Appendix

A. Proof of Remark 1

Let $z \sim \mathcal{N}(0,1)$ and $y = \max\{0, \gamma_c z + \beta_c\}$. For the suffiency, when $\gamma_c > 0$ we have

$$\begin{aligned} \mathbb{E}_z[y] &= \int_{-\infty}^{-\frac{\beta_c}{\gamma_c}} 0 \cdot \frac{1}{\sqrt{2\pi}} \exp^{-\frac{z^2}{2}} dz + \int_{-\frac{\beta_c}{\gamma_c}}^{+\infty} (\gamma_c z + \beta_c) \cdot \frac{1}{\sqrt{2\pi}} \exp^{-\frac{z^2}{2}} dz, \\ &= \frac{\gamma_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \frac{\beta_c}{2}(1 + \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]), \end{aligned} \tag{9}$$

where $\mathrm{Erf}[x] = \frac{2}{\sqrt{\pi}} \int_0^x \exp^{-t^2} dt$ is the error function. From Eqn.(9), we have

$$\lim_{\gamma_c \to 0^+} \mathbb{E}_z[y] = \lim_{\gamma_c \to 0^+} \frac{\gamma_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \lim_{\gamma_c \to 0^+} \frac{\beta_c}{2}(1 + \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]) = 0 \tag{10}$$

In the same way, we can calculate

$$\begin{aligned} \mathbb{E}_z[y^2] &= \int_{-\infty}^{-\frac{\beta_c}{\gamma_c}} 0 \cdot \frac{1}{\sqrt{2\pi}} \exp^{-\frac{z^2}{2}} dz + \int_{-\frac{\beta_c}{\gamma_c}}^{+\infty} (\gamma_c z + \beta_c)^2 \cdot \frac{1}{\sqrt{2\pi}} \exp^{-\frac{z^2}{2}} dz, \\ &= \frac{\gamma_c \beta_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \frac{\gamma_c^2 + \beta_c^2}{2}(1 + \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]), \end{aligned} \tag{11}$$

From Eqn.(11), we have

$$\lim_{\gamma_c \to 0^+} \mathbb{E}_z[y^2] = \lim_{\gamma_c \to 0^+} \frac{\gamma_c \beta_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \lim_{\gamma_c \to 0^+} \frac{\gamma_c^2 + \beta_c^2}{2}(1 + \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]) = 0 \tag{12}$$

When $\gamma_c < 0$, we have

$$\mathbb{E}_z[y] = -\frac{\gamma_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \frac{\beta_c}{2}(1 - \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]), \tag{13}$$

and

$$\mathbb{E}_z[y^2] = -\frac{\gamma_c \beta_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \frac{\gamma_c^2 + \beta_c^2}{2}(1 - \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]), \tag{14}$$

If γ_c sufficiently approaches 0, we arrive at

$$\lim_{\gamma_c \to 0^-} \mathbb{E}_z[y] = \lim_{\gamma_c \to 0^-} -\frac{\gamma_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \lim_{\gamma_c \to 0^-} \frac{\beta_c}{2}(1 - \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]) = 0 \tag{15}$$

and

$$\lim_{\gamma_c \to 0^-} \mathbb{E}_z[y^2] = \lim_{\gamma_c \to 0^-} \frac{-\gamma_c \beta_c \exp^{-\frac{\beta_c^2}{2\gamma_c^2}}}{\sqrt{2\pi}} + \lim_{\gamma_c \to 0^-} \frac{\gamma_c^2 + \beta_c^2}{2}(1 + \mathrm{Erf}[\frac{\beta_c}{\sqrt{2}\gamma_c}]) = 0 \tag{16}$$

For necessity, we show that if $\mathbb{E}_z[y] = 0$ and $\mathbb{E}_z[y^2] = 0$, then $\gamma_c \to 0$ and $\beta_c \leq 0$. First, if $\gamma_c > 0$, combining Eqn Eqn.(9) and Eqn.(11) gives us $\gamma_c \to 0$ and $\beta_c \leq 0$. If $\gamma_c < 0$, combining Eqn Eqn.(13) and Eqn.(14), we can also obtain $\gamma_c \to 0$ and $\beta_c \leq 0$. This completes the proof.

Note that Eqn.(10) and Eqn.(12) are obtained by assuming that $\gamma_c \to 0$ and $\beta_c \leq 0$. The first condition was verified by (Mehta et al., 2019) that showed that inhibited channels and gamma with small values would emerge at the same time. Here, We evaluate the second assumption in various ResNets trained on the ImageNet dataset. The percentage of $\beta_c \leq 0$ in BN after training are reported in Table 6. We see that a large amount of β_c is non-positive.

CNNs	ResNet18	ResNet50	ResNet101
$(\beta_c \leq 0)$	76.0	76.7	81.8

Table 6. Ratios of $(\beta_c \leq 0)$ after traing on various CNNs.

