Information Theoretic Pruning of Coupled Channels in Deep Neural Networks

Supplementary Material

A. Supplementary Materials

Here, we present further results and discussions as supplementary materials to facilitate a more comprehensive understanding of the main findings presented in the main text. The document is structured as follows: Section A.1 delves into the mathematical derivations underlying the tractable regularization term obtained in Section 3.3 of the main text, offering more detailed insights. Section A.3 presents more results on the pruned network configurations achieved through our approach at high sparsity ratios. This includes visualizations of the pruned configurations obtained from ResNet110 network on CIFAR10 and CIFAR100 datasets, as well as the ResNet56 network on the CIFAR10 dataset. Further, Section A.4 provides an elaborate description of the experimental settings adopted in this work. Lastly, we discuss the societal impact of the current work in Section A.5.

A.1. Detailed Mathematical Analysis

In this section, we aim to discuss in greater details the mathematical derivations that led to the tractable upper-bound regularization term obtained in the Right Hand Side (RHS) of the inequality (9) in Section 3.3 of the main paper.

In this regard, recall that as discussed in Section 3.3, the learning process for $\mathbf{h}_s$ could be expressed via Information Bottelneck (IB) [5–7] as

$$\min \left\{ \sum_{s=1}^{S} (\gamma_s I(\mathbf{h}_s, \mathbf{X}) - I(\mathbf{h}_s, y)) \right\} , \tag{1}$$

where $I(\mathbf{h}_s, \mathbf{X})$ is the compression term and $-I(\mathbf{h}_s, y)$ is referred to as the task fidelity term. We further discussed that the task fidelity term can be replaced with the main task loss, and used the compression term to guide the compression of the network. For this term we can write

$$\begin{aligned} I(\mathbf{h}_s, \mathbf{X}) &= \int_{\mathbf{X}, \mathbf{h}_s} p(\mathbf{h}_s, \mathbf{X}) \log \left(\frac{p(\mathbf{h}_s, \mathbf{X})}{p(\mathbf{h}_s) p(\mathbf{X})} \right) d\mathbf{h}_s \, d\mathbf{X} \\ &= \int_{\mathbf{X}, \mathbf{h}_s} p(\mathbf{h}_s, \mathbf{X}) \log \left(\frac{p(\mathbf{h}_s | \mathbf{X}) \cancel{p(\mathbf{X})}}{p(\mathbf{h}_s) \cancel{p(\mathbf{X})}} \right) d\mathbf{h}_s \, d\mathbf{X} \\ &= \int_{\mathbf{X}, \mathbf{h}_s} p(\mathbf{h}_s, \mathbf{X}) \left(\log(p(\mathbf{h}_s | \mathbf{X})) - \log(p(\mathbf{h}_s)) \right) d\mathbf{h}_s \, d\mathbf{X} \, . \end{aligned} \tag{2}$$

It should be noted that in (2), exact calculation of $p(\mathbf{h}_s)$ and consequently, $\log(p(\mathbf{h}_s))$ is intractable. To solve this issue, we aim to find an auxiliary distribution capable of approximating it. Following variational approximation [1, 8], we consider the ratio $\left(\frac{q(\mathbf{h}_s)}{p(\mathbf{h}_s)}\right)$, where $q(\mathbf{h}_s)$ is assumed to be the mentioned tractable distribution. Since $\log(.)$ is a concave function, Jensen's inequality can be applied to $\log\left(\frac{q(\mathbf{h}_s)}{p(\mathbf{h}_s)}\right)$, yielding

$$\begin{aligned} & E_{\mathbf{h}_s} \left\{ \log \left(\frac{q(\mathbf{h}_s)}{p(\mathbf{h}_s)} \right) \right\} \leq \log \left(E_{\mathbf{h}_s} \left\{ \frac{q(\mathbf{h}_s)}{p(\mathbf{h}_s)} \right\} \right) \\ & = \log \left(\int_{\mathbf{h}_s} \cancel{p(\mathbf{h}_s)} \frac{q(\mathbf{h}_s)}{\cancel{p(\mathbf{h}_s)}} d\mathbf{h}_s \right) = \log(1) = 0 \, . \end{aligned} \tag{3}$$

