Supplementary Materials: Understanding the wiring evolution in differentiable NAS

1 Evolution details of DARTS and DSNAS

1.1 DSNAS (single-level optimization)

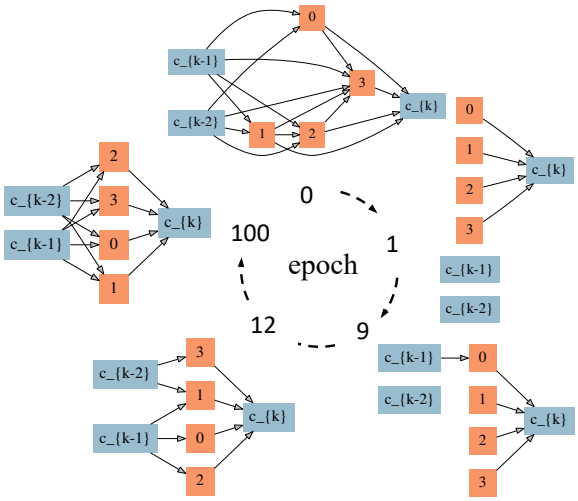

Figure 1: Evolution of the wiring topology with the largest probability in DSNAS. Edges are *dropped* in the beginning. Final cells show preference towards width.

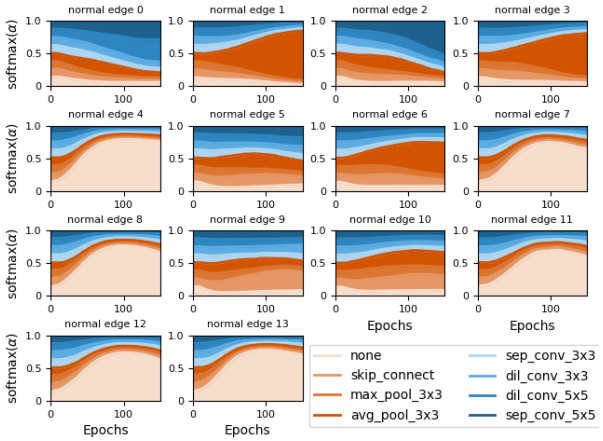

Figure 2: Evolution of p_α on each edge in DSNAS. 4, 7, 8, 11, 12, 13 are intermediate edges where *None* dominates others. This domination is manifested as the *width preference*. The rest are input edges, where *None* only dominates in the beginning. Other operations then become the most likely ones, exhibiting the *growth tendency*.

1.2 DARTS (bi-level optimization)

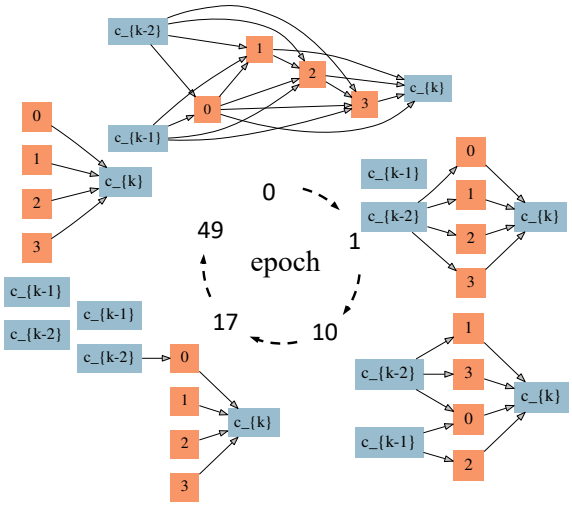

Figure 3: Evolution of the wiring topology with the largest probability in DARTS. Most edges are *dropped* in the beginning, all edges are dropped after 50 epochs of training.

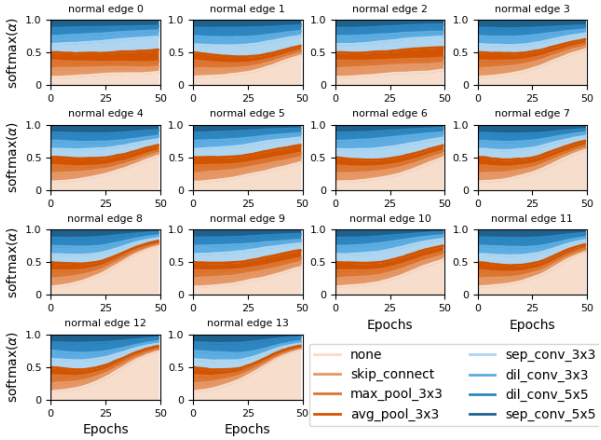

Figure 4: Evolution of p_α with epochs on each edge in DARTS with bi-level optimization. Different from those with single-level optimization, all edges are gradually dropped with training going on. At the end of the evolution, *None* has the largest probability on all edges.

2 Unifying differentiable NAS

2.1 DARTS

Different from SNAS, DARTS (Liu et al., 2018) utilizes an attention mechanism to construct the parent network. To align with (4), we show that with $\hat{\boldsymbol{Z}}_{i,j}^k = \frac{\exp(\boldsymbol{\alpha}_{i,j}^k)}{\sum_m \exp(\boldsymbol{\alpha}_{i,j}^m)}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_{i,j}} &= \frac{\partial \mathcal{L}}{\partial x_j}\frac{\partial x_j}{\partial \boldsymbol{\alpha}_{i,j}} = [\frac{\partial \mathcal{L}}{\partial x_j}\boldsymbol{O}_{i,j}(x_i)]_c \frac{\partial \hat{\boldsymbol{Z}}_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} \\ &= [\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{Z}}_{i,j}}]_c \frac{\partial \hat{\boldsymbol{Z}}_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} = \frac{\partial \hat{\boldsymbol{Z}}_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} C(\hat{\boldsymbol{Z}}_{i,j}),\end{aligned}$$

where as in (4) $[\cdot]_c$ denotes $\cdot$ is a cost function independent from the gradient calculation *w.r.t.* $\boldsymbol{\alpha}$.

