IDa-Det: An Information Discrepancy-aware Distillation for 1-bit Detectors

Sheng Xu[1†], Yanjing Li[1†], Bohan Zeng[1†], Teli Ma[2], Baochang Zhang[1,3*], Xianbin Cao[1], Peng Gao[2], Jinhu Lü[1,3]

[1] Beihang University, Beijing, China
[2] Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3] Zhongguancun Laboratory, Beijing, China
{shengxu, yanjingli, bohanzeng, bczhang}@buaa.edu.cn

Abstract. Knowledge distillation (KD) has been proven to be useful for training compact object detection models. However, we observe that KD is often effective when the teacher model and student counterpart share similar proposal information. This explains why existing KD methods are less effective for 1-bit detectors, caused by a significant information discrepancy between the real-valued teacher and the 1-bit student. This paper presents an Information Discrepancy-aware strategy (IDa-Det) to distill 1-bit detectors that can effectively eliminate information discrepancies and significantly reduce the performance gap between a 1-bit detector and its real-valued counterpart. We formulate the distillation process as a bi-level optimization formulation. At the inner level, we select the representative proposals with maximum information discrepancy. We then introduce a novel entropy distillation loss to reduce the disparity based on the selected proposals. Extensive experiments demonstrate IDa-Det's superiority over state-of-the-art 1-bit detectors and KD methods on both PASCAL VOC and COCO datasets. IDa-Det achieves a 76.9% mAP for a 1-bit Faster-RCNN with ResNet-18 backbone. Our code is open-sourced on https://github.com/SteveTsui/IDa-Det.

Keywords: 1-bit detector, Knowledge distillation, Information discrepancy

1 Introduction

Recently, the object detection task [6,20] has been greatly promoted due to advances in deep convolutional neural networks (DNNs) [12,8]. However, DNN models comprise a large number of parameters and floating-point operations (FLOPs), restricting their deployment on embedded platforms. Techniques such as compact network design [15,24], network pruning [13,16,38], low-rank decomposition [5], and quantization [26,33,36] have been developed to address these restrictions and accomplish an efficient inference on the detection task. Among

† Equal contribution.
∗ Corresponding author.

these, binarized detectors have contributed to object detection by accelerating the CNN feature extracting for real-time bounding box localization and foreground classification [34,31,35]. For example, the 1-bit SSD300 [21] with VGG-16 backbone [28] theoretically achieve the acceleration rate up to 15× with XNOR and Bit-count operations using binarized weights and activations as described in [31]. With extremely high energy-efficiency for embedded devices, they are able to be installed directly on next-generation AI chips. Despite these appealing features, 1-bit detectors' performance often deteriorates to the point, which explains why they are not widely used in real-world embedded systems.

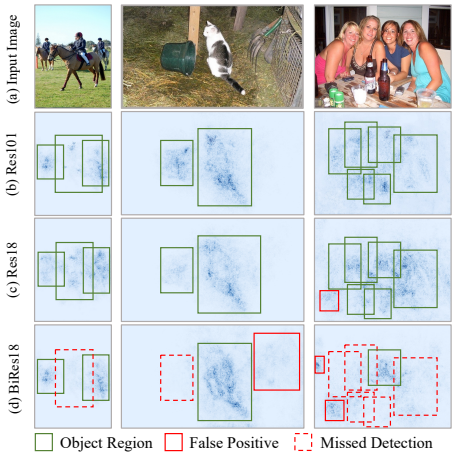
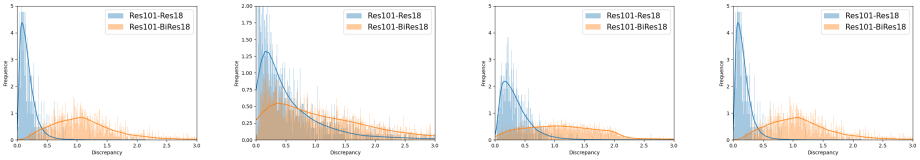


Fig. 1. Input images and saliency maps following [10]. Images are randomly selected from VOC `test2007`. Each row includes: (a) input images, saliency maps of (b) Faster-RCNN with ResNet-101 backbone (Res101), (c) Faster-RCNN with ResNet-18 backbone (Res18), (d) 1-bit Faster-RCNN with ResNet-18 backbone (BiRes18), respectively.

The recent art [35] employs fine-grained feature imitation (FGFI) [30] to enhance the performance of 1-bit detectors. However, it neglects the intrinsic information discrepancy between 1-bit detectors and real-valued detectors. As shown in Fig. 1, we demonstrate that saliency maps of real-valued Faster-RCNN of the ResNet-101 backbone (often used as the teacher network) and the ResNet-18 backbone, in comparison with 1-bit Faster-RCNN of the ResNet-18 backbone (often used as the student network) from top to bottom. They show that knowledge distillation (KD) methods like [30] are effective for distilling real-valued Faster-RCNNs, only when their teacher model and their student counterpart share small information discrepancy on proposals, as shown in Fig. 1 (b) and (c). This phenomenon does not happen for 1-bit Faster-RCNN, as shown in Fig. 1 (b) and (d). This might explain why existing KD methods are less effective in 1-bit detectors. A statistic on COCO and PASCAL VOC datasets in Fig. 2 show that the discrepancy between proposal saliency maps of Res101 and Res18 (blue) is much smaller than that of Res101 and BiRes18 (orange). That is to say, the smaller the distance is, the less the discrepancy is. Briefly, conventional KD methods show their effectiveness on distilling real-valued detectors but seem to be less effective on distilling 1-bit detectors.

