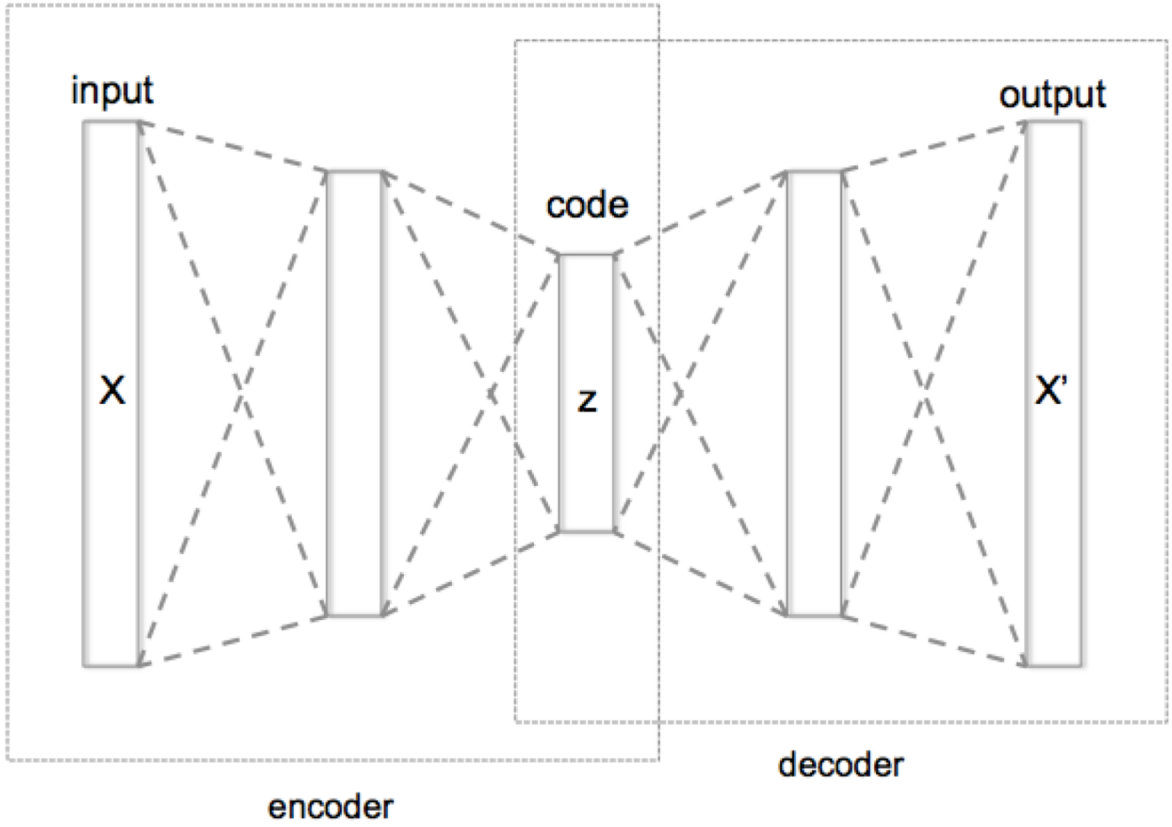
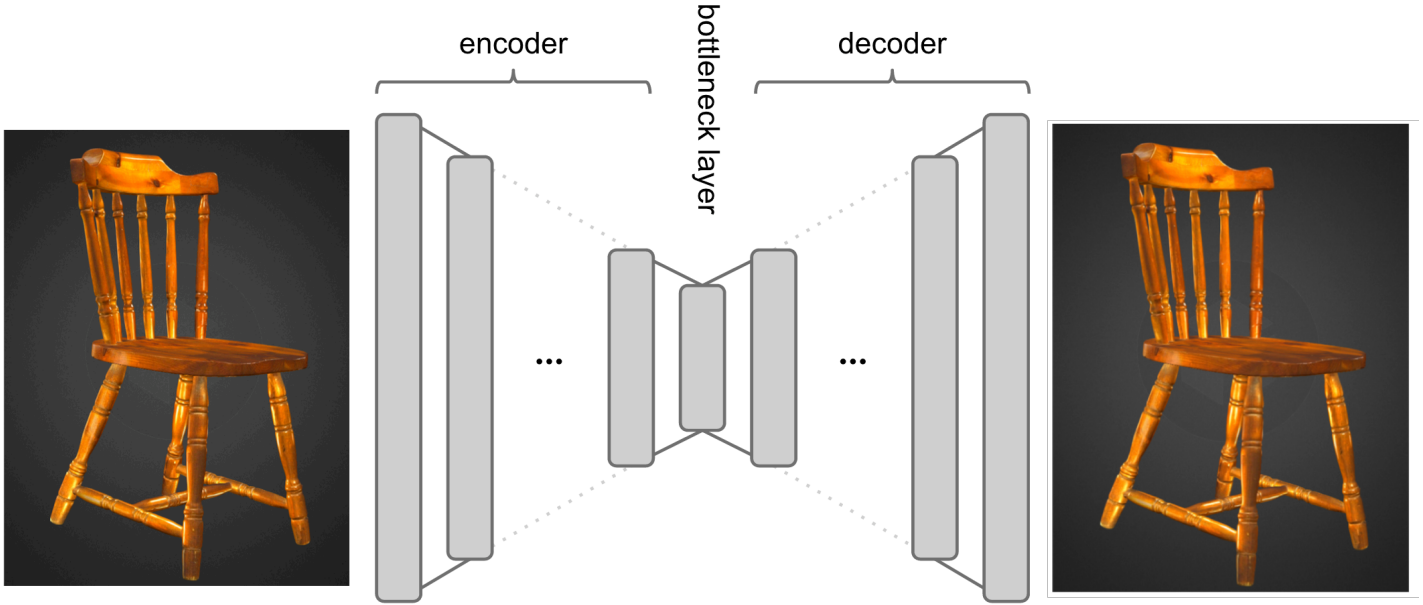
Learning Systems 2018:
Lecture 2– Introduction to Deep Learning Part 2

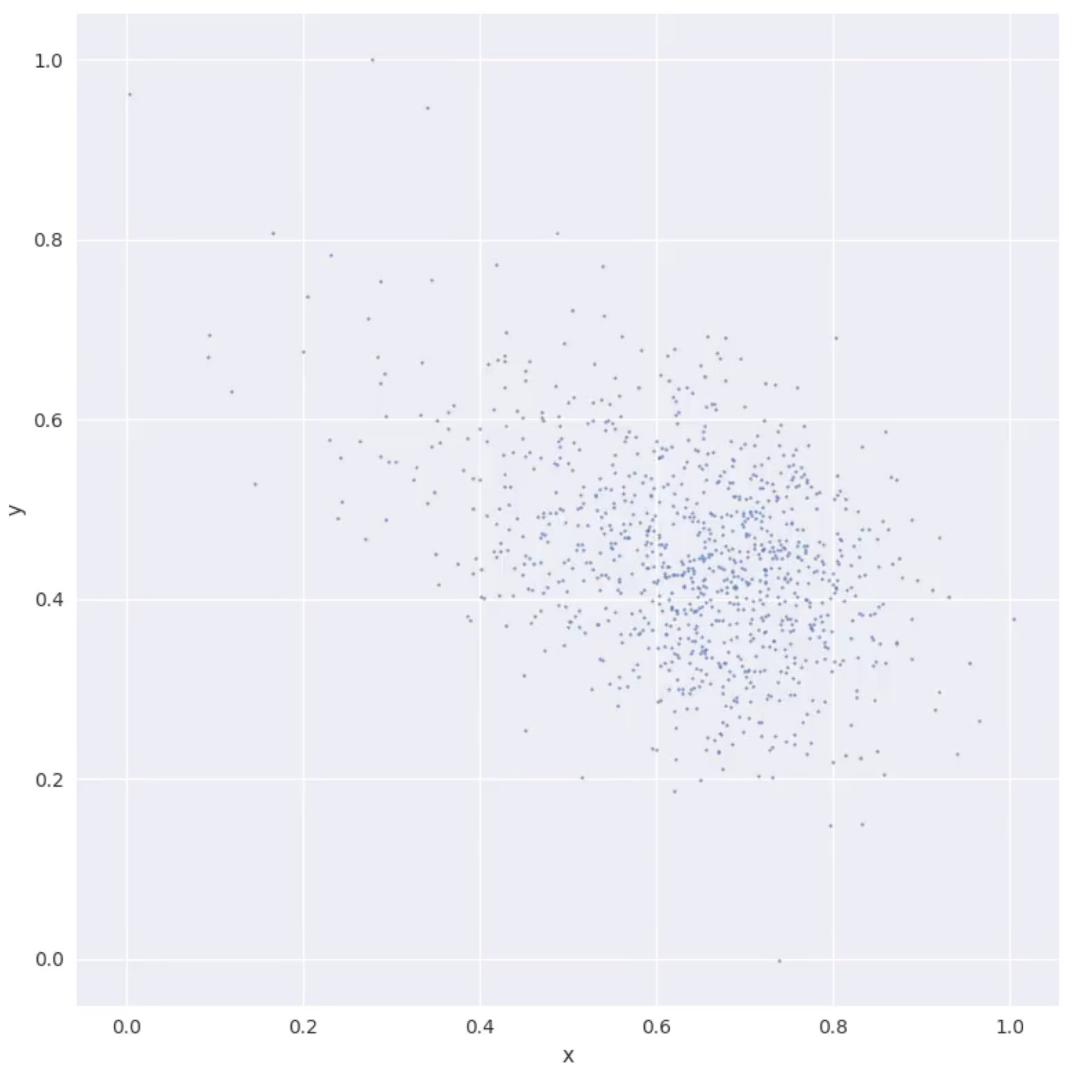
Crescat scientia; vita excolatur

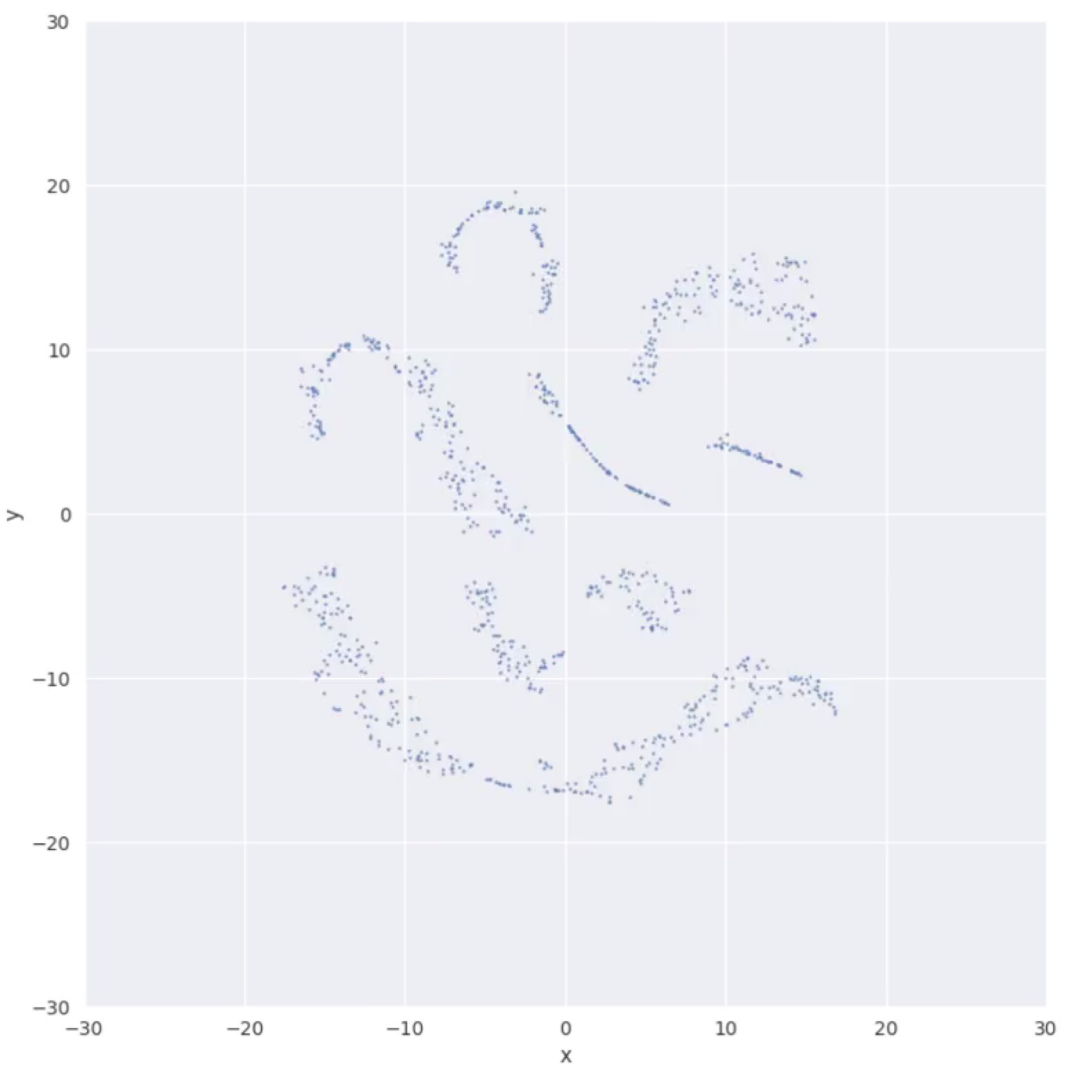
Ian Foster and Rick Stevens
Argonne National Laboratory
The University of Chicago