Now, having obtained $E_{\mathbf{h}_s} \left\{ \log \left(\frac{q(\mathbf{h}_s)}{p(\mathbf{h}_s)} \right) \right\} \leq 0$, a simple rearrangement results in

$$-\log\left(p(\mathbf{h}_s)\right) \leq -\log\left(q(\mathbf{h}_s)\right) . \tag{4}$$

Since the goal is to minimize $I(\mathbf{h}_s, \mathbf{X})$, the tractable upper-bound in the RHS of (4) is used to replace the intractable $-\log\left(p\left(\mathbf{h}_s\right)\right)$ in (2). Hence, an upper-bound for $I(\mathbf{h}_s, \mathbf{X})$ is obtained as

$$\begin{aligned} I(\mathbf{h}_s, \mathbf{X}) \leq \int_{\mathbf{X}, \mathbf{h}_s} & p(\mathbf{h}_s, \mathbf{X}) \left\{ \log(p(\mathbf{h}_s | \mathbf{X})) \right. \\ & \left. - \log(q(\mathbf{h}_s)) \right\} d\mathbf{h}_s \, d\mathbf{X} \ \ . \end{aligned} \tag{5}$$

Moreover, inequality (5) can be rewritten in terms of the KL divergence as

$$\begin{aligned} I(\mathbf{h}_s, \mathbf{X}) &\leq \int_{\mathbf{X}} p(\mathbf{X}) \int_{\mathbf{h}_s} p(\mathbf{h}_s | \mathbf{X}) \log \left(\frac{p(\mathbf{h}_s | \mathbf{X})}{q(\mathbf{h}_s)} \right) d\mathbf{h}_s \, d\mathbf{X} \\ &= E_{\mathbf{X}} \left\{ \mathrm{KL} \left(p(\mathbf{h}_s | \mathbf{X}) \, || \, q(\mathbf{h}_s) \right) \right\} \ \ . \end{aligned} \tag{6}$$

Note that this is the same expression as that of formula (3) in the main paper. As discussed in Section 3.3 of the main paper and following [1], both $p(\mathbf{h}_s | \mathbf{X})$ and $q(\mathbf{h}_s)$ can be modeled with Gaussian distributions as

$$\begin{aligned} p(\mathbf{h}_s | \mathbf{X}) &= \mathcal{N}(\mu_{\mathbf{h}_s | \mathbf{X}}, \sigma^2_{\mathbf{h}_s | \mathbf{X}}) \ \ , \\ q(\mathbf{h}_s) &= \mathcal{N}(0, \eta_s^2) \ \ . \end{aligned} \tag{7}$$

The distribution of $q(\mathbf{h}_s)$ is a degree of freedom at our disposal and can be selected appropriately. Therefore, the optimum value for η_s that allows $q(\mathbf{h}_s)$ to better approximate $p(\mathbf{h}_s | \mathbf{X})$ is found by solving $\frac{d}{d\eta_s} \mathrm{KL} \left(p(\mathbf{h}_s | \mathbf{X}) \, || \, q(\mathbf{h}_s) \right) = 0$. In this regard, we first find the closed form expression for $\mathrm{KL} \left(p(\mathbf{h}_s | \mathbf{X}) || q(\mathbf{h}_s) \right)$, then take its derivative $\frac{d}{d\eta_s}$. We have

