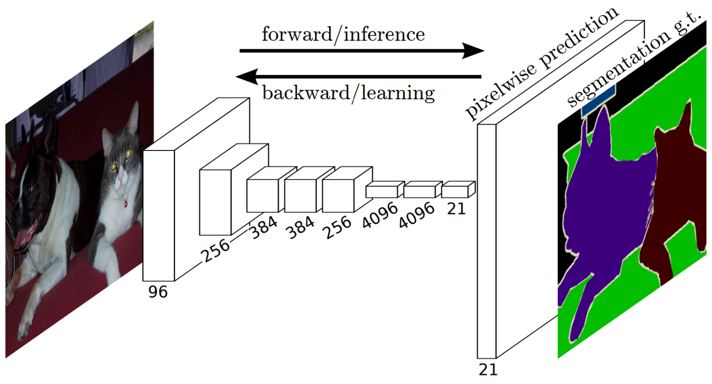
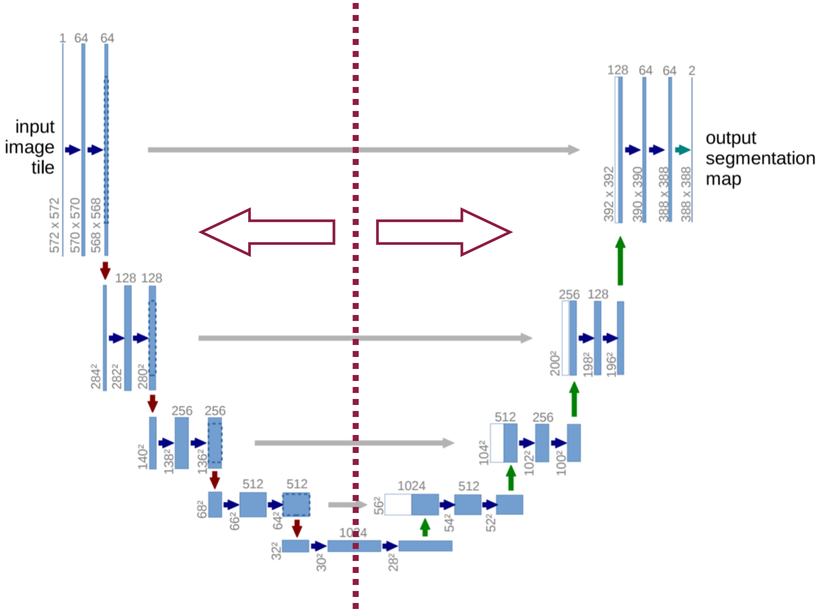
UNet++: A Nested U-Net Architecture for Medical Image Segmentation

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang

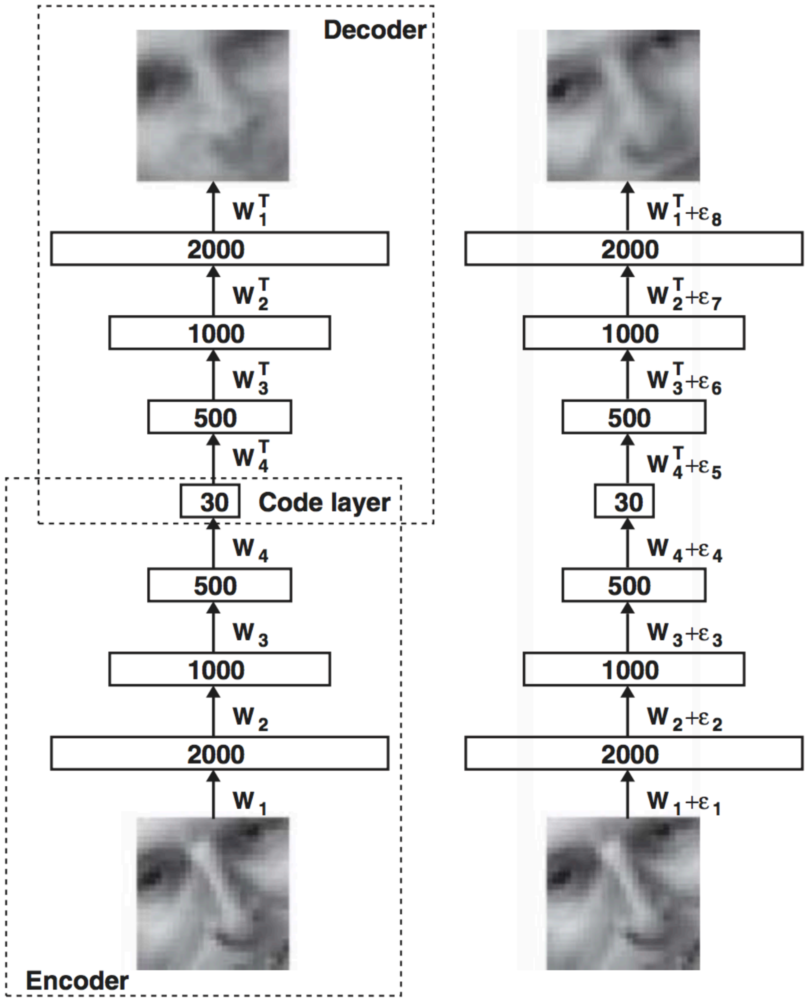
Arizona State University





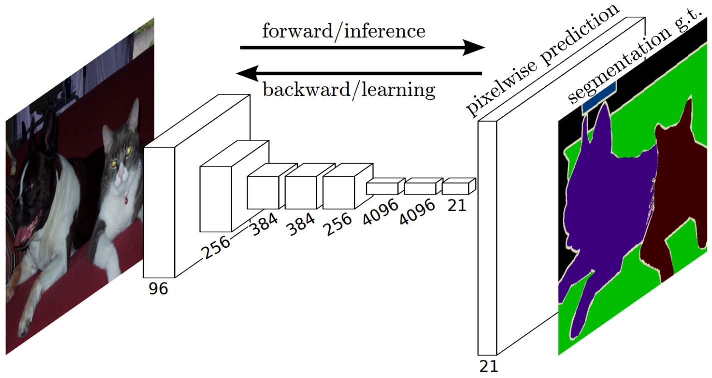
(Ronneberger, 2015)

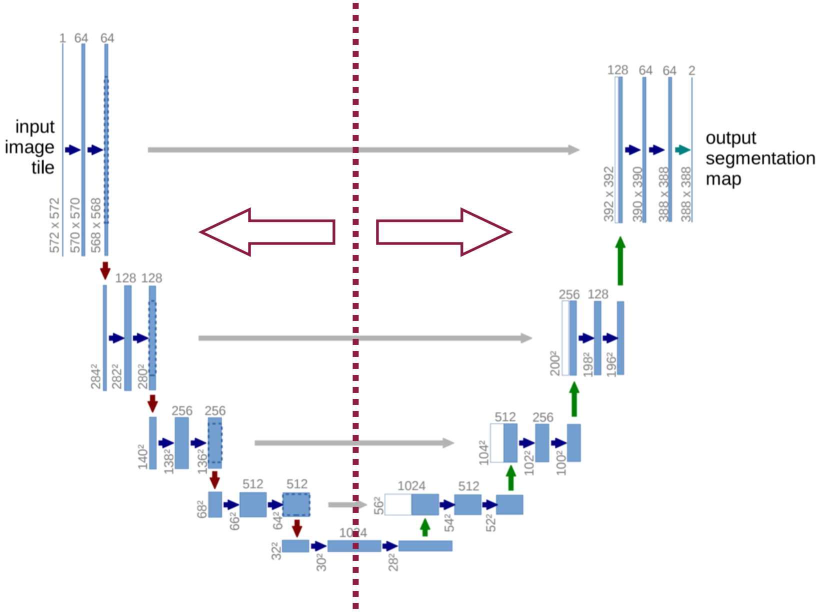
(Long, 2015)



Unrolling

Fine-tuning

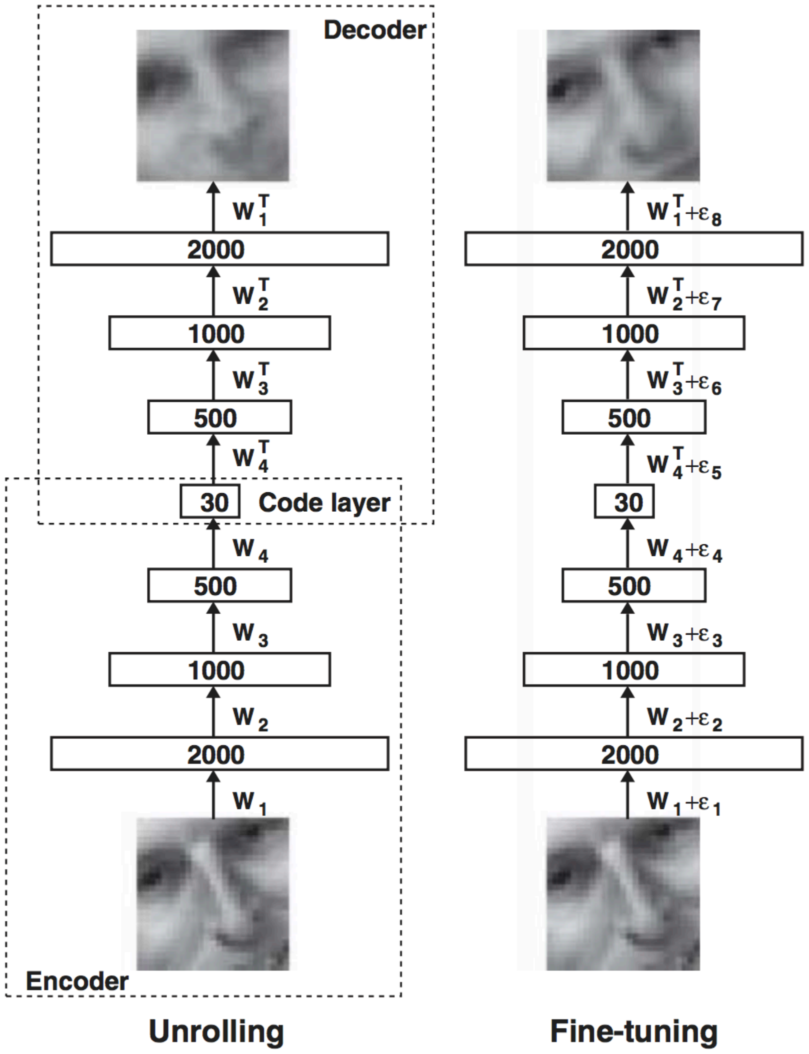


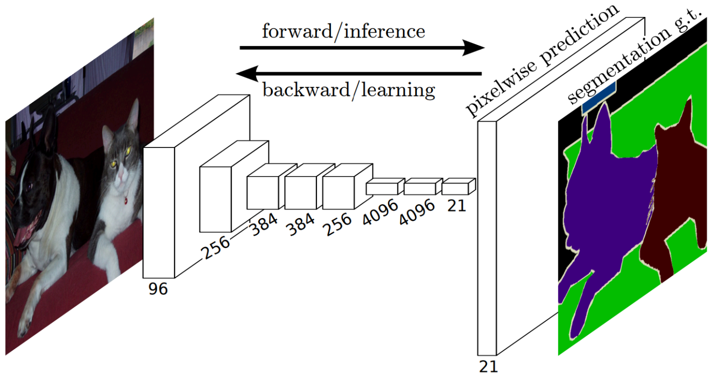


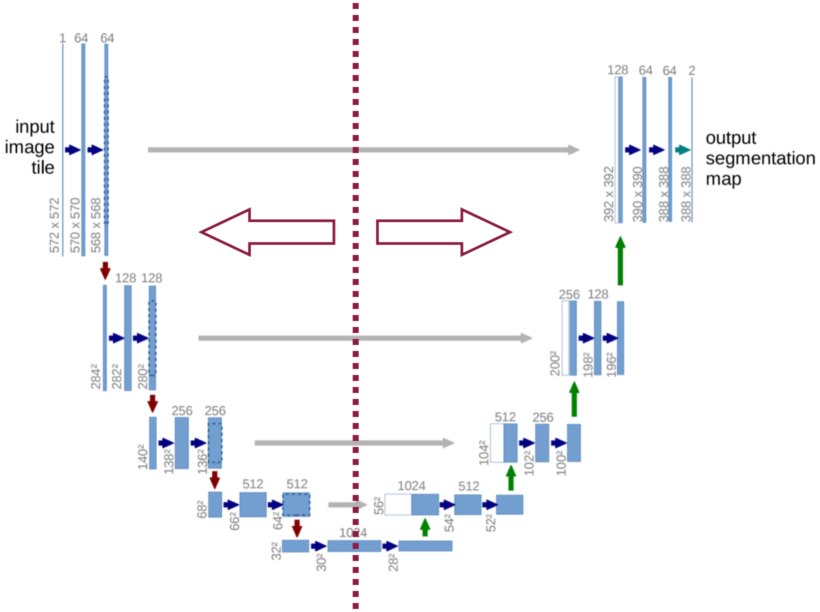
(Ronneberger, 2015)

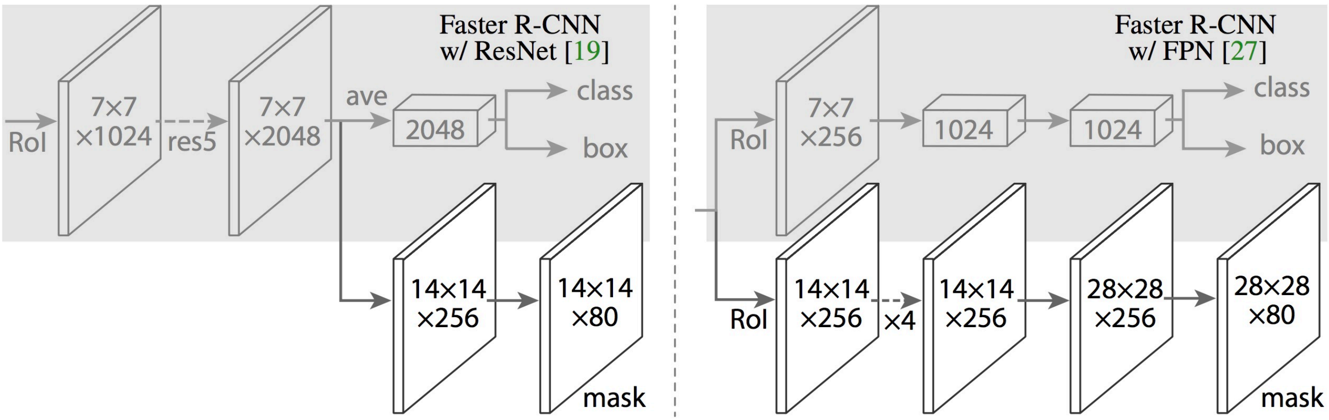
(Hinton, 2006)

(Long, 2015)



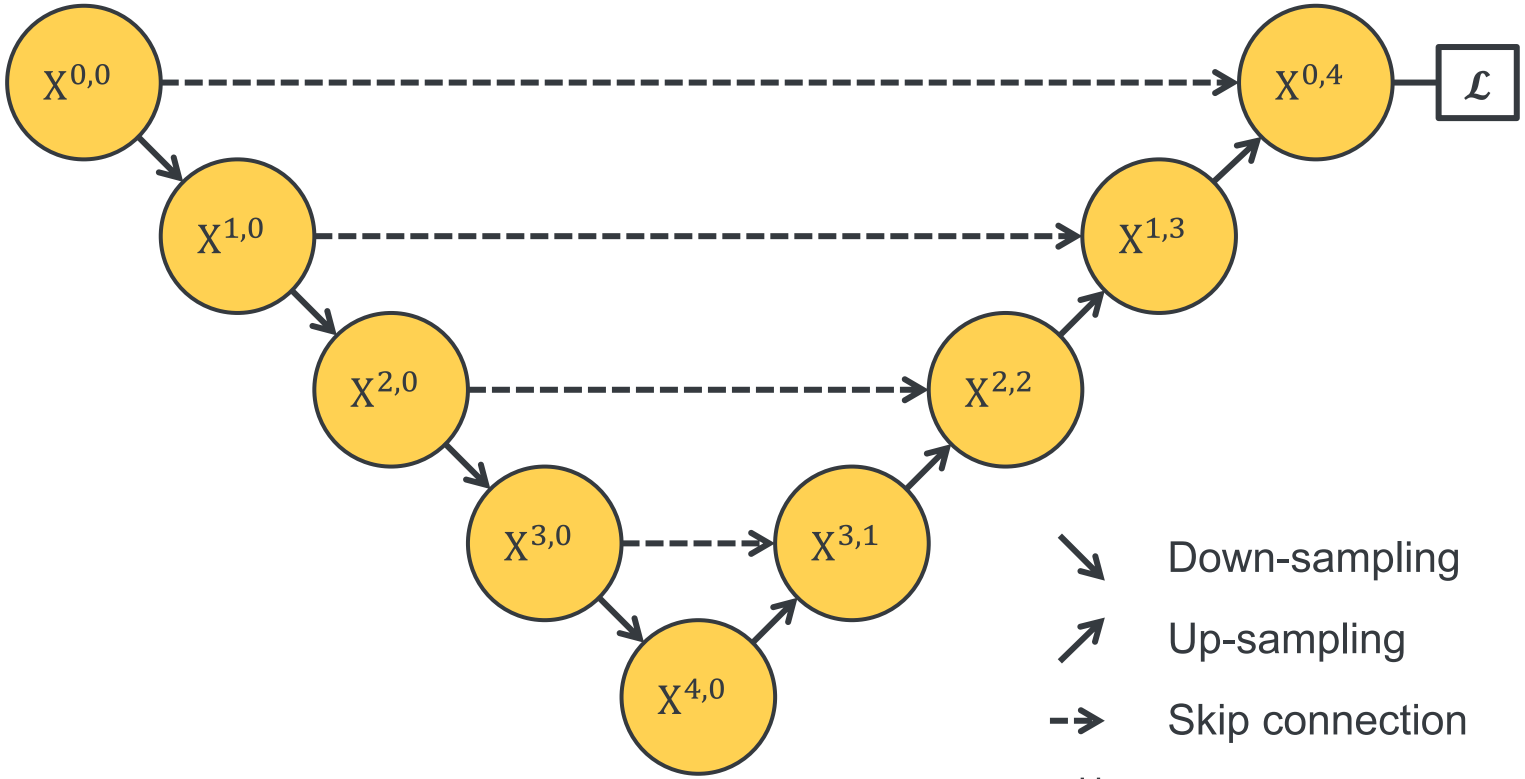


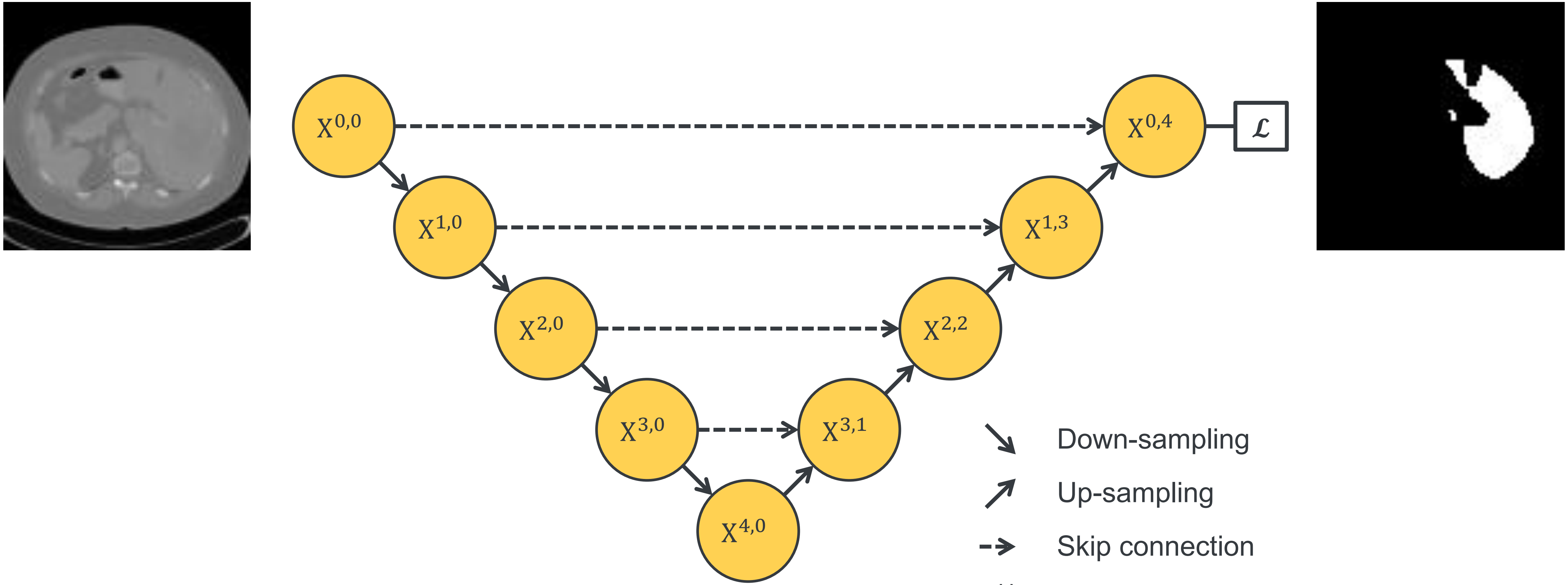


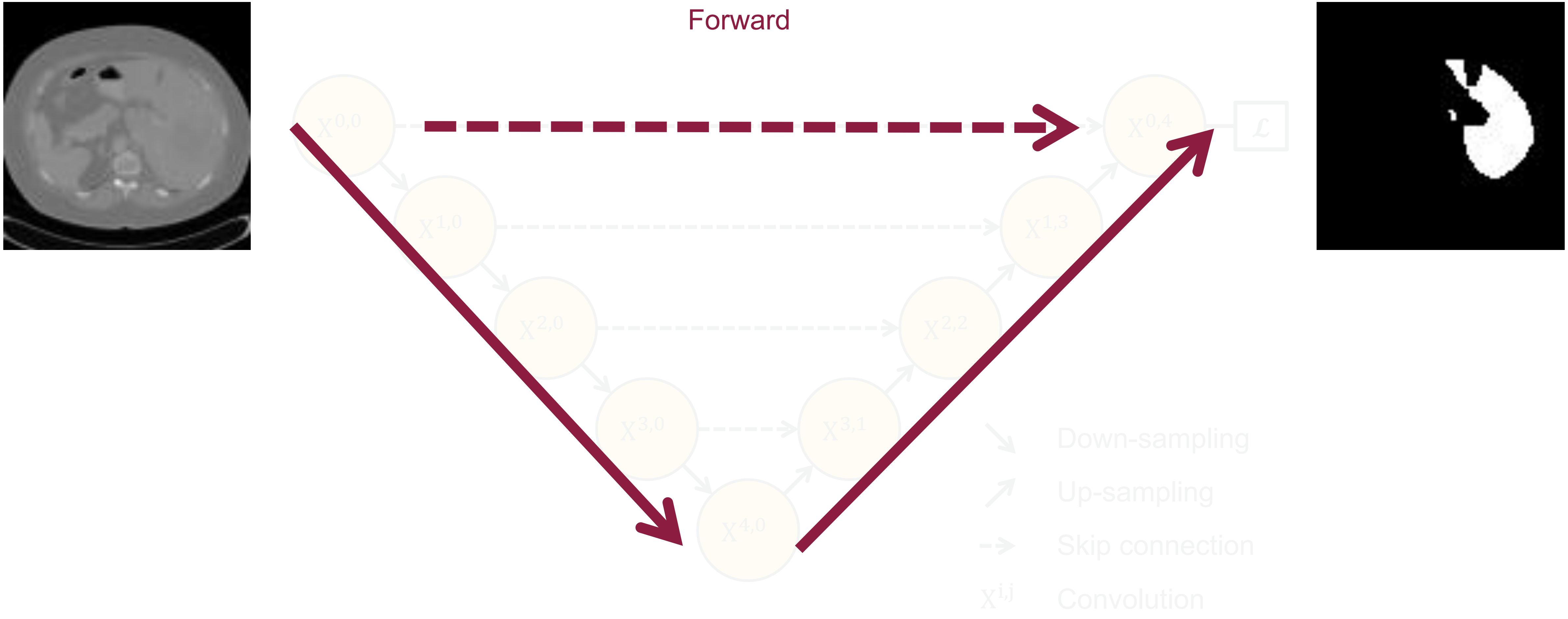


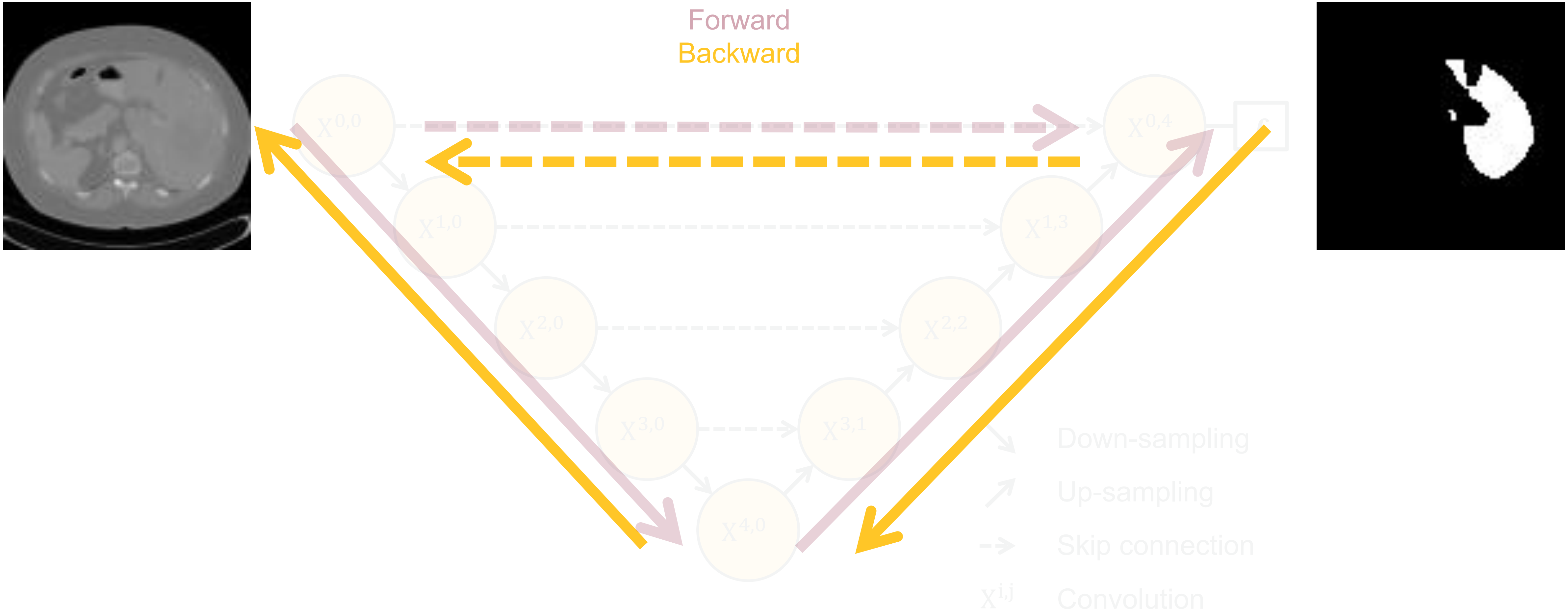
(Ronneberger, 2015)

