CVPR 2020

Learning Dynamic Routing for Semantic Segmentation

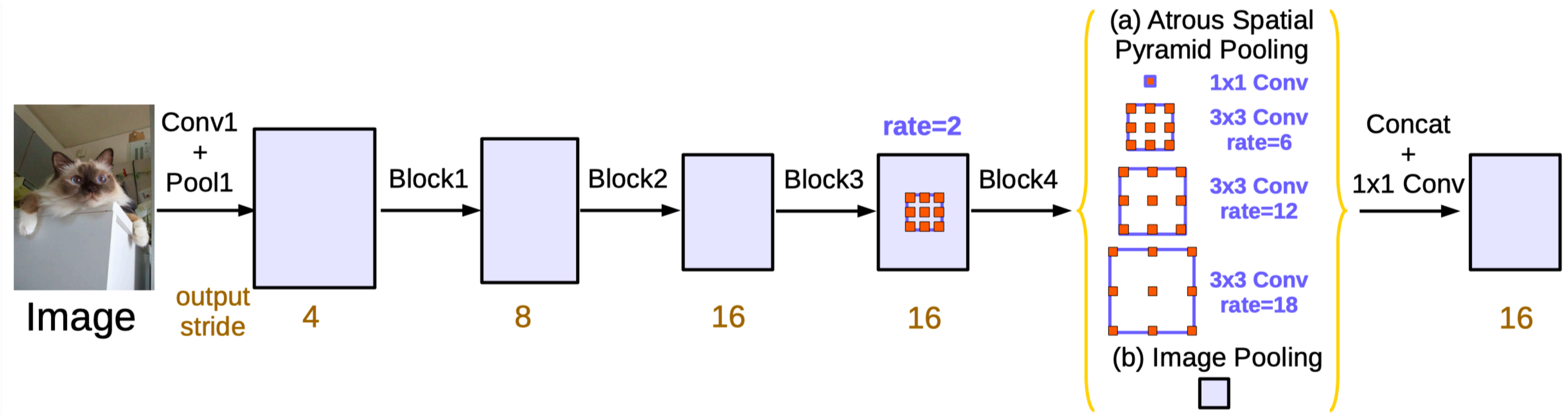
Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang,
Xingang Wang, Jian Sun

MEGVII 旷视

Traditional architecture for semantic segmentation

Human-designed

DeepLab V3: use human-designed pipeline and ASPP module



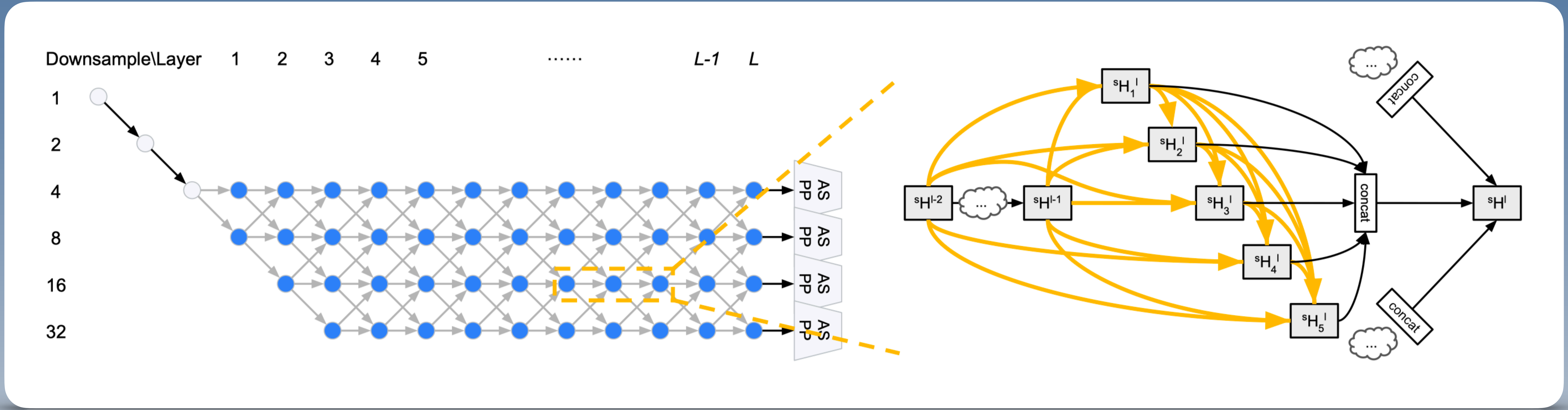
Chen Liang-Chieh, et.al. "Rethinking Atrous Convolution for Semantic Image Segmentation", arXiv:1706.05587

Introduction

Traditional architecture for semantic segmentation

NAS-based

Auto-DeepLab: search in the designed space for a single path



Liu Chenxi, et.al. "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation", In CVPR 2019

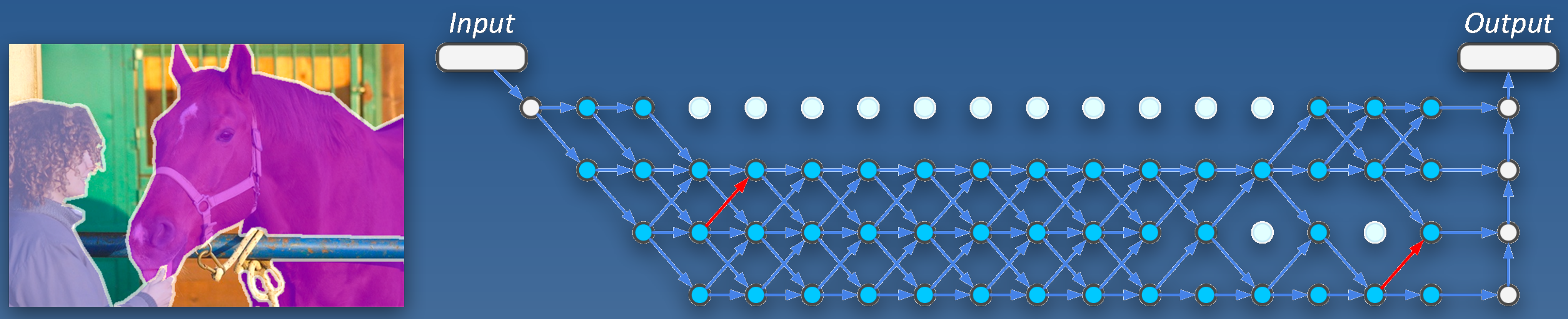
Introduction

Traditional architecture for semantic segmentation

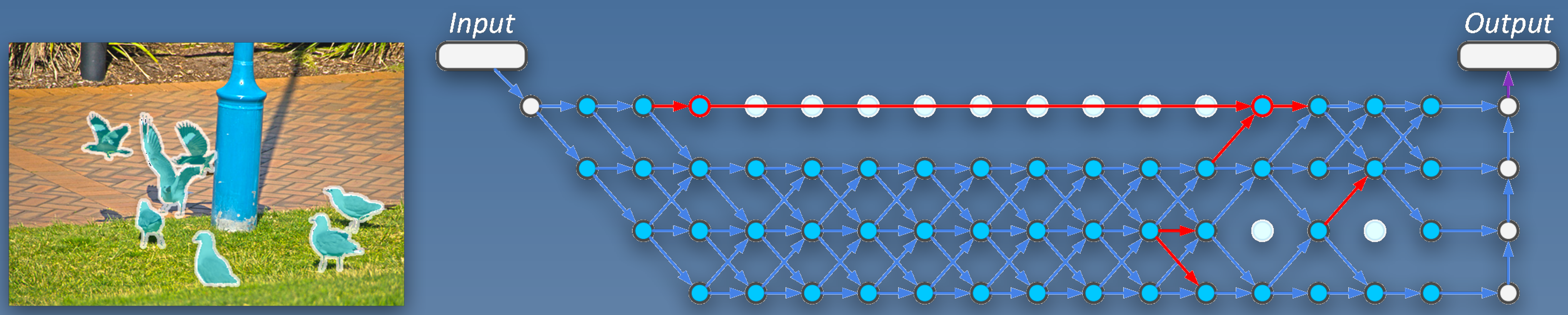
*All of them are **static** or **fixed** for inference*