B. Computation details in 'BN-CE-ReLU' block

As discussed before, CE processes incoming features after normalization layer by combining two branches, $i.e.$, batch decorrelation and instance reweighting. The former computes a covariance matrix and the latter calculates instance variance. We now take 'BN-CE-ReLU' block as an example to show the computation details of statistics in ce. Given a tensor $\boldsymbol{x} \in \mathbb{R}^{N\times C\times H\times W}$, the mean and variance in IN (Ulyanov et al., 2016) are calculated as:

$$(\boldsymbol{\mu}_{\mathrm{IN}})_{nc} = \frac{1}{HW}\sum_{i,j}^{H,W} x_{ncij}, \quad (\boldsymbol{\sigma}^2_{\mathrm{IN}})_{nc} = \frac{1}{HW}\sum_{i,j}^{H,W}(x_{ncij} - (\boldsymbol{\mu}_{\mathrm{IN}})_{nc})^2 \tag{17}$$

Hence, we have $\boldsymbol{\mu}_{\mathrm{IN}}, \boldsymbol{\sigma}^2_{\mathrm{IN}} \in \mathbb{R}^{N\times C}$. Then, the statistics in BN can be reformulated as follows:

$$\begin{aligned}
(\boldsymbol{\mu}_{\mathrm{BN}})_c &= \frac{1}{NHW}\sum_{n,i,j}^{N,H,W} x_{ncij} = \frac{1}{N}\sum_{i}^{N}\frac{1}{HW}\sum_{i,j}^{H,W} x_{ncij} \\
(\boldsymbol{\sigma}^2_{\mathrm{BN}})_c &= \frac{1}{NHW}\sum_{n,i,j}^{N,H,W}(x_{ncij} - (\boldsymbol{\mu}_{\mathrm{BN}})_c)^2 \\
&= \frac{1}{N}\sum_{n}^{N}\frac{1}{HW}\sum_{i,j}^{H,W}(x_{ncij} - (\boldsymbol{\mu}_{\mathrm{IN}})_{nc} + (\boldsymbol{\mu}_{\mathrm{IN}})_{nc} - (\boldsymbol{\mu}_{\mathrm{BN}})_c)^2 \\
&= \frac{1}{N}\sum_{n}^{N}(\frac{1}{HW}\sum_{i,j}^{H,W}(x_{ncij} - (\boldsymbol{\mu}_{\mathrm{IN}})_{nc})^2 + ((\boldsymbol{\mu}_{\mathrm{IN}})_{nc} - (\boldsymbol{\mu}_{\mathrm{BN}})_c)^2) \\
&= \frac{1}{N}\sum_{n}^{N}(\boldsymbol{\sigma}^2_{\mathrm{IN}})_{nc} + \frac{1}{N}\sum_{n}^{N}((\boldsymbol{\mu}_{\mathrm{IN}})_{nc} - (\boldsymbol{\mu}_{\mathrm{BN}})_c)^2
\end{aligned} \tag{18}$$

Then, we have $\mu_{\mathrm{BN}} = \mathbb{E}[\mu_{\mathrm{IN}}]$ and $\sigma^2_{\mathrm{BN}} = \mathbb{E}[\sigma^2_{\mathrm{IN}}] + \mathrm{D}[\mu_{\mathrm{IN}}]$, where $\mathbb{E}[\cdot]$ and $\mathrm{D}[\cdot]$ denote expectation and variance operators over N samples. Further, the input of IR is instance variance of features estimated by $\tilde{\boldsymbol{x}}$, which can be calculated as follows:

$$\begin{aligned}
(\tilde{\boldsymbol{\sigma}}^2_n)_c &= \frac{1}{HW}\sum_{i,j}^{H,W}\left[(\gamma_c\frac{x_{ncij} - (\boldsymbol{\mu}_{\mathrm{BN}})_c}{(\boldsymbol{\sigma}_{\mathrm{BN}})_c} + \beta_c) - (\gamma_c\frac{(\boldsymbol{\mu}_{\mathrm{IN}})_{nc} - (\boldsymbol{\mu}_{\mathrm{BN}})_c}{(\boldsymbol{\sigma}_{\mathrm{BN}})_c} + \beta_c)\right]^2 \\
&= \frac{\gamma_c^2}{(\boldsymbol{\sigma}^2_{\mathrm{BN}})_c}\frac{1}{HW}\sum_{i,j}^{H,W}(x_{ncij} - (\boldsymbol{\mu}_{\mathrm{IN}})_{nc})^2 \\
&= \frac{\gamma_c^2(\boldsymbol{\sigma}^2_{\mathrm{IN}})_{nc}}{(\boldsymbol{\sigma}^2_{\mathrm{BN}})_c}
\end{aligned} \tag{19}$$

