Using AI to improve our Understanding of Waste-water processing

Stephen McGough

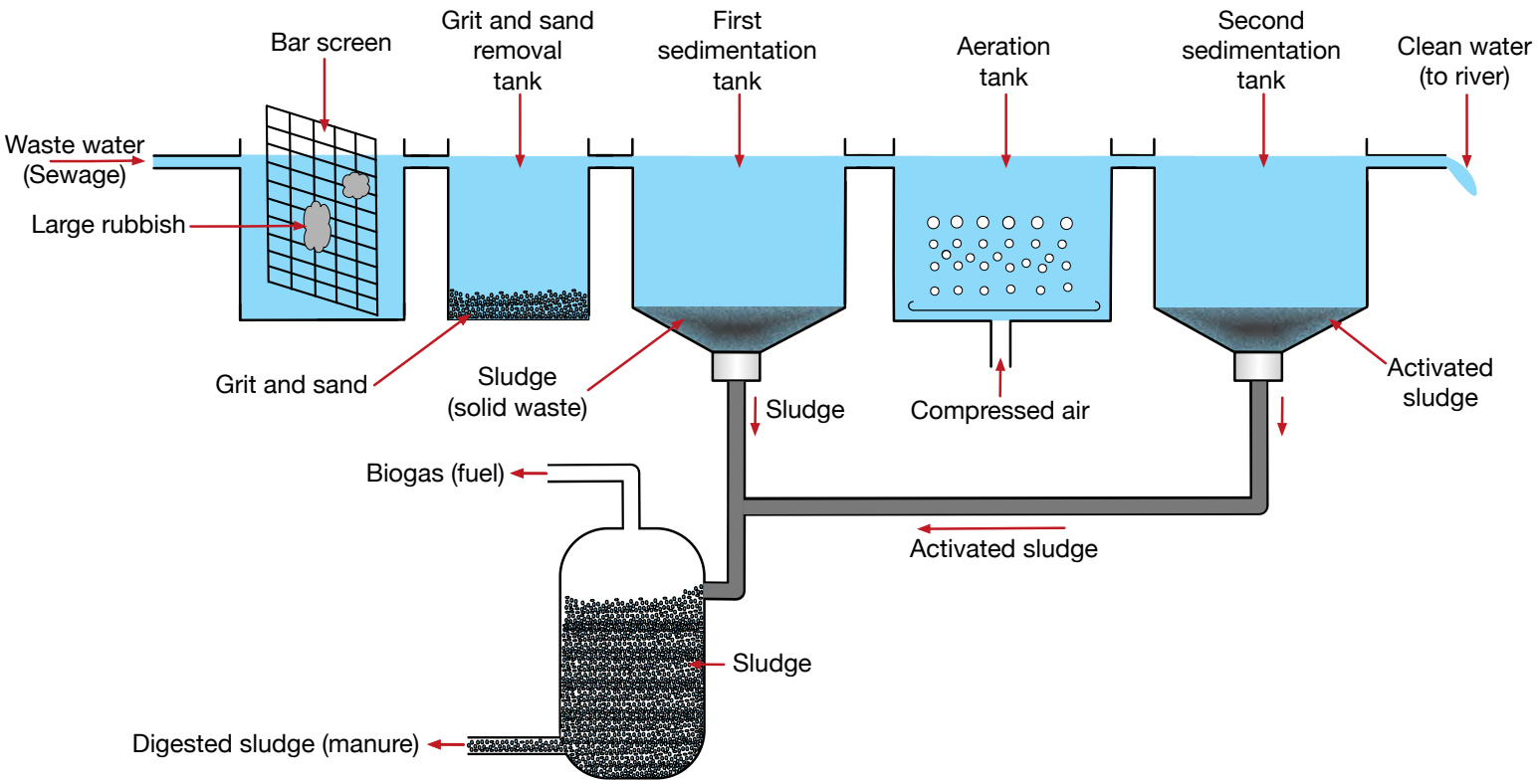
Newcastle University, Fellow Alan Turing Institute

ENBIS Meeting, May 18th, 2021

Outline

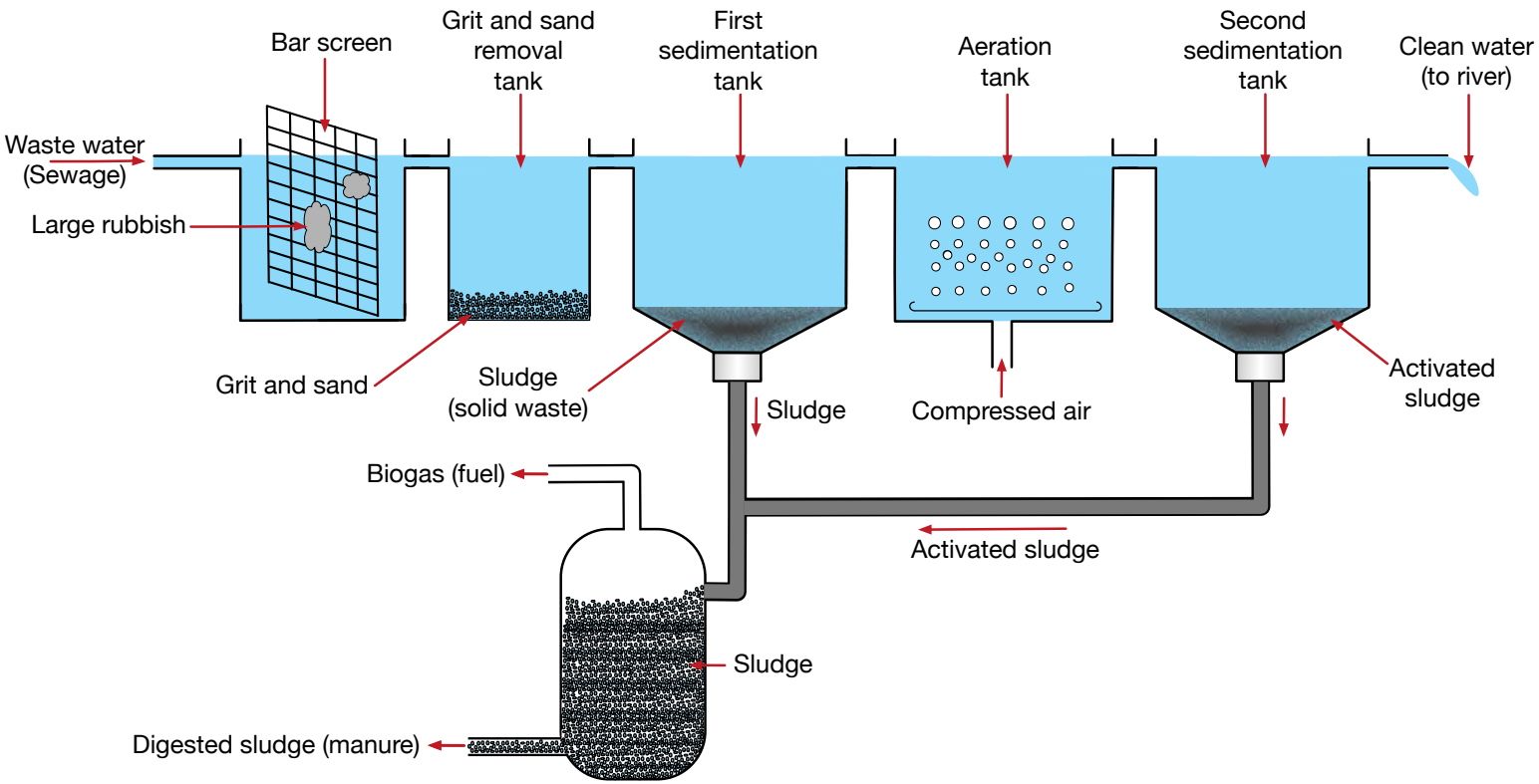
- **Wastewater Treatment Primer**
- Simulating wastewater treatment
- AI Primer
- Using AI for wastewater treatment

Wastewater treatment process

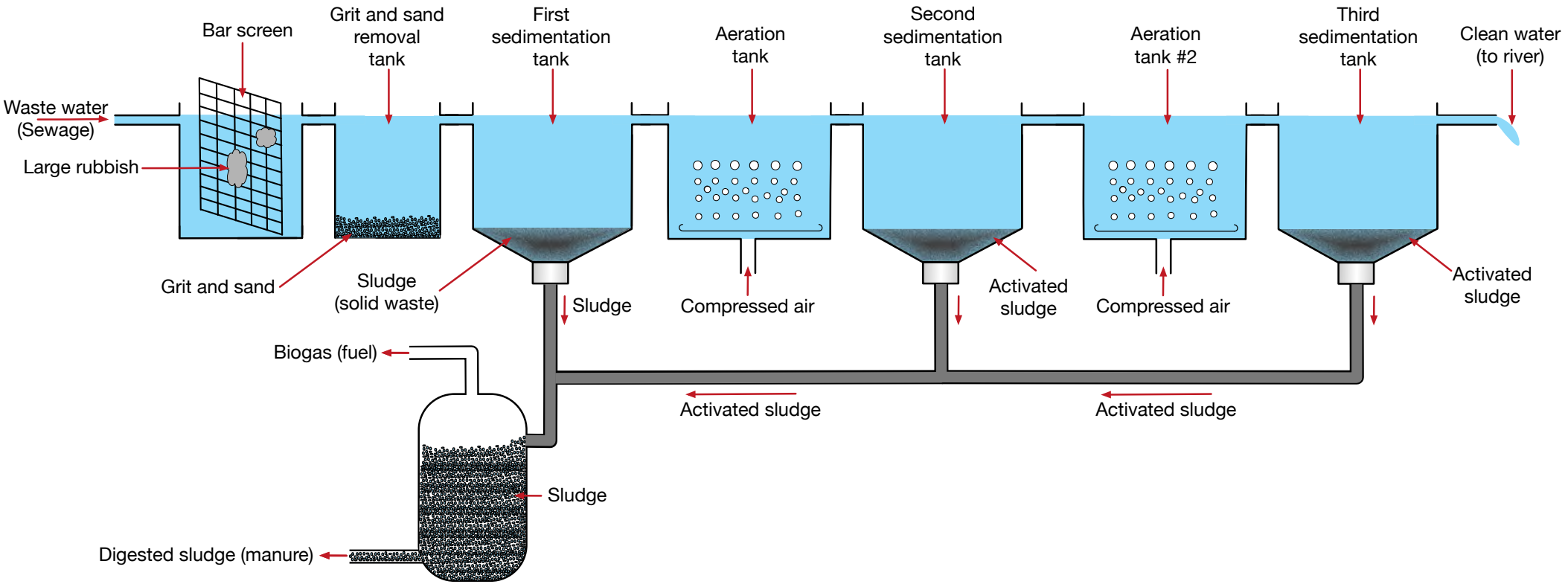


- Process dates back to 1868
- Bacteria in sludge breaks down harmful chemicals
- But new directives require lower levels of contaminates in outflow

Dealing with contaminates



Dealing with contaminates



Want to do this ’better’

- No extra energy input
- No need for extra facilities
- How?
  - The bacteria
  - The environment

Outline

- Wastewater Treatment Primer
- **Simulating wastewater treatment**
- AI Primer
- Using AI for wastewater treatment

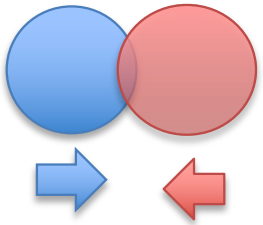
Agent Based Model

- Simulate system – Agent Based Model
- Each bacteria is an agent
  - Acts out the bacteria's biological and mechanical – processes