*However, there exists **huge scale variance** among inputs!*

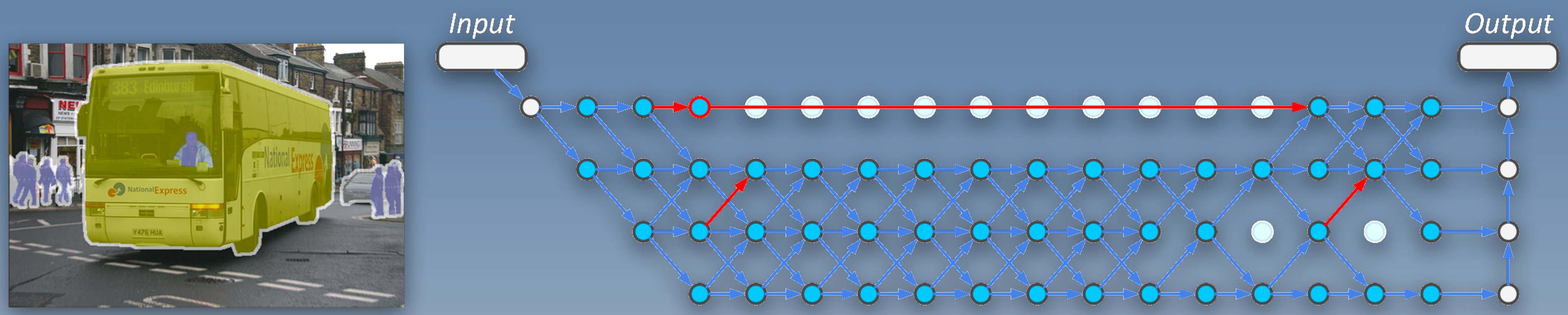
Thus, inference paths should adapt to the input image.
*We need dynamic routing for **data-dependent** architecture!*



Network architecture of large-scale input



Network architecture of small-scale input



Network architecture of mixed-scale input

Inputs with various scales may need different routes

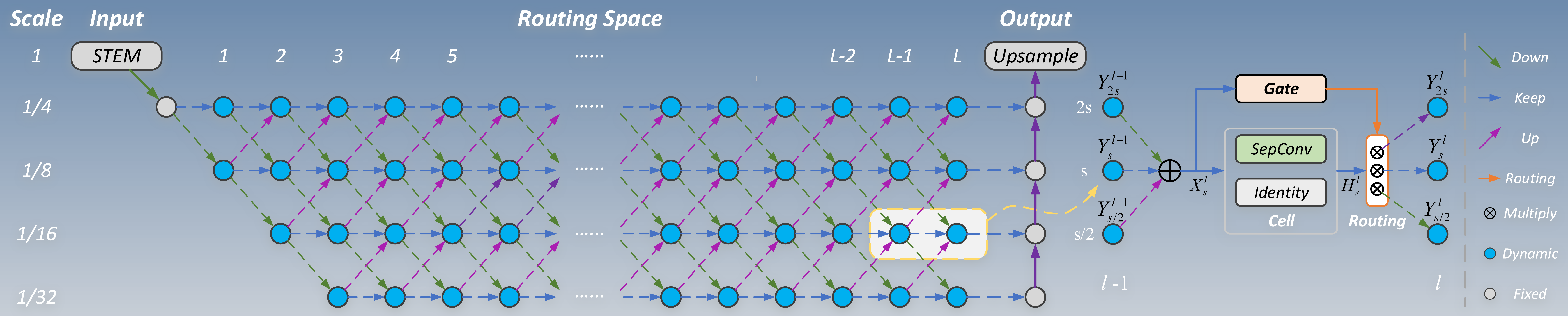
Dynamic Routing

Dynamic framework for semantic segmentation

*Dynamic Routing have the superiority in **network capacity** and **higher performance** with budgeted constraints.*

Here, we give the proposed dynamic routing framework:

Left: The routing space with layer L and max downsampling rate 32. Right: Dynamic routing process at the node level.

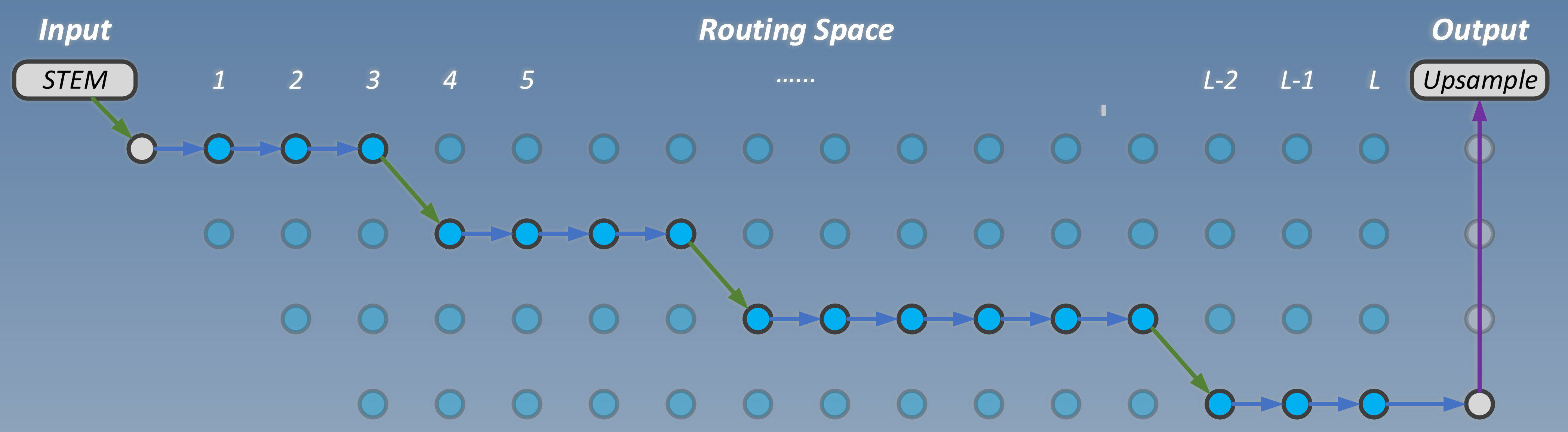


Dynamic Routing

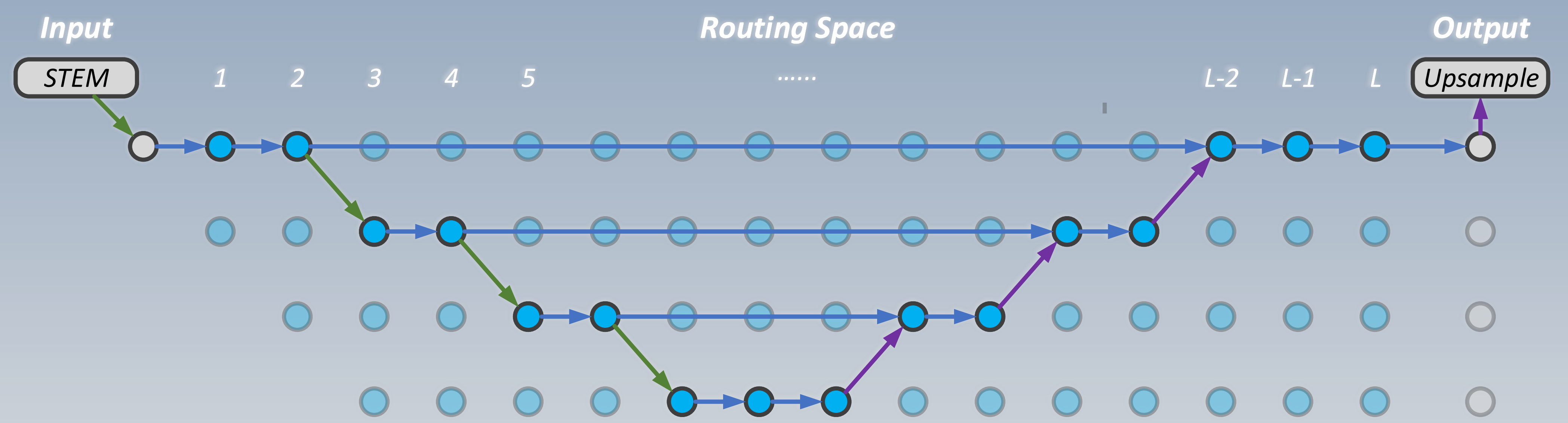
CVPR 2020

Dynamic routing space

With the support for **multi-scale routes** and **skip connection**, several classic architectures can be formulated in similar forms.



