iDisc: Internal Discretization for Monocular Depth Estimation

Luigi Piccinelli  Christos Sakaridis  Fisher Yu
Computer Vision Lab, ETH Zürich

Abstract

Monocular depth estimation is fundamental for 3D scene understanding and downstream applications. However, even under the supervised setup, it is still challenging and ill-posed due to the lack of full geometric constraints. Although a scene can consist of millions of pixels, there are fewer high-level patterns. We propose iDisc to learn those patterns with internal discretized representations. The method implicitly partitions the scene into a set of high-level patterns. In particular, our new module, Internal Discretization (ID), implements a continuous-discrete-continuous bottleneck to learn those concepts without supervision. In contrast to state-of-the-art methods, the proposed model does not enforce any explicit constraints or priors on the depth output. The whole network with the ID module can be trained end-to-end, thanks to the bottleneck module based on attention. Our method sets the new state of the art with significant improvements on NYU-Depth v2 and KITTI, outperforming all published methods on the official KITTI benchmark. iDisc can also achieve state-of-the-art results on surface normal estimation. Further, we explore the model generalization capability via zero-shot testing. We observe the compelling need to promote diversification in the outdoor scenario. Hence, we introduce splits of two autonomous driving datasets, DDAD and Argoverse. Code is available at http://vis.xyz/pub/idisc.

1. Introduction

Depth estimation is essential in computer vision, especially for understanding geometric relations in a scene. This task consists in predicting the distance between the projection center and the 3D point corresponding to each pixel. Depth estimation finds direct significance in downstream applications such as 3D modeling, robotics, and autonomous cars. Some research [62] shows that depth estimation is a crucial prompt to be leveraged for action reasoning and execution. In particular, we tackle the model of monocular depth estimation (MDE). MDE is an ill-posed problem due to its inherent scale ambiguity: the same 2D input image can correspond to an infinite number of 3D scenes.

State-of-the-art (SotA) methods typically involve convolutional networks [12, 13, 24] or, since the advent of vision Transformer [11], transformer architectures [3, 41, 54, 59]. Most methods either impose geometric constraints on the image [22, 33, 38, 55], namely, planarity priors or explicitly discretize the continuous depth range [3, 4, 13]. The latter can be viewed as learning frontoparallel planes. These imposed priors inherently limit the expressiveness of the respective models, as they cannot model arbitrary depth patterns, ubiquitous in real-world scenes.

We instead propose a more general depth estimation model, called iDisc, which does not explicitly impose any constraint on the final prediction. We design an Internal Discretization (ID) of the scene which is in principle depth-agnostic. Our assumption behind this ID is that each scene can be implicitly described by a set of concepts or patterns,
such as objects, planes, edges, and perspective relationships. The specific training signal determines which patterns to learn (see Fig. 1).

We design a continuous-to-discrete bottleneck through which the information is passed in order to obtain such internal scene discretization, namely the underlying patterns. In the bottleneck, the scene feature space is partitioned via learnable and input-dependent quantizers, which in turn transfer the information onto the continuous output space. The ID bottleneck introduced in this work is a general concept and can be implemented in several ways. Our particular ID implementation employs attention-based operators, leading to an end-to-end trainable architecture and input-dependent framework. More specifically, we implement the continuous-to-discrete operation via “transposed” cross-attention, where transposed refers to applying softmax on the output dimension. This softmax formulation enforces the input features to be routed to the internal discrete representations (IDRs) in an exclusive fashion, thus defining an input-dependent soft clustering of the feature space. The discrete-to-continuous transformation is implemented via cross-attention. Supervision is only applied to the final output, without any assumptions or regularization on the IDRs.

We test iDisc on multiple indoor and outdoor datasets and probe its robustness via zero-shot testing. As of today, there is too little variety in MDE benchmarks for the outdoor scenario, since the only established benchmark is KITTI [17]. Moreover, we observe that all methods fail on outdoor zero-shot testing, suggesting that the KITTI dataset is not diverse enough and leads to overfitting, thus implying that it is not indicative of generalized performance. Hence, we find it compelling to establish a new benchmark setup for the MDE community by proposing two new train-test splits of more diverse and challenging high-quality outdoor datasets: Argoverse1.1 [8] and DDAD [18].

