

# Efficient Multitask Dense Predictor via Binarization

Yuzhang Shang<sup>1</sup>, Dan Xu<sup>2</sup>, Gaowen Liu<sup>3</sup>, Ramana Rao Kompella<sup>3</sup>, Yan Yan<sup>1,†</sup>  
<sup>1</sup>Illinois Institute of Technology, <sup>2</sup>HKUST, <sup>3</sup>Cisco Research

yzshawn@outlook.com, danxu@cse.ust.hk, {ga

## Abstract

Multi-task learning for dense prediction has emerged as a pivotal area in computer vision, enabling simultaneous processing of diverse yet interrelated pixel-wise prediction tasks. However, the substantial computational demands of state-of-the-art (SoTA) models often limit their widespread deployment. This paper addresses this challenge by introducing network binarization to compress resource-intensive multi-task dense predictors. Specifically, our goal is to significantly accelerate multi-task dense prediction models via Binary Neural Networks (BNNs) while maintaining and even improving model performance at the same time. To reach this goal, we propose a Binary Multi-task Dense Predictor, *Bi-MTPD*, and several variants of *Bi-MTPD*, in which a multi-task dense predictor is constructed via specified binarized modules. Our systematic analysis of this predictor reveals that performance drop from binarization is primarily caused by severe information degradation. To address this issue, we introduce a deep information bottleneck layer that enforces representations for downstream tasks satisfying Gaussian distribution in forward propagation. Moreover, we introduce a knowledge distillation mechanism to correct the direction of information flow in backward propagation. Intriguingly, one variant of *Bi-MTPD* outperforms full-precision (FP) multi-task dense prediction SoTAs, ARTC [2] (CNN-based) and InvPT [50] (ViT-Based). This result indicates that *Bi-MTPD* is not merely a naive trade-off between performance and efficiency, but is rather a benefit of the redundant information flow thanks to the multi-task architecture. Code is available at [BiMTDP](#).

## 1. Introduction

There is a growing trend in the computer vision community where dense prediction tasks are processed in a multi-task learning manner, such as semantic segmentation, monocular depth estimation, and human parsing [28, 46, 47, 49].

<sup>†</sup>Corresponding author

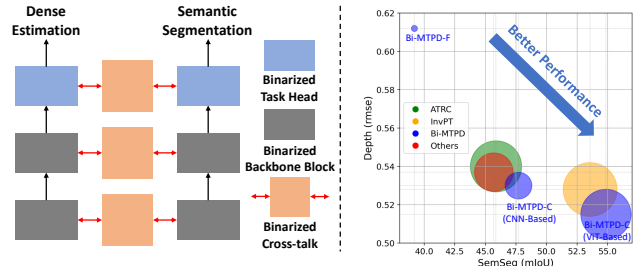


Figure 1. **(Left)** A conceptual illustration of binary dense predictions in a multi-task manner. In contrast to approaching a series of relevant tasks individually, the multitask model benefits from information supplementation among different tasks via cross-talk structures, but the cumbersome cross-talk modules also add additional computational burden. **(Right)** Performance summary on NYUD-v2. X-axis and Y-axis denote the performance on depth estimation (lower is better) and segmentation (higher is better), respectively. Size of dots denotes FLOPs. ATRC [2] and InvPT [50] are previous CNN-based and ViT-based SoTAs, respectively.

Benefited from the information supplementation mechanism via cross-talk structures in the multi-task models, the overall performance for the series of dense prediction tasks has been greatly improved [17] (see Fig. 1). However, the computational demands of State-of-the-Art (SoTA) multi-task dense prediction models, which process multiple complex pixel-wise tasks concurrently, are substantial. This high computational requirement limits their application in resource-constrained environments like autonomous driving, robotics, and virtual reality. Our goal is to optimize these heavy SoTA models for edge devices, balancing speed with performance.

Several strategies for neural network compression have been explored, including pruning [12], network quantization [15, 37, 40] and knowledge distillation [14]. Notably, network binarization, a form of quantization, minimizes weights and activations to  $\pm 1$ , enabling the replacement of computationally intensive inner-product operations in full-precision networks with more efficient xnor-bitcount operations in Binary Neural Networks (BNNs) [15]. Binarization theoretically reduces memory costs by  $32\times$  and increases inference speed by  $64\times$ , making BNNs suitable for edge-device.

While BNNs have shown impressive results in image classification, achieving nearly full-precision ResNet-level accuracy [25], their application has largely been limited to small-scale models, overlooking other computationally intensive computer vision tasks [15, 23, 25, 32, 39]. Extending BNNs to larger models should be the next step. However, this expansion has been hindered by issues such as overfitting [19, 38, 39] and information degradation [30]. Techniques effective in full-precision models, like label smoothing [34], dropout [42], and mixup [52] have less effect on BNNs [18, 38, 39]. Furthermore, SoTA multitask dense prediction tasks often require deep and complex models, equipped with multi-modality fusion structures [2, 47, 50], exacerbating the challenges in implementing binarization effectively.

The primary barrier to applying binarization in multitask dense prediction tasks is the significant degradation of information flow in deep models [15, 30, 32], leading to reduced performance. To address this issue, we first propose a Binary Multitask Dense Predictor (Bi-MTPD) baseline, where a multi-task dense predictor is formulated via binarized modules. Based on a thorough review of this baseline, we conclude that the binarization operation destroys the information flows in multi-task models, and thus representations for downstream tasks are not informative compared with their full-precision counterparts. To tackle this problem, we update Bi-MTPD with additional information flow calibration mechanisms in two directions. First, we implement variational information bottleneck enforcing the embeddings to follow Gaussian distribution with sparsity in forward propagation, in order to filter out the task-irrelevant factors. Second, we leverage the existing FP models via feature-based knowledge distillation to calibrate the gradient of the binary network in backward pass.