input
output
code
X
z
X'
encoder
decoder

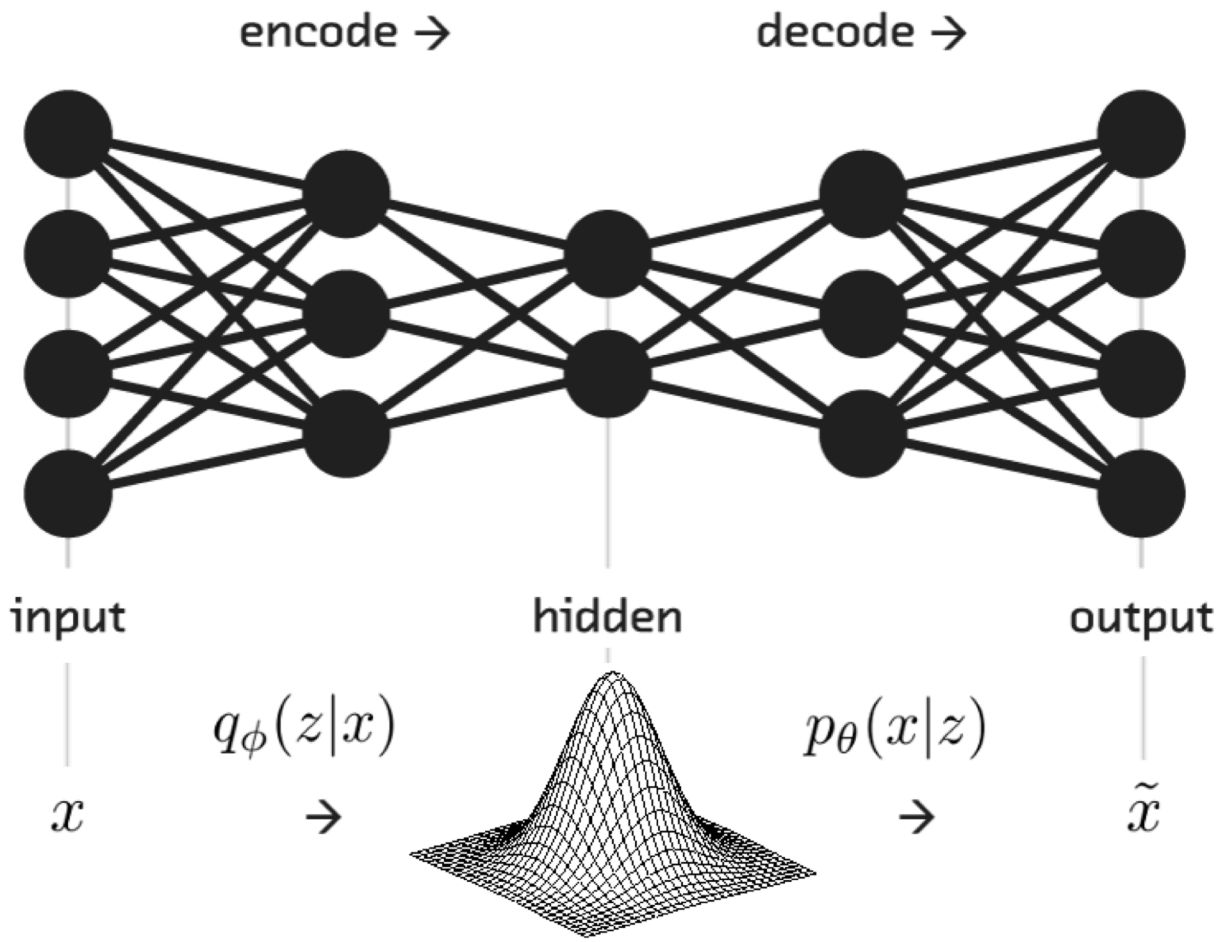
Autoencoder



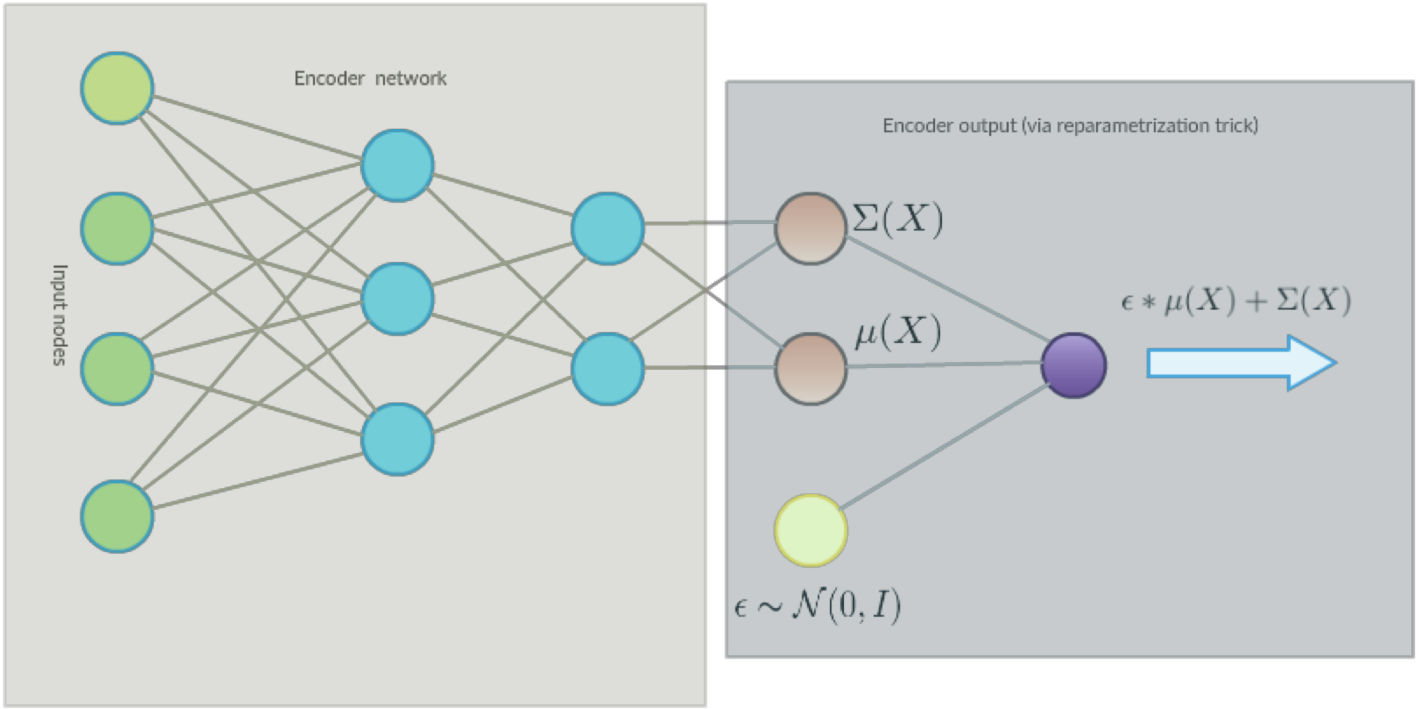

1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x


30
20
10
0
−10
−20
−30
y
−30
−20
−10
0
10
20
30
x

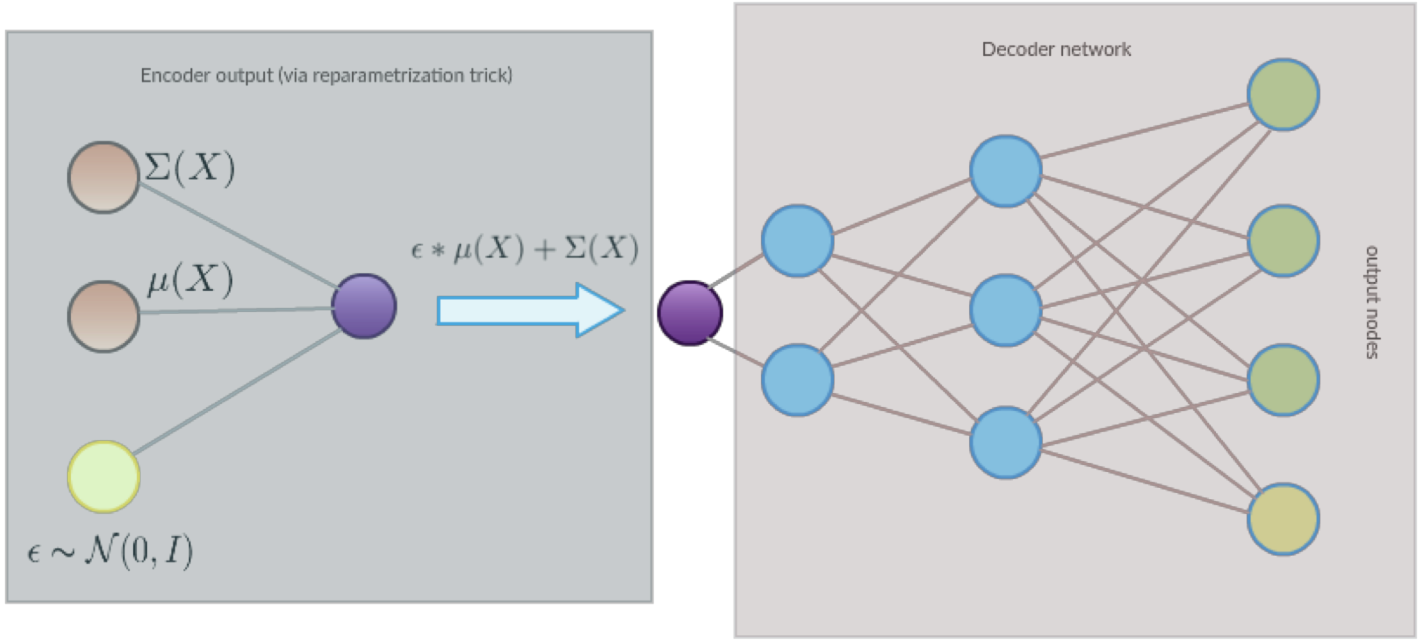
Variational Autoencoder


encode →
decode →
input
hidden
output
$q_\phi(z|x)$
$p_\theta(x|z)$
x
→
→
$\tilde{x}$

Encoder to Latent



Latent to Decoder

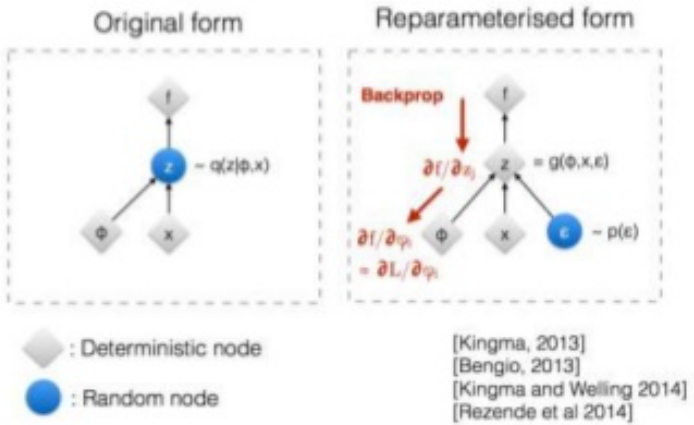


Re-Parameterization Trick

Backpropagation not possible through random sampling!

$$z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \varepsilon_i$$

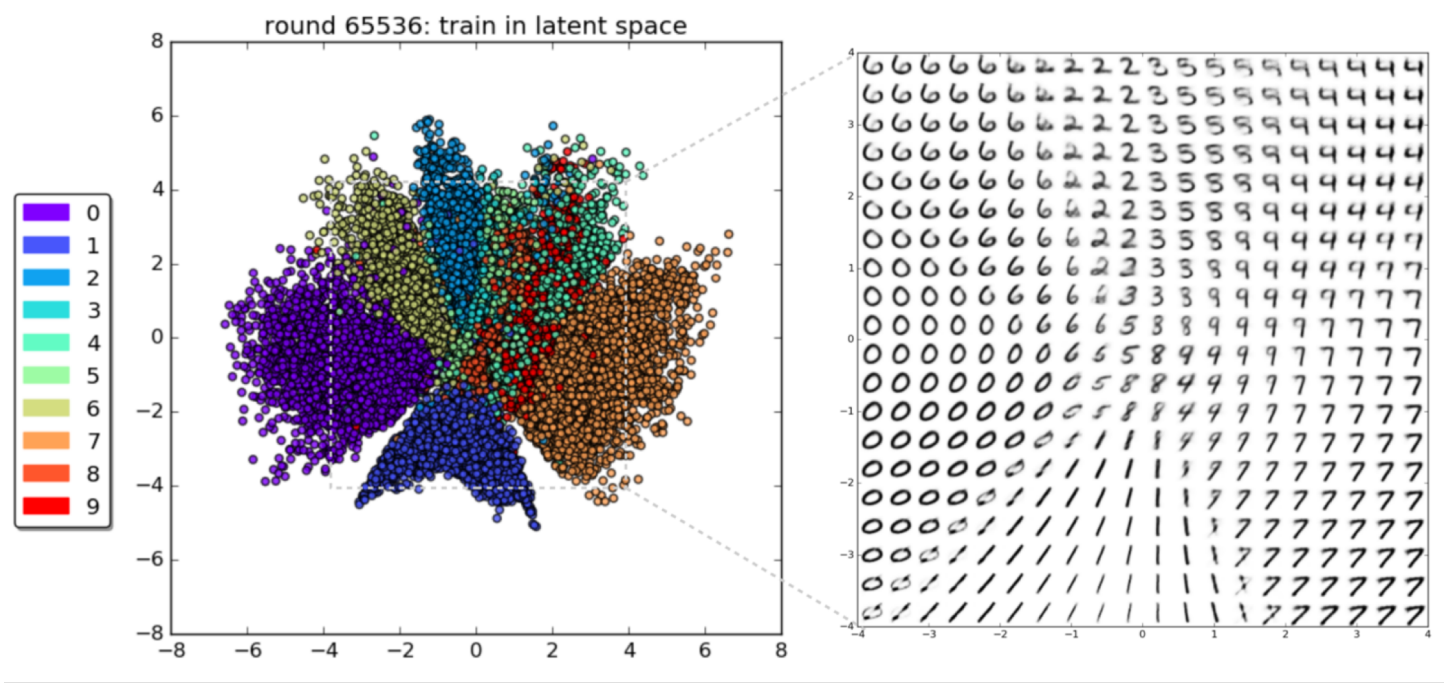
$$\varepsilon_i \sim N(0,1)$$



[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

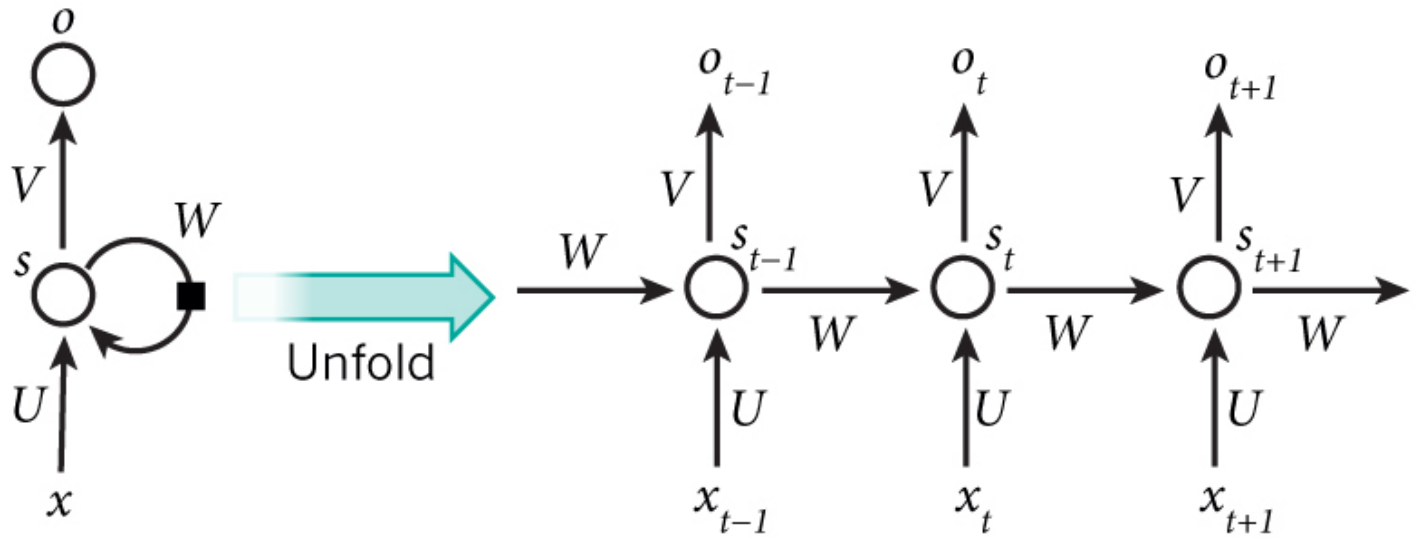
[https://arxiv.org/abs/1609.04468]

MNIST Latent Space Sampling

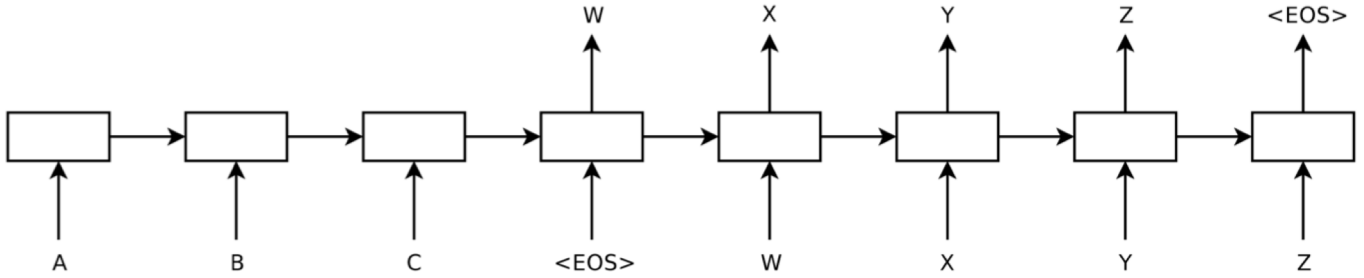


Recurrent Neural Network
Long-Short Term Memory

Recurrent Neural Network



Seq2seq Neural Networks



$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

where $(x_1, \ldots, x_T)$ is input sequence

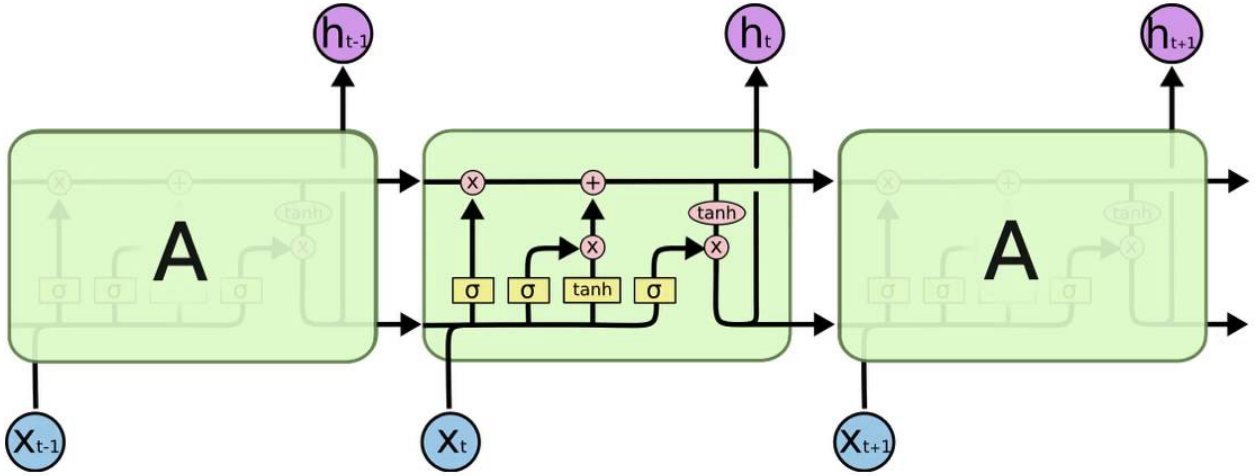
$y_1, \ldots, y_{T'}$ is corresponding output sequence

v the last hidden state of the LSTM

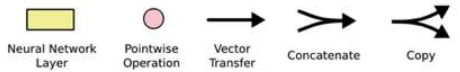
Ref: Sequence to Sequence Learning with Neural Networks; https://arxiv.org/pdf/1409.3215.pdf

The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM.

Long-Short Term Memory module: LSTM



long-short term memory modules used in an RNN



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Variational Autoencoder

+

Recurrent Neural Network

=

Generative Chemistry

Automatic Chemical Design using Variational Autoencoders

Alán Aspuru-Guzik
Harvard University

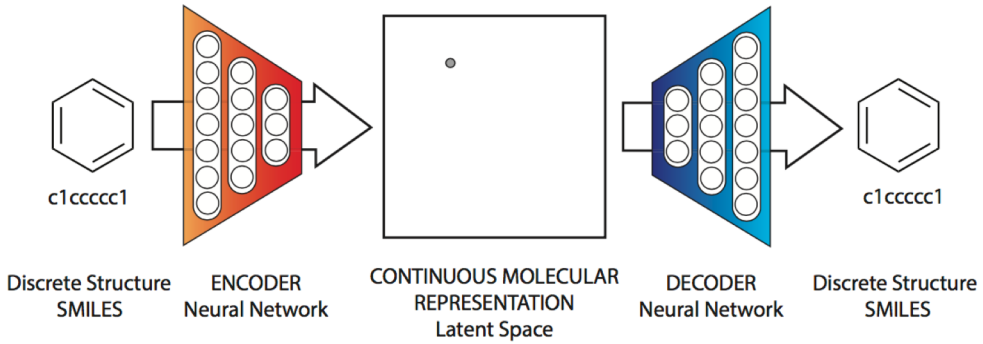
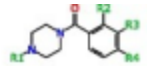
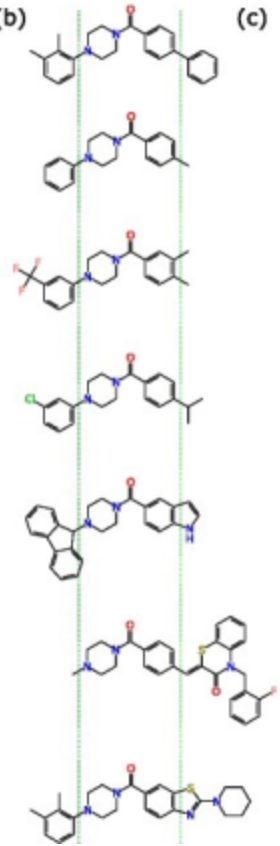


Figure 1: A diagram of the proposed autoencoder for molecular design. Starting from a discrete molecular representation, such as a SMILES string, the encoder network converts each molecule into a vector in the latent space, which is effectively a continuous molecular representation. Given a point in the latent space, the decoder network produces a corresponding SMILES string.