Our main contributions are as follows: (i) we introduce the Internal Discretization module, a novel architectural component that adeptly represents a scene by combining underlying patterns; (ii) we show that it is a generalization of SotA methods involving depth ordinal regression [3, 13]; (iii) we propose splits of two raw outdoor datasets [8, 18] with high-quality LiDAR measurements. We extensively test iDisc on six diverse datasets and, owing to the ID design, our model consistently outperforms SotA methods and presents better transferability. Moreover, we apply iDisc to surface normal estimation. Each of the scalar depth values is associated with a confidence mask which describes the probability of each pixel presenting such a depth value. Hence, SotA methods inherently assume that depth can be represented as a set of frontoparallel planes, that is, depth “masks”.

The main paradigm of ordinal regression methods is to first obtain hidden representations and scalar values of discrete depth values. The dot-product similarity between the feature maps and the depth representations is treated as logits and softmax is applied to extract confidence masks (in Fu et al. [13] this degenerates to argmax). Finally, the final prediction is defined as the per-pixel weighted average of the discrete depth values, with the confidence values serving as the weights. iDisc draws connections with the idea of depth discretization. However, our ID module is designed to be depth-agnostic. The discretization occurs at the abstract level of internal features from the ID bottleneck instead of the output depth level, unlike other methods.

Iterative routing is related to our “transposed” cross-attention. The first approach of this kind was Capsule Networks and their variants [20, 42]. Some formulations [32, 46]
employ different kinds of attention mechanisms. Our attention mechanism draws connections with [32]. However, we do not allow permutation invariance, since our assumption is that each discrete representation internally describes a particular kind of pattern. In addition, we do not introduce any other architectural components such as gated recurrent units (GRU). In contrast to other methods, our attention is employed at a higher abstraction level, namely in the decoder.

3. Method

We propose an Internal Discretization (ID) module, to discretize the internal feature representation of encoder-decoder network architectures. We hypothesize that the module can break down the scenes into coherent concepts without semantic supervision. This section will first describe the module design and then discuss the network architecture. Sec. 3.1.1 defines the formulation of “transposed” cross-attention outlined in Sec. 1 and describes the main difference with previous formulations from Sec. 2. Moreover, we derive in Sec. 3.1.2 how the iDisc formulation can be interpreted as a generalization of SotA ordinal regression methods by reframing their original formulation. Eventually, Sec. 3.2 presents the optimization problem and the overall architecture.

3.1. Internal Discretization Module

The ID module involves a continuous-discrete-continuous bottleneck composed of two main consecutive stages. The overall module is based on our hypothesis that scenes can be represented as a finite set of patterns. The first stage consists in a continuous-to-discrete component, namely soft-exclusive discretization of the feature space. More specifically, it enforces an input-dependent soft clustering on the feature maps in an image-to-set fashion. The second stage completes the internal scene discretization by mapping the learned IDRs onto the continuous output space. IDRs are not bounded to focus exclusively on depth planes but are allowed to represent any high-level pattern or concept, such as objects, relative locations, and planes in the 3D space. In contrast with SotA ordinal regression methods [3, 4, 13], the IDRs are neither explicitly tied to depth values nor directly tied to the output. Moreover, our module operates at multiple intermediate resolutions and merges them only in the last layer. The overall architecture of iDisc, particularly our ID module, is shown in Fig. 2.

3.1.1 Adaptive Feature Partitioning

The first stage of our ID module, Adaptive Feature Partitioning (AFP), generates proper discrete representations (H := {H_l}^3_{l=1}) that quantize the feature space (F := {F_l}^3_{l=1}) at each resolution l. We drop the resolution superscript l since resolutions are independently processed and only one generic resolution is treated here. iDisc does not simply learn fixed centroids, as in standard clustering, but rather learns how to define a partition function in an input-dependent fashion. More specifically, an iterative transposed cross-attention module is utilized. Given the specific input feature maps (F), the iteration process refines (learnable) IDR priors (H_{prior}) over R iterations.
More specifically, the term “transposed” refers to the different axis along which the \texttt{softmax} operation is applied, namely \( \texttt{softmax}(QK^T)^T V \) instead of the canonical dot-product attention \( \texttt{softmax}(QK^T)V \), with \( Q, K, V \) as query, key, and value tensors, respectively. In particular, the tensors are obtained as projections of feature maps and IDR priors, \( f_Q(H_{\text{prior}}), f_K(F), f_V(F) \). The \( t \)-th iteration out of \( T \) can be formulated as follows:

\[
W_{ij}^t = \frac{\exp(k_i^T q_j)}{\sum_k \exp(k_i^T q_k)}, \quad a_{t+1} = \sum_i W_{ij}^t v_i,
\]

where \( q_j, k_i, v_i \in \mathbb{R}^C \) are query, key and value respectively, \( N \) is the number of IDRs, namely, clusters, and \( M \) is the number of pixels. The weights \( W_{ij}^t \) may be normalized to 1 along the \( i \) dimension to avoid vanishing or exploding quantities due to the summation of un-normalized distribution.

The quantization stems from the inherent behavior of \texttt{softmax}. In particular, \texttt{softmax} forces competition among outputs: one output can be large only to the detriment of others. Therefore, when fixing \( i \), namely, given a feature, only a few attention weights \( (W_{ij}^t) \) may be significantly greater than zero. Hence, the content \( v_i \) is routed only to a few IDRs at the successive iteration. Feature maps are fixed during the process and weights are shared by design, thus \( \{k_i, v_i\}_{i=1}^M \) are the same across iterations. The induced competition enforces a soft clustering of the input feature space, where the last-iteration IDR represents the actual partition function \( (H := Q^R) \). The probabilities of belonging to one partition are the attention weights, namely \( W_{ij}^T \) with \( j \)-th query fixed. Since attention weights are inherently dependent on the input, the specific partitioning also depends on the input and takes place at inference time. The entire process of AFP leads to (soft) mutually exclusive IDRs.

As far as the partitioning rationale is concerned, the proposed AFP draws connections with iterative routing methods described in Sec. 2. However, important distinctions apply. First, IDRs are not randomly initialized as the “slots” in Locatello et al. [32] but present a learnable prior. Priors can be seen as learnable positional embeddings in the attention context, thus we do not allow a permutation-invariant set of representations. Moreover, non-adaptive partitioning can still take place via the learnable priors if the iterations are zero. Second, the overall architecture differs noticeably as described in Sec. 2, and in addition, iDisc partitions feature space at the decoder level, corresponding to more abstract, high-level concepts, while the SotA formulations focus on clustering at an abstraction level close to the input image.

One possible alternative approach to obtaining the aforementioned IDRs is the well-known image-to-set proposed in DETR [7], namely via classic cross-attention between representations and image feature maps. However, the corresponding representations might redundantly aggregate features, where the extreme corresponds to each output being the mean of the input. Studies [15, 44] have shown that slow convergence in transformer-based architectures may be due to the non-localized context in cross-attention. The exclusiveness of the IDRs discourages the redundancy of information in different IDRs. We argue that exclusiveness allows the utilization of fewer representations (32 against the 256 utilized in [3] and [13]), and can improve both the interpretability of what IDRs are responsible for and training convergence.

### 3.1.2 Internal Scene Discretization

In the second stage of the ID module, \textit{Internal Scene Discretization} (ISD), the module ingests pixel embeddings \( \{P_i\}_{i=1}^3 \) from the decoder and IDRs \( \mathcal{H} \) from the first stage, both at different resolutions \( l \), as shown in Fig. 2. Each discrete representation carries both the signature, as the \( \mathcal{H} \), and the output-related content, as the \( v_i \), of the pattern it represents. The similarity between IDRs and pixel embeddings is computed in order to spatially localize in the continuous output space where to transfer the information of each IDR. We utilize the dot-product similarity function.