Rewritting Eqn.(19) into the vector form gives us $\tilde{\boldsymbol{\sigma}}^2_n = \mathrm{diag}(\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}) \odot \frac{(\boldsymbol{\sigma}^2_{\mathrm{IN}})_n}{\boldsymbol{\sigma}^2_{\mathrm{BN}}}$, where $\mathrm{diag}(\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}) \in \mathbb{R}^{C\times 1}$ extracts the diagonal of the given matrix. At last, the output of BN is $\tilde{x}_{ncij} = \gamma_c\bar{x}_{ncij} + \beta_c$, then the entry in c-th row and d-th column of covariance matrix Σ of $\tilde{x}$ is calculated as follows:

$$\Sigma_{cd} = \frac{1}{NHW}\sum_{n,i,j}^{N,H,W}(\gamma_c\bar{x}_{ncij})(\gamma_d\bar{x}_{ndij}) = \gamma_c\gamma_d\rho_{cd} \tag{20}$$

where ρ_{cd} is the element in c-th row and j-th column of correlation matrix of $\bar{x}$. Thus, we can write Σ into the vector form: $\boldsymbol{\Sigma} = \boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}} \odot \frac{1}{M}\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^{\mathsf{T}}$ if we reshape $\tilde{\boldsymbol{x}}$ to $\tilde{\boldsymbol{x}} \in \mathbb{R}^{C\times M}$ and $M = N \cdot H \cdot W$.

B.1. Architecture of IR branch

We denote the subnetwork in IR branch as $\tilde{f}$. Note that the activation of $\tilde{f}$ is the Sigmoid function, we formulate $\tilde{f}$ following (Hu et al., 2018),

$$\tilde{f}(\boldsymbol{\sigma}_n^2) = \mathrm{Sigmoid}(\boldsymbol{W}_2\delta_1(\mathrm{LN}(\boldsymbol{W}_1\boldsymbol{\sigma}_n^2))) \tag{21}$$

where δ_1 are ReLU activation function, $\boldsymbol{W}_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ and r is reduction ratio ($r = 4$ in our experiments), $\tilde{f}(\boldsymbol{\sigma}_n^2) \in (0,1)^C$ is treated as a gating mechanism in order to control the strength of the inverse square root of variance for each channel. We see that $\tilde{f}$ is expressed by a bottleneck architecture that is able to model channel dependencies and limit model complexity. Layer normalization (LN) is used inside the bottleneck transform (before ReLU) to ease optimization. It is seen from Eqn.(8) that $s^{-\frac{1}{2}}$ represents the quantity of inverse square root of variance and $\tilde{f}(\boldsymbol{\sigma}_n^2)$ regulates the extend of instance reweighting. $\tilde{f}$ maps the instance variance to a set of channel weights. In this sense, the IR branch intrinsically introduces channel dependencies conditioned on each input.

C. Proof of proposition 1

Proposition 1. *Let $\boldsymbol{\Sigma}$ be covariance matrix of feature maps after batch normalization. Then, (1) assume that $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}^{-\frac{1}{2}}$, $\forall k = 1,2,3,\cdots,T$, we have $|\hat{\gamma_c}| > |\gamma_c|$, $\forall c \in [C]$. (2) Let $\tilde{\boldsymbol{x}}_{nij} = \mathrm{Diag}(\boldsymbol{\gamma})\bar{\boldsymbol{x}}_{nij} + \boldsymbol{\beta}$, assume $\boldsymbol{\Sigma}$ is full-rank, then $\left\|\boldsymbol{\Sigma}^{-\frac{1}{2}}\tilde{\boldsymbol{x}}_{nij}\right\|_2 > \|\tilde{\boldsymbol{x}}_{nij}\|_2$*

Proof. (1) Since $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}^{-\frac{1}{2}}$, $\forall k = 1,2,\cdots,T$, we have $\boldsymbol{\Sigma}_k\boldsymbol{\gamma} = \frac{1}{2}\boldsymbol{\Sigma}_{k-1}(3\boldsymbol{I} - \boldsymbol{\Sigma}_{k-1}^2\boldsymbol{\Sigma})\boldsymbol{\gamma} = \boldsymbol{\Sigma}_{k-1}\boldsymbol{\gamma}$. Therefore, we only need to show $\|\hat{\boldsymbol{\gamma}}\|_1 = \|\boldsymbol{\Sigma}_T\boldsymbol{\gamma}\|_1 = \cdots = \|\boldsymbol{\Sigma}_1\boldsymbol{\gamma}\|_1 > \|\boldsymbol{\gamma}\|_1$. Now, we show that for $k = 1$ we have $\left\|\frac{1}{2}(3\boldsymbol{I} - \boldsymbol{\Sigma})\boldsymbol{\gamma}\right\|_1 > \|\boldsymbol{\gamma}\|_1$. From Eqn.(5), we know that $\boldsymbol{\Sigma} = \frac{\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}}{\|\boldsymbol{\gamma}\|_2^2} \odot \boldsymbol{\rho}$ where $\boldsymbol{\rho}$ is the correlation matrix of $\tilde{\boldsymbol{x}}$ and $-1 \leq \rho_{ij} \leq 1$, $\forall i,j \in [C]$. Then, we have