2.2 ProxylessNAS

ProxylessNAS (Cai et al., 2018) inherits DARTS's learning objective, and introduce the BinnaryConnect technique to empirically save memory. To achieve that, they propose the following approximation:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_{i,j}} = \frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{Z}}_{i,j}} \frac{\partial \hat{\boldsymbol{Z}}_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} \approx \sum_k [\frac{\partial \mathcal{L}}{\partial Z^k_{i,j}}]_c \frac{\partial \hat{Z}^k_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} = \frac{\partial \hat{\boldsymbol{Z}}_{i,j}}{\partial \boldsymbol{\alpha}_{i,j}} C(\boldsymbol{Z}_{i,j}),$$

where as in (4) $[\cdot]_c$ denotes $\cdot$ is a constant for the gradient calculation *w.r.t.* $\boldsymbol{\alpha}$. Obviously, it is also consistent with (4).

3 Cost mean when stacking 8 cells in DSNAS

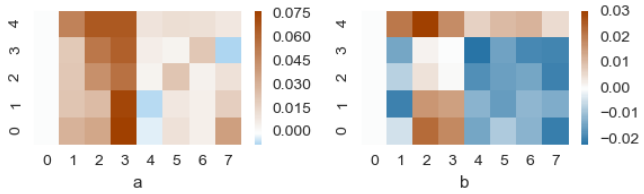

Figure 5: Cost mean statistics for each operation (x-axis) on each edge (y-axis) (a) at initialization and (b) near θ's convergence of by stacking 8 minimal cells. Operation 0: none, Operation 1: skip connect, Operation 2: max_pool_3x3, Operation 3: avg_pool_3x3, Operation 4: sep_conv_3x3, Operation 5: dil_conv_3x3, Operation 6: dil_conv_5x5, Operation 7: sep_conv_5x5.

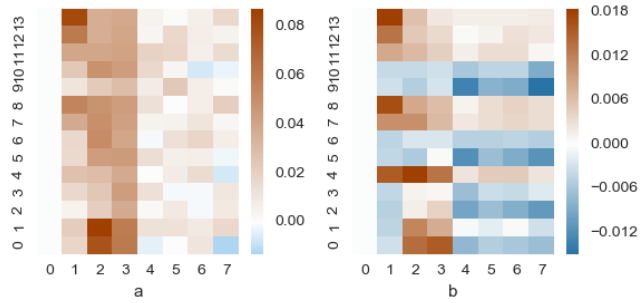

Figure 6: Cost mean statistics for each operation (x-axis) on each edge (y-axis) (a) at initialization and (b) near θ's convergence by stacking 8 original cells as in DARTS and SNAS. Operation 0: none, Operation 1: skip connect, Operation 2: max_pool_3x3, Operation 3: avg_pool_3x3, Operation 4: sep_conv_3x3, Operation 5: dil_conv_3x3, Operation 6: dil_conv_5x5, Operation 7: sep_conv_5x5.

4 Proof of Thm. 1

Theorem 1. *A path does not distribute cost from its output edge after passing one intermediate edge.*

$$C(Z^s_{0,2}) = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}^2_3} \frac{\partial \boldsymbol{X}^2_3}{\partial \boldsymbol{X}^0_2} \boldsymbol{X}^0_2 = 0.$$

Proof. For each intermediate edge, three operations (ReLU, Conv/pooling, BN) are sequentially ordered as shown in Fig. 8. To prove Thm. 1, we analyse the effect of each operation on the cost assignment in the intermediate edge. Let $\boldsymbol{X}^0_2 \in \mathbb{R}^{B\times C_{in}\times W_{in}\times H_{in}}$ denotes the input of Conv operation on edge 4[1], $\boldsymbol{X} \in \mathbb{R}^{B\times C_{out}\times W_{out}\times H_{out}}$ denotes the Conv operation output, $W \in \mathbb{R}^{C_{out}\times C_{in}\times K\times K}$ is the filter weight in the Conv operation. The full proof consists of three following steps.

[1]We do not consider ReLU operation (before Conv) here, since ReLU operation obviously satisfies the proof. Thm. 1 can be easily generalized to pooling operations.

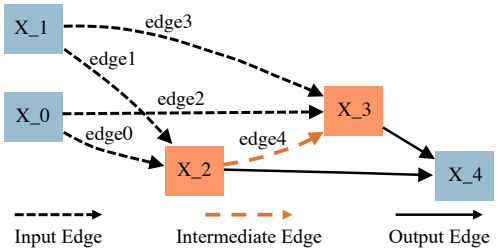

Figure 7: A minimal cell is a cell with 2 *intermediate nodes* (orange), two *input nodes* and one *output node* (blue). Edges connecting input nodes and intermediate nodes ($edge0 - edge3$) are called *input edges*; edges between two intermediate nodes are *intermediate edges* ($edge4$); others are *output edges* which are skip-connections and concatenated together.

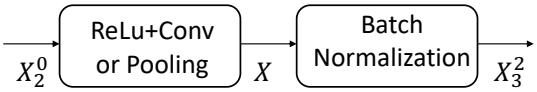

Figure 8: Stacking order of ReLU, Conv/pooling and BN operations on edge 4.

