Medical Vision Seminar

——Wei Lou

DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation (CVPR2021)

—— Yufan He, Dong Yang, Holger Roth, Can Zhao, Daguang Xu
Johns Hopkins University, NVIDIA

Aim: Design a good 3-D medical image segmentation network automatically.

1. Background

Differentiable Neural Network Architecture Search

Richard Shin∗ & Charles Packer∗ & Dawn Song
University of California, Berkeley

(DARTS), (ICLR 2018)

1.1 Neural Network Architecture Search (NAS)

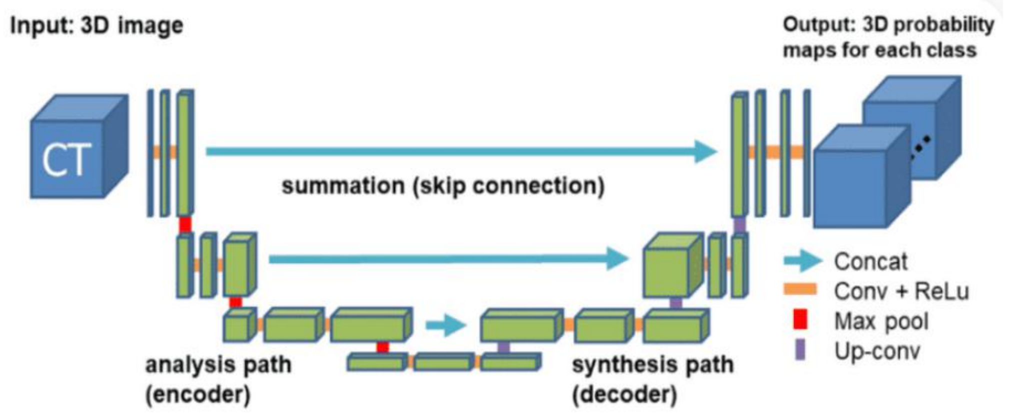


Fig. 1 An example of manual designed network architecture

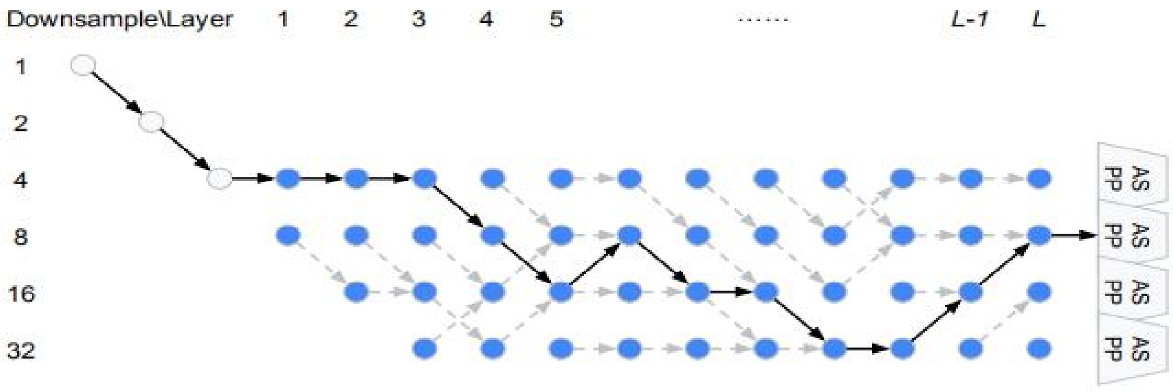


Fig. 2 An example of searched network architecture

1.2 Three key components of NAS (Search Space; Search Strategy; Evaluation Metrics)

- **Search Space**: Topology structure; Layer numbers; Connections; Operations (conv, pooling, skip…) and parameters (stride, channels, height, width, activation functions…);