In this paper, we are motivated by the above observation and present an information discrepancy-aware distillation for 1-bit detectors (IDa-Det), which can effectively address the information discrepancy problem, leading to an efficient distillation process. As shown in Fig. 3, we introduce a discrepancy-aware method to select proposal pairs and facilitate distilling 1-bit detectors, rather



(a) VOC `trainval0712` (b) VOC `test2007` (c) COCO `trainval35k` (d) COCO `minival`

Fig. 2. The Mahalanobis distance of the gradient in the intermediate neck feature between Res101-Res18 (blue) and Res101-BiRes18 (orange) in various datasets.

than only using object anchor locations of student models or ground truth as in existing methods [30,35,10]. We further introduce a novel entropy distillation loss to leverage more comprehensive information than the conventional loss functions. By doing so, we achieve a powerful information discrepancy-aware distillation method for 1-bit detectors (IDa-Det). Our contributions are summarized as:

- Unlike existing KD methods, we distill 1-bit detectors by fully considering the information discrepancy into optimization, which is simple yet effective for learning 1-bit detectors.
- We propose an entropy distillation loss further to improve the representation ability of the 1-bit detector and effectively eliminate the information discrepancy.
- We compare our IDa-Det against state-of-the-art 1-bit detectors and KD methods on the VOC and large-scale COCO datasets. Extensive results reveal that our method outperformas the others by a considerable margin. For instance, on VOC `test2007`, the 1-bit Faster-RCNN with ResNet-18 backbone achieved by IDa-Det obtains 76.9% mAP, achieving a new state-of-the-art.

2 Related Work

1-bit Detectors. By removing the foreground redundancy, BiDet [31] fully exploits the representational capability of the binarized convolutions. In this way, the information bottleneck is introduced, which limits the amount of data in high-level feature maps, while maximizing the mutual information between feature maps and object detection. The performance of the Faster R-CNN detector is significantly enhanced by the ASDA-FRCNN [34] which suppresses the shared amplitude between the real-value and binary kernels. LWS-Det [35] novelly proposes a layer-wise searching approach, minimizing the angular and amplitude errors for 1-bit detectors. Also, FGFI [30] is used by LWS-Det to distill the backbone feature map further.

Knowledge Distillation. Knowledge distillation (KD), a significant subset of model compression methods, aims to transfer knowledge from a well-trained teacher network to a more compact student model. The student is supervised

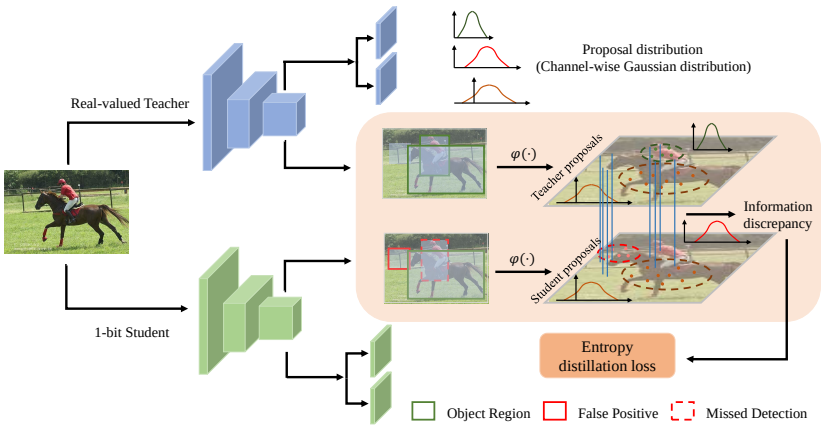


Fig. 3. Overview of the proposed information discrepancy-aware distillation (IDa-Det) framework. We first select representative proposal pairs based on the information discrepancy. Then we propose the entropy distillation loss to eliminate the information discrepancy.

using soft labels created by the teacher, as firstly proposed by [1]. Knowledge distillation is redefined by [14] as training a shallower network after the softmax layer to approximate the teacher's output. Object detectors can be compressed using knowledge distillation, according to numerous recent papers. Chen *et al.* [2] distill the student through all backbone features, regression head, and classification head, but both the imitation of whole feature maps and the distillation in classification head fail to add attention to the important foreground, potentially resulting in a sub-optimal result. Mimicking [17] distills the features from sampled region proposals. However, just replicating the aforementioned regions may lead to misdirection, because the proposals occasionally perform poorly. In order to distill the student, FGFI [30] introduces a unique attention mask to create fine-grained features from foreground object areas. DeFeat [10] balances the background and foreground object regions to efficiently distill the student.

In summary, existing KD frameworks for object detection can only be employed for real-valued students having similar information as their teachers. Thus, they are often ineffective in distilling 1-bit detectors. Unlike prior arts, we identify that the information discrepancy between real-valued teacher and 1-bit students are significant for distillation. We first introduce Mahalanobis distance to identify the information discrepancy and then accordingly distill the features. Meanwhile, we propose a novel entropy distillation loss to prompt the discrimination ability of 1-bit detectors further.

3 The Proposed Method

In this section, we describe our IDa-Det in detail. Firstly, we overview the 1-bit CNNs. We then describe how we employ the information discrepancy method

(IDa) to select representative proposals. Finally, we describe the entropy distillation loss to delicately eliminate the information discrepancy between the real-valued teachers and the 1-bit students.