Step 1: By expanding $C(Z^s_{0,2})$ at path (4-3-2-0), we have:

$$\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0. \tag{1}$$

Step 2: Utilizing the **linear property** of Conv operation, we can simplify the above equation as:

$$\underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}}\boldsymbol{X}_2^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\boldsymbol{X}. \tag{2}$$

Proof of Step 2: To derive Eq.(2), we first calculate the gradient w.r.t Conv input $\boldsymbol{X}_2^0$ (**black-underlined part**). Note that w_{in} (w_{out}), h_{in} (h_{out}), c_{in} (c_{out}) and b denote the index of width, height, channel and batch dimensions of Conv input (output), and K is the Conv filter width and height.

$$\underline{\left[\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}\right]_{bc_{in}w_{in}h_{in}}} = \sum_{c_{out}}\sum_{\substack{w_{out}=w_{in}-K \\ h_{out}=h_{in}-K}}^{\substack{w_{in} \\ h_{in}}} \frac{\partial L_{\boldsymbol{\theta}}}{\boldsymbol{X}_{bc_{out}w_{out}h_{out}}} W_{c_{out}c_{in}(w_{in}-w_{out})(h_{in}-h_{out})} \tag{3}$$

Based on Eq.(3), we can simplify the left part of Eq.(2) by doing an **element-wise calculation** on the width (w), height (h), channel (c) and batch (b) dimensions.

$$\begin{aligned}
&\underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}}\boldsymbol{X}_2^0 \\
&= \sum_{b,c_{out},w_{in},h_{in}}\sum_{w_{out}=w_{in}-K}^{w_{in}}\sum_{h_{out}=h_{in}-K}^{h_{in}} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{bc_{out}w_{out}h_{out}}}\sum_{c_{in}} W_{c_{out}c_{in}(w_{in}-w_{out})(h_{in}-h_{out})}[\boldsymbol{X}_2^0]_{bc_{in}w_{in}h_{in}} \\
&= \sum_{b,c_{out},w_{out},h_{out}}\sum_{w_{in}=w_{out}}^{w_{out}+K}\sum_{h_{in}=h_{out}}^{h_{out}+K} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{bc_{out}w_{out}h_{out}}}\sum_{c_{in}} W_{c_{out}c_{in}(w_{in}-w_{out})(h_{in}-h_{out})}[\boldsymbol{X}_2^0]_{bc_{in}w_{in}h_{in}} \\
&= \sum_{b,c_{out},w_{out},h_{out}} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{bc_{out}w_{out}h_{out}}}\underline{\sum_{w_{in}=w_{out}}^{w_{out}+K}\sum_{h_{in}=h_{out}}^{h_{out}+K}\sum_{c_{in}} W_{c_{out}c_{in}(w_{in}-w_{out})(h_{in}-h_{out})}[\boldsymbol{X}_2^0]_{bc_{in}w_{in}h_{in}}} \\
&= \sum_{b,c_{out},w_{out},h_{out}} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{bc_{out}w_{out}h_{out}}}\underline{\boldsymbol{X}_{bc_{out}w_{out}h_{out}}} \\
&= \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\boldsymbol{X},
\end{aligned} \tag{4}$$

where the second equality holds by changing the order of indexes w_{in}, h_{in}, w_{out}, h_{out}. Using the linear property of Conv operation, the **red-underlined** part in the forth equality can be derived due to:

$$\boldsymbol{X}_{bc_{out}w_{out}h_{out}} = \sum_{c_{in}} \sum_{\substack{w_{in}=w_{out} \\ h_{in}=h_{out}}}^{\substack{w_{out}+K \\ h_{out}+K}} W_{c_{out}c_{in}(w_{in}-w_{out})(h_{in}-h_{out})} [\boldsymbol{X}_2^0]_{bc_{in}w_{in}h_{in}}. \tag{5}$$

Step 3: We have shown Conv/pooling operations does not change the cost value in the intermediate edge, so next we analyse the effect of batch normalization (Ioffe and Szegedy, 2015) on the cost assignment. Exploiting the **property of batch normalization** (Ioffe and Szegedy, 2015), we have:

$$\underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2} \frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}} \boldsymbol{X} = \underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}}} \boldsymbol{X} = 0. \tag{6}$$

Proof of Step 3: Similar as step2, we compute the gradient w.r.t. the Batch Normalization input $\boldsymbol{X}$ (**black-underlined part**). Before we start, we first show the batch normalization process as below:

$$\begin{aligned} \mu_c &= \frac{1}{BD} \sum_{b,d} \boldsymbol{X}_{b,c,d}, \qquad \sigma_c^2 = \frac{1}{BD} \sum_{b,d} (\boldsymbol{X}_{b,c,d} - \mu_c)^2, \\ [\hat{\boldsymbol{X}}_3^2]_{b,c,d} &= \frac{\boldsymbol{X}_{b,c,d} - \mu_c}{\sqrt{\sigma_c^2}}, \quad [\boldsymbol{X}_3^2]_{b,c,d} = \gamma_c * [\hat{\boldsymbol{X}}_3^2]_{b,c,d} + \beta_c. \end{aligned} \tag{7}$$

where $\boldsymbol{X}$ and $\boldsymbol{X}_3^2$ denote the input and output of BN operation, μ_c and σ_c^2 are the mean and variance statistics of c-th channel. Note that d denotes the spatial size $Width \times Height$. Then, the gradients with respect with the output $\boldsymbol{X}_{b,c,d}$ (**black-underlined part**) are calculated based on chain rule:

$$\begin{aligned} \underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{b,c,d}}} &= \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\partial [\boldsymbol{X}_3^2]_{b,c,d}}{\partial \boldsymbol{X}_{b,c,d}} + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \mu_c} \frac{\partial \mu_c}{\partial \boldsymbol{X}_{b,c,d}} + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2} \frac{\partial \sigma_c^2}{\partial \boldsymbol{X}_{b,c,d}} (\textbf{\textit{chain rule}}) \\ &= \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\gamma_c}{\sqrt{\sigma_c^2}} + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \mu_c} \frac{1}{BD} + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2} \frac{2(\boldsymbol{X}_{b,c,d} - \mu_c)}{BD} \\ &= \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\gamma_c}{\sqrt{\sigma_c^2}} - \frac{1}{BD} \sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\gamma_c}{\sqrt{\sigma_c^2}} - \frac{\gamma_c}{\sqrt{\sigma_c^2}} (\sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} [\hat{\boldsymbol{X}}_3^2]_{b,c,d}) \frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}}{CD}. \end{aligned} \tag{8}$$

where the gradients with respect to the mean μ_c ($\frac{\partial L_{\boldsymbol{\theta}}}{\partial \mu_c}$) and variance σ_c^2 ($\frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2}$) are shown as below:

$$\begin{aligned} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \mu_c} &= \sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\partial [\boldsymbol{X}_3^2]_{b,c,d}}{\partial \mu_c} - \frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2} \frac{\partial \sigma_c^2}{\partial \mu_c} (\textbf{\textit{chain rule}}) \\ &= -\sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\gamma_c}{\sqrt{\sigma_c^2}} - \sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2} \frac{X_{b,c,d} - \mu_c}{BD/2} \\ &= -\sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\gamma_c}{\sqrt{\sigma_c^2}} \\ \frac{\partial L_{\boldsymbol{\theta}}}{\partial \sigma_c^2} &= \sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{\partial [\boldsymbol{X}_3^2]_{b,c,d}}{\partial \sigma_c^2} (\textbf{\textit{chain rule}}) \\ &= -\frac{\gamma_c}{2} \sum_{b,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{X_{b,c,d} - \mu_c}{(\sigma_c^2)^{3/2}} \\ &= -\frac{\gamma_c}{2} \sum_{b,d} \frac{\partial \mathcal{L}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}} \frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}}{\sigma_c^2} \end{aligned} \tag{9}$$

where the **orange-colored** term is strictly 0 due to the **standard Gaussian mean property** $\sum_{b,d} \frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}}{BD} = 0$.

After getting the specific form of the **black-underlined part**, we can directly derive Eq.(6) by doing an **element-wise calculation**:

$$
\begin{aligned}
&\underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}}\boldsymbol{X} = \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}\frac{\gamma_c \boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_c^2}} - \frac{1}{BD}\sum_{b,c,d}(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}})\frac{\gamma_c \boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_c^2}} \\
&- \sum_{b,c,d}\frac{\gamma_c}{BD}(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d})[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_c^2}} \\
&= \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}\frac{\gamma_c(\boldsymbol{X}_{b,c,d}-\mu_c)}{\sqrt{\sigma_c^2}} - \underline{\frac{1}{BD}\sum_{b,c,d}(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}})\frac{\gamma_c(\boldsymbol{X}_{b,c,d}-\mu_c)}{\sqrt{\sigma_c^2}}} \\
&- \sum_{b,c,d}\frac{\gamma_c}{BD}(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d})[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_c^2}} \\
&= \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}\gamma_c[\hat{\boldsymbol{X}}_3^2]_{b,c,d} - \sum_{b,c,d}\frac{\gamma_c}{BD}(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d})[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_c^2}} \\
&= \sum_{c}\gamma_c(\sum_{b,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial[\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d})(1-\sum_{b,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}^2}{BD}) \\
&= 0,
\end{aligned}
\tag{10}
$$

where the **orange-underlined** term is strictly 0 due to the **standard Gaussian mean property** $\sum_{b,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}}{BD}=0$, the last equality also holds due to **standard Gaussian variance property** $\sum_{b,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}^2}{BD}=1$.

This result is consistent with (Tian, 2018), in which batch normalization is proved to project the input gradient to the orthogonal complement space spanned by the batch normalization input and vector $\mathbf{1}$. Note that this result can be generalized to arbitrary back-propagation path involving intermediate edges and **other normalization methods**, like instance normalization (Ulyanov et al., 2016), layer normalization (Ba et al., 2016). See Appx.8.

One exception from the proof above is the *skip* operation, in which no BN is added. Let $\boldsymbol{X}_2^0 \in \mathbb{R}^{B\times C_{in}\times W_{in}\times H_{in}}$ denotes the input of skip operation on edge 4. We have

$$
\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\boldsymbol{X}_3^2. \tag{11}
$$

where $\boldsymbol{X}_3^2 = \boldsymbol{X}_2^0$. Obviously, if *skip* is sampled on edge 4, it does not block the cost assignment as in Thm 1. However, the influence of the *skip* operation on edge 4 can be ignored since the cost magnitude is much smaller than other operations. □

5 Validation of Cor. 1.2

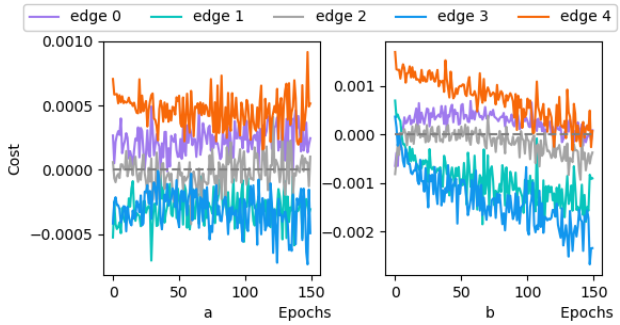

Figure 9: (a) Averaged cost on each edge of the second last cell. (b) Averaged cost on each edge of the last cell.The cost sum of edges in (a) is 0, but it is not 0 in (b).

