EMT-NAS: Transferring architectural knowledge between tasks from different datasets
(Supplementary Material)

Peng Liao¹, Yaochu Jin¹,²*, Wenli Du¹*

¹ Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, ECUST, China
² Faculty of Technology, Bielefeld University, Germany
pengliao@mail.ecust.edu.cn, yaochu.jin@uni-bielefeld.de, wldu@ecust.edu.cn

A. Search Space

Similar to [5], the search space defined in this work is shown in Fig. A. And the optional operations in each block are listed in Table A.

<table>
<thead>
<tr>
<th>Operation Type</th>
<th>Name</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity mapping</td>
<td>Identity</td>
<td>0</td>
</tr>
<tr>
<td>Average pooling</td>
<td>AVG</td>
<td>1</td>
</tr>
<tr>
<td>Max pooling</td>
<td>MAX</td>
<td>2</td>
</tr>
<tr>
<td>Depthwise Separable Convolution</td>
<td>DW3</td>
<td>3</td>
</tr>
<tr>
<td>Depthwise Separable Convolution</td>
<td>DW5</td>
<td>4</td>
</tr>
<tr>
<td>Dilated Convolution</td>
<td>DC3</td>
<td>6</td>
</tr>
<tr>
<td>Dilated Convolution</td>
<td>DC5</td>
<td>7</td>
</tr>
<tr>
<td>Dilated Convolution</td>
<td>DC7</td>
<td>8</td>
</tr>
</tbody>
</table>

B. Pseudo code

Algorithm 1 The pseudo code of EMT-NAS

Input: The number of tasks $N$, population size of each task $K$, maximum number of generations $T$, training dataset $D_{train} = \{D_{train}^1, ..., D_{train}^N\}$ and validation dataset $D_{vali} = \{D_{vali}^1, ..., D_{vali}^N\}$.

Output: The best network architecture for each task.

1: $t = 1$
2: $P_1 \leftarrow \text{Generate an initial population with } N \times K$ and assign the same $\tau$ to each individual on the same task by Algorithm 2
3: $P_2 \leftarrow \text{Train individuals of } P_1 \text{ on } D_{train}$ by Algorithm 3
4: $\text{Evaluate the fitness of trained individuals in } P_2 \text{ on } D_{vali} \text{ by Algorithm 4}$
5: while $t < T$ do
6: $t = t + 1$
7: $\text{Generate offspring population } O_t \text{ and assign the skill factor } \tau \text{ of each offspring by Algorithm 5}$
8: $P_3, O_t \leftarrow \text{Train individuals of } P_t \text{ and } O_t \text{ on } D_{train}$ by Algorithm 3
9: $\text{Evaluate the fitness of trained individuals in } O_t \text{ on } D_{vali} \text{ by Algorithm 4}$
10: $R_t = P_t \cup O_t$
11: $P_{t+1} \leftarrow \text{Select top } K \text{ individuals of every task from } R_t$
12: $(P_{vali})_{t+1} \leftarrow \text{In } P_{t+1}, \text{ the individuals with the highest fitness in each task are evaluated on the validation dataset for the corresponding task}$
13: end while
14: Output the best individuals in $P_{vali}$ of each task and decode them into the corresponding network architecture

*Corresponding author.
C. Hyperparameter Settings

C.1. MedMNIST

MedMNIST [4] is a collection of 10 pre-processed medical datasets, including X-ray, OCT, ultrasound, computed tomography (CT), pathological section, and dermatoscopy, for colorectal cancer, retinal diseases, breast diseases, and liver tumors. We selected the PathMNIST validation dataset for the corresponding task as a candidate for the best individual under that task (Line 12). The steps from Line 6 to Line 12 are repeated for $T - 1$ generations before the best individual (network architecture) for each task is outputted (Line 14).

