Training and Understanding Deep Neural Networks for Robotics, Design, and Perception.

Cornell University, A-Exam

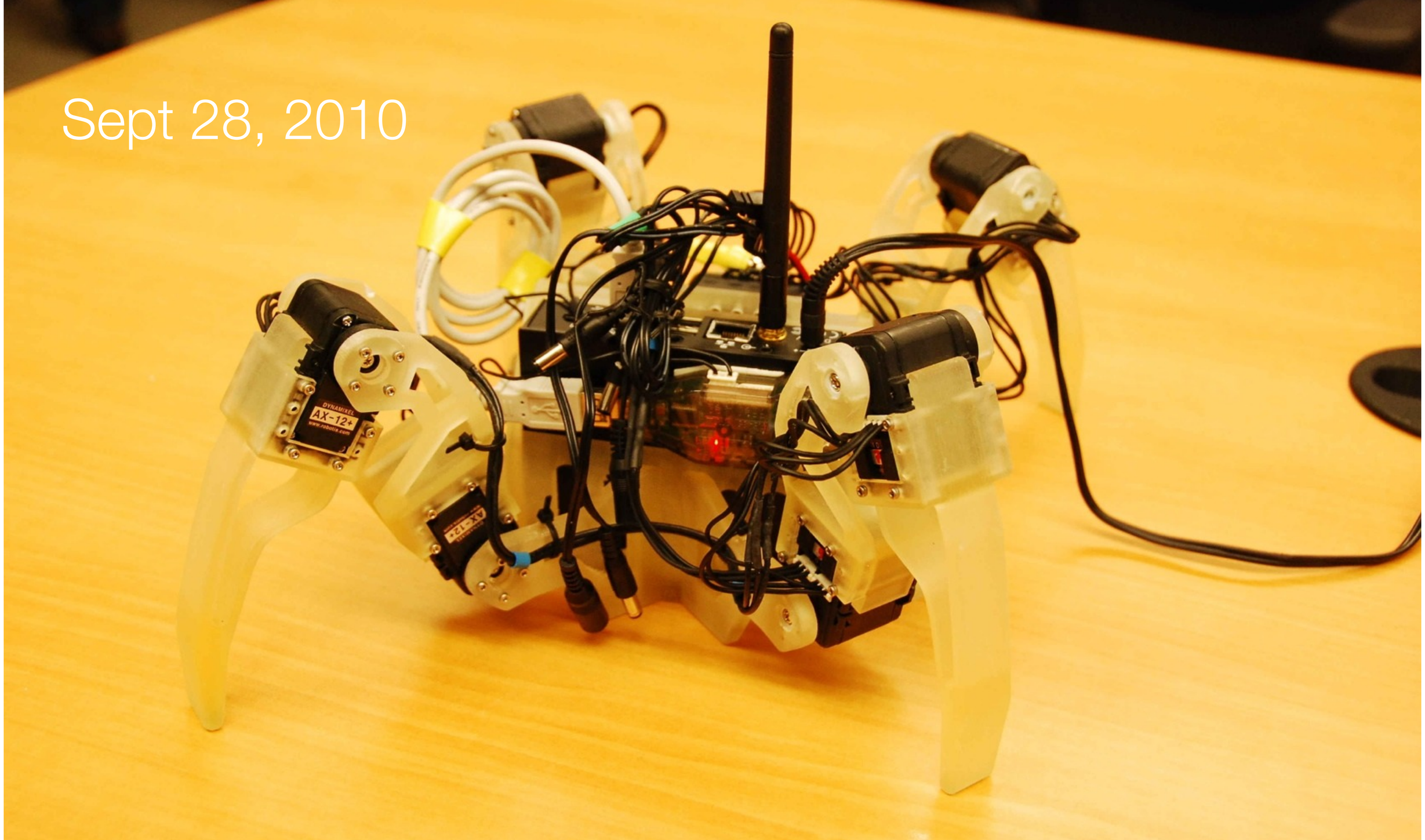
19 February 2015

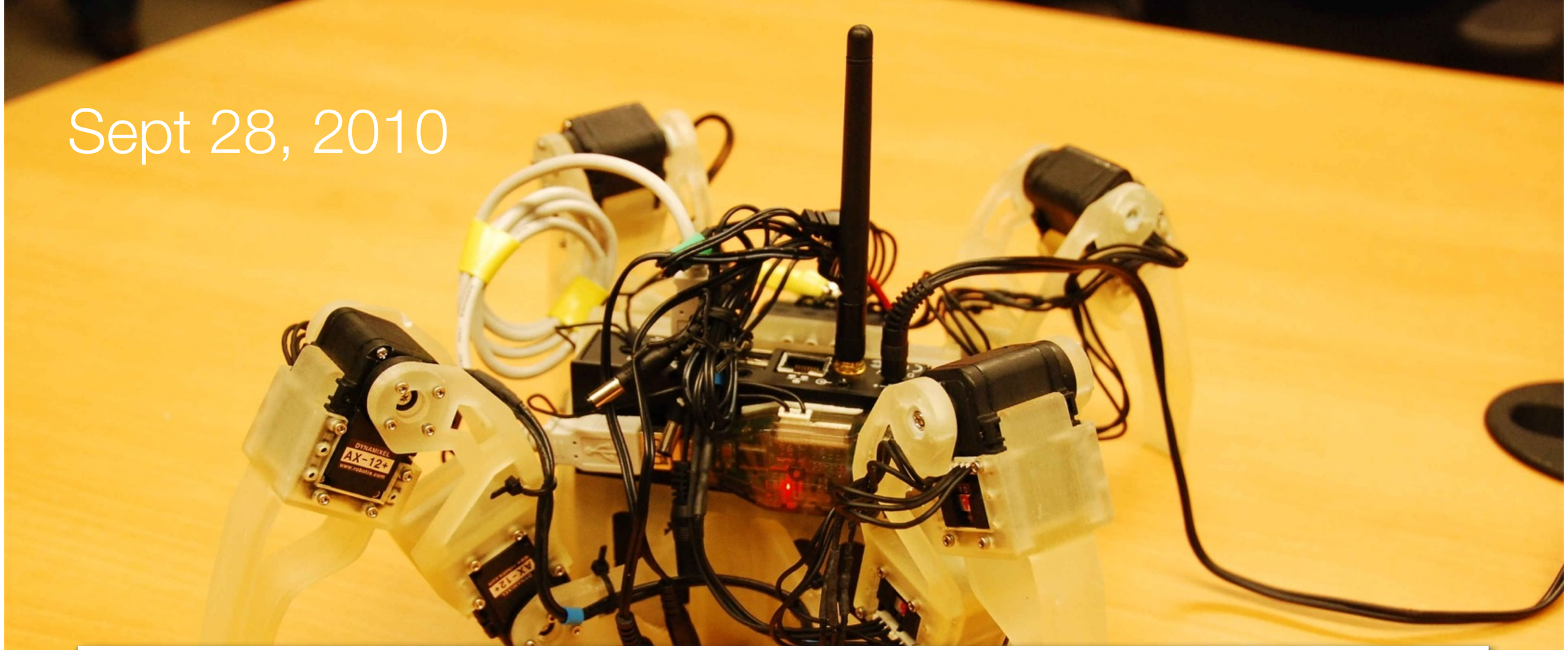
Jason Yosinski

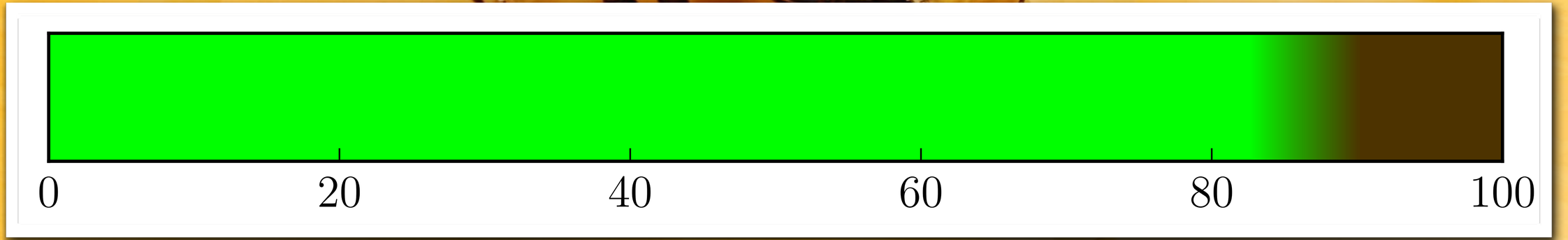
Creative Machines Lab

Cornell University

Original slides available under Creative Commons Attribution-ShareAlike 3.0

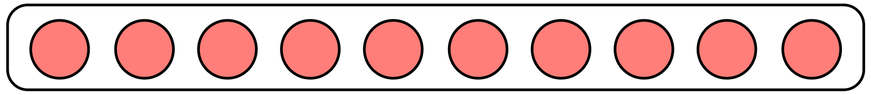

Sept 28, 2010
DYNAMIXEL
AX-12+
www.robotis.com


Sept 28, 2010
DYNAMIXEL
AX-12+

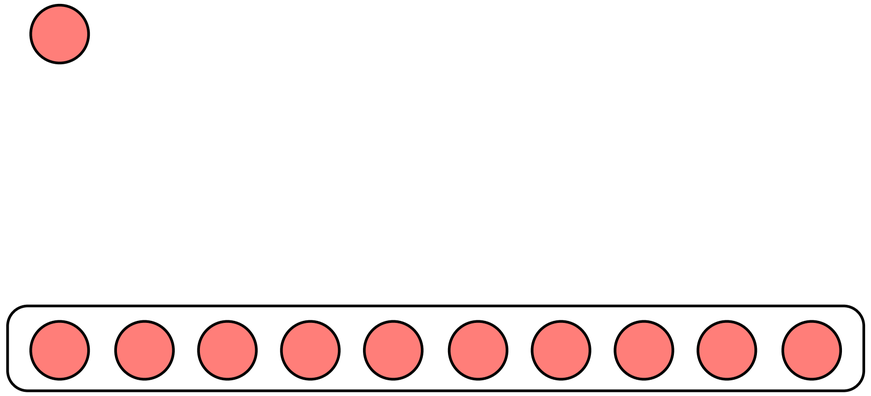

0
20
40
60
80
100

$$
\begin{bmatrix}
1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\
0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\
-0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\
-1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\
1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\
1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\
 & & . \quad . \quad . & & &
\end{bmatrix}
$$

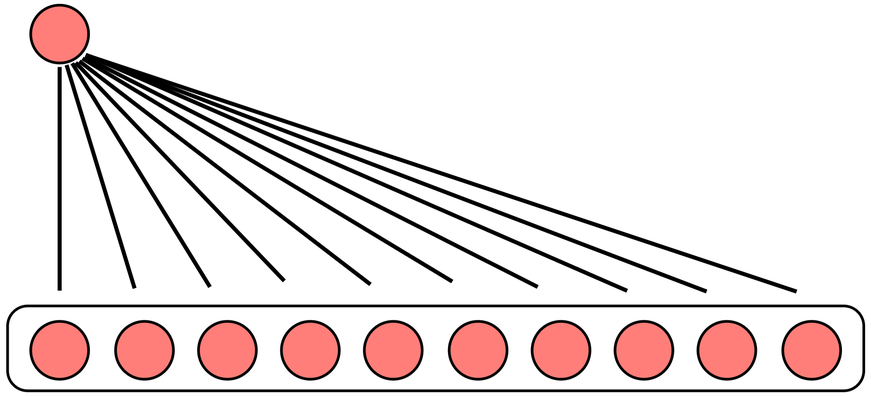
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . & . & . & \end{bmatrix}$$



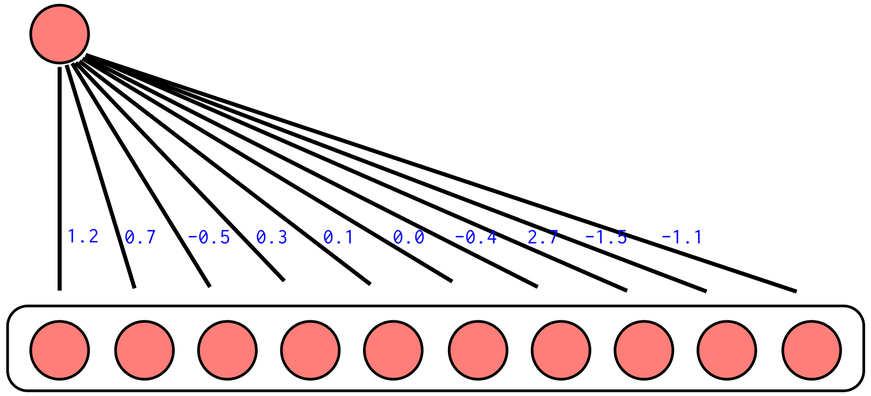
$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . & . & . & \end{bmatrix}$$



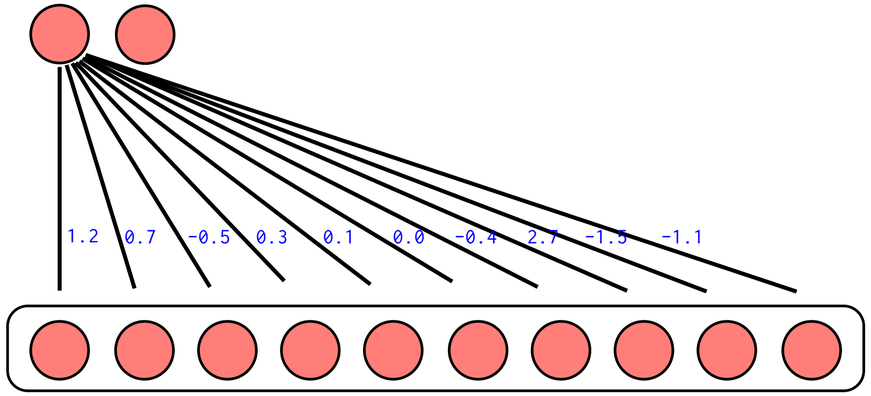
$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



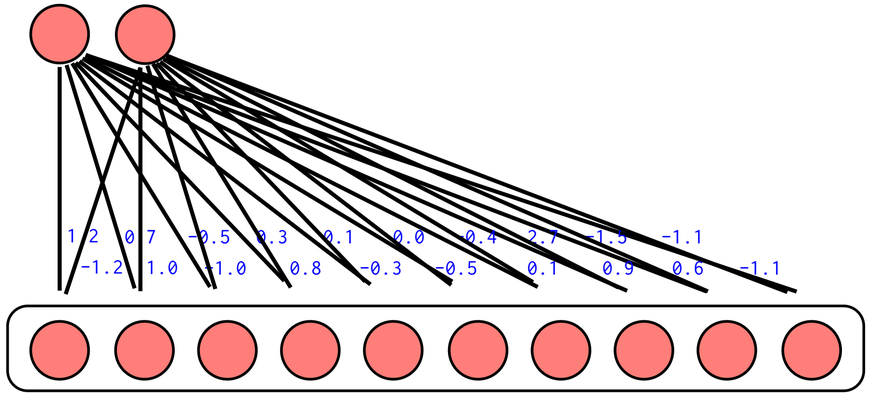
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



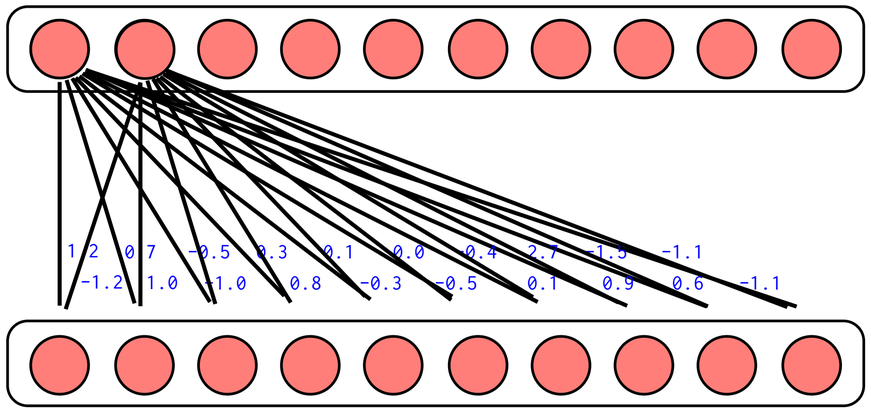
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



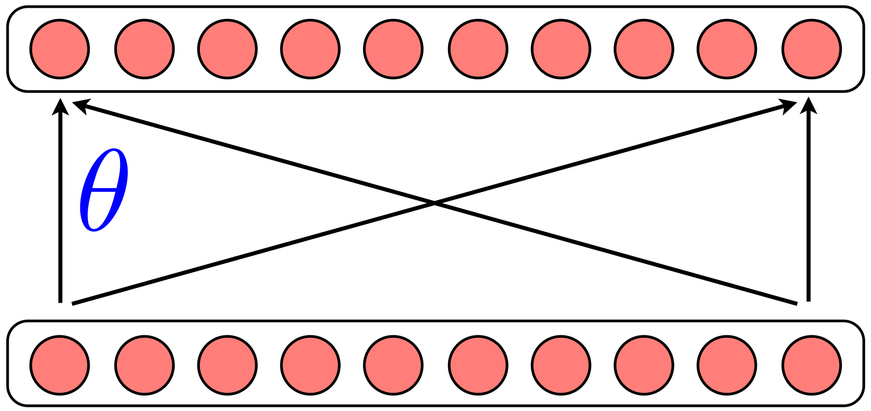
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



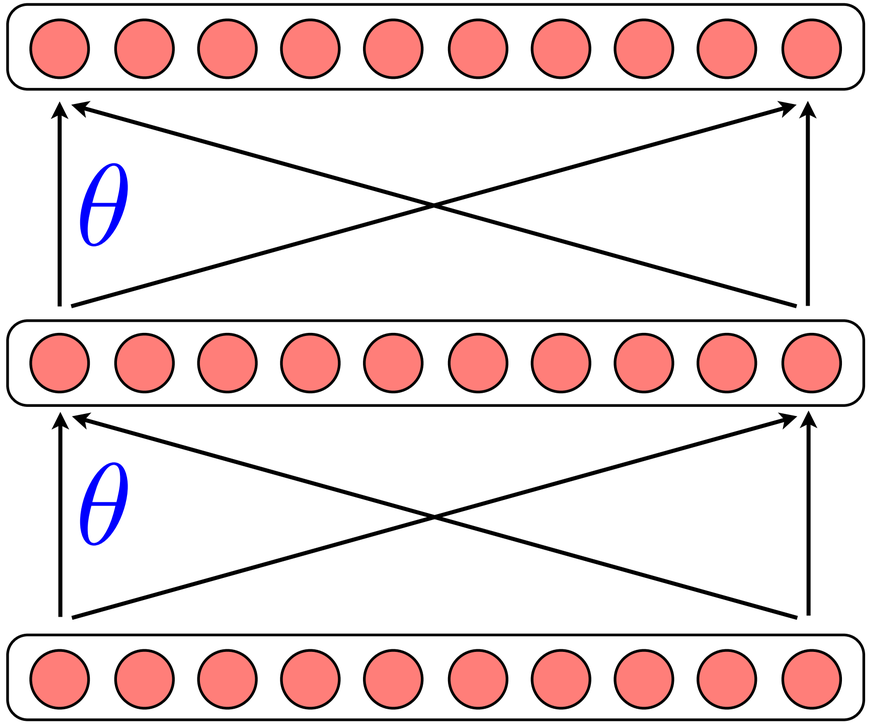
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . & . & . & \end{bmatrix}$$



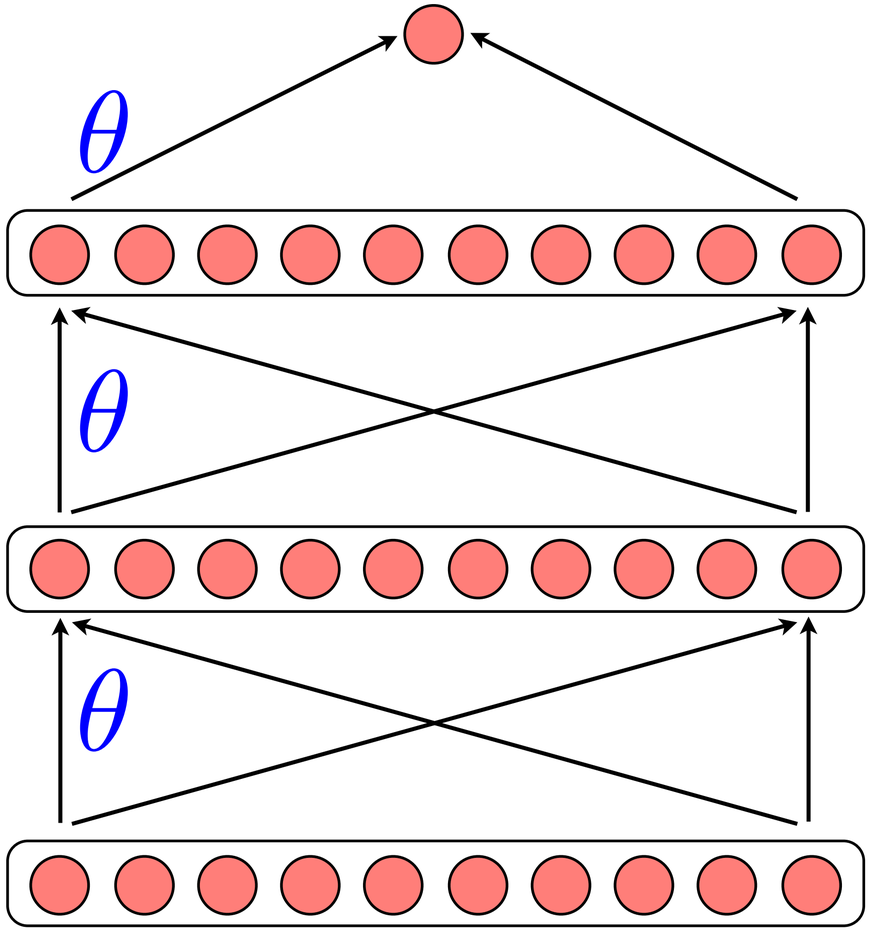
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . & . & . & \end{bmatrix}$$



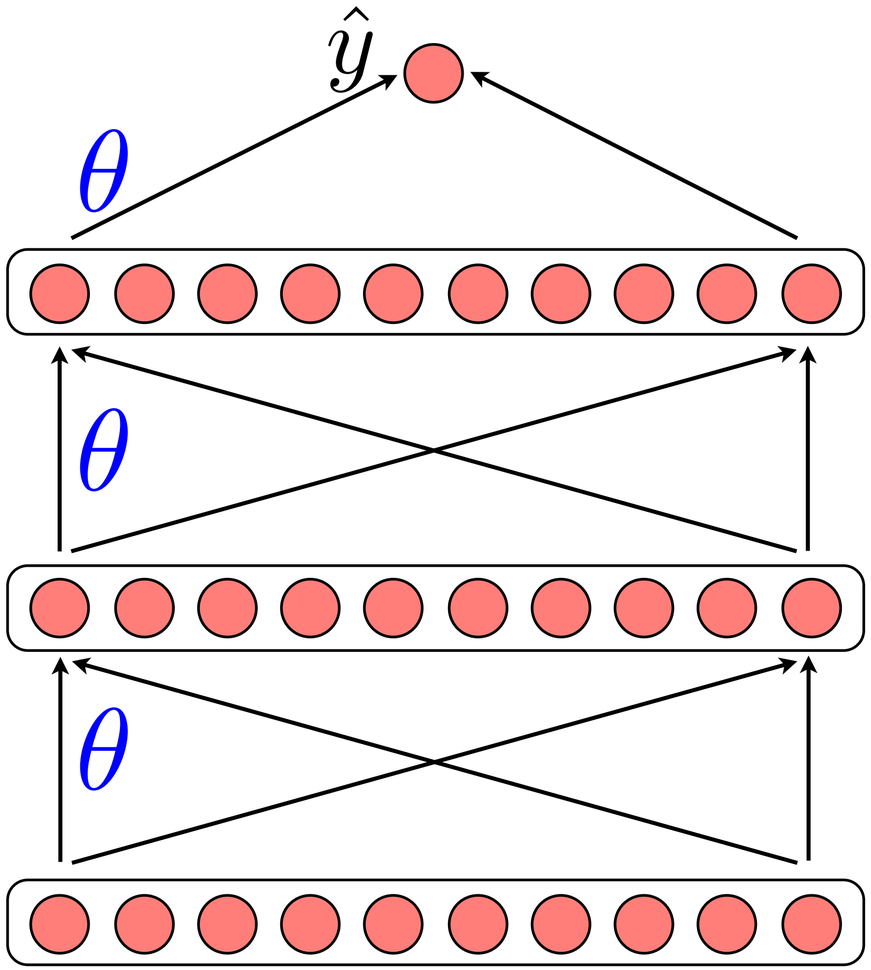
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . & . & . & \end{bmatrix}$$



$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

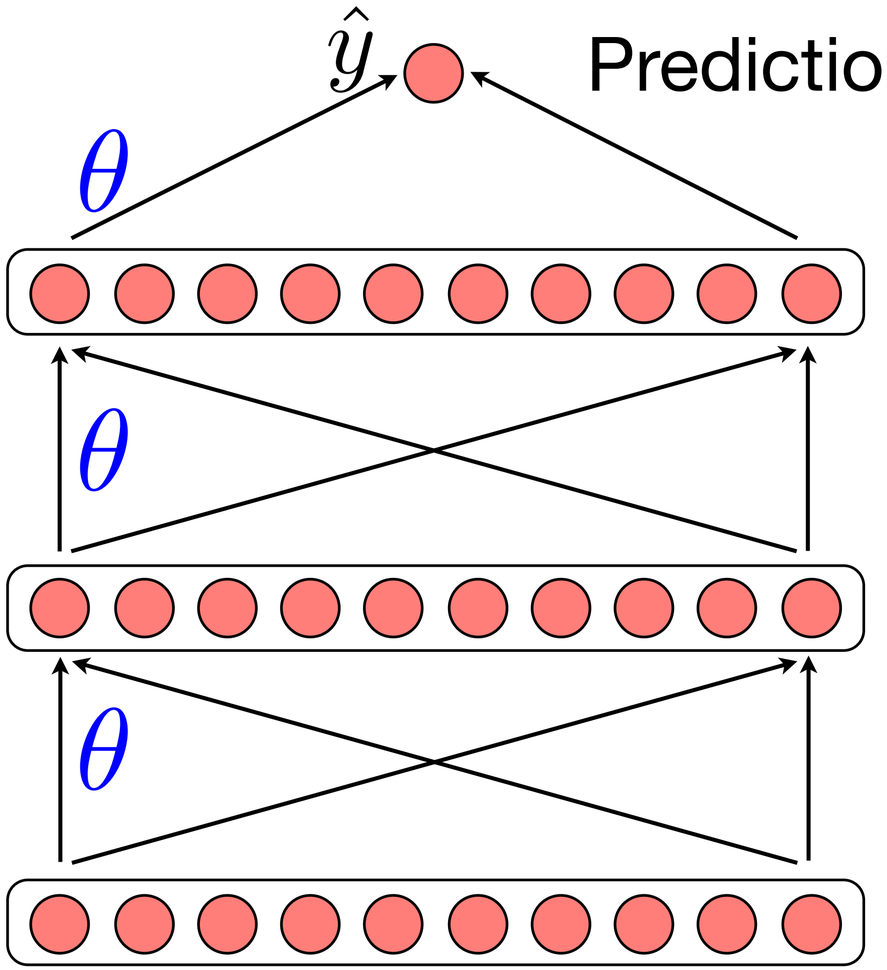


$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

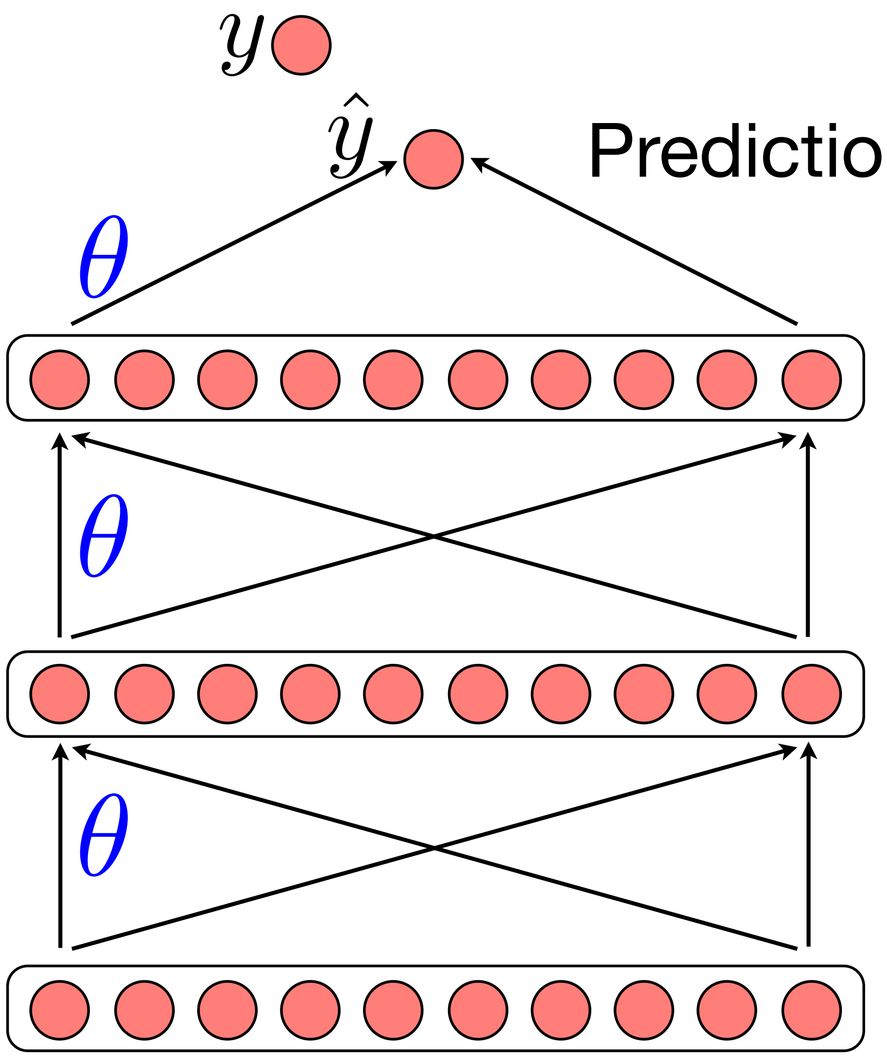


$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

Prediction: dog (`-1.0`) or cat (`1.0`)

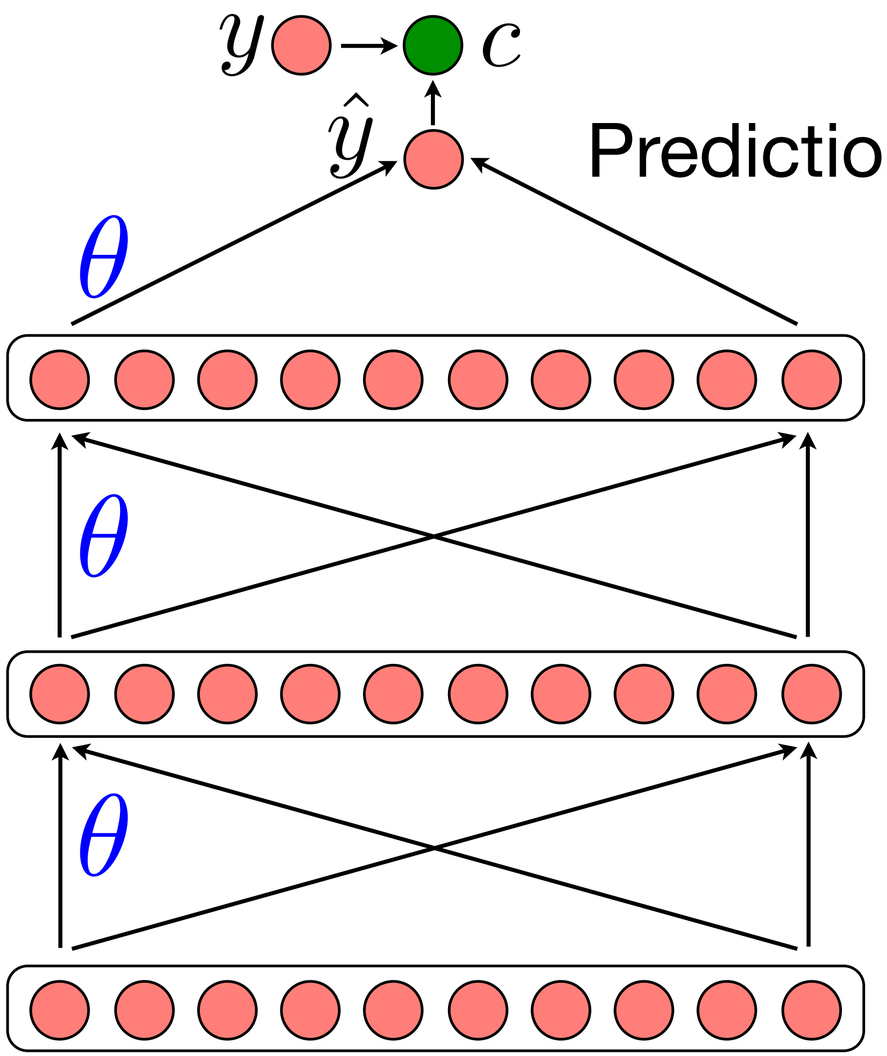


$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 \\ & & \cdot\ \cdot\ \cdot & & \end{bmatrix}$$



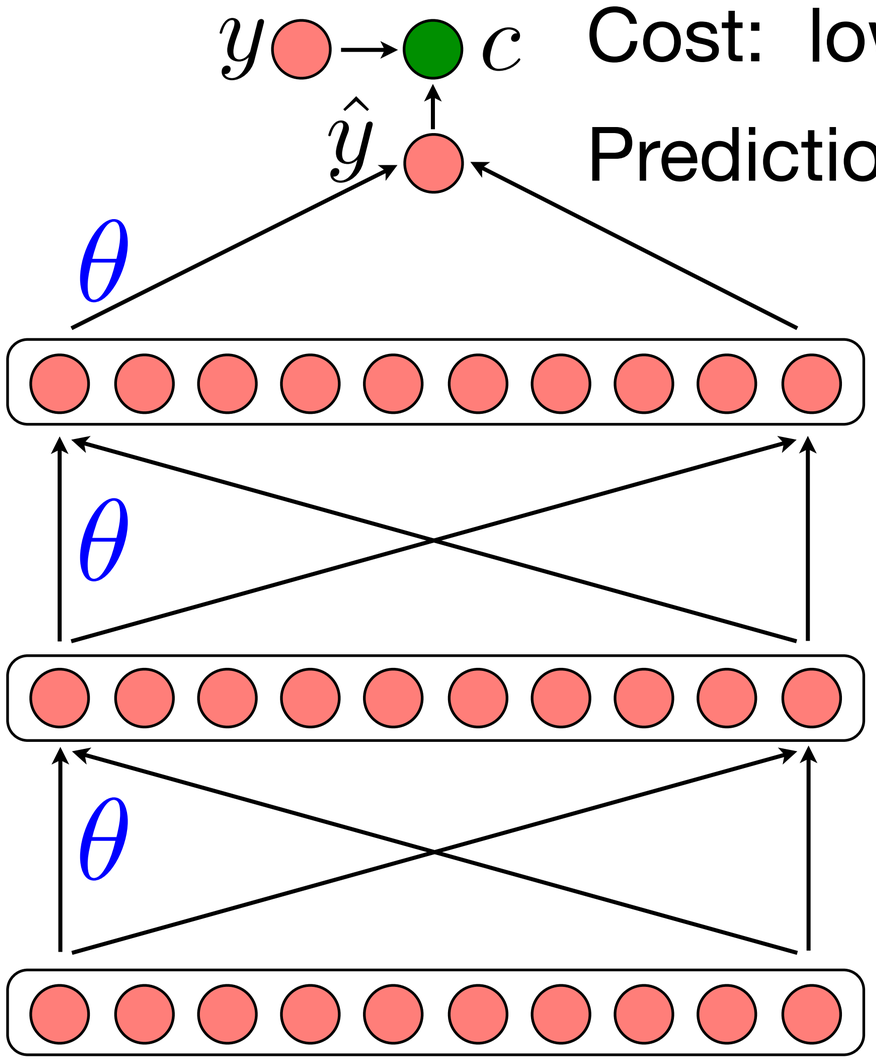
Prediction: dog (`-1.0`) or cat (`1.0`)

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 \\ & & \cdot\ \cdot\ \cdot & & \end{bmatrix}$$



Prediction: dog (`-1.0`) or cat (`1.0`)

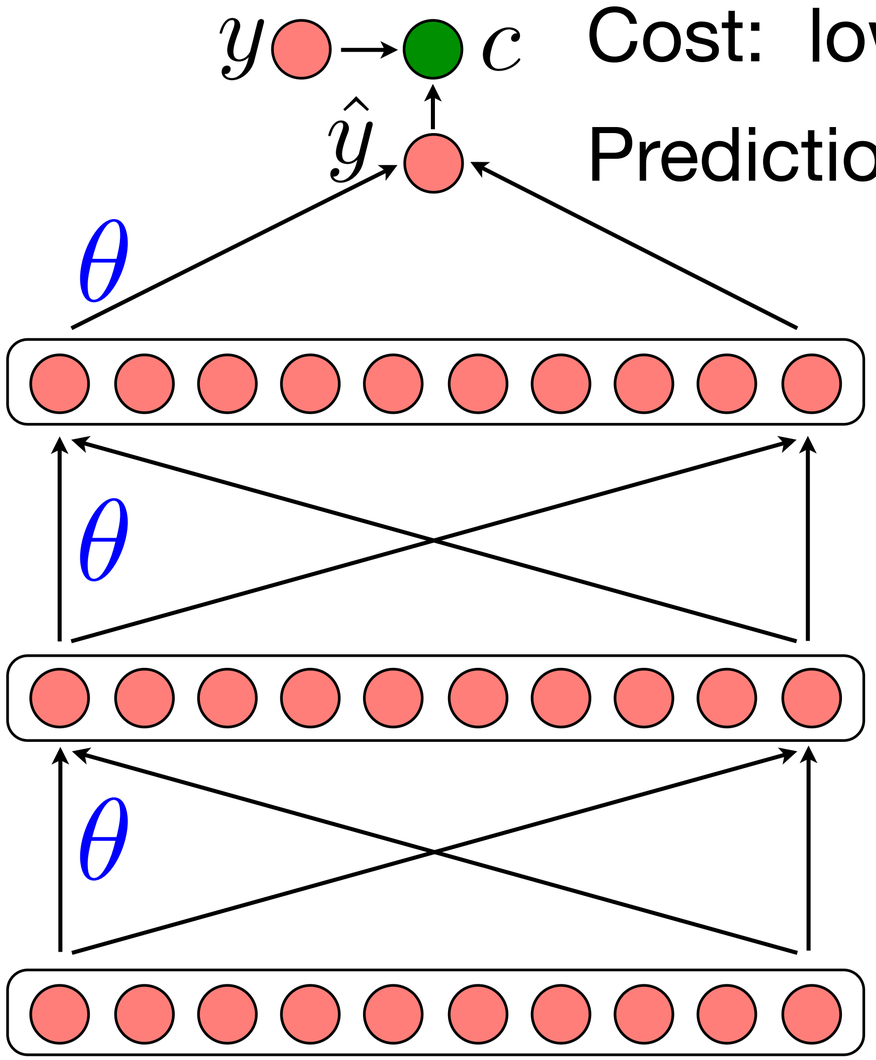
$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

$$X \quad \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

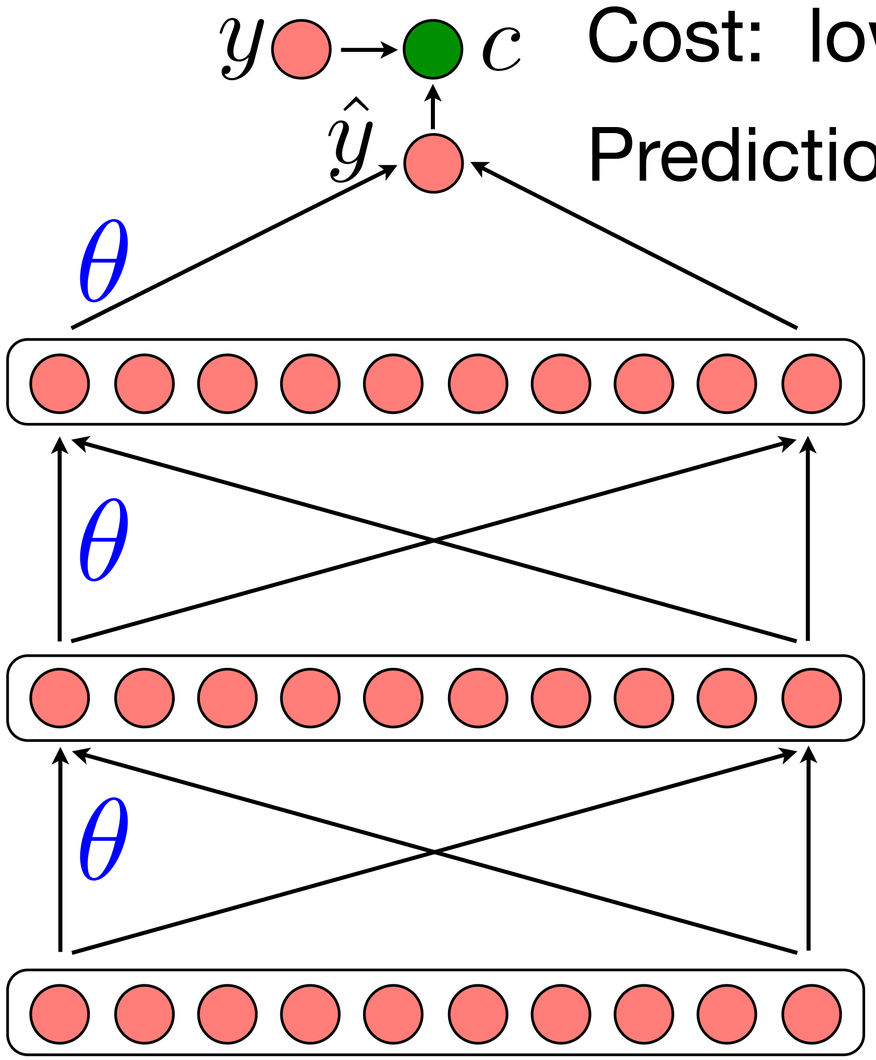


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

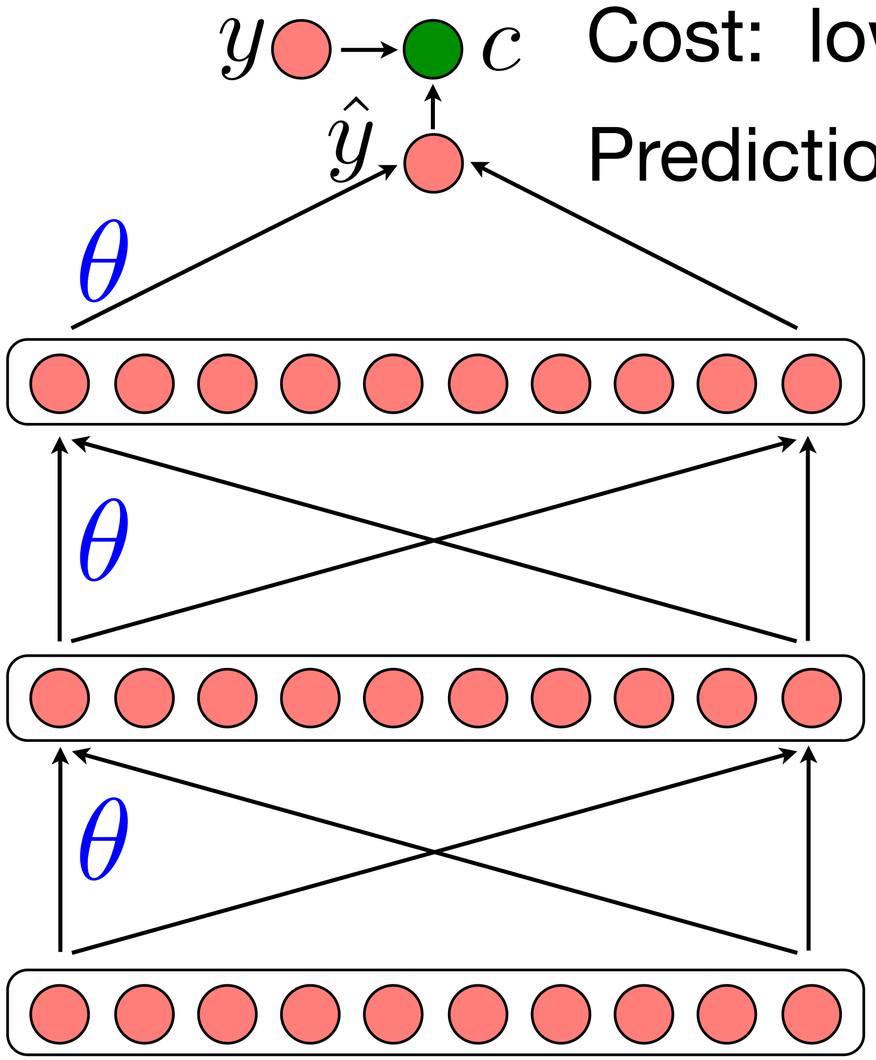


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$

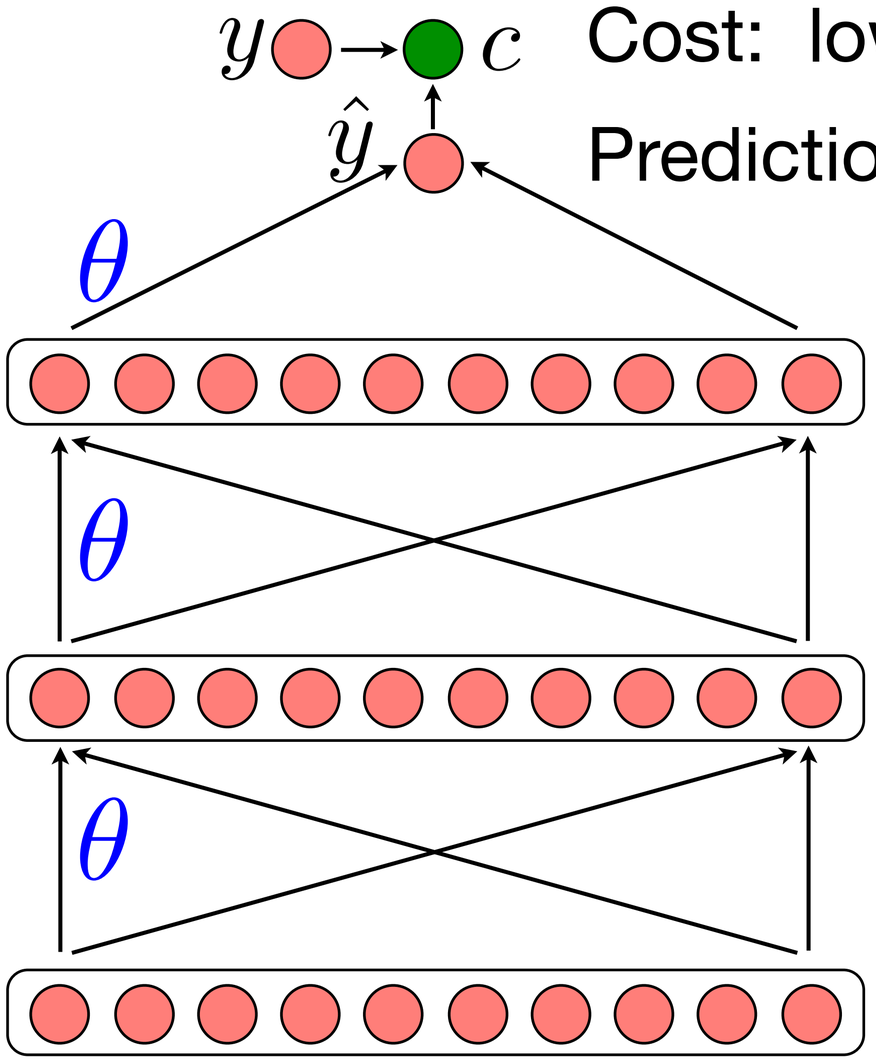


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$

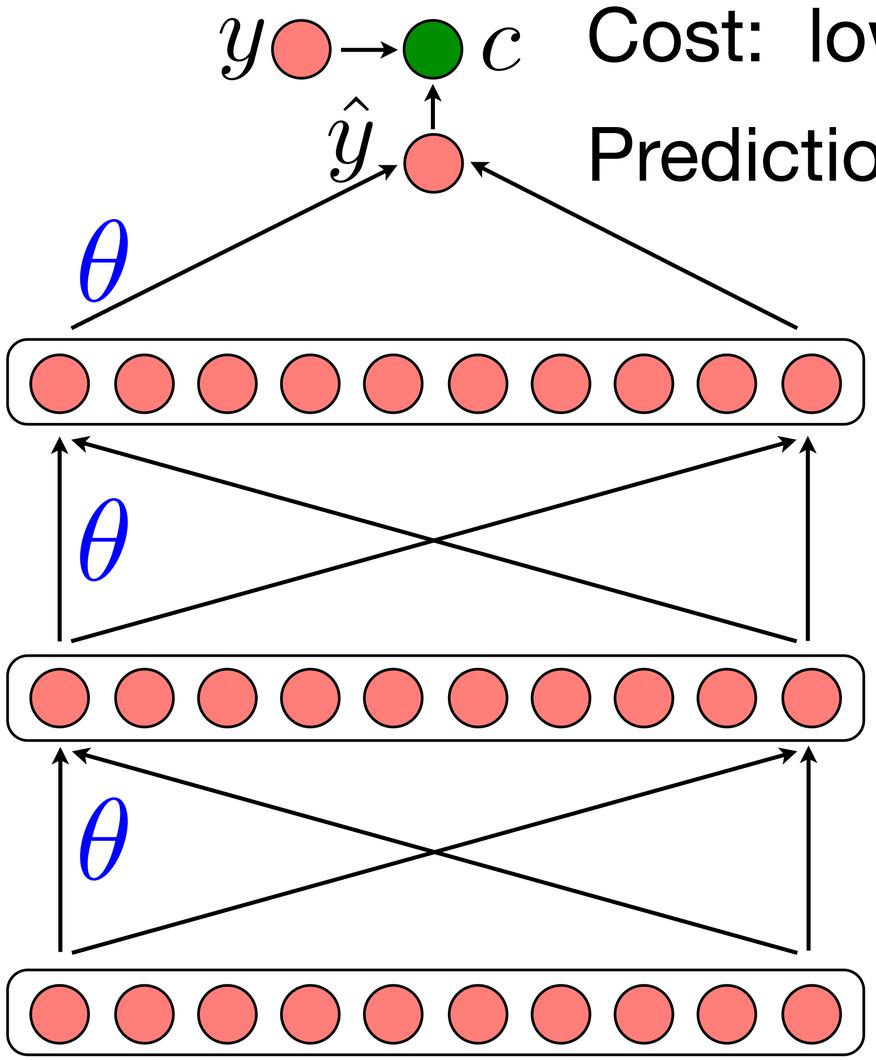


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...
  - ... fiddling with weights

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$

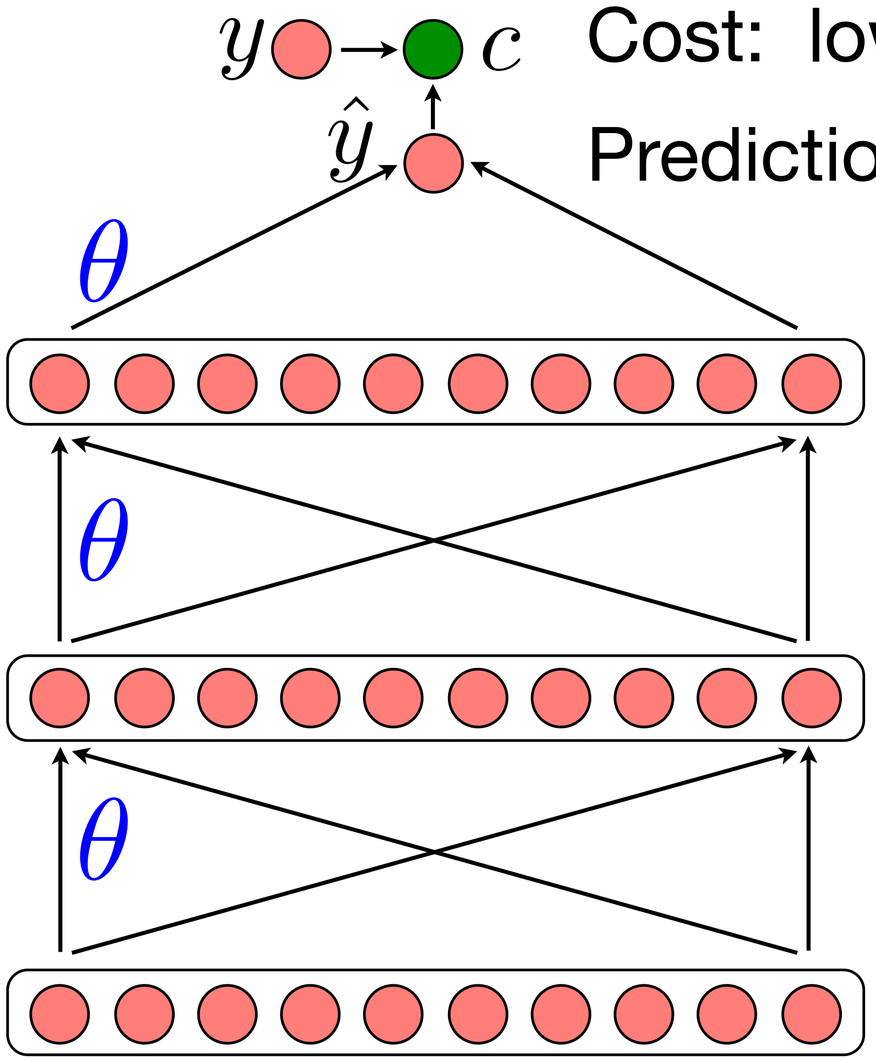


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...
  - ... fiddling with weights
  - ... descending gradient

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$

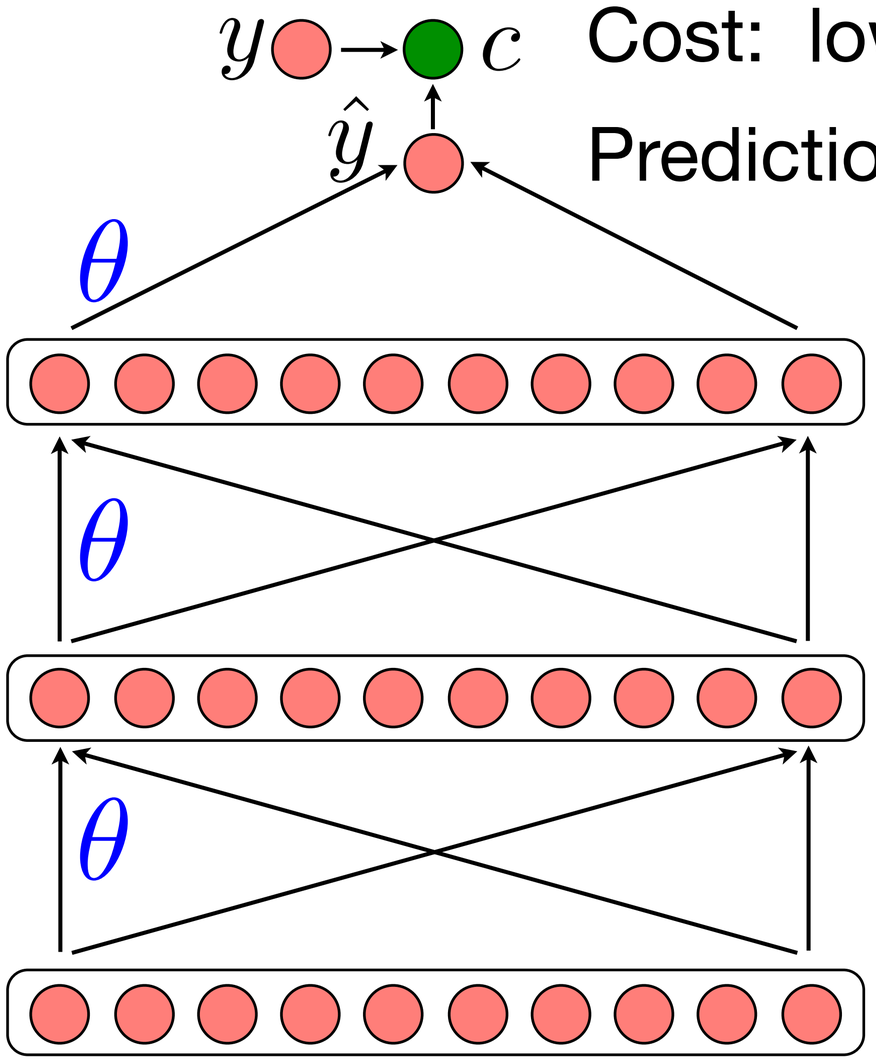


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...
  - ... fiddling with weights
  - ... descending gradient $\frac{\partial c}{\partial \theta}$

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 \\ & & \cdot\ \cdot\ \cdot & & \end{bmatrix}$$

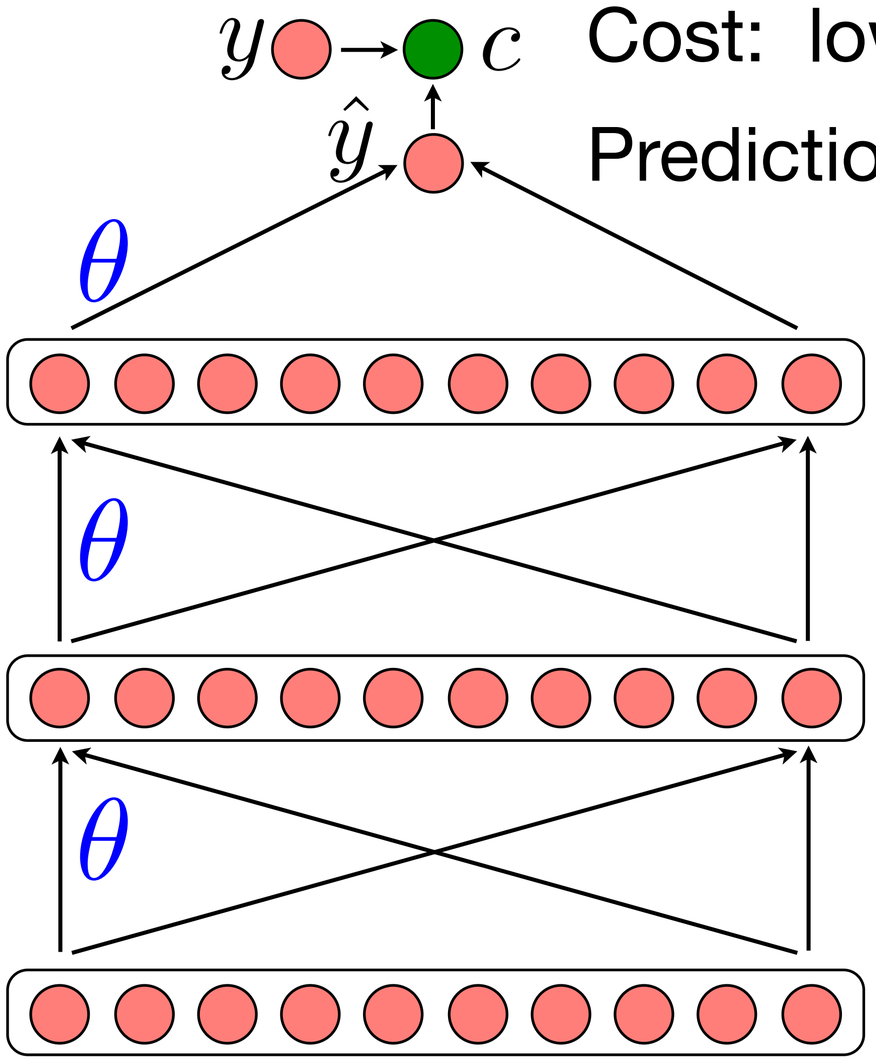


Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...
  - ... fiddling with weights
    (non-differentiable cost)
  - ... descending gradient $\frac{\partial c}{\partial \theta}$
    (differentiable cost)

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$



Cost: lower is good

Prediction: dog (`-1.0`) or cat (`1.0`)

- Initially: garbage output
- How can we train?
- Minimize cost by...
  - ... fiddling with weights
    (non-differentiable cost)
  - ... descending gradient $\frac{\partial c}{\partial \theta}$
    (differentiable cost)

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 \\ & & \cdot\ \cdot\ \cdot & & \end{bmatrix}$$

Compositional and Trainable

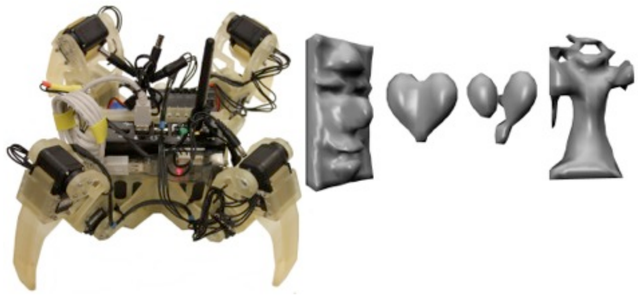
The path of research...

The path of research...

Gaits						

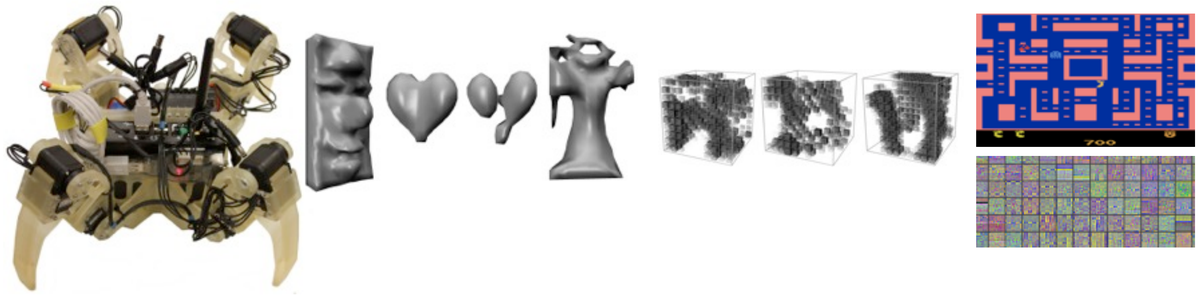
The path of research...

Gaits	3D Design					



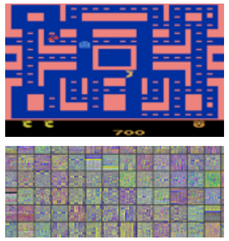
The path of research...

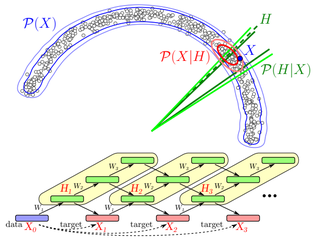
Gaits	3D Design	Unsupervised				



The path of research...

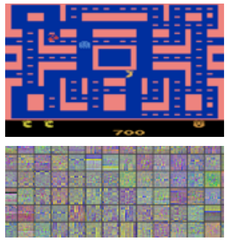
Gaits	3D Design	Unsupervised	GSN			

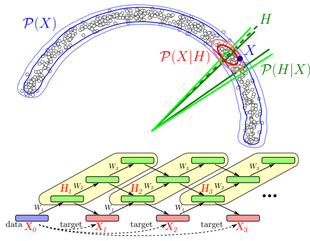


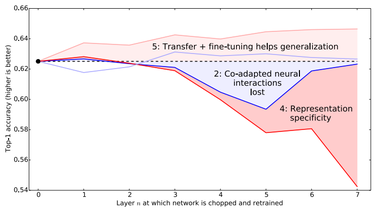


The path of research...

Gaits	3D Design	Unsupervised	GSN	Transfer		

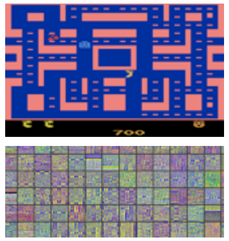


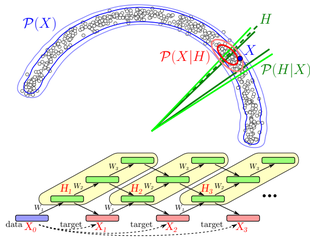


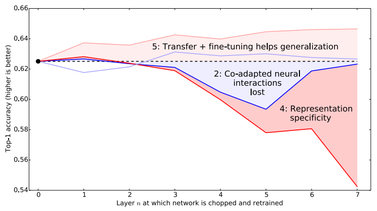


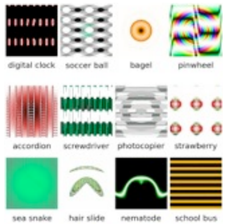
The path of research...

Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	



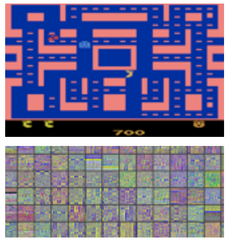


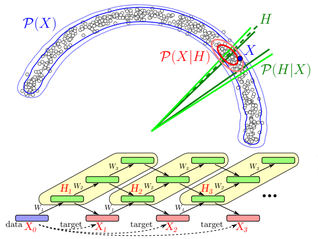


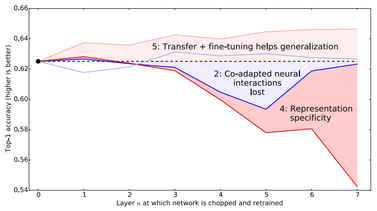


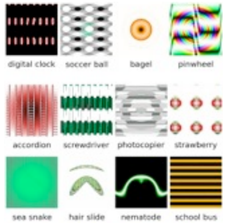
The path of research...

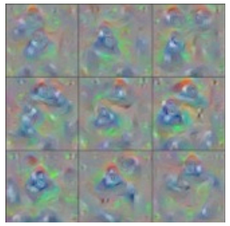
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize



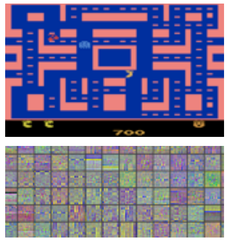


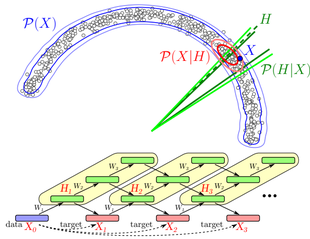


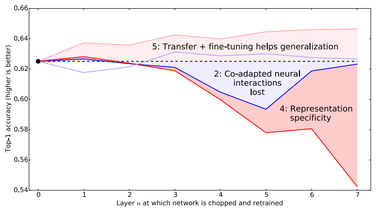


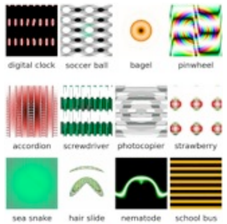


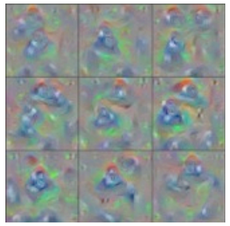
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize



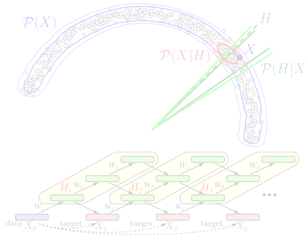




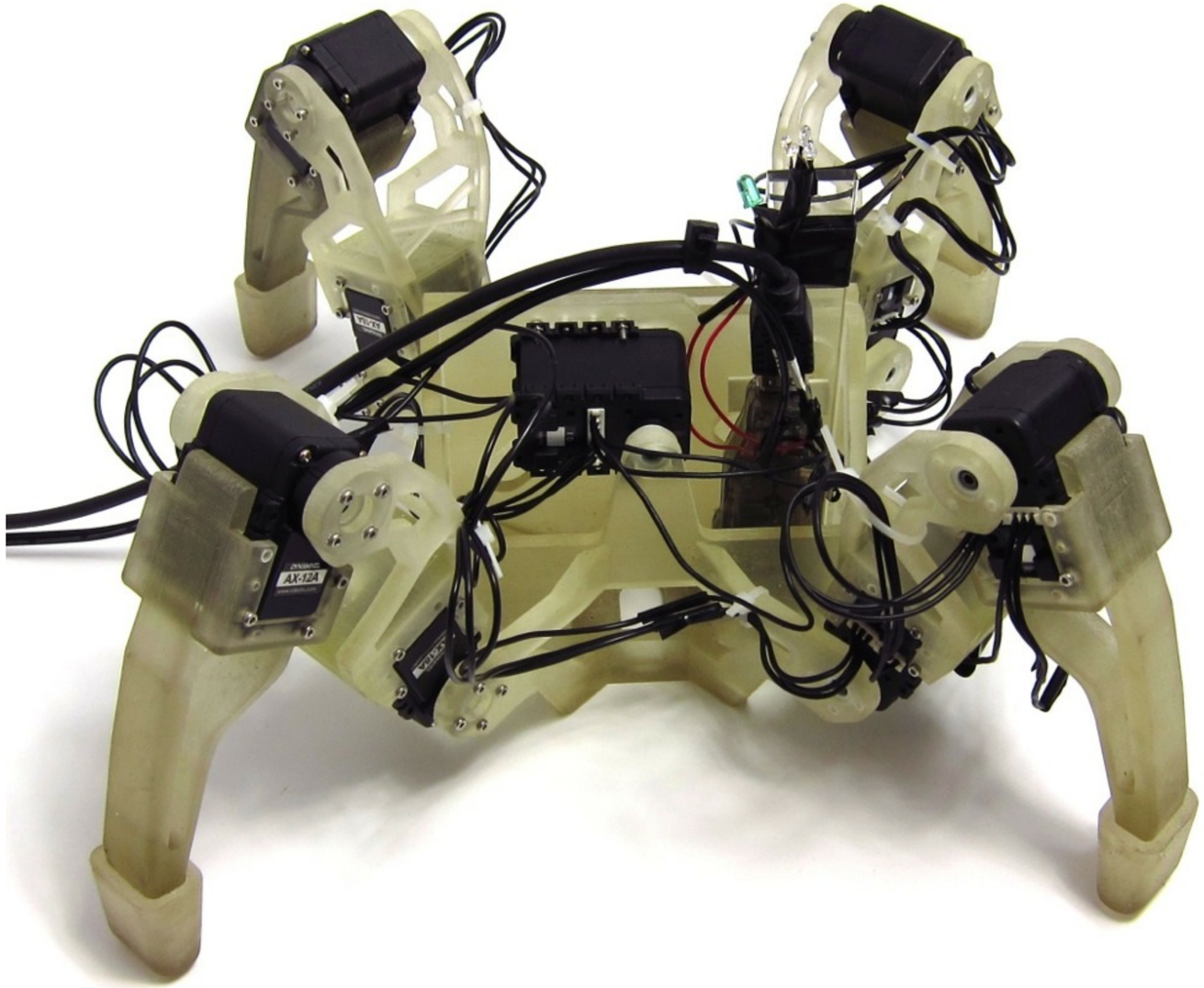




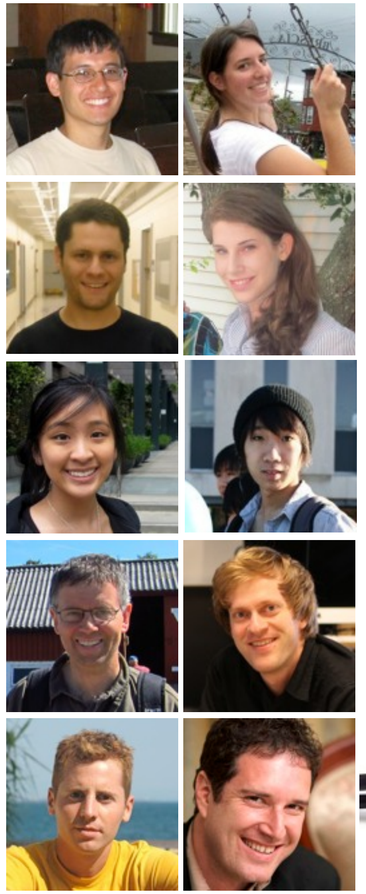
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