Furthermore, the kind of information to transfer onto the final prediction is not constrained, as we never explicitly handle depth values, usually called bins, until the final output. Thus, the IDRs are completely free to carry generic high-level concepts (such as object-ness, relative positioning, and geometric structures). This approach is in stark contrast with SotA methods [3, 4, 13, 27], which explicitly constrain what the representations are about: scalar depth values. Instead, iDisc learns to generate unconstrained representations in an input-dependent fashion. The effective discretization of the scene occurs in the second stage thanks to the information transfer from the set of exclusive concepts \( \{\mathcal{H}\} \) from AFP to the continuous space defined by \( P \). We show that our method is not bounded to depth estimation, but can be applied to generic continuous dense tasks, for instance, surface normal, etc. Consequently, we argue that the training signal of the task at hand determines how to internally discretize the scene, rendering our ID module general and usable in settings other than depth estimation.

From a practical point of view, the whole second stage consists in cross-attention layers applied to IDRs and pixel embeddings. As described in Sec. 3.1.1, we drop the resolution superscript \( l \). After that, the final depth maps are projected onto the output space and the multi-resolution depth predictions are combined. The \( i \)-th layer is defined as:

\[
D_{i+1} = \texttt{softmax}(Q_iK_i^T)V_i + D_i,
\]

where \( Q_i = f_{Q_i}(P) \in \mathbb{R}^{H \times W \times C} \), \( P \) are pixel embeddings with shape \( (H,W) \), and \( K_i, V_i \in \mathbb{R}^{N \times C} \) are the \( N \) IDRs under linear transformations \( f_{K_i}(H), f_{V_i}(H) \). The term \( Q_iK_i^T \) determines the spatial location for which each
specific IDR is responsible, while $V_i$ carries the semantic content to be transferred in the proper spatial locations.

Our approach constitutes a generalization of depth estimation methods that involve (hybrid) ordinal regression. As described in Sec. 2, the common paradigm in ordinal regression methods is to explicitly discretize depth in a set of masks with a scalar depth value associated with it. Then, they predict the likelihood that each pixel belongs to such masks. Our change of paradigm stems from the reinterpretation of the mentioned ordinal regression pipeline which we translate into the following mathematical expression:

$$D = \text{softmax}(P R^T / T)v,$$

where $P$ are the pixel embeddings at maximum resolution and $T$ is the softmax temperature. $v \in \mathbb{R}^{N \times 1}$ are $N$ depth scalar values and $R \in \mathbb{R}^{N \times (C-1)}$ are their hidden representations, both processed as a unique stacked tensor ($R|v \in \mathbb{R}^{N \times C}$). From the reformulation in (3), one can observe that (3) is a degenerate case of (2). In particular, $f_0$ degenerates to the identity function. $f_K$ and $f_V$ degenerate to selector functions: the former function selects up to the $C - 1$ dimensions and the latter selects the last dimension only. Moreover, the hidden representations are refined pixel embeddings ($f(P_i) = H_i = R|v$), and $D$ in (3) is the final output, namely no multiple iterations are performed as in (2). The explicit entanglement between the semantic content of the hidden representations and the final output is due to hard-coding $v$ as depth scalar values.

### 3.2. Network Architecture

Our network described in Fig. 2 comprises first an encoder backbone, interchangeably convolutional or attention-based, producing features at different scales. The encoded features at different resolutions are refined, and information between resolutions is shared, both via four multi-scale deformable attention (MSDA) blocks [63]. The feature maps from MSDA encoding are combined with the respective IDRs in the ISD stage of the ID module to extract the depth maps. The final depth prediction corresponds to the mean of the interpolated $T$ resolutions in the continuous space ($\mathcal{I}$). The feature maps from MSDA encodings at different resolutions are refined, and information between resolutions is shared, both via four multi-scale deformable attention (MSDA) blocks [63].

#### 4. Experiments

### 4.1. Experimental Setup

#### 4.1.1 Datasets

**NYU-Depth V2.** NYU-Depth V2 (NYU) [36] is a dataset consisting of 464 indoor scenes with RGB images and quasi-dense depth images with $640 \times 480$ resolution. Our models are trained on the train-test split proposed by previous methods [24], corresponding to 24,231 samples for training and 654 for testing. In addition to depth, the dataset provides surface normal data utilized for normal estimation. The train split used for normal estimation is the one proposed in [55].