The benefits of Bi-MTPD can be analyzed from two orthogonal perspectives. On one hand, from the perspective of network binarization, the accomplishment of bridging binarization with the multitask dense prediction framework testifies that Bi-MTPD can effectively supplement information, and consequently improve the performance of the individual binary models. On the other hand, from the perspective of multitask dense prediction task, accelerating those cumbersome models is profitable to design more effective and efficient cross-talk modules in it, as shown in Fig. 1. Since existing dense prediction models have severe limitation in modelling the cross-talk modules due to their heavy utilization of convolution operation, it is critical for the multitask dense predictions to learn interactions and inference covering various scopes of the multitask context via the cross-talk mechanism [2, 28, 47, 49, 50, 53]. Intriguingly, a variant of Bi-MTPD outperforms SoTA approach ATRC [2] by 4% over the segmentation task while remaining 43% faster in speed, implying that our proposed method

is not a naive trade-off between performance and efficiency. By empirically investigating this “free lunch” achievement, we conclude that the win-win outcome is benefited from our designed information supplementation mechanism which strengthens the representation ability of the binary model.

## 2. Related Work

**Multitask Dense Prediction.** Multi-Task Learning (MTL) methods can be generally categorized into two main paradigms in terms of the way where model learns shared representations: hard and soft parameter sharing. Hard parameter sharing characterizes architectures which typically share the first hidden representations among the tasks while branching to independent task-specific representations at a later stage. Most approaches split to task-specific heads at a single branch point [5, 17, 21, 36]. However, such naive branching can be sub-optimal, raising interest in mechanisms that allow for finely branched architectures [26, 45]. As a result, in soft parameter sharing, each task is assigned its own set of parameters and a feature sharing mechanism realize the cross-talk as demonstrated in Fig 1. The following works devise the cross-talk mechanisms focusing on the locations in the network where information or features are exchanged or shared between tasks. Apart from the locations, the feature sharing modules are also widely studied. For example, feature fusing can be introduced along the entire network depth [9, 28]; PAD-Net [49] uses multi-modal distillation to enhance task-specific predictions, in which information flow from each source to target task is regulated with a sigmoid-activated gate function; and MTI-Net [47] combines the multi-modal distillation module of PAD-Net with a multi-scale refinement scheme to facilitate cross-task talk at multiple scales.

Although increasing the number of cross-talk modules intuitively benefits the overall performance of the models, computational cost is often an obstacle. To handle this issue, ATRC [2] introduces NAS [55] to automatically design an efficient information fusing modules. From the perspective of the efficient representation cross-talk, our proposed models with the binarization module can be interpreted as a new pathway to feature fusing within a notably lower inference speed level.

**Neural Network Binarization.** As pioneers, [15] use the sign function to quantize weights and activations to  $\pm 1$ , initiating the trends of studies of network binarization. To tackle the vanishing gradient issue induced by the binarization operations, the straight-through estimator (STE) [1] is introduced for the gradient approximation. Rooted in this archetype, considerable studies contribute to improving the performance of BNNs, particularly on ImageNet. For example, [23] propose Bi-Real introducing double residual connections with FP downsampling layers to mitigate the excessive gradient vanishing issue caused by binariza-

tion, and consequently demonstrate that delicately ing additional connections within BNNs benefits t dient propagation. [13] design a proxy matrix as of the latent parameter space to guide the alignmen weights with different bits by recovering the smooth BNNs. In summary, a large number of methods h tended the boundary of the network binarization w curacy over classification (*e.g.*, ReActNet [25] within com- parable FLOPs of binary ResNet-18 achieves 65.9% Top-1 accuracy on ImageNet, while full-precision version is only 70.5%).

However, most of those works validate their effec- tiveness over classification with relatively small architec- tures (mostly ResNet18 and ResNet34). Meanwhile, the network-based models for dense prediction tasks are big- ger and deeper than those toy models, as the information flow in networks is severely degraded. Consequently, di- rectly implementing existing binarization methods can not achieve supposed success. To mitigate the degradation, we propose to binarize those dense prediction models in a mul- titask way.

### 3. Multitask Network Binarization

#### 3.1. Binary Neural Network

To begin with, we briefly review the general idea of binary neural networks (BNNs) in [7, 15]. Here, we only elabo- rate the speedup mechanism and the degradation of infor- mation flow of the binarization. We define a full-precision (FP) neural network with  $K$  layers,  $f(\mathbf{x}) = (\mathbf{W}^K \times \sigma \times \mathbf{W}^{K-1} \dots \sigma \times \mathbf{W}^1)(\mathbf{x})$ , where  $\mathbf{x}$  is the input sample and  $\mathbf{W}^k : \mathbb{R}^{d_{k-1}} \mapsto \mathbb{R}^{d_k}$  ( $k = 1, \dots, K$ ) stands for the weight matrix connecting the  $(k - 1)$ -th and the  $k$ -th layer, with  $d_{k-1}$  and  $d_k$  representing the sizes of the input and output of the  $k$ -th network layer, respectively. The  $\sigma(\cdot)$  function performs element-wise activation for the feature maps.

BNNs vary from FP neural networks in terms of the for- ward operation and the backward gradient approximation. Specifically, in the forward propagation, the BNN main- tains FP latent weights  $\mathbf{W}_F$  for gradient updates, and the  $k$ -th weight matrix  $\mathbf{W}_F^k$  is binarized into  $\pm 1$ , obtaining the binary weight matrix  $\mathbf{W}_B^k$  via the binarize function  $sign(\cdot)$ , *i.e.*  $\mathbf{W}_B^k = sign(\mathbf{W}_F^k)$ . Then the intermediate activation map (full-precision) of the  $k$ -th layer is produced by  $\mathbf{A}_F^k = \mathbf{W}_B^k \mathbf{A}_B^{k-1}$ . The same quantization method is used to binar- ize the full-precision activation map as  $\mathbf{A}_B^k = sign(\mathbf{A}_F^k)$ , and the whole forward pass of binarization is performed by iterating this process for  $L$  times, as shown in Fig. 2. For BNNs, the weights and activations are 1-bit, by which the network is accelerated 32 times in terms of memory cost. Importantly, inference of BNN is accelerated 64 times, as the FP multiplication in FP networks is replaced by Xnor- Bitcount in BNNs.

