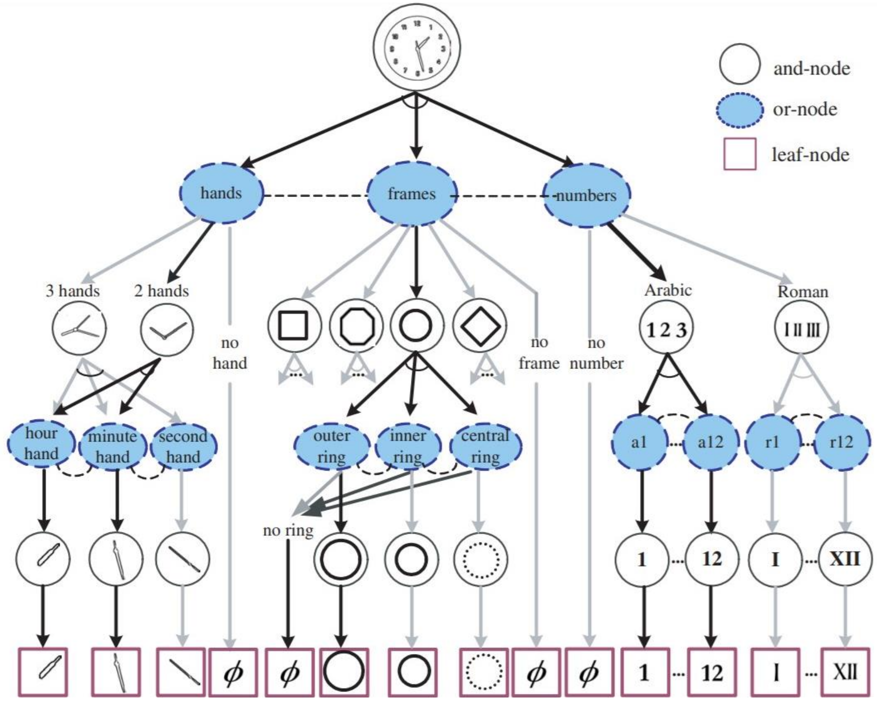
Inducing Hierarchical Compositional Model by Sparsifying Generator Network

Xianglei Xing, Tianfu Wu, Song-Chun Zhu, Ying Nian Wu

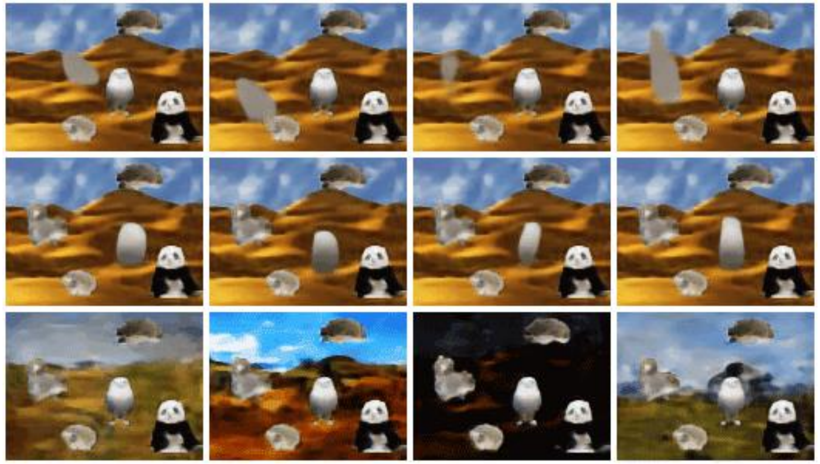
Graph representations of images

- Scenes can be decomposed into a hierarchical graph of sub-components
  - Capture the inherent diversity in images
  - Capture the semantically correlated parts
  - Capture the hierarchy of components
- Unsupervised parsing is a well studied
  - Fully supervised (5+ papers in CVPR 2020)
  - Unsupervised (IODINE, MONET, AIR, SQAIR)
- Generation less well studied
  - GENESIS (no hierarchy)
  - *Learning to manipulate objects (no hierarchy)*

