [104], kMaX-DeepLab [105], and ClustSeg [54] reformulate the cross-attention learning in transformer as a

clustering process. Our proposed PolyMaX builds on top of the cluster-based mask transformer architecture [105]. Rather than focusing on segmentation tasks, PolyMaX unifies dense prediction tasks (*i.e.*, image segmentation, depth estimation, and surface normal estimation) by extending the cluster-based mask transformer to support both discrete and continuous outputs.

### 3 Method

In this section, we first describe how segmentation [105] and depth estimation [7] can be transformed from per-pixel prediction problems to cluster-prediction problems (Sec. 3.1). We then introduce our proposed method, PolyMaX, a new mask transformer framework for general dense predictions (Sec. 3.2). We take surface normal prediction as a concrete example to explain how general dense prediction problems can be reformulated in a similar fashion, allowing us to unify them into the same mask transformer framework.

#### 3.1 Mask-Transformer-Based Segmentation and Depth Estimation

**Cluster-Prediction Paradigm for Segmentation** Two earlier works, MaX-DeepLab [87] and MaskFormer [22], demonstrate how to shift image segmentation from per-pixel classification to the cluster-prediction paradigm. Along the same direction, we follow the recently proposed clustering perspective [104, 105], which casts object queries to cluster centers. This paradigm is realized by the following two steps:

1. *pixel-clustering*: group pixels into  $K$  clusters, represented by segmentation masks  $\{\mathbf{m}_i | \mathbf{m}_i \in [0, 1]^{H \times W}\}_{i=1}^K$ , where  $H$  and  $W$  are height and width. Note that  $\mathbf{m}_i$  denotes soft segmentation masks.

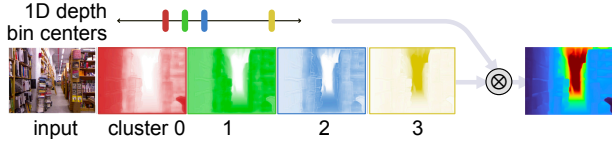


Figure 3. **For the task of depth prediction, we illustrate one example with four depth bins (i.e.,  $K = 4$ ).** The probability distribution map (i.e., output of Eq. (2)) has shape  $HW \times 4$  (generated by four cluster centers), which represents the predicted probability distribution over 4 bins for each pixel. As shown in the figure, the first cluster center (red) learns to highlight closest pixels; the second and third clusters (green/blue) highlight mid-range pixels, and the fourth (yellow) cluster highlights the furthest pixels. These maps are linearly combined (denoted as  $\otimes$ ) with their corresponding 1D bin centers to produce the final depth map prediction.

2. *cluster-classification*: assign semantic label to each cluster with a probability distribution over  $C$  classes. The  $i$ th cluster’s probability distribution is a 1D vector, denoted as  $\mathbf{p}_i$ , where  $\{\mathbf{p}_i | \mathbf{p}_i \in [0, 1]^C, \sum_{c=1}^C \mathbf{p}_{i,c} = 1\}_{i=1}^K$ .

These clusters and probability distributions are jointly learned to predict the output  $S$ , a set of  $K$  cluster-probability pairs:

$$S = \{(\mathbf{m}_i, \mathbf{p}_i)\}_{i=1}^K \quad (1)$$

This cluster-prediction paradigm is general for semantic [38], instance [36], and panoptic [43] segmentation. When only handling semantic segmentation, the framework can be further simplified by setting  $K = C$  (i.e., number of clusters is equal to number of classes) which has a fixed matching between cluster centers (i.e., object queries) and semantic classes. We adopt this simplification, since the datasets we experimented with only support semantic segmentation.

### Cluster-Prediction Paradigm for Depth Estimation

At first glance, depth estimation seems incompatible with this cluster-prediction paradigm because of its continuous nature. However, recent works [7, 31] propose promising solutions by dividing the continuous depth range into  $K$  learnable bins and regarding the task as a classification problem. By this means, the depth estimation task fits neatly into the above cluster-classification paradigm. Specifically, the *pixel-clustering* step outputs the range attention map  $\{\mathbf{r}_i | \mathbf{r}_i \in [0, 1]^{H \times W}\}_{i=1}^K$ , which (after softmax) represents the predicted probability distribution over the  $K$  bins for each pixel. While the *cluster-prediction* step estimates bin centers  $\{\mathbf{b}_i\}_{i=1}^K$ , adaptively discretizing the continuous depth range for each image. Here,  $K$  controls the granularity of the depth range partition. Therefore, the output of depth estimation can be expressed as  $\{(\mathbf{r}_i, \mathbf{b}_i)\}_{i=1}^K$ , sharing the same formulation as segmentation (Eq 1). The final depth prediction is then generated by a linear combination between the depth values of bin centers and the range attention map.