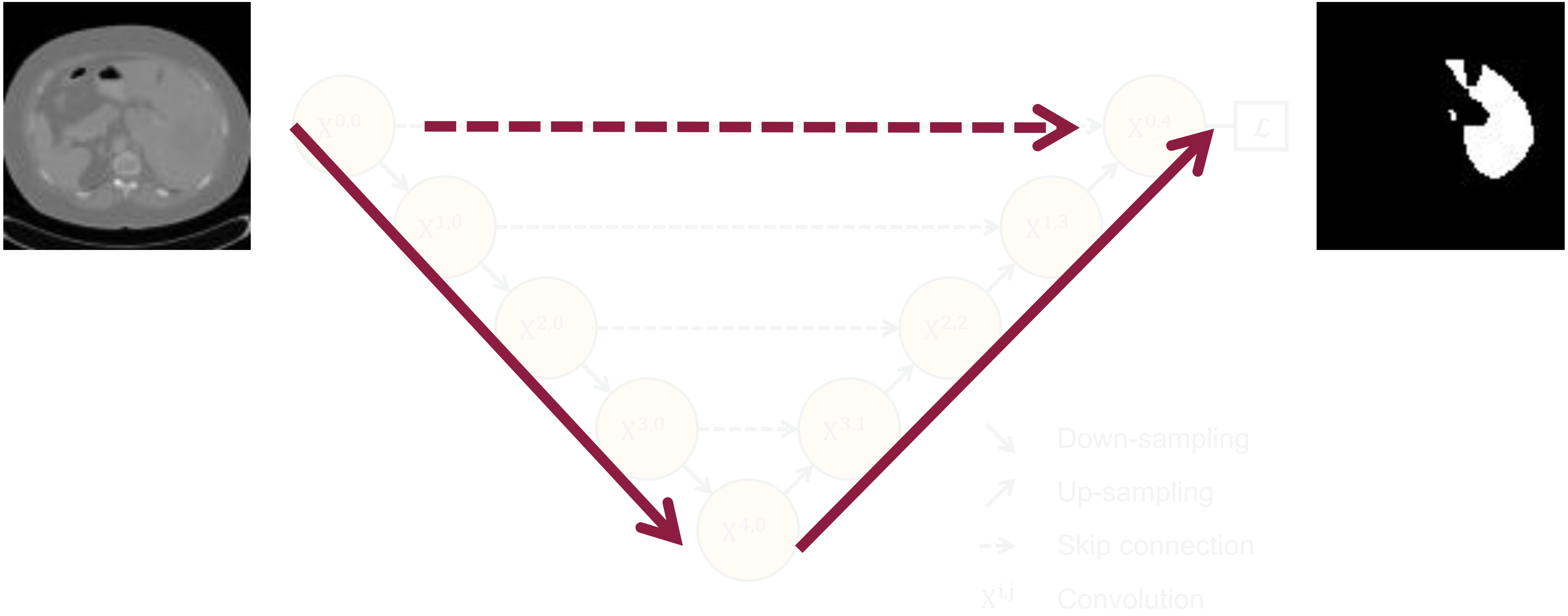
(Hinton, 2006) (Long, 2015) (He, 2017)

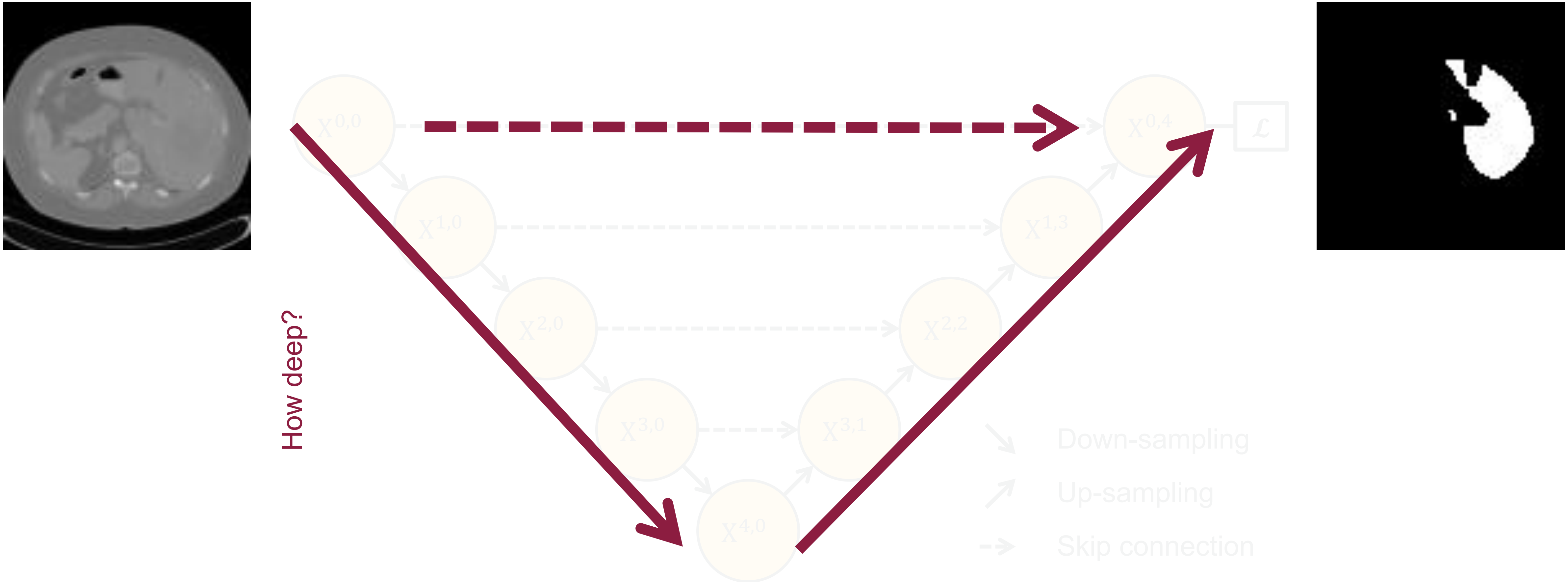

$X^{0,0}$
$X^{1,0}$
$X^{2,0}$
$X^{3,0}$
$X^{4,0}$
$X^{3,1}$
$X^{2,2}$
$X^{1,3}$
$X^{0,4}$
$\mathcal{L}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

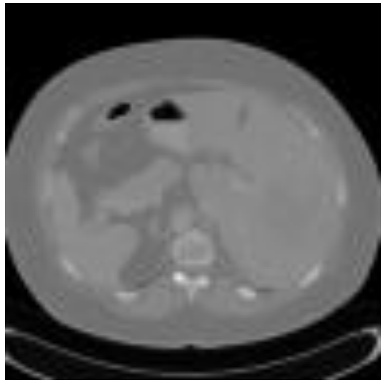

$X^{0,0}$
$X^{1,0}$
$X^{2,0}$
$X^{3,0}$
$X^{4,0}$
$X^{3,1}$
$X^{2,2}$
$X^{1,3}$
$X^{0,4}$
$\mathcal{L}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

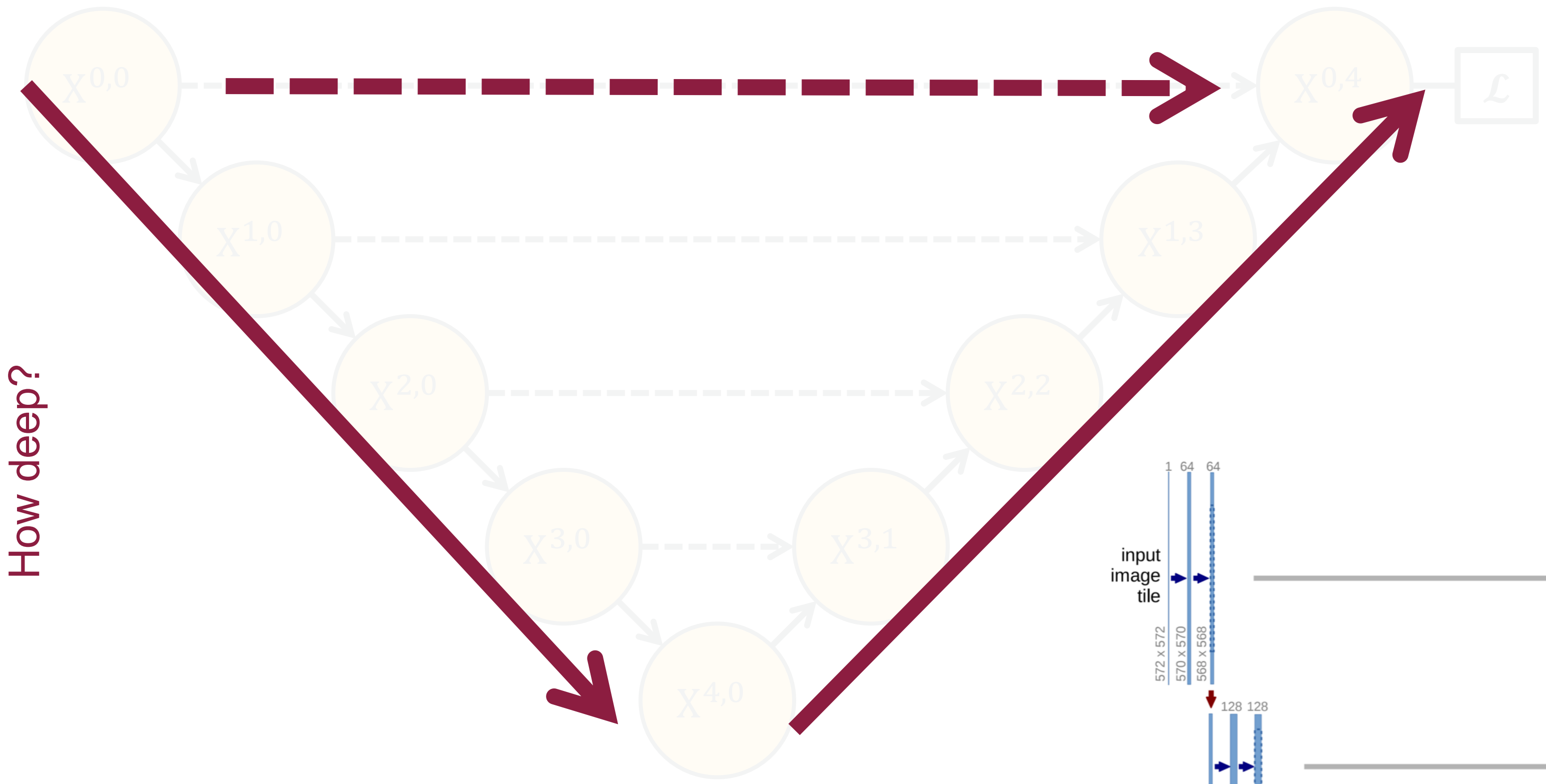

Forward
Down-sampling
Up-sampling
Skip connection
Convolution


Forward
Backward
$X^{0,0}$
$X^{1,0}$
$X^{2,0}$
$X^{3,0}$
$X^{4,0}$
$X^{3,1}$
$X^{2,2}$
$X^{1,3}$
$X^{0,4}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

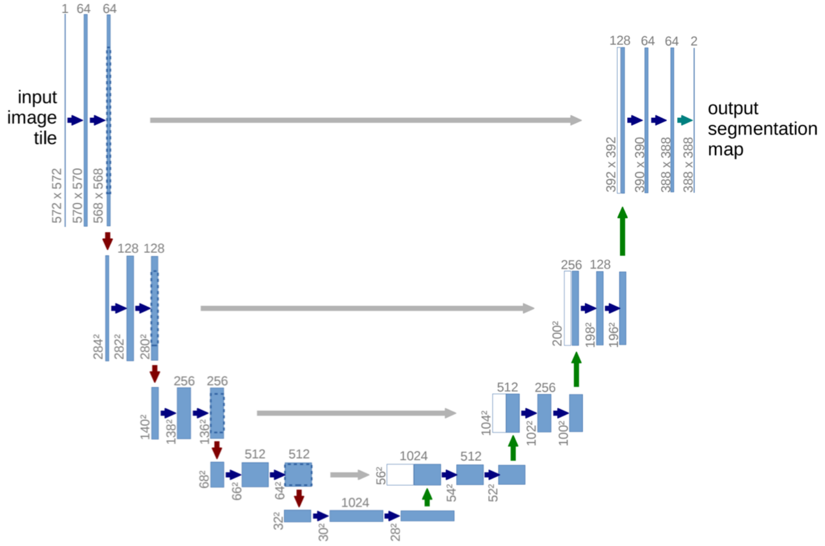

Down-sampling
Up-sampling
Skip connection
Convolution


How deep?
X0,0
X1,0
X2,0
X3,0
X4,0
X3,1
X2,2
X1,3
X0,4
Down-sampling
Up-sampling
Skip connection
Convolution

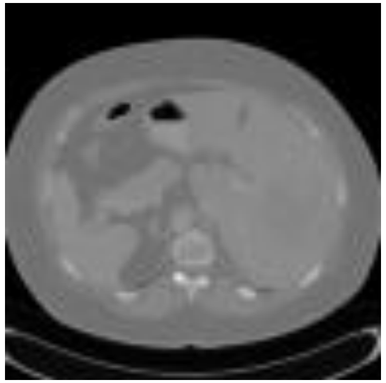




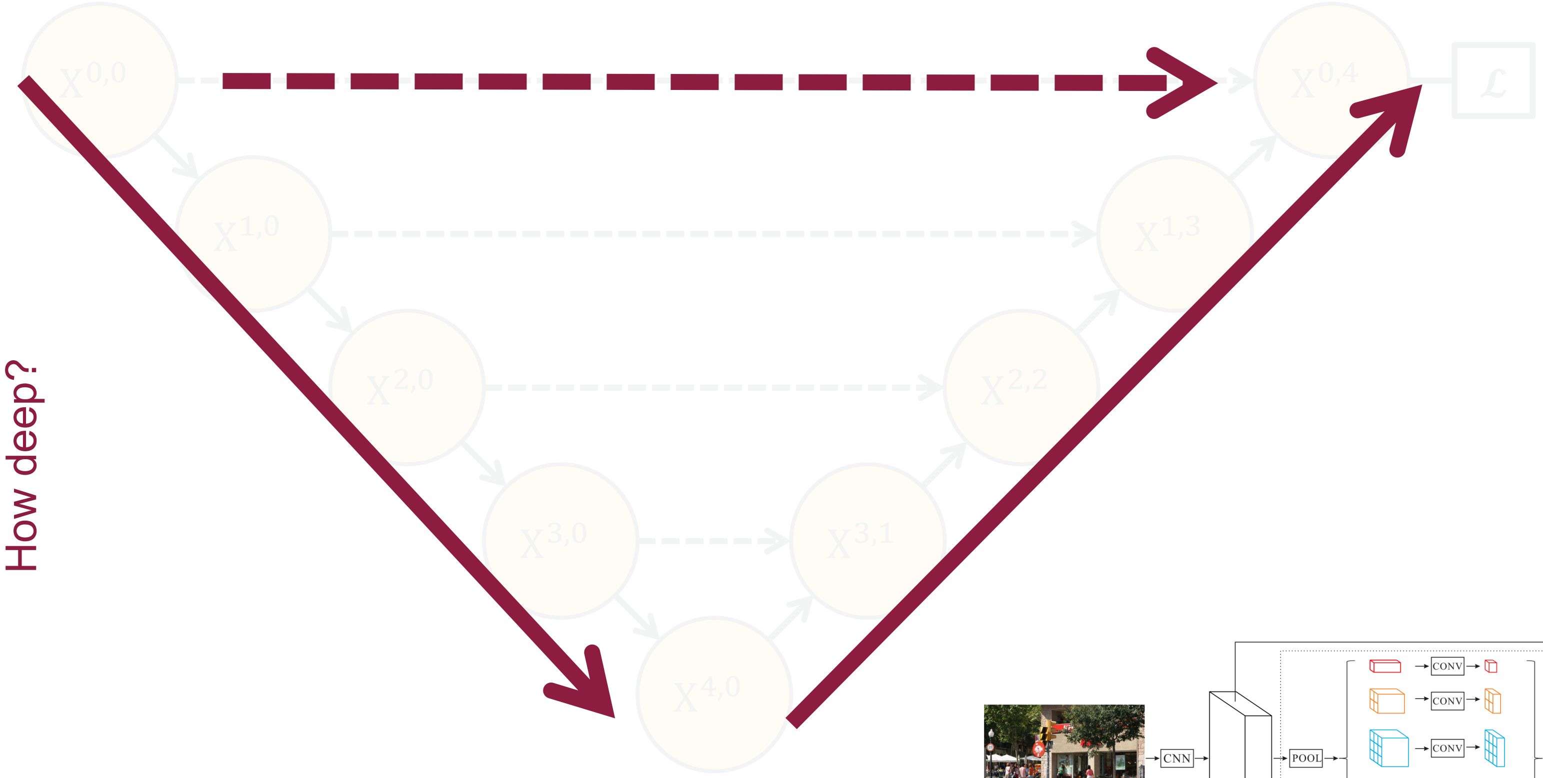
$X^{0,0}$
$X^{1,0}$
$X^{2,0}$
$X^{3,0}$
$X^{4,0}$
$X^{3,1}$
$X^{2,2}$
$X^{1,3}$
$X^{0,4}$
$\mathcal{L}$
How deep?

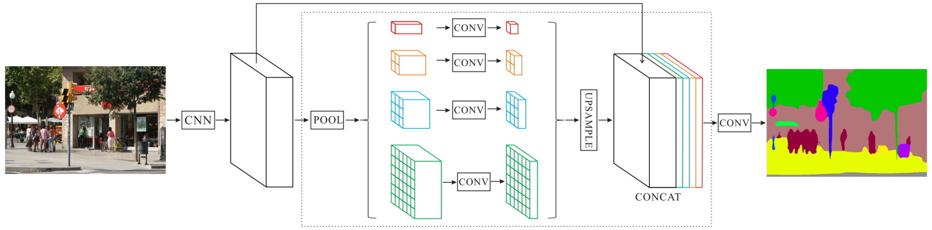


input image tile
output segmentation map
572 x 572
570 x 570
568 x 568
392 x 392
390 x 390
388 x 388
388 x 388

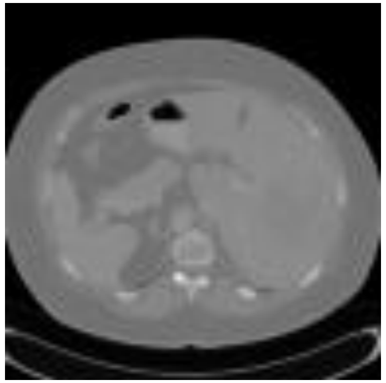


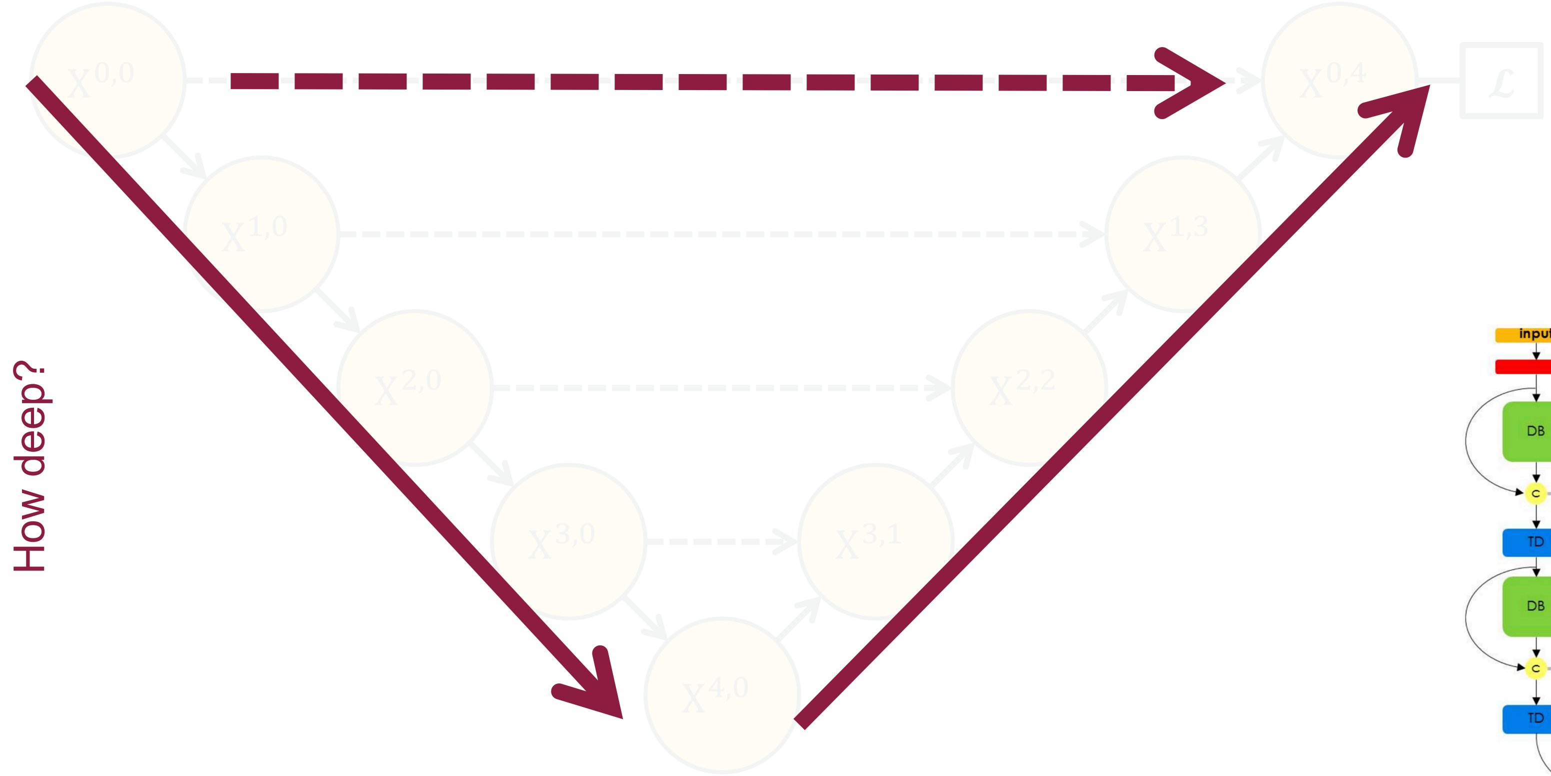
How deep?