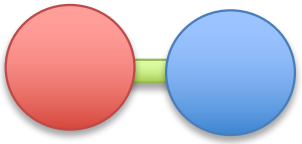
Process Stages

Mechanical Relaxation

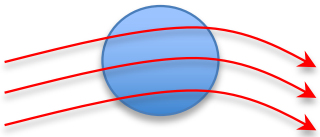
Contact force



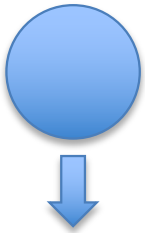
Adhesive force



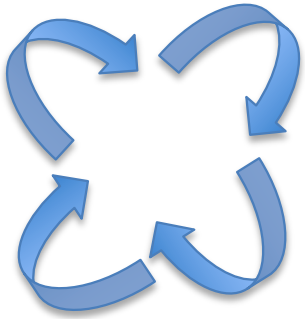
Fluid force



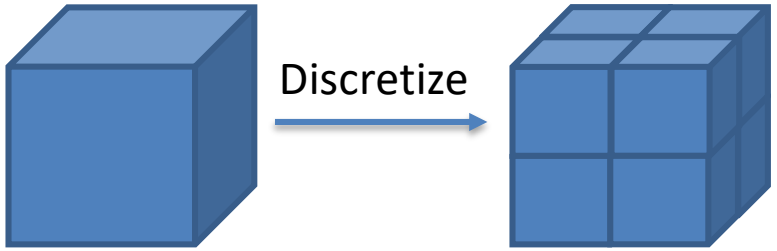
Gravitational force



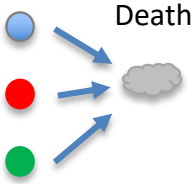
Nutrient Mass Balance

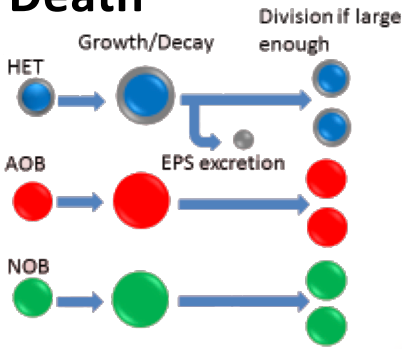


Solve using FDM

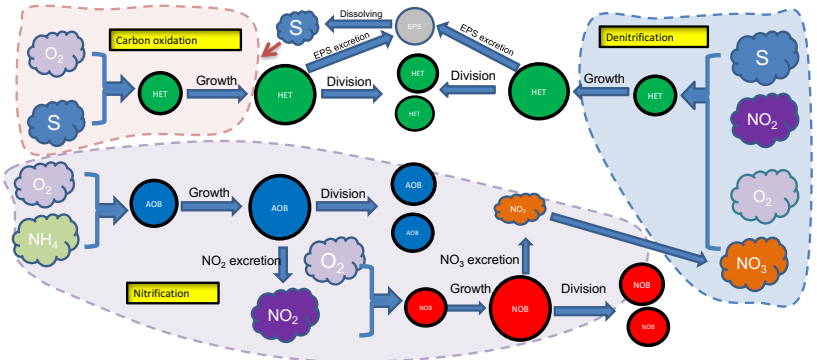


Handle Growth/Decay and Birth / Death



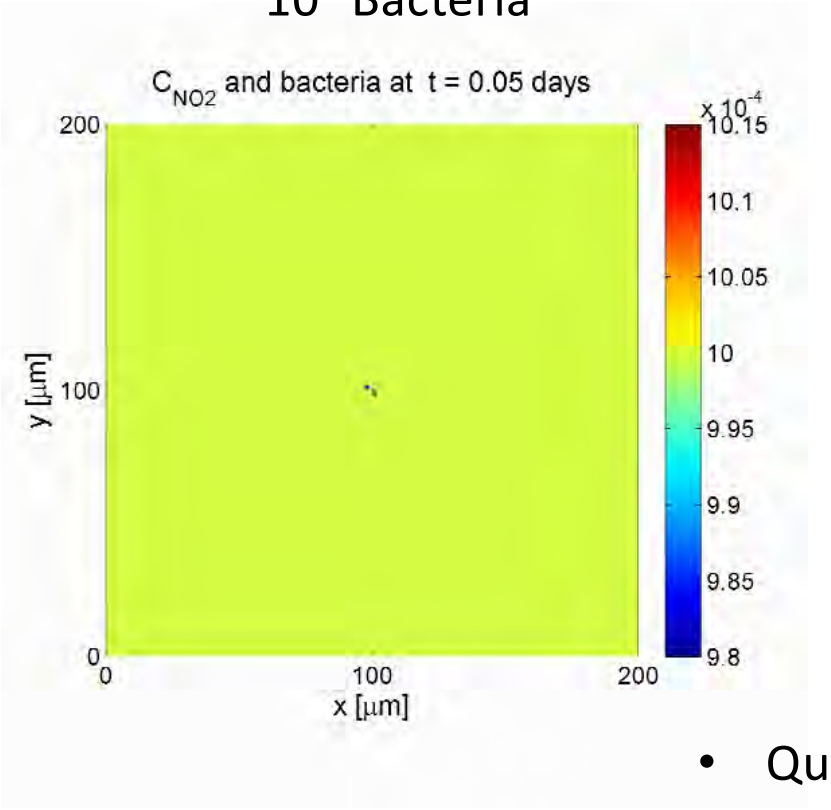


Update Growth rates wrt Nutrient



Mapping this to the Real World

10^4 Bacteria



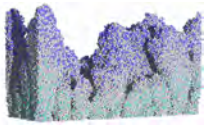
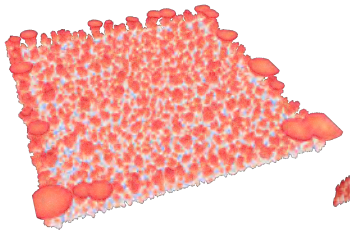
Ofiteru 2014

10^{18} Bacteria

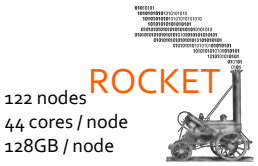
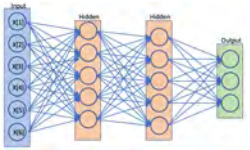
- Quality of the parameters in the model / the model
- Emergent properties as we scale up

Scaling up

frontiers in engineering biology

Under development

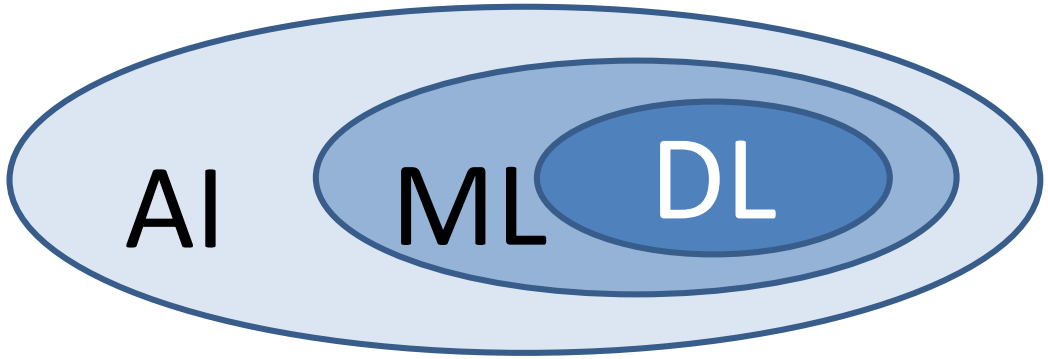

Volume	μm³	~100μm³	mm³	cm³	dm³
Number of Bacteria	10^4	10^6	10^8	10^{10}	10^{12+}
Simulation time	days	10's of days	100's of days	years	decades
Runtime	hours	1 day	4 days	week	week
Hardware			 122 nodes 44 cores / node 128GB / node ROCKET		
Software	MATLAB	LAMMPS	LAMMPS + MPI	LAMMPS + KOKKOS	PyTorch / TensorFlow
Purpose	Proof of concept				Emergent Properties Comparison with real world

Outline

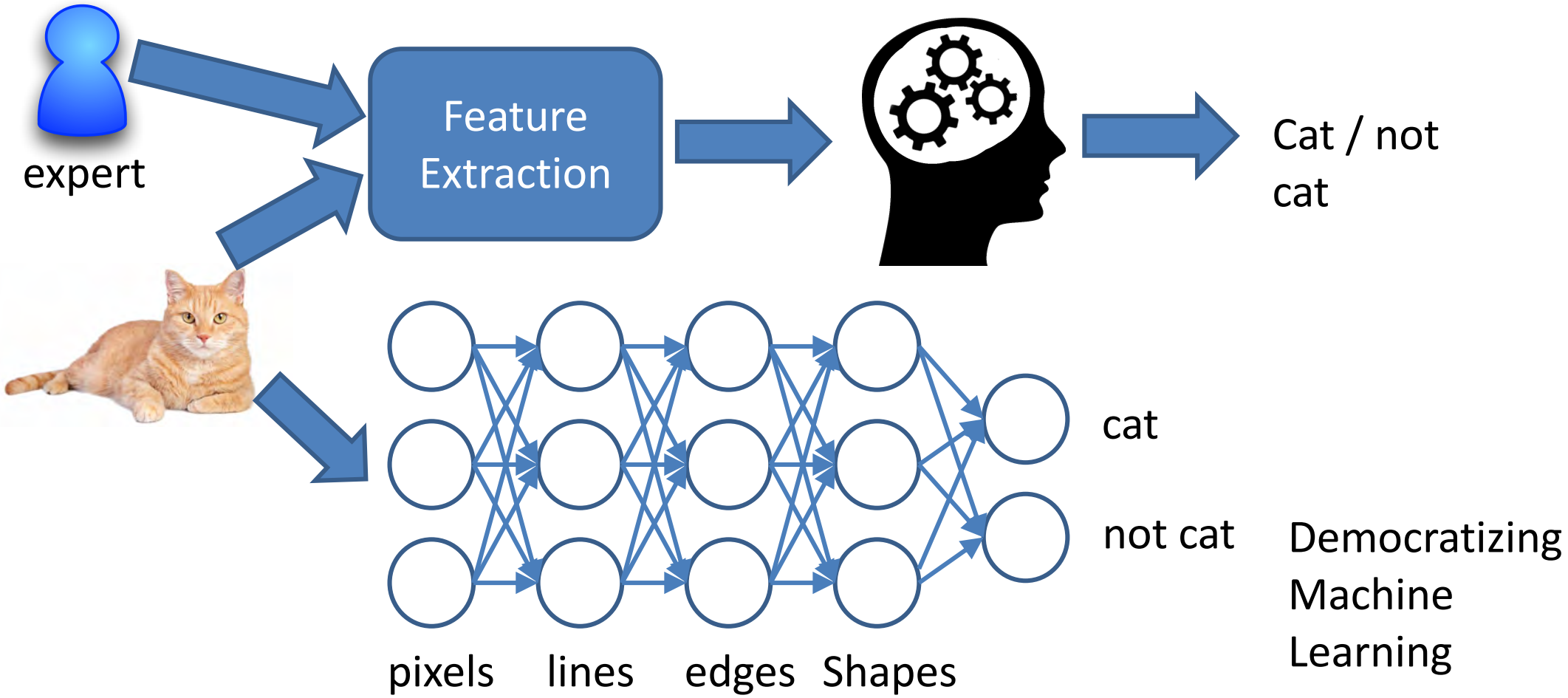
- Wastewater Treatment Primer
- Simulating wastewater treatment
- **AI Primer**
- Using AI for wastewater treatment

What is AI, ML and DL?

- Artificial Intelligence (AI)
  - A system which exhibits characteristics which could be seen as intelligent
- Machine Learning (ML)
  - A system which is able to learn and improve its ability
- Deep Learning (DL)
  - A system which uses (Deep) Neural Networks to exhibit ML



Machine Learning vs Deep Learning



Basic building blocks:
The data

- Data is key here: Sample as 1D array of values

X[1]

X[2]

X[3]

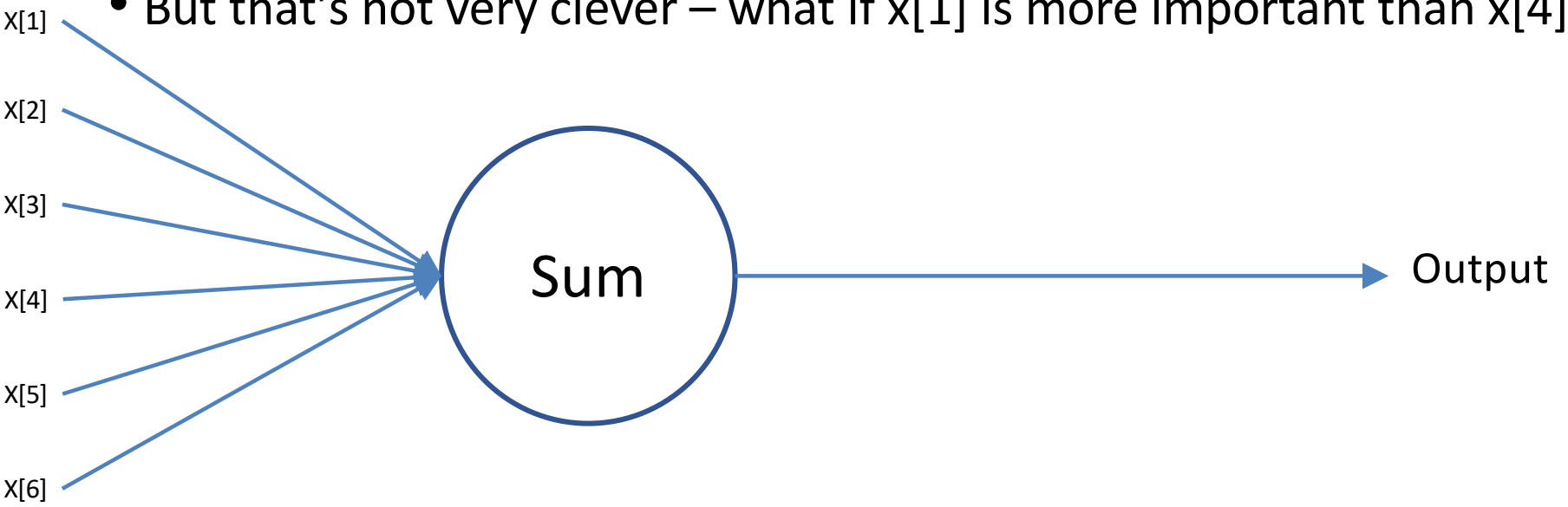
X[4]