$$\begin{aligned}
\frac{1}{2}(3\boldsymbol{I} - \boldsymbol{\Sigma})\boldsymbol{\gamma} &= \frac{1}{2}(3\boldsymbol{I} - \frac{\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}}{\|\boldsymbol{\gamma}\|_2^2} \odot \boldsymbol{\rho})\boldsymbol{\gamma} \\
&= \frac{1}{2}(3\boldsymbol{\gamma} - (\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathsf{T}}}{\|\boldsymbol{\gamma}\|_2^2} \odot \boldsymbol{\rho})\boldsymbol{\gamma}) \\
&= \frac{1}{2}(3\boldsymbol{\gamma} - \frac{1}{\|\boldsymbol{\gamma}\|_2^2}\left[\sum_j^C \gamma_1\gamma_j\rho_{1j}\gamma_j, \sum_j^C \gamma_2\gamma_j\rho_{2j}\gamma_j, \cdots, \sum_j^C \gamma_C\gamma_j\rho_{Cj}\gamma_j\right]^{\mathrm{T}}) \\
&= \frac{1}{2}(3\boldsymbol{\gamma} - \frac{1}{\|\boldsymbol{\gamma}\|_2^2}\left[\sum_j^C \gamma_1\gamma_j\rho_{1j}\gamma_j, \sum_j^C \gamma_2\gamma_j\rho_{2j}\gamma_j, \cdots, \sum_j^C \gamma_C\gamma_j\rho_{Cj}\gamma_j\right]^{\mathrm{T}}) \\
&= \frac{1}{2}\left[(3 - \sum_j^C \frac{\gamma_j^2\rho_{1j}}{\|\boldsymbol{\gamma}\|_2^2})\gamma_1, (3 - \sum_j^C \frac{\gamma_j^2\rho_{2j}}{\|\boldsymbol{\gamma}\|_2^2})\gamma_2, \cdots, (3 - \sum_j^C \frac{\gamma_j^2\rho_{Cj}}{\|\boldsymbol{\gamma}\|_2^2})\gamma_C\right]^{\mathrm{T}}
\end{aligned} \tag{22}$$

Note that $|3 - \sum_j^C \frac{\gamma_j^2\rho_{ij}}{\|\boldsymbol{\gamma}\|_2^2}| \geq 3 - |\sum_j^C \frac{\gamma_j^2\rho_{ij}}{\|\boldsymbol{\gamma}\|_2^2}| \geq 3 - \sum_j^C \frac{\gamma_j^2}{\|\boldsymbol{\gamma}\|_2^2} = 2$, where the last equality holds iff $\rho_{ij} = 1$, $\forall i,j \in [C]$. However, this is not the case in practice. Hence, for all $c \in [C]$ we have

$$\left|\left[\frac{1}{2}(3\boldsymbol{I} - \boldsymbol{\Sigma})\boldsymbol{\gamma}\right]_c\right| = \left|\frac{1}{2}(3 - \sum_j^C \frac{\gamma_j^2\rho_{cj}}{\|\boldsymbol{\gamma}\|_2^2})\gamma_c\right| > |\gamma_c| \tag{23}$$

Note that if other normalization methods such as IN and LN are used, the conclusion in Proposition 1 still can be drawn when $\tilde{x}$ and $-1 \leq \rho_{ij} \leq 1$, $\forall i,j \in [C]$.

(2) We first show that $\lambda_i \in (0,1), \forall i \in [C]$ where λ_i is the i-th largest eigenvalues of $\mathbf{\Sigma}$. Since $\mathbf{\Sigma}$ is the covariance matrix and has full rank, we have $\lambda_i > 0$. Moreover, $\sum_{i=1}^{C} \lambda_i = \mathrm{tr}(\mathbf{\Sigma}) = 1$. Hence, we obtain that $\lambda_i \in (0,1)$. Based on this fact, the inequality below can be derived,

$$\left\| \mathbf{\Sigma}^{-\frac{1}{2}} \tilde{\boldsymbol{x}}_{nij} \right\|_2^2 = \tilde{\boldsymbol{x}}_{nij}{}^{\mathsf{T}} \mathbf{\Sigma}^{-1} \tilde{\boldsymbol{x}}_{nij} > \frac{1}{\lambda_1} \tilde{\boldsymbol{x}}_{nij}{}^{\mathsf{T}} \tilde{\boldsymbol{x}}_{nij} > \tilde{\boldsymbol{x}}_{nij}{}^{\mathsf{T}} \tilde{\boldsymbol{x}}_{nij} = \| \tilde{\boldsymbol{x}}_{nij} \|_2^2 \tag{24}$$

Hence, $\left\| \mathbf{\Sigma}^{-\frac{1}{2}} \tilde{\boldsymbol{x}}_{nij} \right\|_2 > \| \tilde{\boldsymbol{x}}_{nij} \|_2$. Here completes the proof.

