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

Peng Liao[1], Yaochu Jin[1,2]*, Wenli Du[1]*
[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 A. Network operation coding space

Operation Type	Kernel Size	Shot Name	Code
Identity mapping	–	Identity	0
Average pooling	3	AVG	1
Max pooling	3	MAX	2
Depthwise Separable Convolution	3	DW3	3
Depthwise Separable Convolution	5	DW5	4
Depthwise Separable Convolution	7	DW7	5
Dilated Convolution	3	DC3	6
Dilated Convolution	5	DC5	7
Dilated Convolution	7	DC7	8

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 $D_{train} = \left\{D^1_{train}, ..., D^N_{train}\right\}$ and validation dataset $D_{valid} = \left\{D^1_{valid}, ..., D^N_{valid}\right\}$

Output: The best network architecture for each task

1: $t = 1$
2: $P_1 \leftarrow$ Generate an initial population with $N * K$ and assign the same τ to each individual on the same task by Algorithm 2
3: $P_2 \leftarrow$ Train individuals of P_1 on D_{train} by Algorithm 3
4: Evaluate the fitness of trained individuals in P_2 on D_{valid} by Algorithm 4
5: **while** $t < T$ **do**
6: $\quad t = t + 1$;
7: $\quad$ Generate offspring population O_t and assign the skill factor τ of each offspring by Algorithm 5
8: $\quad P_t, O_t \leftarrow$ Train individuals of P_t and O_t on D_{train} by Algorithm 3
9: $\quad$ Evaluate the fitness of trained individuals in O_t on D_{valid} by Algorithm 4
10: $\quad R_t = P_t \cup O_t$
11: $\quad P_{t+1} \leftarrow$ Select top K individuals of every task from R_t;
12: $\quad (P_{best})_t \leftarrow$ In P_{t+1}, 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_{best} of each task and decode them into the corresponding network architecture

The pseudo code of EMT-NAS is given in Algorithm 1. The inputs of the algorithm include the number of tasks N, the population size of each task K, the maximum number of generations T, the training dataset $D_{train} = \left\{D^1_{train}, ..., D^N_{train}\right\}$ (a collection of training sets for all tasks) and the validation dataset $D_{valid} = \left\{D^1_{valid}, ..., D^N_{valid}\right\}$ (a collection of validation sets for all tasks). The algorithm starts with population initialization (Line 2), in which K individuals (network architectures) are randomly generated for each task together with a skill factor τ predefined for each task, referring to Algorithm 2. Then, sampled training of the parents is performed to update the weights of the individuals on the data of the corresponding task using Algorithm 3 (Line 3). As stated above, in the sampled training, an individual having the corresponding skill factor in the population is randomly chosen for each mini-batch of the data to reduce the training time. After training, sampled evaluating is carried out for each individual that evaluates its fitness (classification accuracy) on one mini-batch of the validation data of the corresponding task indicated by its skill factor, referring to Algorithm 4 (Line 4). Next, two individuals are selected from the parent population to generate two offspring by means of crossover or mutation at a probability (Line 7). Details of the crossover and mutation operators are given in Algorithm 5. Then, sampled training of the individuals in the parent and offspring populations are trained on the training data of their corresponding task, referring to Algorithm 3 (Line 8), followed by sampled evaluating of the individuals in the offspring population on the validation data of the corresponding task, referring to Algorithm 4 (Line 9). Then, environmental selection is conducted to select the top K individuals for each task as the parents of the next generation from a combination of parent and offspring individuals by sorting the individuals in an descending order (Line 11). Then, in the parent population of the next generation, the individual with the highest fitness in each task is evaluated on the val-

*Corresponding author.