$$
\begin{aligned}
&\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,\big|\big|\, q(\mathbf{h}_s)\right) \\
&= \int_{\mathbf{h}_s} \{\log(p(\mathbf{h}_s|\mathbf{X})) - \log(q(\mathbf{h}_s))\}\, p(\mathbf{h}_s|\mathbf{X})\, d\mathbf{h}_s \\
&= \int_{\mathbf{h}_s} \left\{ \log\left(\frac{1}{\sqrt{2\pi}\sigma_{\mathbf{h}_s|\mathbf{X}}} \exp\left[-\frac{\left(\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}\right)^2}{2\sigma^2_{\mathbf{h}_s|\mathbf{X}}} \right] \right) \right. \\
&\left. - \log\left(\frac{1}{\sqrt{2\pi}\eta_s} \exp\left[-\frac{\left(\mathbf{h}_s\right)^2}{2\eta_s^2} \right] \right) \right\} \\
&\times \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{h}_s|\mathbf{X}}} \exp\left[-\frac{\left(\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}\right)^2}{2\sigma^2_{\mathbf{h}_s|\mathbf{X}}} \right] d\mathbf{h}_s \\
&= \int_{\mathbf{h}_s} \left\{ -\frac{1}{2}\log(2\pi) - \log(\sigma_{\mathbf{h}_s|\mathbf{X}}) - \frac{1}{2}\left(\frac{\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right)^2 \right. \\
&\left. + \frac{1}{2}\log(2\pi) + \log(\eta_s) + \frac{1}{2}\left(\frac{\mathbf{h}_s - 0}{\eta_s} \right)^2 \right\} \\
&\times \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{h}_s|\mathbf{X}}} \exp\left[-\frac{1}{2}\left(\frac{\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right)^2 \right] d\mathbf{h}_s \\
&= \int \left\{ \log\left(\frac{\eta_s}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right) + \frac{1}{2}\left[\left(\frac{\mathbf{h}_s}{\eta_s} \right)^2 - \left(\frac{\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right)^2 \right] \right\} \\
&\times \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{h}_s|\mathbf{X}}} \exp\left[-\frac{1}{2}\left(\frac{\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right)^2 \right] d\mathbf{h}_s \\
&= E_{\mathbf{h}_s \sim p(\mathbf{h}_s|\mathbf{X})} \left\{ \log\left(\frac{\eta_s}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right) \right. \\
&\left. + \frac{1}{2}\left[\left(\frac{\mathbf{h}_s}{\eta_s} \right)^2 - \left(\frac{\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}}}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right)^2 \right] \right\} \\
&= \log\left(\frac{\eta_s}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right) + \frac{1}{2\eta_s^2} E_{\mathbf{h}_s \sim p(\mathbf{h}_s|\mathbf{X})} \left\{ \mathbf{h}_s^2 \right\} \\
&- \frac{1}{2\sigma^2_{\mathbf{h}_s|\mathbf{X}}} E_{\mathbf{h}_s \sim p(\mathbf{h}_s|\mathbf{X})} \left\{ \left(\mathbf{h}_s - \mu_{\mathbf{h}_s|\mathbf{X}} \right)^2 \right\} \\
&= \log\left(\frac{\eta_s}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right) + \frac{1}{2\eta_s^2}\left(\sigma^2_{\mathbf{h}_s|\mathbf{X}} + \mu^2_{\mathbf{h}_s|\mathbf{X}} \right) - \frac{1}{2} \ . \qquad (8)
\end{aligned}
$$

Therefore, the closed form expression for $\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,\big|\big|\, q(\mathbf{h}_s)\right)$ is found as

$$
\begin{aligned}
&\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,\big|\big|\, q(\mathbf{h}_s)\right) \\
&= \log\left(\frac{\eta_s}{\sigma_{\mathbf{h}_s|\mathbf{X}}} \right) + \frac{1}{2\eta_s^2}\left(\sigma^2_{\mathbf{h}_s|\mathbf{X}} + \mu^2_{\mathbf{h}_s|\mathbf{X}} \right) - \frac{1}{2} \ . \qquad (9)
\end{aligned}
$$

Now, solving $\frac{d}{d\eta_s}\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\big|\big|q(\mathbf{h}_s)\right) = 0$ will yield the optimum value η_s^* for the auxiliary distribution $q(\mathbf{h}_s) = \mathcal{N}(0, \eta_s^2)$. This is given by

$$
\begin{aligned}
&\frac{d}{d\eta_s}\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,\big|\big|\, q(\mathbf{h}_s)\right) \\
&= \frac{1}{\eta_s} - \frac{2}{2\eta_s^3}\left(\sigma^2_{\mathbf{h}_s|\mathbf{X}} + \mu^2_{\mathbf{h}_s|\mathbf{X}} \right) = 0 \ , \qquad (10)
\end{aligned}
$$

from which the optimum value η_s^* is found as

$$
\eta_s^* = \sqrt{\sigma^2_{\mathbf{h}_s|\mathbf{X}} + \mu^2_{\mathbf{h}_s|\mathbf{X}}} \ . \qquad (11)
$$

