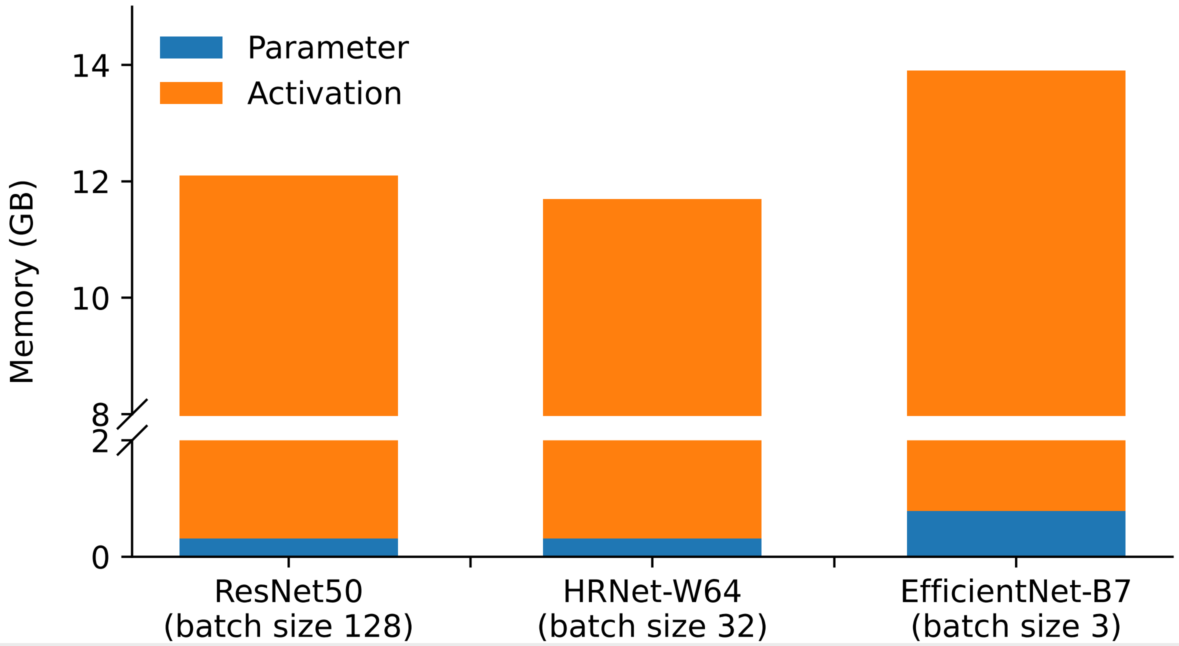
RevBiFPN: The Fully Reversible Bidirectional Feature Pyramid Network

Vitaliy Chiley, Vithursan Thangarasa, Abhay Gupta, Anshul Samar, Joel Hestness, Dennis DeCoste

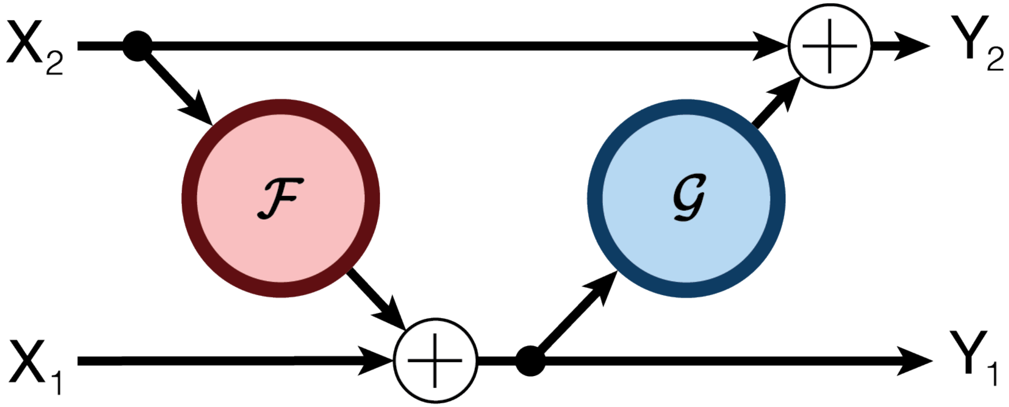
Cerebras Systems, Inc.

Network Training Memory

- Most CV networks (e.g. ResNet, EfficientNet, HRNet) use under 1GB for storing parameters, gradients, and optimizer state
  - Parameter Memory Complexity: $O(c^2 d)$
- Activations dominate the use of accelerator memory
  - Activation Memory Complexity: $O(nchwd)$
- The memory used for activations limits neural network scaling



Reversible Residual Block (RevResBlock)



RevResBlock (Gomez et al. 2017)

Memory Saving Techniques

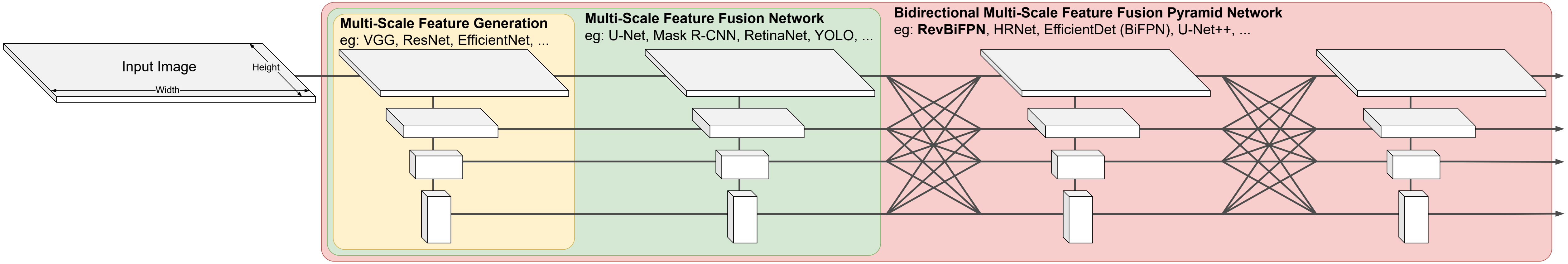
- Activation Memory Complexity: $O(nchwd)$
  - With respect to depth activation memory complexity is linear $O(d)$
- **Reverse Checkpointing**: caches activations at $\sqrt{d}$ intervals
  - Complexity: $O(nchw\sqrt{d})$
- **Reversible Recomputation**: When a network is built using invertible modules, the activations can be recomputed backwards during the backwards pass
  - Complexity: $O\left(nchw\right)$
    - Activation memory complexity is **constant** with respect to depth
  - Problem: no reversible building block operate across multi-scale features

	Memory		Compute	
	Layer Sequential	Pipelined Parallel	Forward Pass	Backward Pass
SGD Baseline	$O(D)$	$O(D^2)$	$O(D)$	$O(2D)$
with Checkpointing	$O(\sqrt{D})$	$O(D^{\frac{3}{2}})$	$O(2D)$	$O(2D)$
with Reversble Recomputation	$O(1)$	$O(D)$	$O(2D)$	$O(2D)$

BiFPN

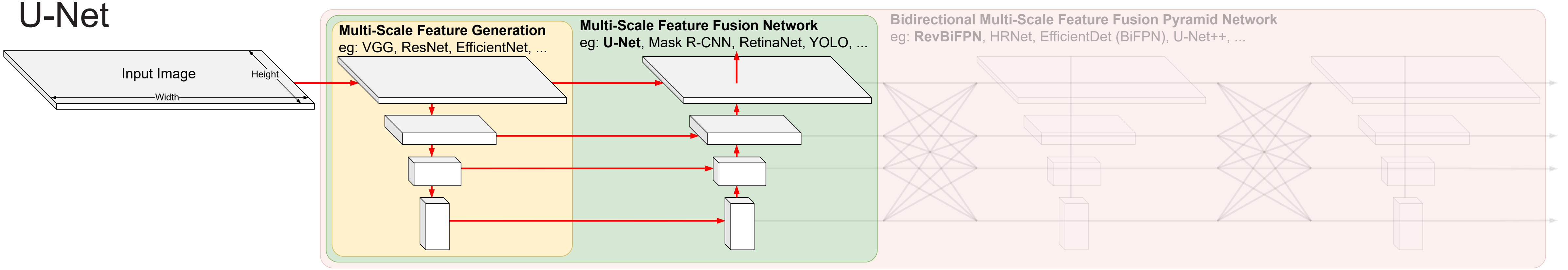
Bidirectional Feature Pyramid Networks (e.g. HRNet, EfficientDet, NAS-FPN)