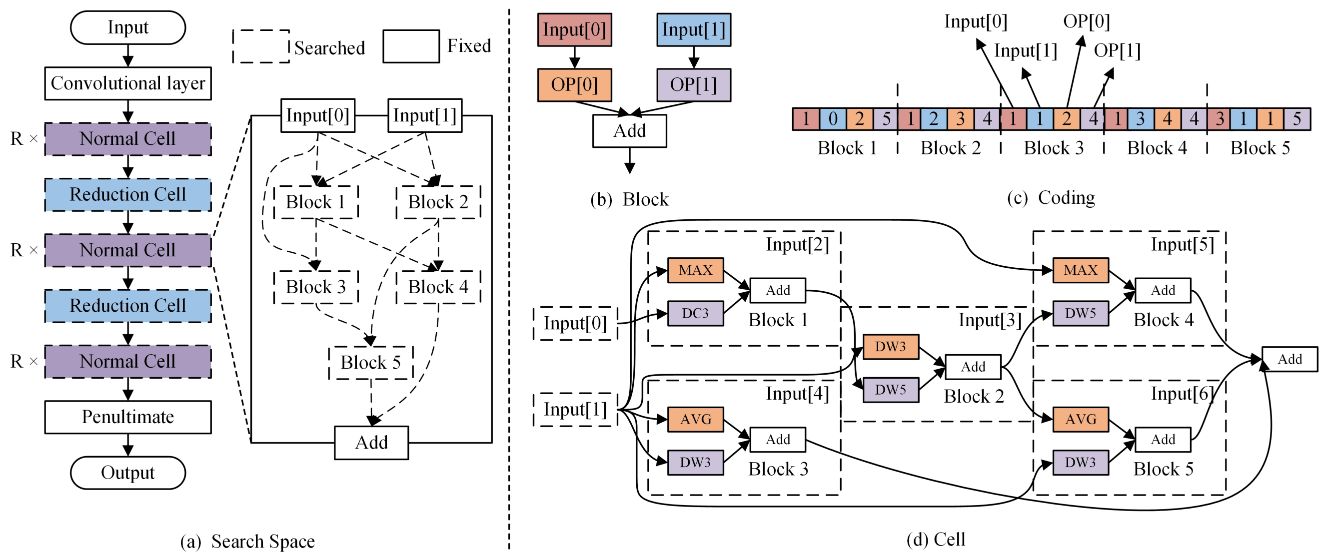


Figure A. Overview of the search space. (a) The search space consists of a convolution layer, two types of searchable cells and a fully connected layer. Normal cell can be stacked R times, reduction cell only once. Each cell consists of two inputs (the output and input of the previous block respectively), 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.

idation 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
$P_1 = \{\}$
for $n \leftarrow 1$ to N **do**
  $\tau = n$
  for $k \leftarrow 1$ to K **do**
    $individual = \{\}$
    for $e \leftarrow 1$ to 2 **do**
      $cellcode = \{\}$
      for $h \leftarrow 1$ to 5 **do**
        $blockcode = \{\}$
        $x_1, x_2 \leftarrow$ Randomly select two positions in X
        $s_1, s_2 \leftarrow$ Randomly select two operations in S
        $blockcode \leftarrow s_1, s_2 \cup x_1, x_2$
        $cellcode \leftarrow cellcode \cup blockcode$
      end for
      $individual = individual \cup cellcode$
    end for
    $P_1 \leftarrow P_1 \cup (individual \cup \tau)$
  end for
end for
Output population P_1

Algorithm 3 Sampled Training

Input: Number of tasks N, input population I, training dataset $D_{train} = \left\{D_{train}^1, \ldots, D_{train}^N\right\}$
Output: Trained population I
According to the skill factor τ, the input population I is divided into N sub-population of the same task $I_1, \ldots, I_N$
for $n \leftarrow 1$ to N **do**
  for each batch $\mathscr{D}$ in D_{train}^n **do**
    Sample an individual $\mathscr{I}$ from I_n by the binary tournament selection
    $net, \omega \leftarrow$ Decode individual $\mathscr{I}$ to activate the corresponding network
    $\nabla\omega \leftarrow$ Compute the gradient
    $\omega \leftarrow \omega - r\nabla\omega$
  end for
end for
Output the trained population I

Algorithm 4 Sampled Evaluating

Input: Number of tasks N, input population I, current generation t, training dataset $D_{valid} = \left\{D_{valid}^1, \ldots, D_{valid}^N\right\}$
Output: Evaluated population I
According to skill factor τ, the input population I is divided into N sub-population of the same task $I_1, \ldots, I_N$
for $n \leftarrow 1$ to N **do**
  for each $\mathscr{I}$ in I_n **do**
    $net, \omega \leftarrow$ Decode individual $\mathscr{I}$ to activate the corresponding network
    $\mathscr{D} \leftarrow$ A batch randomly selected from D_{valid}^n
    $acc \leftarrow net(\mathscr{D})$
    if $t > 1$ **then**
      $acc \leftarrow$ Compute by equation 2 in the main paper
    end if
  end for
end for
Output the evaluated population I

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

Table B. Medical image dataset

Dataset	Image Size	Data Modality	Tasks (Classes)	D_{train}	D_{valid}	D_{test}
PathMNIST	3×28×28	Pathology	Multi-Class(9)	89,996	10,004	7,180
OrganMNIST_Axial	1×28×28	Abdominal CT	Multi-Class(11)	34,581	6,491	17,778
OragnMNIST_Coronal	1×28×28	Abdominal CT	Multi-Class(11)	13,000	2,392	8,268
OrganMNIST_Sagittal	1×28×28	Abdominal CT	Multi-Class(11)	13,940	2,452	8,829