Therefore, by substituting the optimum value η_s^* in the definition of $q(\mathbf{h}_s)$ in (7), the proper choice for $q(\mathbf{h}_s)$ is selected as

$$
q(\mathbf{h}_s) = \mathcal{N}\left(0,\ \mu^2_{\mathbf{h}_s|\mathbf{X}} + \sigma^2_{\mathbf{h}_s|\mathbf{X}}\right) \ . \qquad (12)
$$

Further, by substituting the optimum η_s^* in (9), the minimum for the KL divergence is found to be

$$
\begin{aligned}
&\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,\big|\big|\, q(\mathbf{h}_s)\right) = \frac{1}{2}\log\left(\frac{\mu^2_{\mathbf{h}_s|\mathbf{X}} + \sigma^2_{\mathbf{h}_s|\mathbf{X}}}{\sigma^2_{\mathbf{h}_s|\mathbf{X}}} \right) \\
&\equiv \log\left(\frac{\mu^2_{\mathbf{h}_s|\mathbf{X}} + \sigma^2_{\mathbf{h}_s|\mathbf{X}}}{\sigma^2_{\mathbf{h}_s|\mathbf{X}}} \right) \ . \qquad (13)
\end{aligned}
$$

The aim is to minimize the RHS of (13). However, it is worth noting that both $\mu_{\mathbf{h}_s|\mathbf{X}}$ and $\sigma_{\mathbf{h}_s|\mathbf{X}}$ are intractable. Alternatively, we can write them in terms of $\mu_{\texttt{SCS}_s|\mathbf{X}}$ and $\sigma_{\texttt{SCS}_s|\mathbf{X}}$, given that the representation of $\mathbf{h}_s$ is considered to be formed through

$$
\begin{aligned}
\texttt{SCS}_s &= f_s(\mathbf{X}) \ , \\
\mathbf{h}_s &= (\mu_s + \epsilon\sigma_s)\,\texttt{SCS}_s \ . \qquad (14)
\end{aligned}
$$

As discussed previously, $f_s(.)$ is a nonlinear function that also exhibits stochasticity because it is affected by all of the random variables controlling the saliency of the coupled sets in the preceding layers. We assume a random variable ϕ to represent this stochasticity. Then, $\mu_{\mathbf{h}_s|\mathbf{X}}$ is calculated as

$$
\begin{aligned}
\mu_{\mathbf{h}_s|\mathbf{X}} &= E_{\phi,\epsilon}\left\{(\mu_s + \epsilon\sigma_s)\texttt{SCS}_s \,|\, \mathbf{X}\right\} \\
&= E_{\epsilon}\left\{\mu_s + \epsilon\sigma_s\right\} E_{\phi}\left\{\texttt{SCS}_s|\mathbf{X}\right\} \\
&= \mu_s \cdot \mu_{\texttt{SCS}_s|\mathbf{X}} \ , \qquad (15)
\end{aligned}
$$

and for $\sigma_{\mathbf{h}_s|\mathbf{X}}$ we can write

$$
\begin{aligned}
\sigma^2_{\mathbf{h}_s|\mathbf{X}} &= E_{\phi,\epsilon}\left\{(\mu_s + \epsilon\sigma_s)^2\, \texttt{SCS}_s^2 \,|\, \mathbf{X}\right\} \\
&- E^2_{\phi,\epsilon}\left\{(\mu_s + \epsilon\sigma_s)\, \texttt{SCS}_s \,|\, \mathbf{X}\right\} \\
&= E_{\epsilon}\left\{\left(\mu_s^2 + 2\mu_s\sigma_s\epsilon + \sigma_s^2\epsilon^2\right)\right\} E_{\phi}\left\{\texttt{SCS}_s^2 \,|\, \mathbf{X}\right\} \\
&- \left(\mu_s \cdot \mu_{\texttt{SCS}_s|\mathbf{X}}\right)^2 \\
&= \left(\mu_s^2 + 0 + \sigma_s^2(0+1)\right)\left(\mu^2_{\texttt{SCS}_s|\mathbf{X}} + \sigma^2_{\texttt{SCS}_s|\mathbf{X}}\right) \\
&- \mu_s^2 \cdot \mu^2_{\texttt{SCS}_s|\mathbf{X}} \\
&= \left(\mu_s^2 + \sigma_s^2\right)\sigma^2_{\texttt{SCS}_s|\mathbf{X}} + \sigma_s^2 \cdot \mu^2_{\texttt{SCS}_s|\mathbf{X}} \ . \qquad (16)
\end{aligned}
$$