- Iteratively fuse high and low resolution feature maps
  - Promotes scale invariant detection and segmentation
  - Provide local and global coherence
- Drive SOTA results for spatially sensitive tasks when paired with high resolution images
  - Consume a lot of memory for storing activations at high and low resolution scale along the entire semantic depth

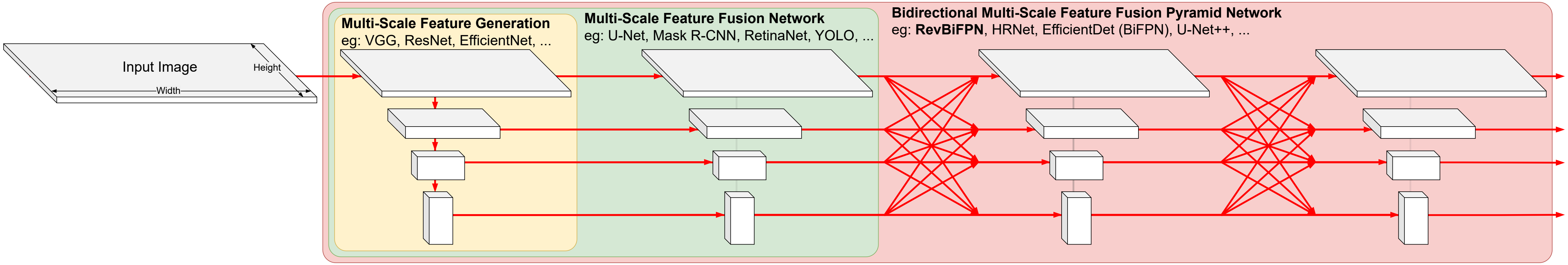


Multi-Scale Feature Networks

U-Net

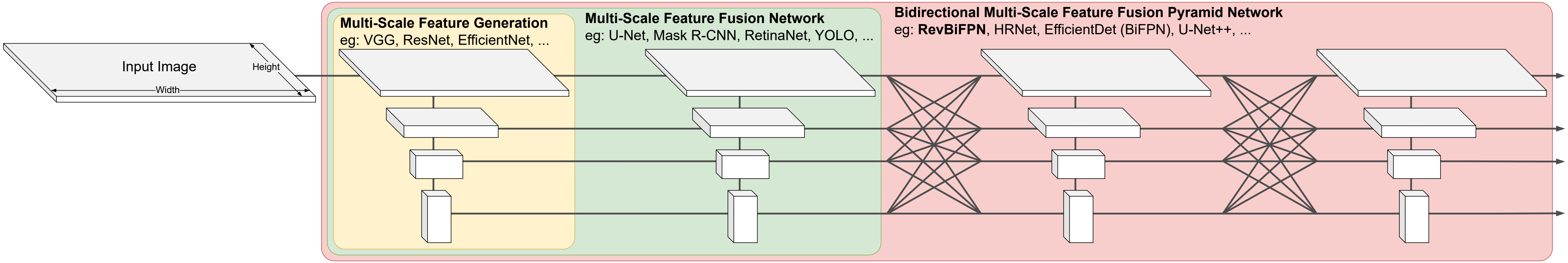


HRNet

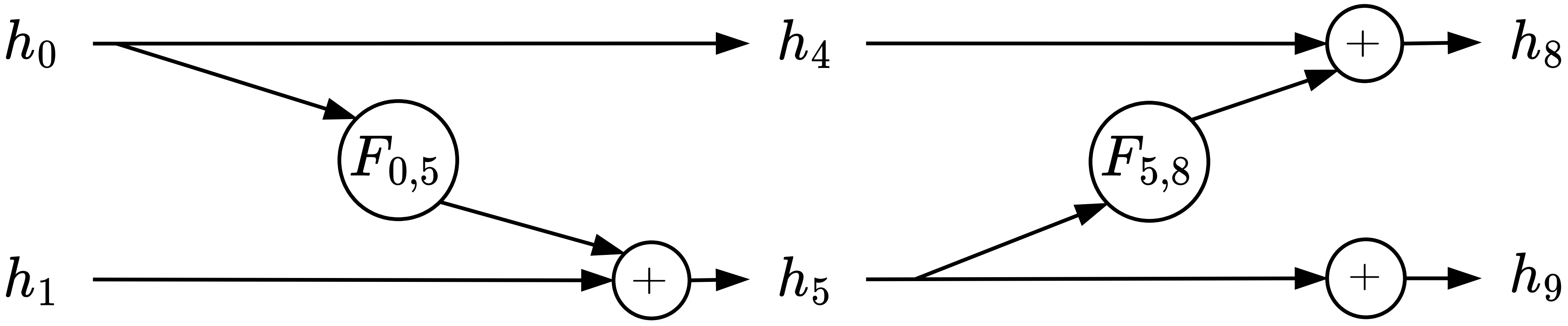


BiFPN

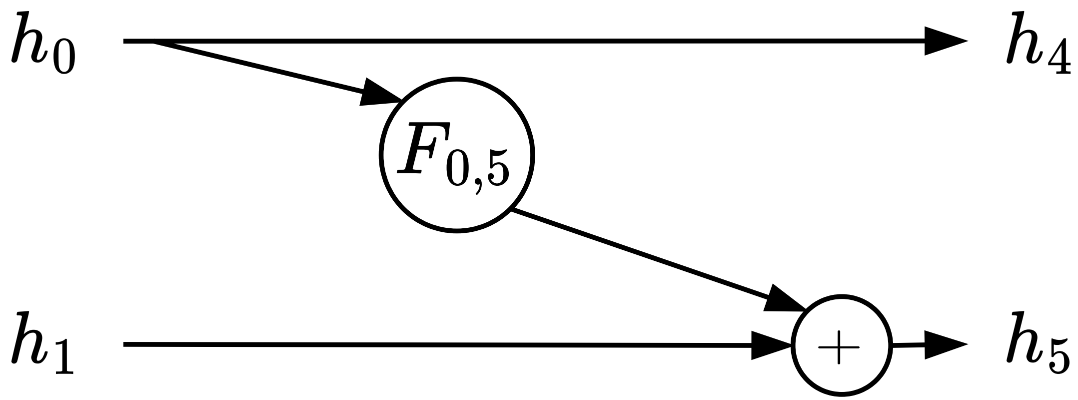
How do we apply reversible recompilation to BiFPN style networks???