Network architecture modeled from FCN-32s



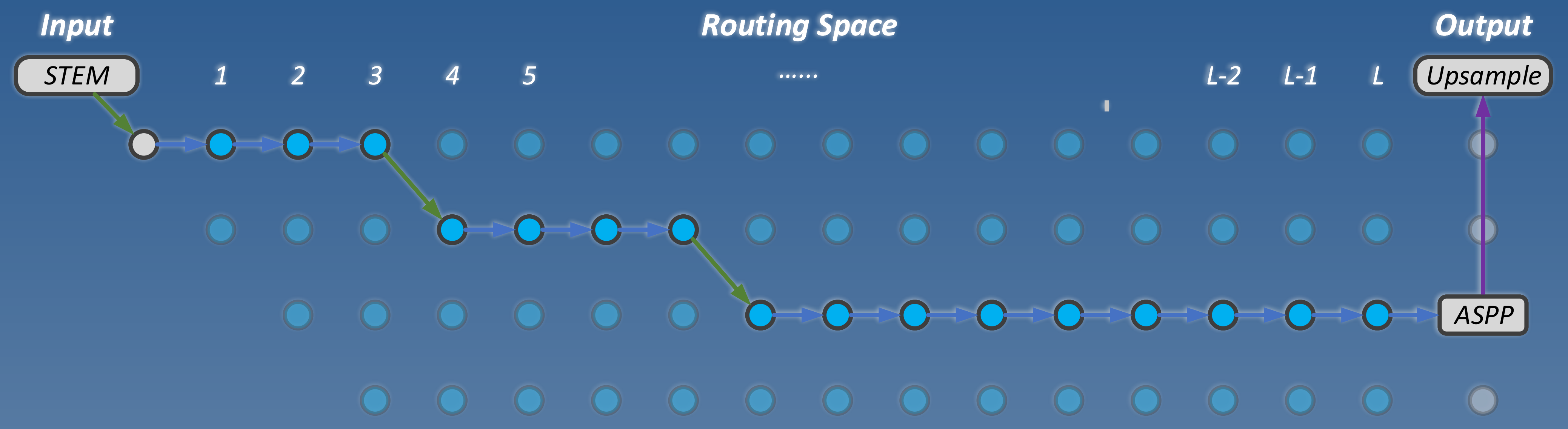
Network architecture modeled from U-Net

Dynamic Routing

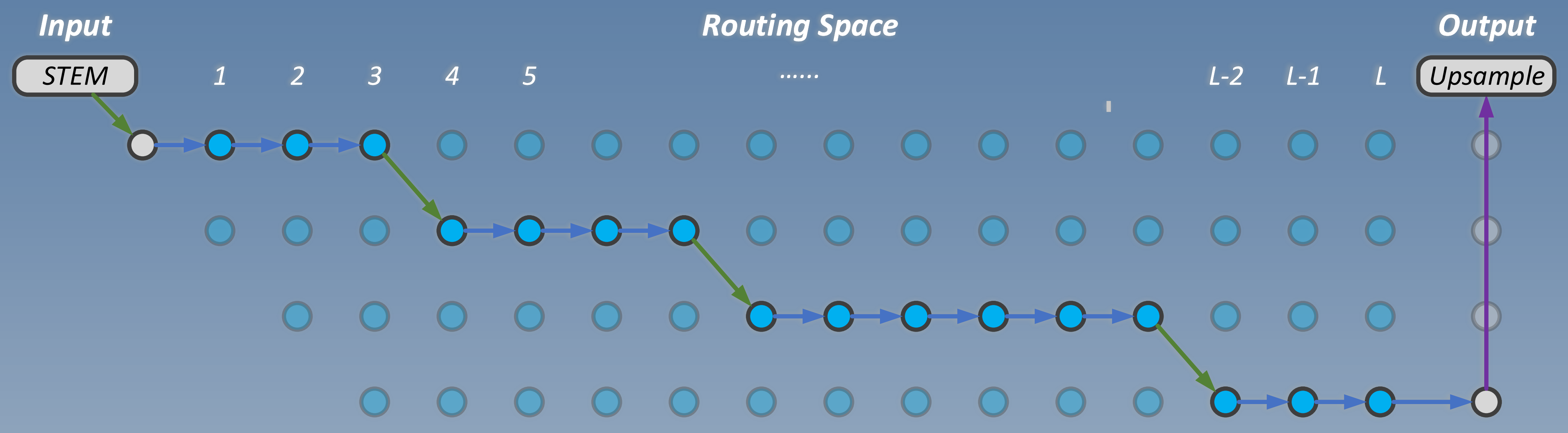
CVPR 2020

Dynamic routing space

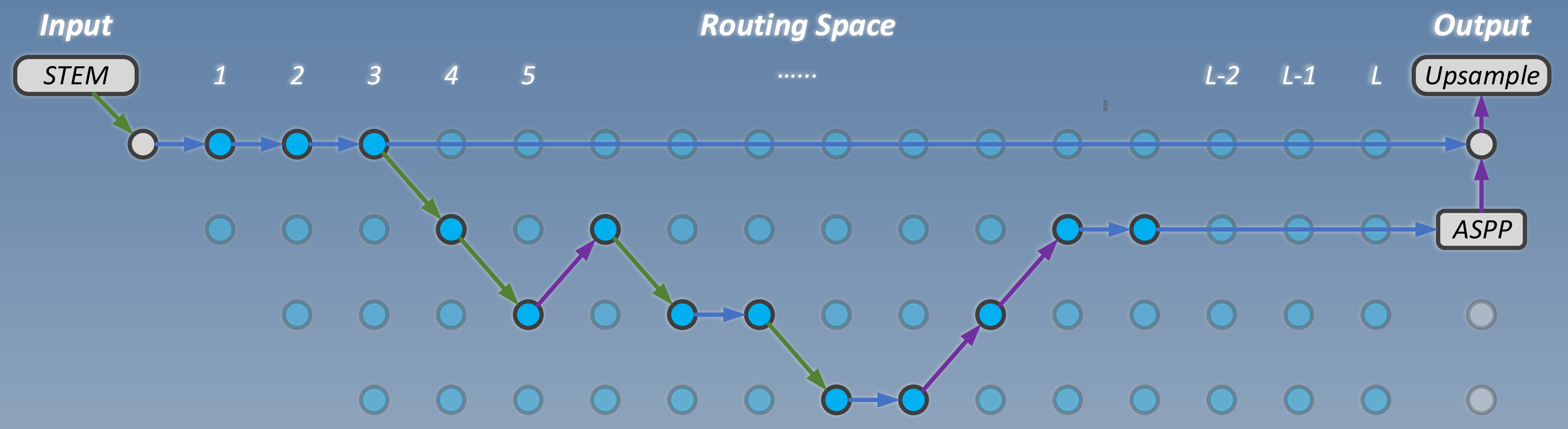
*With the support for **multi-scale routes** and **skip connection**, several classic architectures can be formulated in similar forms.*



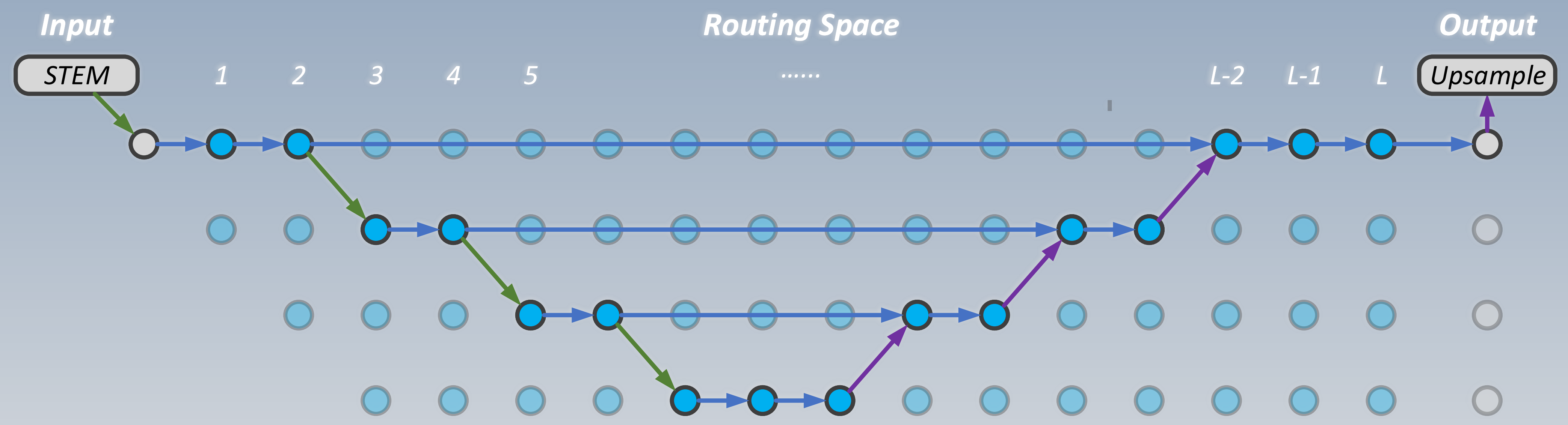
Network architecture modeled from DeepLab V3