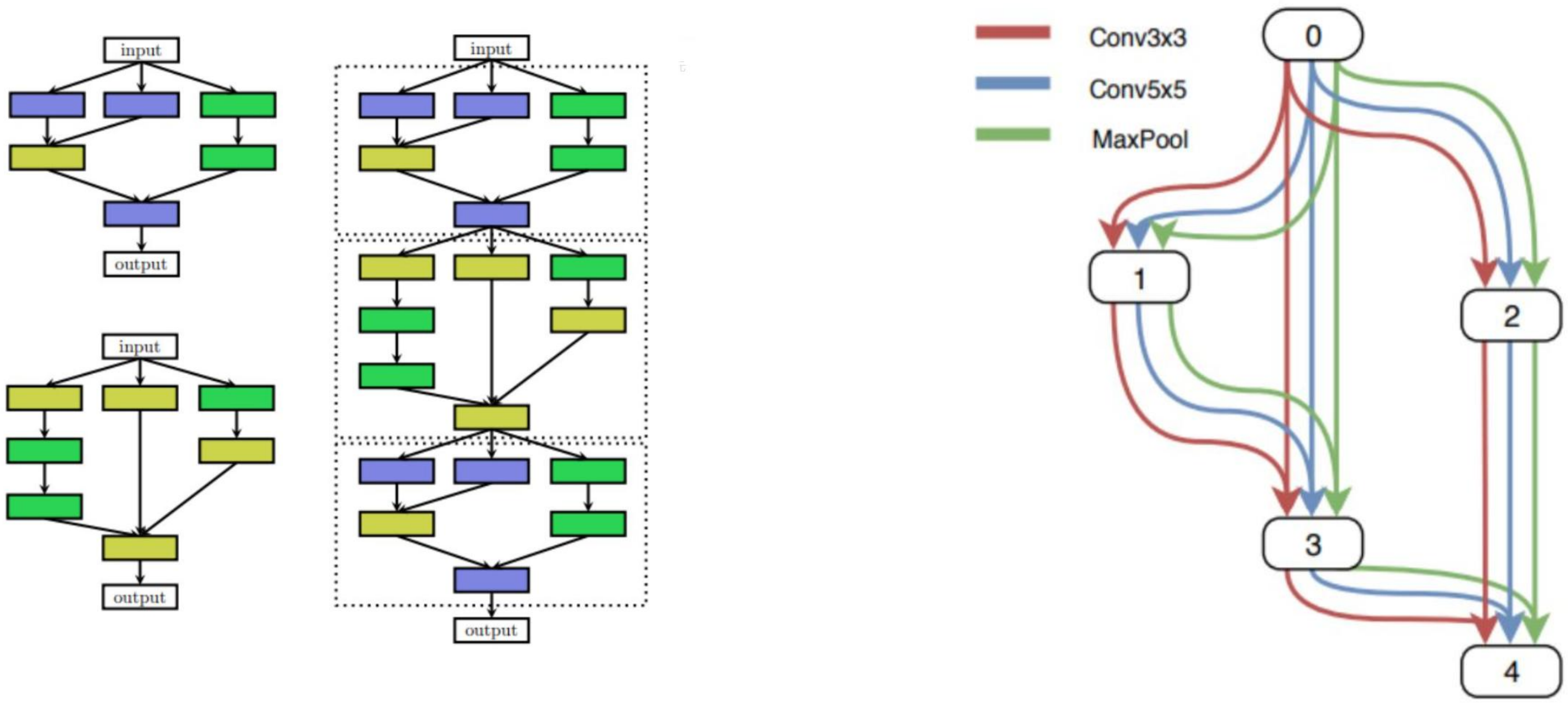


Fig.3 Search Space. (Left) Topology; (Right) Operations and connections.

1.2 Three key components of NAS (Search Space; Search Strategy; Evaluation Metrics)

- **Search Strategy**: Reinforcement Learning; Random Search; Genetic Algorithm; **Differentiable Search Algorithm**.

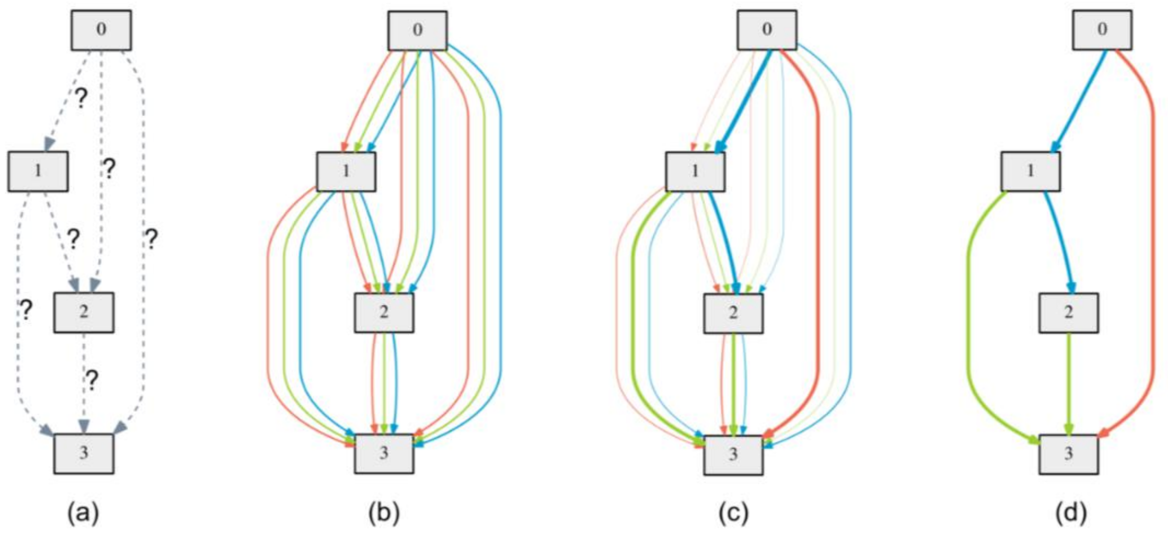


Fig.4 **Differentiable Neural Network Architecture Search (DARTS), (ICLR 2018)**

Definition of search space:
From node i to node j, o represents different operations, x represents features.