Algorithm 5 Implicit Knowledge Transfer

Input: Parent population P_t, crossover probability of individuals from different tasks RMP, number of tasks N, and population size of each task K
Output: the offspring population O_t

$O_t = \{\}$
while The number of offspring of each task does not reach K **do**
  $p_1, p_2 \leftarrow$ Select two individuals from P_t
  if $\tau_{p_1} = \tau_{p_2}$ or $rand < RMP$ **then**
    $q_1, q_2 \leftarrow$ crossover(p_1, p_2)
    $o_1 \leftarrow$ mutate(q_1)
    $o_2 \leftarrow$ mutate(q_2)
    if $rand < 0.5$ **then**
      $(\tau_{o_1}, \tau_{o_2}) \leftarrow (\tau_{p_1}, \tau_{p_2})$
    else
      $(\tau_{o_1}, \tau_{o_2}) \leftarrow (\tau_{p_2}, \tau_{p_1})$
    end if
  else
    $o_1 \leftarrow$ mutate(p_1)
    $o_2 \leftarrow$ mutate(p_2)
    $(\tau_{o_1}, \tau_{o_2}) \leftarrow (\tau_{p_1}, \tau_{p_2})$
  end if
  if the number of offspring of each task does not reach K **then**
    The newly generated individual is added into the offspring population of the corresponding task until the population size of the task reaches K
  end if
end while
Output the offspring population O_t

dataset of colon pathology, and OrganMNIST_Axial, OragnMNIST_Coronal, and OrganMNIST_Sagittal through different processing methods in abdominal 3D CT along three axes, listed in Table B.

we use the same parameter settings recommended in [4] for a fair comparison, meaning the maximum number of generations is set to 100 for EMT-NAS, R is set to 1 (normal cells stacked once) and the initial channel number is set to 48 during the search process. Experiments on MedMNIST with a search phase only. When the search phase is over, the optimal network searched (including the network architecture and its weights) is retested for classification accuracy on the test set and then compared to the networks found by other NAS algorithms. Similar to previous work [5], other parameters are the same as those in Table 1 in the main paper.

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 32*32. 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 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.

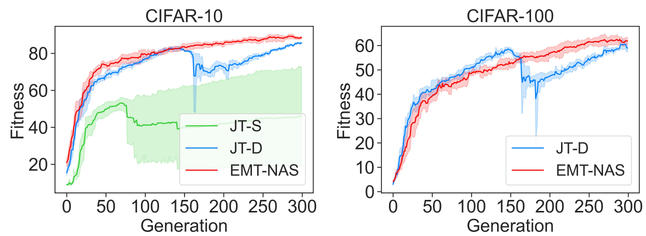
For re-training on ImageNet, we follow the previous work [5], three convolution layers with 3×3 convolution kernels are added to the convolution layer of the model found on the CIFAR-10 and CIFAR-100 datasets. we adopt pre-processing with image size 224×224 [5], 300 epochs, batch size of 512, SGD optimizer with linearly decayed learning rate initialized as 0.05, momentum of 0.9, and weight decay of 1×10^{-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 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.

Model	GPU days	Task 1 (C-10)			Task 2 (C-100)		
		Validation ACC (%)	Test ACC (%)	Params (M)	Validation ACC (%)	Test ACC (%)	Params (M)
JT-S (Baseline)	0.50	72.57±10.8	87.21±12.3	1.76	$*$	72.26±6.76	1.73
JT-D (Baseline)	0.46	85.14±1.04	96.22±0.22	1.90	60.15±1.07	79.53±0.21	1.91
EMT-NAS (Ours)	**0.42**	**88.76±1.05**	**96.73±0.15**	2.17	**61.69±2.15**	**81.86±0.10**	2.30



(a) (b)

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.