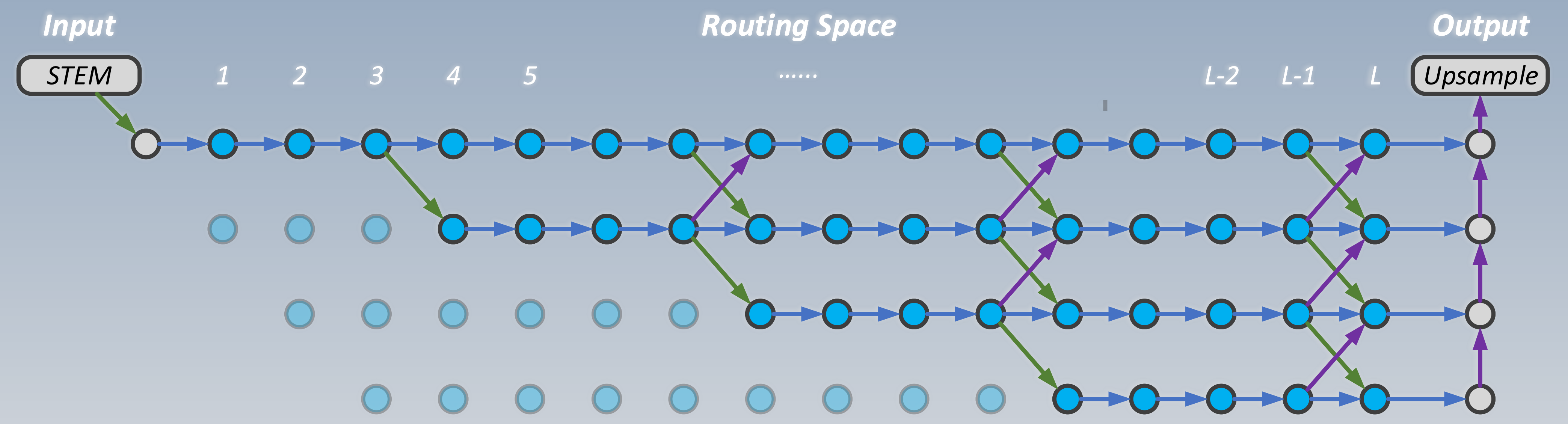
Network architecture modeled from FCN-32s



Network architecture modeled from Auto-DeepLab



Network architecture modeled from U-Net

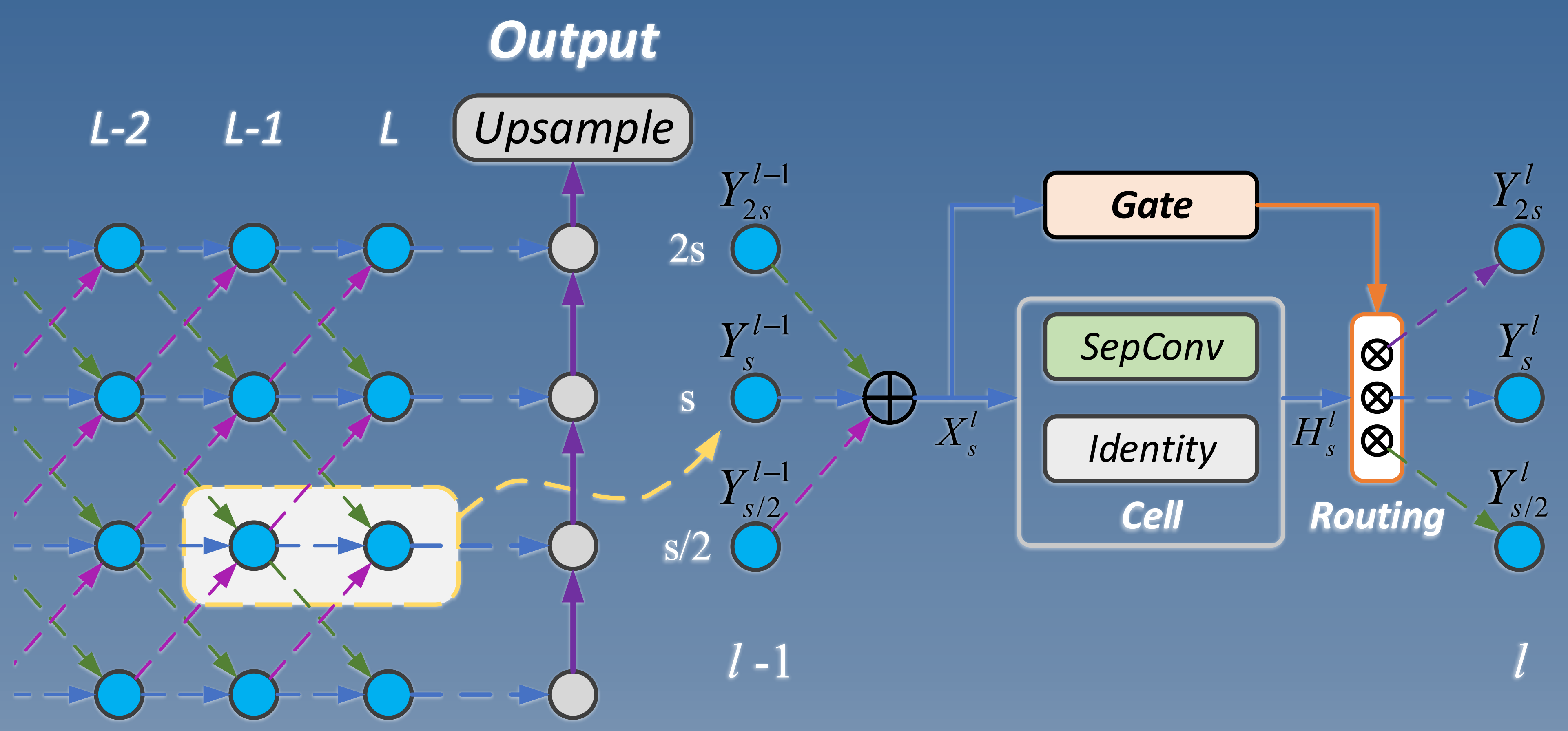


Network architecture modeled from HRNet V2

Dynamic Routing

Dynamic routing process

*Given the routing space with several individual nodes, we adopt a basic **cell** and a **gate** inside **each node**.*



Cell Operation:
aggregate multi-scale features from the former layer

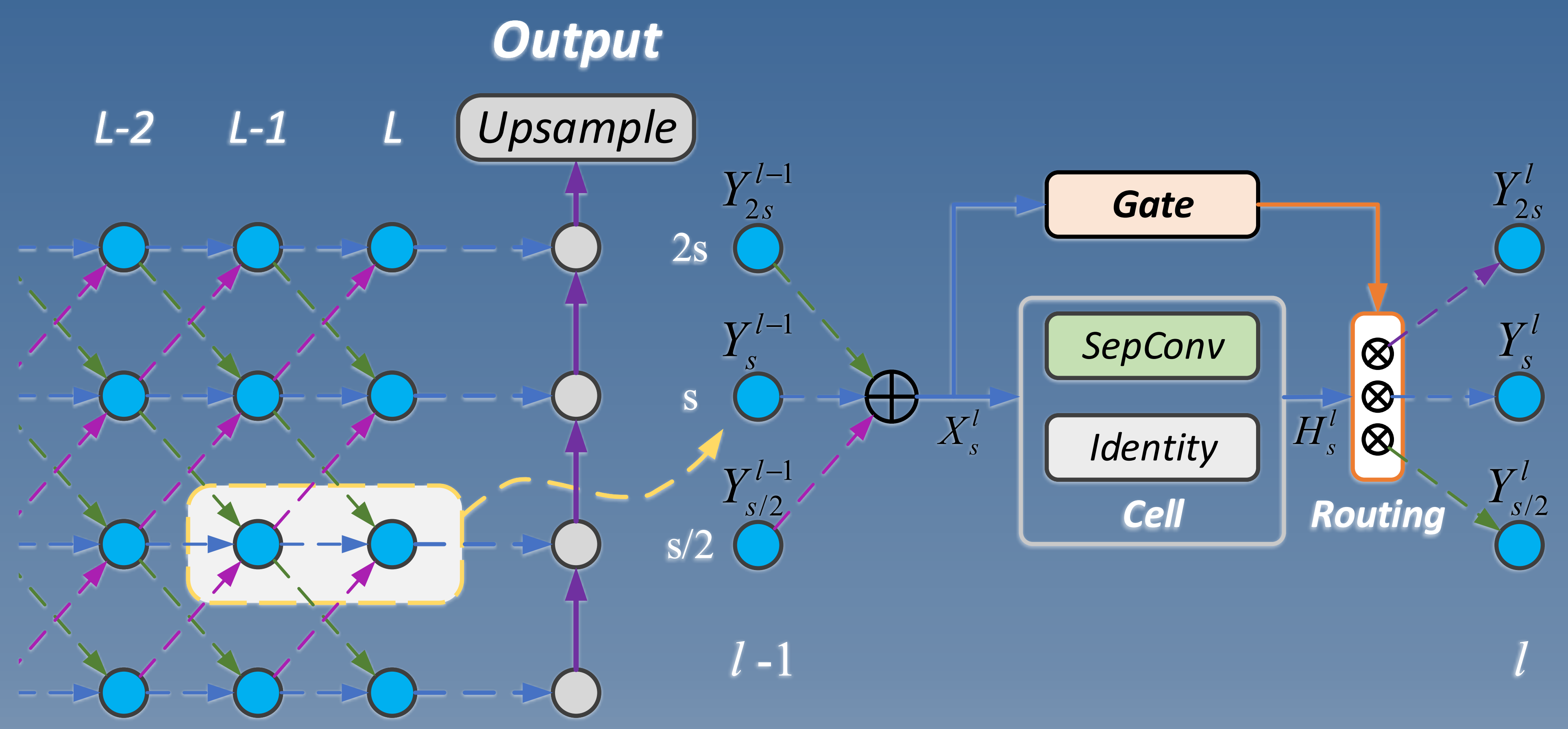
Hidden state feature

$$\mathbf{H}_s^l = \sum_{O^i \in \mathcal{O}} O^i(\mathbf{X}_s^l)$$

Dynamic Routing

Dynamic routing process

Given the routing space with several individual nodes, we adopt a basic cell and a gate inside each node.