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)})$$

Continuous relaxation:
Define continuous weights α_i for each operation o_i.
Relax discrete operations to continuous using Softmax activation function:

$$\bar{o}_{(i,j)}(x) = \sum_{o \in O} \frac{exp(\alpha_{o,(i,j)})}{\sum_{o \in O} exp(\alpha_{o,(i,j)})} o(x)$$

$$\alpha = [\alpha_{1(i,j)} \dots \alpha_{|O|,(i,j)}]$$

Bi-level optimization problem:

$$min_\alpha L_{val}(w^*(\alpha), \alpha)$$
$$s.t \quad w^*(\alpha) = argmin_w L_{train}(w, \alpha)$$

- **Evaluation Metrics**: Performance; Efficiency…

1.3 Conclusion: DARTS training pipeline

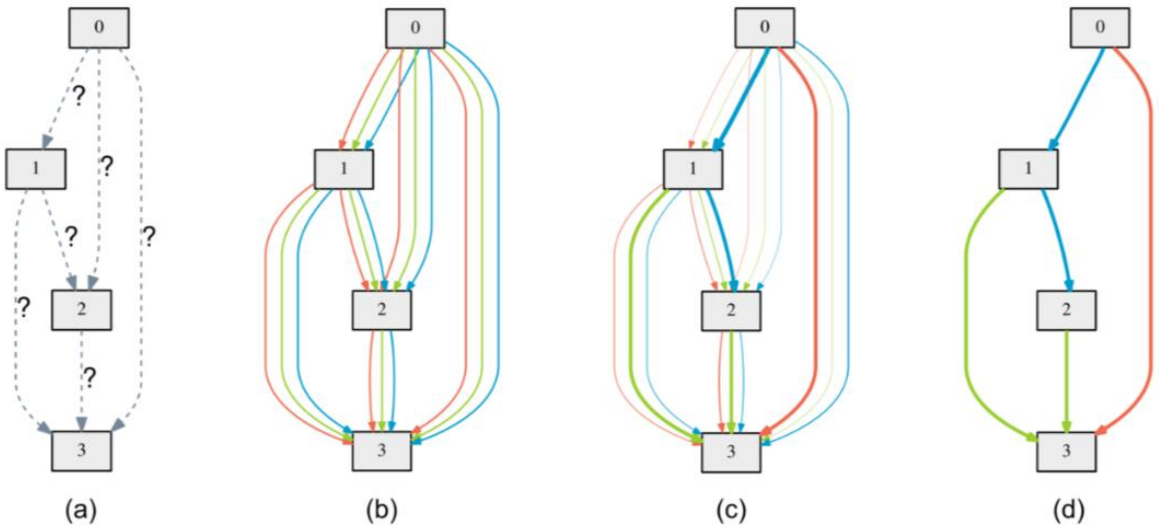


Fig.4 Typical NAS training process

1. Define a search space
2. Continuous relaxation
3. Optimization
4. Operation selection / Discretization

2. Related Work

Auto-DeepLab:
Hierarchical Neural Architecture Search for Semantic Image Segmentation

Chenxi Liu,Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua,
Alan Yuille, Li Fei-Fei
Johns Hopkins University, Google, Stanford University

2.1 Auto-DeepLab (CVPR 2019)

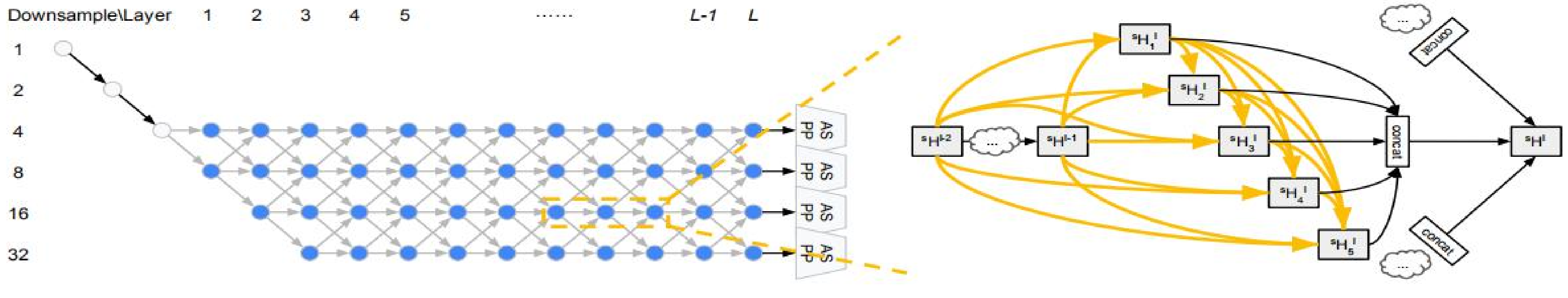


Fig.5 Search Space. (Right) Outer network level; (Left) Inner cell level

Key Contribution:

1. Two level hierarchical architecture search space (inner cell level / outer network level)：

Inner cell level: Operations inside convolutional module.

Outer network level: Topology path among cells.