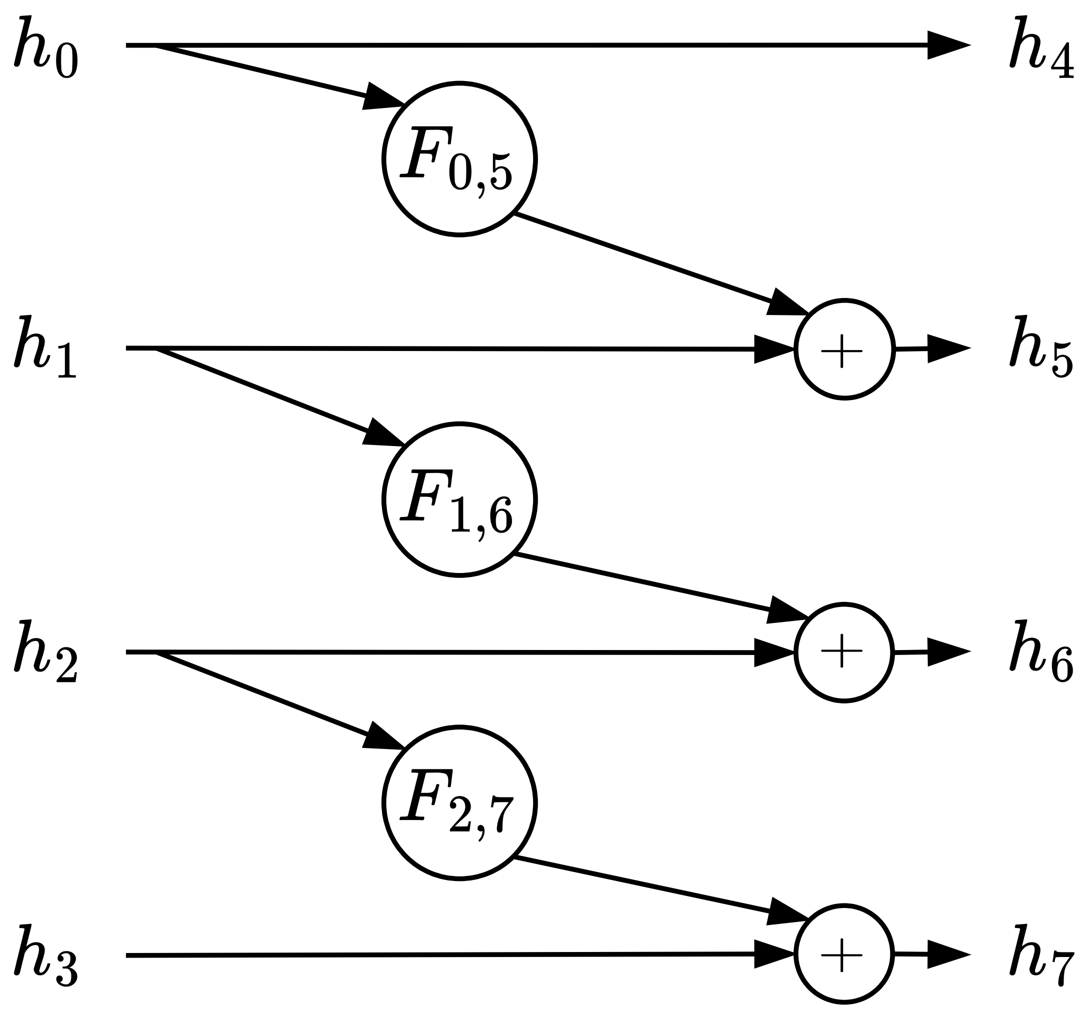
RevResBlock -> RevSilo



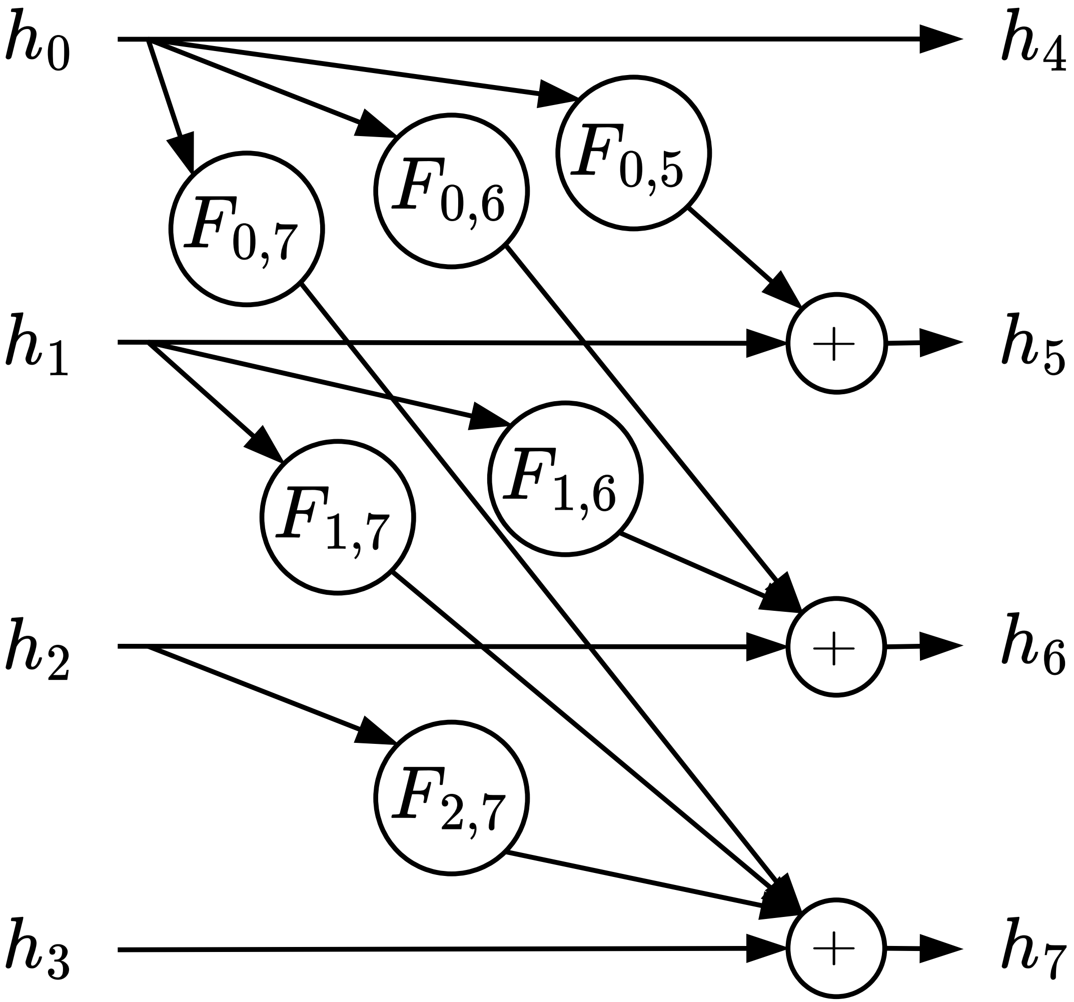
RevResBlock -> RevSilo



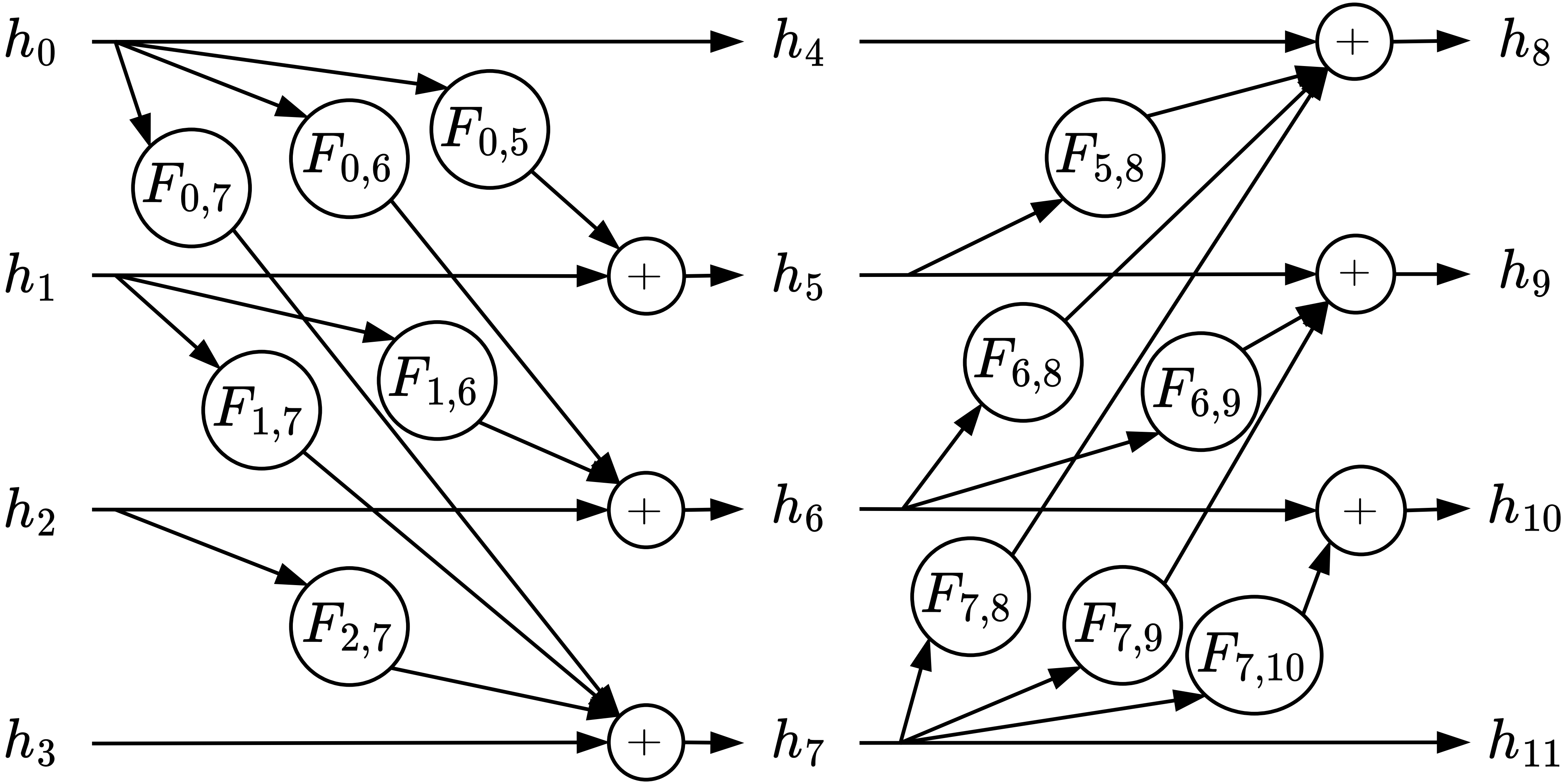
RevResBlock -> RevSilo



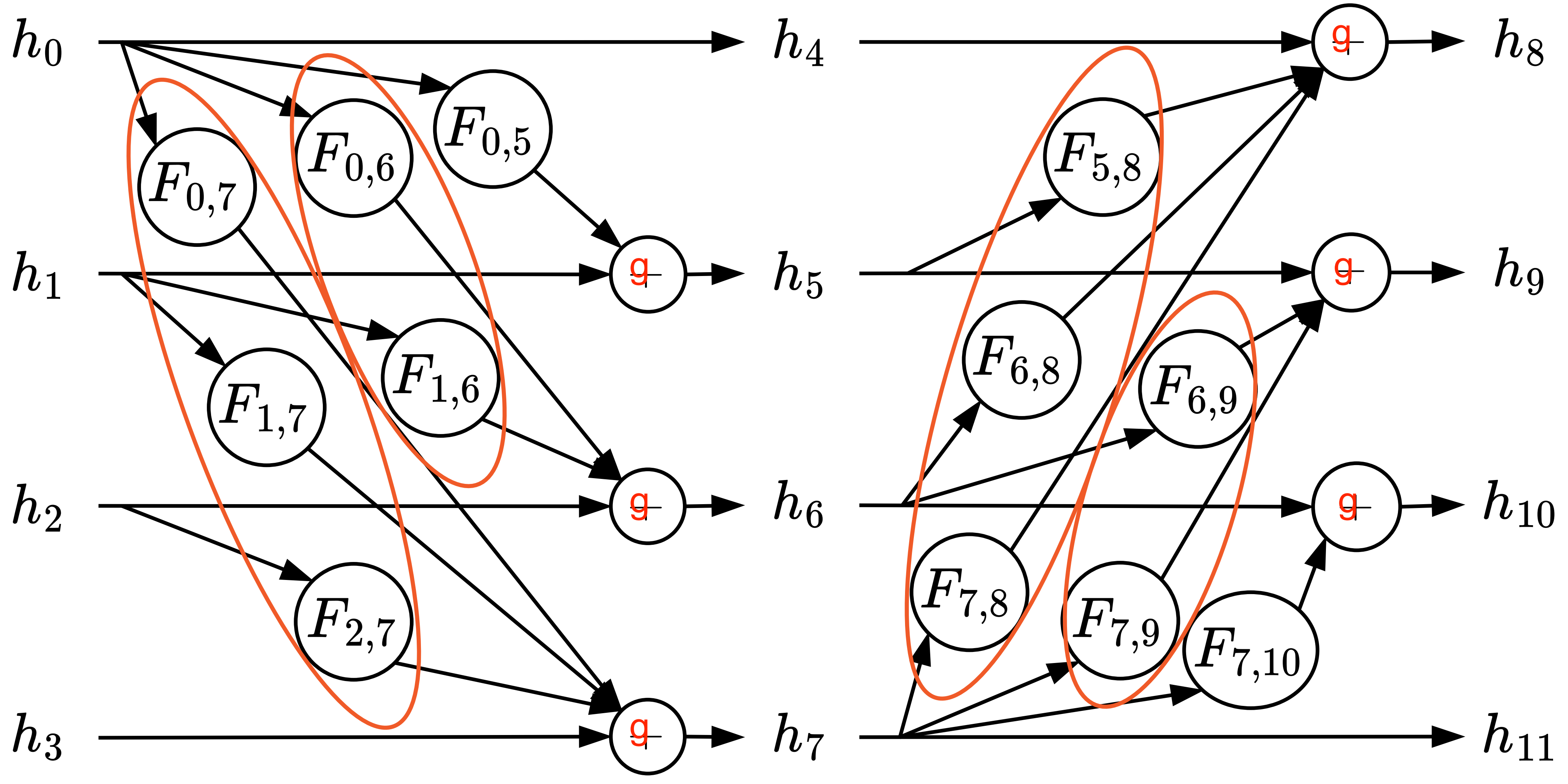