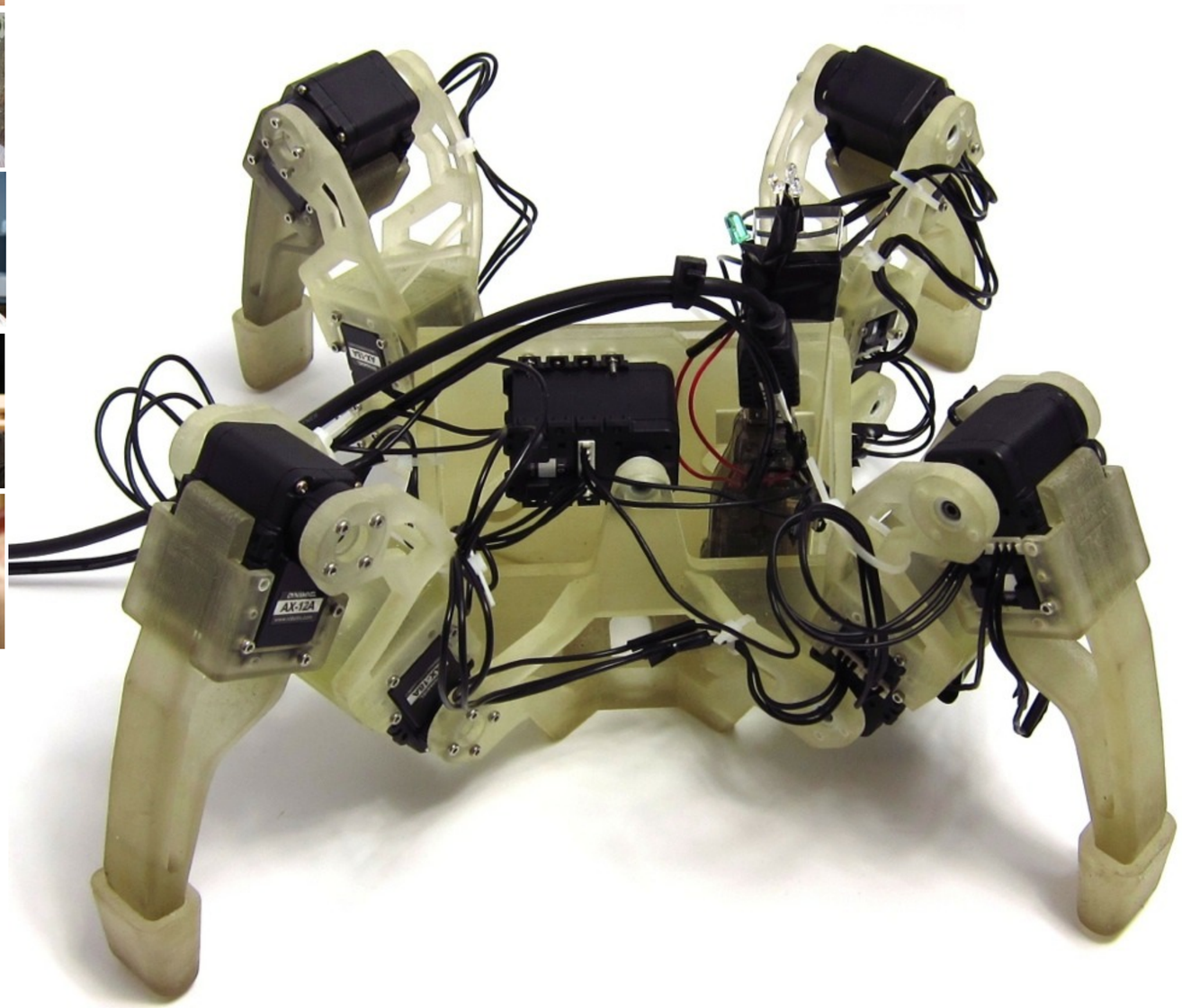


Robot Gait Discovery

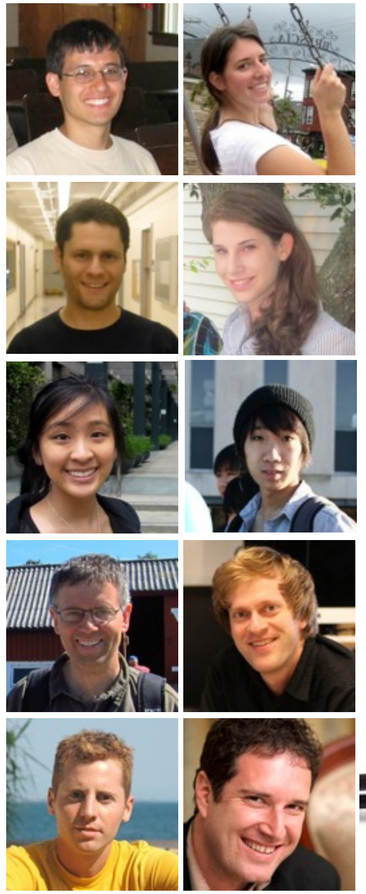


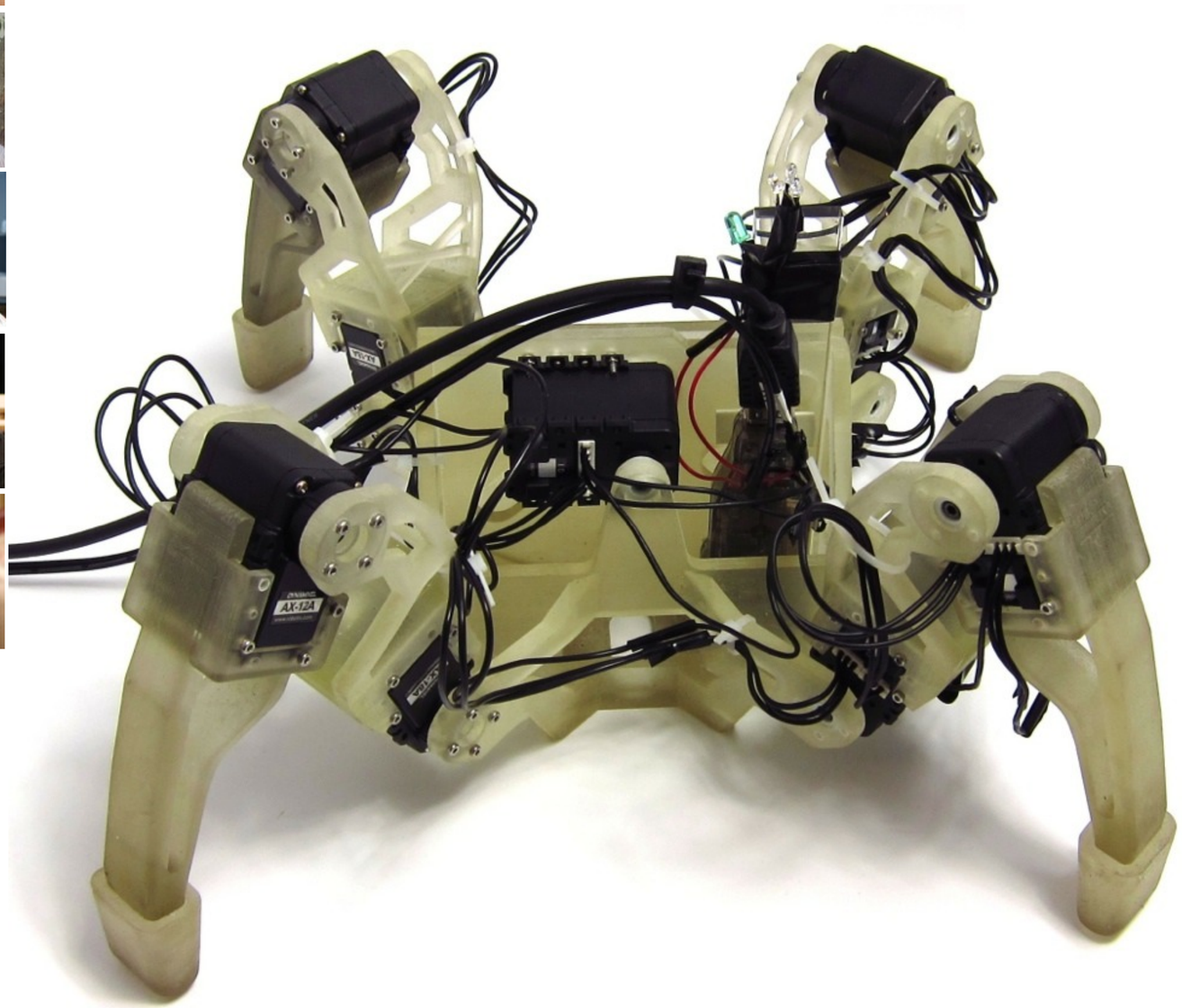
Robot Gait Discovery



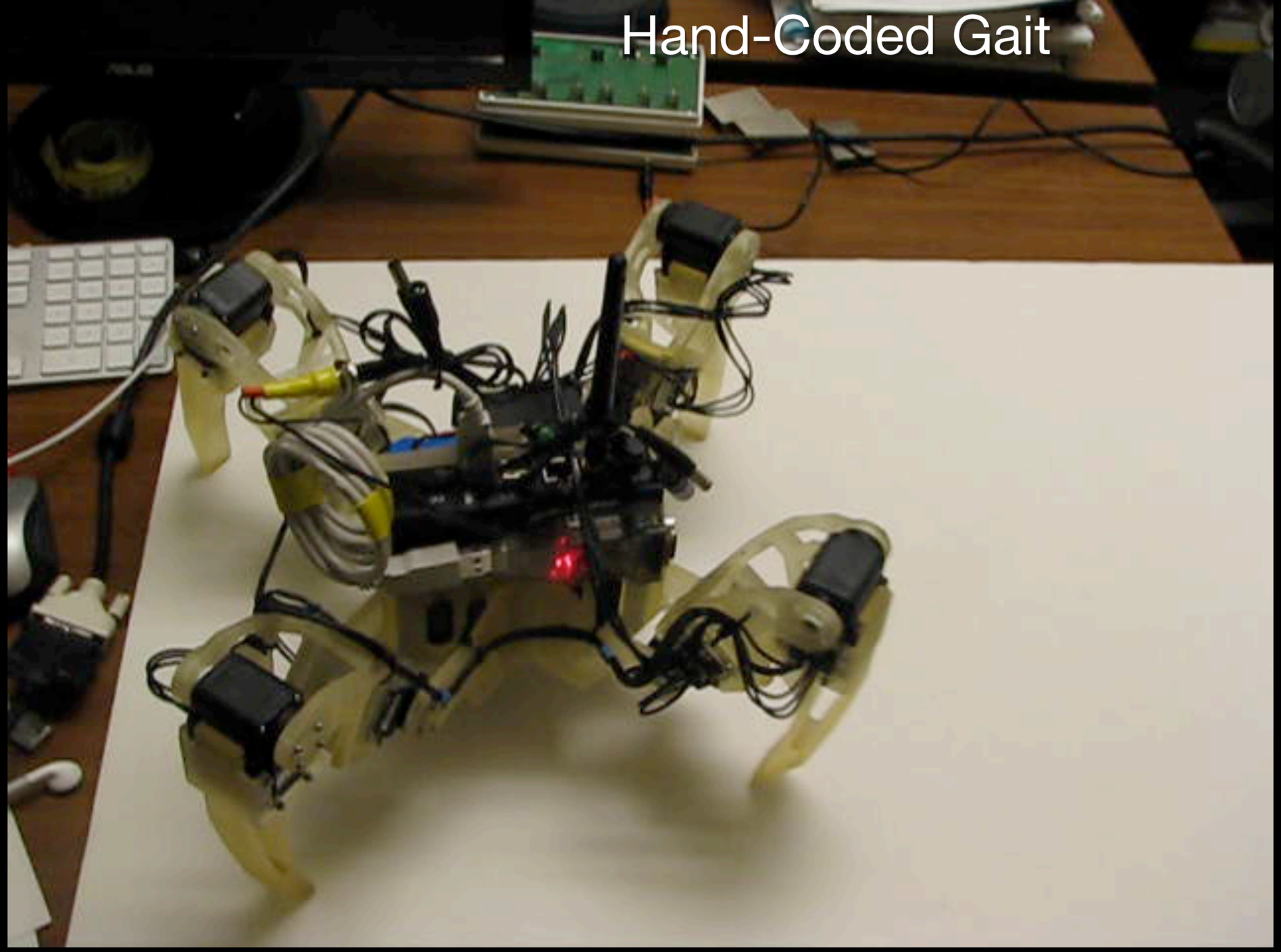


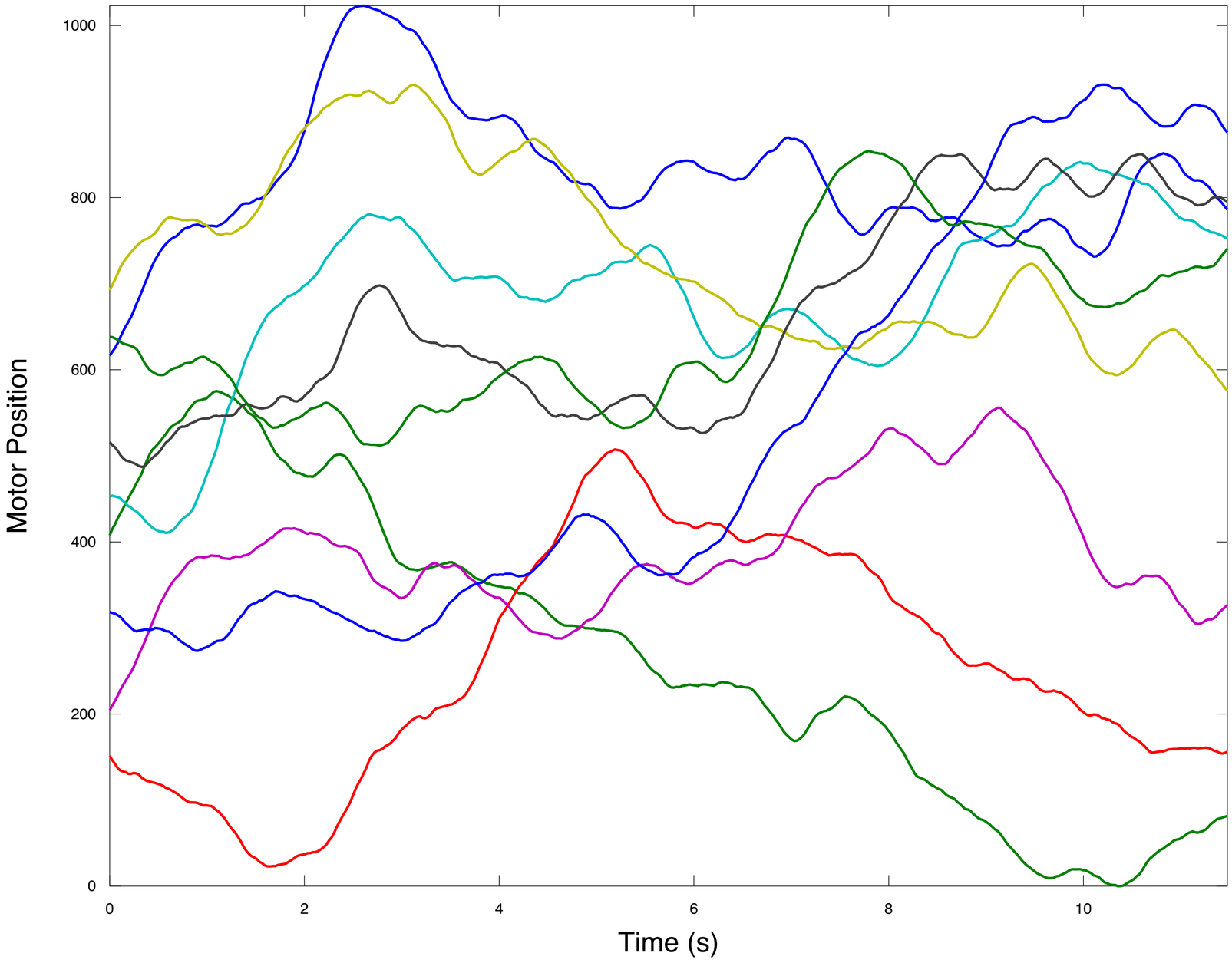
Robot Gait Discovery



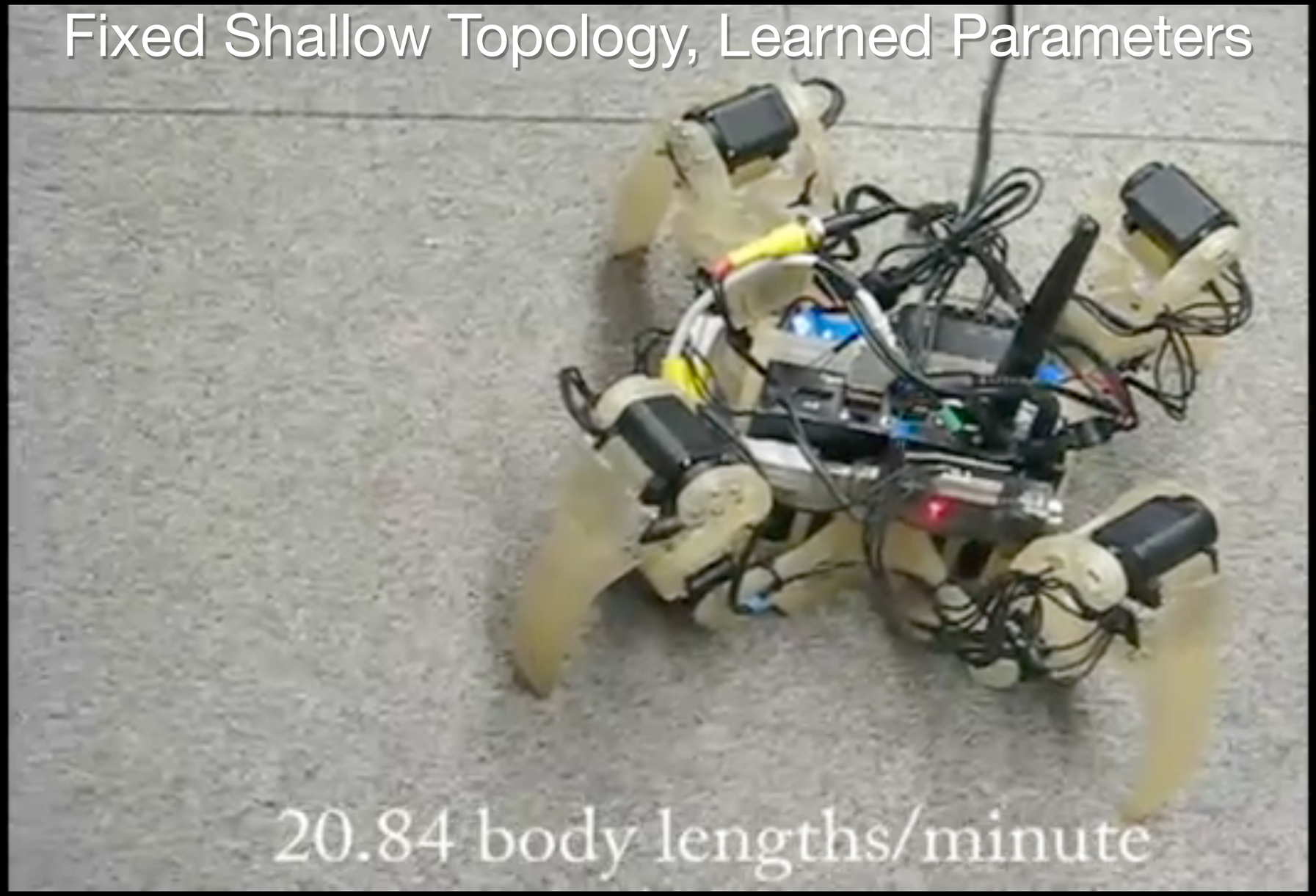


Hand-Coded Gait

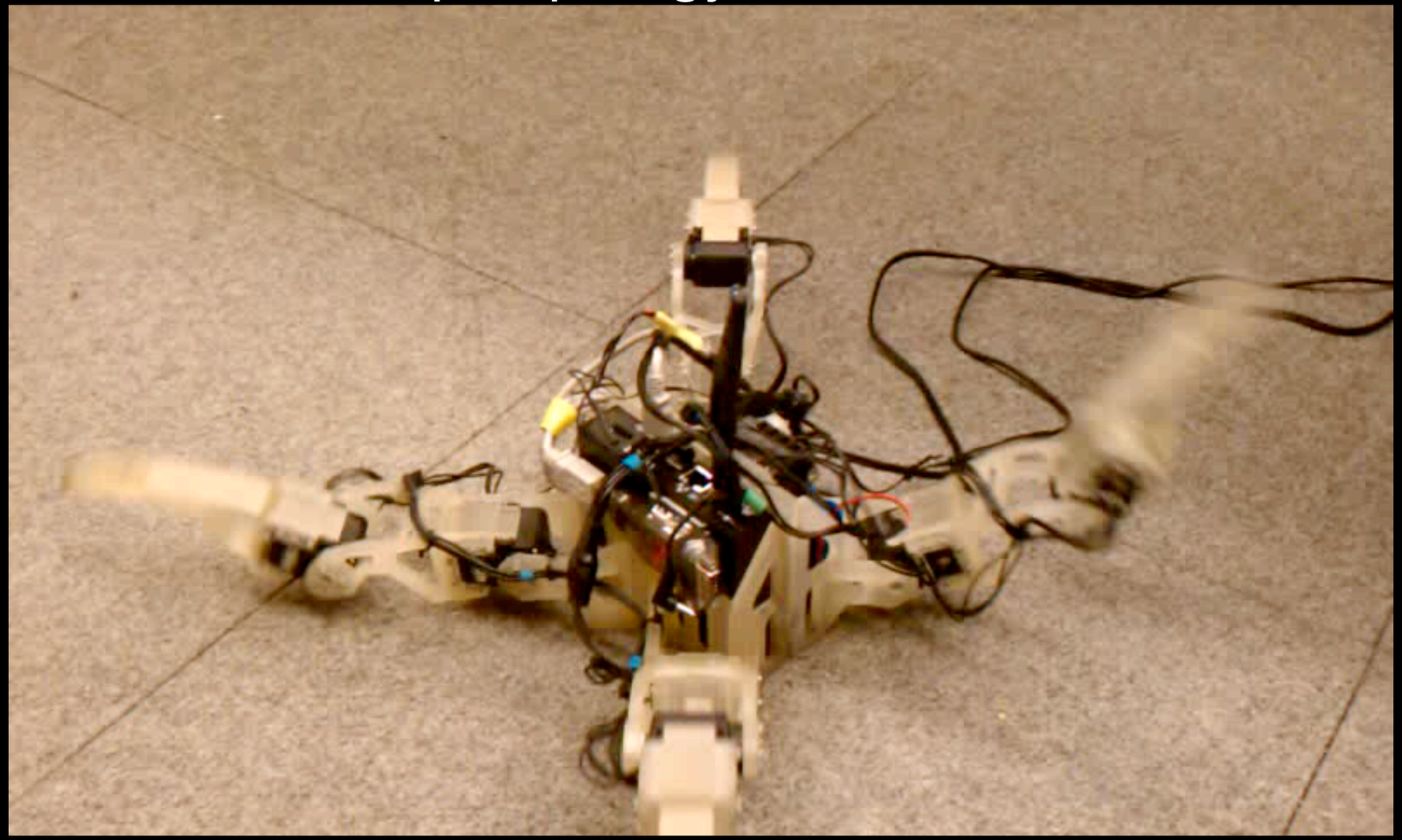



Motor Position
Time (s)
0
200
400
600
800
1000
0
2
4
6
8
10

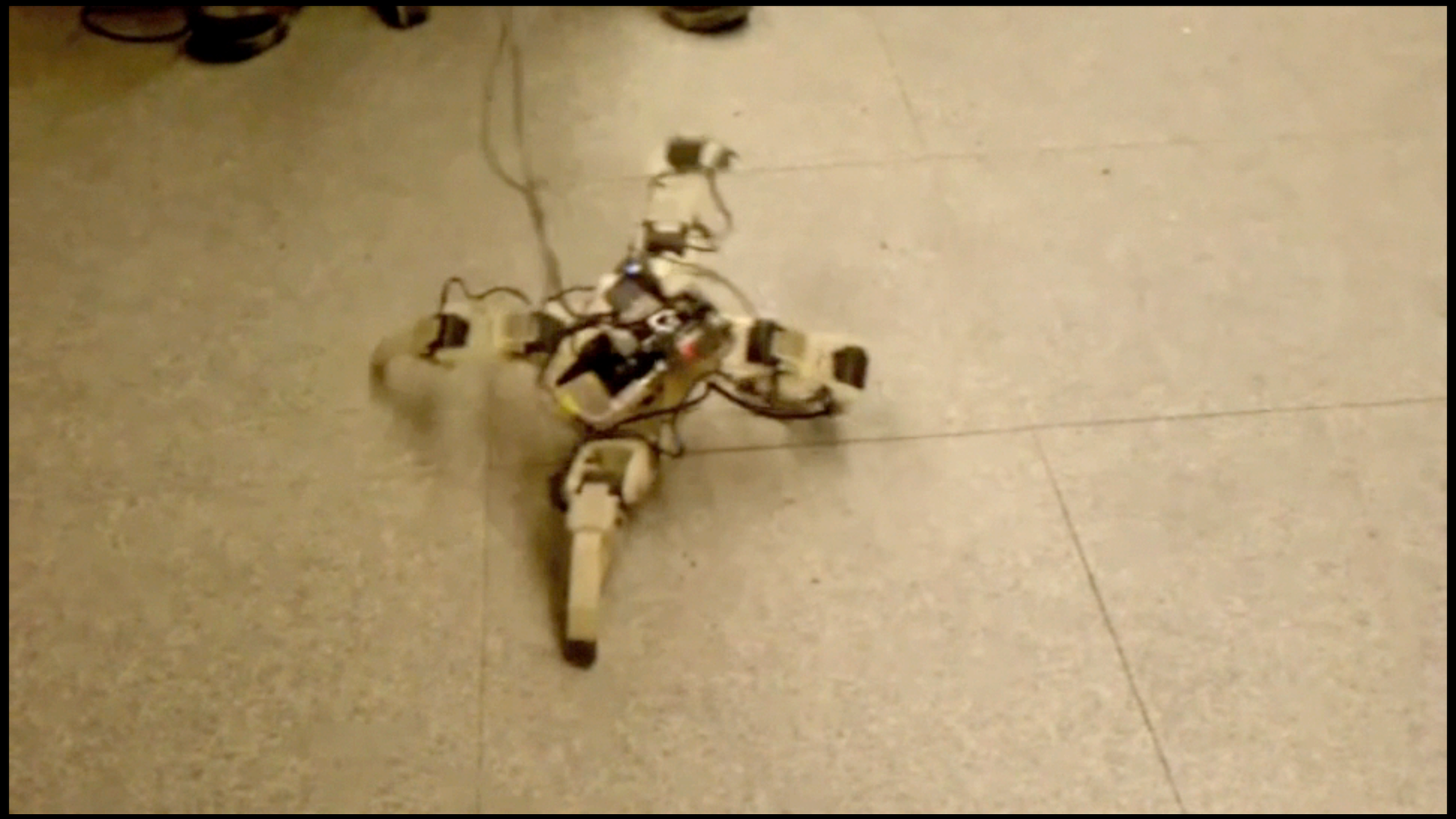
Fixed Shallow Topology, Learned Parameters



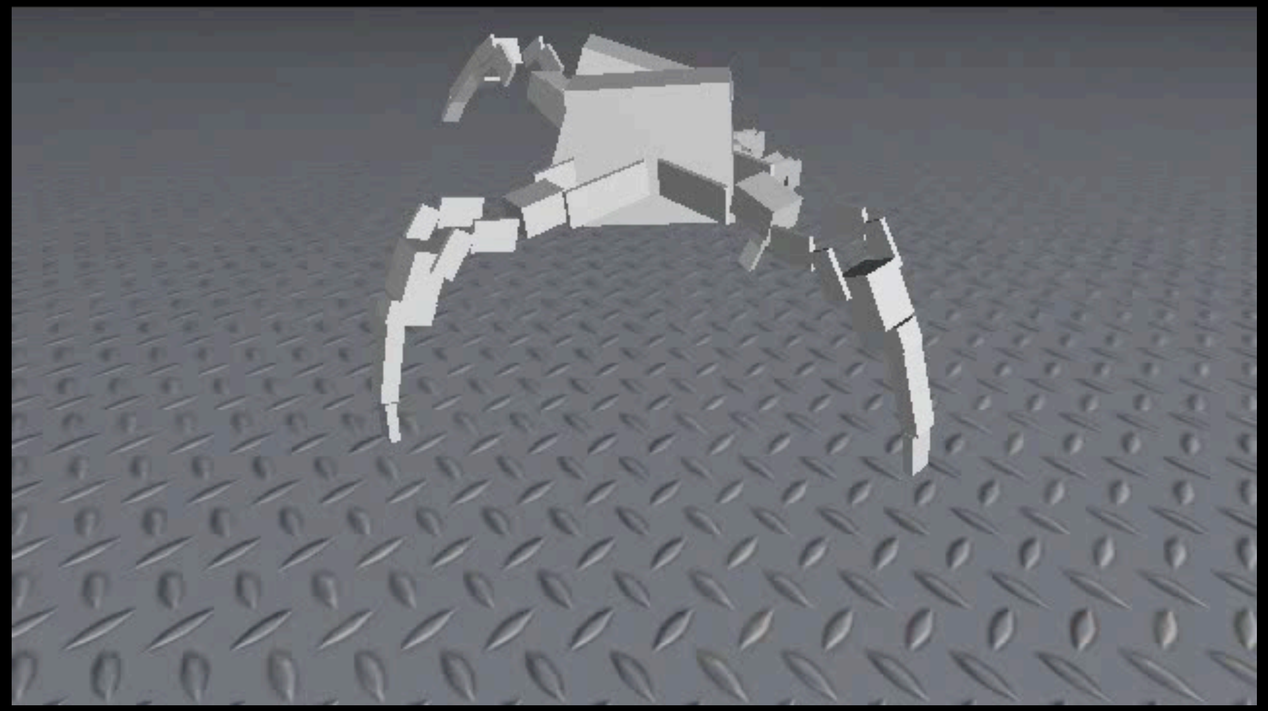
Learned Deep Topology, Learned Parameters

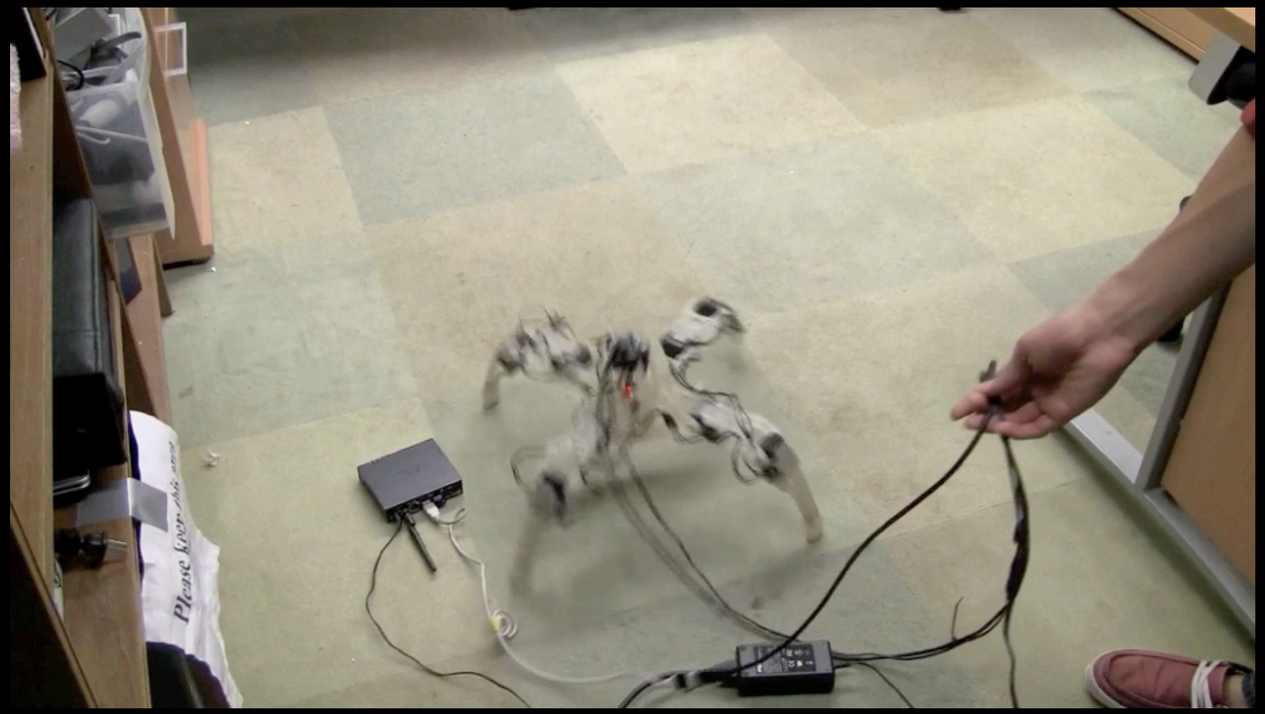


Learned Deep Topology, Learned Parameters

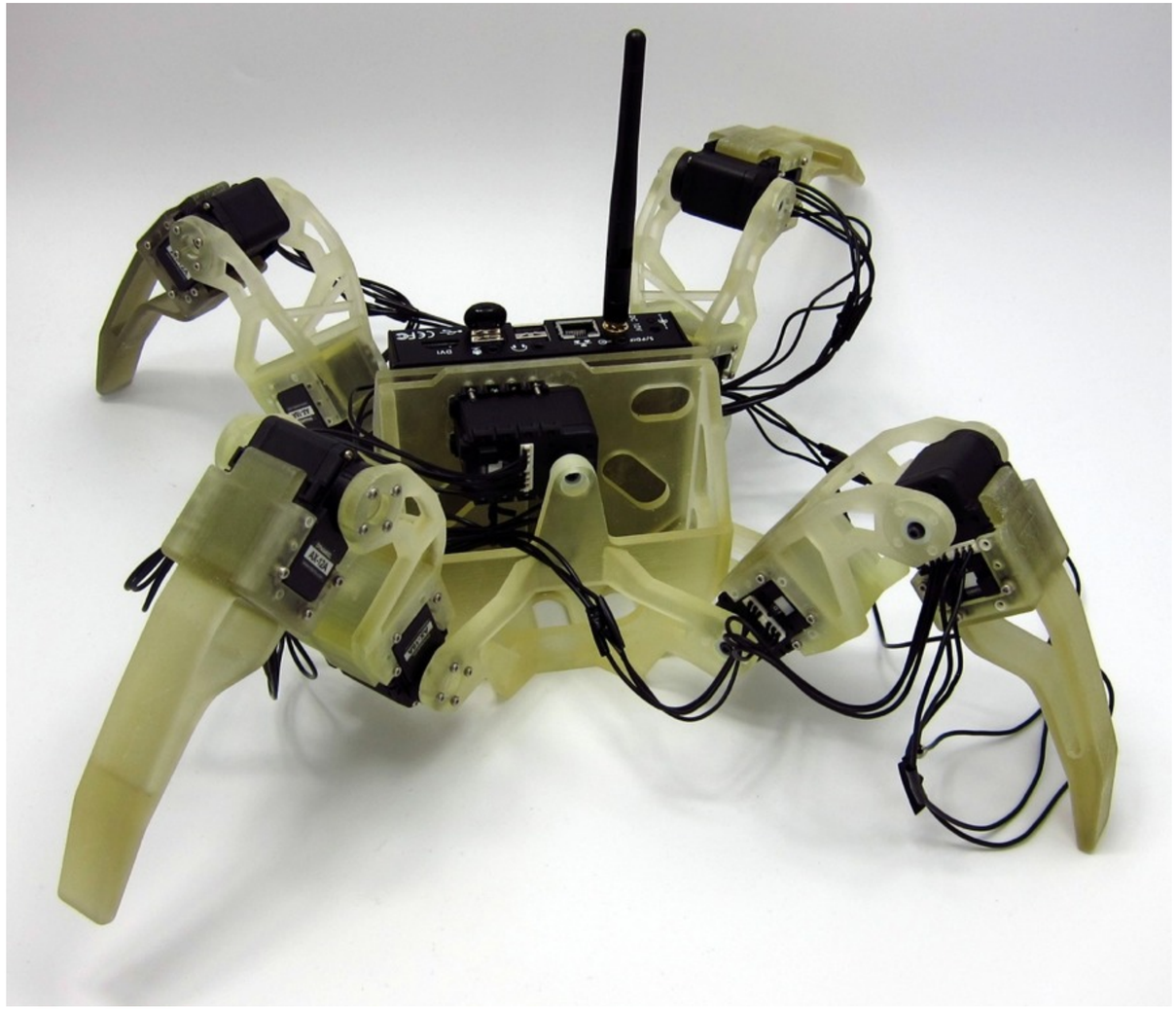


... And add a Simulator

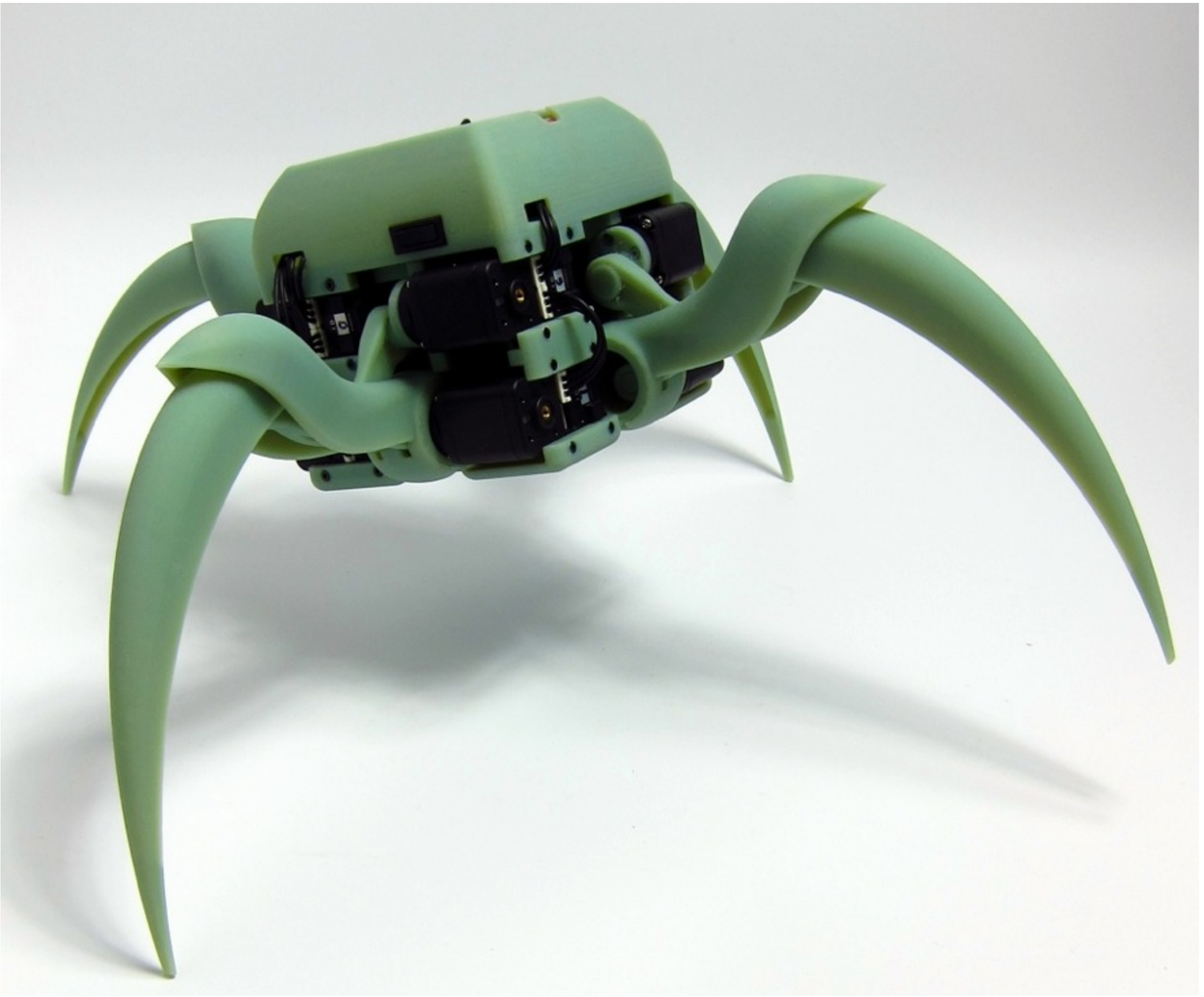




QuadraTot



Aracna

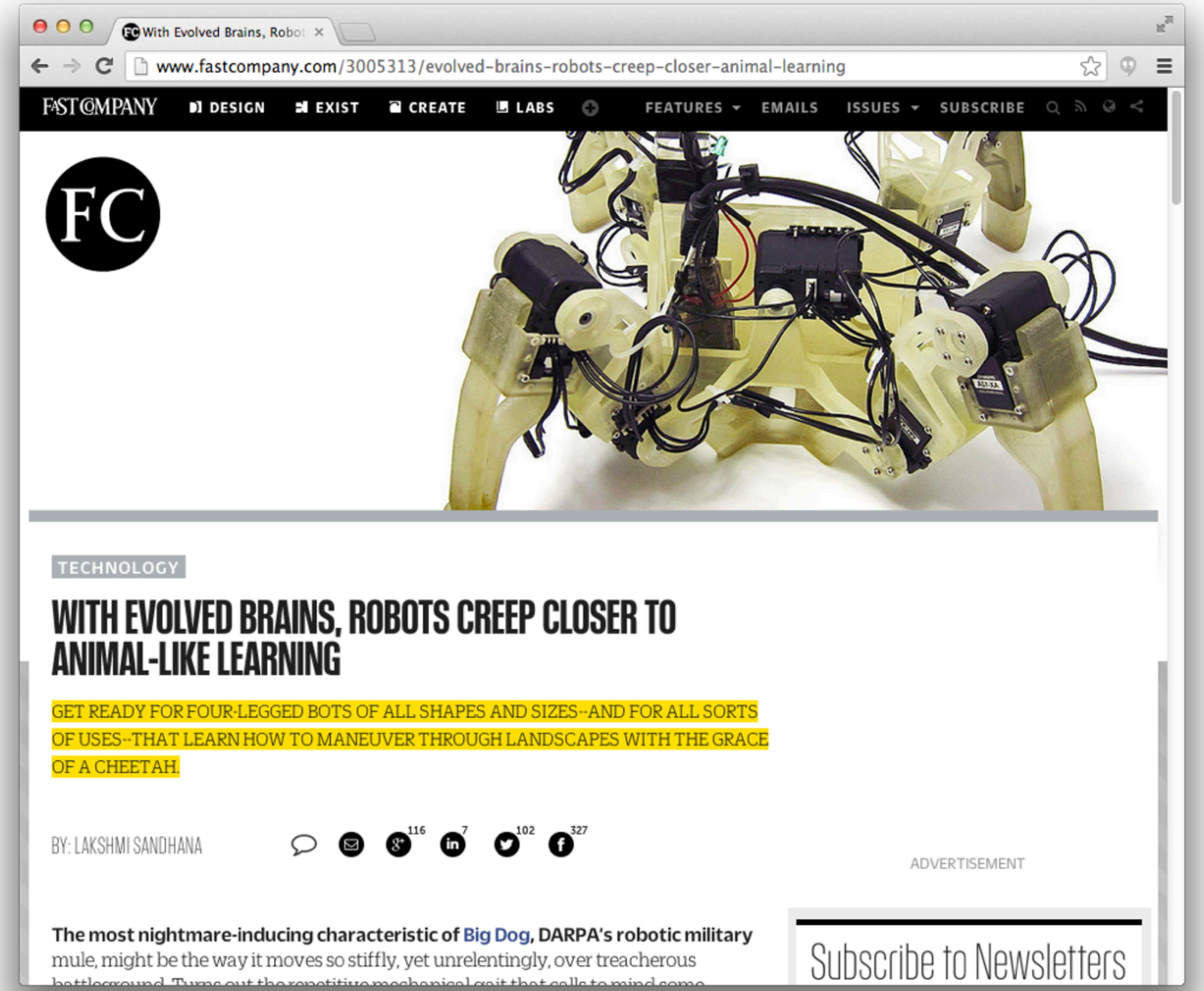


GECCO ’11, ECAL ’11, ALife ’12, Evo* ’13

(best presentation award)

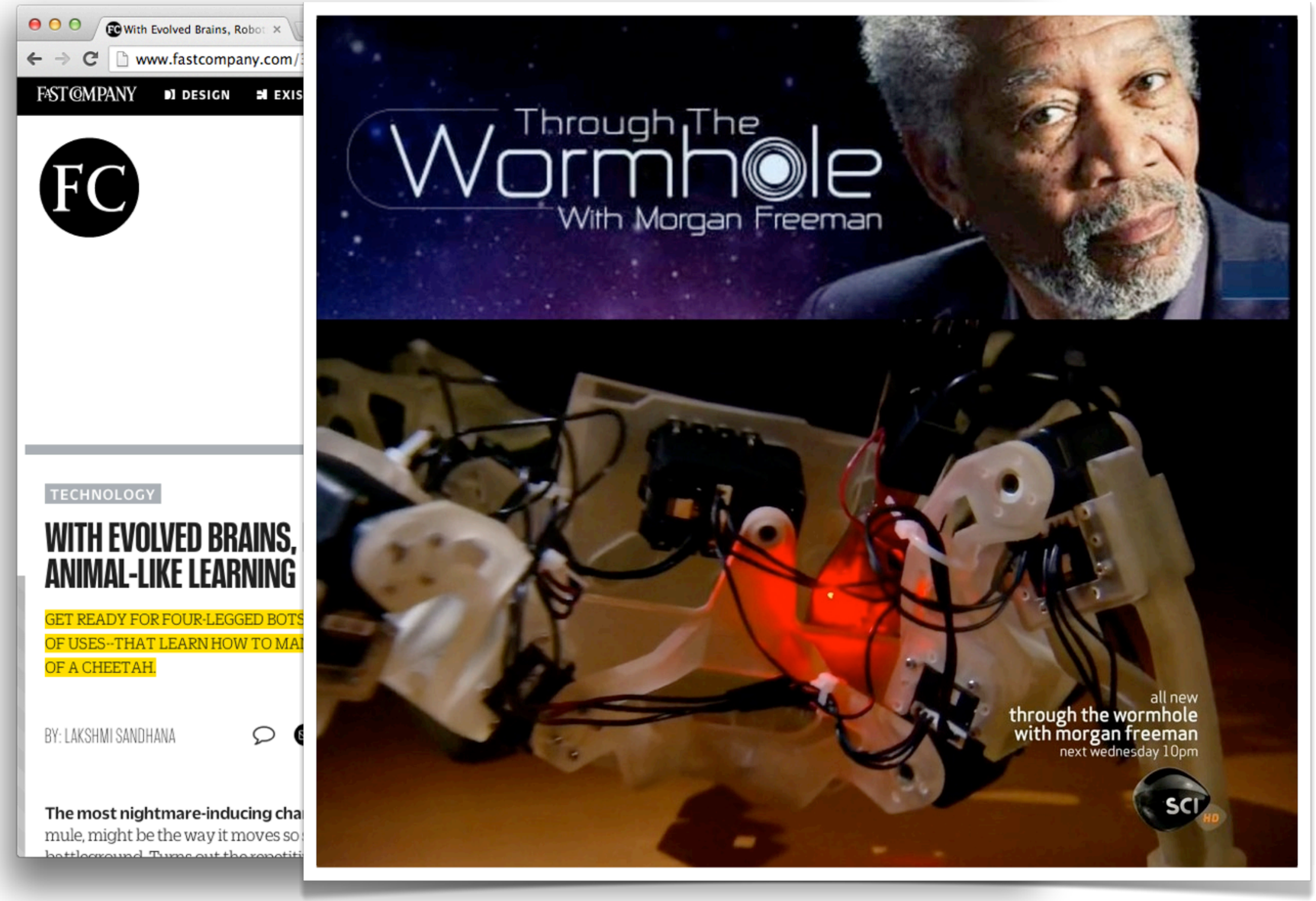
GECCO ’11, ECAL ’11, ALife ’12, Evo* ’13

(best presentation award)

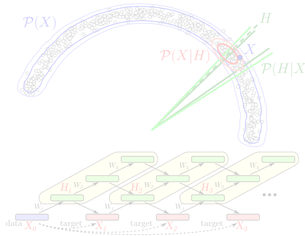


GECCO ’11, ECAL ’11, ALife ’12, Evo* ’13

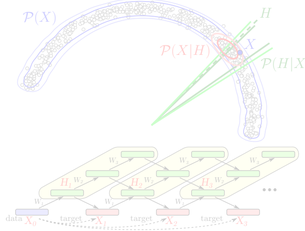
(best presentation award)

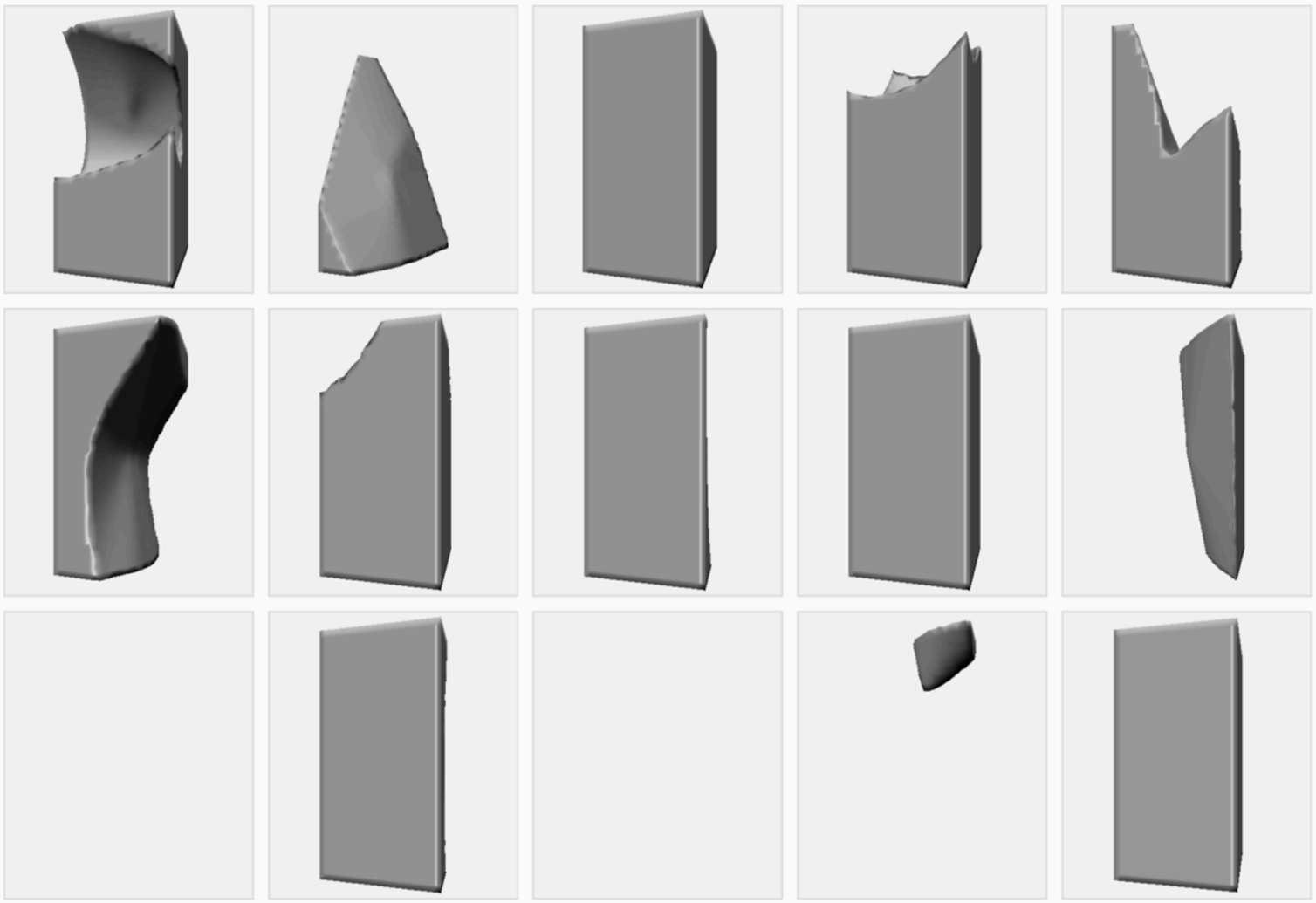


Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize



Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize





EndlessForms

login register search

Start Anew

Evolve

Save / Publish

Browse

best new
highest rated
newest
random
category
chess challenge

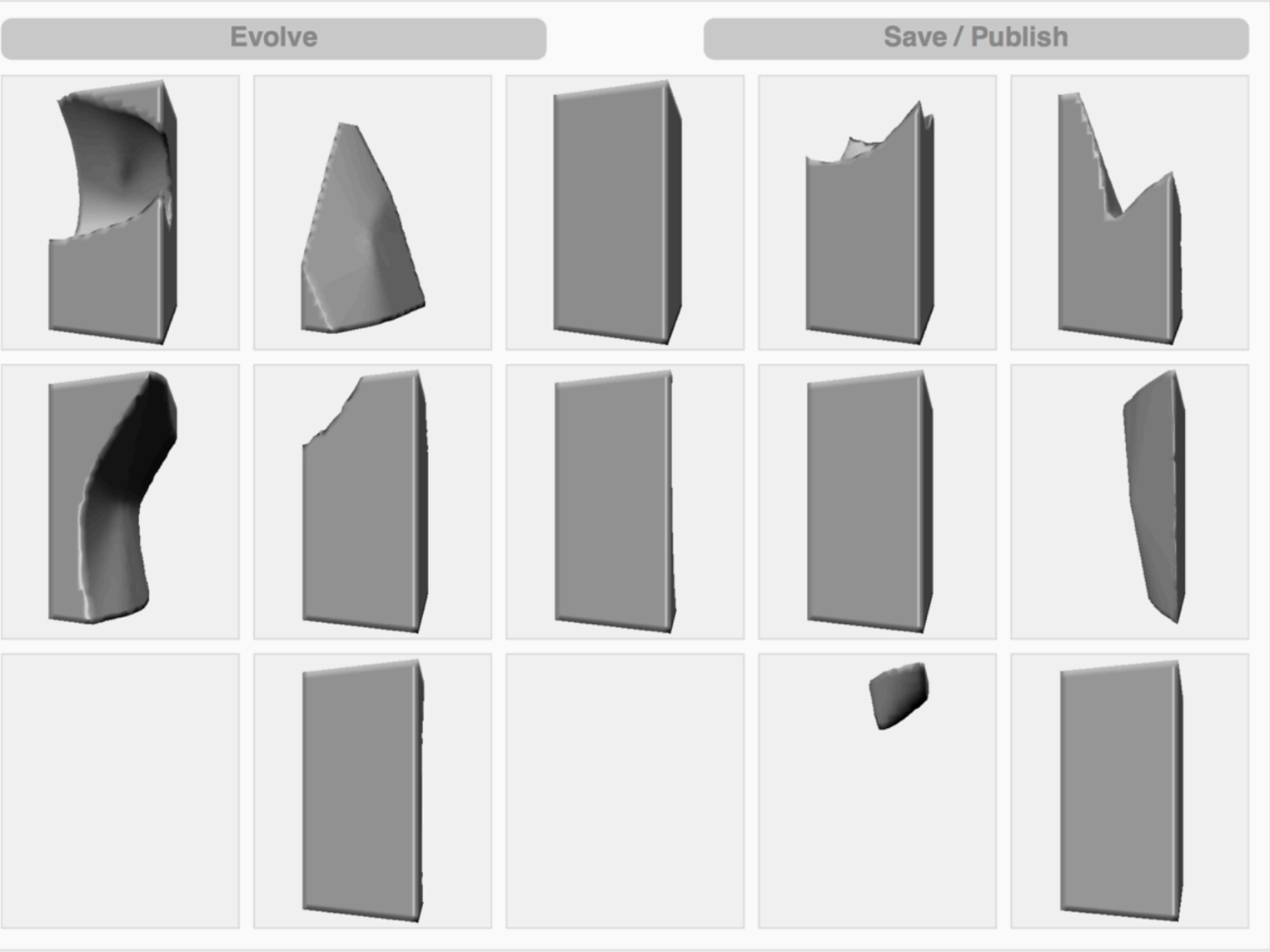
Best Users

current week
last week

Share It

Like 436
Tweet 274
g+1 126

Follow Us



EndlessForms

login register search

Start Anew

Browse

best new
highest rated
newest
random
category
chess challenge

Best Users

current week
last week

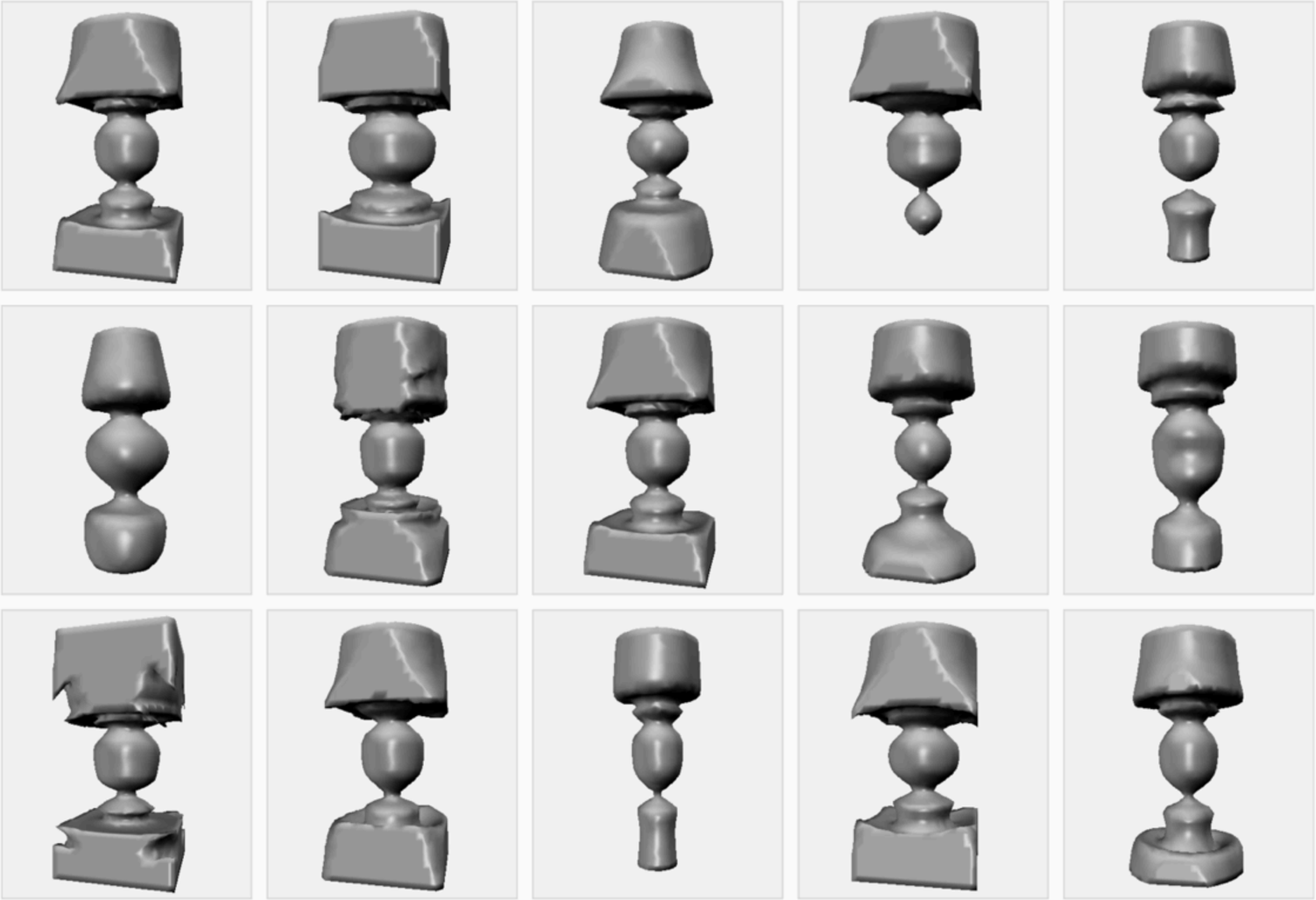
Share It

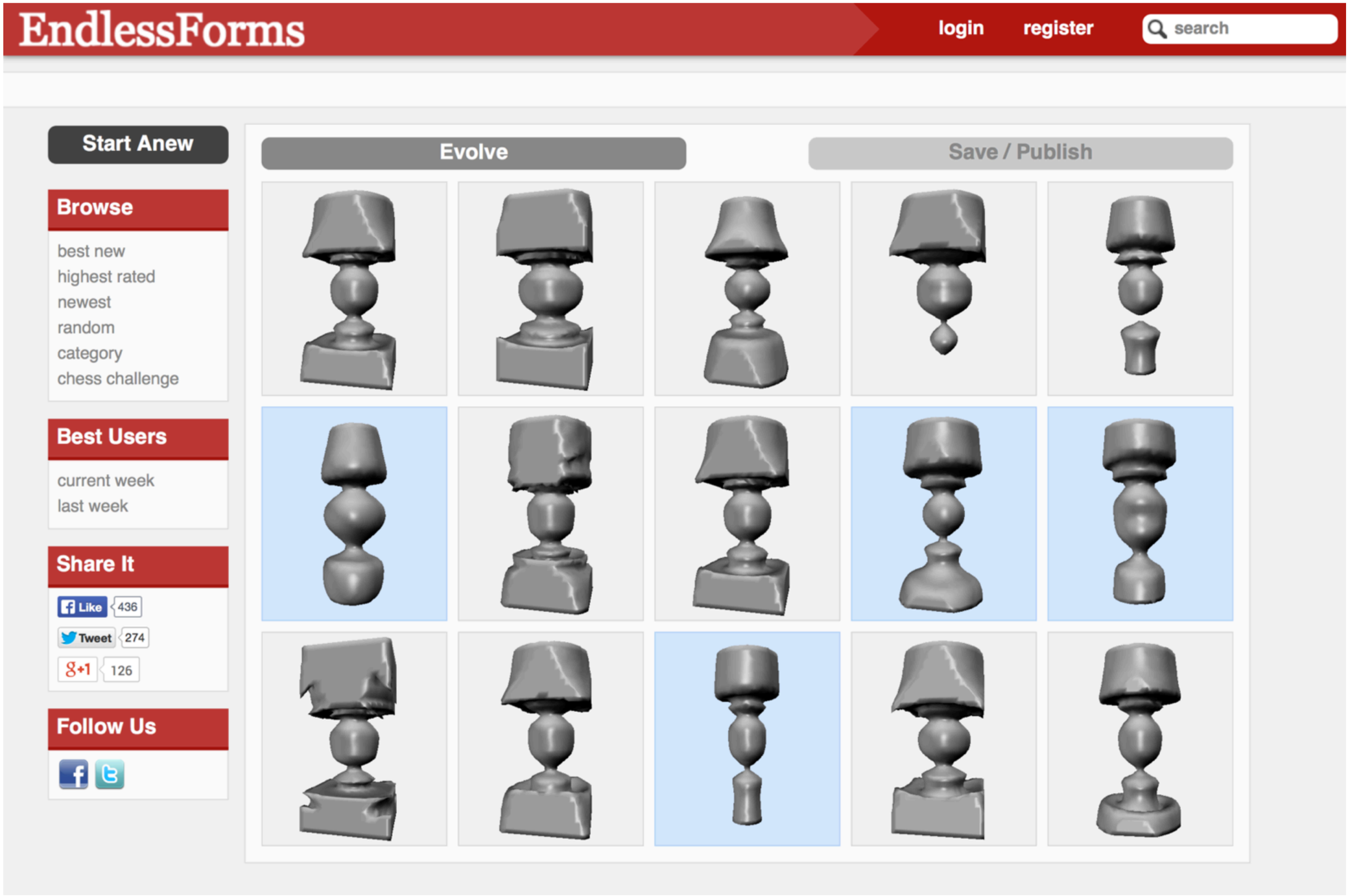
Like 436
Tweet 274
g+1 126

Follow Us

Evolve

Save / Publish




EndlessForms
login
register
search
Start Anew
Evolve
Save / Publish
Browse
best new
highest rated
newest
random
category
chess challenge
Best Users
current week
last week
Share It
Like 436
Tweet 274
g+1 126
Follow Us

EndlessForms

login register search

Start Anew

Browse

best new
highest rated
newest
random
category
chess challenge

Best Users

current week
last week

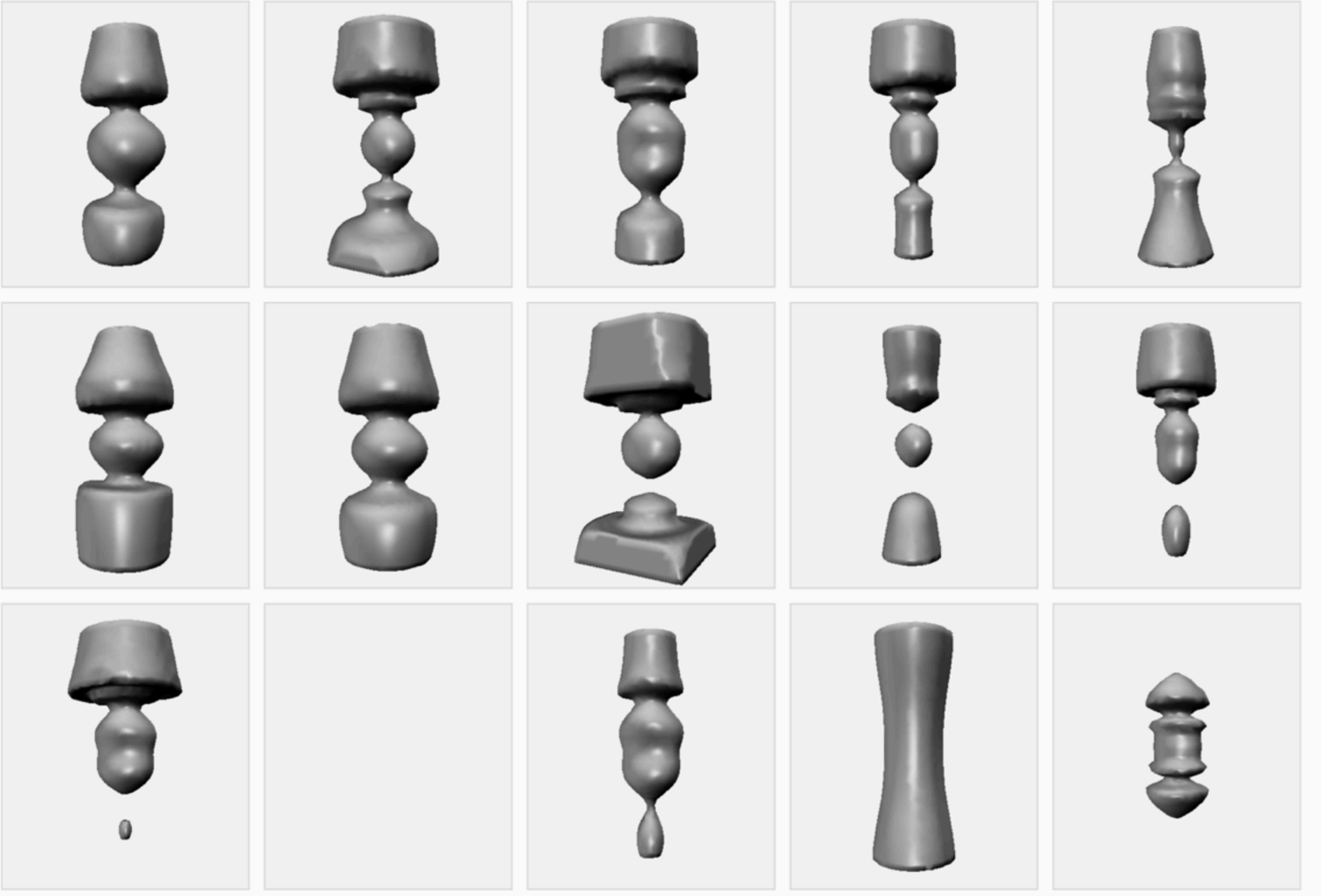
Share It

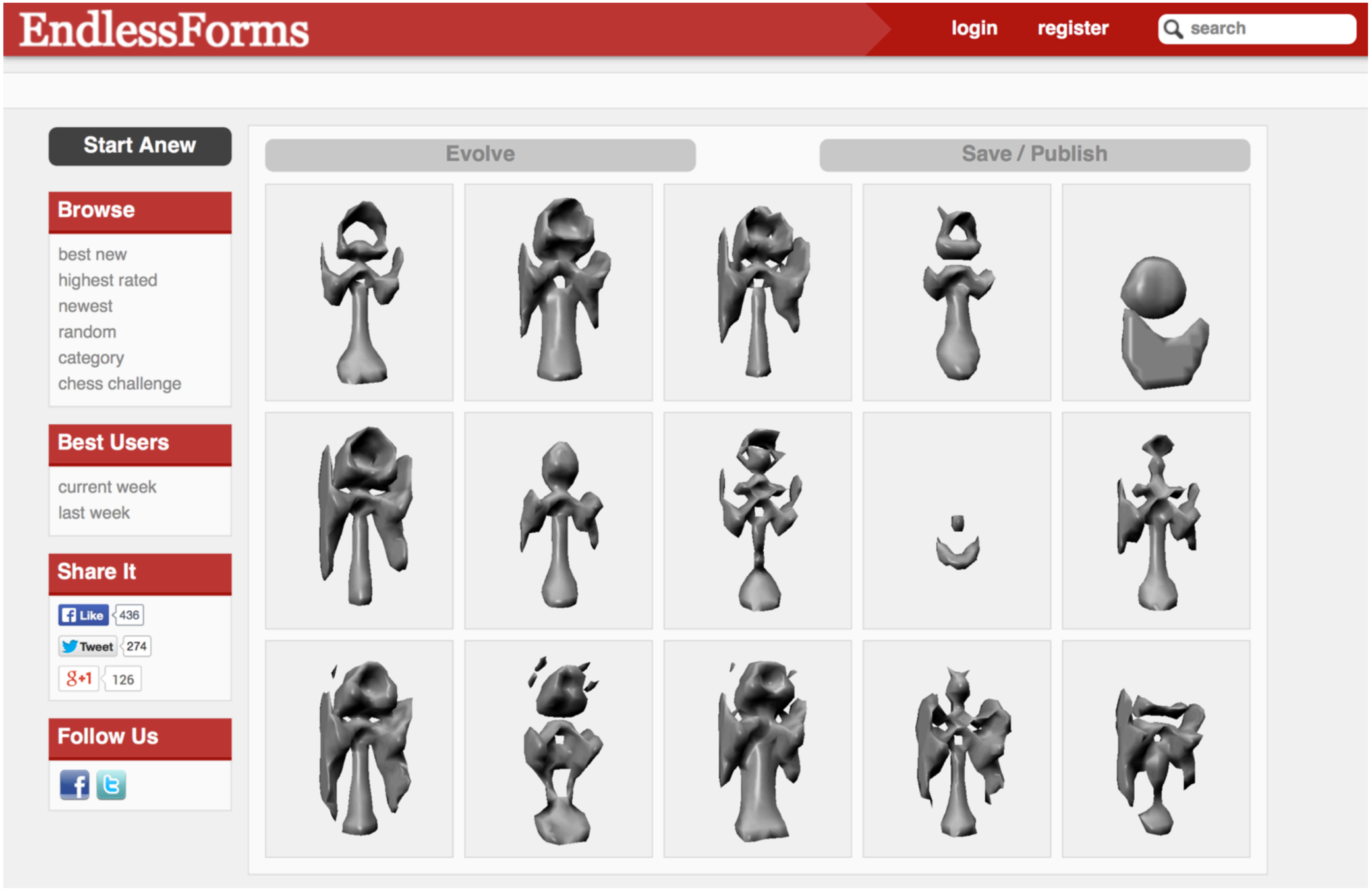
Like 436
Tweet 274
g+1 126

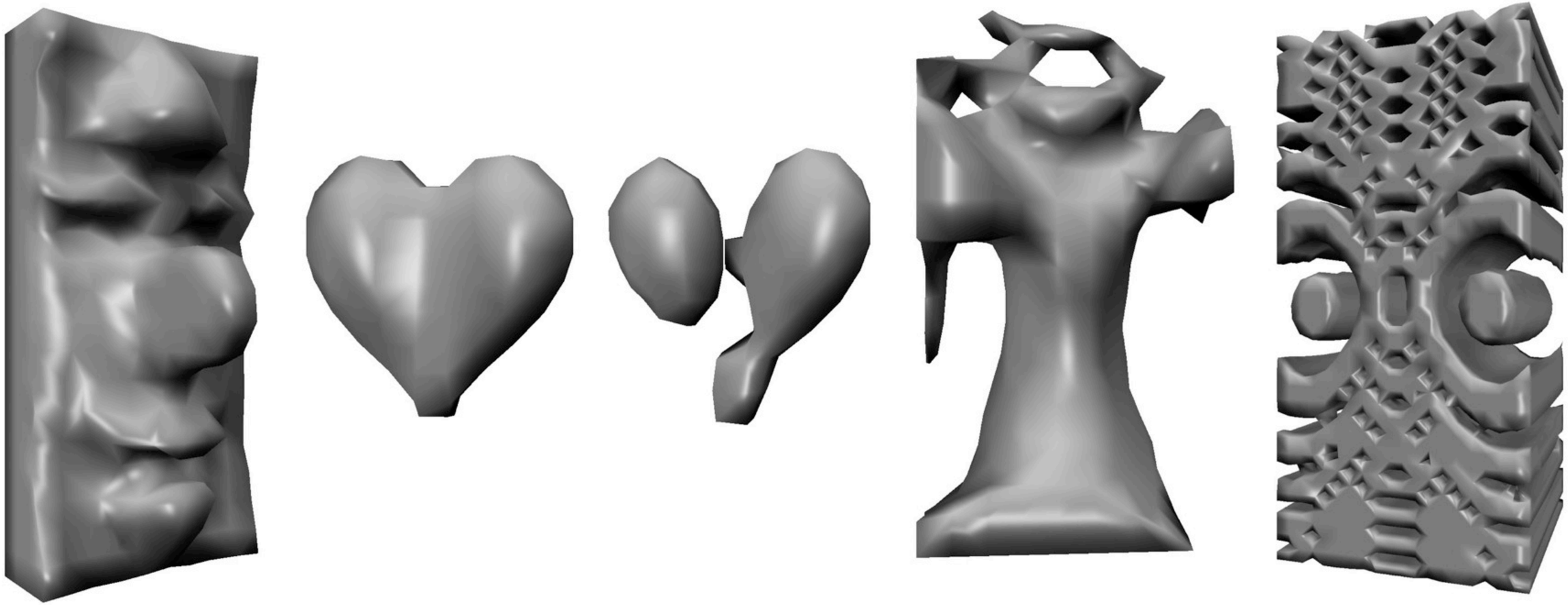
Follow Us

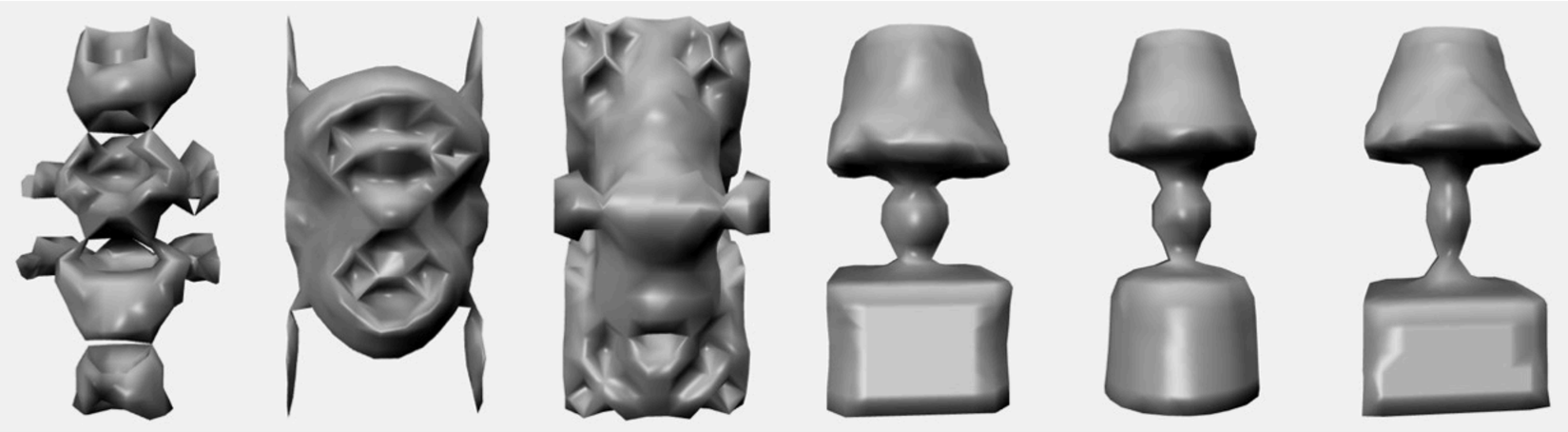
Evolve

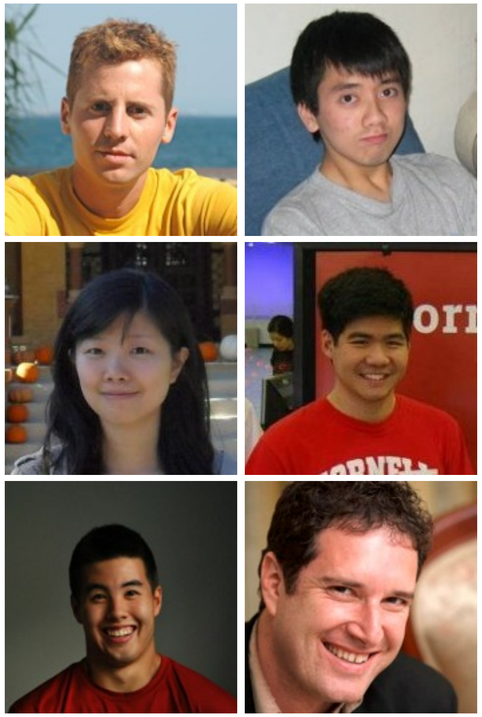
Save / Publish

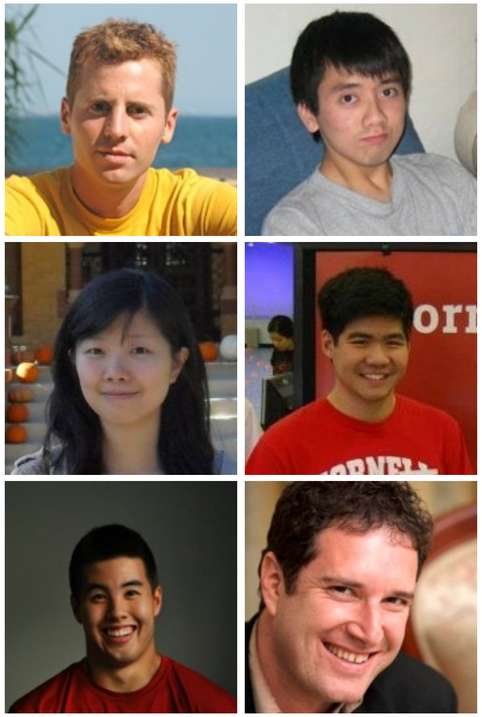



EndlessForms
login
register
search
Start Anew
Evolve
Save / Publish
Browse
best new
highest rated
newest
random
category
chess challenge
Best Users
current week
last week
Share It
Like 436
Tweet 274
g+1 126
Follow Us









4,800,000+

shapes

Slashdot

COMMUNICATIONS OF THE ACM

Technology PUBLISHED BY MIT Review

Y Hacker News

KURZWEIL ACCELERATING INTELLIGENCE

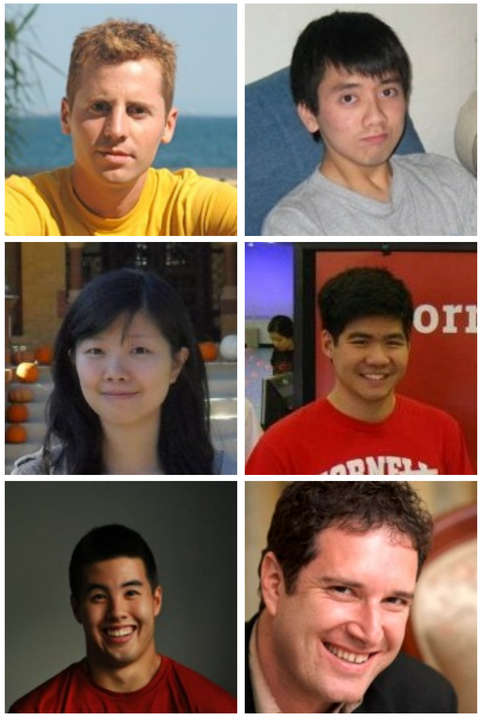
msnbc.com

CHRONICLE ONLINE Daily News From Cornell University

NewScientist

ALife ’12, (submitted) Siggraph ’15 (x2), patent

(best poster award)



4,800,000+

shapes

Slashdot

COMMUNICATIONS OF THE ACM

Technology Review PUBLISHED BY MIT

Y Hacker News

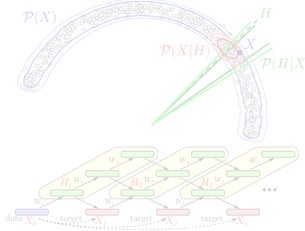
KURZWEIL ACCELERATING INTELLIGENCE

msnbc.com

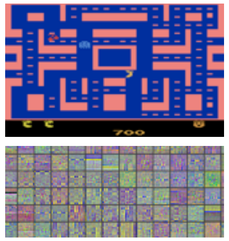
CHRONICLE ONLINE Daily News From Cornell University

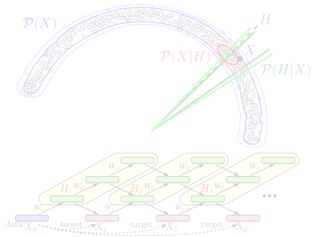
NewScientist

Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

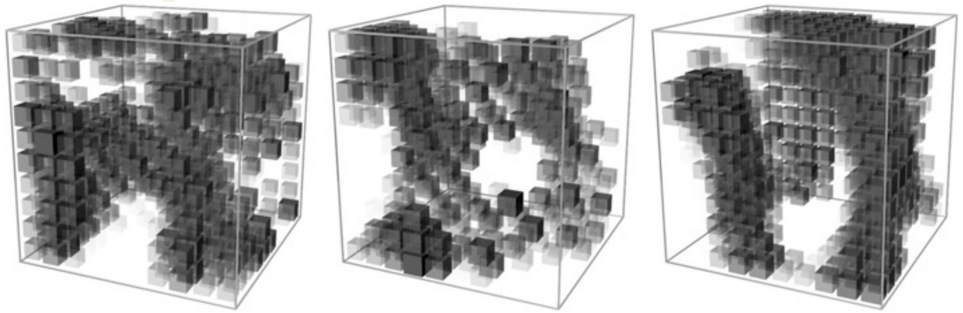


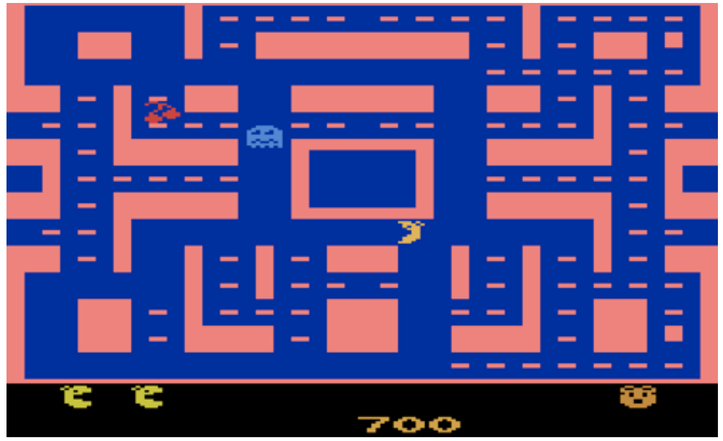
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

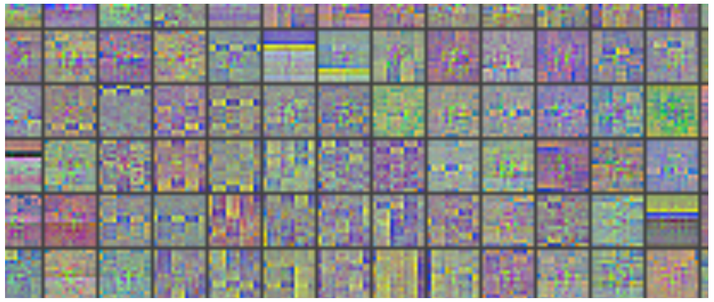




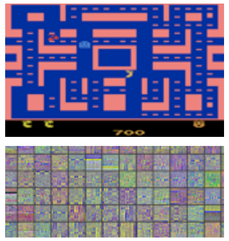
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

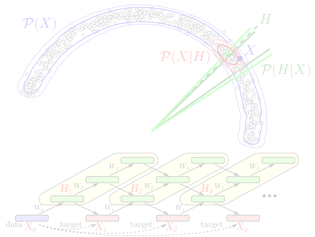




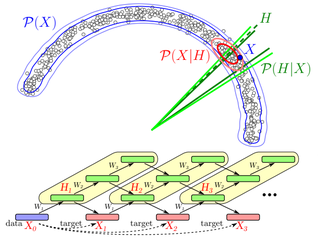


Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

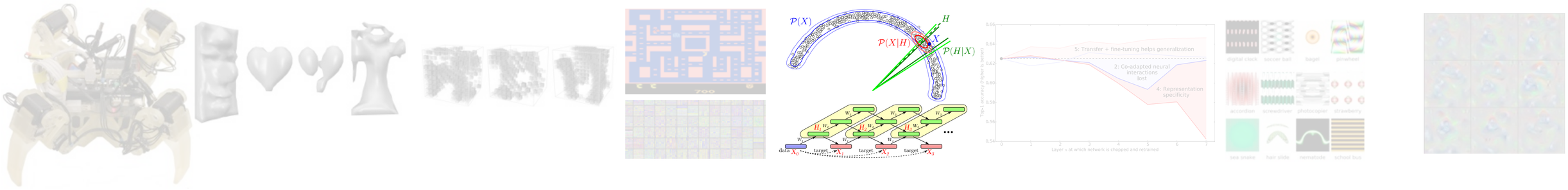




Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

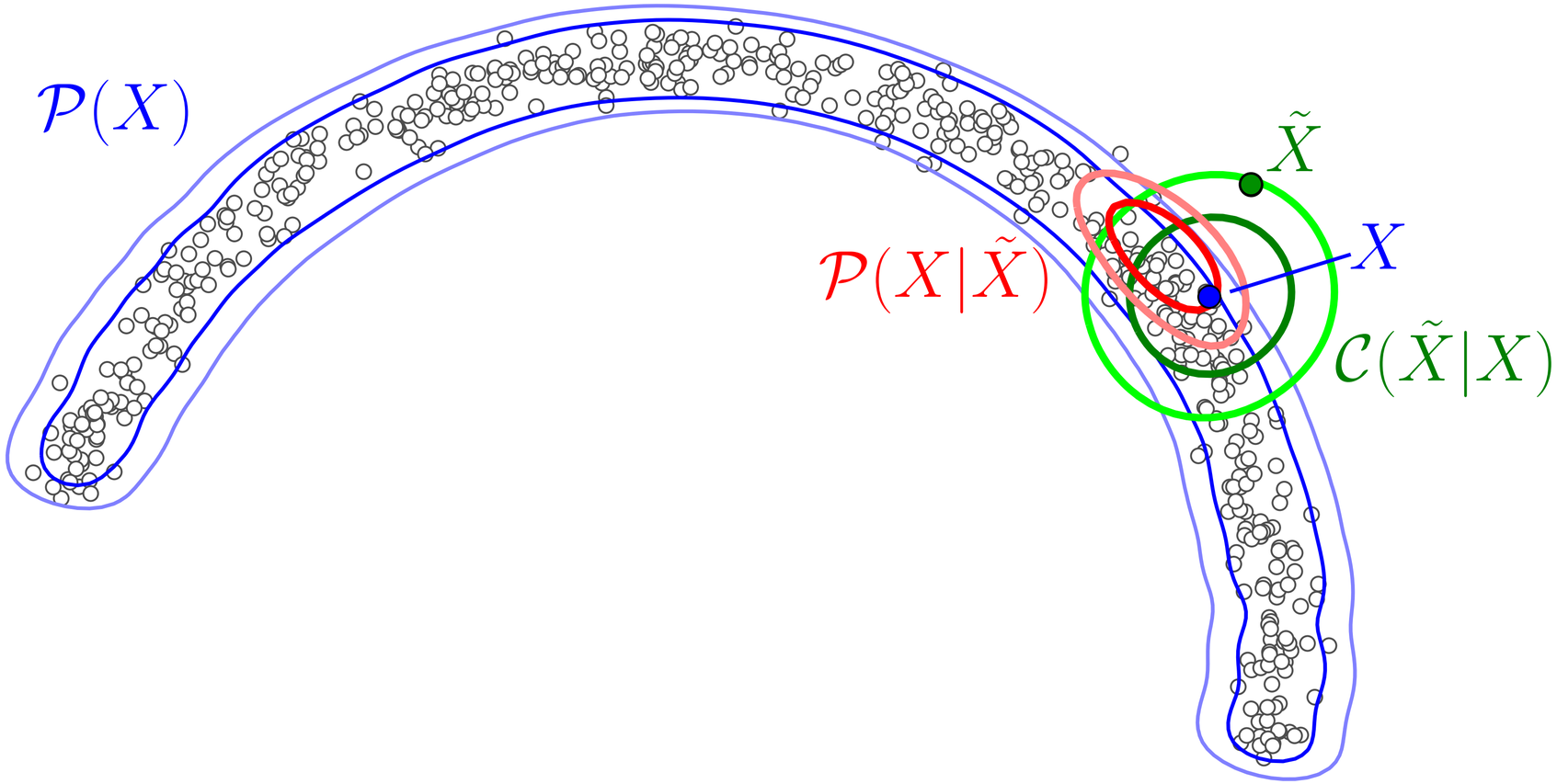


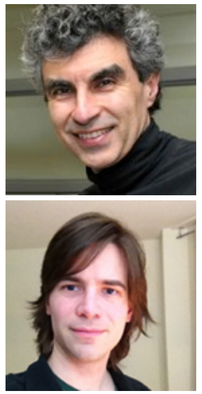
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize



See: Cornell Brown Bag talk slides, 19 March 2014.