6 Proof of Thm. 2

Theorem 2. *Cost at output edges of the last cell has the form $C_{\boldsymbol{Z}} = L_{\boldsymbol{\theta}} - H_{\boldsymbol{\theta}}$. It is negatively related to classification accuracy. It tends to be positive at low accuracy, negative at high accuracy.*

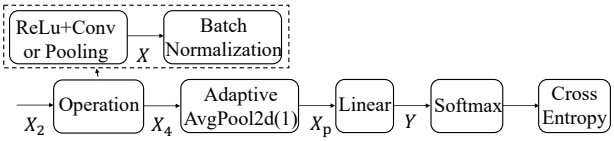

Figure 10: Post-processing the output from the last cell for loss.

Proof. **Negatively related**: We can first prove that for the last cell's output edges cost of one batch M with sampled architecture $\boldsymbol{Z}$ has an equivalent form:

$$
\begin{aligned}
C_{\boldsymbol{Z}} &= \sum_{b,c,d} \underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_4]_{b,c,d}}} [\boldsymbol{X}_4]_{b,c,d} \\
&= \frac{1}{BD} \sum_{b,c,d,n} \underline{[-w_{cn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} w_{cn}]} [\boldsymbol{X}_4]_{b,c,d} \\
&= \frac{1}{B} \sum_{b,c,n} [-w_{cn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} w_{cn}] [\boldsymbol{X}_p]_{b,c} \\
&= \frac{1}{B} \sum_{b,n} [-Y_{bn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} Y_{bn}] \\
&= \frac{1}{B} \sum_{b,n} [-\log(\exp(Y_{bn_b})) + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} Y_{bn} + \log(\sum_q \exp(Y_{bq})) - \log(\sum_q \exp(Y_{bq}))] \\
&= \frac{1}{B} \sum_{b} [-\log \frac{\exp(Y_{bn_b})}{\sum_q \exp(Y_{bq})} + \sum_n \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} \log \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})}] \\
&= L_{\boldsymbol{\theta}} - H_{\boldsymbol{\theta}},
\end{aligned}
\tag{12}
$$

where n_b is the corresponding b-th image class label, $L_{\boldsymbol{\theta}} = -\frac{1}{B} \log \frac{\exp(Y_{bn_b})}{\sum_q \exp(Y_{bq})}$ is Eq.3, $H_{\boldsymbol{\theta}}$ is the entropy of network output, w_{ch} is the weight parameter in the linear layer, $[\boldsymbol{X}_p]_{b,c} = \frac{1}{D} \sum_d [\boldsymbol{X}_4]_{b,c,d}$ is an element of the output from the adaptive average pooling, $Y_{bn} = \sum_c [\boldsymbol{X}_p]_{b,c} w_{cn}$ is an element of the output of the last linear layer. Obviously, the cost sum is positively correlated to the loss, thus negatively correlated to the accuracy. The **black-underlined part** is derived by chain rule:

$$
\begin{aligned}
\underline{\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_4]_{b,c,d}}} &= \sum_n \frac{\partial L_{\boldsymbol{\theta}}}{\partial Y_{bn}} \frac{\partial Y_{bn}}{\partial [\boldsymbol{X}_p]_{b,c}} \frac{\partial [\boldsymbol{X}_p]_{b,c}}{\partial [\boldsymbol{X}_4]_{b,c,d}} \\
&= \frac{1}{BD} \sum_n [-w_{cn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} w_{cn}].
\end{aligned}
\tag{13}
$$

We conduct experiments and record on $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ for a single minimal cell by fixing $\boldsymbol{\alpha}$ and and uniformly sampling networks in DSNAS, updating $\boldsymbol{\theta}$ for 150 epochs (Fig. 11). We also fix $\boldsymbol{\alpha}$ and update $\boldsymbol{\theta}$ for 50 epochs in DARTS (Fig. 12). The results validate this proof. Their dynamical trends also validate Eq. 5.2 and Remark 1.

Fig. 13 shows the result on the search set of DARTS. Note that the search set if a set of data used to train $\boldsymbol{\alpha}$, held-out from the training of $\boldsymbol{\theta}$. Curves in this figure illustrate our analysis in Remark 4. The loss decelerates its decreasing compared with the loss in training set (Fig. 12), while the entropy keeps decreasing.

Positive at low accuracy: Exploiting normalization and weight initialization, we have:

$$
\mathbb{E}_{\boldsymbol{\theta}_0}[C_{\boldsymbol{Z}} > 0],
$$

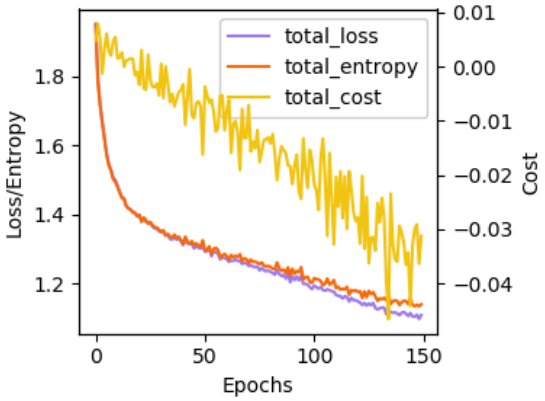

Figure 11: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DSNAS. X-axis is the epoch number, y-axis on the left is for $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ and the right is for $C_{\boldsymbol{Z}}$. Note that the unit for $C_{\boldsymbol{Z}}$ is magnified to show its trend.