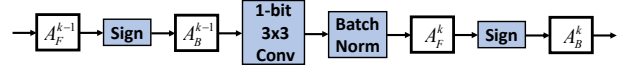


Figure 2. A general illustration of the forward propagation of the  $k$ -th layer in the BNN.

In the backward propagation, the main challenge is that the pervasive sign functions are theoretically non- differentiable, and thus extremely destroy the information flow via gradient the propagation. To address this prob- lem, a large number of researchers [31] widely exploit the straight-through estimator (STE) [1] to numerically approx- imate the derivative of the whole BNN, *i.e.*

$$\text{Backward: } \frac{\partial \mathcal{L}}{\partial x} = \begin{cases} \frac{\partial \mathcal{L}}{\partial sign(x)} & |x| \leq 1 \\ 0 & |x| > 1. \end{cases} \quad (1)$$

It is worth noting that we do not implement the aforemen- tioned vanilla approximation method in practice, while we utilize the prevalent Bi-Real [23] and IR-Net [32] to gradu- ally approximate the *sign* function, which have been proven to be better estimation approaches [29, 30].

Even though numerous methods have been proposed to eliminate the deterioration of the information flow induced by the binarization, the deterioration is still inevitable due to the severe accuracy loss of weights, activations and gra- dients [29, 30]. Consequently, binarization destroys the per- formance of the complicated computer vision models.

#### 3.2. Multitask Dense Predictor

After deploying binarization techniques in the models for dense prediction tasks, the performance of the binarized models is unacceptable, as shown in Fig. 1 and the binary single result in Table. 3. Since the architectures of those SoTA dense prediction models are relatively heavier and deeper (*e.g.*, HRNet-48 or ResNet-101 with a task-specific head [27]) than the ones for classification (*e.g.*, ResNet- 18 with a fully-connected layer as the classification head). Moreover, the information passing in the binary models via back-propagation, especially in deep models, is notoriously inefficient [23].

Dense prediction tasks can mutually supplement infor- mation, *e.g.*, surface normals and depth can directly be de- rived from each other, which can be modeled as the regular- ization of each other [47]. The relevancy among dense pre- diction tasks is worth being utilized to improve the overall performance of models. For example, before the deep learn- ing era, pioneering work [11] utilizes RGB-D images with depth information to predict scene semantics to improve the quality of the prediction. In the deep learning era, recent attention-based multitask learning methods [2, 47, 49] ex- plicitly and implicitly distill information from other tasks as a complementary signal to improve the targeted task perfor- mance. Briefly, the above-mentioned methods are achieved

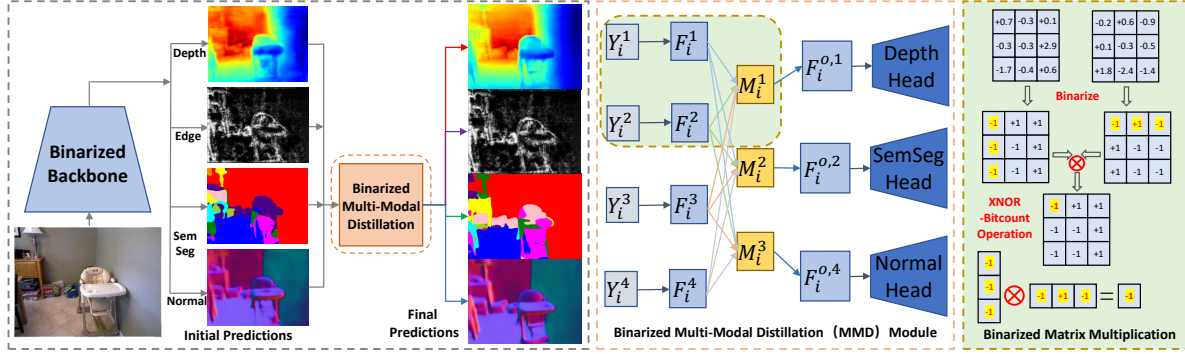


Figure 3. **(Left)** The illustration of the baseline multitask framework. **(Middle)** The designed MMD modules for binary representations. Importantly, the MMD module can pass information among different predictions, acting as a cross-talk mechanism. **(Right)** As all fundamental modules in Bi-MTPD baseline are binarized, inferences can be performed by complete Bool operations, which are very computationally cheap.

by combining an existing backbone network for initial task predictions with a subsequent decoding process, as shown in Fig. 3 (Left).

Specifically, the shared features extracted by the backbone network are then processed by a set of task-specific heads, which produce a series of initial predictions for  $T$  tasks, *i.e.*  $\{Y_i^k\}$  ( $k = 1, \dots, T$ ) (the backbone and the task-specific heads are referred as the front-end of the network [47]). Transforming and binarizing  $Y_i^t$  into the form of a 1-bit feature map, we obtain a set of corresponding binary feature maps of the scene, *i.e.*  $\{F_{B,i}^t\}$  ( $t = 1, \dots, T$ ) which are more task-aware than the shared binary features of the backbone network. The information from these task-specific feature representations is then fused via a *multi-modal distillation via binarized attention mechanism* before making the final task predictions. As previous work featured, our method is also task-saleable. Especially, it is possible that some tasks are only predicted in the front-end of the network (initial prediction). The initial tasks are also called auxiliary tasks since they serve as proxies in order to improve the performance of the final tasks, as shown in Fig. 3.