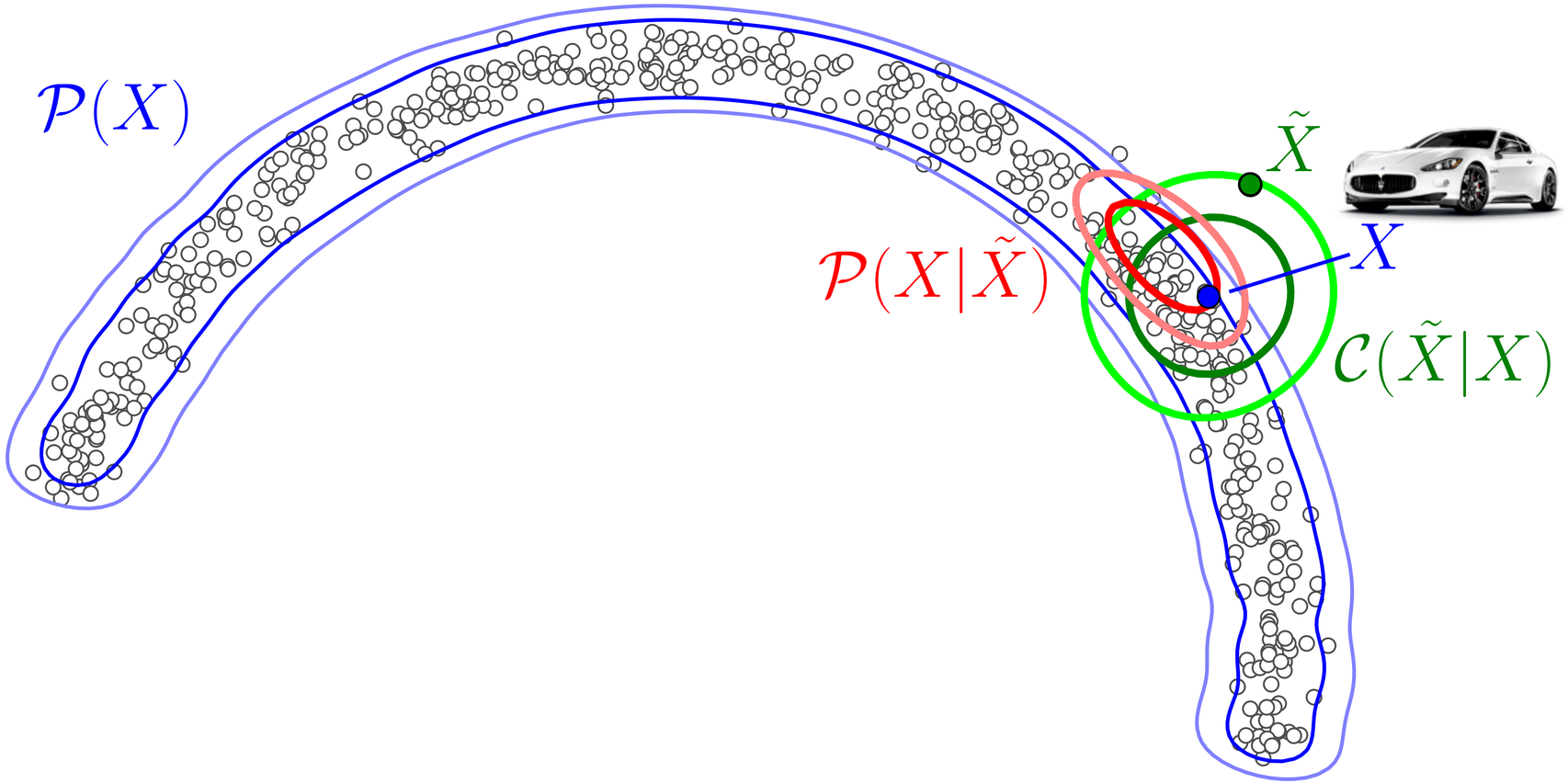
Generative Stochastic Networks

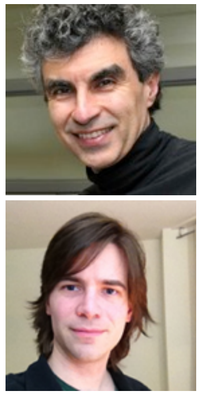




Eric Thibodeau-Laufer

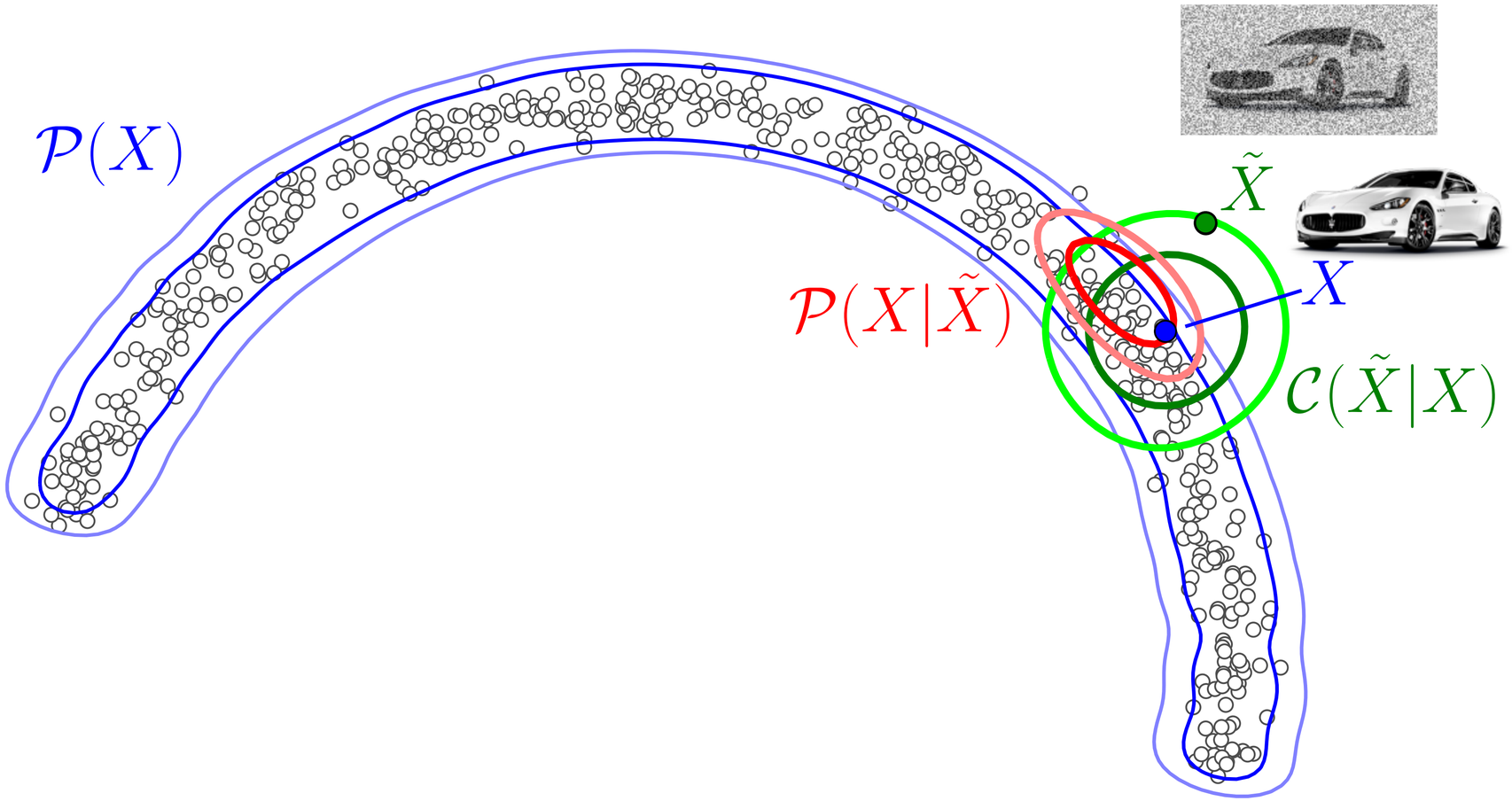
Generative Stochastic Networks

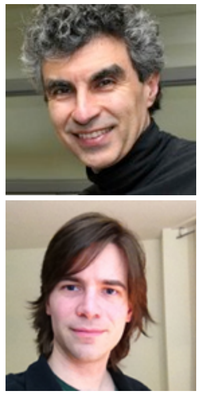




Eric Thibodeau-Laufer

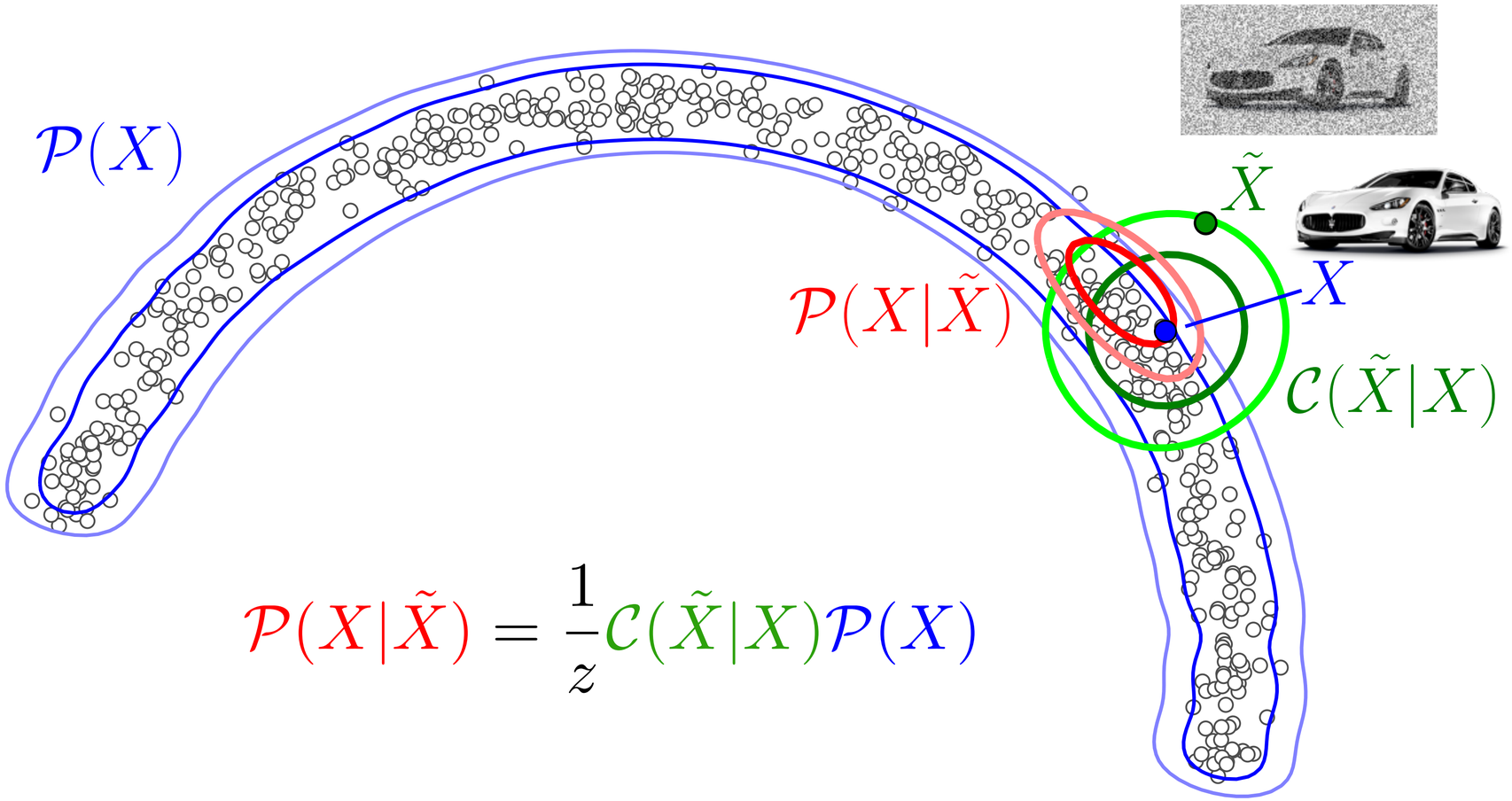
Generative Stochastic Networks



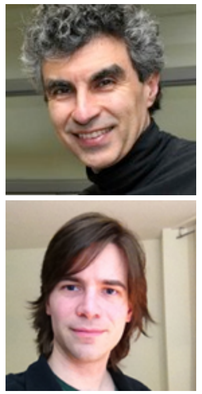


Eric Thibodeau-Laufer

Generative Stochastic Networks

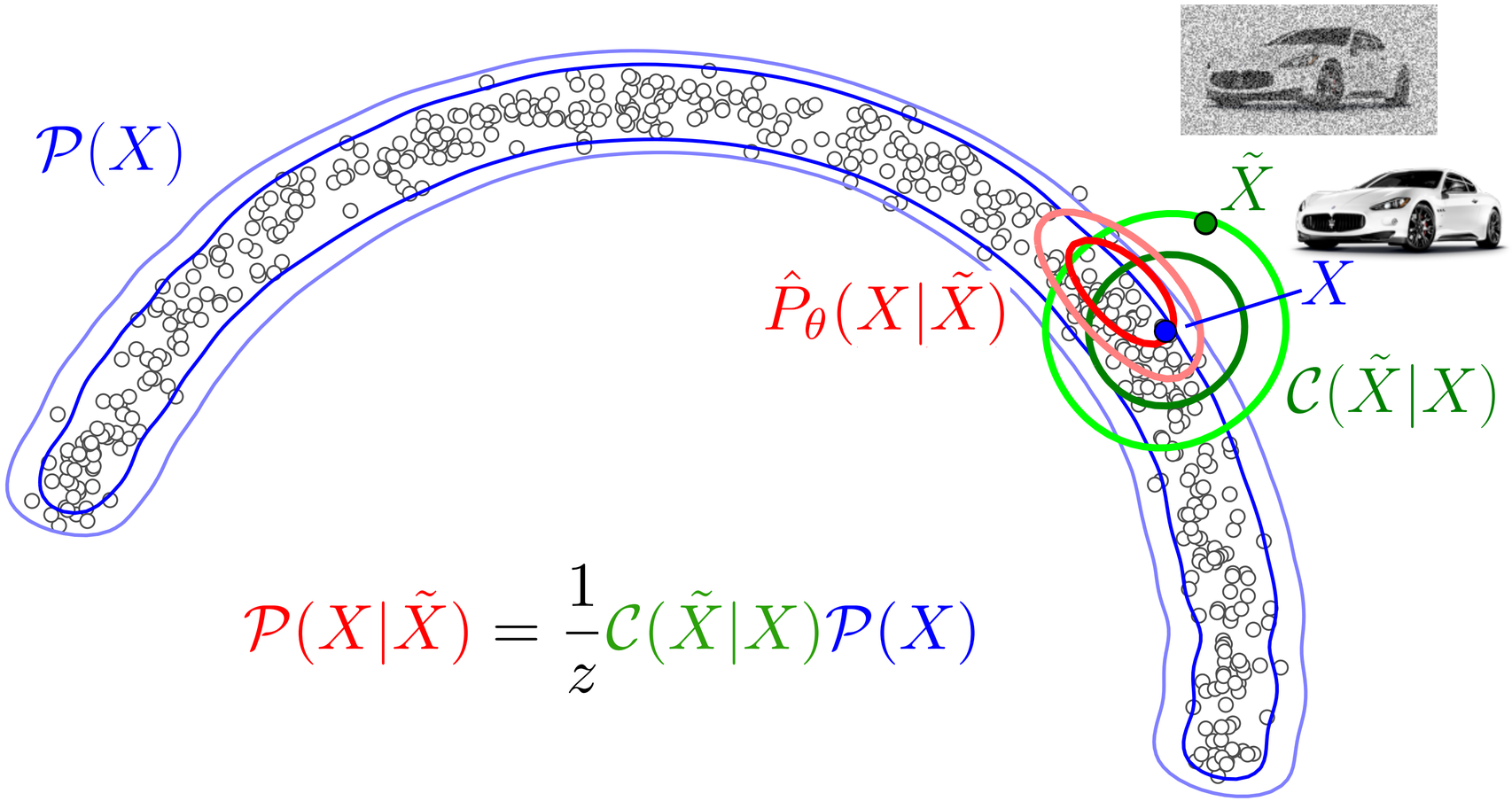


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

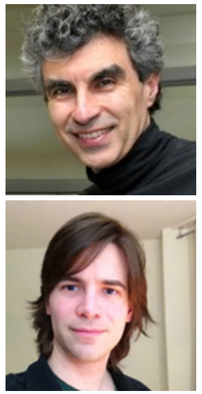


Eric Thibodeau-Laufer

Generative Stochastic Networks

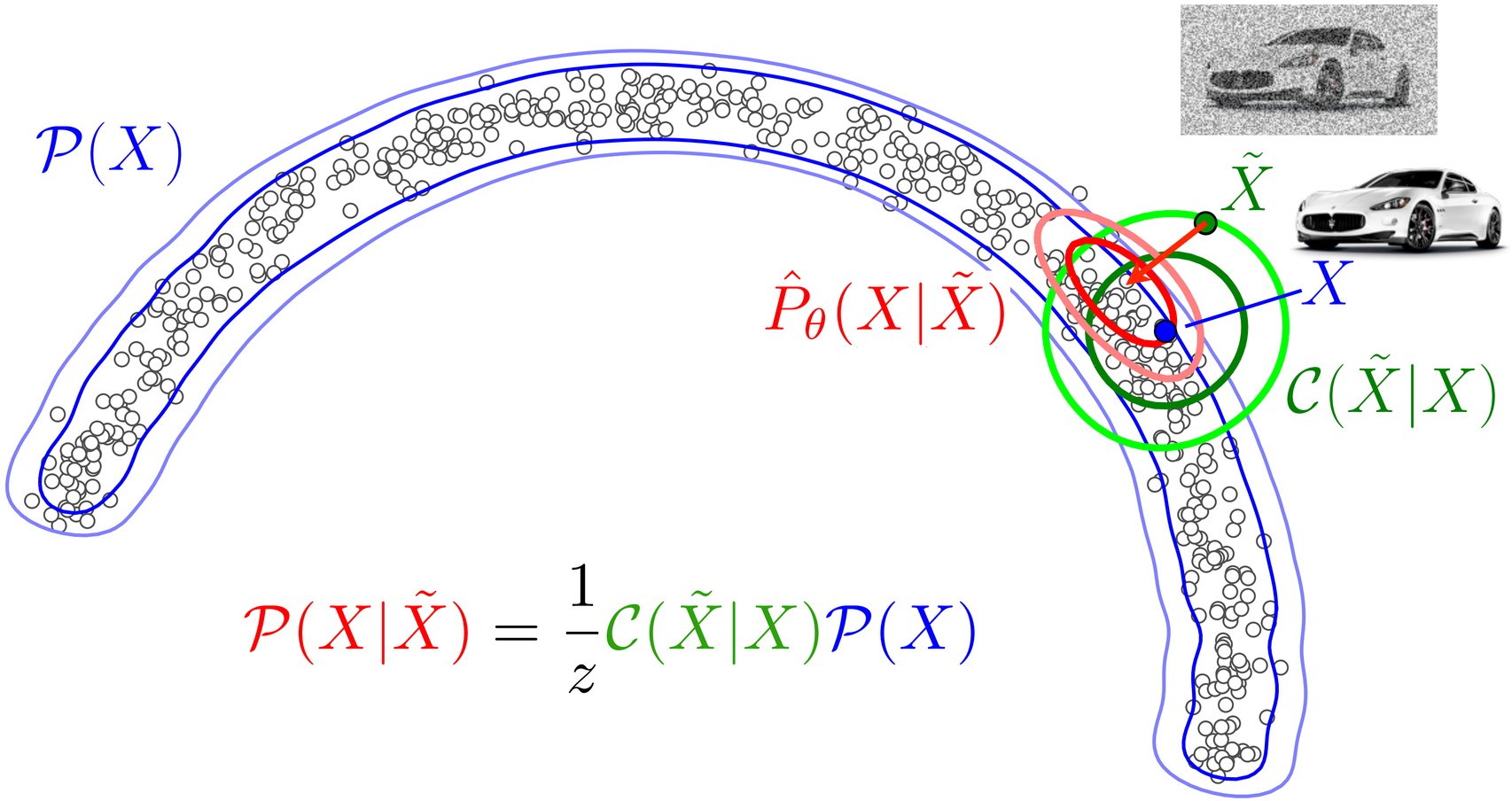


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

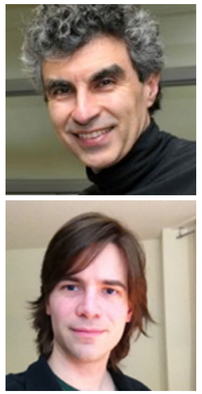


Eric Thibodeau-Laufer

Generative Stochastic Networks

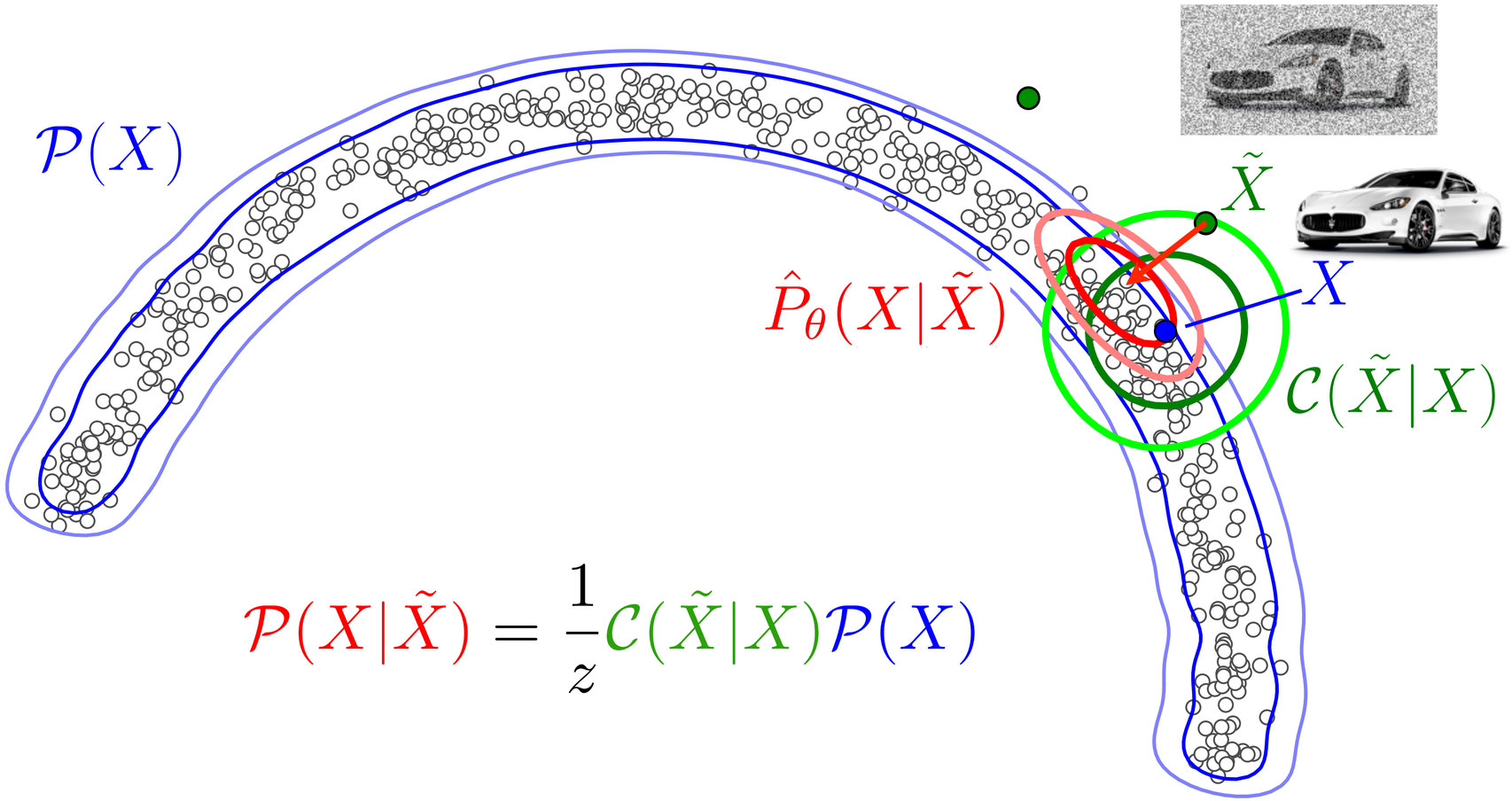


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

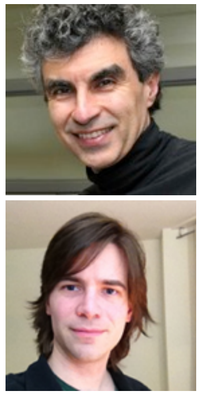


Eric Thibodeau-Laufer

Generative Stochastic Networks

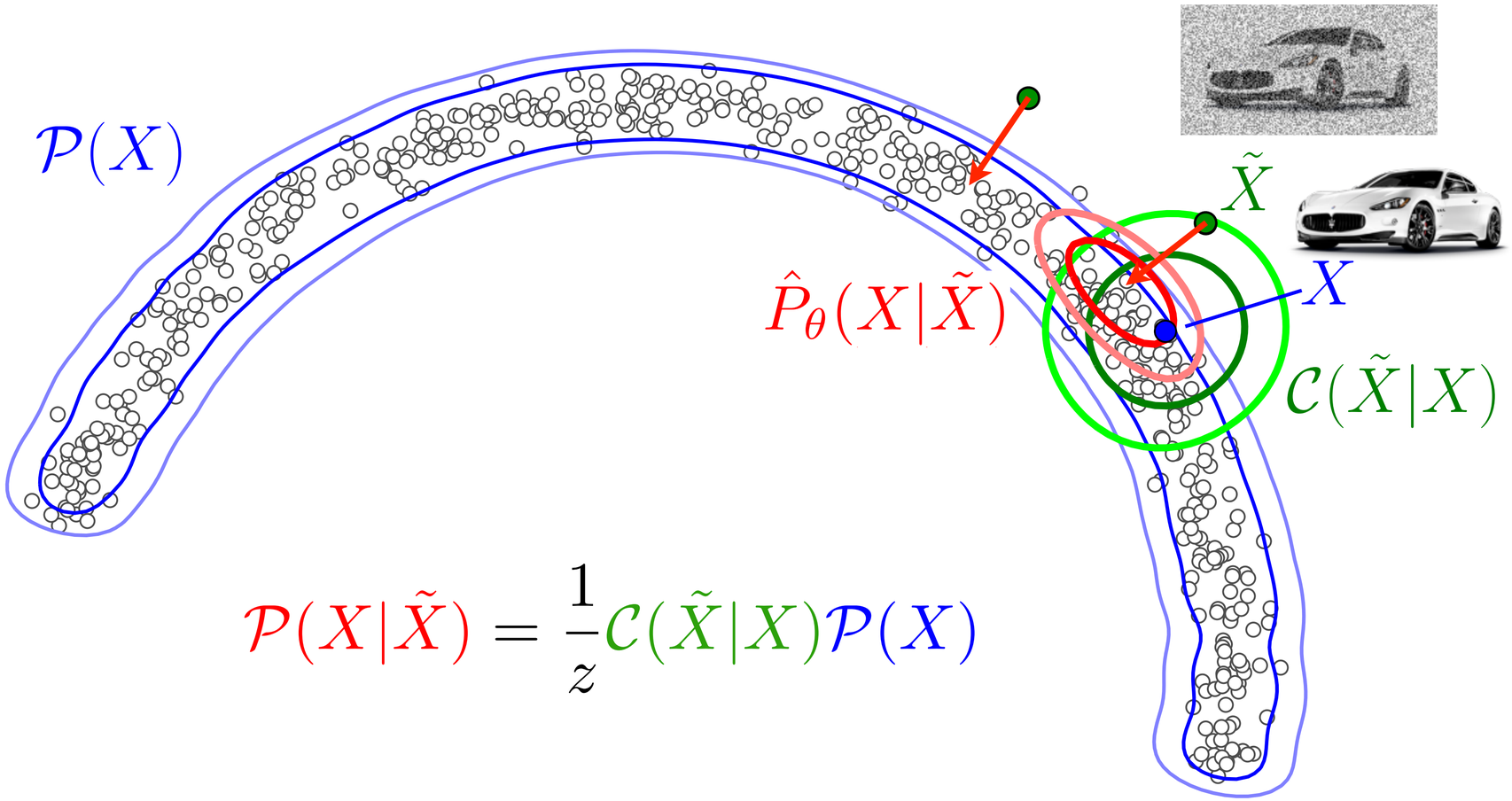


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

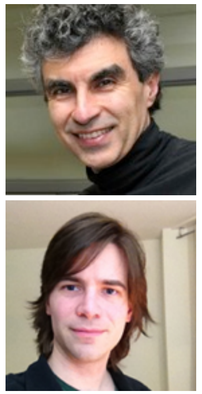


Eric Thibodeau-Laufer

Generative Stochastic Networks

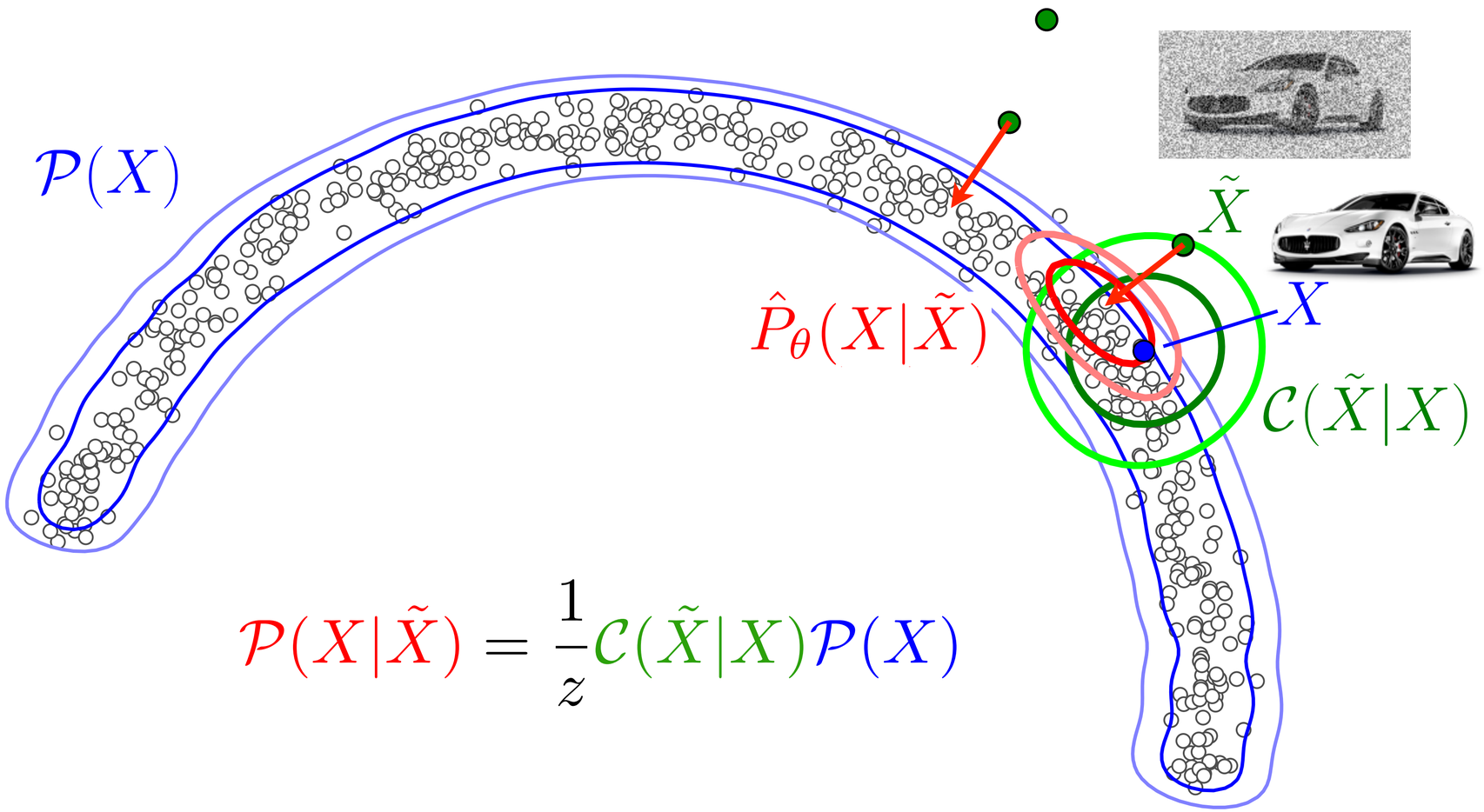


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

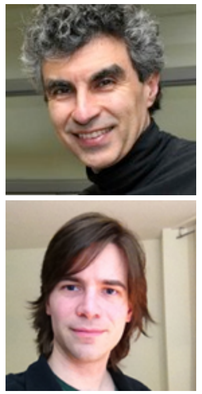


Eric Thibodeau-Laufer

Generative Stochastic Networks

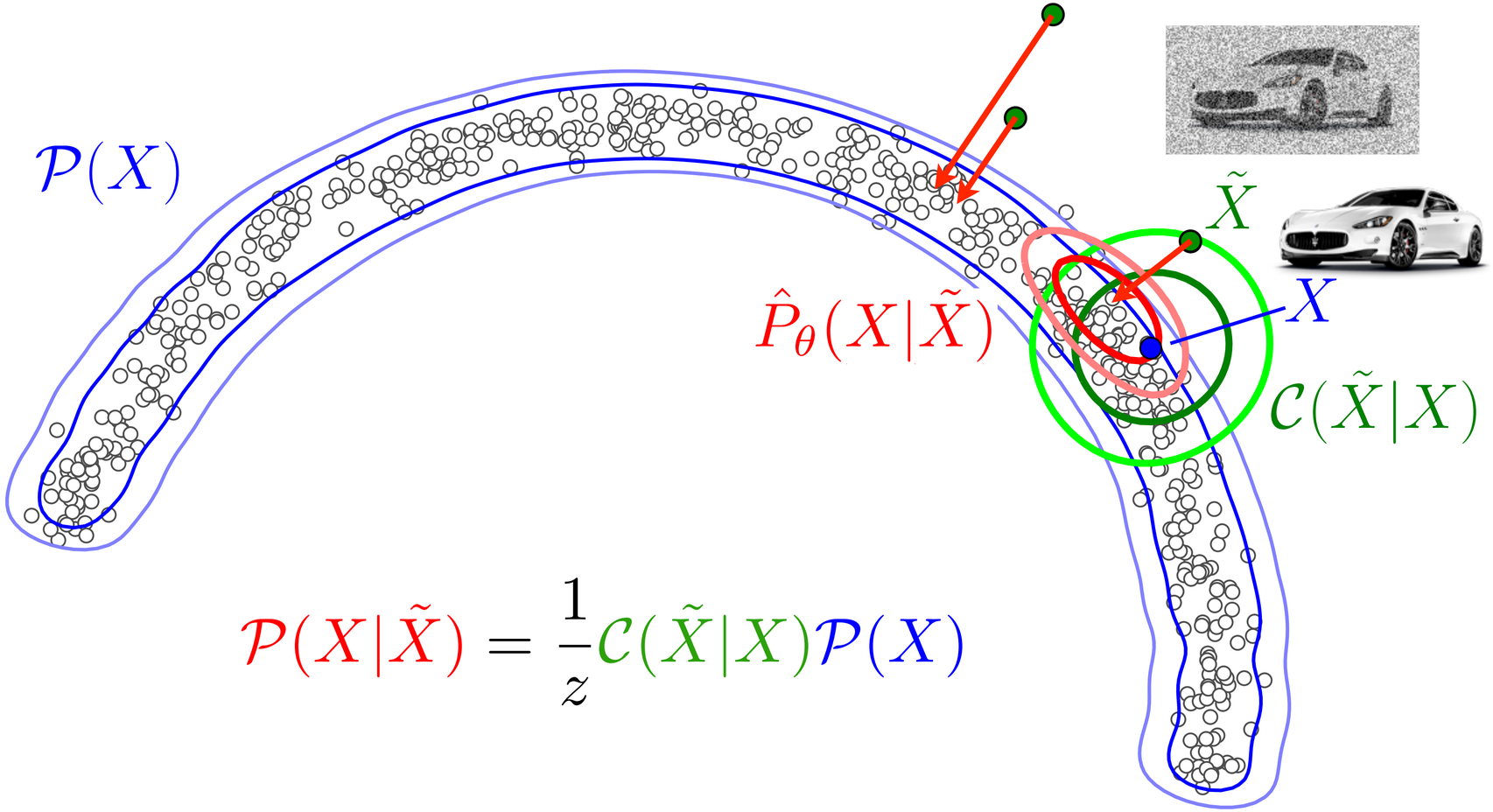


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

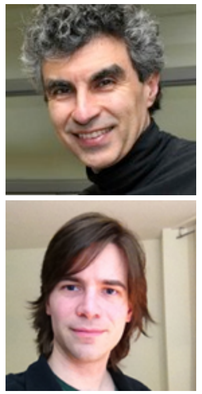


Eric Thibodeau-Laufer

Generative Stochastic Networks

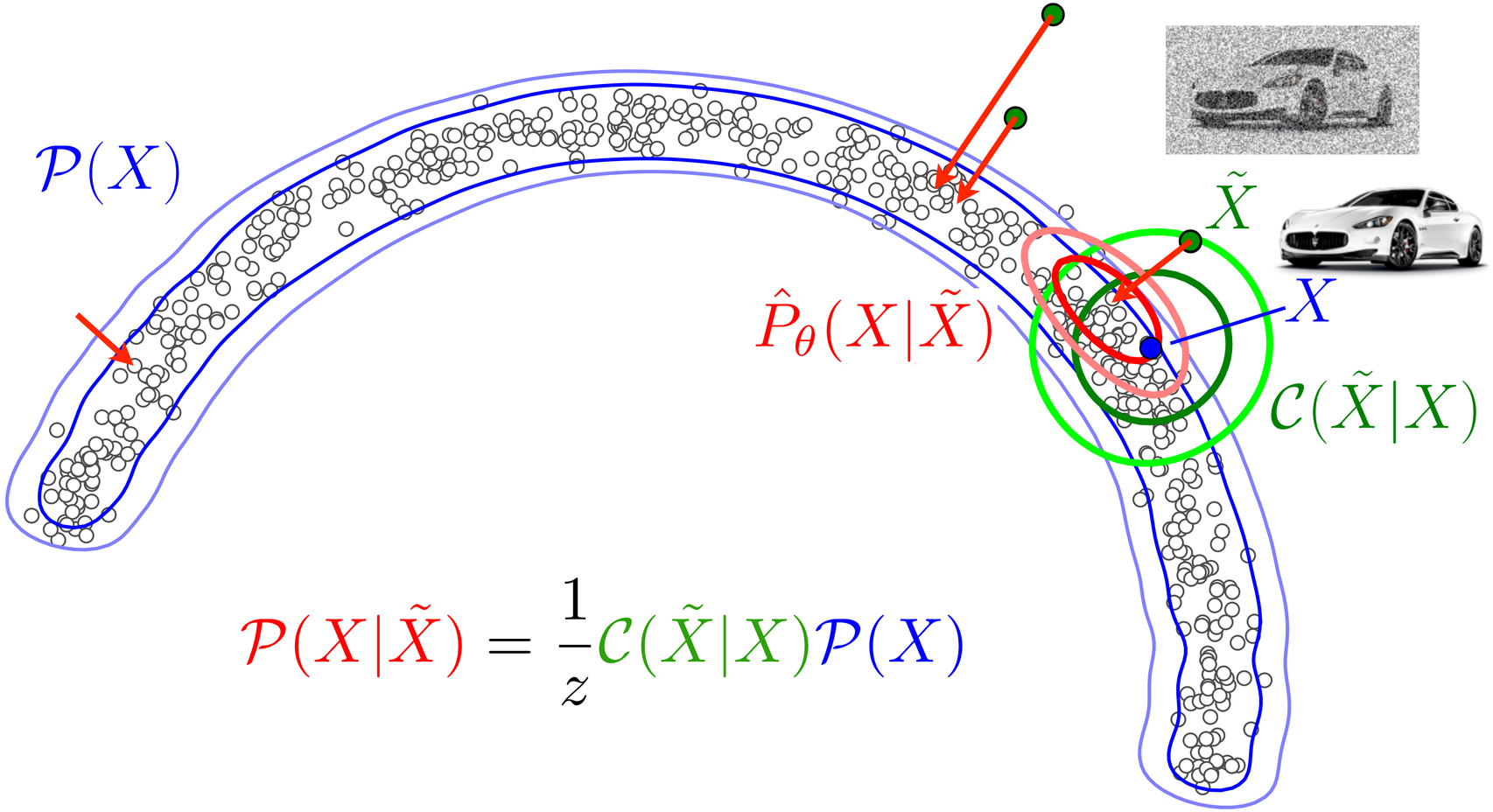


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

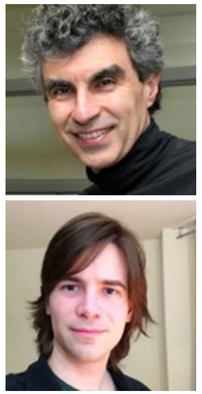


Eric Thibodeau-Laufer

Generative Stochastic Networks

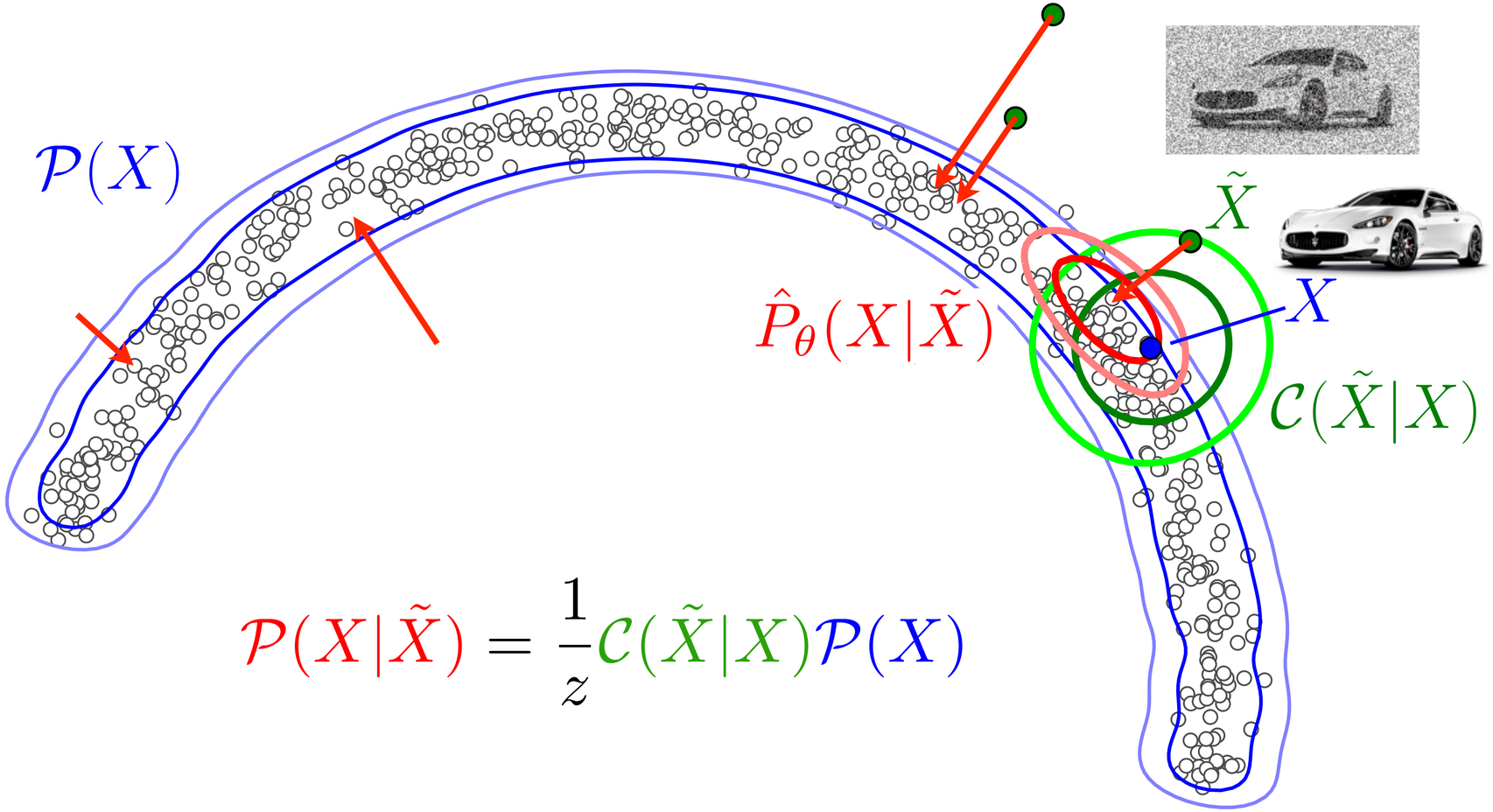


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

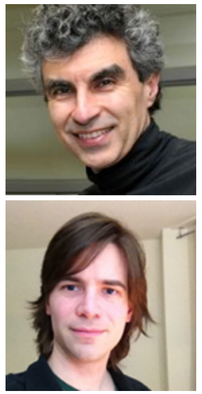


Eric Thibodeau-Laufer

Generative Stochastic Networks

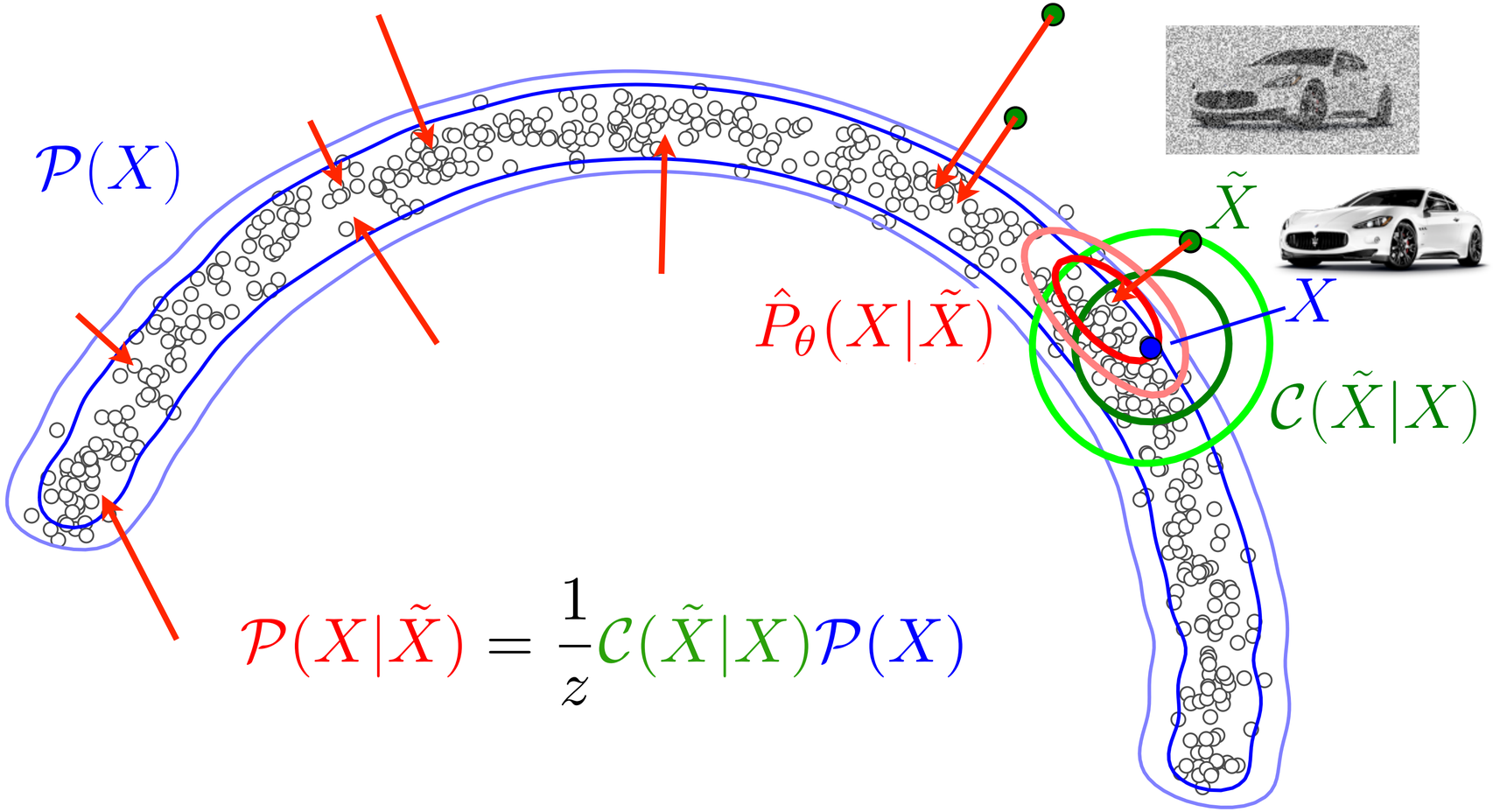


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

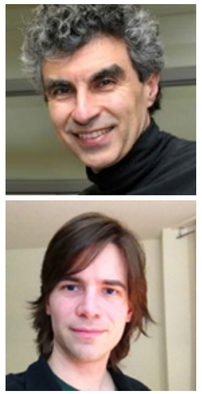


Eric Thibodeau-Laufer

Generative Stochastic Networks

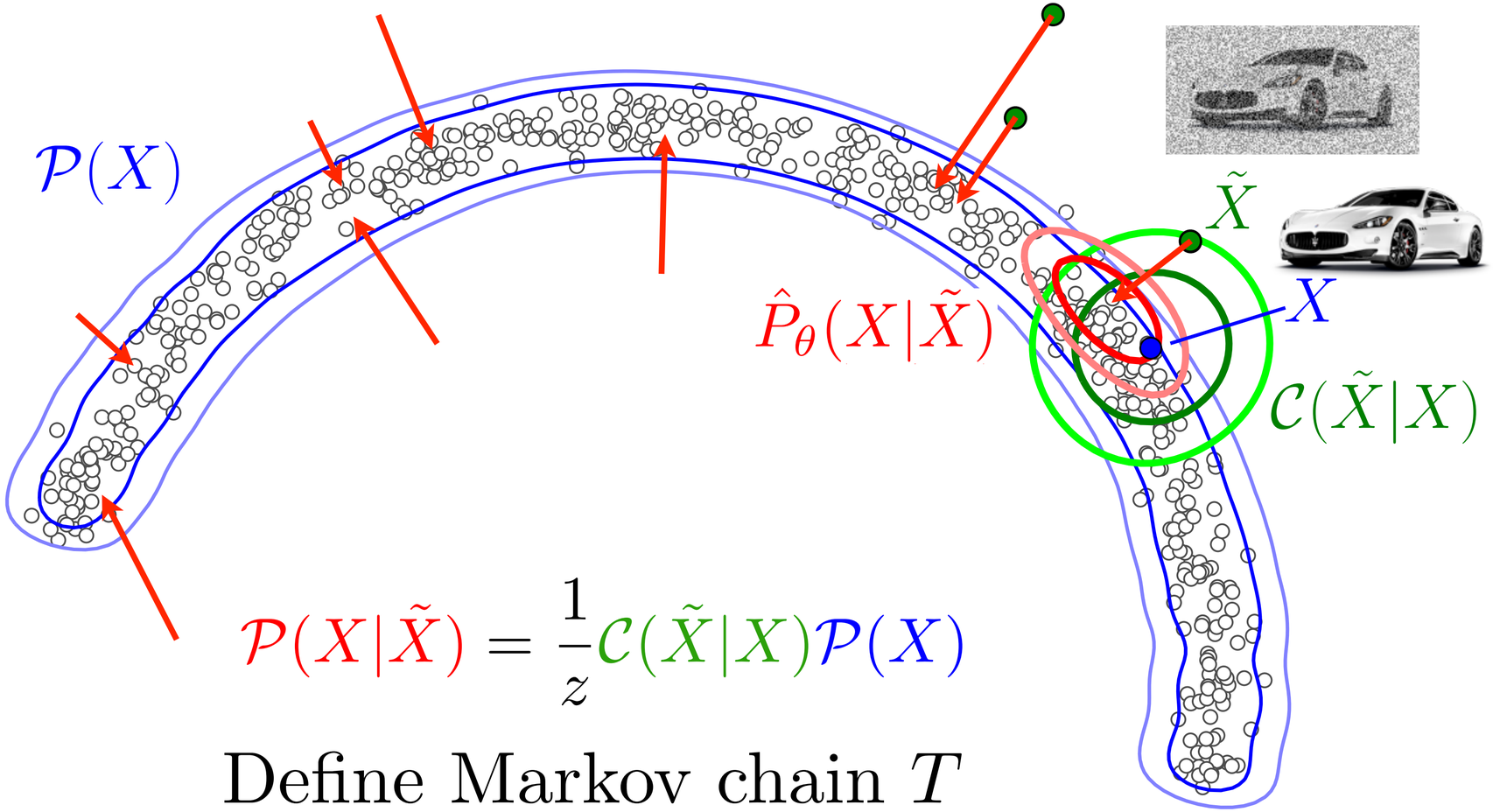


$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$



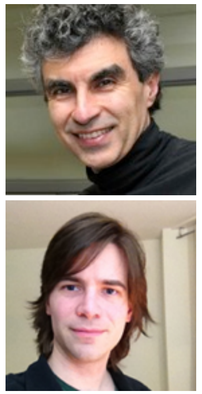
Eric Thibodeau-Laufer

Generative Stochastic Networks



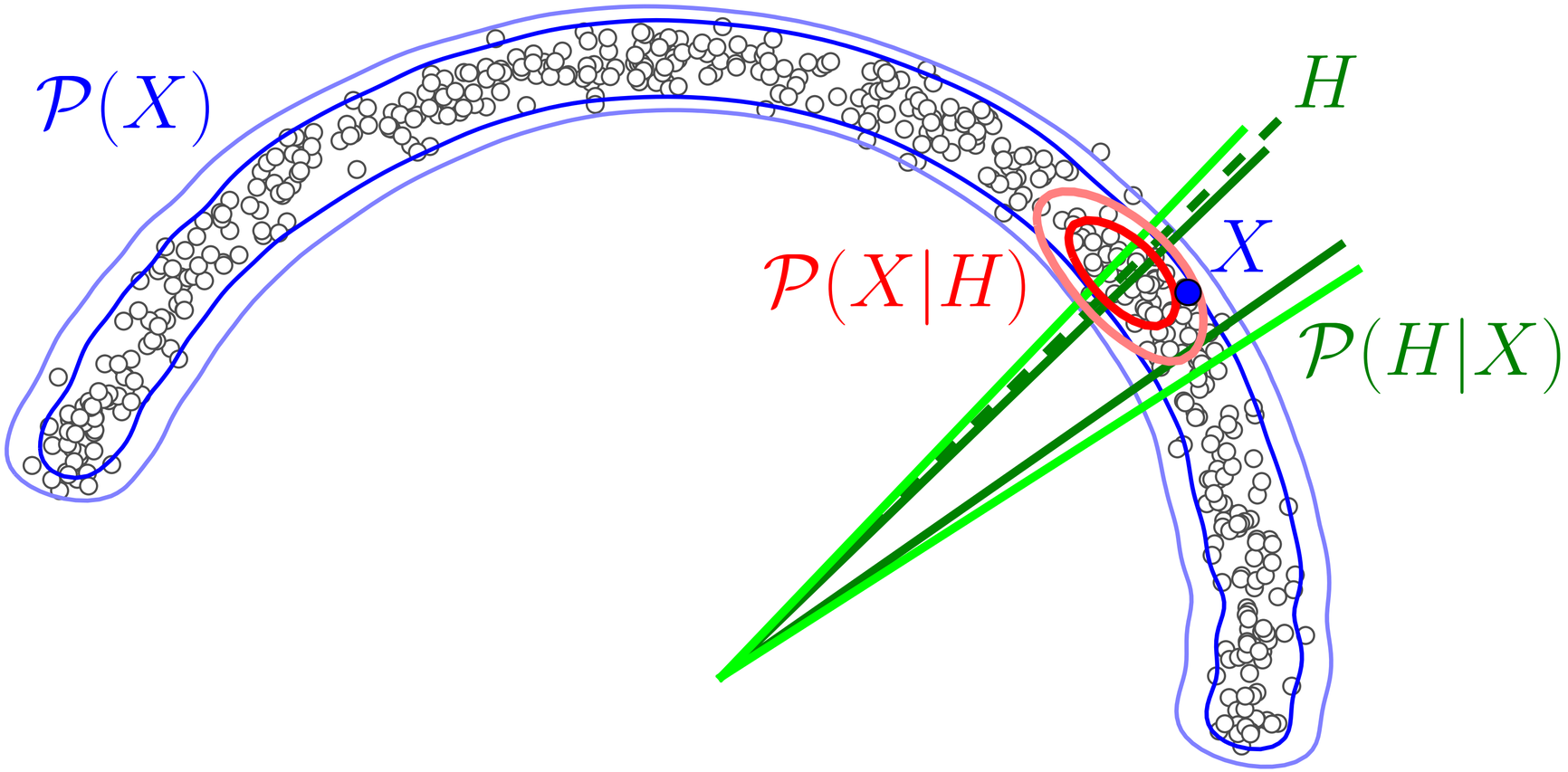
$$\mathcal{P}(X|\tilde{X}) = \frac{1}{z}\mathcal{C}(\tilde{X}|X)\mathcal{P}(X)$$

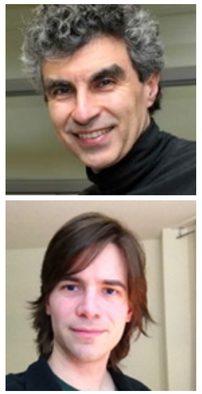
Define Markov chain T



Eric Thibodeau-Laufer

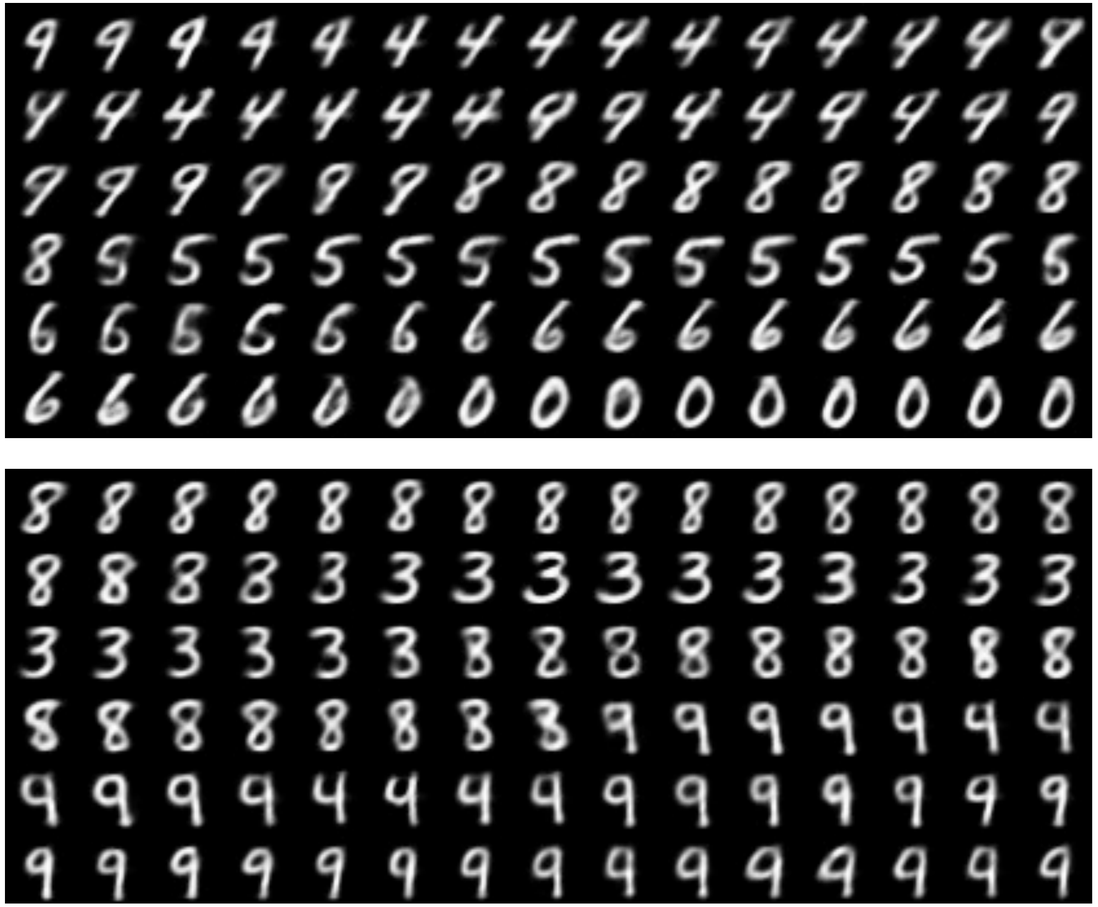
Generative Stochastic Networks





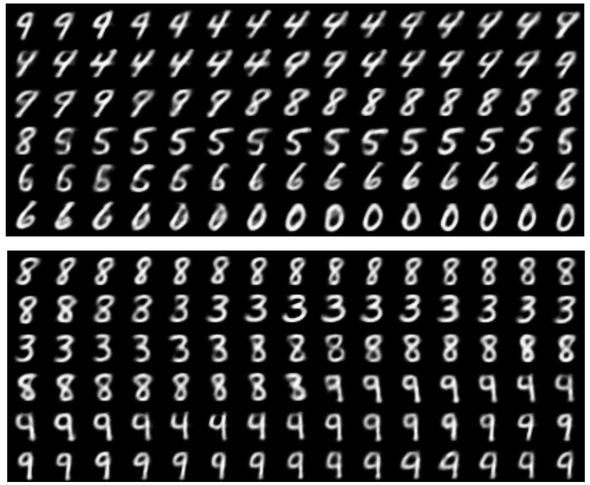
Eric Thibodeau-Laufer

Generative Stochastic Networks

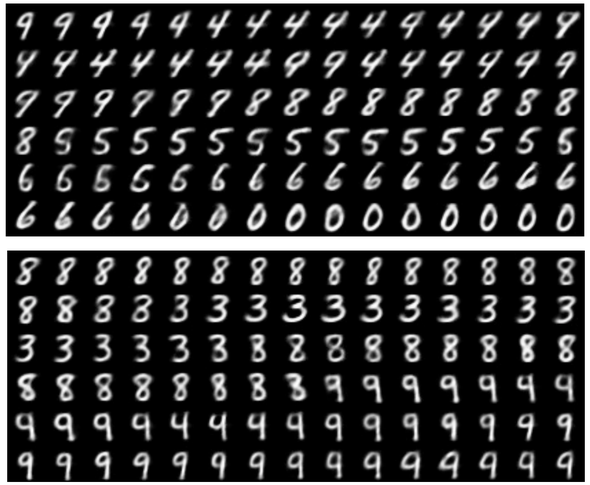


Learned distribution P(X)

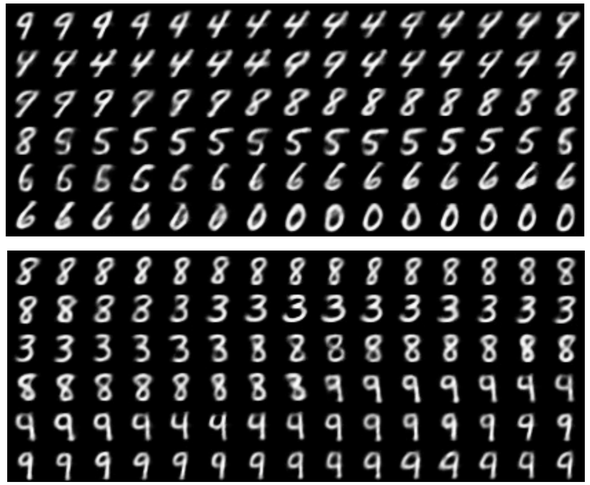
Generative Stochastic Networks



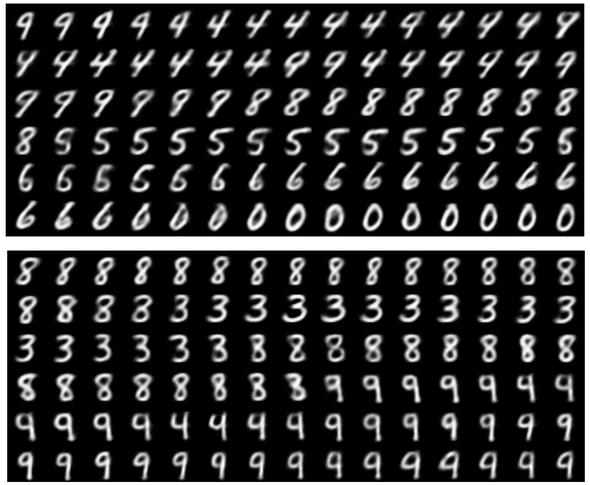
Generative Stochastic Networks

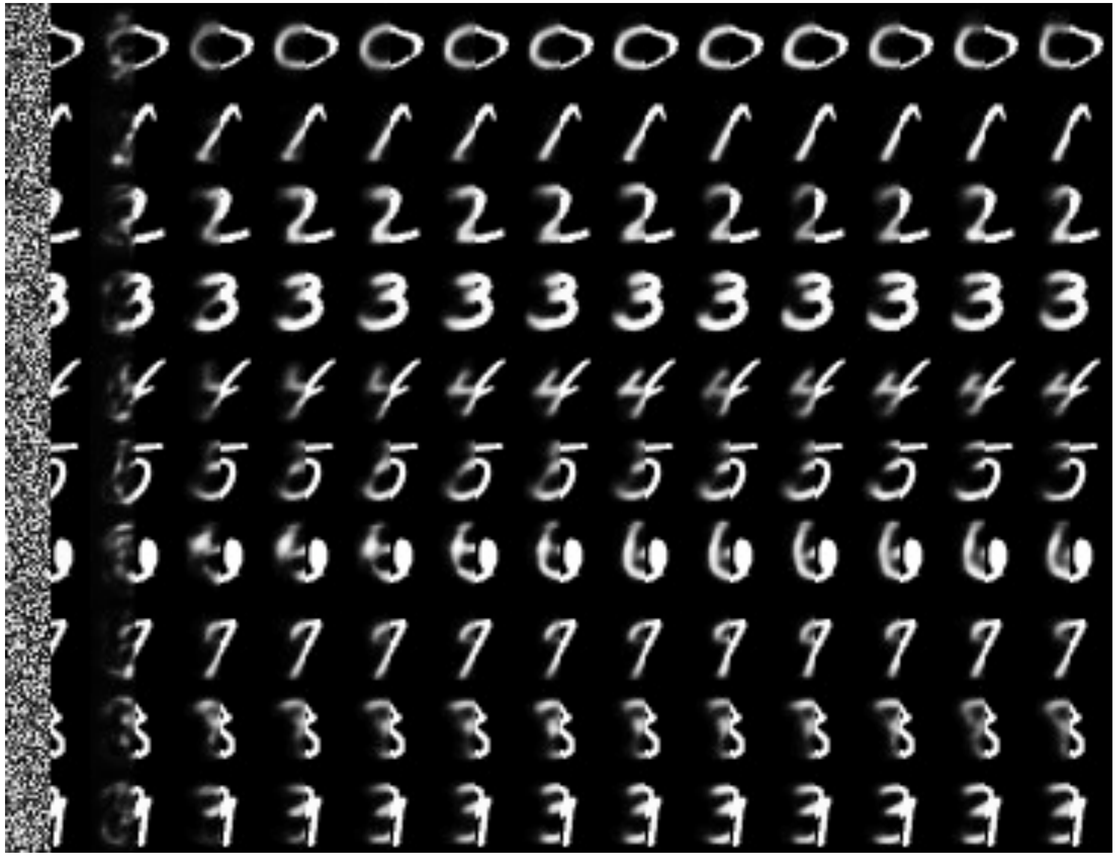


Generative Stochastic Networks



Generative Stochastic Networks

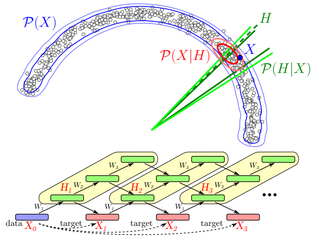




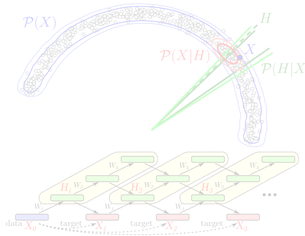
Generative Stochastic Networks

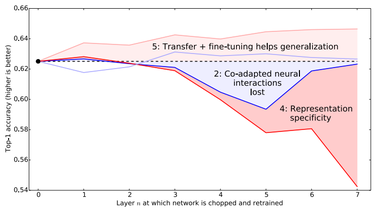
Generative Stochastic Networks

Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize



Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize





Great recent progress in Computer Vision

Great recent progress in Computer Vision

Better Modeling

Great recent progress in Computer Vision

Better Modeling

- Dropout

Great recent progress in Computer Vision

Better Modeling

- Dropout
- Rectified Linear units

Great recent progress in Computer Vision

Better Modeling

- Dropout
- Rectified Linear units
- Convolution and pooling

Great recent progress in Computer Vision

Better Modeling

- Dropout
- Rectified Linear units
- Convolution and pooling

Computation

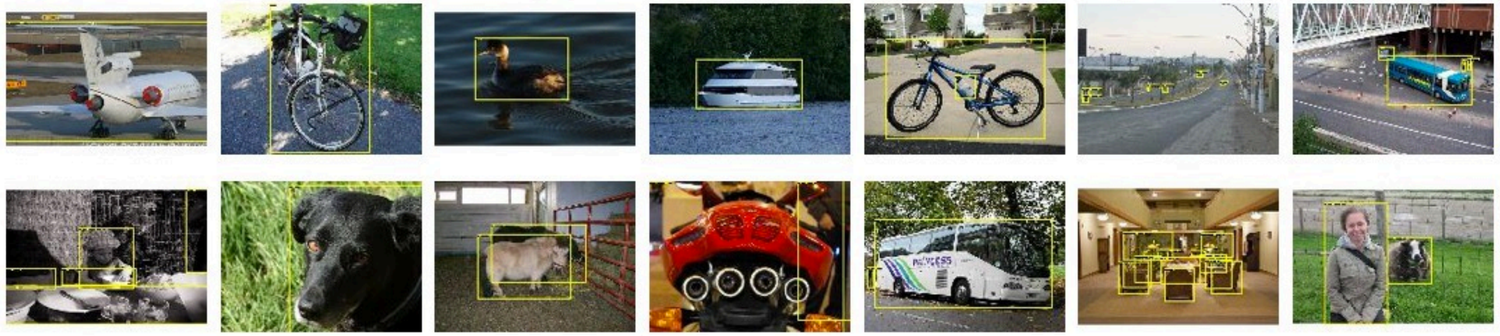
- GPUs

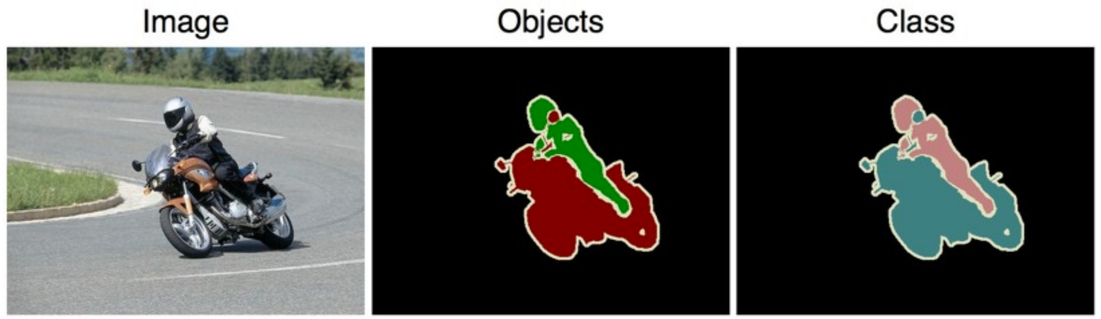
Great recent progress in Computer Vision

Better Modeling
- Dropout
- Rectified Linear units
- Convolution and pooling

Computation
- GPUs

Data
- Bigger datasets



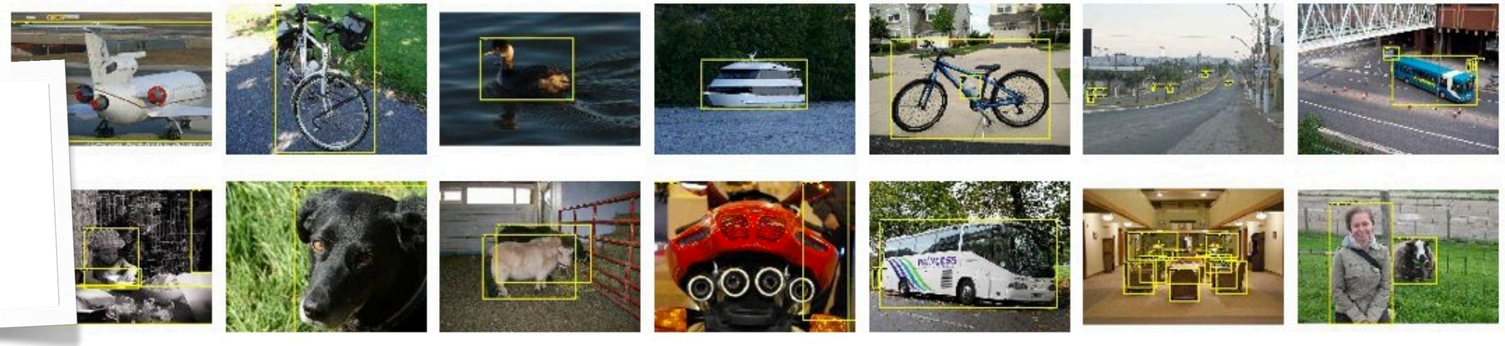


Pascal VOC
(11,500 images)

(not shown)

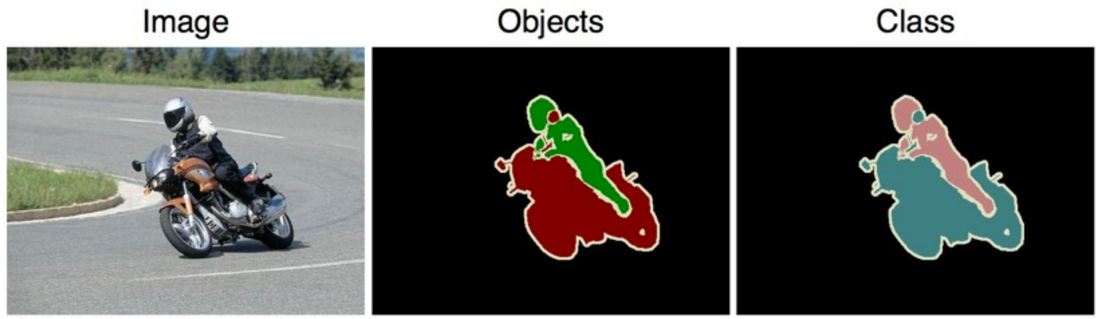
Caltech-101

(9,146 images)



Pascal VOC

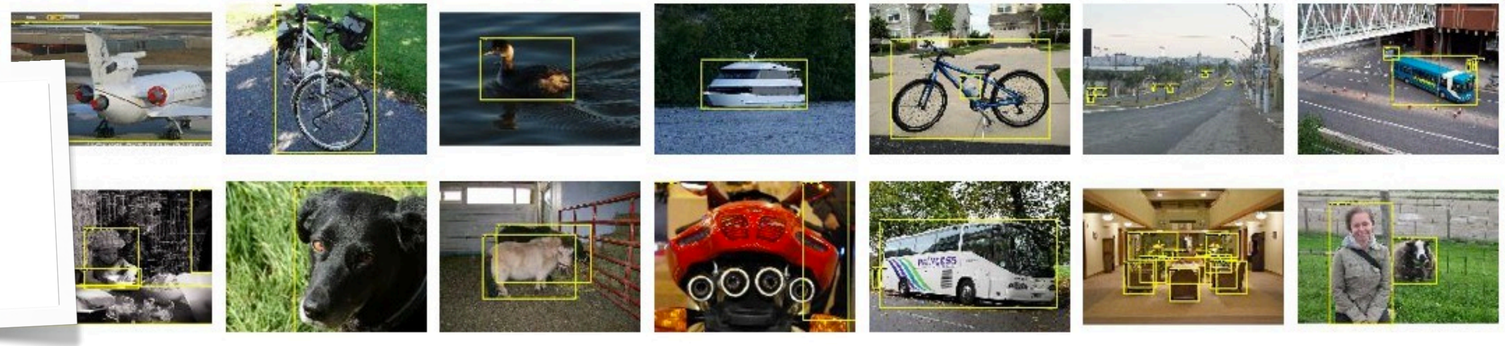
(11,500 images)



(not shown)

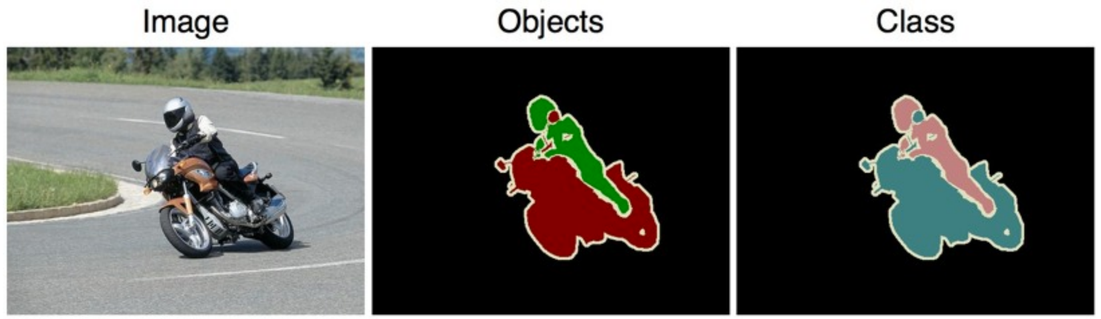
Caltech-101

(9,146 images)



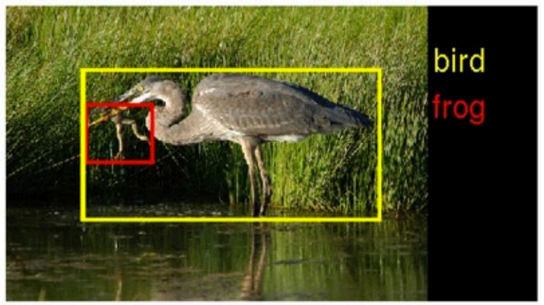
Pascal VOC

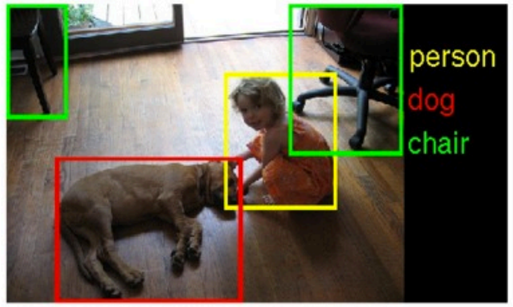
(11,500 images)

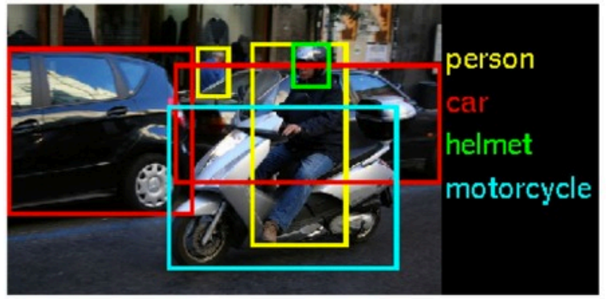


ImageNet

(1,300,000 images,
1000 classes)

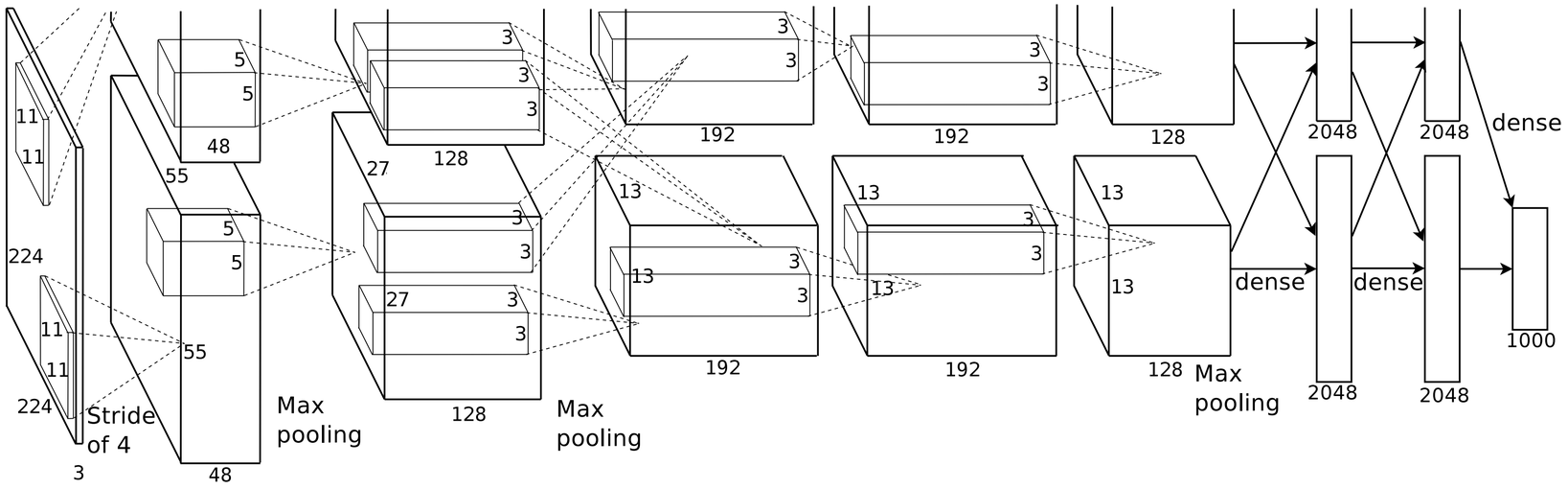




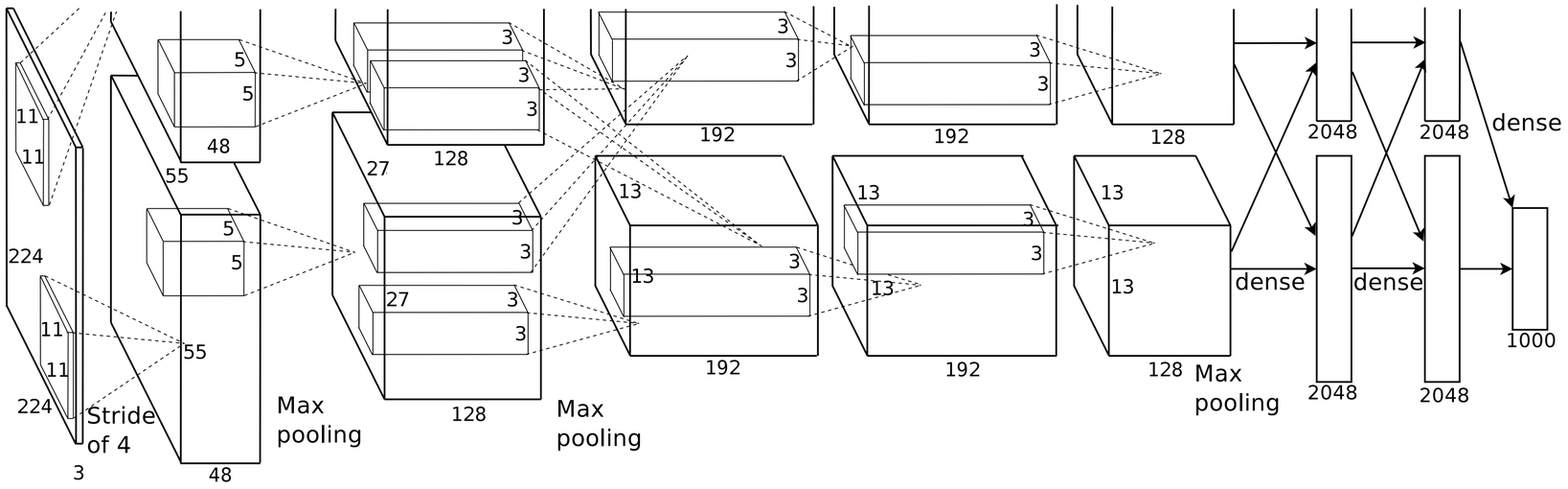


Deng et al., 2009

Krizhevsky, Sutskever, and Hinton, NIPS 2012.

