Appendix A

In this appendix, we include: (I) proof of the property stated in Section 3.2, (II) the detailed supernet structure and search space.

A-I: Proof of Diversity Score Property

In this section, we show a more detailed formula of the property stated in Section 3.2 and the proof of the property.

Property: Assume that \( h_m := (o_{1,m}, \cdots, o_{j,m}, \cdots, o_{K,m}) \) and \( h'_m := (o_{1,m}, \cdots, o'_{j,m}, \cdots, o_{K,m}) \) are different only by \( j^{th} \) operator. Denote the indexes of operators in \( h_m \) and \( h'_m \) as \( \sigma_1, \sigma_2, \cdots, \sigma_K \) and \( \sigma'_1, \sigma'_2, \cdots, \sigma'_K \). If \( S_{i,k}^m < S'_{i,k} \) for \( k = 1, 2, \cdots, K \) and \( r_{2}^{\sigma_i} > r_{2}^{\sigma'_i} \), then we have:

\[
\text{Score}(h_m) > \text{Score}(h'_m),
\]

where \( \sigma_j \) and \( \sigma'_j \) equal to \( i \) and \( i' \).

Proof: Given the property of matrix determinant and definition of \( L_y^m \), the diversity score of \( h_m \) could be expressed as:

\[
\text{Score}(h_m) = \prod_{i=1}^{K} r_{2}^{\sigma_i} \cdot \det(S_{i,k}^m).
\]

where \( S_{i,k}^m \) are the corresponding submatrixs of \( h_m \) in \( S_m \).

According to the assumption, we know that \( \prod_{i=1}^{K} r_{2}^{\sigma_i} > \prod_{i=1}^{K} r_{2}^{\sigma'_i} \). Now, if \( \det(S_{i,k}^m) \) is greater than \( \det(S'_{i,k}) \) then the property holds easily. Because \( h_m \) and \( h'_m \) are only different by the \( j^{th} \) operator and \( S_m \) is a symmetry matrix, the number of total different entries between \( S_{i,k}^m \) and \( S'_{i,k} \) is less than \( 2K \). We could construct a series of matrixs \( B_i \in \mathbb{R}^{K \times K}, i = 0, 1, 2, \cdots, K \) as following:

\[
B_i(k,l) = \begin{cases} 
S_{i,k}^m(k,l), & k < i, l = j, \\
S_{i,k}^m(k,l), & l < i, k = j, \\
S'_{i,k}^m(k,l), & \text{Otherwise},
\end{cases}
\]

where \( B_i(k,l) \) is the entry in row \( k \) column \( l \). We then prove the following inequality by induction:

\[
\det(B_i) \leq \det(B_{i+1}), i = 0, 1, 2, \cdots, K - 1.
\]

*This work is done when Minghao is an intern at Microsoft.
For $i = 0$, consider matrix $A$ defined as follow:

$$A(k,l) = \begin{cases} 
S^{w}_{m}(1,j) & k = 1, l = j, \\
S^{w}_{m}(1,j) & l = 1, k = j, \\
S^{w}_{m}(j,1) & S^{w}_{m}(1,j) = 1, \\
S^{w}_{m}(j,1) & Otherwise.
\end{cases}$$  \hspace{1cm} (5)

Given the assumption that $S^{m}_{i,k} < S^{m}_{i,k} \prime$ for $k = 1, 2, \cdots, K$, we have $S^{w}_{m}(1,j) < S^{w}_{m}(1,j) \prime$. Then we could get $A$ is a positive define matrix easily using the definition of positive define matrixx. Regarding $A$ and $B_{0}$ are both semi-positive define matrix, we have following statement using \textbf{Oppenheim’s inequality}:

$$\det(A \circ B_{0}) = \det(B_{1}) \geq \det(B_{0}) \prod_{i}^{K} A(i,i) = \det(B_{0}),$$  \hspace{1cm} (6)

where $A \circ B_{0}$ is the \textbf{Hadamard product} (element-wise product) of $A$ and $B_{0}$. Besides, $B_{1}$ is also a semi-positive define matrix according to \textbf{Schur product theorem}.

