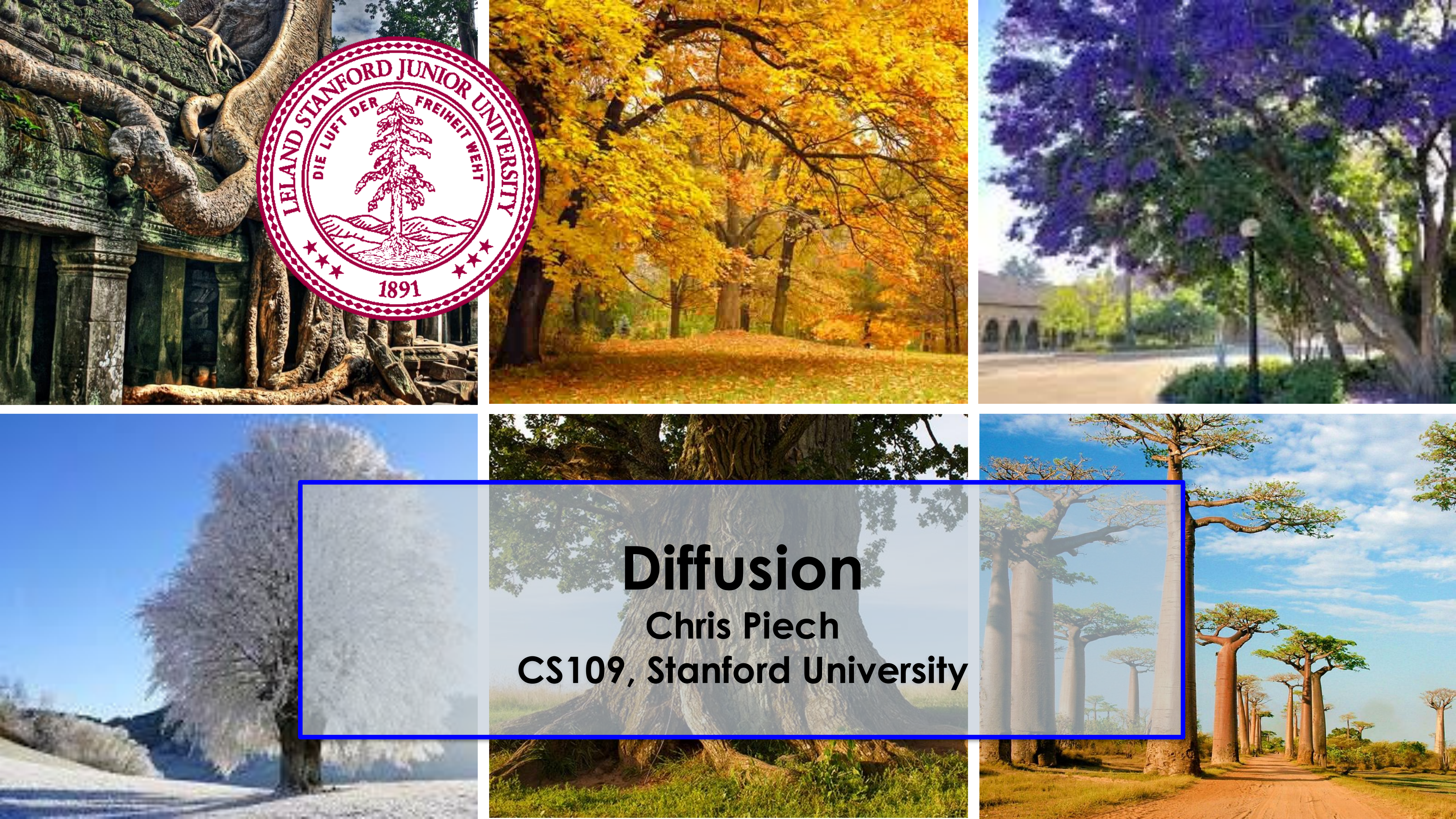


Diffusion

Chris Piech
CS109, Stanford University

First: Learning Goals…

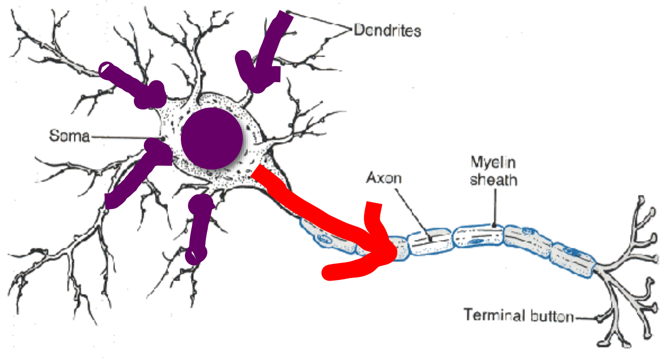
1. See **beautiful** math

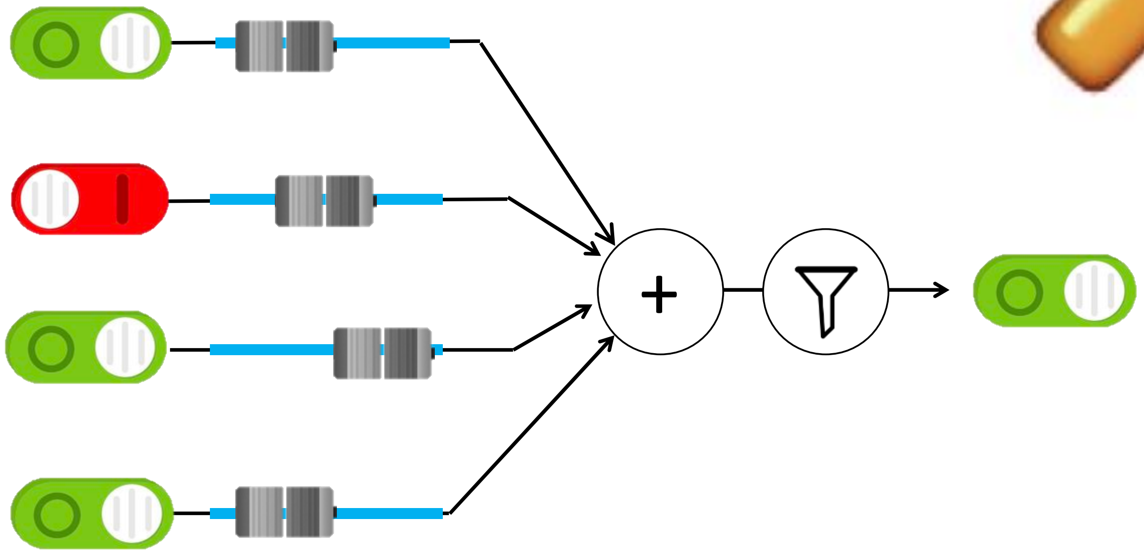
2. See **impactful** math

3. Review concepts

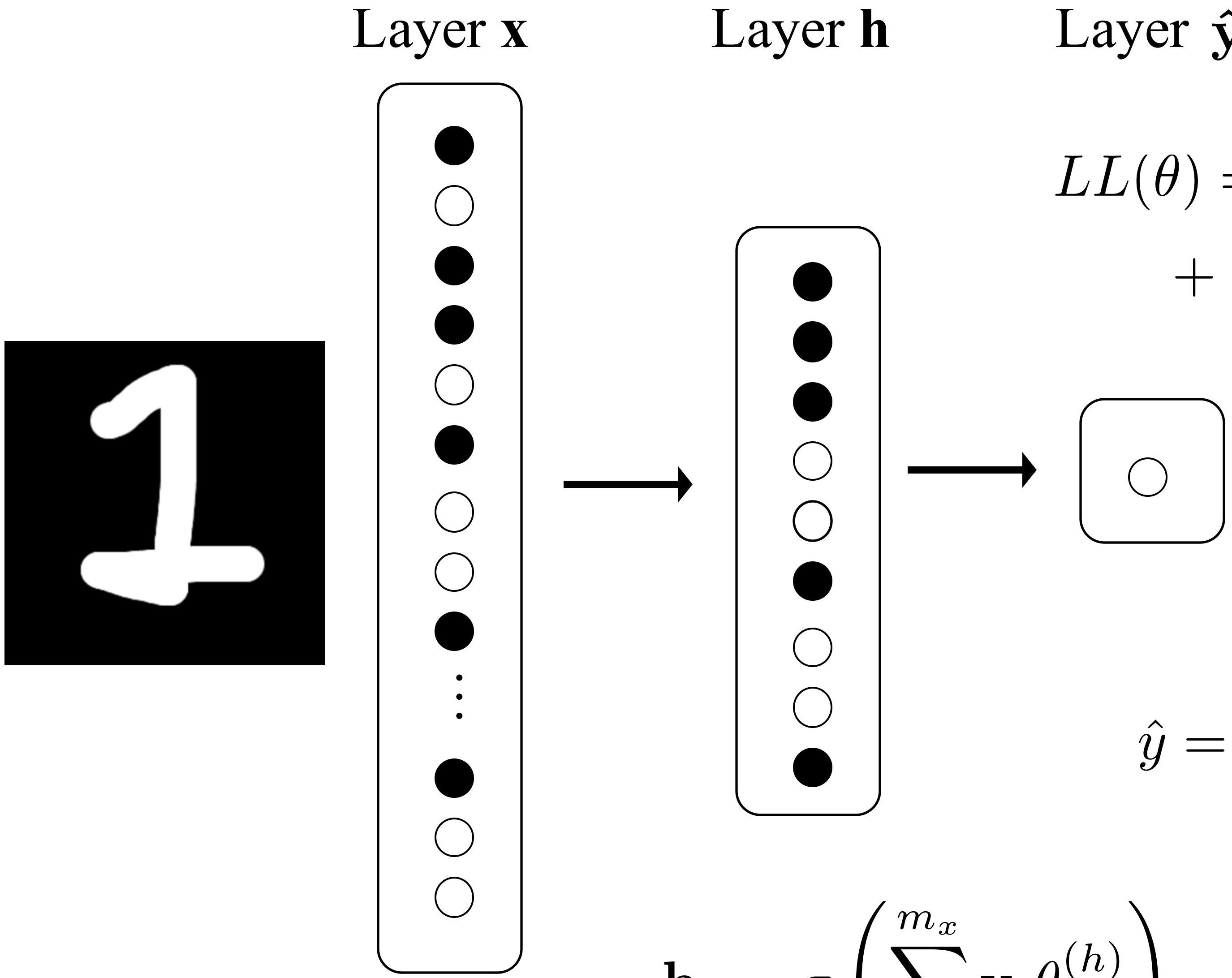
Review

Artificial Neurons





Deep Learning

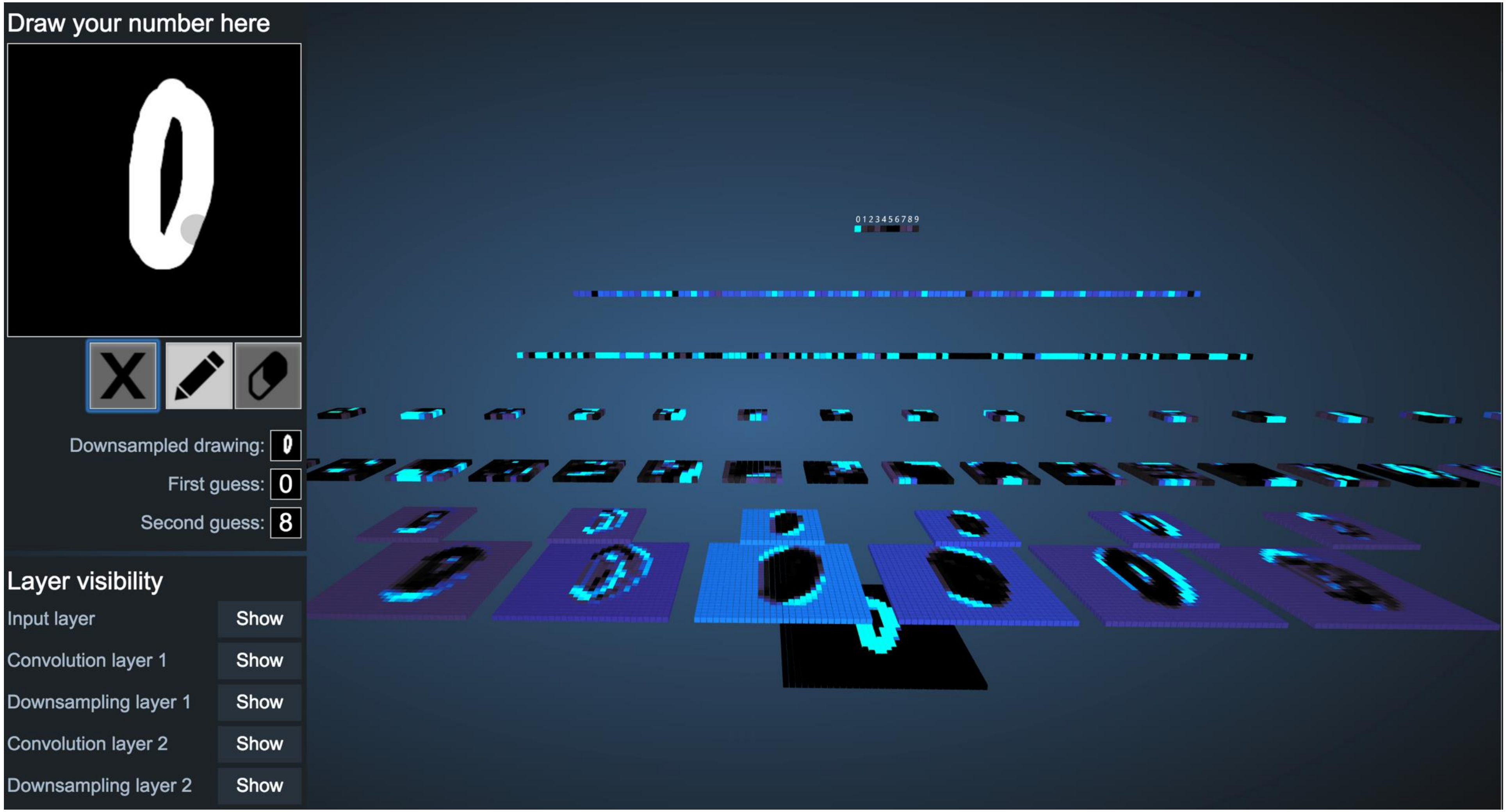


$$LL(\theta) = y \log \hat{y} + (1-y)\log[1-\hat{y}]$$

$$\hat{y} = \sigma\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right)$$

$$\mathbf{h}_j = \sigma\left(\sum_{i=0}^{m_x} \mathbf{x}_i \theta_{i,j}^{(h)}\right)$$

Demonstration



https://web.archive.org/web/20211117115916/https://www.cs.ryerson.ca/~aharley/vis/conv/

Deep Learning Code

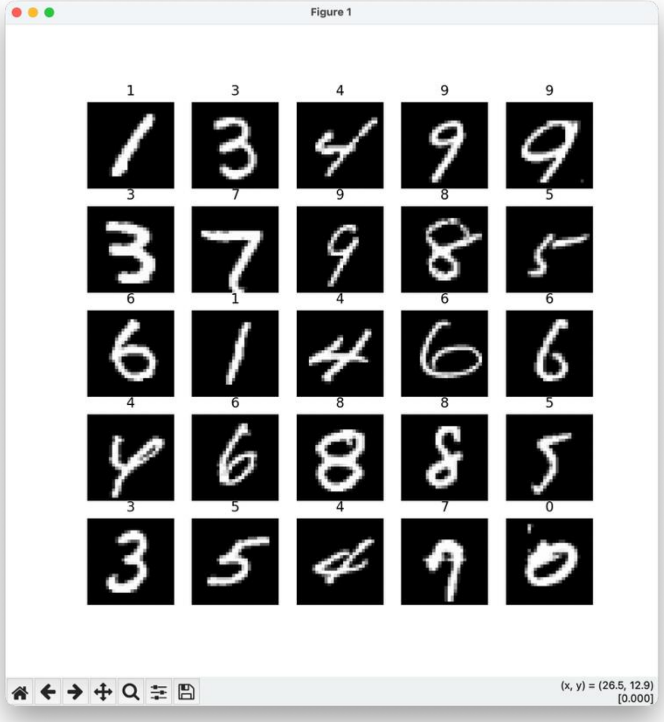
Build model

Train model

```python
def run_train(train, model):
    model.train()
    loss_function = nn.NLLLoss(reduction="sum")
    optimizer = torch.optim.Adam(model.parameters(), lr=0.0001)

    log_likelihoods = []

    for image_batch, truth_batch in train:
        # does one batch of images simultaneously
        optimizer.zero_grad()  # start with gradient zero
        pred = model(image_batch)  # predict the label
        loss = loss_function(pred, truth_batch) # calc loss
        loss.backward() # calculate gradients
        optimizer.step() # update parameters
        log_likelihoods.append(-loss.item()) # keep track of likelihoods

    return np.mean(log_likelihoods)
```

```python
def main():
    # get the data
    train, test = download_data()
    print(f"Training examples: {len(train.dataset)}")