**Algorithm 2** Population Initialization

**Input:** Number of tasks $N$, population size of each task $K$, operation space $S$, and position space $X$

**Output:** Initial population $P_1$

1: $P_1 = \{\}$
2: for $n \leftarrow 1$ to $N$ do
3:   $\tau = n$
4:   for $k \leftarrow 1$ to $K$ do
5:     individual = \{}
6:   for $e \leftarrow 1$ to 2 do
7:     cellcode = \{}
8:   for $h \leftarrow 1$ to 5 do
9:     blockcode = \{}
10:    $x_1, x_2 \leftarrow$ Randomly select two positions in $X$
11:    $s_1, s_2 \leftarrow$ Randomly select two operations in $S$
12:    blockcode = blockcode $\cup$ blockcode
13:   end for
14:   individual = individual $\cup$ cellcode
15: end for
16: $P_1 \leftarrow P_1 \cup$ (individual $\cup$ $\tau$)
17: end for
18: Output population $P_1$

Figure A. Overview of the search space. (a) The search space consists of a convolution layer, two types of searchable cells and a fully connected layer. Normal cell can be stacked $R$ times, reduction cell only once. Each cell consists of two inputs (the output and input of the previous block respectively), five blocks, and one output. And each block has two inputs that are separately connected to the operation, which adds up to one output, as shown in (b). Hence, each block code has four integer bits. The cell code consists of five blocks, and its corresponding cell structure is shown in (c) and (d), respectively.

**Algorithm 3** Sampled Training

**Input:** Number of tasks $N$, input population $I$, training dataset $D_{train} = \{D_{train}^1, ..., D_{train}^N\}$

**Output:** Trained population $I$

1: According to the skill factor $\tau$, the input population $I$ is divided into $N$ sub-population of the same task $I_1, ..., I_N$
2: for $n \leftarrow 1$ to $N$ do
3:   for each batch $\mathcal{D}$ in $D_{train}^n$ do
4:     Sample an individual $\mathcal{F}$ from $I_n$ by the binary tournament selection
5:     $\mathbf{net}, \omega \leftarrow$ Decode individual $\mathcal{F}$ to activate the corresponding network
6:     $\nabla \omega \leftarrow$ Compute the gradient
7:     $\omega \leftarrow \omega - \tau \nabla \omega$
8:   end for
9: end for
10: Output the trained population $I$

**Algorithm 4** Sampled Evaluating

**Input:** Number of tasks $N$, input population $I$, current generation $t$, training dataset $D_{valid} = \{D_{valid}^1, ..., D_{valid}^N\}$

**Output:** Evaluated population $I$

1: According to skill factor $\tau$, the input population $I$ is divided into $N$ sub-population of the same task $I_1, ..., I_N$
2: for $n \leftarrow 1$ to $N$ do
3:   for each $\mathcal{F}$ in $I_n$ do
4:     $\mathbf{net}, \omega \leftarrow$ Decode individual $\mathcal{F}$ to activate the corresponding network
5:     $\mathcal{D} \leftarrow$ A batch randomly selected from $D_{valid}^n$
6:     $acc \leftarrow net(\mathcal{D})$
7:   if $t > 1$ then
8:     $acc \leftarrow$ Compute by equation 2 in the main paper
9: end if
10: end for
11: end for
12: Output the evaluated population $I$
Algorithm 5 Implicit Knowledge Transfer

\begin{verbatim}
Input: Parent population \( P_i \), crossover probability of individuals from different tasks \( RMP \), number of tasks \( N \), and population size of each task \( K \)
Output: the offspring population \( O_i \)

1: \( O_i := \{\} \)
2: while The number of offspring of each task does not reach \( K \) do
3: \( p_1, p_2 \leftarrow \text{Select two individuals from } P_i \)
4: if \( \tau_{p_1} = \tau_{p_2} \) or \( \text{rand} < RMP \) then
5: \( q_1, q_2 \leftarrow \text{crossover}(p_1, p_2) \)
6: \( o_1 \leftarrow \text{mutate}(q_1) \)
7: \( o_2 \leftarrow \text{mutate}(q_2) \)
8: if \( \text{rand} < 0.5 \) then
9: \( (\tau_{q_1}, \tau_{q_2}) \leftarrow (\tau_{p_1}, \tau_{p_2}) \)
10: else
11: \( (\tau_{q_1}, \tau_{q_2}) \leftarrow (\tau_{q_1}, \tau_{q_2}) \)
12: end if
13: else
14: \( o_1 \leftarrow \text{mutate}(p_1) \)
15: \( o_2 \leftarrow \text{mutate}(p_2) \)
16: \( (\tau_{o_1}, \tau_{o_2}) \leftarrow (\tau_{p_1}, \tau_{p_2}) \)
17: end if
18: if the number of offspring of each task does not reach \( K \) then
19: The newly generated individual is added into the offspring population of the corresponding task until the population size of the task reaches \( K \)
20: end if
21: end while
22: Output the offspring population \( O_i \)
\end{verbatim}

are listed in Table 1 in the main paper. In the search phase, \( R \) in is set to 1, meaning that normal cell is stacked only once in each network architecture and the initial number of channels is 20. In the retraining stage, \( R \) is set to 2 (normal cells in the best network architecture is increased to 2), the initial channel number is set to 48, and all models use an auxiliary classifier located 2/3 of the way up the network and the loss weight of the auxiliary classifier is 0.4, for a total of 600 epochs.