**Zero-shot testing datasets.** We evaluate the generalizability of indoor models on two indoor datasets which are not seen during training. The selected datasets are SUN-RGBD [43] and DIODE-Indoor [47]. For both datasets, the resolution is reduced to match that of NYU, which is $640 \times 480$.

**KITTI.** The KITTI dataset provides stereo images and corresponding Velodyne LiDAR scans of outdoor scenes captured from a moving vehicle [17]. RGB and depth images have (mean) resolution of $1241 \times 376$. The split proposed by [12] (Eigen-split) with corrected depth is utilized as training and testing set, namely, 23,158 and 652 samples. The evaluation crop corresponds to the crop defined by [16]. All methods in
Sec. 4.2 that have source code and pre-trained models available are re-evaluated on KITTI with the evaluation mask from [16] to have consistent results.

**Argoverse1.1 and DDAD.** We propose splits of two autonomous driving datasets, Argoverse1.1 (Argoverse) [8] and DDAD [18], for depth estimation. Argoverse and DDAD are both outdoor datasets that provide 360° HD images and the corresponding LiDAR scans from moving vehicles. We pre-process the original datasets to extract depth maps and avoid redundancy. Training set scenes are sampled when the vehicle has been displaced by at least 2 meters from the previous sample. For the testing set scenes, we increase this threshold to 50 meters to further diminish redundancy. Our Argoverse split accounts for 21,672 training samples and 476 test samples, while DDAD for 18,380 training and 860 testing samples. Samples in Argoverse are taken from the 6 cameras covering the full 360° panorama. For DDAD, we exclude 2 out of the 6 cameras since they have more than 30% pixels occluded by the camera capture system. We crop both RGB images and depth maps to have 1920×870 resolution that is 180px and 210px cropped from the top for Argoverse and DDAD, respectively, to crop out a large portion of the sky and regions occluded by the ego-vehicle. For both datasets, we clip the maximum depth at 150m.

### 4.1.2 Implementation Details

**Evaluation Details.** In all experiments, we do not exploit any test-time augmentations (TTA), camera parameters, or other tricks and regularizations, in contrast to many previous methods [3, 13, 24, 38, 59]. This provides a more challenging setup, which allows us to show the effectiveness of iDisc. As depth estimation metrics, we utilize root mean square error (RMS) and its log variant (RMS_{log}), absolute error in log-scale (Log_{10}A), absolute (A.Rel) and squared (S.rel) mean relative error, the percentage of inlier pixels (δ_{t}) with threshold 1.25\(^{t}\), and scale-invariant error in log-scale (SI_{log}): 
\[ 100\sqrt{\text{Var}(_{Log})} \]

The maximum depth for NYU and all zero-shot testing in indoor datasets, specifically SUN-RGBD and Diode Indoor, is set to 10m, while for KITTI it is set to 80m and for Argoverse and DDAD to 150m. Zero-shot testing is performed by evaluating a model trained on either KITTI or NYU and tested on either outdoor or indoor datasets, respectively, without additional fine-tuning. For surface normals estimation, the metrics are mean (Mean) and median (Med) absolute error, RMS angular error, and percentages of inlier pixels with thresholds at 11.5°, 22.5°, and 30°. GT-based mean depth rescaling is applied only on Diode Indoor for all methods since the dataset presents largely scale-equivariant scenes, such as plain walls with tiny details.

**Training Details.** We implement iDisc in PyTorch [37]. For training, we use the AdamW [34] optimizer (\(\beta_{1} = 0.9, \beta_{2} = 0.999\)) with an initial learning rate of 0.0002 for every experiment, and weight decay set to 0.02. As a scheduler, we exploit Cosine Annealing starting from 30% of the training, with final learning rate of 0.00002. We run 45k optimization iterations with a batch size of 16. All backbones are initialized with weights from ImageNet-pretrained models. The augmentations include both geometric (random rotation and scale) and appearance (random brightness, gamma, saturation, hue shift) augmentations. The required training time amounts to 20 hours on 4 NVidia Titan RTX.