    print(f"Test examples: {len(test.dataset)}")

    # a very simple and fast nn
    model = nn.Sequential(
        nn.Flatten(),
        nn.Linear(28 * 28, 512),  # images are 28 x 28 pixels
        nn.Sigmoid(),
        nn.Linear(512, 512),
        nn.Sigmoid(),
        nn.Linear(512, 10),
        nn.LogSoftmax(dim=1),
    ).to(device)

    # see how bad it is before training
    accuracy = run_test(test, model, device)
    print(f"Untrained, Test Accuracy: {accuracy}")
```

Training our Classifier was fun!




```
piech@Chriss-MacBook-Pro-4 DeepLearning % python train.py
Using device: mps
Training examples: 60000
Test examples: 10000
Untrained, Test Accuracy: 8.91
training...
Epoch 0, Test Accuracy: 85.48, LogLikelihood: -1.302721969763438
Epoch 1, Test Accuracy: 90.28, LogLikelihood: -0.4389506075223287
Epoch 2, Test Accuracy: 91.36, LogLikelihood: -0.3278572145620982
Epoch 3, Test Accuracy: 92.08, LogLikelihood: -0.2859212871313095
Epoch 4, Test Accuracy: 92.5, LogLikelihood: -0.2597165879646937
Epoch 5, Test Accuracy: 93.1, LogLikelihood: -0.23922272586425145
Epoch 6, Test Accuracy: 93.61, LogLikelihood: -0.22143801993131637
```

Types of Machine Learning Tasks

Muli-Class Classification

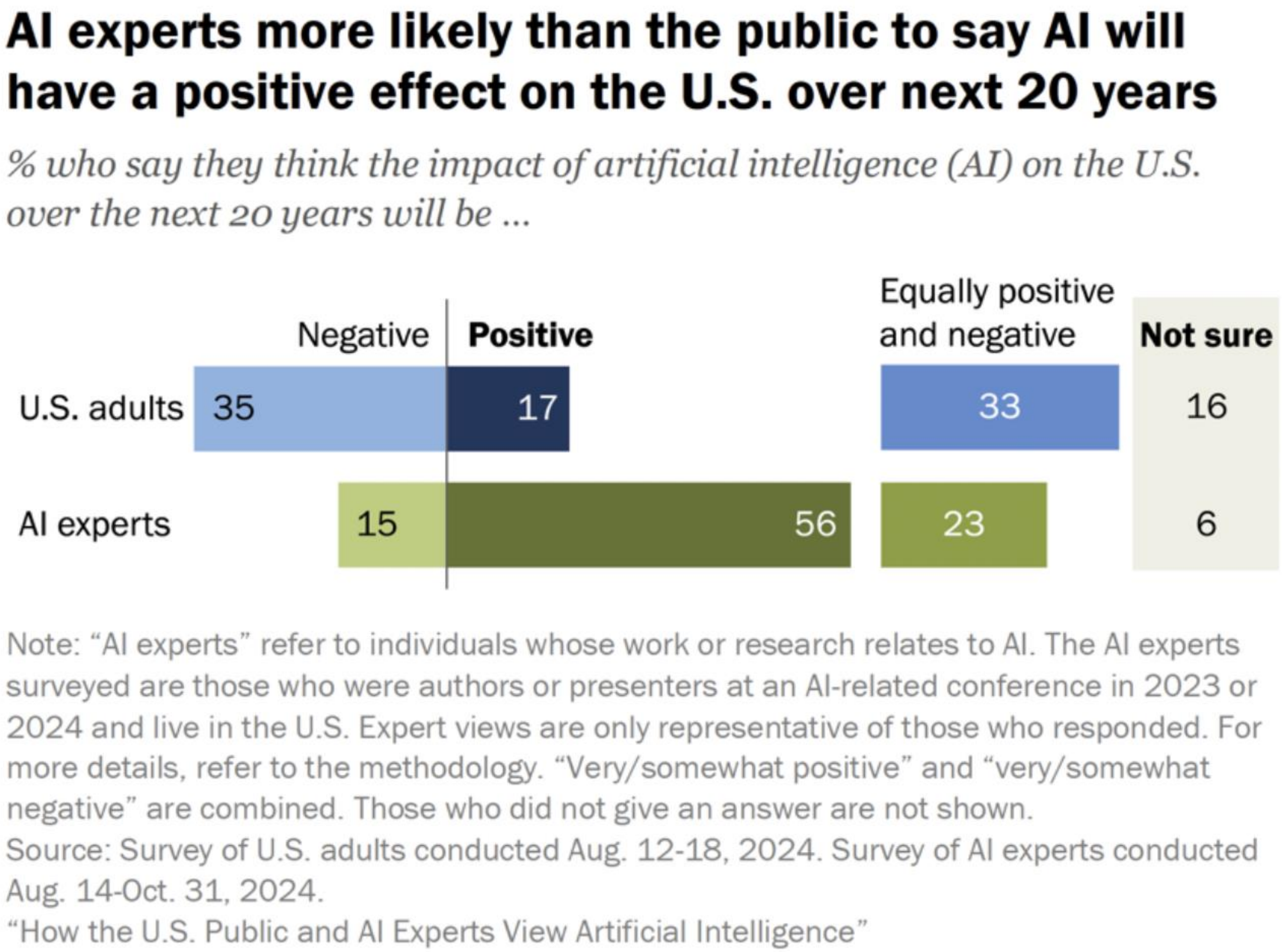
Regression

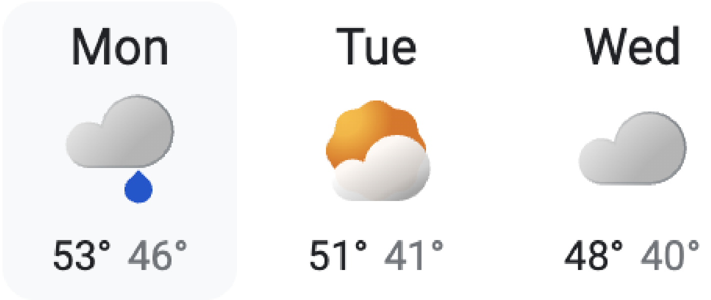
Reinforcement Learning

Generation

End Review

Impact of AI is Really Wild!





Chris' take: similar to how I understand the weather

I am curious…

How Does This Work?

create an image of a tree

Here's an image of a serene tree in a peaceful meadow.

another tree

Here's another beautiful tree, this time showcasing the vibrant colors of autumn.