**Multi-Modal Distillation (MMD) via Binarized Attention Mechanism.** The multi-modal distillation module is the key in the multi-task dense prediction model. Specifically, we utilize the attention mechanism for guiding the information passing between the binary feature maps generated from different modalities for different tasks. Since the passed information flow is not always helpful, the attention can act as a gate function to control the flow, in other words, to make the network automatically learn to focus or to ignore information from other binary features [47, 49]. Including the binarization operations, we can formalize the MMD via binarized attention as follows. While passing information to the  $k$ -th task, we first obtain a binarized atten-

tion map  $\mathbf{A}_{B,i}^k$ , *i.e.*,

$$\mathbf{A}_{B,i}^k \leftarrow \text{bool}(\mathbf{W}_B^k \otimes \mathbf{F}_{B,i}^k) \quad (2)$$

where  $\mathbf{W}_B^k$  is the parameters of the binarized convolution layer,  $\mathbf{F}_{B,i}^k$  is the binary feature map of the initial prediction, and  $\otimes$  denotes convolution operation. Then the information is passed with the attention map controlled as follows:

$$\mathbf{F}_{B,i}^{o,k} \leftarrow \text{sign} \left[ \mathbf{F}_{B,i}^k + \sum_{t=1, t \neq k}^T \mathbf{A}_{B,i}^k \odot (\mathbf{W}_{B,t} \otimes \mathbf{F}_{B,i}^t) \right] \quad (3)$$

where  $\odot$  element-wise multiplication. The general demonstration of the distillation process is presented in Fig. 3 Left. The output binary feature map  $\mathbf{F}_{B,i}^{o,k}$  is then used by the head for the corresponding  $t$ -th task in Fig. 3 Right. By using the task-specific distillation activations, the network can preserve more information for each task [2, 47, 49], which especially benefits the BNNs where the deteriorated information flow mainly induce the performance drop.

On the other hand, multitask dense prediction models benefit from the network binarization in terms of performance. Although these multitask models have achieved a promising performance, they are still limited by the nature of convolution-based distillation modules that are heavily used in a multi-scale way, which model critical spatial and task-related interactions in relatively local perceptive fields [50]. Theoretically, more distillation modules in different network nodes can contribute to model performance, yet we cannot unrestrictedly add distillation modules to the existing model due to the computational limitation. Fortunately, with the saved computational cost of the binary networks, we can implement additional distillation modules in our model.

**Binary Baseline for Multitask Dense Prediction, Bi-MTPD.** To obtain dense predictions with BNNs under a multitask learning framework, we practically binarize

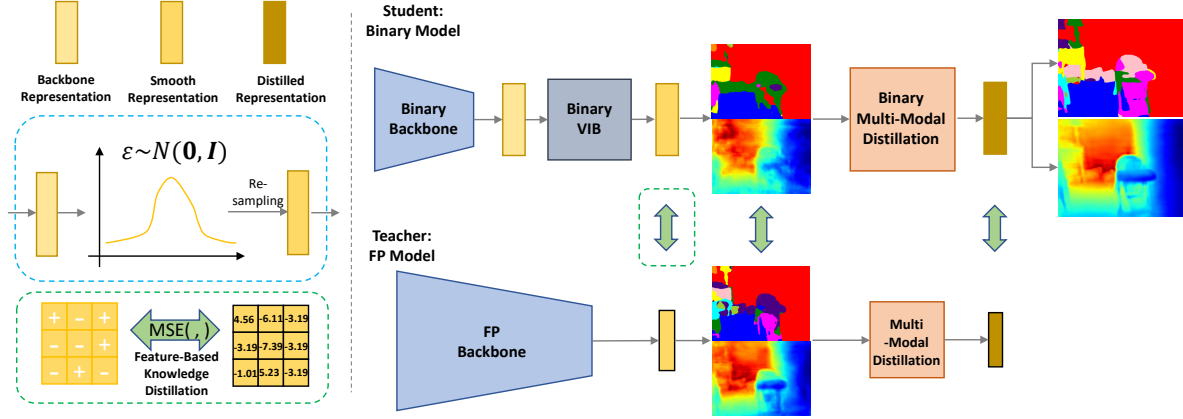


Figure 4. **The pipeline of Bi-MTPD.** We introduce a VIB layer after the backbone network to filter-out the nuisance factors which may lead to model overfitting issue in the forward propagation. In addition, we deploy a feature-based knowledge distillation mechanism to guide the optimization direction in the backward propagation.

the MTI-Net [47] as the binary baseline. Specifically, the main modules in the full-precision MTI-Net, including backbone, heads, and multi-modal distillation module, are replaced with binary modules (both weights and activations are 1-bit). We call this baseline as Bi-MTPD.

### 3.3. Information Flow Supplementation

Although we build a fully binarized baseline Bi-MTPD for multitask dense predictions and train the pipeline with common techniques, the performance is still of major concern. The baseline suffers an immense information degradation as the *nuisance factors are over-fitted in the forward propagation and optimization directions severely mismatch in the backward propagation.* To solve these problems, in this section, we further propose the variant of Bi-MTPD, Bi-MTPD-F. Specifically, we introduce a variational information bottleneck (VIB) layer after the output of the shared binary backbone to precisely enforces the feature extractor to preserve the minimal sufficient information of the input data. As well, we deploy the feature-based knowledge distillation to guided the optimization direction. We present more details in the following section.

**Variational Information Bottleneck for Filter-Out Nuisance Factors.** Obtaining the initial binary representations of input images by the shared backbone, we need to train a series of targeted heads to split them. A straightforward strategy is to feed these representations into the following MMD module. However, the binarized representations lack homogenization leading to model overfitting issue [48]. Therefore, the need to regularize the binarized representations, while the regularization would not to contaminate the information flow in the representations. Fortunately, the information bottleneck (IB) principle directly relates to compression with the best hypothesis that the data misfit and the model complexity should simultaneously be minimized [43, 48].