Cell Operation:

aggregate multi-scale features from the former layer

Hidden state feature

$$\mathbf{H}_s^l = \sum_{O^i \in \mathcal{O}} O^i(\mathbf{X}_s^l)$$

Soft Conditional Gate:

choose routing paths to the next layer

Gating feature

$$\mathbf{G}_s^l = \mathrm{Conv}(\mathrm{Pool}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(\mathbf{X}_s^l))))) + \beta_s^l$$

Activating weights

$$\alpha_s^l = \max(0, \mathrm{Tanh}(\mathbf{G}_s^l))$$

Activated feature

$$\mathbf{H}_s^l = \begin{cases} \mathbf{X}_s^l & \sum_j \alpha_{s\to j}^l = 0 \\ \sum_{O^i \in \mathcal{O}} O^i(\mathbf{X}_s^l) & \sum_j \alpha_{s\to j}^l > 0 \end{cases}$$

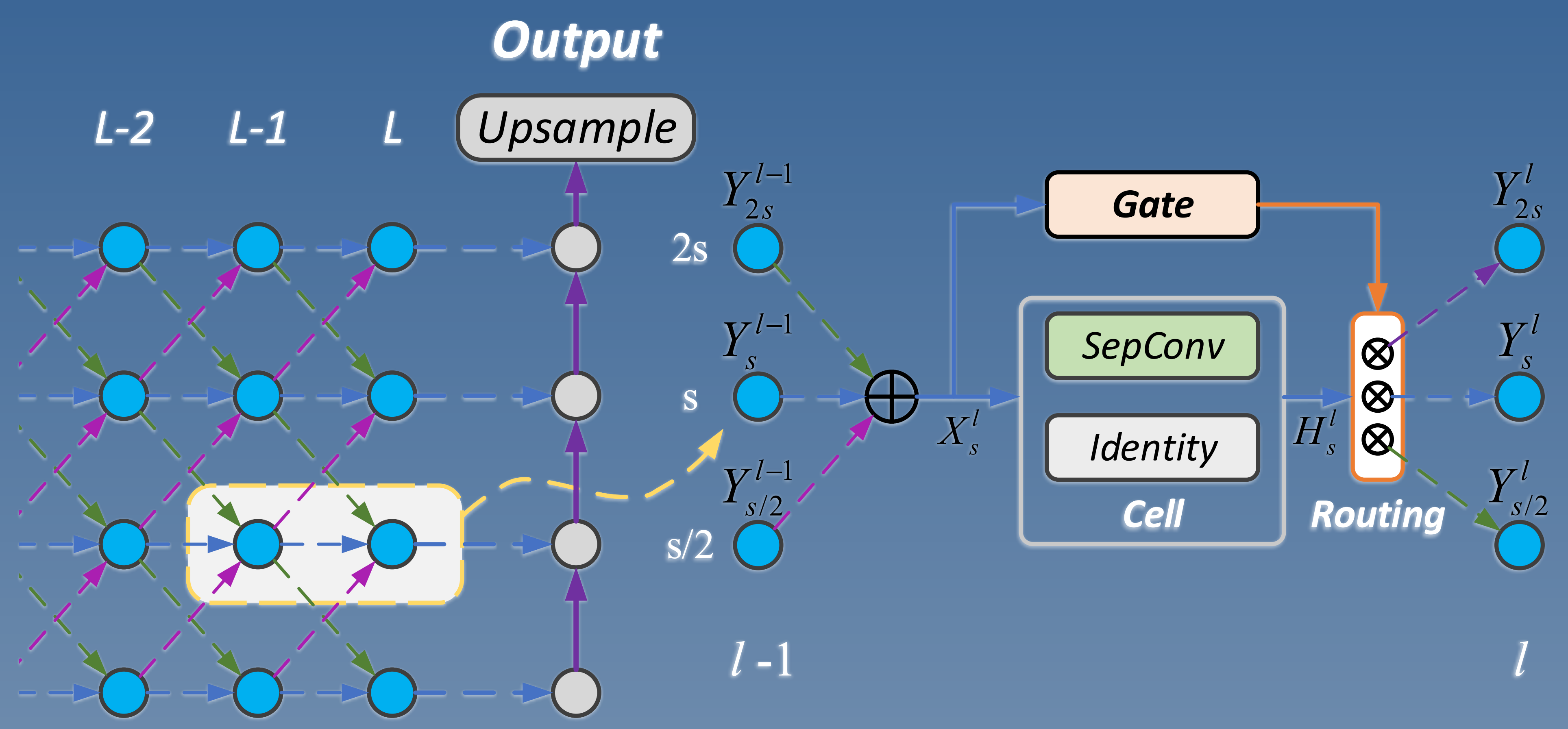
Output feature

$$\mathbf{Y}_j^l = \begin{cases} 0 & \sum_j \alpha_{s\to j}^l = 0,\ j \neq s \\ \mathbf{H}_s^l & \sum_j \alpha_{s\to j}^l = 0,\ j = s \\ \alpha_{s\to j}^l \mathscr{T}_{s\to j}(\mathbf{H}_s^l) & \sum_j \alpha_{s\to j}^l > 0 \end{cases}$$

Dynamic Routing

Dynamic routing process

*Given the routing space with several individual nodes, we adopt a basic **cell** and a **gate** inside **each node**.*



Cell Operation:

aggregate multi-scale features from the former layer

Hidden state feature

$$\mathbf{H}_s^l = \sum_{O^i \in \mathcal{O}} O^i(\mathbf{X}_s^l)$$

Soft Conditional Gate:

choose routing paths to the next layer

Gating feature

$$\mathbf{G}_s^l = \mathrm{Conv}(\mathrm{Pool}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(\mathbf{X}_s^l))))) + \beta_s^l$$

Activating weights

$$\alpha_s^l = \max(0, \mathrm{Tanh}(\mathbf{G}_s^l))$$

Activated feature

$$\mathbf{H}_s^l = \begin{cases} \mathbf{X}_s^l & \sum_j \alpha_{s\to j}^l = 0 \\ \sum_{O^i \in \mathcal{O}} O^i\left(\mathbf{X}_s^l\right) & \sum_j \alpha_{s\to j}^l > 0 \end{cases}$$

Output feature

$$\mathbf{Y}_j^l = \begin{cases} 0 & \sum_j \alpha_{s\to j}^l = 0,\ j \neq s \\ \mathbf{H}_s^l & \sum_j \alpha_{s\to j}^l = 0,\ j = s \\ \alpha_{s\to j}^l \mathcal{T}_{s\to j}\left(\mathbf{H}_s^l\right) & \sum_j \alpha_{s\to j}^l > 0 \end{cases}$$

Budget Constraint:

consider budget constraint for efficient inference

Per-node cost

$$\mathscr{C}(\mathrm{Node}_s^l) = \mathscr{C}(\mathrm{Cell}_s^l) + \mathscr{C}(\mathrm{Gate}_s^l) + \mathscr{C}(\mathrm{Trans}_s^l)$$

Space-level cost

$$\mathscr{C}(\mathrm{Space}) = \sum_{l \le L} \sum_{s \le 1/4} \mathscr{C}(\mathrm{Node}_s^l)$$

Constraint loss

$$\mathscr{L}_C = (\mathscr{C}(\mathrm{Space})/C - \mu)^2$$

Experiments

Ablation Studies

We compare with several classic architectures under similar FLOPs, which are modeled on the same routing space.