(a)



(b)



(c)

Cc1cccc(N2CCN(CC2)C(=O)c3ccc(cc3)c4ccccc4)c1C

Cc1ccc(cc1)C(=O)N2CCN(CC2)c3ccccc3

Cc1ccc(cc1C)C(=O)N2CCN(CC2)c3ccccc(c3)C(F)(F)F

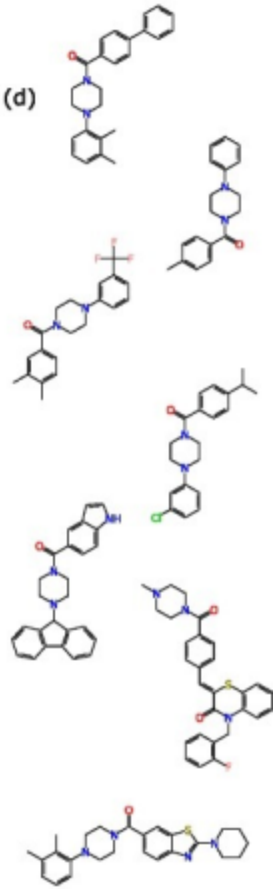
CC(C)c1ccc(cc1)C(=O)N2CCN(CC2)c3cccc(Cl)c3

O=C(N1CCN(CC1)C2c3ccccc3c4ccccc24)c5ccc6[nH]ccc6c5

CN1CCN(CC1)C(=O)c2ccc(cc2)\C=C3/Sc4ccccc4N(Cc5ccccc5F)C3=O

Cc1cccc(N2CCN(CC2)C(=O)c3ccc4nc(sc4c3)N5CCCCC5)c1C

(d)



CNN to RNN (LSTM)

- Three layers of CNN filters 9,9,11 and 9,9,10 convolutional kernels
- Two fully connected layers of size 435, 292

- Three layers of gated RNN with hidden dimension of 501

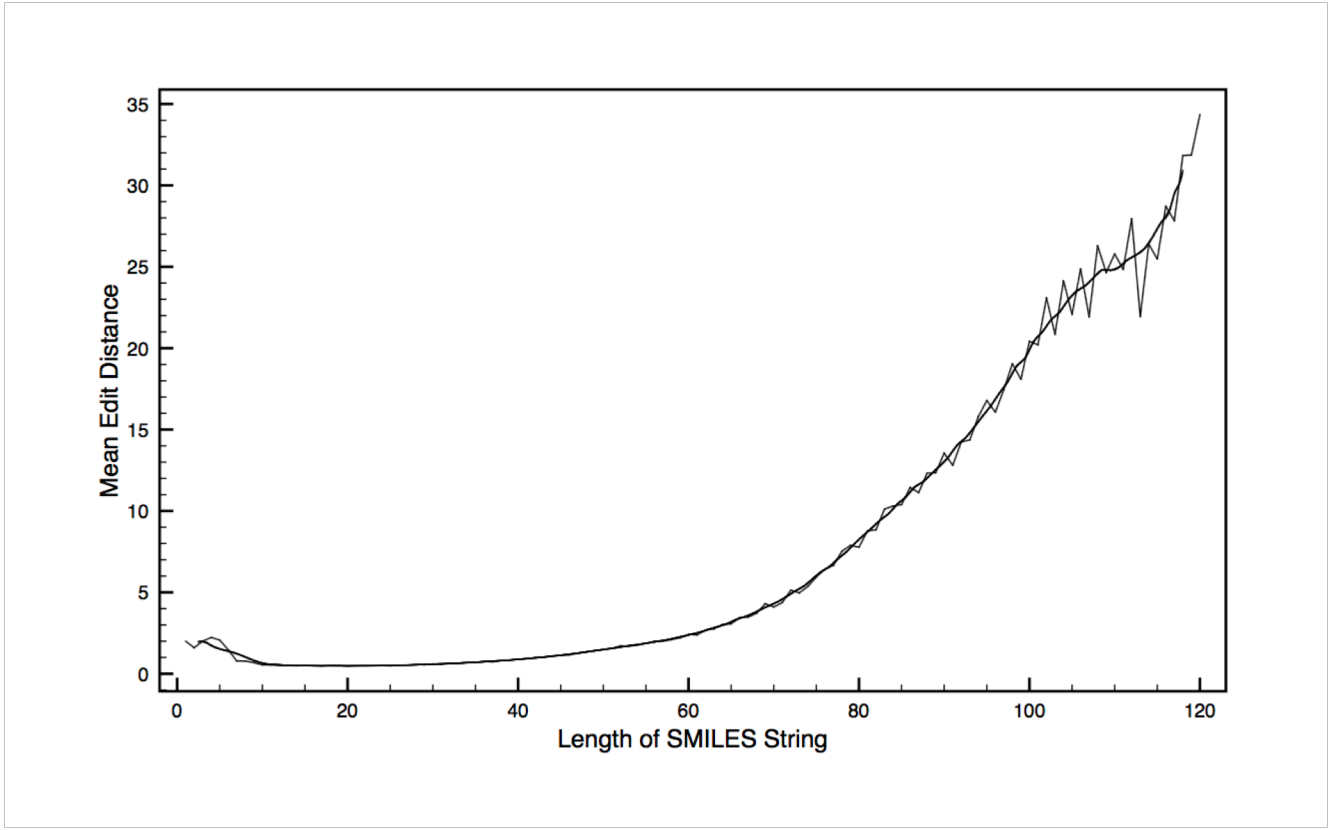
- Input SMILES upto 120 characters

Aspuru-Guzik et. al. Results

Molecular family	Autoencoder training loss	Latent dimension	Training set reconstruction %	Test set reconstruction %
drug-like	naïve	56	99.1	98.3
drug-like	variational	292	96.4	95.3
OLED	naïve	56	96.7	91.2
OLED	variational	292	91.4	79.4

Table 1: Reconstruction accuracy for the deep autoencoders used in this work. Accuracy is defined as the percentage of correct characters in decoded SMILES strings. An autoencoder with a large enough latent dimension could achieve perfect reconstruction, but exploration of the latent space tends to become more difficult as the latent dimension increases.

Error rate for the Molecular VAE



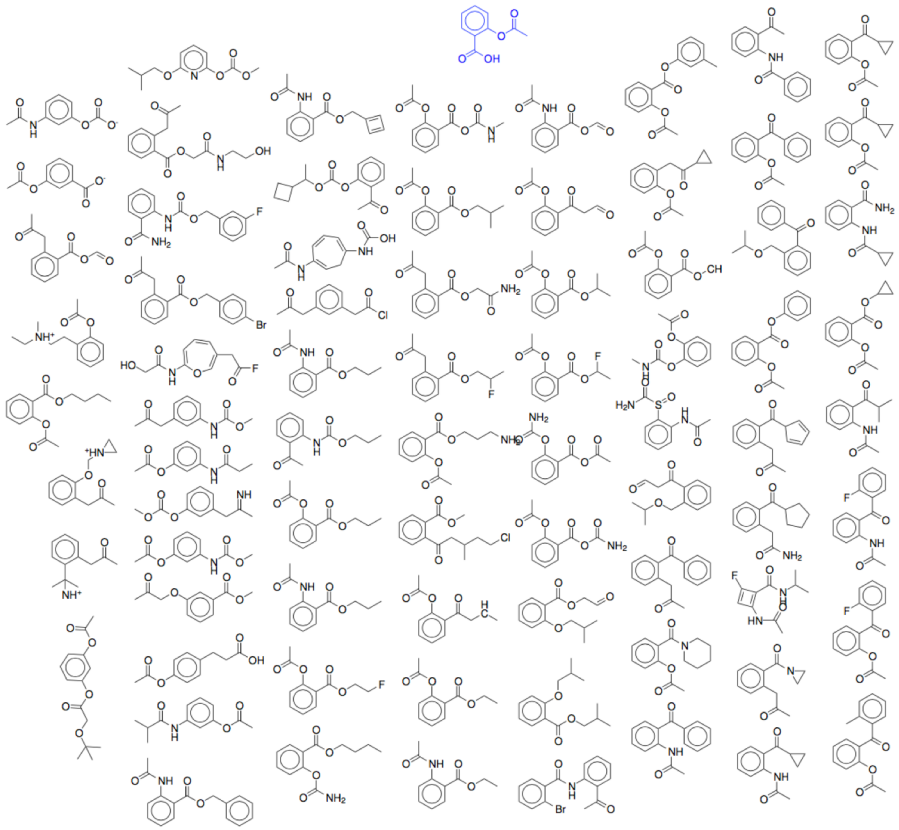


Figure 4: Molecules decoded from randomly-sampled points in the latent space of a variational autoencoder, near to a given molecule (aspirin [2-(acetyloxy)benzoic acid], highlighted in blue).

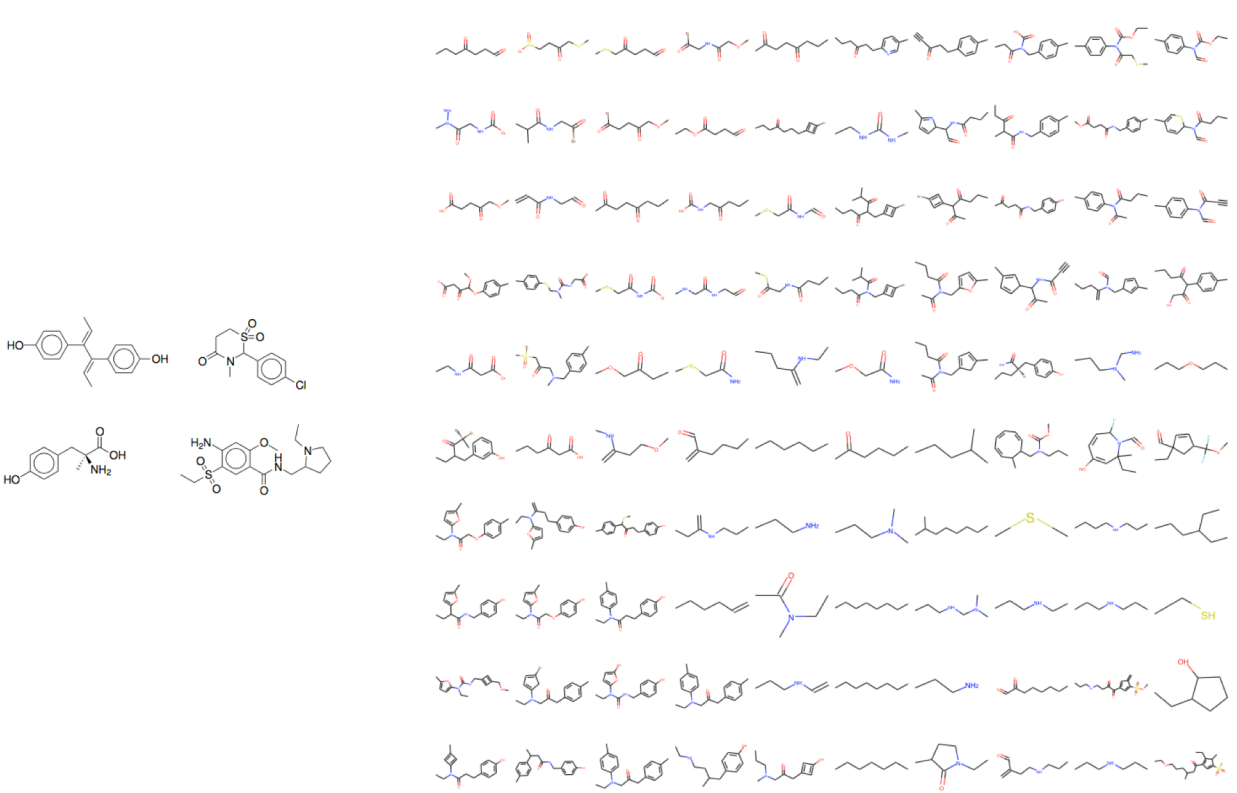
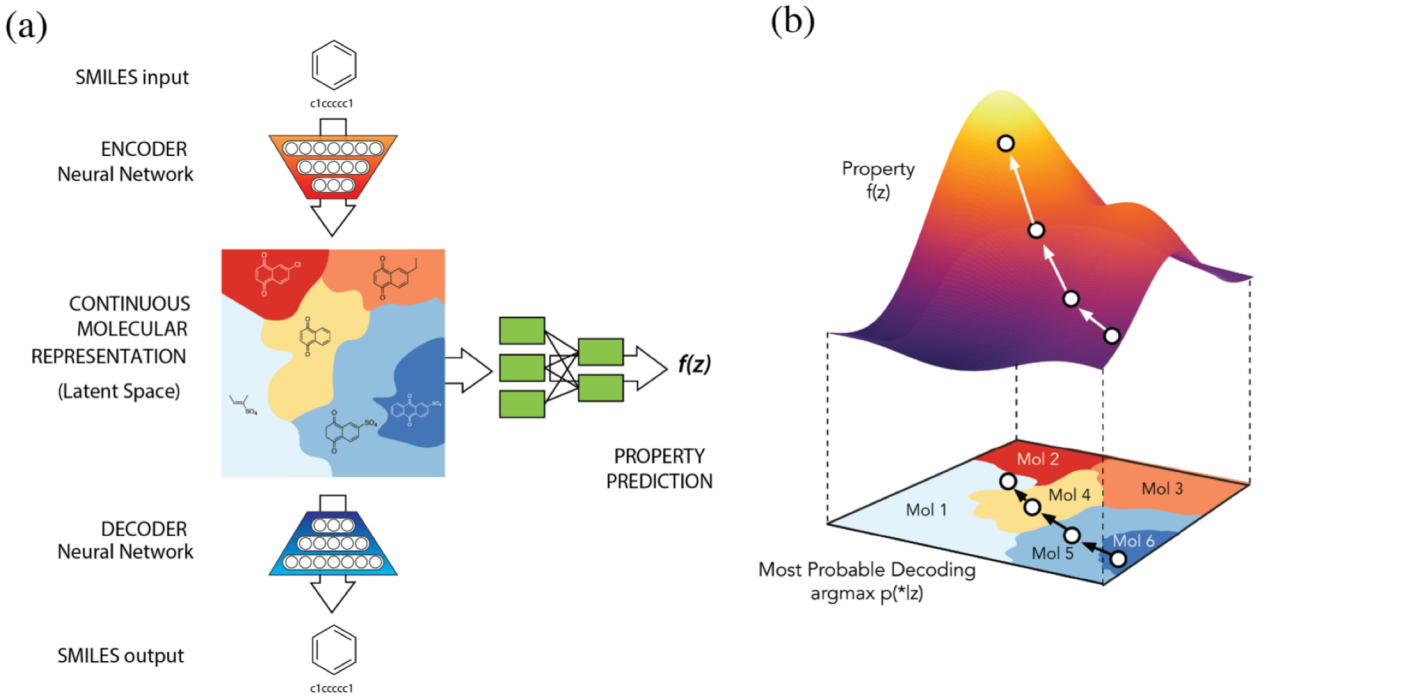


Figure 5: Interpolation. Two-dimensional interpolation between four random drugs. *Left* Starting molecules encoded, whose decodings correspond to the respective four corners of the figure to the right. *Right* Decodings of interpolating linearly between the latent representations of the four molecules to the right.

Associating Properties with Latent Space Representation



Visualization of Latent Space

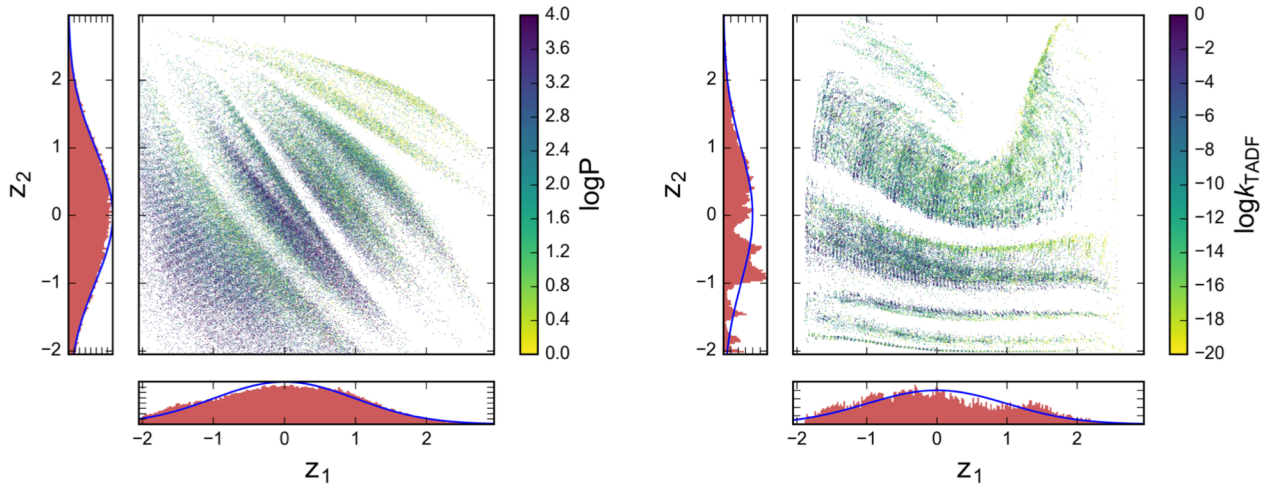


Figure 2: Projection of the molecular training sets onto learned two-dimensional latent spaces. The one-dimensional histograms show the distribution of the training data along each dimension, overlaid with the Gaussian prior imposed in the variational autoencoder. The points are colored along a chemical property that is relevant to their function, and will be the target of optimization experiments. *Left*: A natural library of drug-like molecules, colored by their predicted water-octanol partition coefficient. *Right*: A combinatorially-generated library of organic LED molecules, colored by their predicted delayed fluorescent emission rate (k_{TADF} in μs^{-1}).

Mining the Learned Representations

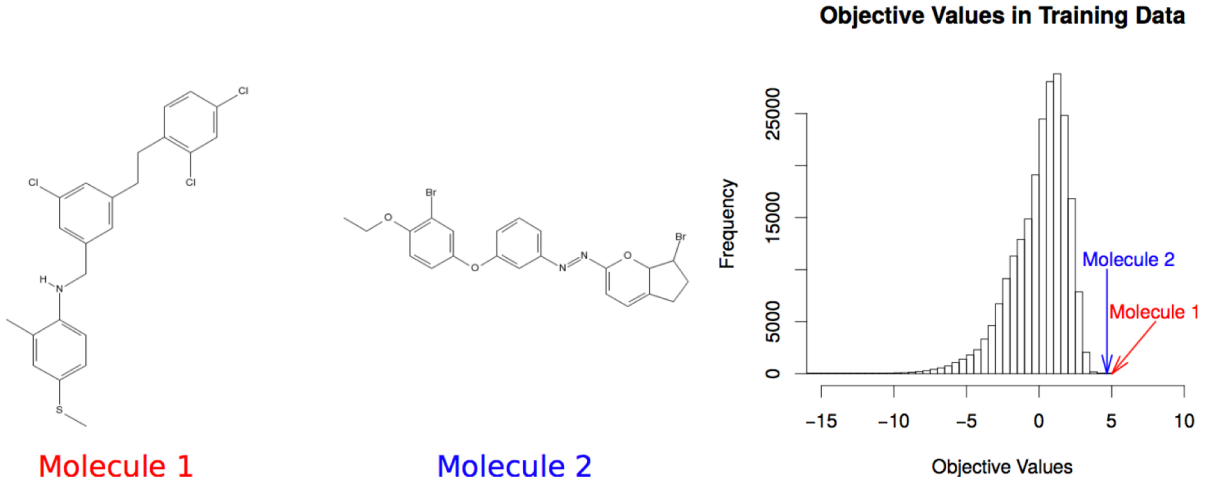
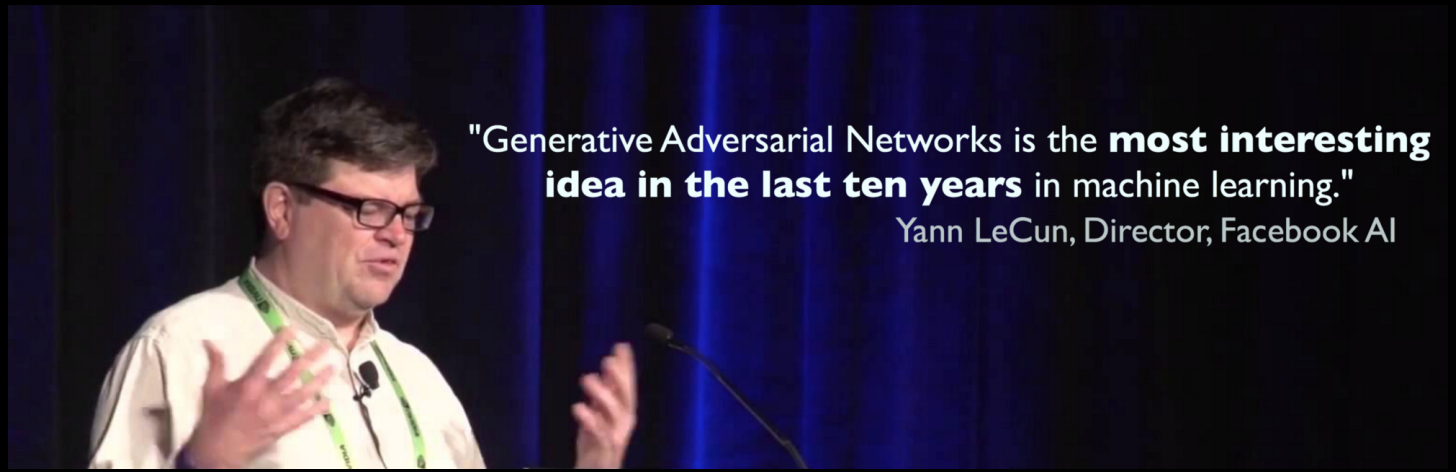


Figure 7: *Left*: Molecules generated by the optimization process with better score values than any other molecule in the training data. *Right*: Histogram of objective values in the training data.

Generative Adversarial Networks


"Generative Adversarial Networks is the **most interesting idea in the last ten years** in machine learning."
Yann LeCun, Director, Facebook AI

What are Generative Models?