As the VIB could effectively capture the relevant parts and filter out the irrelevant ones from inputs, we design a novel VIB-based layer after the backbone. In particular, it explicitly enforces the feature extractor to preserve the minimal sufficient information of the input data. In other words, it can help ensure the information flow flexibly to learn clean representation for the targeted tasks. The objective function of our VIB-based classification can be formulated as a term of information loss, written as follows:

$$\mathcal{L}_{vib} = KL[p(\mathbf{Z} | \mathbf{A}_B), r(\mathbf{Z})], \quad (4)$$

where  $\mathbf{A}_B$  is the input binary backbone representation,  $\mathbf{B}$  is the latent representation variable,  $p(\mathbf{Z} | \mathbf{A}_B)$  is a multivariate Gaussian distribution, and  $r(\mathbf{Z})$  is a standard normal distribution. Generally, the latter is a regularization term controlling how much information of the input is filtered out. A more detailed discussion about the VIB for binarized models filtering out irrelevant information is in Supplemental Materials.

**Feature-based Knowledge Distillation for Guiding the Direction of Information Flow.** Distillation is a common and essential optimization approach to alleviate the performance drop of quantized models on ultra-low bit-width settings, which can be flexibly deployed for any architectures to utilize the knowledge of a full-precision teacher model [3, 16, 30, 51]. The usual practice is to distill the activations in a layerwise manner from the full-precision teacher to the quantized counterparts, *i.e.*,  $\mathbf{F}_{B,l}$  and  $\mathbf{F}_{FP,l}$  ( $l = 1, \dots, L$ , where  $L$  represents the number of network layers), respectively. We use the mean squared errors (MSE) as the distance function to measure the difference between corresponding from features student and teacher. The knowledge distillation loss can be written as follows:

$$\mathcal{L}_{kd} = \sum_{l=1}^L MSE(\mathbf{F}_{B,l}, \mathbf{F}_{FP,l}). \quad (5)$$

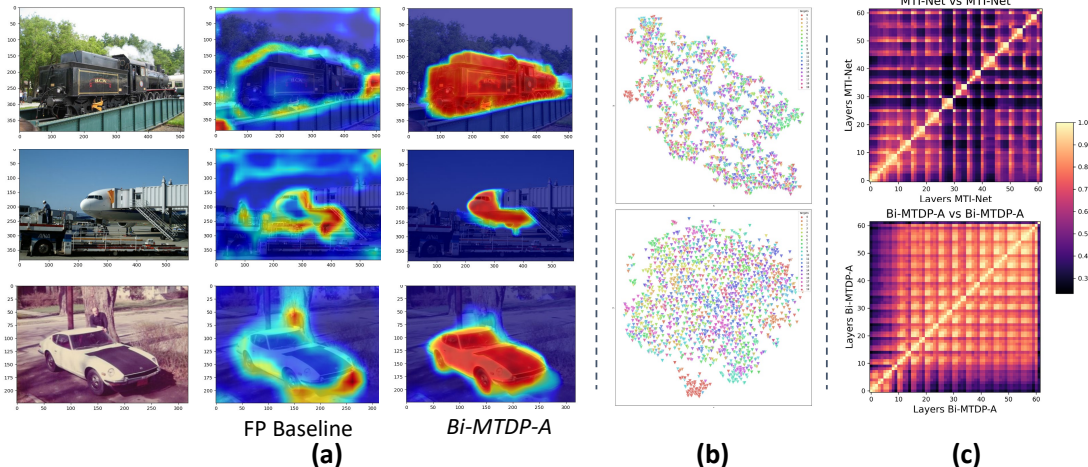


Figure 5. (a) Grad-cam visualization of feature maps of different multitask dense prediction methods. (b) t-SNE visualization of learned features of all 20 classes on Pascal-Context. (c) Centered Kernel Alignment analyzing the information flow within different networks.

### 3.4. Counter-Intuitive Results of Bi-MTDP-A

Intuitively, implementing binarization on FP network inevitably induces representations degradation, as the gradient of the *sign* function cannot be perfectly estimated [1]. Thus, binarized models are impossible to outperform their full-precision counterpart models. However, Bi-MTDP-C, a variant of Bi-MTDP (*i.e.*, full-precision backbone with only binarized multi-modal distillation) outperforms its fully FP version. Specifically, just binarizing the multi-modal distillation can simultaneously accelerate the model by  $\sim 39\%$  and improve the mIoU for segmentation by  $\sim 4\%$ , as shown in Table 1. This result demonstrates that our method is not a naive trade-off between model performance and efficiency but a powerful tool for boosting multitask dense predictors. This exciting ‘free-lunch’ achievement is even a bit of counter-intuitive. We speculate the reasons are that i) binarization on MMD can filter out task-irrelevant information; ii) and thus the information flow in the network is more effective. To testify this speculation, we conduct a series of experiments in two aspects, the representation ability of Bi-MTDP-C and information flow supplementation within the network.

**Qualitative Study of Learned Features with Bi-MTDP** To investigate the representation ability of Bi-MTDP-C and its FP counterparts, we visualize i) the feature maps behind the Binarized Multi-Modal Distillation (MMD) module in 2-D space via the t-SNE [44] algorithm, and ii) the regions where the network considers important via the Grad-Cam algorithm [35]. The results are shown in Fig. 5. It is clear that binarized model, Bi-MTDP-C is able to filter the irrelevant information out via the binarized attention module (see Fig. 5 (a)), and thus helps learn more discriminative features (see Fig. 5 (b)) resulting in higher quantitative results. Overall, the generated spatial feature maps for segmentation are better. The enhanced

representative ability can contribute to higher quantitative results.