3.1 Preliminaries

In a specific convolution layer, $\mathbf{w} \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$, $\mathbf{a}_{in} \in \mathbb{R}^{C_{in} \times W_{in} \times H_{in}}$, and $\mathbf{a}_{out} \in \mathbb{R}^{C_{out} \times W_{out} \times H_{out}}$ represent its weights and feature maps, where C_{in} and C_{out} represents the number of channels. (H, W) are the height and width of the feature maps, and K denotes the kernel size. We then have

$$\mathbf{a}_{out} = \mathbf{a}_{in} \otimes \mathbf{w}, \tag{1}$$

where $\otimes$ is the convolution operation. We omit the batch normalization (BN) and activation layers for simplicity. The 1-bit model aims to quantize $\mathbf{w}$ and $\mathbf{a}_{in}$ into $\mathbf{b}^{\mathbf{w}} \in \{-1, +1\}^{C_{out} \times C_{in} \times K \times K}$ and $\mathbf{b}^{\mathbf{a}_{in}} \in \{-1, +1\}^{C_{in} \times H \times W}$ using the efficient XNOR and Bit-count operations to replace full-precision operations. Following [3,4], the forward process of the 1-bit CNN is

$$\mathbf{a}_{out} = \boldsymbol{\alpha} \circ \mathbf{b}^{\mathbf{a}_{in}} \odot \mathbf{b}^{\mathbf{w}}, \tag{2}$$

where $\odot$ is the XNOR, and bit-count operations and $\circ$ denotes the channel-wise multiplication. $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_{C_{out}}] \in \mathbb{R}_+$ is the vector consisting of channel-wise scale factors. $\mathbf{b} = \text{sign}(\cdot)$ denotes the binarized variable using the sign function, which returns 1 if the input is greater than zero, and -1 otherwise. It then enters several non-linear layers, *e.g.*, BN layer, non-linear activation layer, and max-pooling layer. We omit these for simplification. And then, the output $\mathbf{a}_{out}$ is binarized to $\mathbf{b}^{\mathbf{a}_{out}}$ via the sign function. The fundamental objective of BNNs is calculating $\mathbf{w}$. We want it to be as close as possible before and after binarization, such that the binarization effect is minimized. Then, we define the reconstruction error as

$$L_R(\mathbf{w}, \boldsymbol{\alpha}) = \mathbf{w} - \boldsymbol{\alpha} \circ \mathbf{b}^{\mathbf{w}}. \tag{3}$$

3.2 Select proposals with Information Discrepancy

To eliminate the large magnitude scale difference between the real-valued teacher and the 1-bit student, we introduce a channel-wise transformation for the proposals[1] of the intermediate neck. We first apply a transformation $\varphi(\cdot)$ on a proposal $\tilde{R}_n \in \mathbb{R}^{C \times W \times H}$ and have

$$R_{n;c}(x, y) = \varphi(\tilde{R}_{n;c}(x, y)) = \frac{\exp(\frac{\tilde{R}_{n;c}(x,y)}{\mathcal{T}})}{\sum_{(x',y') \in (W,H)} \exp(\frac{\tilde{R}_{n;c}(x'y')}{\mathcal{T}})}, \tag{4}$$

[1] In this paper, the proposal denotes the neck/backbone feature map patched by the region proposal of detectors.

where $(x, y) \in (W, H)$ denote a specific spatial location (x, y) in spatial range (W, H), and $c \in \{1, \cdots, C\}$ is the channel index. $n \in \{1, \cdots, N\}$ is the proposal index. N denotes the number of proposals. $\mathcal{T}$ denotes a hyper-parameter controlling the statistical attributions of the channel-wise alignment operation[2]. After the transformation, features in each channel of a proposal projected into the same feature space [29] and follow a Gaussian distribution as

$$p(R_{n;c}) \sim \mathcal{N}(\mu_{n;c}, \sigma^2_{n;c}). \tag{5}$$

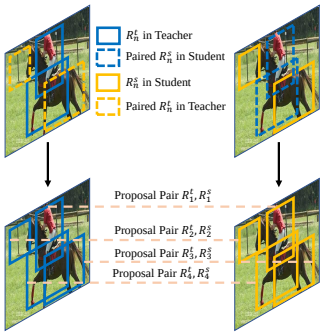


Fig. 4. Illustration for the generation of the proposal pairs. Each single proposal in one model generates a counterpart feature map patch in the same location of the other model.

We further evaluate the information discrepancy between the proposals of the teacher and the student. As shown in Fig. 4, the teacher and student have N_T and N_S proposals, respectively. Each single proposal in one model generates a counterpart feature map patch in the same location of the other model, thus total $N_T + N_S$ proposal pairs are considered. To evaluate the information discrepancy, we introduce the Mahalanobis distance of each channel-wise proposal feature and measure the discrepancy as

$$\varepsilon_n = \sum_{c=1}^{C} ||(R^t_{n;c} - R^s_{n;c})^T \Sigma^{-1}_{n;c} (R^t_{n;c} - R^s_{n;c})||_2, \tag{6}$$

where $\Sigma_{n;c}$ denotes the covariance matrix of the teacher and student in the c-th channel of the n-th proposal pair. The Mahalanobis distance takes both the pixel-level distance between proposals and the statistical characteristics differences in proposal pairs into account.