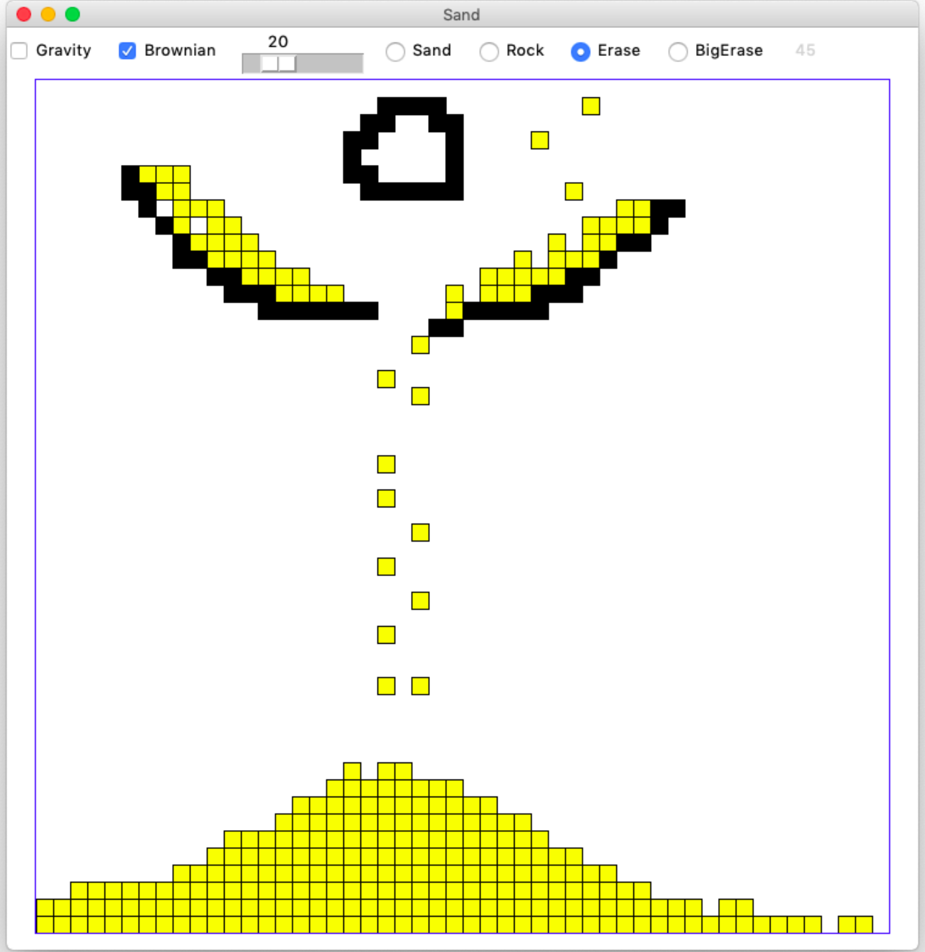
Inference is easy, but sampling is hard?

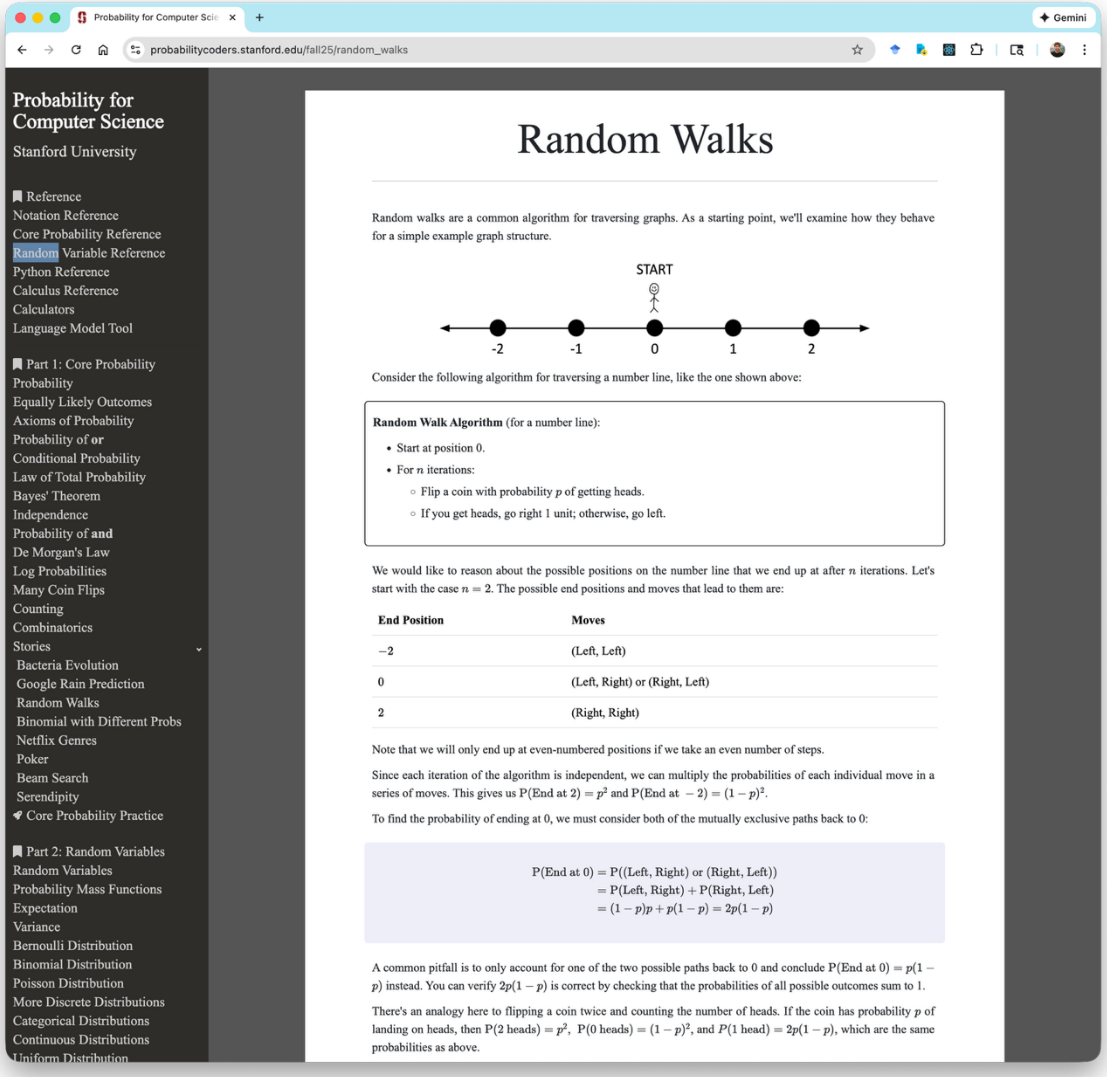
Goal: write an algorithm that can **generate** hand drawn number 8s

samples at step 236000 (class 8)

Challenge: can you come up with a (hacky, brute force) way to generate images if I give you a working classifier?

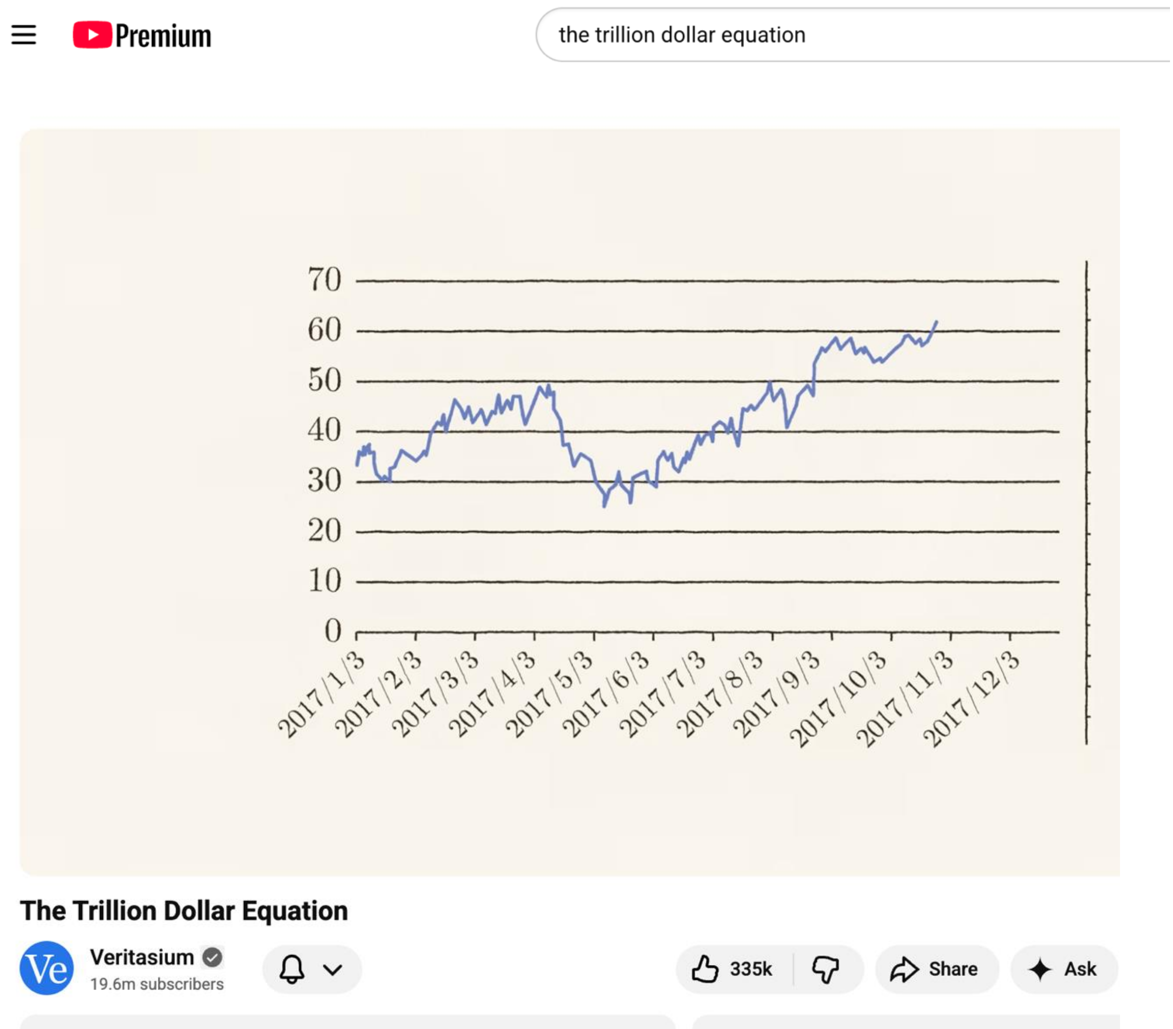
Random Walks

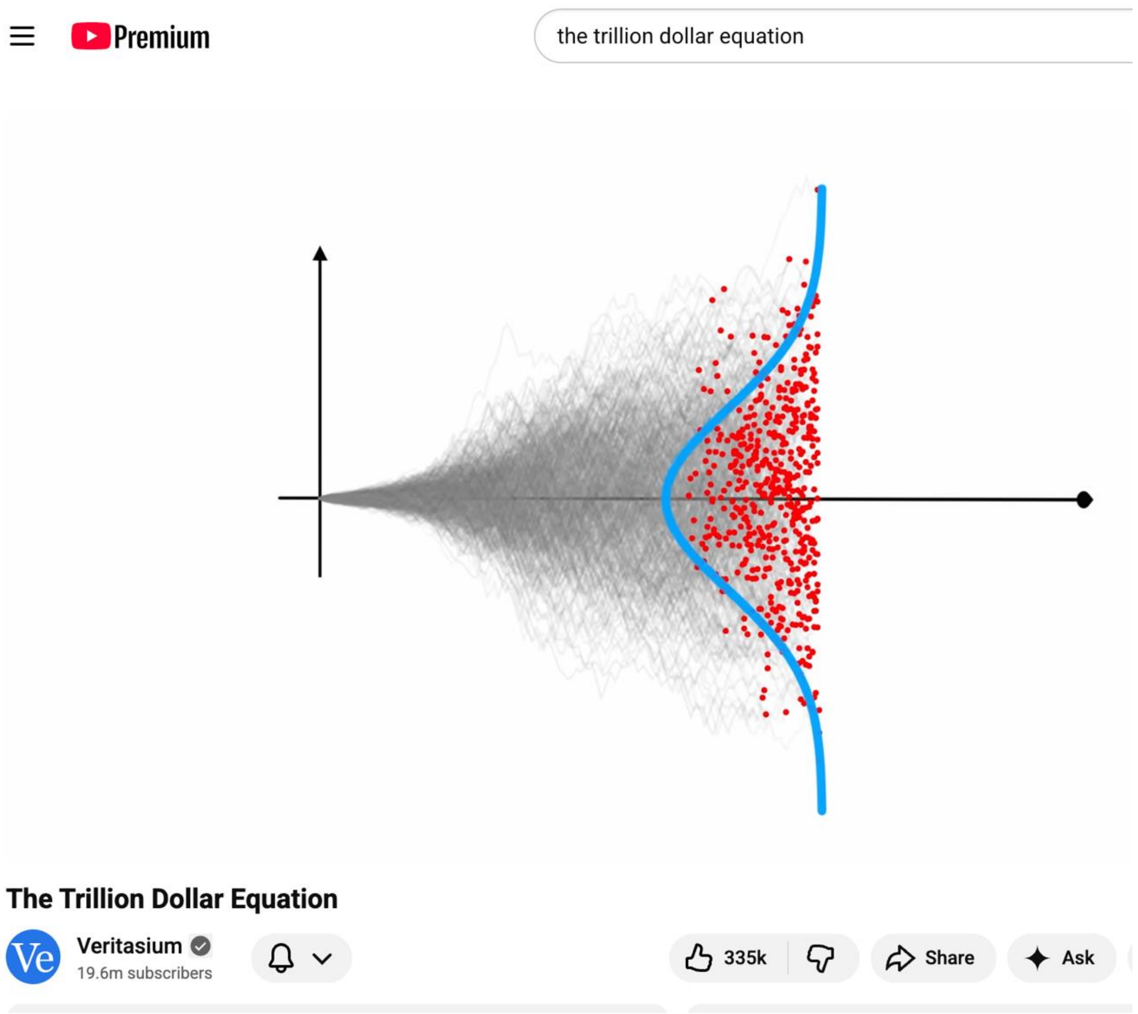




The "brownian" parameter is a number in the range 0..100 inclusive. When brownian is 20, that means there is a 20% chance that each sand will randomly try to move one square left or right each turn.