X[5]

X[6]

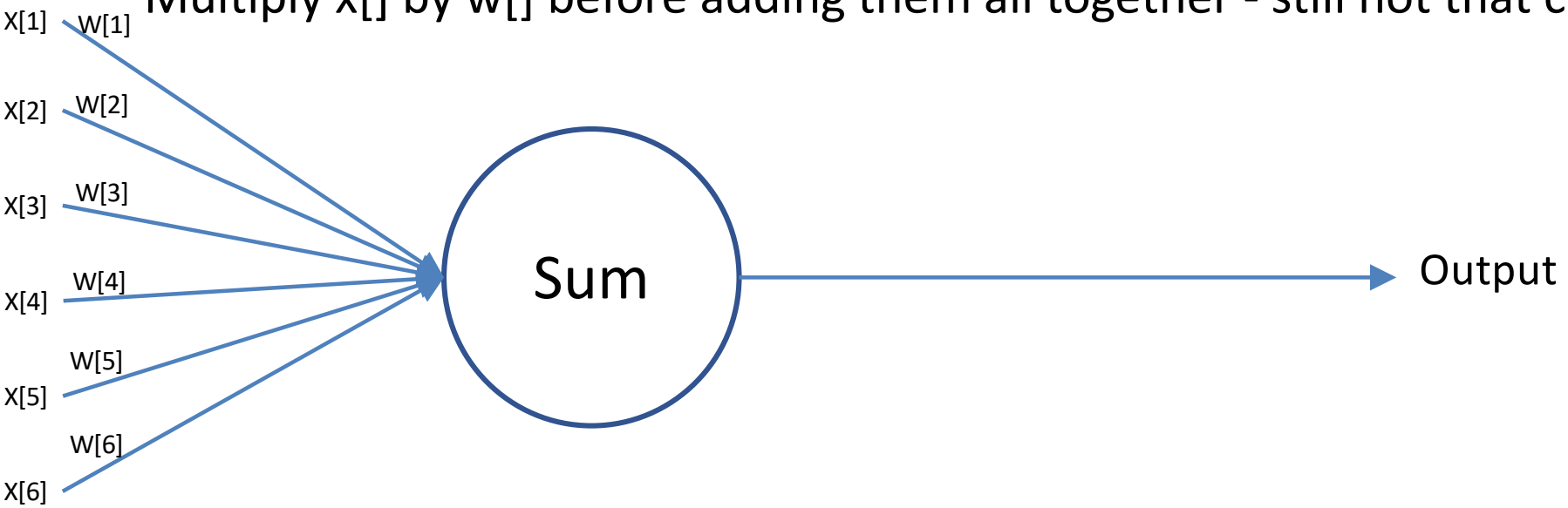
Basic building blocks: The neuron

- Sums up all of the input values
  - But that's not very clever – what if x[1] is more important than x[4]?



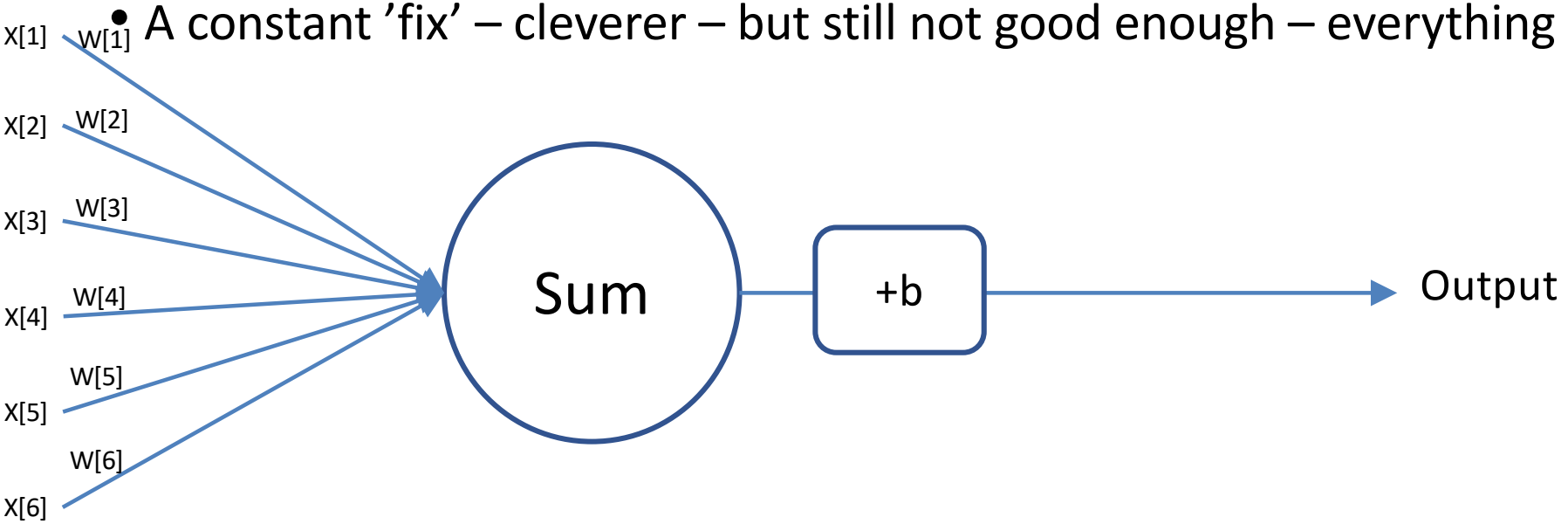
Basic building blocks: Weights

- The weights – so we can attribute importance to each x[]
  - Multiply x[] by w[] before adding them all together - still not that clever



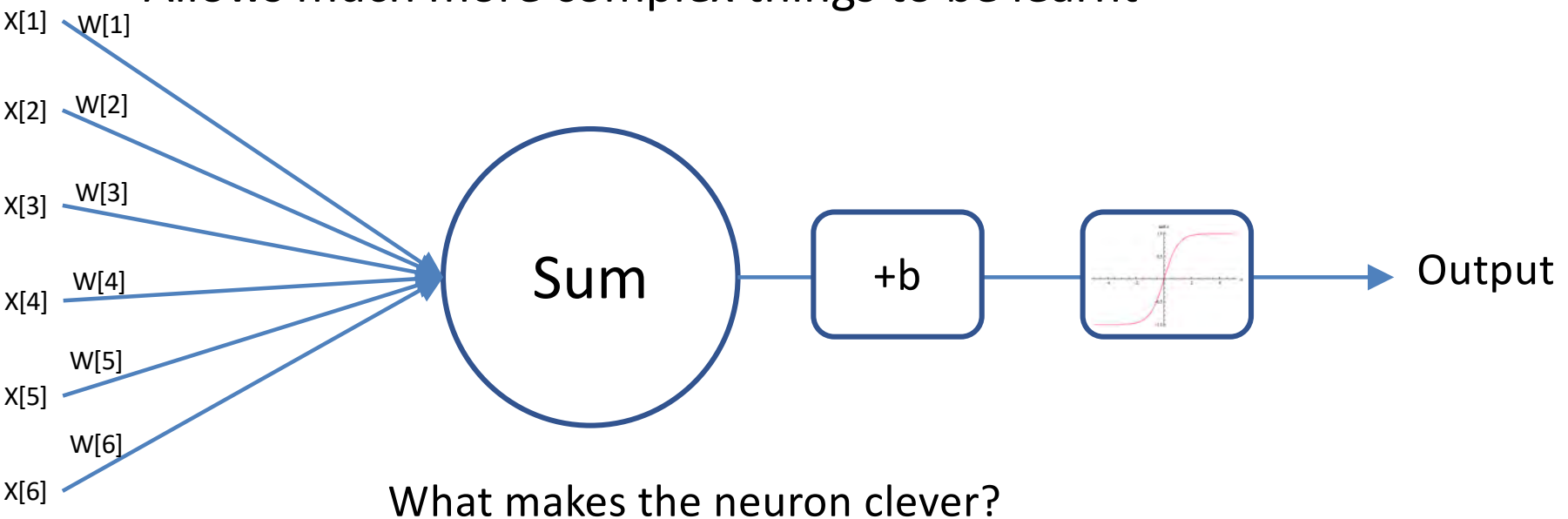
Basic building blocks:
Bias

- The bias is a value we add to the output
  - A constant ’fix’ – cleverer – but still not good enough – everything is liner



Basic building blocks: Activation function

- Activation function - non-linear operation applied to output
  - Allows much more complex things to be learnt

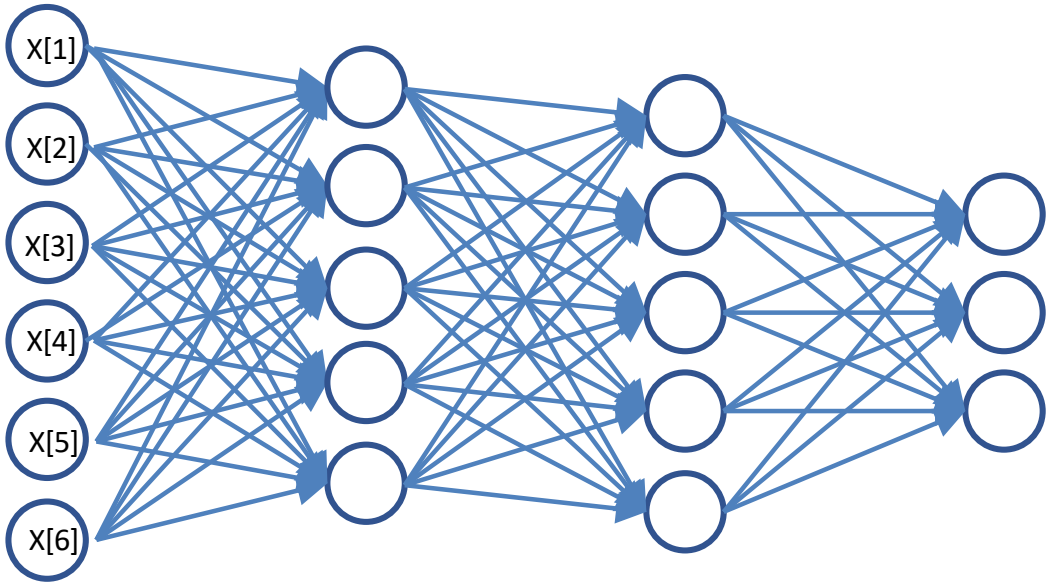


What makes the neuron clever?

- The right values of w[], b
- Trained by passing lots of examples through and modifying these values

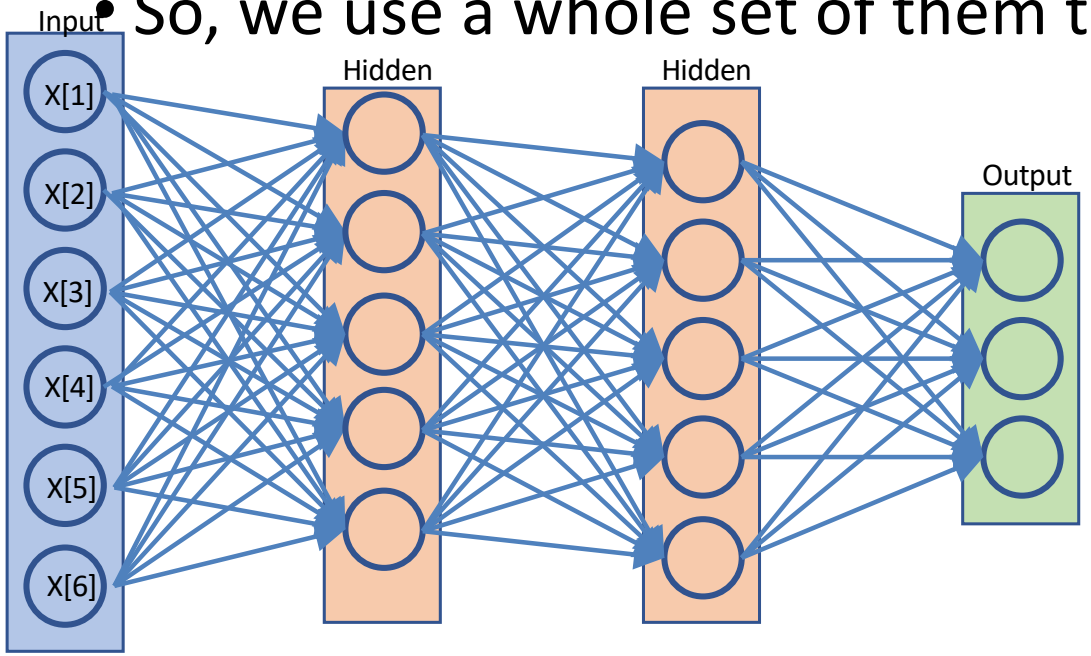
Building a full Neural Network

- A single block on its own can’t do much
  - So, we use a whole set of them to make a neural network