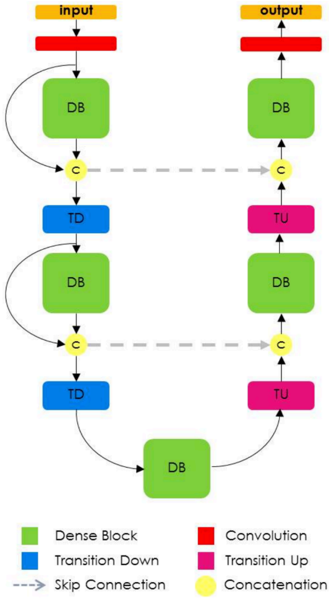




(Zhao, 2017)



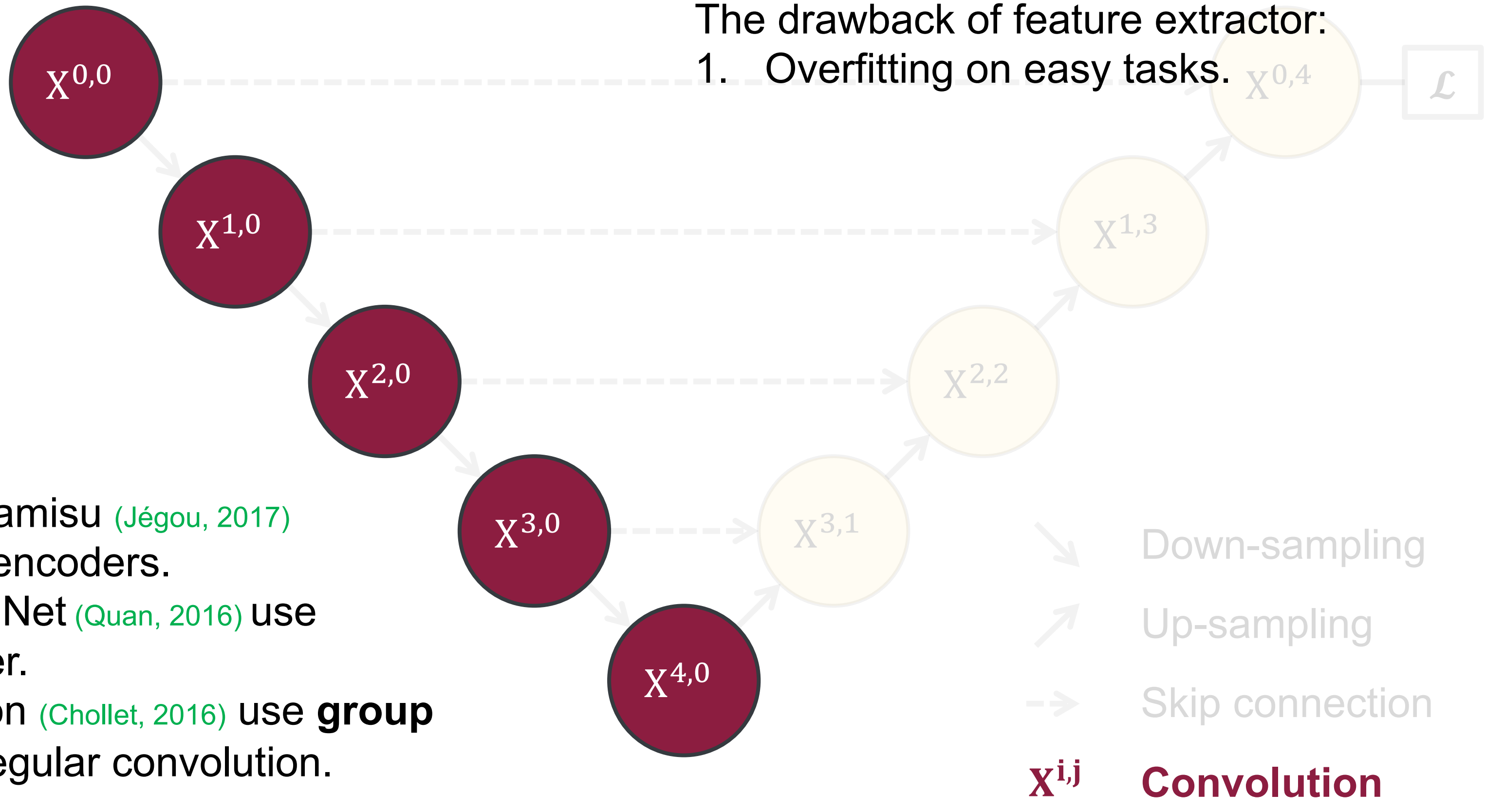

$X^{0,0}$
$X^{1,0}$
$X^{2,0}$
$X^{3,0}$
$X^{4,0}$
$X^{3,1}$
$X^{2,2}$
$X^{1,3}$
$X^{0,4}$
$\mathcal{L}$
How deep?


input
output
DB
C
TD
TU
Dense Block
Transition Down
Skip Connection
Convolution
Transition Up
Concatenation

(Jegou, 2017)

The feature extractor is important:
1. Good feature representation.
2. Fast convergence speed.

The drawback of feature extractor:
1. Overfitting on easy tasks.



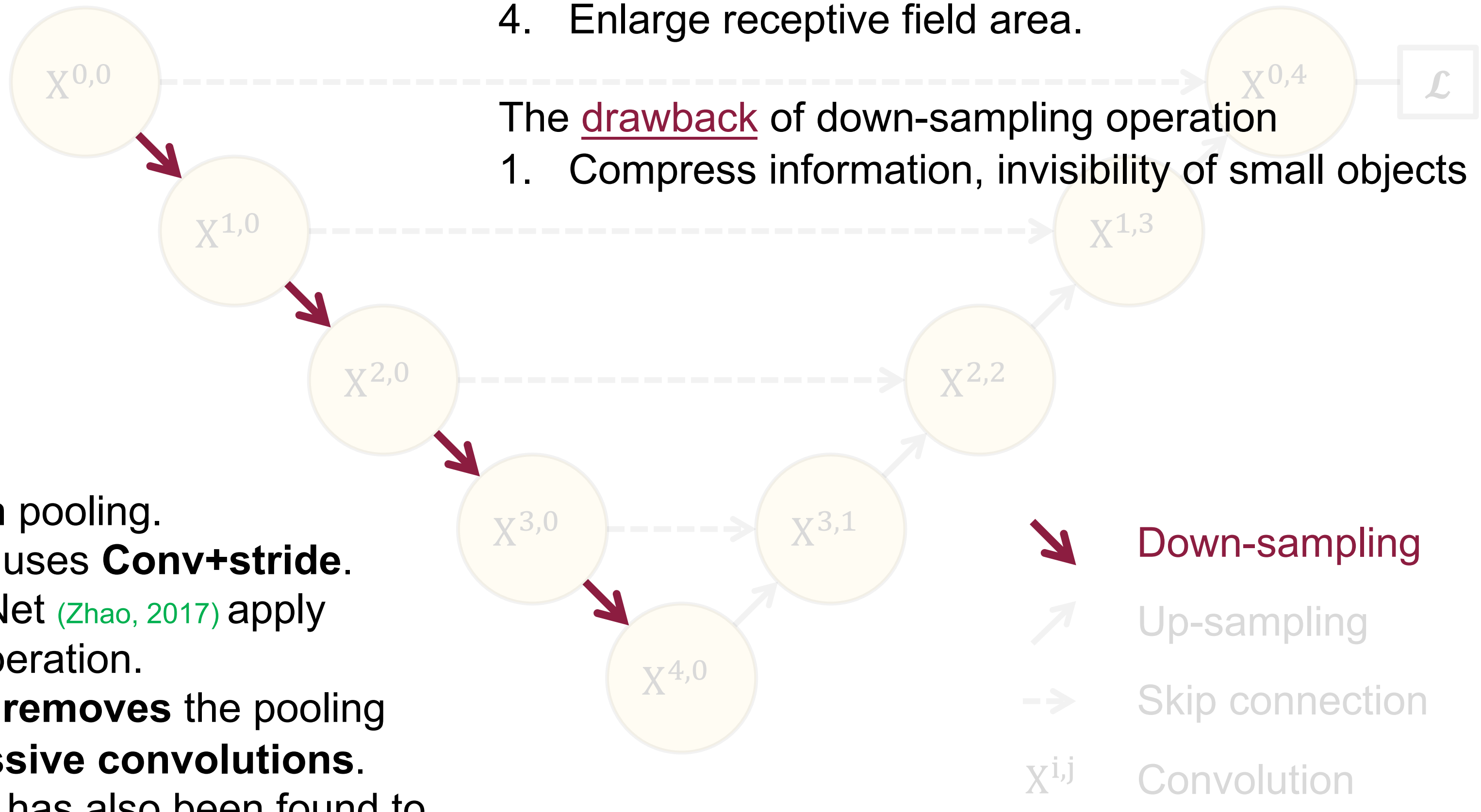
- H-DenseUNet (Li, 2017), Tiramisu (Jégou, 2017) use **dense units** in their encoders.
- PSPNet (Zhao, 2017), FusionNet (Quan, 2016) use **residual units** as encoder.
- ResNext (Xie, 2016), Xception (Chollet, 2016) use **group convolution** instead of regular convolution.

The down-sampling is important:

1. Robust against small input variance.
2. Reduce overfitting.
3. Reduce computation cost.
4. Enlarge receptive field area.

The drawback of down-sampling operation

1. Compress information, invisibility of small objects



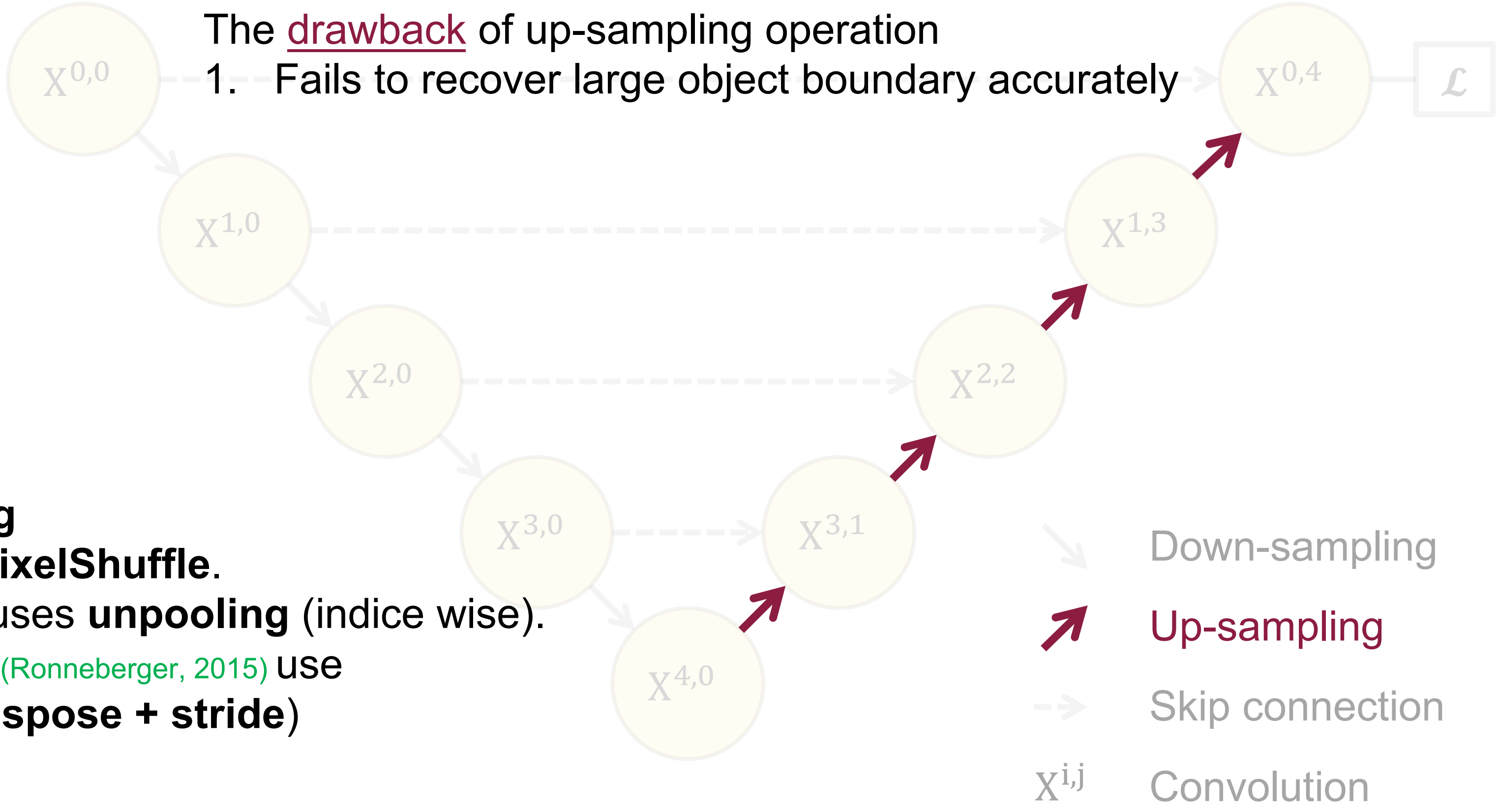
- **Max** vs. **Ave** vs. **L2-norm** pooling.
- ALL-CNN (Springenberg, 2015) uses **Conv+stride**.
- DeepLab (Chen, 2017), PSPNet (Zhao, 2017) apply **dilated convolutional** operation.
- HyperDenseNet (Dolz, 2018) **removes** the pooling layers, only leave **successive convolutions**.
- Discarding pooling layers has also been found to be important in training **VAEs** or **GANs**.

The up-sampling is important:

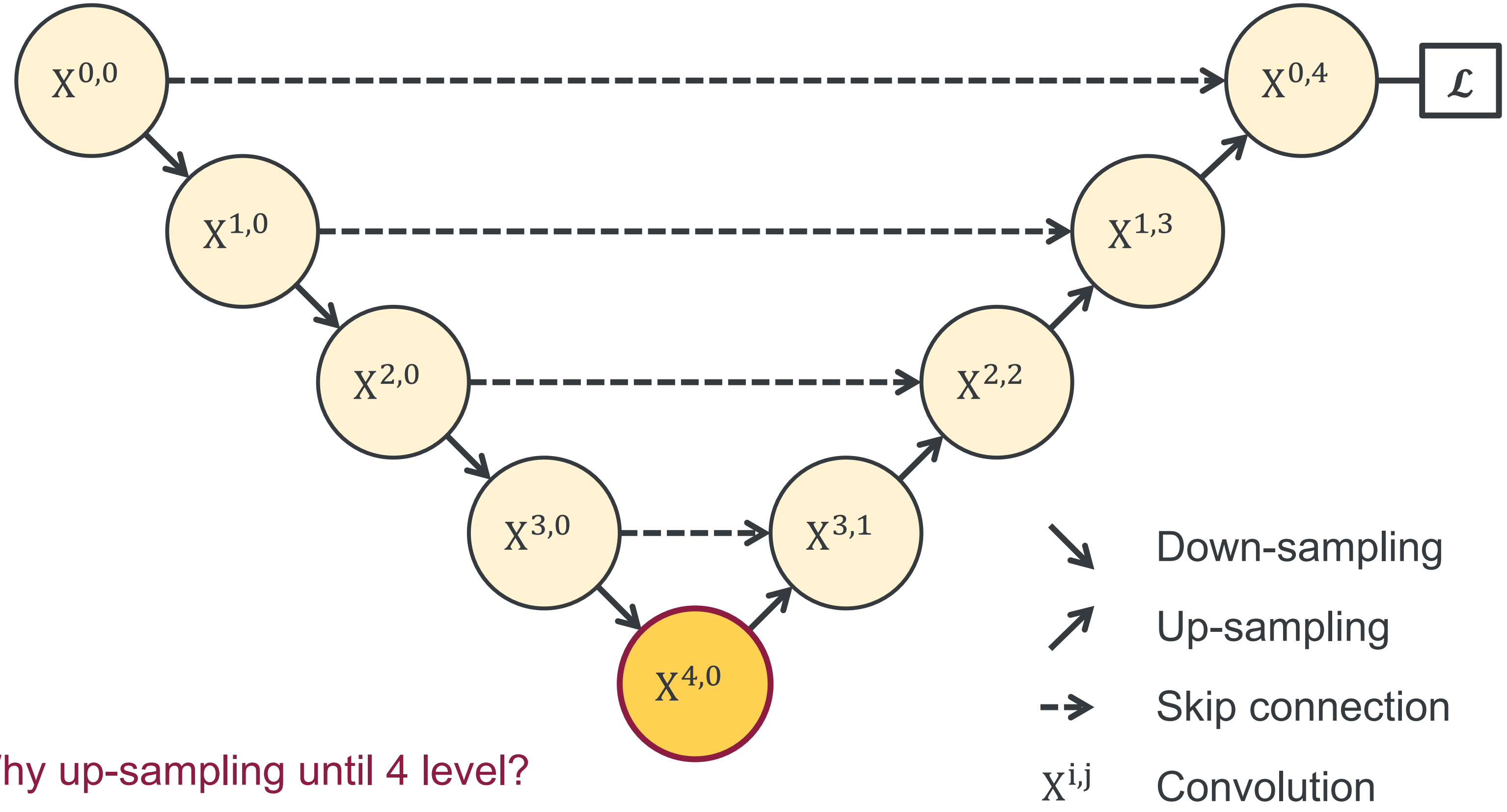
1. Recovers lost resolution in down-sampling
2. Guides encoder to select important information

The drawback of up-sampling operation

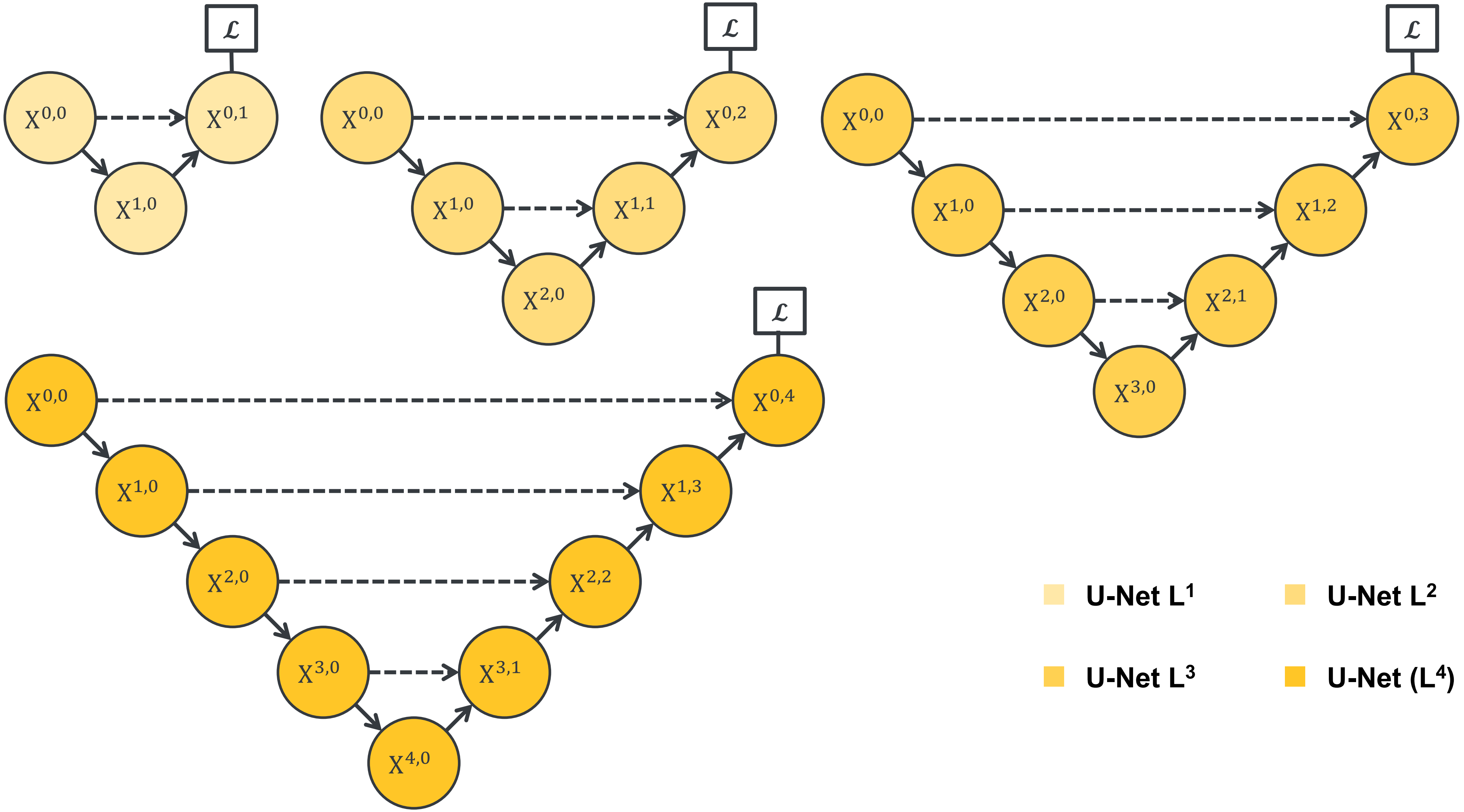
1. Fails to recover large object boundary accurately