Random Walks -> Trillion Dollars





Black-Sholes Equation is based off thinking of stocks as random walks

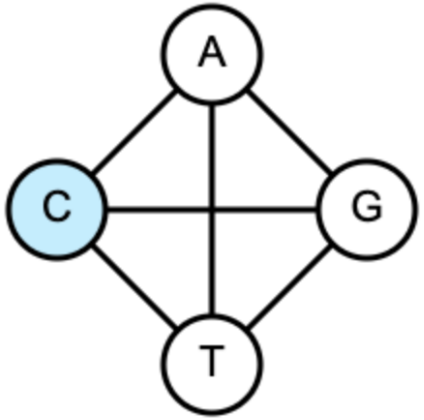
DNA Mutation Clocks

Now let's look at a more complex graph to analyze a problem about DNA sequence mutations.

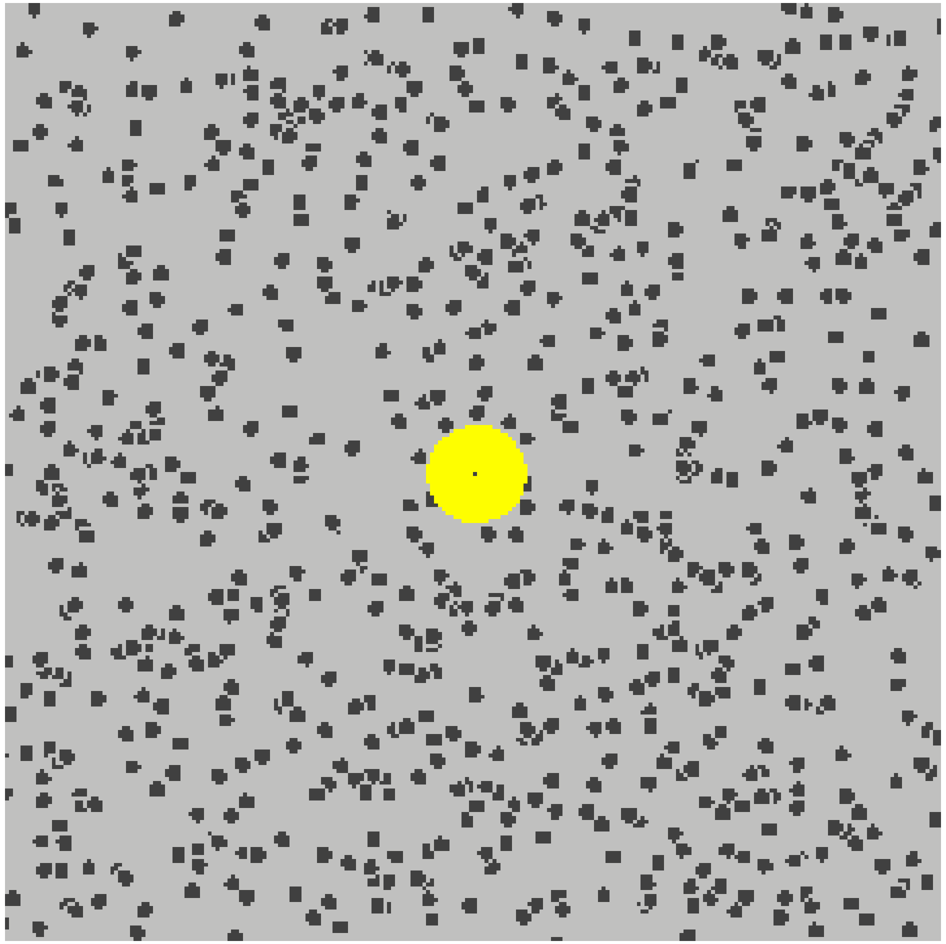
A species of bacteria is rapidly multiplying inside a dish. Each time a bacterium clones itself, it copies its DNA sequence (a very long ordered list made up of the letters A, C, G, and T). DNA copying is an error-prone process, so mutations sometimes occur -- i.e., at a certain position in the DNA sequence, the letter A changes to, say, a C. Each time a mutation occurs, the original letter changes to one of the three other letters with equal likelihood.

Assume that before any mutations occur, there is an A at a particular position in the DNA sequence. What is the probability that after n mutations, there is an A at this position again?

This problem might not sound like it's about a random walk at first, but it can be represented as one!



Diffusion in Physics!



Many **cool problems** related to random walks…

Today’s mystery: Can you infer where you came from?

Random Walks

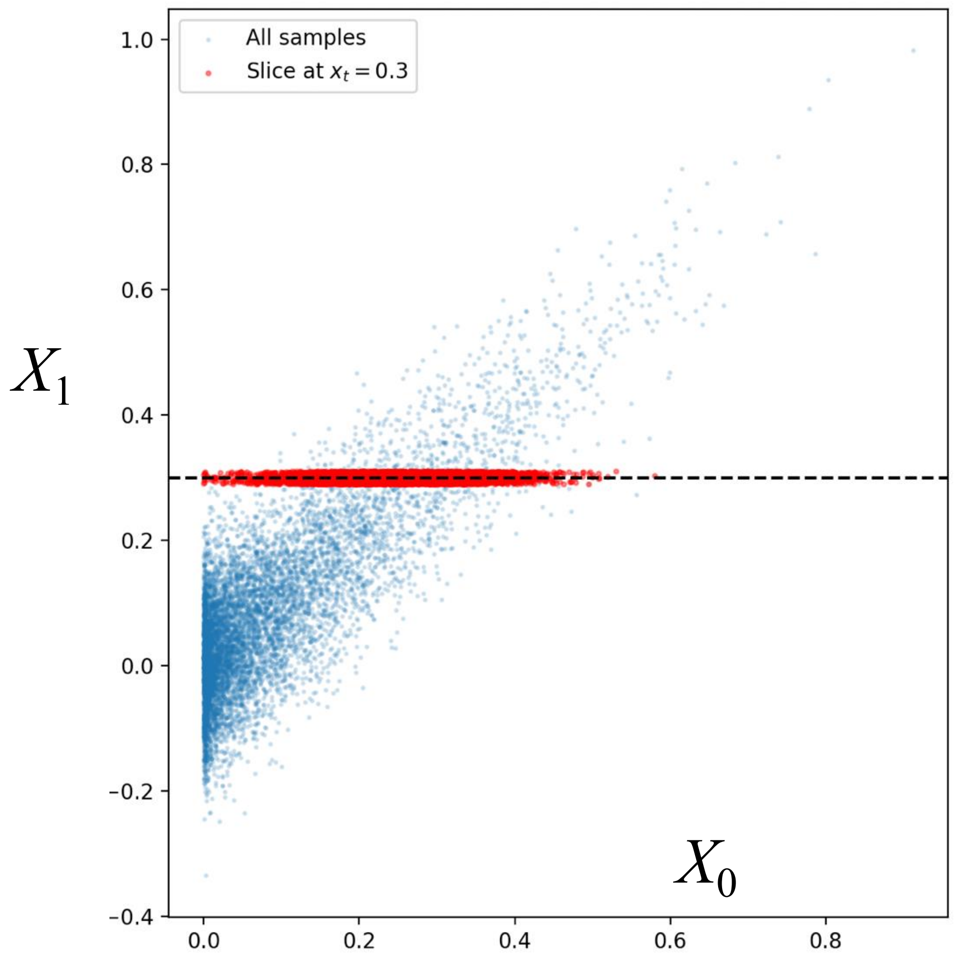
$$X_0 = 30$$

$$X_{t+1} = X_t + N_t \qquad N_i \sim N(\mu = 0, \sigma^2)$$

What is the probability distribution of X_{100}?

You observe $X_{101} = 10$
What is the probability density of X_{100}?

Random Walks



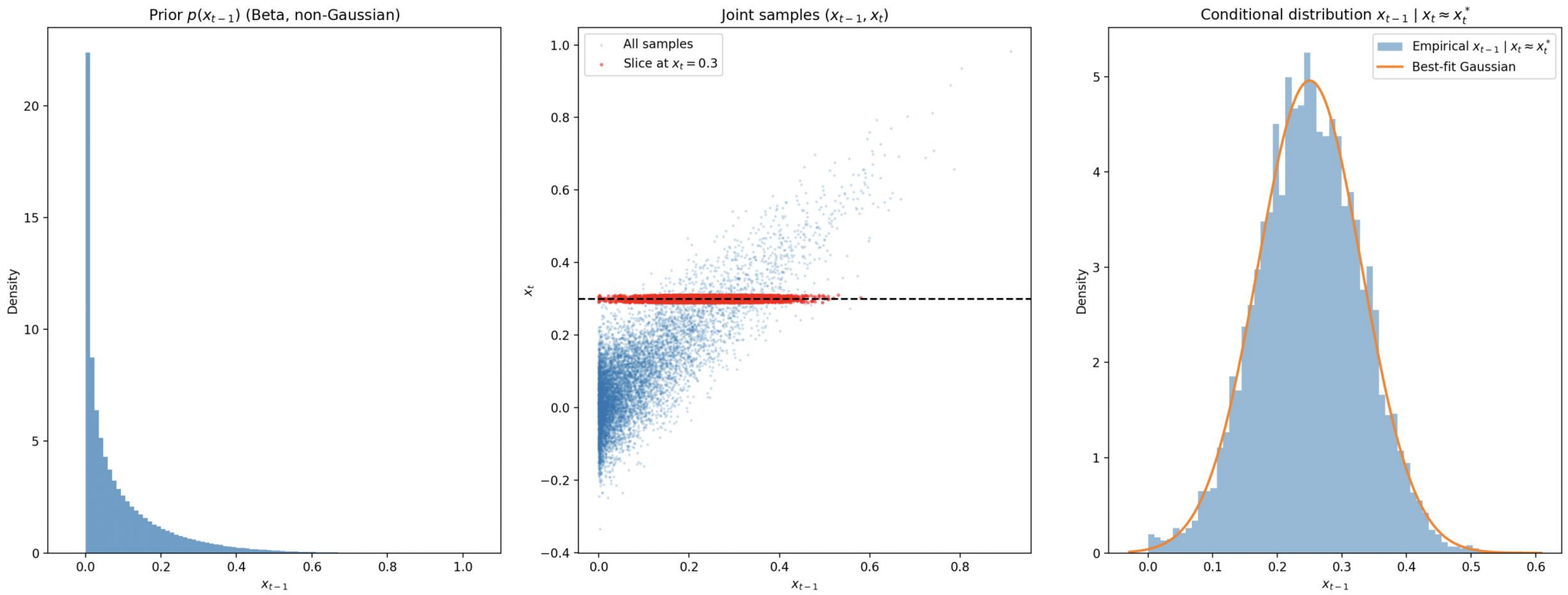
$$X_0 \sim \text{Beta}(2, 7)$$

$$X_{t+1} = X_t + N_t \qquad N_i \sim N(\mu = 0, \sigma^2)$$

You observe $X_1 = 0.3$

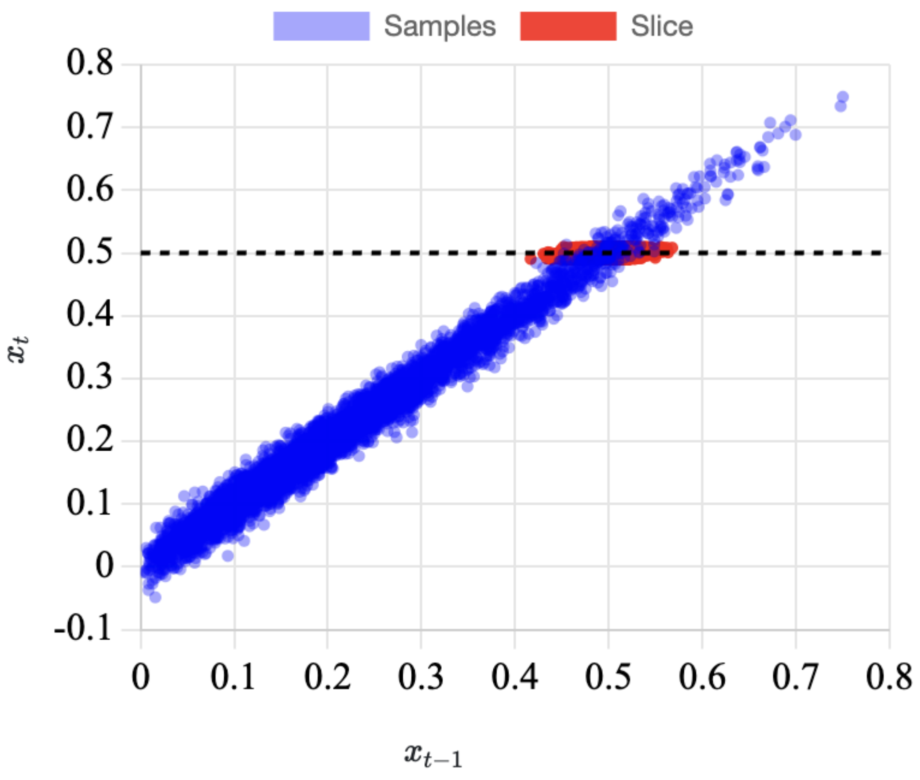
What is the probability density of X_0?