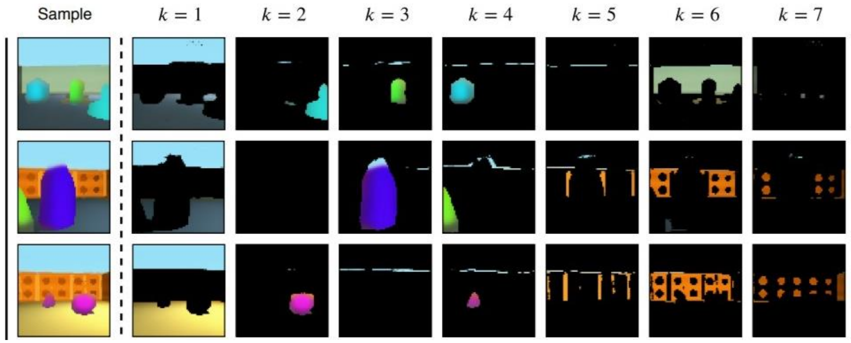


Unsupervised decomposition of visual inputs

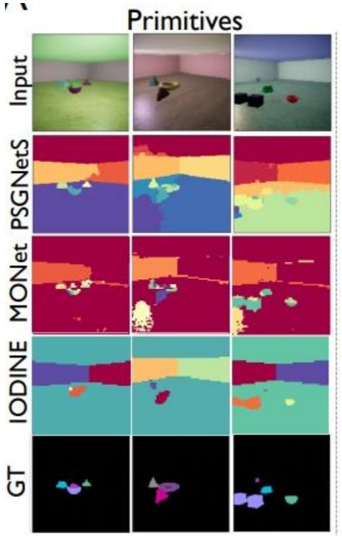
Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects



Learning to Manipulate Individual Objects in an Image



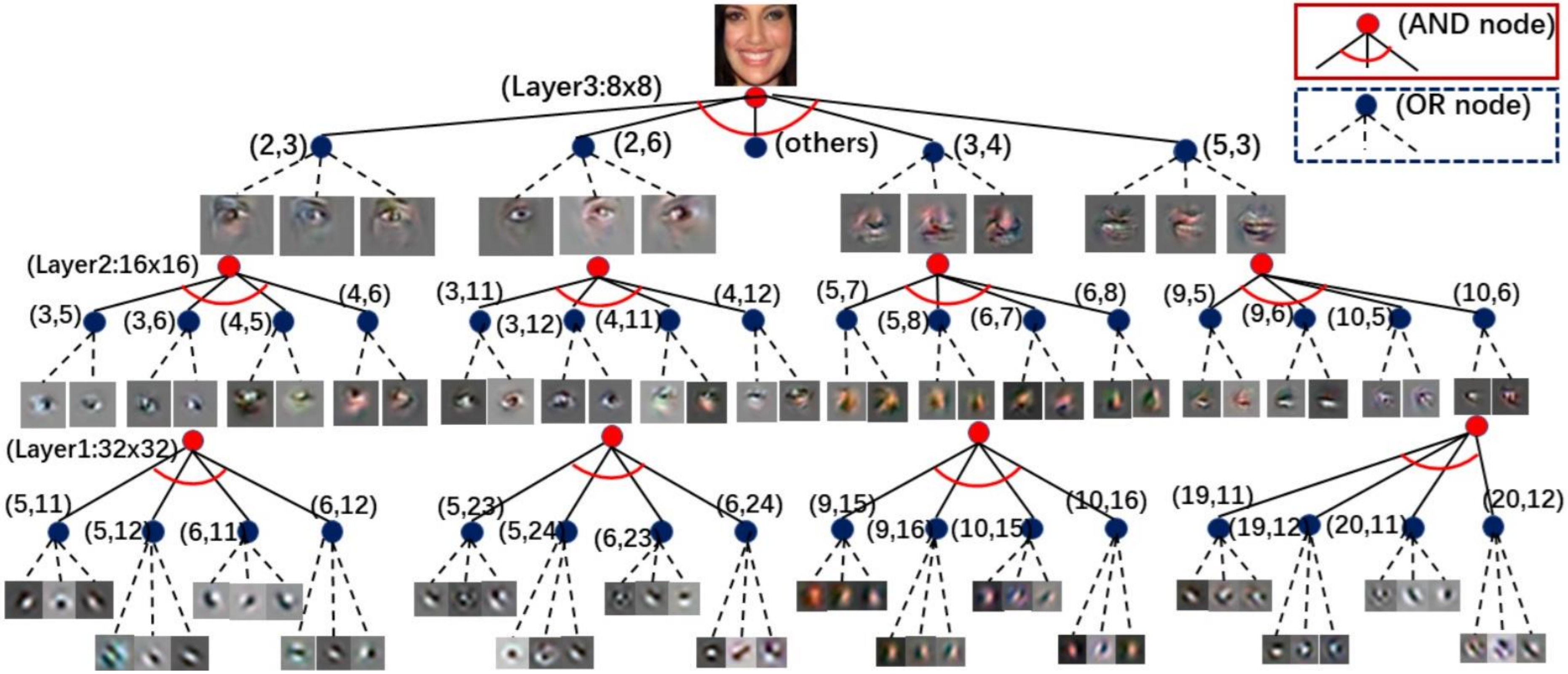
GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations



Inducing sparsity

- L1 norm
- SparseMax, SparsestMax, Sparse Switchable Normalization (SSN)
- Routing (requires reinforce to backprop gradients)
- Gumble-Softmax (Ben Poole)

- Top-k (what they use)


(AND node)
(OR node)
(Layer3:8x8)
(2,3)
(2,6)
(others)
(3,4)
(5,3)
(Layer2:16x16)
(3,5)
(3,6)
(4,5)
(4,6)
(3,11)
(3,12)
(4,11)
(4,12)
(5,7)
(5,8)
(6,7)
(6,8)
(9,5)
(9,6)
(10,5)
(10,6)
(Layer1:32x32)
(5,11)
(5,12)
(6,11)
(6,12)
(5,23)
(5,24)
(6,23)
(6,24)
(9,15)
(9,16)
(10,15)
(10,16)
(19,11)
(19,12)
(20,11)
(20,12)

Why sparsity?

- Helps learn interpretable basis functions
  - Forces the network to compress meaningful representations into a few activations
- Potentially more efficient (not in their case)
  - Their implementations does not save computation
- Their sparsity operation is non-differentiable
  - In this respect, this design choice is questionable

```
input_vector # of shape N, C, H, W
top_activations = top_k(input_vector, dim=1)
input_vector[input_vector < top_activations] = 0
```

```
input_vector # of shape N, C, H, W
input_vector = input_vector.reshape(N, C, H*W)
top_activations = top_k(input_vector, dim=2)
input_vector[input_vector < top_activations] = 0
```

On the energy based “critic”

- Similar to the non-saturating variant of the GAN loss
- They have a section on it in paper, but in code they use WGAN-GP
- soft-reverse-KL
- Full proof available on request

$$\min_{\Theta}\max_{\Phi} T(\Theta,\Phi),$$
$$T(\Theta,\Phi) = H(P(Y), P(Y;\Theta,\mathbf{k})) \tag{18}$$
$$- H(P(Y), P(Y;\Phi)) + H(P(Y;\Theta,\mathbf{k}), P(Y;\Phi)),$$

$$\mathbb{E}_{z\sim p_z(z)}\left[\nabla_\theta \log(1-D^*(G_\theta(z)))|_{\theta=\theta_0}\right] = \nabla_\theta 2\mathcal{D}_{JS}(p_r\|p_g)|_{\theta=\theta_0}$$

We then subtract the gradient for JSD from the gradient for KL, and that gives us:

$$\nabla_\theta \mathcal{D}_{KL}(p_{g_\theta}\|p_r) - 2\mathcal{D}_{JS}(p_r\|p_g)|_{\theta=\theta_0} = \nabla_\theta \mathbb{E}_{z\sim p_z(z)}\left[-\log(D^*(G_\theta(z)))|_{\theta=\theta_0}\right]$$