Then, substituting $\mu_{\mathbf{h}_s|\mathbf{X}}$ and $\sigma^2_{\mathbf{h}_s|\mathbf{X}}$ from (15) and (16) into (13) yields

$$\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,||\,q(\mathbf{h}_s)\right) \equiv \log\left(1 + \frac{\mu_s^2 \cdot \mu^2_{\texttt{SCS}_s|\mathbf{X}}}{\sigma_s^2 \cdot \mu^2_{\texttt{SCS}_s|\mathbf{X}} + \left(\mu_s^2 + \sigma_s^2\right)\sigma^2_{\texttt{SCS}_s|\mathbf{X}}}\right) \quad , \tag{17}$$

which is the same as formula (7) in the paper. Noting that $\sigma^2_{\texttt{SCS}_s|\mathbf{X}} \geq 0$ appears in the denominator, and that $\log(.)$ is a non-decreasing function, an upper-bound can be obtained for (17) corresponding to $\sigma^2_{\texttt{SCS}_s|\mathbf{X}} = 0$. Substituting $\sigma^2_{\texttt{SCS}_s|\mathbf{X}} = 0$ in (17) results in

$$\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,||\,q(\mathbf{h}_s)\right) \leq \log\left(1 + \frac{\mu_s^2}{\sigma_s^2}\right) \quad . \tag{18}$$

Furthermore, taking expectation with respect to the input $\mathbf{X}$ from both sides of (18) produces an upper-bound for the compression term in IB (*i.e.*, $I(\mathbf{h}_s, \mathbf{X})$) as

$$I(\mathbf{h}_s, \mathbf{X}) \leq E_{\mathbf{X}}\left\{\mathrm{KL}\left(p(\mathbf{h}_s|\mathbf{X})\,||\,q(\mathbf{h}_s)\right)\right\} \leq E_{\mathbf{X}}\left\{\log\left(1 + \frac{\mu_s^2}{\sigma_s^2}\right)\right\} \approx \frac{1}{N_b}\log\left(1 + \frac{\mu_s^2}{\sigma_s^2}\right) \quad . \tag{19}$$

Note that the RHS of (19) should be summed up for all SCS, weighted with proper $\gamma_s = \gamma^{'}|\texttt{SCS}_s|$ in Eq. (1), and optimized along with the primary task loss that represents the task fidelity term in IB. Hence, the overall training loss is

$$CE(y, \hat{y}) + \frac{1}{N_b}\gamma^{'}\sum_{s=1}^{S}|\texttt{SCS}_s|\left[\log\left(1 + \frac{\mu_s^2}{\sigma_s^2}\right)\right] \quad , \tag{20}$$

where $|\texttt{SCS}_s|$ is the number of coupled channels in $\texttt{SCS}_s$ and $\gamma^{'}$ balances the primary task loss and the tractable regularization term (*i.e.*, the compression-accuracy trade-off).

A.2. Transferability to Object Detection

In this subsection, we aim to investigate the transferability of the pruned network configurations obtained via ITPCC to the task of object detection. To this end, we consider the unpruned ResNet50 and the three pruned configurations which were derived from it (in Section 4.3 of the main paper) on the ImageNet classification task, namely, ITPCC-A, ITPCC-B, and ITPCC-C. We utilize them as backbones in Single Shot Detector (SSD) architectures [3], employing both SSD300 and SSD512 variants. Following [3], we train the whole network (including the backbone and the head) on the combined VOC2007 [2] and VOC2012 train/val sets, and evaluate it on the VOC2007 test set. We adopt the head architecture from the original SSD paper and replace the VGG16 backbone with the aforementioned networks.