Key Idea: our model cares about what distribution generated the input data points, and we want to mimic it with our probabilistic model. **Our learned model should be able to make up new samples from the distribution, not just copy and paste existing samples!**

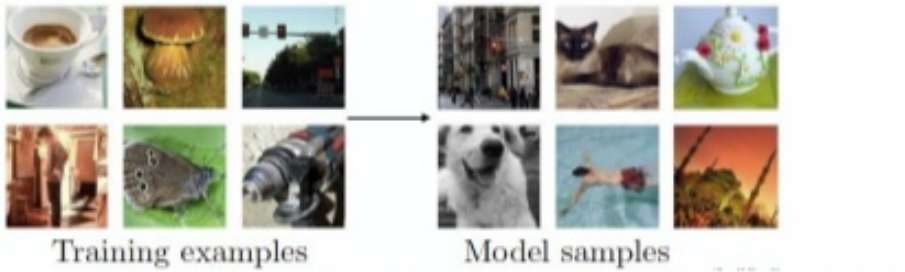
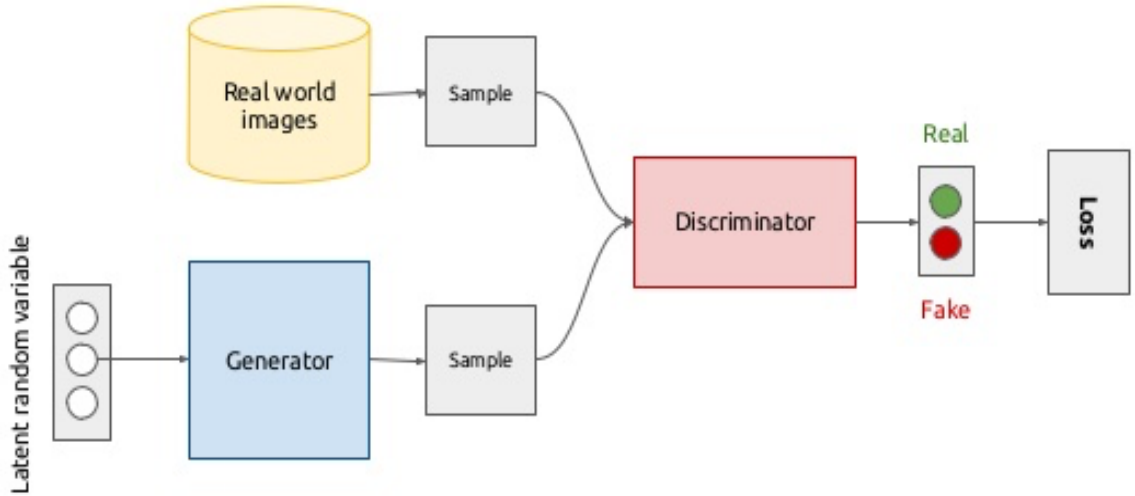
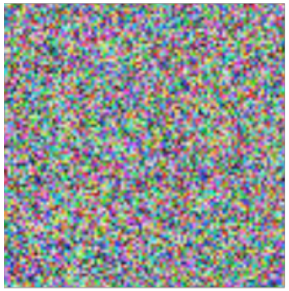


Figure from NIPS 2016 Tutorial: Generative Adversarial Networks (I. Goodfellow)

Generative adversarial networks (conceptual)

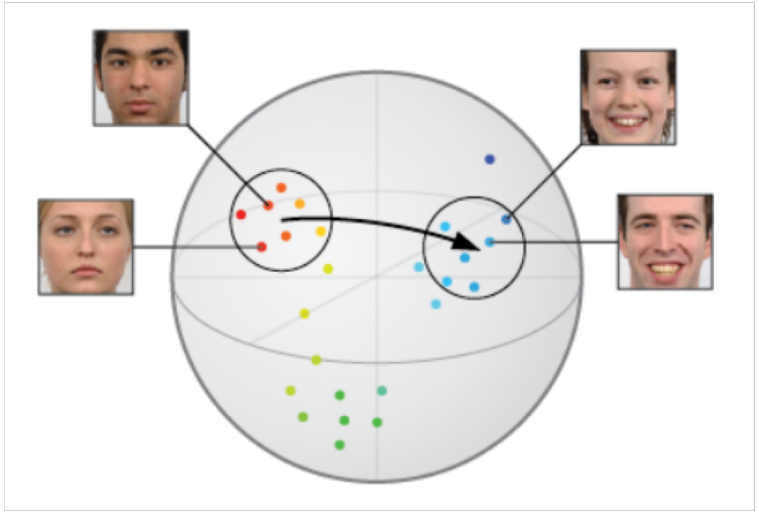


Noise ~ N(0,1)

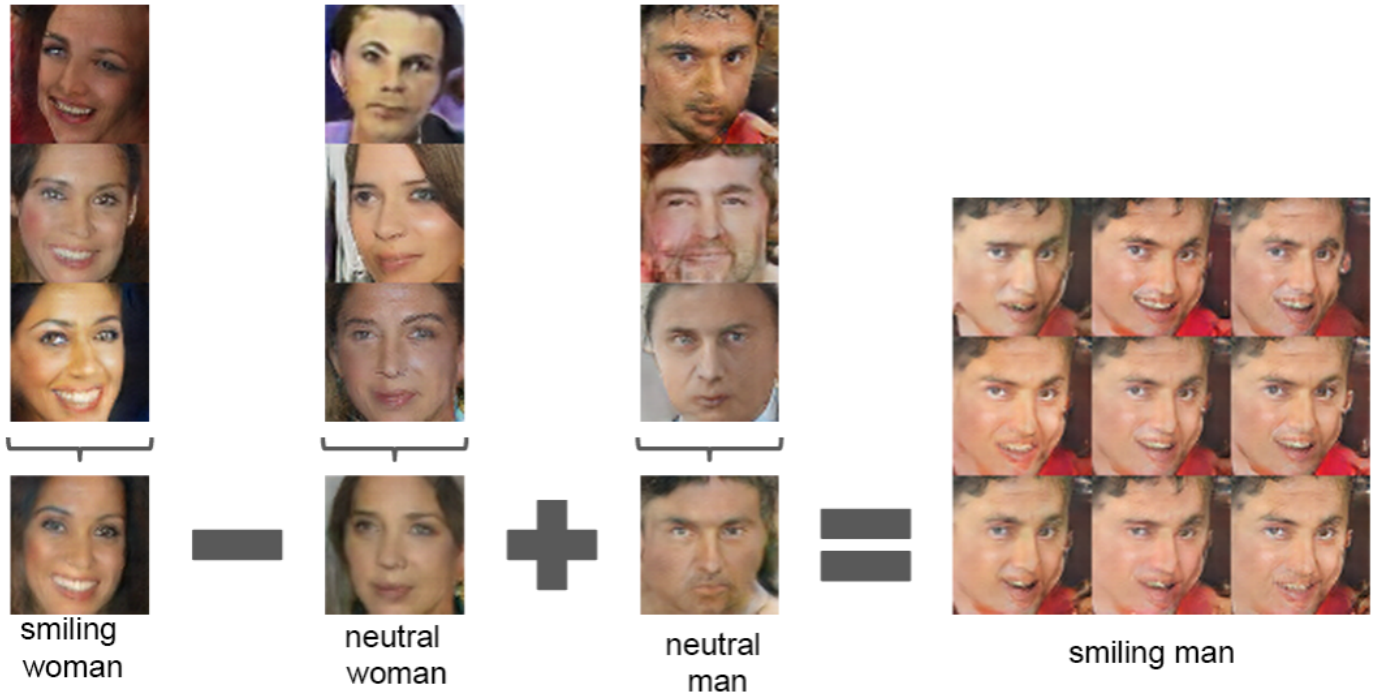


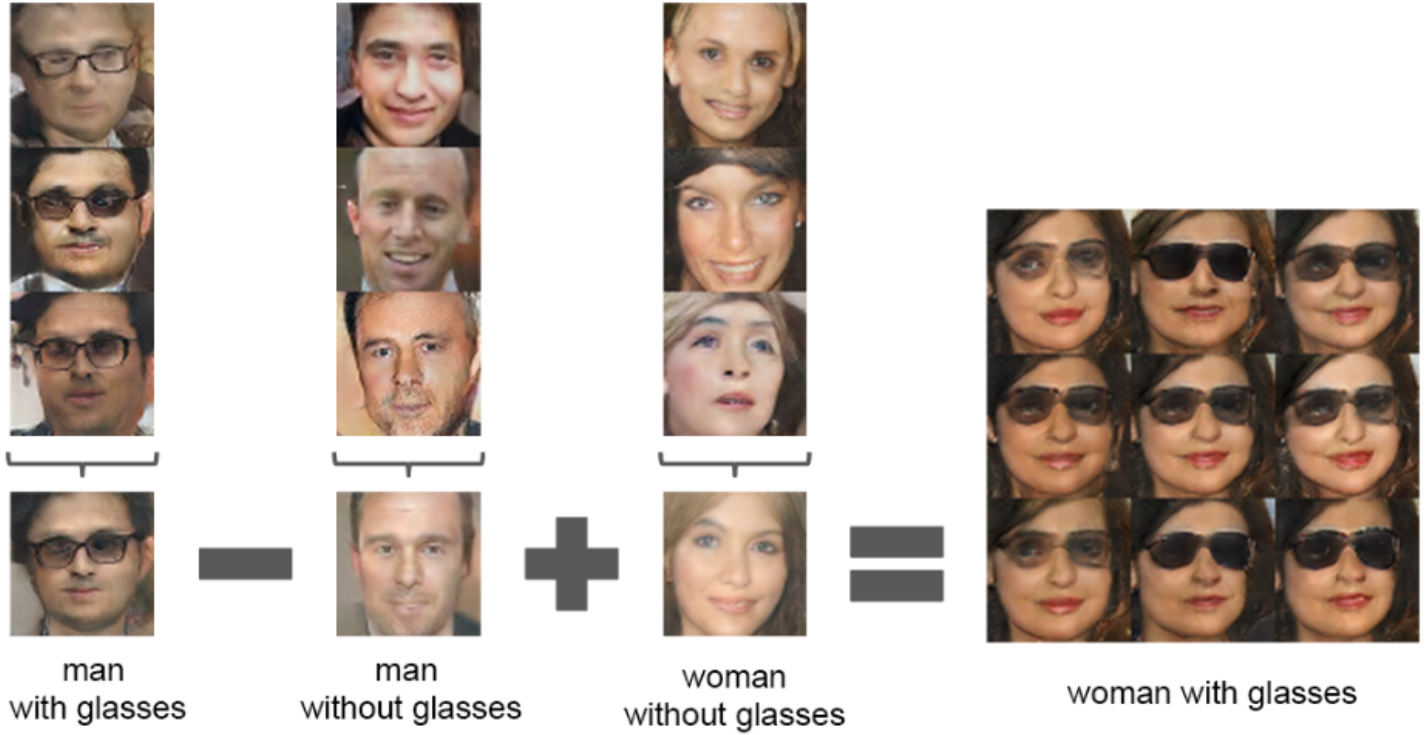
Generative Model

If you do it right!

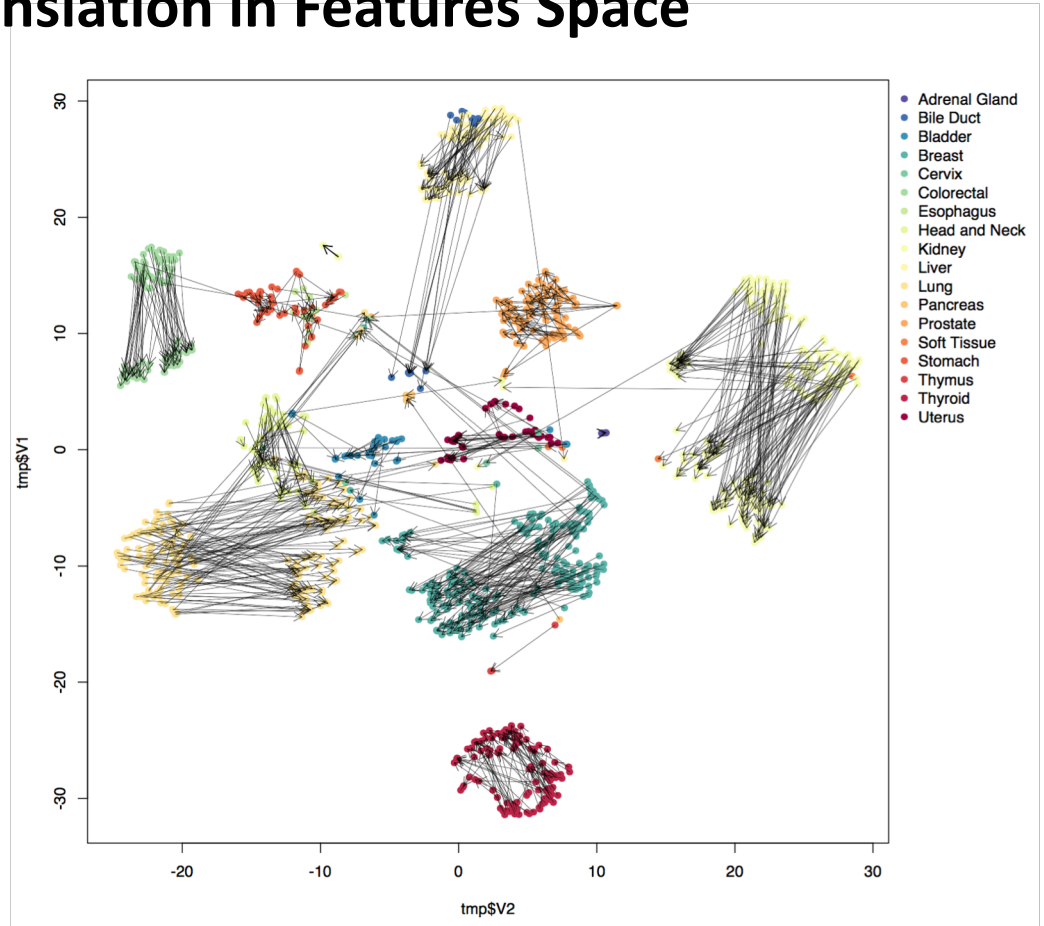


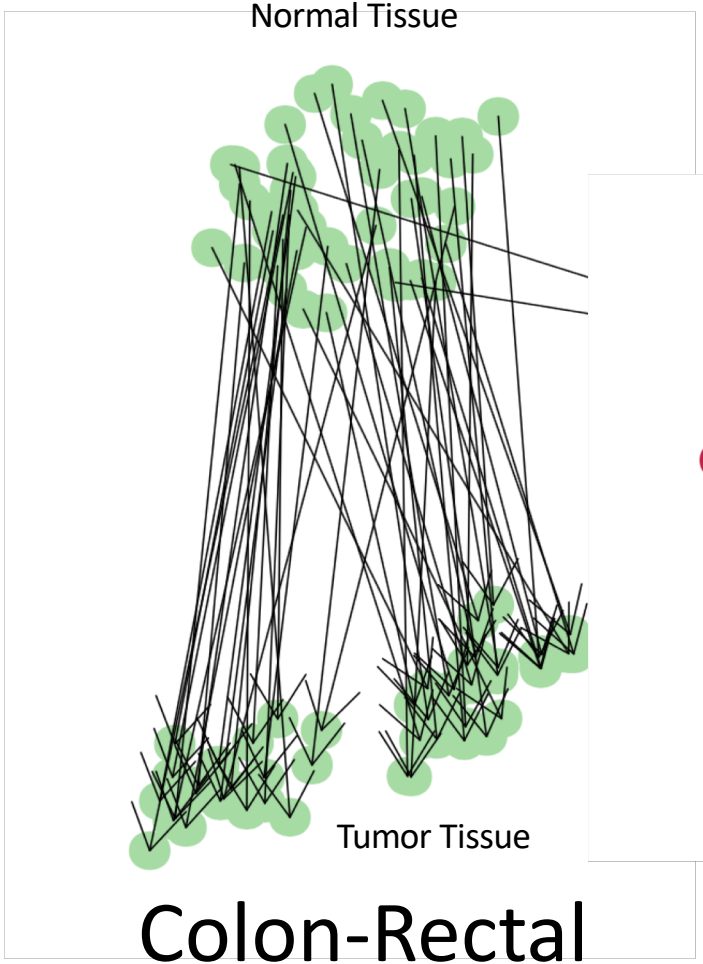
Arithmetic in the Latent Vector Space


smiling woman
neutral woman
neutral man
smiling man


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

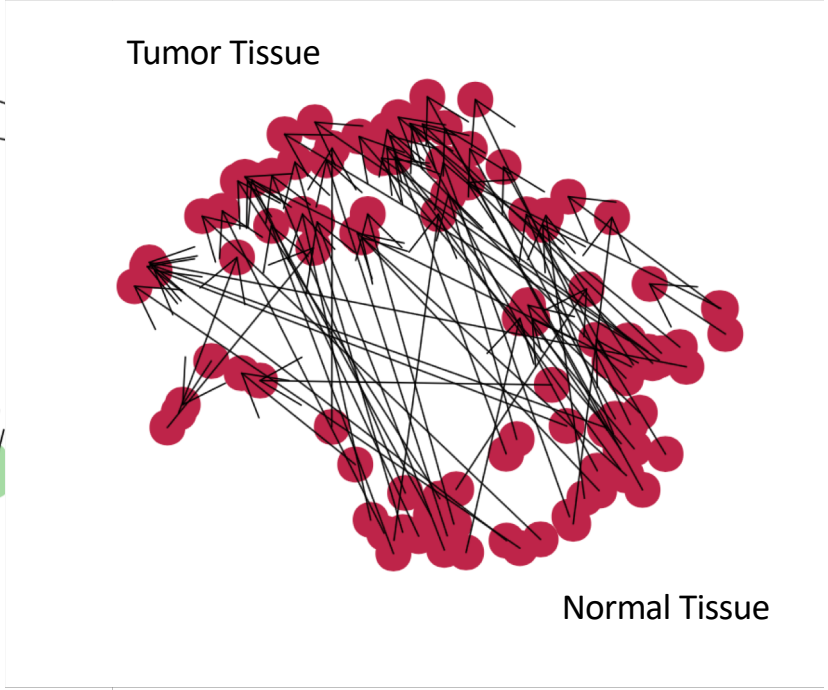
t-sne Plot of Matched Normal Pairs Showing Translation in Features Space





Colon-Rectal

Uterus



Cycle Consistent Generative Adversarial Networks

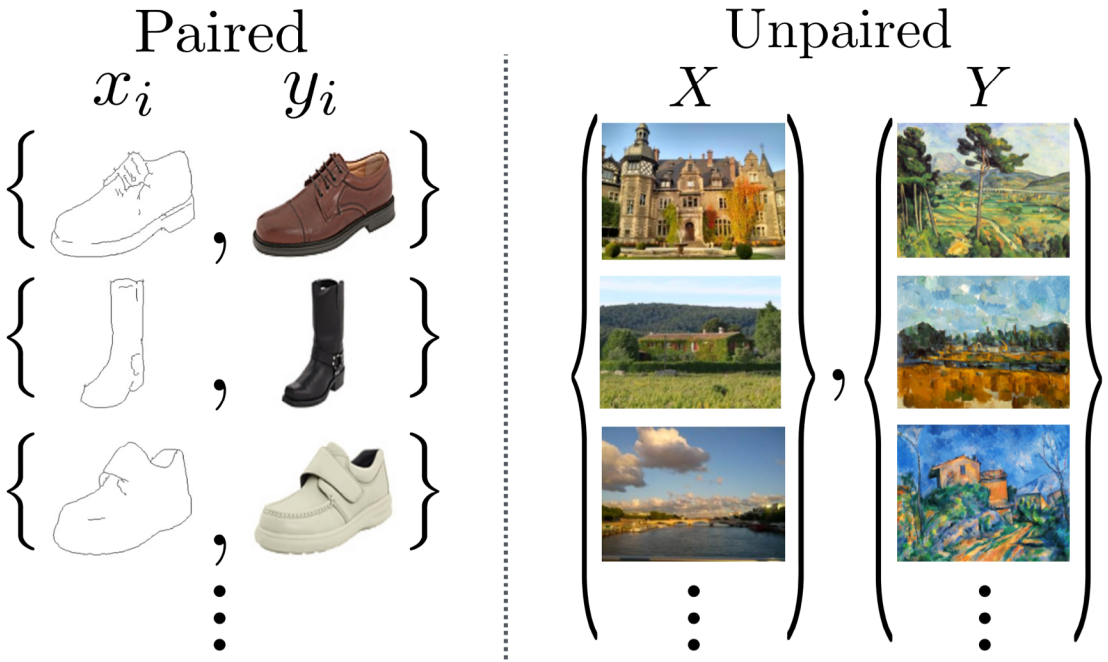


Figure 2: *Paired* training data (left) consists of training examples $\{x_i, y_i\}_{i=1}^N$, where the correspondence between x_i and y_i exists [21]. We instead consider *unpaired* training data (right), consisting of a source set $\{x_i\}_{i=1}^N$ ($x_i \in X$) and a target set $\{y_j\}_{j=1}$ ($y_j \in Y$), with no information provided as to which x_i matches which y_j.