This One Strange Fact

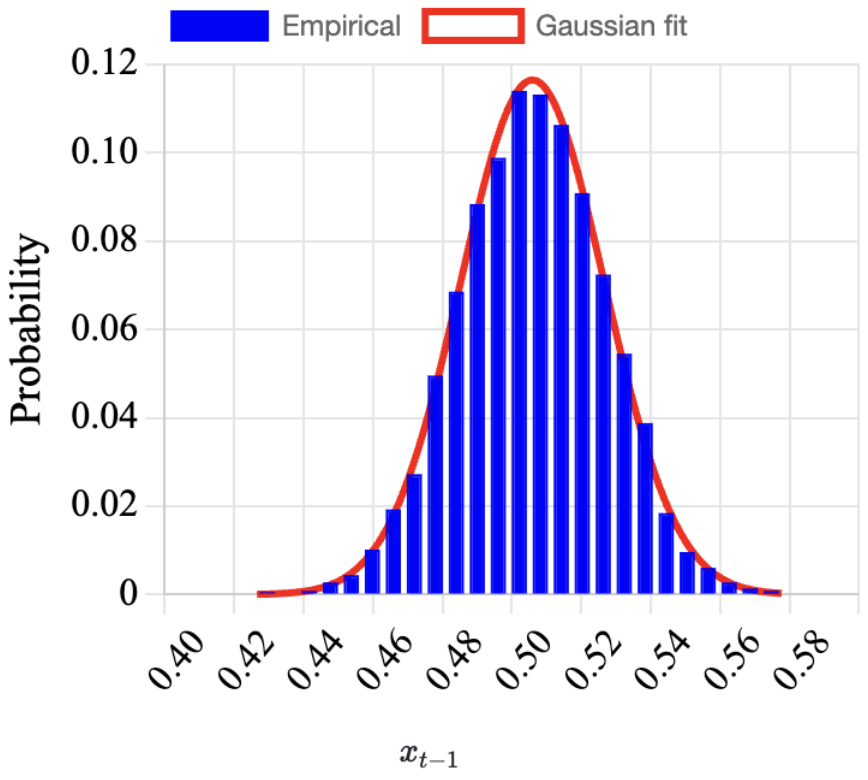


This One Strange Fact: Generalizes

Joint $P(X_{t-1} = x_{t-1}, X_t = x_t)$



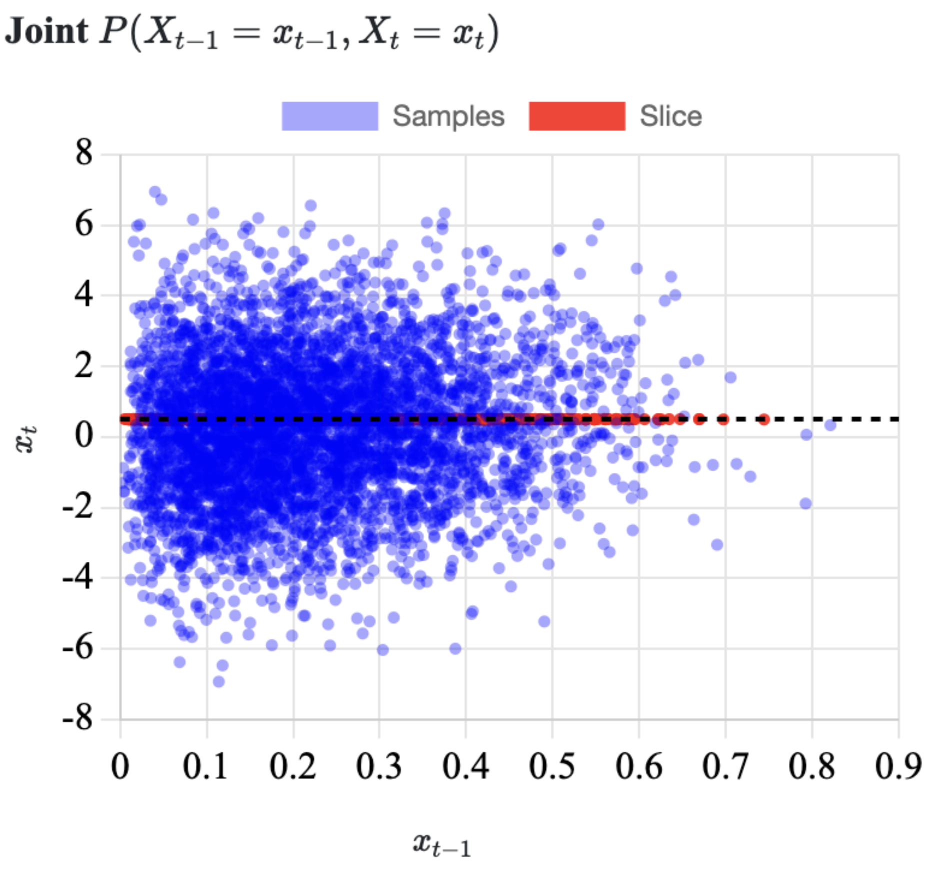
Posterior $P(X_{t-1} = x_{t-1} \mid X_t \approx k)$

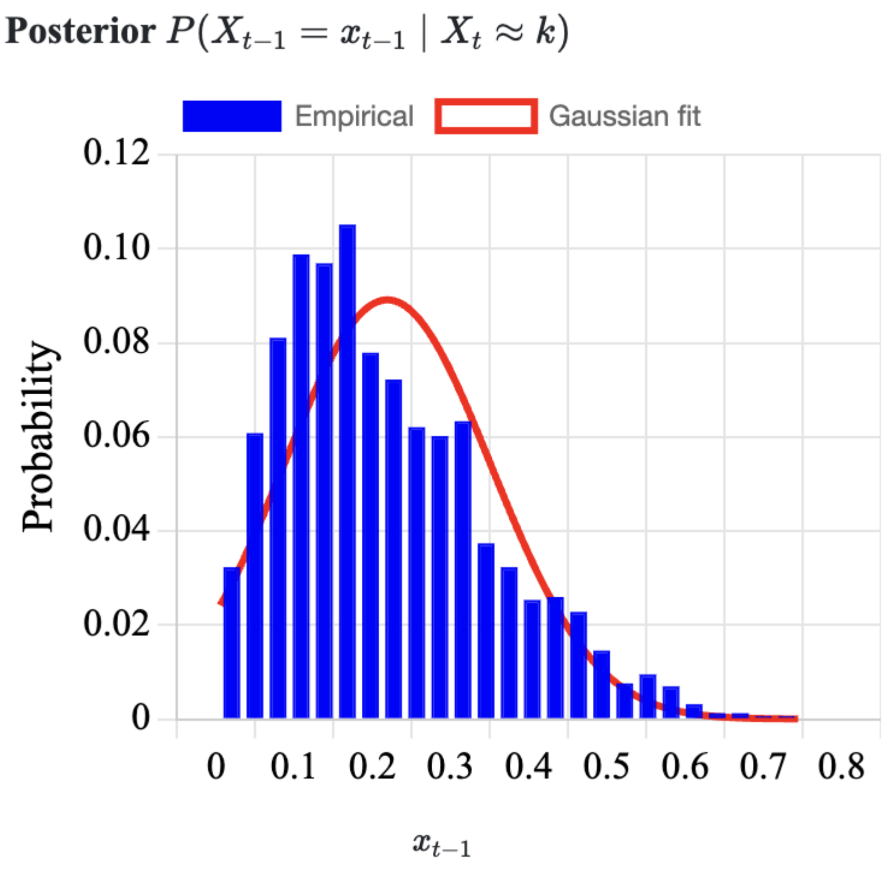


k: 0.5 σ: 0.02 Slice width: 0.01 Samples in slice: **3631**

https://probabilitycoders.stanford.edu/fall25/diffusion

This One Strange Fact: Generalizes (if σ is small)





k: 0.5 σ: 2 Slice width: 0.01 Samples in slice: **1579**

https://probabilitycoders.stanford.edu/fall25/diffusion

Diffusion Critical Fact

If

$$X_{t+1} = X_t + N_t \qquad N_i \sim N(\mu = 0, \sigma^2)$$

This must be a small known value

Then

$$X_t | X_{t+1} \sim N(\mu, \sigma^2)$$

This value is unknown. Must figure it out

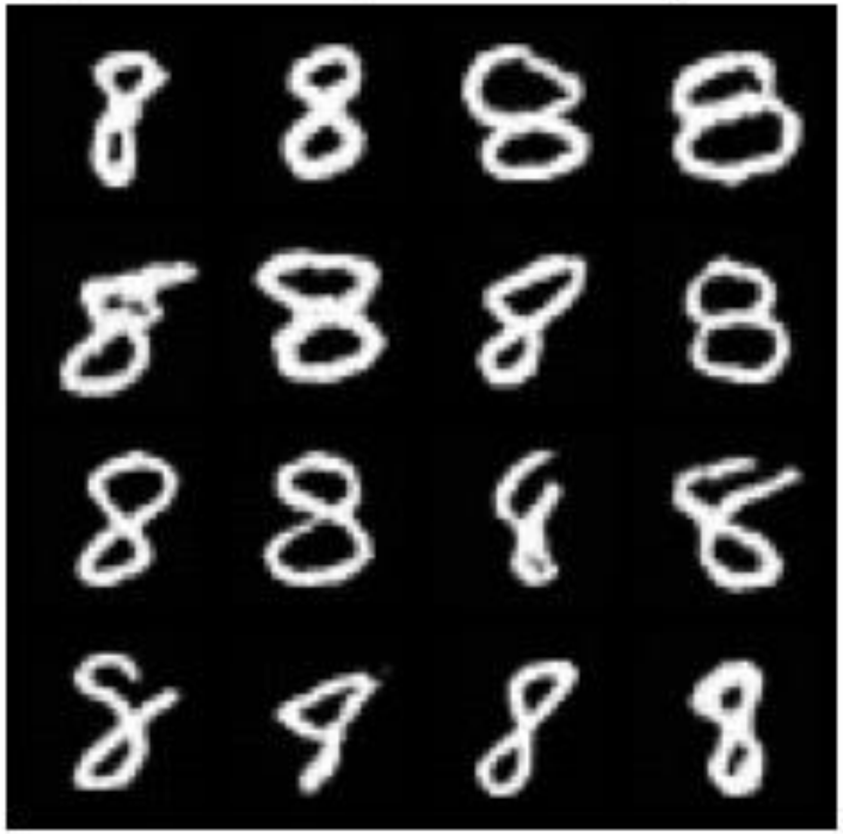
Regardless of the prior distribution of X_t

https://probabilitycoders.stanford.edu/fall25/diffusion

TLDR: We just need a neural network that can predict the added noise (and we get a backwards process we can sample from)