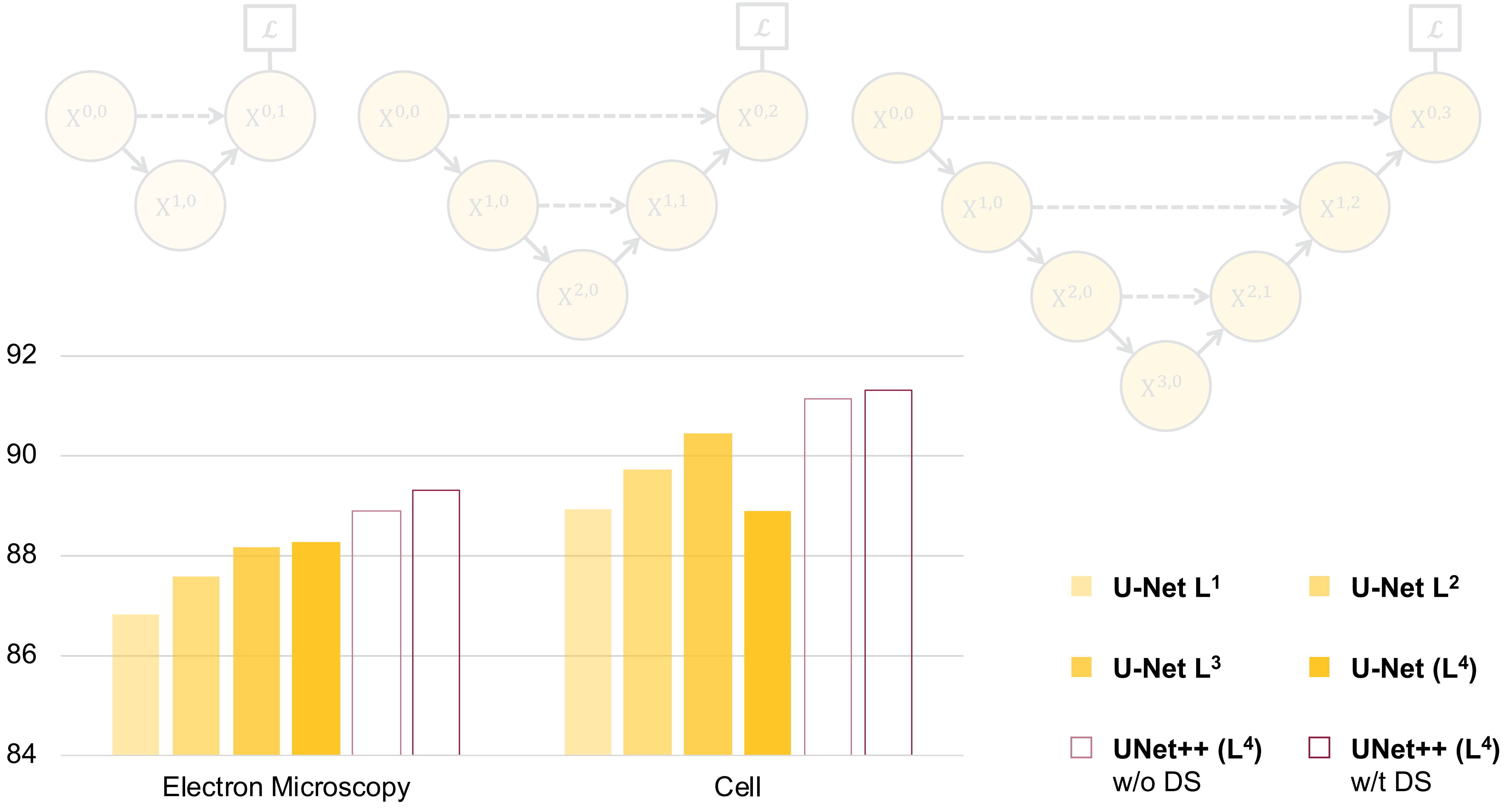


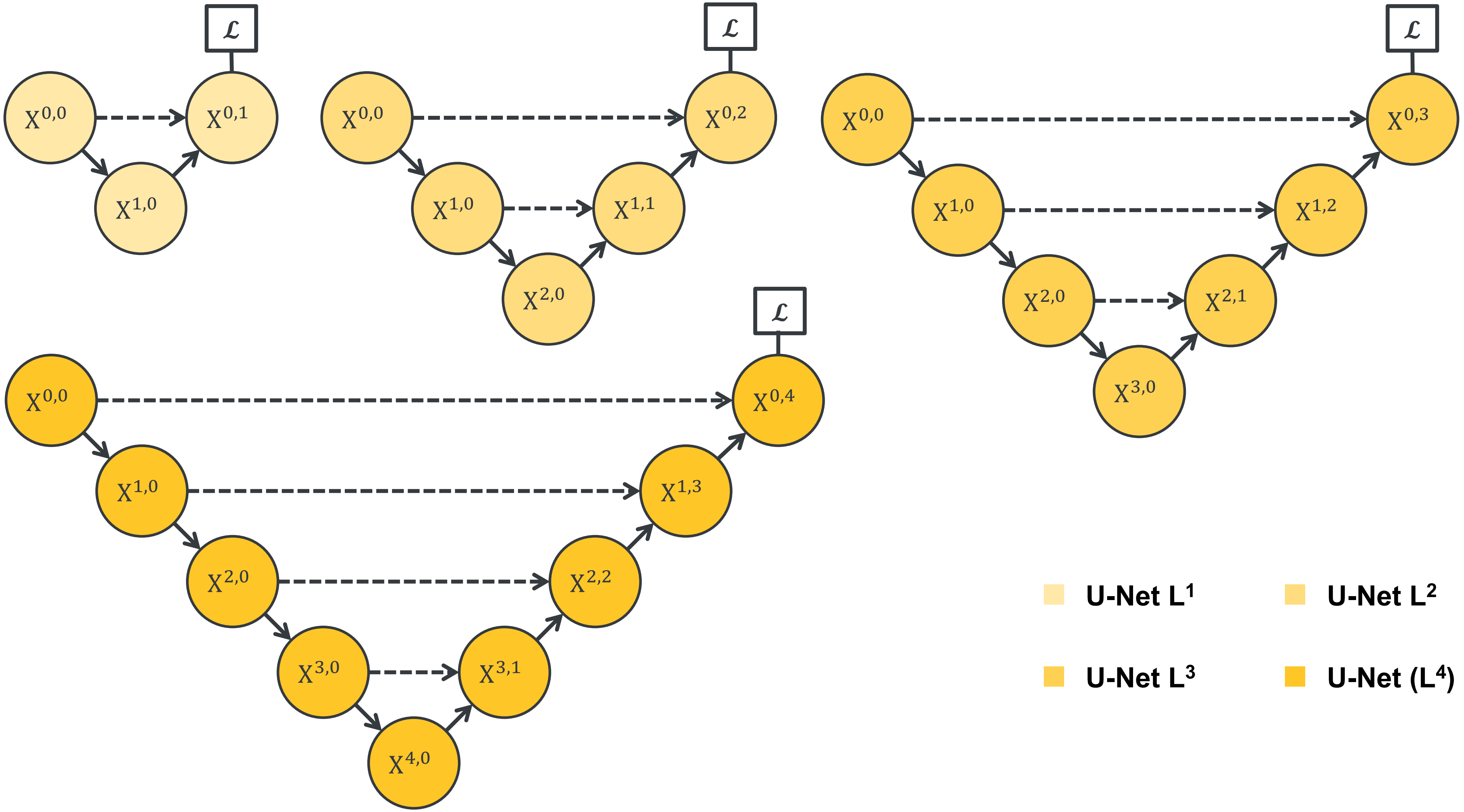
- Up-sampling by **repeating**
- PixelDCL (Gao, 2017) uses **PixelShuffle.**
- SegNet (Badrinarayanan, 2016) uses **unpooling** (indice wise).
- FCN (Long, 2015) and U-Net (Ronneberger, 2015) use deconvolution (**ConvTranspose + stride**)



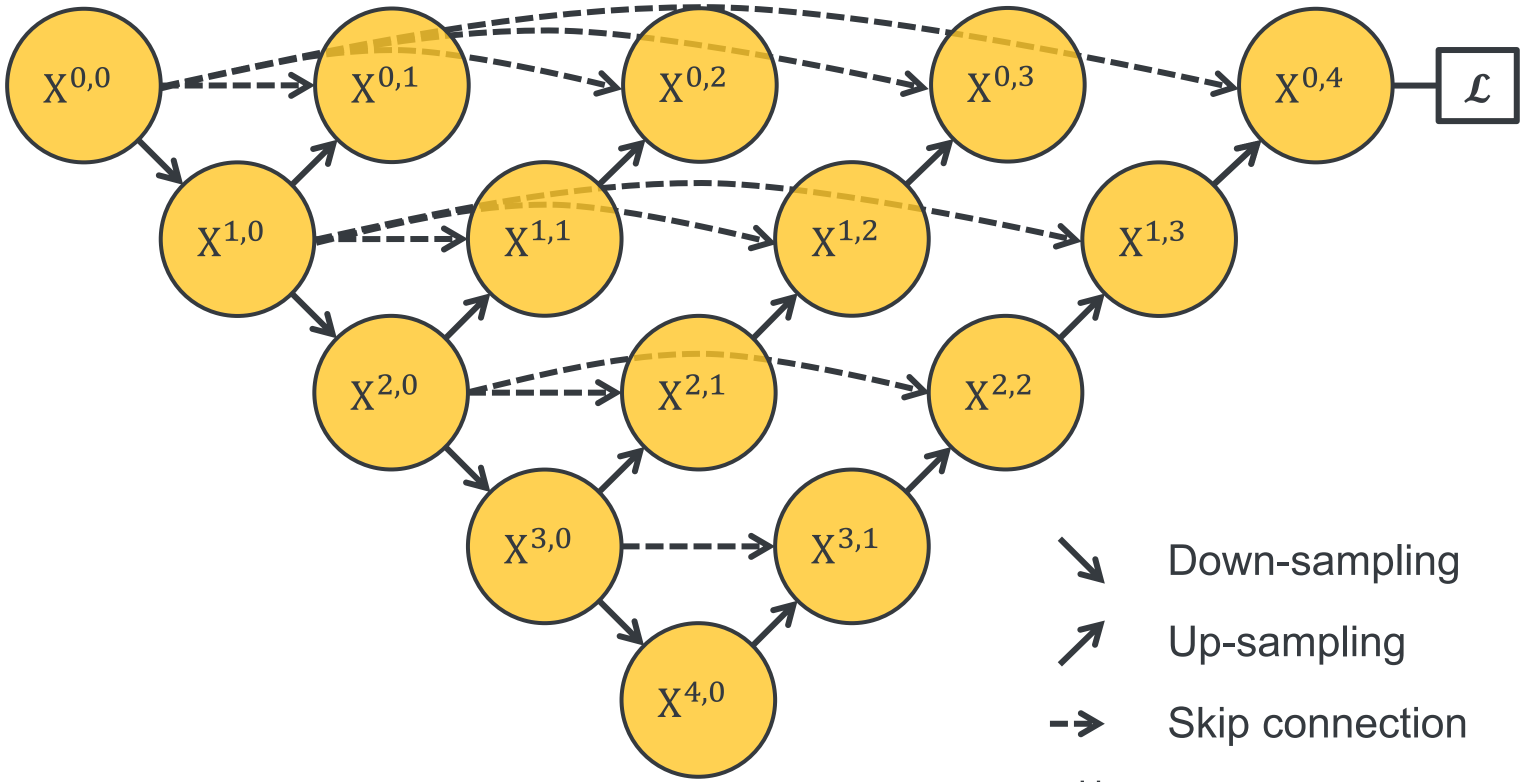
Why up-sampling until 4 level?

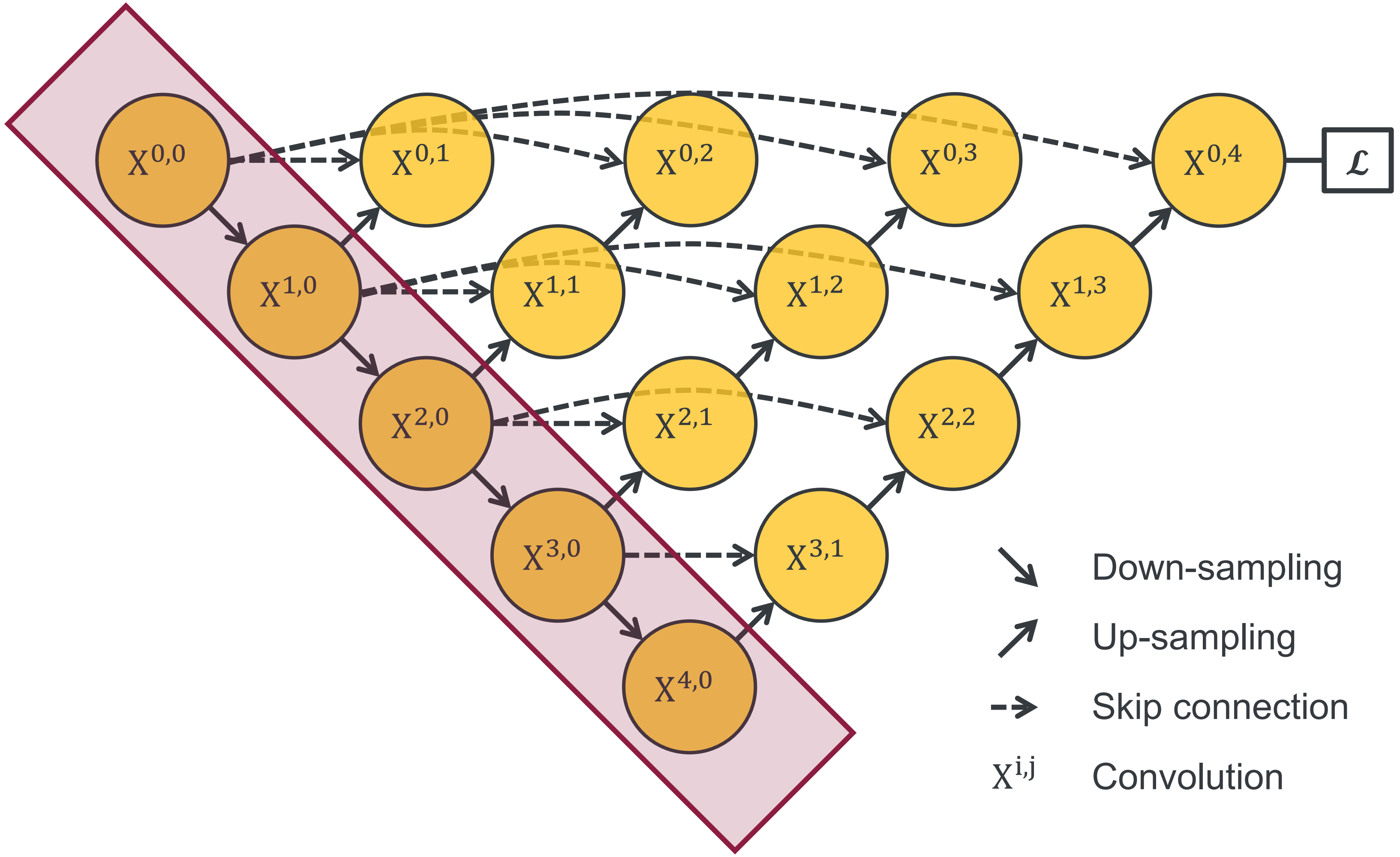

$\mathcal{L}$
$X^{0,0}$ $X^{0,1}$ $X^{1,0}$
$\mathcal{L}$
$X^{0,0}$ $X^{0,2}$ $X^{1,0}$ $X^{1,1}$ $X^{2,0}$
$\mathcal{L}$
$X^{0,0}$ $X^{0,3}$ $X^{1,0}$ $X^{1,2}$ $X^{2,0}$ $X^{2,1}$ $X^{3,0}$
$\mathcal{L}$
$X^{0,0}$ $X^{0,4}$ $X^{1,0}$ $X^{1,3}$ $X^{2,0}$ $X^{2,2}$ $X^{3,0}$ $X^{3,1}$ $X^{4,0}$
U-Net L^1
U-Net L^2
U-Net L^3
U-Net (L^4)

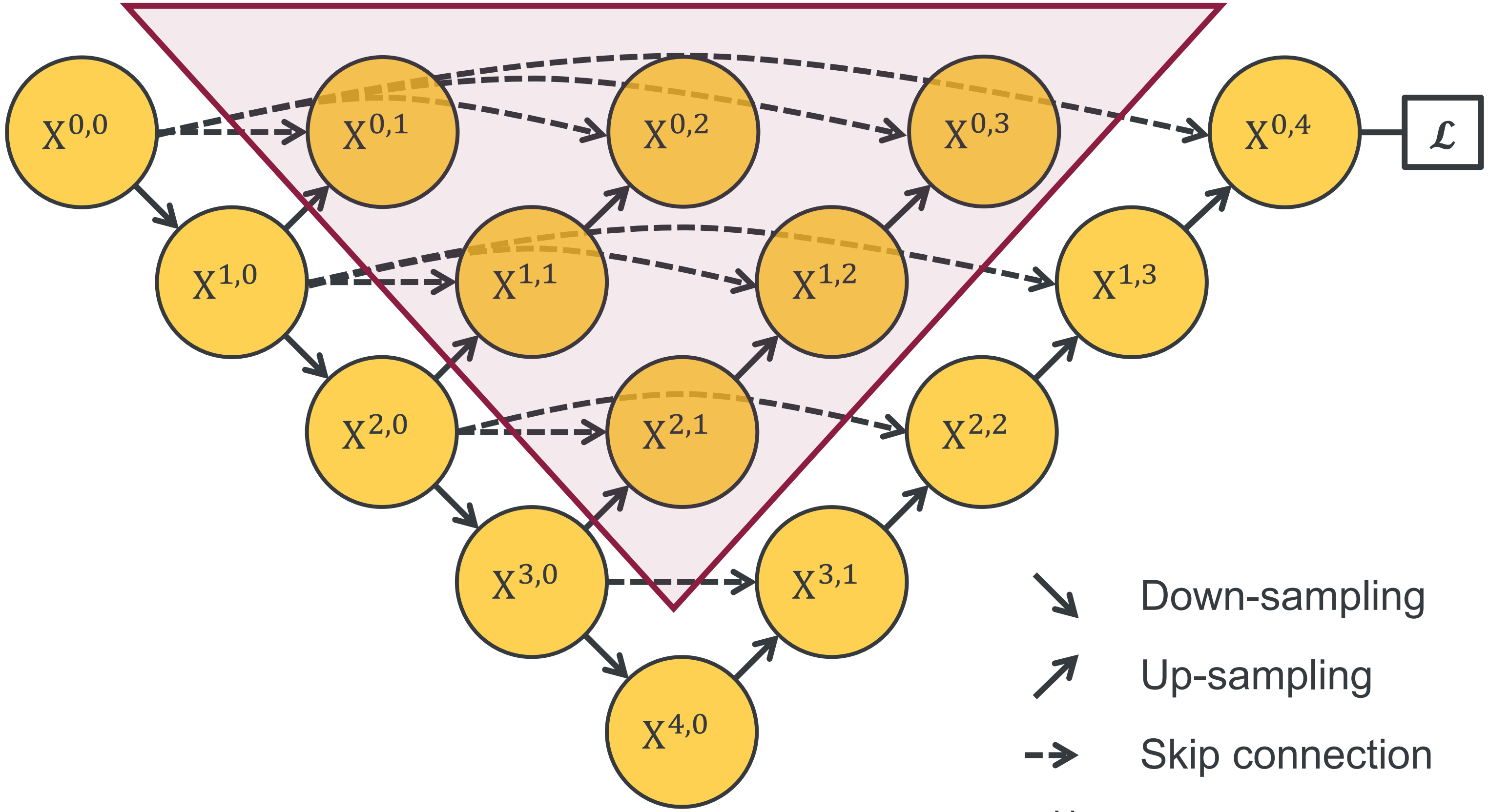

$\mathcal{L}$
$X^{0,0}$
$X^{0,1}$
$X^{1,0}$
$X^{0,2}$
$X^{1,1}$
$X^{2,0}$
$X^{0,3}$
$X^{1,2}$
$X^{2,1}$
$X^{3,0}$
92
90
88
86
84
Electron Microscopy
Cell
U-Net L^1
U-Net L^2
U-Net L^3
U-Net (L^4)
UNet++ (L^4) w/o DS
UNet++ (L^4) w/t DS

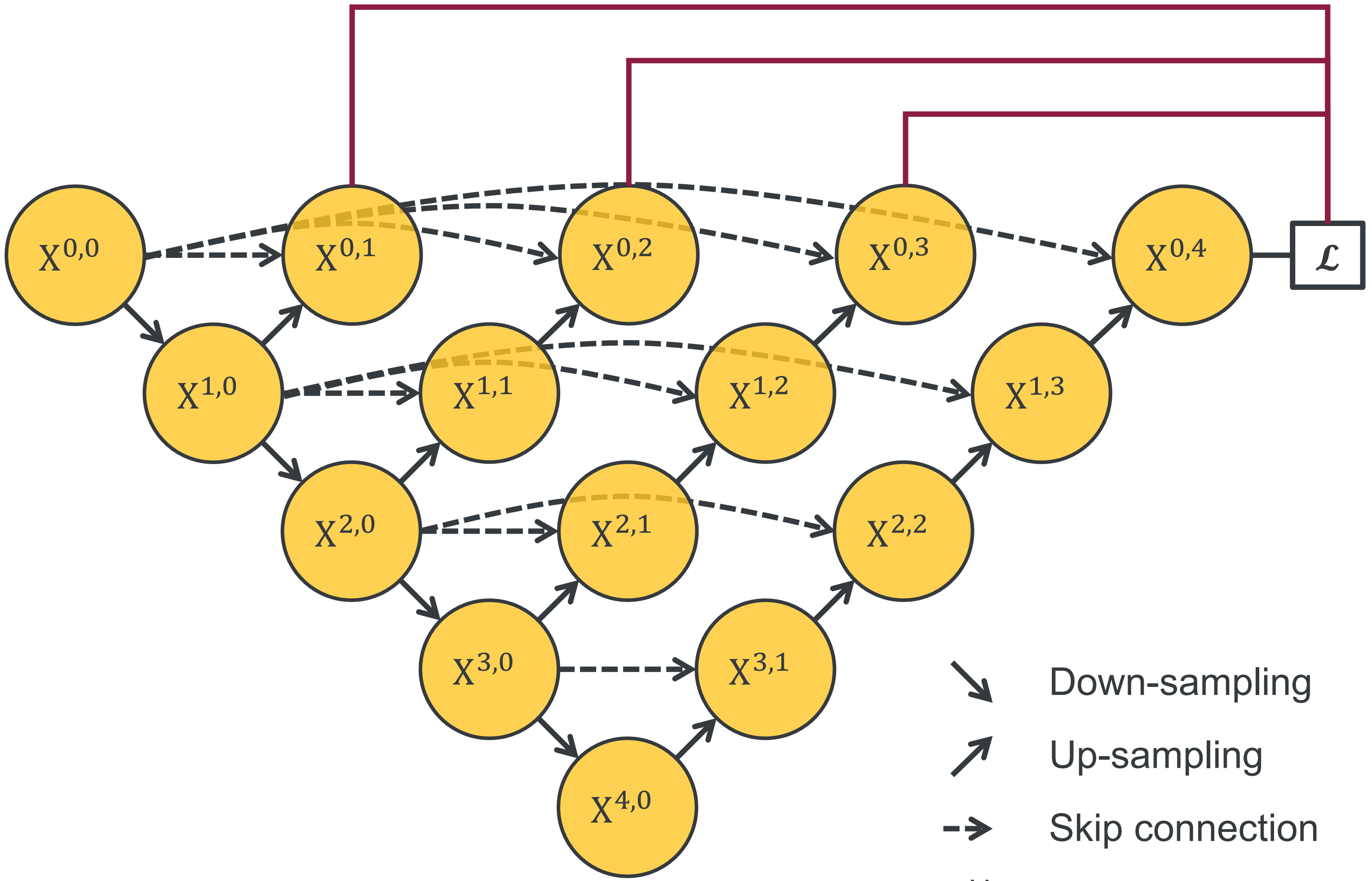


$\mathcal{L}$
$X^{0,0}$
$X^{0,1}$
$X^{1,0}$
$\mathcal{L}$
$X^{0,0}$
$X^{0,2}$
$X^{1,0}$
$X^{1,1}$
$X^{2,0}$
$\mathcal{L}$
$X^{0,0}$
$X^{0,3}$
$X^{1,0}$
$X^{1,2}$
$X^{2,0}$
$X^{2,1}$
$X^{3,0}$
$\mathcal{L}$
$X^{0,0}$
$X^{0,4}$
$X^{1,0}$
$X^{1,3}$
$X^{2,0}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
U-Net L^1
U-Net L^2
U-Net L^3
U-Net (L^4)