### 4.2. Comparison with the State of the Art

**Indoor Datasets.** Results on NYU are presented in Table 1. The results show that we set the new state of the art on the benchmark, improving by more than 6% on RMS and 9% on A.Rel over the previous Sota. Moreover, results highlight

![Figure 4. Attention maps on NYU for three different IDRs. Each row presents the attention map of a specific IDR for four test images. Each discrete representation focuses on a specific high-level concept. The first two rows pertain to IDRs at the lowest resolution while the last corresponds to the highest resolution. Best viewed on a screen and zoomed in.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Encoder</th>
<th>δ_{1}</th>
<th>δ_{2}</th>
<th>δ_{3}</th>
<th>RMS</th>
<th>A Rel</th>
<th>Log_{10}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigen et al.</td>
<td>-</td>
<td>0.769</td>
<td>0.950</td>
<td>0.988</td>
<td>0.641</td>
<td>0.158</td>
<td></td>
</tr>
<tr>
<td>DORN [13]</td>
<td>R101</td>
<td>0.828</td>
<td>0.965</td>
<td>0.992</td>
<td>0.509</td>
<td>0.115</td>
<td>0.051</td>
</tr>
<tr>
<td>VNL [53]</td>
<td>-</td>
<td>0.875</td>
<td>0.976</td>
<td>0.994</td>
<td>0.416</td>
<td>0.108</td>
<td>0.048</td>
</tr>
<tr>
<td>BTA [24]</td>
<td>D161</td>
<td>0.885</td>
<td>0.978</td>
<td>0.994</td>
<td>0.392</td>
<td>0.110</td>
<td>0.047</td>
</tr>
<tr>
<td>Adalfar [3]</td>
<td>MVIT</td>
<td>0.903</td>
<td>0.984</td>
<td>0.997</td>
<td>0.364</td>
<td>0.103</td>
<td>0.044</td>
</tr>
<tr>
<td>DAV [23]</td>
<td>DD22</td>
<td>0.882</td>
<td>0.980</td>
<td>0.996</td>
<td>0.412</td>
<td>0.108</td>
<td></td>
</tr>
<tr>
<td>Long et al.</td>
<td>[33]</td>
<td>HR48</td>
<td>0.890</td>
<td>0.982</td>
<td>0.996</td>
<td>0.377</td>
<td>0.101</td>
</tr>
<tr>
<td>TransDepth [34]</td>
<td>VITB</td>
<td>0.900</td>
<td>0.983</td>
<td>0.996</td>
<td>0.365</td>
<td>0.106</td>
<td>0.045</td>
</tr>
<tr>
<td>DPT [1]</td>
<td>VITB</td>
<td>0.904</td>
<td>0.988</td>
<td>0.998</td>
<td>0.357</td>
<td>0.110</td>
<td>0.045</td>
</tr>
<tr>
<td>PiDepth [38]</td>
<td>R101</td>
<td>0.830</td>
<td>0.971</td>
<td>0.995</td>
<td>0.450</td>
<td>0.130</td>
<td>0.056</td>
</tr>
<tr>
<td>NeWCRF [59]</td>
<td>SwinL</td>
<td>0.922</td>
<td>0.992</td>
<td>0.998</td>
<td>0.334</td>
<td>0.095</td>
<td>0.041</td>
</tr>
<tr>
<td>LocalBasis [4]</td>
<td>MVIT</td>
<td>0.967</td>
<td>0.987</td>
<td>0.998</td>
<td>0.357</td>
<td>0.099</td>
<td>0.042</td>
</tr>
<tr>
<td>Ours</td>
<td>R101</td>
<td>0.892</td>
<td>0.983</td>
<td>0.995</td>
<td>0.380</td>
<td>0.109</td>
<td>0.046</td>
</tr>
<tr>
<td>EB5</td>
<td>MVIT</td>
<td>0.903</td>
<td>0.986</td>
<td>0.997</td>
<td>0.369</td>
<td>0.104</td>
<td>0.044</td>
</tr>
<tr>
<td>SwinT</td>
<td>0.954</td>
<td>0.983</td>
<td>0.996</td>
<td>0.377</td>
<td>0.109</td>
<td>0.045</td>
<td></td>
</tr>
<tr>
<td>SwinB</td>
<td>0.926</td>
<td>0.989</td>
<td>0.997</td>
<td>0.327</td>
<td>0.091</td>
<td>0.039</td>
<td></td>
</tr>
<tr>
<td>SwinL</td>
<td>0.940</td>
<td>0.993</td>
<td>0.999</td>
<td>0.313</td>
<td>0.086</td>
<td>0.037</td>
<td></td>
</tr>
</tbody>
</table>