To select representative proposals with maximum information discrepancy, we first define a binary distillation mask m_n as

$$m_n = \begin{cases} 1, & \text{if pair } (R^t_n, R^s_n) \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $m_n = 1$ denotes distillation will be applied on such proposal pair, otherwise remain unchanged. For each proposal pair, only when their distribution is quite different, the student model can learn from the teacher counterpart, where a distillation process is needed.

Based on the derivation above, discrepant proposal pairs will be optimized through distillation. For distilling the selected pairs, we resort to maximize conditional probability $p(R^s_n | R^t_n)$. That is to say, after distillation or optimization, feature distributions of teacher proposal and student counterpart become similar

[2] In this paper, we set $\mathcal{T} = 4$.

with each other. To this end, we define $p(R_n^s|R_n^t)$ with $m_n, n \in \{1, \cdots, N_T+N_S\}$ in consideration as

$$p(R_n^s|R_n^t; m_n) \sim m_n\mathcal{N}(\mu_n^t, {\sigma_n^t}^2) + (1-m_n)\mathcal{N}(\mu_n^s, {\sigma_n^s}^2). \tag{8}$$

Subsequently, we introduce a bi-level optimization formulation to solve the distillation problem as

$$\begin{aligned} &\max_{R_n^s} p(R_n^s|R_n^t; \mathbf{m}^*), \ \ \forall\, n \in \{0, \cdots, N_T+N_S\}, \\ &\text{s.t. } \ \mathbf{m}^* = \arg\max_{\mathbf{m}} \sum_{n=1}^{N_T+N_S} m_n\varepsilon_n, \end{aligned} \tag{9}$$

where $\mathbf{m} = [m_1, \cdots, m_{N_T+N_S}]$ and $||\mathbf{m}||_0 = \gamma \cdot (N_T + N_S)$. γ is a hyper-parameter. In this way, we select $\gamma \cdot (N_T + N_S)$ pairs of proposals containing the most representative information discrepancy for distillation. γ controls the proportion of discrepant proposal pairs, further validated in Sec. 4.2.

For each iteration, we first solve the inner-level optimization, *i.e.*, proposal selection, via exhaustive sorting [32]; and then solve the upper-level optimization, distilling the selected pair, based on entropy distillation loss discussed in Sec. 3.3. Considering that there are not too many proposals involved, the process is relatively efficient for the inner-level optimization.

3.3 Entropy distillation loss

After selecting a specific number of proposals, we crop the feature based on the proposals we obtained. Most of the SOTA detection models are based on Feature Pyramid Networks (FPN) [19], which can significantly improve the robustness of multi-scale detection. For the Faster-RCNN framework in this paper, we resize the proposals and crop the features from each stage of the neck feature maps. We generate the proposals from the regression layer of the SSD framework and crop the features from the feature map of maximum spatial size. Then we formulate the entropy distillation process as

$$\max_{R_n^s} \ p(R_n^s|R_n^t). \tag{10}$$

Here is **the upper level** of the bi-level optimization, where $\boldsymbol{m}$ is solved and therefore omitted. We rewrite Equ. 10 and further achieve our entropy distillation loss as

$$L_P(\mathbf{w}, \boldsymbol{\alpha}; \gamma) = (R_n^s - R_n^t) + \mathrm{Cov}(R_n^s, R_n^t)^{-1}(R_n^s - R_n^t)^2 + \log(\mathrm{Cov}(R_n^s, R_n^t)), \tag{11}$$

where $\mathrm{Cov}(R_n^s, R_n^t) = \mathbb{E}(R_n^s R_n^t) - \mathbb{E}(R_n^s)\mathbb{E}(R_n^t)$ denotes the covariance matrix.

Hence, we trained the 1-bit student model end-to-end, the total loss for distilling the student model is defined as

$$L = L_{GT}(\mathbf{w}, \boldsymbol{\alpha}) + \lambda L_P(\mathbf{w}, \boldsymbol{\alpha}; \gamma) + \mu L_R(\mathbf{w}, \boldsymbol{\alpha}), \tag{12}$$

where L_{GT} is the detection loss derived from the ground truth label and L_R is defined in Equ. 3.

4 Experiments

On two mainstream object detection datasets, *i.e,* PASCAL VOC [6] and COCO [20], extensive experiments are undertaken to test our proposed method. First, we go through the implementation specifics of our IDa-Det. Then, in the ablation studies, we set different hyper-parameters and validate the effectiveness of the components as well as the convergence of our method. Moreover, we illustrate the superiority of our IDa-Det by comparing it to previous state-of-the-art 1-bit CNNs and other KD approaches on 1-bit detectors. Finally, we analyze the deploy efficiency of our IDa-Det on hardware.