D. Connection between CE block and Nash Equilibrium

We first introduce the definition of Gaussian interference game in context of CNN and then build the connection between a CE block and Nash Equilibrium. For clarity of notation, we omit the subscript n for a concrete sample.

We suppose that each channel can transmit a power vector $\boldsymbol{p}_c = (p_{c11}, \cdots, p_{cHW})$ where p_{cij} denotes the transmit power to the neuron at location (i,j) in the c-th channel. Since normalization layer is often followed by a ReLU activation, we restrict $p_{cij} \geq 0$. In game theory, all channels maximizes sum of maximum information rate of all neurons. We consider dependencies among channels, then channels are thought to play a Gaussian interference game, which can be described by the following maximization problem, for the c-th channel,

$$\begin{aligned} &\max \ \mathcal{C}_c(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C) = \sum_{i,j=1}^{h,W} \ln \left(1 + \frac{g_{cc} p_{cij}}{\sum_{d \neq c} g_{cd} p_{dij} + \sigma_c / h_{cij}} \right) \\ &s.t. \quad \begin{cases} \sum_{i,j=1}^{H,W} p_{cij} = P_c, & \\ p_{cij} \geq 0, & \forall i \in [H], j \in [W] \end{cases} \end{aligned} \tag{25}$$

where g_{cd} represents dependencies between the c-th channel and d-th channel, and $\mathcal{C}_c$ is the sum of maximum information rate with respect to to the c-th channel given transit power distributions $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C$. We also term it pay-off of the c-th channel. In game theory, C channels and solution space of $\{p_{cij}\}_{c,i,j=1}^{C,H,W}$ together with pay-off vector $\boldsymbol{\mathcal{C}} = (\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_C)$ form a Gaussian interference game $\mathbb{G}$. Different from basic settings in $\mathbb{G}$, here we do not restrict dependencies g_{cd} to $(0,1)$. It is known that $\mathbb{G}$ has a unique Nash Equilibrium point whose definition is given as below,

Definition 1. *An C-tuple of strategies $(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C)$ for channels $1, 2, \cdots, C$ respectively is called a Nash equilibrium iff for all c and for all $\mathbf{p}$ ($\mathbf{p}$ a strategy for channel c)*

$$\mathcal{C}_c(\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{c-1}, \boldsymbol{p}, \boldsymbol{p}_{c+1}, \cdots, \boldsymbol{p}_C) \leq \mathcal{C}_c(\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C) \tag{26}$$

i.e., given that all other channels $d \neq c$ use strategies $\boldsymbol{p}_d$, channel c best response is $\boldsymbol{p}_c$. Since $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_C$ are concave in $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_C$ respectively, KKT conditions imply the following theorem.

Theorem 1. *Given pay-off in Eqn.(25), $(\boldsymbol{p}_1^*, \cdots, \boldsymbol{p}_C^*)$ is a Nash equilibrium point if and only if there exist $\boldsymbol{v}_0 = (v_0^1, \cdots, v_0^C)$ (Lagrange multiplier) such that for all $i \in [H]$ and $j \in [W]$,*

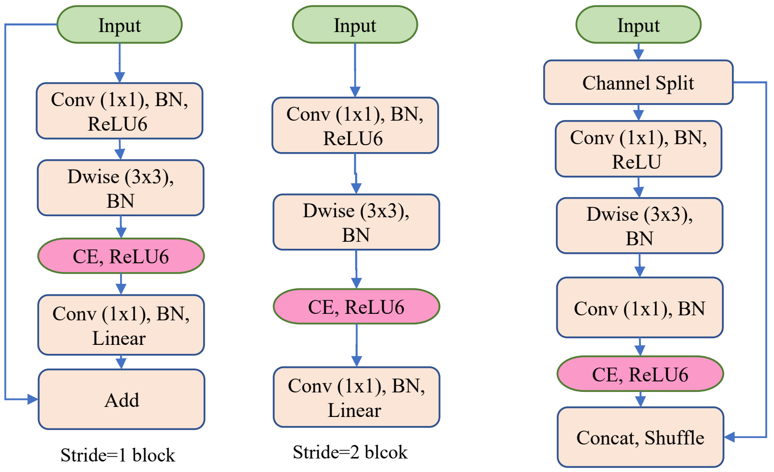
$$\frac{g_{cc}}{\sum_d g_{cd} p_{dij}^* + \sigma_c / h_{cij}} \begin{cases} = v_0^c \text{ for } p_{cij}^* > 0 \\ \leq v_0^c \text{ for } p_{cij}^* = 0 \end{cases} \tag{27}$$