21482
how iDisc is more sample-efficient than other transformer-based architectures \cite{3,4,41,54,59} since we achieve better results even when employing smaller and less heavily pretrained backbone architectures. In addition, results show a significant improvement in performance with our model instantiated with a full-convolutional backbone over other full-convolutional-based models \cite{12,13,22,24,38}. Table 2 presents zero-shot testing of NYU models on SUN-RGBD and Diode. In both cases, iDisc exhibits a compelling generalization performance, which we argue is due to implicitly learning the underlying patterns, namely, IDRs, of indoor scene structure via the ID module.

Qualitative results in Fig. 3 emphasize how the method excels in capturing the overall scene complexity. In particular, iDisc correctly captures discontinuities without depth over-excitation due to chromatic edges, such as the sink in row 1, and captures the right perspective between foreground and background depth planes such as between the bed (row 2) or sofa (row 3) and the walls behind. In addition, the model presents a reduced error around edges, even when compared to higher-resolution models such as \cite{3}. We argue that iDisc actually reasons at the pattern level, thus capturing better the structure of the scene. This is particularly appreciable in indoor scenes, since these are usually populated by a multitude of objects. This behavior is displayed in the attention maps of Fig. 4. Fig. 4 shows how IDRs at lower resolution capture specific components, such as the relative position of the background (row 1) and foreground objects (row 2), while IDRs at higher resolution behave as depth refiners, attending typically to high-frequency features, such as upper (row 3) or lower borders of objects. It is worth noting that an IDR attends to the image borders when the particular concept it looks for is not present in the image. That is, the borders are the last resort in which the IDR tries to find its corresponding pattern (e.g., row 2, col. 1).

**Outdoor Datasets.** Results on KITTI in Table 3 demonstrate that iDisc sets the new SotA for this primary outdoor dataset, improving by more than 3% in RMS and by 0.9% in \(\delta_{0.5}\) over the previous SotA. However, KITTI results present saturated metrics. For instance, \(\delta_{3}\) is not reported since every method scores \(>0.99\), with recent ones scoring 0.999. Therefore, we propose to utilize the metric \(\delta_{0.5}\), to better convey meaningful evaluation information. In addition, iDisc performs remarkably well on the highly competitive official KITTI benchmark, ranking 3rd among all methods and 1st among all published MDE methods.

Moreover, Table 4 shows the results of methods trained and evaluated on the splits from Argoverse and DDAD proposed in this work. All methods have been trained with the same architecture and pipeline utilized for training on KITTI. We argue that the high degree of sparseness in GT of the two proposed datasets, in contrast to KITTI, deeply affects windowed methods such as \cite{3,59}. Qualitative results in Fig. 5 suggest that the scene level discretization leads to retaining small objects and sharp transitions between foreground objects and background: background in row 1, and boxes in row 2. These results show the better ability of iDisc to capture fine-grained depth variations on close-by and similar objects, including crowd in row 3. Zero-shot testing from KITTI to DDAD and Argoverse are presented in Supplement.

**Surface Normals Estimation.** We emphasize that the proposed method has more general applications by testing iDisc on a different continuous dense prediction task such as surface normals estimation. Results in Table 5 evidence that we
Figure 5. **Qualitative results on KITTI.** Three zoomed-in crops of different test images are shown. The comparisons show the ability of iDisc to capture small details, proper background transition, and fine-grained variations in, e.g., crowded scenes. Best viewed on a screen.