**Mask Transformer Framework** After describing the cluster-prediction paradigm for both segmentation and depth

target domain	output space $\mathbb{R}$	$D_{\mathcal{T}}$
segmentation	discrete	1
depth estimation	continuous	1
surface normal	continuous	3

Table 1. **General dense prediction learns the pixel mapping from the input RGB space to the structured output space  $\mathbb{R}$  with different dimensionalities  $D_{\mathcal{T}}$ .**

estimation, we now explain how to integrate them to the mask transformer framework. In the framework, the  $K$  cluster centers are learned from the  $K$  input queries through the transformer-decoder branch (blue block in Fig. 2). It associates the clusters, represented as query embedding vectors, with the pixel features extracted from the pixel encoder-decoder path [79] (pink block in Fig. 2) through self-attention and cross-attention blocks. The aggregated information is gradually refined as the pixel features go from lower-resolution to higher-resolution. The *probability distribution map* is then generated from the following equation:

$$\text{softmax}_K(\mathbf{F} \times \mathbf{Q}^T), \quad (2)$$

where  $\mathbf{F} \in \mathbb{R}^{HW \times D}$  and  $\mathbf{Q} \in \mathbb{R}^{K \times D}$  denote the final pixel features (i.e., per-pixel embeddings) and cluster centers (i.e., object queries).  $D$  is the channel dimension of pixel features and queries. For segmentation, the probability distribution map corresponds to the segmentation masks  $\{\mathbf{m}_i\}$ , while for depth estimation, it becomes the range attention map  $\{\mathbf{r}_i\}$ . Finally, another Multi-Layer Perceptrons (MLPs) are added on top of the cluster centers  $\mathbf{Q}$  to predict the probability distribution  $\mathbf{p}_i$  or bin centers  $\mathbf{b}_i$  for segmentation and depth estimation, respectively. Consequently, the cluster centers in the mask transformer are used as an intermediate representation in both segmentation and depth estimation.

## 3.2 Mask-Transformer-Based General Dense Prediction

General dense prediction tasks can be expressed as follows:

$$\mathcal{T} : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times D_{\mathcal{T}}}, D_{\mathcal{T}} \in \mathbb{N} \quad (3)$$

As presented in the above formulation, the objective of general dense prediction is to learn the pixel mapping  $\mathcal{T}$  from the input RGB space to the structured output space  $\mathbb{R}$  with different dimensionalities ( $D_{\mathcal{T}}$ ) and topologies (output space  $\mathbb{R}$  is either discrete or continuous). For instance, segmentation and depth estimation task correspond to the single-dimensional ( $D_{\mathcal{T}} = 1$ ) output space of discrete and continuous topology, respectively (Tab. 1). Given that the cluster-prediction methods have been effectively applied to these tasks (as described in Sec. 3.1), we take a step further and study how general dense prediction tasks, particularly multi-dimensional continuous target space (e.g., surface normal), can be learned by the cluster-prediction approach.

**Task of Surface Normal** Surface normal is a representative dense prediction task that targets a multi-dimensional

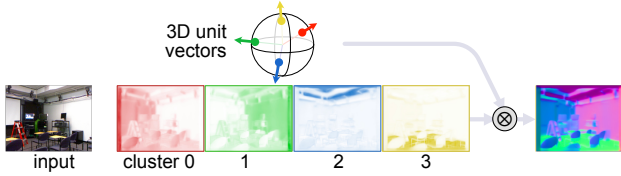


Figure 4. **To illustrate the task of surface normal prediction, we show an example model with four cluster centers** that roughly correspond to cardinal directions (3-vectors on the unit ball). Each channel of the  $HW \times 4$  probability distribution map represents the predicted probability of the corresponding vector direction. These maps are linearly combined (denoted as  $\otimes$ ) with their corresponding 3D unit vectors to generate the final surface normal prediction.

continuous output space, in which case  $D_{\mathcal{T}} = 3$  and output space  $\mathbb{R}$  is continuous. This task aims at predicting, for each pixel, the direction that is perpendicular to the tangent plane of the point in a 3D world, which can be expressed by a 3D unit vector  $\mathbf{v} = [N_x, N_y, N_z]$ , where  $N_x^2 + N_y^2 + N_z^2 = 1$ .

**Cluster-Prediction for Surface Normal** From the geometric perspective, a surface normal lies on the surface of a unit 3D ball. Therefore, the problem essentially becomes learning the pixel mapping  $\mathcal{T}$  from the 2D image space to the 3D point on the unit ball surface. The geometric meaning of the surface normal prediction allows us to naturally adopt the clustering-prediction approach.

Similar to [7, 31], we discretize the surface normal output space into  $K$  pieces of partitioned 3D spheres, converting the problem to classification among the  $K$  pieces. We refer the partitioned sphere as sphere segments for simplicity<sup>2</sup>, which can be viewed as clusters of 3D points on the unit ball surface, and thus are likely to have irregular shapes. Fortunately, with such a simplification, we can adopt the same paradigm as segmentation and depth estimation to tackle surface normals, where  $\mathbf{p}_i$  can be regarded as the probability over the  $K$  sphere segments. We jointly learn the center coordinates of sphere segments with the 3D-point-cluster and probability distribution. The final surface normal is predicted as a linear combination of the center coordinates of  $K$  sphere segments and the range attention map.