Comparisons with classic architectures on the Cityscapes val set

Method	Dynamicc	Modeled from	mIoU (%)	$FLOPS_{Avg}$ (G)	$FLOPS_{Max}$ (G)	$FLOPS_{Min}$ (G)
Handcrafted	×	FCN-32s	66.9	35.1	35.1	35.1
	×	DeepLab V3	67.0	42.5	42.5	42.5
	×	U-Net	71.6	53.9	53.9	53.9
	×	HRNetV2	72.5	62.5	62.5	62.5
Searched	×	Auto-DeepLab	67.2	33.1	33.1	33.1
Dynamic-A	✓	***Routing-Space***	***72.8***	***44.9***	***48.2***	***43.5***
Dynamic-B	✓	***Routing-Space***	***73.8***	***58.7***	***63.5***	***56.8***
Dynamic-C	✓	***Routing-Space***	***74.6***	***66.6***	***71.6***	***64.3***

Experiments

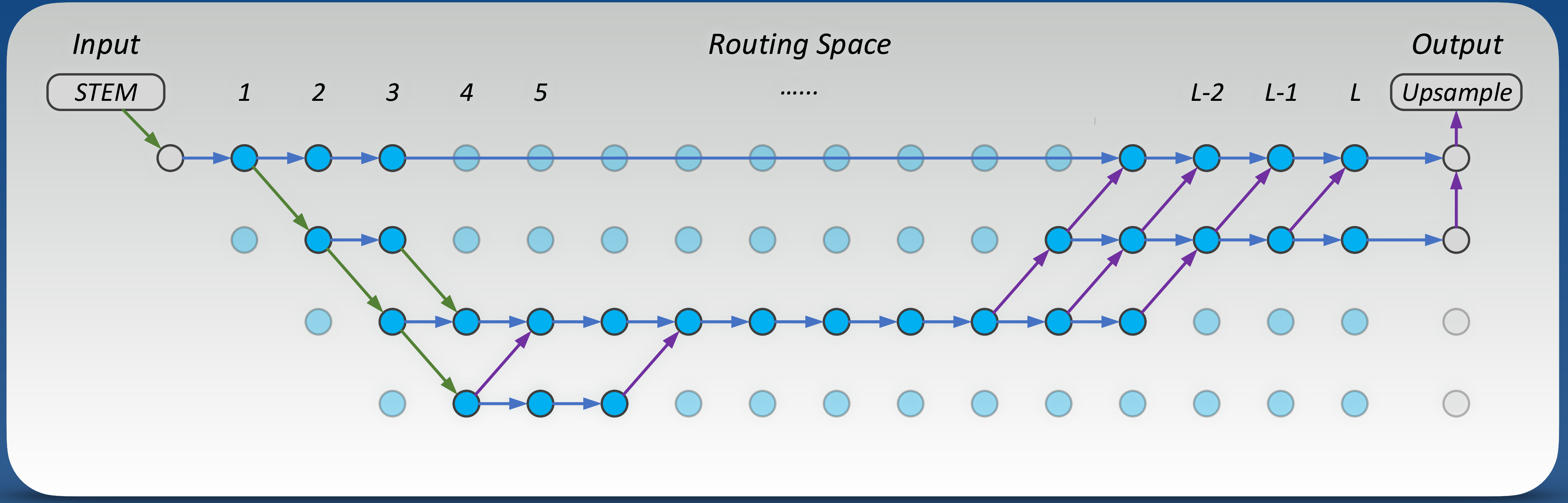
Ablation Studies

We compare with several classic architectures under similar FLOPs, which are modeled on the same routing space.

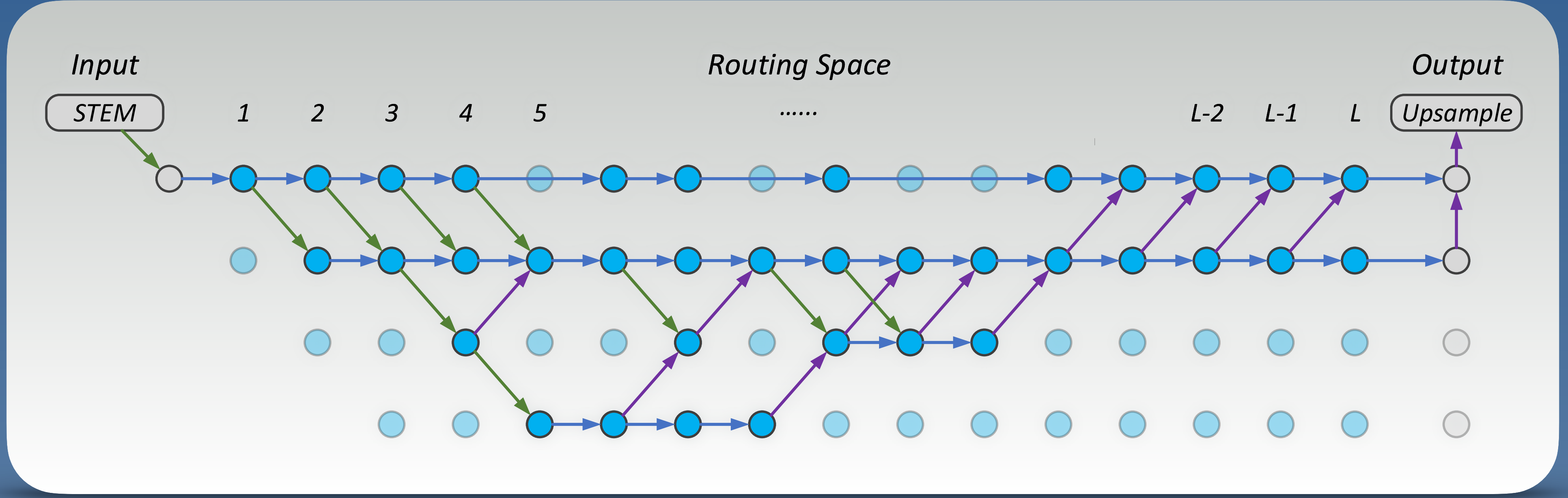
Actually, some paths are always kept with different inputs. The paths, which are preserved over 95% inferences, are defined as Common networks.

Comparisons with classic architectures on the Cityscapes val set

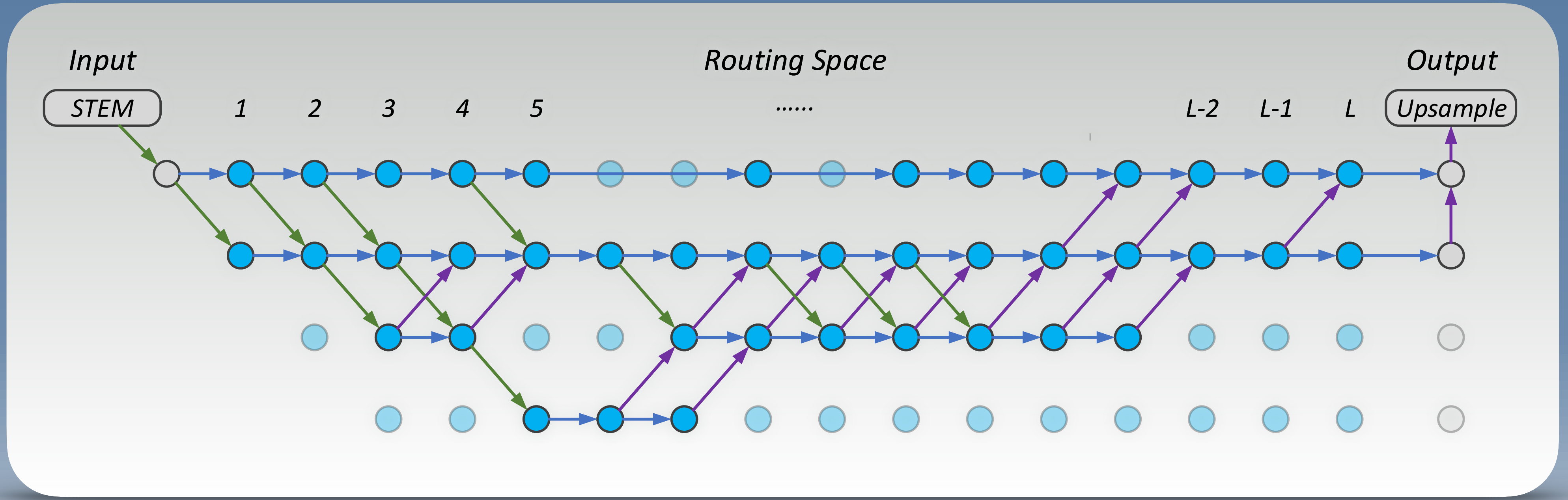
Method	Dynamicc	Modeled from	mIoU (%)	$FLOPS_{Avg}$ (G)	$FLOPS_{Max}$ (G)	$FLOPS_{Min}$ (G)
Handcrafted	×	FCN-32s	66.9	35.1	35.1	35.1
	×	DeepLab V3	67.0	42.5	42.5	42.5
	×	U-Net	71.6	53.9	53.9	53.9
	×	HRNetV2	72.5	62.5	62.5	62.5
Searched	×	Auto-DeepLab	67.2	33.1	33.1	33.1
Common-A	×	Dynamic-A	71.6	41.6	41.6	41.6
Common-B	×	Dynamic-B	73.0	53.7	53.7	53.7
Common-C	×	Dynamic-C	73.2	57.1	57.1	57.1
Dynamic-A	✓	**Routing-Space**	**72.8**	**44.9**	**48.2**	**43.5**
Dynamic-B	✓	**Routing-Space**	**73.8**	**58.7**	**63.5**	**56.8**
Dynamic-C	✓	**Routing-Space**	**74.6**	**66.6**	**71.6**	**64.3**