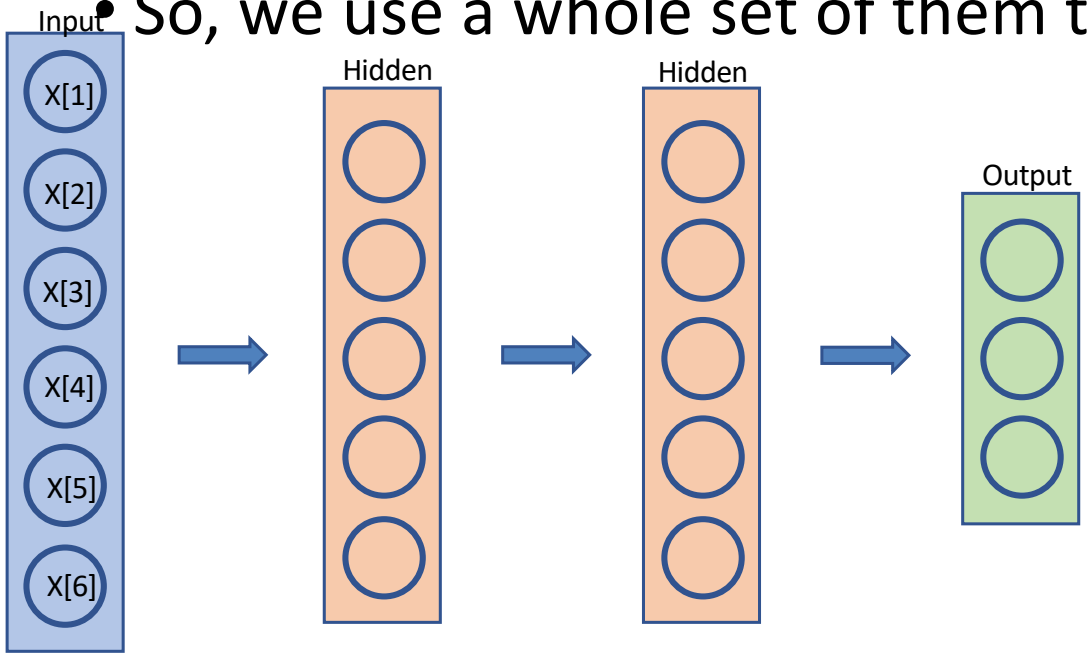
Building a full Neural Network

- A single block on its own can't do much
- So, we use a whole set of them to make a neural network



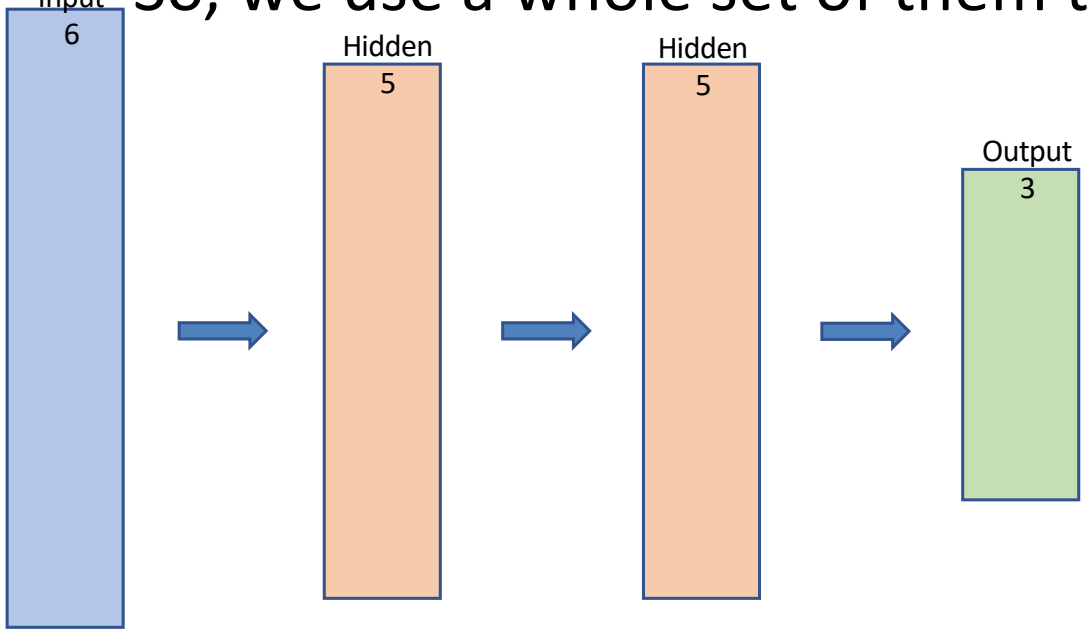
Building a full Neural Network

- A single block on its own can't do much
- So, we use a whole set of them to make a neural network



Building a full Neural Network

- A single block on its own can't do much
- So, we use a whole set of them to make a neural network

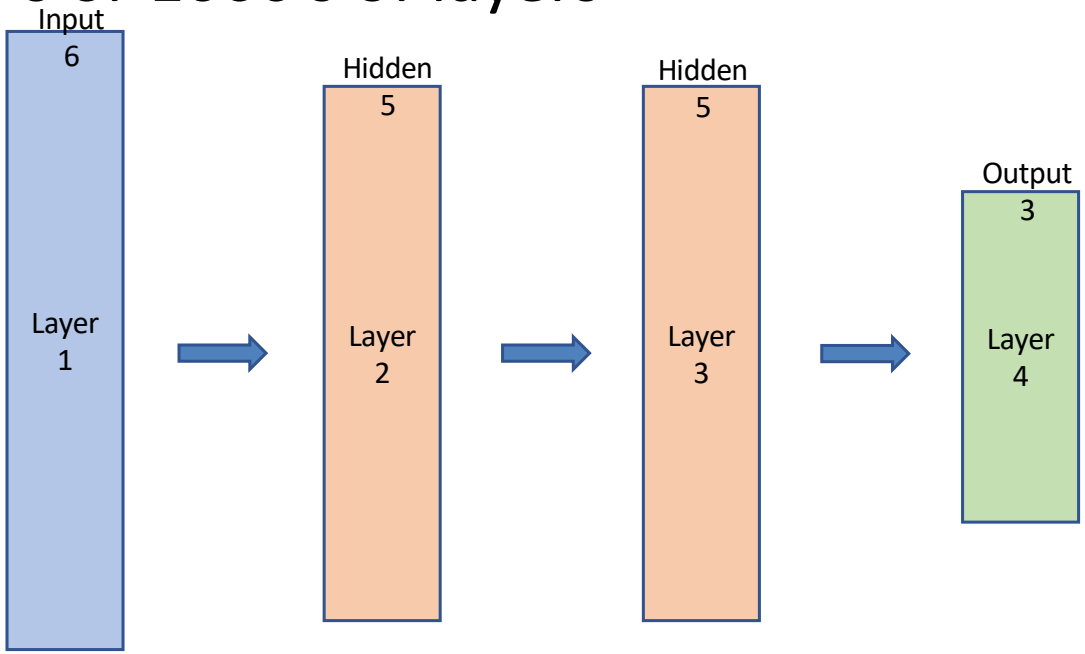


Called:

- Multi-Layer Perceptron (MLP)
- Fully Connected Layers
- Dense layers

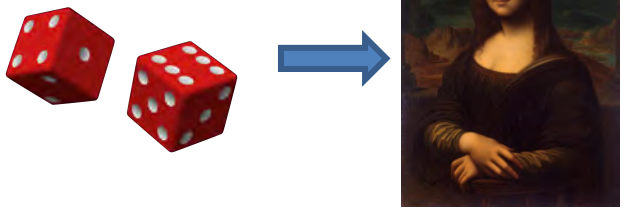
So Why Deep Learning?

- Named due to the depth of the network
- Number of layers in the network
- ‘Real’ networks have 10’s, 100’s or 1000’s of layers
- These networks are often referred to as models
- Why now?
  - Data is available
  - Powerful (GPUs) to train them



Generative Adversarial Network (GAN)

- Main aim: generate fake samples from some input domain that are as close to the real data as possible. E.g., random input -> Italian Renaissance portraits



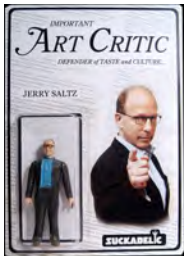
- Needs two components:

Generator

- Generates fake samples
- Tries to make the samples as 'real' as possible to fool the discriminator

Discriminator

- Identifies if a sample is fake
- Tries to identify if a sample is from the real set or a fake from the generator



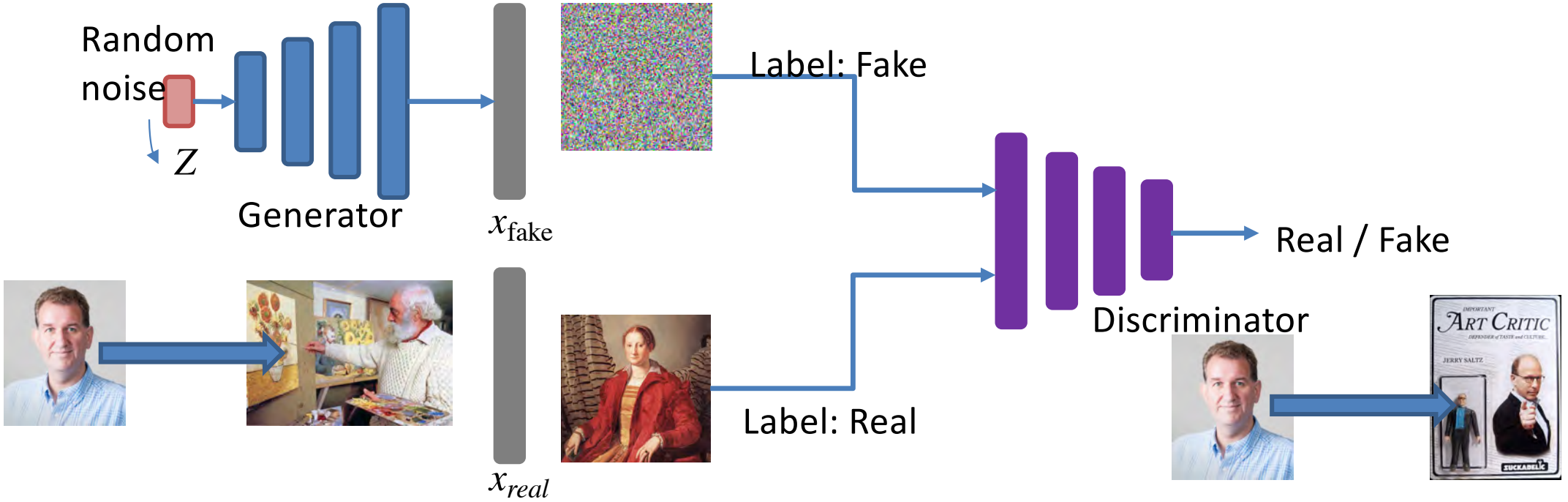
Generative Adversarial Network (GAN)

Generator

- Generates fake samples
- Forger (e.g. of art)