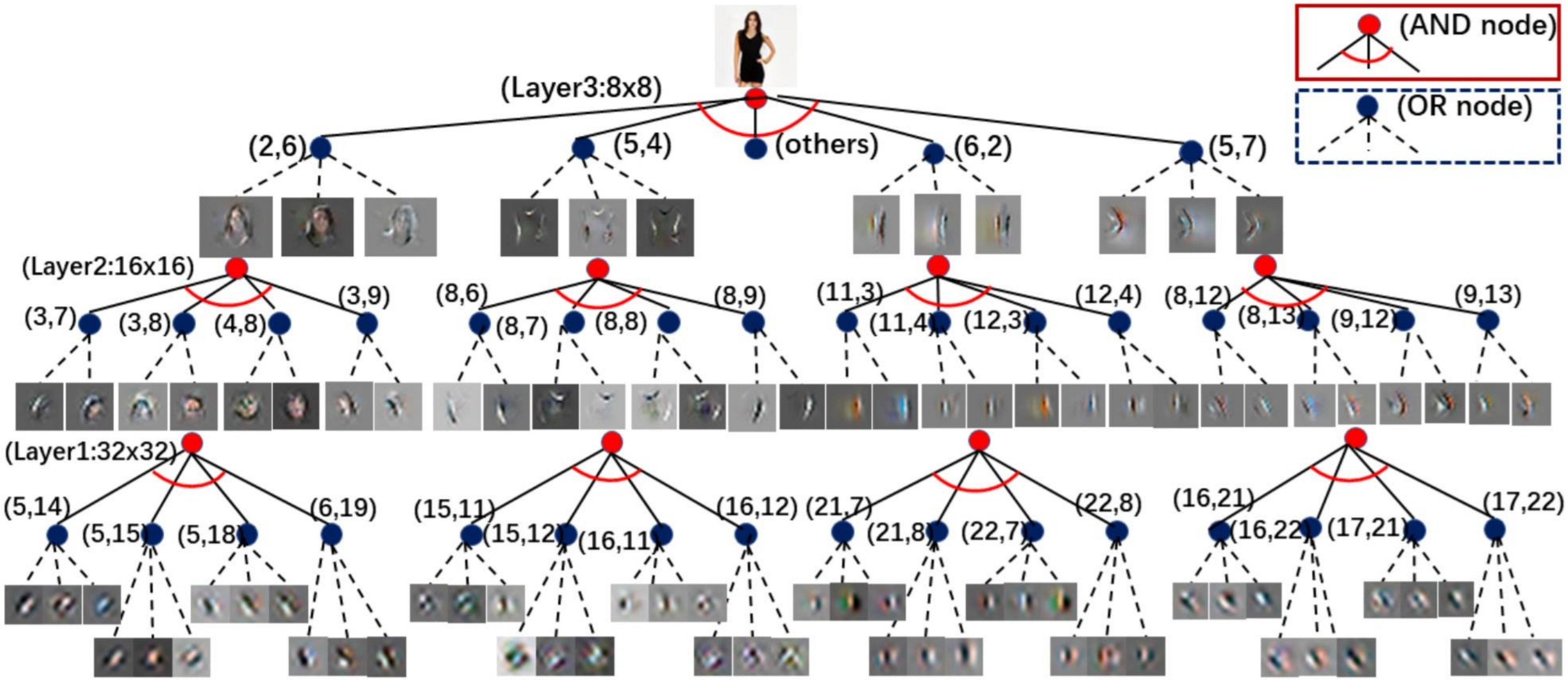
We then re-write the result into the reverse KL formulation:

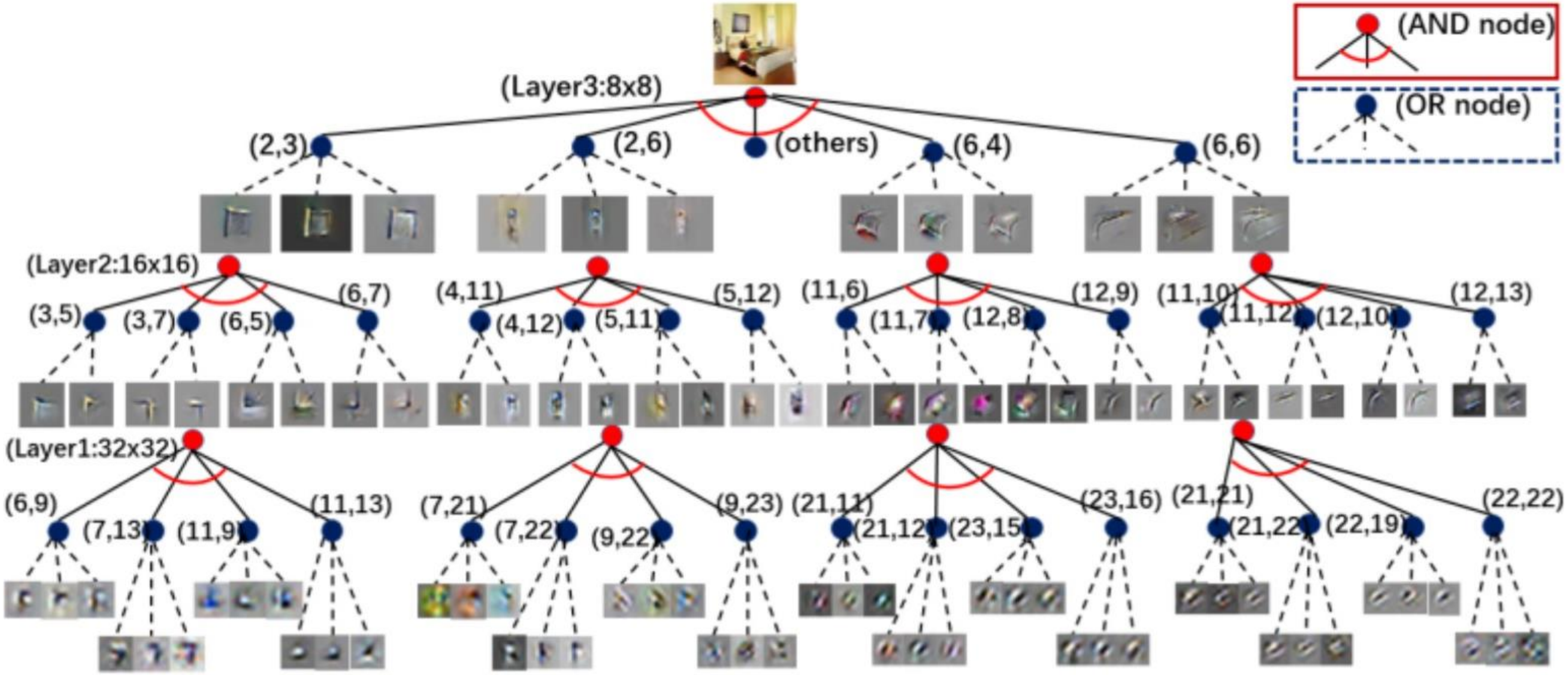
$$\begin{aligned}
&\mathcal{D}_{KL}(p_g\|p_r) - 2\mathcal{D}_{JS}(p_r\|p_g)\\
&= \int_x p_g(x)\log\frac{p_g(x)}{p_r(x)} - \Big(\int_x p_r(x)\log\frac{2p_r(x)}{p_r(x)+p_g(x)}\Big) - \Big(\int_x p_g(x)\log\frac{2p_g(x)}{p_r(x)+p_g(x)}\Big)\\
&= \int_x \left[p_g(x)\log\frac{p_g(x)}{p_r(x)} - \log\frac{2p_g(x)}{p_r(x)+p_g(x)}\right] - \Big(\int_x p_r(x)\log\frac{2p_r(x)}{p_r(x)+p_g(x)}\Big)\\
&= \int_x p_g(x)\log\frac{p_r(x)+p_g(x)}{2p_r(x)} - \int_x p_r(x)\log\frac{2p_r(x)}{p_r(x)+p_g(x)}\\
&= \int_x p_g(x)\log\frac{p_r(x)+p_g(x)}{2p_r(x)} + \int_x p_r(x)\log\frac{p_r(x)+p_g(x)}{2p_r(x)}\\
&= \int_x (p_g(x)+p_r(x))\log\frac{p_r(x)+p_g(x)}{2p_r(x)}\\
&= 2\mathcal{D}_{KL}\Big(\frac{1}{2}p_r + \frac{1}{2}p_g\|p_r\Big)
\end{aligned}$$