Proof. The Lagrangian corresponding to minimization of $-C_c$ subject to the equality constraint and non-negative constraints on p_{cij} is given by

$$L_c = -\sum_{i,j=1}^{h,W} \ln \left(1 + \frac{g_{cc} p_{cij}}{\sum_{d \neq c} g_{cd} p_{dij} + \sigma_c / h_{cij}} \right) + v_0^c \left(\sum_{i,j=1}^{H,W} p_{cij} - P_c \right) + \sum_{i,j=1}^{H,W} v_1^{cij} (-p_{cij}). \tag{28}$$

Differentiating the Lagrangian with respect to p_{cij} and equating the derivative to zero, we obtain

$$\frac{g_c C}{\sum_d g_{cd} p_{cij} + \sigma_c / h_{cij}} + v_1^{cij} = v_0^c \tag{29}$$



(a) CE blocks in MobileNetv2 **(b) CE block in ShuffleNetv2**

Figure 6. CE blocks in MobileNetv2 (a) and ShuffleNetv2 (b). ‘Add’ denotes broadcast element-wise addition. ‘Concat’ indicates the concatenation of channels. ‘Dwise’ represents the depthwise convolution.

Now, using the complementary slackness condition $v_1^{cij}p_{cij} = 0$ and $v_1^{cij} \geq 0$, we obtain condition (27). This completes the proof.

By Theorem 1, the unique Nash Equilibrium point can be explicitly written as follows when $p^*_{cij} > 0$,

$$\boldsymbol{p}^*_{ij} = \boldsymbol{G}^{-1}\left(\mathrm{Diag}(\boldsymbol{v}_0)^{-1}\mathrm{diag}(\boldsymbol{G}) - \mathrm{Diag}(\boldsymbol{h}_{ij})^{-1}\boldsymbol{\sigma}\right) \tag{30}$$

where $\boldsymbol{p}^*_{ij}, \boldsymbol{h}_{ij}, \boldsymbol{\sigma} \in \mathbb{R}^{C\times 1}$ and $\boldsymbol{v}_0 \in \mathbb{R}^{C\times 1}$ are Lagrangian multipliers corresponding to equality constraints. Note that a approximation can be made using Taylor expansion as follow: $-\frac{\sigma_c}{h_{cij}} = \sigma_c(2 + h_{cij} + \mathcal{O}((1+h_{cij})^2))$. Thus, a linear proxy to Eqn.(30) can be written as

$$\boldsymbol{p}^*_{ij} = \boldsymbol{G}^{-1}\left(\mathrm{Diag}(\boldsymbol{\sigma})\bar{\boldsymbol{h}}_{ij} + \mathrm{Diag}(\boldsymbol{v}_0)^{-1}\mathrm{diag}(\boldsymbol{G}) + (2+\boldsymbol{\delta})\boldsymbol{\sigma}\right) \tag{31}$$

Let $\boldsymbol{G} = [\boldsymbol{D}_n]^{\frac{1}{2}}, \boldsymbol{h}_{ij} = \bar{\boldsymbol{x}}_{ij}, \boldsymbol{\gamma} = \boldsymbol{\sigma}$ and $\boldsymbol{\beta} = \mathrm{Diag}(\boldsymbol{v}_0)^{-1}\mathrm{diag}(\boldsymbol{G}) + (2+\boldsymbol{\delta})\boldsymbol{\sigma}$, Eqn.(31) can surprisingly match CE unit in Eqn.(2), implying that the proposed CE block indeed encourages all the channels to contribute to the layer output. In Gaussian interference game, $\boldsymbol{\sigma}$ is known and v_0 can be determined when budget P_c's are given. However, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learned by SGD in deep neural networks. Note that the Nash Equilibrium solution can be derived for every single sample, implying that the decorrelation operation should be performed conditioned on each instance sample. This is consistent with our design of the CE block.

E. Network Architecture

CE-MobileNetv2 and CE-ShuffleNetv2. As for CE-MobileNetv2, since the last ‘1×1’ convolution layer in the bottleneck is not followed by a Rectified unit, we insert CE in the ‘3×3’ convolution layer which also has the largest number of channels in the bottleneck, as shown in Fig.6(a). Following similar strategies, CE is further integrated into ShuffleNetv2 to construct CE-ShuffleNetv2, as shown in Fig.6(b).