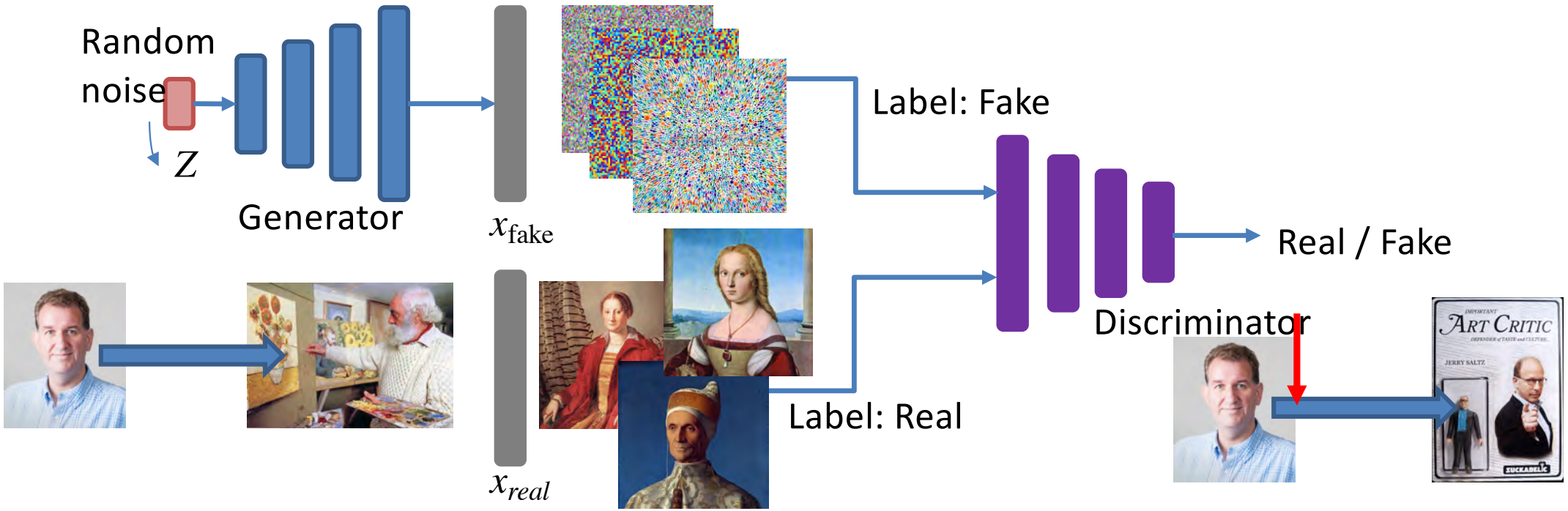
Discriminator

- Identifies if a sample is fake
- E.g., art critic



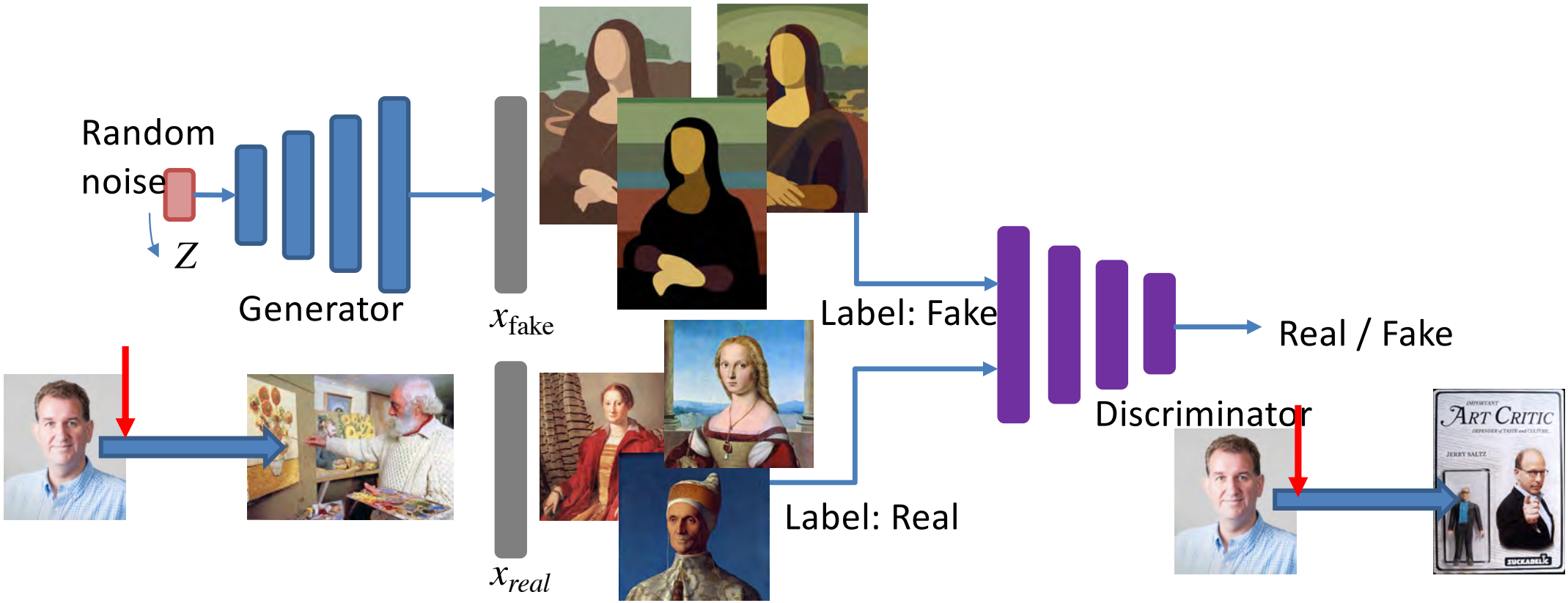
Generative Adversarial Network (GAN)

- Iteratively train **discriminator** and then generator



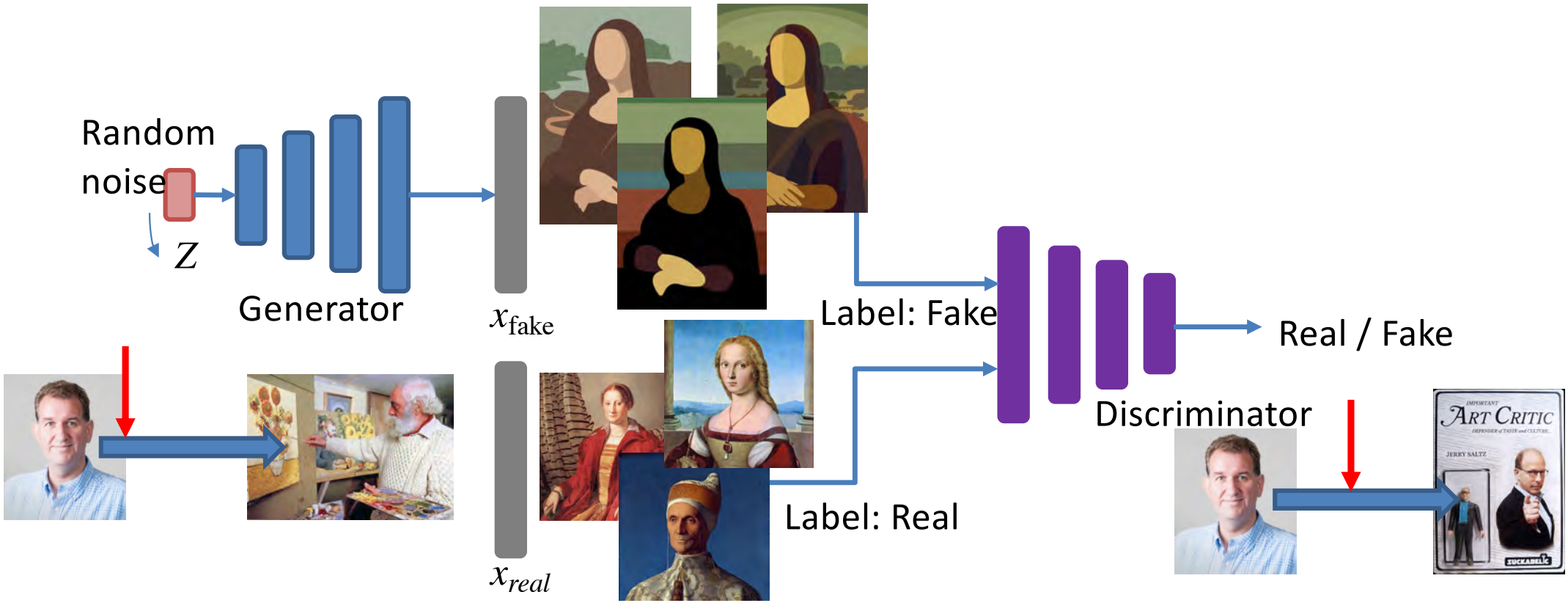
Generative Adversarial Network (GAN)

- Iteratively train discriminator and then **generator**



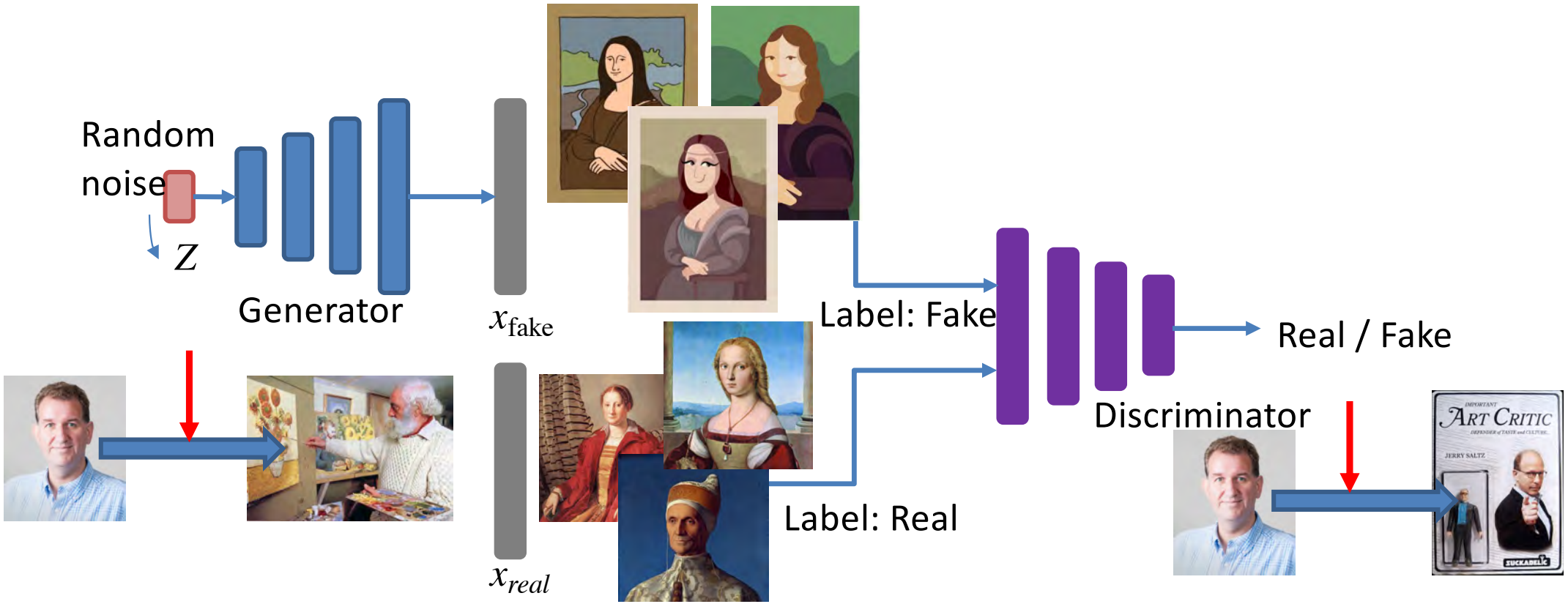
Generative Adversarial Network (GAN)

- Iteratively train **discriminator** and then generator



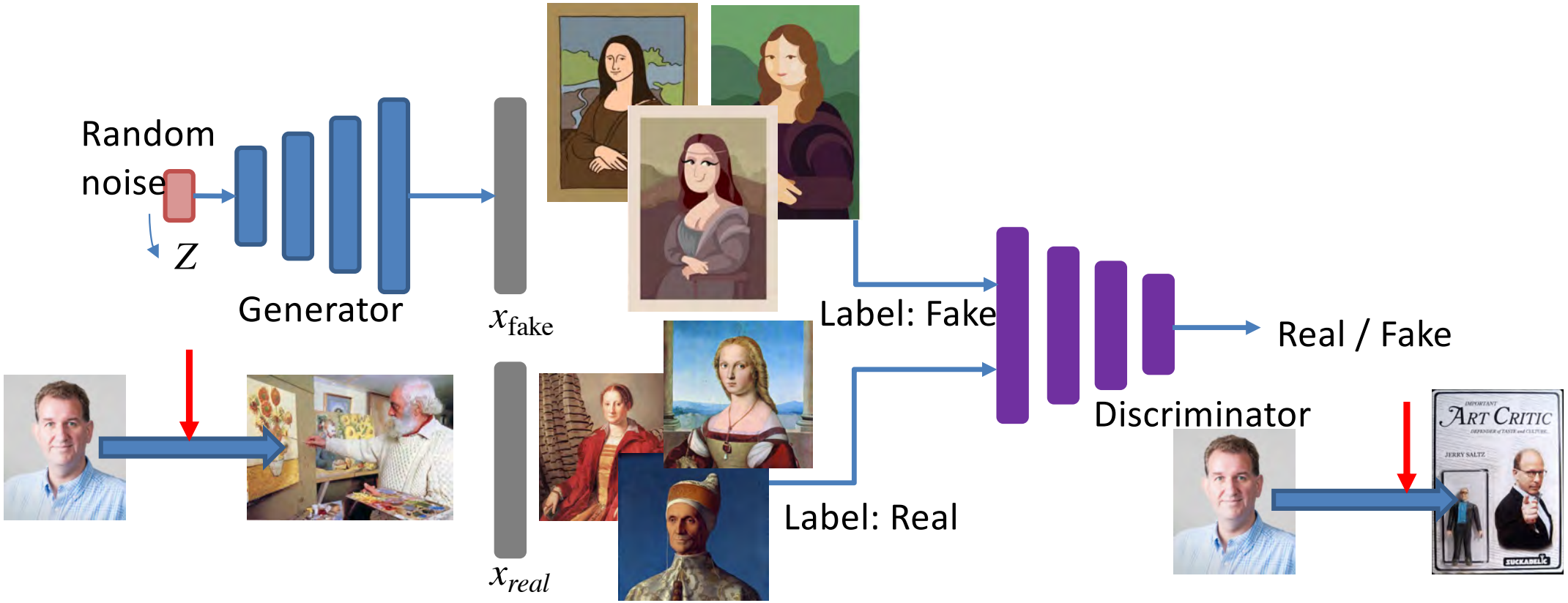
Generative Adversarial Network (GAN)