Training details

- Standard GAN training paradigm
- They use a WGAN-GP based discriminator
- Their generated images are 64x64
- 200-dimensional latent vector

- For reconstruction, they train a separate encoder
  - Due to their top-k, they likely cannot perform gradient descent on latent space


(AND node)
(OR node)
(Layer3:8x8)
(2,6)
(5,4)
(others)
(6,2)
(5,7)
(Layer2:16x16)
(3,7)
(3,8)
(4,8)
(3,9)
(8,6)
(8,7)
(8,8)
(8,9)
(11,3)
(11,4)
(12,3)
(12,4)
(8,12)
(8,13)
(9,12)
(9,13)
(Layer1:32x32)
(5,14)
(5,15)
(5,18)
(6,19)
(15,11)
(15,12)
(16,11)
(16,12)
(21,7)
(21,8)
(22,7)
(22,8)
(16,21)
(16,22)
(17,21)
(17,22)


(AND node)
(OR node)
(Layer3:8x8)
(2,3)
(2,6)
(others)
(6,4)
(6,6)
(Layer2:16x16)
(3,5)
(3,7)
(6,5)
(6,7)
(4,11)
(4,12)
(5,11)
(5,12)
(11,6)
(11,7)
(12,8)
(12,9)
(11,10)
(11,12)
(12,10)
(12,13)
(Layer1:32x32)
(6,9)
(7,13)
(11,9)
(11,13)
(7,21)
(7,22)
(9,22)
(9,23)
(21,11)
(21,12)
(23,15)
(23,16)
(21,21)
(21,22)
(22,19)
(22,22)

Possible Future Work

- Spatially hierarchical models
  - Current model has fixed hierarchy (grid based composition)
  - More like MONet/IODINE/PSGNet
  - Very useful for robotics
  - Maybe combine self attention w/ sparsemax w/ spatial transformer?
- Multi-modal models
  - Class conditional imagenet is well-defined task
  - Most people can do 128x128, but since Google's BigGAN paper no longer an area of study
  - Must demonstrate unsupervised separation of classes (maybe via infogan conditioning)
- Efficient models
  - Current implementation is leads to no memory improvements
  - Can we implement ***Learning Dynamic Routing for Semantic Segmentation*** but for generation?
  - Dynamic selection of basis depending on image complexity