RevResBlock -> RevSilo



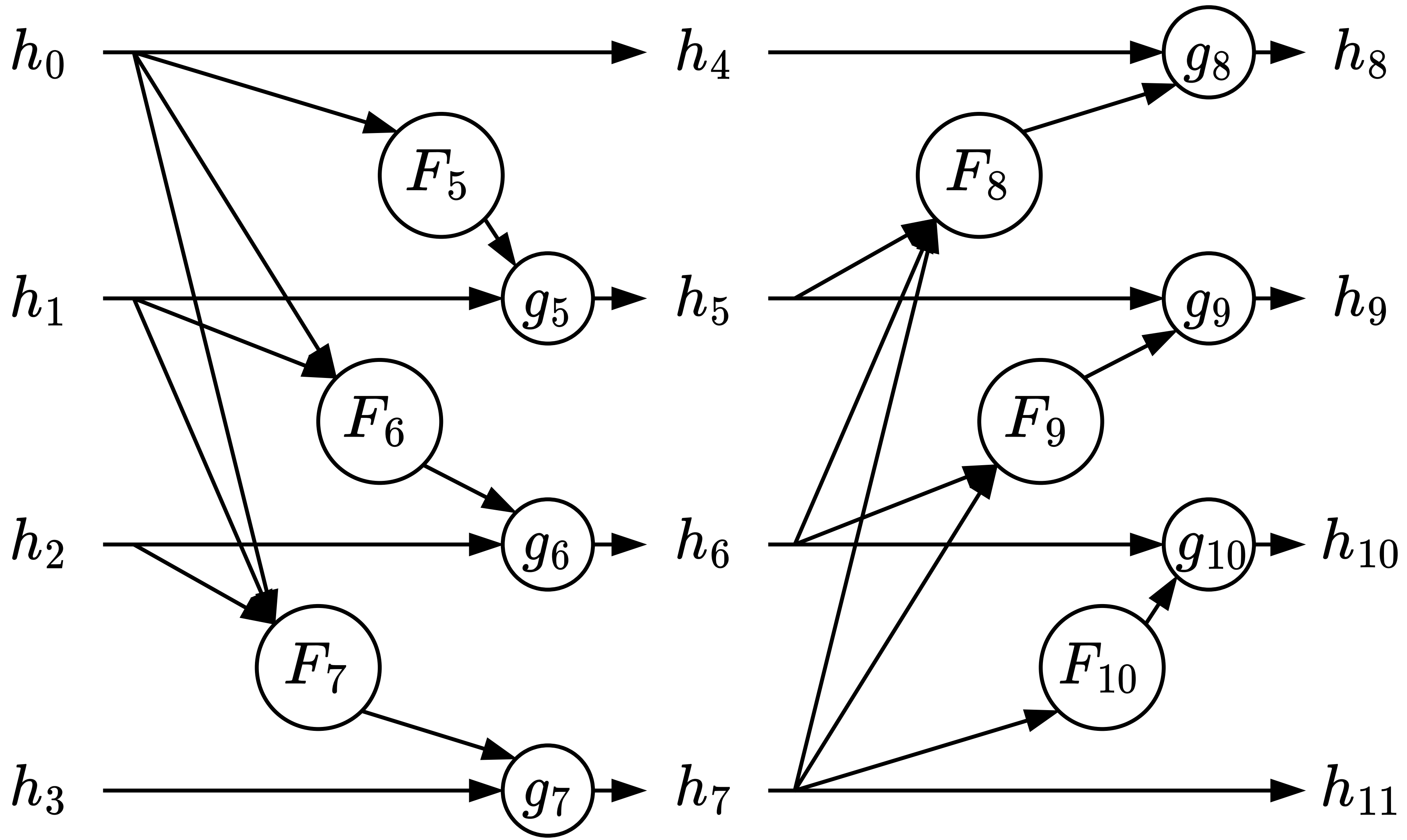
RevResBlock -> RevSilo



RevResBlock -> RevSilo

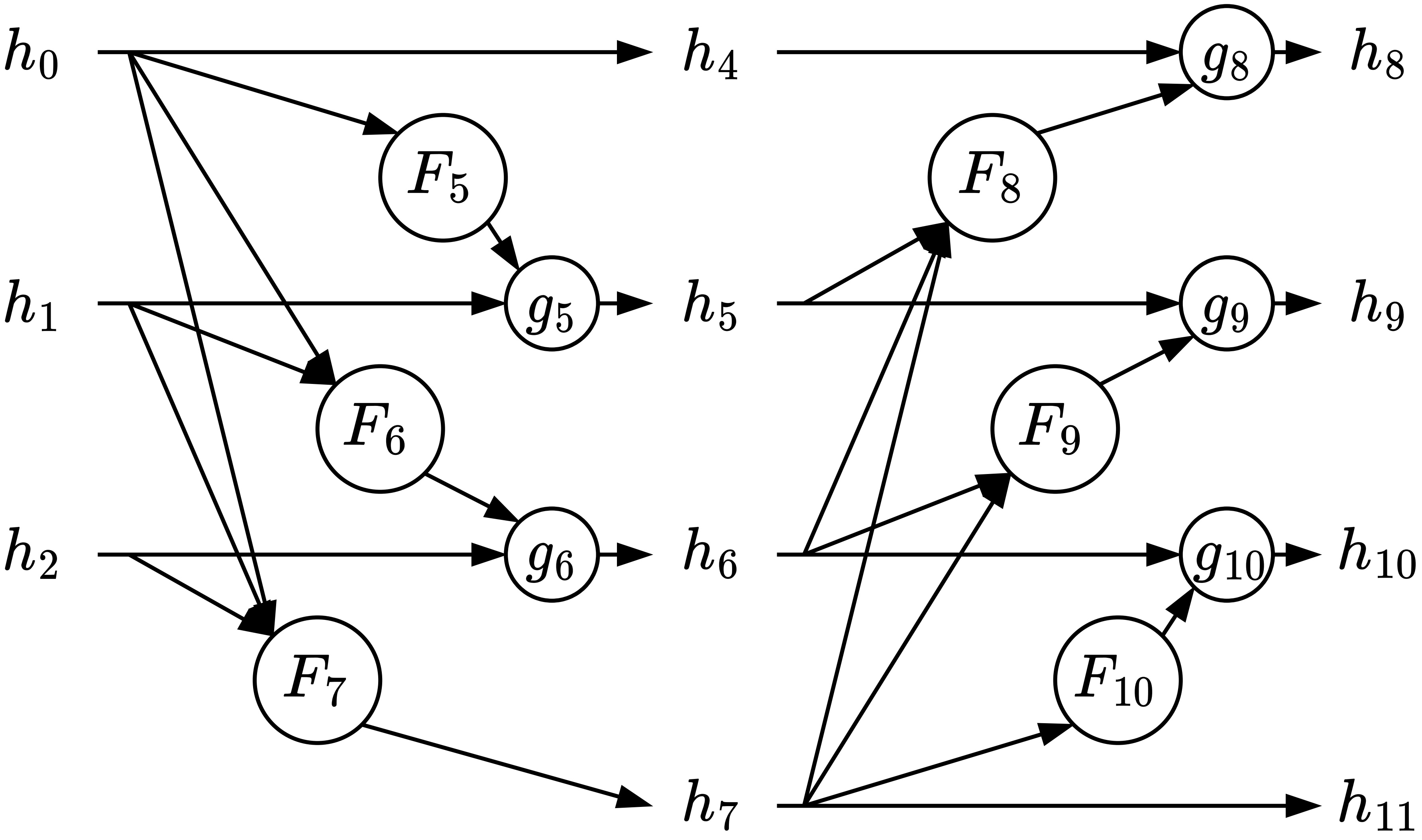


RevSilo



h_0
h_4
g_8
h_8
F_5
F_8
h_1
g_5
h_5
g_9
h_9
F_6
F_9
h_2
g_6
h_6
g_{10}
h_{10}
F_7
F_{10}
h_3