Two Mapping Functions $G: X \rightarrow Y$ and $F: Y \rightarrow X$ and discriminators D_Y and D_X

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x$$

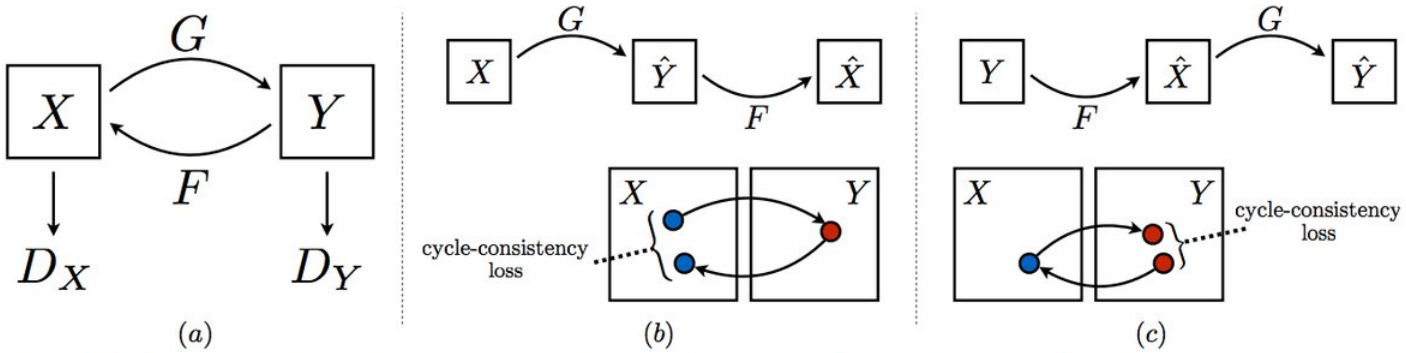
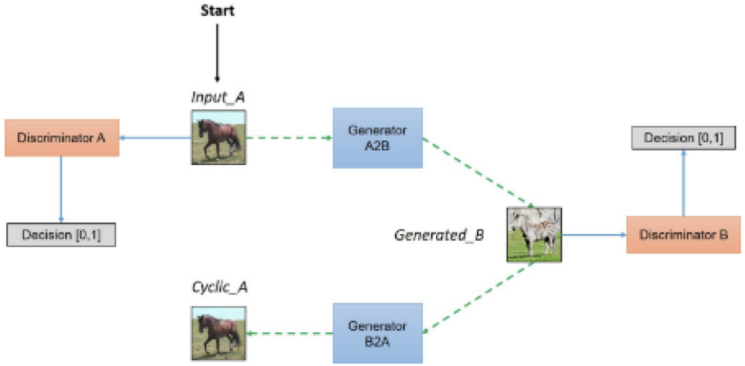
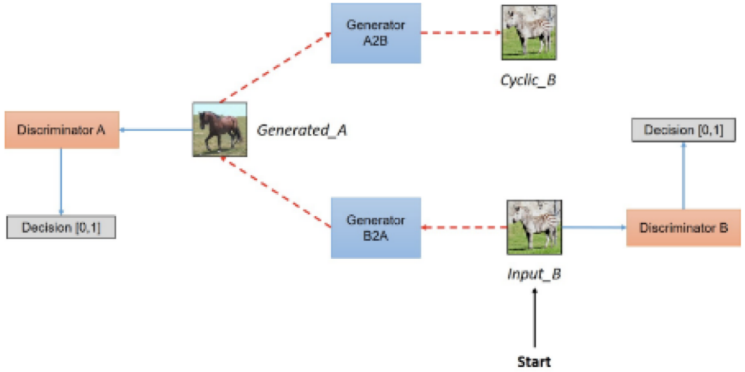


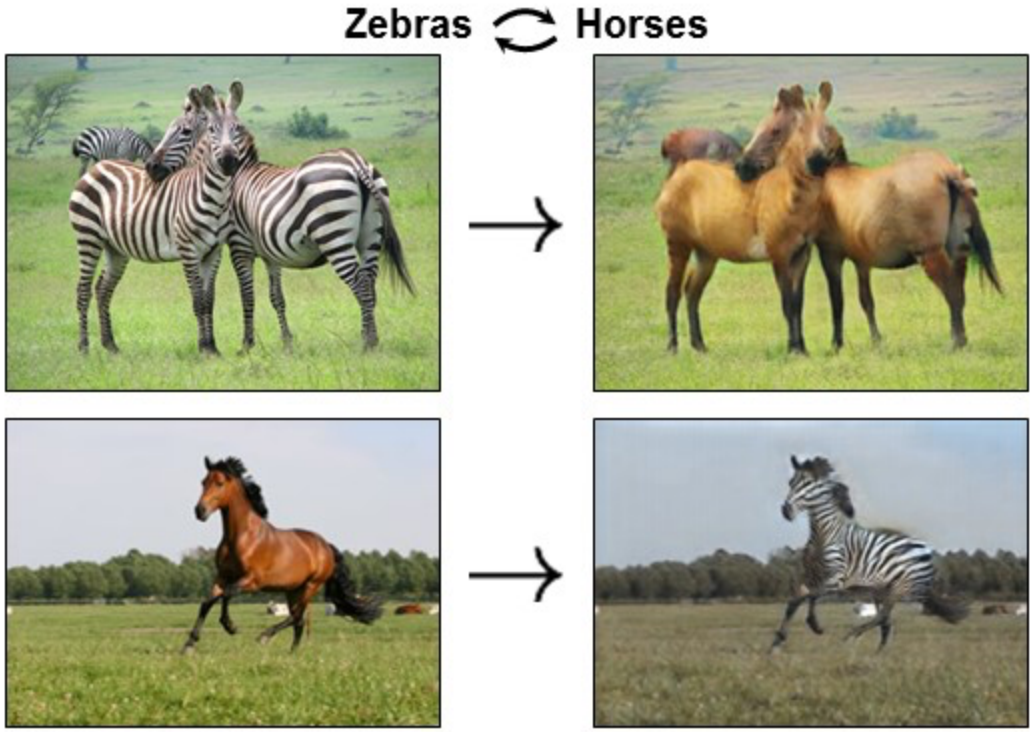
Figure 3: (a) Our model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X, F, and X. To further regularize the mappings, we introduce two "cycle consistency losses" that capture the intuition that if we translate from one domain to the other and back again we should arrive where we started: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

Network Architecture





Simplified view of CycleGAN architecture


Zebras
Horses


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

Combination Generative Adversarial Networks

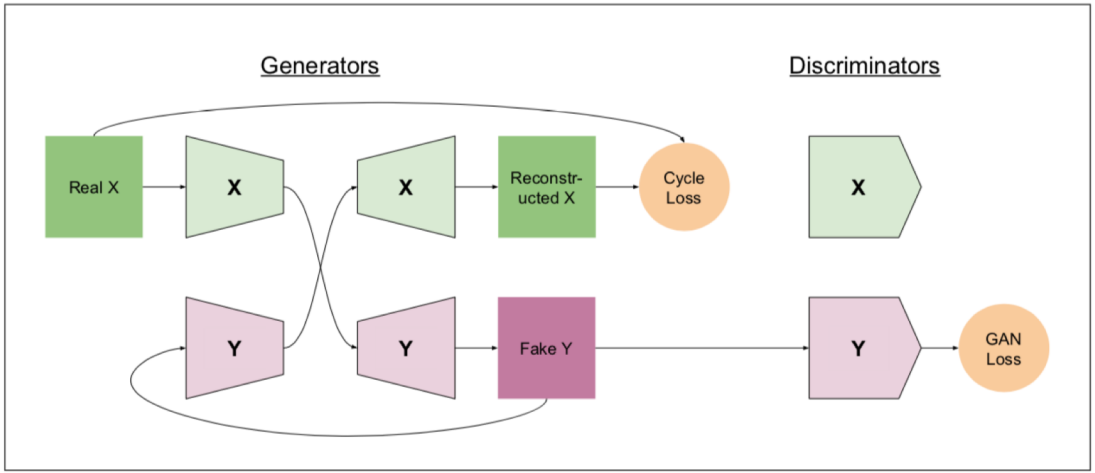


Figure 2. Generator training pass for direction $X \rightarrow Y$, where $X, Y \in \{1, .., n\} : X \neq Y$ are randomly chosen from our n domains at the start of every iteration. This pass is always repeated symmetrically for direction $Y \rightarrow X$ as well.

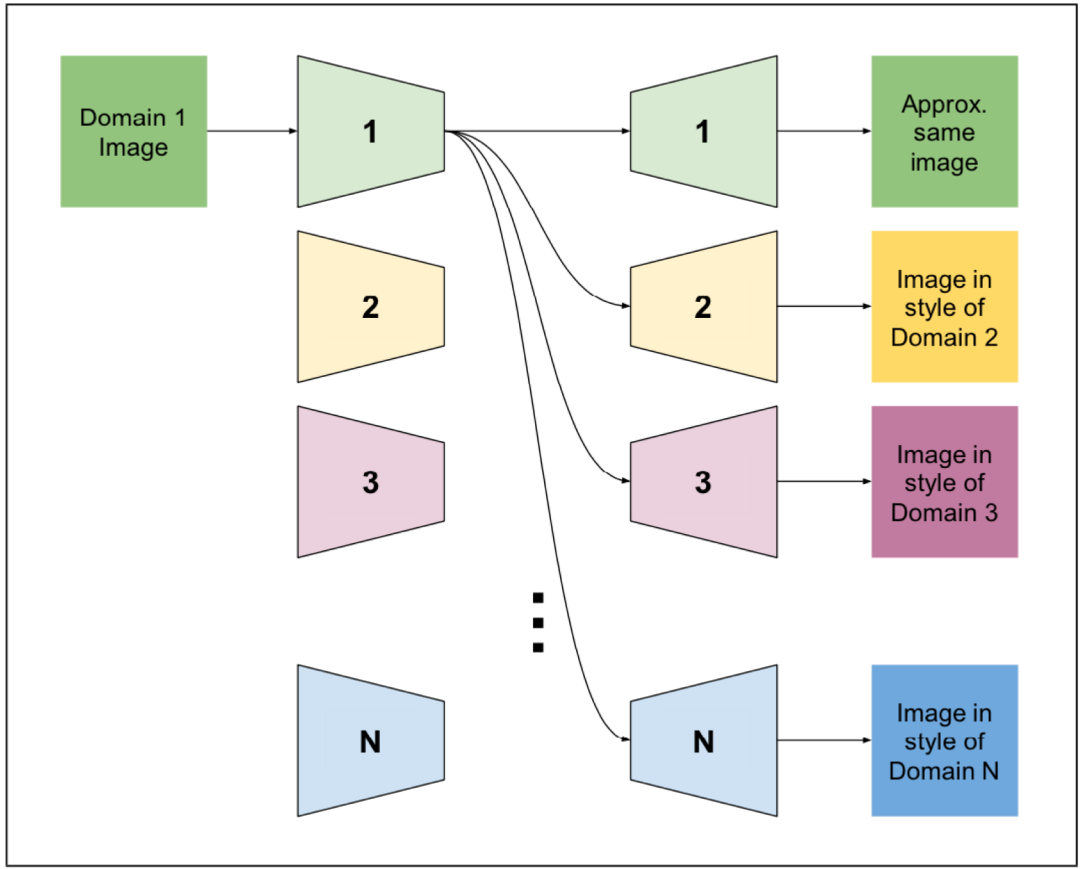


Figure 3. Example inference functionality of translation from one domain to all others.

Figure 4. Validation results for pictures of the Alps in all four seasons. Original images lie on the diagonal.

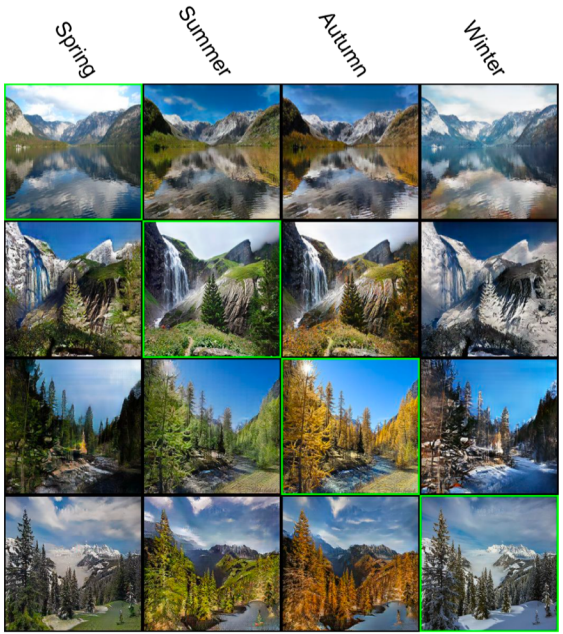
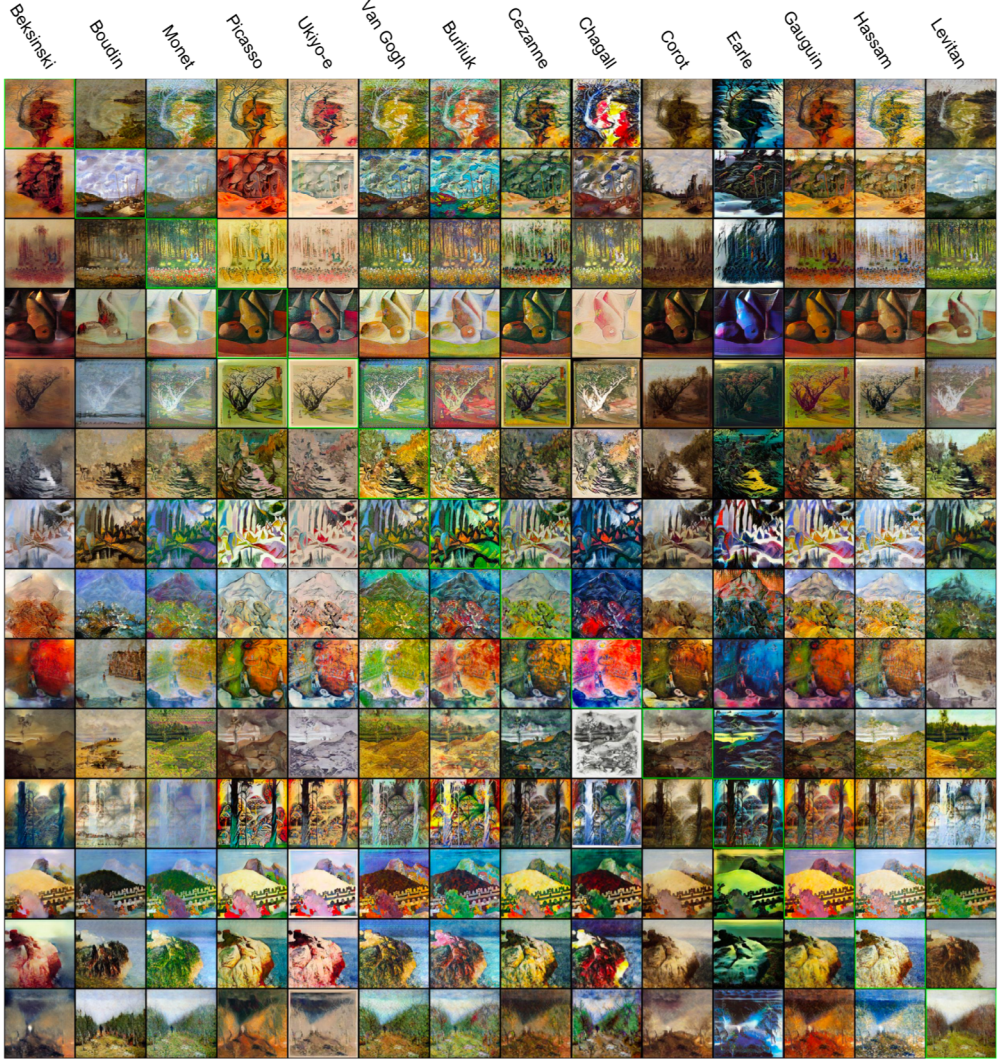
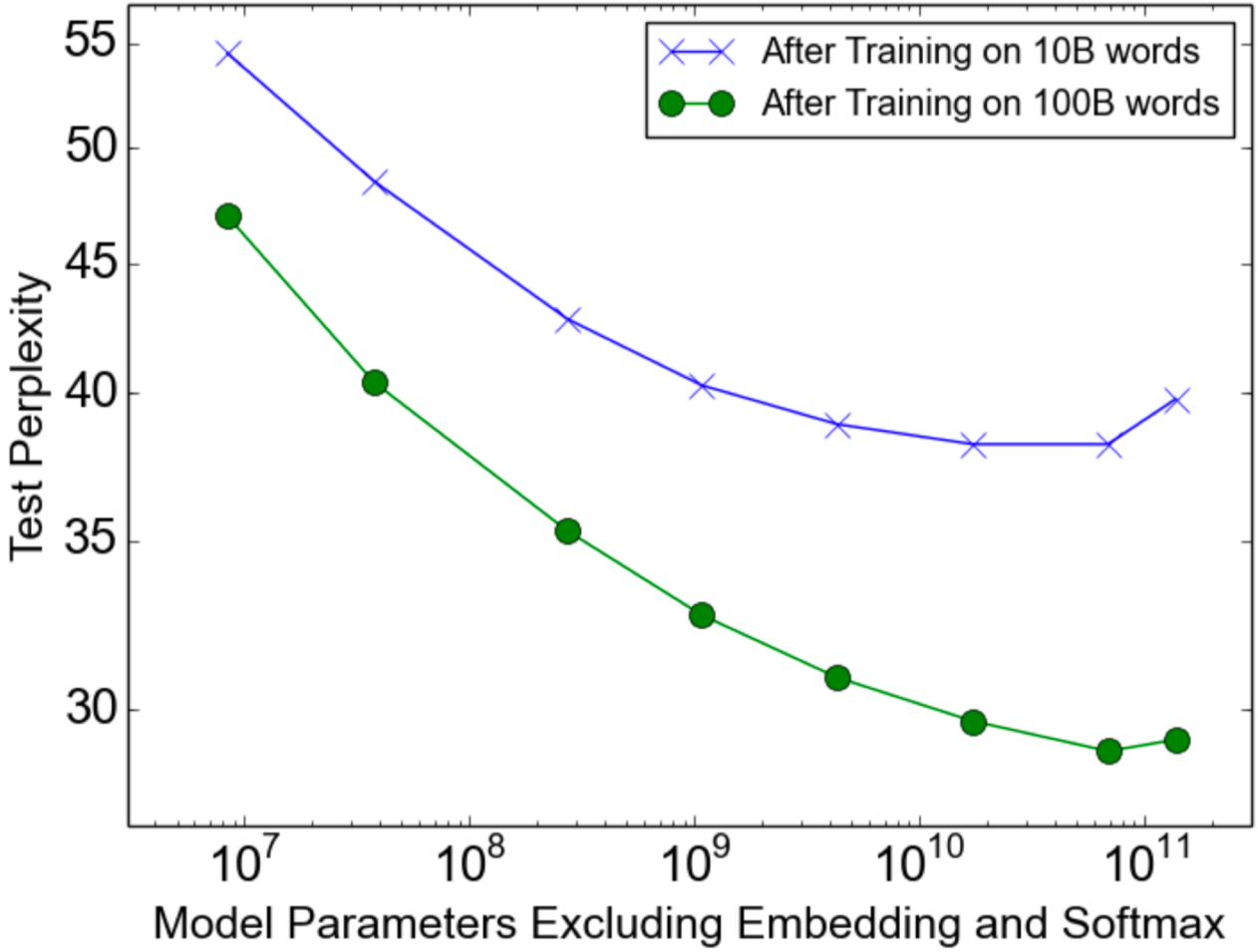


Figure 5. Same Alps images but from standard CycleGAN results instead. Original images lie on the diagonal.