4.1 Datasets and Implementation Details

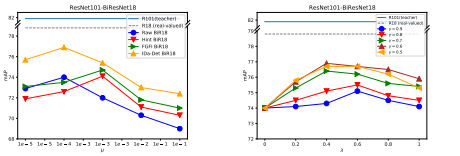
PASCAL VOC. Natural images from 20 different classes are included in the VOC datasets. We use the VOC `trainval2012` and VOC `trainval2007` sets to train our model, which contain around 16k images, and the VOC `test2007` set to evaluate our IDa-Det, which contains 4952 images. We utilize the mean average precision (mAP) as the evaluation matrices, as suggested by [6].
COCO. All our experiments on COCO dataset are conducted on the COCO `2014` [20] object detection track in the training stage, which contains the combination of 80k images and 80 different categories from the COCO `train2014` and 35k images sampled from COCO `val2014`, *i.e.*, COCO `trainval35k`. Then we evaluate our method on the remaining 5k images from the COCO `minival`. We list the average precision (AP) for IoUs$\in$ $[0.5 : 0.05 : 0.95]$, designated as mAP@$[.5, .95]$, using COCO's standard evaluation metric. For further analyzing our method, we also list AP_{50}, AP_{75}, AP_s, AP_m, and AP_l.
Implementation Details. Our IDa-Det is trained with two mainstream object detectors, *i.e.*, two-stage Faster-RCNN[3] [27] and one-stage SSD [21]. In Faster-RCNN, we utilize ResNet-18 and ResNet-34 [12] as backbones. And we utilize VGG-16 [28] as the backbone of SSD framework. PyTorch [25] is used for implementing IDa-Det. We run the experiments on 4 NVIDIA GTX 2080Ti GPUs with 11 GB memory and 128 GB of RAM. We use ImageNet ILSVRC12 to pre-train the backbone of a 1-bit student, following [23]. The SGD optimizer is utilized and the batch size is set as 24 for SSD and 4 for Faster-RCNN, respectively.

We keep the shortcut, first layer, and the last layer (the 1×1 convolution layer of RPN and a FC layer of the bbox head) in the detectors as real-valued on Faster-RCNN framework after implementing 1-bit CNNs following [23]. Following BiDet [31], the extra layer is likewise retained as real-valued for the SSD framework. Following [31] and [9], we modify the network of ResNet-18/34 with an extra shortcut and PReLU [11].

The architecture of VGG-16 is modified with extra residual connections, following [31]. The lateral connection of the FPN [19] neck is replaced with 3×3 1-bit convolution for improving performance. This adjustment is implemented in

[3] In this paper, Faster-RCNN denotes the Faster-RCNN implemented with FPN neck.

all of the Faster-RCNN experiments. For Faster-RCNN, we train the model with two stages. Only the backbone is binarized at the first stage. Then we binarize all layers in the second stage. Each stage counts 12 epochs. The learning rate is set as 0.004 and decays by multiplying 0.1 in the 9-th and 11-th epochs. We use a loss coefficient set as 5 and multi-scale training method. For fair comparison, all the methods in this paper are implemented with the same training setup. The real-valued counterparts in this paper are also trained for 24 epochs for fair comparison. For SSD, the model is trained for 24 epochs with a learning rate of 0.01, which decays to 0.1 at the 16-th and 22-nd epochs by multiplying 0.1.



(a) Effect of μ. (b) Effect of λ and γ.

Fig. 5. On VOC, we (a) select μ on raw detector and different KD methods including Hint [2], FGFI [30], and IDa-Det; (b) select λ and γ on IDa-Det with μ set as $1e$–4.

We select real-valued Faster-RCNN with ResNet101 backbone (81.9% mAP on VOC and 39.8% mAP on COCO) and real-valued SSD300 with VGG16 backbone (74.5% mAP on VOC and 25.0% mAP on COCO) as teacher network.

4.2 Ablation Study

Selecting the hyper-parameter. As mentioned above, we select hyper-parame ters λ, γ, and μ in this part. We first select the μ, which controls the binarization process. As plotted in Fig. 5 (a), we first fine-tune the hyper-parameter μ controlling the binarization process in four situations: raw BiRes18, and BiRes18 distilled via Hint [2], FGFI [30], and our IDa-Det, respectively. Overall, the performances increase first and then decrease when increasing the value of μ. On raw BiRes18 and IDa-Det BiRes18, the 1-bit student achieves the best performance when μ is set as $1e$-4. And μ valued $1e$-3 is better for the Hint, and the FGFI distilled 1-bit student. Thus, we set μ as $1e$-4 for an extended ablation study. Fig. 5 (b) shows that the performances increase first and then decrease with the increase of λ from left to right. In general, the IDa-Det obtains better performances with λ set as 0.4 and 0.6. With varying value of γ, we find $\{\lambda, \gamma\}$ = $\{0.4, 0.6\}$ boost the performance of IDa-Det most, achieving 76.9% mAP on VOC `test2007`. Based on the ablative study above, we set hyper-parameters λ, γ, and μ as 0.4, 0.6, and $1e$-4 for the experiments in this paper.

Effectiveness of components. We first compare our information discrepancy-aware (IDa) proposal selecting method with other methods to select proposals: Hint [2] (using the neck feature without region mask) and FGFI [30]. We show the effectiveness of IDa on two-stage Faster-RCNN in Tab. 1. On the Faster-RCNN, the introducing of IDa achieves improvements of the mAP by 2.5%, 2.4%, and 1.8% compared to non-distillation, Hint, and FGFI, under the same student-teacher framework. Then we evaluate the proposed entropy distillation loss against the conventional ℓ_2 loss, inner-product loss, and cosine similarity

Model	Proposal selection	Distillation method	mAP
Res18	✗	✗	78.6
BiRes18	✗	✗	74.0
Res101-BiRes18	Hint	ℓ_2	74.1
Res101-BiRes18	Hint	Entropy loss	74.5
Res101-BiRes18	FGFI	ℓ_2	74.7
Res101-BiRes18	FGFI	Entropy loss	75.0
Res101-BiRes18	IDa	Inner-product	74.8
Res101-BiRes18	IDa	Consine similarity	76.4
Res101-BiRes18	IDa	ℓ_2	76.5
Res101-BiRes18	**IDa**	**Entropy loss**	**76.9**

Table 1. The effects of different components in IDa-Det with Faster-RCNN model on PASCAL VOC dataset. Hint [2] and FGFI[30] are used to compare with our information discrepancy-aware proposal selection (IDa). IDa and Entropy loss denote main components of the proposed IDa-Det.

loss. As depicted in Tab. 1, our entropy distillation loss improves the distillation performance by 0.4%, 0.3%, and 0.4% with Hint, FGFI, and IDa method compared with ℓ_2 loss. Compared with inner-product and cosine similarity loss, entropy loss outperforms them by 2.1% and 0.5% in mAP on our framework, which further reflects the effectiveness of our method.