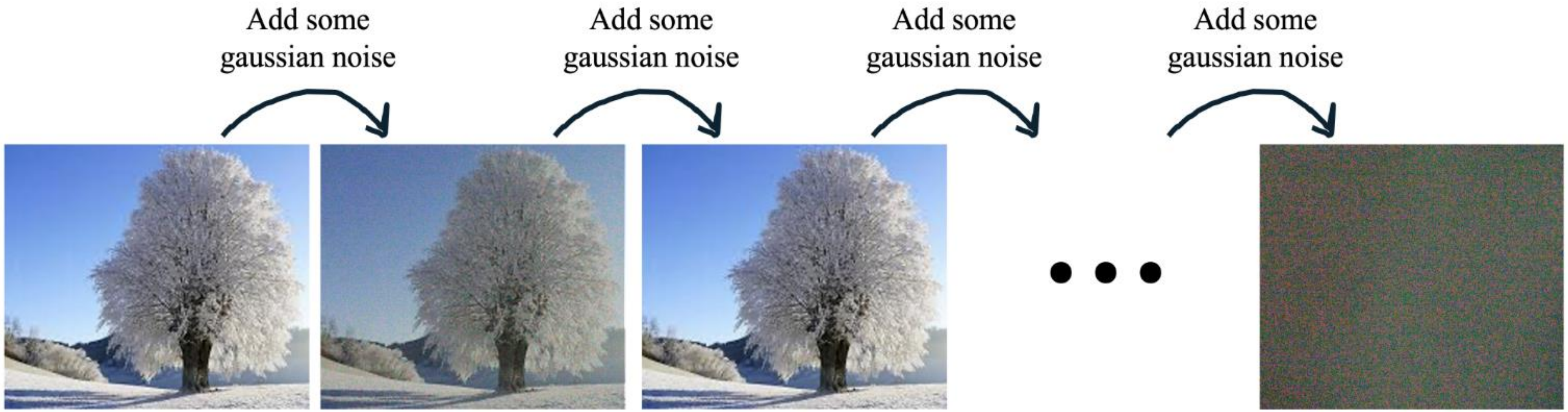
I generated this 8s this morning!

samples at step 236000 (class 8)

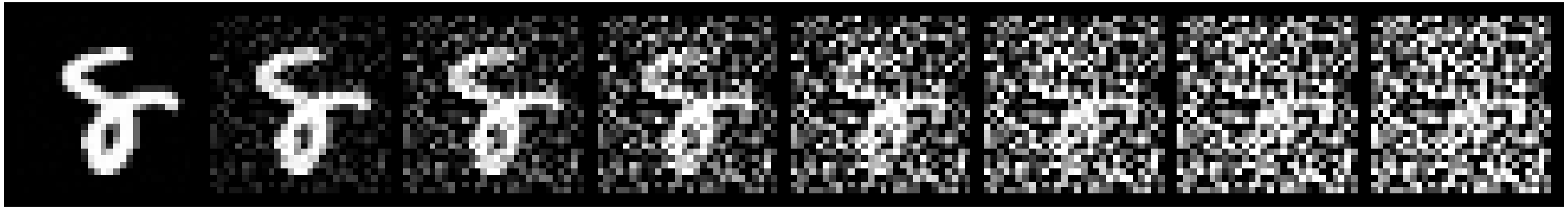


How?

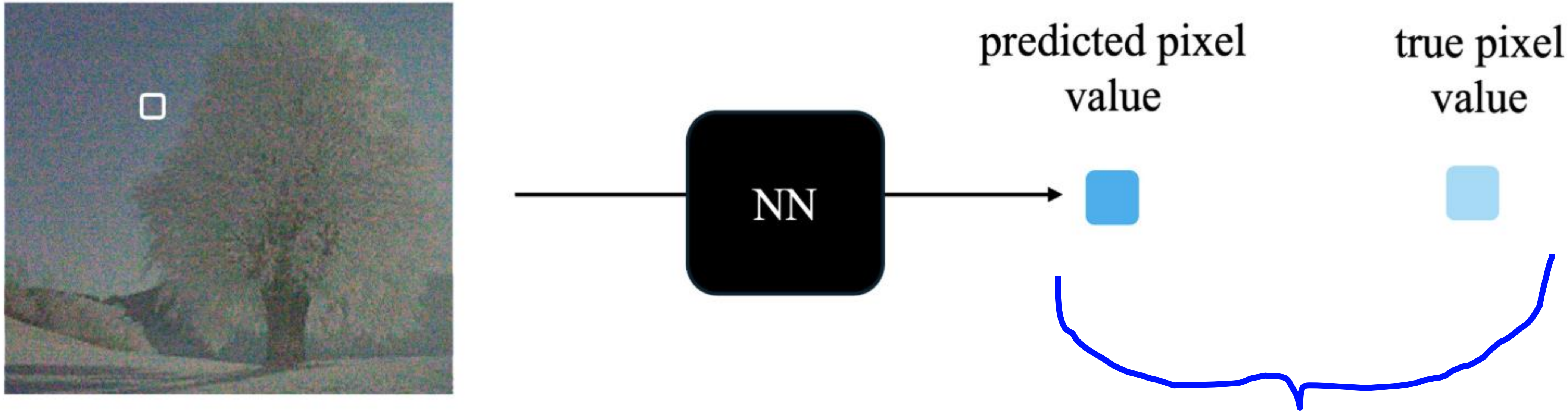
Take Your Images on a Walk!



Forward noising (class 8)

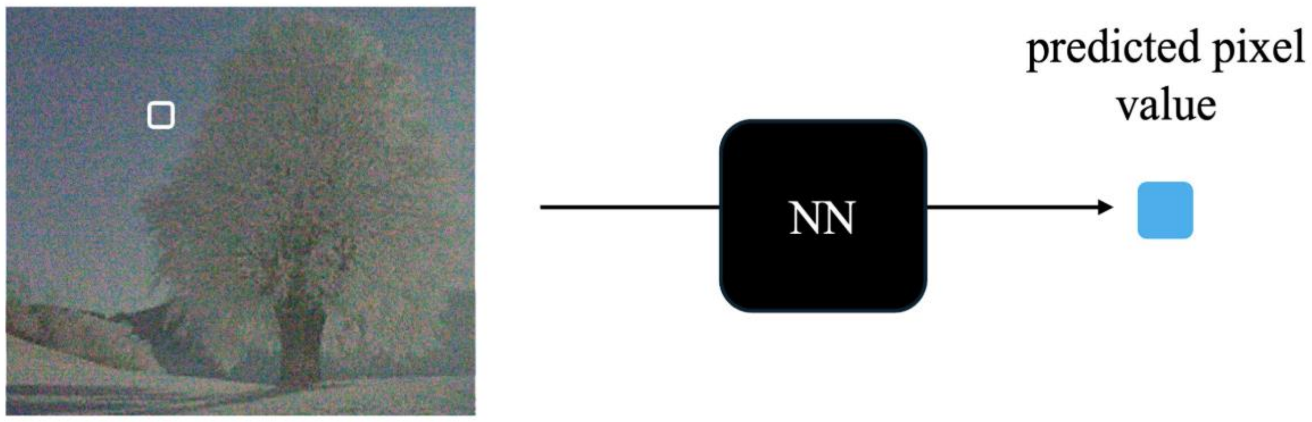


We Need a Neural Network that Denoises

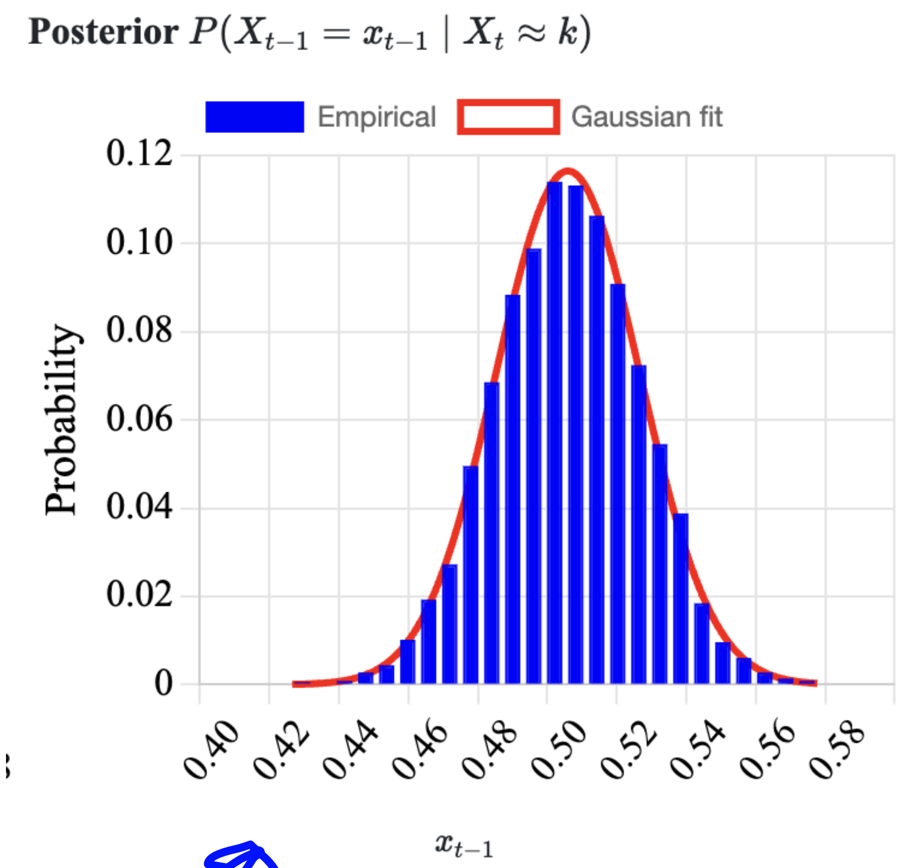


Used KL divergence to show you just need to predict the squared error

Learning the mean is all we need!



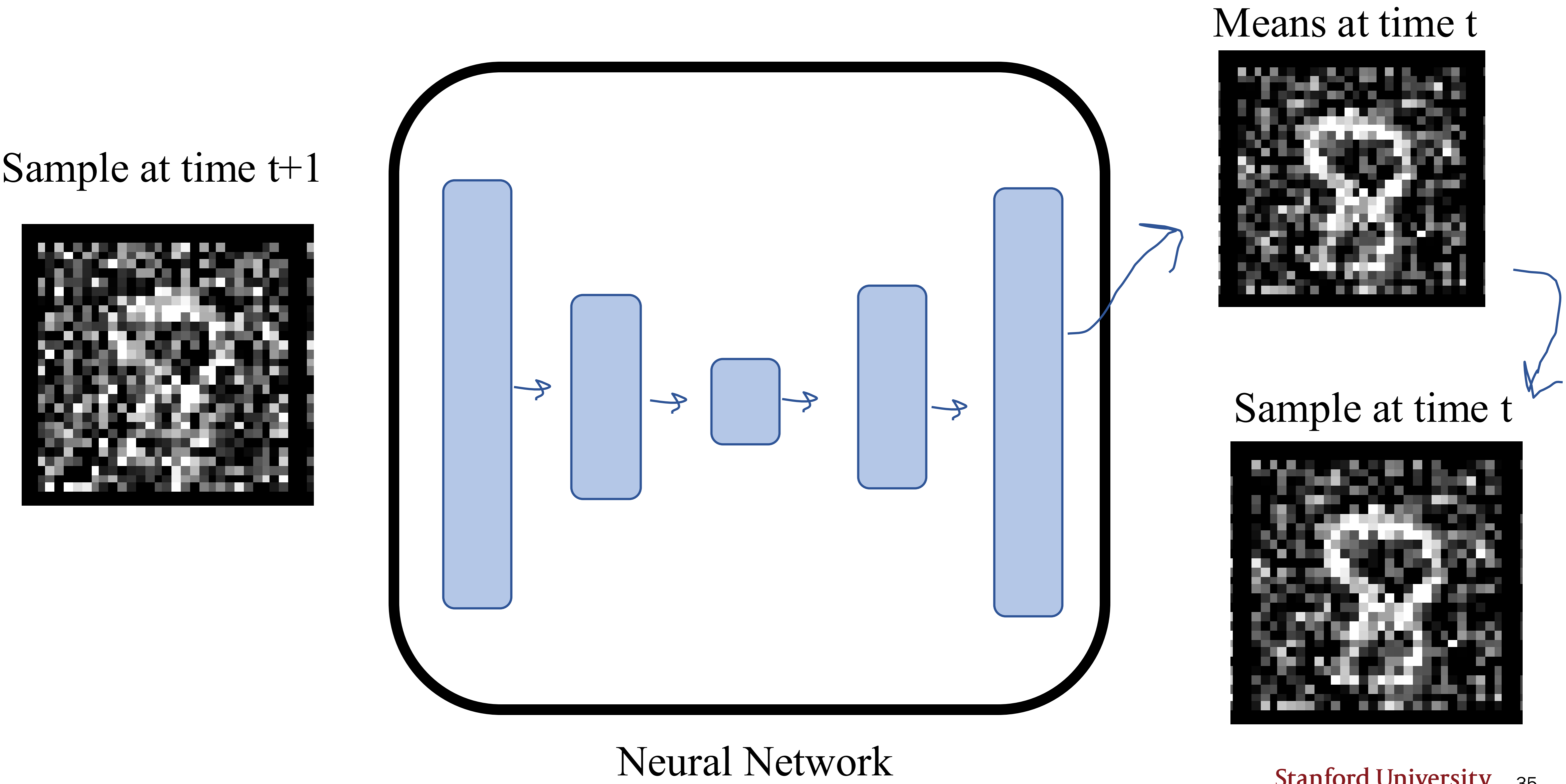
This is the mean of the posterior gaussian



