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

Chenxi Liu[1] Liang-Chieh Chen[2] Florian Schroff[2] Hartwig Adam[2] Wei Hua[2] Alan Yuille[1] Li Fei-Fei[3]
[1]Johns Hopkins University [2]Google AI [3]Stanford University

JOHNS HOPKINS UNIVERSITY

Google AI

Motivation

Problem Description:

- **Automatically design CNN architectures that surpass human expert designs**: Neural Architecture Search (NAS)
- Part of the AutoML initiative
- Modern CNNs usually follow a two-level hierarchy:
  - Inner cell level governs specific layer-wise computations
  - Outer network level controls spatial resolution changes
- Most previous approaches:
  - Focused on image classification
  - Search inner cell level; hand-specify outer network level

Our Goal:

- **NAS for dense image prediction: semantic segmentation**
  - Challenge 1: Search outer network level
  - Challenge 2: Computationally friendly

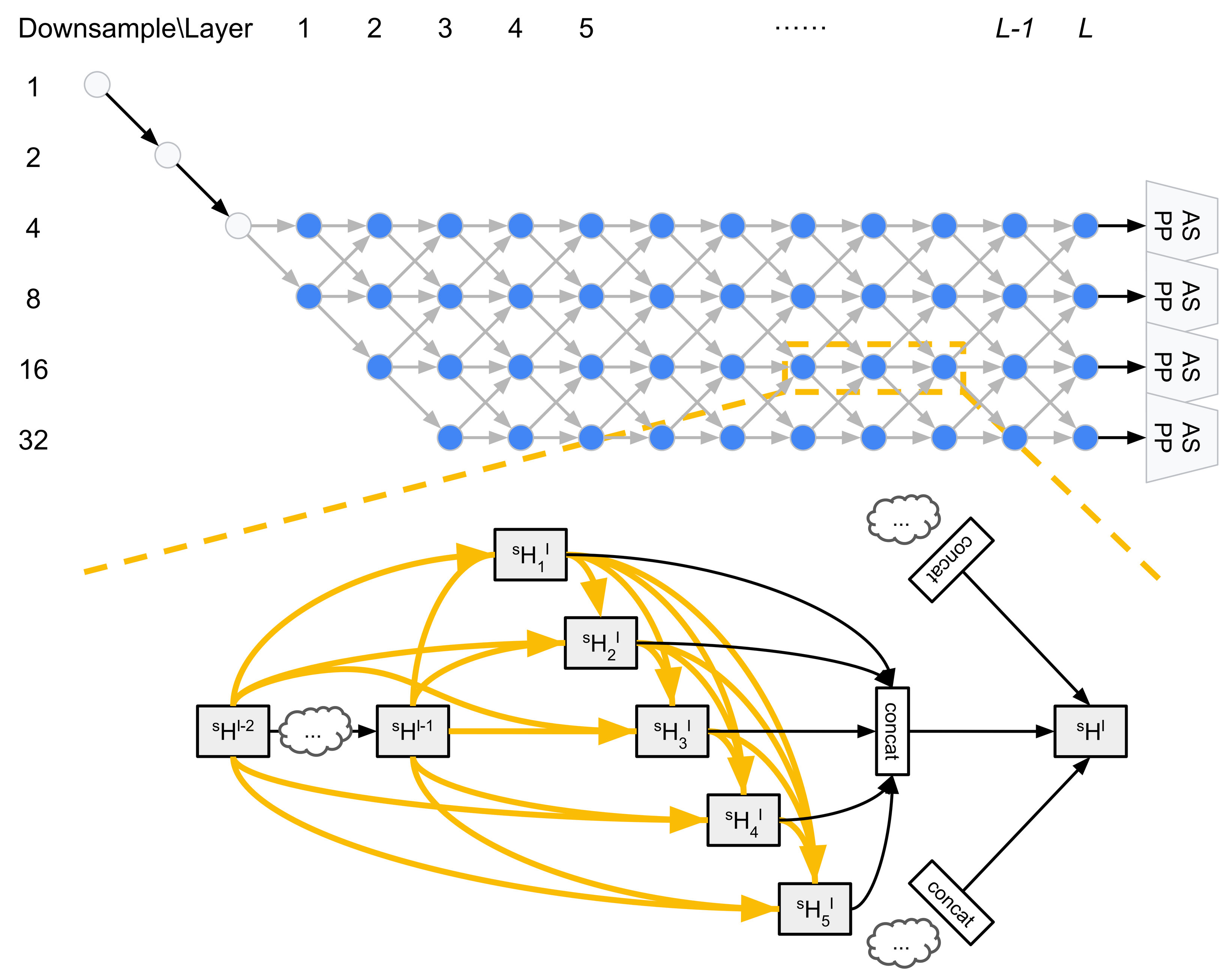
Architecture Search Space

Inner Cell Level 10^{14} **different architectures**

- Same as the one used in NASNet, PNASNet, DARTS...
- Each cell consists of $B = 5$ blocks