For $i = 1, 2, \cdots, K - 1$, it is easy to construct $A$ with similar definition like above and get the statement that $\det(B_{i}) \leq \det(B_{i+1})$. Now, combining the chain of inequality, we have:

$$\det(S^{w}_{m}) = \det(B_{K-1}) \geq \det(B_{0}) = \det(S^{w}_{m} \prime).$$  \hspace{1cm} (7)

Using Eq. (2)(7), the property holds easily.

\section*{A-II: Supernet Structure and Search Space}

In this section we give the detailed supernet structer and space of the new dimension \textit{Splint Point}.

<table>
<thead>
<tr>
<th>Input Shape</th>
<th>Operators</th>
<th>Channels</th>
<th>Repeat</th>
<th>Stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>$224^2 \times 3$</td>
<td>$3 \times 3$ Conv</td>
<td>16</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$112^2 \times 16$</td>
<td>$3 \times 3$ Depthwise Separable Conv</td>
<td>16</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$56^2 \times 16$</td>
<td>MBConv / SkipConnect</td>
<td>24</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$28^2 \times 24$</td>
<td>MBConv / SkipConnect</td>
<td>40</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$14^2 \times 40$</td>
<td>MBConv / SkipConnect</td>
<td>80</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>$14^2 \times 80$</td>
<td>MBConv / SkipConnect</td>
<td>112</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$7^2 \times 112$</td>
<td>MBConv / SkipConnect</td>
<td>160</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>$7^2 \times 160$</td>
<td>$1 \times 1$ Conv</td>
<td>960</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$7^2 \times 960$</td>
<td>Global Avg. Pooling</td>
<td>960</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>960</td>
<td>$1 \times 1$ Conv</td>
<td>1,280</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1,280</td>
<td>Fully Connect</td>
<td>1,000</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1. The structure of the supernet. The "MBConv" contains 6 inverted bottleneck residual block MBConv [1] (kernel sizes of {3,5,7}) with the squeeze and excitation module (expansion rates {4,6}). The "Repeat" represents the maximum number of repeated blocks in a group. The "Stride" indicates the convolutional stride of the first block in each repeated group. (9, 20, 1) means space starts from 9 to 20 with a step of 1.
Appendix B

In appendix B, we show the detailed evolution algorithm, with the detailed algorithm of $K$-path evolution search below. Specific steps of Crossover, Mutation are presented in Section 3.4.

**Algorithm 1 K-Path Evolution Search**

**Input:**
- Shrunk search space $\tilde{S}$, weights $W_{\tilde{S}}$, population size $P$, resources constraints $C$, number of generation iteration $T$, validation dataset $D_{val}$, training dataset $D_{train}$, Mutation probability of split point $P_s$, Mutation probability of layer combination $P_m$.

**Output:** The most promising ensemble architecture $A^*$.

1. $G(0) :=$ Random sample $P$ ensemble architectures $\{A_1, A_2, \cdots, A_P\}$ from $\tilde{S}$ with constrain $C$;
2. while search step $t \in (0, T)$ do
3.   while $A_i \in G(t)$ do
4.     Recalculate the statistics of BN on $D_{train}$;
5.     Obtain the accuracy of $\Phi(\cdot; A_i, W_{\tilde{S}})$ on $D_{val}$;
6.   end while
7.   $G_{topk} :=$ the Top $K$ candidates by accuracy order;
8.   $G_{crossover} :=$ Crossover($G_{topk}$, $\tilde{S}$, $C$);
9.   $G_{mutation} :=$ Mutation($G_{topk}$, $P_s$, $P_m$, $\tilde{S}$, $C$);
10.  $G(t+1) = G_{crossover} \cup G_{mutation}$
11. end while

**References**