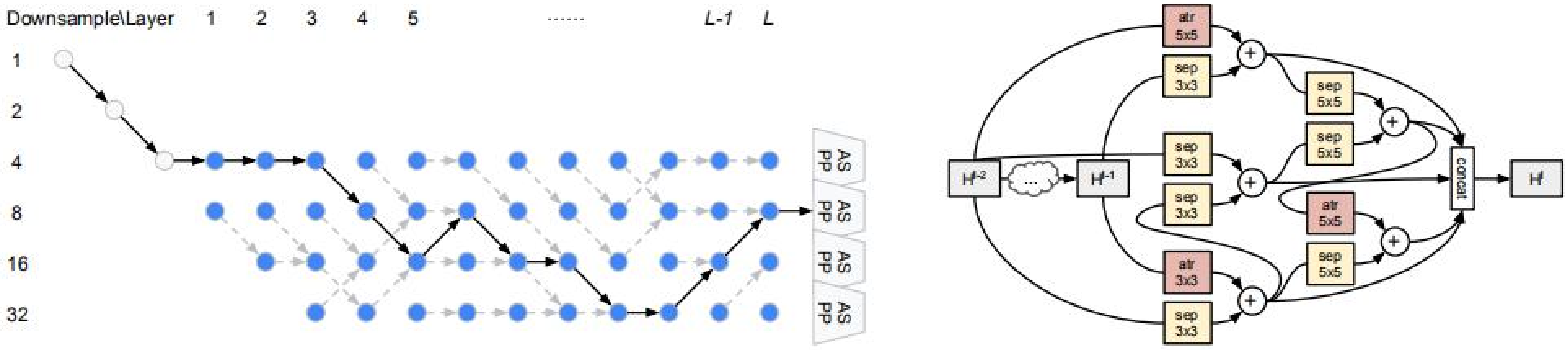


Fig.6 Searched result

2.2 Remain problems (NAS; Medical image segmentation)

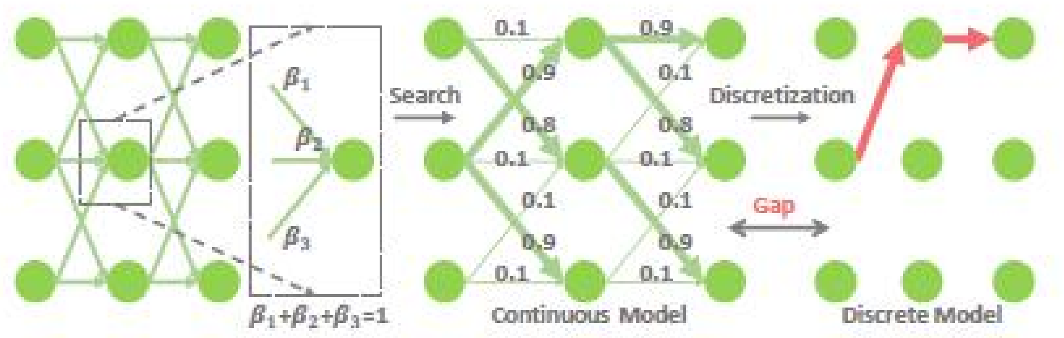


Fig.7 Limitation of formal single-path discrete model

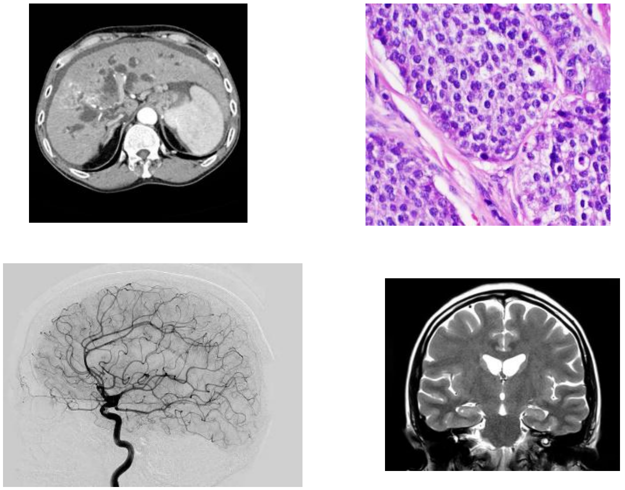


Fig.8 Different medical images

(1) Current single-path discrete model can result in a large 'discretization gap' with searched continuous model.
(2) NAS based methods usually cost tremendous training time and memory usage.
(3) Manually designed networks, like U-Net, are less likely to be optimal for different type of medical images.

3. Method

DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation

—— Yufan He, Dong Yang, Holger Roth, Can Zhao, Daguang Xu
Johns Hopkins University, NVIDIA

(CVPR2021)