Beksinski
Boudin
Monet
Picasso
Ukiyo-e
Van Gogh
Burliuk
Cezanne
Chagall
Corot
Earle
Gauguin
Hassam
Levitan

Really Large Networks
Multimodal Networks
Multitask Networks


After Training on 10B words
After Training on 100B words
Test Perplexity
55
50
45
40
35
30
10^7
10^8
10^9
10^{10}
10^{11}
Model Parameters Excluding Embedding and Softmax

1000x Model Capacity, 137 Billion Parameters

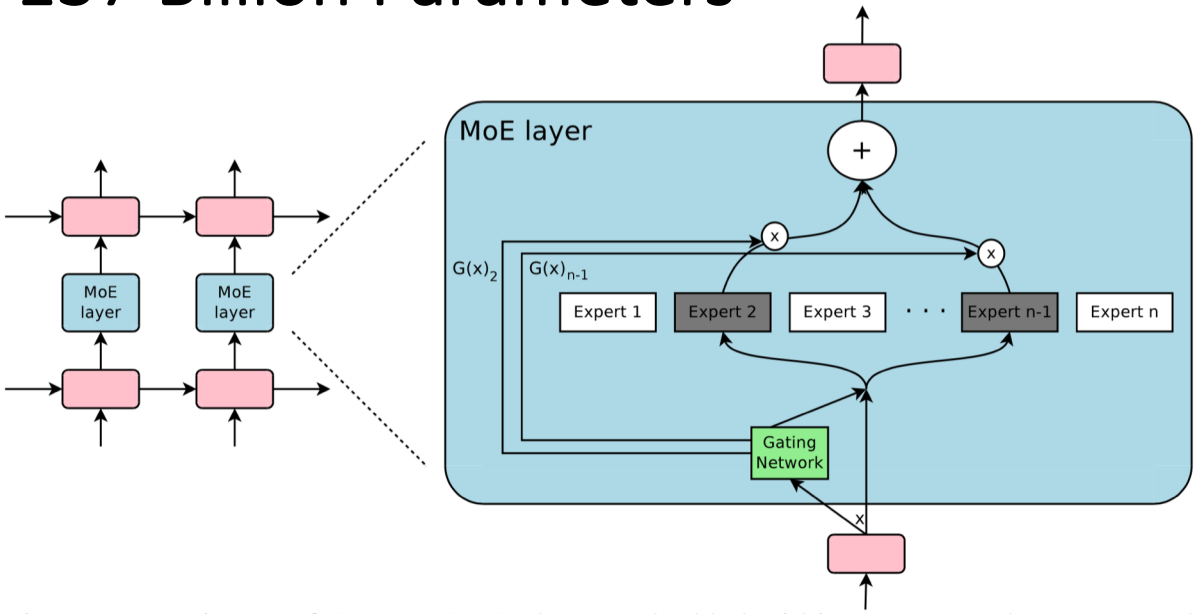


Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.

OUTRAGEOUSLY LARGE NEURAL NETWORKS:
THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

Can we create a unified deep learning model to solve tasks across multiple domains?

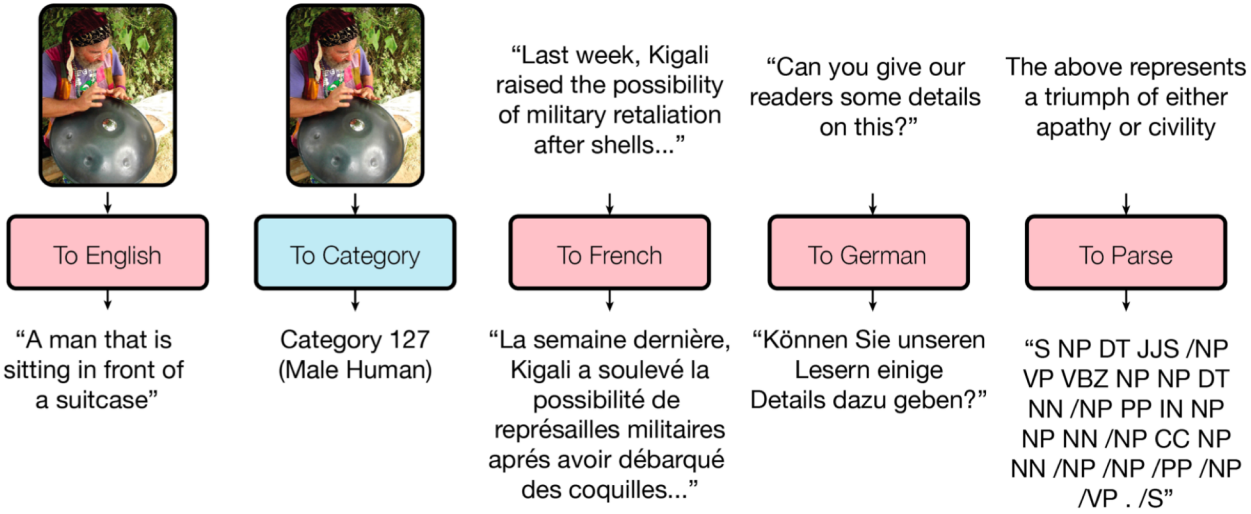


Figure 1: Examples decoded from a single MultiModel trained jointly on 8 tasks. Red depicts a language modality while blue depicts a categorical modality.

One Model To Learn Them All

Aggregating Blocks with Gates

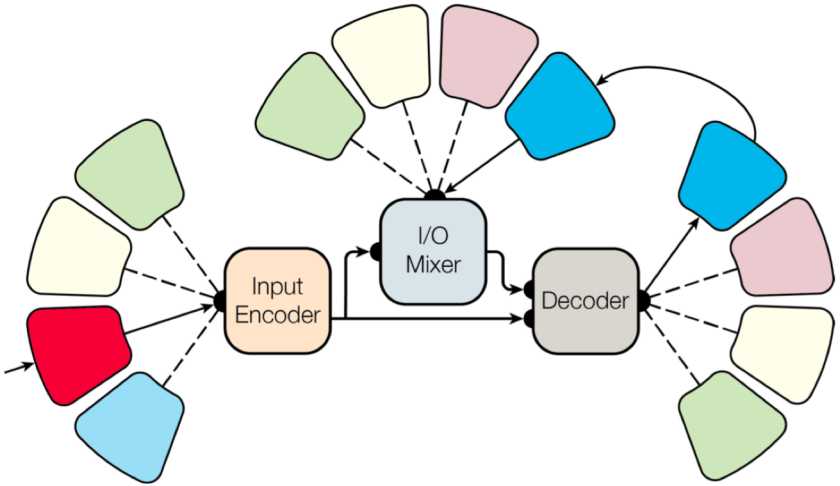


Figure 2: The MultiModel, with modality-nets, an encoder, and an autoregressive decoder.

“This leads us to conclude that mixing different computation blocks is in fact a good way to improve performance on many various tasks.”

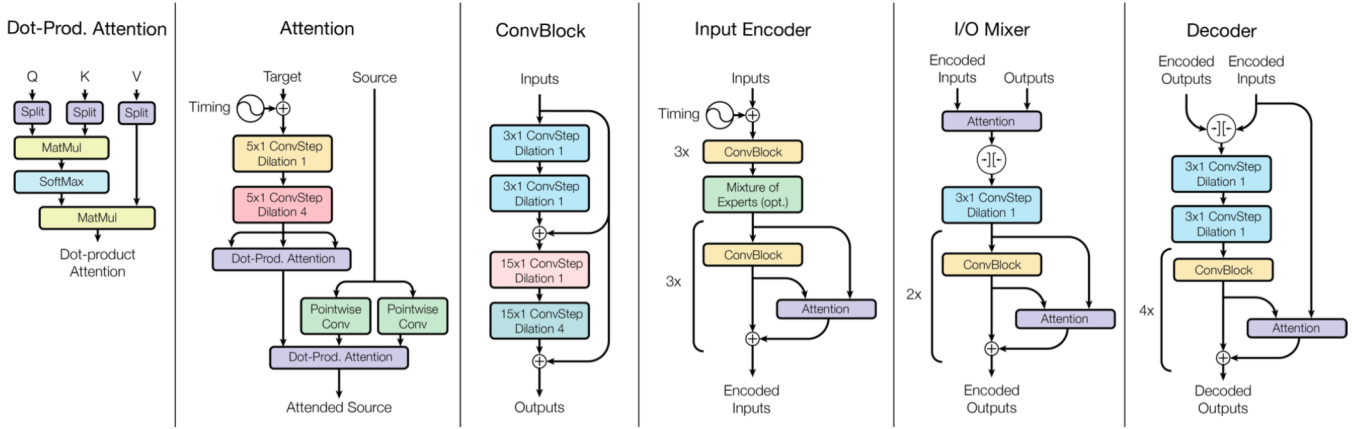


Figure 3: Architecture of the MultiModel; see text for details.

Problem	Alone			W/ ImageNet			W/ 8 Problems		
	log(ppl)	acc.	full	log(ppl)	acc.	full	log(ppl)	acc.	full
Parsing	0.20	97.1%	11.7%	0.16	97.5%	12.7%	0.15	97.9%	14.5%

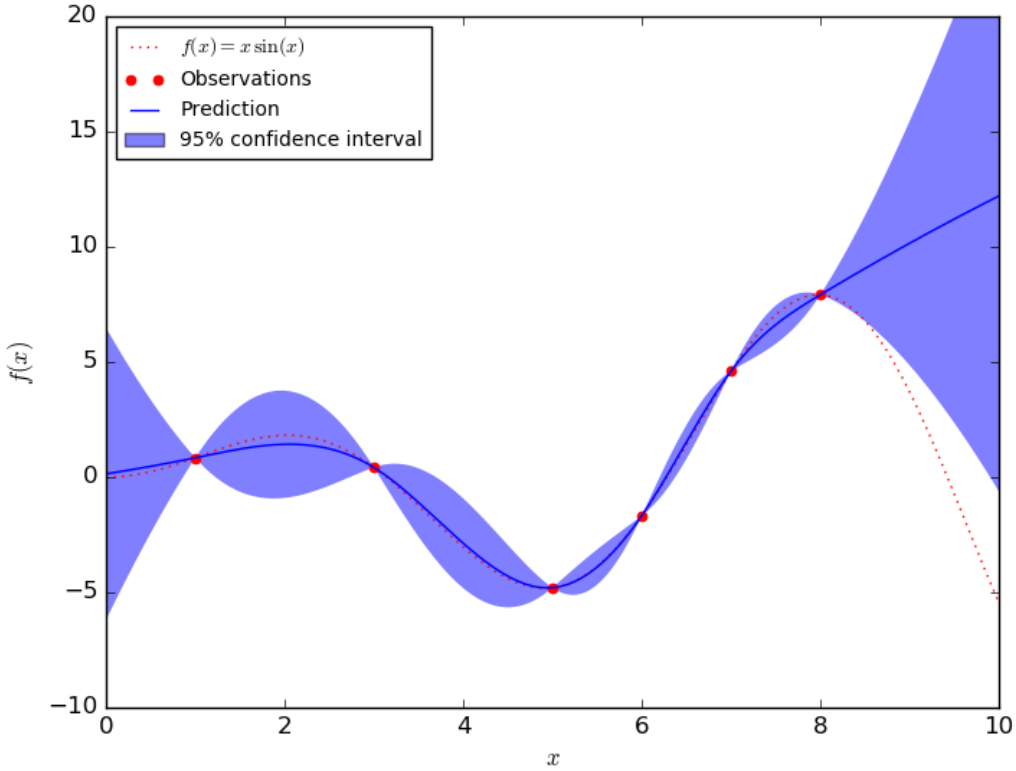
Table 3: Results on training parsing alone, with ImageNet, and with 8 other tasks. We report log-perplexity, per-token accuracy, and the percentage of fully correct parse trees.

Problem	All Blocks		Without MoE		Without Attention	
	log(perpexity)	accuracy	log(perplexity)	accuracy	log(perplexity)	accuracy
ImageNet	1.6	67%	1.6	66%	1.6	67%
WMT EN→FR	1.2	76%	1.3	74%	1.4	72%

Table 4: Ablating mixture-of-experts and attention from MultiModel training.

Deep Learning Uncertainty Quantification

Intuition behind UQ



Three Approaches to Uncertainty Quantification

- Train on distributions and predict distributions
- Bootstrap with ensembles during training
- Dropout during inference as a Bayesian approximation

  (Yarin Gal, University of Cambridge)

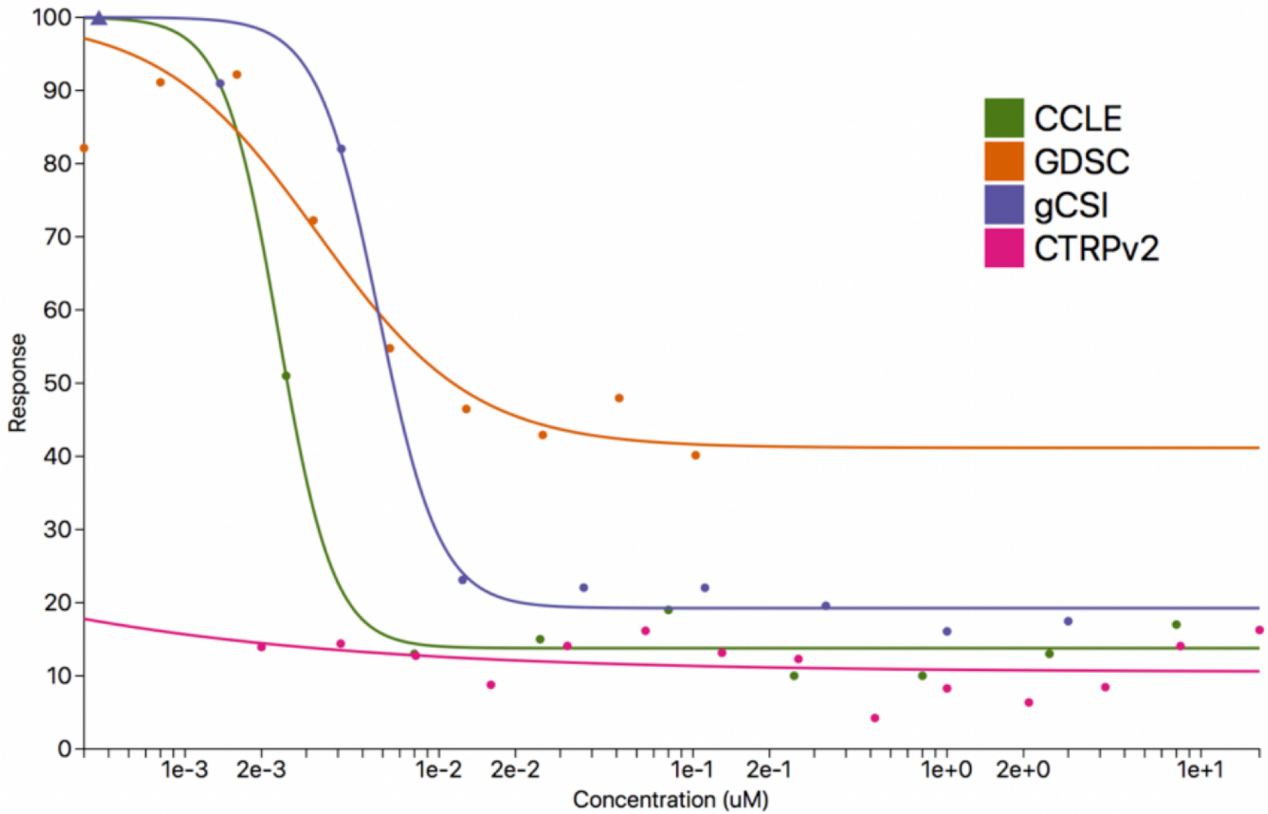
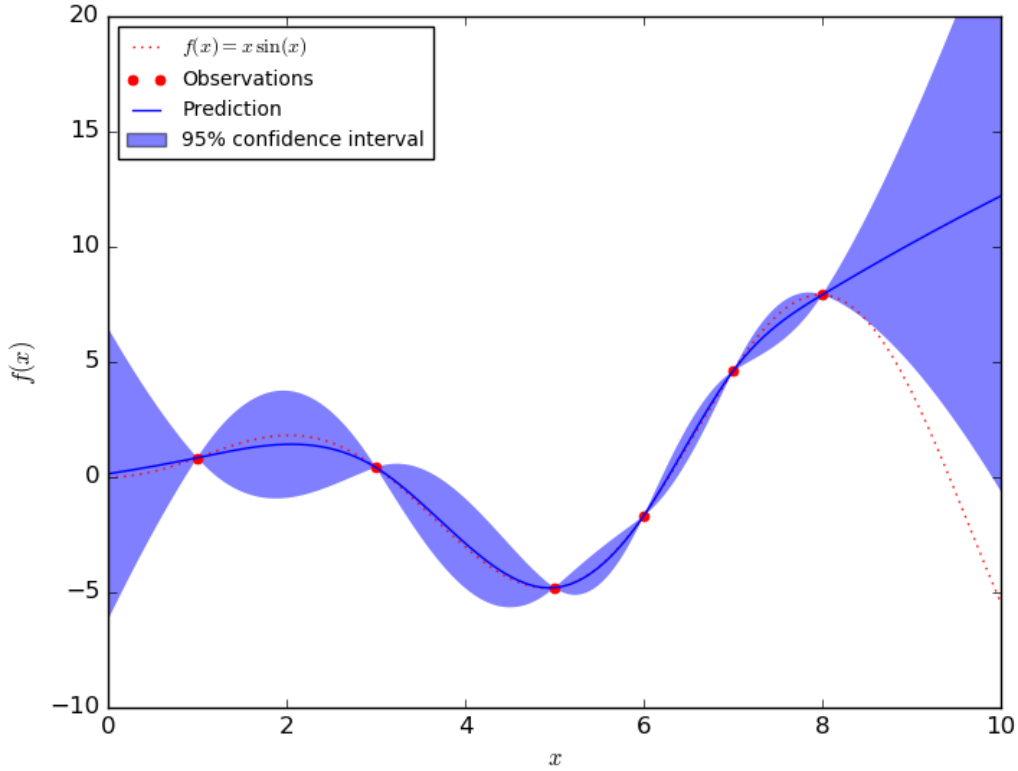
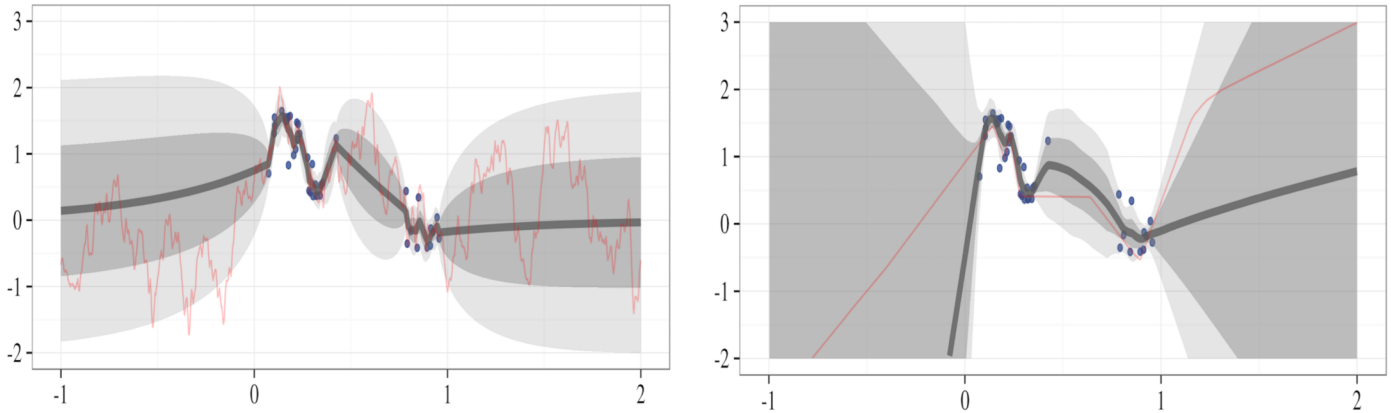


Figure 2. An example of dose response data from multiple studies. The figure adapted from PharmacoDB [2] shows the fitted dose response curves of the SU-DHL-8 lymphoma cell line treated with paclitaxel. Experimental measurements from multiple sources are not in complete agreement.