C.3. ImageNet

ImageNet [3] contains a thousand classification datasets, of which the training set has 1,281,167 images and the validation set has 50,000 images, making it one of the most challenging datasets in image classification.

For re-training on ImageNet, we follow the previous work [5], three convolution layers with 3x3 convolution kernels are added to the convolution layer of the model found on the CIFAR-10 and CIFAR-100 datasets. We adopt pre-processing with image size 224x224 [5], 300 epochs, batch size of 512, SGD optimizer with linearly decayed learning rate initialized as 0.05, momentum of 0.9, and weight decay of 1 \( \times 10^{-4} \). Learning rate warming starts [2] in the initial 5 epochs, it then decays every 100 epochs at a rate of 0.1. A Dropout layer [5] of rate 0.05 is added before the last linear layer.

D. Two Baselines for Joint Training

We have compared the two baselines, JT-S (combine data from all tasks and train them together, but share the same architecture in the NAS.), and JT-D (combine data from all tasks and jointly train them, and each task has individual architecture, but the supernet weight is shared across all tasks) on C-10 and C-100. The hyperparameters settings for JT-S and JT-D are the same as for EMT-NAS and their results are listed in Table C. From these results, we see that EMT-NAS outperformed both JT-S and JT-D in terms of validation accuracy of the best individual at the end of the search and its test accuracy after retraining. This might be attributed to the fact that jointly training the weights of the supernet may exacerbate catastrophic forgetting, making the algorithm unable to distinguish between different architectures. This effect will be more serious for JT-S, where the architecture is the same. In Fig. B, we note that the variance of JT-S is larger than that of JT-D, further confirming our hypothesis.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image Size</th>
<th>Data Modality</th>
<th>Tasks (Classes)</th>
<th>( D_{\text{train}} )</th>
<th>( D_{\text{valid}} )</th>
<th>( D_{\text{test}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathMNIST</td>
<td>3x28x28 Pathology</td>
<td>Multi-Class(9)</td>
<td>8,999</td>
<td>5,589</td>
<td>7,180</td>
<td></td>
</tr>
<tr>
<td>OrganMNIST_Axial</td>
<td>1x28x28 Abdominal CT</td>
<td>Multi-Class(11)</td>
<td>13,451</td>
<td>6,491</td>
<td>17,778</td>
<td></td>
</tr>
<tr>
<td>OrganMNIST_Coronar</td>
<td>1x28x28 Abdominal CT</td>
<td>Multi-Class(11)</td>
<td>13,000</td>
<td>2,392</td>
<td>8,268</td>
<td></td>
</tr>
<tr>
<td>OrganMNIST_Sagittal</td>
<td>1x28x28 Abdominal CT</td>
<td>Multi-Class(11)</td>
<td>13,940</td>
<td>2,452</td>
<td>8,829</td>
<td></td>
</tr>
</tbody>
</table>

C.2. CIFAR-10 and CIFAR-100

CIFAR-10 and CIFAR-100 [1] are datasets of color images of 10 and 100 classes, respectively, with a size of 32x32. The training set contains 50,000 images and the test set has 10,000 images. We divide the training set into a new training set and a validation set on a 4-to-1 ratio. The new training and validation sets are used for the EMT-NAS search phase, while the test set is used for the performance evaluation of the network architecture.

Similar to [5], the main parameter settings of EMT-NAS
Table C. Comparison of two baselines JT-S and JT-D on C-10 and C-100. * indicates that the average validation results of JT-S on C-100 is the same as on C-10.