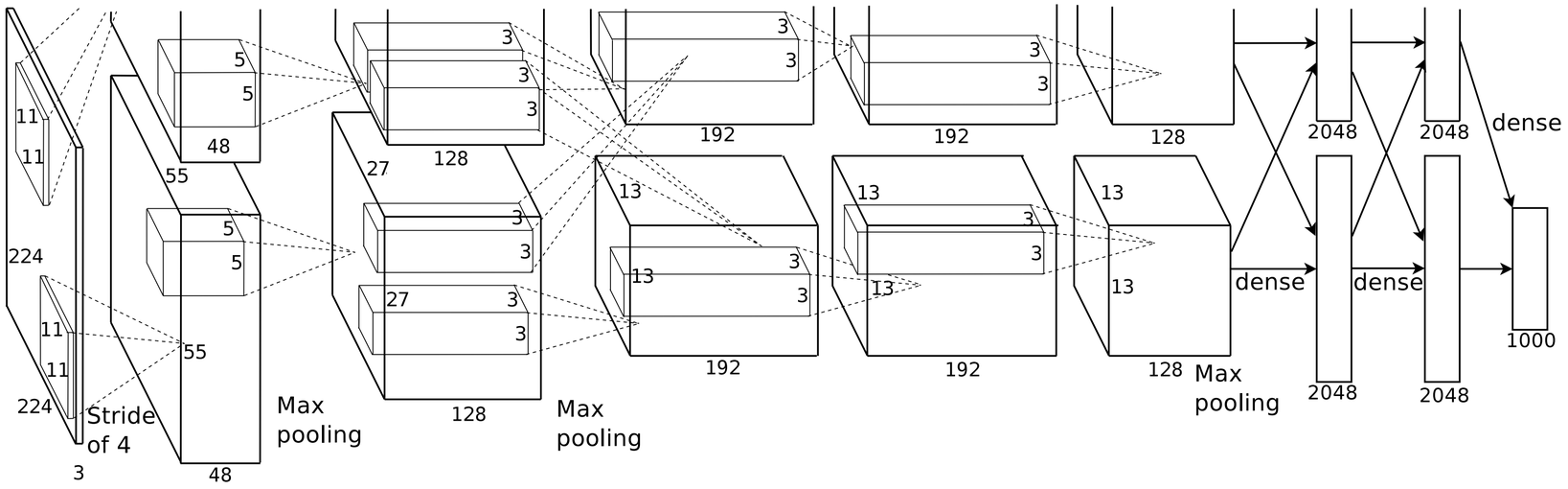


Krizhevsky, Sutskever, and Hinton, NIPS 2012.



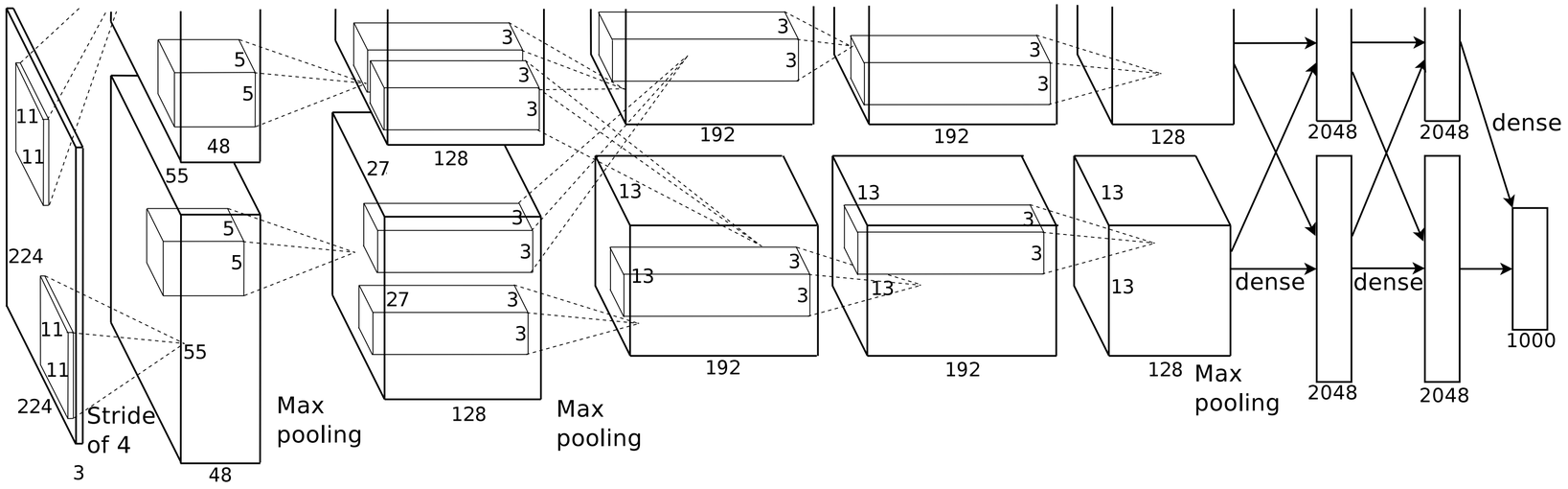
	Top-5 val err
Previous SotA	26%

Krizhevsky, Sutskever, and Hinton, NIPS 2012.



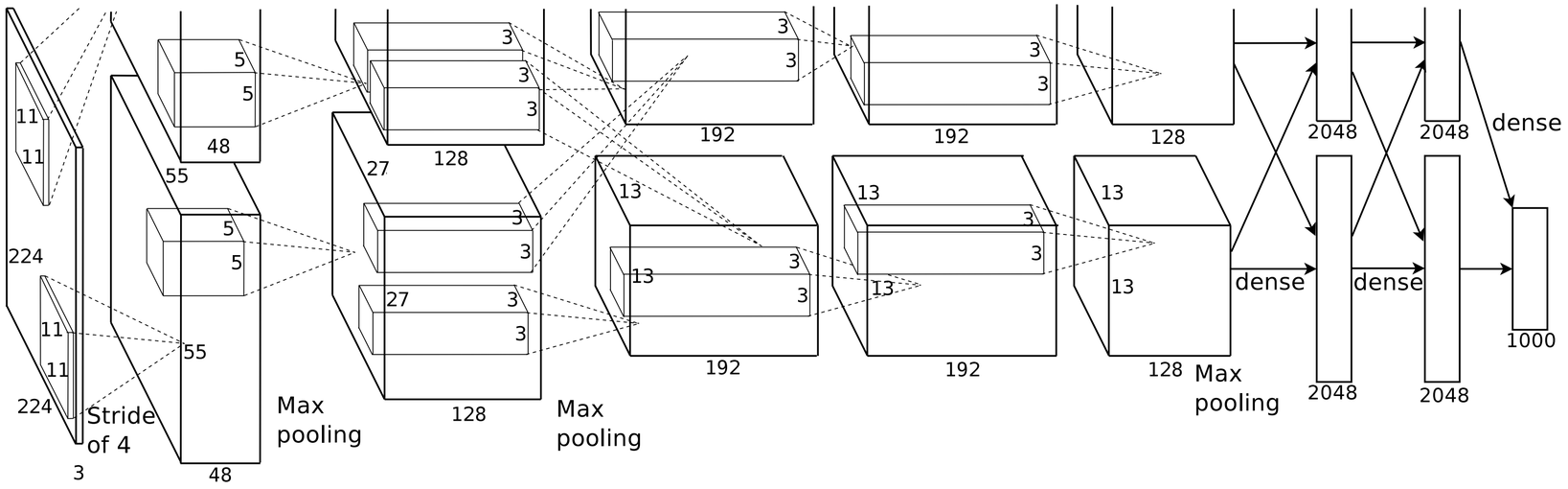
	Top-5 val err
Previous SotA	26%
Krizhevsky et al. (2012)	15%

Krizhevsky, Sutskever, and Hinton, NIPS 2012.



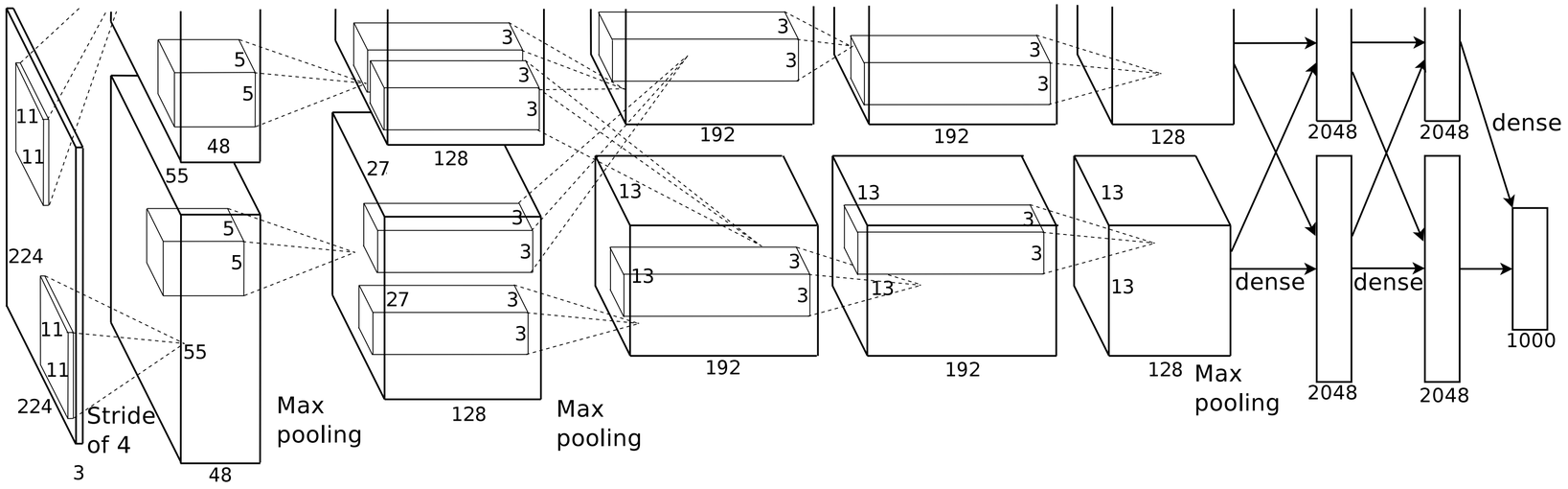
	Top-5 val err
Previous SotA	26%
Krizhevsky et al. (2012)	15%
GoogLe Net (2014)	6.7%

Krizhevsky, Sutskever, and Hinton, NIPS 2012.



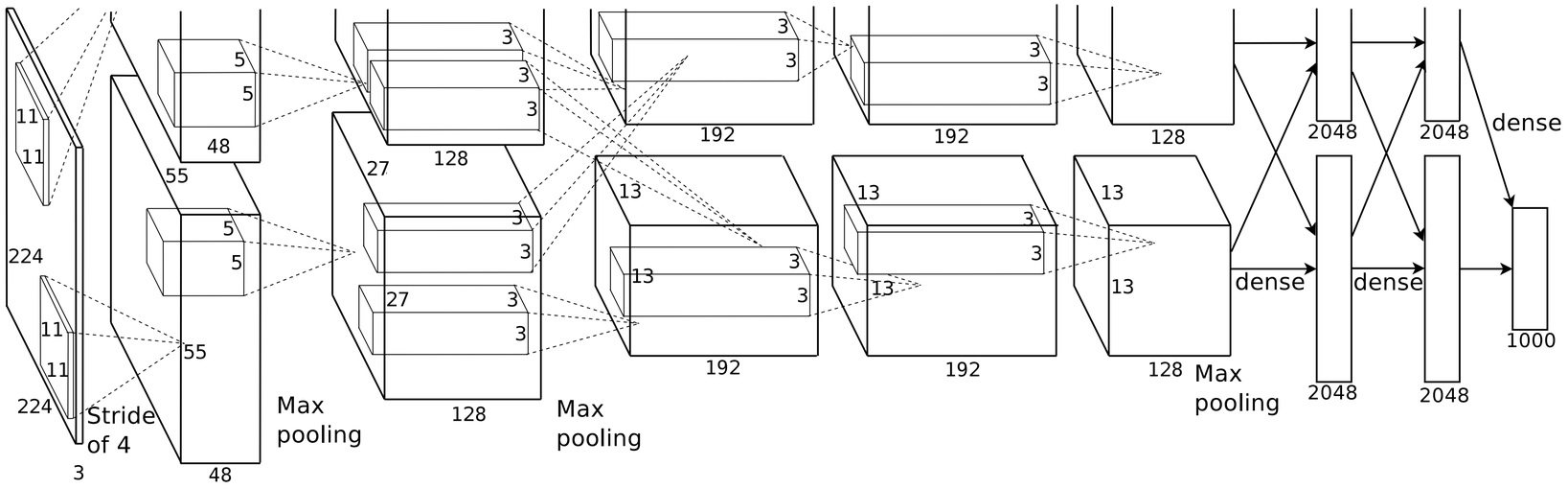
	Top-5 val err
Previous SotA	26%
Krizhevsky et al. (2012)	15%
GoogLe Net (2014)	6.7%
Human Karpathy/Russakovsky (2014)	5.1%

Krizhevsky, Sutskever, and Hinton, NIPS 2012.



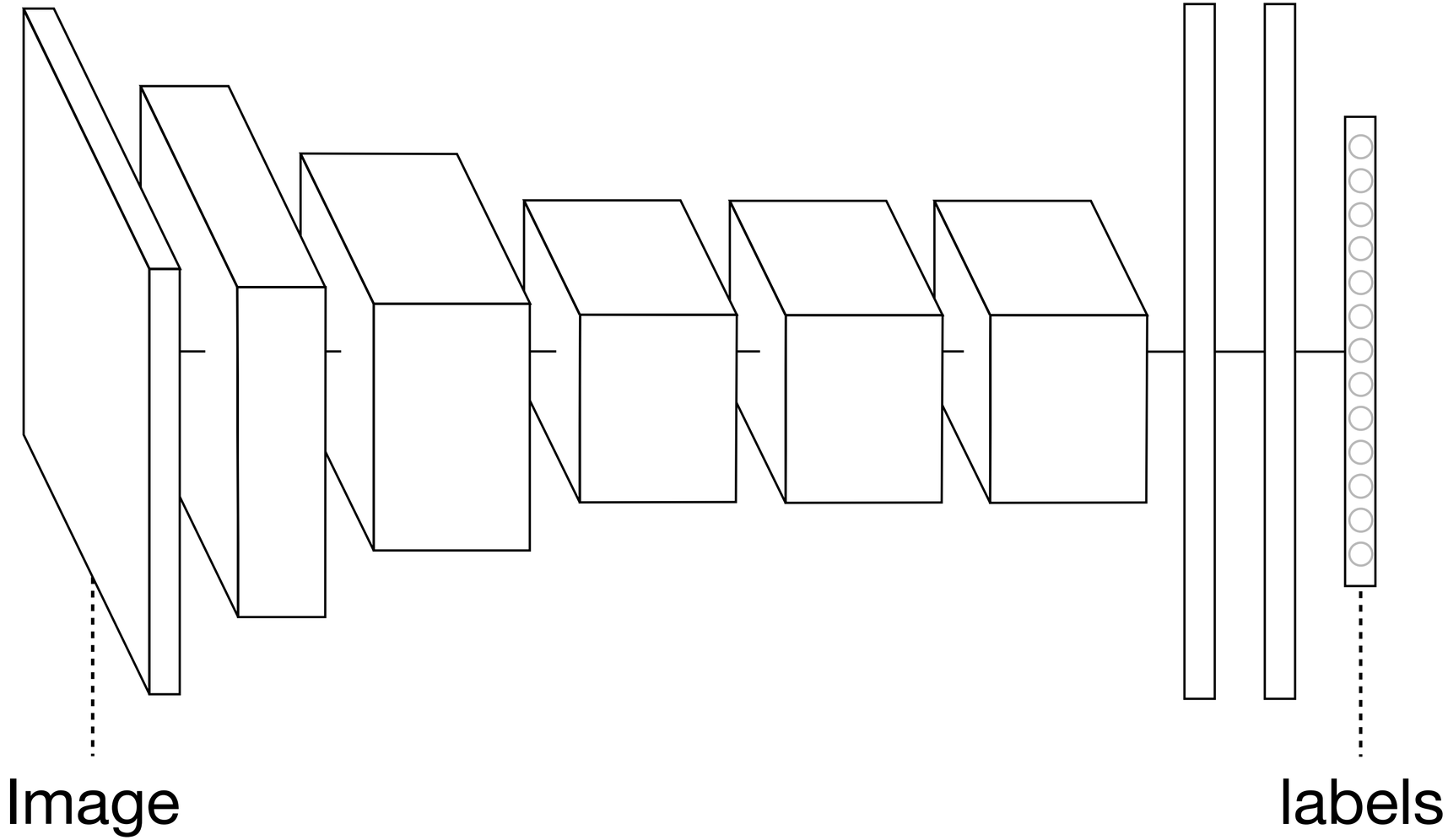
	Top-5 val err
Previous SotA	26%
Krizhevsky et al. (2012)	15%
GoogLe Net (2014)	6.7%
Human Karpathy/Russakovsky (2014)	5.1%
MSR (2015 Feb 6)	4.94%

Krizhevsky, Sutskever, and Hinton, NIPS 2012.



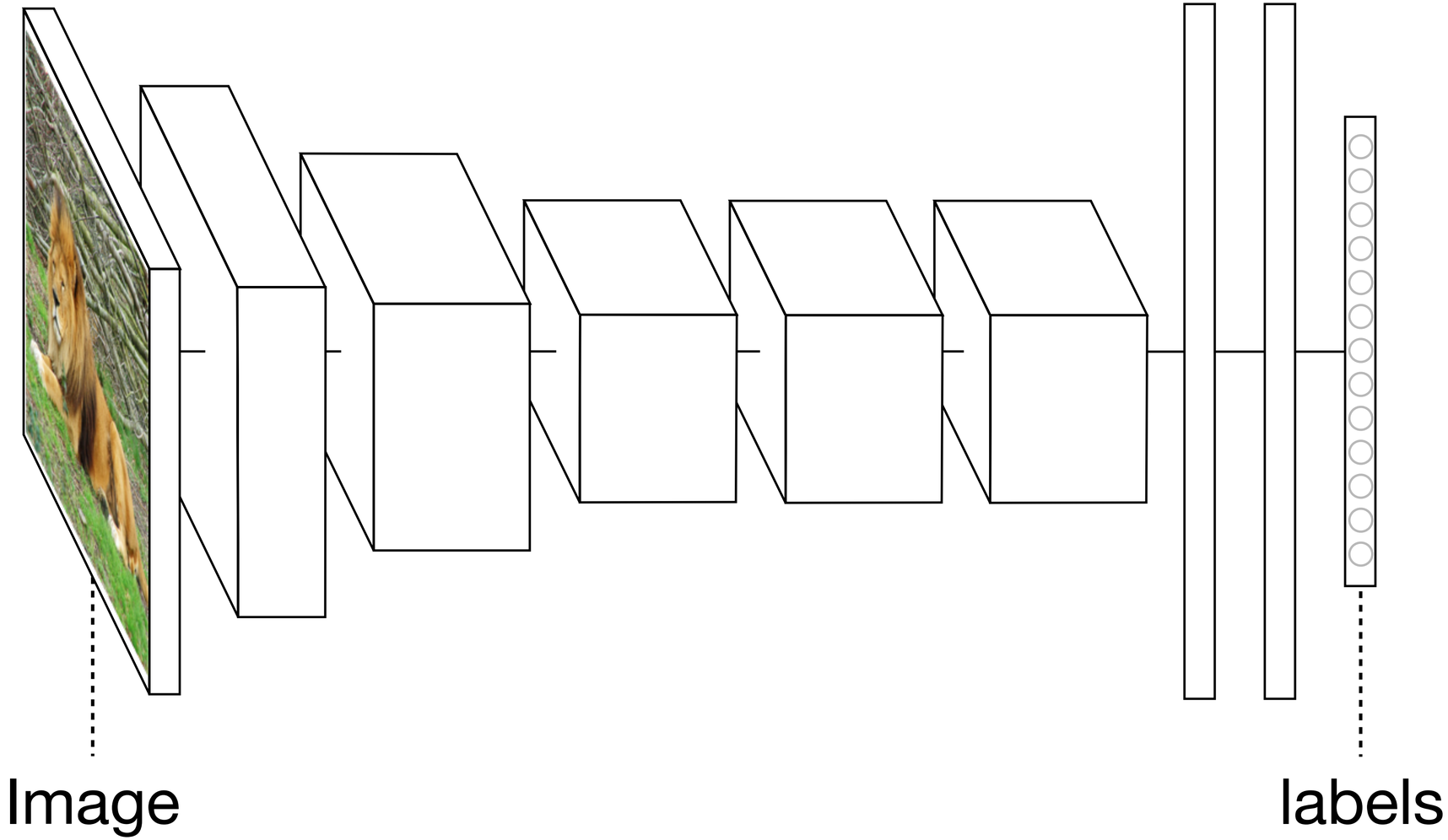
	Top-5 val err
Previous SotA	26%
Krizhevsky et al. (2012)	15%
GoogLe Net (2014)	6.7%
Human Karpathy/Russakovsky (2014)	5.1%
MSR (2015 Feb 6)	4.94%
Google (2015 Feb 11)	4.9%

AlexNet



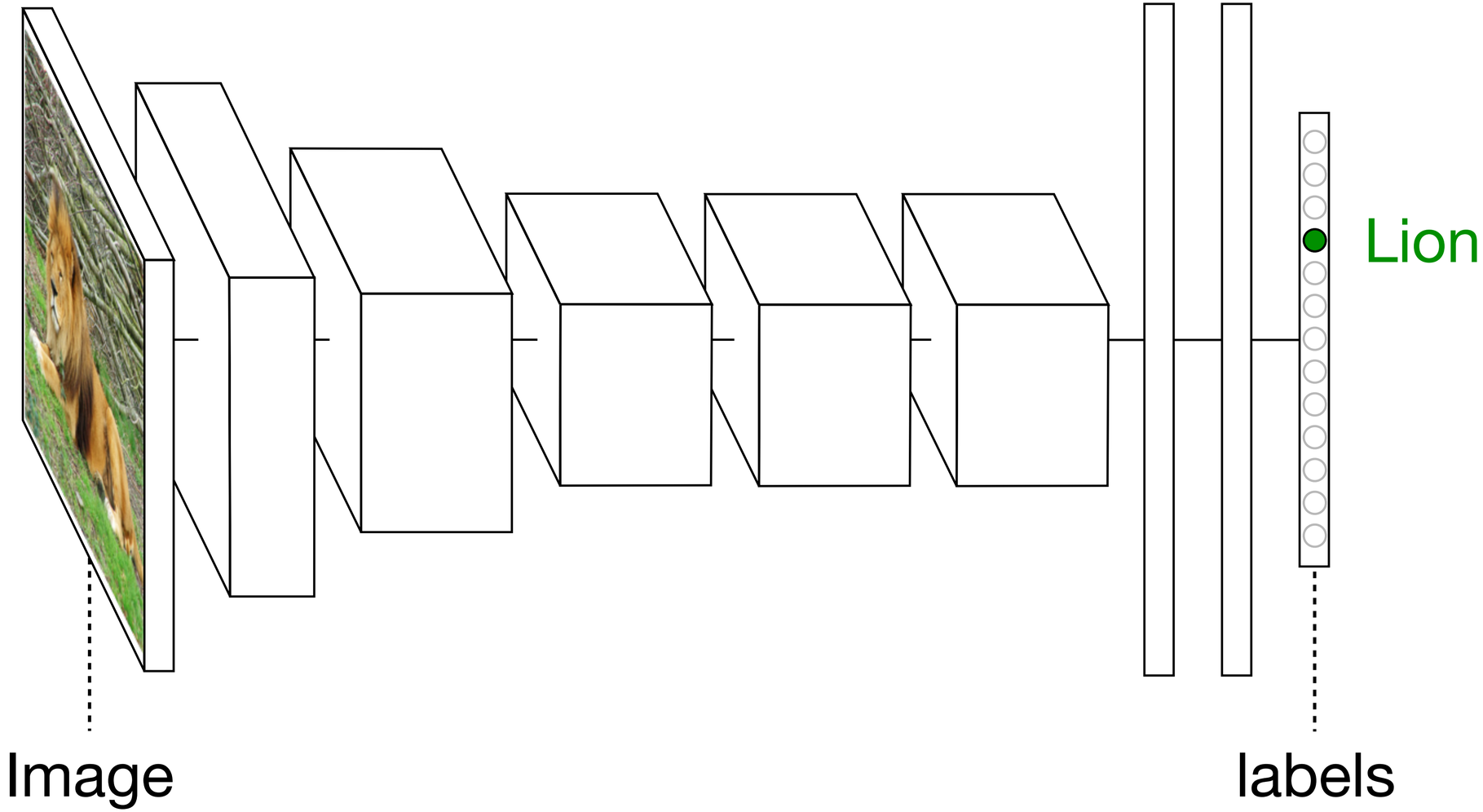
Krizhevsky, Sutskever, Hinton — NIPS 2012

AlexNet



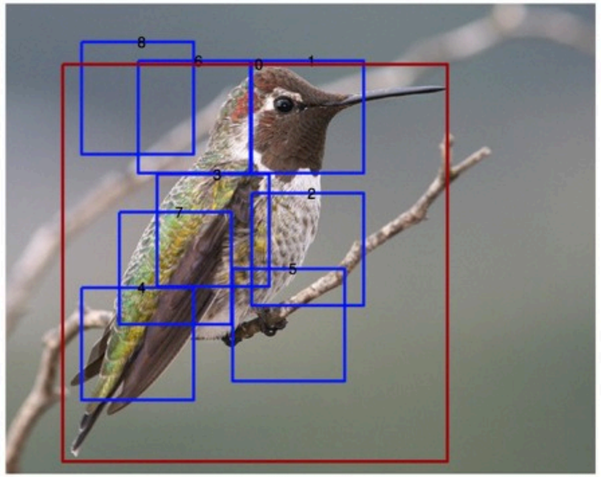
Krizhevsky, Sutskever, Hinton — NIPS 2012

AlexNet

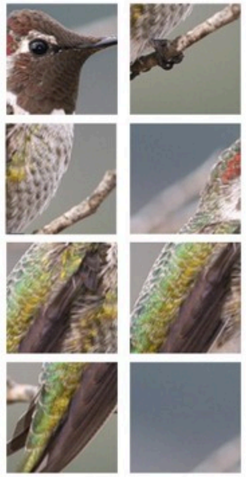


Krizhevsky, Sutskever, Hinton — NIPS 2012

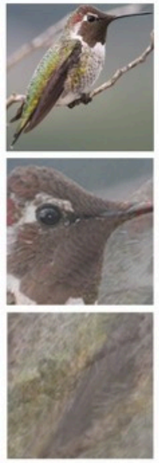
DeCAF Donahue, Jia, et al. arXiv 2013



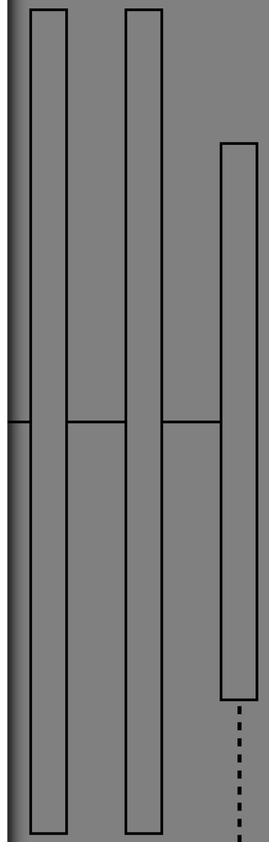
(a) DPM detections



(b) Parts



(c) DPD

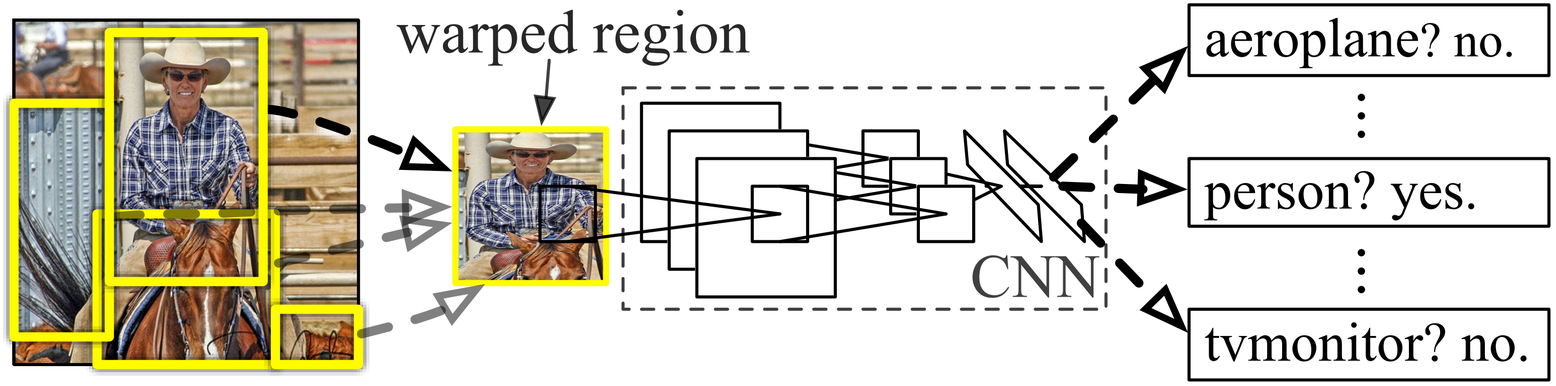


Image

labels

R-CNN

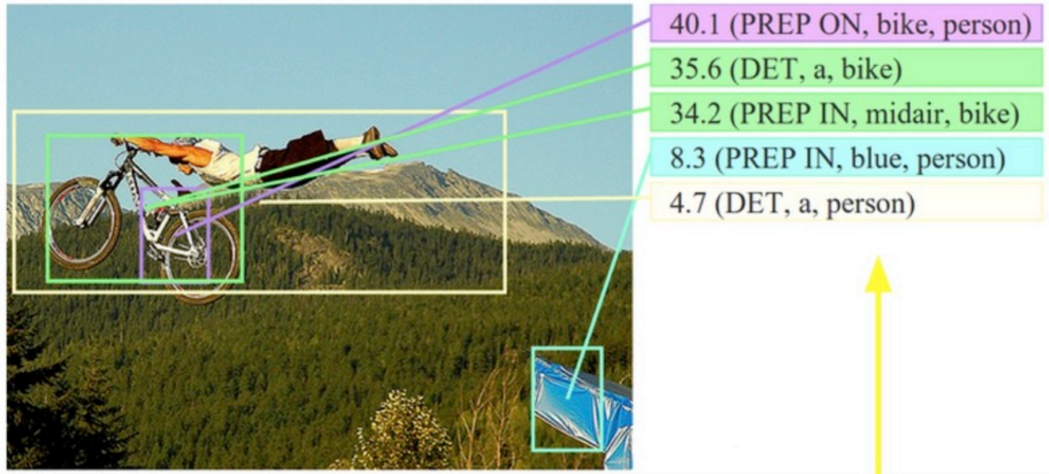
Girshick et al., CVPR 2014



labels

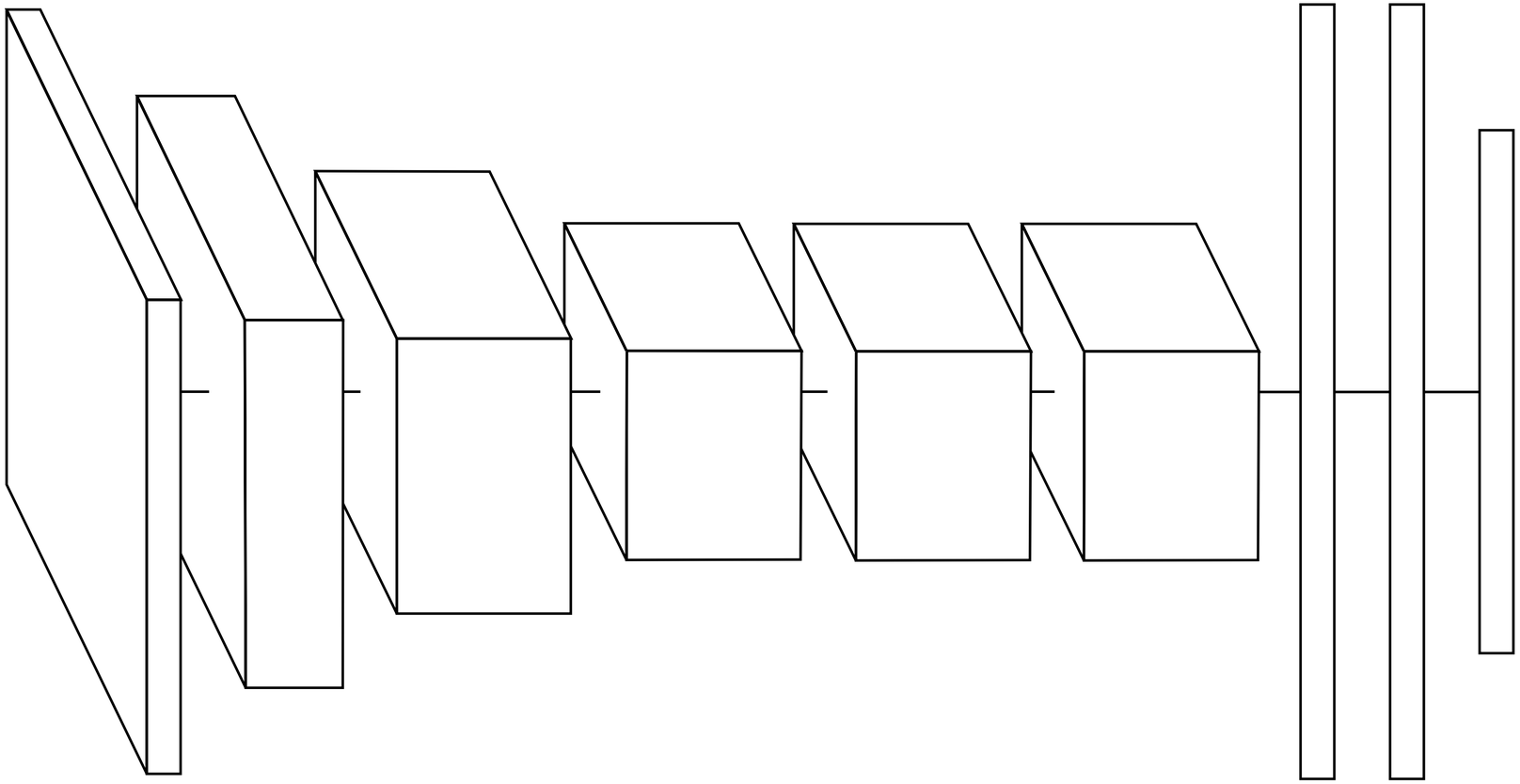
Deep Fragment Embeddings

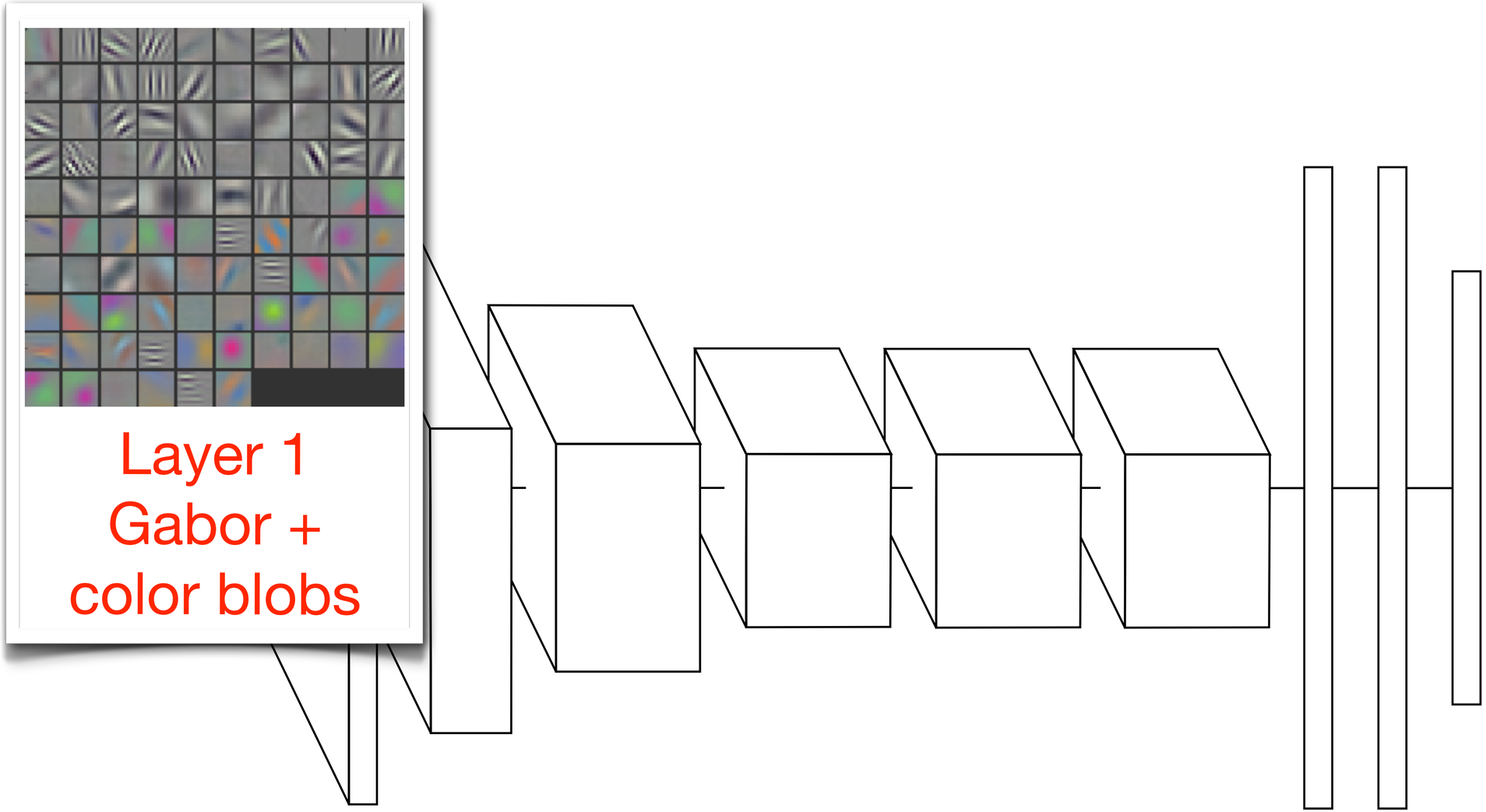
Karpathy et al. NIPS 2014

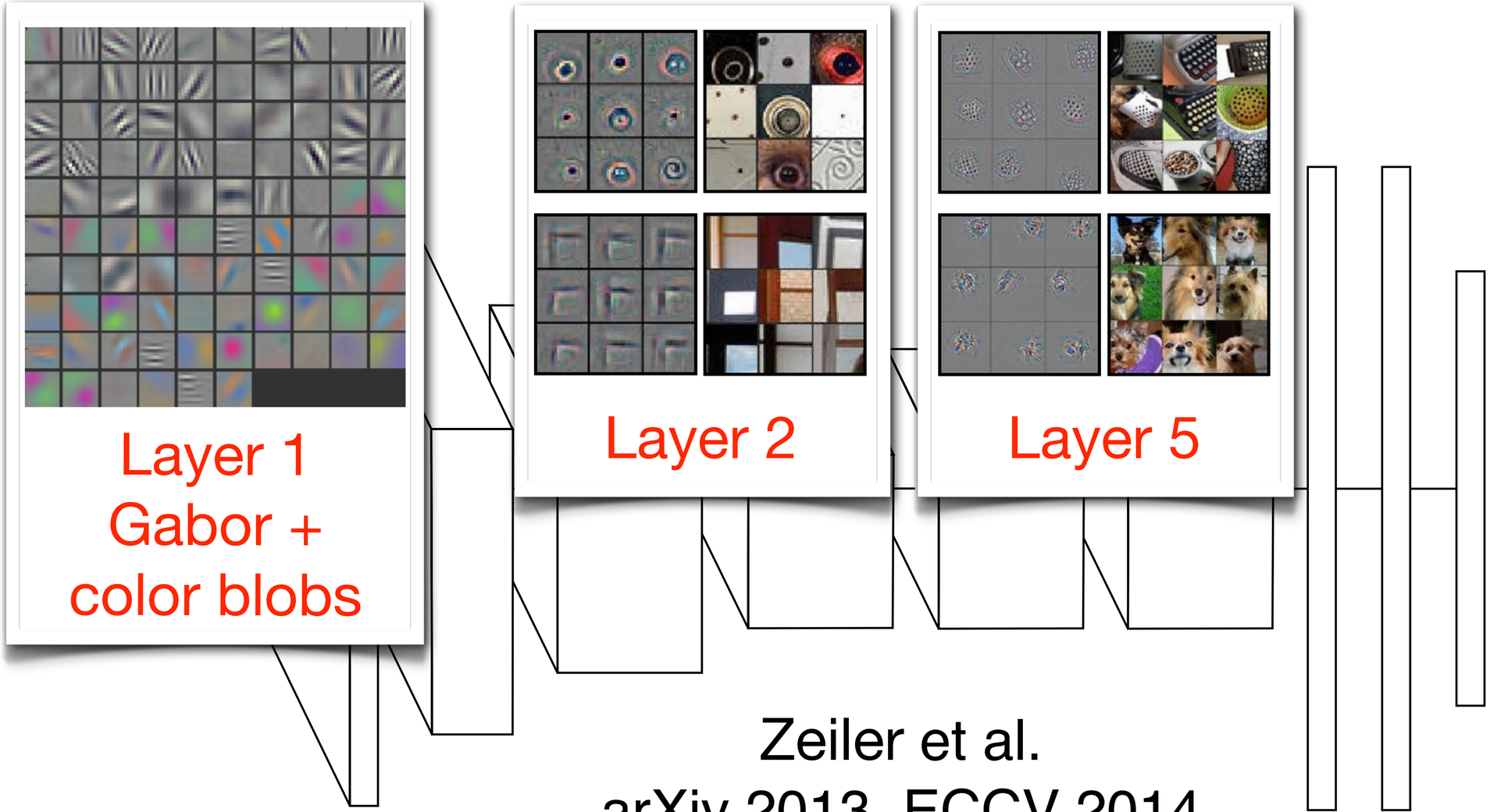


1. A person in blue on a bike in midair
2. Man on a dirt bike
3. A dirt biker flies through the air
4. A person on a dirt bike soaring through the air sideways
5. A dirt biker leaps through the air

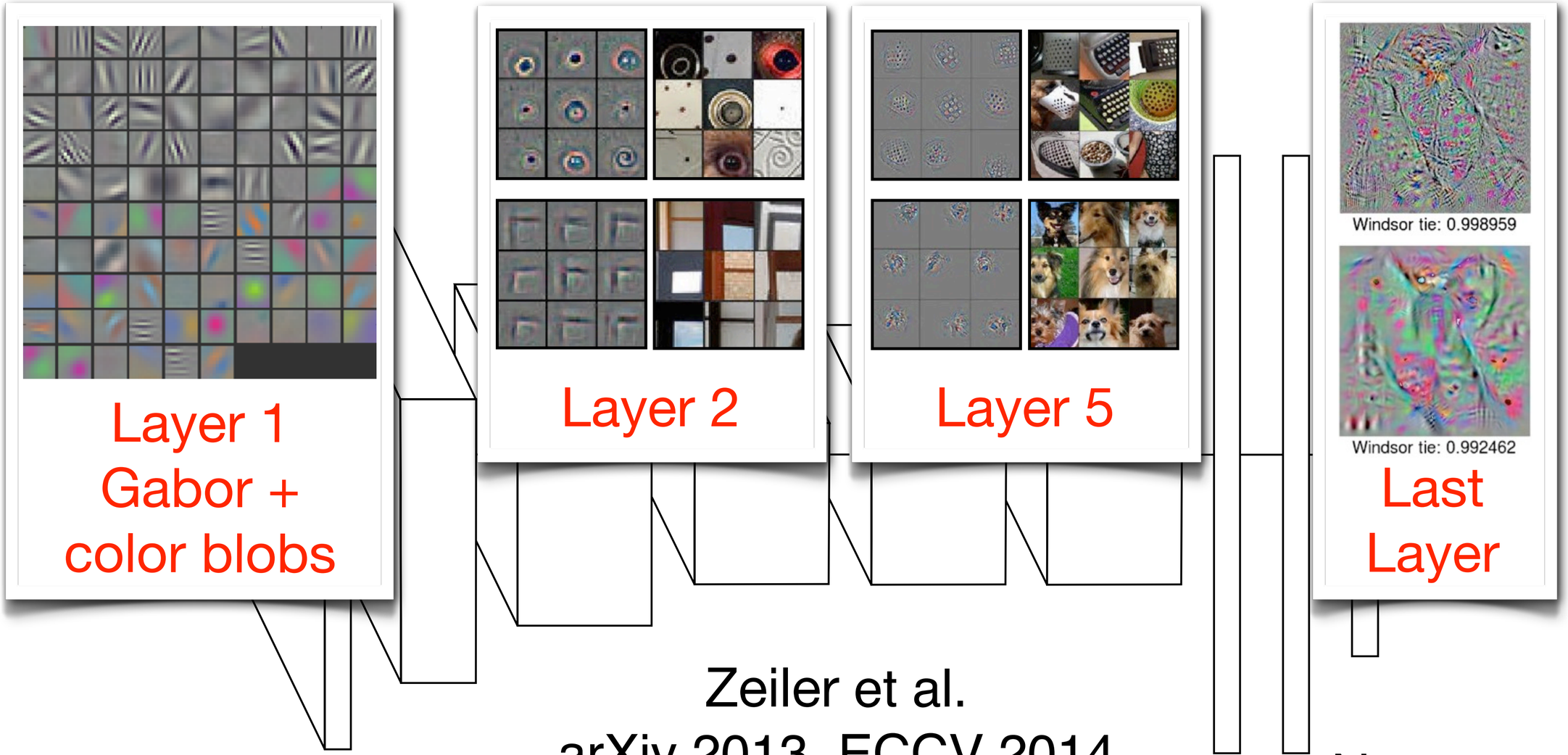
bels




Layer 1
Gabor +
color blobs

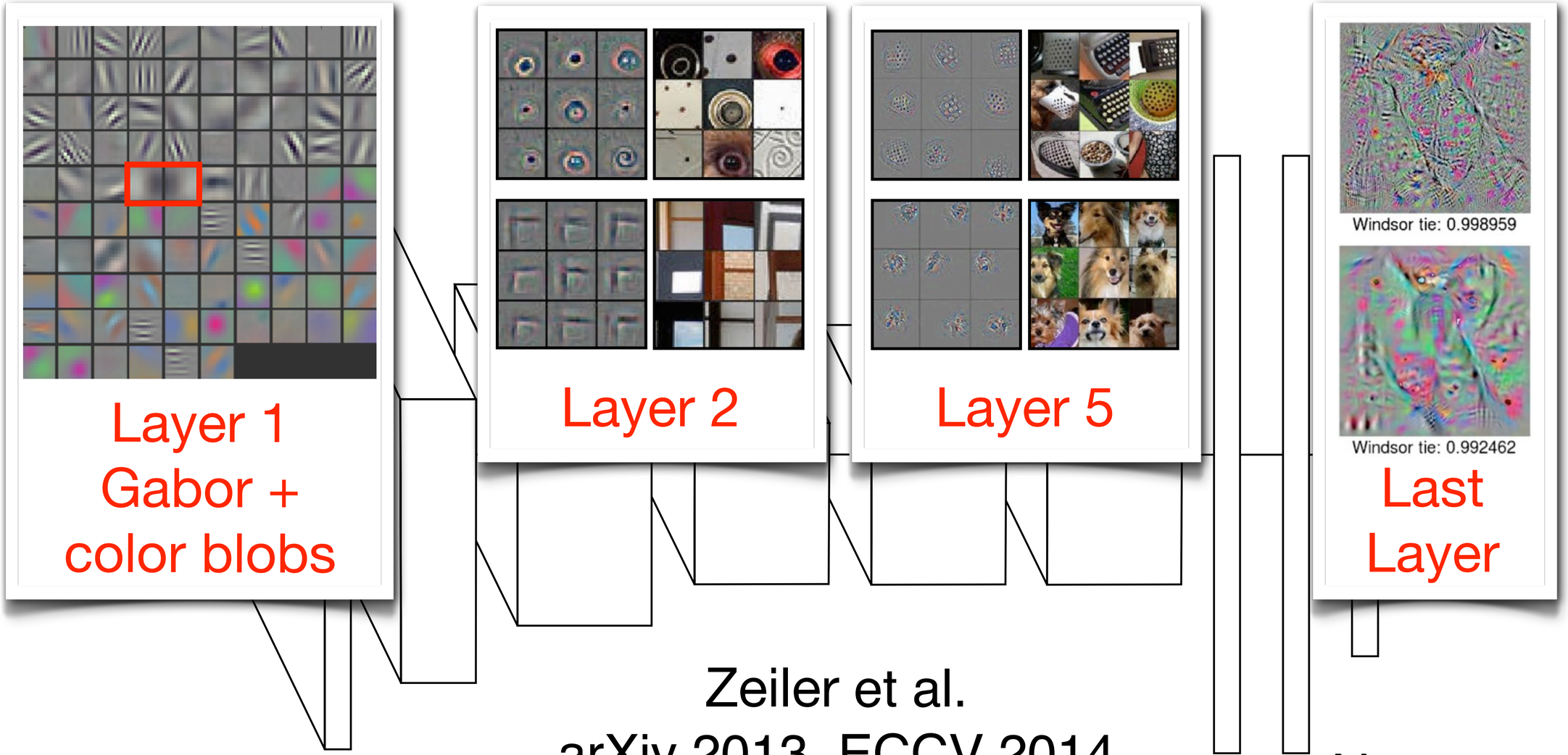


Zeiler et al.
arXiv 2013, ECCV 2014



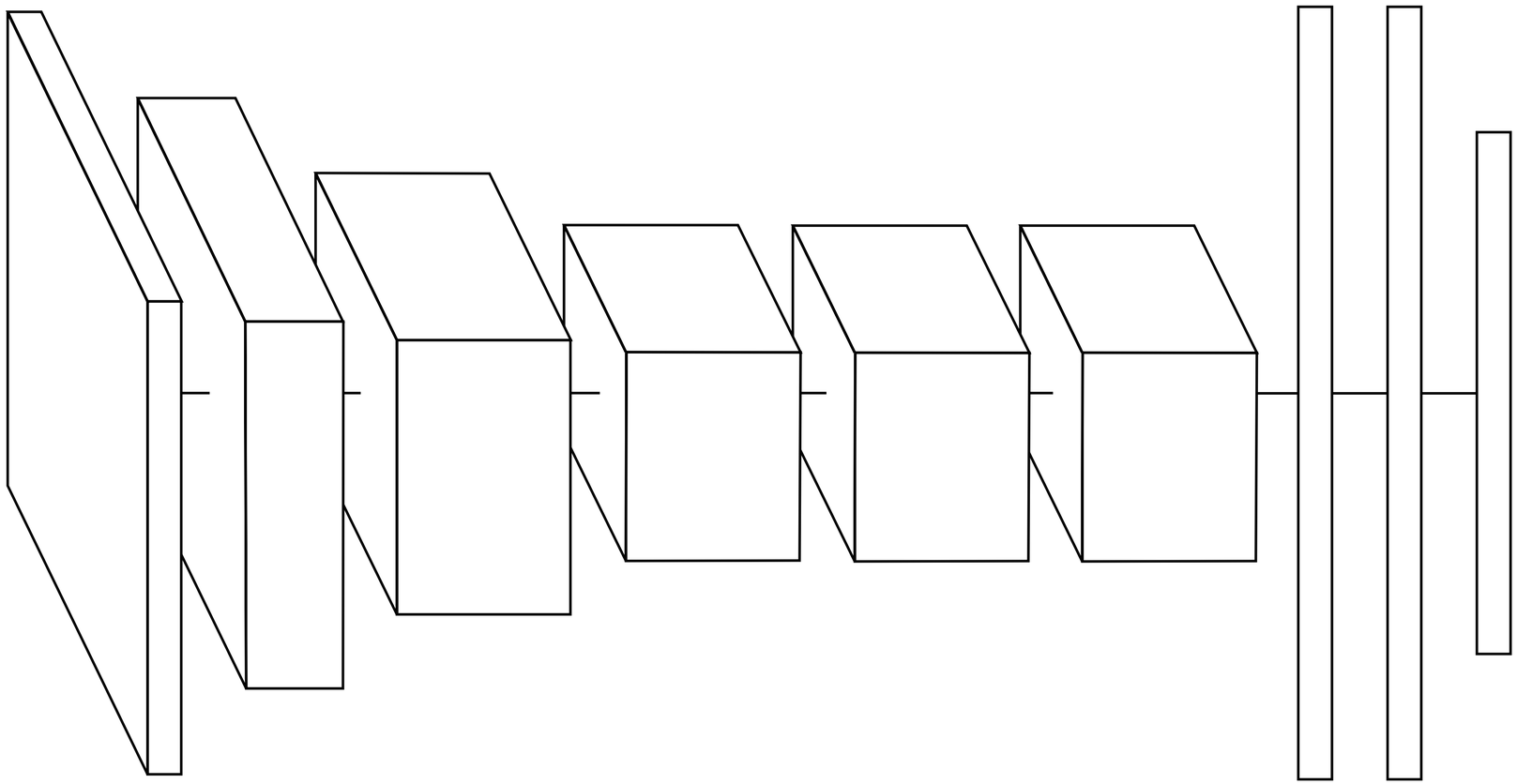
Zeiler et al.
arXiv 2013, ECCV 2014

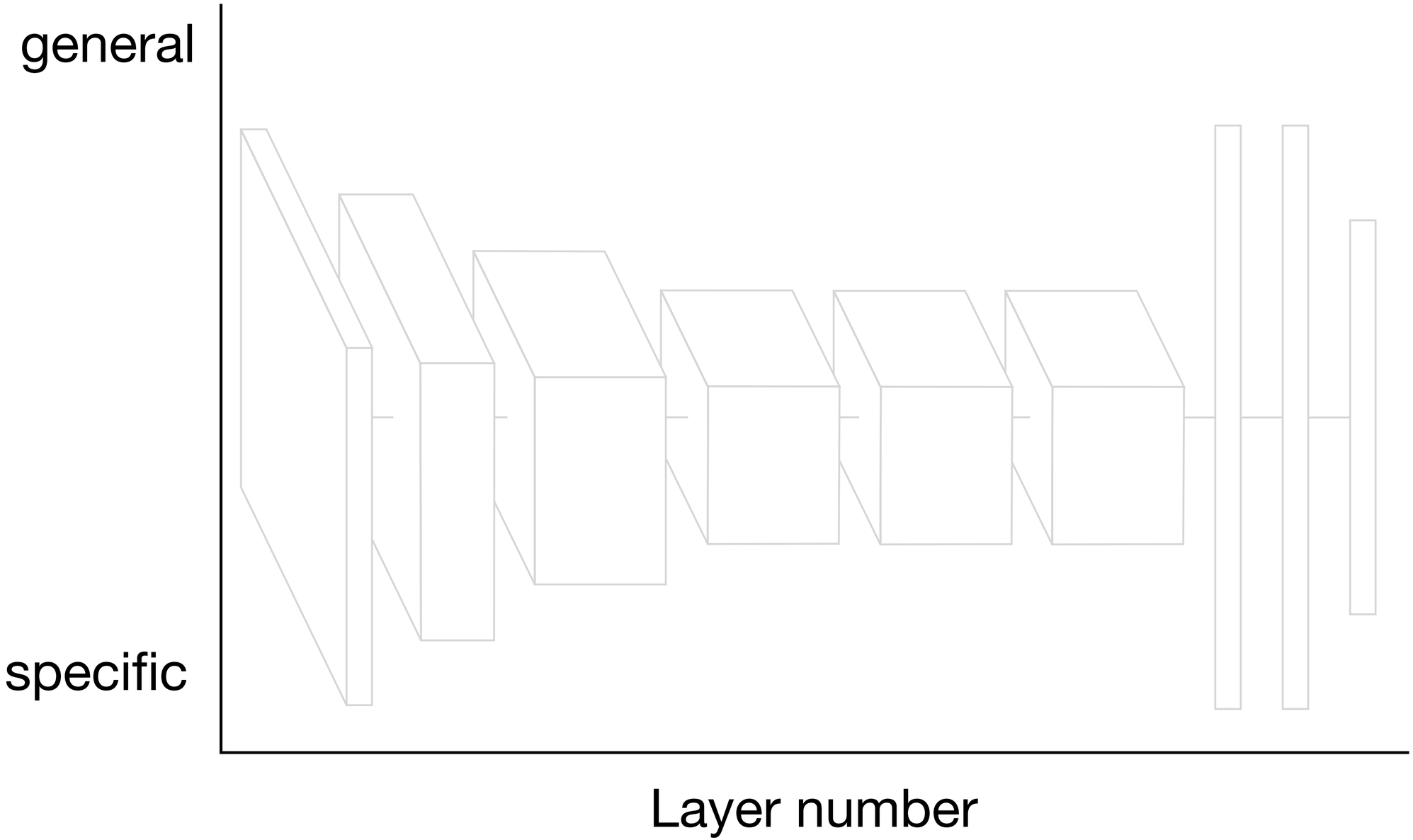
Nguyen,
Yosinski, Clune,
arXiv 2014

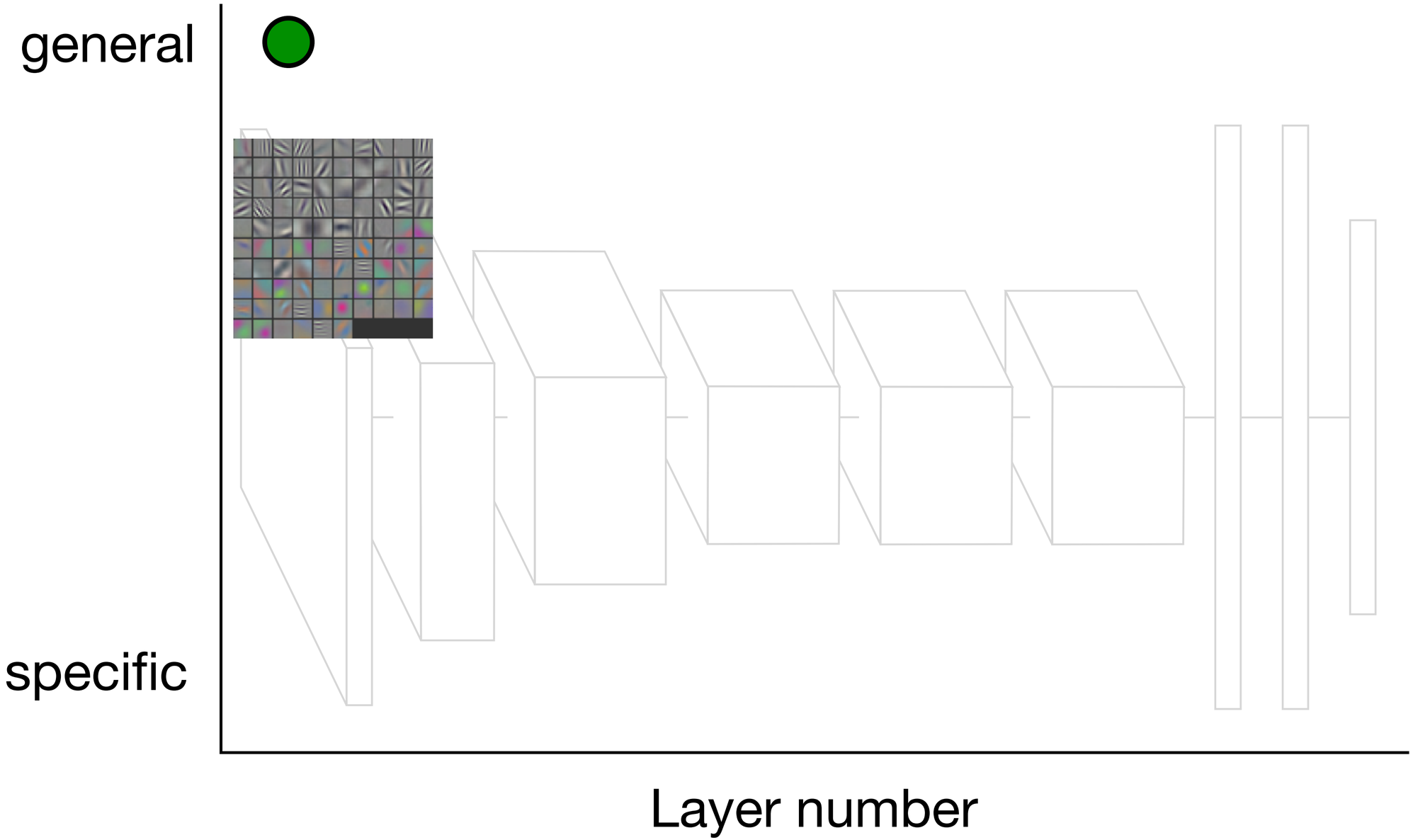


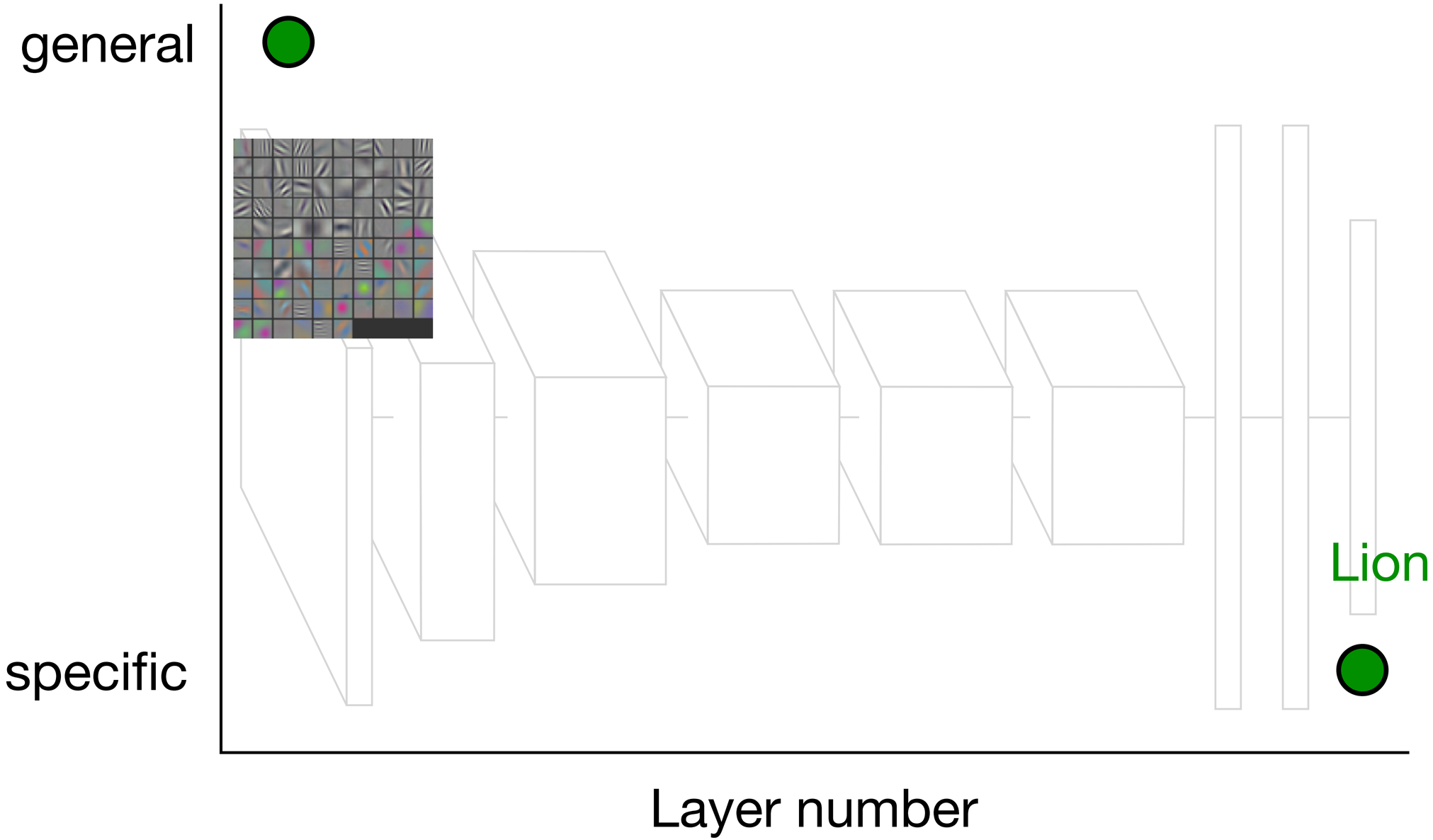
Zeiler et al.
arXiv 2013, ECCV 2014

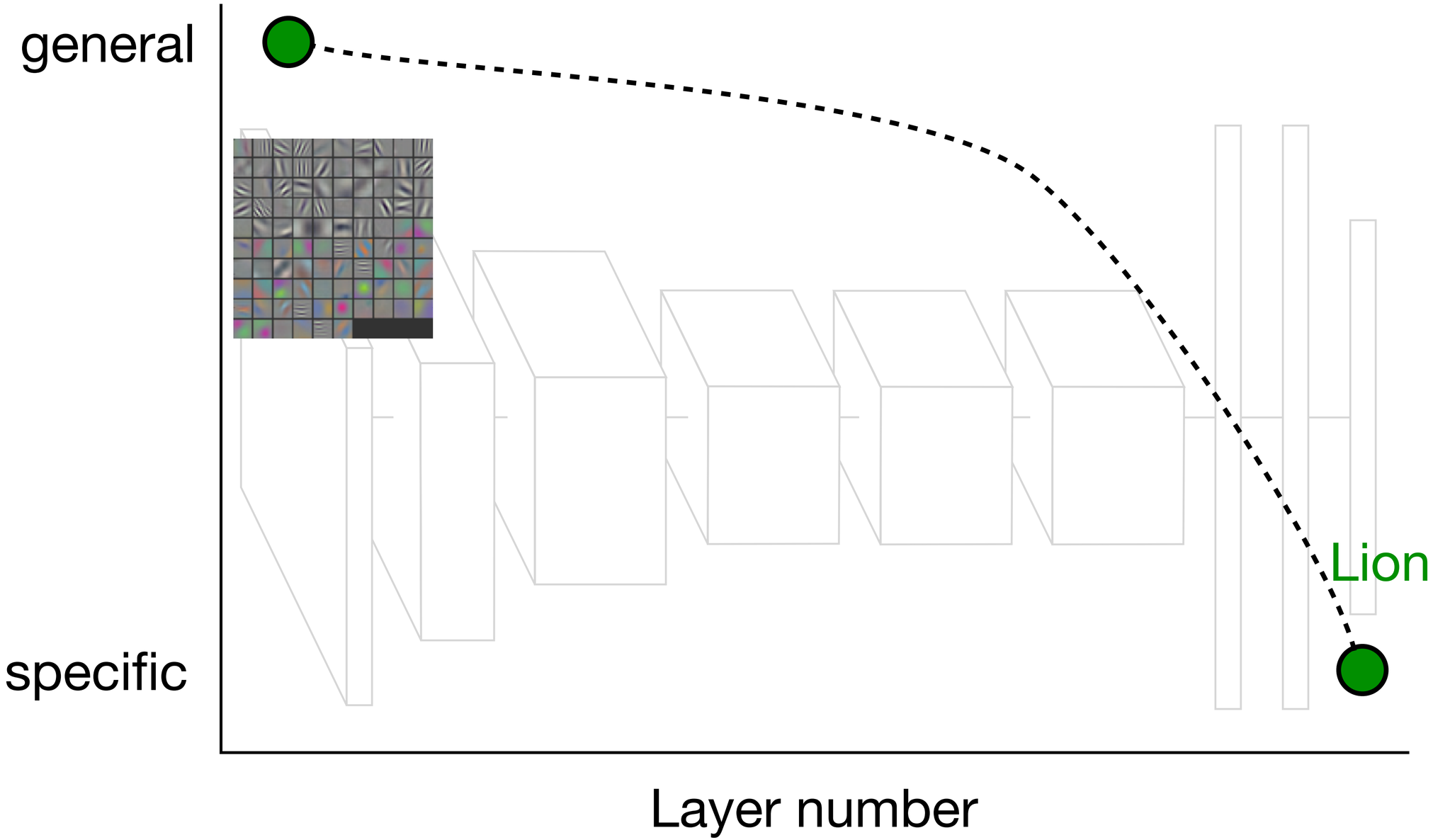
Nguyen,
Yosinski, Clune,
arXiv 2014

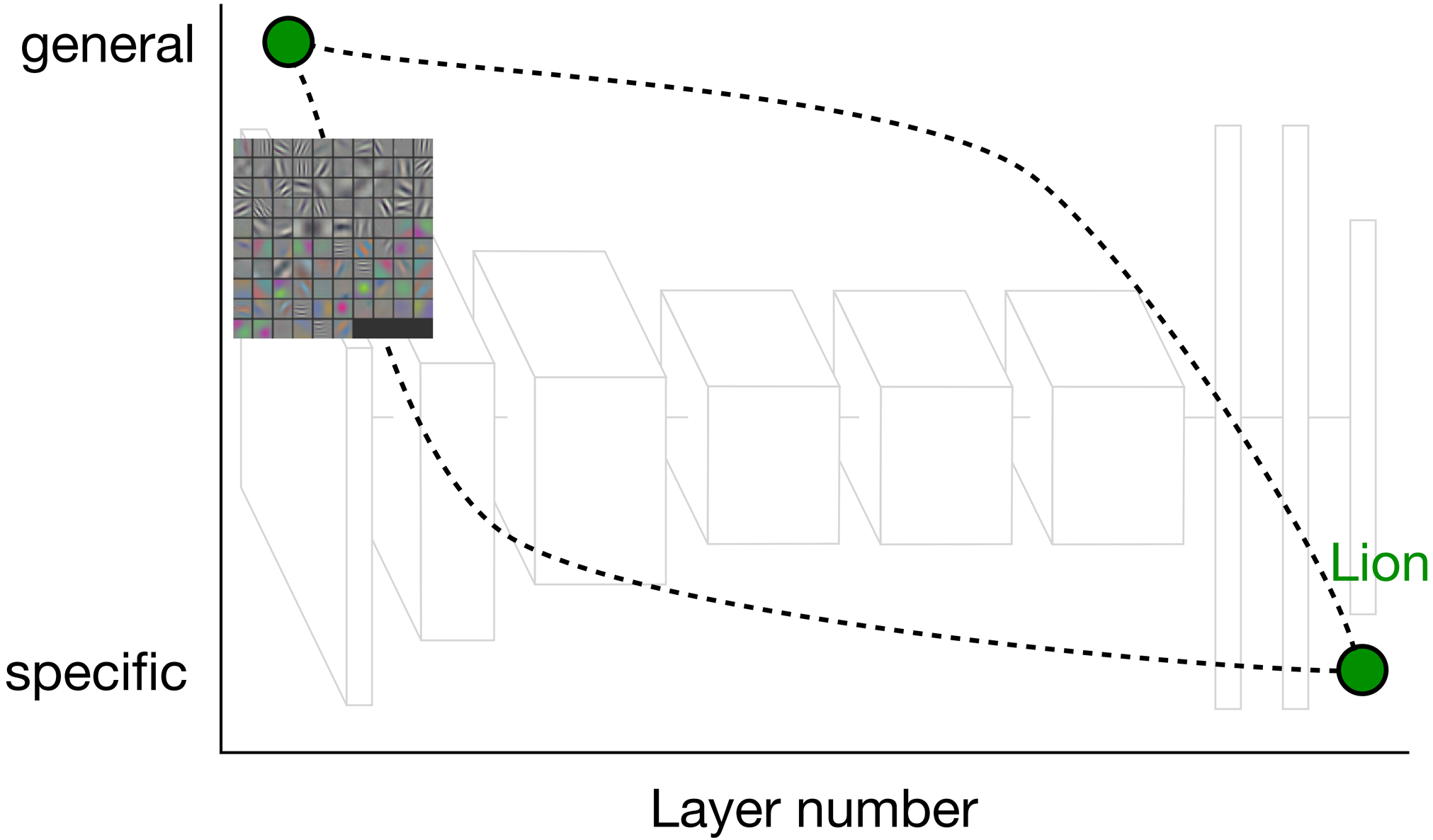


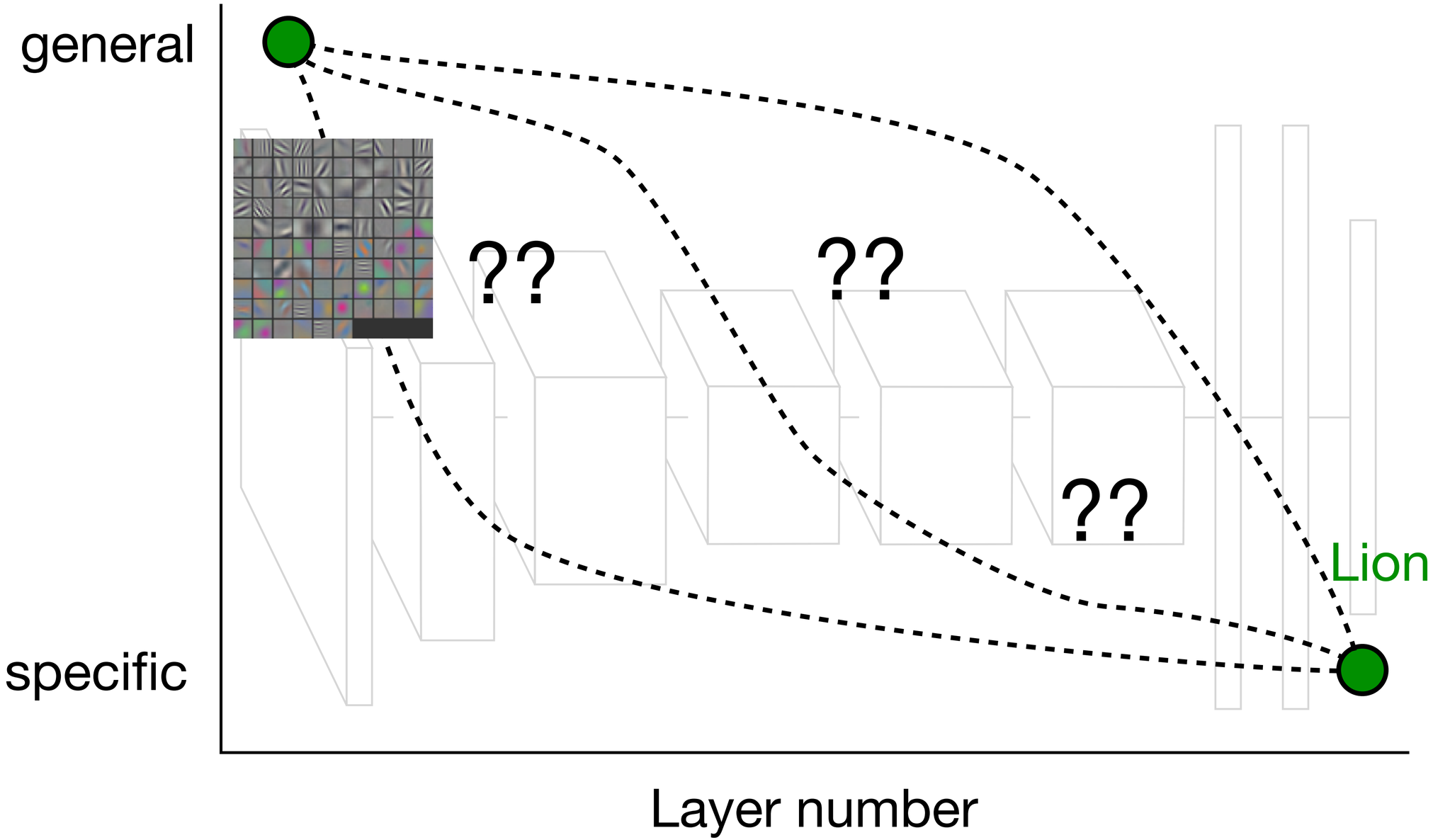

general
specific
Layer number

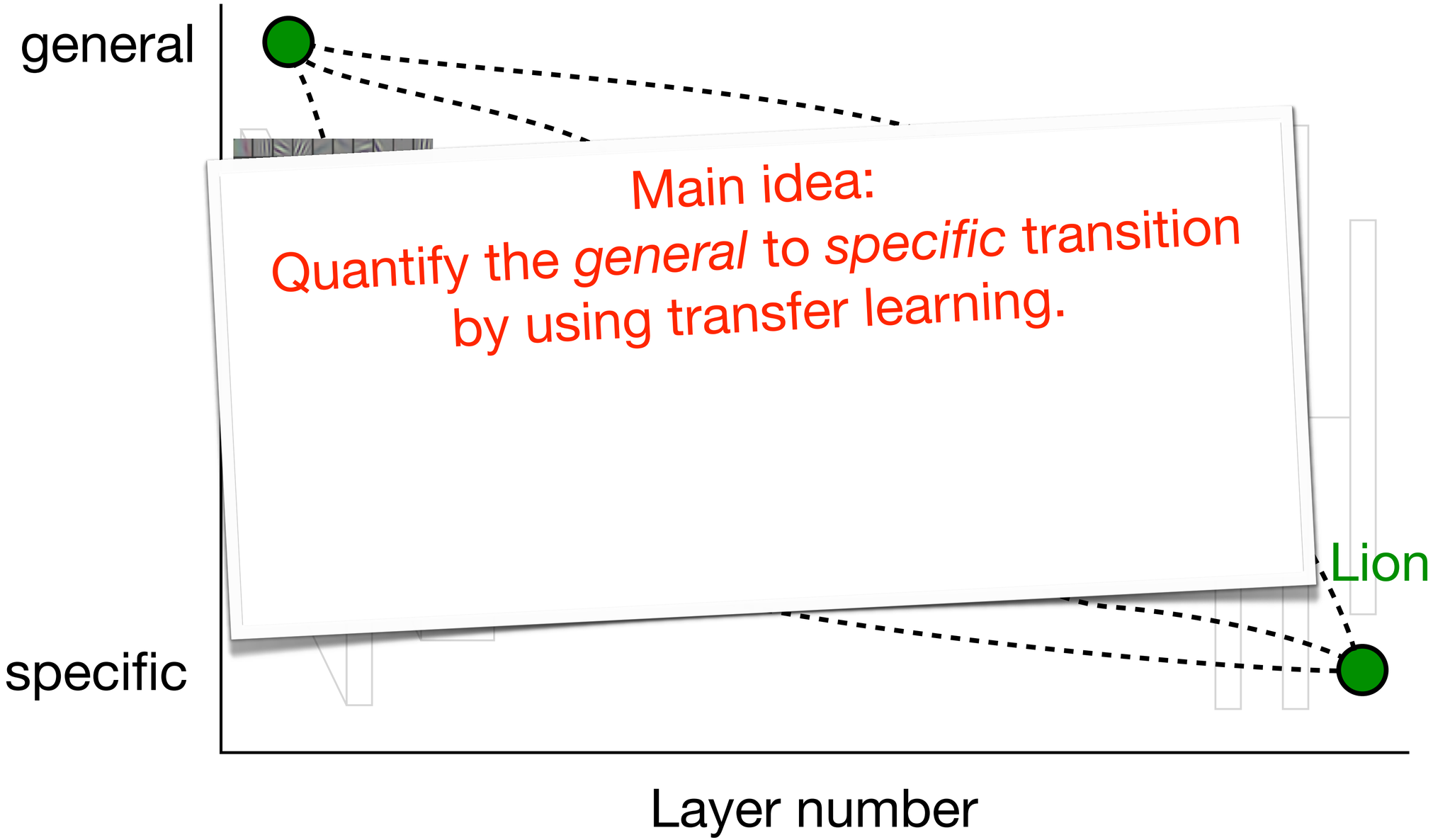

general
specific
Layer number


general
specific
Lion
Layer number

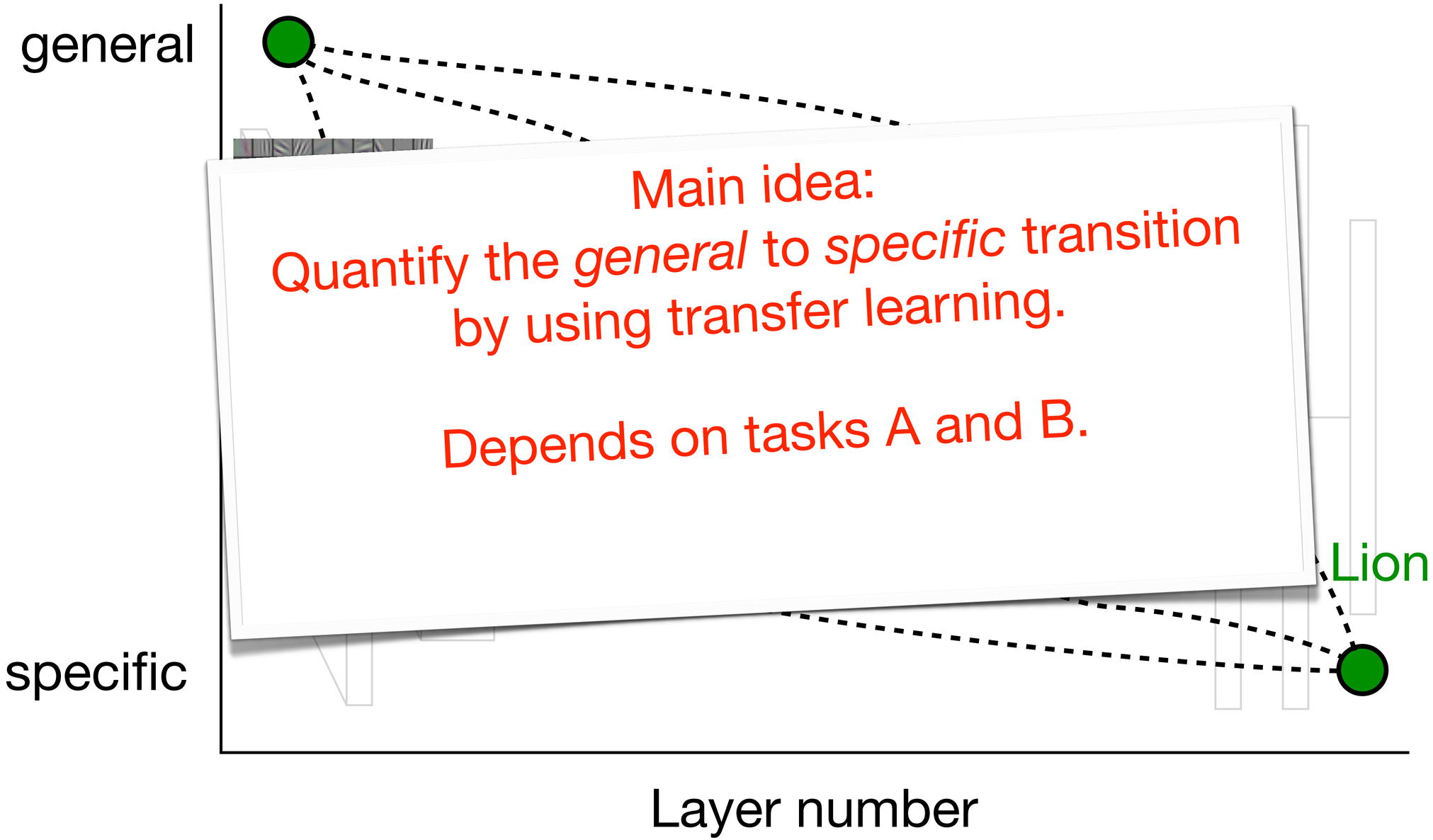

general
specific
Lion
Layer number


general
specific
Lion
Layer number


general
specific
??
??
??
Lion
Layer number

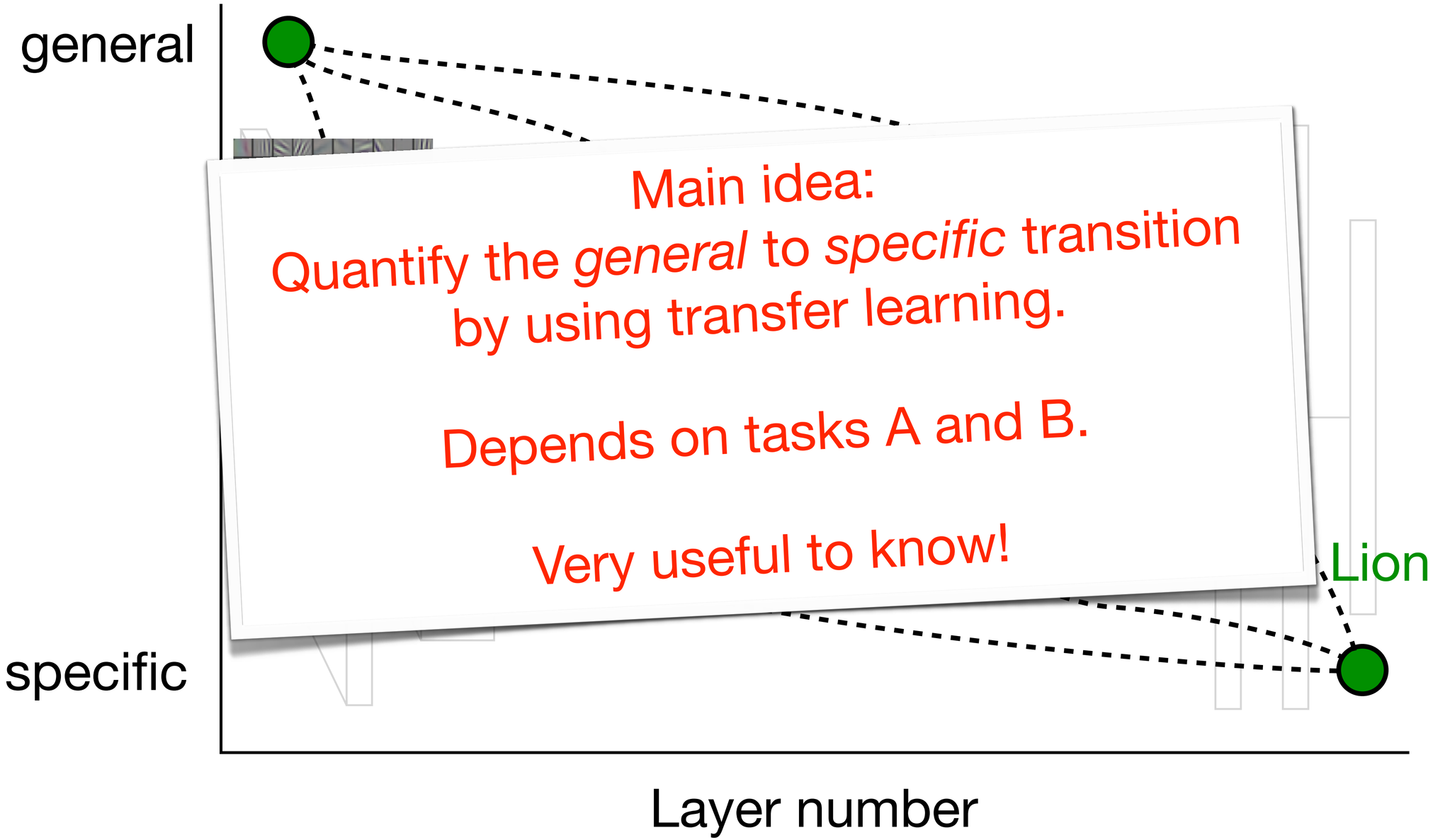


Main idea:
Quantify the *general* to *specific* transition
by using transfer learning.