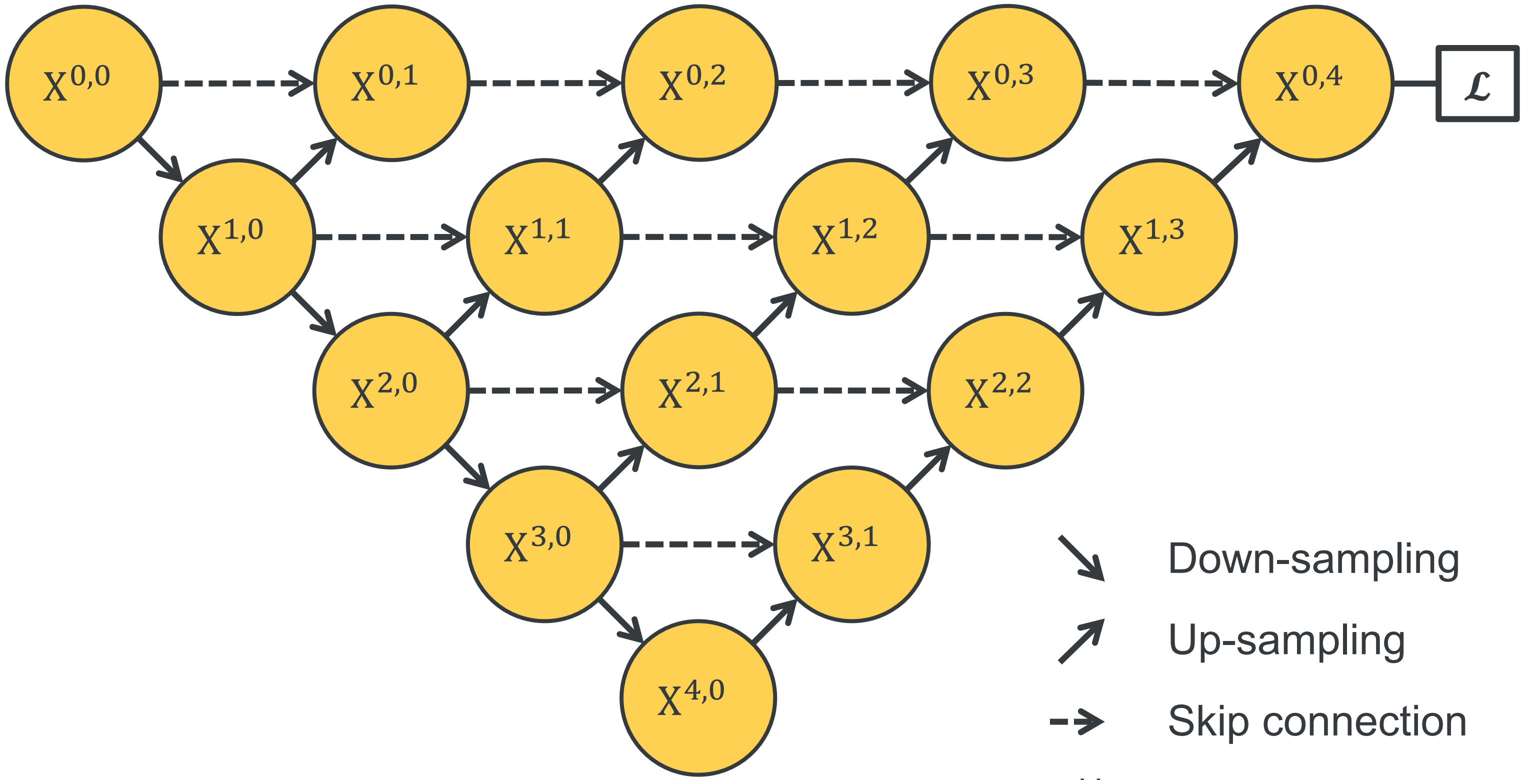

X0,0
X0,1
X0,2
X0,3
X0,4
ℒ
X1,0
X1,1
X1,2
X1,3
X2,0
X2,1
X2,2
X3,0
X3,1
X4,0
Down-sampling
Up-sampling
Skip connection
Xi,j
Convolution


$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

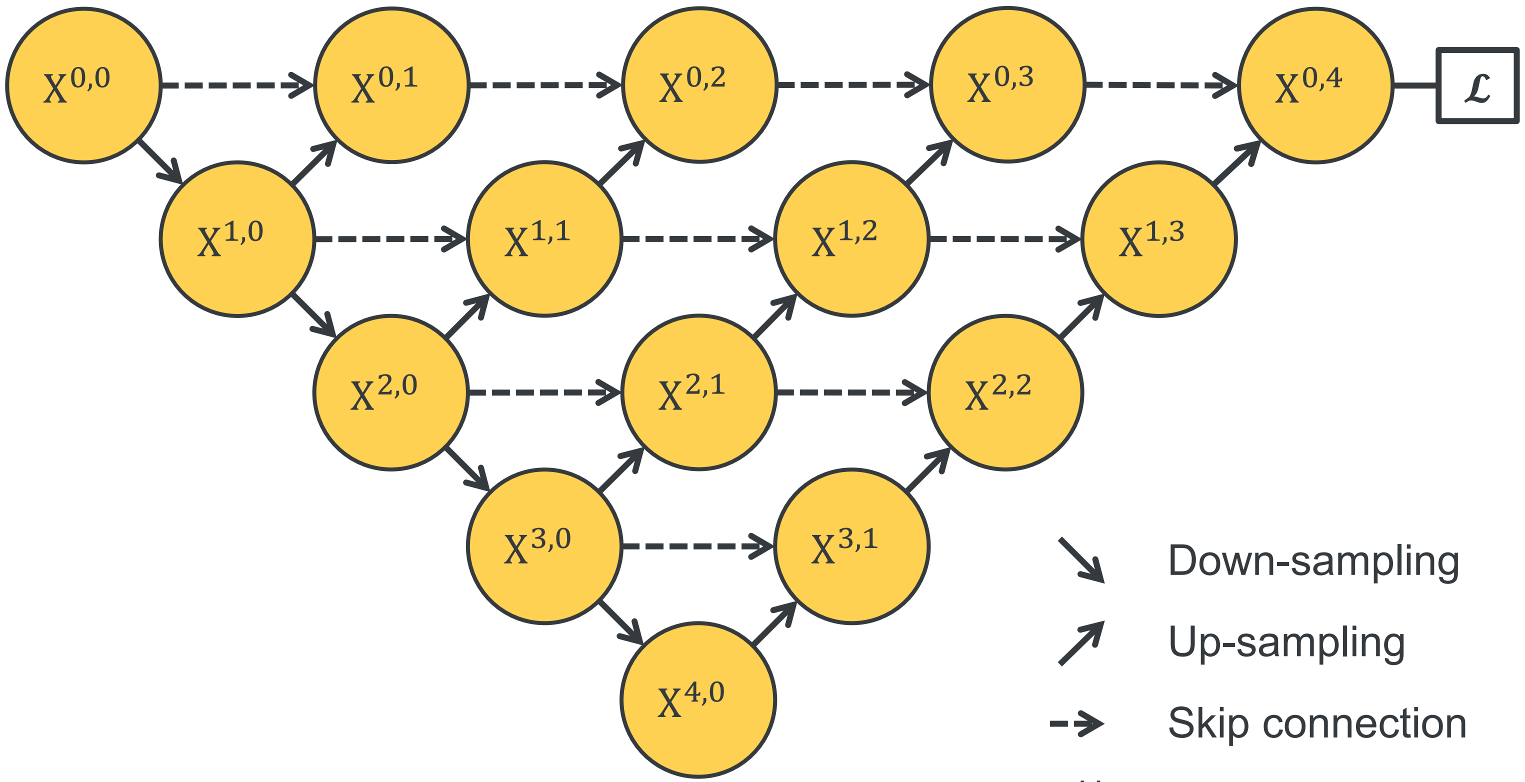

X0,0
X0,1
X0,2
X0,3
X0,4
X1,0
X1,1
X1,2
X1,3
X2,0
X2,1
X2,2
X3,0
X3,1
X4,0
ℒ
Down-sampling
Up-sampling
Skip connection
Xi,j
Convolution



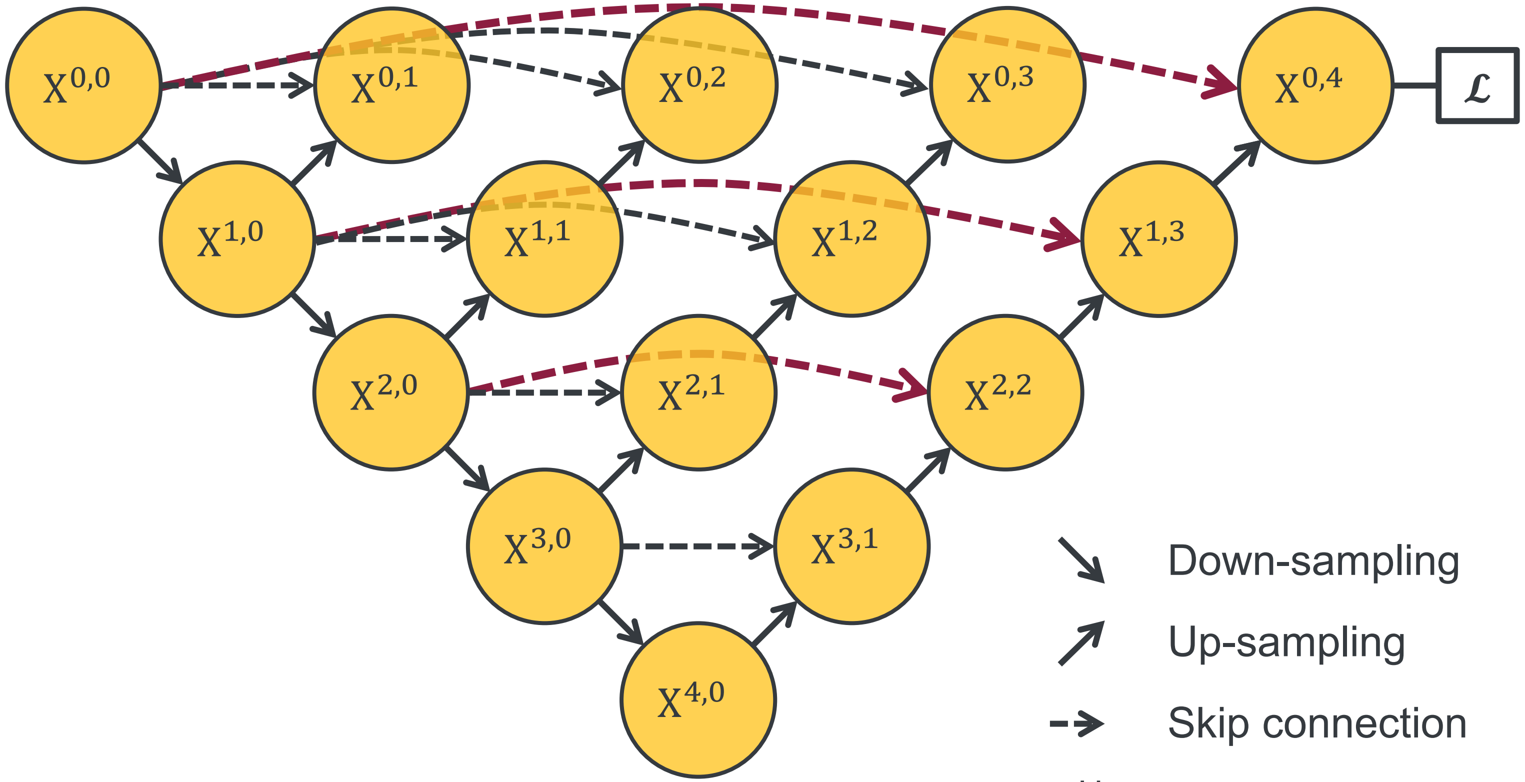
$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



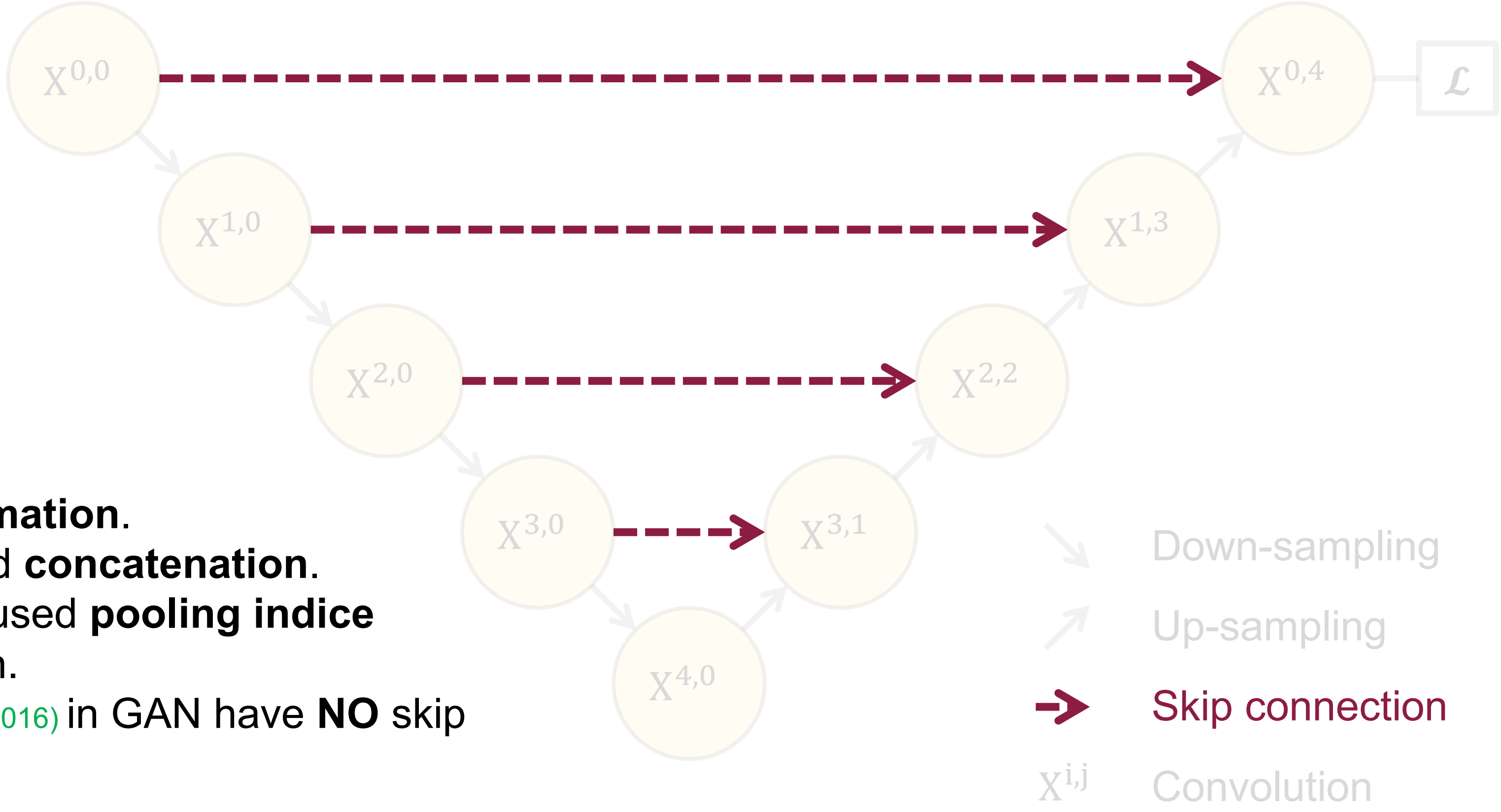
(Yu, CVPR 2018)
Deep Layer Aggregation



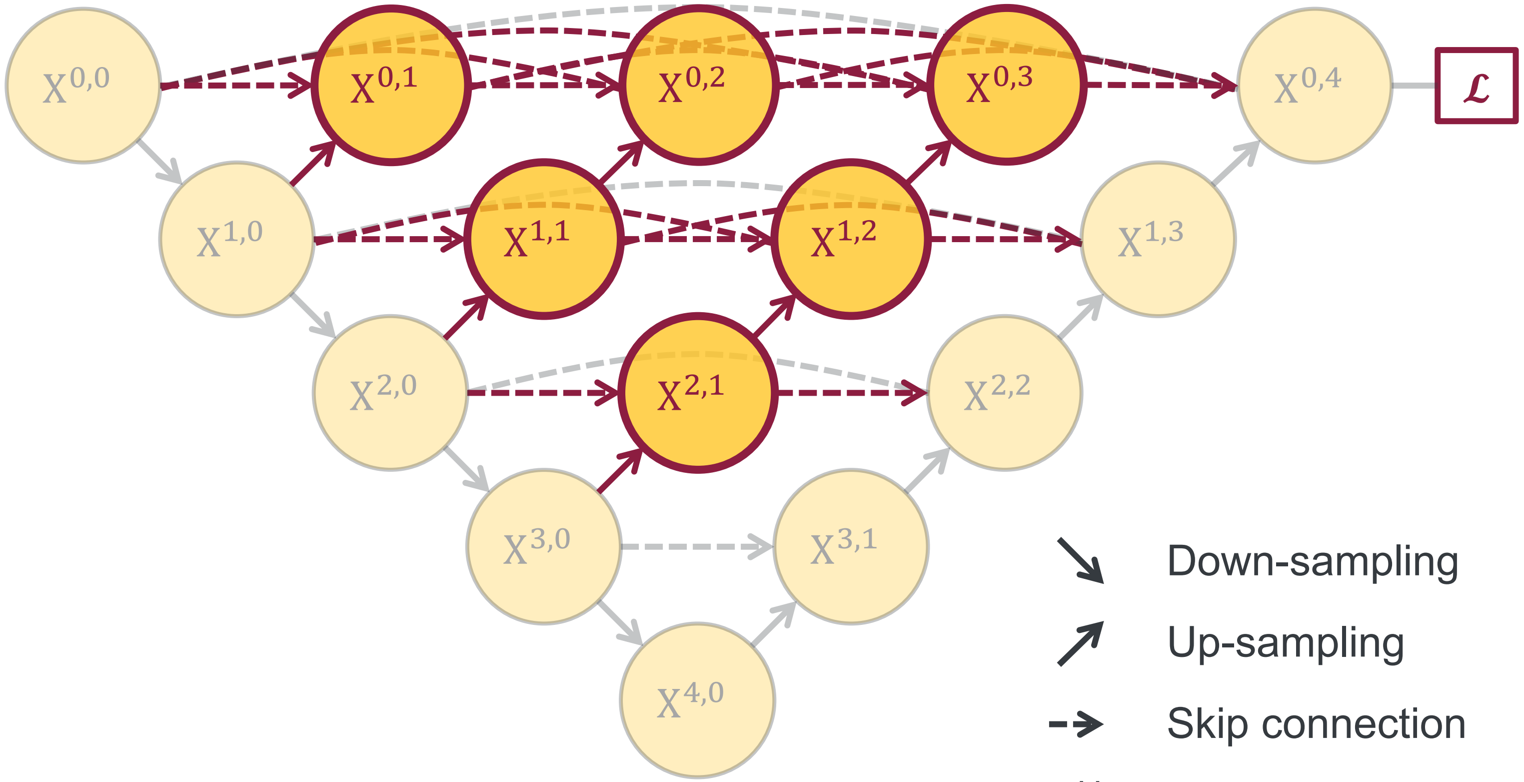
$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

The skip connection is important:

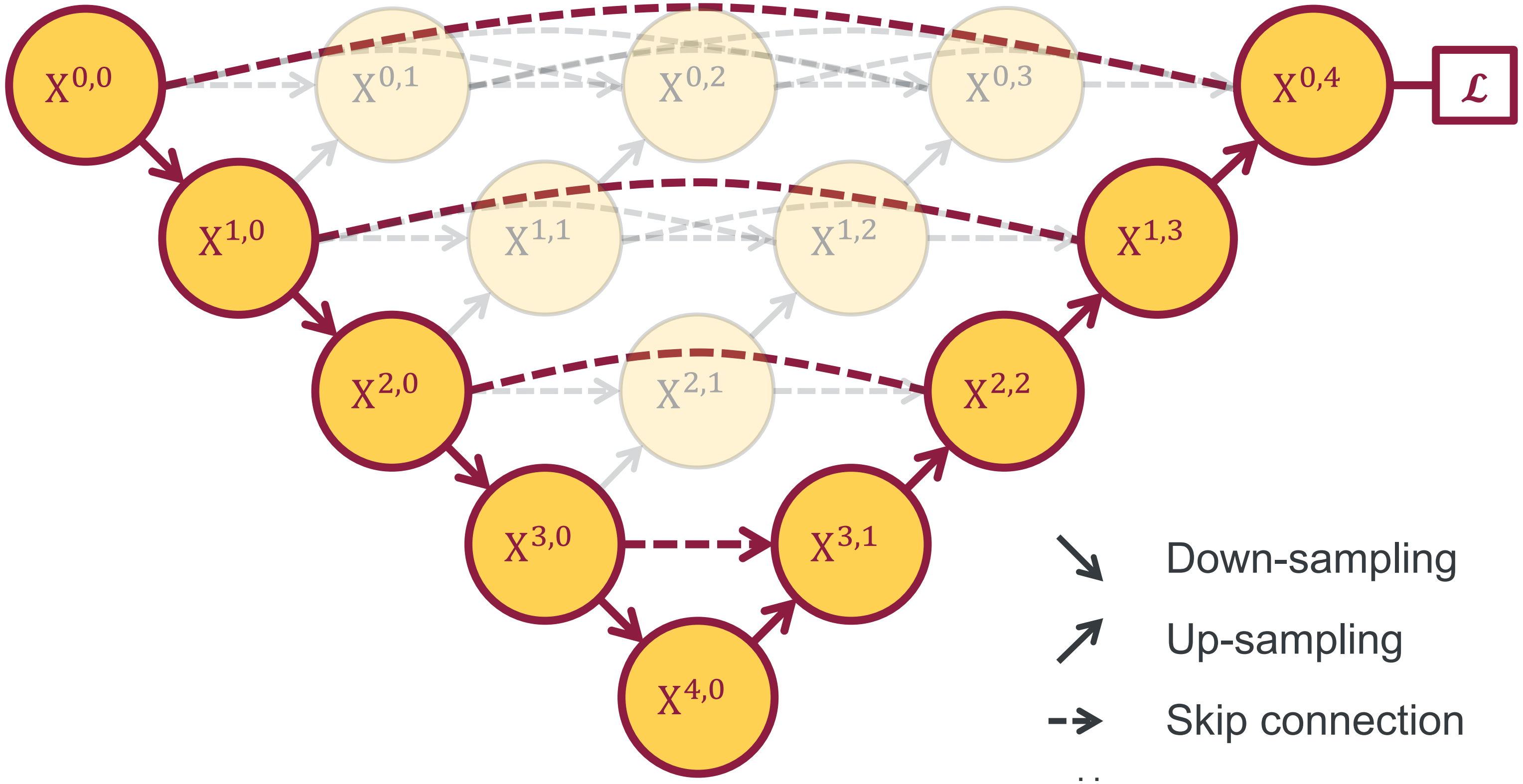
1. Fights the vanishing gradient problem.
2. Learns pyramid level features.
3. Recover info loss in down-sampling.

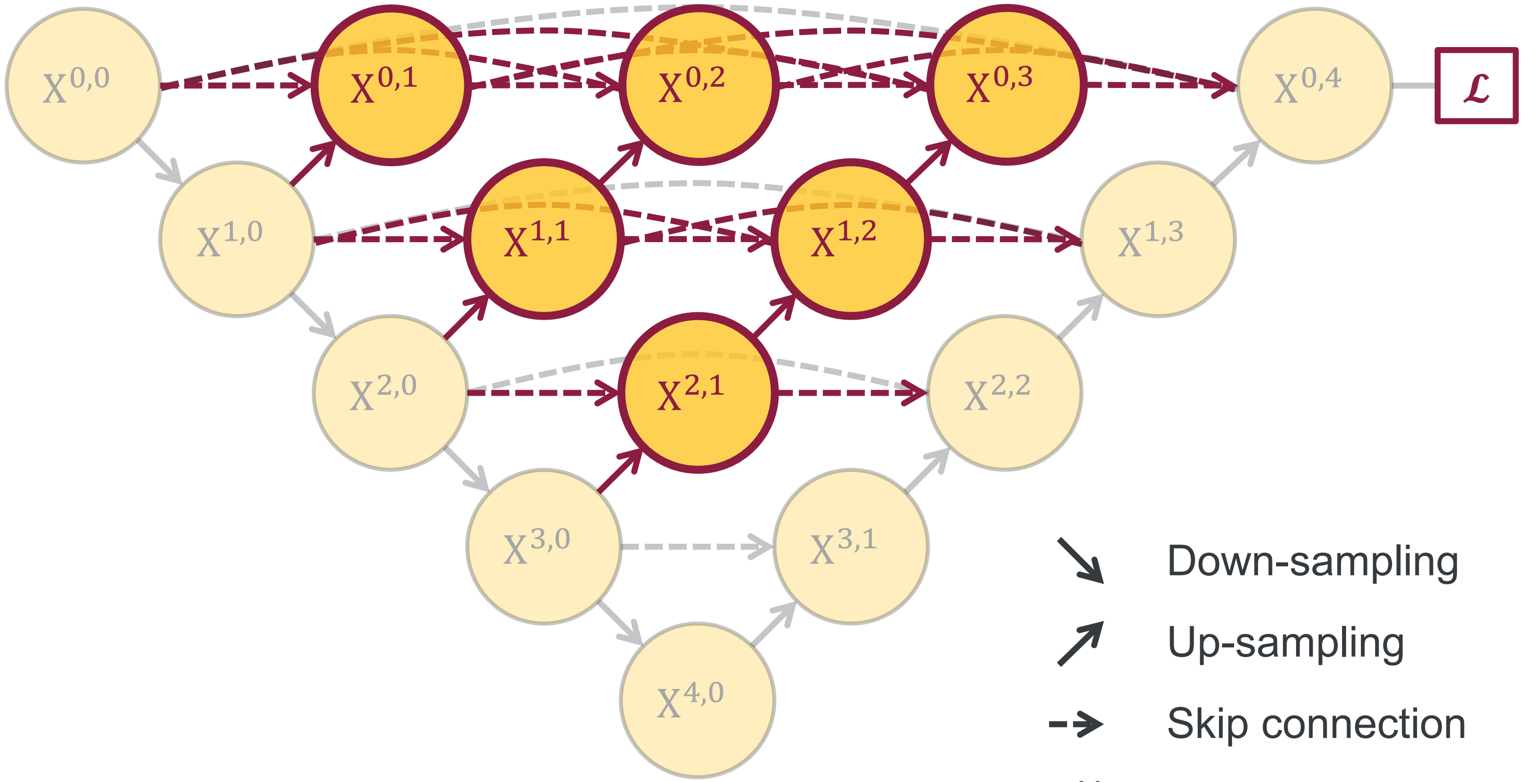


- FCN (Long, 2015) used **summation**.
- U-Net (Ronneberger, 2015) used **concatenation**.
- SegNet (Badrinarayanan, 2016) used **pooling indice** instead of skip connection.
- Some generators (Johnson 2016) in GAN have **NO** skip connection.