**Analysis of Information Flow Supplementation within Network via Centered Kernel Alignment.** Analyzing distributional information flow within layers of neural networks is challenging because outputs of layers are distributed across a large number of neurons. Centered kernel alignment (CKA) [6, 22, 33] can address these challenges, by quantitatively comparing activations within or across networks. Specifically, for a network feed by  $m$  samples, CKA algorithm takes  $\mathbf{X} \in \mathbb{R}^{m \times p_1}$  and  $\mathbf{Y} \in \mathbb{R}^{m \times p_2}$  as inputs which are output activations of two layers (with  $p_1$  and  $p_2$  neurons respectively). Letting  $\mathbf{K} \triangleq \mathbf{X}\mathbf{X}^\top$  and  $\mathbf{L} \triangleq \mathbf{Y}\mathbf{Y}^\top$  denote the Gram matrices for the two layers CKA computes:

$$\text{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\text{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\text{HSIC}(\mathbf{K}, \mathbf{K})\text{HSIC}(\mathbf{L}, \mathbf{L})}}, \quad (6)$$

where HSIC is the Hilbert-Schmidt independence criterion [10]. Given the centering matrix  $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$  and the centered Gram matrices  $\mathbf{K}' = \mathbf{H}\mathbf{K}\mathbf{H}$  and  $\mathbf{L}' = \mathbf{H}\mathbf{L}\mathbf{H}$ ,  $\text{HSIC} = \frac{\text{vec}(\mathbf{K}')\text{vec}(\mathbf{L}')}{(m-1)^2}$ , the similarity between these centered Gram matrices. Importantly, CKA is invariant to orthogonal transformation of representations (including permutation of neurons), and the normalization term ensures invariance to isotropic scaling. These properties enable meaningful comparison and analysis of neural network hidden representations.

Therefore, we introduce CKA to study the information flow in the multitask dense prediction models. In the heatmap, the lighter the dot, the more similar the two corresponding layers. Higher similar score between two layers’ output representations means those two layers share more information. The results are presented in Fig. 5 (c), we can see that the similar scores among front layers and back lay-

Table 1. Results on NYUD-v2. Bi-MTPD-C: only implementing the binarization in the multi-modal distillation module, Bi-MTPD-F: fully-binarized model.

Model		SemSeg $\uparrow$	Depth $\downarrow$	Normal $\downarrow$	Bound $\uparrow$	float32	binary	FLOPs
Method	Backbone					Params (M)	Params (M)	(G)
Cross [28]	CNN	36.34	0.629	20.88	76.38	241.46	0	338.09
PAP [53]		36.72	0.617	20.82	76.42	189.10	0	256.86
PSD [54]		36.69	0.625	20.87	76.42	224.67	0	315.60
PAD [49]		36.61	0.627	20.85	76.38	170.98	0	230.91
MTI [47]		45.70	0.537	20.27	77.86	144.87	0	212.98
ATRC [2]		45.87	0.540	20.09	77.34	180.00	0	249.24
MTI + Bi-MTPD-F		39.20	0.612	21.04	76.86	6.45	138.42	18.07
MTI + Bi-MTPD-C		47.71	0.530	20.06	77.36	90.28	54.59	130.65
InvPT [50]	ViT-B	50.30	0.536	19.00	77.60	154.79	0	244.71
InvPT + Bi-MTPD-C		51.20(0.90 $\uparrow$ )	0.528(0.08 $\downarrow$ )	19.50(0.50 $\uparrow$ )	77.68(0.08 $\uparrow$ )	127.92	26.87	183.81
InvPT [50]	ViT-L	53.56	0.518	19.04	78.10	239.22	0	331.60
InvPT + Bi-MTPD-C		54.86(1.30 $\uparrow$ )	0.515(0.03 $\downarrow$ )	19.50(0.46 $\uparrow$ )	78.20(0.10 $\uparrow$ )	212.34	26.87	301.88

ers in Bi-MTPD-C is much higher than the ones in MTI-Net [47]. This indicates that Bi-MTPD-C is able to supplement information flow within network, and thus boost the model performance.

## 4. Experiments

In this section, we conducted comprehensive experiments to evaluate our proposed method on two datasets for dense prediction tasks: PASCAL Context [8] and NYUD-v2 [41]. We first describe the implementation details of Bi-MTPD, and then compare our method with SoTA binary neural networks in the task of object detection to demonstrate superiority of the proposed method. Finally, we validate the effectiveness of information bottleneck and feature-based knowledge distillation by a series of ablative studies.

### 4.1. Datasets, Evaluation, Implementation Details

**Datasets.** *PASCAL-Content* is a popular dataset for dense prediction tasks. We use the split from PASCAL-Context which has annotations for semantic segmentation, human part segmentation, semantic edge detection, surface normals prediction and saliency detection. Note that some annotations were distilled by [27] using pre-trained SoTA models [4]. *NYUD-v2* contains various indoor scenes such as offices and living rooms with 795 training and 654 testing images. It provides different dense labels, including semantic segmentation, monocular depth estimation, surface normal estimation and object boundary detection.

**Evaluation.** Semantic segmentation (Semseg) and human parsing (Parsing) are evaluated with mean Intersection over Union (mIoU); monocular depth estimation (Depth) is evaluated with Root Mean Square Error (RMSE); surface normal estimation (Normal) is evaluated by the mean error (mErr) of predicted angles; saliency detection (Saliency) is evaluated with maximal F-measure (maxF); object boundary detection (Boundary) is evaluated with the optimal-dataset-scale F-measure (odsF). To evaluate the model efficiency *w.r.t.* memory cost and inference speed, we adopt the number of parameters and FLOPs for a single round of

the model inferring an input image.

**Implementation details.** We build our approach on the most prevalent backbone architecture, *i.e.*, HRNet as previous SoTA methods [2, 47]. The task-specific heads are also implemented as two basic residual blocks, *i.e.*, binarized BasicBlock and binarized Bottleneck with additional binary shortcuts as BiReal-Net [23]. We use  $\ell_1$  loss for depth estimation and cross-entropy loss for semantic segmentation on NYUD-v2. As in the prior work, the edge detection task is trained with a positive weighted  $w_{pos} = 0:95$  binary cross-entropy loss. We do not adopt a particular loss weighing strategy on NYUD-v2, but simply sum the losses together. On PASCAL, we reuse the training setup from [47] to facilitate a fair comparison. We reuse the loss weights from there. The initial task predictions in the front-end of the network use the same loss weighing as the final task predictions. We refer to the supplementary material for further implementation details. Importantly, we use Adam optimizer [20] in training, but with different learning rates for binary parameters (1e-5) and FP parameters (1e-4), as Adam with a larger learning rate for binary parameters can lead to better training results [24]. Note that our project is based on the codebases for MTI-Net [47] and more details can be found in the **codes** in the **Supplemental Materials**.