Now you have a distribution for the reverse process

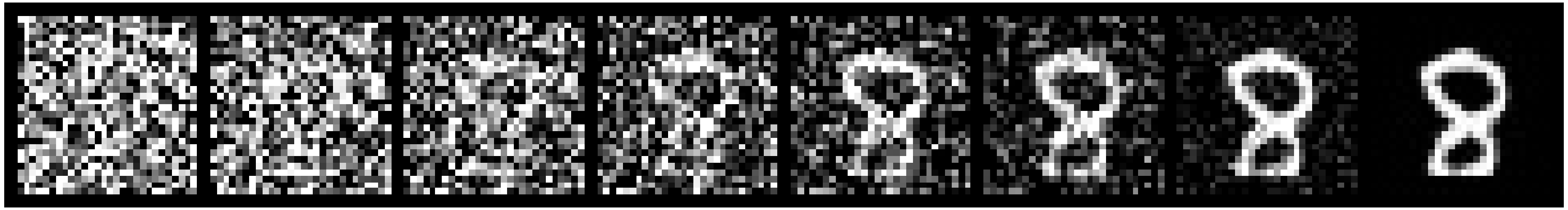
https://probabilitycoders.stanford.edu/fall25/diffusion

Inside the Neural Network

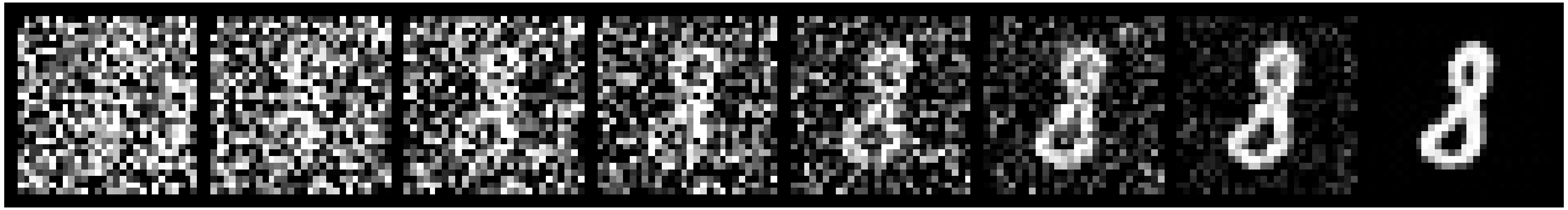


We learn how to remove noise!

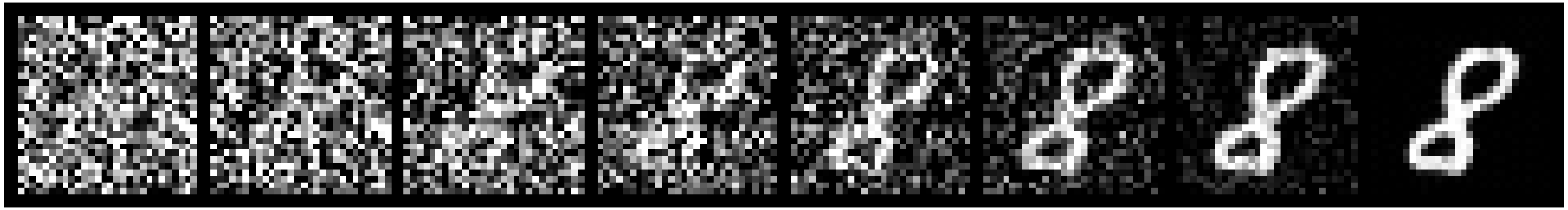
Reverse denoising (class 8)



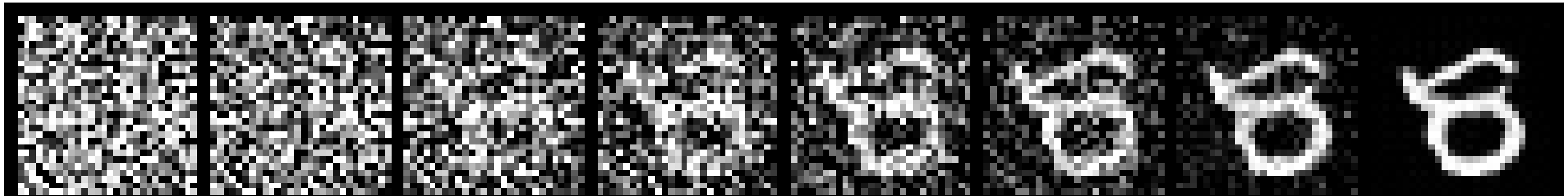
Reverse denoising (class 8)



Reverse denoising (class 8)

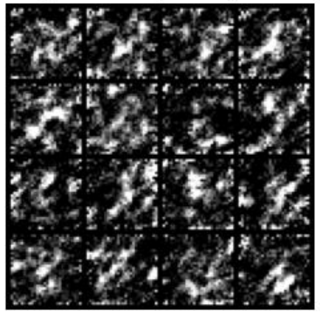


Reverse denoising (class 8)

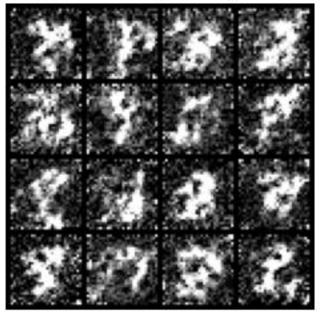


Training a Neural Network

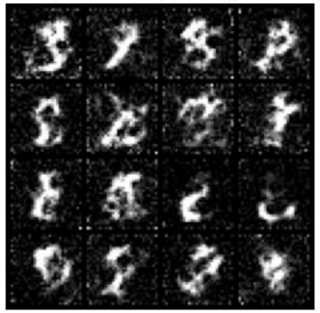
samples at step 100 (class 8)



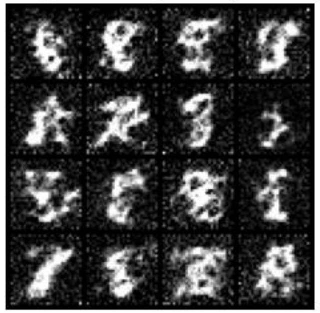
samples at step 200 (class 8)



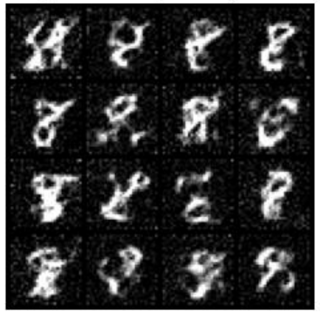
samples at step 300 (class 8)



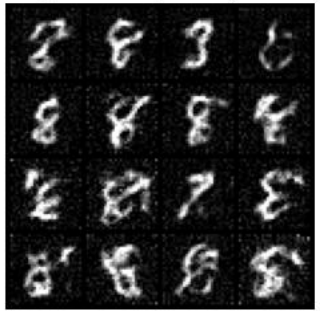
samples at step 400 (class 8)



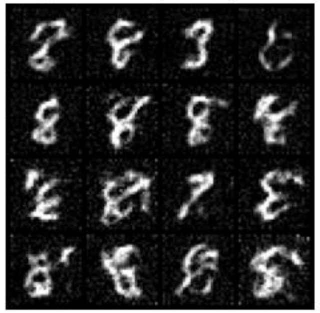
samples at step 500 (class 8)



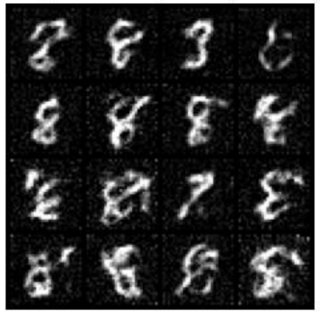
samples at step 600 (class 8)



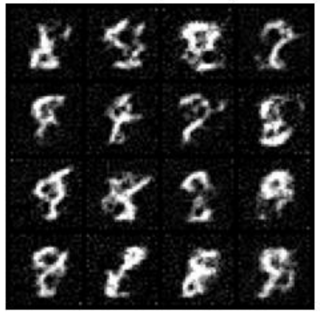
samples at step 600 (class 8)



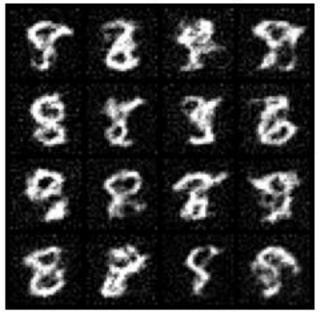
samples at step 600 (class 8)



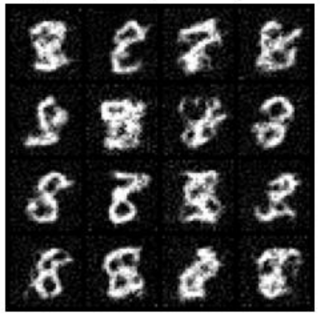
samples at step 900 (class 8)



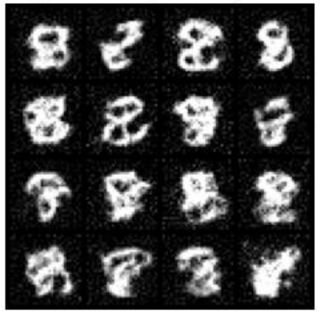
samples at step 1000 (class 8)



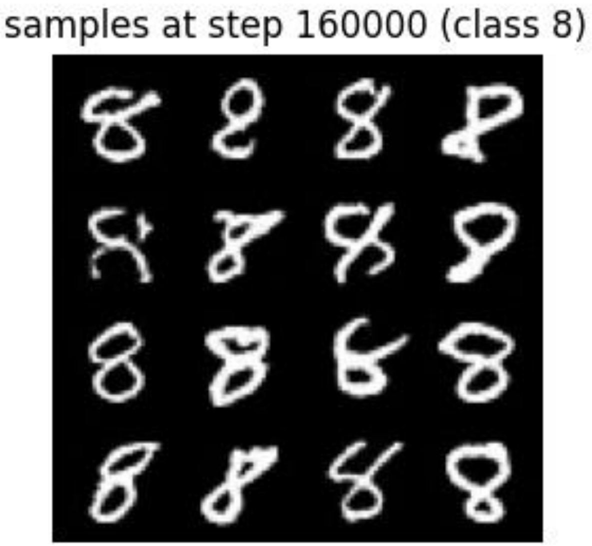
samples at step 1100 (class 8)

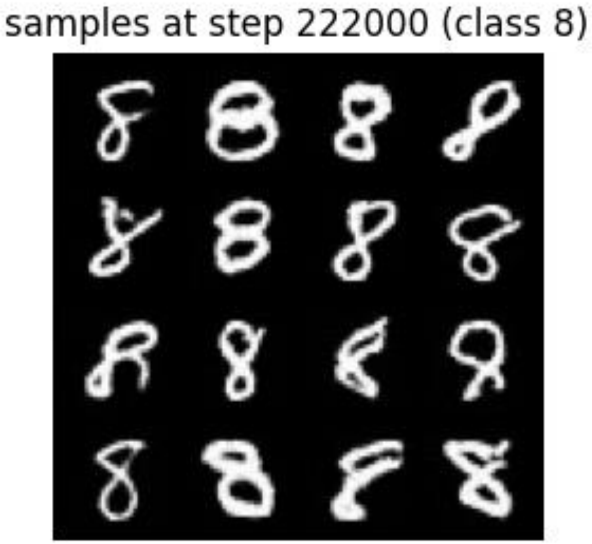


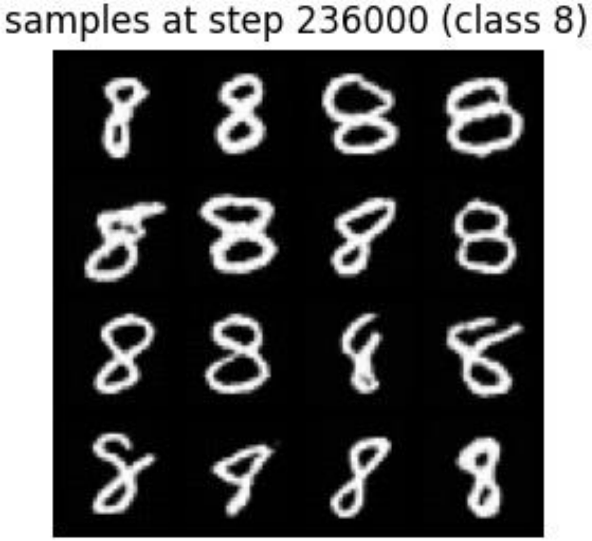
samples at step 1200 (class 8)