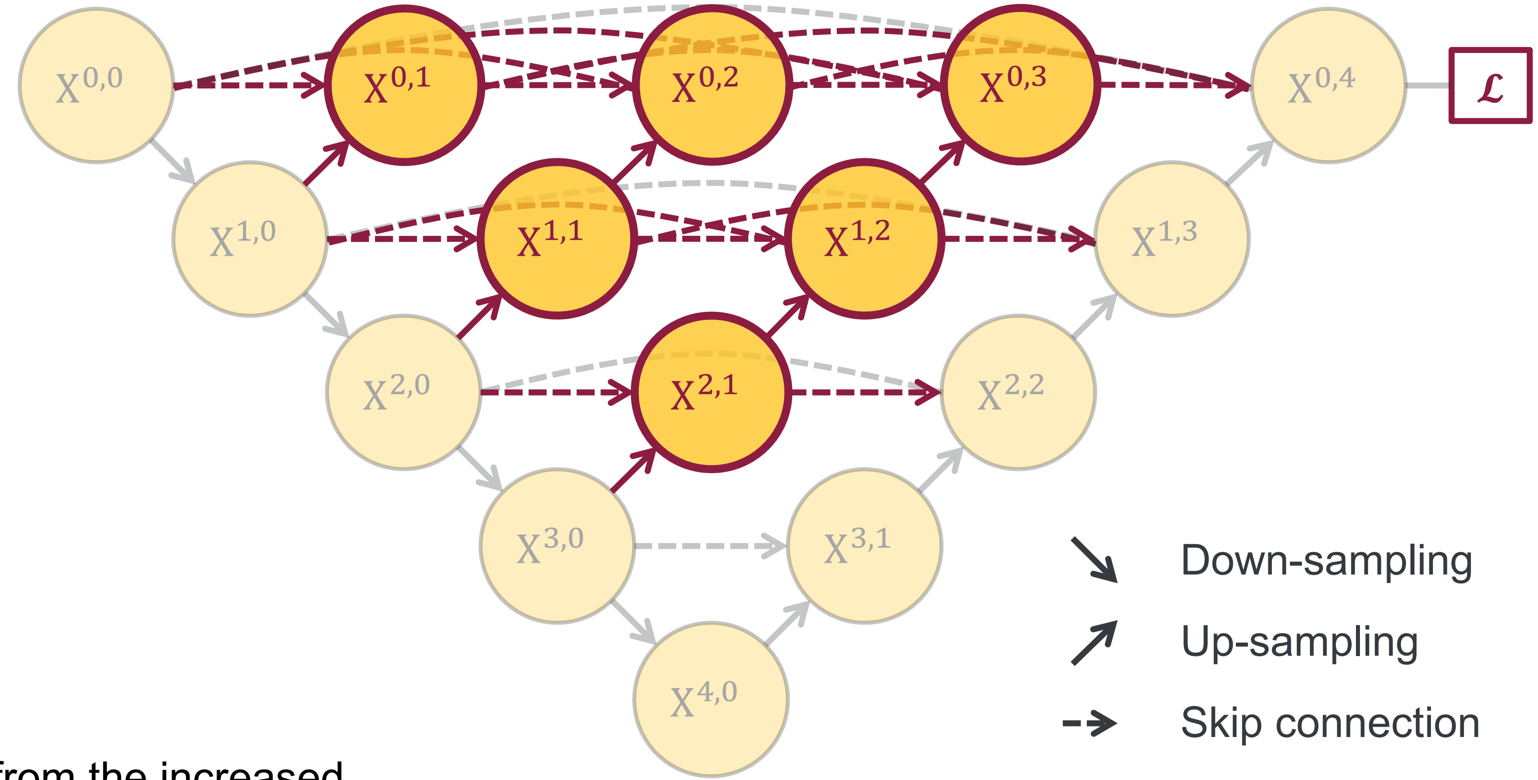


X0,0
X0,1
X0,2
X0,3
X0,4
ℒ
X1,0
X1,1
X1,2
X1,3
X2,0
X2,1
X2,2
X3,0
X3,1
X4,0
Down-sampling
Up-sampling
Skip connection
Xi,j
Convolution

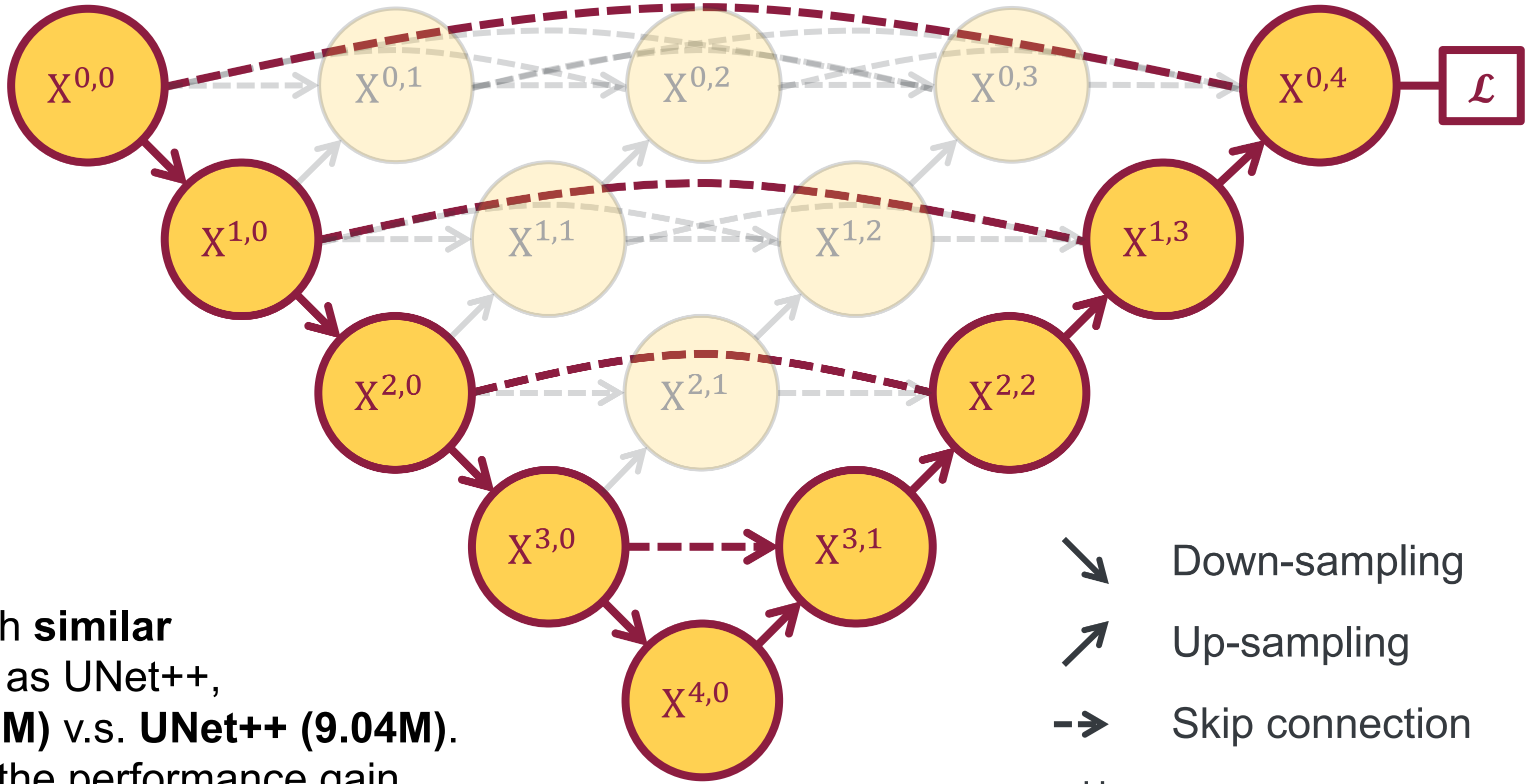

$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



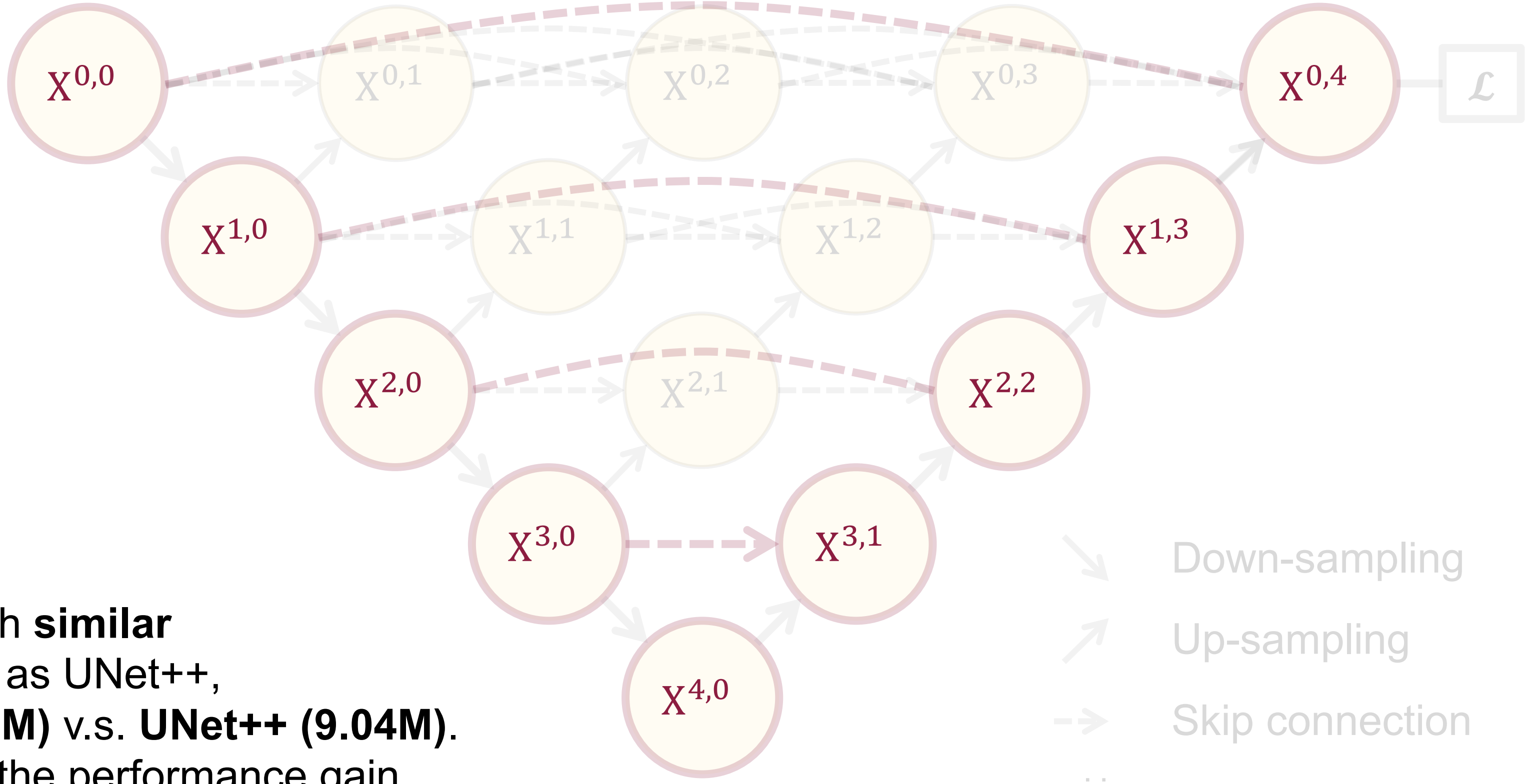
Q: Does the benefit come from the increased number of parameters? That is, U-Net (7.76M) vs. UNet++ (9.04M)?



"Wide" U-Net

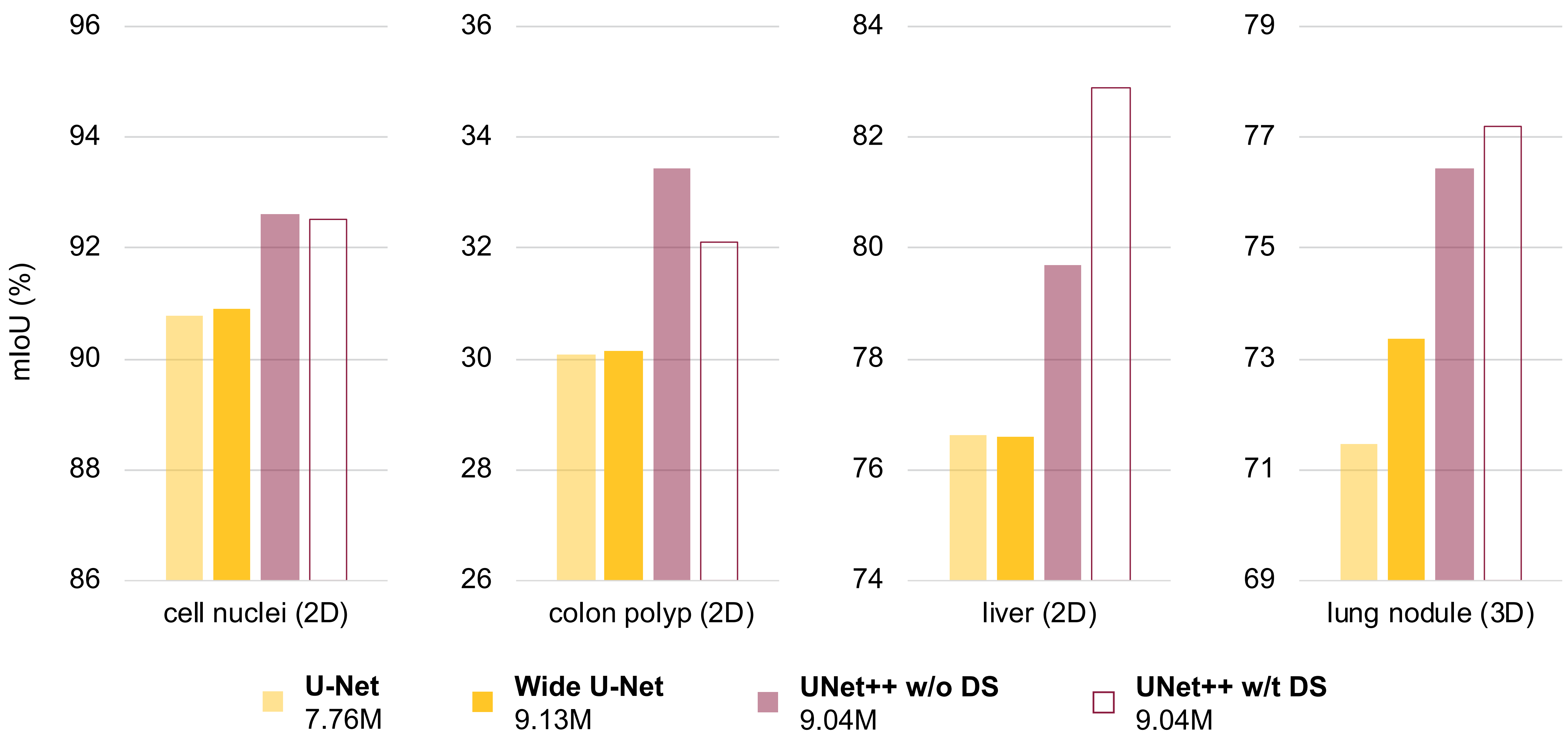
- Design a wide U-Net with **similar number of parameters** as UNet++, where **wide U-Net (9.13M)** v.s. **UNet++ (9.04M)**.
- This was to ensure that the performance gain yielded by UNet++ is **not** simply due to increased number of parameters.

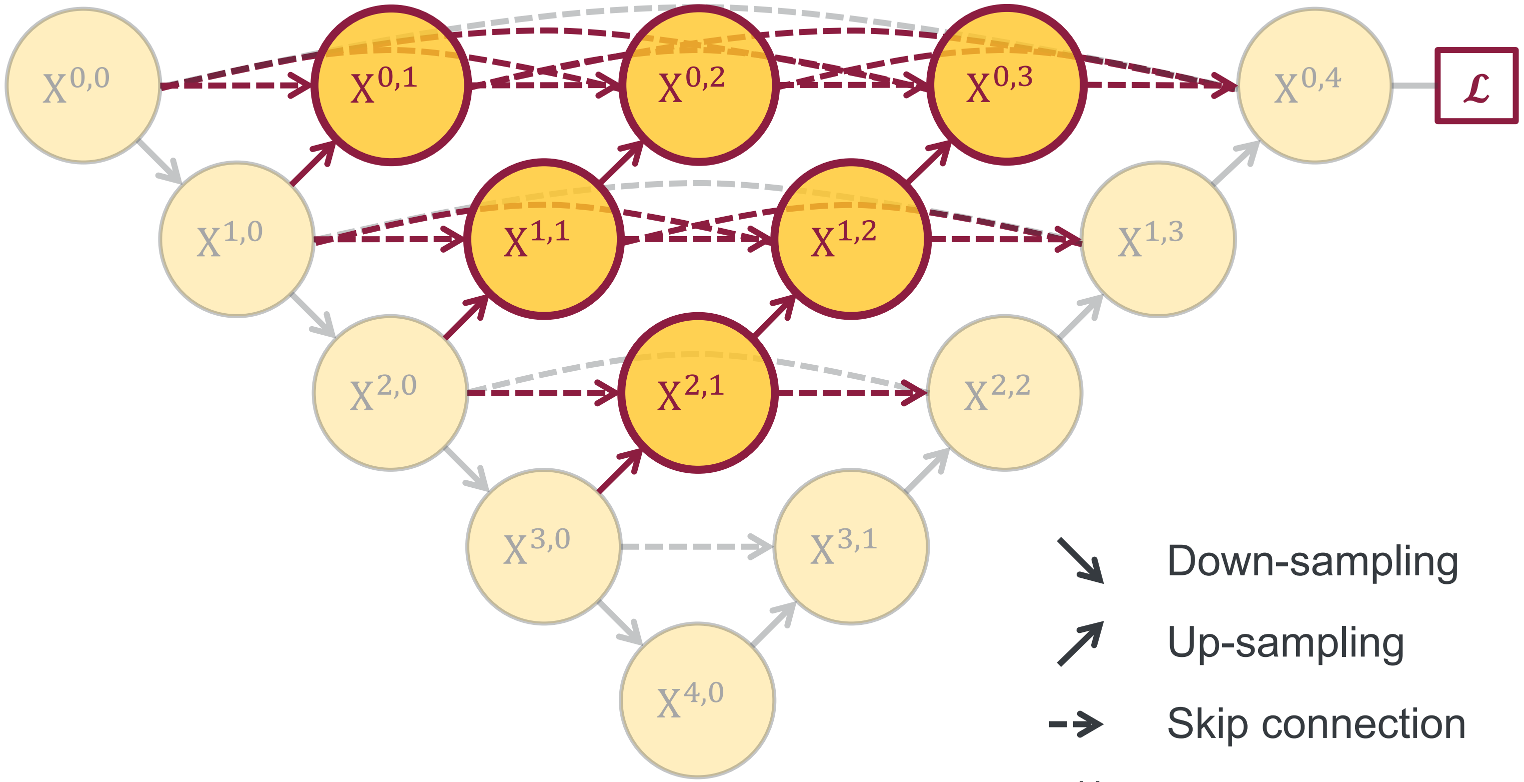
encoder / decoder	$X^{0,0}$ / $X^{0,4}$	$X^{1,0}$ / $X^{1,3}$	$X^{2,0}$ / $X^{2,2}$	$X^{3,0}$ / $X^{3,1}$	$X^{4,0}$ / $X^{4,0}$
U-Net **[7.76M]**	32	64	128	256	512
"Wide" U-Net **[9.13M]**	35	70	140	280	560
UNet++ **[9.04M]**	32	64	128	256	512

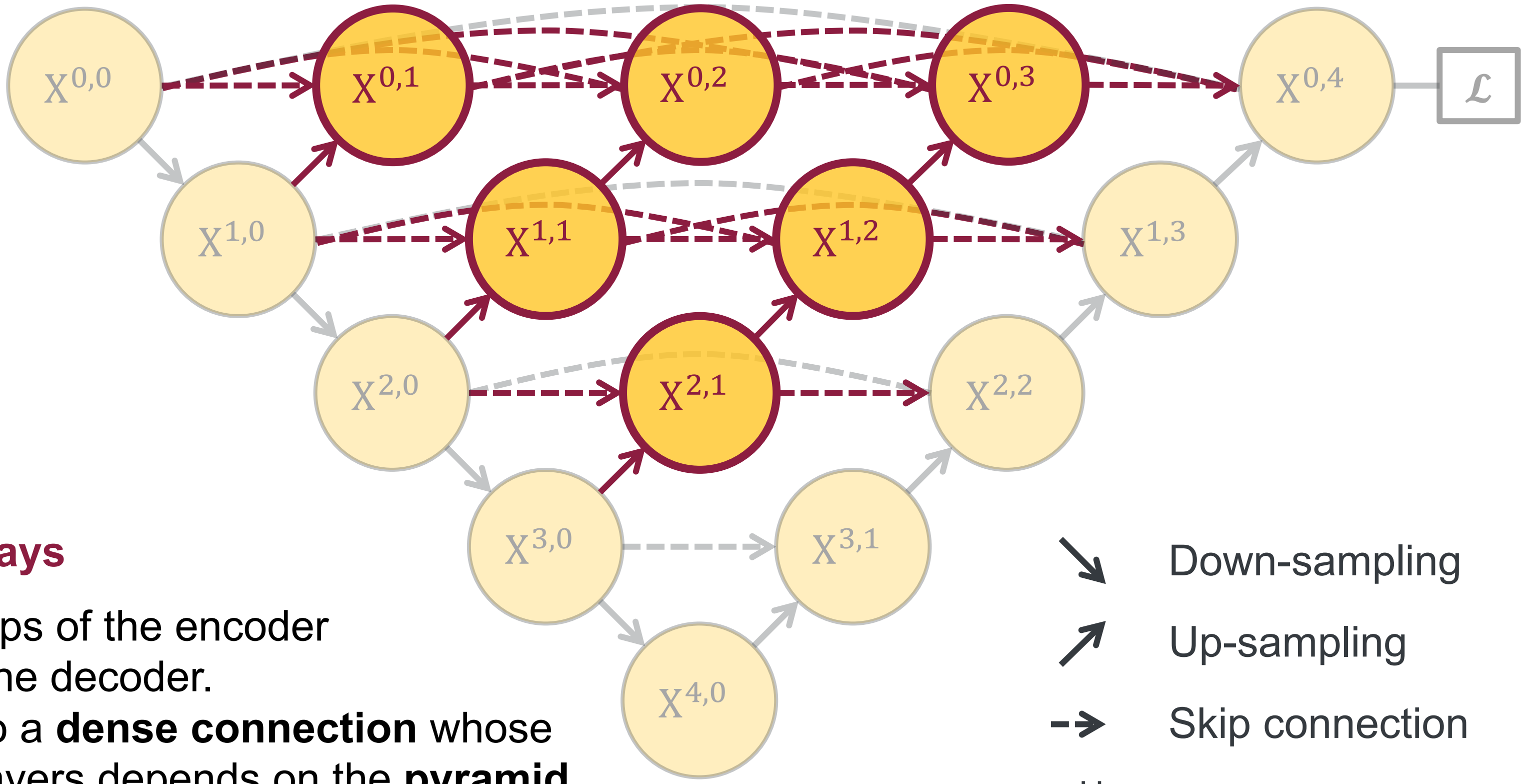


"Wide" U-Net

- Design a wide U-Net with **similar number of parameters** as UNet++, where **wide U-Net (9.13M)** v.s. **UNet++ (9.04M)**.
- This was to ensure that the performance gain yielded by UNet++ is **not** simply due to increased number of parameters.