3 Overall

3.1 Network Topology Search Space

网络搜索空间

3.2 Continuous Relaxation and Discretization

连续松弛和离散化操作

3.3 Discretization with topology constraints

用拓扑算法来进行离散操作

3.4 Bridging the Discretization Gap

缓解离散-连续转化Gap

3.5 Memory Budget Constraints

控制内存的消耗

3.6 Optimization

3.1 Network Topology Search Space

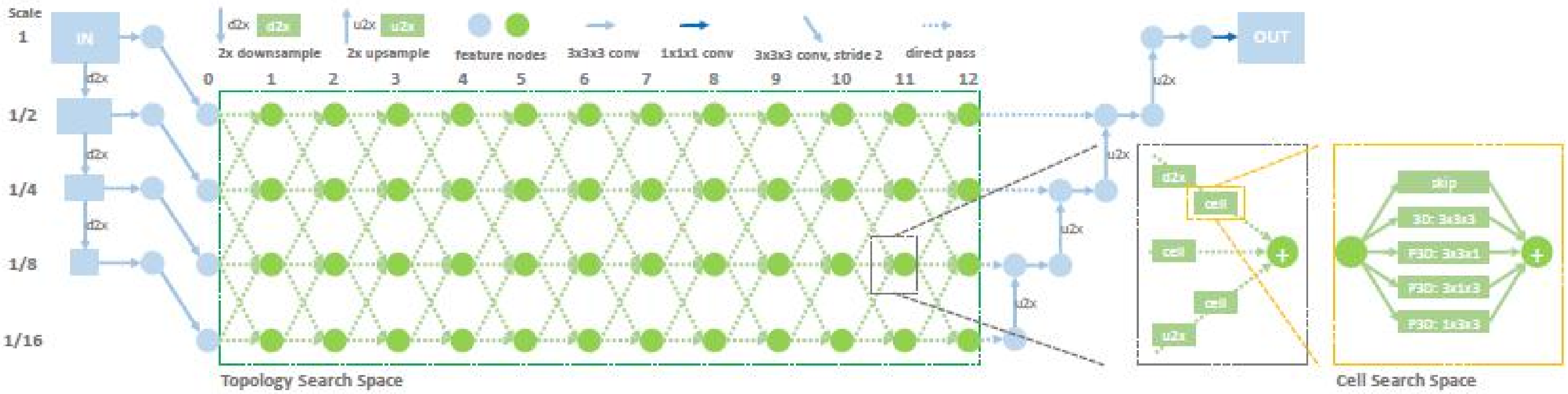


Fig.8 Left: Topology search space; Right: Cell search space

Outer network search space:
Layer number: L = 12.
Input for layer 0: D = 4 feature map of different scale.
Input for layer i: E = 4 nodes * j=3 inputs (upsampling/no change/downsampling) – 2 (first/end node).
Edge: Each edge contains a cell and a spatial operation.
Output: Summation of edge outputs.

Inner cell operations:
Note that the spatial operations are not included in cells
P3D： Pseudo 3D

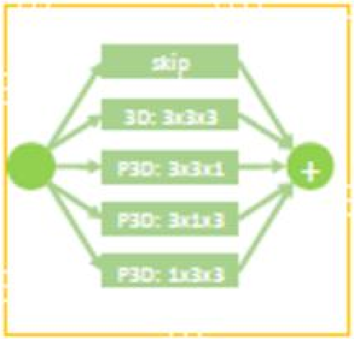
(1) skip connection (2) 3x3x3 3D convolution
(3) P3D 3x3x1: 3x3x1 followed by 1x1x3 convolution
(4) P3D 3x1x3: 3x1x3 followed by 1x3x1 convolution
(5) P3D 1x3x3: 1x3x3 followed by 3x1x1 convolution

3.2 **Continuous Relaxation and Discretization**

3.2.1 Cell Space Relaxation follow the same method with DARTS.

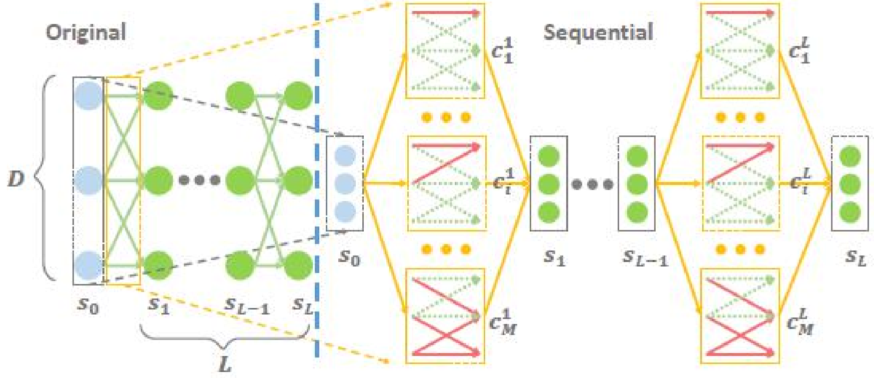
$$\bar{o}_{(i,j)}(x) = \sum_{o \in O} \frac{exp(\alpha_{o,(i,j)})}{\sum_{o \in O} exp(\alpha_{o,(i,j)})} o(x)$$

$$\alpha = [\alpha_{1(i,j)} \dots \alpha_{|O|,(i,j)}]$$



Cell Search Space

3.2.2 Network Topology Space Relaxation.



select one from $c_1^1, c_2^1, \cdots, c_M^1$ with probability $\eta_1^1, \eta_2^1, \cdots, \eta_M^1$

1. D feature nodes of each layer are combined as a 'super feature node' s_i
2. From s_{i-1} to s_i, there are M = E^2-1 connection pattern cp.
3. Define the input connection operation to s_i with connection pattern cp_j as c_j^i, it also includes cell operations on the selected edges in cp_j.
4. Associate a variable η_j^i to the connection operation c_j^i.