g_7
h_7
h_{11}

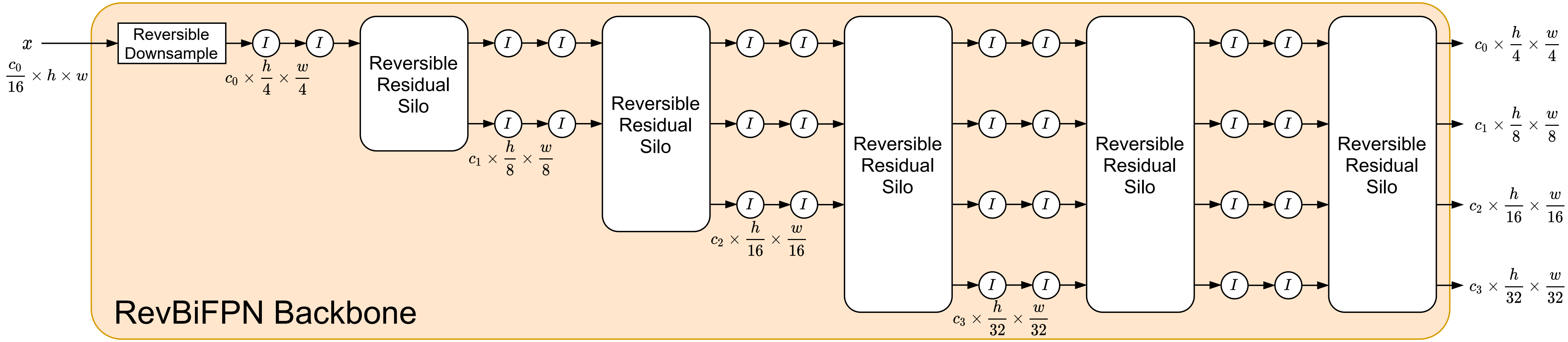
Expanding the Feature Pyramid



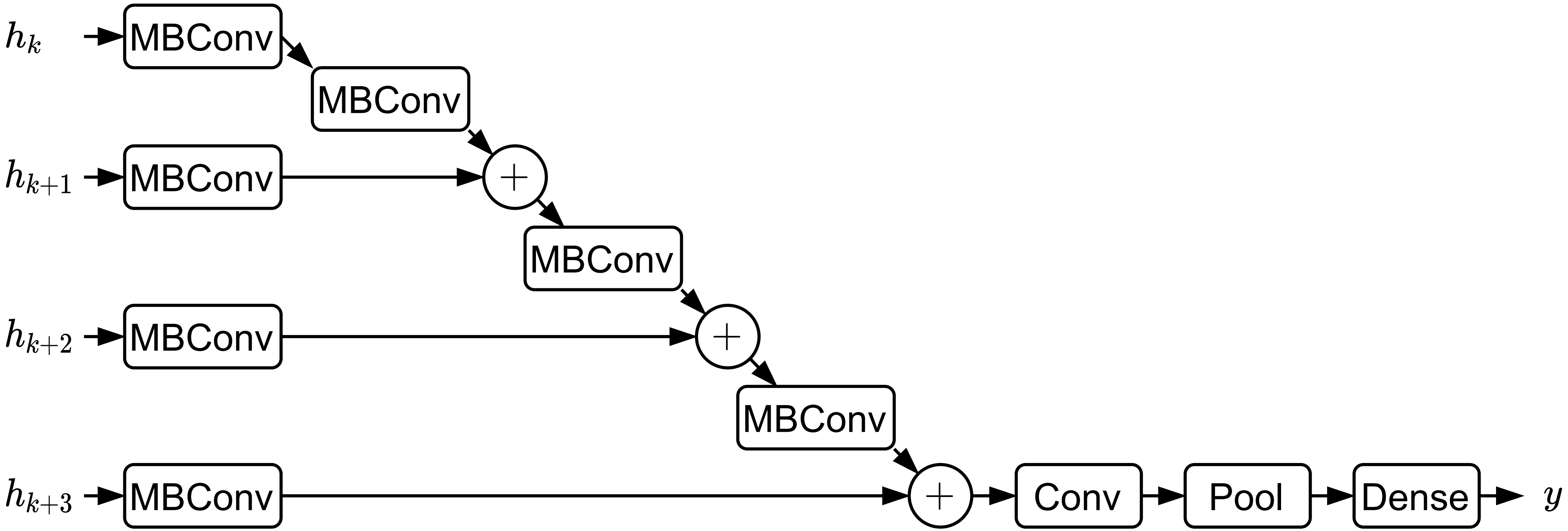
RevBiFPN

Using the RevSilo we built RevBiFPN, a fully reversible bidirectional multi-scale feature fusion pyramid network