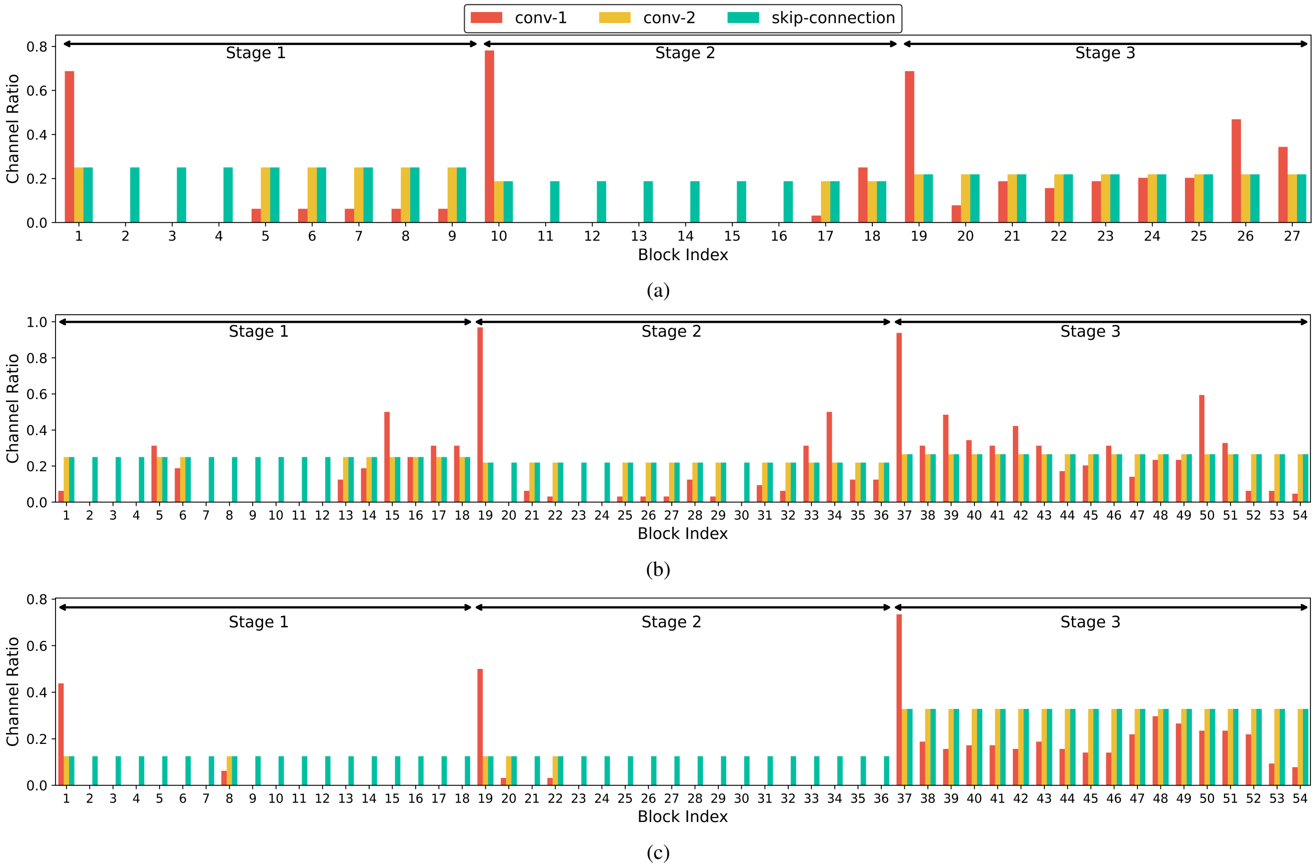
Model	mAP (%)	FLOPs (G)
SSD300-VGG16 [4]	76.72	31.44
FasterRCNN-VGG16 [4]	70.10	91.23
RetinaNet-RN50 [4]	77.27	106.5
SSD300-RN50 (base)	77.79	11.1
SSD300-ITPCC-A (Ours)	**77.86**	6.85
SSD300-ITPCC-B (Ours)	77.06	5.08
SSD300-ITPCC-C (Ours)	75.08	**3.38**
SSD512-RN50-slim [4]	75.83	46.09
SSD512-RN50 (base) [4]	77.98	65.56
SSD512-RN50-HALP [4]	77.42	**15.38**
SSD512-RN50 (base)	80.9	46.24
SSD512-ITPCC-A (Ours)	**81.05**	31.42
SSD512-ITPCC-B (Ours)	80.45	25.6
SSD512-ITPCC-C (Ours)	78.82	20.15