Goal: Select one connection pattern to connect super feature nodes.

$$s_i = \sum_{j=1}^{M} (\eta_j^i * c_j^i(s_{i-1})) \quad i = 1 \cdots, L \qquad (1)$$

$$\sum_{j=1}^{M} \eta_j^i = 1, \eta_j \geq 0 \quad \forall i, j$$

3.3 **Discretization with topology constraints**

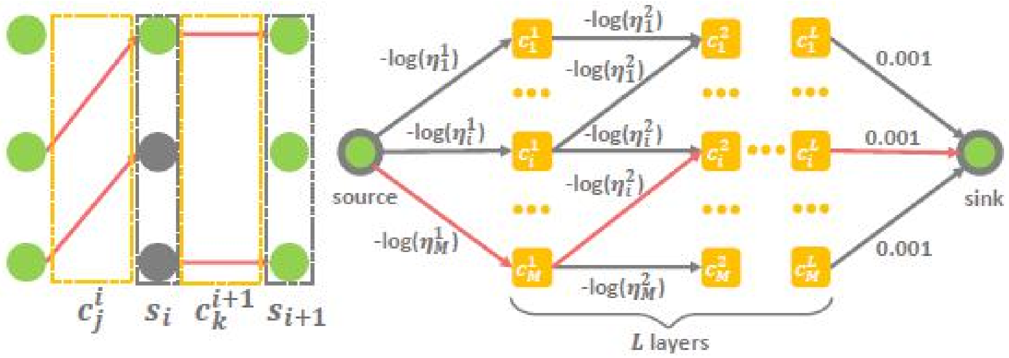


Fig.9 Derive discrete architecture

A directed graph G contains L*M+2 nodes. Each input edge cost is -log η_j^i. Those L nodes on the shortest path from source to sink in G represents the optimal connection operations (Dijkstra Algorithm). For cell operations, simply use the operation with the largest α.

Why we need topology constraints —— Because some connection pattern may not feasible (gray nodes in Fig. 9)

I: Array of selected input connection pattern indexes (whole path)

F(j): All the feasible output connection pattern with input pattern j

p(I): Distribution of input connection pattern indexes

$$p(I) = \begin{cases} \prod_{i=1}^{L} \eta_i^{I(i)}, & \forall i: \quad I(i+1) \in \mathcal{F}(I(i)) \\ 0, & \text{else.} \end{cases} \quad (3)$$

$$I = \operatorname*{argmin}_{I} \sum_{i=1}^{L} \text{-}\log(\eta_i^{I(i)}), \ \forall i: I(i+1) \in \mathcal{F}(I(i)) \quad (4)$$

3.4 **Bridging the Discretization Gap**

3.4.1 Encourage binarization of α and η.

$$
\begin{aligned}
\mathcal{L}_\alpha &= \frac{-1}{L*E*N}\sum_{i=1}^{L}\sum_{e=1}^{E}\sum_{n=1}^{N}\alpha_n^{i,e}*log(\alpha_n^{i,e}) \\
\mathcal{L}_\eta &= \frac{-1}{L*M}\sum_{i=1}^{L}\sum_{j=1}^{M}\eta_j^i*log(\eta_j^i)
\end{aligned}
\quad (5)
$$

3.4.2 Try to constraint the network to build feasible connection patterns.

$$
p_{in}^i(a) = \sum_{j\in F_{in}(a)}\eta_j^i, \quad p_{out}^i(a) = \sum_{j\in F_{out}(a)}\eta_j^{i+1} \quad (6)
$$

$$
\begin{aligned}
\mathcal{L}_{tp} = -\sum_{i=1}^{L-1}\sum_{a\in\mathcal{A}}(\quad & p_{in}^i(a)\log(p_{out}^i(a)) \quad + \\
& (1-p_{in}^i(a))\log(1-p_{out}^i(a)) \quad)
\end{aligned}
\quad (7)
$$

a: Indication function of length D, a(i) = 1 if i-th node of a super feature node is activated.

F_{in}(a)/F_{out}: All feasible input and output connection pattern indexes for activated a.

p_{in}^i(a): The probability that the activation pattern for s_i is a.

p_{out}^i(a): The probability that the s_i with pattern a is feasible.

L_{tp}: By minimizing L_{tp}, the search stage is aware of topology constraints and encourages all the super feature nodes to be topologically feasible.

3.5 **Memory Budget Constraints**

Reason: For segmentation tasks, the searched model is usually retrained under different training setting (input size, filter number, datasets). The memory budget normally huge in 3D image training tasks.

Solution: Consider memory usage in architecture search

$$M_e = \sum_{i=1}^{L}\sum_{j=1}^{M} \eta_j^i * (\sum_{e=1}^{E} M^{i,e} * cp_j(e)) \quad M^{i,e} = \sum_{n=1}^{N} \alpha_n^{i,e} M_n.$$

$$M_a = \sum_{i=1}^{L}\sum_{j=1}^{M} *(\sum_{e=1}^{E}(\sum_{n=1}^{N} M_n) * cp_j(e))$$

$$\mathcal{L}_m = |M_e/M_a - \sigma|_1$$

M_n: Estimate memory usage for operation O_n

M_e: Expected memory usage of searched model

M_a: Maximum memory usage of whole model

σ: Memory budget

3.6 **Optimization**

1. Split the training data into train1 and train2.
2. The optimization using 2 different loss alternately.

Optimize network weight w using L_seg (DICE + cross-entropy loss) with train1;
Optimize architecture weights α and η using L_arch with train2:

$$\mathcal{L}_{arch} = \mathcal{L}_{seg} + t/t_{all} * (\mathcal{L}_{\alpha} + \mathcal{L}_{\eta} + \lambda * \mathcal{L}_{tp} + \mathcal{L}_{m})$$