<table>
<thead>
<tr>
<th>Model</th>
<th>GPU days</th>
<th>Validation Task 1 (C-10)</th>
<th>Test Task 1 (C-10)</th>
<th>Params (M)</th>
<th>Validation Task 2 (C-100)</th>
<th>Test Task 2 (C-100)</th>
<th>Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JT-S (Baseline)</td>
<td>0.50</td>
<td>88.76±2.05 69.73±0.15 1.89</td>
<td>0.37</td>
<td>JT-D (Baseline)</td>
<td>0.46</td>
<td>85.14±1.04 96.22±0.22 1.90</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Figure B. Comparison of best individual fitness during the search process on C-10 and -100. JT-S fitness is averaged on C-10 and -100, so the values are not comparable with ST-D and EMT-NAS.

Table D. Comparison of EMT-NAS when the population size is set to 10, 20, 30, and 40.

<table>
<thead>
<tr>
<th>POP</th>
<th>Task 1 Validation Task 1 (C-10)</th>
<th>Test Task 1 (C-10)</th>
<th>POP</th>
<th>Task 2 Validation Task 2 (C-100)</th>
<th>Test Task 2 (C-100)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>C-10 88.9±1.18 96.41±0.21 1.89</td>
<td>0.42</td>
<td>C-10 67.83±2.10 1.14±0.18 0.08</td>
<td>0.02</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.06</td>
</tr>
<tr>
<td>20</td>
<td>C-10 88.76±2.05 96.73±0.15 1.89</td>
<td>0.37</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.08</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.10</td>
</tr>
<tr>
<td>30</td>
<td>C-10 87.48±0.17 96.9±0.18 1.89</td>
<td>0.37</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.10</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.15</td>
</tr>
<tr>
<td>40</td>
<td>C-10 88.49±1.28 96.38±0.27 1.89</td>
<td>0.42</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.10</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table E. Comparison of EMT-NAS when the crossover probability is set to 1.00, 0.95, and 0.90.

<table>
<thead>
<tr>
<th>CP</th>
<th>Task 1 Validation Task 1 (C-10)</th>
<th>Test Task 1 (C-10)</th>
<th>CP</th>
<th>Task 2 Validation Task 2 (C-100)</th>
<th>Test Task 2 (C-100)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>C-10 88.76±2.05 96.73±0.15 1.89</td>
<td>0.37</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.02</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.04</td>
</tr>
<tr>
<td>0.95</td>
<td>C-10 88.63±0.20 96.39±0.20 1.89</td>
<td>0.37</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.06</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.08</td>
</tr>
<tr>
<td>0.90</td>
<td>C-10 86.54±0.17 96.44±0.17 1.89</td>
<td>0.42</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.08</td>
<td>C-10 61.69±2.15 81.86±0.19 2.30</td>
<td>0.10</td>
</tr>
</tbody>
</table>

E. Analysis of Parameters

Population Size: We set the population size POP = 10, 20, 30, and 40 and the statistical results are given in Table D. The overall runtime of EMT-NAS increases as POP slightly increases mainly because the number of networks to be assessed on the validation set increases. The performance on the test dataset achieves the best when the population size is set to 20.

Crossover Probability: We set the crossover probability CP = 1.00, 0.95, 0.90 and the statistical results are presented in Table E. From these results, we can conclude that the best performance was achieved on both the validation and test sets, when CP = 1.00. Hence, the crossover probability is set to 1.00.

Mutation Probability: We set the mutation probability MP = 0.02, 0.04, 0.06, 0.08, 0.10 and the statistical results are given in Table F. We found that as MP increases, the validation accuracy on both tasks decrease. However, the test accuracy varies. On the one hand, these results indicate that the mutation probability should be kept small, which is consistent with the known knowledge in evolutionary optimization. On the other hand, the increase in mutation probability may be attributed to the fact that higher mutation rate makes the sampled validation less accuracy. Hence, the mutation probability is set to 0.02.

Generation Number: We set the generation number GN = 100, 200, 300, and 400 and the statistical result are given in Table G. We found that as GN increases, the validation accuracy and test accuracy increased and then decreased for both tasks. And, as GN increases, the running time of the EMT-NAS increases. Hence, the generation number is set to 300.

F. Network Architectures Visualisation

References


Figure C. Normal and reduction Cells discovered by EMT-NAS on Task 1 (CIFAR-10)

Figure D. Normal and reduction Cells discovered by EMT-NAS on Task 2 (CIFAR-10)