Main idea:
Quantify the *general* to *specific* transition by using transfer learning.

Depends on tasks A and B.



Main idea:

Quantify the *general* to *specific* transition by using transfer learning.

Depends on tasks A and B.

Very useful to know!

ImageNet

1000 Classes

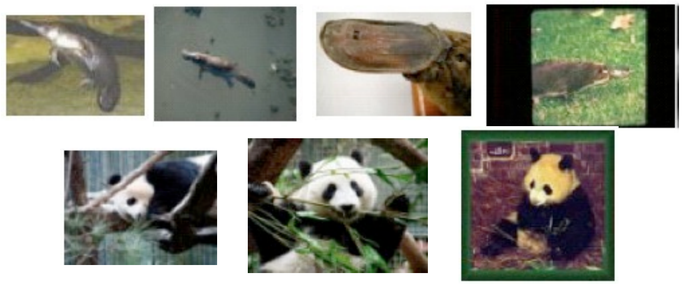
Deng et al., 2009

ImageNet

1000 Classes

dataset

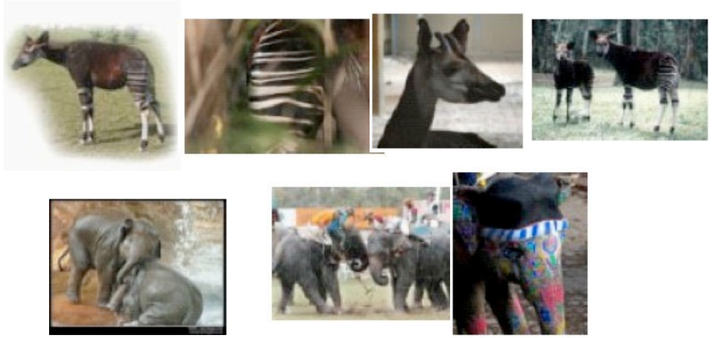
A



500 Classes

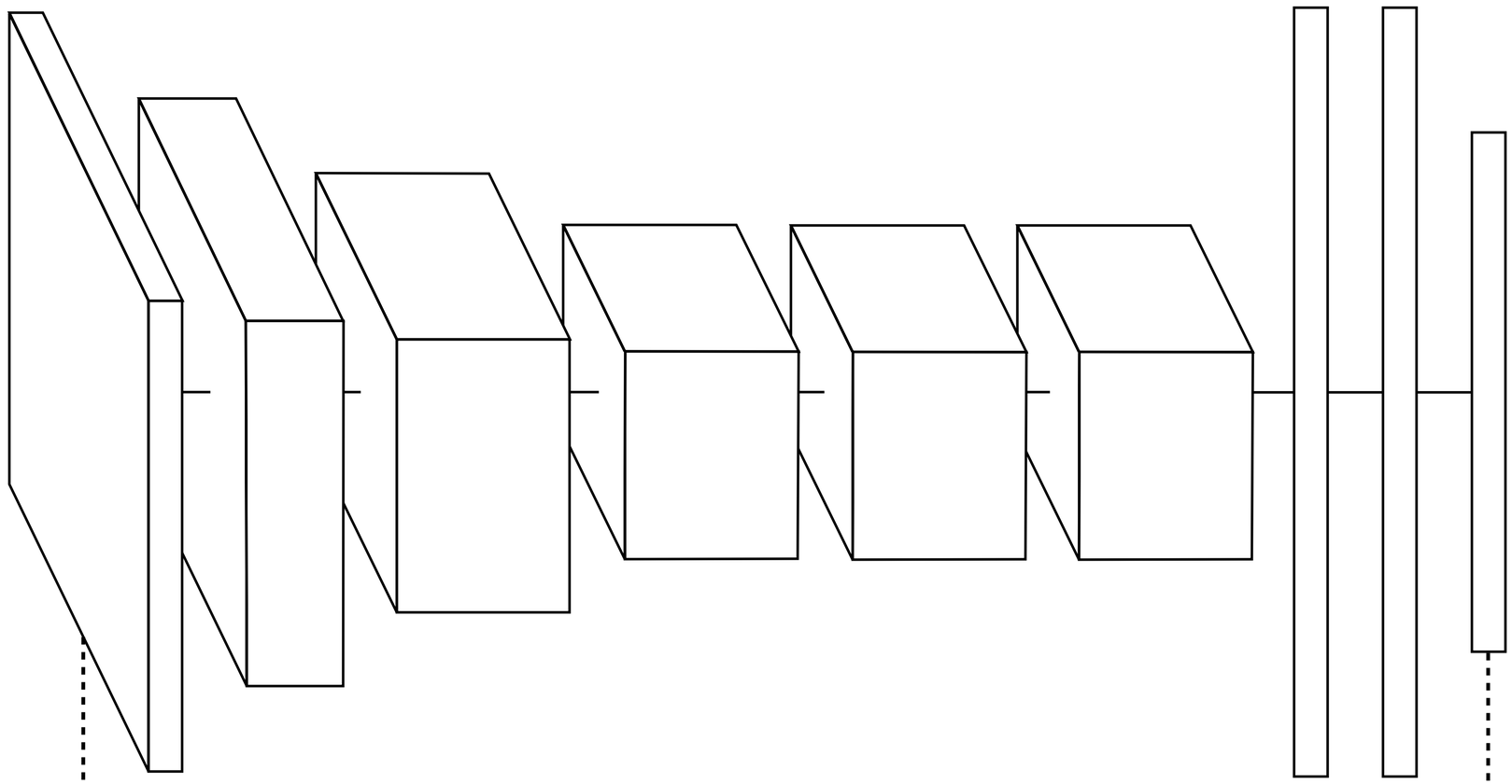
dataset

B



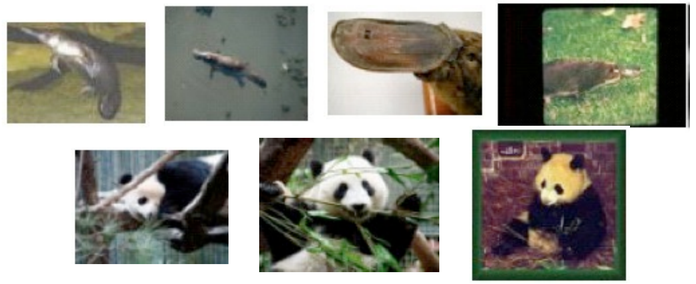
500 Classes

Deng et al., 2009



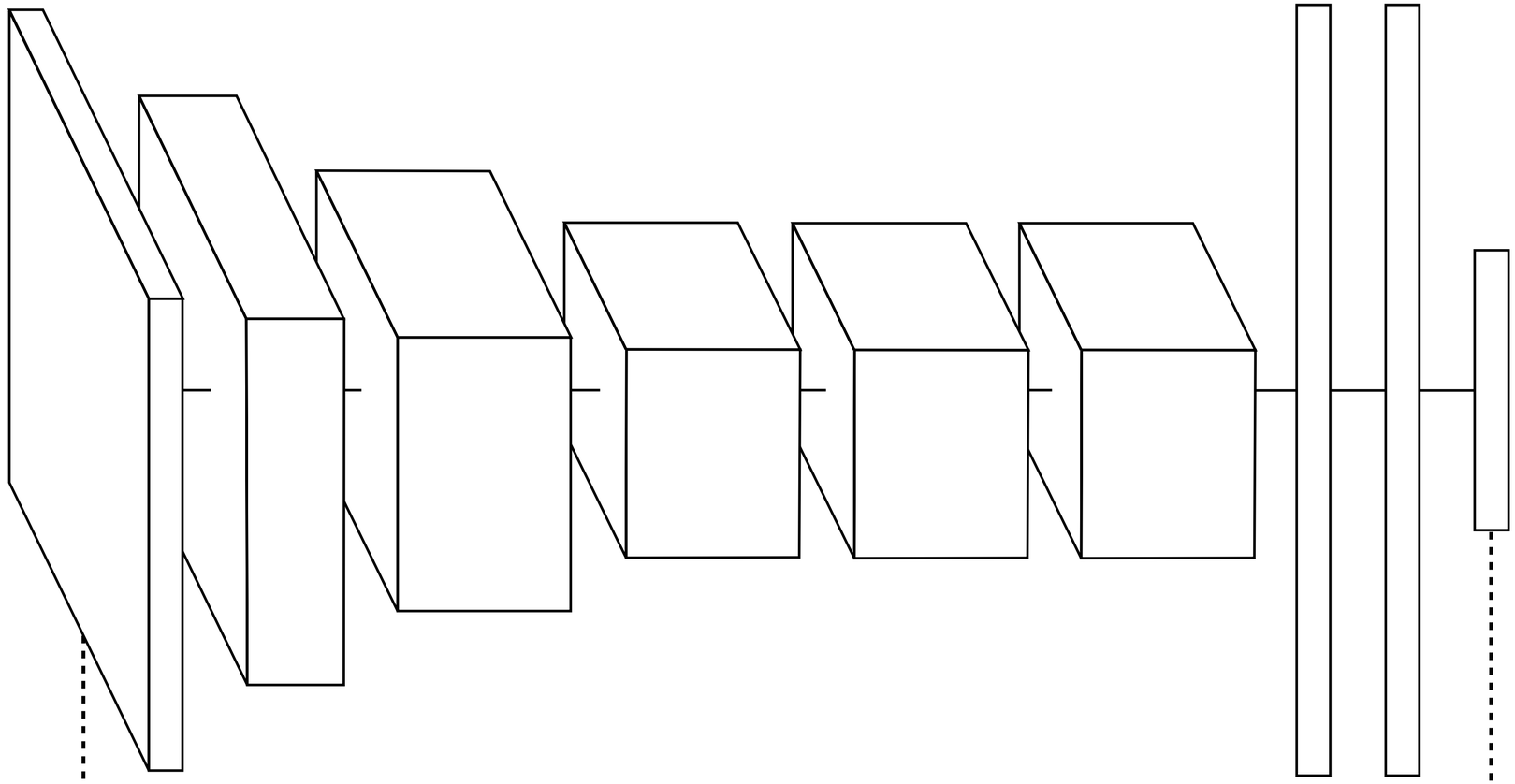
A Images

A Labels



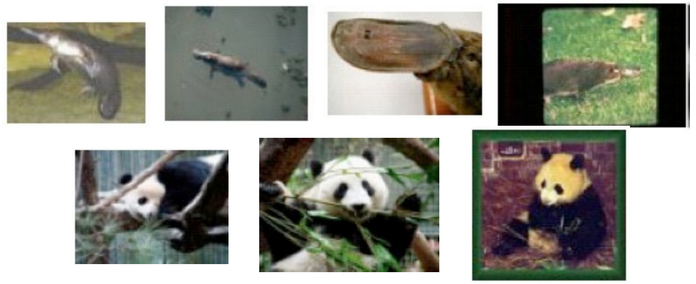
500 Classes

Train using Caffe framework (Jia et al. 2014)



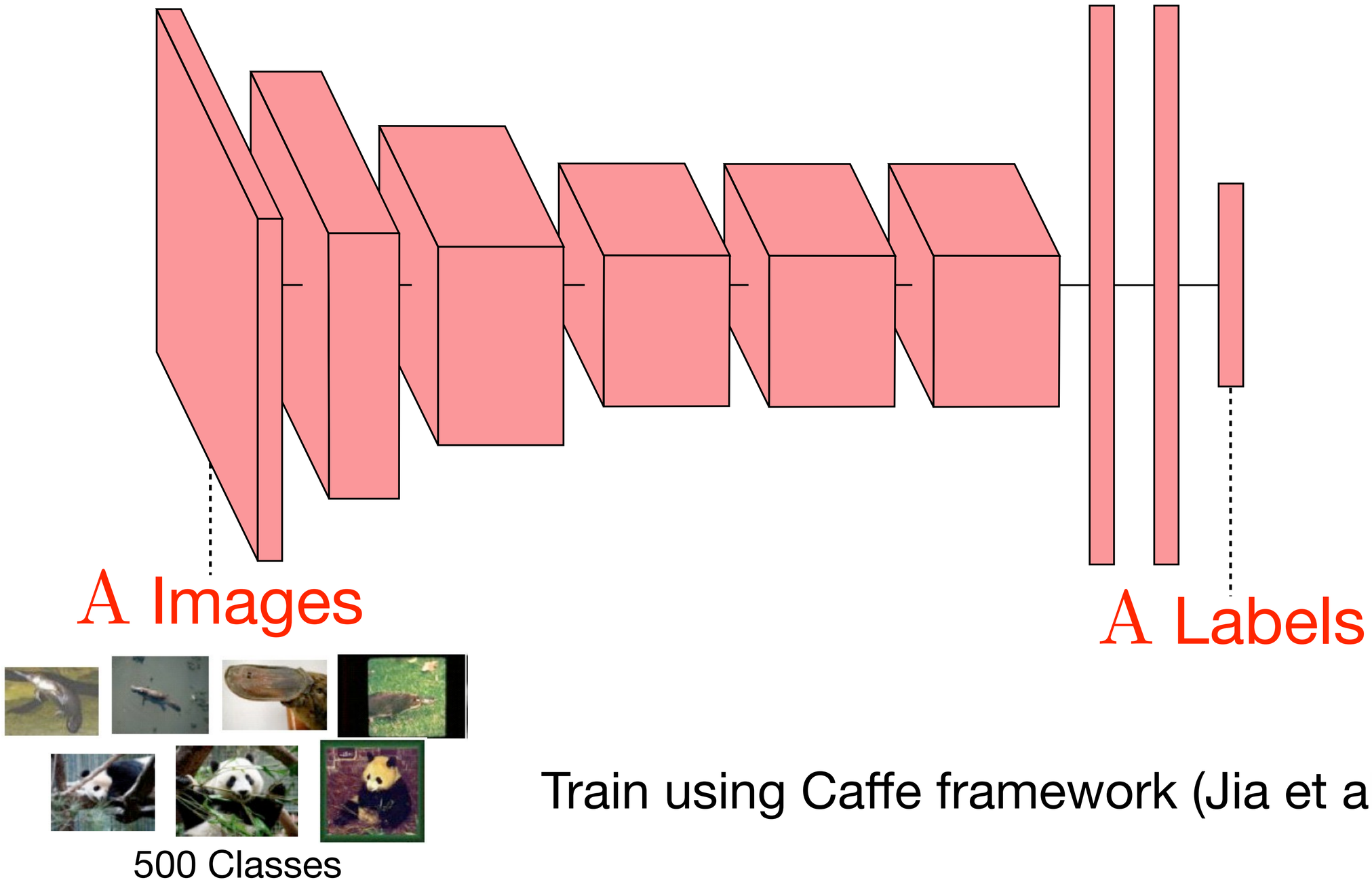
A Images

A Labels

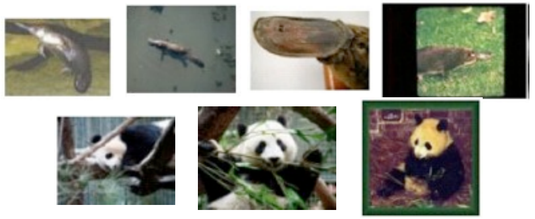


500 Classes

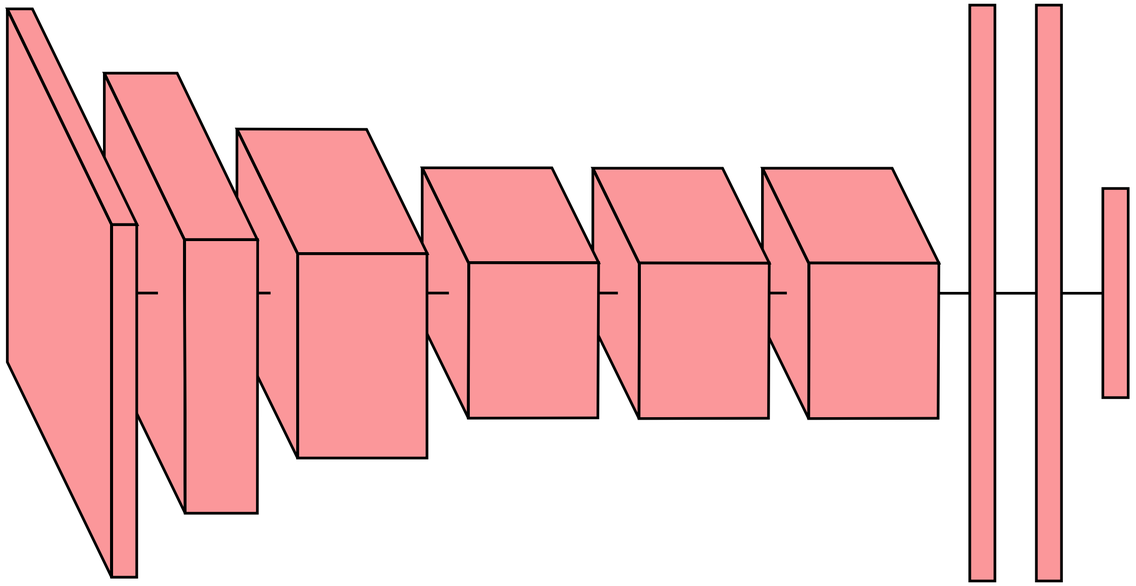
Train using Caffe framework (Jia et al. 2014)



Train using Caffe framework (Jia et al.)

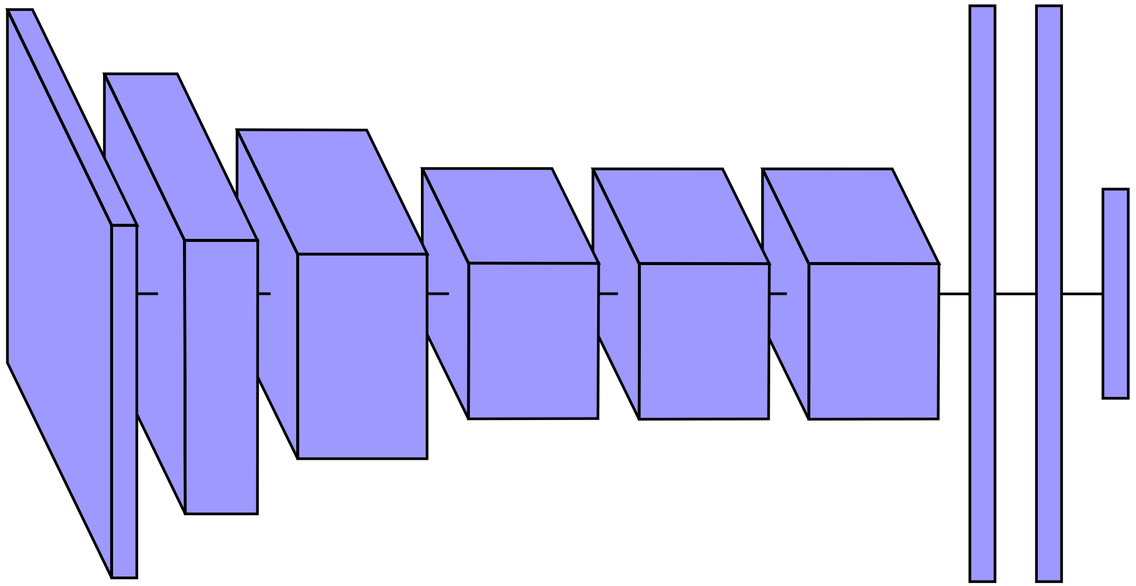


A Images

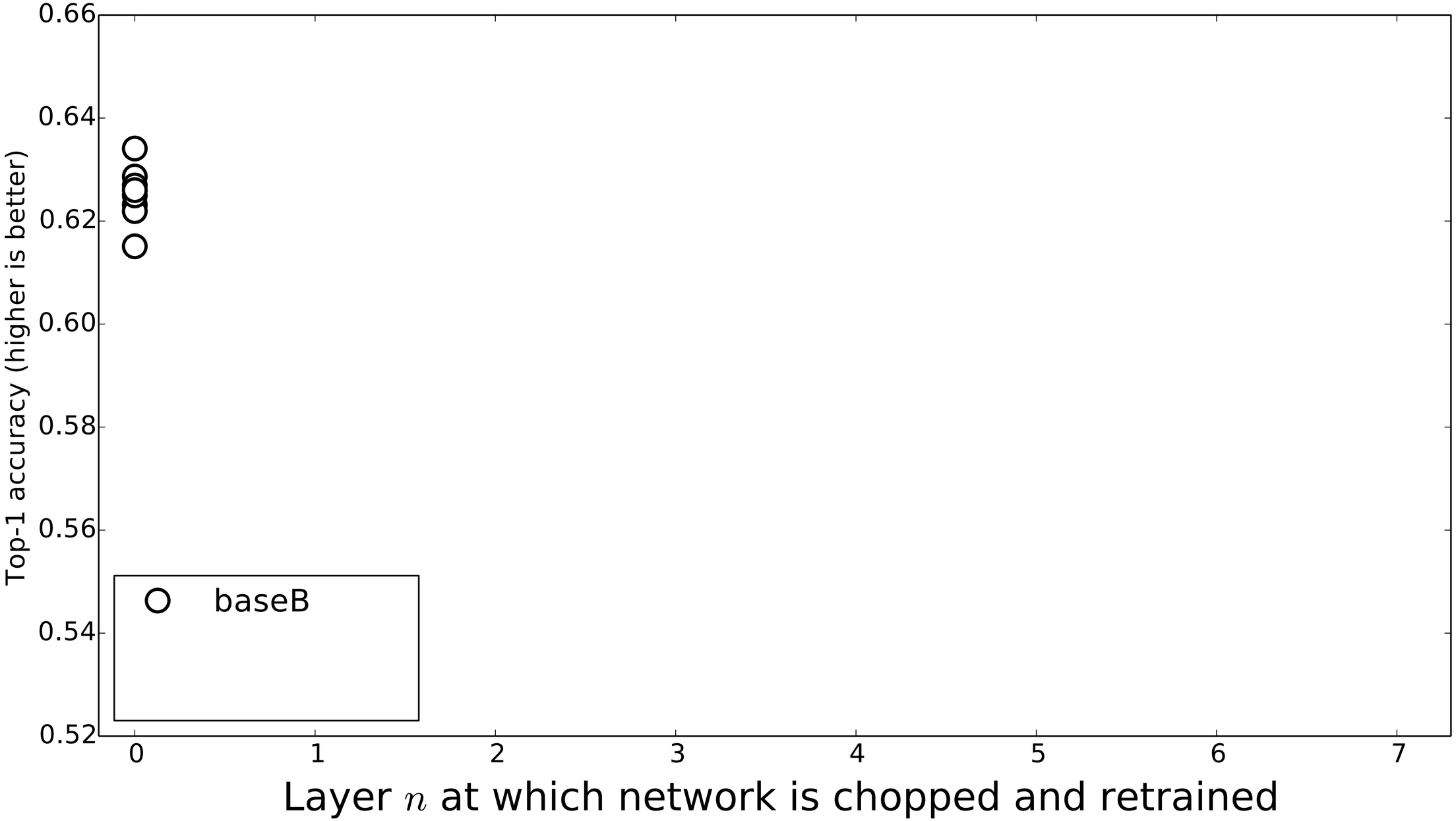


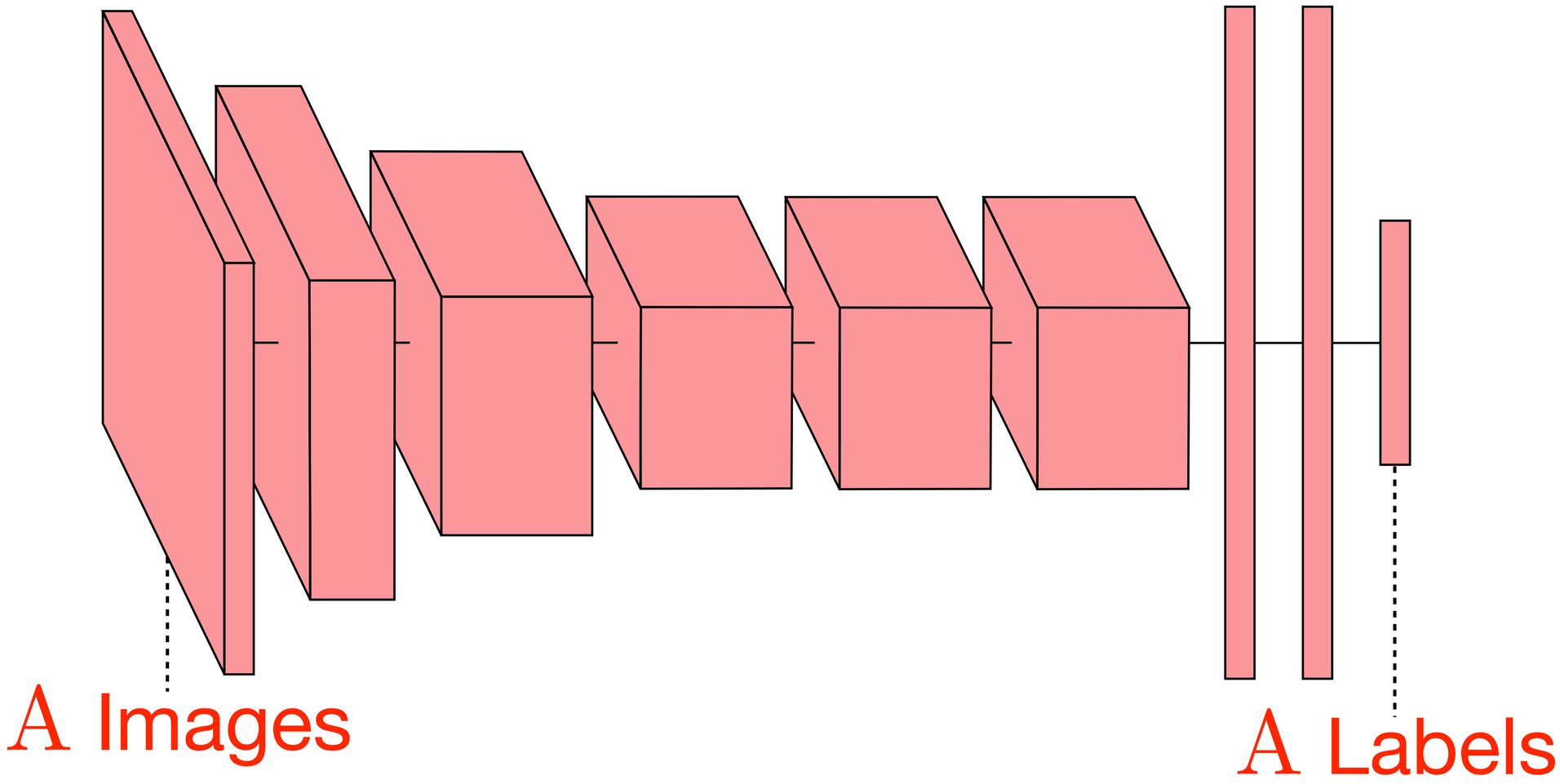
baseA

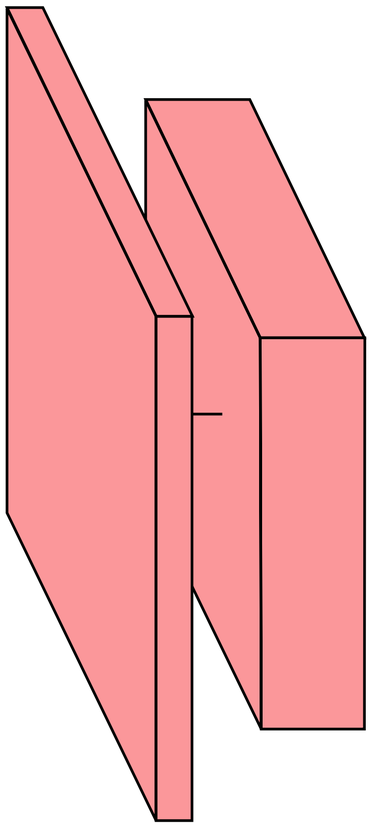
B Images

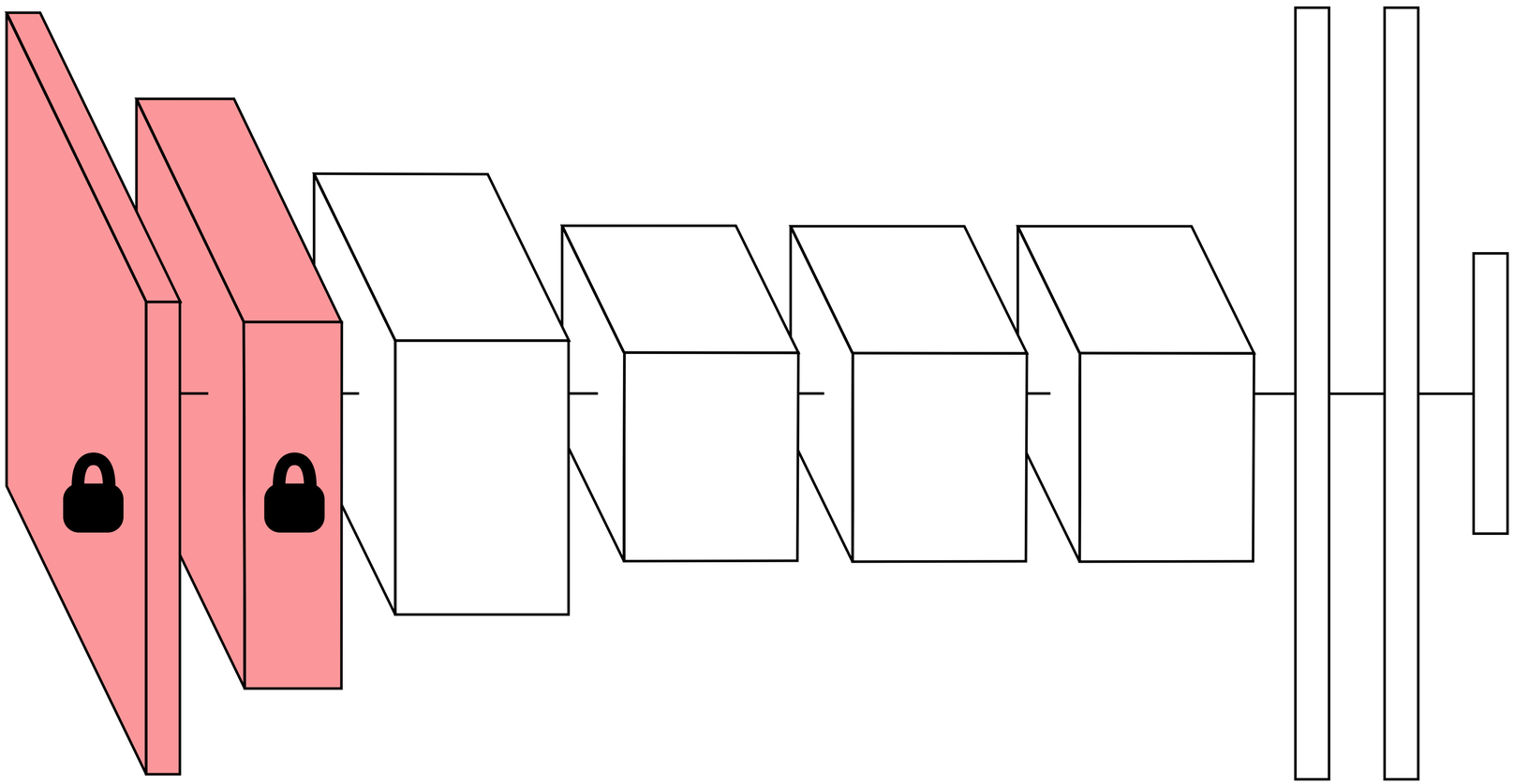


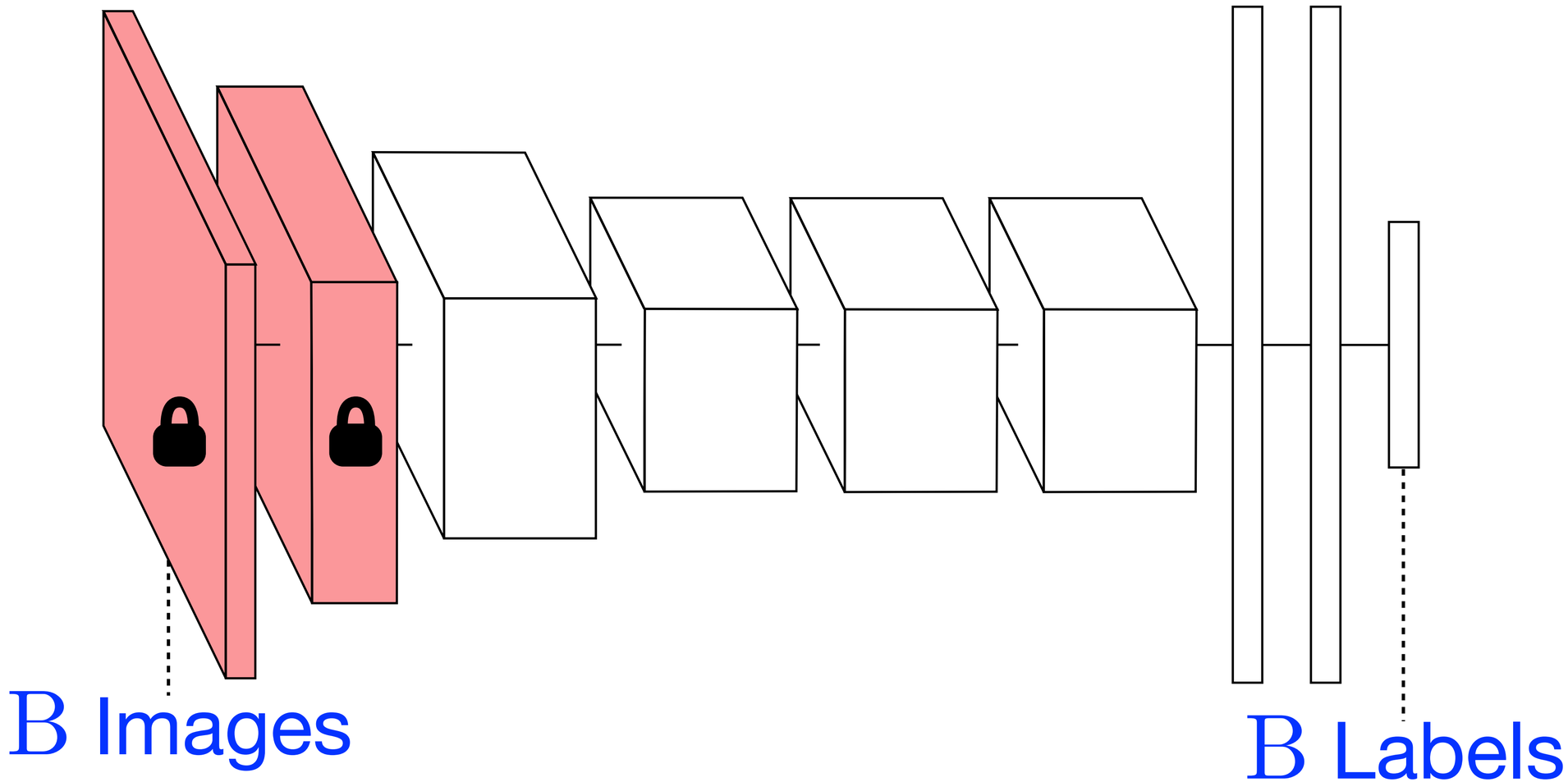
baseB

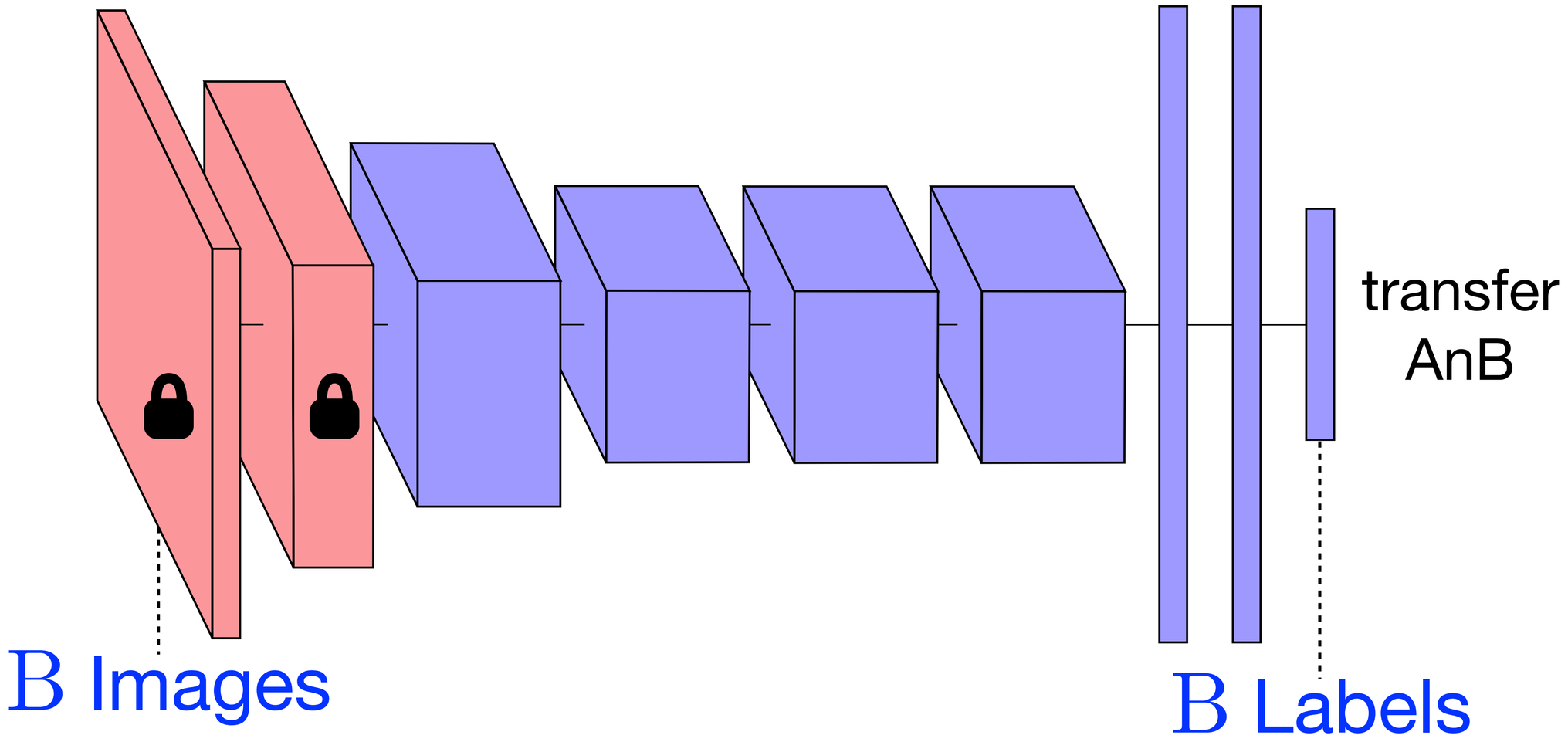

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
baseB
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained

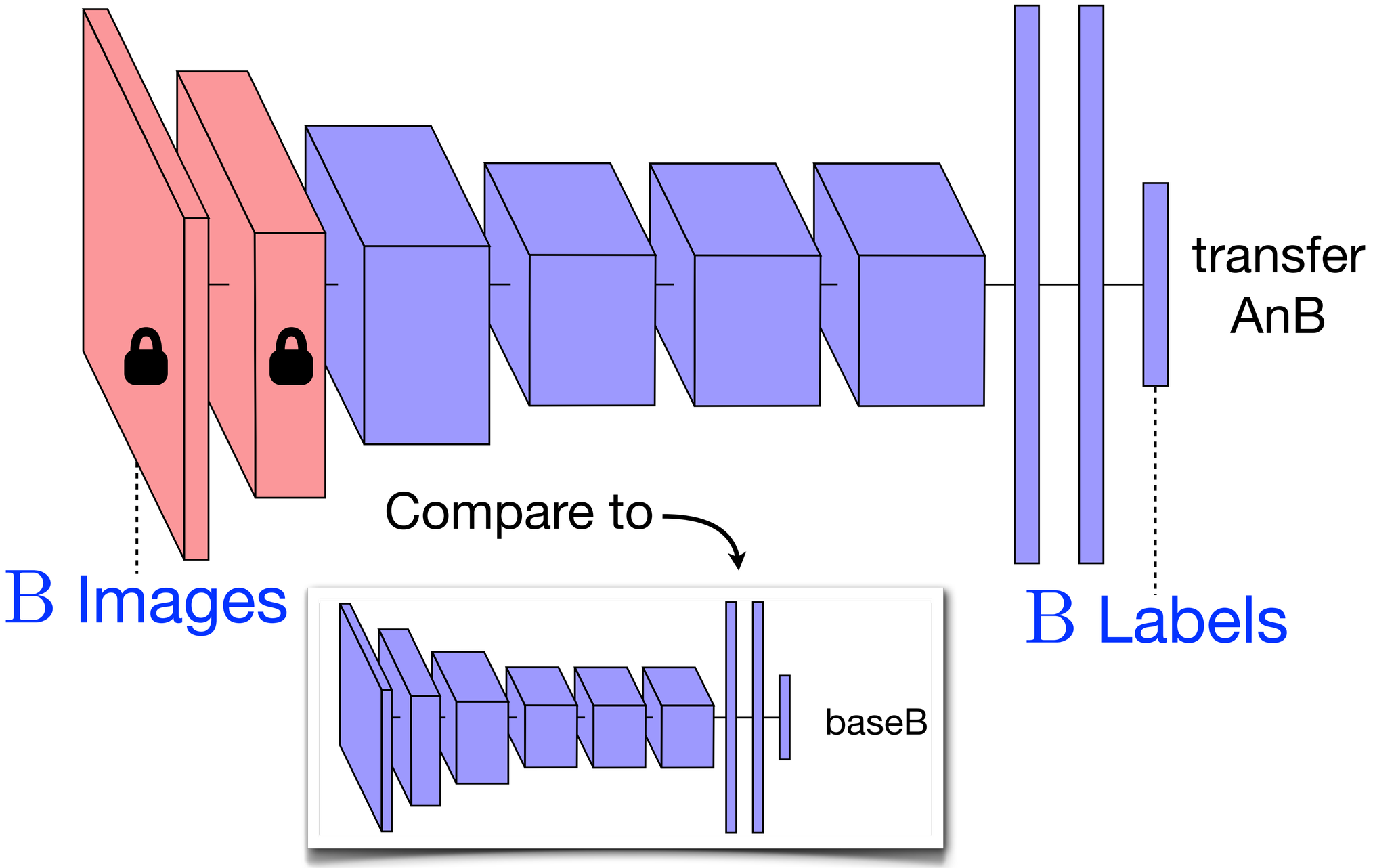

A Images
A Labels

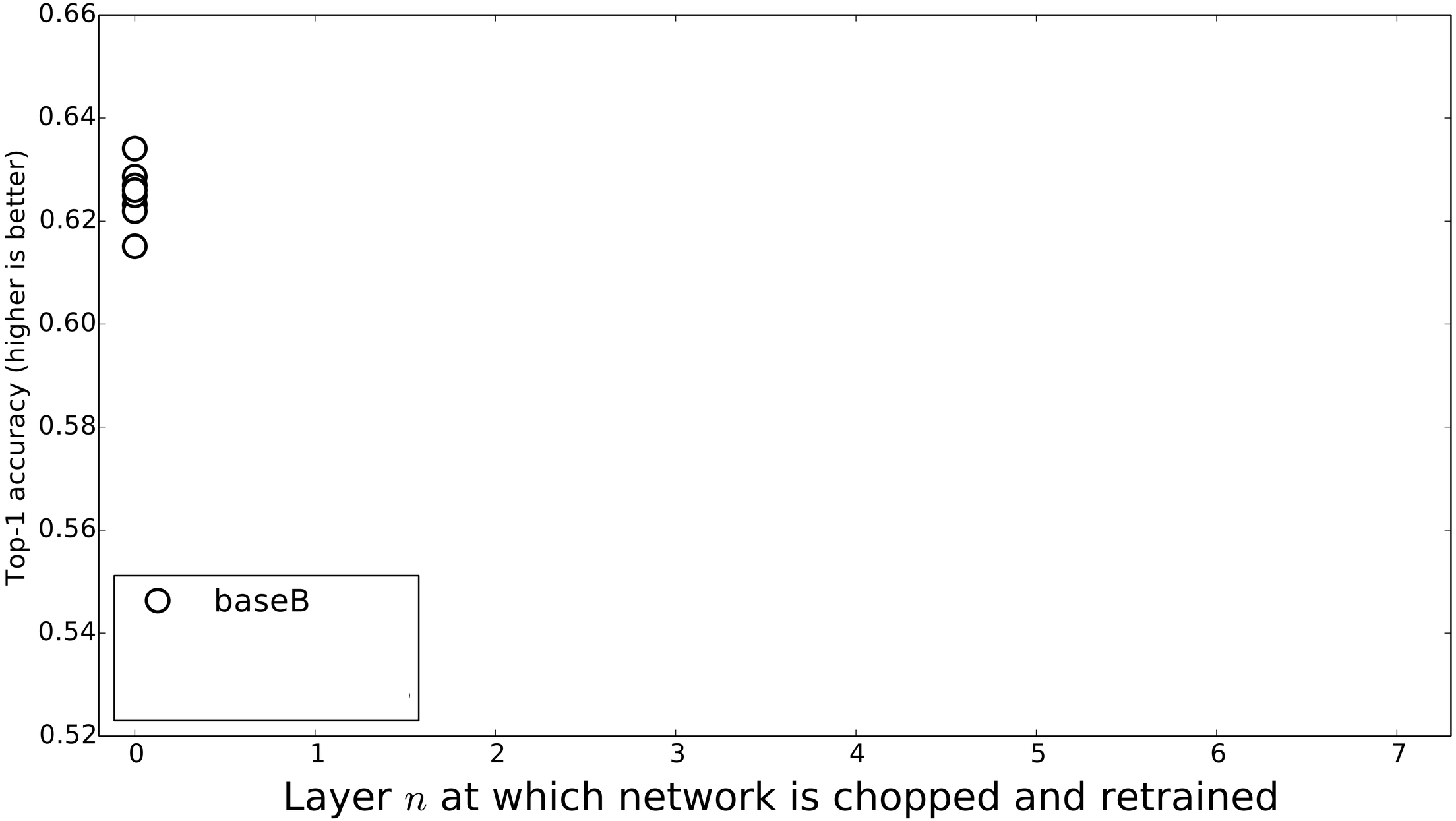


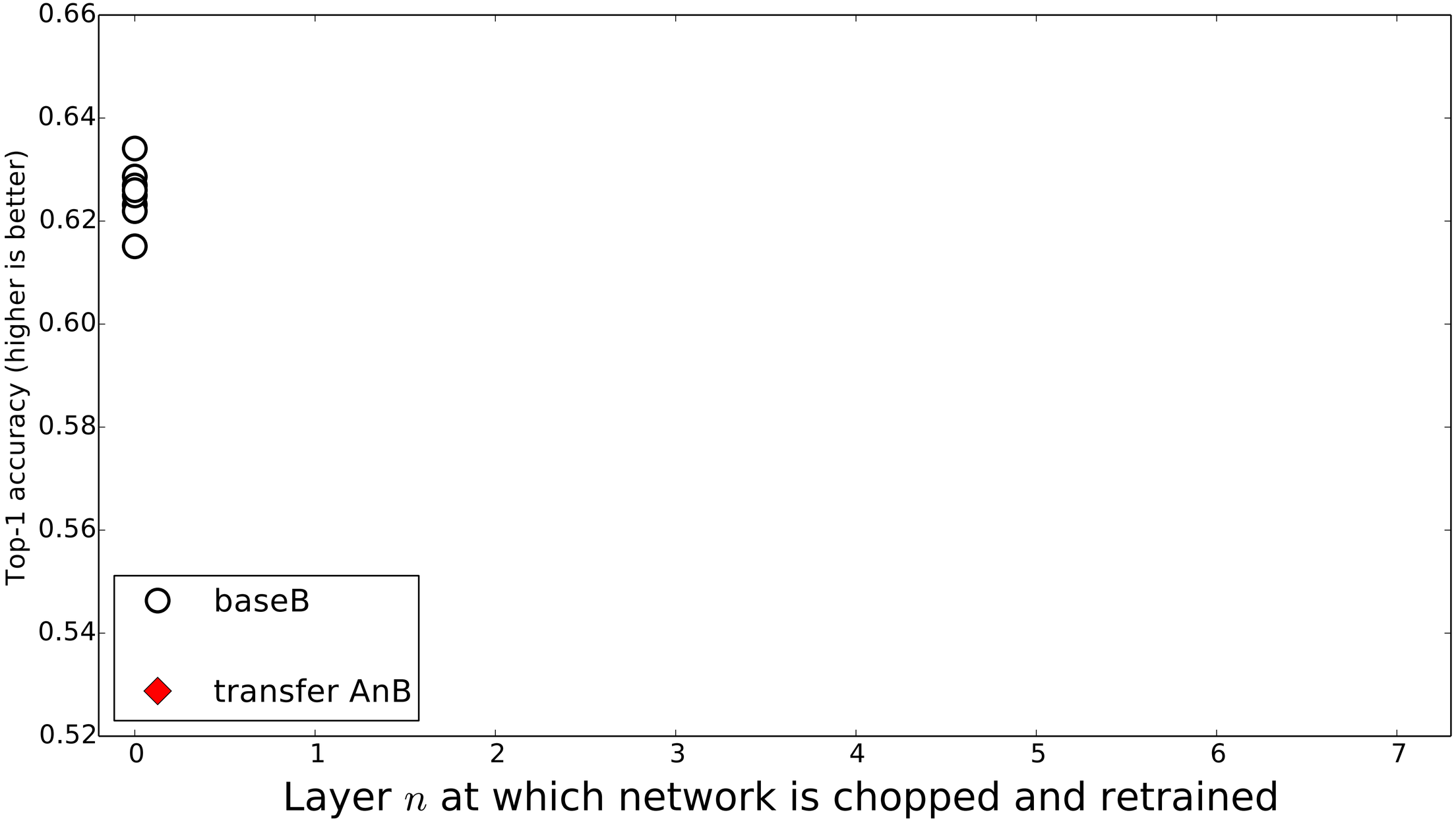


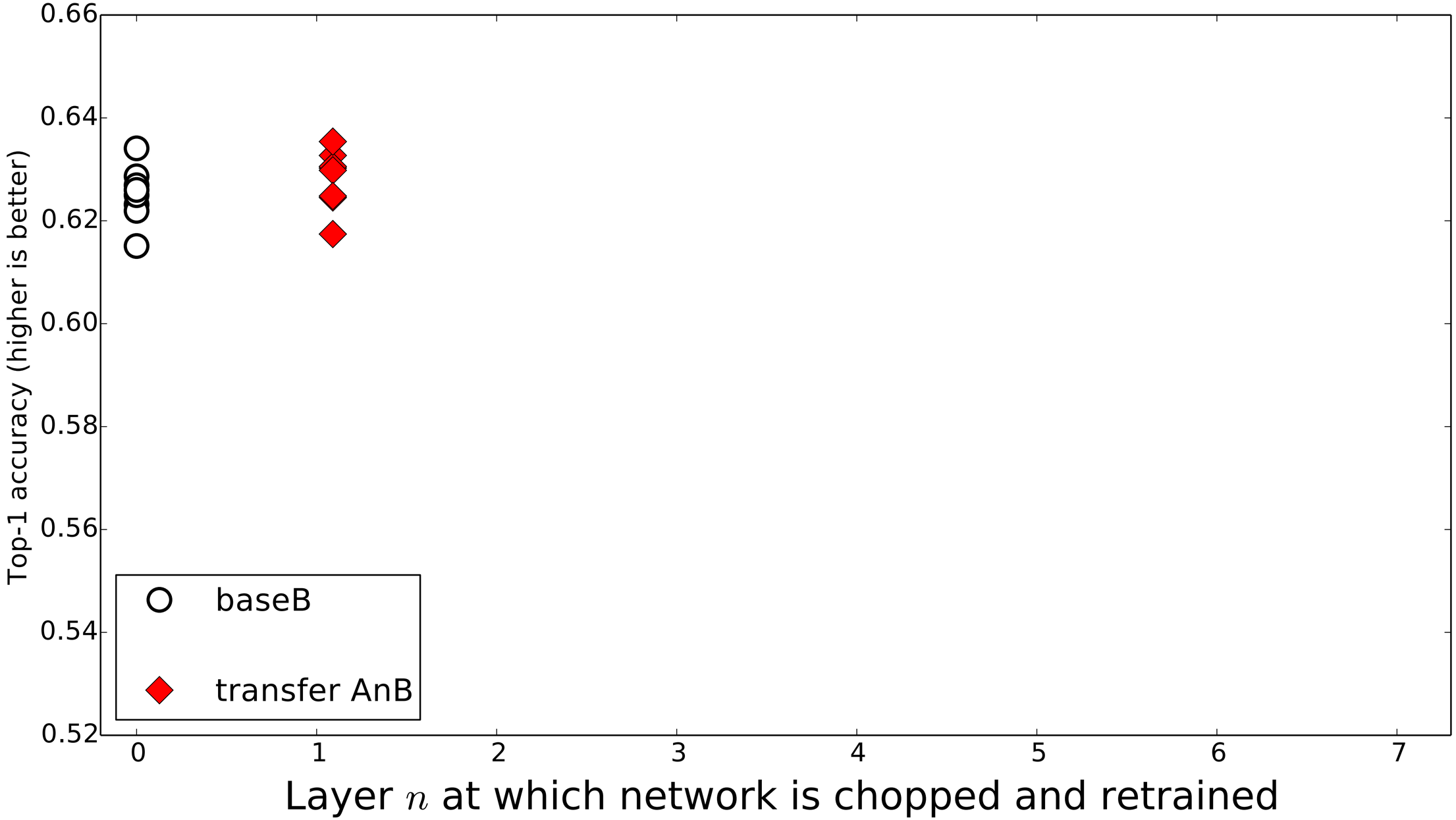

B Images
B Labels

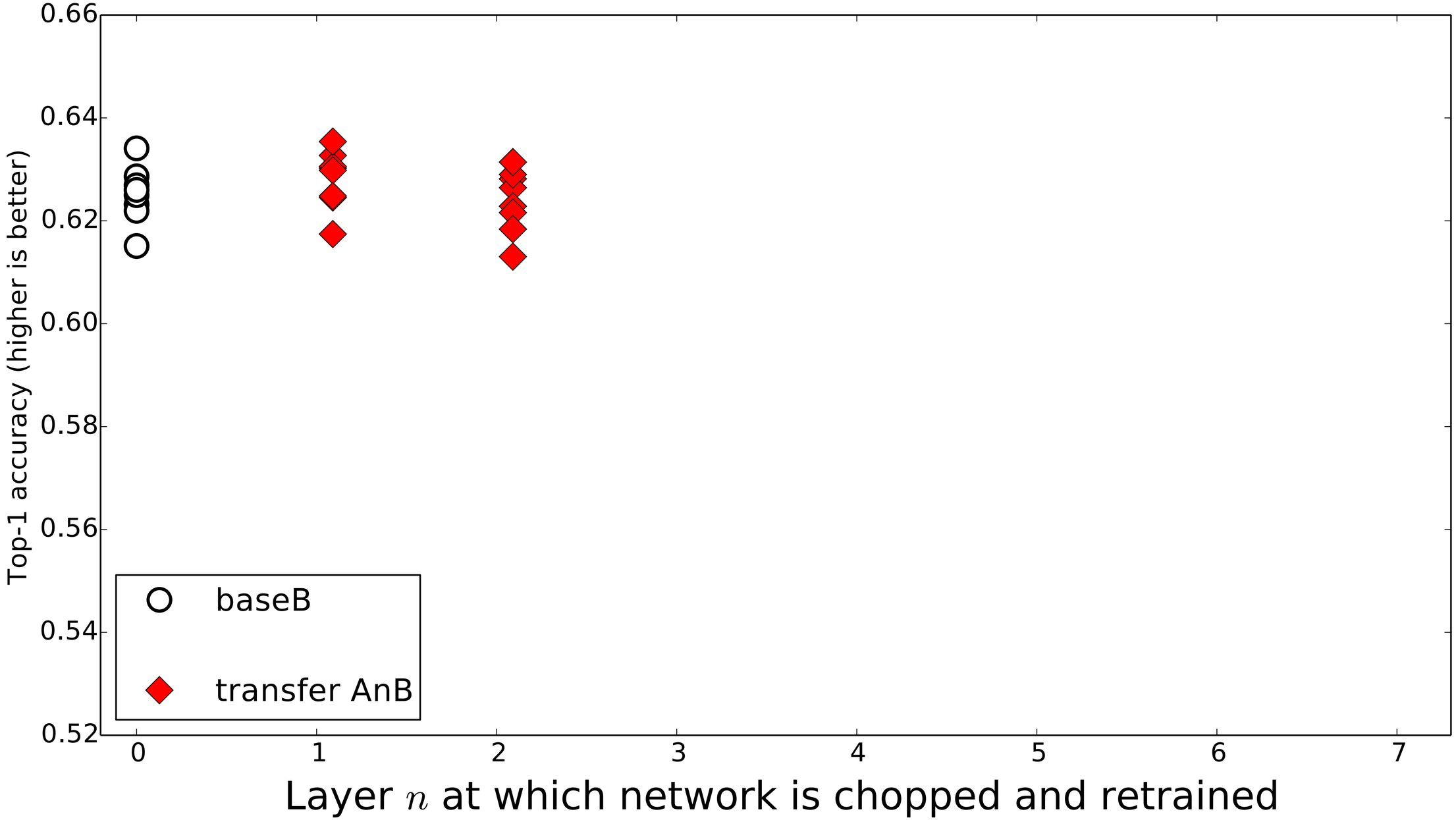

transfer
AnB
B Images
B Labels

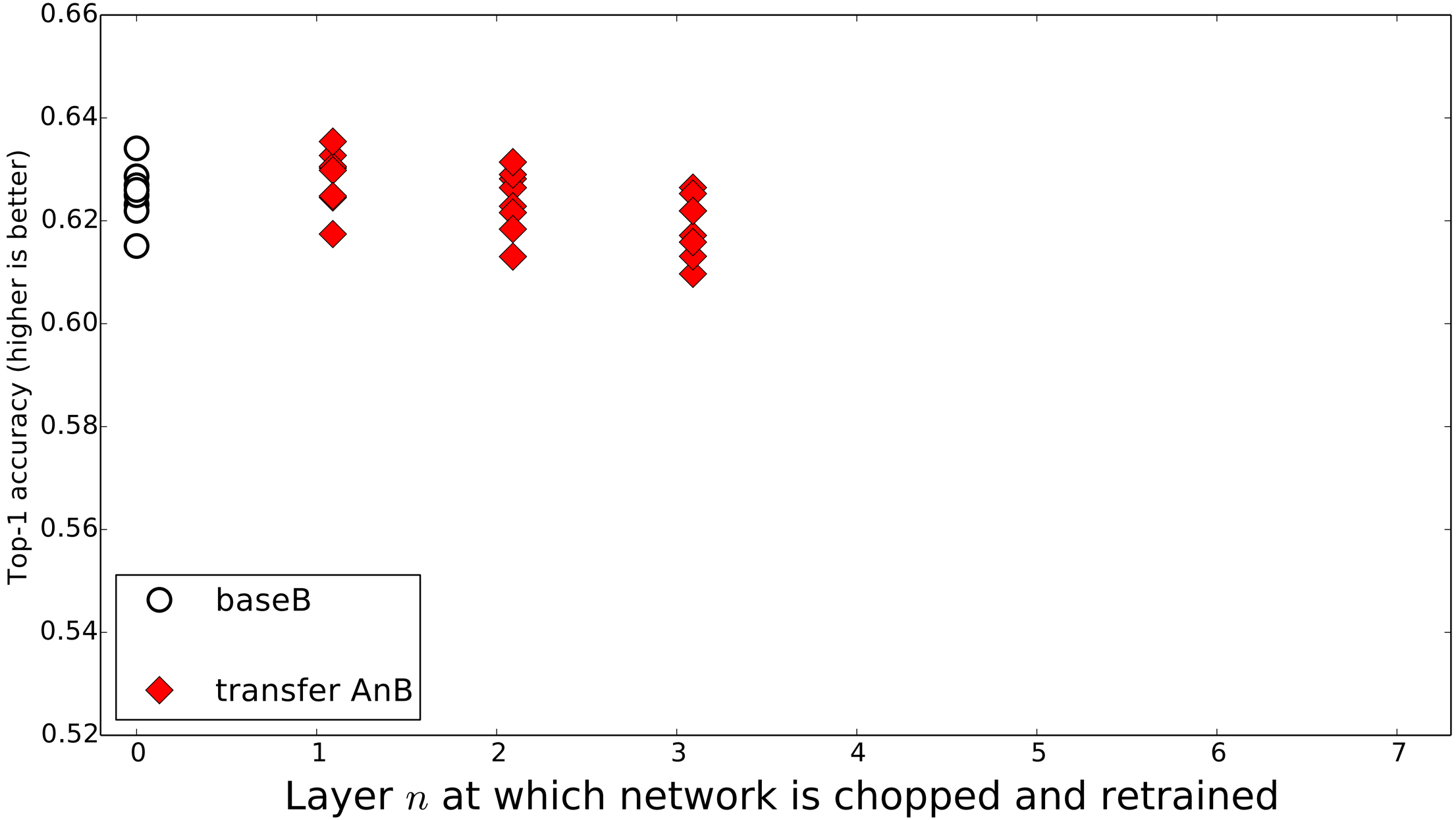

transfer
AnB
Compare to
B Images
B Labels
baseB

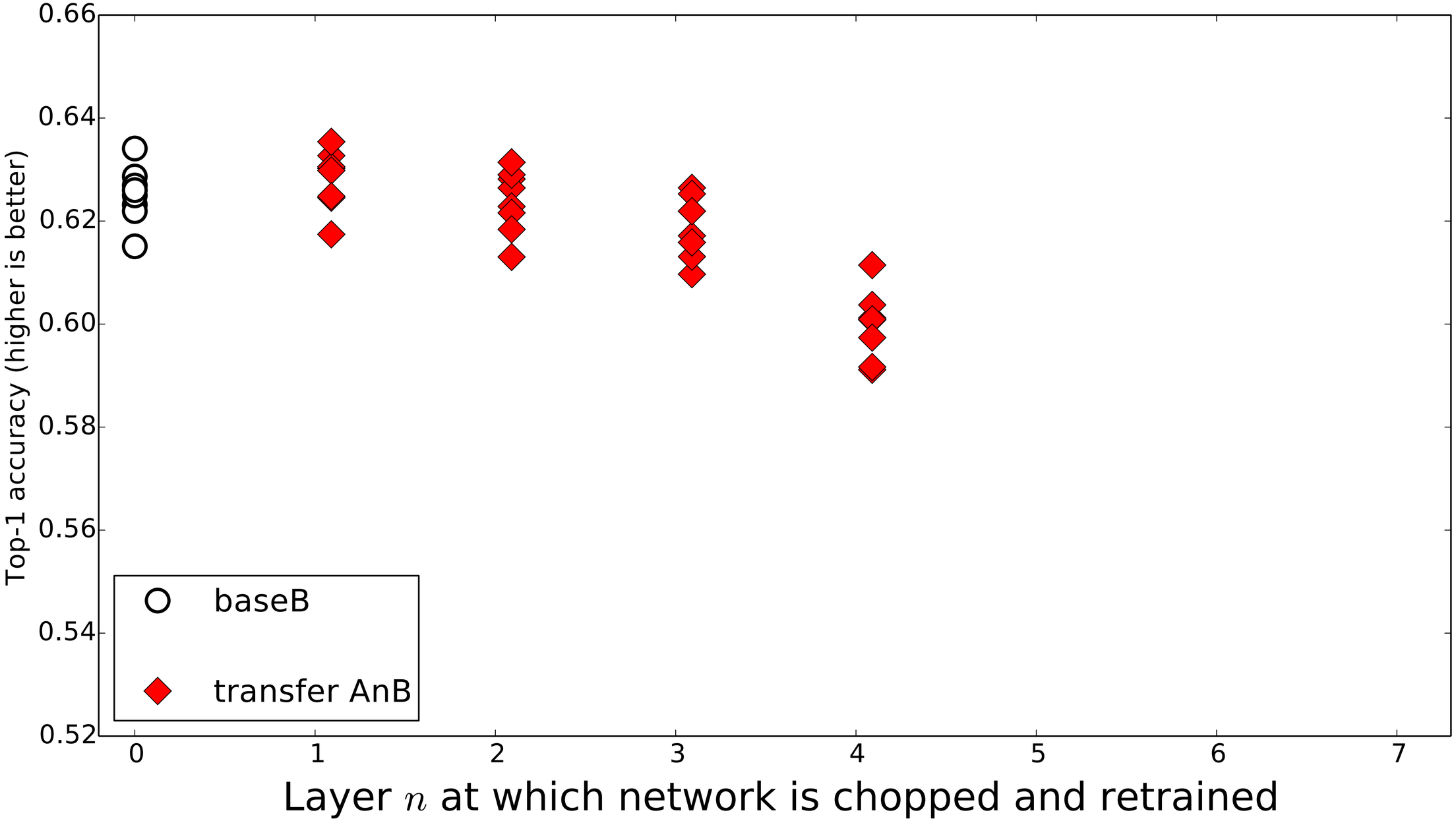

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
baseB
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained

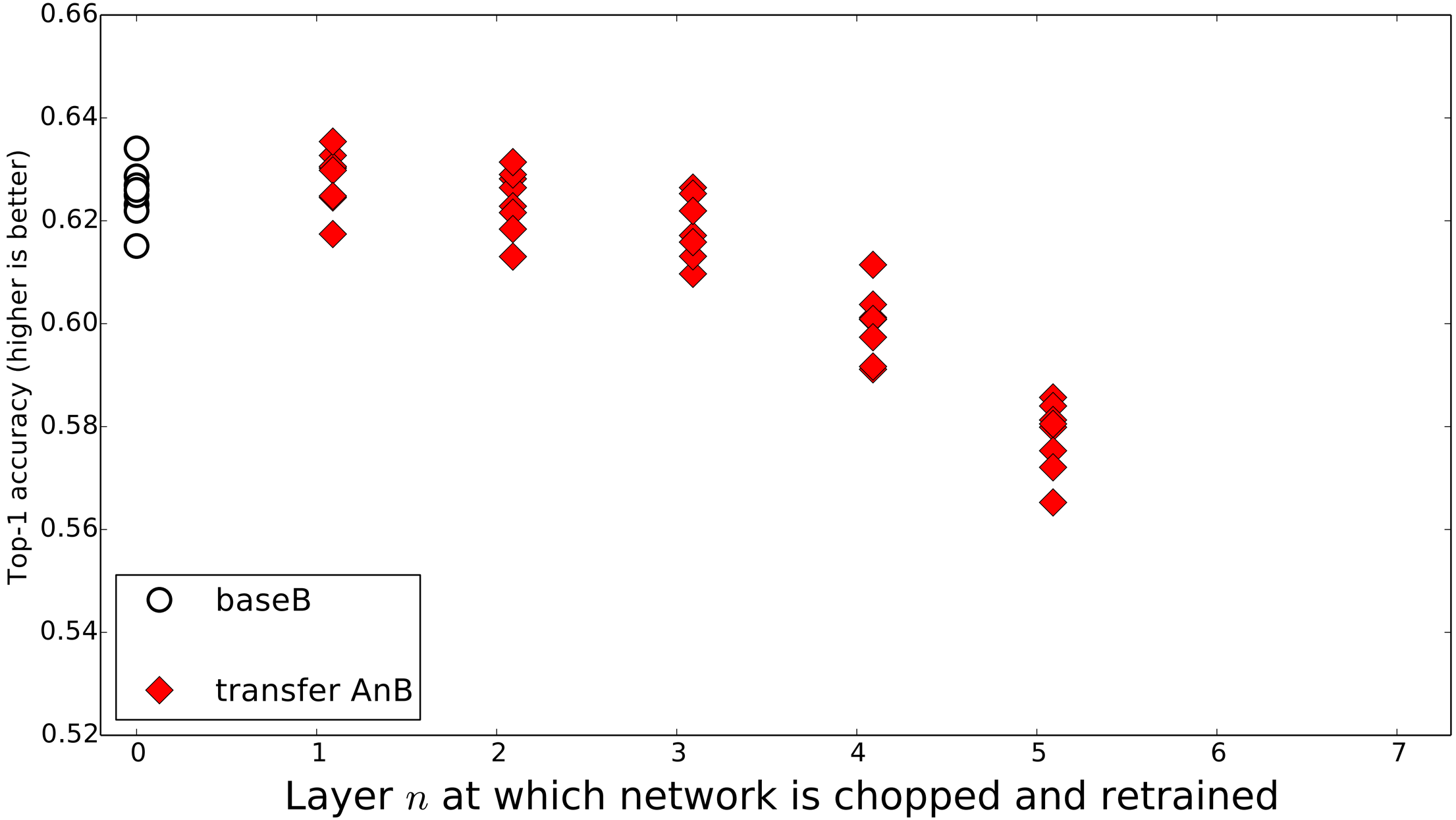

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

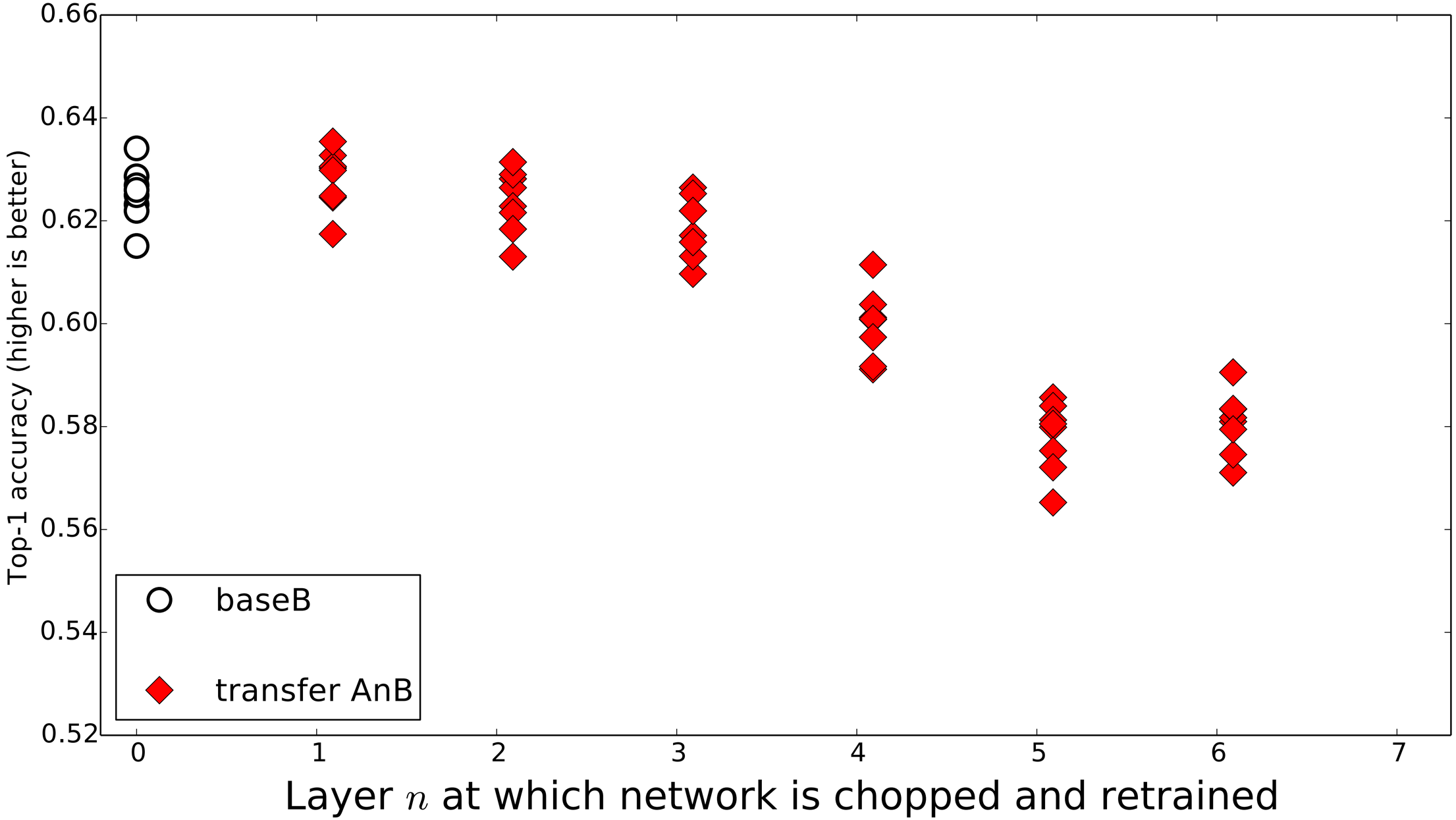

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

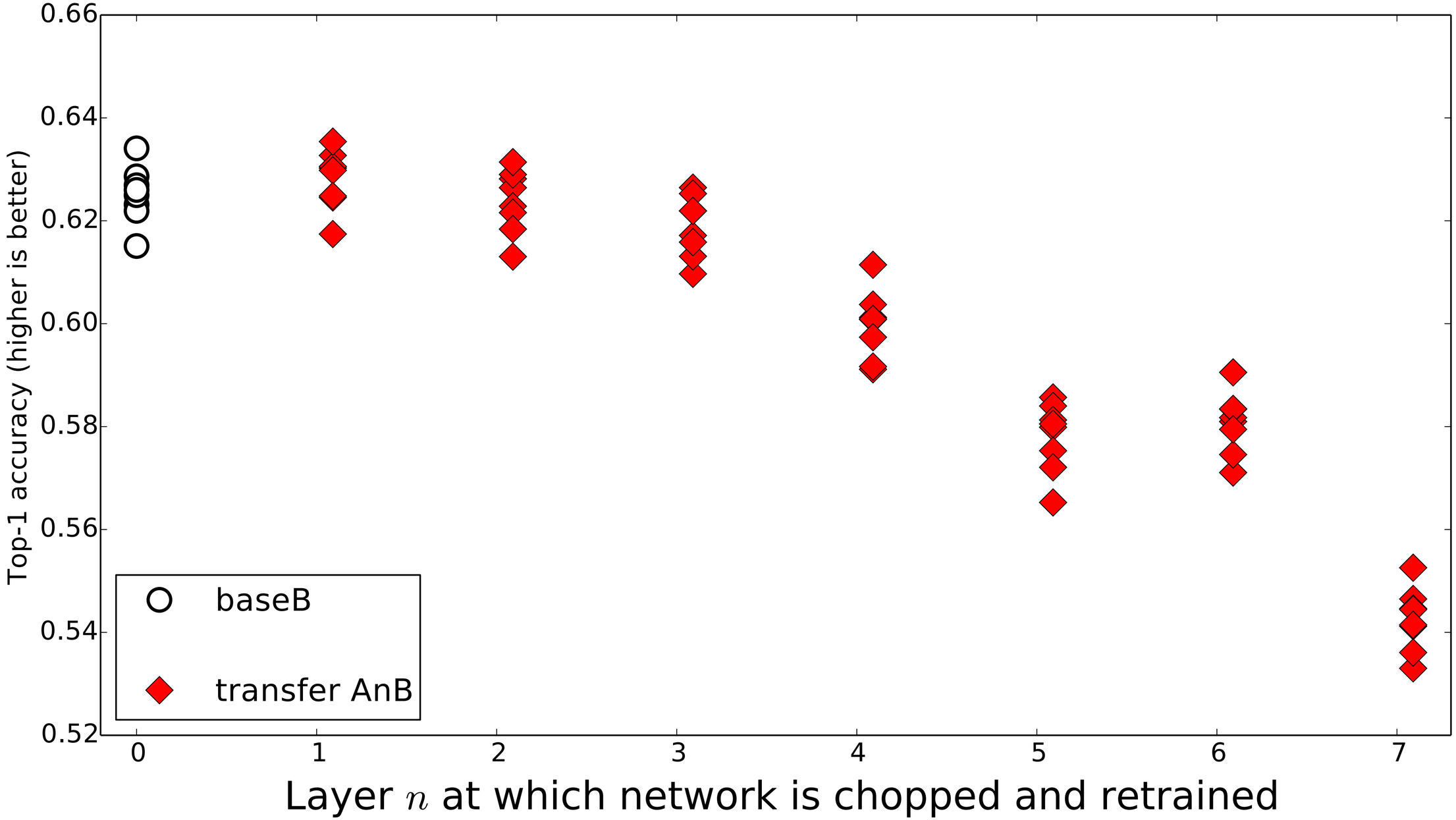

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

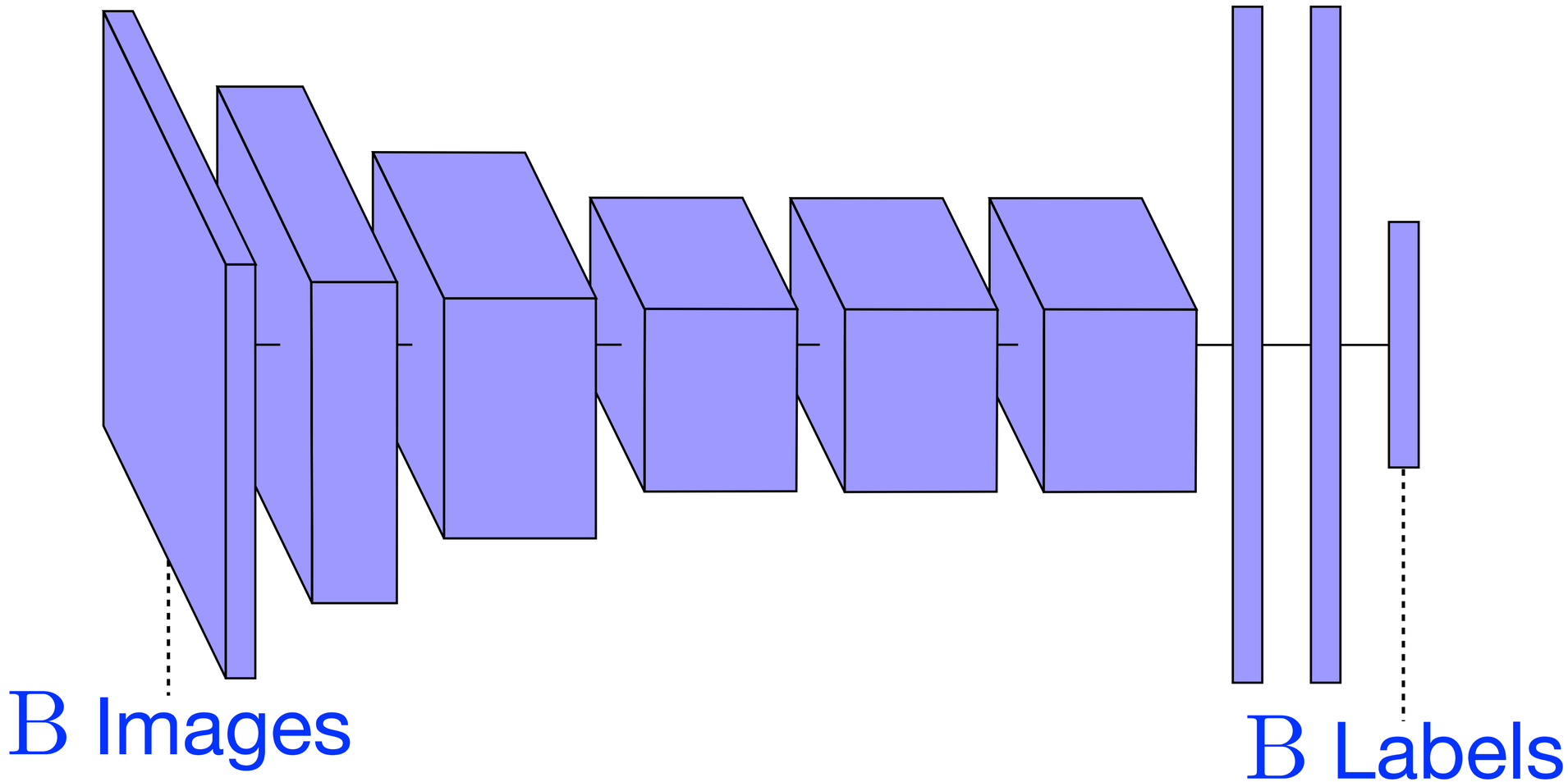

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

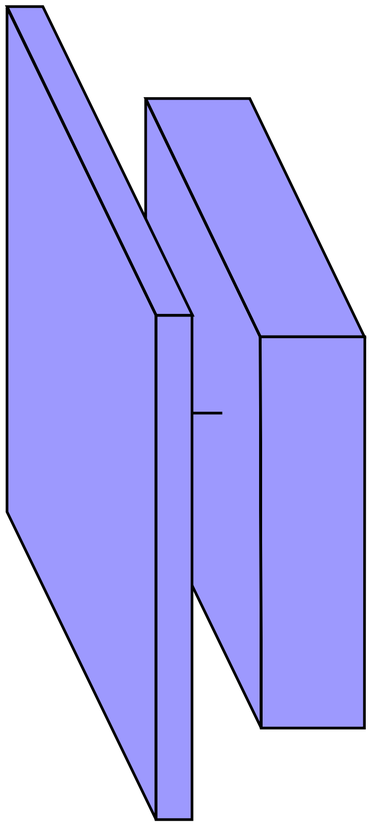

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

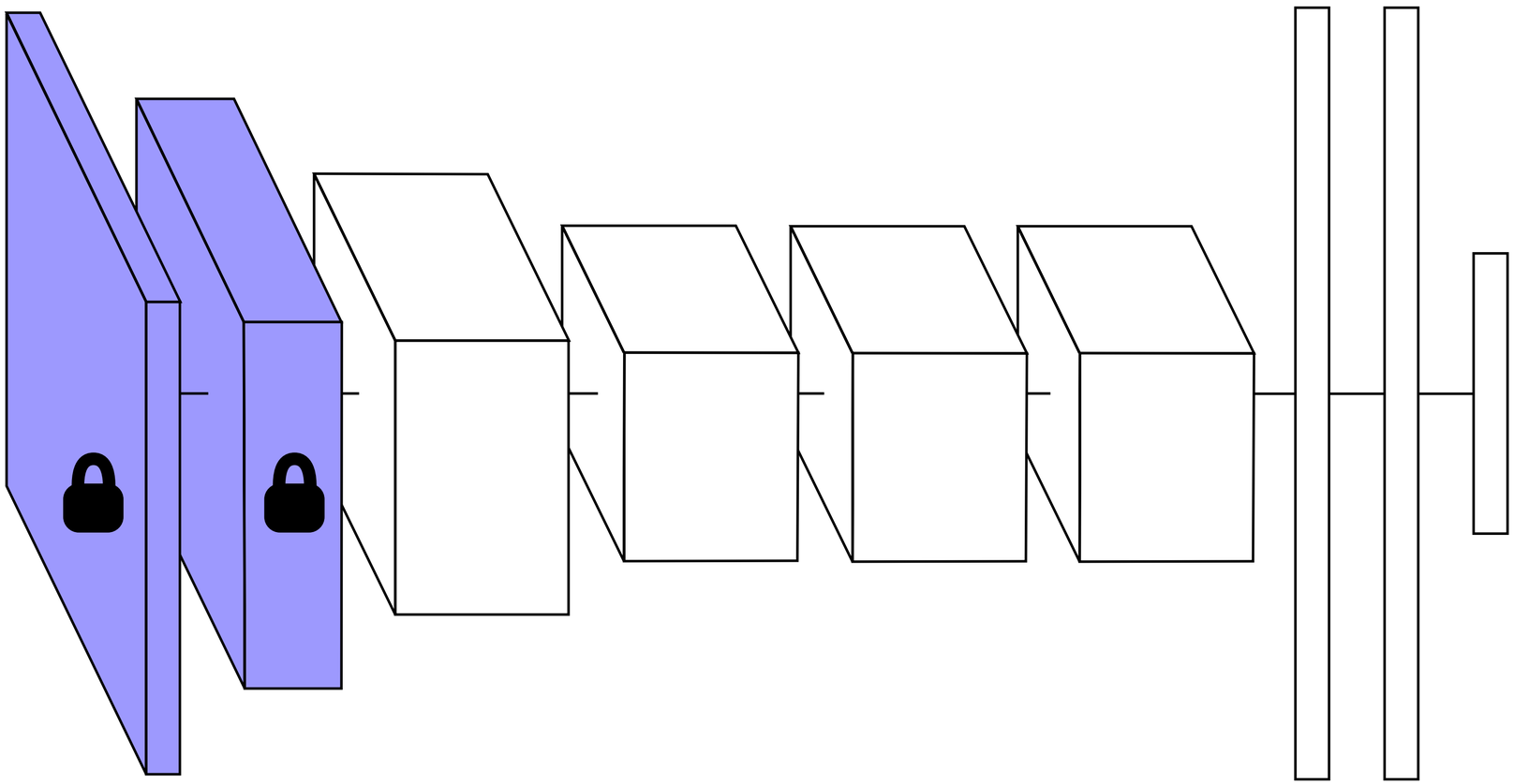

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

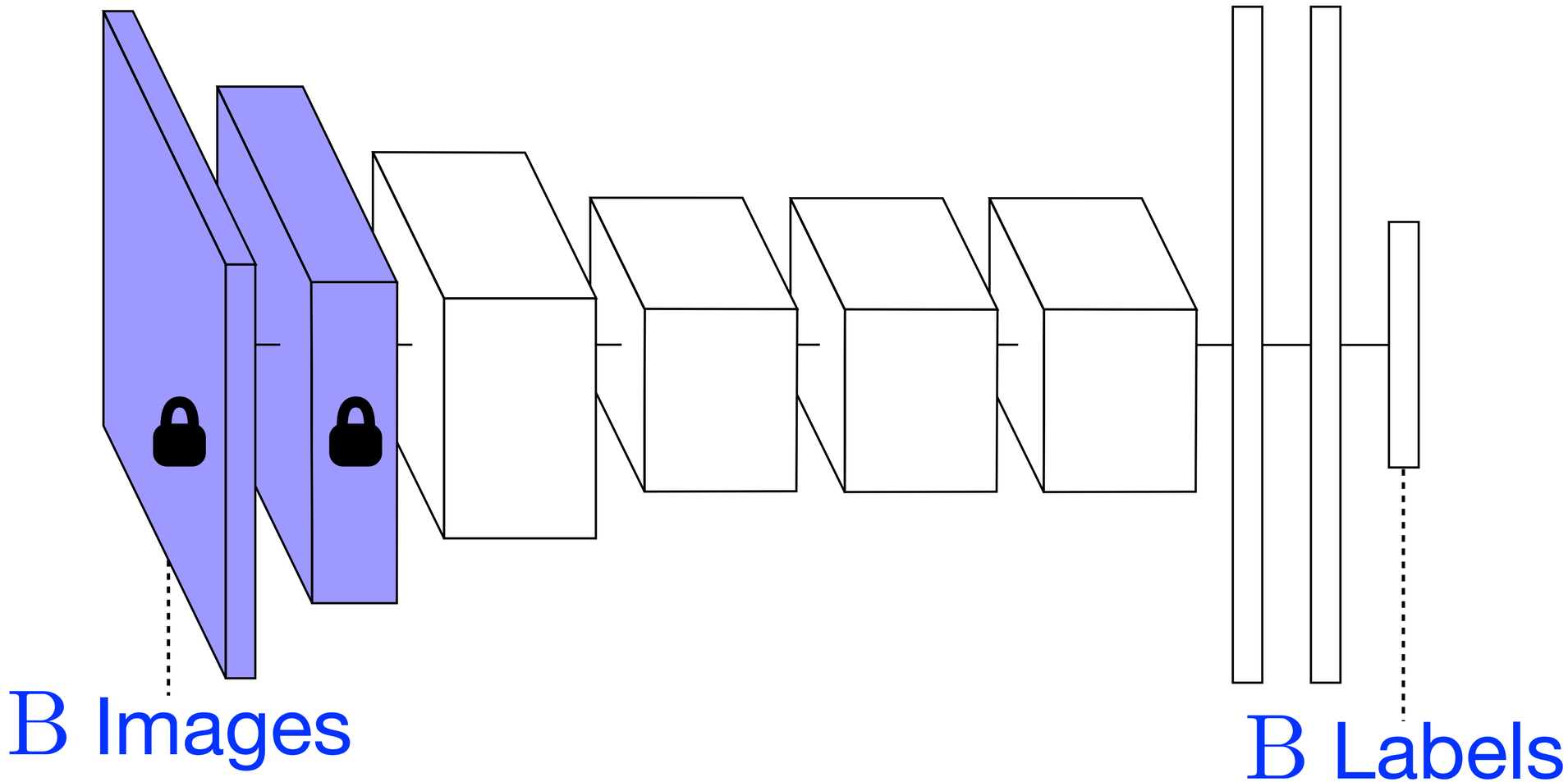

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

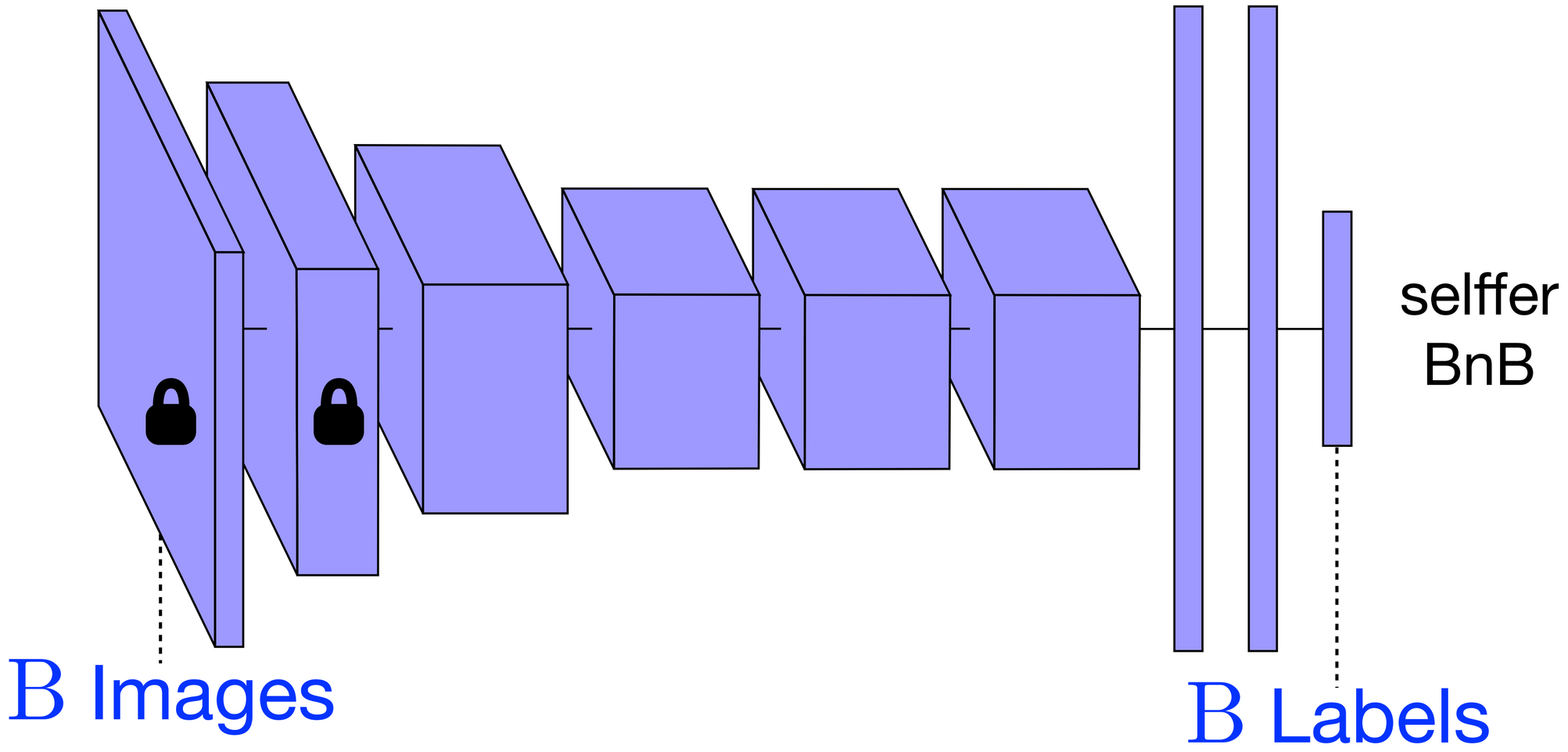

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

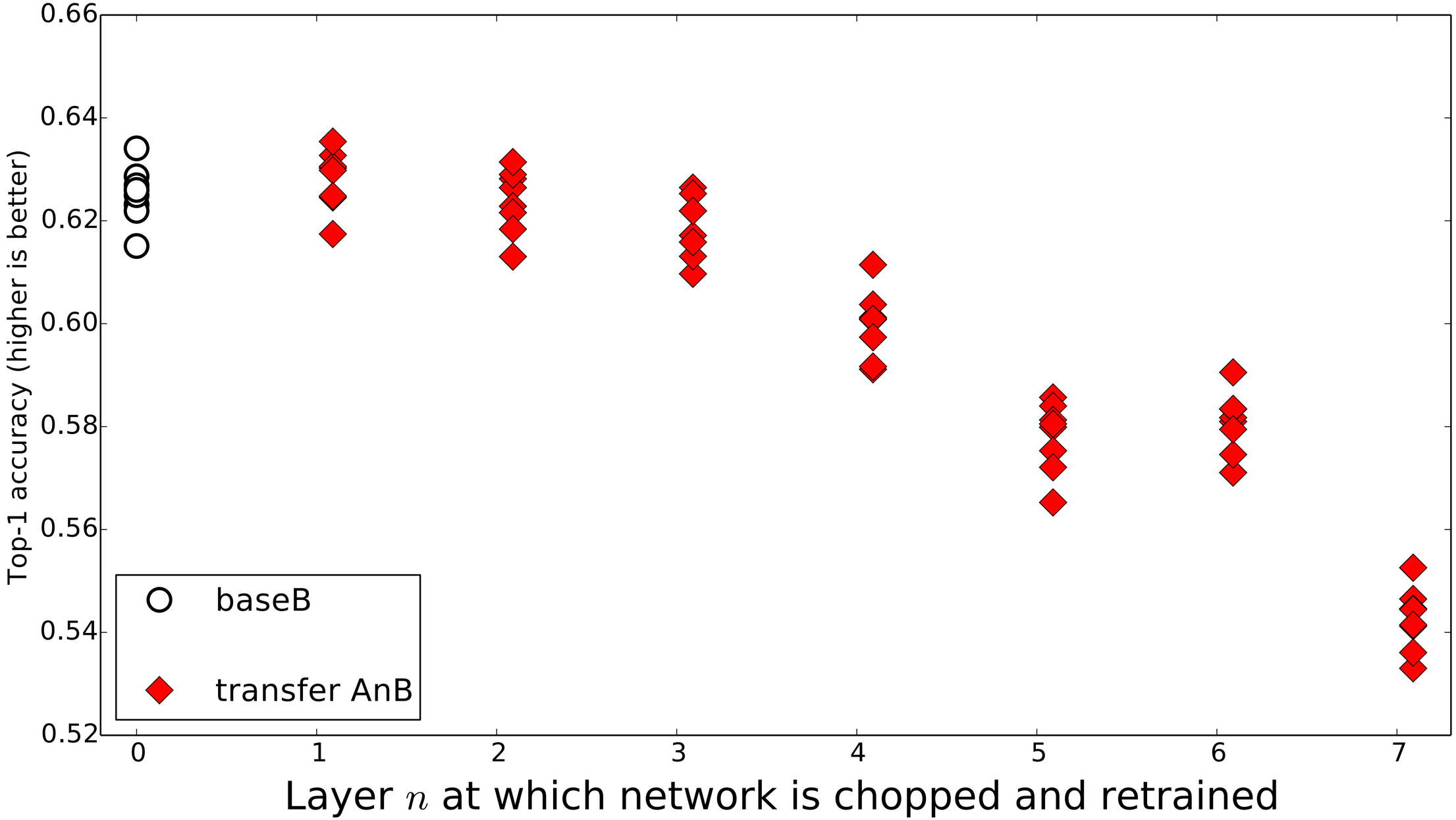

B Images
B Labels

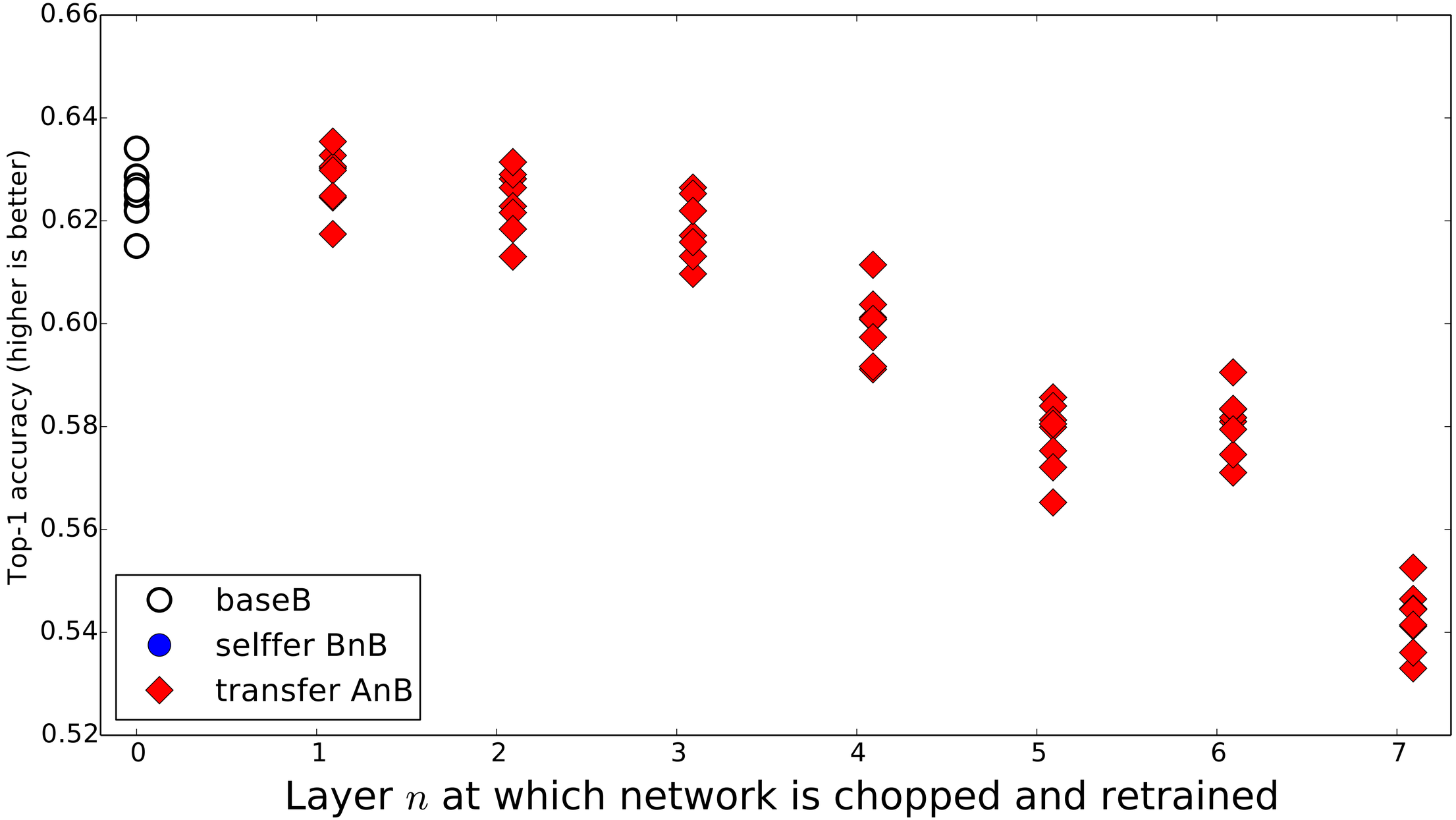


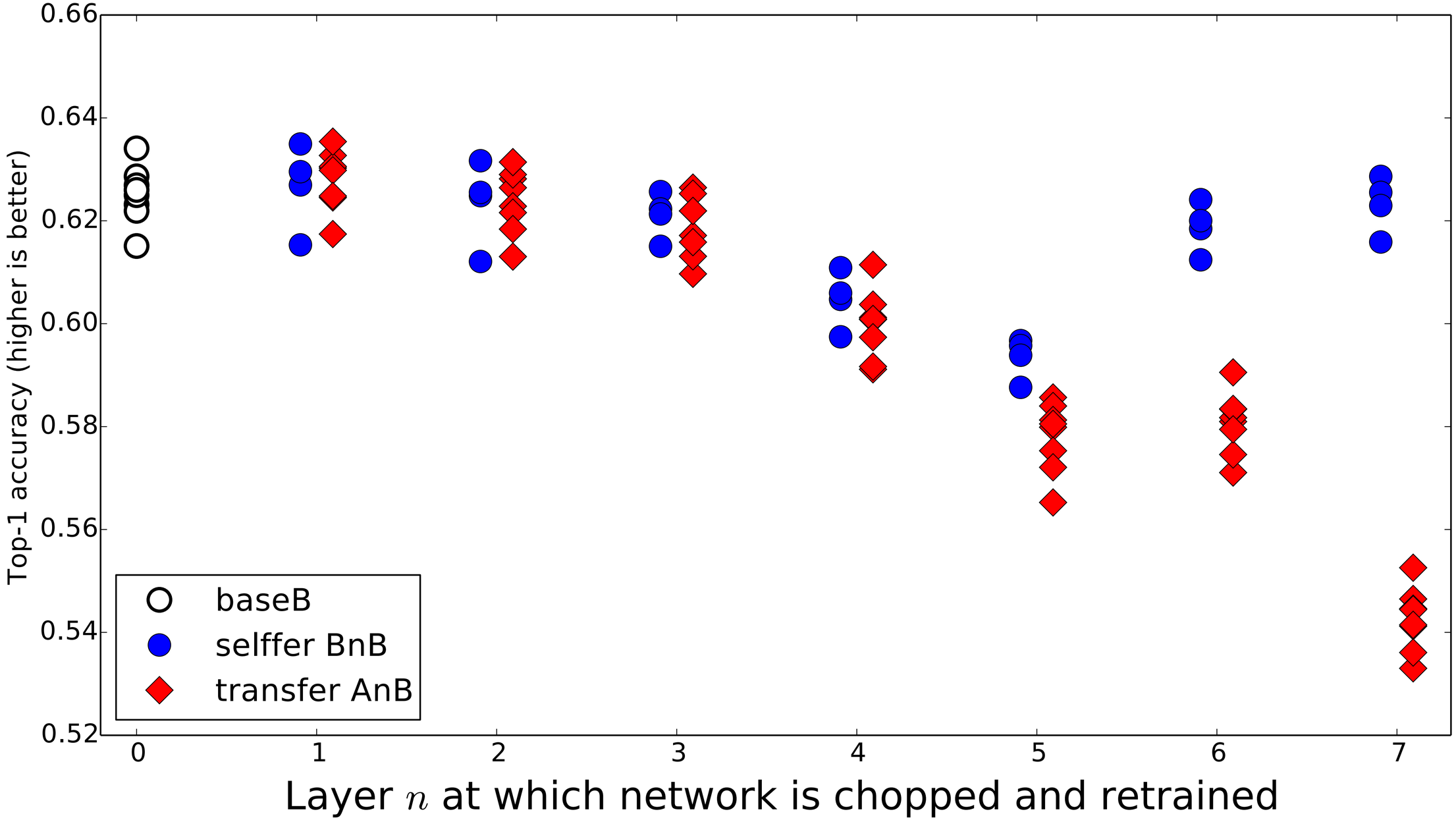


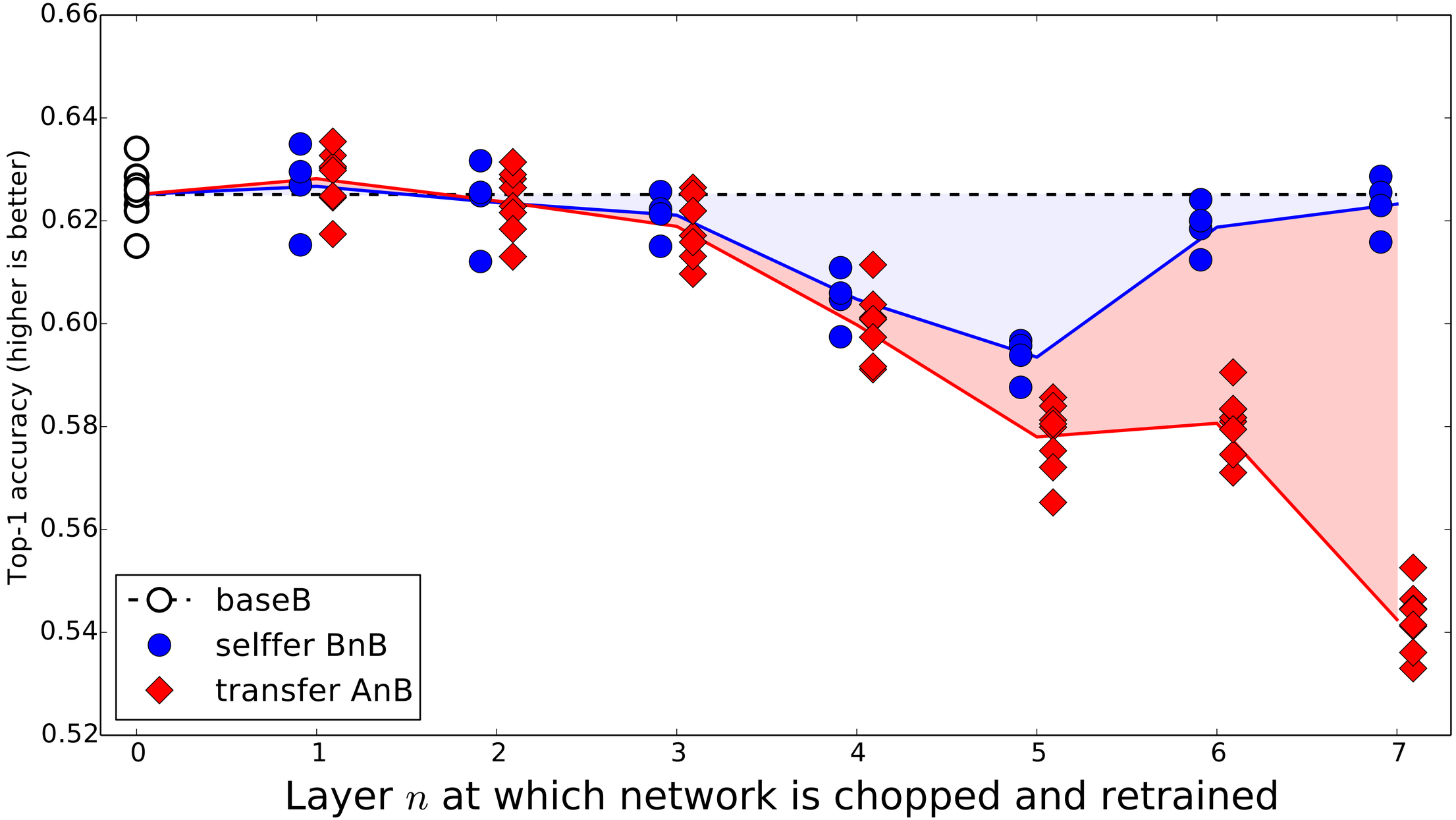

B Images
B Labels

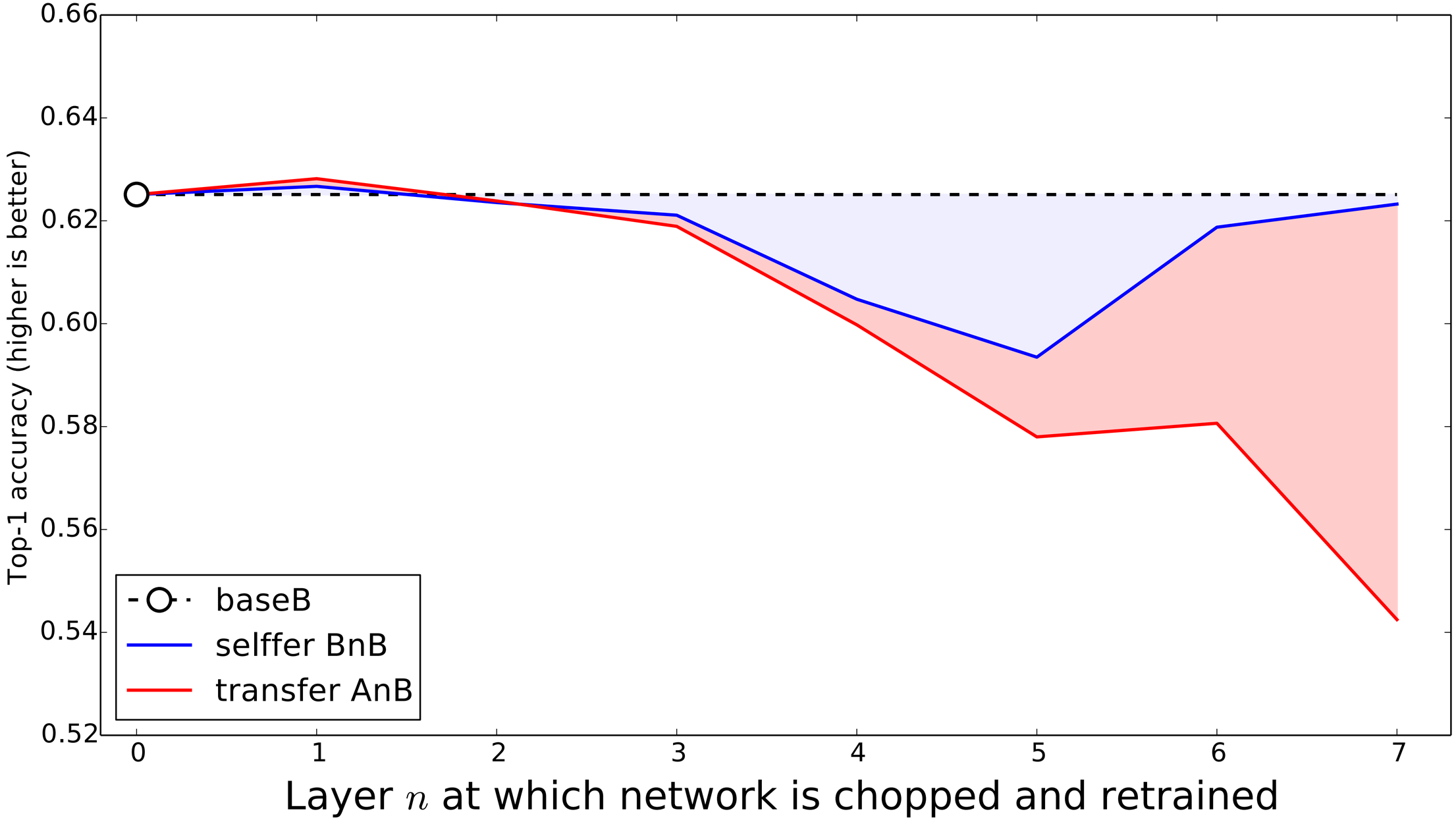

selffer
BnB
B Images
B Labels

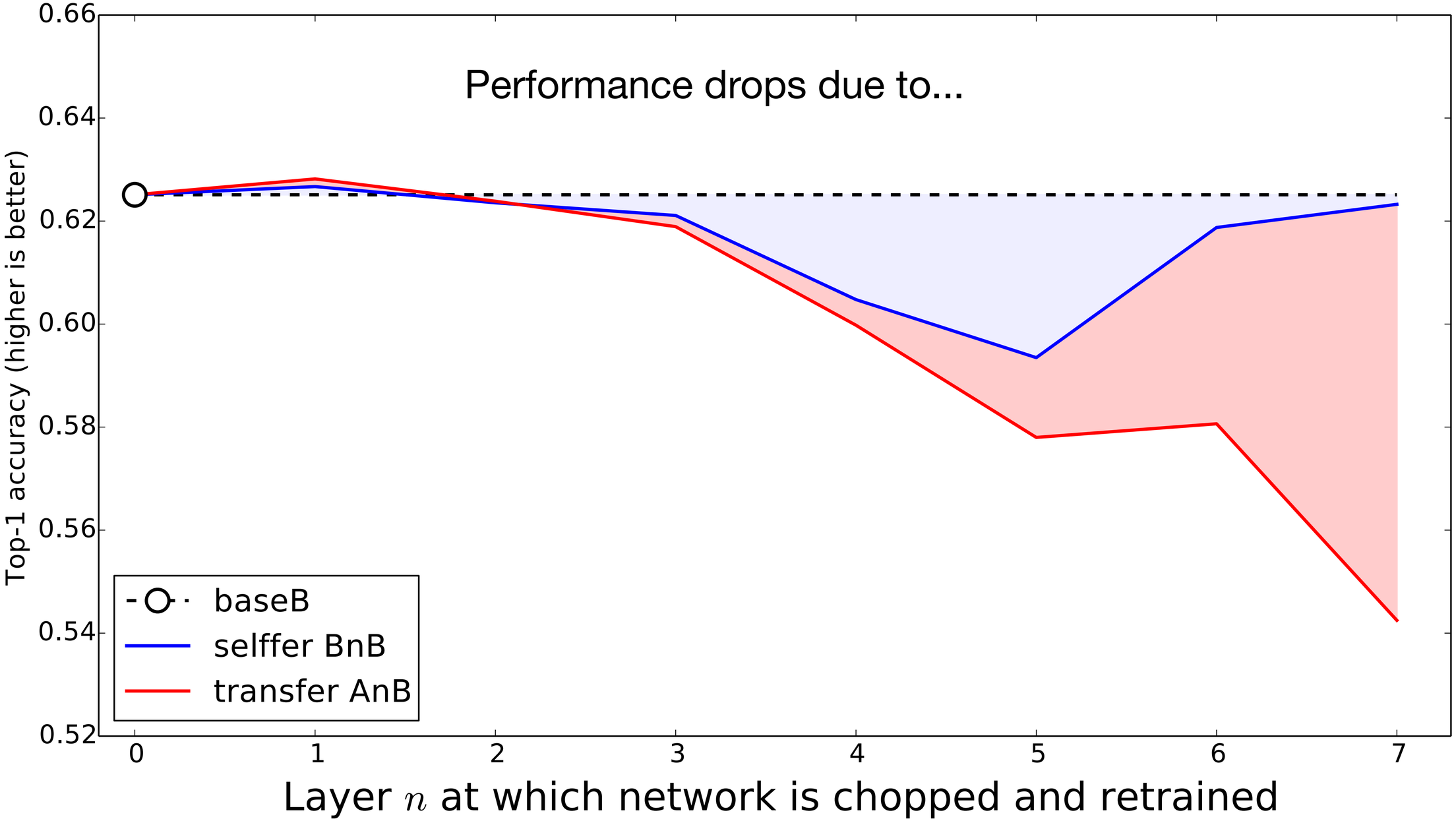

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
transfer AnB

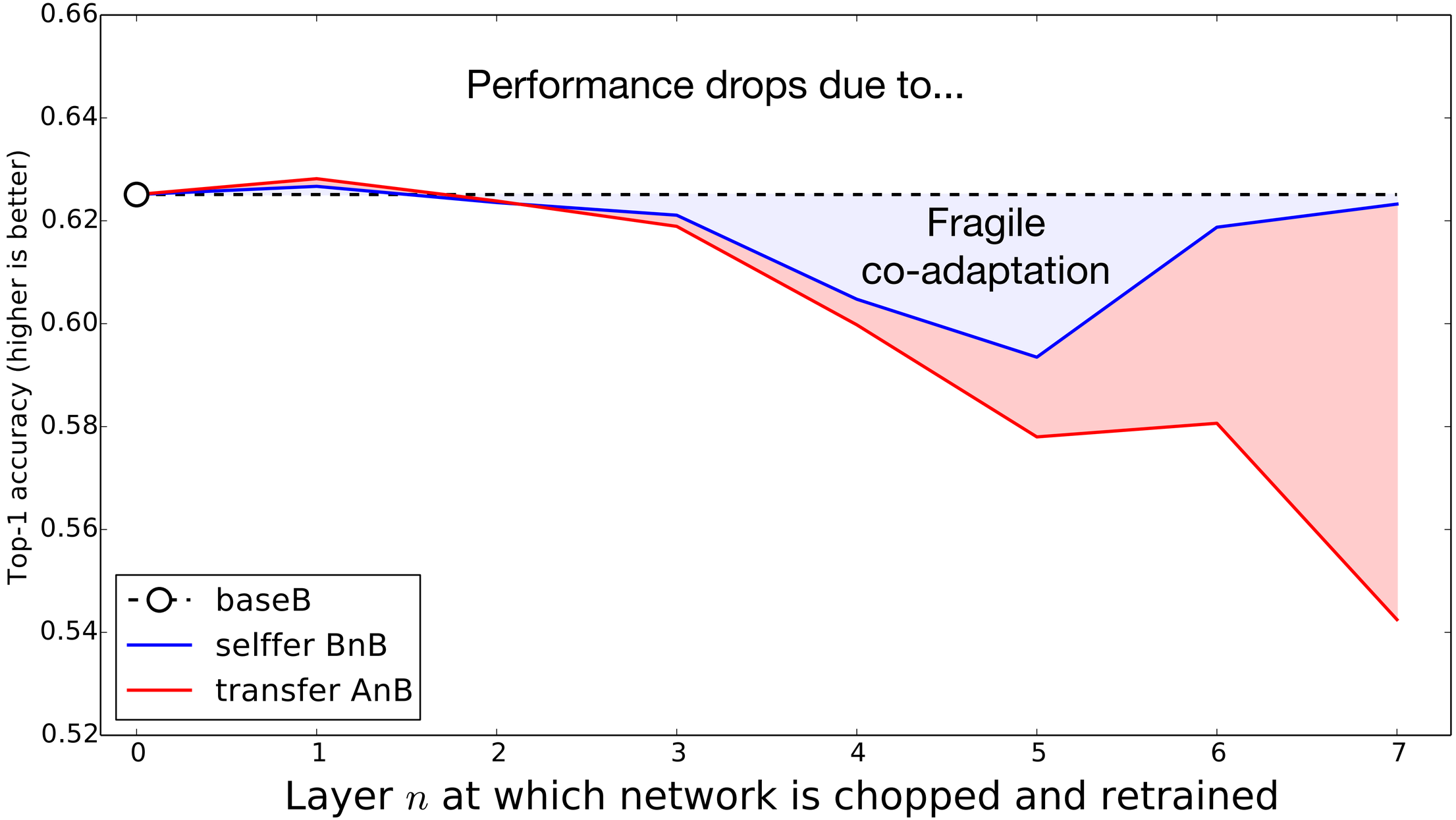

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
transfer AnB

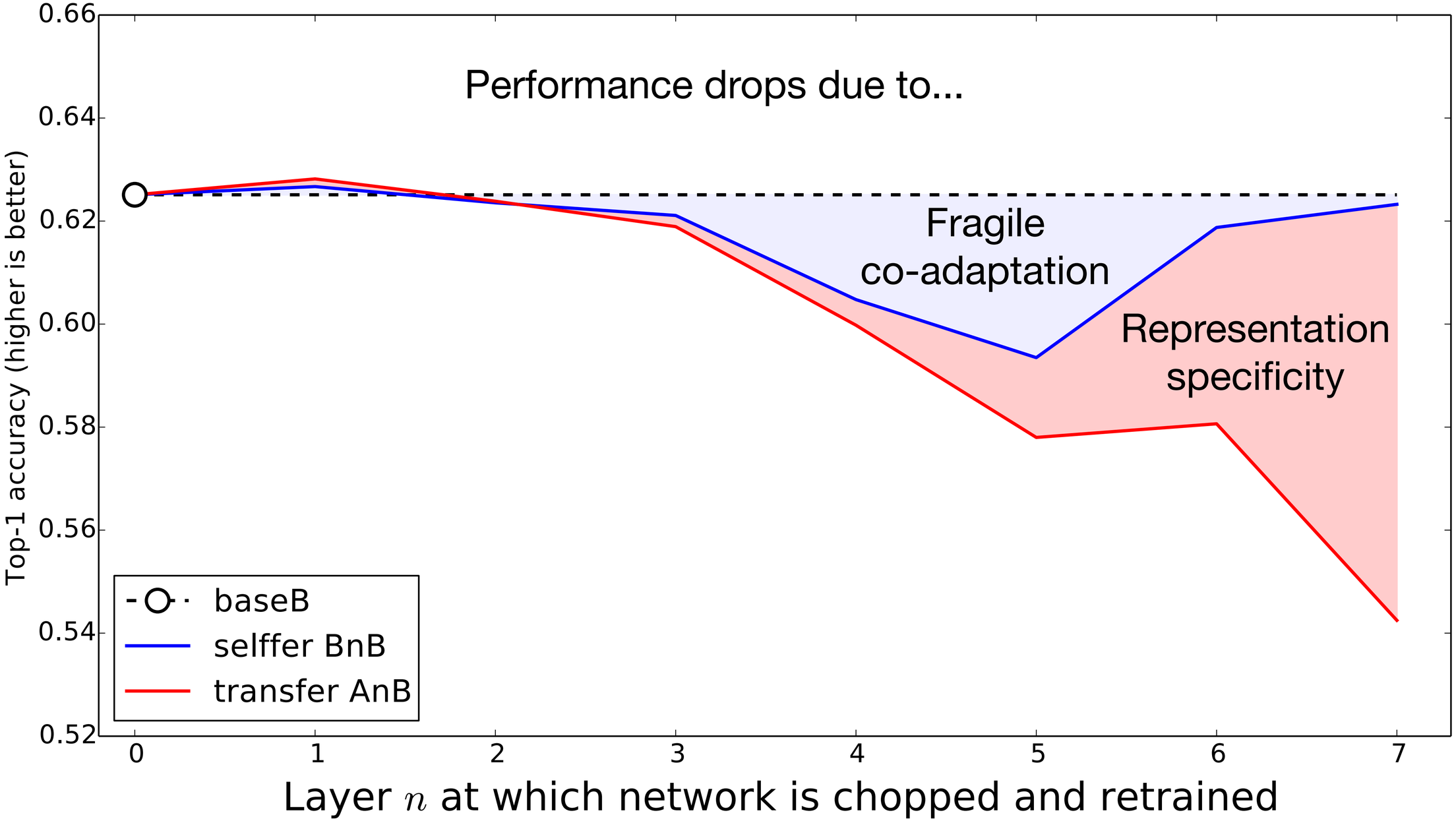

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
transfer AnB

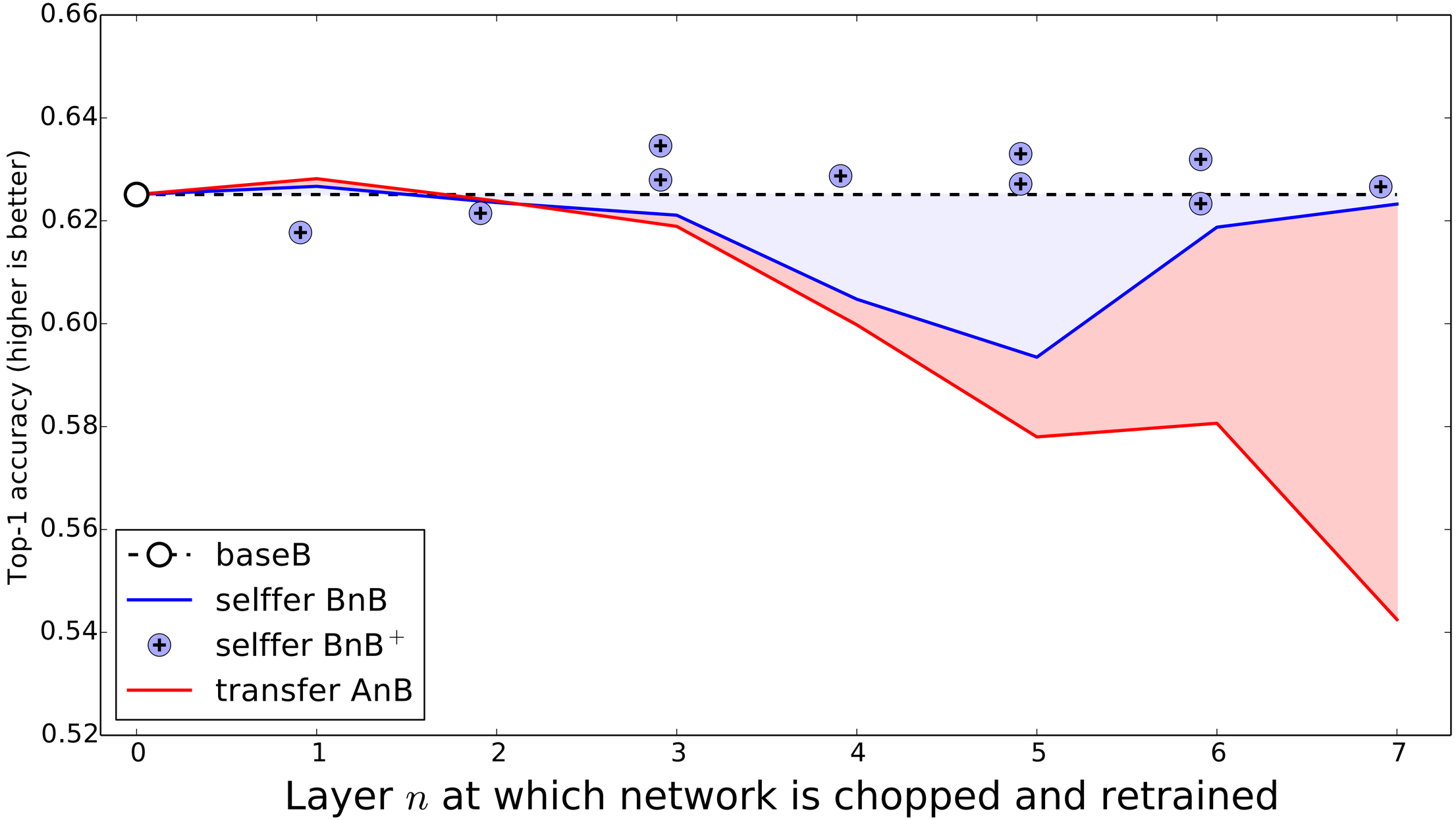

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
transfer AnB

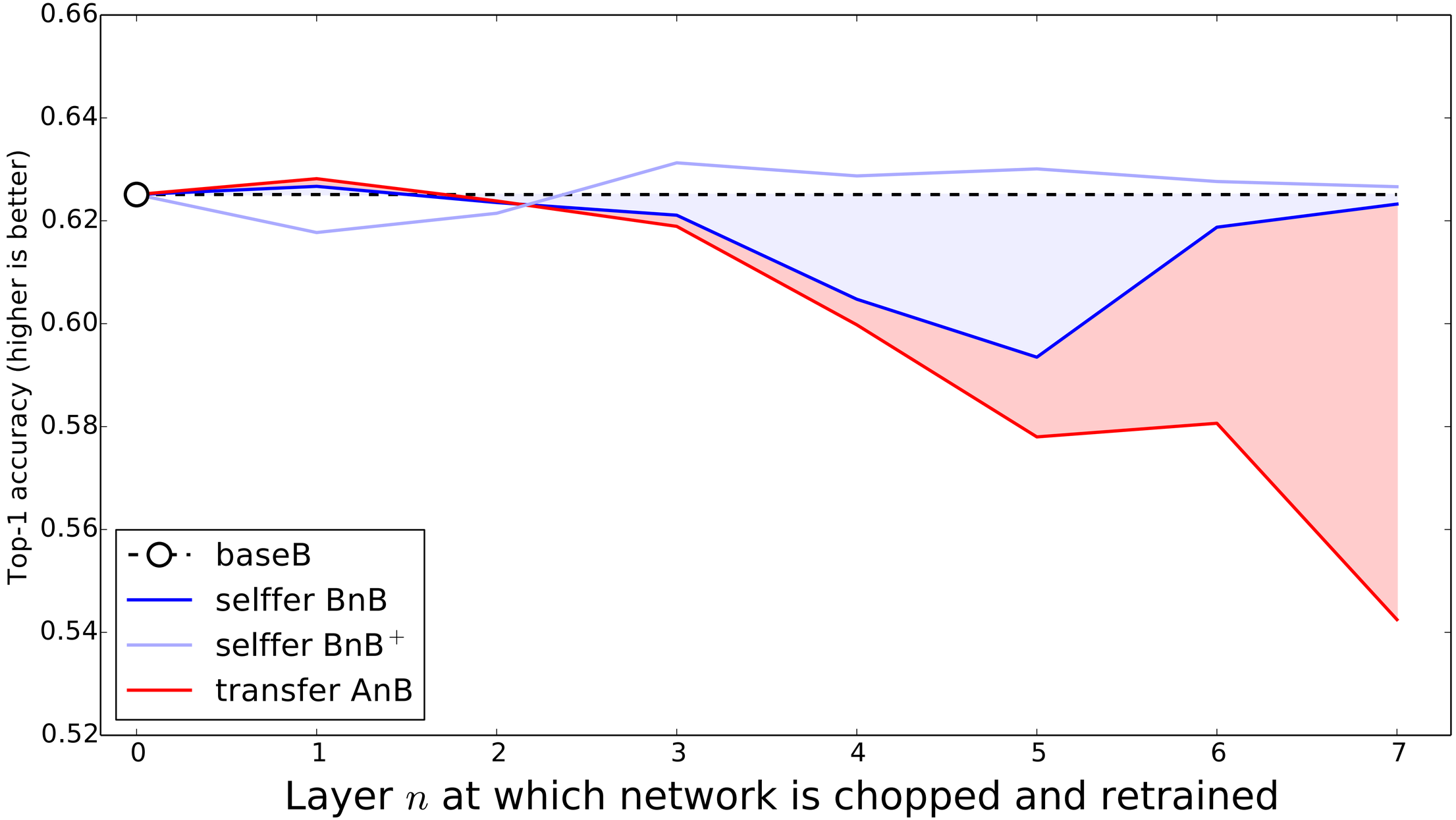

0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
transfer AnB

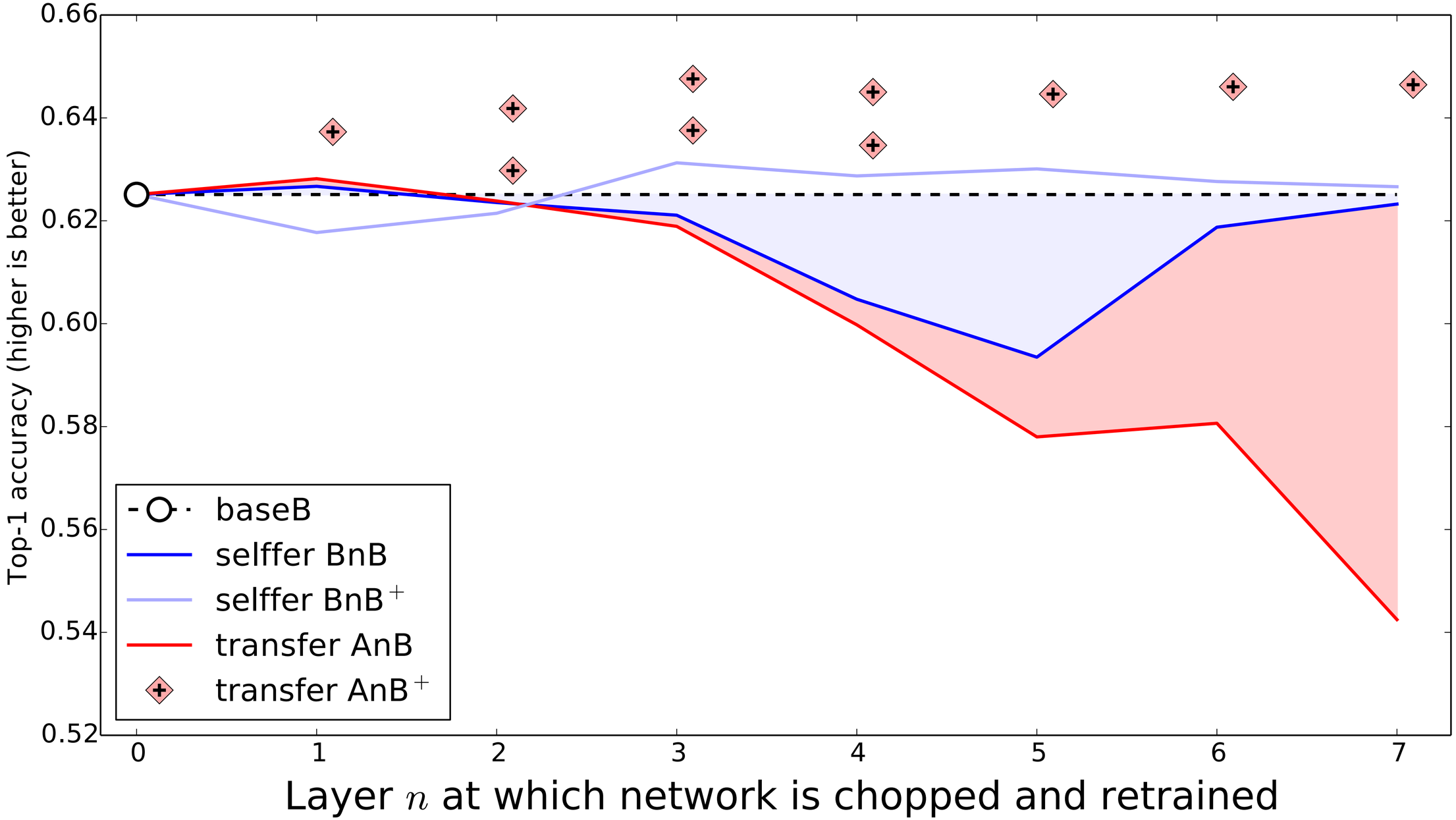

Performance drops due to...
Top-1 accuracy (higher is better)
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
transfer AnB

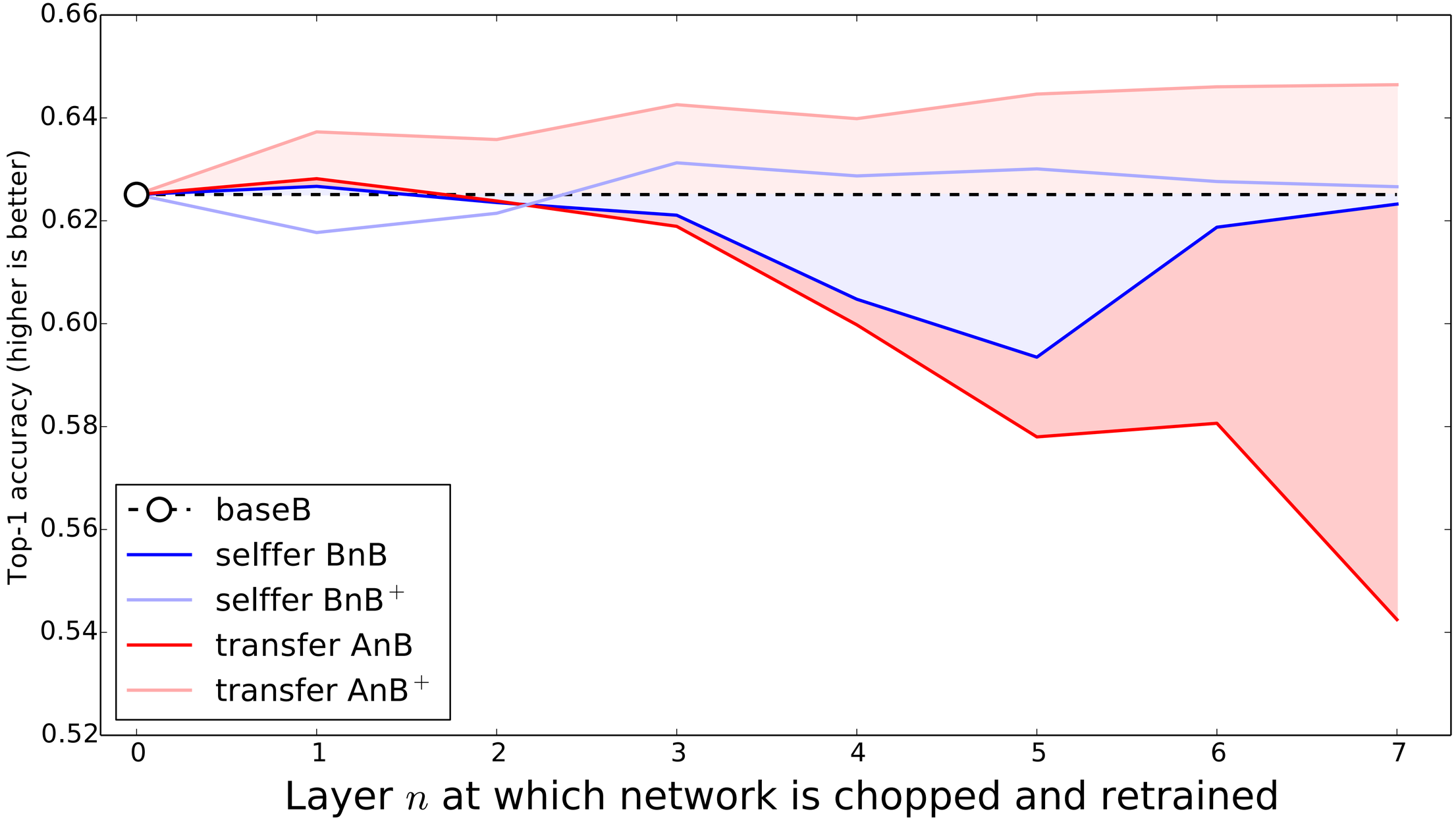

Performance drops due to...
Fragile
co-adaptation
baseB
selffer BnB
transfer AnB
Top-1 accuracy (higher is better)
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained

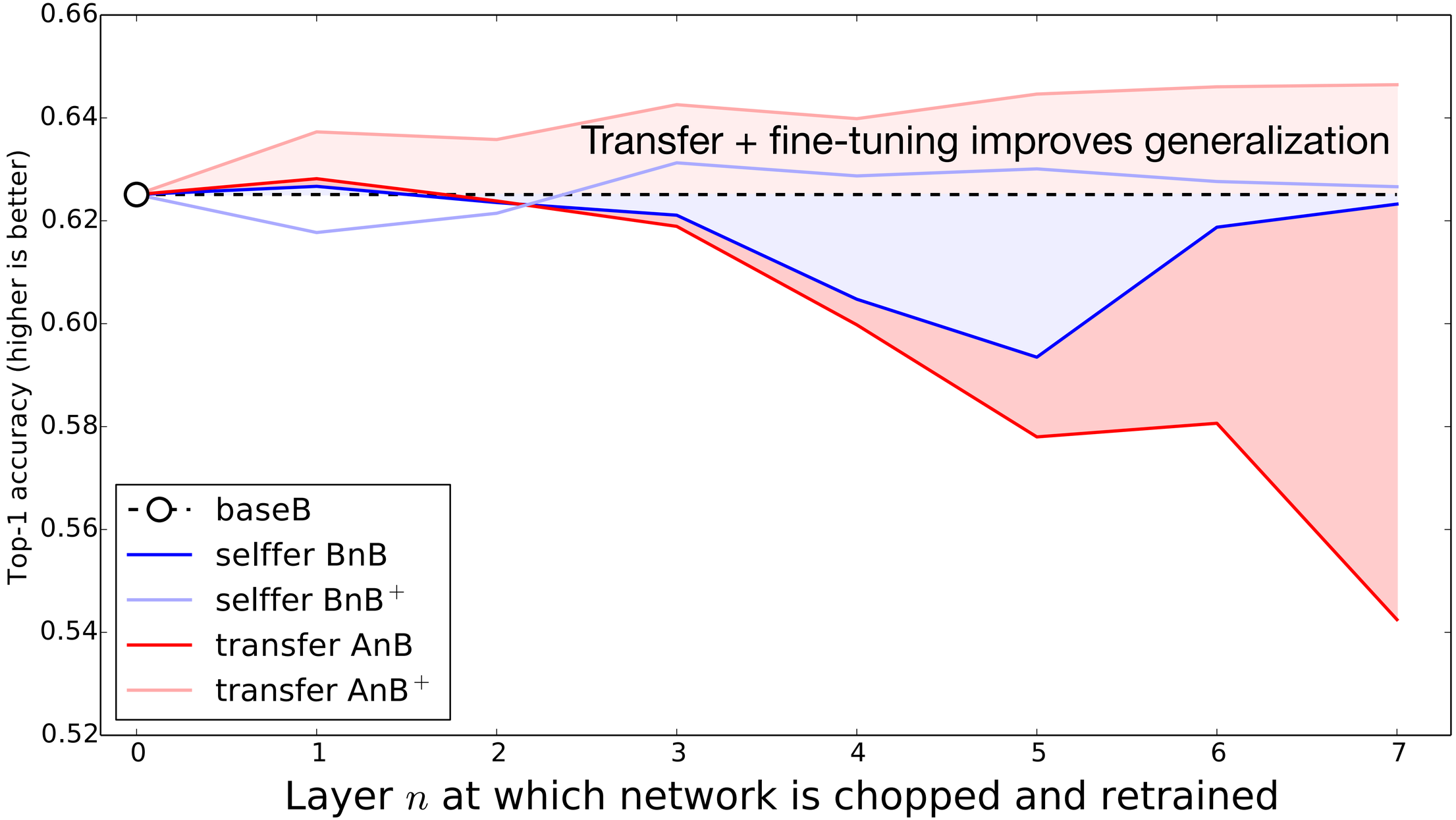

Performance drops due to...
Fragile
co-adaptation
Representation
specificity
baseB
selffer BnB
transfer AnB
Top-1 accuracy (higher is better)
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained


0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
selffer BnB$^+$
transfer AnB


0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
selffer BnB$^+$
transfer AnB


0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
selffer BnB$^+$
transfer AnB
transfer AnB$^+$


0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Top-1 accuracy (higher is better)
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
selffer BnB$^+$
transfer AnB
transfer AnB$^+$


Transfer + fine-tuning improves generalization
Top-1 accuracy (higher is better)
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
baseB
selffer BnB
selffer BnB$^+$
transfer AnB
transfer AnB$^+$

ImageNet has many related categories...

Dataset A: random

gecko

fire truck

baseball

panther

rabbit

gorilla

Dataset B: random

garbage truck

toucan

radiator

binoculars

lion

bookshop

ImageNet has many related categories...