t/t_{all}: current iteration / total iterations； λ = 0.001 ；

$$\mathcal{L}_{\alpha} = \frac{-1}{L * E * N} \sum_{i=1}^{L} \sum_{e=1}^{E} \sum_{n=1}^{N} \alpha_n^{i,e} * log(\alpha_n^{i,e})$$

$$\mathcal{L}_{\eta} = \frac{-1}{L * M} \sum_{i=1}^{L} \sum_{j=1}^{M} \eta_j^i * log(\eta_j^i)$$

$$\mathcal{L}_{tp} = -\sum_{i=1}^{L-1} \sum_{a \in \mathcal{A}} (\quad p_{in}^i(a) \log(p_{out}^i(a)) \quad + \\ (1 - p_{in}^i(a)) \log(1 - p_{out}^i(a)) \quad)$$

$$\mathcal{L}_{m} = |M_e/M_a - \sigma|_1$$

4. Experiments

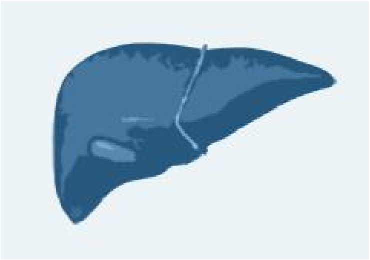
4.1 Training and Testing datasets:

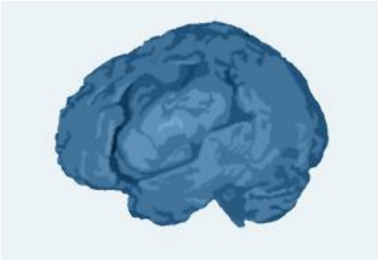
MSD dataset has ten segmentation tasks (CT/MRI Liver/Brain/Hippocampus(海马体)/Lung/Prostate(前列腺)/Cardiac(心脏)/ Pancreas (胰腺)/ Colon(结肠)/ Hepatic(肝脏)/ Spleen(脾脏))

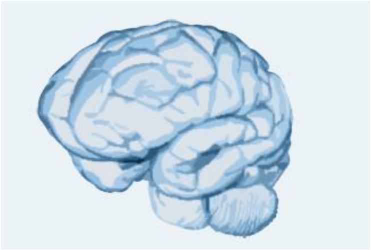
Numbers: 131/70; 484/266; 263/131; 64/32; 32/16; 20/10; 282/139; 126/64; 303/140; 41/20.

Training set for search: Pancreas dataset (282 3D CT images).

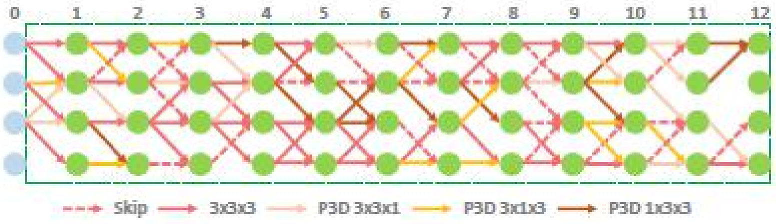
Testing set: All the testing set of ten tasks.



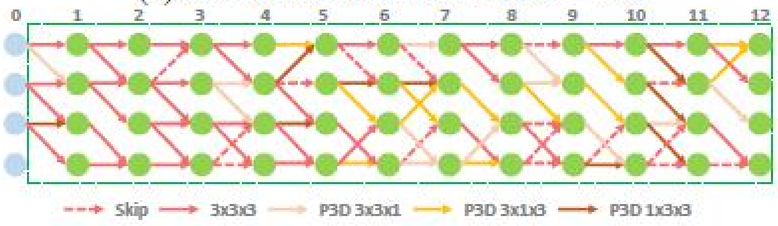




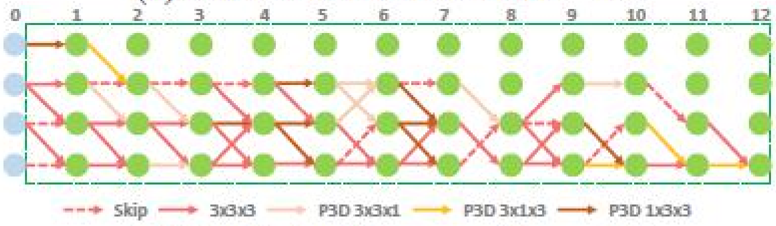
4.2 Evaluation Metrics (time/efficiency)



(a) Searched architecture with $\sigma = 0.8$



(b) Searched architecture with $\sigma = 0.5$



(c) Searched architecture with $\sigma = 0.2$

Table 1. Comparison of FLOPs, Parameters and Retraining GPU memory usage and the 5-Fold cross validation Dice-Sørensen score of our searched architectures on Pancreas dataset

Model	FLOPs (G)	Params. (M)	Memory (MB)	DSC1	DSC2	Avg.
3D UNet [6] (nn-UNet)	658	18	9176	-	-	-
Attention UNet [28]	1163	104	13465	-	-	-
C2FNAS [45]	151	17	5730	-	-	-
DiNTS (σ=0.2)	146	163	5787	77.94	48.07	63.00
DiNTS (σ=0.5)	308	147	10110	**80.20**	52.25	66.23
DiNTS (σ=0.8)	334	152	13018	80.06	**52.53**	**66.29**

- Search Time: DiNTS/C2FNAS 5.8 GPU(V100) days / 333 GPU(V100) days
- FLOPS/Params/Memory: No advantage but achieve 3D NAS successfully.