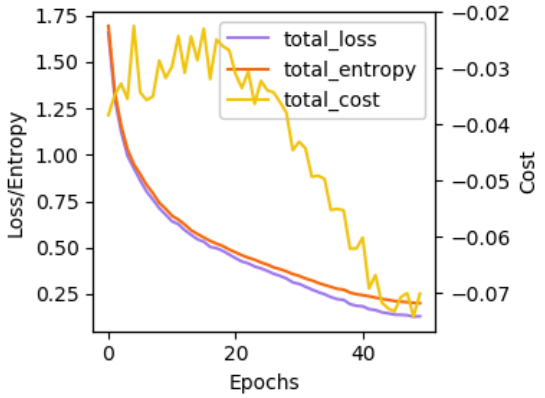

Figure 12: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DARTS in training set. X-axis is the epoch number, y-axis on the left is for $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ and the right is for $C_{\boldsymbol{Z}}$. Note that the unit for $C_{\boldsymbol{Z}}$ is magnified to show its trend.

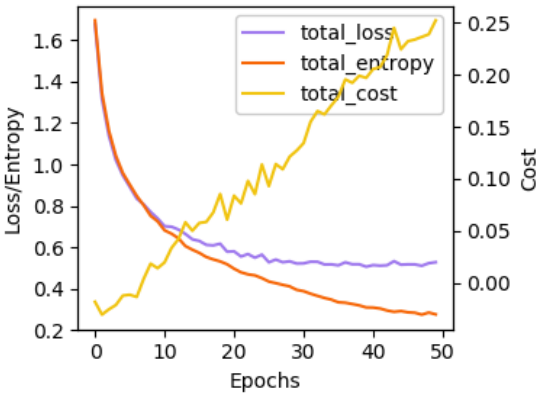

Figure 13: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DARTS in search set. X-axis is the epoch number, y-axis on the left is for $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ and the right is for $C_{\boldsymbol{Z}}$. Note that the unit for $C_{\boldsymbol{Z}}$ is magnified to show its trend.

Note that in our derivation, we follow the commonly used parameter initialization method. With $[\boldsymbol{X}_p]_{b,c} \sim \mathcal{N}(0,\cdot)$, $w_{c,n} \sim \mathcal{N}(0,\cdot)$ at the initialization point, we have $y_1, y_2 \sim \mathcal{N}(0,\cdot)$ and $\mathbb{E}_{y_1,y_2\sim\mathcal{N}(0,\cdot)}[y_1 \exp(y_1+y_2)] > 0$, thus we can prove $\mathbb{E}_{\boldsymbol{\theta}_0}[C_{\boldsymbol{Z}} > 0]$.

Negative at high accuracy: With operation parameters updated towards convergence, the probability of b-th

image being classified to the correct label n_b increases towards 1. Since $Y_{bn_b} = \max\{Y_{bn}\}$, we have

$$C_{\boldsymbol{Z}} \propto \sum_n [-Y_{bn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} Y_{bn}]$$
$$\leq \sum_n [-Y_{bn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})} Y_{bn_b}] = 0.$$

Fig. 14, Fig. 15 and Fig. 16 show the comparison of $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ in the training set and the search set, which validate this proof. Note that the increase of the negative cost for correct classification in later epochs is mainly because the probability of b-th image being classified to the correct label n_b increases towards 1.

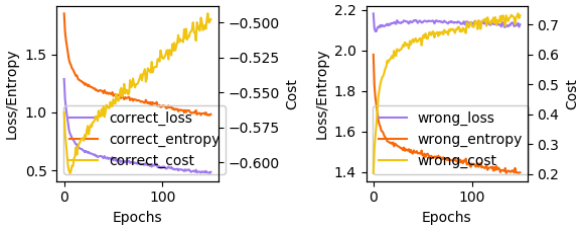

Figure 14: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DSNAS for (a) correct classification and (b) wrong classification.

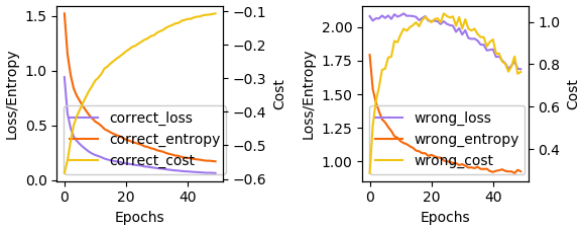

Figure 15: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DARTS for (a) correct classification and (b) wrong classification in training set

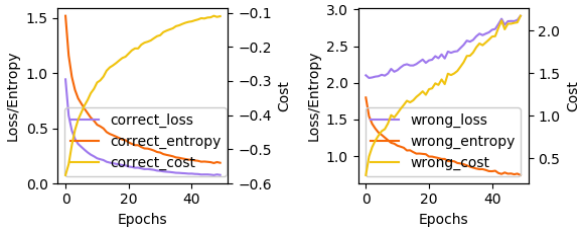

Figure 16: $C_{\boldsymbol{Z}}$, $L_{\boldsymbol{\theta}}$, $H_{\boldsymbol{\theta}}$ in DARTS for (a) correct classification and (b) wrong classification in search set

□

7 Details for empirical study in Sec. 5.3

In Sec. 5.3, we introduced our empirical study to illustrate the distinctive role of intermediate edges. Here we provide more details to help readers understand our design of ablation. Fig. 17 shows the cost of edges in simplified cell (Fig. 8(a)), averaged over operations. The variation here is mainly different operation combination on edges, including fixing to one operation. Ones can see that $edge(1,2)$ always has the largest cost. Fig. 18 shows experiments on modified cell (Fig. 8(b)), where $edge(1,3)$ is deleted. When we still sample operation on $edge(0,1)$, the cost on $edge(1,2)$ is still larger than $edge(0,1)$ and $edge(0,2)$, as shown in Fig. 18(a)&(b). This is only altered by also fixing the operation on $edge(0,1)$. In sum, the existence of $edge(1,3)$ and the sampling on $edge(0,1)$ are two major factors causing the discrimination towards the intermediate edge.