Intuition behind UQ (Gaussian Process Models)



Bootstrapping UQ in Deep Neural Networks

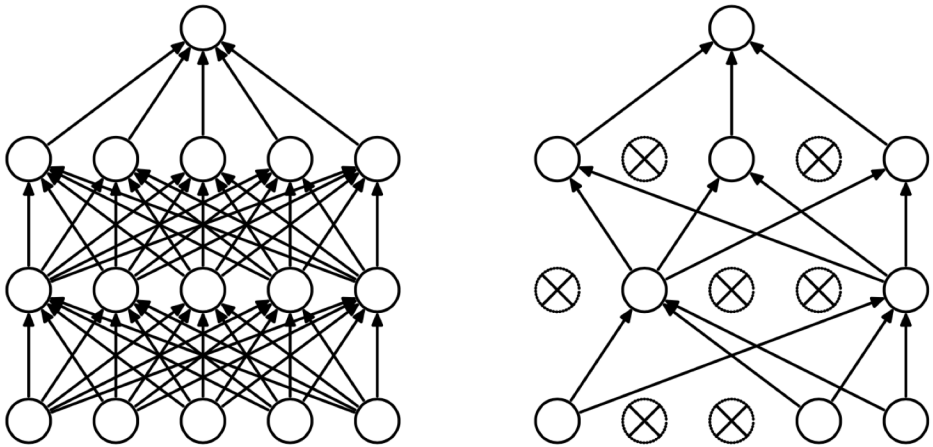


(b) Gaussian process posterior (c) Bootstrapped neural nets

Gerhard Paass, Assessing and Improving Neural Network Predictions by the Bootstrap Algorithm, NIPS, 659

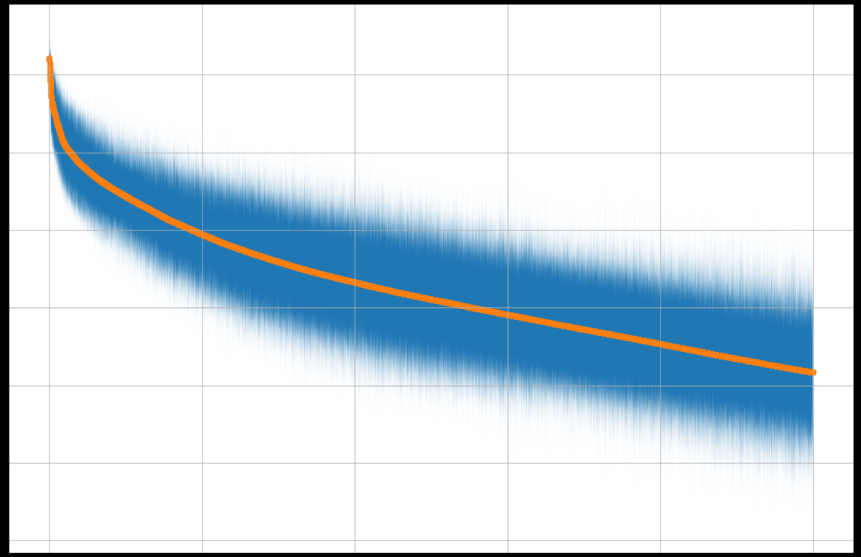
Dropout!

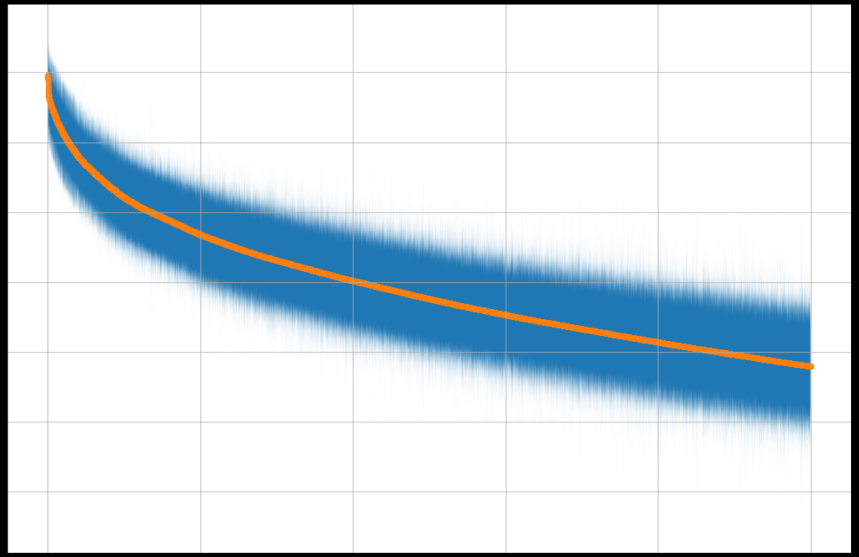
Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov



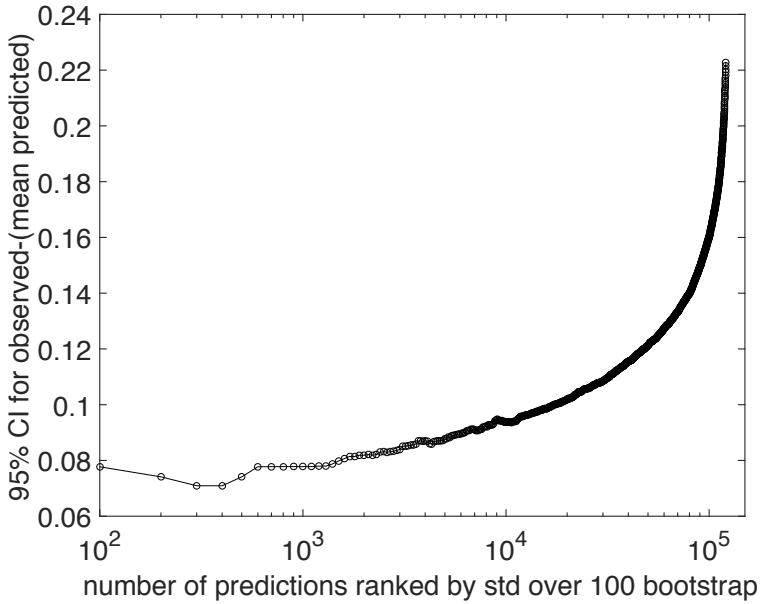
(a) Standard Neural Net (b) After applying dropout.

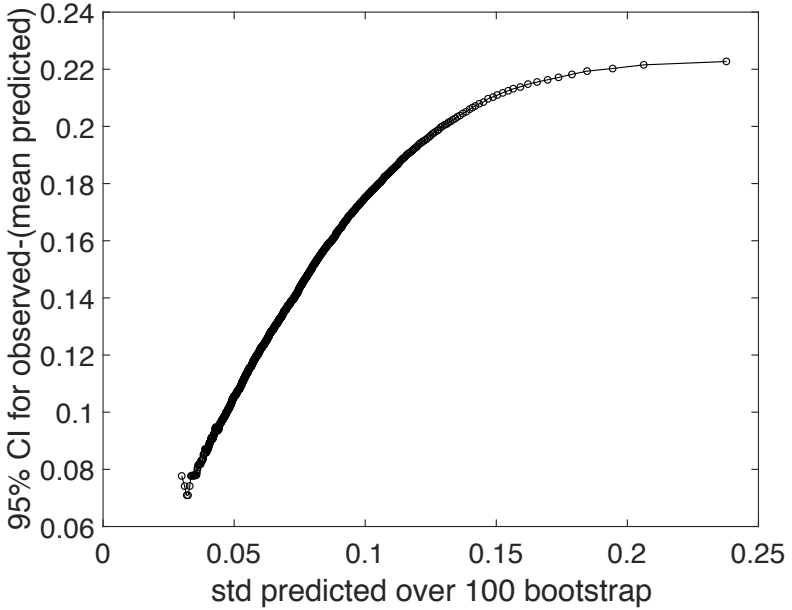
Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.



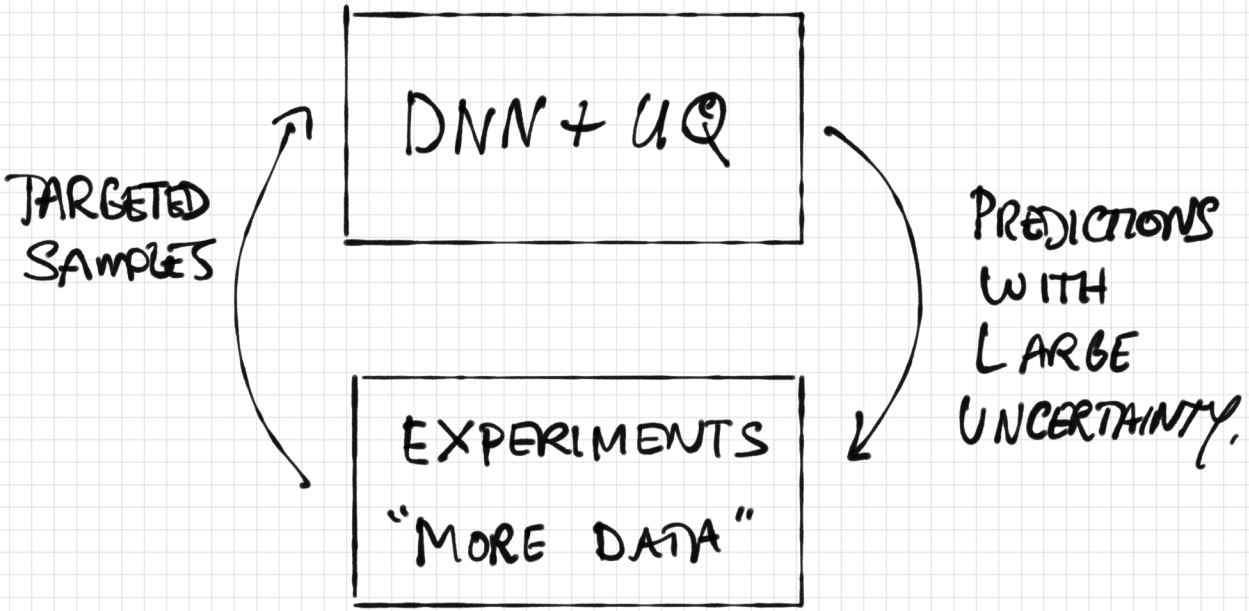


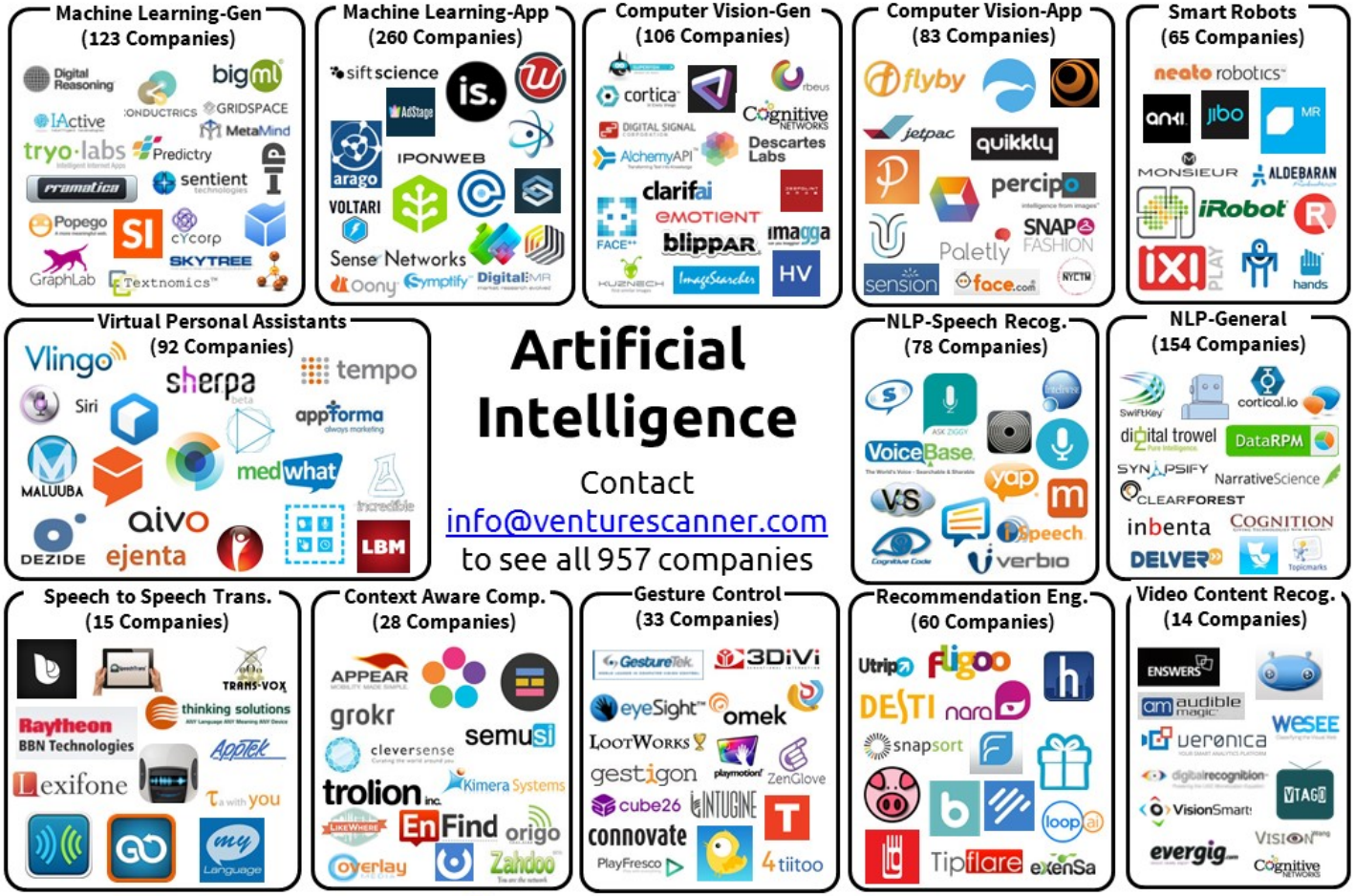
Order coherence and calibration





Highly confident predictions (small bootstrap std) have high accuracy (with high confidence the predictions are in a small interval around the true value).

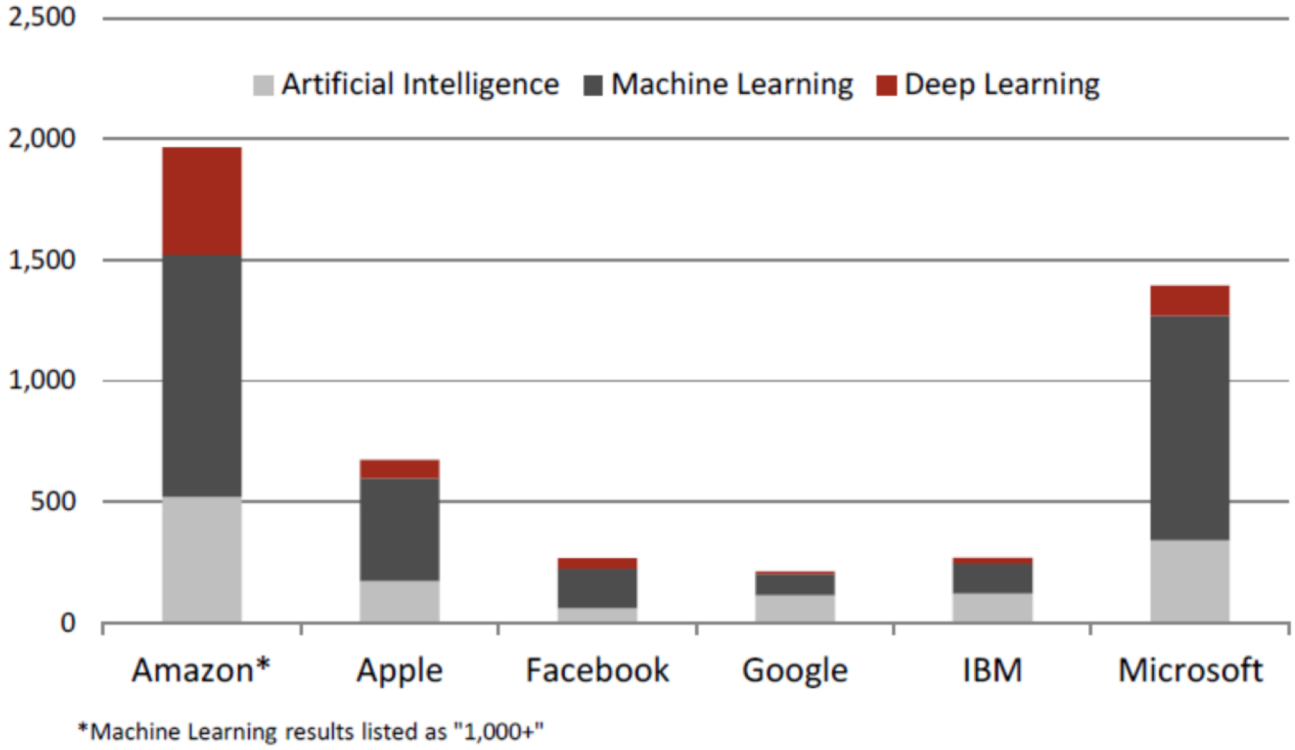

DNN + UQ
TARGETED SAMPLES
PREDICTIONS WITH LARGE UNCERTAINTY.
EXPERIMENTS
"MORE DATA"


Machine Learning-Gen
(123 Companies)
Machine Learning-App
(260 Companies)
Computer Vision-Gen
(106 Companies)
Computer Vision-App
(83 Companies)
Smart Robots
(65 Companies)
Virtual Personal Assistants
(92 Companies)
Artificial Intelligence
Contact
info@venturescanner.com
to see all 957 companies
NLP-Speech Recog.
(78 Companies)
NLP-General
(154 Companies)
Speech to Speech Trans.
(15 Companies)
Context Aware Comp.
(28 Companies)
Gesture Control
(33 Companies)
Recommendation Eng.
(60 Companies)
Video Content Recog.
(14 Companies)
Venture Scanner