**Mask Transformer Framework for General Dense Prediction** Extending the cluster-prediction paradigm to general dense prediction tasks allows us to deploy and unify them into the mask transformer framework described in Sec. 3.1. Going from the single-dimension ( $D_{\mathcal{T}} = 1$ ) to multi-dimension ( $D_{\mathcal{T}} > 1$ ), the major upgrade happens at the place where the cluster center  $\mathbf{Q} \in \mathbb{R}^{K \times D}$  is mapped (via MLPs) to  $\mathbb{R}^{K \times D_{\mathcal{T}}}$  ( $D_{\mathcal{T}} > 1$ ), instead of  $\mathbb{R}^K$ , when predicting  $D_{\mathcal{T}}$ -dimensional bin centers. For instance, in the case of surface normal, each cluster center (*i.e.*, object query) predicts a unit vector  $\mathbf{v}_i \in \mathbb{R}^3$ .

**Model Instantiation** We build the proposed general

<sup>2</sup>Note that here the sphere segment does not rigorously match the mathematical definition of spherical segment, which refers to the solid produced by cutting a sphere with a pair of parallel planes.

framework on top of kMaX-DeepLab [105] with the official code-base [94]. We call the resulting model PolyMaX, a **Polymath** with **Mask Xformer** for general dense prediction tasks. The model architecture is illustrated in Fig. 2.

## 4 Experimental Results

In this section, we first provide the details of our experimental setup. We then report our main results on NYUD-v2 [67] and Taskonomy [108]. We also present visualizations to obtain deeper insights of the proposed cluster-prediction paradigm, followed by ablation studies.

### 4.1 Experimental Setup

**Dataset** We focus on semantic segmentation, monocular depth estimation and surface normal tasks during experiments, as they comprehensively represent the diverse target space of dense prediction (one-dimensional to multi-dimensional, and discrete to continuous). Specifically, we use NYUD-v2 [67] dataset. The official release provides 795 training and 654 testing images with real (instead of pseudo labels) ground-truths of those three tasks.

To complement the small scale of NYUD-v2 dataset, we also conduct experiments on the Taskonomy [108] dataset, which is composed of 4.6 million images (train: 3.4M, val: 538K; test: 629K images) from 537 different buildings with indoor scenes. This dataset is originally developed to facilitate the study of task transfer learning, thereby containing ground-truths of various tasks, including semantic segmentation, depth estimation, surface normal, and so on. The depth Z-buffer and surface normal ground-truth are programmatically computed from image registration and mesh alignment, resulting in high quality annotations. However, the provided semantic segmentation annotations are only pseudo-labels generated by Li et al. [50], an out-dated model trained on COCO [56]. After carefully examining these pseudo-labels, we notice that their quality does not satisfy the requirement for evaluating or improving state-of-the-art models. Therefore, we adopt the kMaX-DeepLab [105] model with ConvNeXt-L [61] backbone pretrained on COCO dataset to regenerate the pseudo-labels for semantic segmentation. We visually compare both labels in Fig. 5. The Taskonomy dataset enhanced by the new pseudo labels can serve as a meaningful complementary to this research field, particularly given the scarcity of the available large-scale high-quality multi-task dense prediction datasets. *We will release the high-quality pseudo labels to facilitate future research*<sup>3</sup>

**Evaluation Metrics** Semantic segmentation is evaluated with mean Intersection-over-Union (mIoU) [29]. For depth estimation, the metrics are root mean square error (RMS), Absolute mean relative error (A.Rel), absolute error in log-scale ( $\text{Log}_{10}$ ) and pixel inlier ratio ( $\delta_i$ ) with error threshold at  $1.25^i$  [28]. For surface normal metrics, following [27, 30] we use mean (Mean) and median (Median)

<sup>3</sup>Will release at <https://github.com/google-research/deeplab2>

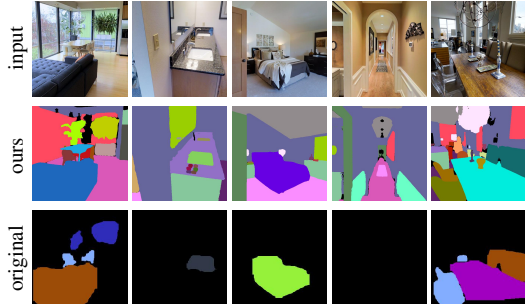


Figure 5. **Visualization of Taskonomy pseudo-labels: ours (middle) vs. original ones by Li *et al.* [50] (bottom).** Our pseudo-labels demonstrate higher quality than the existing ones.

absolute error, RMS angular error, and pixel inlier ratio ( $\delta_1$ ,  $\delta_2$ ,  $\delta_3$ ) with thresholds at  $11.5^\circ$ ,  $22.5^\circ$  and  $30^\circ$ , respectively.