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

Tabs. 1 and 2 present a comparative analysis of the proposed Binary Multitask Dense Predictor (Bi-MTPD) with current state-of-the-art models on the NYUD-v2 and PASCAL-Context datasets. This comparison includes notable CNN-based methods such as PAD-Net [49], ASTMT [27], MTI-Net [47], and ATRC [2], among others. Bi-MTPD demonstrates exceptional performance, outperforming other models in 6 of the 9 evaluated metrics, particularly in complex scene understanding tasks like Semantic Segmentation and Parsing. Remarkably, on the NYUD-v2 benchmark, Bi-MTPD surpasses the previously best-performing CNN-based method (ATRC) by a margin of +1.8 (mIoU) in Semantic Segmentation, while requiring

Table 2. Results on PASCAL-VOC. Bi-MTPD-C: only implementing the binarization in the multi-modal distillation module, Bi-MTPD-F: fully-binarized model.

Model		SemSeg $\uparrow$	Parsing $\uparrow$	Saliency $\uparrow$	Normal $\downarrow$	Bound $\uparrow$	float32 Params (M)	binary Params (M)	FLOPs (G)
Method	Backbone								
ASTMT [27]	CNN	68.00	61.10	65.70	14.70	72.40	364.72	0	501.27
PAD [49]		53.60	59.60	65.80	15.30	72.50	231.80	0	289.46
MTI [47]		61.70	60.18	84.78	14.23	70.80	218.56	0	280.12
ATRC [2]		62.69	59.42	84.70	14.20	70.96	241.45	0	310.19
ATRC-A [2]		63.60	62.23	83.91	14.30	70.86	249.87	0	320.57
ATRC-B[2]		67.67	62.93	82.29	14.24	72.42	280.01	0	383.21
MTI [47] + Bi-MTPD-F		48.10	56.28	64.42	14.29	76.89	13.67	204.89	31.38
MTI [47] + Bi-MTPD-C		62.98(1.28 $\uparrow$ )	60.44(0.26 $\uparrow$ )	83.56(1.22 $\downarrow$ )	14.31(0.08 $\uparrow$ )	71.28(0.26 $\uparrow$ )	153.71	64.85	194.51
InvPT[50]	ViT-B	77.50	66.83	83.65	14.63	73.00	176.35	0	274.68
InvPT[50] + Bi-MTPD-C		76.84 (0.66 $\downarrow$ )	67.10(0.27 $\uparrow$ )	84.97 (1.32 $\uparrow$ )	13.69(0.94 $\downarrow$ )	73.04 (0.04 $\uparrow$ )	154.68	21.67	220.76
InvPT[50]		79.03	67.61	84.81	14.15	73.00	422.93	0	425.37
InvPT[50] + Bi-MTPD-C	ViT-L	79.83 (0.80 $\uparrow$ )	68.17(0.56 $\uparrow$ )						

only 62% of the storage space for weights and 56% of computational FLOPs.

Furthermore, the application of Bi-MTPD to the ViT-based state-of-the-art method, InvPT [50], shows Bi-MTPD’s ability to enhance model performance while also improving efficiency. This demonstrates the broad applicability and generalization capability of Bi-MTPD across different architectural frameworks.

For a qualitative assessment, Figs. 7 and 6 displays prediction examples from various models. These examples illustrate that Bi-MTPD-C not only competes with but occasionally surpasses the state-of-the-art ATRC in qualitative performance.

### 4.3. Ablative Studies

Tab. 3 shows a series of ablative studies of Bi-MTPD-C and Bi-MTPD-F with an HRNet-48 backbone on NYUD-v2. We verify how different components of our model contribute to the multi-task improvements. In summary, every designed module positively impacts the overall model. We would like to highlight the main intuition of our method that binarized dense prediction models in a multitask manner largely outperform the binarized single models through Bi-MTPD-F w.o. VIB & KD vs. Bi-Single.

Table 3. Ablative studies on NYUD-v2.

Model	SemSeg $\uparrow$	Depth $\downarrow$	Normal $\downarrow$	Bound $\uparrow$
Bi-MTPD-C	<b>47.71</b>	<b>0.530</b>	<b>20.06</b>	<b>77.36</b>
Bi-MTPD-C w.o. binary MMD	45.70	0.537	20.27	77.86
Bi-MTPD-F	<b>39.20</b>	<b>0.612</b>	<b>21.04</b>	<b>76.86</b>
Bi-MTPD-F w.o. VIB	35.04	0.652	22.31	73.88
Bi-MTPD-F w.o. KD	36.20	0.640	21.99	74.86
Bi-MTPD-F w.o. VIB & KD	33.99	0.667	23.24	71.30
Bi-Single	16.20	0.872	29.36	69.24

## 5. Conclusion

In this paper, we significantly accelerate cumbersome dense prediction models, in which BNNs for relevant tasks are modeled and optimized under a multitask framework to supplement degraded information caused by binarization op-

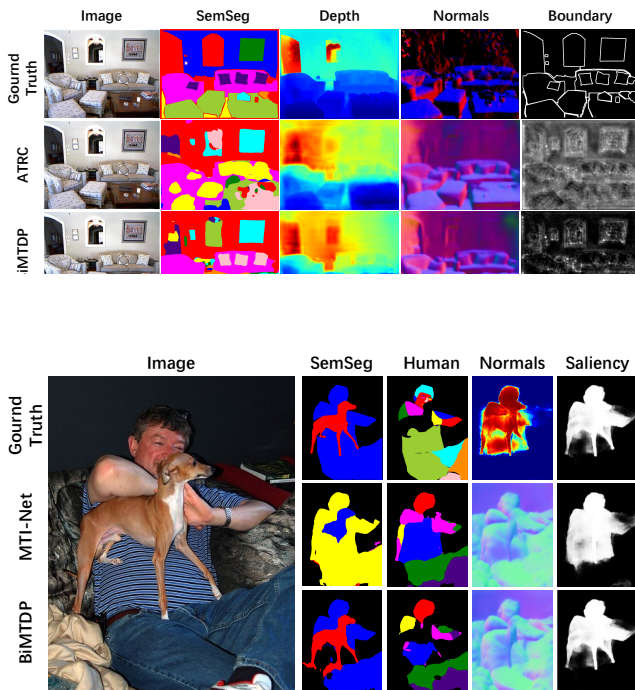


Figure 7. Qualitative comparison with MTI-Net on PASCAL-Context.