- Iteratively train discriminator and then **generator**



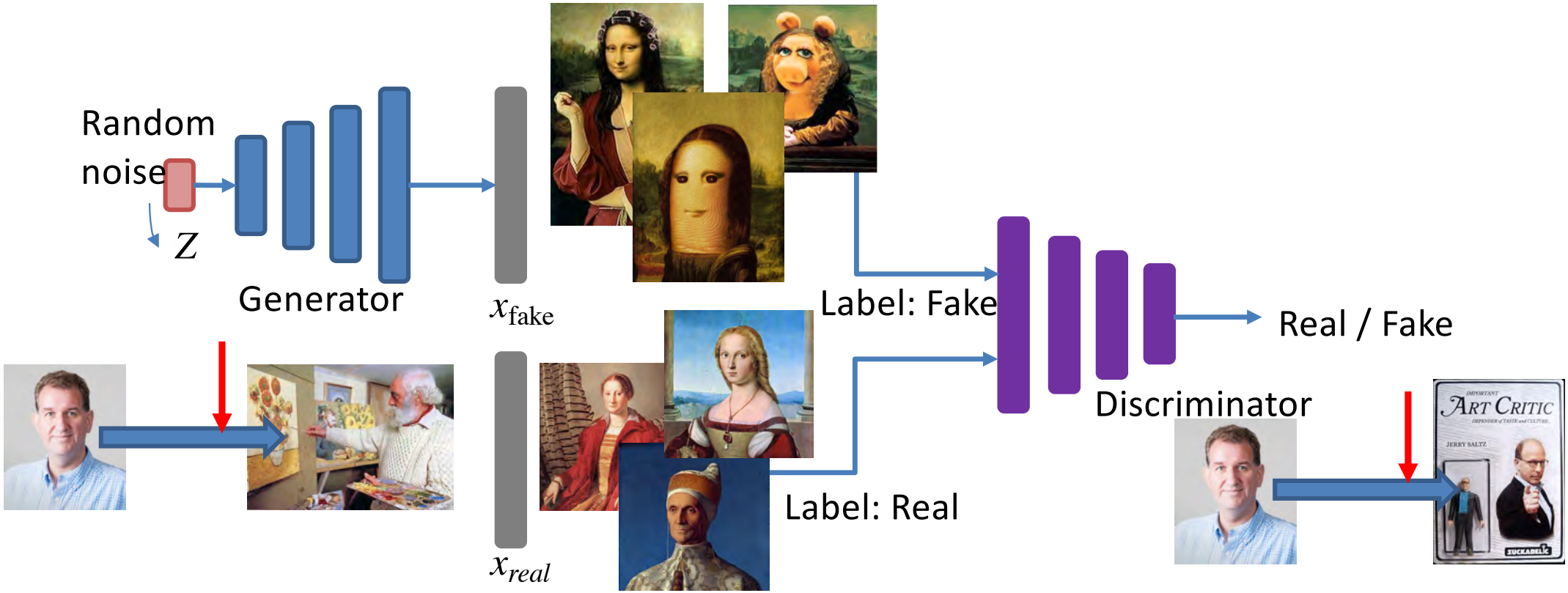
Generative Adversarial Network (GAN)

- Iteratively train **discriminator** and then generator



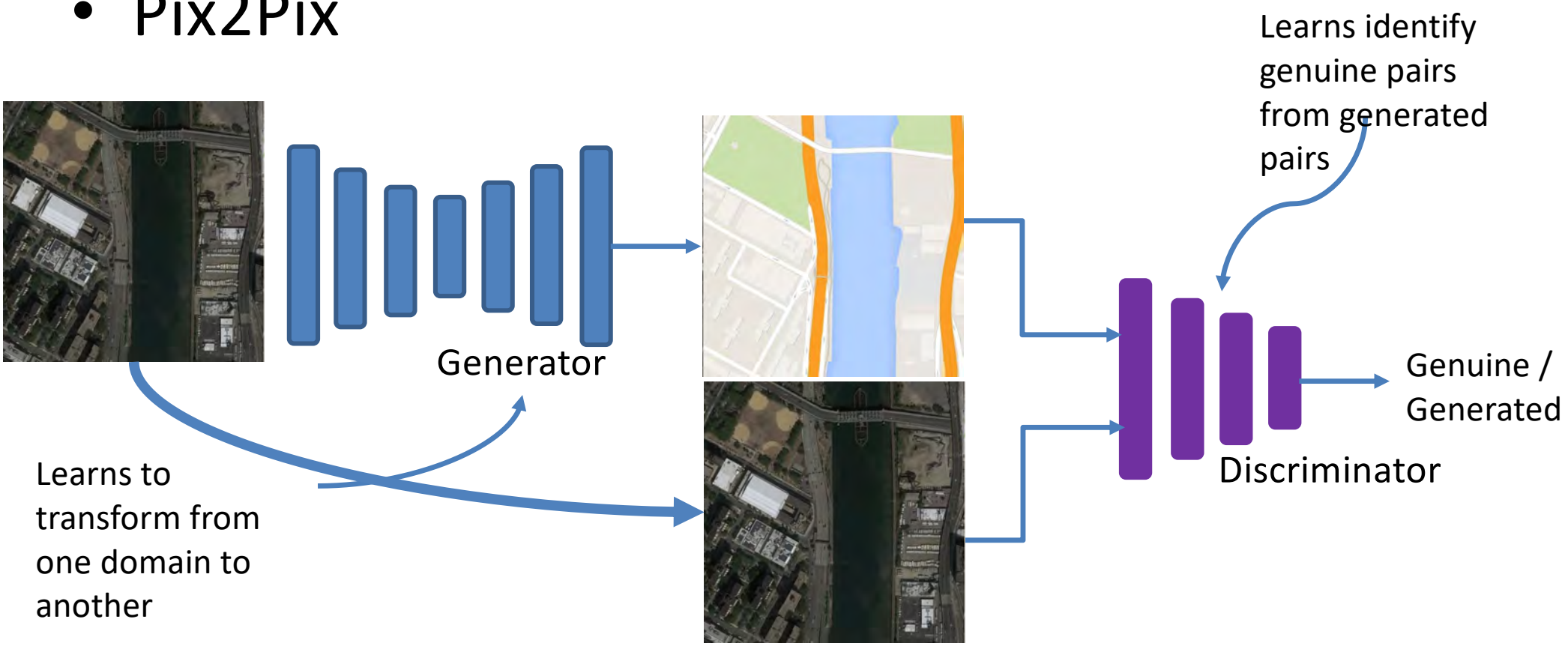
Generative Adversarial Network (GAN)

- Iteratively train discriminator and then **generator**



Conditional GAN

- Pix2Pix



Outline

- Wastewater Treatment Primer
- Simulating wastewater treatment
- AI Primer
- **Using AI for wastewater treatment**
  - **Scaling up**
  - Fine Tuning the simulation

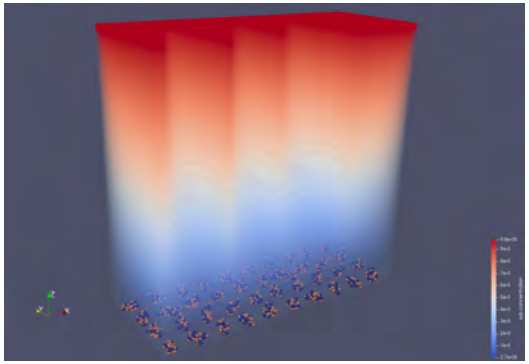
Building a Deep Learning Emulator

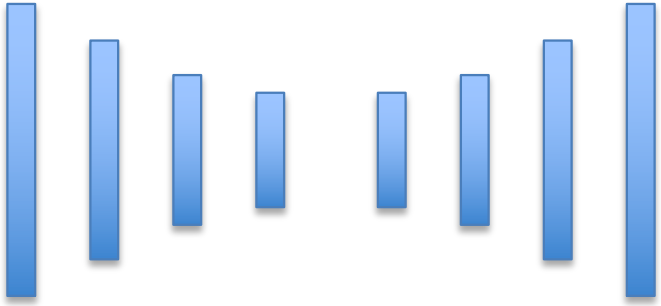
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

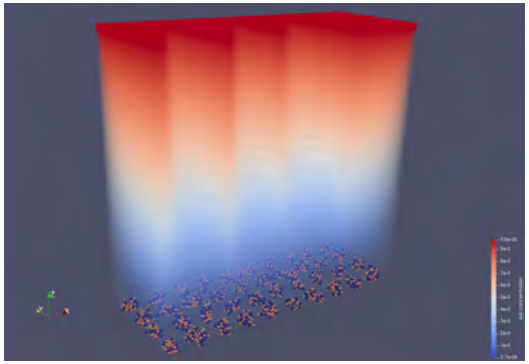
  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN



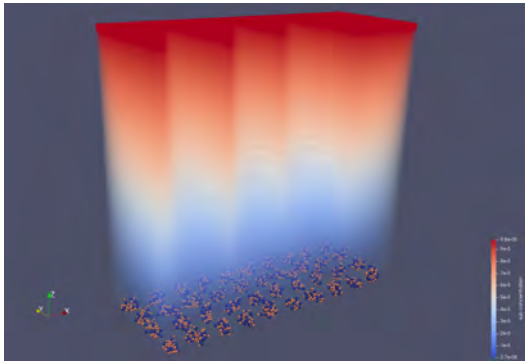
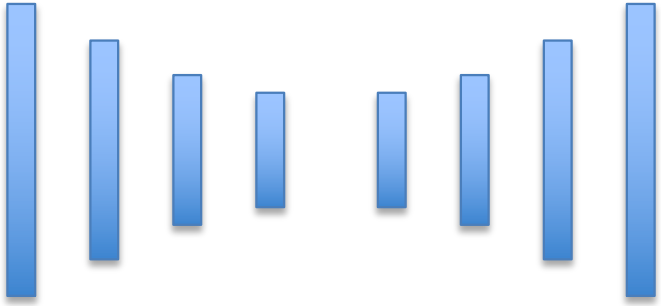
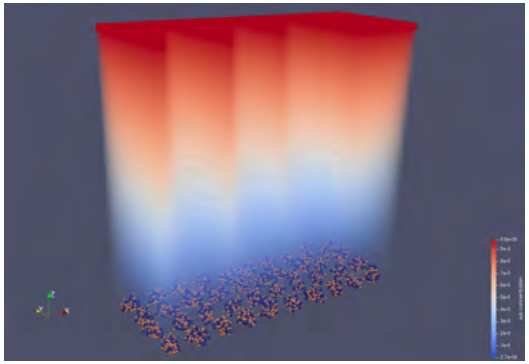




  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

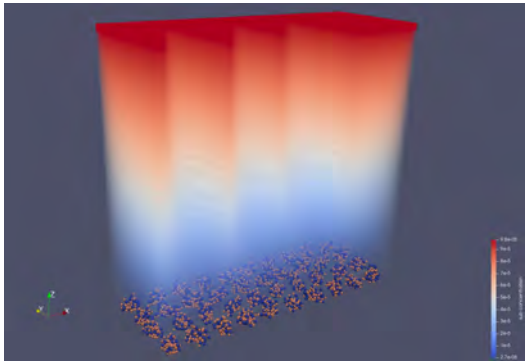
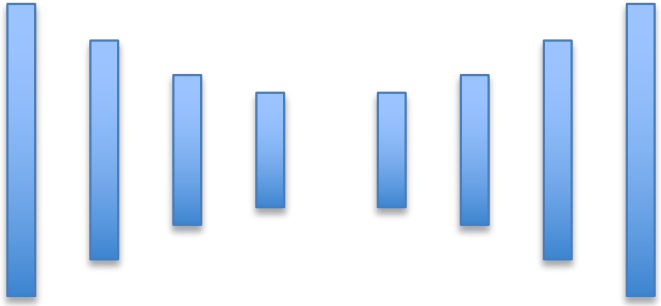
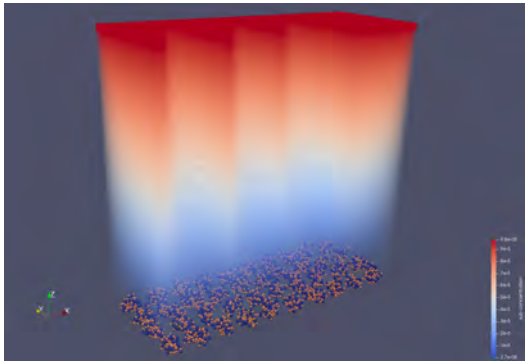
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