Dataset A: man-made

garbage truck

fire truck

radiator

baseball

binoculars

bookshop

Dataset B: natural

gorilla

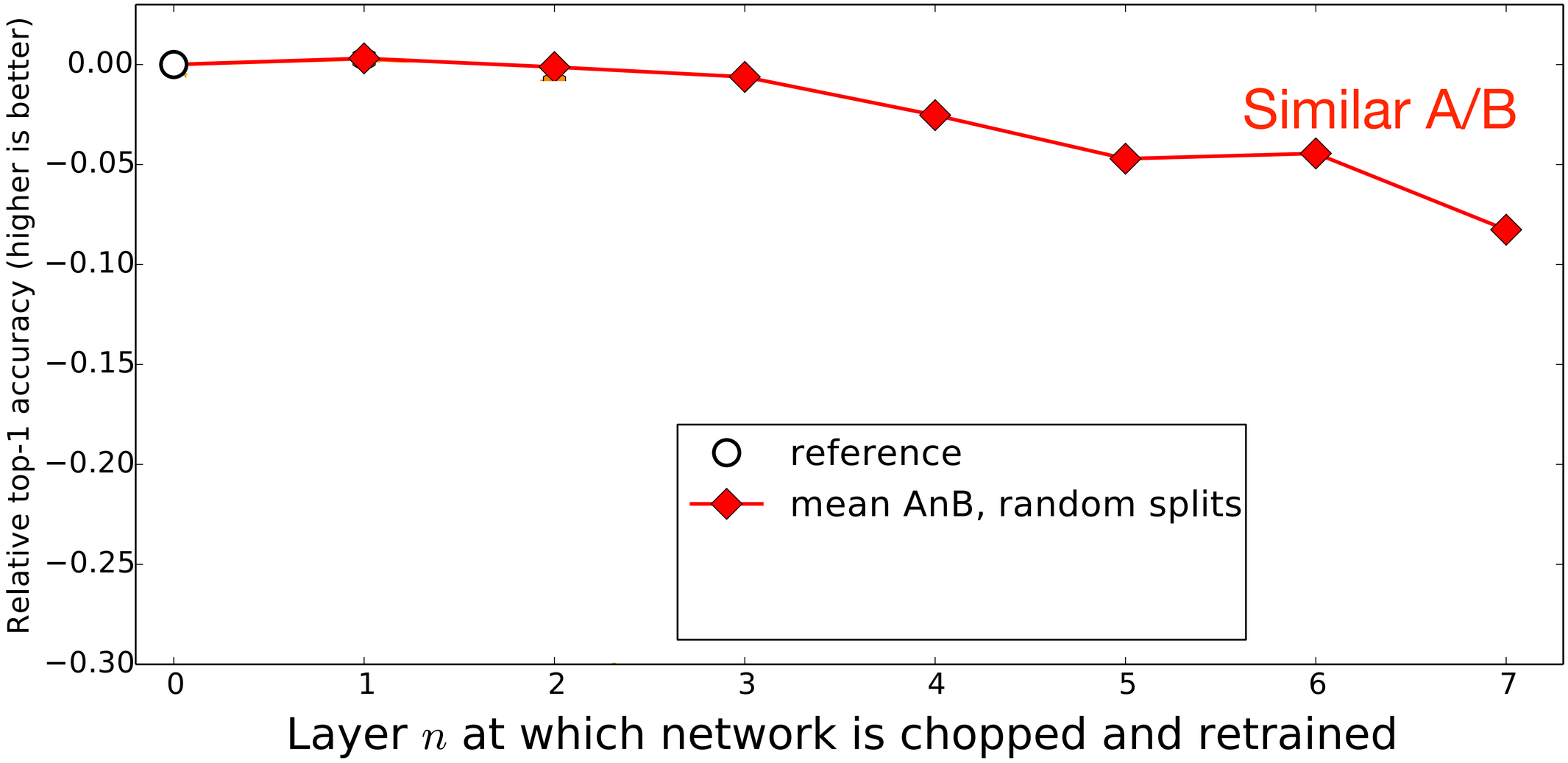
gecko

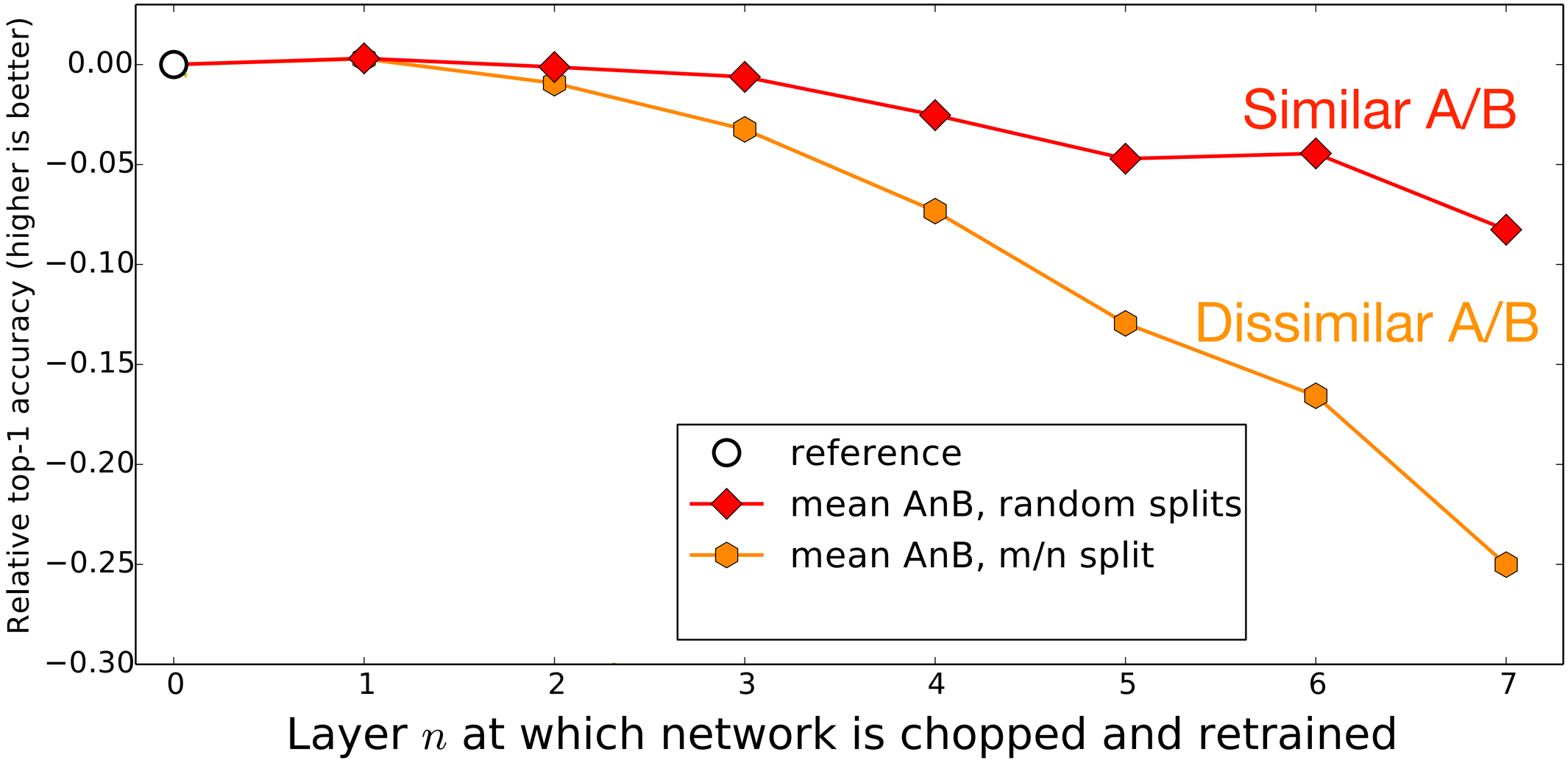
toucan

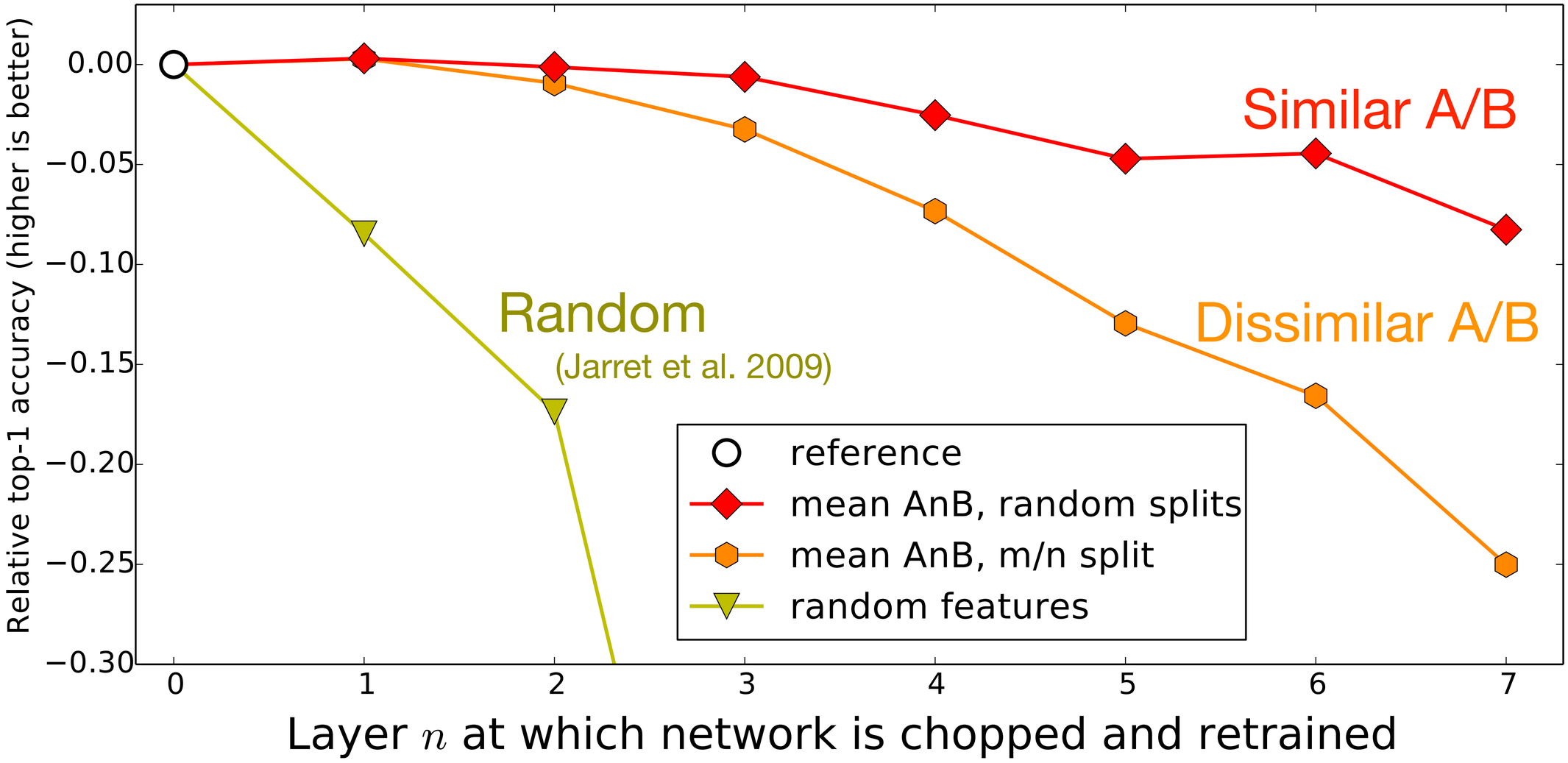
rabbit

panther

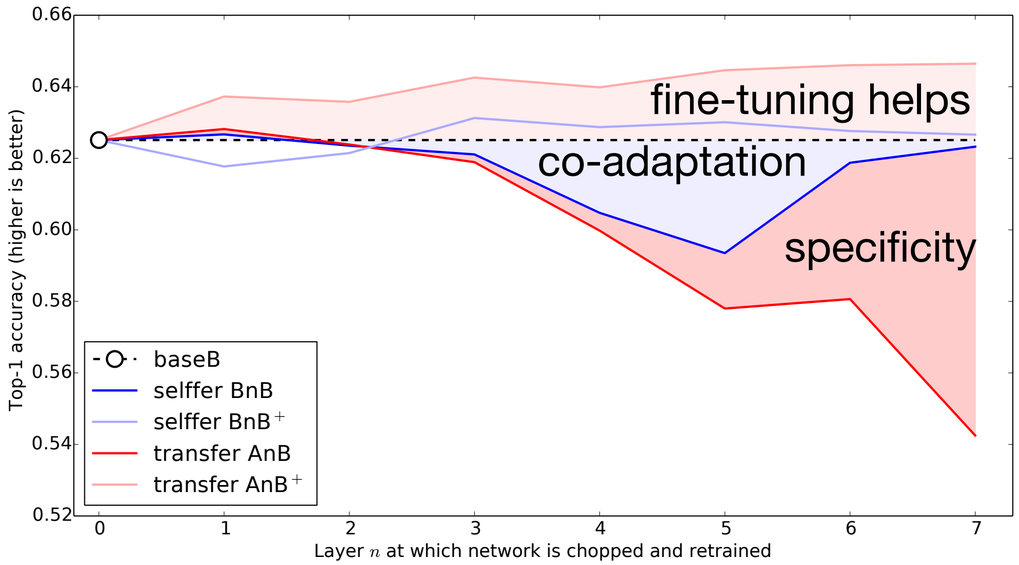
lion

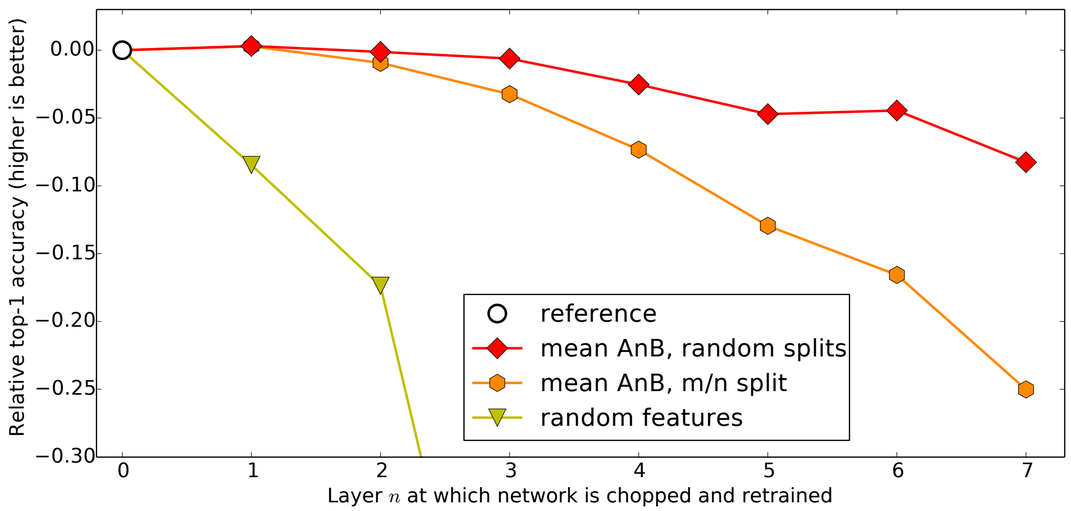

Similar A/B
reference
mean AnB, random splits
Relative top-1 accuracy (higher is better)
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained


Relative top-1 accuracy (higher is better)
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
Similar A/B
Dissimilar A/B
reference
mean AnB, random splits
mean AnB, m/n split

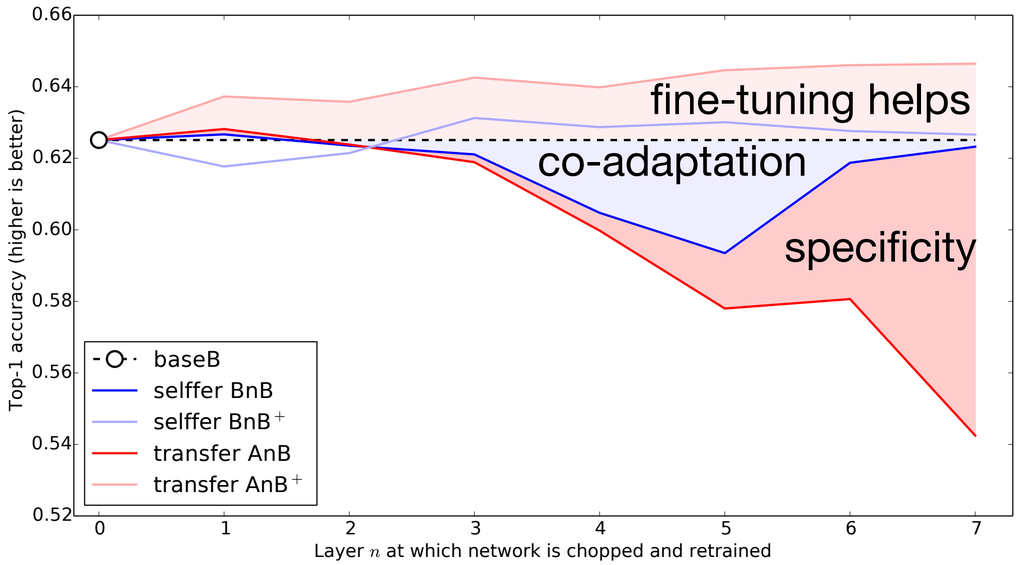

Relative top-1 accuracy (higher is better)
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
0
1
2
3
4
5
6
7
Layer n at which network is chopped and retrained
Similar A/B
Dissimilar A/B
Random
(Jarret et al. 2009)
reference
mean AnB, random splits
mean AnB, m/n split
random features

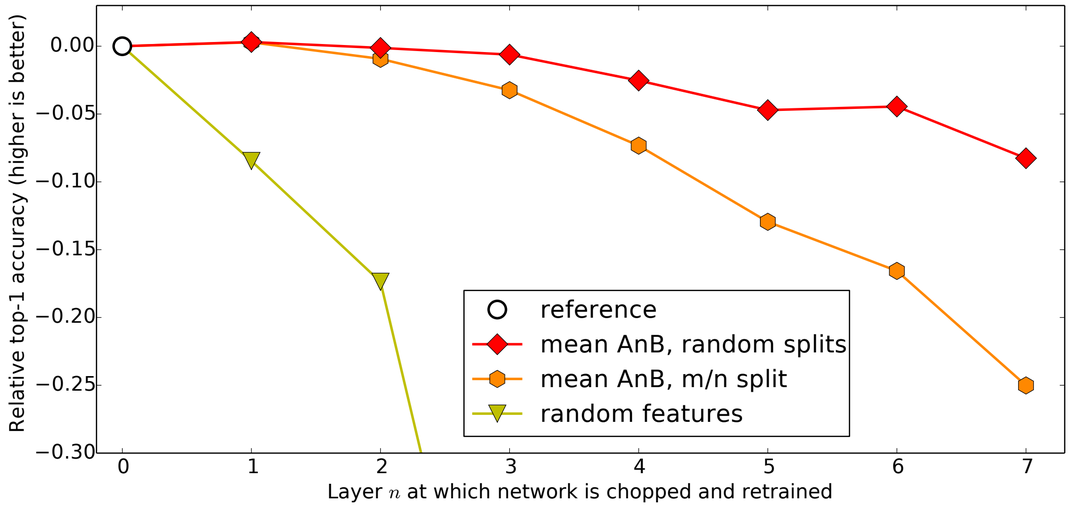
Conclusions





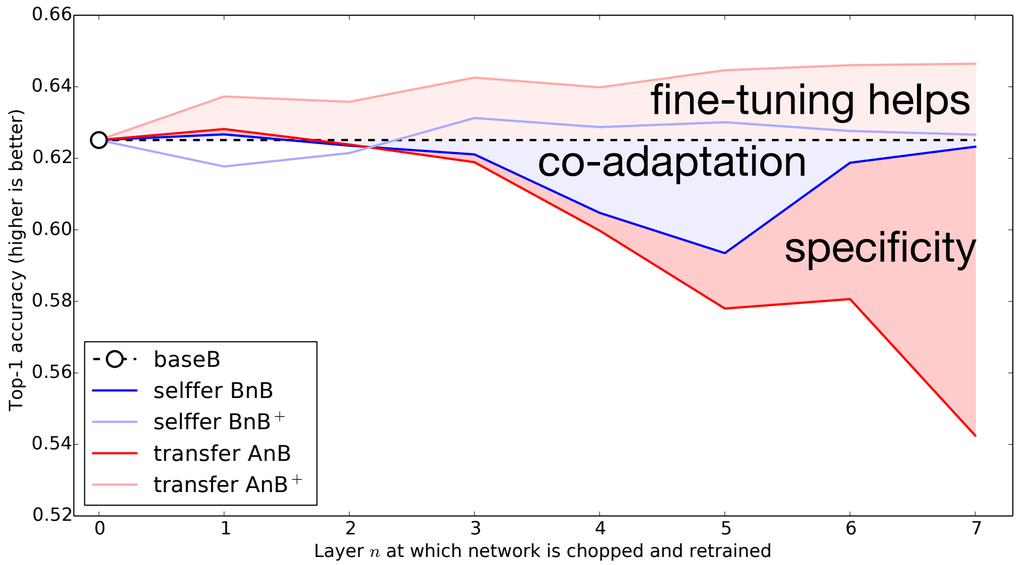
Conclusions

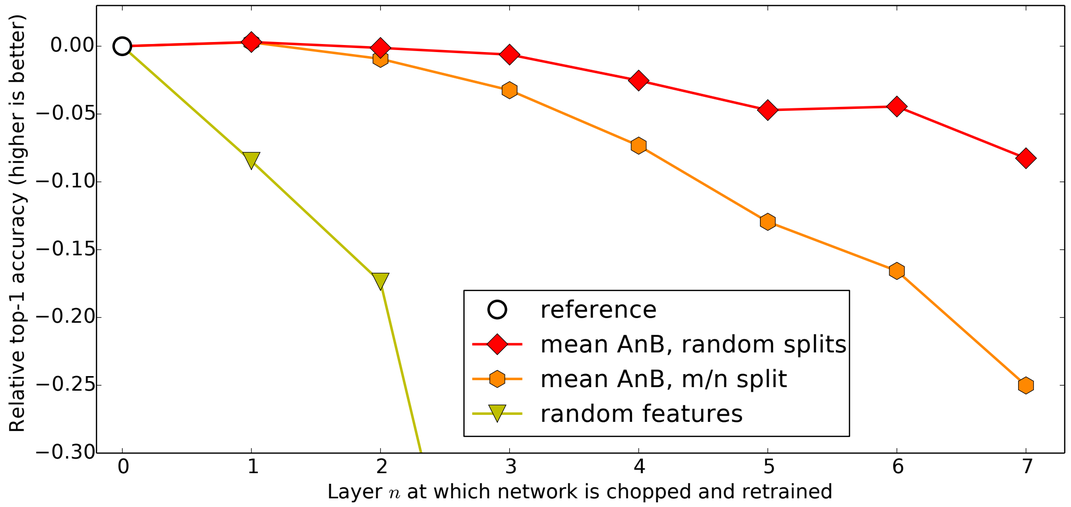




- Measure *general* to *specific* transition layer by layer

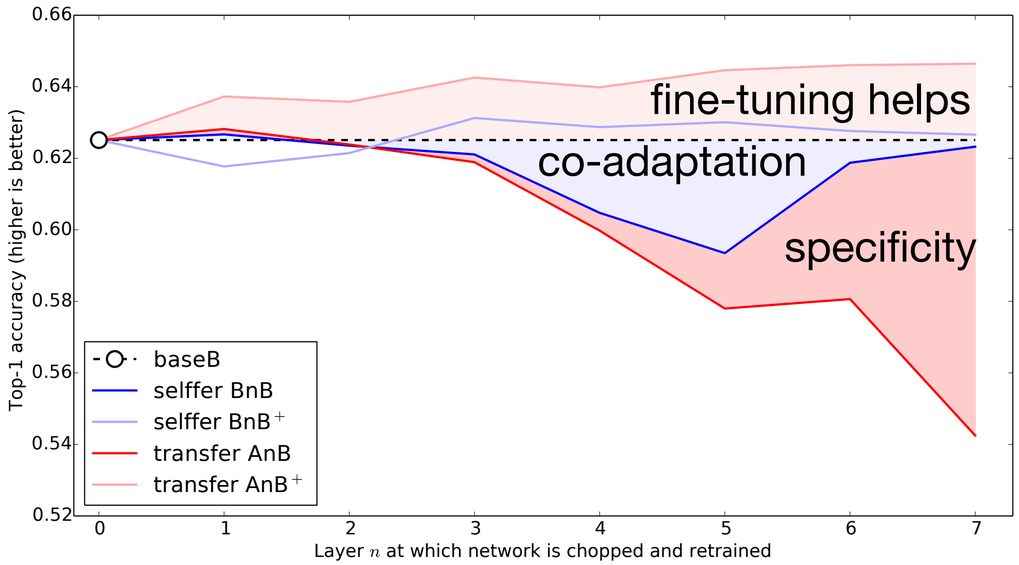
Conclusions

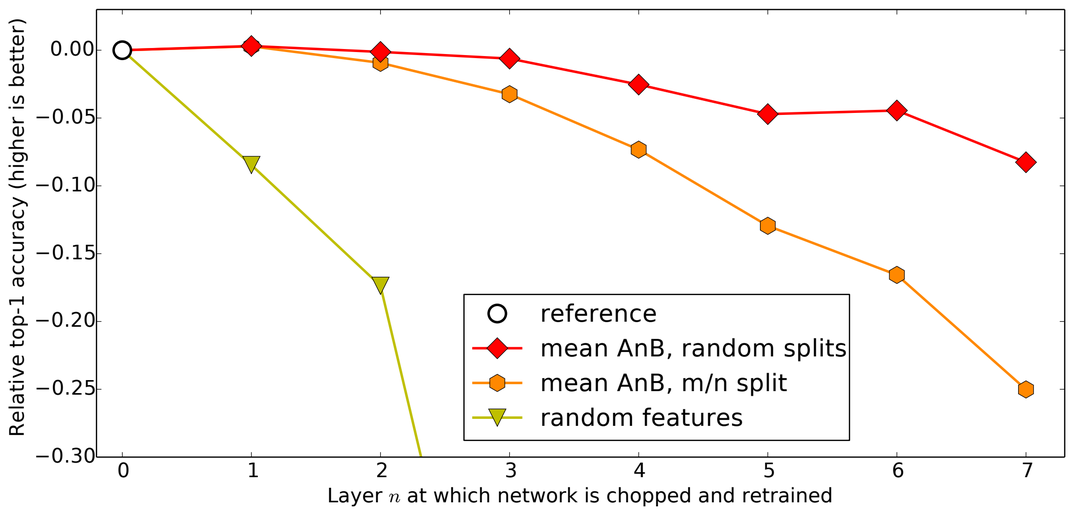




- Measure *general* to *specific* transition layer by layer
- Transferability governed by:

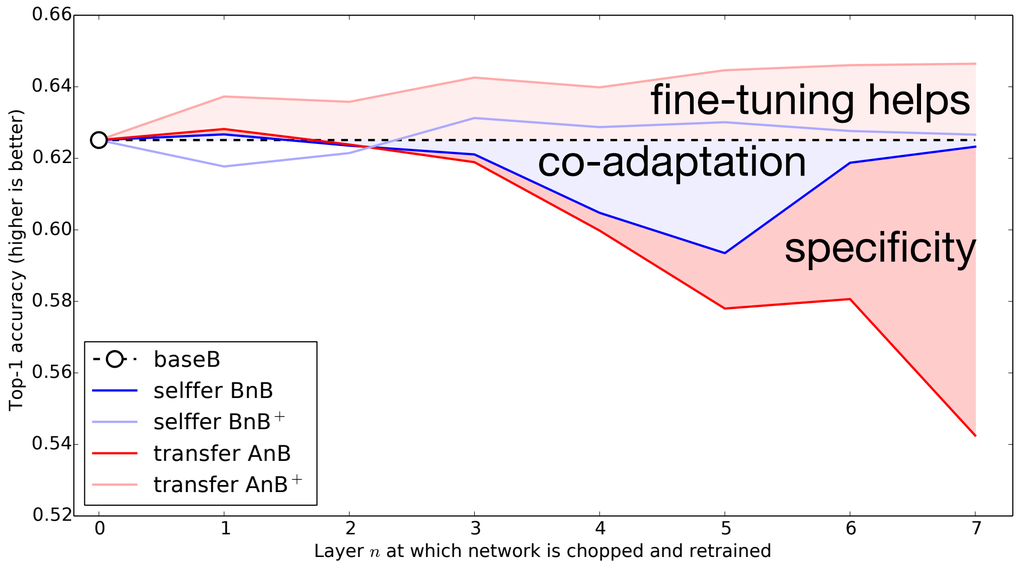
Conclusions

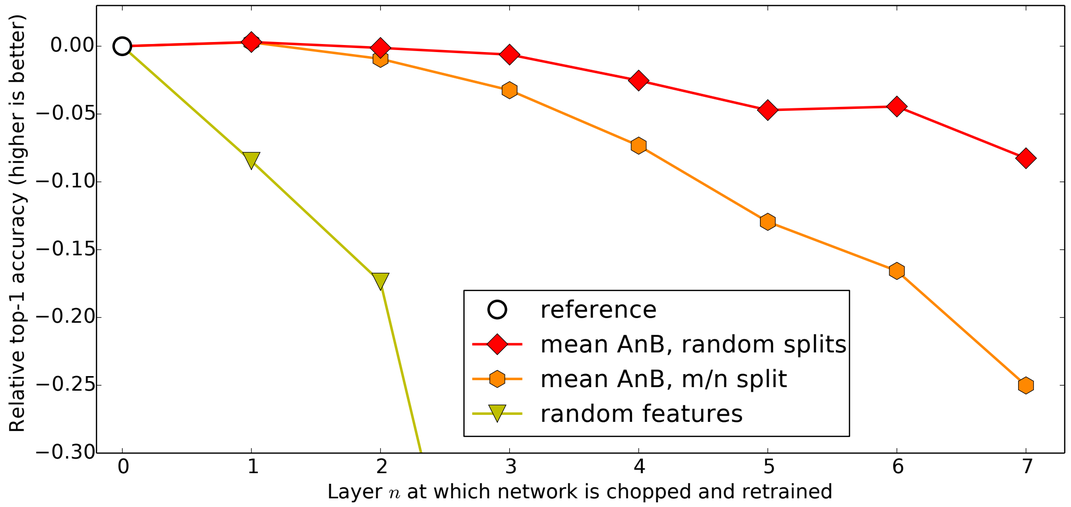




- Measure *general* to *specific* transition layer by layer
- Transferability governed by:
  - lost co-adaptations

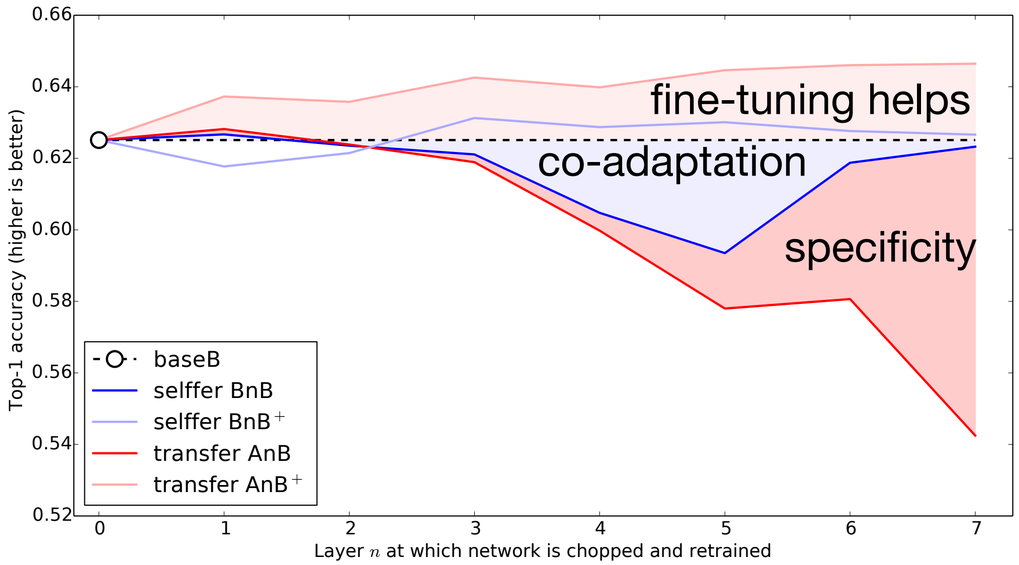
Conclusions

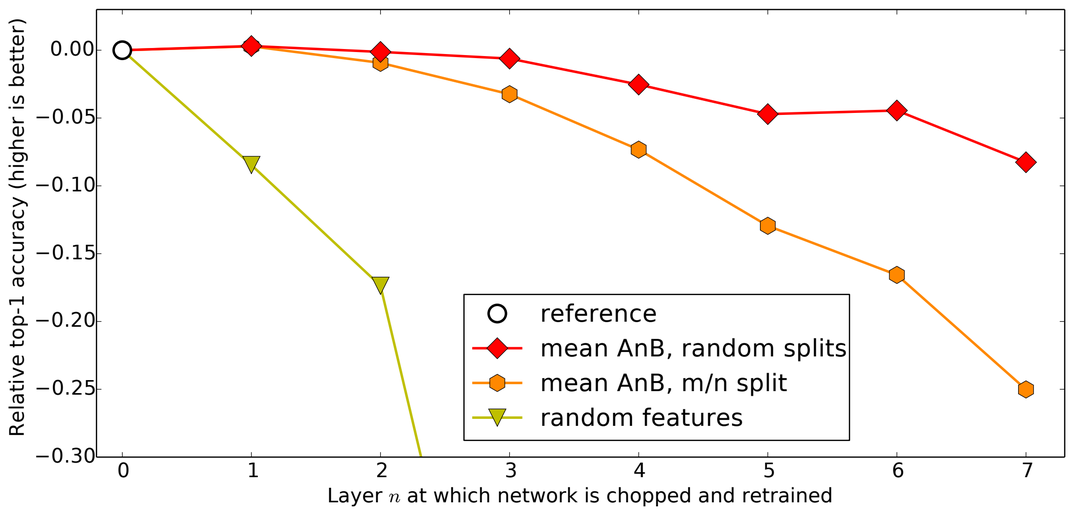




- Measure *general* to *specific* transition layer by layer
- Transferability governed by:
  - lost co-adaptations
  - specificity

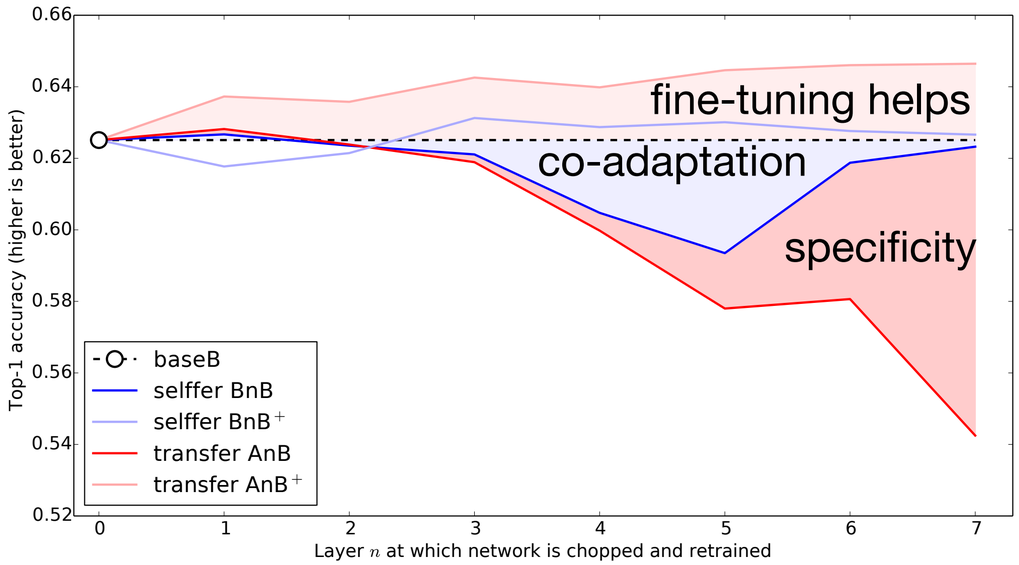
Conclusions

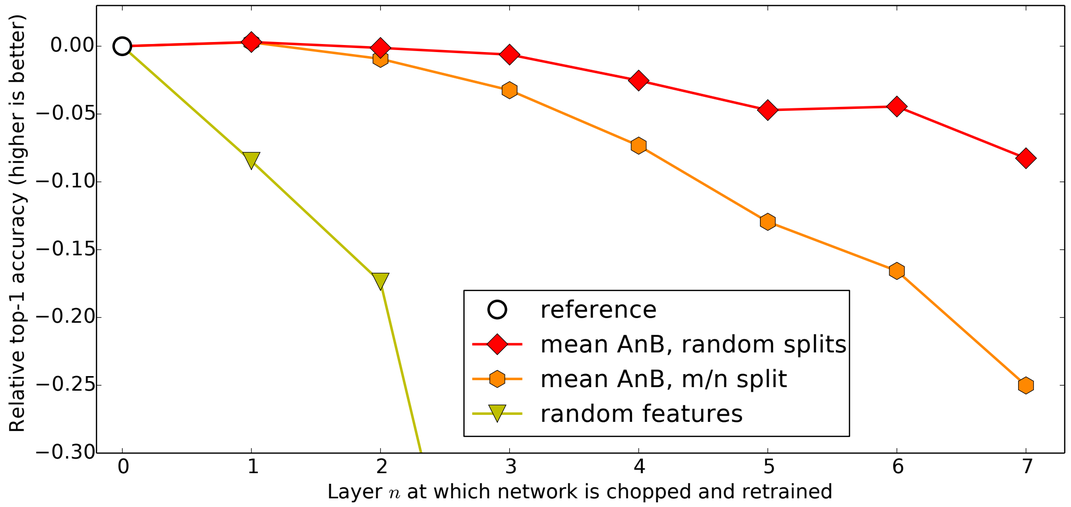




- Measure *general* to *specific* transition layer by layer
- Transferability governed by:
  - lost co-adaptations
  - specificity
  - difference between base and target dataset

Conclusions





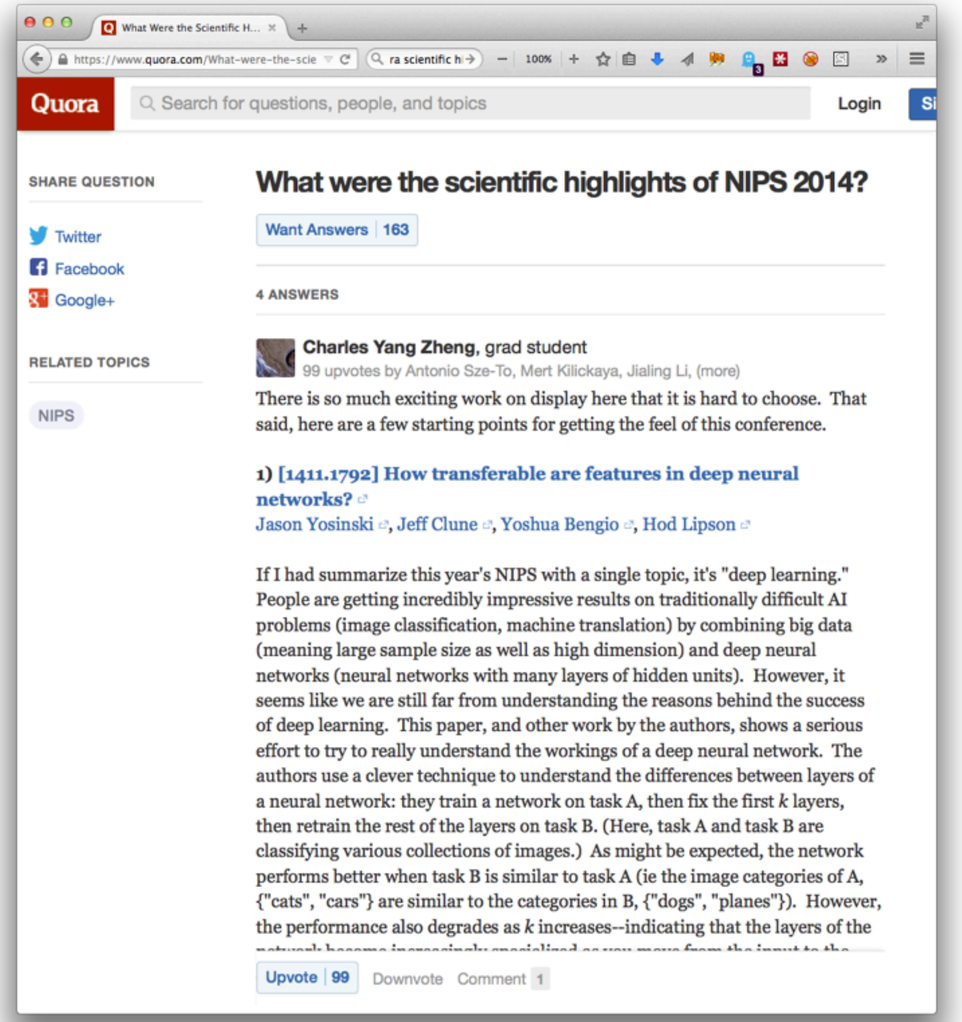
- Measure *general* to *specific* transition layer by layer
- Transferability governed by:
  - lost co-adaptations
  - specificity
  - difference between base and target dataset
- Fine-tuning helps even on large target dataset

NIPS ’14

(oral)

NIPS ’14

(oral)



Quora

Search for questions, people, and topics

Login

SHARE QUESTION

Twitter

Facebook

Google+

RELATED TOPICS

NIPS

What were the scientific highlights of NIPS 2014?

Want Answers | 163

4 ANSWERS

Charles Yang Zheng, grad student

99 upvotes by Antonio Sze-To, Mert Kilickaya, Jialing Li, (more)

There is so much exciting work on display here that it is hard to choose. That said, here are a few starting points for getting the feel of this conference.

1) [1411.1792] How transferable are features in deep neural networks?

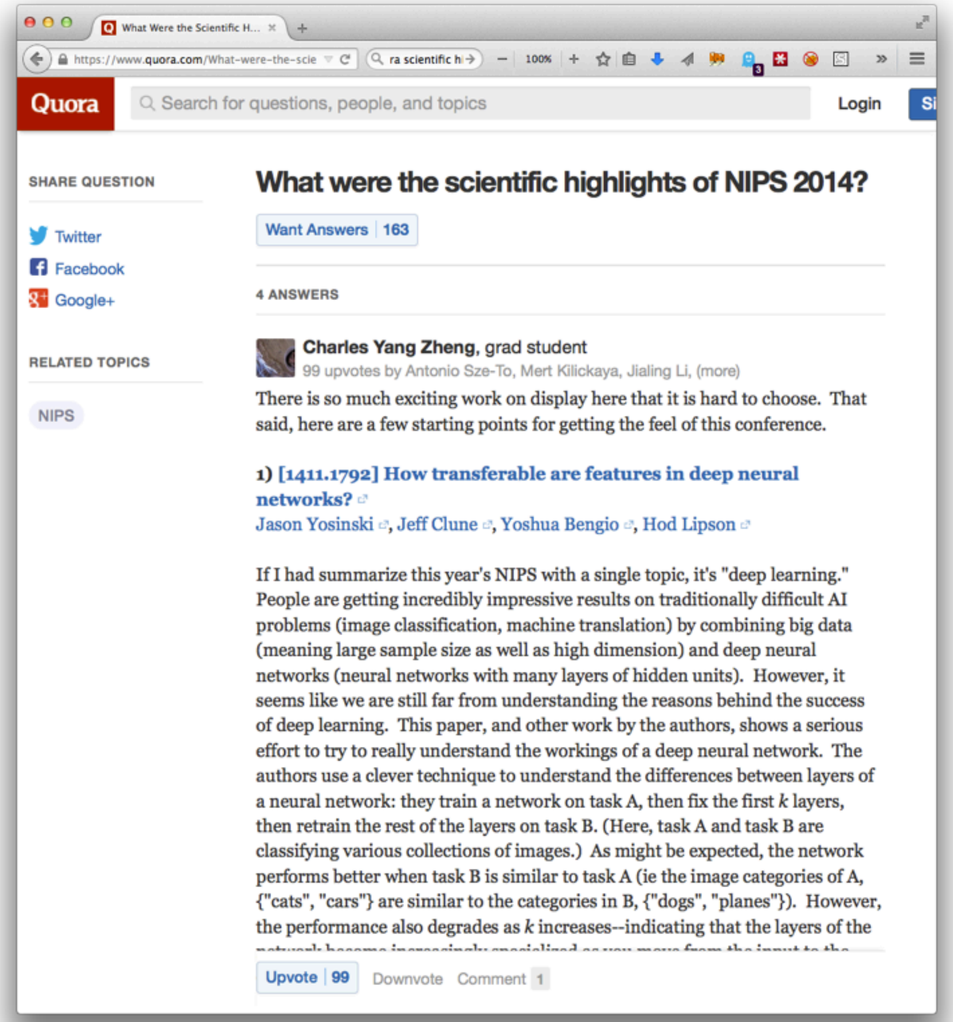
Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson

If I had summarize this year's NIPS with a single topic, it's "deep learning." People are getting incredibly impressive results on traditionally difficult AI problems (image classification, machine translation) by combining big data (meaning large sample size as well as high dimension) and deep neural networks (neural networks with many layers of hidden units). However, it seems like we are still far from understanding the reasons behind the success of deep learning. This paper, and other work by the authors, shows a serious effort to try to really understand the workings of a deep neural network. The authors use a clever technique to understand the differences between layers of a neural network: they train a network on task A, then fix the first *k* layers, then retrain the rest of the layers on task B. (Here, task A and task B are classifying various collections of images.) As might be expected, the network performs better when task B is similar to task A (ie the image categories of A, {"cats", "cars"} are similar to the categories in B, {"dogs", "planes"}). However, the performance also degrades as *k* increases--indicating that the layers of the

Upvote | 99 Downvote Comment 1

NIPS ’14

(oral)



What Were the Scientific H...

https://www.quora.com/What-were-the-scie

Quora | Search for questions, people, and topics | Login

SHARE QUESTION

Twitter

Facebook

Google+

RELATED TOPICS

NIPS

What were the scientific highlights of NIPS 2014?

Want Answers | 163

4 ANSWERS

Charles Yang Zheng, grad student

99 upvotes by Antonio Sze-To, Mert Kilickaya, Jialing Li, (more)

There is so much exciting work on display here that it is hard to choose. That said, here are a few starting points for getting the feel of this conference.

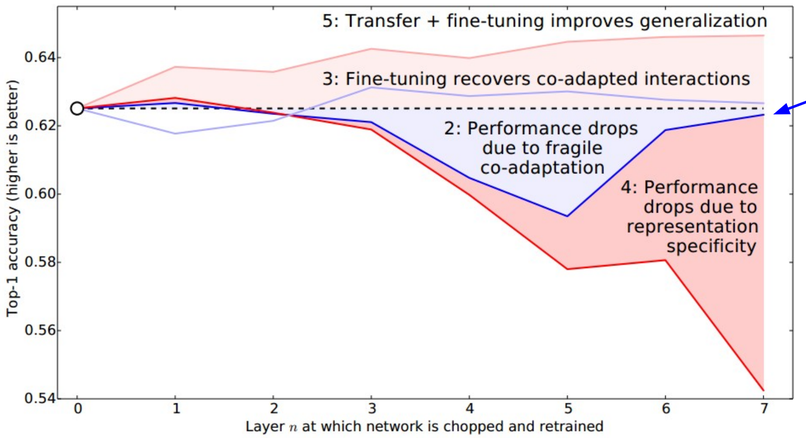
1) **[1411.1792] How transferable are features in deep neural networks?**

Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson

If I had summarize this year's NIPS with a single topic, it's "deep learning." People are getting incredibly impressive results on traditionally difficult AI problems (image classification, machine translation) by combining big data (meaning large sample size as well as high dimension) and deep neural networks (neural networks with many layers of hidden units). However, it seems like we are still far from understanding the reasons behind the success of deep learning. This paper, and other work by the authors, shows a serious effort to try to really understand the workings of a deep neural network. The authors use a clever technique to understand the differences between layers of a neural network: they train a network on task A, then fix the first *k* layers, then retrain the rest of the layers on task B. (Here, task A and task B are classifying various collections of images.) As might be expected, the network performs better when task B is similar to task A (ie the image categories of A, {"cats", "cars"} are similar to the categories in B, {"dogs", "planes"}). However, the performance also degrades as *k* increases--indicating that the layers of the

Upvote | 99 Downvote Comment | 1

How transferable are features in deep neural networks?
[Yosinski et al., 2014]



Split ImageNet classes in half to two sets: A/B.

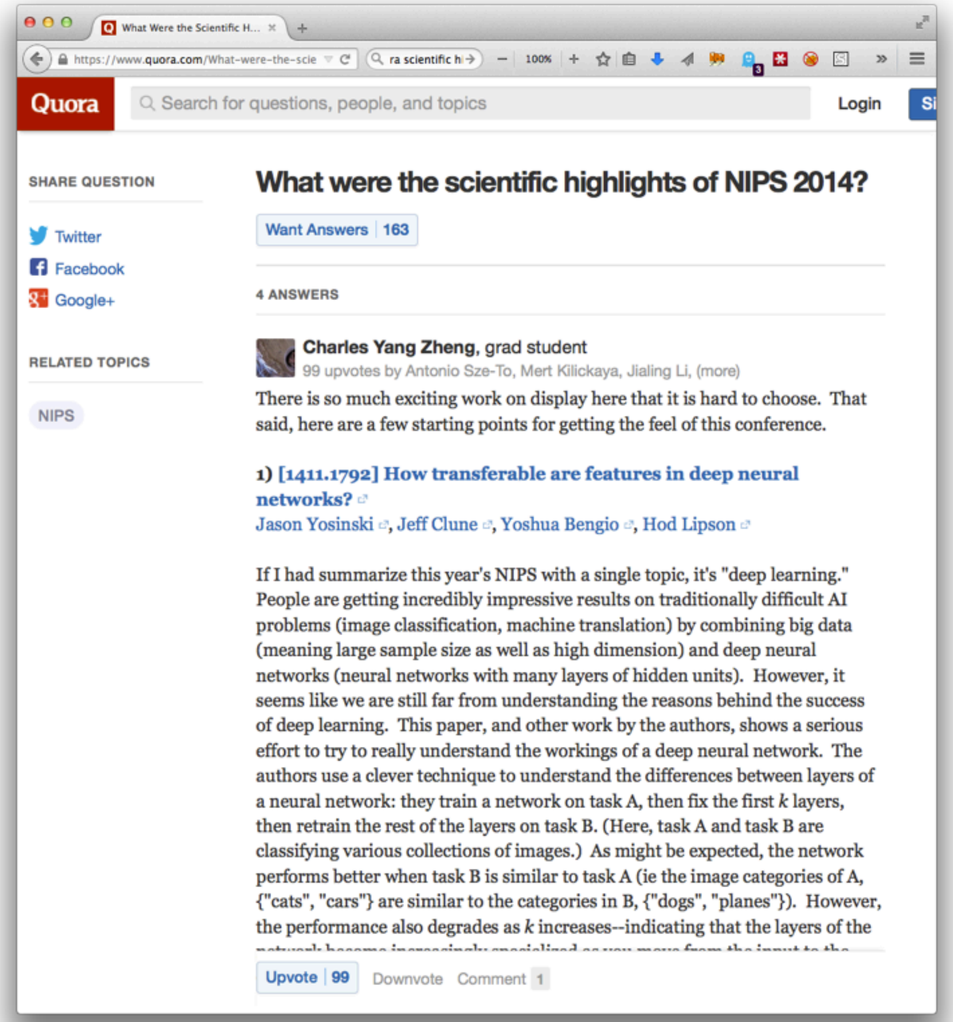
Train on B, fix the first n layers, re-initialize layers n+, train on B again and test on B val

=> performance degrades when n = 4. wat.

Stanford CS231n

NIPS ’14

(oral)



What Were the Scientific H...

https://www.quora.com/What-were-the-scie

Quora | Search for questions, people, and topics | Login

SHARE QUESTION

Twitter

Facebook

Google+

RELATED TOPICS

NIPS

What were the scientific highlights of NIPS 2014?

Want Answers | 163

4 ANSWERS

Charles Yang Zheng, grad student

99 upvotes by Antonio Sze-To, Mert Kilickaya, Jialing Li, (more)

There is so much exciting work on display here that it is hard to choose. That said, here are a few starting points for getting the feel of this conference.

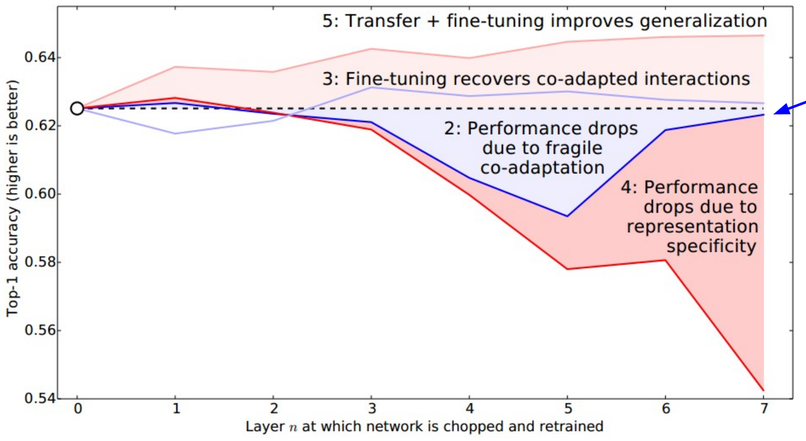
1) **[1411.1792] How transferable are features in deep neural networks?**

Jason Yosinski, Jeff Clune, Yoshua Bengio, Hod Lipson

If I had summarize this year's NIPS with a single topic, it's "deep learning." People are getting incredibly impressive results on traditionally difficult AI problems (image classification, machine translation) by combining big data (meaning large sample size as well as high dimension) and deep neural networks (neural networks with many layers of hidden units). However, it seems like we are still far from understanding the reasons behind the success of deep learning. This paper, and other work by the authors, shows a serious effort to try to really understand the workings of a deep neural network. The authors use a clever technique to understand the differences between layers of a neural network: they train a network on task A, then fix the first *k* layers, then retrain the rest of the layers on task B. (Here, task A and task B are classifying various collections of images.) As might be expected, the network performs better when task B is similar to task A (ie the image categories of A, {"cats", "cars"} are similar to the categories in B, {"dogs", "planes"}). However, the performance also degrades as *k* increases--indicating that the layers of the

Upvote | 99 Downvote Comment | 1

How transferable are features in deep neural networks?
[Yosinski et al., 2014]



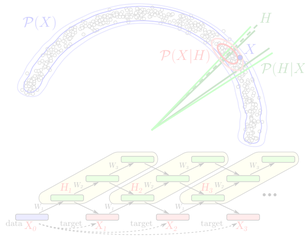
Split ImageNet classes in half to two sets: A/B.

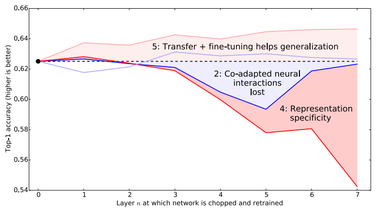
Train on B, fix the first n layers, re-initialize layers n+, train on B again and test on B val

=> performance degrades when n = 4. wat.

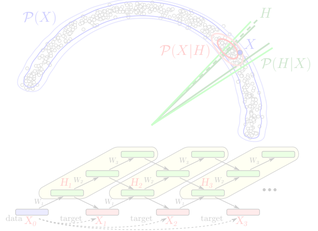
Stanford CS231n

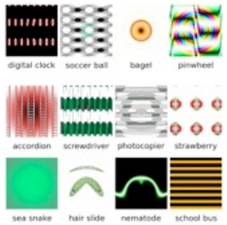
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

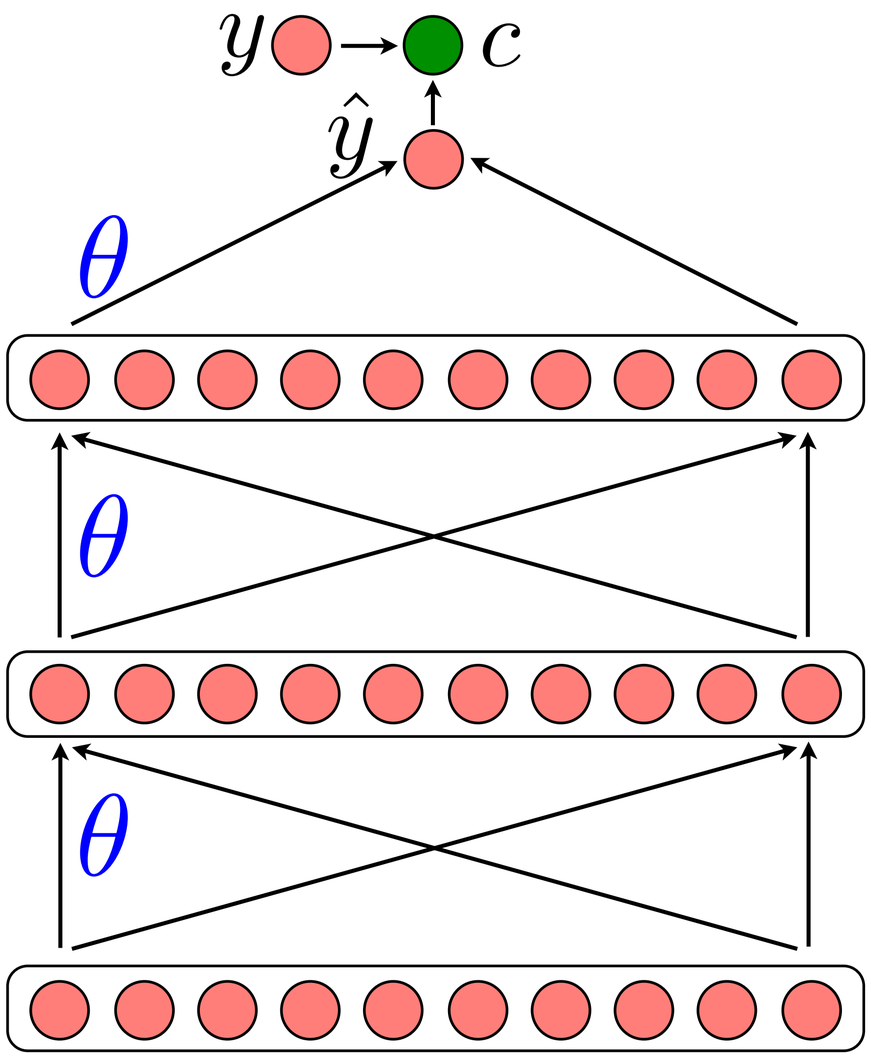




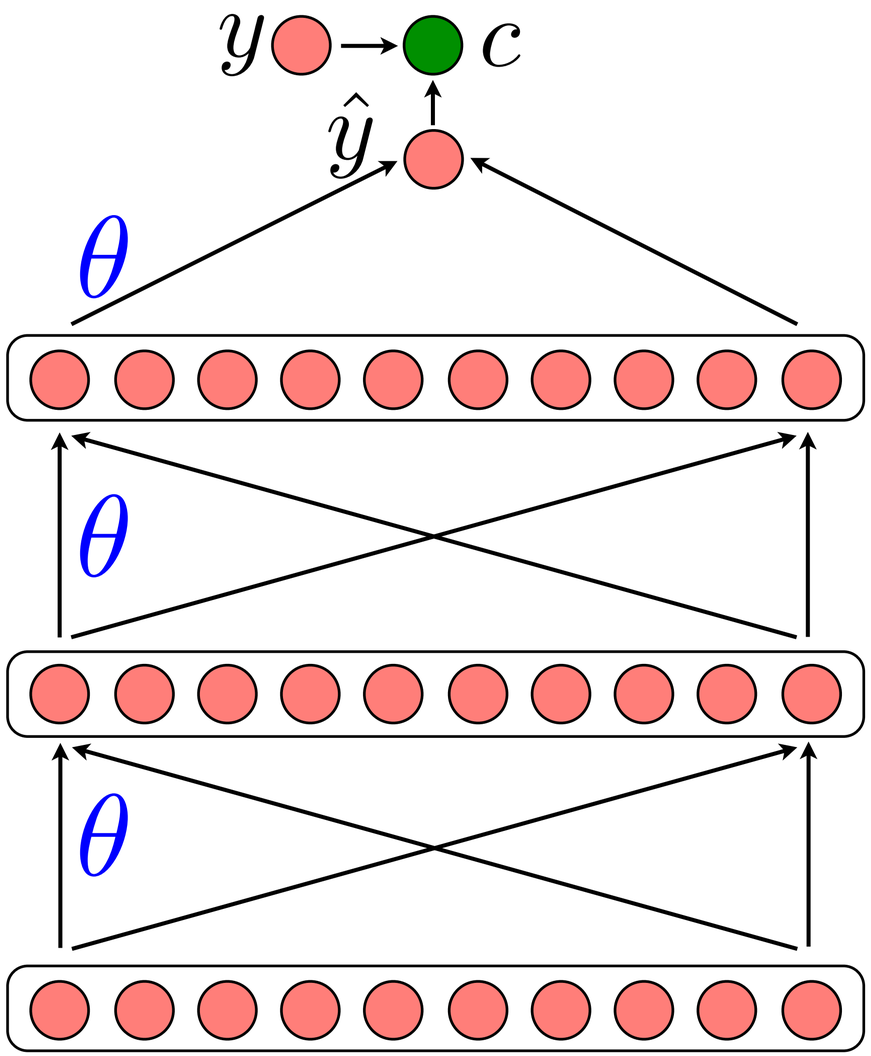
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize





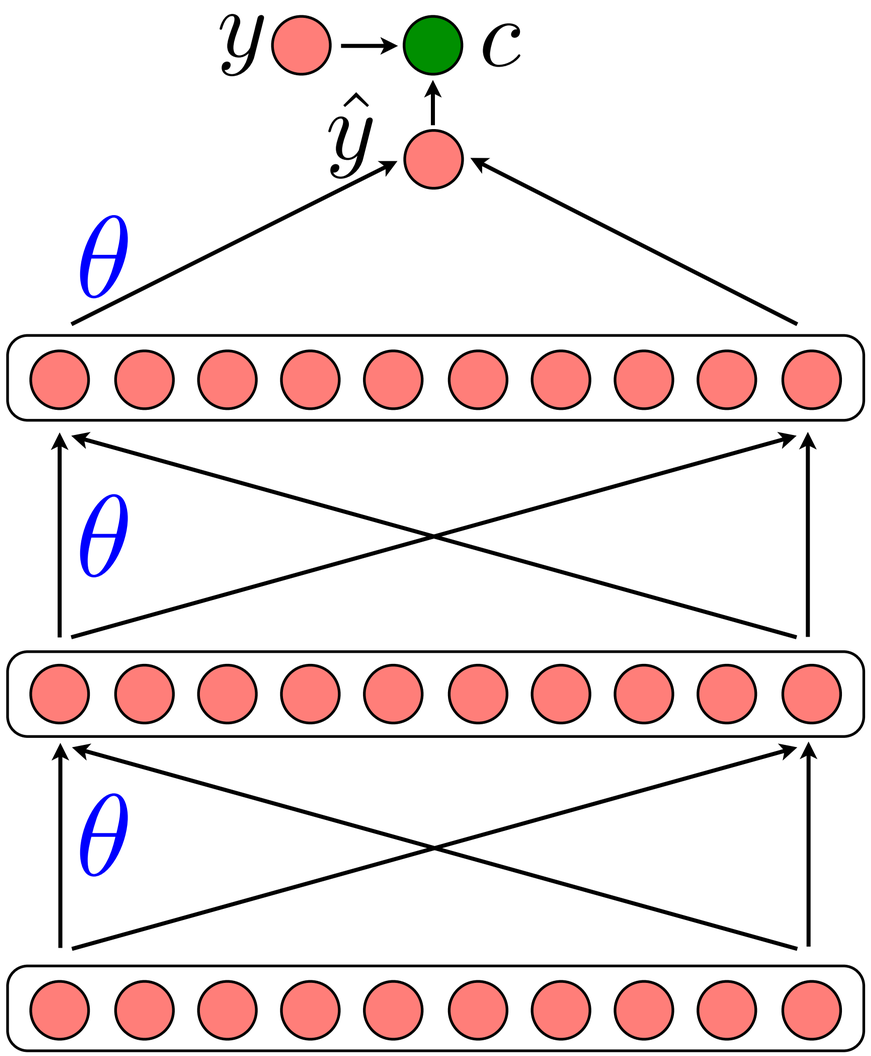


$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$



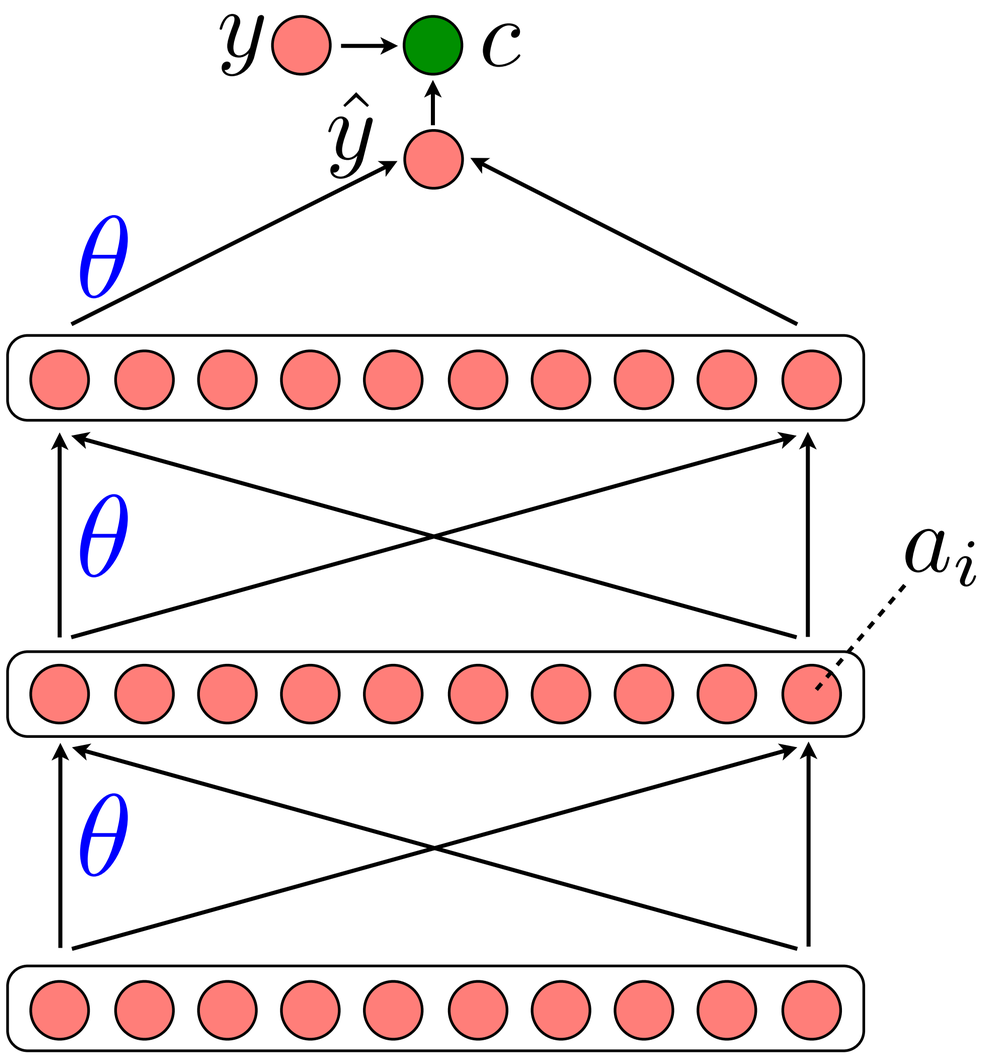
$$\frac{\partial c}{\partial \theta}$$

$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 \\ & & \cdot\ \cdot\ \cdot & & \end{bmatrix}$$



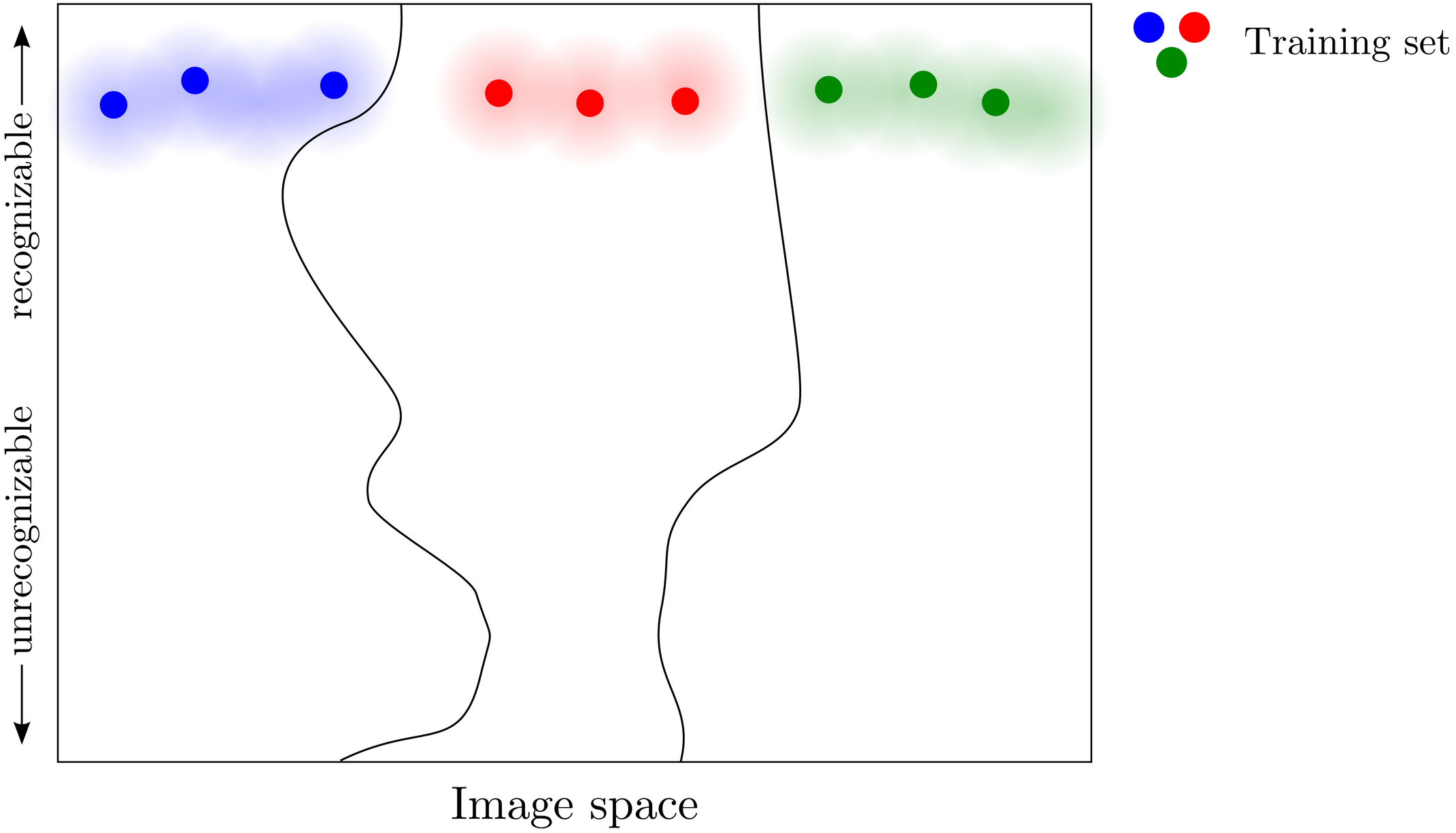
$$\frac{\partial c}{\partial \theta} \quad \frac{\partial c}{\partial X}$$

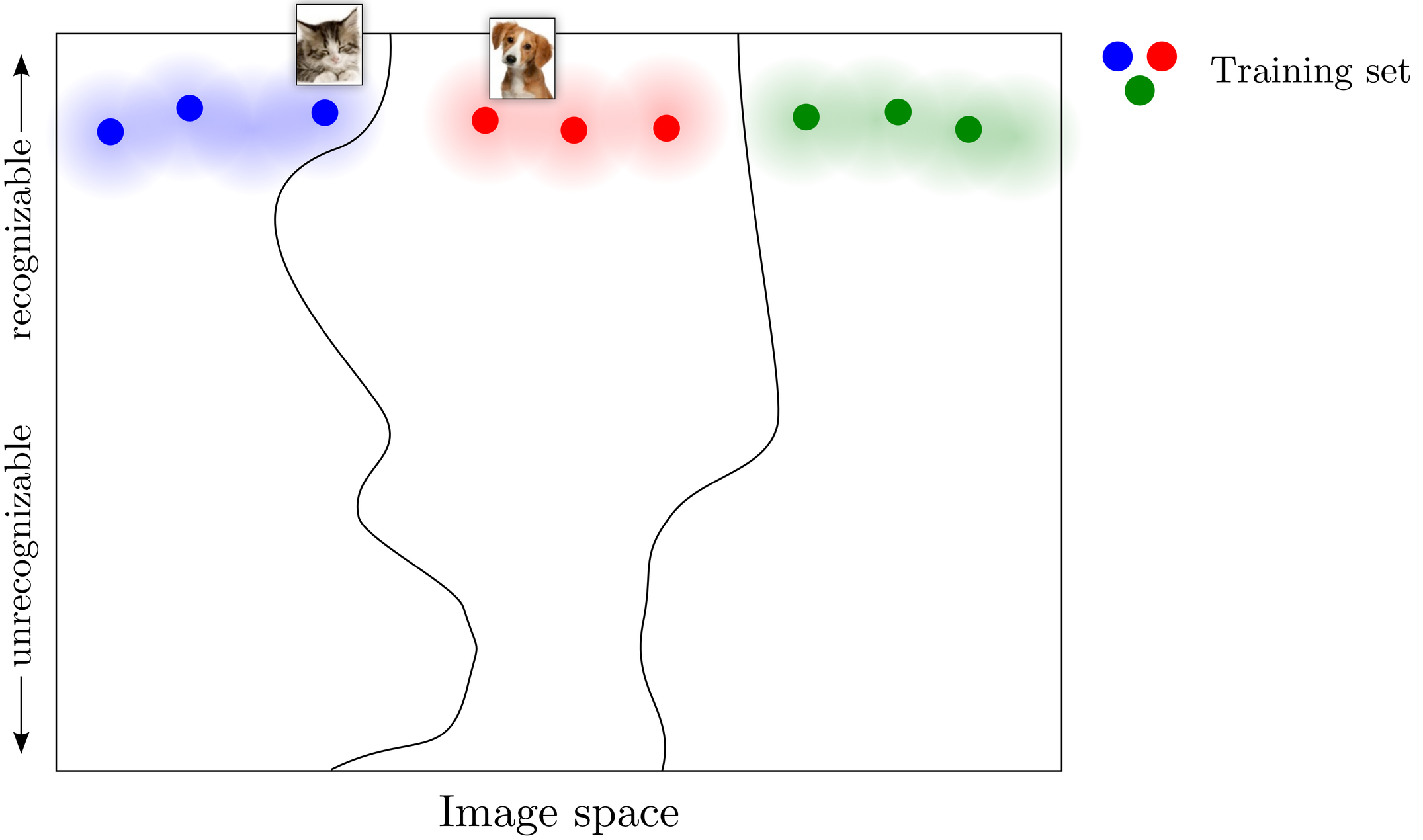
$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \quad . \quad . & & & \end{bmatrix}$$

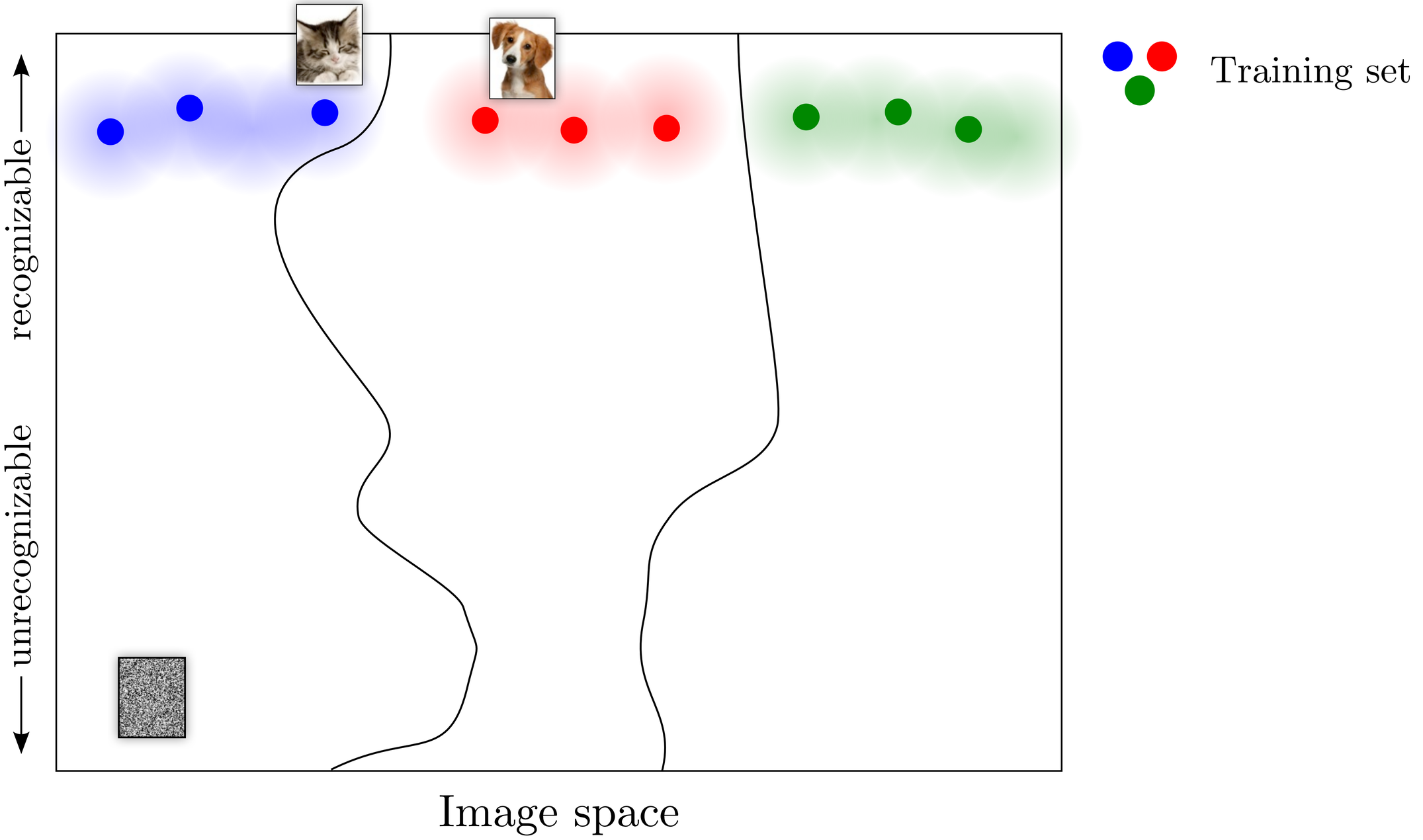


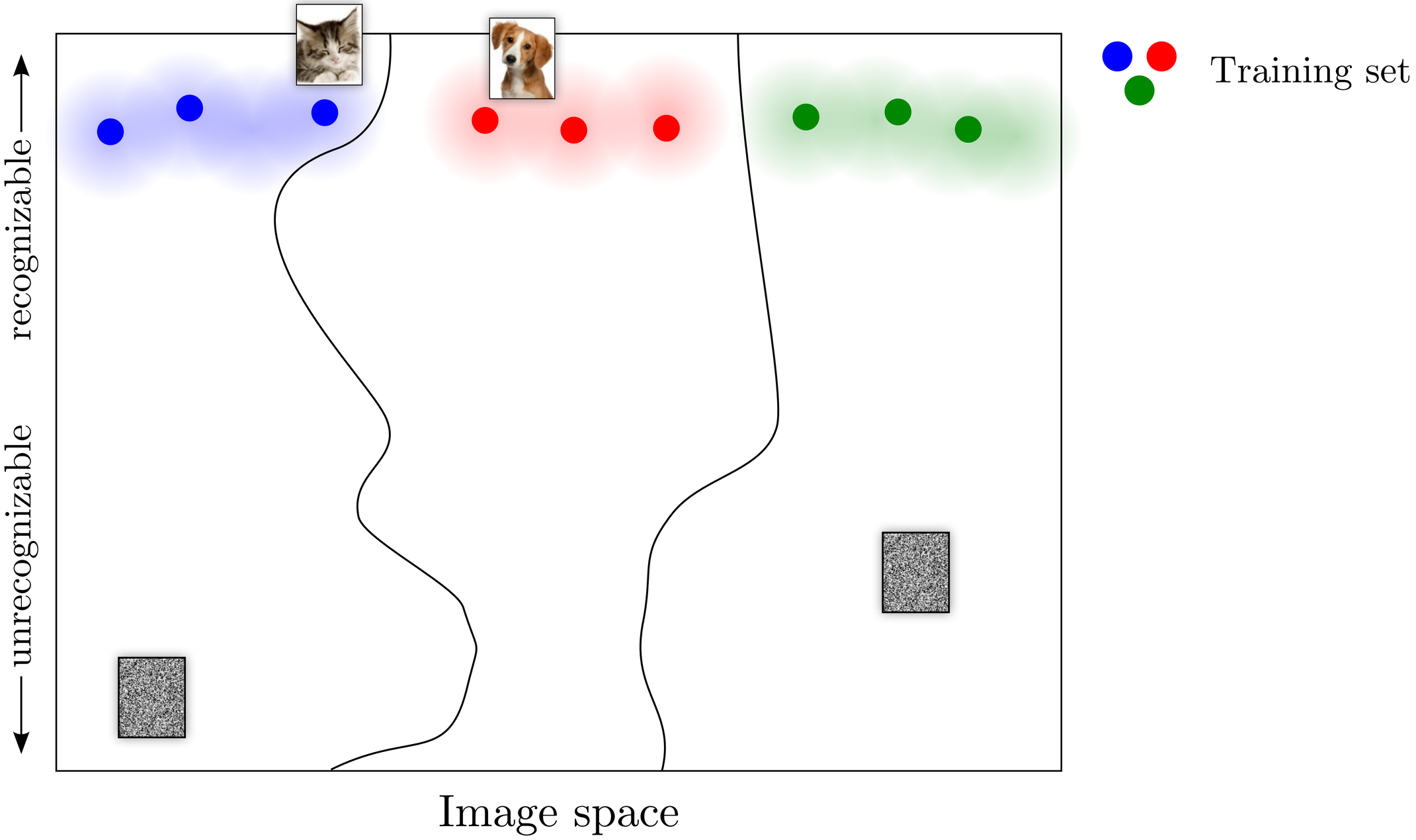
$$\frac{\partial c}{\partial \theta} \quad \frac{\partial c}{\partial X} \quad \frac{\partial a_i}{\partial X}$$

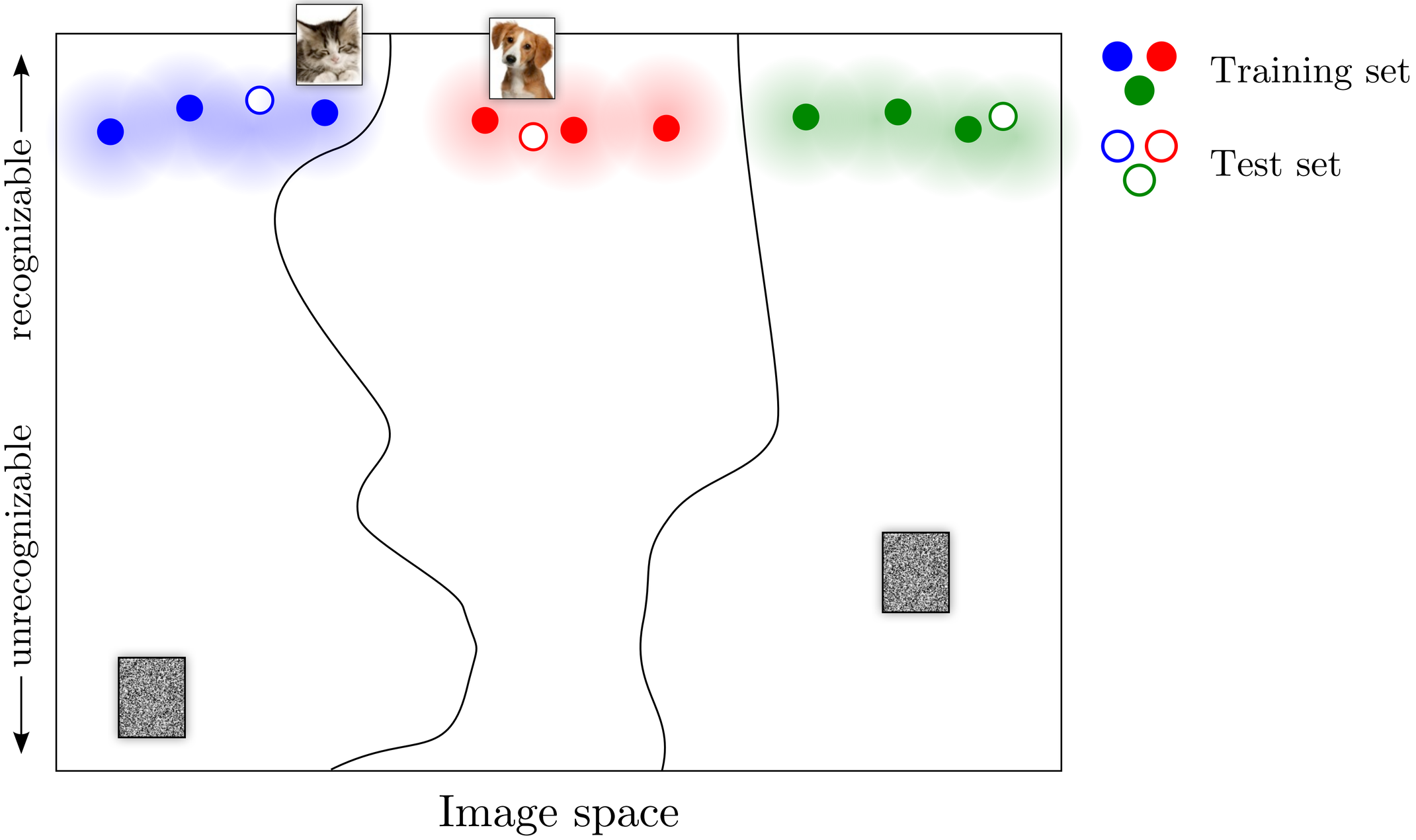
$$X \begin{bmatrix} 1.5 & -0.7 & 0.2 & -0.2 & 0.0 & \\ 0.4 & -0.7 & -0.7 & 0.7 & 1.2 & . \\ -0.6 & 1.8 & 1.7 & -1.1 & -0.6 & . \\ -1.1 & 0.4 & -2.0 & -0.3 & 1.1 & . \\ 1.7 & 0.0 & -0.1 & -0.5 & 0.1 & \\ 1.2 & 1.5 & 0.3 & 0.4 & -1.5 & \\ & & . \; . \; . & & & \end{bmatrix}$$