4.3 Results on PASCAL VOC

With the same student framework, we compare our IDa-Det with the state-of-the-art 1-bit ReActNet [23] and other KD methods, such as FGFI [30], DeFeat [10], and LWS-Det [35], in the task of object detection with the VOC datasets. The detection results of the multi-bit quantization method DoReFa-Net [37] is also reported. We use the input resolution following [35], *i.e.* 1000×600 for Faster-RCNN and 300×300 for SSD.

Tab. 2 lists the comparison of several quantization approaches and detection frameworks in terms of computing complexity, storage cost, and the mAP. Our IDa-Det significantly accelerates computation and reduces storage requirements for various detectors. We follow XNOR-Net [26] to calculate memory usage, which is estimated by adding $32\times$ the number of full-precision kernels and $1\times$ the number of binary kernels in the networks. The number of float operations (FLOPs) is calculated in the same way as Bi-Real-Net [22]. The current CPUs can handle both bit-wise XNOR and bit-count operations in parallel. The number of real-valued FLOPs plus $\frac{1}{64}$ of the number of 1-bit multiplications equals the OPs following [22].

Faster-RCNN. We summarize the experimental results on VOC `test2007` of 1-bit Faster-RCNNs from lines 2 to 17 in Tab. 2. Compared with raw ReActNet [23], our baseline binarization method achieves 4.4%, and 2.7% mAP improvement with ResNet-18/34 backbone respectively. Compared with other KD

Framework	Backbone	Quantization Method	KD Method	W/A	Memory Usage (MB)	OPs ($\times 10^9$)	mAP
Faster-RCNN	ResNet-18	Real-valued	✗	32/32	112.88	96.40	78.8
		DoReFa-Net	✗	4/4	21.59	27.15	73.3
		ReActNet	✗	1/1	16.61	18.49	69.6
		Ours	✗				74.0
		LWS-Det					73.2
		Ours	FGFI				74.7
		Ours	DeFeat				74.9
		IDa-Det					**76.9**
	ResNet-34	Real-valued	✗	32/32	145.12	118.80	80.0
		DoReFa-Net	✗	4/4	29.65	32.31	75.6
		ReActNet	✗	1/1	24.68	21.49	72.3
		Ours	✗				75.0
		Ours	FGFI				75.4
		Ours	DeFeat				75.7
		LWS-Det					75.8
		IDa-Det					**78.0**
SSD	VGG-16	Real-valued	✗	32/32	105.16	31.44	74.3
		DoReFa-Net	✗	4/4	29.58	6.67	69.2
		ReActNet	✗	1/1	21.88	2.13	68.4
		Ours	✗				69.5
		LWS-Det					69.5
		Ours	FGFI				70.0
		Ours	DeFeat				71.4
		IDa-Det					**72.5**

Table 2. We report memory usage, FLOPs, and mAP (%) with state-of-the-art 1-bit detectors, other KD methods on VOC `test2007`. The best results are **bold**.

methods on the 1-bit detector with the same train and test settings, our IDa-Det surpasses FGFI and DeFeat distillation method in a clear margin of 2.2%, 2.0% with ResNet-18 backbone, and 2.6%, 2.3% with ResNet-34 backbone. Our IDa-Det significantly surpasses the prior state-of-the-art, LWS-Det, by 3.7% in ResNet-18 backbone, and 2.2% in ResNet-34 backbone with the same FLOPs and memory usage. All of the improvements have impacts on object detection.

Compared with the raw real-valued detectors, the proposed IDa-Det surpasses real-valued Faster-RCNN with ResNet-18/34 backbone ({76.9%, 78.0%} *vs.*{76.4%, 77.8%}) by obviously computation acceleration and storage savings by 5.21×/5.53× and 6.80×/5.88×. The above results are of great significance in the real-time inference of object detection.

SSD. On the SSD300 framework with a VGG-16 backbone, our IDa-Det can accelerate computation and save storage by 14.76× and 4.81× faster than real-valued counterparts, respectively, as illustrated in the bottom section of Tab. 2. The drop in the performance is relatively minor (72.5% *vs.* 74.3%). Also, our

(a) False positives (b) Missed detection

Fig. 6. Qualitative results on the gain from information discrepancy-aware distilling. The top row shows IDa-Det student's output. The bottom row images are raw student model's output without information discrepancy-aware distilling.

method surpasses other 1-bit networks and KD methods by a sizable margin. Compared to 1-bit ReActNet, our raw 1-bit model can achieve 1.1% higher mAP with the same computation. Compared with FGFI, DeFeat, and LWS-Det, our IDa-Det exceeds them by 3.0%, 2.5%, and 1.1%, respectively.