Outer Network Level (New) 10^5 **different architectures**

- Next layer is either twice as large, or twice as small, or same
- The smallest spatial resolution is downsampled by 32
- NAS = find a good path in this L-layer trellis



Method

Continuous Relaxation of Architectures

- *Cell Architecture*
  Approximate each operation with its continuous relaxation:
  $$\bar{O}_{j\to i}(H_j^l) = \sum_{O^k \in \mathcal{O}} \alpha^k_{j\to i} O^k(H_j^l) \quad (1)$$
  where $\alpha^k_{j\to i}$ are normalized scalars, implemented as softmax. The cell level update may be summarized as:
  $$H^l = \text{Cell}(H^{l-1}, H^{l-2}; \alpha) \quad (2)$$
- *Network Architecture*
  Associated a scalar β with each gray arrow:
  $$ {}^sH^l = \beta^l_{\frac{s}{2}\to s}\text{Cell}({}^{\frac{s}{2}}H^{l-1}, {}^sH^{l-2}; \alpha) \quad (3)$$
  $$ + \beta^l_{s\to s}\text{Cell}({}^sH^{l-1}, {}^sH^{l-2}; \alpha) + \beta^l_{2s\to s}\text{Cell}({}^{2s}H^{l-1}, {}^sH^{l-2}; \alpha)$$
  where $s = 4, 8, 16, 32$ and $l = 1, 2, \dots, L$. The scalars β are normalized such that $\beta^l_{s\to\frac{s}{2}} + \beta^l_{s\to s} + \beta^l_{s\to 2s} = 1 \quad \forall s, l$.