### Table 5. Comparison of surface normals estimation methods on NYU official test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>11.5°</th>
<th>22.5°</th>
<th>30°</th>
<th>RMS</th>
<th>Mean</th>
<th>Med</th>
<th>Lower is better</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURGE [49]</td>
<td>0.473</td>
<td>0.689</td>
<td>0.766</td>
<td>20.6</td>
<td>12.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeoNet [39]</td>
<td>0.484</td>
<td>0.481</td>
<td>0.795</td>
<td>26.9</td>
<td>19.0</td>
<td>11.8</td>
<td></td>
</tr>
<tr>
<td>PAP [51]</td>
<td>0.488</td>
<td>0.722</td>
<td>0.798</td>
<td>25.5</td>
<td>18.6</td>
<td>11.7</td>
<td></td>
</tr>
<tr>
<td>GeoNet++ [40]</td>
<td>0.502</td>
<td>0.732</td>
<td>0.807</td>
<td>26.7</td>
<td>18.5</td>
<td>11.2</td>
<td></td>
</tr>
<tr>
<td>Bae et al. [1]</td>
<td>0.622</td>
<td>0.793</td>
<td>0.852</td>
<td>23.5</td>
<td>14.9</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>0.638</td>
<td>0.798</td>
<td>0.856</td>
<td>22.8</td>
<td>14.6</td>
<td>7.3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EDD</th>
<th>ISD</th>
<th>AFP</th>
<th>MSDA</th>
<th>δ↑</th>
<th>RMS</th>
<th>A.Rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.890</td>
<td>0.370</td>
<td>0.104</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>0.905</td>
<td>0.367</td>
<td>0.102</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>0.919</td>
<td>0.340</td>
<td>0.096</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>0.931</td>
<td>0.319</td>
<td>0.091</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>0.931</td>
<td>0.326</td>
<td>0.091</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>0.934</td>
<td>0.319</td>
<td>0.088</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.930</td>
<td>0.319</td>
<td>0.089</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.940</td>
<td>0.313</td>
<td>0.086</td>
</tr>
</tbody>
</table>

Table 5. Comparison of surface normals estimation methods on NYU official test set. iDisc architecture and training pipeline is the same as the one utilized for indoor depth estimation.

set the new state of the art on surface normals estimation. It is worth mentioning that all other methods are specifically designed for normals estimation, while we keep the same architecture and framework from indoor depth estimation.

### 4.3. Ablation study

The importance of each component introduced in iDisc is evaluated by ablating the method in Table 6.

**Depth Discretization.** Internal scene discretization provides a clear improvement over its explicit counterpart (row 3 vs. 2), which is already beneficial in terms of robustness. Adding the MSDA module on top of explicit discretization (row 5) recovers part of the performance gap between the latter and our full method (row 8). We argue that MSDA recovers a better scene scale by refining feature maps at different scales at once, which is helpful for higher-resolution feature maps.

**Component Interactions.** Using either the MSDA module or the AFP module together with internal scene discretization results in similar performance (rows 4 and 6). We argue that the two modules are complementary, and they synergize when combined (row 8). The complementarity can be explained as follows: in the former scenario (row 4), MSDA preemptively refines feature maps to be partitioned by the non-adaptive clustering, that is, by the IDR priors described in Sec. 3, while on latter one (row 6), AFP allows the IDRs to adapt themselves to partition the unrefined feature space properly. Row 7 shows that the architecture closer to the one in [32], particularly random initialization, hurts performance since the internal representations do not embody any domain-specific prior information.

### 5. Conclusion

We have introduced a new module, called Internal Discretization, for MDE. The module represents the assumption that scenes can be represented as a finite set of patterns. Hence, iDisc leverages an internally discretized representation of the scene that is enforced via a continuous-discrete-continuous bottleneck, namely ID module. We have validated the proposed method, without any TTA or tricks, on the primary indoor and outdoor benchmarks for MDE, and have set the new state of the art among supervised approaches. Results showed that learning the underlying patterns, while not imposing any explicit constraints or regularization on the output, is beneficial for performance and generalization. iDisc also works out-of-the-box for normal estimation, beating all specialized SotA methods. In addition, we propose two new challenging outdoor dataset splits, aiming to benefit the community with more general and diverse benchmarks.

**Acknowledgment.** This work is funded by Toyota Motor Europe via the research project TRACE-Zürich.
References


[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 7th International Conference on Learning Representations, ICLR 2019, 11 2017. 6