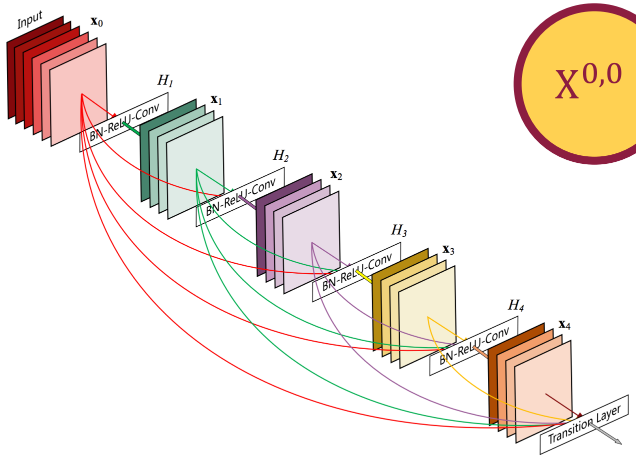

mIoU (%)
96
94
92
90
88
86
cell nuclei (2D)
36
34
32
30
28
26
colon polyp (2D)
84
82
80
78
76
74
liver (2D)
79
77
75
73
71
69
lung nodule (3D)
U-Net
7.76M
Wide U-Net
9.13M
UNet++ w/o DS
9.04M
UNet++ w/t DS
9.04M

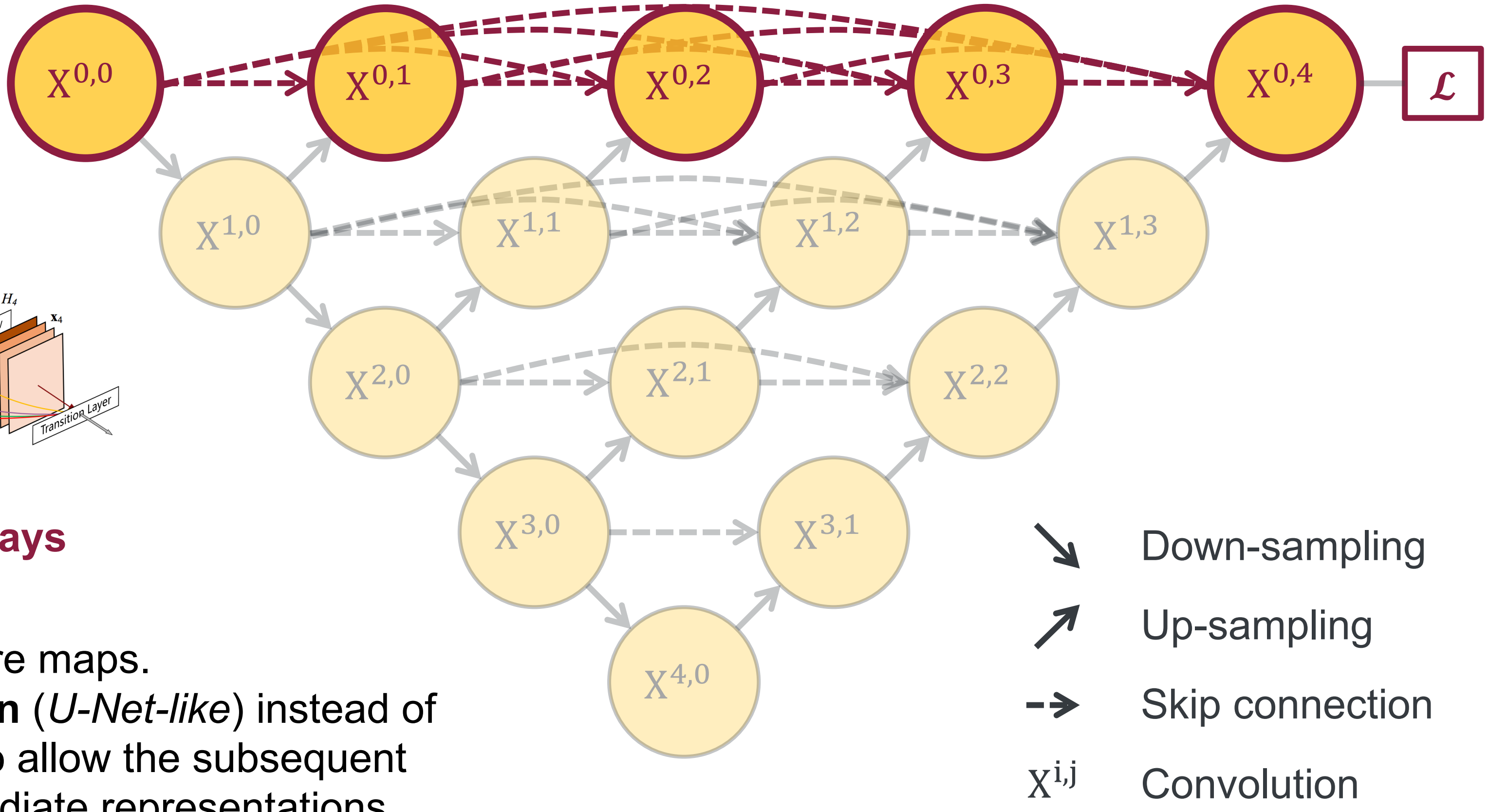

$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



Re-design the skip pathways

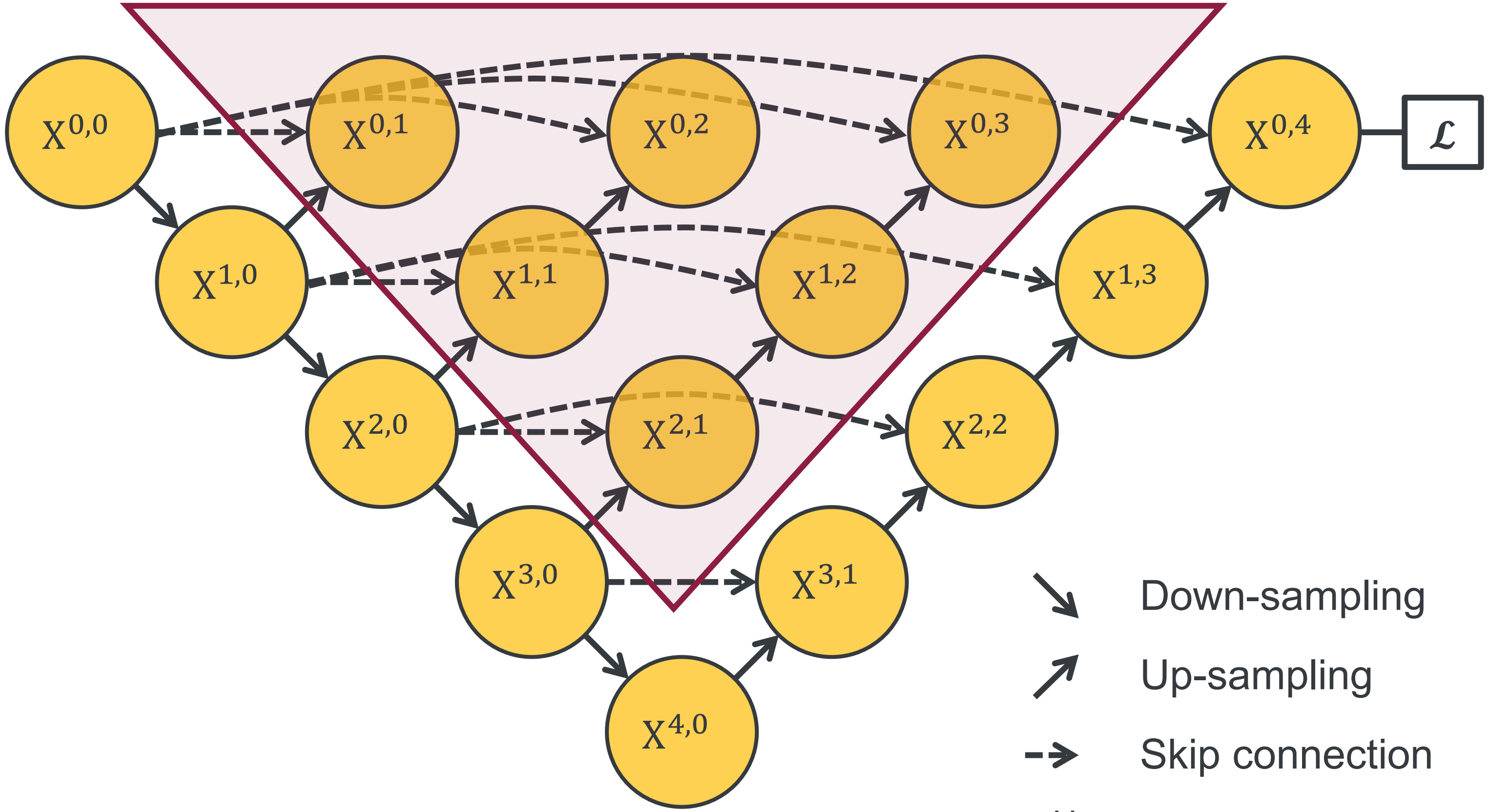
- In U-Net, the feature maps of the encoder are directly received in the decoder.
- In UNet++, they undergo a **dense connection** whose number of convolution layers depends on the **pyramid level**.

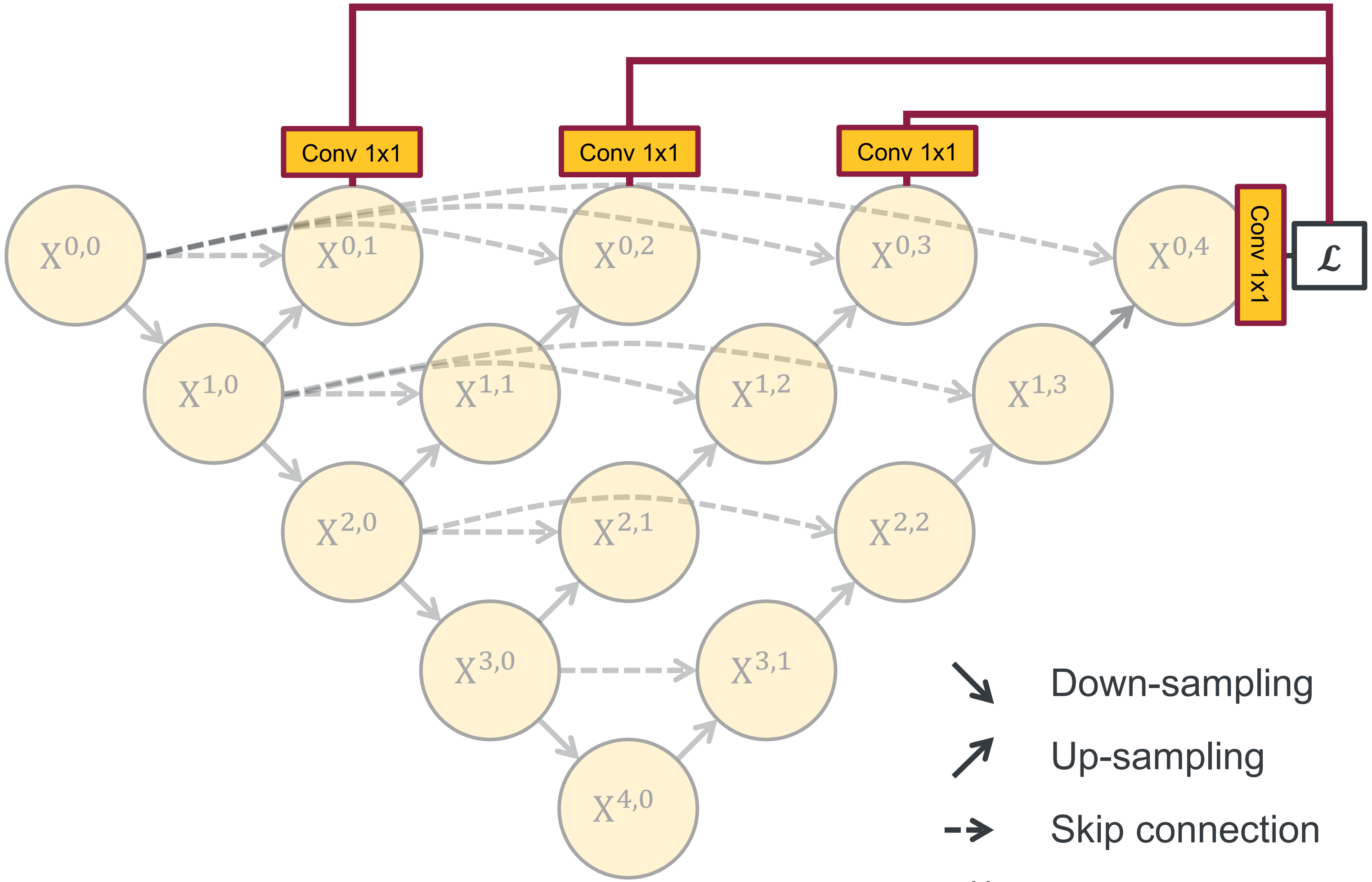


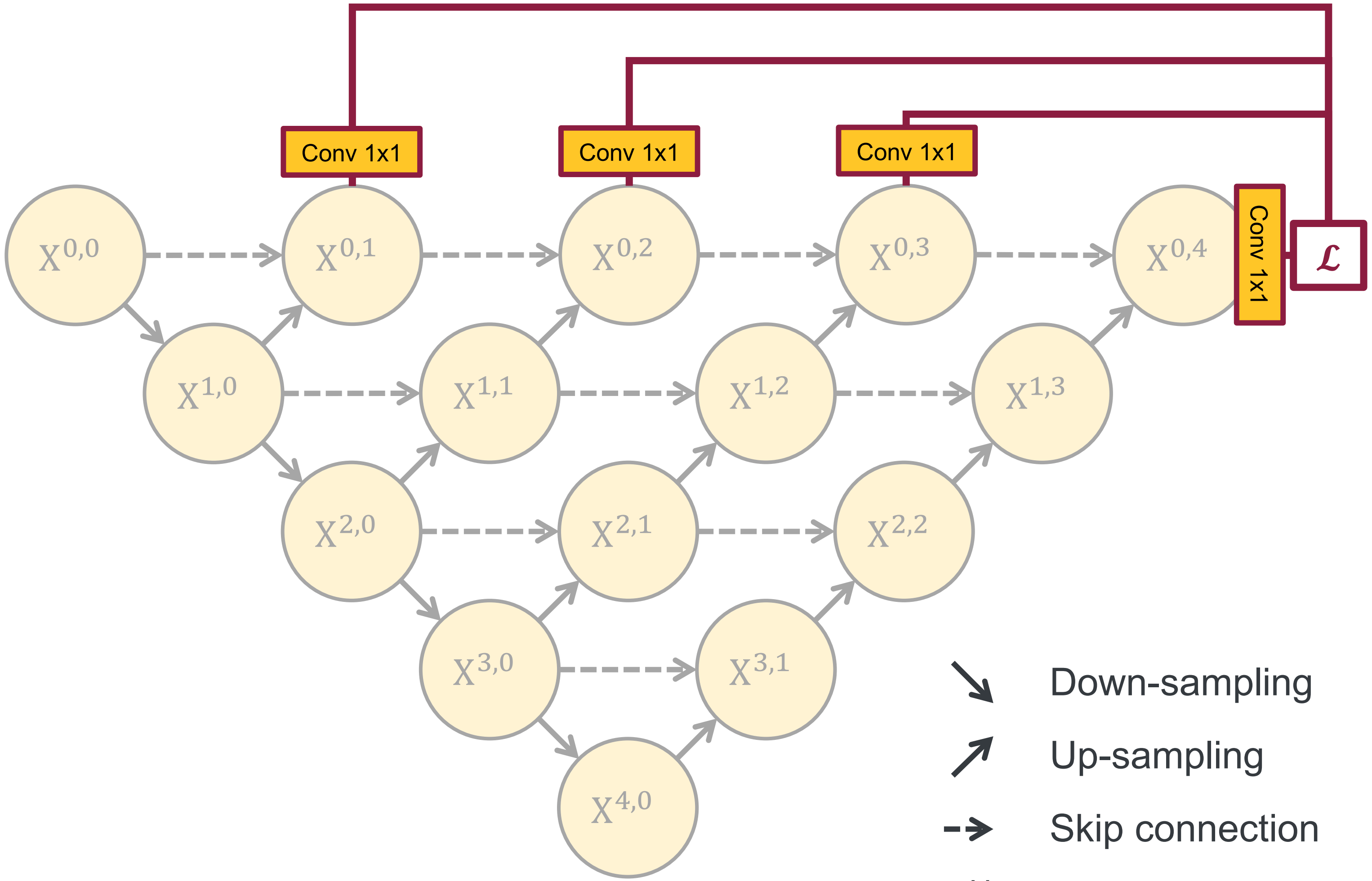


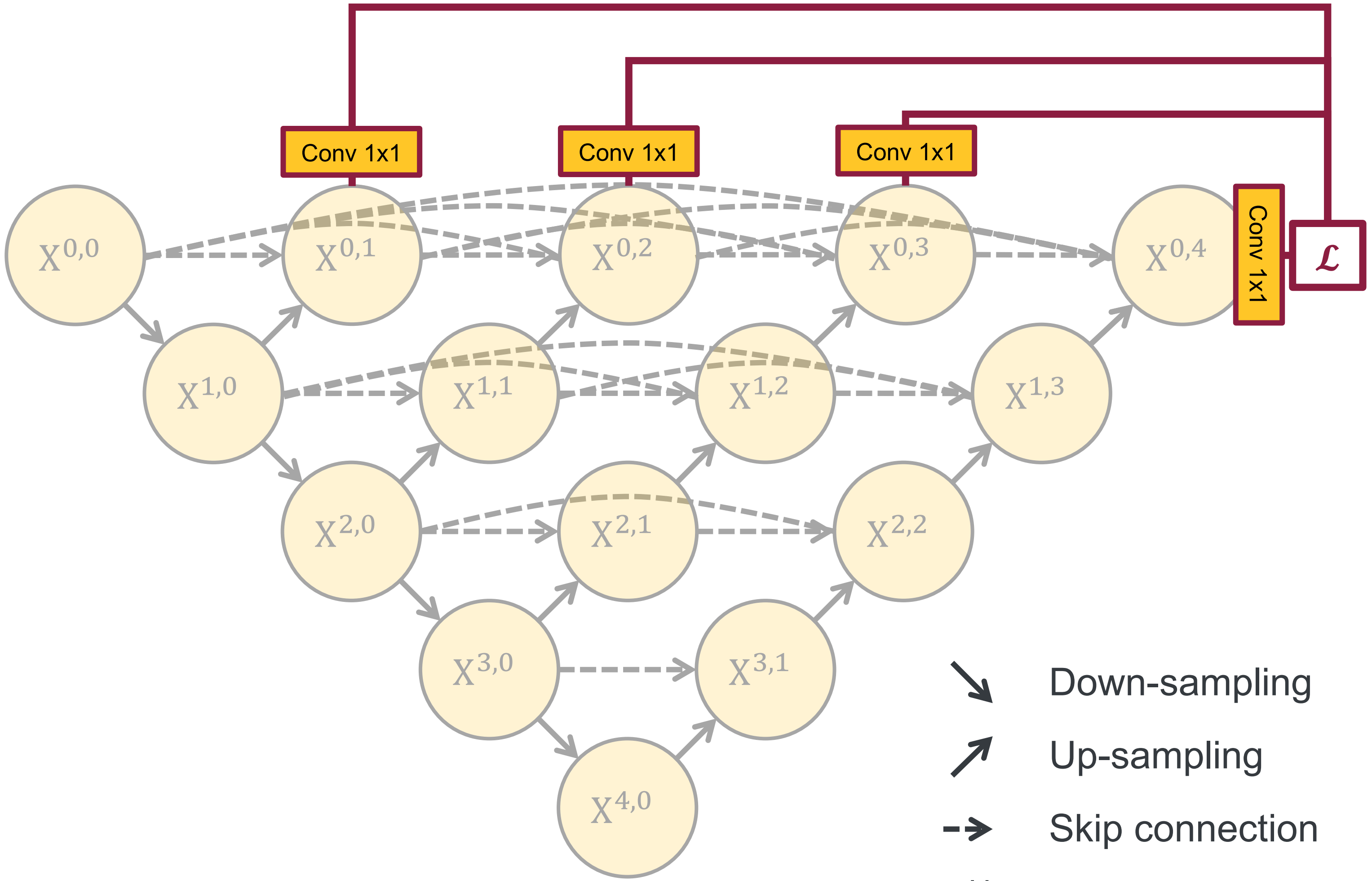
Re-design the skip pathways

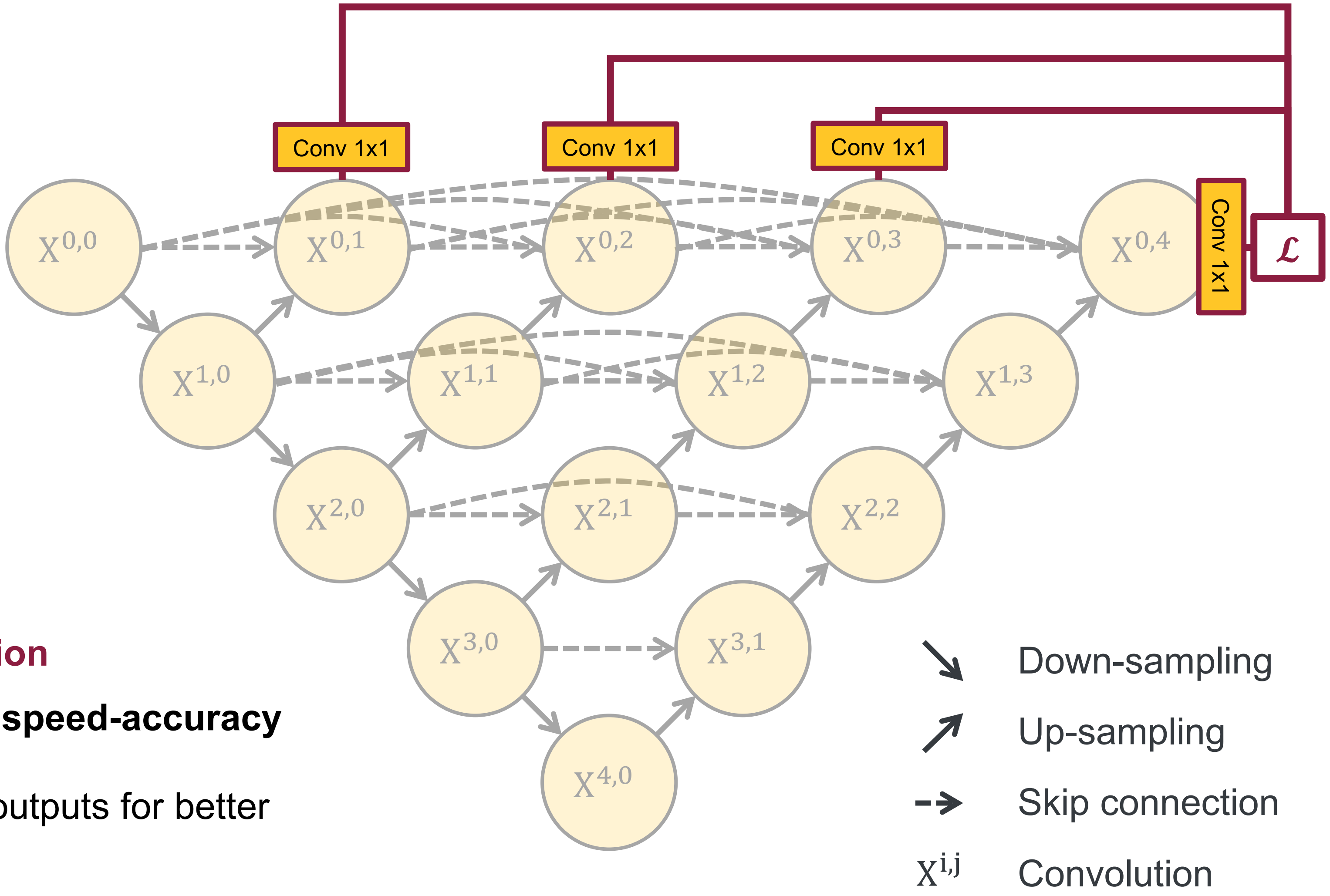
- Increase feature **fusion**.
- **Densely connect** feature maps.
- We apply **concatenation** (*U-Net-like*) instead of summation (*FCN-like*) to allow the subsequent layers to **re-use** intermediate representations.