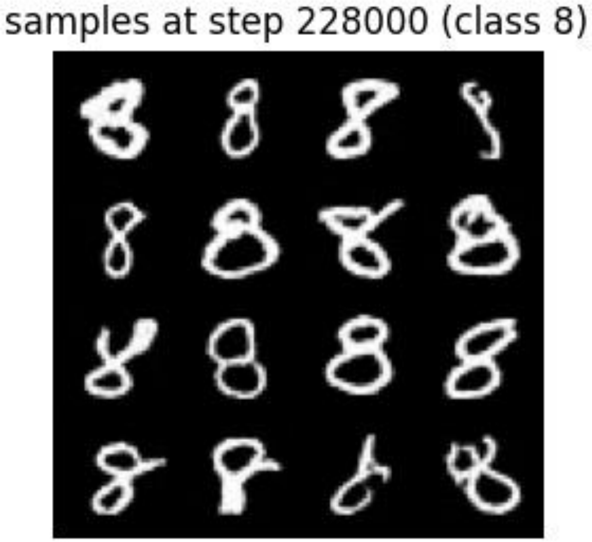


Trained for 20 mins. Works fine!



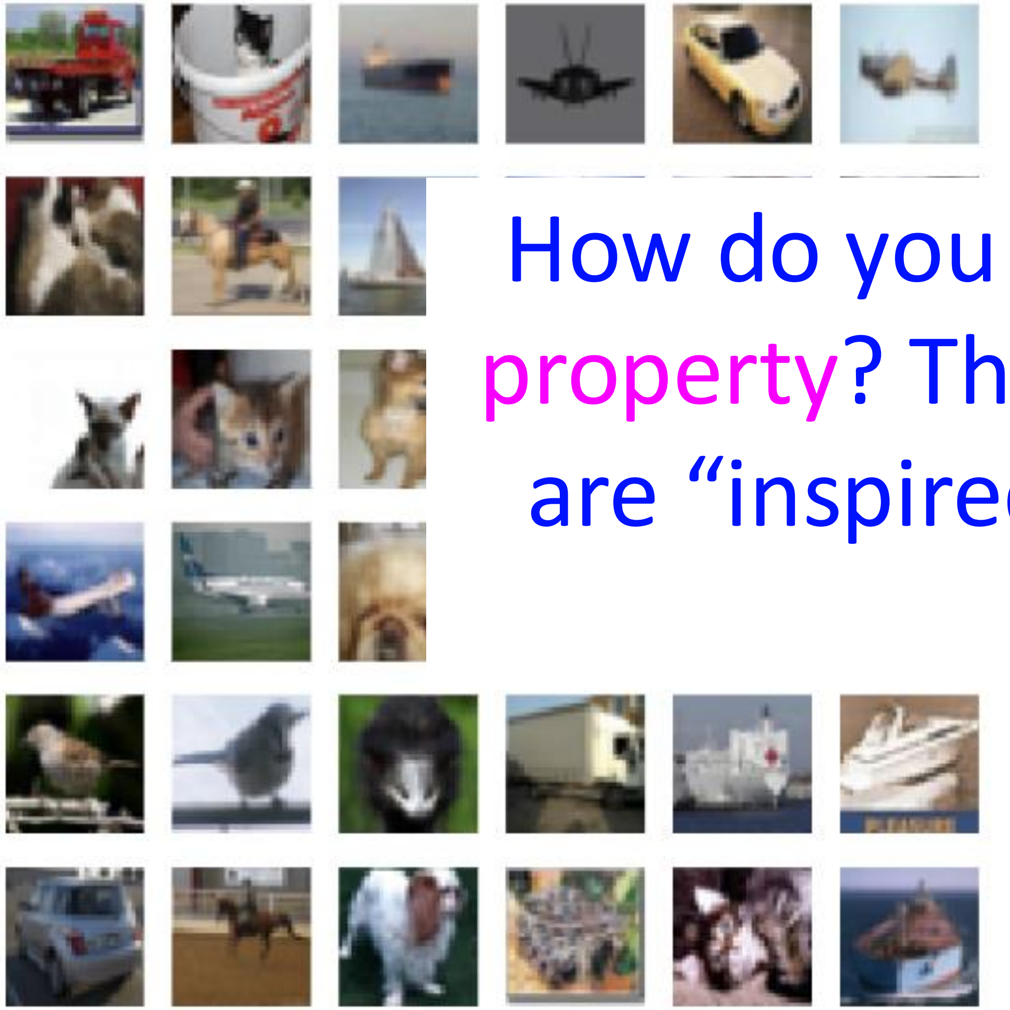






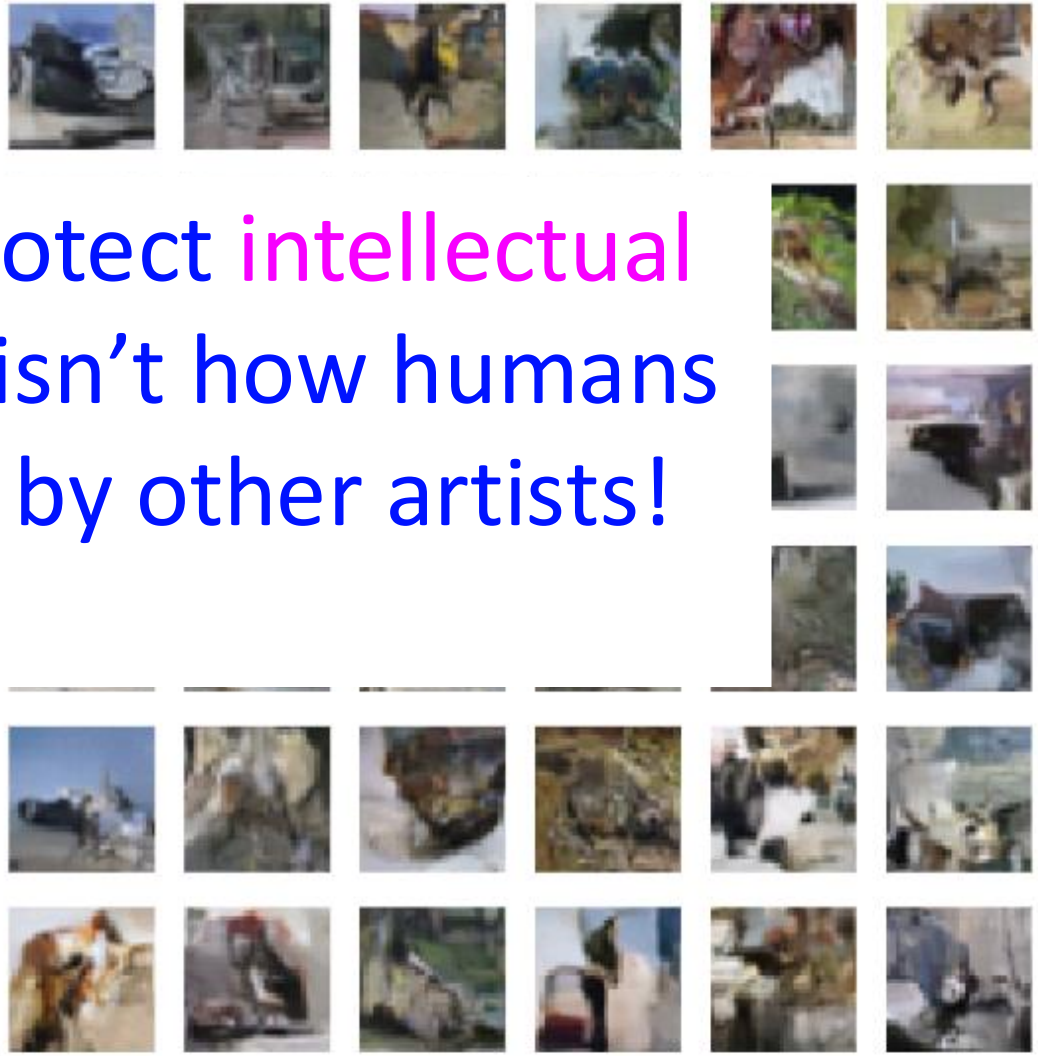
History at Stanford: Original Diffusion Paper

History at Stanford: Original Diffusion Paper



(a)

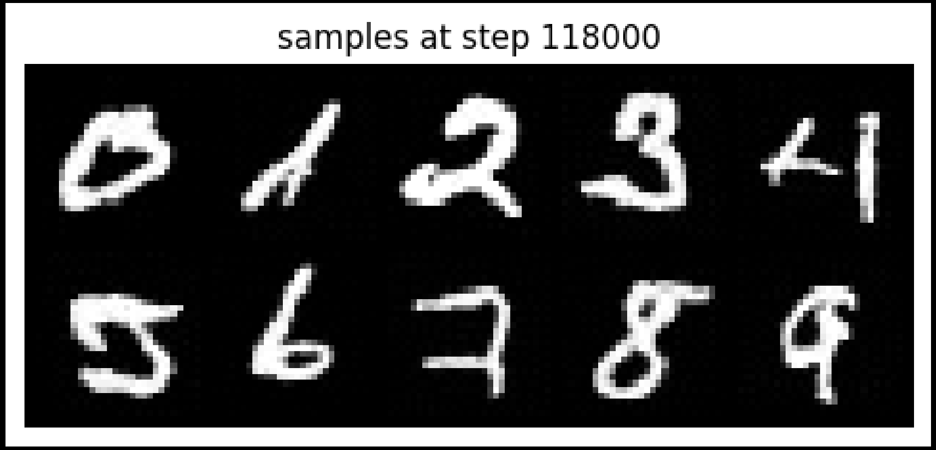
How do you protect intellectual property? This isn't how humans are "inspired" by other artists!



(d)

Which artists was this based off?

How do you use text to guide the generation of images?



Ran into Jascha over Thanksgiving

Can you use a classifier to guide generation of an image?

On Wed!

CS109 App

cs109psets.netlify.app/fall24/music/ratings

Gemini

#	song	Sample Mean PDF	votes	numVotes	SampleMean	SEOM	songId	Pr(Top16)	Pr(Best)
1	Get Lucky - Daft Punk			45	3.82	0.18	117	0.845	0.500
2	Let It Be - The Beatles			41	3.8	0.19	150	0.806	0.470
3	Life is a Highway - Rascal Flatts			36	3.78	0.24	67	0.730	0.447
4	Upside Down - Jack Johnson			93	3.68	0.14	137	0.630	0.270
5	September - Earth, Wind & Fire			55	3.6	0.16	15	0.424	0.180
6	Time of Our Lives - Pitbull			24	3.58	0.26	39	0.420	0.224
7	Vienna - Billy Joel			24	3.58	0.28	140	0.425	0.235
8	Just the Two of Us (feat. Bill Withers) - Grover Washing			25	3.56	0.22	78	0.372	0.180
9	Voulez-Vous - ABBA			20	3.55	0.27	69	0.381	0.203
10	Let it Happen - Tame			22	3.55	0.29	73	0.388	0.214
11	Careless Whisper - George Michael			24	3.54	0.24	87	0.350	0.175
12	Take Five - Dave Brubeck			18	3.5	0.33	37	0.348	0.197
13	Clairo - Juna			18	3.5	0.28	57	0.318	0.168
14	We Are The Champions - Queen			22	3.5	0.24	77	0.289	0.143
15	All Star - Smash Mouth			17	3.41	0.37	0	0.272	0.160
16	Feel it Still - Portugal. The Man			17	3.41	0.32	68	0.244	0.132