- I are reversible residual blocks from Gomez et al. (2017)
- High level network design is similar to HRNet, but uses the MBConv block and invertible modules



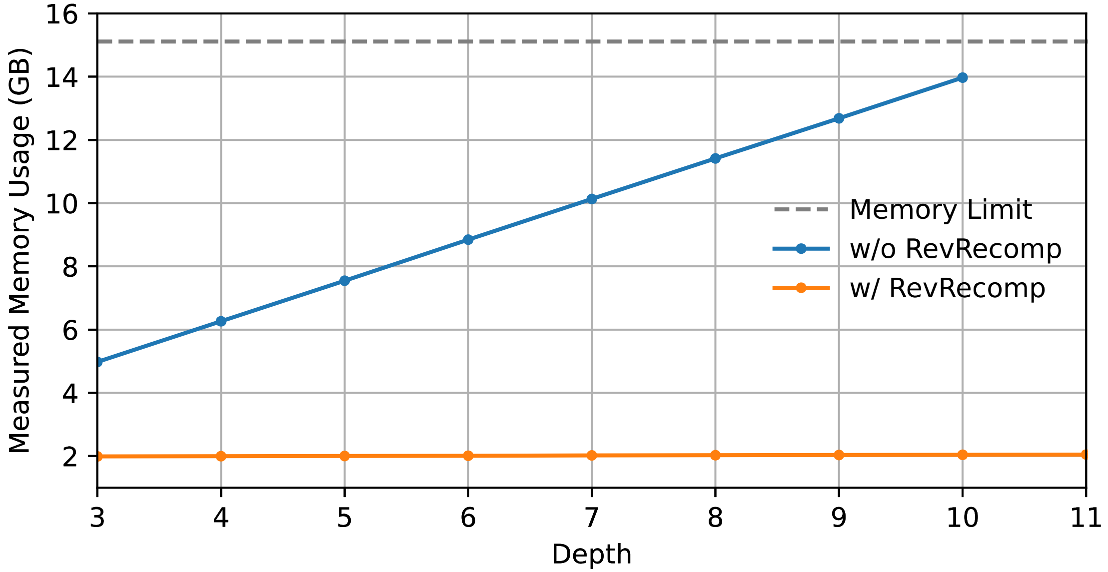
Classification Head

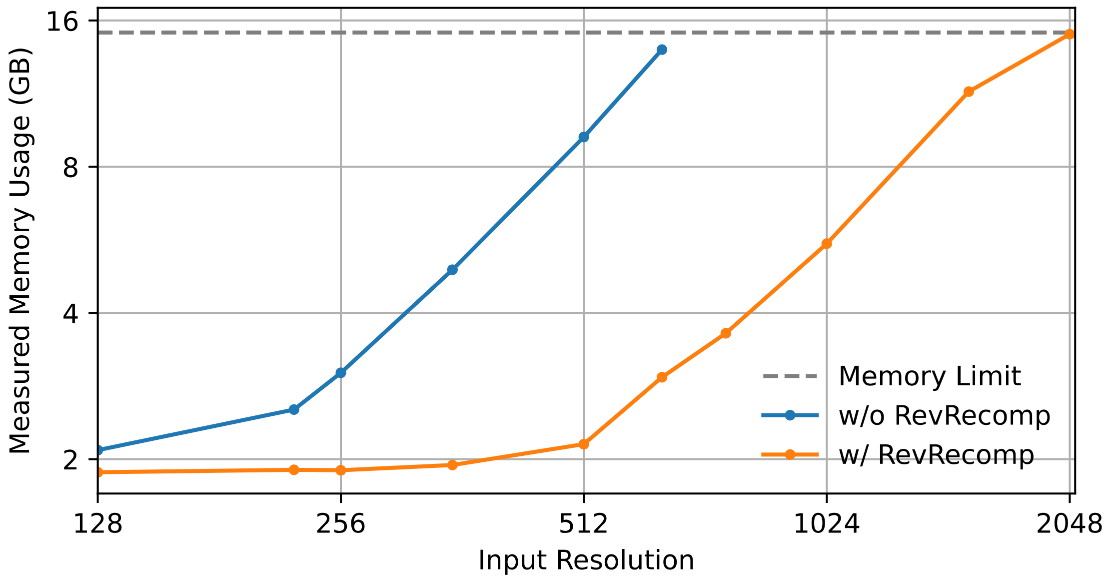


Memory With and Without Reversible Recomputation

$$O(nchwd) \text{ vs } O(nchw)$$

When scaling other dimensions, the memory complexity is still the same, but the memory has a substantial offset allowing for larger networks.

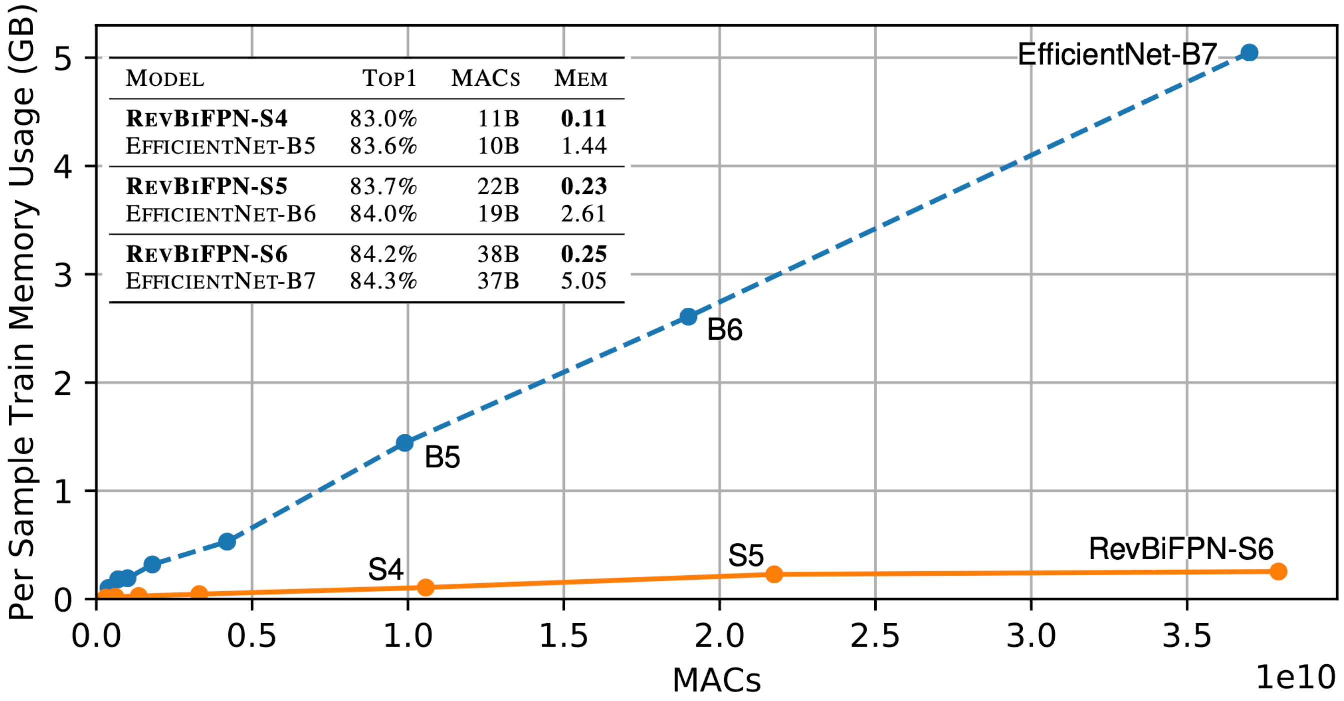




ImageNet

RevBiFPN can be scaled to have similar performance as EfficientNet but uses far less memory