Interestingly, even though in DARTS *None* is excluded and a *post-hoc* derivation scheme that select edges with top-k α is designed, intermediate edges are barely chosen. This can be explained by the observations here. Since the cost on intermediate edges are positive most of the time, their α must be smaller than other edges.

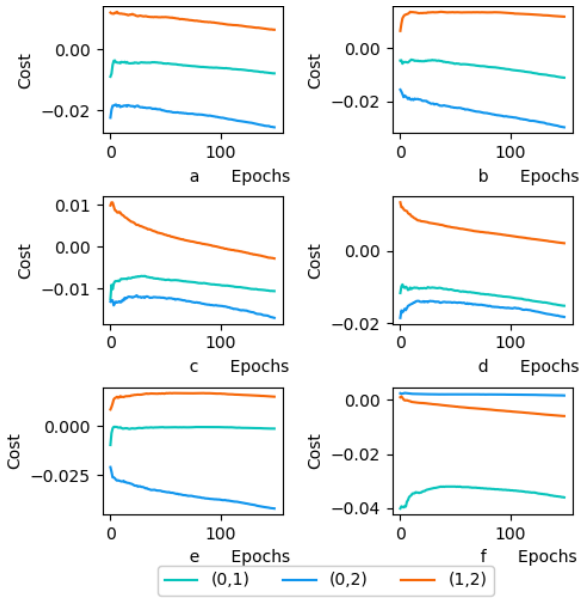

Figure 17: Averaged cost on each edge by (a) sampling none/skip/pool/conv on $edge(0,1)$, $edge(0,2)$, $edge(1,2)$, (b) sampling none/skip/pool/conv on $edge(0,1)$, $edge(0,2)$ and sampling none/skip/pool on $edge(1,2)$, (c) sampling none/skip/pool/conv on $edge(0,1)$, $edge(0,2)$ and sampling none/conv on $edge(1,2)$, (d) sampling none/skip/pool/conv on $edge(0,2)$, $edge(1,2)$ and sampling none/conv on $edge(0,1)$, (e) sampling none/skip/pool/conv on $edge(0,2)$, $edge(1,2)$ and sampling none/pool/skip on $edge(0,1)$, (f) sampling none/skip/pool/conv on $edge(1,2)$ and fixing operation on $edge(0,1)$, $edge(0,2)$. For almost all cases, $edge(1,2)$ has the greatest cost.

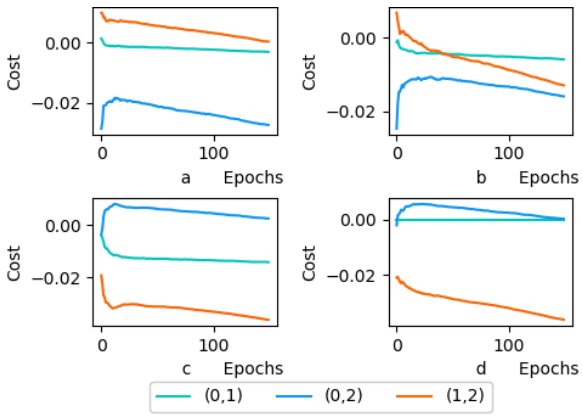

Figure 18: Averaged cost on each edge by deleting $edge(1,3)$ and (a) sampling none/skip/pool/conv on $edge(0,1)$, $edge(0,2)$, $edge(1,2)$, (b) sampling none/skip/pool/conv on $edge(0,2)$, $edge(1,2)$ and sampling skip/pool/conv on $edge(0,1)$, (c) sampling none/skip/pool/conv on $edge(0,2)$, $edge(1,2)$ and fixing operation on $edge(0,1)$, (d) sampling none/skip/pool/conv on $edge(0,2)$ and sampling none/pool/conv on $edge(1,2)$ and fixing operation on $edge(0,1)$. Only in (c) and (d) $edge(1,2)$ has the lowest cost. The cost on $edge(0,1)$ being close to zero here also validates our proof of Thm. 1 in Appx. 4.

8 Further discussion

8.1 Other Normalization Methods

Ones may wonder if the inductive bias discussed in the work could be avoid by shifting to other normalization units, *i.e.* instance normalization (Ulyanov et al., 2016), layer normalization (Ba et al., 2016). However, they seem no option than BN. The cost of instance normalization (affine-enabled) and layer normalization (affine-enabled) is the same as that of batch normalization. As an example, we show that layer normalization still satisfies Thm. 1. The proof of instance normalization is similar.

Proof: The layer normalization process is shown as below:

$$\begin{aligned} \mu_b &= \frac{1}{CD}\sum_{c,d} \boldsymbol{X}_{b,c,d}, \qquad \sigma_b^2 = \frac{1}{CD}\sum_{c,d}\left(\boldsymbol{X}_{b,c,d} - \mu_b\right)^2, \\ [\hat{\boldsymbol{X}}_3^2]_{b,c,d} &= \frac{\boldsymbol{X}_{b,c,d} - \mu_b}{\sqrt{\sigma_b^2}}, \quad [\boldsymbol{X}_3^2]_{b,c,d} = \gamma_b * [\hat{\boldsymbol{X}}_3^2]_{b,c,d} + \beta_b. \end{aligned} \tag{14}$$