As shown in Fig. 6, BiRes18 achieved by IDa-Det effectively eliminates the false positives and missed detections compared with raw BiRes18. In summary, we achieve new state-of-the-art performance on PASCAL VOC compared to previous 1-bit detectors and KD methods on various frameworks and backbones. We also achieve competitive results, demonstrating the IDa-Det's superiority.

4.4 Results on COCO

Because of its diversity and scale, the COCO dataset presents a greater challenge in the object detection task, compared with PASCAL VOC. On COCO, we compare our proposed IDa-Det with the state-of-the-art 1-bit ReActNet [23], as well as the KD techniques FGFI [30], DeFeat [10], and LWS-Det [35]. We also report the detection result of the 4-bit quantization method FQN [18] and the DoReFa-Net [37] for reference. Following [35], the input images are scaled to 1333×800 for Faster-RCNN and 300×300 for SSD.

The mAP, AP with different IoU thresholds, and AP of objects with varying scales are all reported in Tab. 3. Due to the constraints in the width of page, we do not report the FLOPs and memory use in Tab. 3. We conduct experiments on Faster-RCNN and SSD detectors, the results of which are presented in the folllowing two parts.

Faster-RCNN. Comparing to the state-of-the-art 1-bit ReActNet, our baseline binarized model achieves a 5.7% improvement on mAP@[.5, .95] with the ResNet-18 backbone. Compared to state-of-the-art LWS-Det, FGFI, and DeFeat, our IDa-Det prompts the mAP@[.5, .95] by 2.4%, 1.8%, and 1.4%, respectively. With the ResNet-34 backbone, the proposed IDa-Det surpasses FGFI, DeFeat, and LWS-Det by 1.1%, 0.7%, and 0.6%, respectively. IDa-Det, nevertheless, has substantially reduced FLOPs and memory use.

Framework	Backbone	Quantization Method	KD Method	W/A	mAP @[.5, .95]	AP_{50}	AP_{75}	AP_s	AP_m	AP_1
Faster-RCNN	ResNet-18	Real-valued	✗	32/32	32.2	53.8	34.0	18.0	34.7	41.9
		FQN	✗	4/4	28.1	48.4	29.3	14.5	30.4	38.1
		ReActNet	✗	1/1	21.1	38.5	20.5	9.7	23.5	32.1
		Ours	✗		26.8	46.1	27.9	14.7	28.4	36.0
		LWS-Det			26.9	44.9	27.7	12.9	28.7	38.3
		Ours	FGFI		27.5	46.5	28.8	15.2	28.7	37.5
		Ours	DeFeat		27.9	46.9	29.3	15.8	29.0	38.2
		IDa-Det			**29.3**	**48.7**	**30.9**	**16.7**	**20.8**	**39.9**
	ResNet-34	Real-valued	✗	32/32	35.8	57.6	38.4	21.1	39.0	46.1
		FQN	✗	4/4	31.8	52.9	33.9	17.6	34.4	42.2
		ReActNet	✗	1/1	23.4	43.3	24.4	10.7	25.9	35.5
		Ours	✗		29.0	47.7	30.9	16.6	30.5	39.0
		Ours	FGFI		29.4	48.4	30.3	17.1	30.7	39.7
		Ours	DeFeat		29.8	48.7	30.9	17.6	31.4	40.5
		LWS-Det			29.9	49.2	30.1	15.1	**32.1**	**40.9**
		IDa-Det			**30.5**	**49.2**	**31.8**	**17.7**	31.3	40.6
SSD	VGG-16	Real-valued	✗	32/32	23.2	41.2	23.4	5.3	23.2	39.6
		DoReFa-Net	✗	4/4	19.5	35.0	19.6	5.1	20.5	32.8
		ReActNet	✗	1/1	15.3	30.0	13.2	5.4	16.3	25.0
		Ours	✗		17.2	31.5	16.8	3.2	18.2	31.3
		LWS-Det			17.1	32.9	16.1	**5.5**	17.4	26.7
		Ours	FGFI		17.7	32.3	17.3	3.3	18.9	31.8
		Ours	DeFeat		18.1	32.8	17.9	3.3	19.4	32.6
		IDa-Det			**19.4**	**34.5**	**19.3**	3.7	**21.1**	**35.0**

Table 3. Comparison with state-of-the-art 1-bit detectors and KD methods on COCO `minival`. Optimal results are **bold**.

SSD. On the SSD300 framework, our IDa-Det achieves 19.4% mAP@[.5, .95] with the VGG-16 backbone, surpassing LWS-Det, FGFI, and DeFeat by 2.3%, 1.7%, and 1.3% mAP, respectively.

To summarize, our method outperforms baseline quantization methods and other KD methods in terms of the AP with various IoU thresholds and the AP for objects of varied sizes on COCO, indicating IDa-Det's superiority and generality in many application settings.