**Implementation Details** We adopt the pixel encoder/decoder and the transformer decoder modules from the kMaX-DeepLab [105]. Additional L2 normalization is needed at the end of surface normal head to yield 3D unit vectors. During training, we adopt the same losses from kMaX-DeepLab for semantic segmentation. For depth estimation, when training on NYUD-v2 dataset, we use scale invariant logarithmic error [28], relative squared error [33], following ViP-DeepLab [74]. We also include the multi-scale gradient loss proposed by MegaDepth [51] to improve the visual sharpness of the predicted depth map. While training depth estimation on Taskonomy, we switch to robust Charbonnier loss [6], since this dataset has maximum depth value 128m with more outliers. The loss function we apply to surface normal is simply L2 loss, since it has better training stability than truncated angular loss [24]. The experiments on NYUD-v2 are conducted by first pretraining PolyMaX on Taskonomy dataset, and then finetuning on NYUD-v2 train split. The finetuning step is unnecessary when evaluating on Taskonomy test split. The learning rate at the pretraining and finetuning stages are  $5e-4$  and  $5e-5$ , respectively. To ensure fair comparisons with previous works, we closely follow their experiment setup. However, due to the use of different training data splits for each of the three tasks in previous studies, we have to train each task independently.

## 4.2 Main Results

**NYUD-v2** In Tab. 2, we compare PolyMaX with state-of-the-art models for dense prediction tasks on the NYUD-v2 dataset. We group the existing models based on the number and type of the tasks they support on this dataset. We choose the best numbers reported in the prior works. As shown in the table, in such a competitive comparison, PolyMaX still significantly outperforms all the existing models.

Specifically, in the semantic segmentation task, most of the recent models use additional modalities such as depth, but PolyMaX surpasses the existing best two models CMX [110] and CMNeXt [111] by 1.2% mIoU, without

using any additional modalities. In the depth estimation task, the current best model VPD [113] is built upon Stable-Diffusion [78] and pretrained with LAION-5B dataset [81]. Despite of only using a pretraining dataset (*i.e.*, Taskonomy) with 0.1% of that scale, PolyMaX achieves better performance on all the depth metrics than VPD. Furthermore, PolyMaX breaks the close competition among non-diffusion-based models by a meaningful improvement from above 0.27 to 0.25 RMS (all the recent non-diffusion-based models achieve around 0.27 RMS), similarly for all the other metrics. This is non-trivial, particularly given that the best non-diffusion-based depth model ZoeDepth [9] uses the pretrained BeiT<sub>384</sub>-L [5] backbone and 2-stage training with a mixture of 12 datasets. On the surface normal benchmark, PolyMaX continues to outperform all the existing models by a substantial margin, with the mean error being reduced from above 14.60 to 13.09. Finally, when comparing with models that support all three tasks on this dataset, the improvement of PolyMaX is further amplified for all metrics. Overall, PolyMaX is not only among the few models that support all the three tasks, but also sets a new state-of-the-art on all three dense prediction benchmarks.

**Taskonomy** In Tab. 3, we report our results on Taskonomy, along with a solid baseline<sup>4</sup> DeepLabv3+ [19]. The performance of the same baseline model on NYUD-v2 benchmarks in Tab. 2, compared with state-of-the-art models, can justify our choice of it as baseline on Taskonomy benchmarks. On all three tasks, PolyMaX significantly outperforms the baseline model. Specifically, the performance on semantic segmentation, depth estimation and surface normal is improved by 8.3 mIoU, 0.13 RMS and 1.0 mean error, respectively. These results further validate the effectiveness of the proposed framework.

**Visualizations and Limitations** We visualize the predictions of PolyMaX on all the three dense prediction tasks with their corresponding ground-truth in Fig. 6. As shown in the top row, PolyMaX successfully resolves the fine-grained details and irregular object shapes, and predicts high quality results. The bottom row shows a challenging case where PolyMaX can be further improved. Similar to many other depth models, it is difficult to correctly infer the depth map with glass, mirror and other reflective surfaces, due to the artifacts in the available ground-truth. Lastly, another direction for future improvement is the visual sharpness of the predicted surface normal. We find that the multi-scale gradient loss proposed by MegaDepth [51] can effectively improve the edge sharpness for depth estimation, and slightly improves depth metrics (*e.g.*, around 0.002 RMS improvement). It is promising to adapt the loss to surface normal prediction to improve visual quality.

<sup>4</sup>We notice recent works [10, 11] also report numbers on this benchmark, but with the unreleased code and different experimental setup, we can not consider them here as fair baselines.

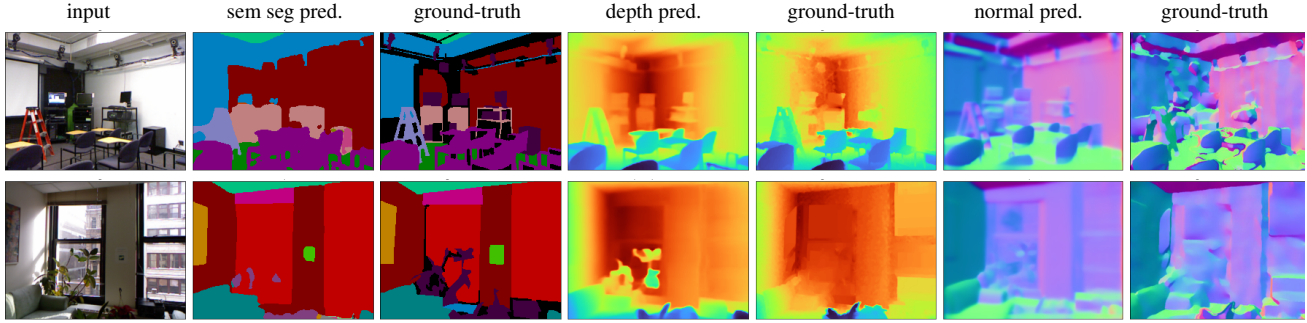


Figure 6. Visualization of model inputs and outputs for semantic segmentation, depth estimation and normal prediction. PolyMaX resolves fine details on scenes with complex structure (top row), and even performs acceptably when ground-truth is unreliable (last row).