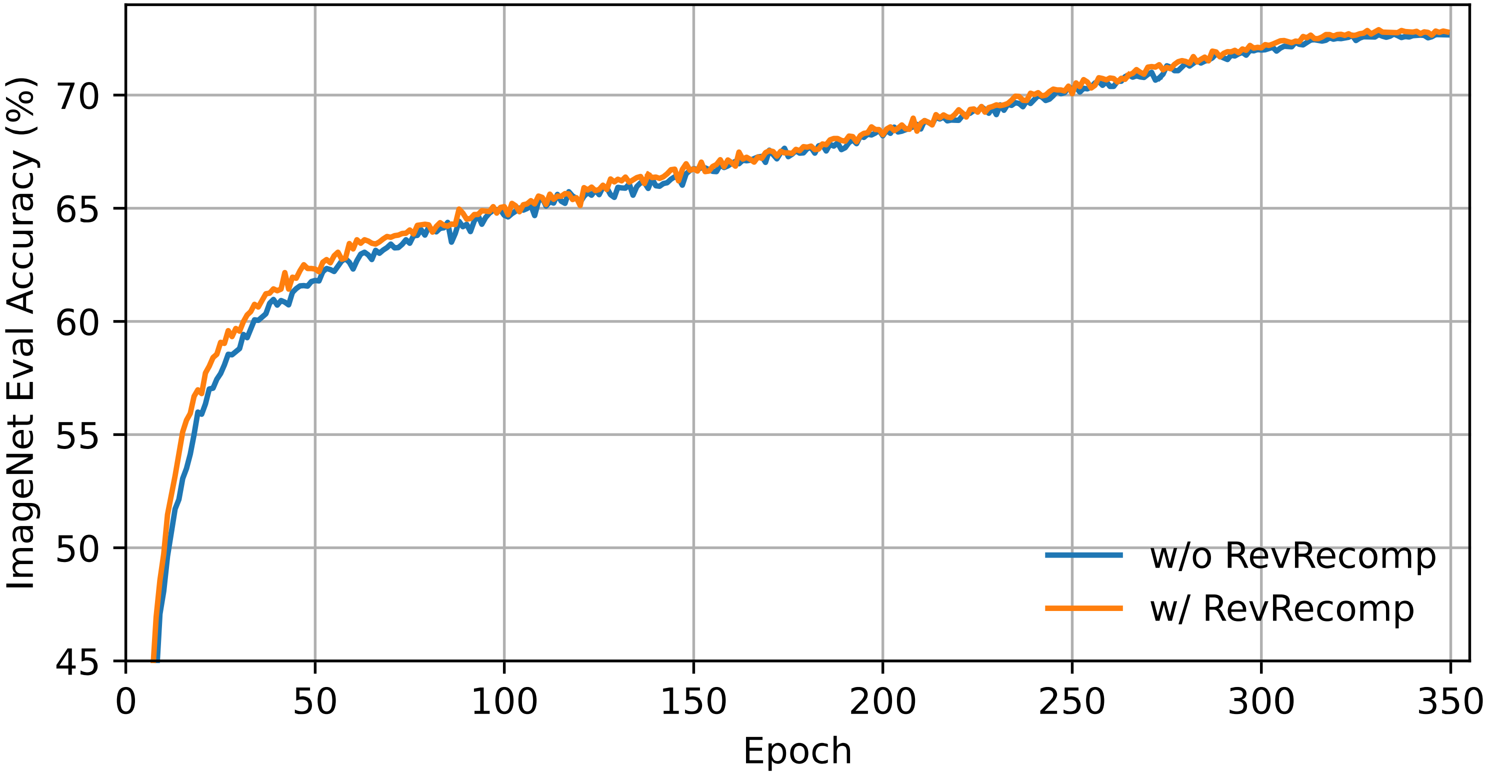
Model	Params	MACs	Top1
RevBiFPN-S0	3.42M	0.31B	72.8%
RevBiFPN-S1	5.11M	0.62B	75.9%
RevBiFPN-S2	10.6M	1.37B	79.0%
RevBiFPN-S3	19.6M	3.33B	81.1%
RevBiFPN-S4	48.7M	10.6B	83.0%
RevBiFPN-S5	82.0M	21.8B	83.7%
RevBiFPN-S6	142.3M	38.1B	84.2%



Training With and Without Reversible Recomputation

Training with reversible recomputation is nearly indistinguishable from regular training

- No approximations -> little reconstruction error



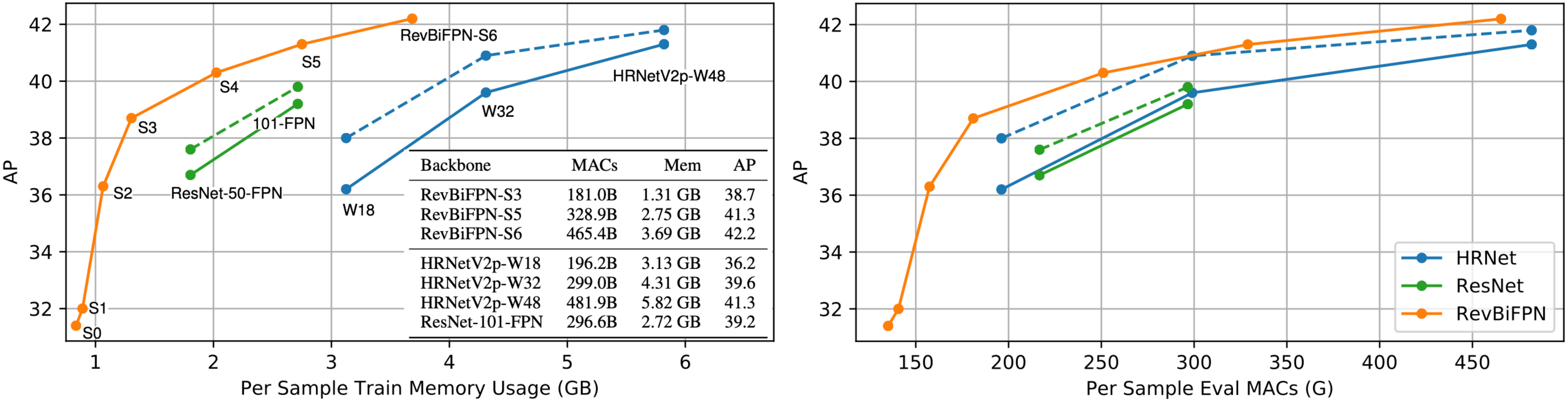
Computational Overhead of Recomputation

Theoretical Slowdown: 33%

Model	Slowdown
RevBiFPN-S0	25.02%
RevBiFPN-S2	21.96%
RevBiFPN-S4	15.73%
RevBiFPN-S6	12.73%

MS COCO Detection

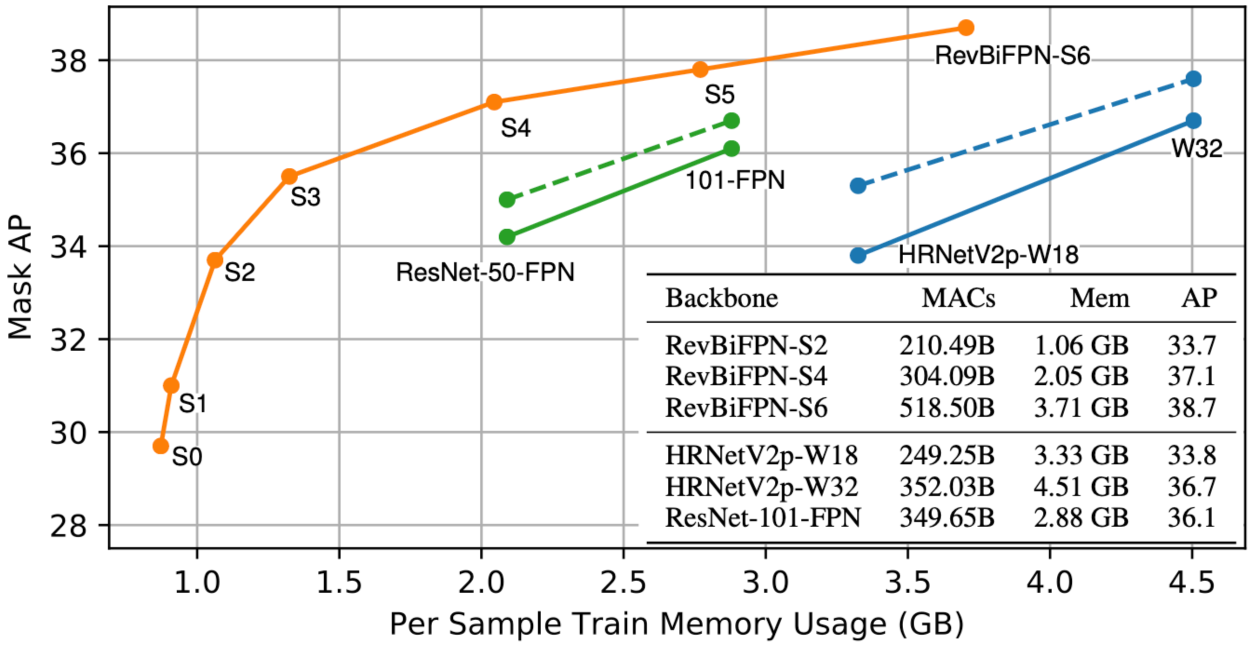
Head: Faster R-CNN (from MMDetection)