$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$\mathcal{L}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution


Conv 1x1
Conv 1x1
Conv 1x1
Conv 1x1
$\mathcal{L}$
$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution


Conv 1x1
Conv 1x1
Conv 1x1
Conv 1x1
$\mathcal{L}$
$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution

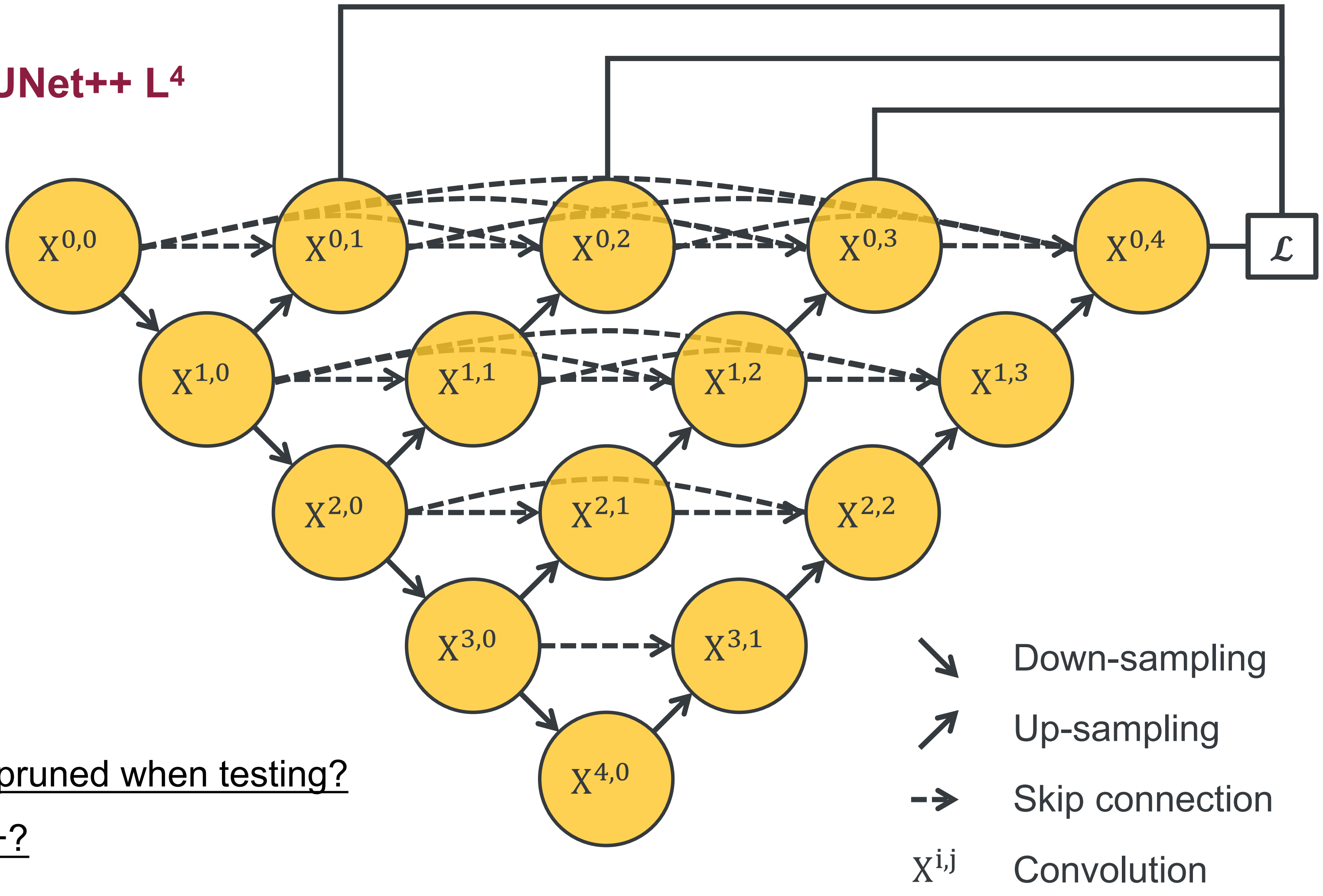

Conv 1x1
Conv 1x1
Conv 1x1
Conv 1x1
$\mathcal{L}$
$X^{0,0}$
$X^{0,1}$
$X^{0,2}$
$X^{0,3}$
$X^{0,4}$
$X^{1,0}$
$X^{1,1}$
$X^{1,2}$
$X^{1,3}$
$X^{2,0}$
$X^{2,1}$
$X^{2,2}$
$X^{3,0}$
$X^{3,1}$
$X^{4,0}$
Down-sampling
Up-sampling
Skip connection
$X^{i,j}$ Convolution



The use of deep supervision

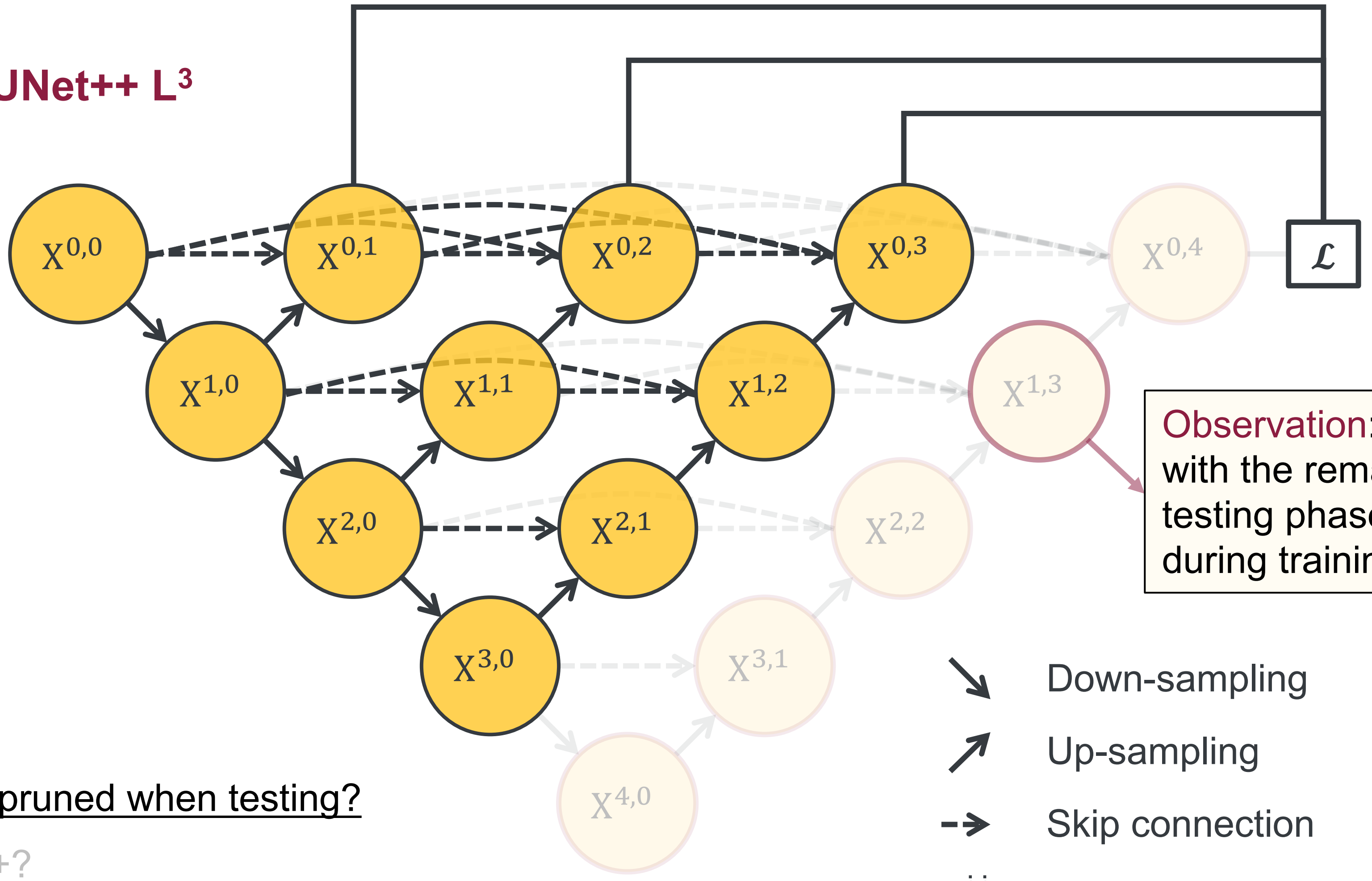
- Allow model pruning via **speed-accuracy** trade-off.
- Ensemble **multi-depth** outputs for better accuracy.

UNet++ L^4



- Q: Why UNet++ can be pruned when testing?
- Q: How to prune UNet++?
- Q: What's the benefit?

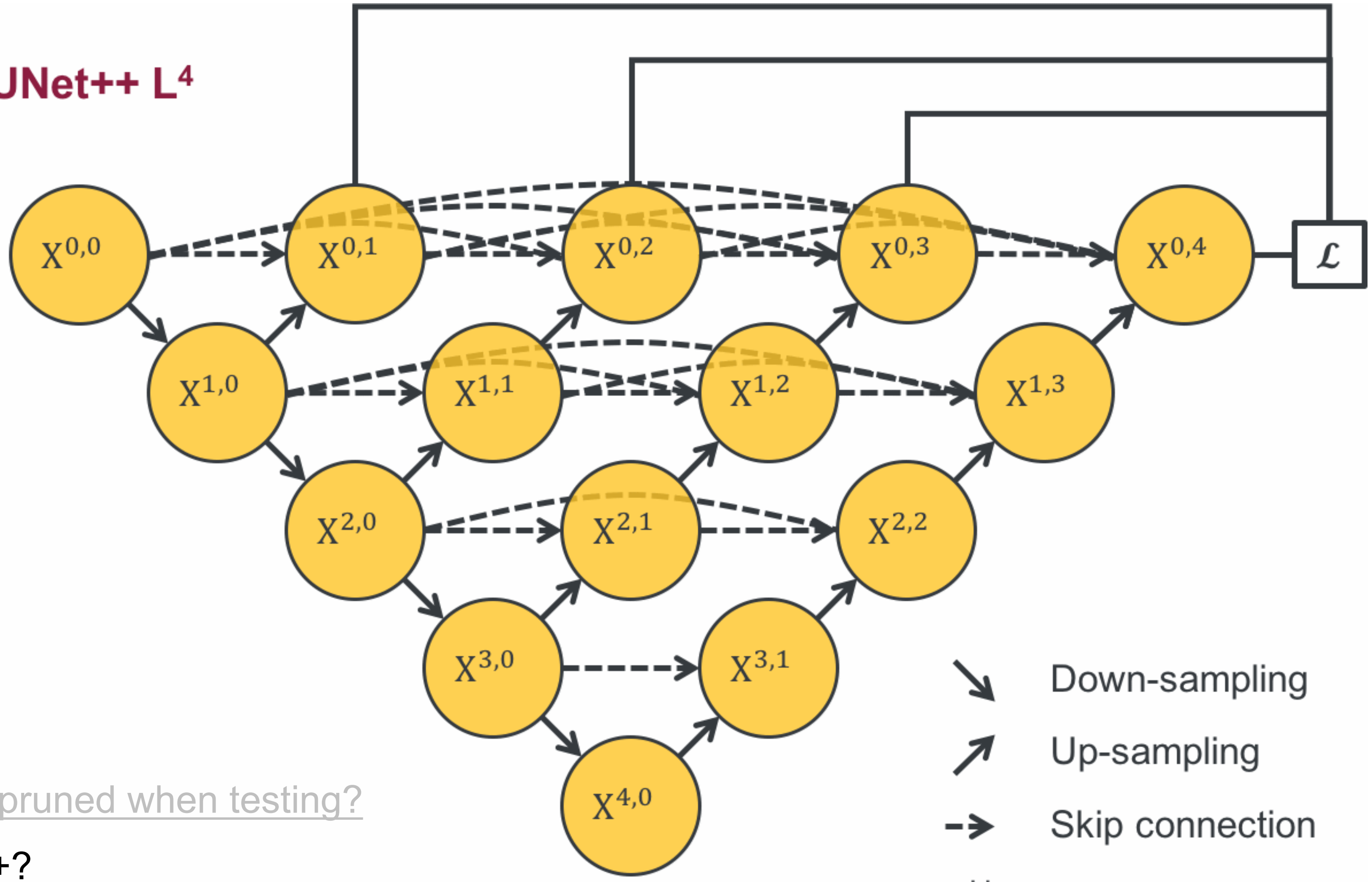
UNet++ L3



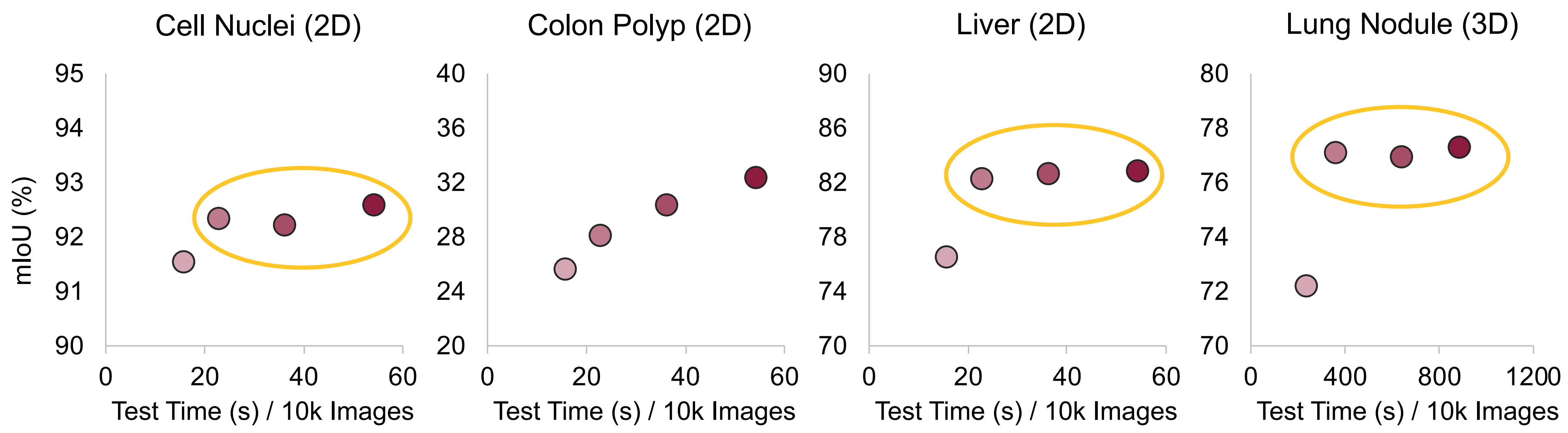
Observation: It is **independent** with the remaining parts during testing phase, but **contributes** during training phase.

- Q: Why UNet++ can be pruned when testing?
- Q: How to prune UNet++?
- Q: What's the benefit?

UNet++ L^4



- Q: Why UNet++ can be pruned when testing?
- Q: How to prune UNet++?
- Q: What's the benefit?



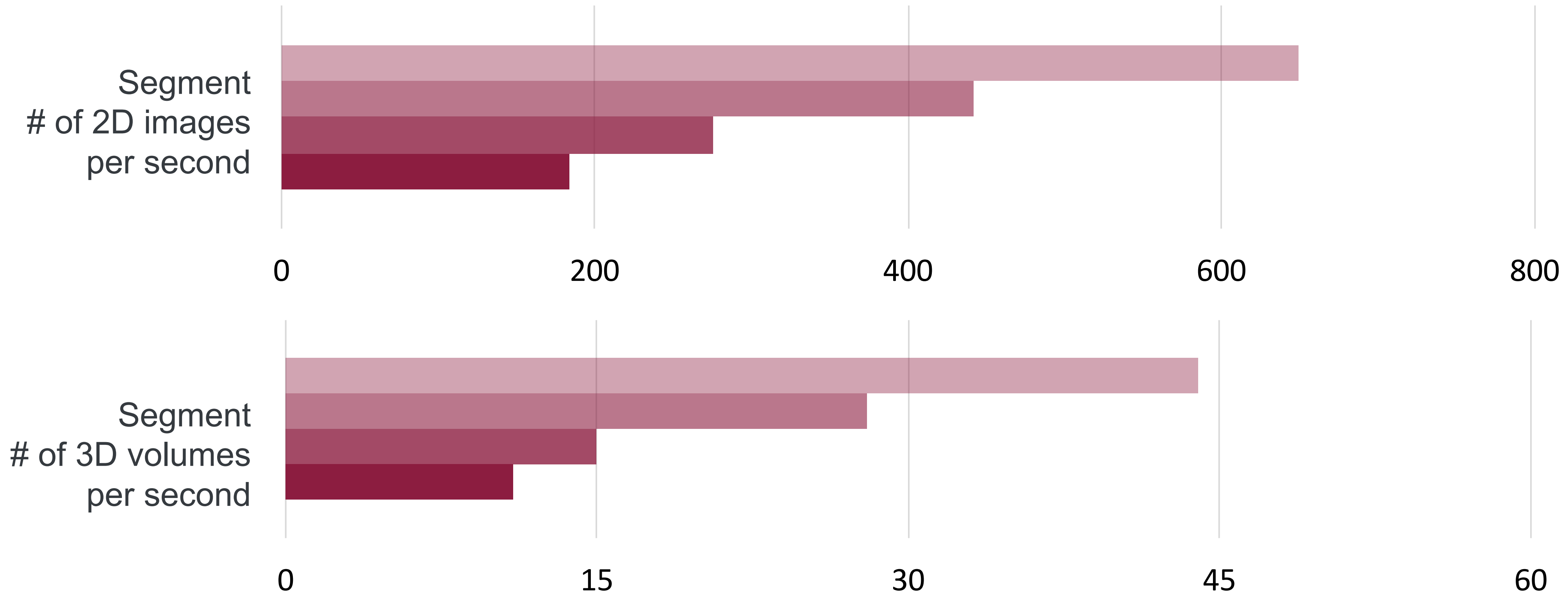
UNet++ L^1
Param = 0.1M

UNet++ L^2
Param = 0.5M

UNet++ L^3
Param = 2.2M

UNet++ L^4
Param = 9.0M

Q: What's the benefit?



In one second, it segments

- **260 2D images more,**
- **16 3D volumes more.**

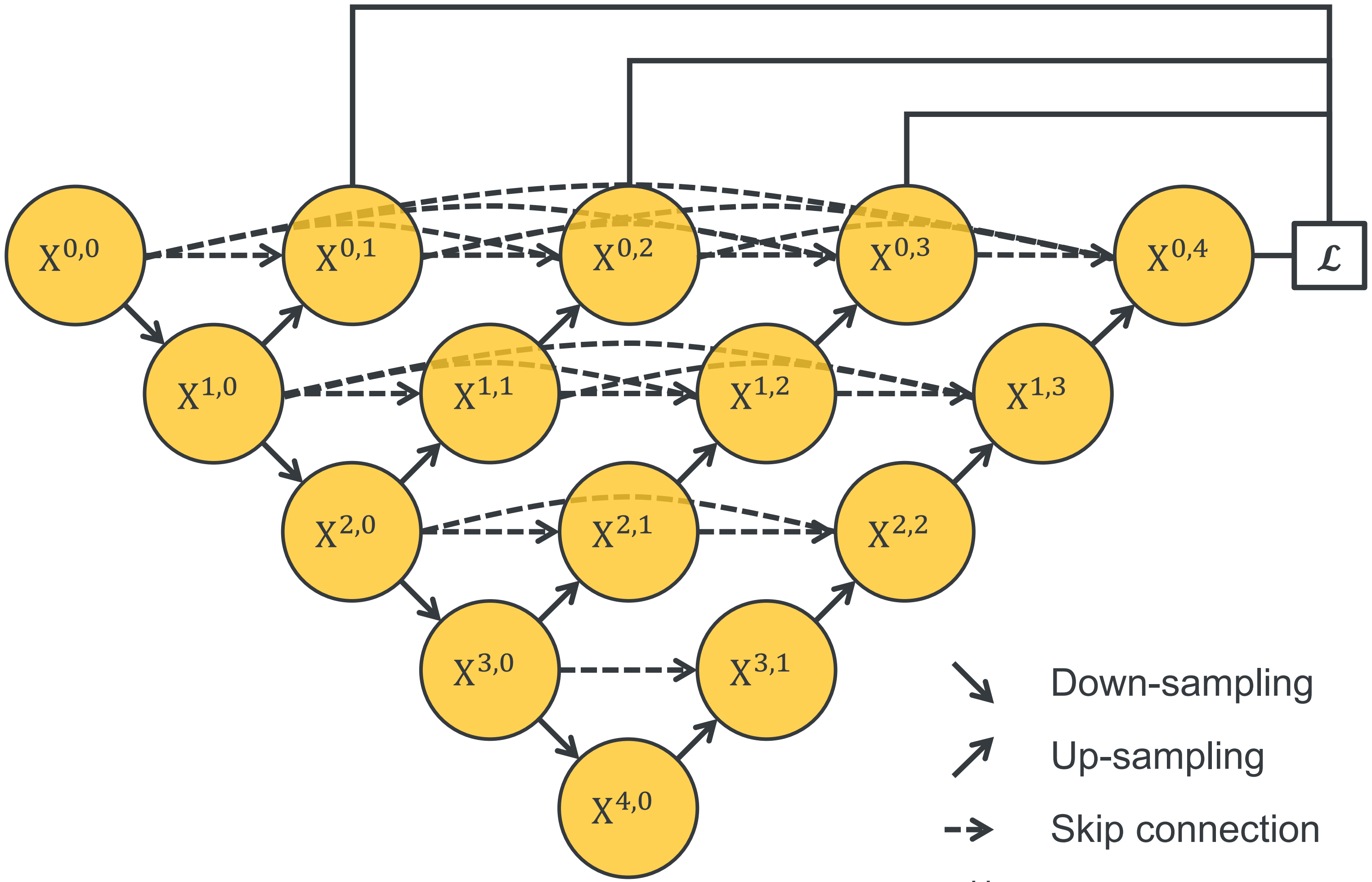
Q: What's the benefit?

UNet++ L^1 Param = 0.1M

UNet++ L^2 Param = 0.5M

UNet++ L^3 Param = 2.2M

UNet++ L^4 Param = 9.0M



- **Paper** https://arxiv.org/abs/1807.10165
- **Code** https://github.com/MrGiovanni/Nested-UNet
- **Weibo** @MrGiovanni

UNet++: A Nested U-Net Architecture for Medical Image Segmentation

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang

Arizona State University

- **Paper** https://arxiv.org/abs/1807.10165
- **Code** https://github.com/MrGiovanni/Nested-UNet
- **Weibo** @MrGiovanni