erations. Based on this binary baseline, we further introduce variational information bottleneck and feature-based knowledge distillation to supplement information flow. Experiment results show that our method significantly accelerates existing SoTA methods with comparably small performance drop over the mainstream dense prediction tasks on PASCAL VOC and NYUD-v2. Intriguingly, Bi-MTPD not only reaches SoTA w.r.t. performance but also saves computational costs, compared with SoTA method ATRC [2].

**Acknowledgments:** This research is partially supported by NSF IIS-2309073, ECCS-212352101 and Cisco unrestricted gift. This article solely reflects the opinions and conclusions of its authors and not the funding agencies.



## References

- [1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv:1308.3432*, 2013. [2](#), [3](#), [6](#)
- [2] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021. [1](#), [2](#), [3](#), [4](#), [7](#), [8](#)
- [3] Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. High-capacity expert binary networks. In *ICLR*, 2020. [5](#)
- [4] 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. [7](#)
- [5] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018. [2](#)
- [6] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *JMLR*, 2012. [6](#)
- [7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *NeurIPS*, 2016. [3](#)
- [8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. [7](#)
- [9] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, 2019. [2](#)
- [10] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In *NeurIPS*, 2007. [6](#)
- [11] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. [3](#)
- [12] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016. [1](#)
- [13] Xiangyu He, Zitao Mo, Ke Cheng, Weixiang Xu, Qinghao Hu, Peisong Wang, Qingshan Liu, and Jian Cheng. Proxybnn: Learning binarized neural networks via proxy matrices. In *CVPR*, 2020. [3](#)
- [14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS*, 2014. [1](#)
- [15] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *NeurIPS*, 2016. [1](#), [2](#), [3](#)
- [16] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *EMNLP*, 2020. [5](#)
- [17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. [1](#), [2](#)
- [18] Hyungjun Kim, Kyungsu Kim, Jinseok Kim, and Jae-Joon Kim. Binaryduo: Reducing gradient mismatch in binary activation network by coupling binary activations. In *ICLR*, 2020. [2](#)
- [19] Hyungjun Kim, Jihoon Park, Changhun Lee, and Jae-Joon Kim. Improving accuracy of binary neural networks using unbalanced activation distribution. In *CVPR*, 2021. [2](#)
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. [7](#)
- [21] 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. [2](#)
- [22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019. [6](#)
- [23] Zechun Liu, Wenhan Luo, Baoyuan Wu, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Binarizing deep network towards real-network performance. *IJCV*, 2020. [2](#), [3](#), [7](#)
- [24] Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization. In *ICML*, 2021. [7](#)
- [25] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *ECCV*, 2020. [2](#), [3](#)
- [26] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, 2017. [2](#)
- [27] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, 2019. [3](#), [7](#), [8](#)
- [28] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. [1](#), [2](#), [7](#)
- [29] Haotong Qin, Zhongang Cai, Mingyuan Zhang, Yifu Ding, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Hao Su. Bipointnet: Binary neural network for point clouds. In *ICLR*, 2021. [3](#)
- [30] Haotong Qin, Yifu Ding, Mingyuan Zhang, YAN Qinghua, Aishan Liu, Qingqing Dang, Ziwei Liu, and Xianglong Liu. Bibert: Accurate fully binarized bert. In *ICLR*, 2022. [2](#), [3](#), [5](#)
- [31] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe. Binary neural networks: A survey. *PR*, 2020. [3](#)
- [32] Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In *CVPR*, 2020. [2](#), [3](#)
- [33] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *NeurIPS*, 2021. [6](#)
- [34] Lothar Reichel and Qiang Ye. Simple square smoothing regularization operators. *Electronic Transactions on Numerical Analysis*, 33:63, 2009. [2](#)
- [35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. [6](#)
- [36] Ozan Sener and Vladlen Koltun. Multi-task learning as

- multi-objective optimization. In *NeurIPS*, 2018. 2
- [37] Yuzhang Shang, Bingxin Xu, Gaowen Liu, Ramana Rao Kompella, and Yan Yan. Causal-dfq: Causality guided data-free network quantization. In *ICCV*, 2023. 1
- [38] Yuzhang Shang, Dan Xu, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity retained binary neural network. In *ECCV*, 2022. 2
- [39] Yuzhang Shang, Dan Xu, Ziliang Zong, and Yan Yan. Network binarization via contrastive learning. In *ECCV*, 2022. 2
- [40] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, pages 1972–1981, 2023. 1
- [41] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 7
- [42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 2
- [43] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 5
- [44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 6
- [45] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *BMVC*, 2020. 2
- [46] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. Multi-task learning for dense prediction tasks: A survey. *TPAMI*, 2021. 1
- [47] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *CVPR*, 2020. 1, 2, 3, 4, 5, 7, 8
- [48] Ziwei Wang, Ziyi Wu, Jiwen Lu, and Jie Zhou. Bidet: An efficient binarized object detector. In *CVPR*, 2020. 5
- [49] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018. 1, 2, 3, 4, 7, 8
- [50] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. *arXiv preprint arXiv:2203.07997*, 2022. 1, 2, 4, 7, 8
- [51] Shang Yuzhang, Duan Bin, Zong Ziliang, Nie Liqiang, and Yan Yan. Lipschitz continuity guided knowledge distillation. In *ICCV*, 2021. 5
- [52] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2
- [53] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, 2019. 2, 7
- [54] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaogun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *CVPR*, 2020. 7
- [55] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016. 2