Moving average in inference. Unlike previous methods in manual architecture design that do not depend on batch estimated statistics, the proposed CE block requires computing the inverse square root of a batch covariance matrix $\boldsymbol{\Sigma}$ and a global variance scale s in Eqn.(8) in each training step. To make the output depend only on the input, deterministically in inference, we use the moving average to calculate the population estimate of $\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}$ and $\hat{s}^{-\frac{1}{2}}$ by following the below updating rules:

$$\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}} = (1-m)\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}} + m\boldsymbol{\Sigma}^{-\frac{1}{2}}, \quad \hat{s}^{-\frac{1}{2}} = (1-m)\hat{s}^{-\frac{1}{2}} + m\cdot s^{-\frac{1}{2}} \tag{32}$$

where s and $\boldsymbol{\Sigma}$ are the variance scale and covariance calculated within each mini-batch during training, and m denotes the momentum of moving average. It is worth noting that since $\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}$ is fixed during inference, the BD branch does not introduce extra costs in memory or computation except for a simple linear transformation ($\hat{\boldsymbol{\Sigma}}^{-\frac{1}{2}}\boldsymbol{x}_{\tilde{nij}}$).

Backbone	ResNet50						ResNet18			
Block	GN+ReLU		IN+ReLU		LN+ReLU		BN+ReLU		BN+ELU	
Acc	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
Baseline	75.6	92.8	74.2	91.9	71.6	89.9	70.5	89.4	68.1	87.6
Baseline+CE	**76.2**	**92.9**	**76.0**	**92.7**	**73.3**	**91.3**	**71.9**	**90.2**	**68.7**	**88.5**
Increase	+0.6	+0.1	+1.8	+0.8	+1.7	+1.4	+1.4	+0.8	+0.6	+0.9

Table 7. CE improves top-1 and top-5 accuracy of various normalization methods and rectified units on ImageNet with ResNet50 or ResNet18 as backbones.

Model and computational complexity. The main computation of our CE includes calculating the covariance and inverse square root of it in the BD branch and computing two FC layers in the IR branch. We see that there is a lot of space to reduce computational cost of CE. For BD branch, given an internal feature $\boldsymbol{x} \in \mathbb{R}^{N\times C\times H\times W}$, the cost of calculating a covariance matrix is $2NHWC^2$, which is comparable to the cost of convolution operation. A pooling operation can be employed to downsample featuremap for too large H and W. In this way, the complexity can be reduced to $2NHWC^2/k^2 + CHW$ where k is kernel size of the window of pooling. Further, we can use group-wise whitening to improve efficiency, reducing the cost of computing $\Sigma^{-\frac{1}{2}}$ from TC^3 to TCg^2 (g is group size). For IR branch, we focus on the additional parameters introduced by two FC layers. In fact, the reduction ratio r can be appropriately chosen to balance model complexity and representational power. Besides, the majority of these parameters come from the final block of the network. For example, a single IR in the final block of ResNet-50 has $2 * 2048^2/r$ parameters. In practice, the CE blocks in the final stages of networks are removed to reduce additional parameters. We provide the measurement of computational burden and Flops in Table 1.

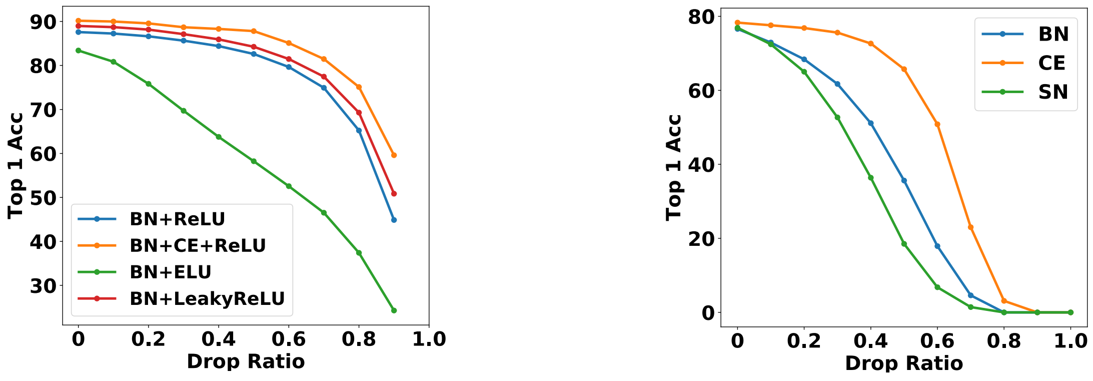
F. Ablative Experiments

CE improves various normalization methods and rectified units. In addition to BN, CE is also effective for other normalization technologies, as inhibited channel emerges in many well-known normalizers as shown in Fig.1. To prove this, we conduct experiments using ResNet-50 under different normalizers including, group normalization (GN), instance normalization (IN), and layer normalization (LN). For these experiments, we stack CE block after the above normalizers to see whether CE helps other normalization methods. Table 7 confirms that CE generalize well over different normalization technologies, improving their generalization on testing samples by 0.6-1.8 top-1 accuracy. On the other hand, CE is also superior to many rectified units such as ELU (Clevert et al., 2015)

Many methods have been proposed to improve normalizers such as switchable normalization (SN) (Luo et al., 2018) and ReLU activation such as exponential linear unit (ELU) (Clevert et al., 2015) and leaky ReLU (LReLU) (Maas et al., 2013). The ablation approach in (Morcos et al., 2018) is used to see whether and how these methods help channel equalization. We demonstrate the effectiveness of CE by answering the following questions.