Similarly, we have:

$$\begin{aligned} &\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\boldsymbol{X} = \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}\frac{\gamma_b \boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_b^2}} - \frac{1}{CD}\sum_{b,c,d}\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}\Big)\frac{\gamma_b \boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_b^2}} \\ &- \sum_{b,c,d}\frac{\gamma_b}{CD}\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\Big)[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_b^2}} \\ &= \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}\frac{\gamma_b(\boldsymbol{X}_{b,c,d} - \mu_b)}{\sqrt{\sigma_b^2}} - \underline{\frac{1}{CD}\sum_{b,c,d}\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}\Big)\frac{\gamma_b(\boldsymbol{X}_{b,c,d} - \mu_b)}{\sqrt{\sigma_b^2}}} \\ &- \sum_{b,c,d}\frac{\gamma_b}{CD}\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\Big)[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_b^2}} \\ &= \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}\gamma_b[\hat{\boldsymbol{X}}_3^2]_{b,c,d} - \sum_{b,c,d}\frac{\gamma_b}{CD}\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\Big)[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\frac{\boldsymbol{X}_{b,c,d}}{\sqrt{\sigma_b^2}} \\ &= \sum_{b}\gamma_b\Big(\sum_{c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_3^2]_{b,c,d}}[\hat{\boldsymbol{X}}_3^2]_{b,c,d}\Big)\Big(1 - \sum_{c,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}^2}{CD}\Big) \\ &= 0, \end{aligned} \tag{15}$$

where the **orange-underlined** term is strictly 0 due to the **standard Gaussian mean property** $\sum_{c,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}}{CD} = 0$, the last equality also holds due to **standard Gaussian variance property** $\sum_{c,d}\frac{[\hat{\boldsymbol{X}}_3^2]_{b,c,d}^2}{CD} = 1$.

Interestingly, the cost mean statistic collected on different edges by using affine-disabled instance normalization are exactly zero, and they will still be zero after training. Fig. 19 shows the cost mean statistic.

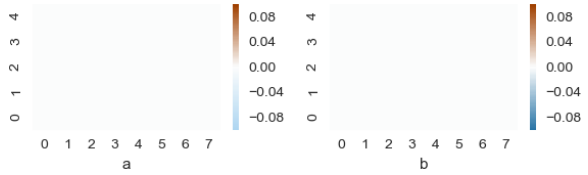

Figure 19: Cost mean statistics for each operation (x-axis) on each edge (y-axis) by using affine-disabled instance normalization.

We can derive the cost function $C(Z_{i,j}^s)$ by using affine-disabled instance normalization. Based on the path (4-3-0) (or (4-3-1), (4-3-2)), we write the cost function explicitly on *edge*2 (*edge*3, *edge*4). The cost function on *edge*2

($edge3$, $edge4$) is derived to be zero for all operations.

$$\begin{aligned} C(Z^s_{i,j}) &= \sum_{b,c,d} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}^3_4]_{b,c,d}} [\boldsymbol{X}^0_3]_{b,c,d} \\ &= \frac{1}{D} \sum_{b,c} \frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_p]_{b,c}} \sum_d [\boldsymbol{X}^0_3]_{b,c,d} \\ &= 0 \end{aligned} \tag{16}$$

where $\sum_d [\boldsymbol{X}^3_4]_{b,c,d} = 0$ (instance normalization property).

The cost on $edge0$ ($edge1$) can be derived similarly and follows the same conclusion as $edge2$ ($edge3$, $edge4$).

8.2 Stacking order in operations

Another consideration could be to change the order of Conv, BN and ReLU units in operations. In our proposed framework, we can show that both theoretically and empirically, same phenomenon would occur.

Fig. 20 shows the cost mean statistic on different edges by using *Conv-ReLU-BN* or *Pooling-BN* order in the minimal cell structure.

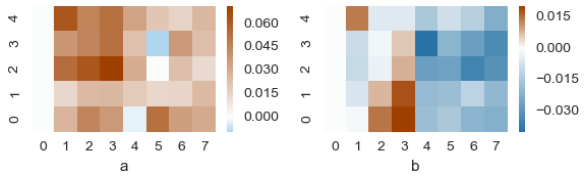

Figure 20: Cost mean statistics for each operation (x-axis) on each edge (y-axis) (a) at initialization and (b) near convergence of θ by using *Conv-ReLU-BN* order. Operation 0: none, Operation 1: skip connect, Operation 2: max_pool_3x3, Operation 3: avg_pool_3x3, Operation 4: sep_conv_3x3, Operation 5: dil_conv_3x3, Operation 6: dil_conv_5x5, Operation 7: sep_conv_5x5.

Fig. 21 shows the cost mean statistic on different edges by using *Conv-BN-ReLU* or *Pooling-BN* order in the minimal cell structure.

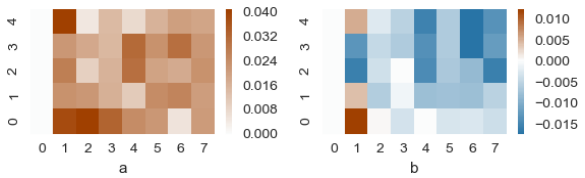

Figure 21: Cost mean statistics for each operation (x-axis) on each edge (y-axis) (a) at initialization and (b) near convergence of θ by using *Conv-BN-ReLU* order. Operation 0: none, Operation 1: skip connect, Operation 2: max_pool_3x3, Operation 3: avg_pool_3x3, Operation 4: sep_conv_3x3, Operation 5: dil_conv_3x3, Operation 6: dil_conv_5x5, Operation 7: sep_conv_5x5.

The proof in Appx.4 and Appx.6 are also valid after the change of order, since the order of *ReLU* operation does not make a difference.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Cai, H., Zhu, L., and Han, S. (2018). Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Tian, Y. (2018). A theoretical framework for deep locally connected relu network. *arXiv preprint arXiv:1809.10829*.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.