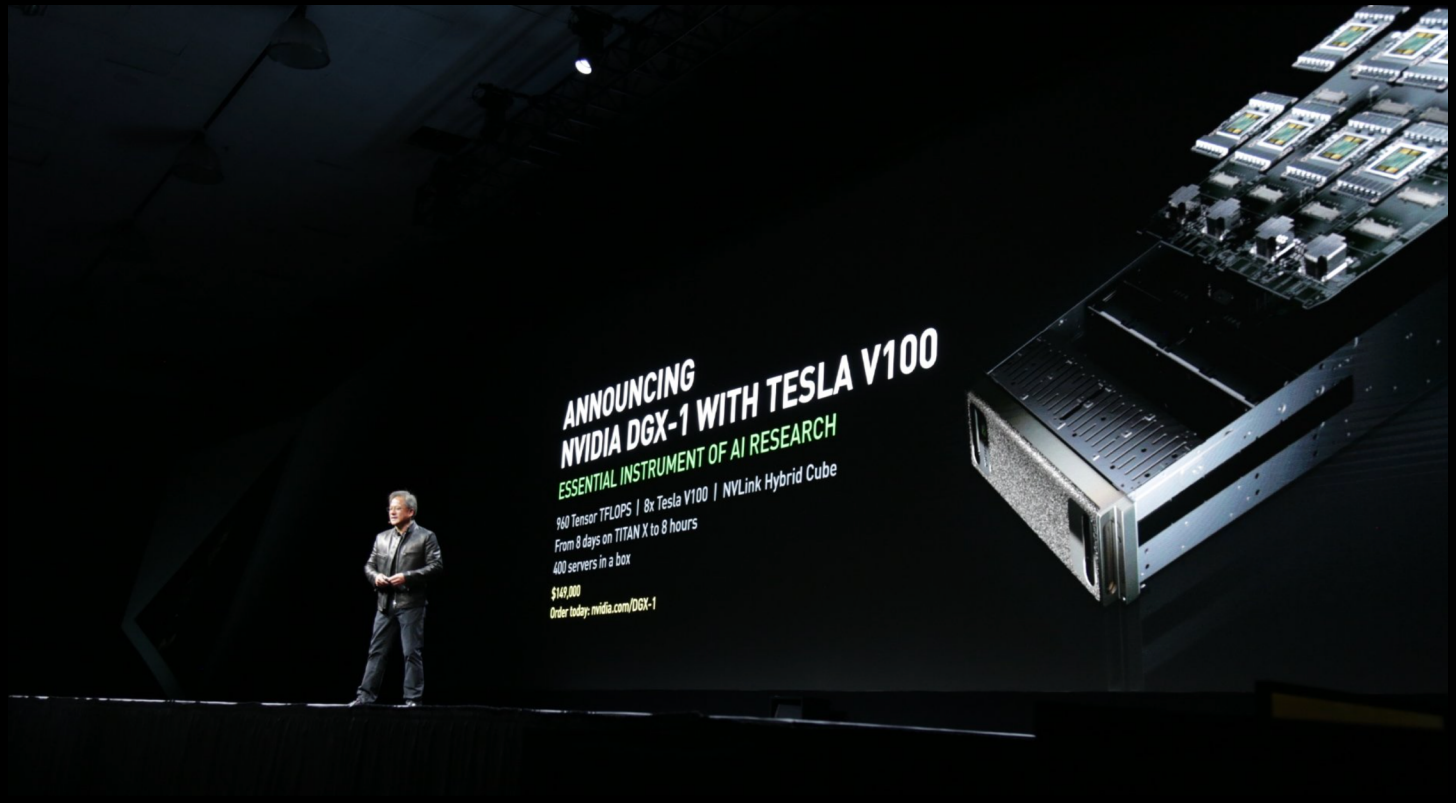
By 2020, the market for machine learning will reach $40 billion, according to market research firm IDC.

Deep Learning market is projected to be ~$5B by 2020

Exhibit 23: Monster.com Postings by Company, Search Terms: Artificial Intelligence, Machine Learning, and Deep Learning



Source: monster.com


ANNOUNCING
NVIDIA DGX-1 WITH TESLA V100
ESSENTIAL INSTRUMENT OF AI RESEARCH
960 Tensor TFLOPS | 8x Tesla V100 | NVLink Hybrid Cube
From 8 days on TITAN X to 8 hours
400 servers in a box
$149,000
Order today: nvidia.com/DGX-1


DGX STATION
TESLA
TESLA
TESLA
TESLA
NVIDIA


intel
AI AND MORE ON IA
Dr. Rajeeb Hazra
Vice President, Data Center Group
General Manager, Enterprise and Government Group
ISC High Performance
ISC High Performance

According to the latest market research report "**Artificial Intelligence Market by Offering (Hardware, Software, Services), Technology (Machine Learning, Natural Language Processing, Context-Aware Computing, Computer Vision), End-User Industry, and Geography - Global Forecast to 2025**", The artificial intelligence market is expected to grow from USD 21.46 Billion in 2018 to USD 190.61 Billion by 2025, at a CAGR of 36.62% between 2018 and 2025. Major drivers for the market are growing big data, the increasing adoption of cloud-based applications and services, and increasing demand for intelligent virtual assistants. The major restraint for the market is the limited number of AI technology experts.


CBINSIGHTS
ai.100
2018
CYBERSECURITY
ENDGAME.
DARKTRACE
cybereason
CROWDSTRIKE
OBSIDIAN
sparkcognition
Shift Technology
appthority
VERSIVE
SHAPE
perimeterx
ROBOTICS
vicarious
KINDRED
anki
UBTECH
brain corp
neurala
CloudMinds
FINTECH & INSURANCE
zestfinance
NUMERAI
Upstart
affirm
CAPE
HARDWARE
GRAPHCORE
MYTHIC
cerebras
Cambricon
EDUCATION
英语流利说 Liulishuo
Osmo
HR TECH
mya
textio
workey
ENTERPRISE
DataRobot
TRIFACTA
algolia
tamr
LEAPMIND
PETUUM
ELEMENT AI
HEALTHCARE
freenome
babylon
RECURSION
INSILICO MEDICINE
TEMPUS
ARTERYS
AiCure
flatiron
AUTO TECH
Mighty AI
drive.ai
ZOOX
nauto
AEYE
COMMERCE
DYNAMIC YIELD
Reflektion
Twiggle
aquifi
Appier
AGRICULTURE
BENSON-HILL BIOSYSTEMS
prospera
PHYSICAL SECURITY
SHIELD AI
Deep Sentinel
RISK & REGULATORY COMPLIANCE
WorkFusion
SOCURE
onfido
text IQ
Merlon Intelligence
MARKETING, SALES, CRM
Amplero
INVOCA
Gong
INSIDE SALES.COM
conversica
IOT
C3 IoT
MAANA
Preferred Networks
FOGHORN
ORCAM
NEWS & MEDIA
SoundHound Inc.
今日头条 BYTEDANCE
GEOSPATIAL ANALYTICS
Orbital Insight
Descartes Labs
LIFE SCIENCE
zymergen
LEGAL TECH
casetext
IT & NETWORKS
Moogsoft
SPORTS
SPORTLOGIQ
ESPORTS
MOBALYTICS
PERSONAL ASSISTANTS
sherpa
TRAVEL
FLYR
SOFTWARE DEVELOPMENT
applitools eyes
CROSS-INDUSTRY
CognitiveScale
Captricity
Face++ 旷视
TRACTABLE
商汤 sensetime
INSIGHT ENGINES
narrative science
Mobvoi
PRIMER
KYNDI

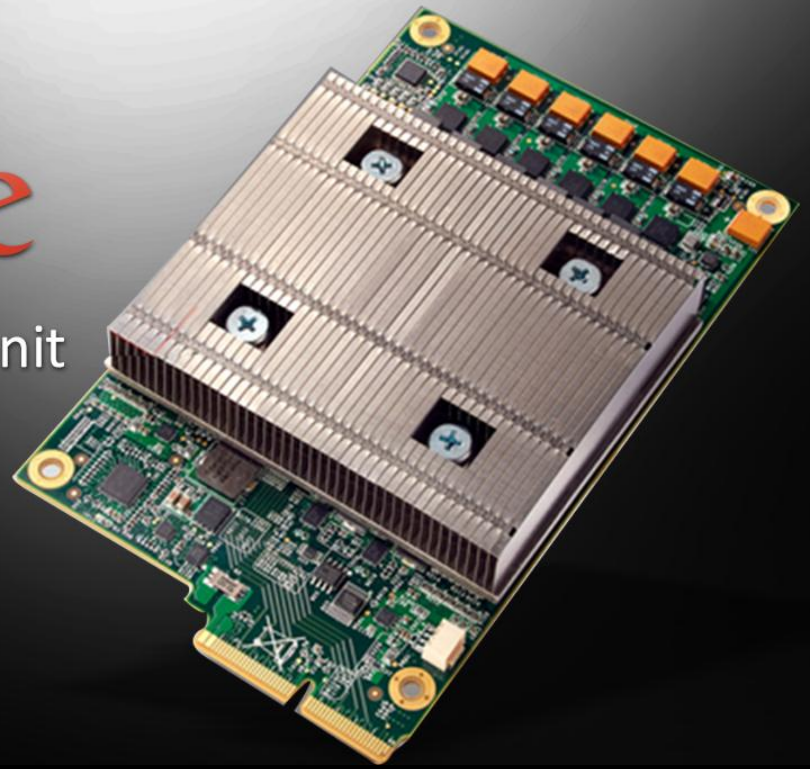
AI Hardware and Systems Startups

- Cerebras
- Wave Computing
- KnuEdge
- TensTorrent
- THINCI
- Knowm
- Mythic
- BrainChip
- InnoGrit
- LightMatter
- SambaNova
- Esperanto
- Almotive
- Deepscale
- LeapMind
- NovuMind
- REM
- Deep Vision
- Groq
- Kneron
- Hailo
- Think Silicon
- LightIntelligence
- Gyrfalcon

Specialized hardware is emerging that will be many times (100x) the performance of general purpose CPU and GPU designs

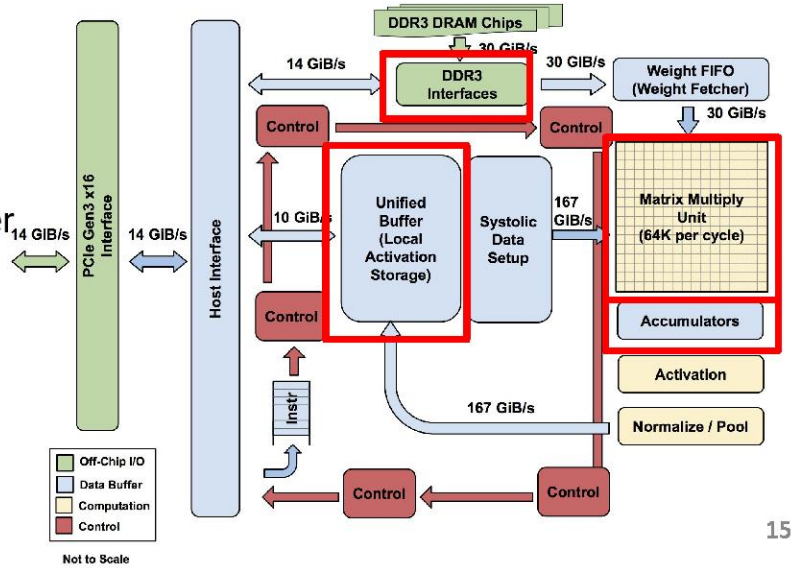
Google

Tensor Processing Unit



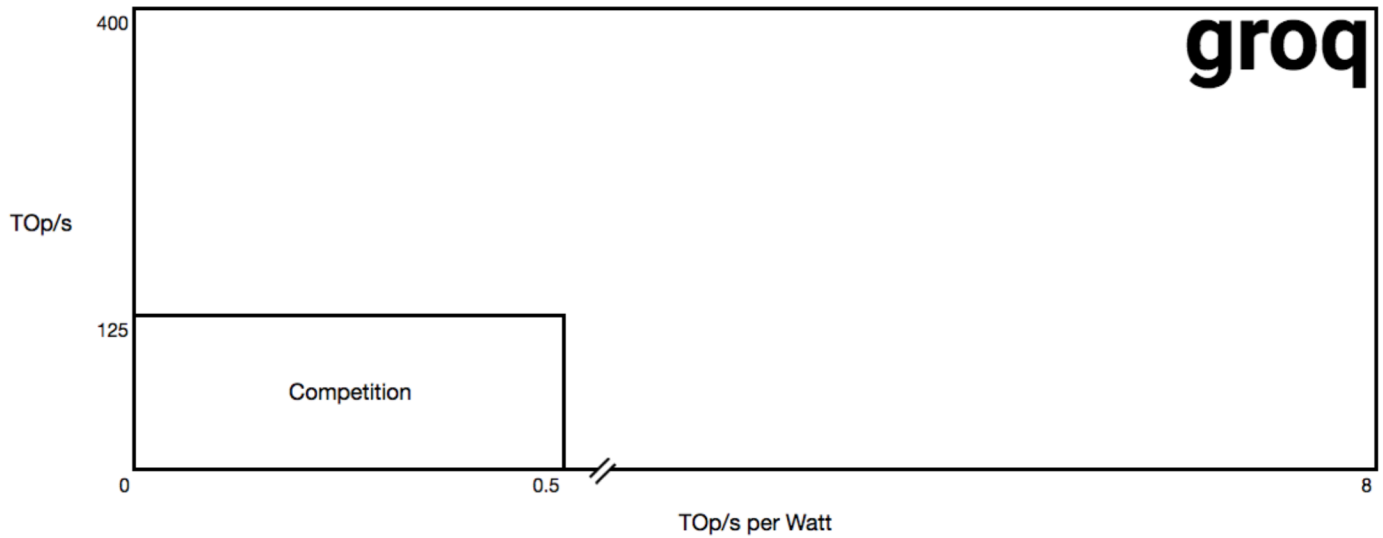
TPU: High-level Chip Architecture

- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units
- 700 MHz clock rate
- Peak: 92T operations/second
  - 65,536 * 2 * 700M
- >25X as many MACs vs GPU
- >100X as many MACs vs CPU
- 4 MiB of on-chip Accumulator memory
- 24 MiB of on-chip Unified Buffer (activation memory)
- 3.5X as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GiB of off-chip weight DRAM memory



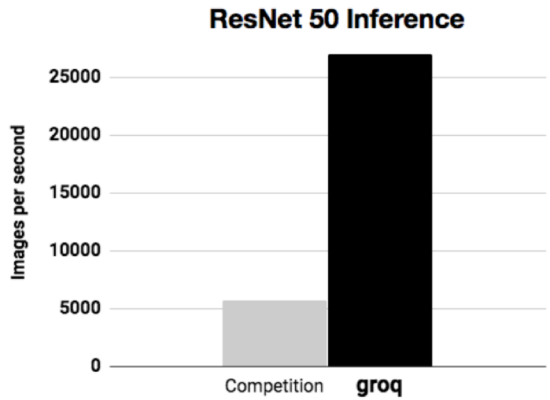

A
B
C
D

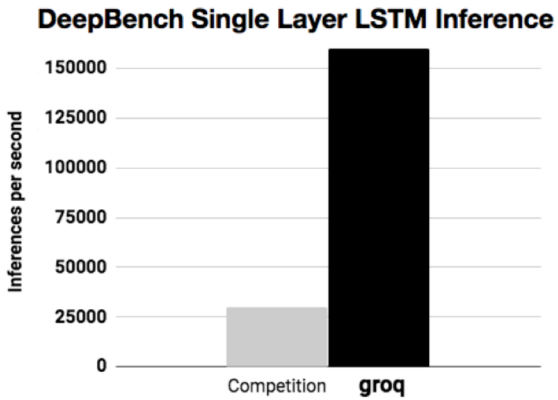
Groq (grok)



Our first machine learning product. Single chip. 2018.

November 9, 2017





2018 performance estimates. Single chip. <1ms latency.

November 21, 2017

Wave Computing

Wave's Compute Appliance is Redefining How Machine Learning is Done

- 2.9 PetaOps per second of performance
- More than 2TB of high-speed memory
- Up to 256,000 processing elements per appliance
- Scales up to four appliances per data center node
- Initially supporting TensorFlow

ABOUT THE WAVE COMPUTE APPLIANCE

Specifications for each Wave Compute Appliance

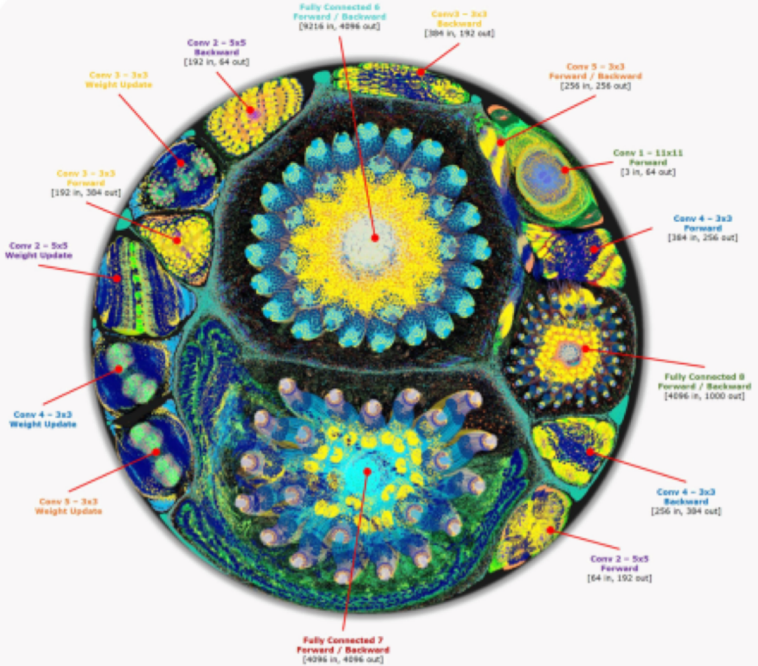
Performance	Performance/computer (peak)	2.9 PetaOPS/second
	Performance/node (peak)	11.6 PetaOPS/second
	Dataflow Processing Elements (PE's)	Up to 256,000 (16,000 PE's per Wave DPU chip)
Scalability	Wave machine learning computers per data center node	Up to 4 computers delivering 1,000,000 PE's
Memory	High-speed memory	128 GB HMC DRAM
	SSD storage	16 TB
	Bulk storage	2 TB DDR4 DRAM
Connections	Data center backbone connection	10 GbE or 40 GbE
	High-speed inter-computer communication within a single data center node	Wave's proprietary communication system that connects up to 4 computers within a single data center node
Physical	Data center form factor	Each Wave computer comes in a 3U form factor; up to 4 computers can be added per data center node
	Dimensions per each 3U computer	866D x 444W x 131H (mm)
	Operating temperature	10° – 35° C
Software	Machine learning framework	TensorFlow (initially)
	Operating system for Wave Session Manager server	Linux Server
	Library	WaveFlow Agent Library
	Development toolkit	WaveFlow SDK
	Data runtime	WaveFlow Execution Engine

Graphcore.ai

KNOWLEDGE MODELS
ARE NATURALLY REPRESENTED
AS GRAPHS...

VERTICES ARE FEATURES

EDGES ARE CORRELATIONS OR
CAUSATIONS



POPLAR™ SEAMLESS DEVELOPMENT