POP	Task 1	Validation ACC (%)	Test ACC (%)	Task 2	Validation ACC (%)	Test ACC (%)	Time (%) ↓
10	C-10	88.91±1.18	96.41±0.21	C-100	61.28±3.20	81.14±0.18	+0.0
20	C-10	**88.76±1.05**	**96.73±0.15**	C-100	**61.69±2.01**	**81.86±0.10**	**+8.2**
30	C-10	87.48±0.73	96.59±0.18	C-100	58.96±2.66	81.07±0.46	+13.3
40	C-10	86.49±1.28	96.38±0.27	C-100	57.74±2.03	80.85±0.52	+18.8

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

CP	Task 1	Validation ACC (%)	Test ACC (%)	Task 2	Validation ACC (%)	Test ACC (%)
1.00	C-10	**88.76±1.05**	**96.73±0.15**	C-100	**61.69±2.15**	**81.86±0.10**
0.95	C-10	88.63±0.70	96.39±0.20	C-100	61.63±0.89	81.25±0.58
0.90	C-10	88.10±0.87	96.44±0.12	C-100	60.51±1.97	81.02±0.47

E. Analysis of Parameters

Population Size: we set the population size $POP = 10, 20, 30, 40$ 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 $CR = 1.00$. 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 result 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 indicates 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 $GN = 100, 200, 300, 400$ and the statistical result are given in Table G. We found that as GN increased, 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.

Table F. Comparison of EMT-NAS when the mutation probability is set to 0.02, 0.04, 0.06, 0.08, and 0.10.

MP	Task 1	Validation ACC (%)	Test ACC (%)	Task 2	Validation ACC (%)	Test ACC (%)
0.02	C-10	**88.76±1.05**	**96.73±0.15**	C-100	**61.69±2.15**	**81.86±0.10**
0.04	C-10	86.29±1.40	96.45±0.20	C-100	57.46±3.79	80.97±0.43
0.06	C-10	85.81±1.23	96.40±0.06	C-100	56.27±1.48	81.31±0.29
0.08	C-10	84.03±0.78	96.52±0.28	C-100	55.12±0.64	80.90±1.12
0.10	C-10	83.31±2.07	96.36±0.24	C-100	53.56±1.28	80.45±0.43

Table G. Comparison of EMT-NAS when the generation number is set to 100, 200, 300, and 400.

GN	Task 1	Validation ACC (%)	Test ACC (%)	Task 2	Validation ACC (%)	Test ACC (%)
100	C-10	82.44±1.19	96.33±0.19	C-100	51.42±2.44	80.26±1.05
200	C-10	86.45±0.84	96.49±0.16	C-100	58.87±0.74	80.68±0.25
300	C-10	**88.76±1.05**	**96.73±0.15**	C-100	**61.69±2.15**	**81.86±0.10**
400	C-10	87.35±1.19	96.62±0.08	C-100	61.15±2.29	81.03±0.55

F. Network Architectures Visualisation

References

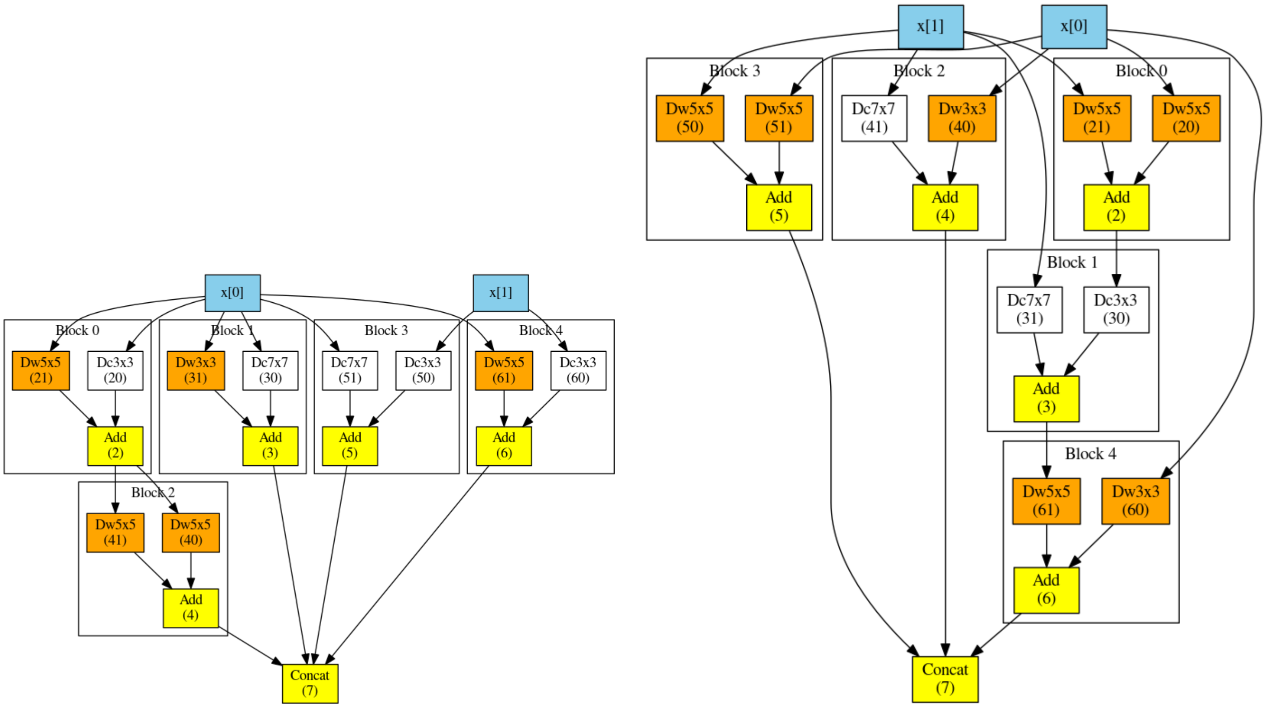
[1] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[2] Zhichao Lu, Kalyanmoy Deb, and Vishnu Naresh Boddeti. Muxconv: Information multiplexing in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12044–12053, 2020. 3