Table 1. Object detection transferability results on PASCAL VOC

Our experimental results and those taken from [4] are presented in Table 1, comparing the mean Average Precision (mAP) and the number of Giga Floating Point Operations denoted as FLOPs (G). Notably, our models showcase significant enhancements in both mAP and FLOPs when contrasted with other commonly used detectors. In comparison to the HALP model [4], which was pruned specifically for object detection task, our transferred SSD512-ITPCC-C strikes a relatively fair compression-mAP trade-off, achieving a 1.4% higher mAP with a moderate increase in FLOPs (20.15 *vs*. 15.38). However, even with a smaller resolution, our SSD300-ITPCC-A still obtains a slightly higher mAP than HALP (77.86 *vs*. 77.42) at a significantly lower FLOPs (6.85 *vs*. 15.38). These findings demonstrate the strong transferability capability of the pruned network configurations uncovered with ITPCC to the task of object detection. Such superior results pave the way for promising future directions towards applying ITPCC to pruning for other tasks such as object detection and image segmentation.

A.3. Further Results on High-sparsity Pruned Configuration Analysis

In this section, further results on the high sparsity network configuration for the ResNet110 network on both CIFAR10 and CIFAR100 datasets as well as for the ResNet56 network on the CIFAR10 dataset are presented in Figures (1a, 1b, 1c). For all of these three scenarios, a similar pruning pattern to that of Figure 4 in the main paper is observed.



(a)

(b)

(c)

Figure 1. Pruned network configuration of (a) ResNet56 on CIFAR10 at 23.80× acceleration with 85.41% accuracy, (b) ResNet110 on CIFAR10 at 23.28× acceleration with 89.45% accuracy, and (c) ResNet110 on CIFAR100 at 37.85× acceleration with 52.77% accuracy.

A.4. Experimental Settings

This section provides details about the settings used for different experiments. The code used for the experiments will be released upon the acceptance of the paper under Creative Commons (🅭) License (CC BY).

ImageNet Classification Experiments: We evaluate the performance of our proposed method on the ResNet50 architecture using the ImageNet dataset. Random flip and crop data augmentation techniques are applied during training. To ensure fair comparison with other pruning algorithms, we perform our experiments starting from the official torch-vision base model (76.13% top-1 accuracy). In the pruning phase, we prune this base model over 180 epochs, starting at an initial learning rate of 10^{-3} and gradually decreasing it to 10^{-4} using a cosine annealing learning rate scheduler to optimize the non-variational network parameters. The variational parameters (μ and σ) maintain a fixed learning rate of 3×10^{-3}. We employ Stochastic Gradient Descent (SGD) optimizer with a batch size of 256, a momentum of 0.9, and a weight decay of 1.5×10^{-3} to optimize the weights. Following pruning, we fix the pruned architecture by freezing the variational parameters. We then fine-tune the non-variational parameters for 90 epochs, starting with a learning rate of 10^{-3} and gradually reduce it to 10^{-5} using the cosine annealing learning rate scheduler. Other training hyper-parameters (*e.g.*, batch size, weight decay, etc.) remain the same as those of the pruning phase. All experiments on ImageNet were performed on three Nvidia V100 (16GB) GPUs.

CIFAR10/100 Classification Experiments: Experiments for the CIFAR10 and CIFAR100 datasets are performed on ResNet56, ResNet110, and MobileNetV2 networks, starting from unpruned models with reported accuracies. Each pruning experiment on CIFAR10/100 is repeated 5 times with different random seeds, and we report mean and standard deviation of the results.

For ResNet56 and ResNet110 networks, the pruning phase lasts for 250 epochs and follows a similar protocol to the ImageNet experiments, except that the initial learning rate is 3×10^{-2} and gradually reduces to 10^{-4}. The fixed learning rate for the variational parameters is 2×10^{-3}. Once the pruning process is completed, we fine-tune the remaining architecture for 200 epochs. We start with an ini-

Layer	setting
SSD300	
MaxPool	(3, 1, 1)
Conv2D_1	(1024, 3, 6, 6)
Conv2D_2	(1024, 1, 0, 1)
SSD512	
MaxPool	(3, 1, 1)
Conv2D_1	(1024, 3, 6, 2)
Conv2D_2	(1024, 3, 6, 2)

Table 2. Details on the layers added on top of the backbone for SSD300 and SSD512 variants. For the MaxPool layers, the reported values in parenthesis are kernel size, stride, and padding values, respectively. As for the Conv2D layers, the reported values are the number of output channels, kernel size, padding, and dilation values.

tial learning rate of 10^{-4} and gradually reduce it to 10^{-5}. A weight decay of 5×10^{-4} is applied for both the pruning and fine-tuning phases. Other hyper-parameters remain consistent with those used in the ImageNet experiments.