model	encoder	pretraining data	sem seg	depth estimation				surface normal				
			mIoU $\uparrow$	RMS $\downarrow$	A.Rel $\downarrow$	Log <sub>10</sub> $\downarrow$	$\delta_1$ $\uparrow$	Mean $\downarrow$	Med $\downarrow$	RMS $\downarrow$	$\delta_1$ $\uparrow$	
OMNIVORE [34]	Swin-L [60]	IN-1K [80], Kinetics [41], SUN [95]	56.80	-	-	-	-	-	-	-	-	-
CMX [110]	MiT-B5 [96]	IN-1K [80]	56.90	-	-	-	-	-	-	-	-	-
CMNeXt [111]	MiT-B4 [96]	IN-1K [80]	56.90	-	-	-	-	-	-	-	-	-
BinsFormer [53]	Swin-L [60]	IN-22K [80]	-	0.330	0.094	0.040	0.925	-	-	-	-	-
DINov2 [70]	ViT-g [109]	LVD-142M [70]	-	0.279	0.091	0.037	0.950	-	-	-	-	-
AiT-P [68]	Swin-L [60]	IN-22K [80]	-	0.275	0.076	0.033	0.954	-	-	-	-	-
ZoeDepth [9]	BEiT <sub>384</sub> -L [5]	M12 [9]	-	0.270	0.075	0.032	0.955	-	-	-	-	-
Do et al. [24]	ResNeXt-101 [97]	IN-1K [80]	-	-	-	-	-	16.20	8.200	25.30	59.50	-
Bae et al. [4]	MiT-B5 [96]	IN-1K [80]	-	-	-	-	-	14.90	7.500	23.50	62.20	-
MultiMAE [2]	ViT-B [25]	IN-1K [80]	56.00	-	-	-	0.864	-	-	-	-	-
Painter [92]	ViT-L [25]	IN-1K [80]	-	0.288	0.080	-	0.950	-	-	-	-	-
MTFormer [99]	Swin [60]	IN-22K [80]	50.04	0.490	-	-	-	-	-	-	-	-
VPD [113]	SD [78]	LAION-5B [81], CLIP [75]	-	0.254	0.069	0.030	0.964	-	-	-	-	-
ASNDepth [64]	HRNet-48 [89]	IN-1K [80]	-	0.377	0.101	0.044	0.890	20.00	13.400	-	43.50	-
iDisc [71]	Swin-L [60]	IN-22K [80]	-	0.313	0.086	0.037	0.940	14.60	7.300	22.80	63.80	-
ARTC [12]	HRNet-48 [89]	IN-1K [80]	46.33	0.536	-	-	-	20.18	-	-	-	-
InvPT [102]	Swin-L [60]	IN-22K [80]	51.76	0.502	-	-	-	19.39	-	-	-	-
InvPT [102]	ViT-L [25]	IN-22K [80]	53.56	0.518	-	-	-	19.04	-	-	-	-
MQTrans. [100]	Swin-L [60]	ADE20K [115]	54.84	0.533	-	-	-	19.69	-	-	-	-
DeepLabv3+ [19]†	ResNet-50 [37]	IN-1K [80], Taskonomy [108]	50.14	0.375	0.108	0.046	0.887	14.81	8.291	22.49	61.44	-
PolyMaX	ResNet-50 [37]	IN-1K [80], Taskonomy [108]	51.30	0.317	0.084	0.037	0.936	14.03	7.789	21.45	63.25	-
PolyMaX	ConvNeXt-L [61]	IN-22K [80], Taskonomy [108]	54.59	0.282	0.076	0.033	0.954	13.73	7.533	21.18	64.07	-
PolyMaX	ConvNeXt-S [61]	IN-22K [80], Taskonomy [108]	56.34	0.273	0.073	0.032	0.959	13.42	7.351	20.74	64.73	-
PolyMaX	ConvNeXt-B [61]	IN-22K [80], Taskonomy [108]	57.21	0.260	0.071	0.030	0.965	13.23	7.181	20.58	65.48	-
PolyMaX	ConvNeXt-L [61]	IN-22K [80], Taskonomy [108]	<b>58.08</b>	<b>0.250</b>	<b>0.067</b>	<b>0.029</b>	<b>0.969</b>	<b>13.09</b>	<b>7.117</b>	<b>20.40</b>	<b>65.66</b>	-