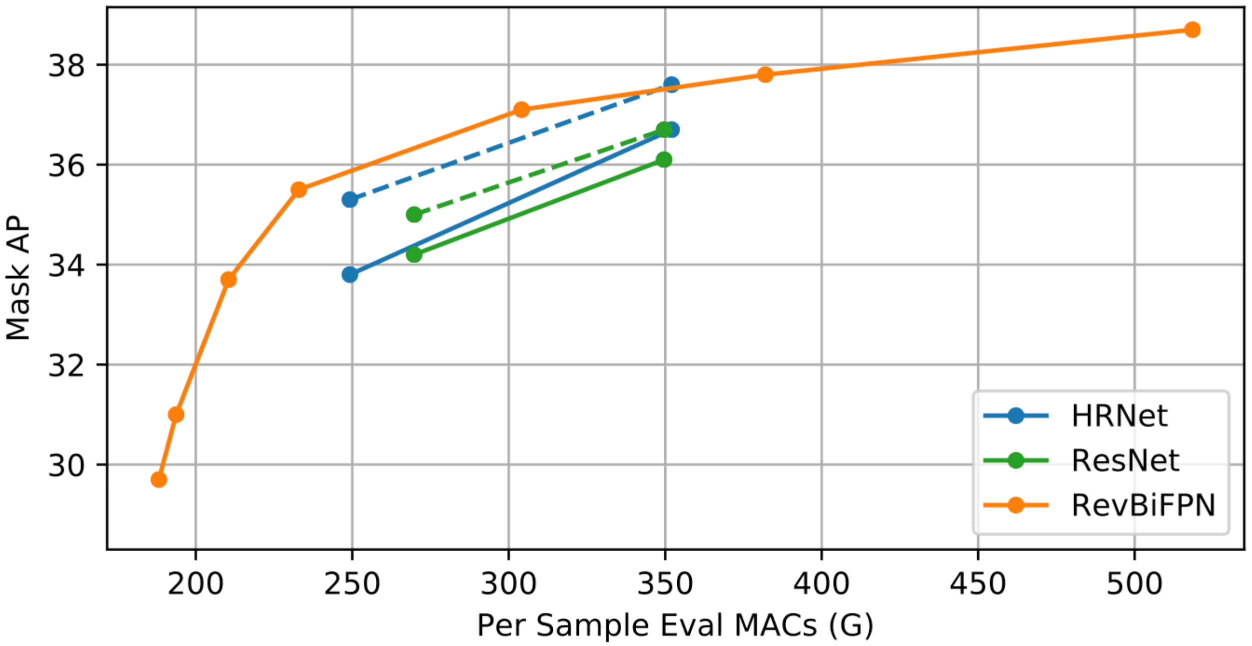
Backbone	MACs	Mem	AP
RevBiFPN-S3	181.0B	1.31 GB	38.7
RevBiFPN-S5	328.9B	2.75 GB	41.3
RevBiFPN-S6	465.4B	3.69 GB	42.2
HRNetV2p-W18	196.2B	3.13 GB	36.2
HRNetV2p-W32	299.0B	4.31 GB	39.6
HRNetV2p-W48	481.9B	5.82 GB	41.3
ResNet-101-FPN	296.6B	2.72 GB	39.2

MS COCO Instance Segmentation

Head: Mask R-CNN (from MMSegmentation)

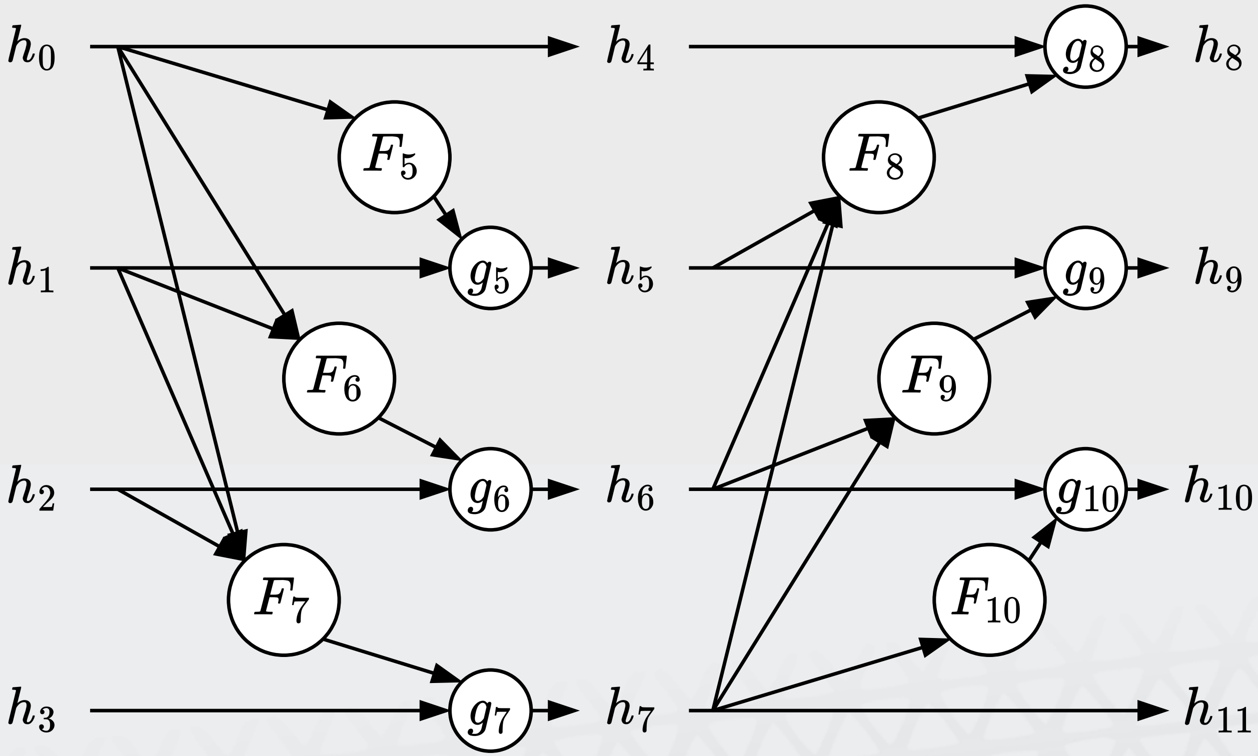


Backbone	MACs	Mem	AP
RevBiFPN-S2	210.49B	1.06 GB	33.7
RevBiFPN-S4	304.09B	2.05 GB	37.1
RevBiFPN-S6	518.50B	3.71 GB	38.7
HRNetV2p-W18	249.25B	3.33 GB	33.8
HRNetV2p-W32	352.03B	4.51 GB	36.7
ResNet-101-FPN	349.65B	2.88 GB	36.1



Future Work

- Dig into RevBiFPN's sensitivity to
  - Reconstruction error
  - Sparsity
  - Different Normalization Methods
  - Gradient delay (ASGD)
- Tune network / building block for different compute platforms
  - Improve network design using NAS
- Apply to 3D tasks and other memory intensive tasks
- Apply to flow based generation
  - RevBiFPN can iteratively fuse high and low resolution feature maps to promote local and global coherence in flow based generation


h_0
h_1
h_2
h_3
F_5
F_6
F_7
g_5
g_6
g_7
h_4
h_5
h_6
h_7
F_8
F_9
F_{10}
g_8
g_9
g_{10}
h_8
h_9
h_{10}
h_{11}