4.2 Evaluation Metrics (performance)

- Segmentation performance:

MSD challenge champion nnUNet:
ensembles 2D/3D/Cascaded-3D Unet on different tasks,
hand-crafted hyper-parameters.

Latest NAS SOTA on MSD dataset: C2FNAS

Results:
DiNTS is better on bigger dataset (Pancrease, Brain, Colon), worse on smaller dataset (Heart (10), Prostate (16), Spleen(20)). DiNTS outperforms manual-designed and NAS SOTA with best average performance for all ten tasks.

	Brain							
Metric	DSC1	DSC2	DSC3	Avg.	NSD1	NSD2	NSD3	Avg.
CerebriuDIKU [30]	**69.52**	43.11	66.74	59.79	88.25	68.98	88.90	82.04
NVDLMED [41]	67.52	45.00	68.01	60.18	86.99	69.77	89.82	82.19
Kim et al [15]	67.40	45.75	68.26	60.47	86.65	72.03	90.28	82.99
nnUNet [14]	68.04	46.81	68.46	61.10	87.51	72.47	90.78	83.59
C2FNAS [45]	67.62	48.60	69.72	61.98	87.61	72.87	91.16	83.88
DiNTS	69.28	**48.65**	**69.75**	**62.56**	**89.33**	**73.16**	**91.69**	**84.73**

	Heart		Liver					
Metric	DSC1	NSD1	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.
CerebriuDIKU [30]	89.47	90.63	94.27	57.25	75.76	96.68	72.60	84.64
NVDLMED [41]	92.46	95.57	95.06	71.40	83.23	98.26	87.16	92.71
Kim et al [15]	93.11	96.44	94.25	72.96	83.605	96.76	88.58	92.67
nnUNet [14]	**93.30**	**96.74**	**95.75**	**75.97**	**85.86**	98.55	90.65	94.60
C2FNAS [45]	92.49	95.81	94.98	72.89	83.94	98.38	89.15	93.77
DiNTS	92.99	96.35	95.35	74.62	84.99	**98.69**	**91.02**	**94.86**

	Lung		Hippocampus					
Metric	DSC1	NSD1	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.
CerebriuDIKU [30]	58.71	56.10	89.68	88.31	89.00	97.42	97.42	97.42
NVDLMED [41]	52.15	50.23	87.97	86.71	87.34	96.07	96.59	96.33
Kim et al [15]	63.10	62.51	90.11	88.72	89.42	97.77	**97.73**	**97.75**
nnUNet [14]	73.97	76.02	**90.23**	**88.69**	**89.46**	**97.79**	97.53	97.66
C2FNAS [45]	70.44	72.22	89.37	87.96	88.67	97.27	97.35	97.31
DiNTS	**74.75**	**77.02**	89.91	88.41	89.16	97.76	97.56	97.66

	Spleen		Prostate					
Metric	DSC1	NSD1	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.
CerebriuDIKU [30]	95.00	98.00	69.11	86.34	77.73	94.72	97.90	96.31
NVDLMED [41]	96.01	99.72	69.36	86.66	78.01	92.96	97.45	95.21
Kim et al [15]	91.92	94.83	72.64	89.02	80.83	95.05	98.03	96.54
nnUNet [14]	**97.43**	**99.89**	**76.59**	**89.62**	**83.11**	**96.27**	**98.85**	**97.56**
C2FNAS [45]	96.28	97.66	74.88	88.75	81.82	98.79	95.12	96.96
DiNTS	96.98	99.83	75.37	89.25	82.31	95.96	98.82	97.39

	Colon		Hepatic Vessels					
Metric	DSC1	NSD1	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.
CerebriuDIKU [30]	28.00	43.00	59.00	38.00	48.50	79.00	44.00	61.50
NVDLMED [41]	55.63	66.47	61.74	61.37	61.56	81.61	68.82	75.22
Kim et al [15]	49.32	62.21	62.34	68.63	65.485	83.22	78.43	80.825
nnUNet [14]	58.33	68.43	**66.46**	**71.78**	**69.12**	**84.43**	80.72	**82.58**
C2FNAS [45]	58.90	**72.56**	64.30	71.00	67.65	83.78	80.66	82.22
DiNTS	**59.21**	70.34	64.50	71.76	68.13	83.98	**81.03**	82.51

	Pancreas						**Overall**	
Metric	DSC1	DSC2	Avg.	NSD1	NSD2	Avg.	DSC	NSD
CerebriuDIKU [30]	71.23	24.98	48.11	91.57	46.43	69.00	67.01	77.86
NVDLMED [41]	77.97	44.49	61.23	94.43	63.45	78.94	72.78	83.26
Kim et al [15]	80.61	51.75	66.18	95.83	73.09	84.46	74.34	85.12
nnUNet [14]	**81.64**	52.78	67.21	96.14	71.47	83.81	77.89	88.09
C2FNAS [45]	80.76	54.41	67.59	96.16	75.58	85.87	76.97	87.83
DiNTS	81.02	**55.35**	**68.19**	**96.26**	**75.90**	**86.08**	**77.93**	**88.68**