Network architecture of Common-A



Network architecture of Common-B

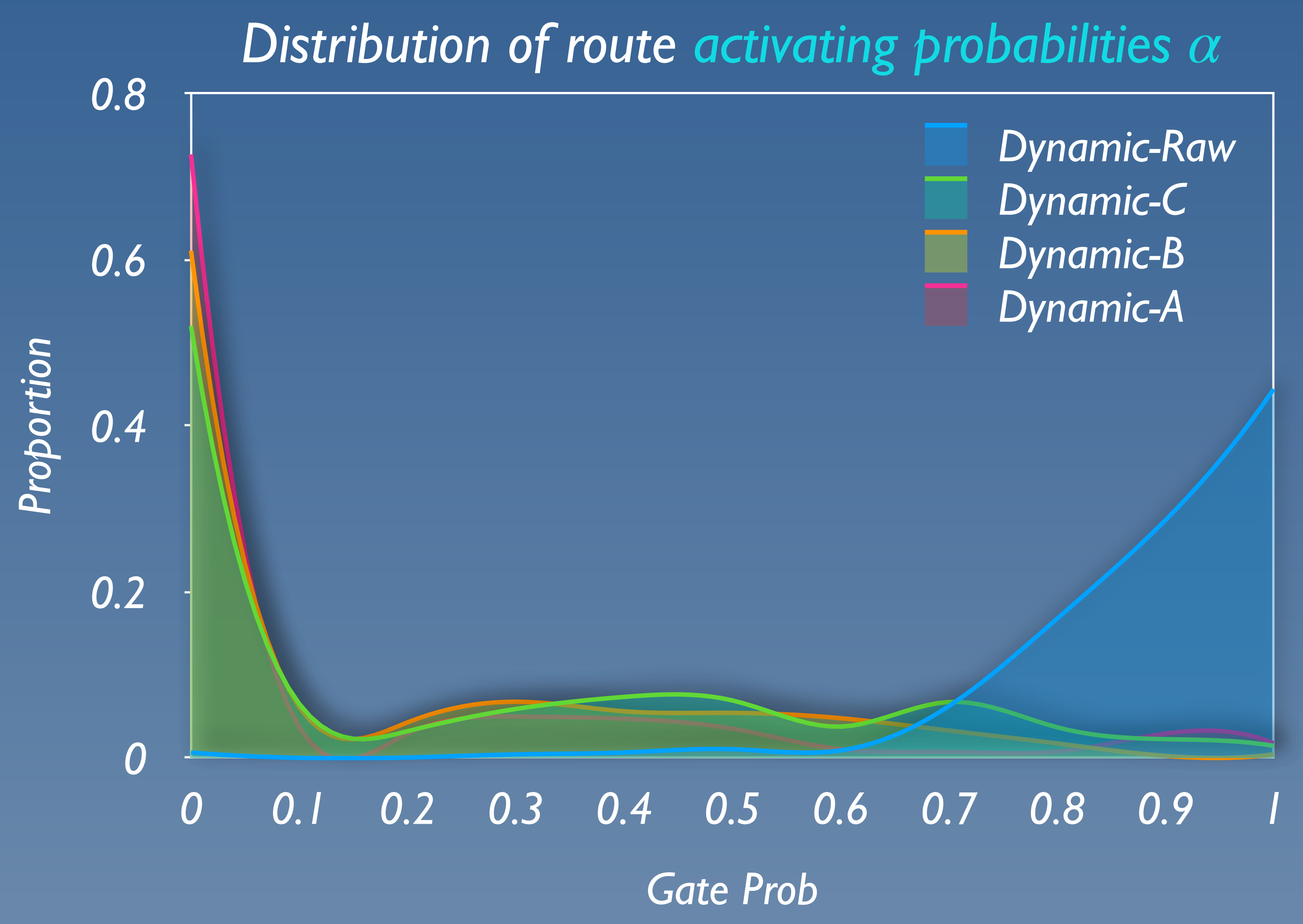


Network architecture of Common-C

Experiments

Visualization

Most of the paths tend to be preserved in Dynamic-Raw.
Different proportions of routes will be dropped if given budgets.