4.5 Deployment Efficiency

We implement the 1-bit models achieved by our IDa-Det on ODROID C4, which has a 2.016 GHz 64-bit quad-core ARM Cortex-A55. With evaluating its real speed in practice, the efficiency of our IDa-Det is proved when deployed into real-world mobile devices. We use the SIMD instruction SSHL on ARM NEON, for making inference framework BOLT [7] compatible with our IDa-Det. We compare our IDa-Det to the real-valued backbones in Tab. 4. We utilize VOC

Framework	Network	Method	W/A	Latency (ms)	Acceleration
Faster-RCNN	ResNet-18	Real-valued	32/32	12043.8	-
		IDa-Det	1/1	2474.4	4.87×
	ResNet-34	Real-valued	32/32	14550.2	-
		IDa-Det	1/1	2971.3	4.72×
SSD	VGG-16	Real-valued	32/32	2788.7	-
		IDa-Det	1/1	200.5	13.91×

Table 4. Comparison of time cost of real-valued and 1-bit models (Faster-RCNN and SSD) on hardware (single thread).

dataset for testing the model. For Faster-RCNN, the input images were scaled to 1000×600 and 300×300 for SSD. We can plainly see that IDa-Det's inference speed is substantially faster with the highly efficient BOLT framework. For example, the acceleration rate achieves about $4.7\times$ on Faster-RCNN, which is slightly lower than the theoretical acceleration rate discussed in Sec. 4.3. Furthermore, IDa-Det achieves $13.91\times$ acceleration with SSD. All deployment results in the object detection are significant on real-world edge devices.

5 Conclusion

This paper presents a novel method for training 1-bit detectors with knowledge distillation to minimize the information discrepancy. IDa-Det employs a maximum entropy model to select the proposals with maximum information discrepancy and proposes a novel entropy distillation loss to supervise the information discrepancy. As a result, our IDa-Det significantly boosts the performance of 1-bit detectors. Extensive experiments show that IDa-Det surpasses state-of-the-art 1-bit detectors and other knowledge distillation methods in object detection.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 62076016, 92067204, 62141604 and the Shanghai Committee of Science and Technology under Grant No. 21DZ1100100.

References

1. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: Proc. of NeurIPS Workshop (2014)
2. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Proc. of NeurIPS (2017)
3. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. In: Proc. of NeurIPS (2015)
4. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv (2016)
5. Denil, M., Shakibi, B., Dinh, L., Ranzato, M., De Freitas, N.: Predicting parameters in deep learning. In: Proc. of NeurIPS (2013)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (2010)
7. Feng, J.: Bolt. `https://github.com/huawei-noah/bolt` (2021)
8. Gao, P., Ma, T., Li, H., Dai, J., Qiao, Y.: Convmae: Masked convolution meets masked autoencoders. arXiv preprint arXiv:2205.03892 (2022)
9. Gu, J., Li, C., Zhang, B., Han, J., Cao, X., Liu, J., Doermann, D.: Convolutional neural networks for 1-bit cnns via discrete back propagation. In: Proc. of AAAI (2019)
10. Guo, J., Han, K., Wang, Y., Wu, H., Chen, X., Xu, C., Xu, C.: Distilling object detectors via decoupled features. In: Proc. of CVPR (2021)
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. of ICCV (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of CVPR (2016)
13. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks. In: Proc. of IJCAI (2018)
14. Hinton, G., Oriol, Dean, J.: Distilling the knowledge in a neural network. In: Proc. of NeurIPS (2014)
15. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. In: Proc. of CVPR (2017)
16. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: Proc. of ICLR (2016)
17. Li, Q., Jin, S., Yan, J.: Mimicking very efficient network for object detection. In: Proc. of CVPR (2017)
18. Li, R., Wang, Y., Liang, F., Qin, H., Yan, J., Fan, R.: Fully quantized network for object detection. In: Proc. of CVPR (2019)
19. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proc. of CVPR (2017)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proc. of ECCV (2014)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Proc. of ECCV (2016)
22. Liu, Z., Luo, W., Wu, B., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Binarizing deep network towards real-network performance. International Journal of Computer Vision **128**(1), 202–219 (2020)

23. Liu, Z., Shen, Z., Savvides, M., Cheng, K.T.: Reactnet: Towards precise binary neural network with generalized activation functions. In: Proc. of ECCV (2020)
24. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proc. of ECCV (2018)
25. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NeurIPS Workshops (2017)
26. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: Proc. of ECCV (2016)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. of ICLR (2015)
29. Wang, G.H., Ge, Y., Wu, J.: Distilling knowledge by mimicking features. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
30. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: Proc. of CVPR (2019)
31. Wang, Z., Wu, Z., Lu, J., Zhou, J.: Bidet: An efficient binarized object detector. In: Proc. of CVPR (2020)
32. Wu, N.: The maximum entropy method, vol. 32. Springer Science & Business Media (2012)
33. Xu, S., Li, Y., Zhao, J., Zhang, B., Guo, G.: Poem: 1-bit point-wise operations based on expectation-maximization for efficient point cloud processing. In: Proc. of BMVC. pp. 1–10 (2021)
34. Xu, S., Liu, Z., Gong, X., Liu, C., Mao, M., Zhang, B.: Amplitude suppression and direction activation in networks for 1-bit faster r-cnn. In: Proc. of EMDL (2020)
35. Xu, S., Zhao, J., Lu, J., Zhang, B., Han, S., Doermann, D.: Layer-wise searching for 1-bit detectors. In: Proc. of CVPR (2021)
36. Zhao, J., Xu, S., Zhang, B., Gu, J., Doermann, D., Guo, G.: Towards compact 1-bit cnns via bayesian learning. International Journal of Computer Vision **130**(2), 201–225 (2022)
37. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv (2016)
38. Zhuo, L., Zhang, B., Yang, L., Chen, H., Ye, Q., Doermann, D., Ji, R., Guo, G.: Cogradient descent for bilinear optimization. In: Proc. of CVPR (2020)