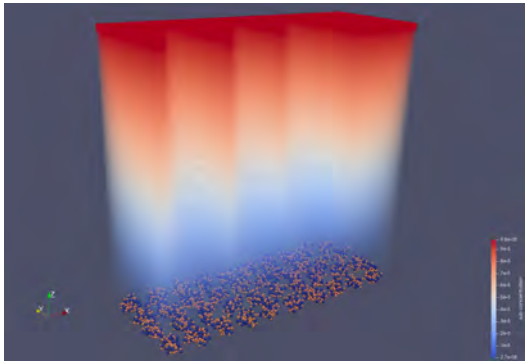
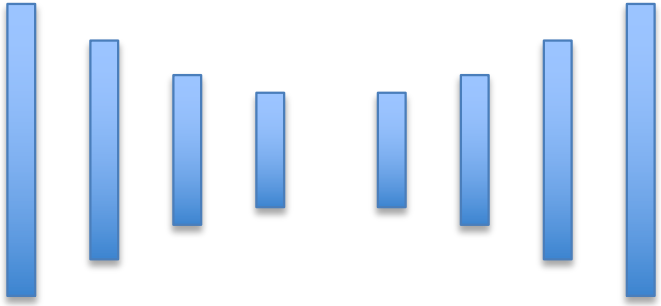
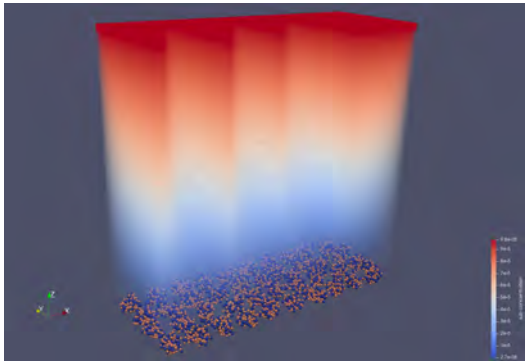
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

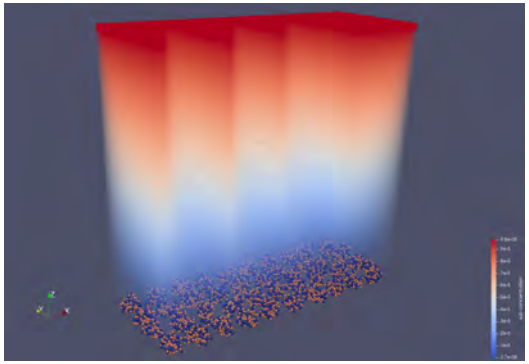
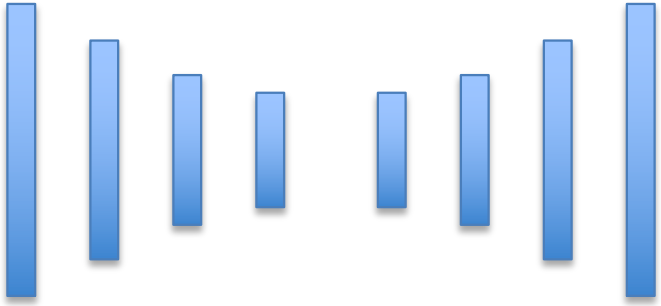
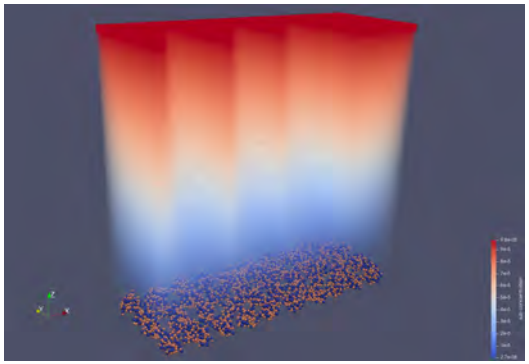
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

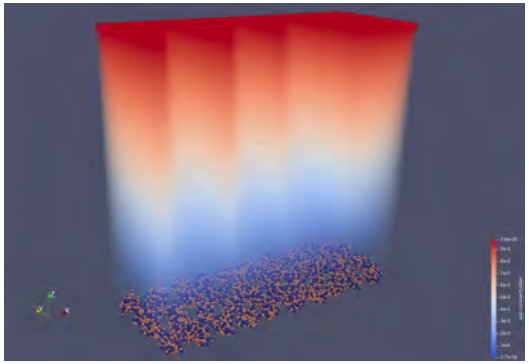
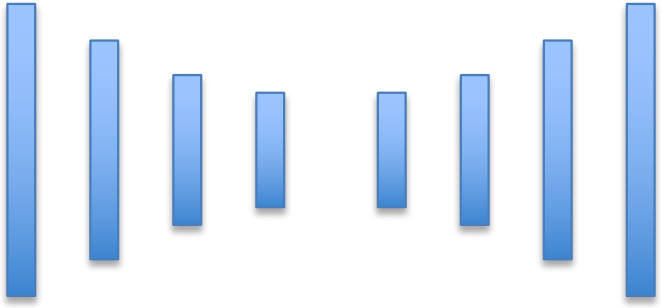
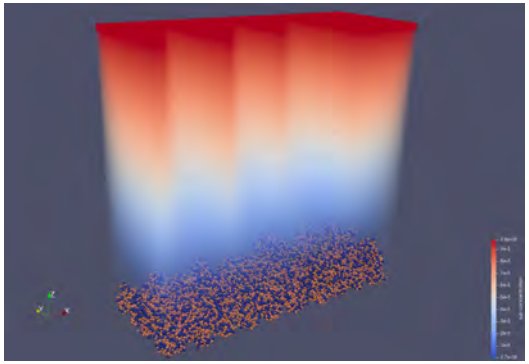
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

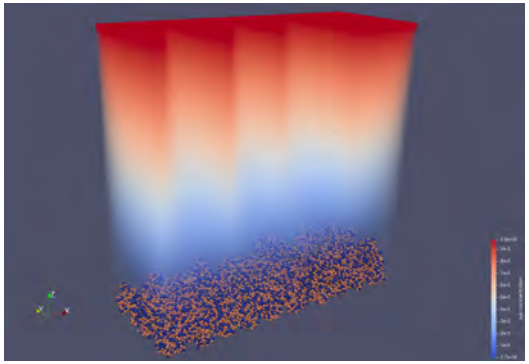
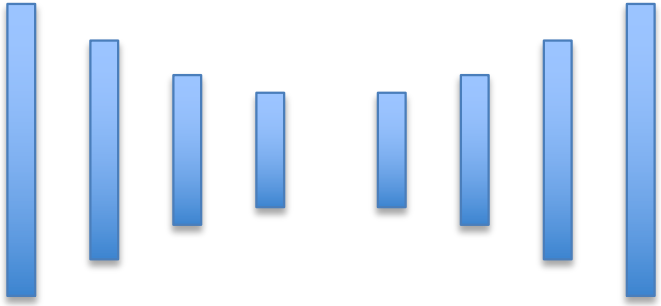
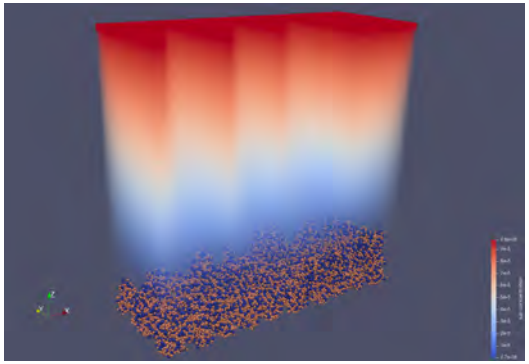
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

Building a Deep Learning Emulator

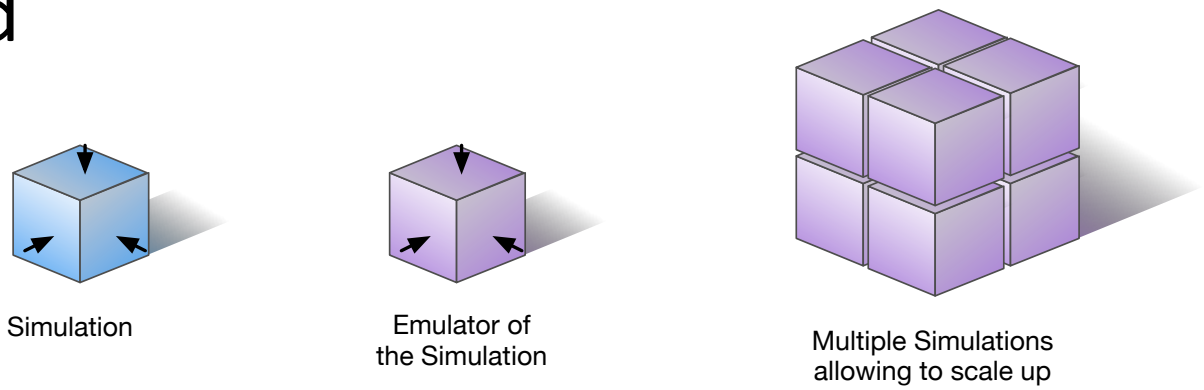
- Predict next step using Deep Learning
  - Autoencoder, GAN, RNN

  - Done for large enough volume s.t.
    sim_time >> prediction_time

How to use this to scale up

- Focus DL emulator on the Outside edges of the volume
- Can then 3D ‘tile’ volumes together
  - Nontrivial – requires massive DL Emulator, well trained



Outline

- Wastewater Treatment Primer
- Simulating wastewater treatment
- AI Primer
- **Using AI for wastewater treatment**
  - Scaling up
  - **Fine Tuning the simulation**

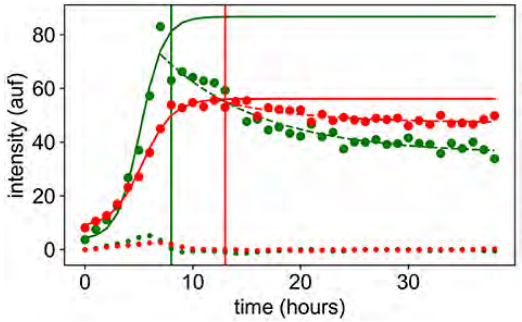
Why might the simulation need tuning?

- Simulations are ‘best guesses’ as to how a system works
  - Parameters often based on results from papers/ books
  - Model is based on our understanding of how the system works (often from books)
  - Can we match the output of the simulation to the real world?

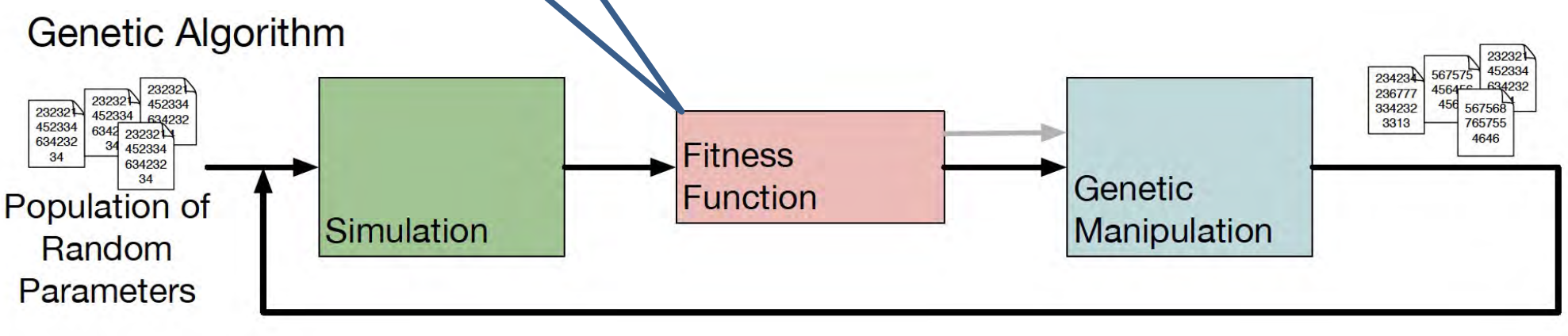
Start with a fairly simple Genetic Algorithm approach

$$\frac{A}{1 + e^{(\frac{4\mu}{A}(\tau - t) + 2)}}$$

A, μ and τ



Genetic Algorithm



Hand-Crafted Fitness

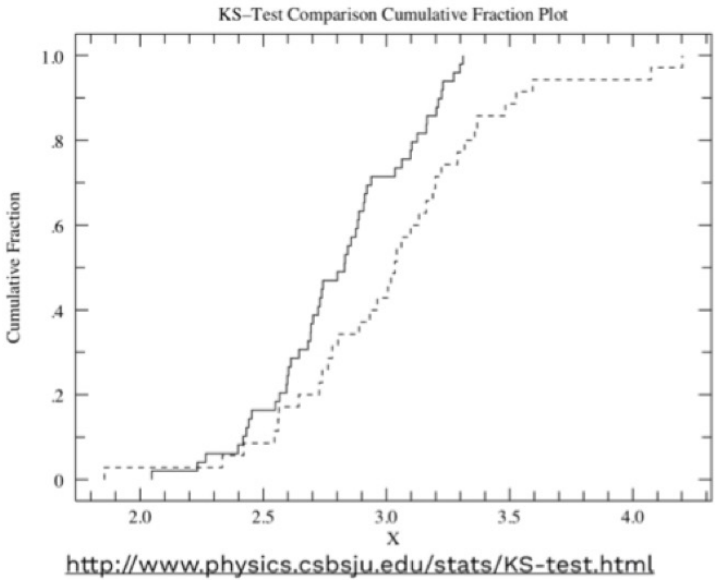
- Fitness function defined by comparing experimental and simulation data
- Multiple runs of simulation
- From each simulation compute A, μ and τ
- Compute empirical CDFs
- Compare with same for experiments