Table 2. NYUD-v2 test set results. Our PolyMaX with ConvNeXt-L consistently outperforms prior arts on three benchmarks of NYUD-v2. †: Our reimplemented DeepLabv3+ of the pixel-regression method for depth estimation and surface normal prediction serves as an additional strong baseline.

model	encoder	sem seg	depth estimation				surface normal			
		mIoU $\uparrow$	RMS $\downarrow$	A.Rel $\downarrow$	Log <sub>10</sub> $\downarrow$	$\delta_1$ $\uparrow$	Mean $\downarrow$	Med $\downarrow$	RMS $\downarrow$	$\delta_1$ $\uparrow$
DeepLabv3+ [19]	ResNet-50 [37]	32.87	0.6303	0.1472	0.0618	82.61	11.45	4.938	19.65	56.13
PolyMaX	ResNet-50 [37]	35.00	0.5885	0.1236	0.0538	86.49	10.96	4.922	18.83	55.27
PolyMaX	ConvNeXt-T [61]	35.23	0.5287	0.1140	0.0479	88.56	10.87	4.859	18.49	55.39
PolyMaX	ConvNeXt-S [61]	38.99	0.5097	0.1077	0.0452	89.59	10.67	4.709	18.46	56.05
PolyMaX	ConvNeXt-B [61]	<b>41.17</b>	<b>0.4981</b>	0.1034	0.0433	90.37	10.58	4.646	18.37	56.43
PolyMaX	ConvNeXt-L [61]	40.94	0.4988	<b>0.1029</b>	<b>0.0432</b>	<b>90.41</b>	<b>10.46</b>	<b>4.550</b>	<b>18.25</b>	<b>56.76</b>

Table 3. Taskonomy test set results. The DeepLabv3+ baseline and PolyMaX are trained with the same configuration.

### 4.3 Ablation Studies

#### Scalability of Mask-Transformer-Based Framework

We investigate the scalability of PolyMaX, since this property has become increasingly crucial in the large scale model era. Fig. 7 demonstrates that PolyMaX’s scalability with the percentage of pretraining dataset. For a fair comparison, we build the baseline models upon DeepLabv3+ [19], using pixel-classification prediction head for semantic segmentation, and pixel-regression prediction head for depth estimation and surface normal prediction. The training con-

figurations are the same within each task, except that the DeepLabv3+ based depth model requires the robust Charbonnier loss [6] to overcome the typical training instability issue of regression with outliers.

As shown in Fig. 7a, PolyMaX scales significantly better than the baseline model on semantic segmentation. When all models are pretrained with only ImageNet [80] (i.e., 0% Taskonomy data is introduced in pretraining), PolyMaX performs worse than the baseline, especially the ConvNeXt-L model variant fails disappointingly. This is because the

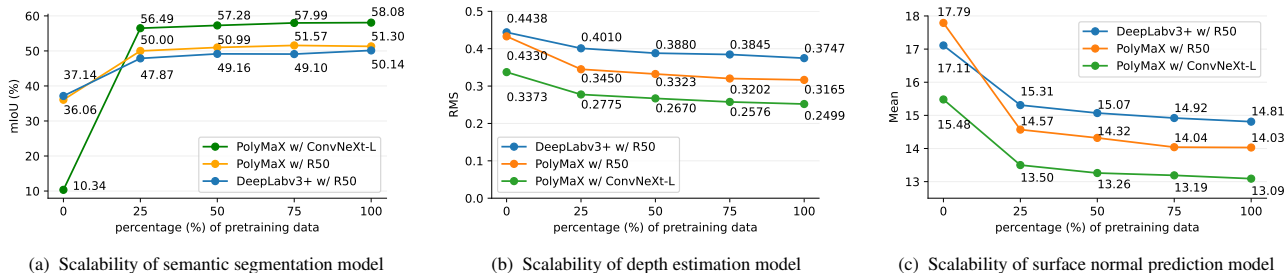


Figure 7. Experiments of model scalability with different pretraining data percentage of Taskonomy on NYUD-v2. Our PolyMaX demonstrates better scalability than the solid per-pixel prediction baseline DeepLabv3+ on all three tasks.

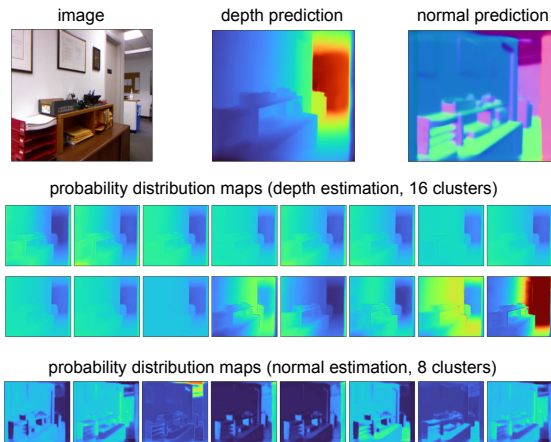


Figure 8. Visualization of probability distribution maps for depth prediction (middle) and normal estimation (bottom) tasks. It illustrates the attention on regions with different depth and different angles, which validate the effectiveness of PolyMaX with cluster-prediction paradigm.

larger capacity of PolyMaX can not be fully exploited with only 795 training samples that are available for semantic segmentation from NYUD-v2 dataset. Thus, after including only 25% of the Taskonomy data in pretraining, PolyMaX with ResNet-50 and ConvNeXt-L immediately outperform the baseline model with more than 2% mIoU and 9% mIoU, respectively. This performance gaps remain when the percentage of pretraining data increases, and stay at more than 1% and 7% mIoU after using the entire Taskonomy dataset for pretraining.