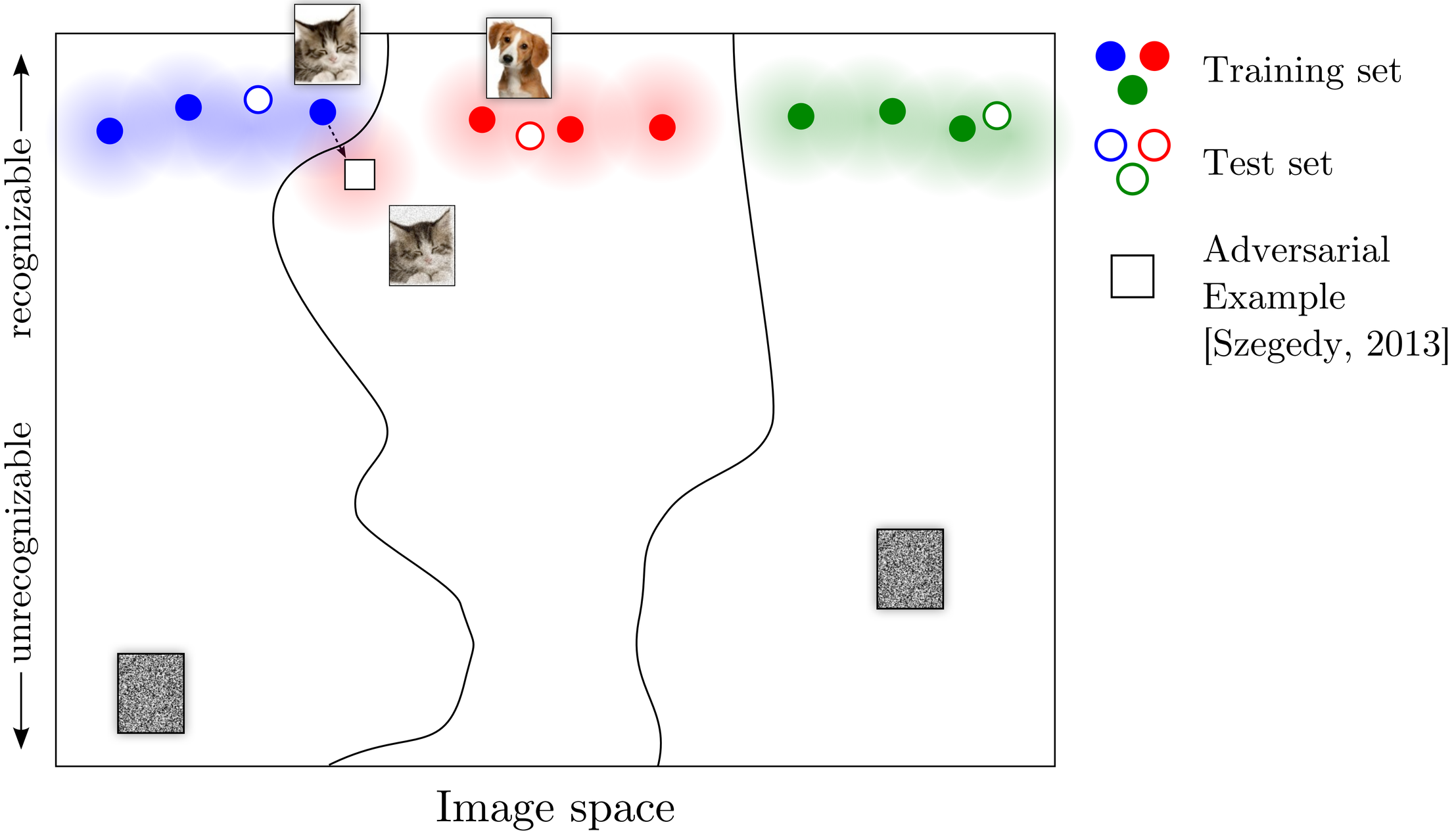

recognizable
unrecognizable
Image space
Training set

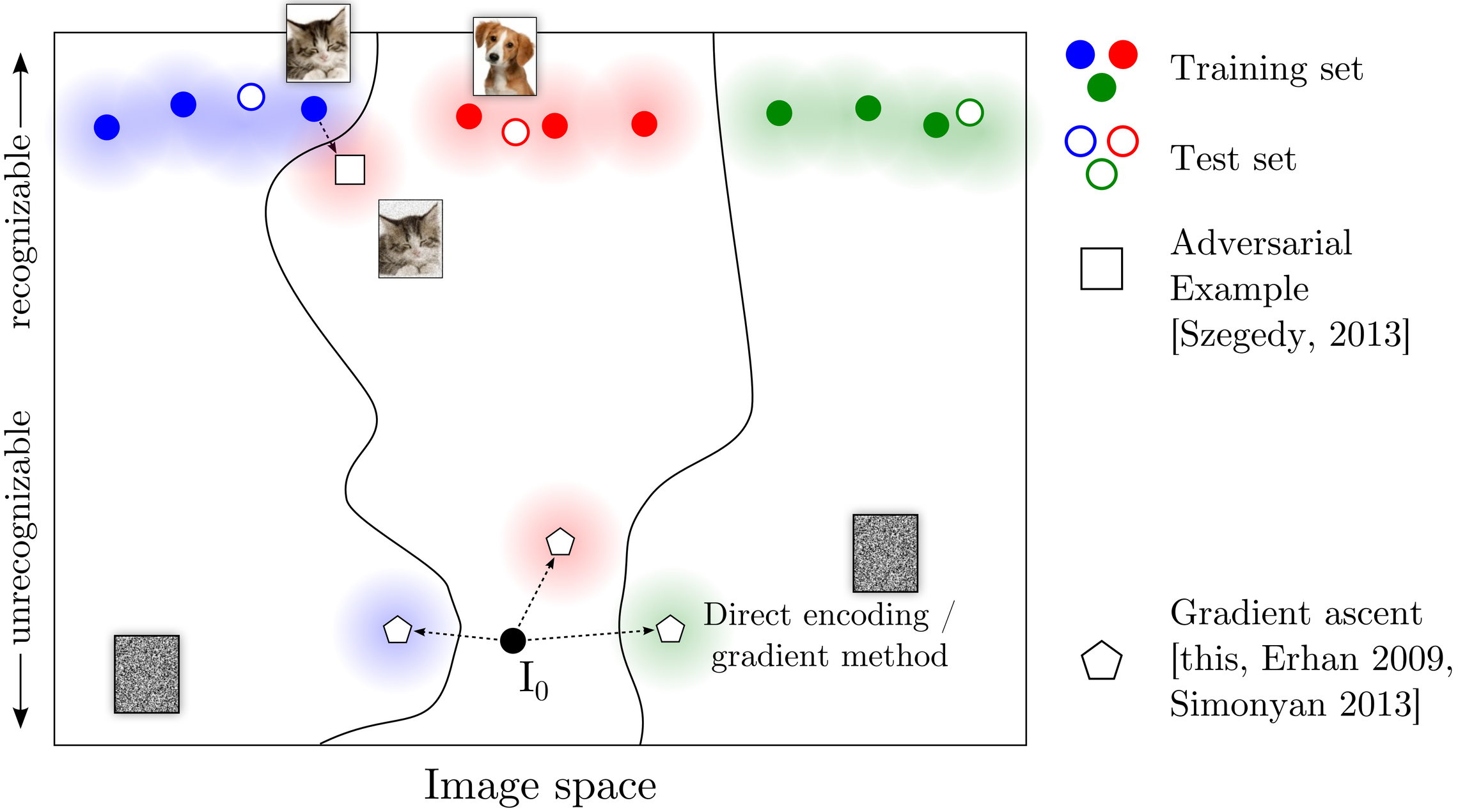

recognizable
unrecognizable
Image space
Training set

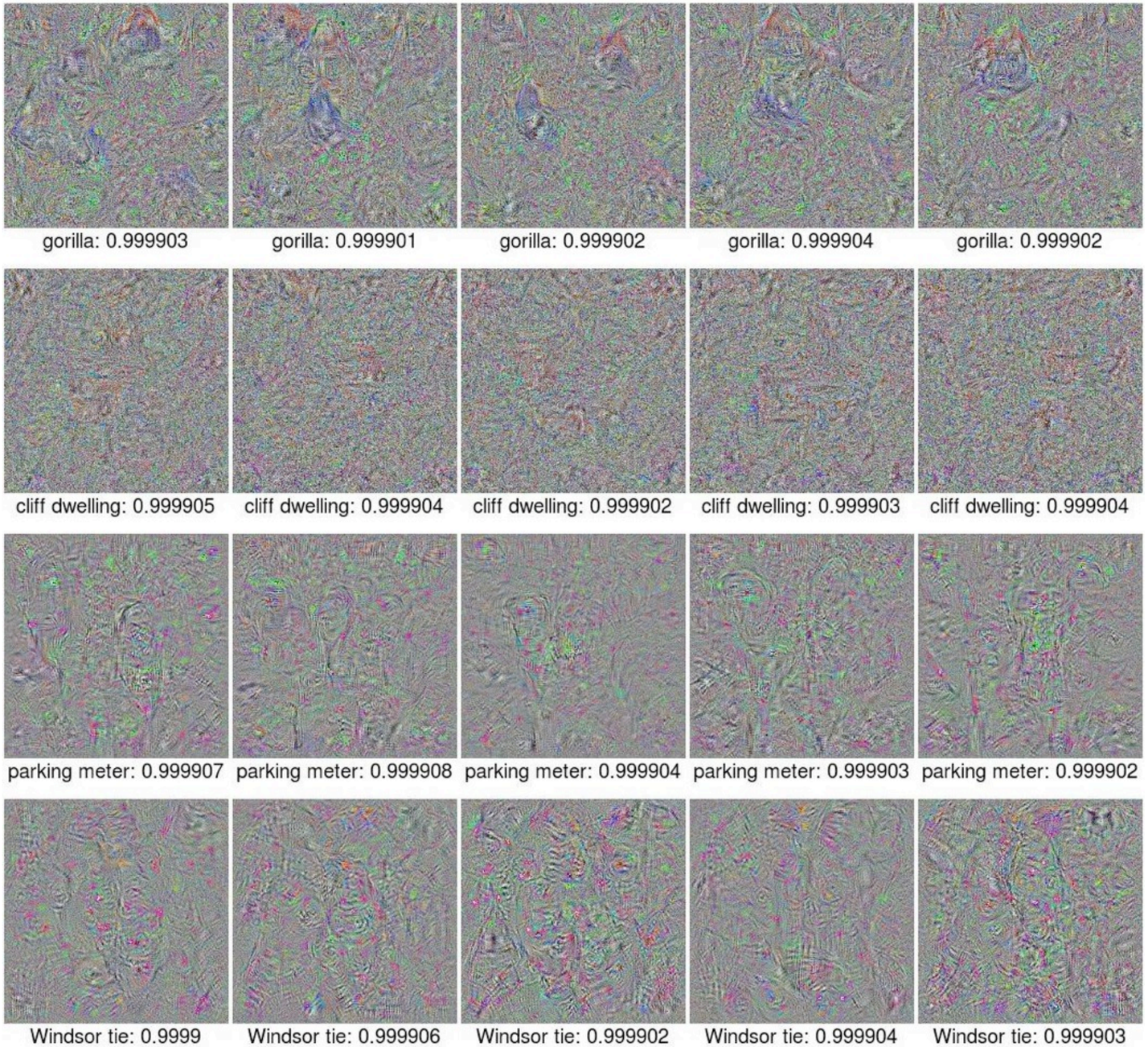

recognizable
unrecognizable
Training set
Image space

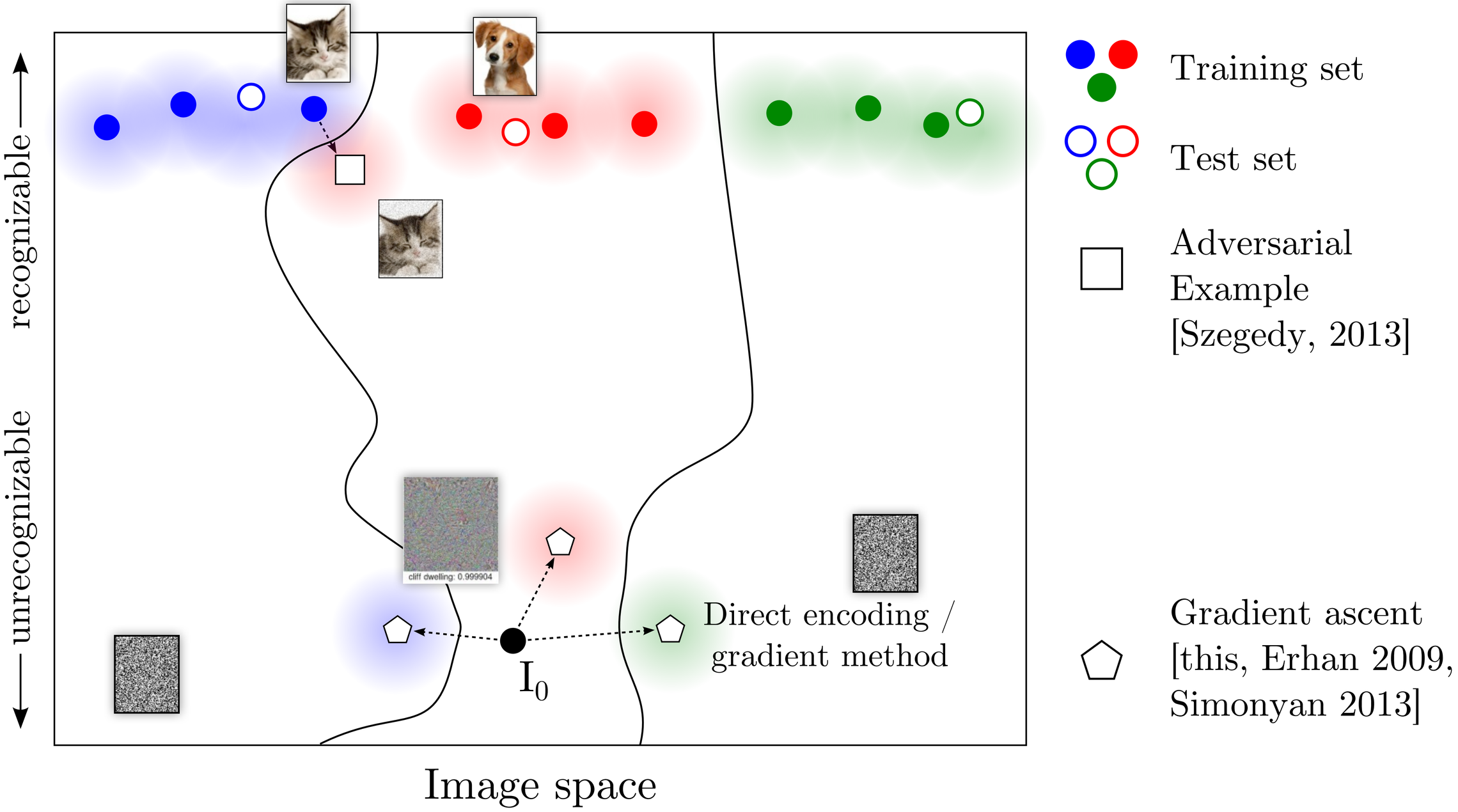

recognizable
unrecognizable
Training set
Image space

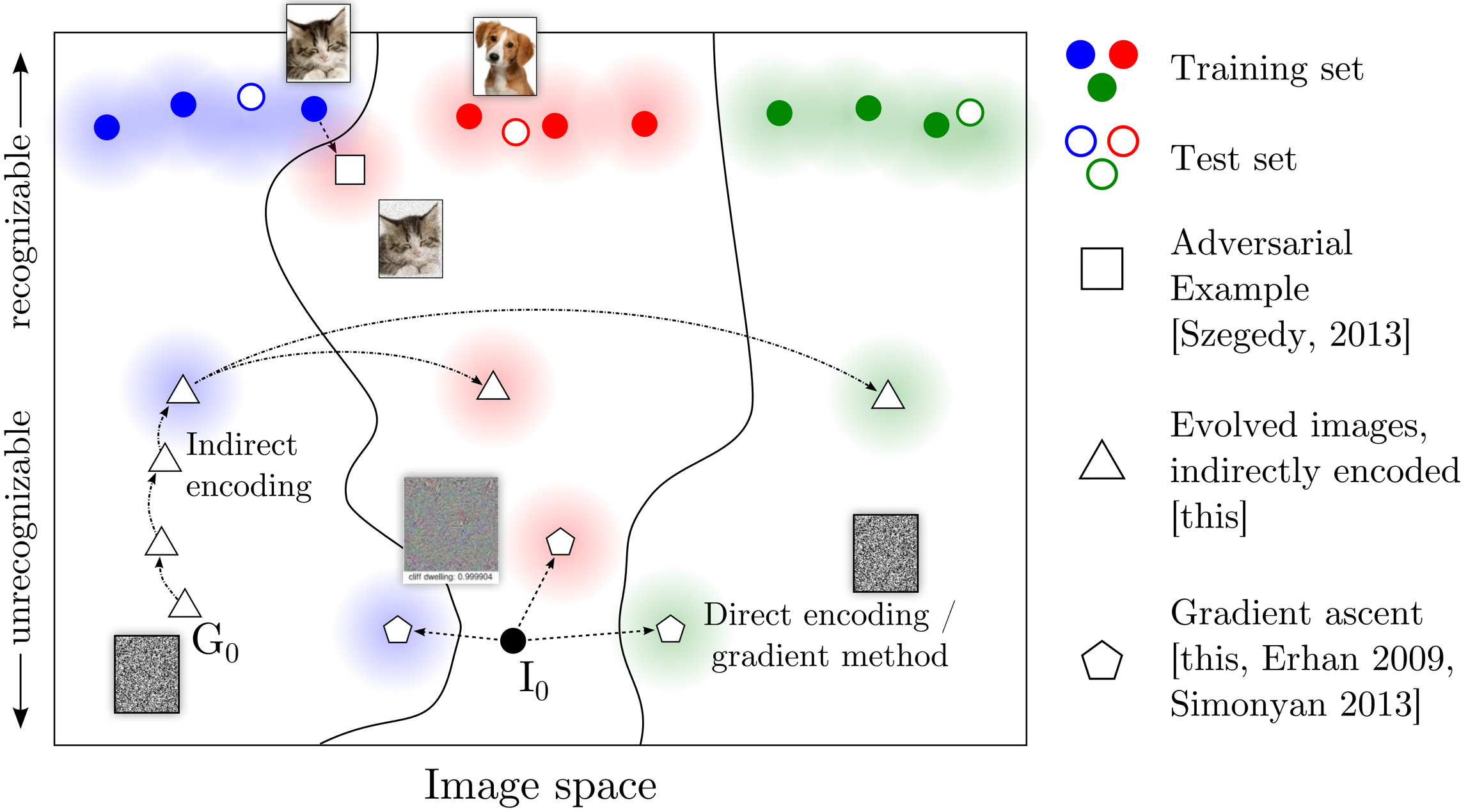

recognizable
unrecognizable
Image space
Training set
Test set

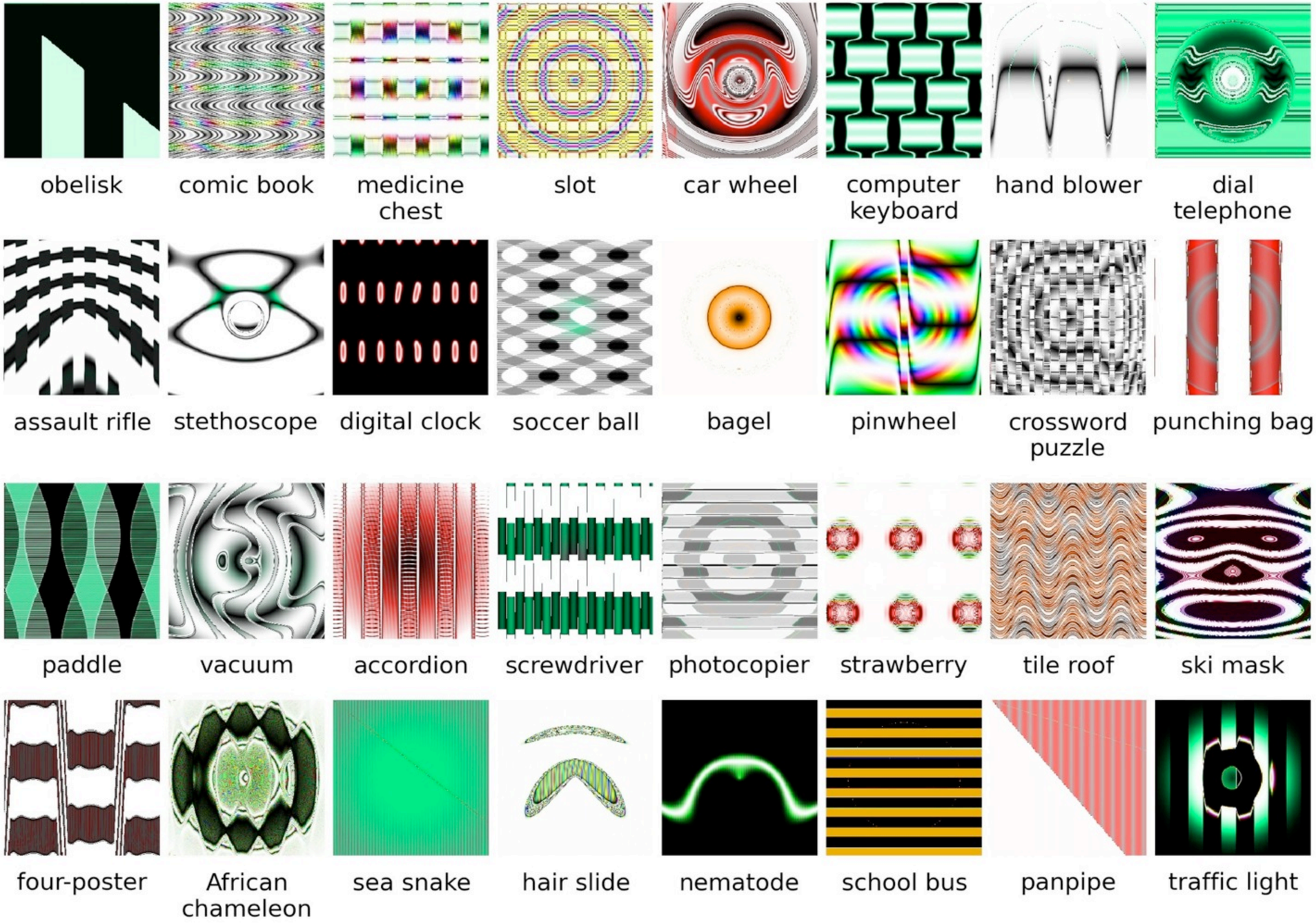

recognizable
unrecognizable
Image space
Training set
Test set
Adversarial Example [Szegedy, 2013]


recognizable
unrecognizable
Image space
I_0
Direct encoding / gradient method
Training set
Test set
Adversarial Example [Szegedy, 2013]
Gradient ascent [this, Erhan 2009, Simonyan 2013]


gorilla: 0.999903
gorilla: 0.999901
gorilla: 0.999902
gorilla: 0.999904
gorilla: 0.999902
cliff dwelling: 0.999905
cliff dwelling: 0.999904
cliff dwelling: 0.999902
cliff dwelling: 0.999903
cliff dwelling: 0.999904
parking meter: 0.999907
parking meter: 0.999908
parking meter: 0.999904
parking meter: 0.999903
parking meter: 0.999902
Windsor tie: 0.9999
Windsor tie: 0.999906
Windsor tie: 0.999902
Windsor tie: 0.999904
Windsor tie: 0.999903


recognizable
unrecognizable
cliff dwelling: 0.999904
I_0
Direct encoding / gradient method
Image space
Training set
Test set
Adversarial Example [Szegedy, 2013]
Gradient ascent [this, Erhan 2009, Simonyan 2013]


recognizable
unrecognizable
Indirect encoding
cliff dwelling: 0.999904
G_0
I_0
Direct encoding / gradient method
Image space
Training set
Test set
Adversarial Example [Szegedy, 2013]
Evolved images, indirectly encoded [this]
Gradient ascent [this, Erhan 2009, Simonyan 2013]


obelisk
comic book
medicine chest
slot
car wheel
computer keyboard
hand blower
dial telephone
assault rifle
stethoscope
digital clock
soccer ball
bagel
pinwheel
crossword puzzle
punching bag
paddle
vacuum
accordion
screwdriver
photocopier
strawberry
tile roof
ski mask
four-poster
African chameleon
sea snake
hair slide
nematode
school bus
panpipe
traffic light

Reactions

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

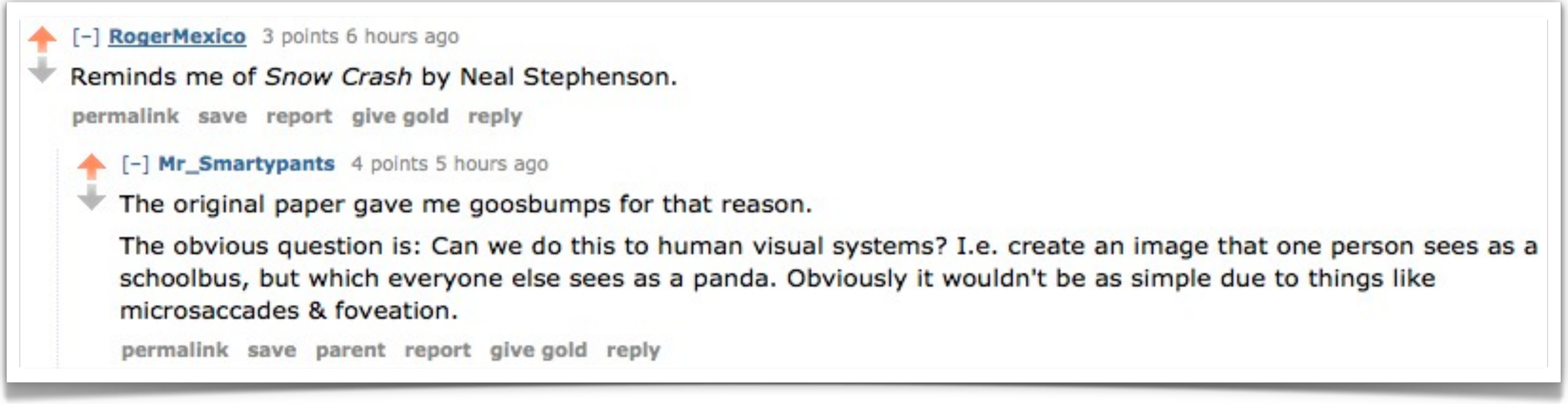
- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low.
  This wouldn’t happen to a human or with a good generative model.”


[–] RogerMexico 3 points 6 hours ago

Reminds me of *Snow Crash* by Neal Stephenson.

permalink save report give gold reply

[–] Mr_Smartypants 4 points 5 hours ago

The original paper gave me goosbumps for that reason.

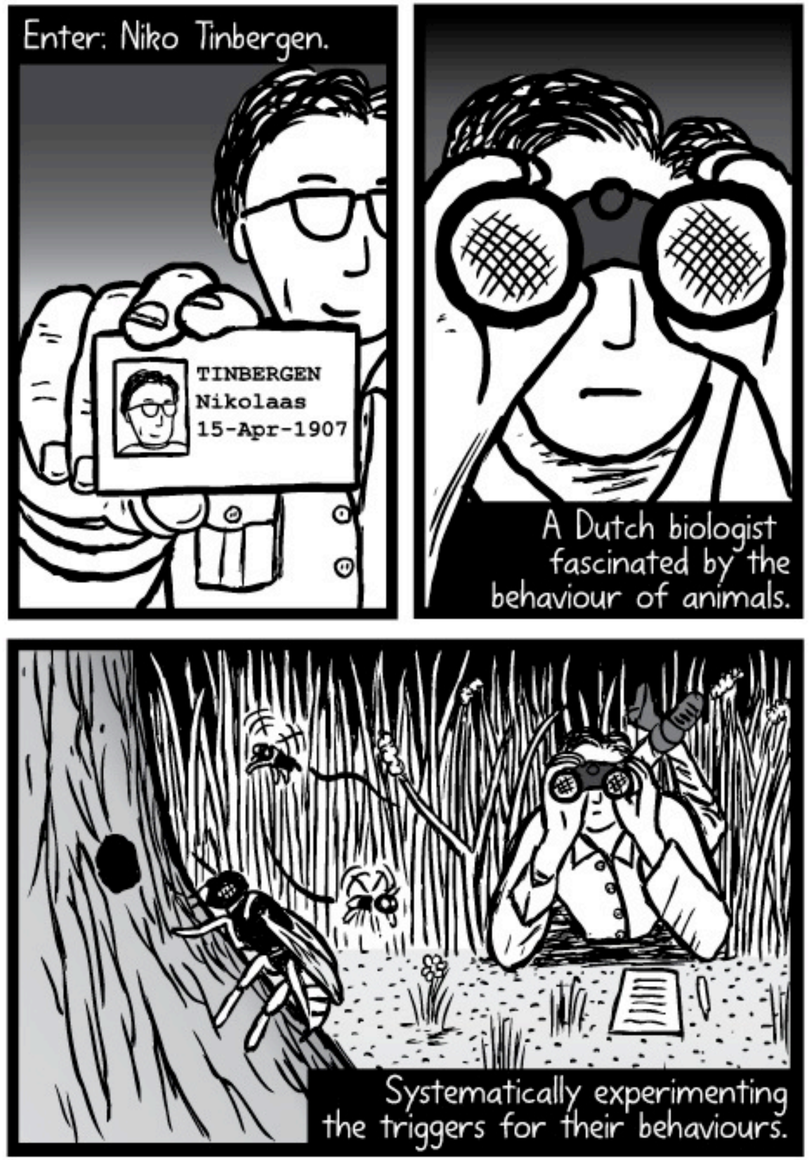
The obvious question is: Can we do this to human visual systems? I.e. create an image that one person sees as a schoolbus, but which everyone else sees as a panda. Obviously it wouldn't be as simple due to things like microsaccades & foveation.

permalink save parent report give gold reply

Nikolaas Tinbergen, 1950’s

www.stuartmcmillen.com/comics_en/supernormal-stimuli

Nikolaas Tinbergen, 1950's

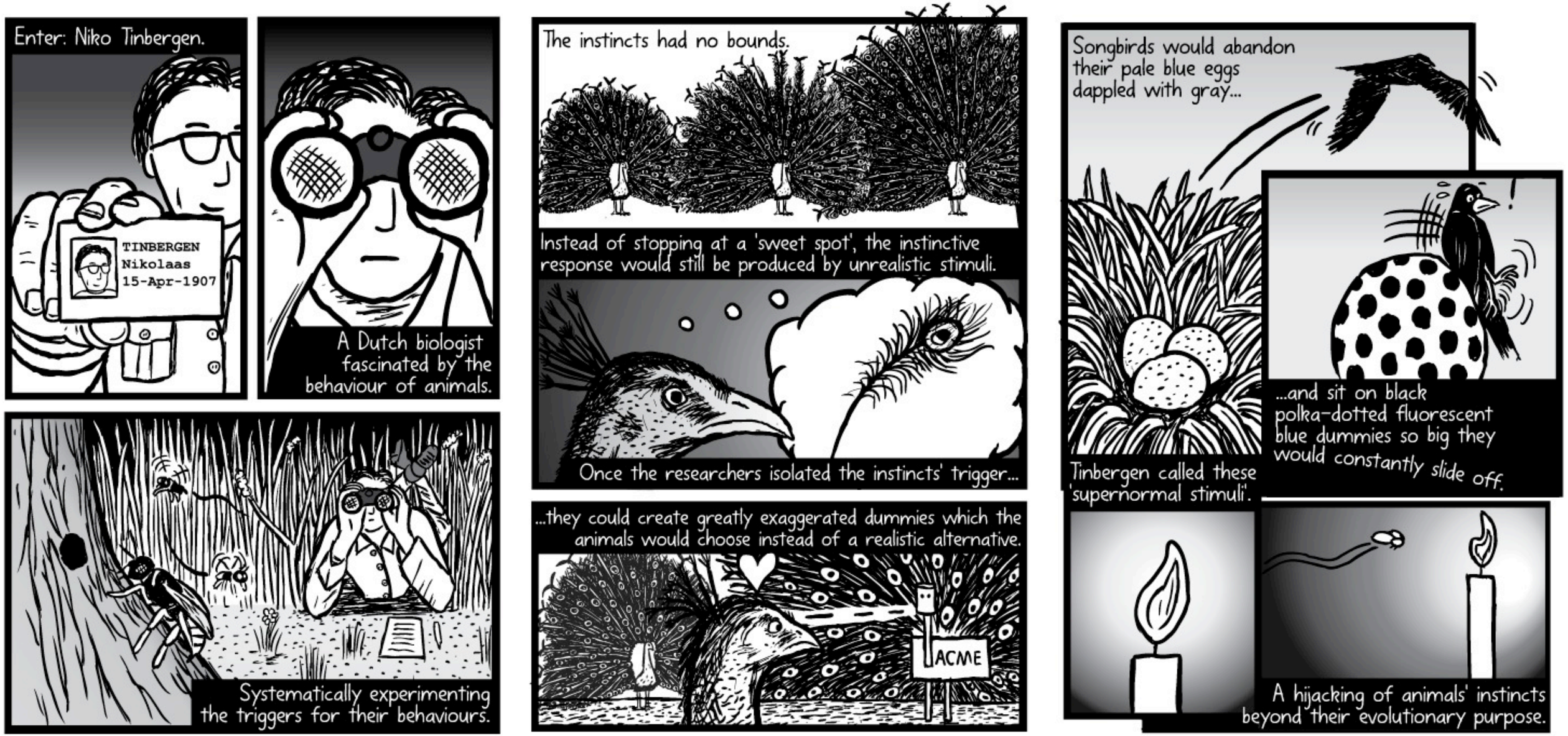


www.stuartmcmillen.com/comics_en/supernormal-stimuli

Nikolaas Tinbergen, 1950's

www.stuartmcmillen.com/comics_en/supernormal-stimuli

Nikolaas Tinbergen, 1950's



www.stuartmcmillen.com/comics_en/supernormal-stimuli

Reactions

- Doesn’t everyone already know this?

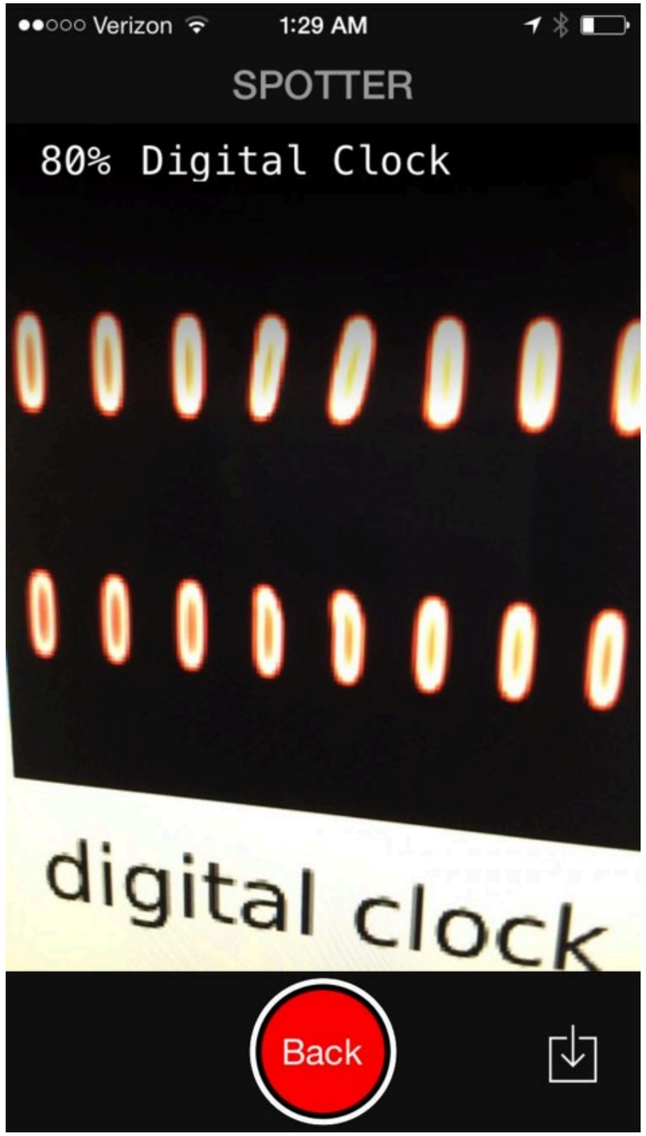
  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

Reactions

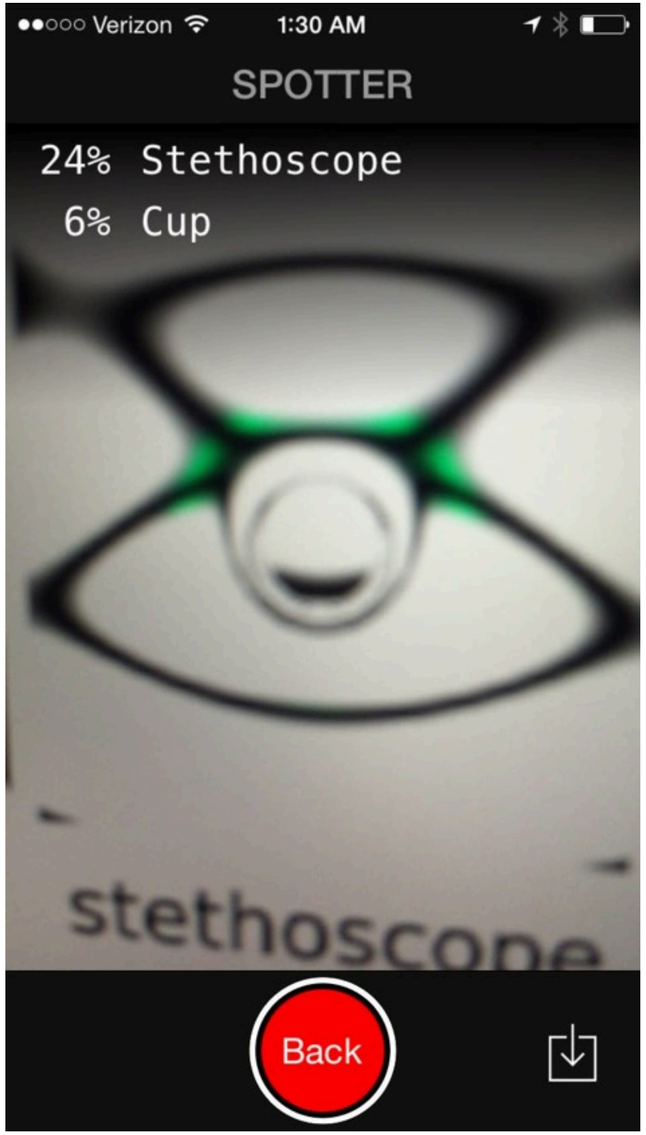
- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

- How robust are these examples?


Verizon
1:29 AM
SPOTTER
80% Digital Clock
digital clock
Back

Verizon
1:26 AM
SPOTTER
51% Green Snake
13% Brain Coral
green snake
Back


Verizon
1:30 AM
SPOTTER
24% Stethoscope
6% Cup
stethoscope
Back

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

- How robust are these examples?

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

- How robust are these examples?

- With/without gradient, high confidence, any class

Reactions

- Doesn’t everyone already know this?

  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”

- How robust are these examples?
- With/without gradient, high confidence, any class
- Security implications

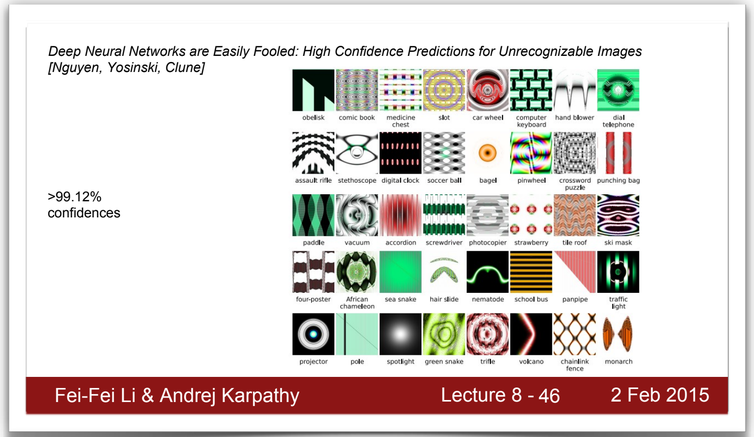
Reactions

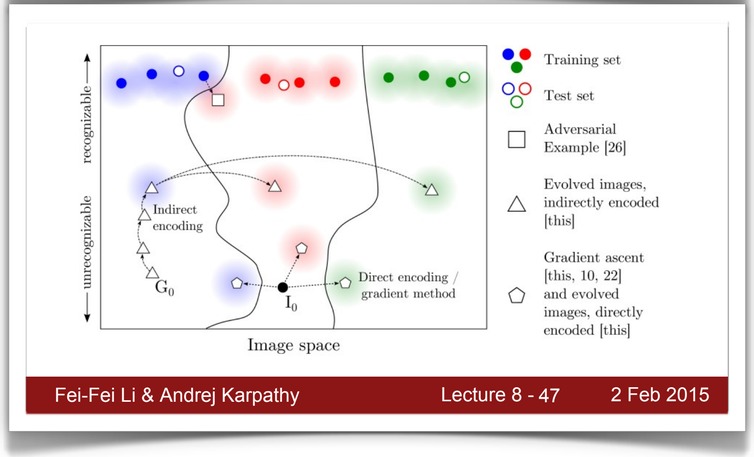
- Doesn’t everyone already know this?
  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”
- How robust are these examples?
- With/without gradient, high confidence, any class
- Security implications
- Useful debugging tool?

Reactions

- Doesn't everyone already know this?
  "Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn't happen to a human or with a good generative model."
- How robust are these examples?
- With/without gradient, high confidence, any class
- Security implications
- Useful debugging tool?

Stanford CS231n



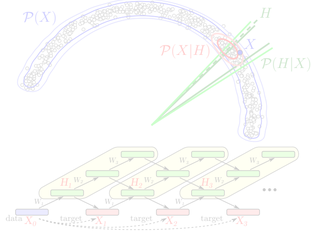


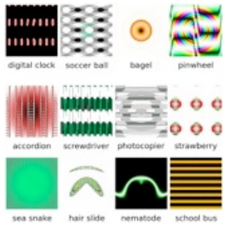
Reactions

arXiv ’14, (submitted) CVPR ’15

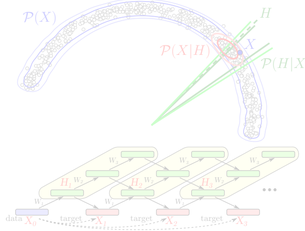
- Doesn’t everyone already know this?
  “Of course a learned p(y|X) model will output garbage when p(X) is low. This wouldn’t happen to a human or with a good generative model.”
- How robust are these examples?
- With/without gradient, high confidence, any class
- Security implications
- Useful debugging tool?

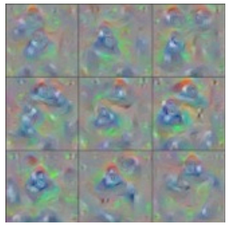
Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

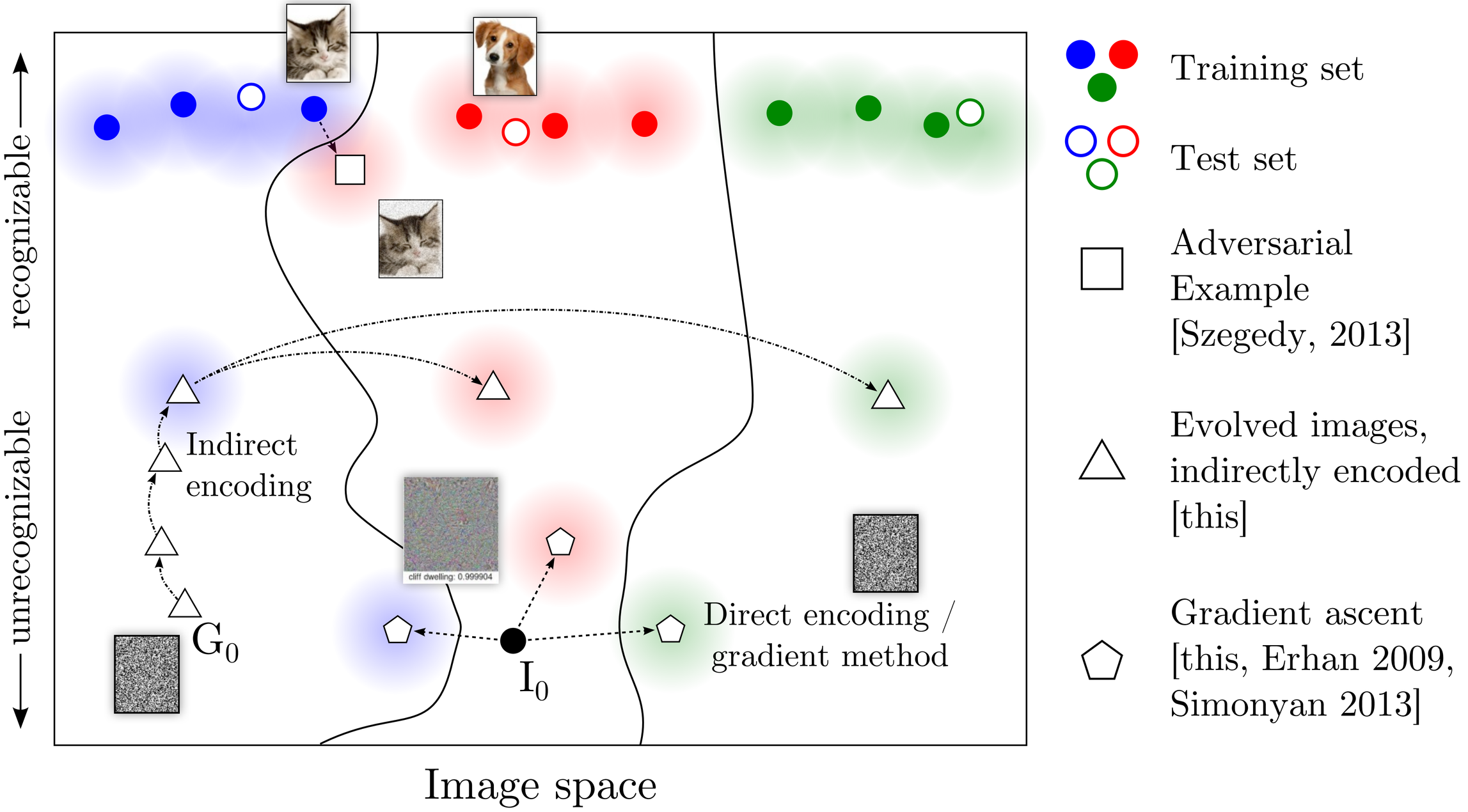




Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize

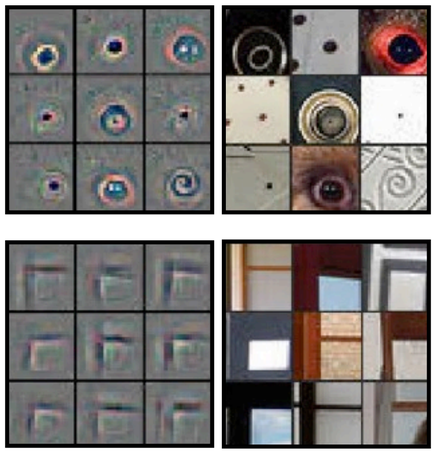




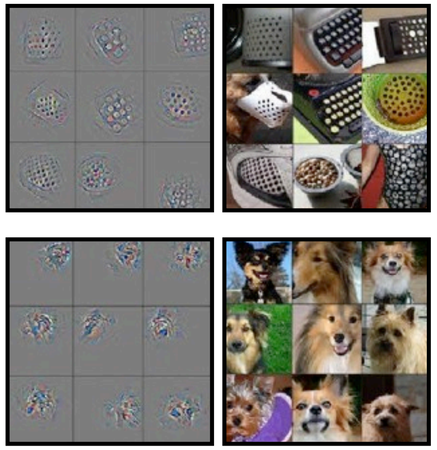

recognizable
unrecognizable
Indirect encoding
cliff dwelling: 0.999904
G_0
I_0
Direct encoding / gradient method
Image space
Training set
Test set
Adversarial Example [Szegedy, 2013]
Evolved images, indirectly encoded [this]
Gradient ascent [this, Erhan 2009, Simonyan 2013]

With dataset

Deconvnets, Zeiler et al arXiv 2013, ECCV 2014.



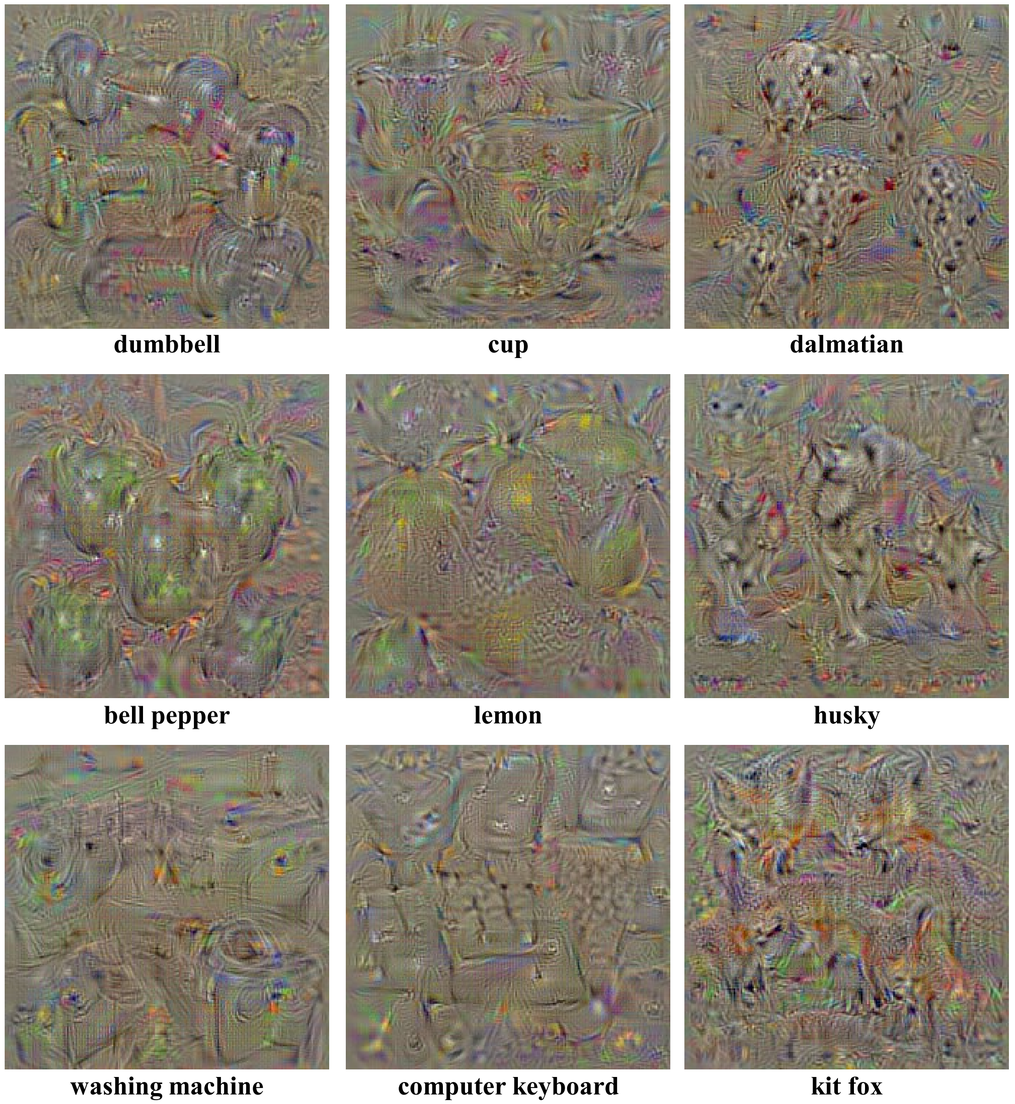
Layer 2

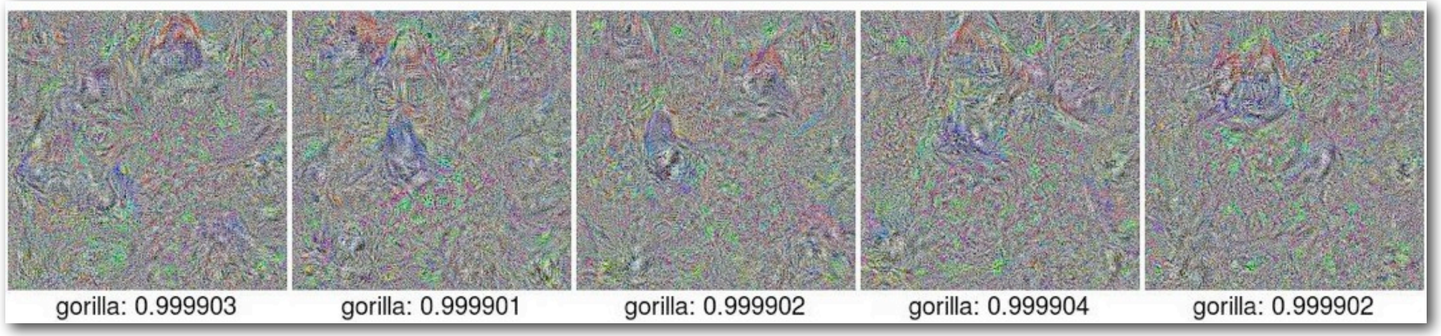


Layer 5

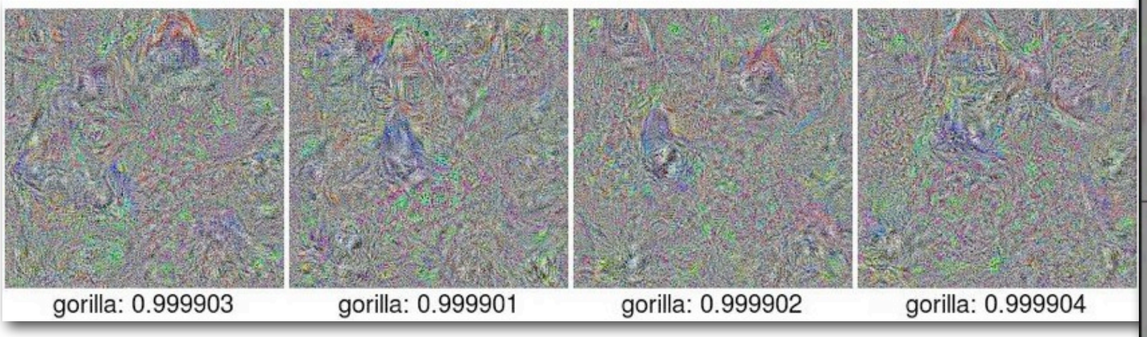
Without dataset

Simonyan et al., arXiv/ICLR 2014.



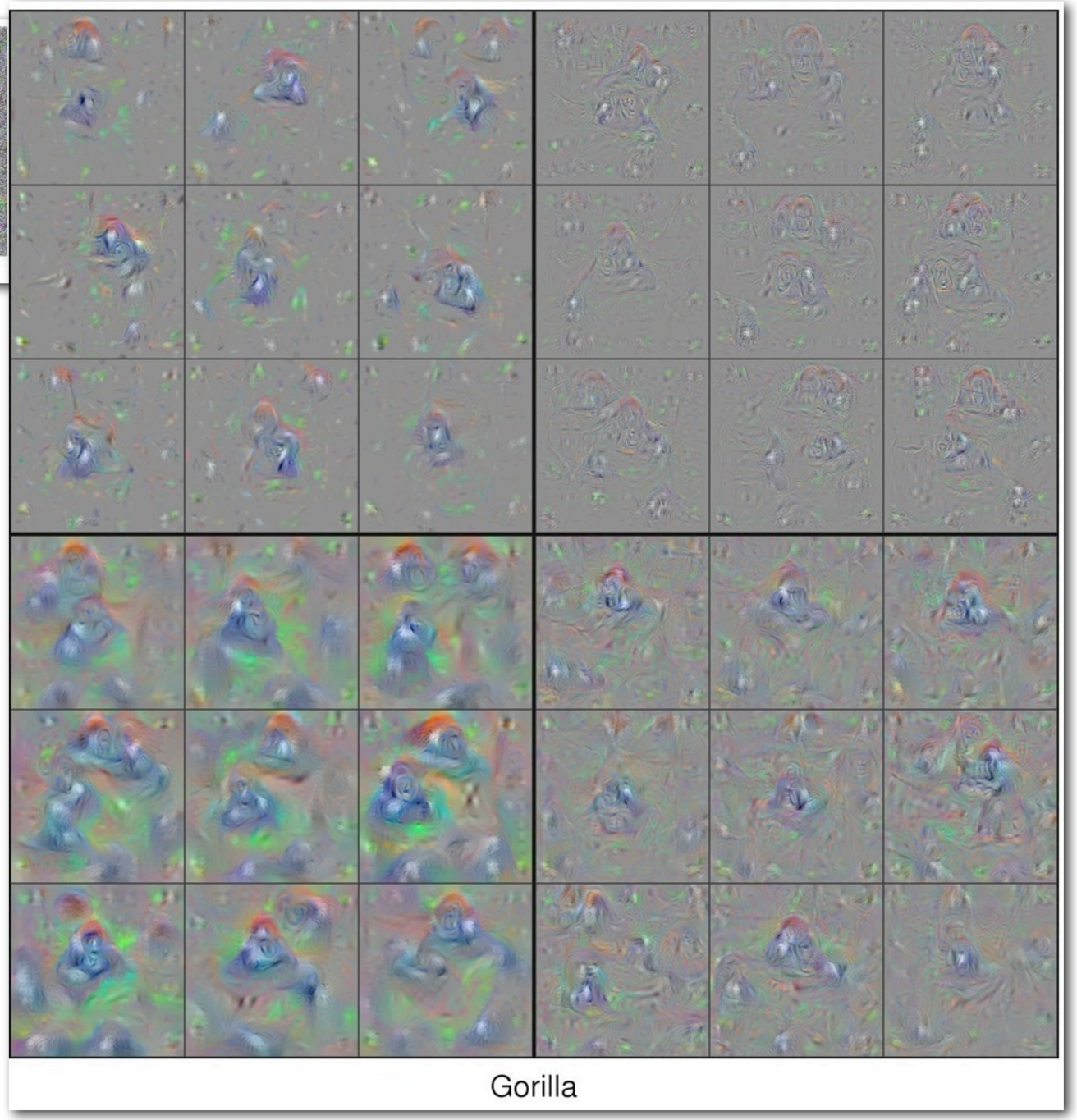


Without regularization

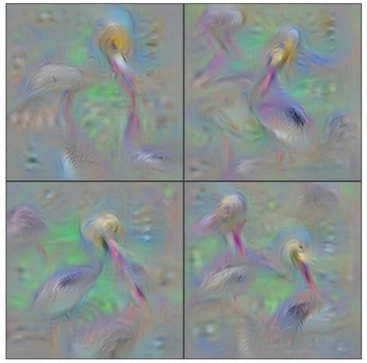


Without regularization

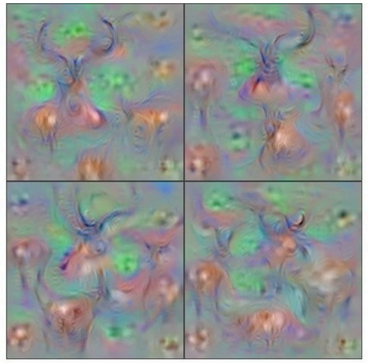
With regularization



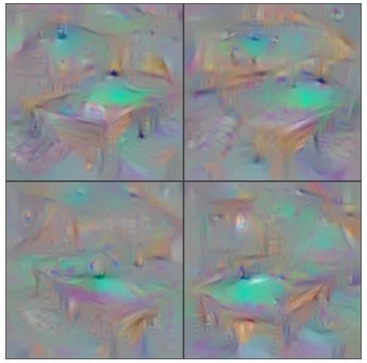
Flamingo



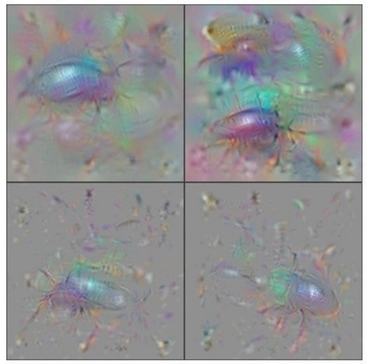
Pelican



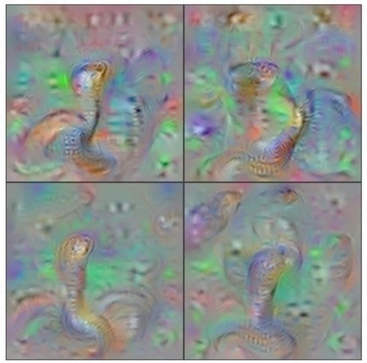
Hartebeest



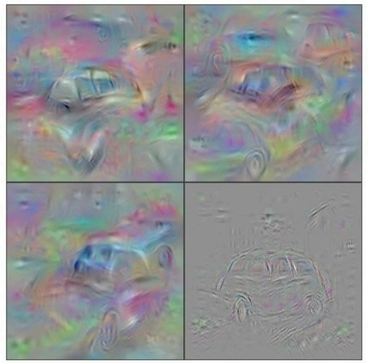
Billiard Table



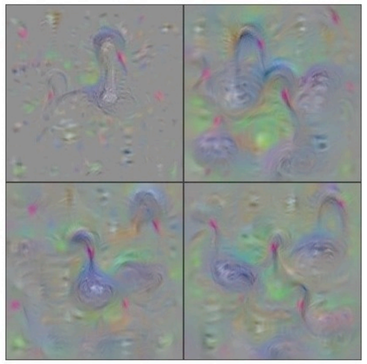
Ground Beetle



Indian Cobra



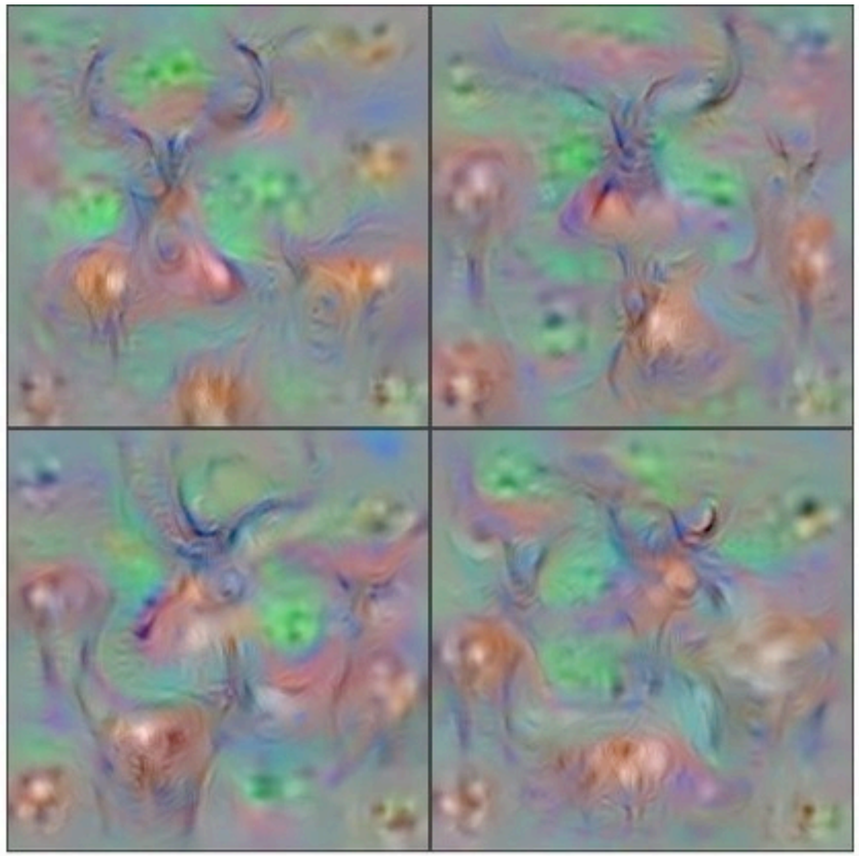
Station Wagon



Black Swan

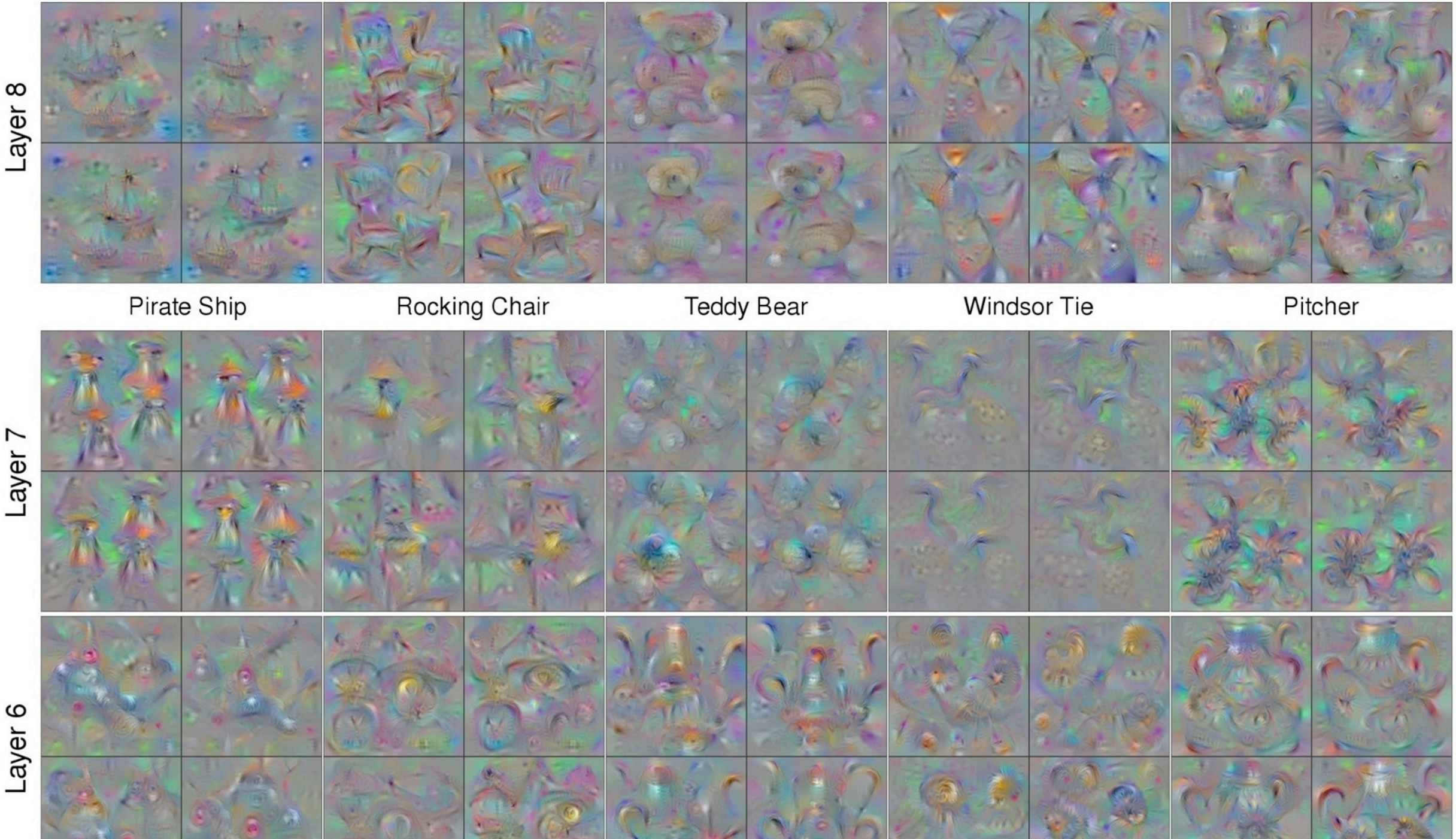
School Bus

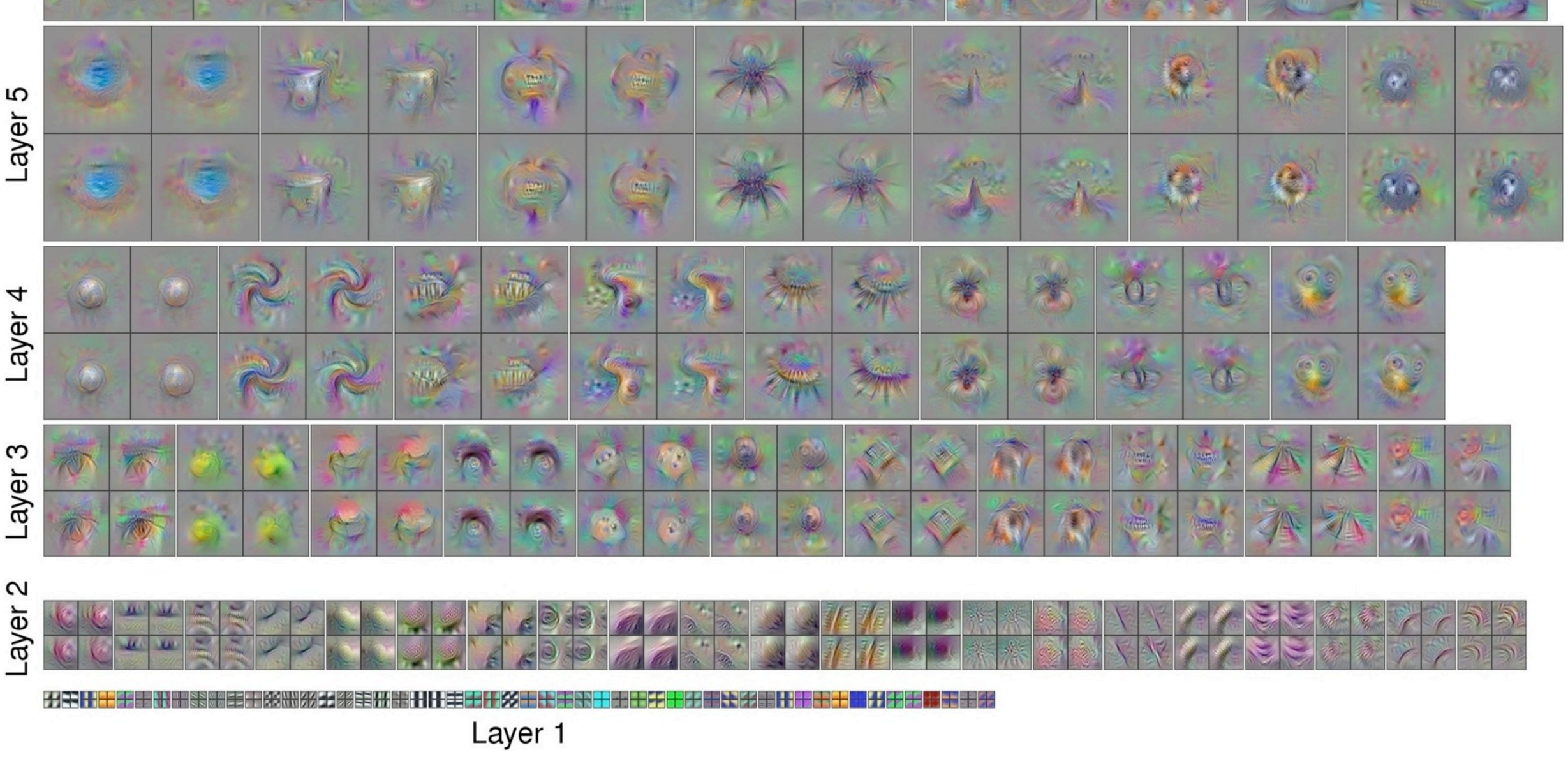
Tricycle



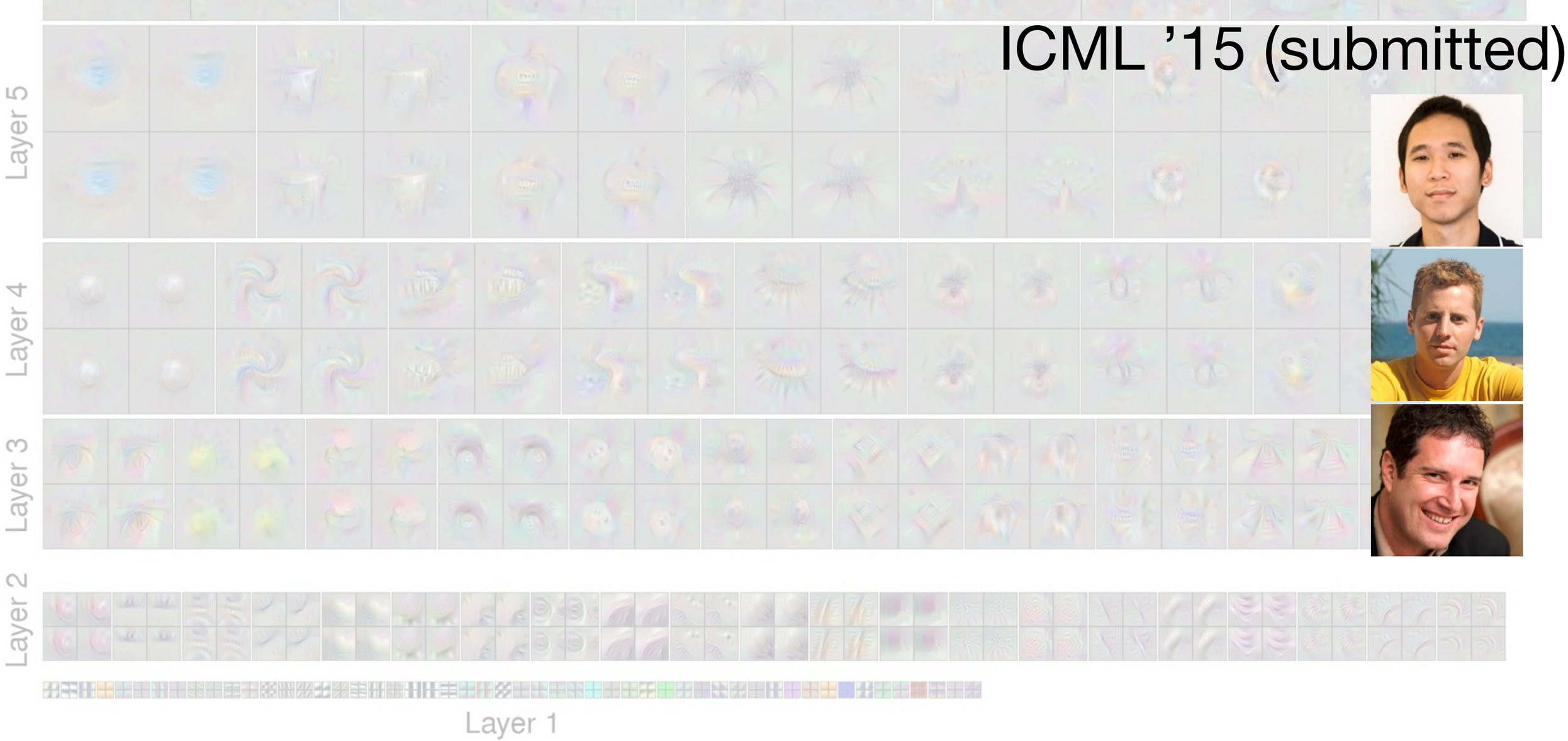
Hartebeest

School Bus

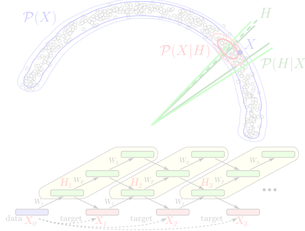

Layer 8
Pirate Ship
Rocking Chair
Teddy Bear
Windsor Tie
Pitcher
Layer 7
Layer 6

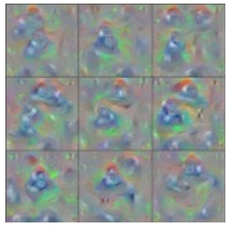

Layer 5
Layer 4
Layer 3
Layer 2
Layer 1

ICML ’15 (submitted)



Gaits	3D Design	Unsupervised	GSN	Transfer	Fooling	Visualize





Future plans

Future plans

- ICML Vis paper cleanup / code / arXiv version

Future plans

- ICML Vis paper cleanup / code / arXiv version
- Learned representations: distributed or local??

Future plans

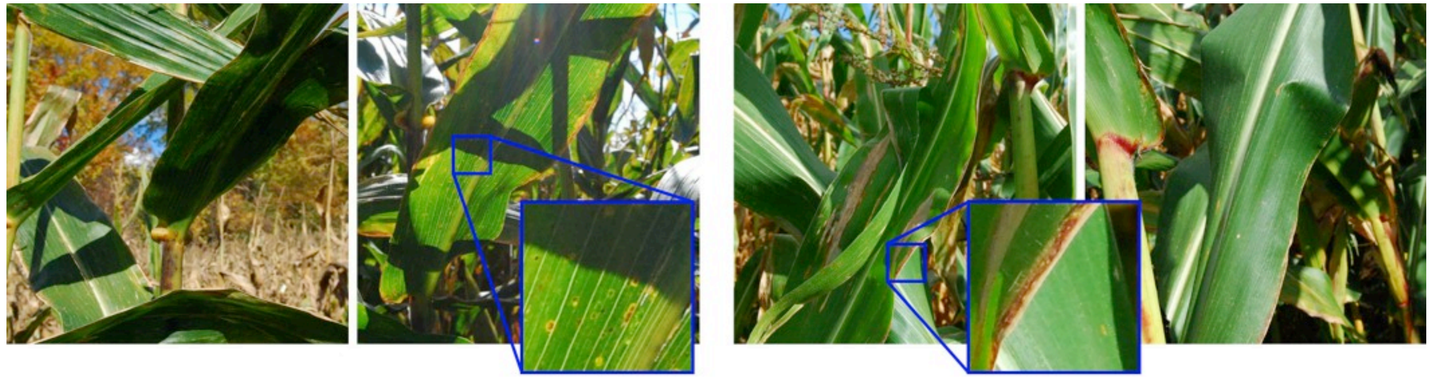
- ICML Vis paper cleanup / code / arXiv version
- Learned representations: distributed or local??

Future plans

- ICML Vis paper cleanup / code / arXiv version
- Learned representations: distributed or local??

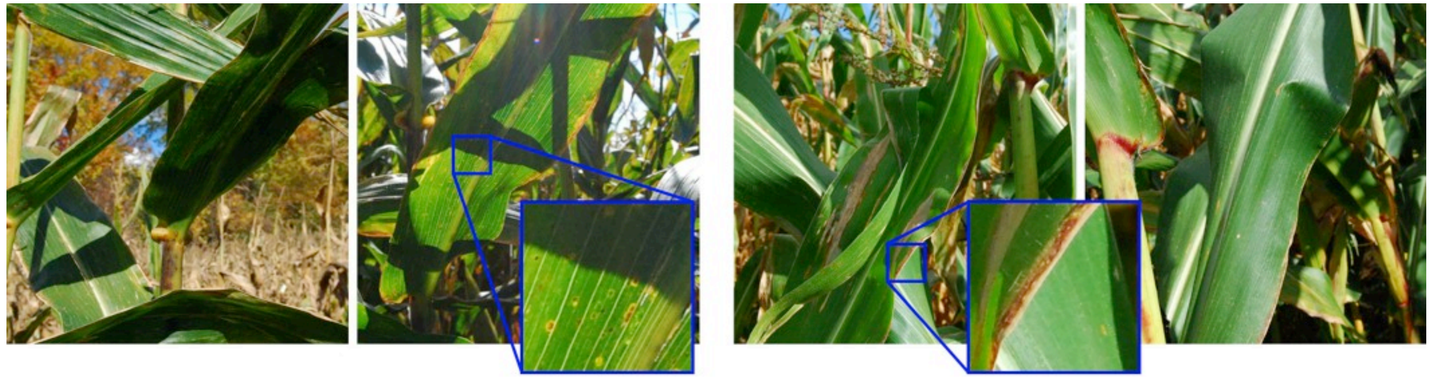
Future plans

- ICML Vis paper cleanup / code / arXiv version
- Learned representations: distributed or local??
- Applications...



Future plans

- ICML Vis paper cleanup / code / arXiv version
- Learned representations: distributed or local??
- Applications...
- Robots



Slides: http://s.yosinski.com/a-exam.pdf

Hod Lipson

Slides: http://s.yosinski.com/a-exam.pdf

Hod Lipson

Jeff Clune

Slides: http://s.yosinski.com/a-exam.pdf

Hod Lipson

Jeff Clune

Yoshua Bengio

Slides: http://s.yosinski.com/a-exam.pdf

Hod Lipson

Jeff Clune

Yoshua Bengio

Slides: http://s.yosinski.com/a-exam.pdf

Funding:

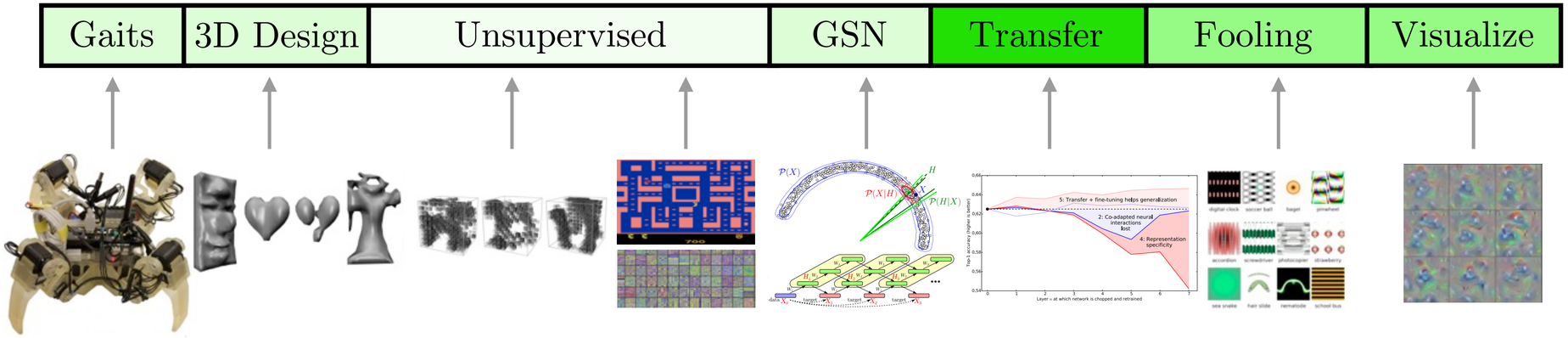
- NSF CDI Grant ECCS 0941561
- DARPA Matter Compiler Grant
- NSERC
- Ubisoft
- CIFAR
- NASA Space Technology Research Fellowship

Thanks!

Hod Lipson

Jeff Clune

Yoshua Bengio



Slides: http://s.yosinski.com/a-exam.pdf

Funding:

- NSF CDI Grant ECCS 0941561
- DARPA Matter Compiler Grant
- NSERC
- Ubisoft
- CIFAR
- NASA Space Technology Research Fellowship