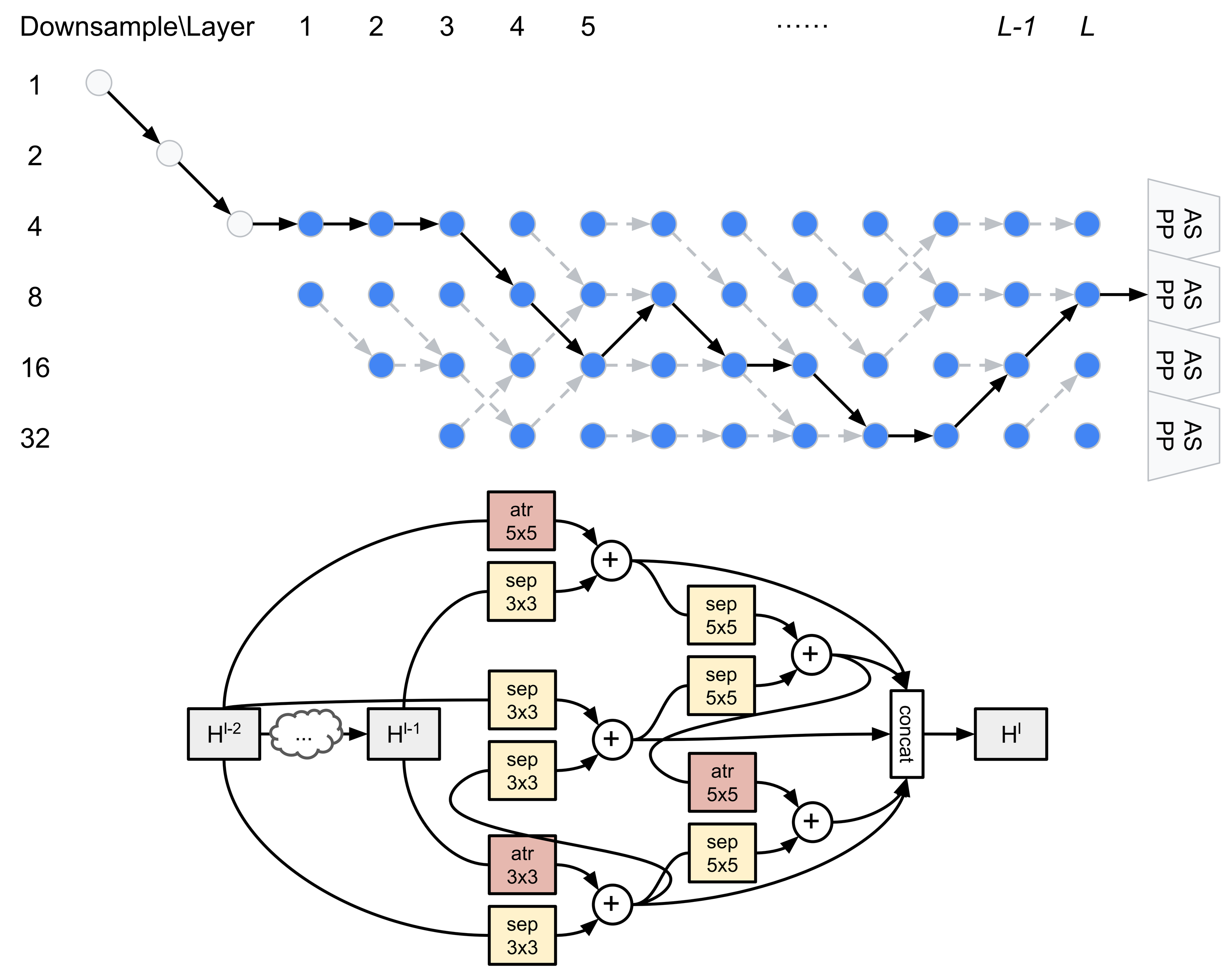
Optimization

1. Update network weights w by $\nabla_w \mathcal{L}_{trainA}(w, \alpha, \beta)$
2. Update architecture α, β by $\nabla_{\alpha,\beta} \mathcal{L}_{trainB}(w, \alpha, \beta)$

Decoding Discrete Architectures

- *Cell Architecture*: Greedy argmax
- *Network Architecture*: Viterbi algorithm

Found Architecture: Auto-DeepLab



Experiments & Results

About the Auto-DeepLab Architecture

- Downsample in the first 3/4 layers; upsample in the last 1/4
- Atrous conv often used; learned the importance of context

Results on Cityscapes

Method	ImageNet	Multi-Adds	Params	mIOU (val)
Auto-DeepLab-S		333.25B	10.15M	79.74
Auto-DeepLab-M		460.93B	21.62M	80.04
Auto-DeepLab-L		695.03B	44.42M	80.33
FRRN-A		-	17.76M	65.7
FRRN-B		-	24.78M	-
DeepLabv3+	✓	1551.05B	43.48M	79.55

Method	ImageNet	Coarse	mIOU (test)
FRRN-A			63.0
FRRN-B			71.8
Auto-DeepLab-S			79.9
Auto-DeepLab-L			80.4
Auto-DeepLab-S		✓	80.9
Auto-DeepLab-L		✓	82.1
DeepLabv3+	✓	✓	82.1
DPC	✓	✓	82.7

Results on PASCAL VOC 2012 (test set)

Method	ImageNet	COCO	mIOU (%)
Auto-DeepLab-S		✓	82.5
Auto-DeepLab-M		✓	84.1
Auto-DeepLab-L		✓	85.6
PSPNet	✓	✓	85.4
DeepLabv3+	✓	✓	87.8

Results on ADE20K (val set)

Method	ImageNet	mIOU (%)	Pixel-Acc (%)	Avg (%)
Auto-DeepLab-S		40.69	80.60	60.65
Auto-DeepLab-M		42.19	81.09	61.64
Auto-DeepLab-L		43.98	81.72	62.85
PSPNet	✓	43.51	81.38	62.45
DeepLabv3+	✓	45.65	82.52	64.09

Conclusion

- Novel: One of the first attempts to extend NAS beyond image classification to dense image prediction
- Challenging: A network level search space that augments and complements the cell level one; joint, hierarchical search
- Efficient: 3 GPU days on 321 × 321 Cityscapes image crops
- Competitive: Auto-DeepLab (always trained from scratch) outperforms other models trained from scratch significantly, and is comparable with other ImageNet-pretrained models