The scalability of depth estimation is shown in Fig. 7b. When no Taskonomy pretraining data is used, the improvement of PolyMaX over the baseline (both with ResNet-50) is only 0.01 RMS, implying negligible positive impact of cluster-prediction paradigm. The more than 0.1 RMS performance improvement of PolyMaX with ConvNeXt-L mainly comes from the more powerful backbone. However, PolyMaX has better scalability with the increase of pretraining data. When 25% Taskonomy data is involved in pretraining, the model performance improvements of PolyMaX with ResNet-50 and ConvNeXt-L increase from 0.01 and 0.1 RMS to 0.05 and 0.12 RMS, respectively. With all Taskonomy

dataset being used in pretraining, the performance gaps of the two PolyMaX come to 0.6 and 0.12 RMS.

The scalability of PolyMaX on surface normal, showing a similar trend as depth estimation, is illustrated in Fig. 7c. Even though PolyMaX with ResNet-50 does not perform better than the baseline model when no Taskonomy pretraining data is used, it significantly surpasses the baseline model by 0.7 and 0.9 mean error at 25% and 100% of pretraining data. While PolyMaX with ConvNeXt-L substantially outperforms the baseline model by 1.2 and 1.8 mean error at 25% and 100% of pretraining data.

**Visualization of Cluster Representation** To obtain deeper understanding of the model behaviour, we further look into the visualization of the learned cluster centers and the probability distribution across those clusters for depth estimation and surface normal, shown in Fig. 8. Interestingly, 12 out of the 16 probability maps from the depth estimation model look similar, implying redundancy in the learned 16 cluster centers. Similar findings are observed for surface normal. This can also explain the small impact of cluster count  $K$  on model performance in tables of supplementary.

Despite the redundancy, the rest unique probability maps clearly depict the cluster-prediction paradigm. For instance, the last 4 probability maps of the depth model correspond to the attention on close, medium, further and furthest regions in the scene. Similarly, the unique probability maps of surface normal illustrate the attention on surfaces with different angles, including left, right, upwards, downwards.

## 5 Conclusion

We have generalized dense prediction tasks with the same mask transformer framework, realized by casting semantic segmentation, depth estimation and surface normal to cluster-prediction paradigm. The proposed PolyMaX has demonstrated state-of-the-art results on the three benchmarks of NYUD-v2 dataset. We hope the superior performance of PolyMaX can enable many impactful applications, including but not limited to scene understanding, image generation and editing, and augmented reality.

*Acknowledgement* We thank Yukun Zhu, Jun Xie and Shuyang Sun for their support on the code-base.



## References

- [1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, May 2011. 1
- [2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimaec: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022. 7
- [3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 39(12):2481–2495, 2017. 2
- [4] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *ICCV*, 2021. 2, 7
- [5] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 6, 7
- [6] Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, 2019. 6, 7
- [7] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021. 2, 3, 4, 5
- [8] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *ECCV*, 2020. 2
- [9] Shariq Farooq Bhat, Reiner Birkel, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv:2302.12288*, 2023. 2, 6, 7
- [10] Deblina Bhattacharjee, Sabine Süsstrunk, and Mathieu Salzmann. Vision transformer adapters for generalizable multi-task learning. *arXiv:2308.12372*, 2023. 3, 6
- [11] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *CVPR*, 2022. 3, 6
- [12] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021. 3, 7
- [13] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1
- [14] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 3
- [15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 1, 2
- [16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 2017. 1, 2
- [17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017. 1, 2
- [18] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 2
- [19] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 6, 7
- [20] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. In *CVPR*, 2022. 3
- [21] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *CVPR*, 2020. 2
- [22] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 1, 3
- [23] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1
- [24] Tien Do, Khiem Vuong, Stergios I Roulletis, and Hyun Soo Park. Surface normal estimation of tilted images via spatial rectifier. In *ECCV*, 2020. 2, 6, 7
- [25] 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. In *ICLR*, 2021. 3, 7
- [26] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1
- [27] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 1, 5
- [28] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 1, 2, 5, 6
- [29] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 5
- [30] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, 2013. 5
- [31] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 1, 2, 4, 5
- [32] Jun Fu, Jing Liu, Yuhang Wang, Jin Zhou, Changyong Wang, and Hanqing Lu. Stacked deconvolutional network for semantic segmentation. *IEEE TIP*, 2019. 2
- [33] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 6
- [34] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der

- Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 7
- [35] Xiuye Gu, Yin Cui, Jonathan Huang, Abdullah Rashwan, Xuan Yang, Xingyi Zhou, Golnaz Ghiasi, Weicheng Kuo, Huizhong Chen, Liang-Chieh Chen, and David A Ross. Dataseg: Taming a universal multi-dataset multi-task segmentation model. *NeurIPS*, 2023. 2
- [36] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 4
- [37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 7
- [38] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 4
- [39] Steven Hickson, Karthik Raveendran, Alireza Fathi, Kevin Murphy, and Irfan Essa. Floors are flat: Leveraging semantics for real-time surface normal prediction. In *ICCV workshop*, 2019. 2
- [40] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 1, 3
- [41] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv:1705.06950*, 2017. 7
- [42] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv:1511.02680*, 2015. 2
- [43] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 4
- [44] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 1
- [45] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. In *NeurIPS*, 2022. 3
- [46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1
- [47] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 2
- [48] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv:1907.10326*, 2019. 1
- [49] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Learning multiple dense prediction tasks from partially annotated data. In *CVPR*, 2022. 3
- [50] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2, 5, 6
- [51] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6
- [52] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Tong Lu, and Ping Luo. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *CVPR*, 2022. 3
- [53] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv:2204.00987*, 2022. 2, 7
- [54] James Liang, Tianfei Zhou, Dongfang Liu, and Wenguan Wang. Clustseg: Clustering for universal segmentation. In *ICML*, 2023. 3
- [55] Shuai Liao, Efstratios Gavves, and Cees GM Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *CVPR*, 2019. 2
- [56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 5
- [57] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE TPAMI*, 38(10):2024–2039, 2015. 2
- [58] Nianfeng Liu, Haiqing Li, Man Zhang, Jing Liu, Zhenan Sun, and Tieniu Tan. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In *ICB*, 2016. 2
- [59] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv:1506.04579*, 2015. 2
- [60] 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 *ICCV*, 2021. 7
- [61] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 5, 7
- [62] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1
- [63] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2
- [64] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *ICCV*, 2021. 3, 7
- [65] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 1
- [66] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 1
- [67] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 1, 2, 5
- [68] Jia Ning, Chen Li, Zheng Zhang, Zigang Geng, Qi Dai, Kun He, and Han Hu. All in tokens: Unifying output space of visual tasks via soft token. *arXiv:2301.02229*, 2023. 3, 7
- [69] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation.

- In *ICCV*, 2015. 2
- [70] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023. 7
- [71] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *CVPR*, 2023. 2, 7
- [72] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018. 2
- [73] Xiaojuan Qi, Zhengzhe Liu, Renjie Liao, Philip HS Torr, Raquel Urtasun, and Jiaya Jia. Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. *IEEE TPAMI*, 44(2):969–984, 2020. 2
- [74] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *CVPR*, 2021. 2, 6
- [75] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 7
- [76] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 1
- [77] 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 TPAMI*, 44(3), 2022. 1
- [78] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 6, 7
- [79] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4
- [80] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 7
- [81] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 2, 6, 7
- [82] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1
- [83] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 3
- [84] Shuyang Sun, Weijun Wang, Qihang Yu, Andrew Howard, Philip Torr, and Liang-Chieh Chen. Remax: Relaxing for better training on efficient panoptic segmentation. *NeurIPS*, 2023. 2
- [85] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 1
- [86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3
- [87] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 1, 3
- [88] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020. 2
- [89] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE TPAMI*, 2020. 7
- [90] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 2
- [91] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 1
- [92] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023. 3, 7
- [93] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020. 1
- [94] Mark Weber, Huiyu Wang, Siyuan Qiao, Jun Xie, Maxwell D. Collins, Yukun Zhu, Liangzhe Yuan, Dahun Kim, Qihang Yu, Daniel Cremers, Laura Leal-Taixe, Alan L. Yuille, Florian Schroff, Hartwig Adam, and Liang-Chieh Chen. DeepLab2: A TensorFlow Library for Deep Labeling. *arXiv: 2106.09748*, 2021. 5
- [95] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *IJCV*, 119:3–22, 2016. 7
- [96] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 3, 7
- [97] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 7
- [98] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018. 2
- [99] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtformer: Multi-task learning via transformer and cross-task reasoning. In *ECCV*, 2022. 3, 7
- [100] Y Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *arXiv:2205.14354*, 2022. 3, 7
- [101] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes.

- In *CVPR*, 2018. 2
- [102] Hanrong Ye and Dan Xu. Invpt: Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, 2022. 3, 7
- [103] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *NeurIPS*, 2023. 2
- [104] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *CVPR*, 2022. 3
- [105] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *ECCV*, 2022. 2, 3, 5, 6
- [106] Yading Yuan, Ming Chao, and Yeh-Chi Lo. Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance. *IEEE transactions on medical imaging*, 36(9):1876–1886, 2017. 2
- [107] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 2
- [108] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 1, 2, 5, 7
- [109] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. 7
- [110] Jiaming Zhang, Huayao Liu, Kailun Yang, Xinxin Hu, Ruiping Liu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 6, 7
- [111] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *CVPR*, 2023. 2, 6, 7
- [112] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1
- [113] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. *arXiv:2303.02153*, 2023. 2, 6, 7
- [114] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 3
- [115] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 7
- [116] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 3