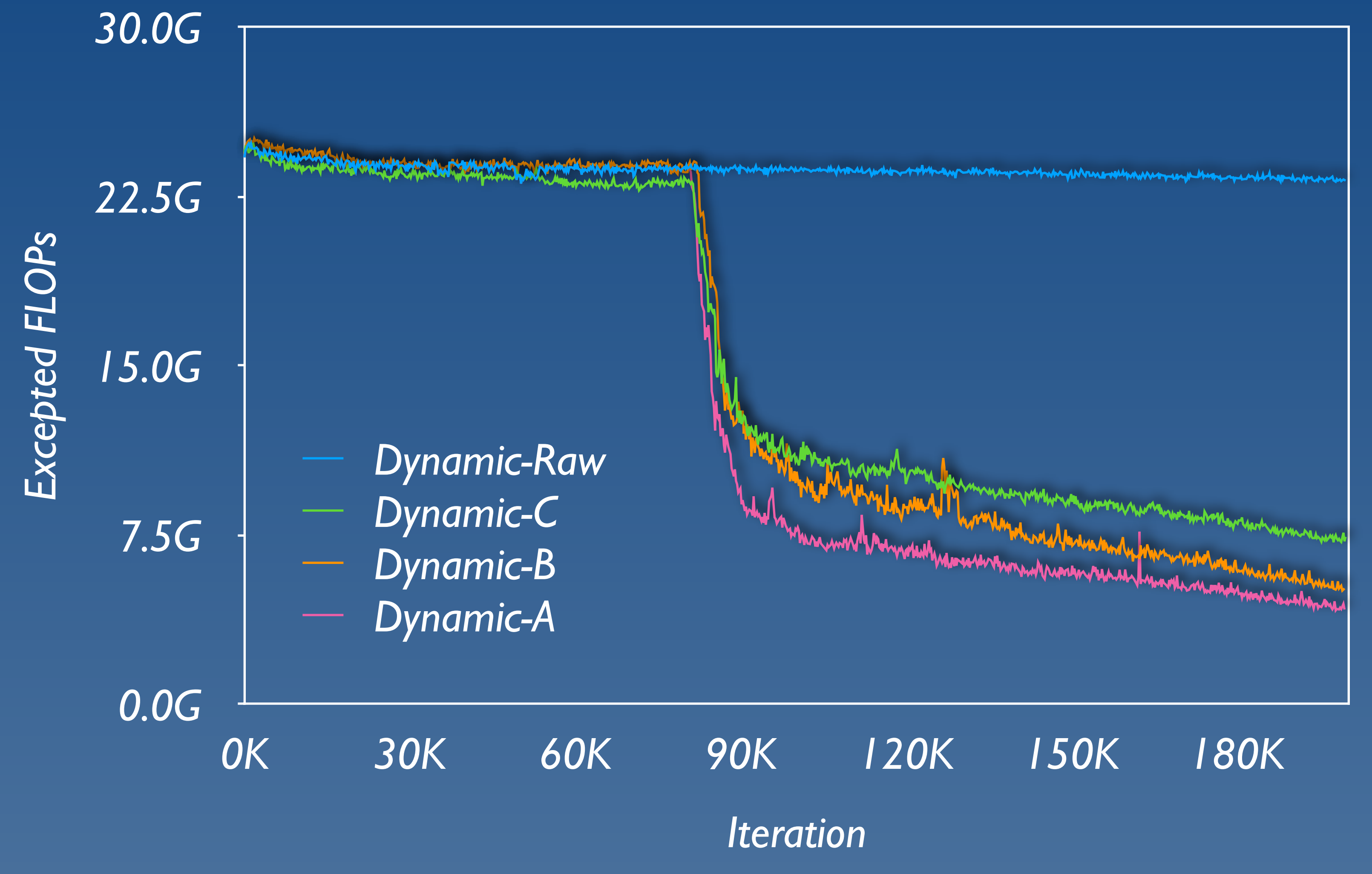


Visualization

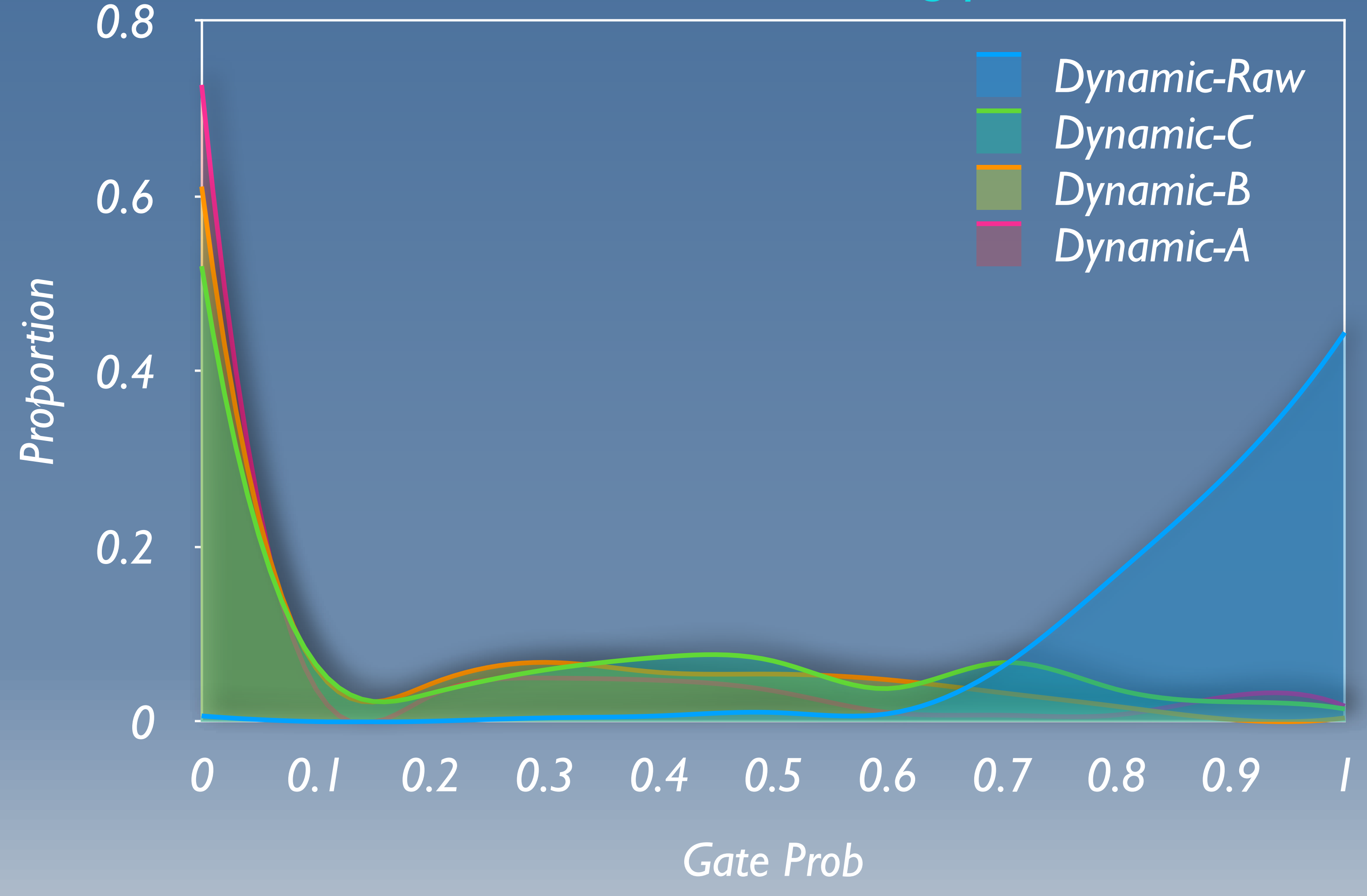
Most of the paths tend to be preserved in Dynamic-Raw.
Different proportions of routes will be dropped if given budgets.

The expected FLOPs $\mathscr{C}(\text{Space})$ and the resource cost $\mathscr{L}_{\text{C}}$ will be optimized steadily with different budget constraints.

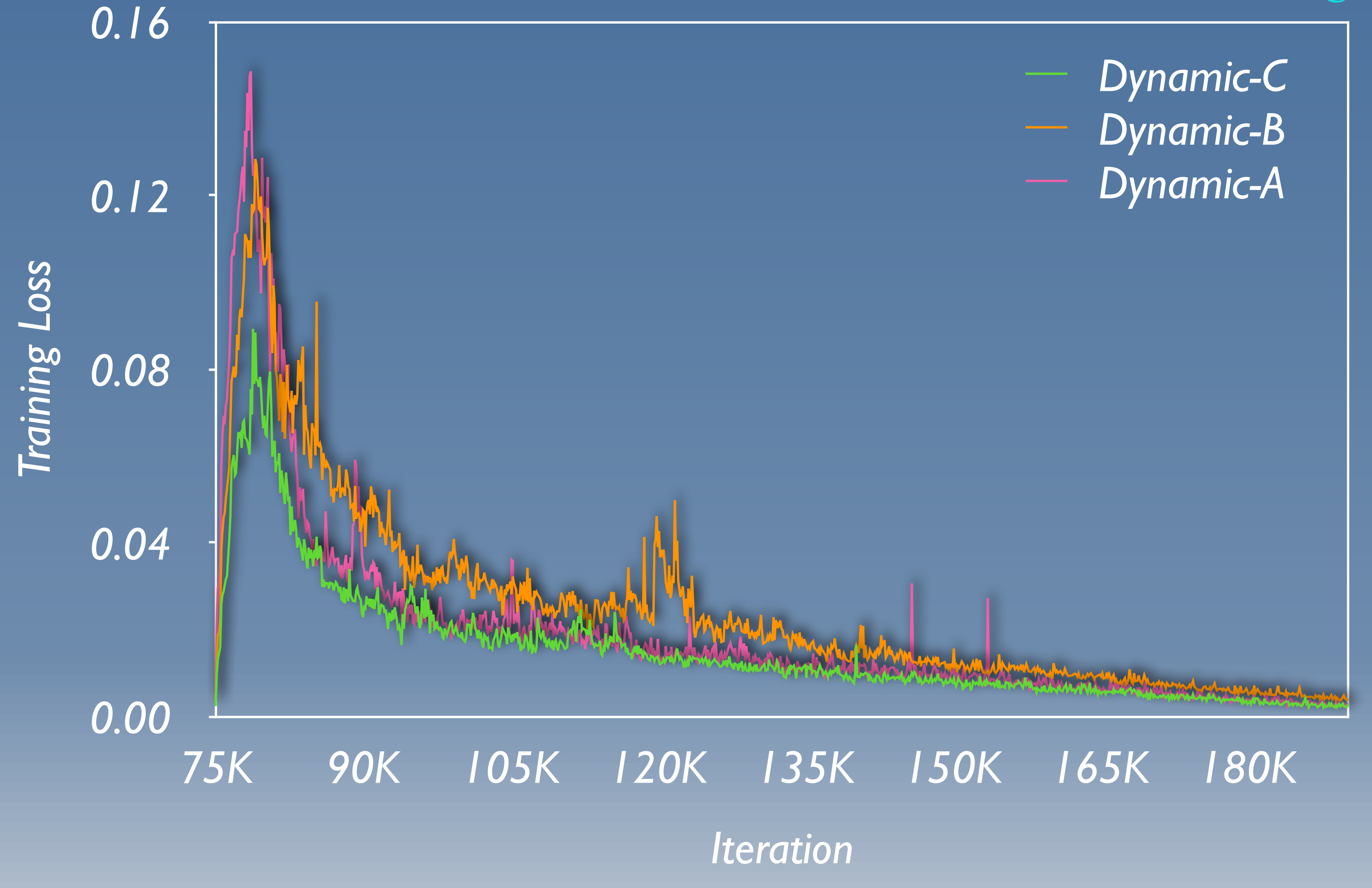
The optimization process of the expected FLOPs $\mathscr{C}(\text{Space})$



Distribution of route activating probabilities α



The optimization process of the resource cost $\mathscr{L}_{\text{C}}$



Results

Compared with previous works, the proposed Dynamic Routing achieve similar performance with much less resource consumption.

Comparisons with others on Cityscapes dataset with input size 1024x2048.

Method	backbone	$mIoU_{test}$ (%)	$mIoU_{val}$ (%)	FLOPS (G)
BiSeNet	ResNet-18	77.7	74.8	98.3
DeepLab V3	ResNet-101-ASPP	-	78.5	1778.7
DeepLab V3+	Xception-71-ASPP	-	79.6	1551.1
PSPNet	ResNet-101-PSP	78.4	79.7	2017.6
Auto-DeepLab*	Searched-F20-ASPP	79.9	79.7	333.3
Auto-DeepLab*	Searched-F48-ASPP	80.4	80.3	695.0
Dynamic*	*Layer16*	*79.1*	*78.3*	*111.7*
Dynamic	*Layer16*	*79.7*	*78.6*	*119.4*
Dynamic	*Layer33*	*80.0*	*79.2*	*242.3*
Dynamic	*Layer33-PSP*	***80.7***	***79.7***	***270.0***

Comparisons with others on PASCAL VOC 2012 dataset with input size 512x512.

Method	backbone	$mIoU_{test}$ (%)	$mIoU_{val}$ (%)	FLOPS (G)
DeepLab V3	MobileNet-ASPP	-	75.3	14.3
DeepLab V3	MobileNetV2-ASPP	-	75.7	5.8
Auto-DeepLab	Searched-F20-ASPP	82.5	78.3	41.7
Dynamic	*Layer16*	*82.8*	*78.6*	*14.9*
Dynamic	*Layer33*	***84.0***	***79.0***	***30.8***

Thanks

For more questions, please contact

www.yanwei-li.com
liyanwei2017@ia.ac.cn

Paper | **Code**

https://arxiv.org/abs/2003.10401

https://github.com/yanwei-li/DynamicRouting