Do other ReLU-like activation functions help channel equalization? Two representative improvements on ReLU function, i.e. ELU (Clevert et al., 2015) and LReLU (Maas et al., 2013), are employed to see whether other ReLU-like activation functions can help channel equalization. We plot the cumulative ablation curve that depicts ablation ratio versus the top-1 accuracy on CIFAR10 dataset in Fig.7(a). The baseline curve is 'BN+ReLU'. As we can see, the top-1 accuracy curve of 'BN+LReLU' drops more gently, implying that LReLU helps channel equalization. But 'ELU+ReLU' has worse cumulative ablation curve than 'BN+ReLU'. By contrast, the proposed CE block improves the recognition performance of 'BN+ReLU' (higher top-1 accuracy) and promotes channel equalization most (the most gentle cumulative ablation curve).

Do the adaptive normalizers help channel equalization? We experiment on a representative adaptive normalization method (i.e. SN), to see whether it helps channel equalization. SN learns to select an appropriate normalizer from IN, BN and LN for each channel. The cumulative ablation curves are plotted on ImageNet dataset with ResNet-50 under blocks of 'BN+ReLU', 'SN+ReLU' and 'BN+CE+ReLU'. As shown in Fig.7(b), SN even does damage to channel equalization when it is used to replace BN. However, 'BN+CE+ReLU' shows the most gentle cumulative ablation curve, indicating the effectiveness of CE block in channel equalization. Compared with SN, ResNet-50 with CE block also achieves better top-1 accuracy (78.2 vs 76.9), showing that channel equalization is important for block design in a CNN.



(a)

(b)

Figure 7. (a) compares the cumulative ablation curves of ’BN+ReLU’, ’BN+ELU’, ’BN+LReLU’ and ’BN+CE+ReLU’ with VGGNet on CIFAR-10 dataset. We see that the Both LReLU and CE can improve the channel equalization in ’BN+ReLU’ block. (b) compares the cumulative ablation curves of ’BN+ReLU’, ’SN+ReLU’ and ’BN+CE+ReLU’ with ResNet-50 on ImageNet dataset. The proposed CE consistently improves the channel equalization of ’BN+RelU’ block. Note that ’BN+CE+ReLU’ achieves the highest top-1 accuracy on both two datasets compared to its counterparts (when drop ration is 0).

G. Experimental Setup

ResNet Training Setting on ImageNet. All networks are trained using 8 GPUs with a mini-batch of 32 per GPU. We train all the architectures from scratch for 100 epochs using stochastic gradient descent (SGD) with momentum 0.9 and weight decay 1e-4. The cross entropy loss with label smooth (smooth ratio 0.1) is employed. The base learning rate is set to 0.1 and is multiplied by 0.1 after $30, 60$ and 90 epochs. Besides, the covariance matrix in BD branch is calculated within each GPU. Since the computation of covariance matrix involves heavy computation when the size of feature map is large, a 2×2 maximum pooling is adopted to down-sample the feature map after the first batch normalization layer. Like (Huang et al., 2019), we also use group-wise decorrelation with group size 16 across the network to improve the efficiency in the BD branch. By default, the reduction ratio r in IR branch is set to 4.

MobileNet V2 training setting on ImageNet. All networks are trained using 8 GPUs with a mini-batch of 32 per GPU for 150 epochs with cosine learning rate. The base learning rate is set to 0.05 and the weight decay is 4e-5. The cross entropy loss with label smooth (smooth ratio 0.1) is employed.

ShuffleNet V2 training setting on ImageNet. All networks are trained using 8 GPUs with a mini-batch of 128 per GPU for 240 epochs with poly learning rate. The base learning rate is set to 0.5 and weight decay is 4e-5. We also adopt warmup and label smoothing tricks.

VGG networks training setting on CIFAR10. We adopt CIFAR10 that contains 60k images of 10 categories, where 50k images for training and 10k images for test. We train VGG networks with a batch size of 256 on a single GPU for 160 epochs. The initial learning rate is 0.1 and is decreased by 10 times every 60 epochs. The inhibited channel ratios in Fig. 1 and Fig.4(c) is measured by the average ratio for the first six layers. For inference drop experiments in Fig.1(c), we randomly drop channels of the output in the third layer with different dropout ratio. For each ratio, we run the experiment 5 times and average the top 1 accuracy.

Mask-RCNN training setting on COCO. We fine-tune the ImageNet pretrained model in COCO for 24 epoch with base learning rate 0.02 and multiply it by 0.1 after 16 and 22 epochs. All the models are trained using 8 GPUs with a mini-batch of 2 images. The basic backbone structure is adopted from the ResNet50/ResNet101 trained on ImageNet.