[3] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3

[4] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195. IEEE, 2021. 2, 3

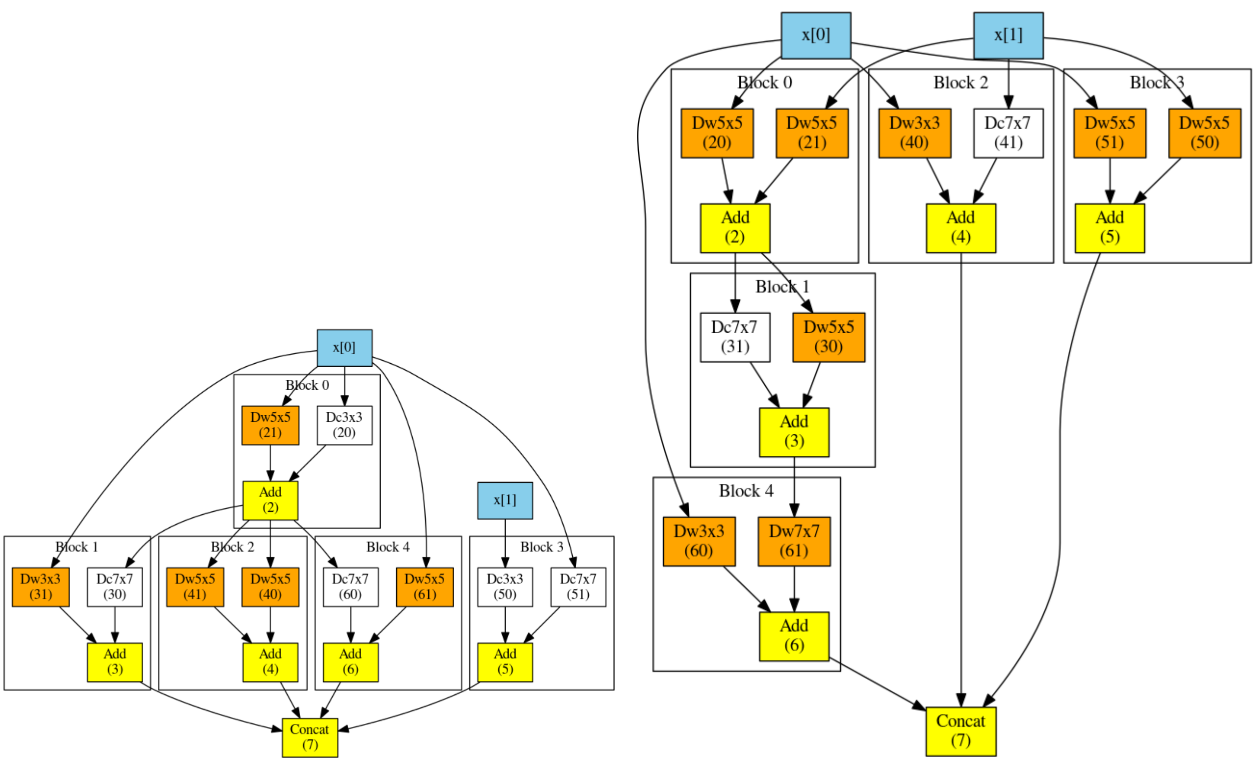
[5] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018. 1, 3



(a) Normal Cell

(b) Reduction Cell

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



(a) Normal Cell

(b) Reduction Cell

Figure D. Normal and reduction Cells discovered by EMT-NAS on Task 2 (CIFAR-100)