For the MobileNetV2 network, we follow a similar training strategy to those of the ResNet56/110 experiments except that a smaller weight decay of 4×10^{-5} is used in both the pruning and finetuning phases. Additionally, in the finetuning phase, we start with a higher learning rate of 10^{-3} and gradually reduce it to 10^{-5} during the course of 250 finetuning epochs, using the cosine annealing learning rate scheduler.

In the ablation experiments, to ensure a fair and consistent comparison, the experimental settings for scenarios (a) and (b) are identical and align with those adopted for the CIFAR100 experiments. For scenario (c), however, we reinitialize the pruned architecture found in scenario (a) and conduct a training process lasting for 450 epochs. It starts with an initial learning rate of 10^{-1} that gradually reduces to 10^{-5}, using a cosine annealing learning rate scheduler. All other training settings remain the same for this scenario. All experiment on CIFAR10 and CIFAR100 datasets were performed on a single Nvidia Titan V (12GB) GPU.

Object Detection Transferability Experiments: In our investigations into object detection transferability, we assess the performance of various backbone architectures, including the full ResNet50 (using torch vision's official pretrained model), and the pruned versions ITPCC-A, ITPCC-B, and ITPCC-C obtained from it in Table 2 of the main paper. We employ the same head as the original Single Shot multibox Detector (SSD) paper [3], but replace the VGG16 backbone with the mentioned ones. To align the backbone's output with the head's input, we streamlined the integration process by removing the last fully connected and average pooling layers of the backbone. Instead, we replace them for a max pooling layer followed by two convolution layers with ReLu activations in between. Details on these added layers are presented in Table 2.

The mentioned backbones pretrained/pruned on ImageNet classification, together with their randomly initialized head, are trained on the union of VOC2007 and VOC2012 train/val sets for the object detection task. This training procedure lasts for 120 epochs, starting from a learning rate of 10^{-3} that gradually decays to 10^{-5} using the cosine annealing scheduler. We employ Stochastic Gradient Descent (SGD) optimizer with a batch size of 32, a momentum of 0.9, and a weight decay of 5×10^{-4} to optimize the weights. All other training settings remain the same as the original SSD paper [3]. All transferability experiments were performed on a single NVIDIA RTX A4000 (16GB) GPU.

A.5. Societal impacts

Our pruning algorithm could have several societal impacts:

- **Increased Accessibility of AI**: By reducing the size and computational demands of deep learning models, this algorithm could make AI technology more accessible to individuals and organizations with limited resources. This could lead to a wider range of applications of AI in areas like healthcare, education, and environmental monitoring.
- **Reduced Energy Consumption**: Smaller models require less power to run, which could contribute to a reduction in the overall energy consumption of AI systems. This could have a positive impact on the environment.
- **Faster Deployment on Devices**: Pruned models can run faster on devices with limited processing power, like smartphones and Internet of Things (IoT) devices. This could enable the development of new AI-powered applications on these devices.

References

[1] Bin Dai, Chen Zhu, and David Paul Wipf. Compressing neural networks using the variational information bottleneck. In *International Conference on Machine Learning*, 2018. 1

[2] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 3

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The*

Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer, 2016. 3, 5

[4] Maying Shen, Hongxu Yin, Pavlo Molchanov, Lei Mao, Jianna Liu, and Jose M. Alvarez. Structural pruning via latency-saliency knapsack. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 12894–12908. Curran Associates, Inc., 2022. 3

[5] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information (2017). *arXiv preprint arXiv:1703.00810*, 1195, 2017. 1

[6] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 1

[7] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015. 1

[8] Chenglong Zhao, Bingbing Ni, Jian Zhang, Qiwei Zhao, Wenjun Zhang, and Qi Tian. Variational convolutional neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2780–2789, 2019. 1