Let F and G be empirical CSFs for simulation / experiment data

$$L(F,G) = \int_{-\infty}^{\infty} |F(x) - G(x)|\, dx$$

To obtain a fitness function f from a loss L

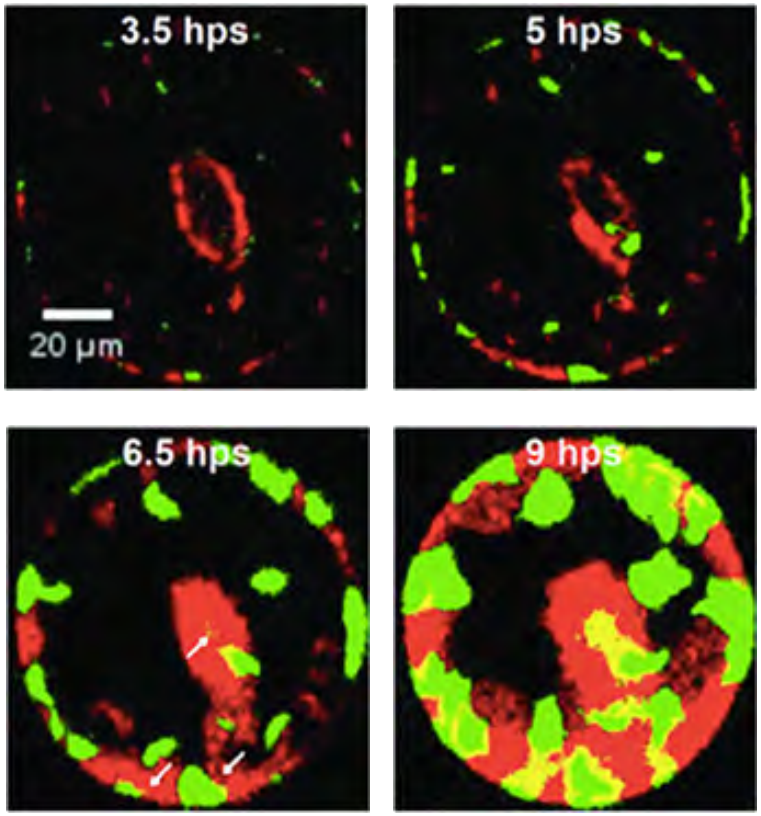
$$f = \frac{1}{0.1 + L}$$



http://www.physics.csbsju.edu/stats/KS-test.html

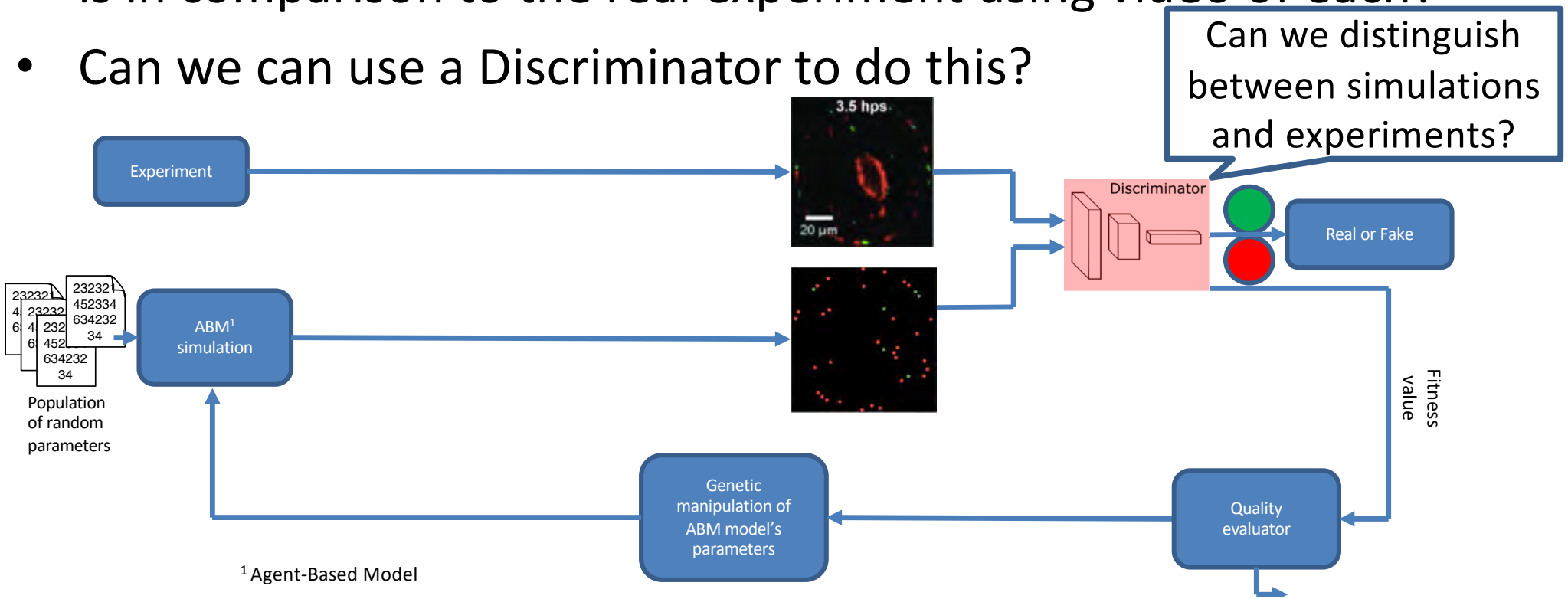
Brain storm – what else is there?

- The number of regions of each bacteria
- Their shape
- Their relative locations
- Are they touching?
- How these things change over time
- …



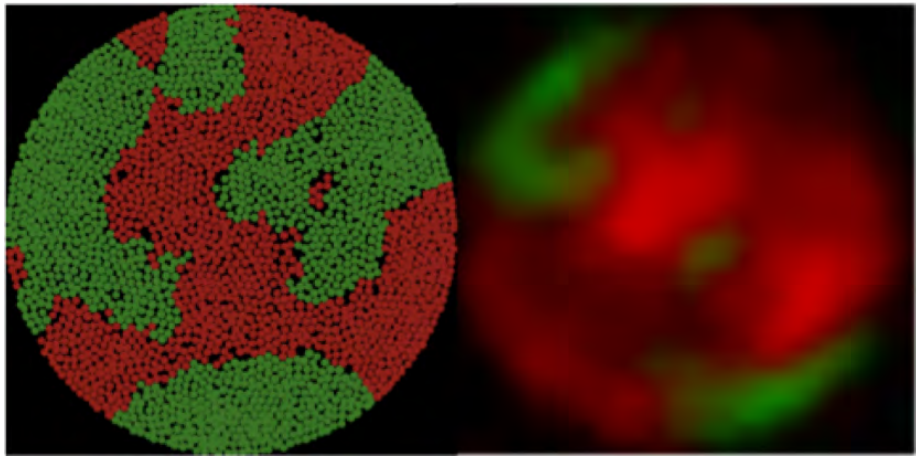
Deep Learning says ‘don’t do feature extraction’

- Can we get Deep Learning to tell us how good our simulation is in comparison to the real experiment using video of each?
- Can we can use a Discriminator to do this?



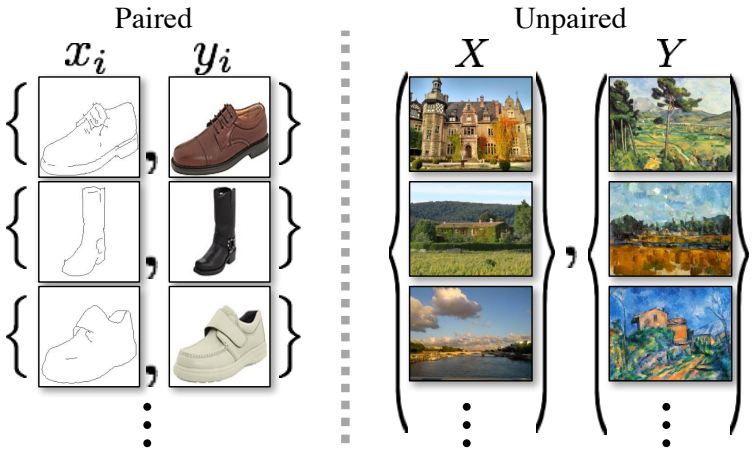
This will fail: Simulation looks nothing like Experiment

- Simulation is nice crisp and clean
- Fluorescence of tightly packed bacteria

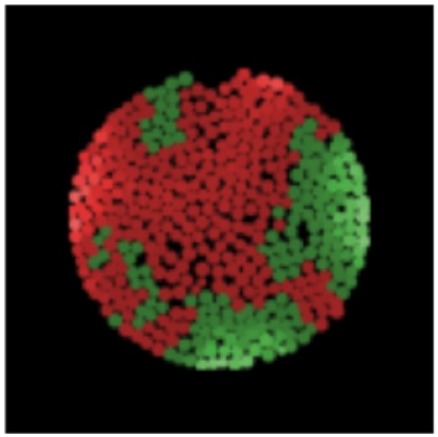


Making Simulations look more 'Real'

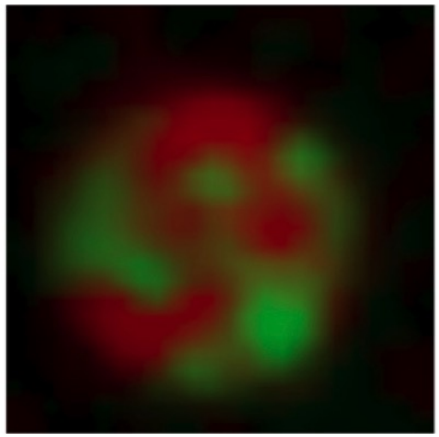
- Style Transfer
- Using CycleGAN
- No need for paired images



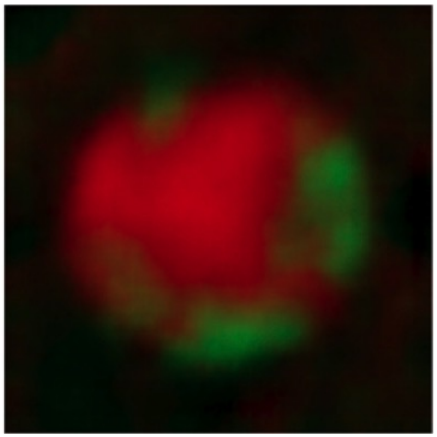
Input (simulation)



Source (experiment)

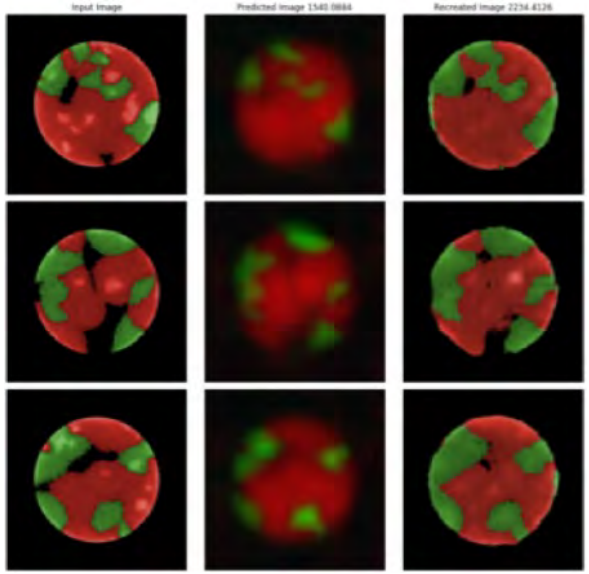


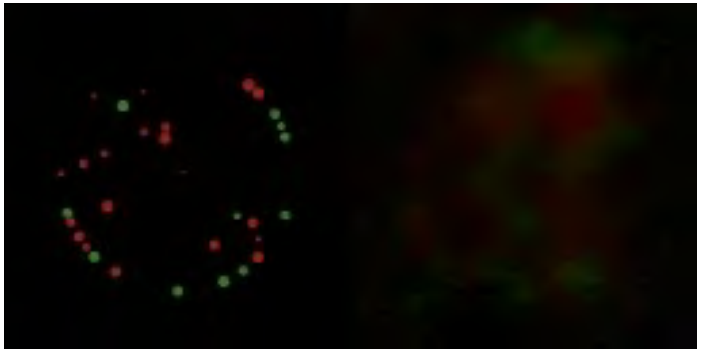
Output (experiment-like sim)



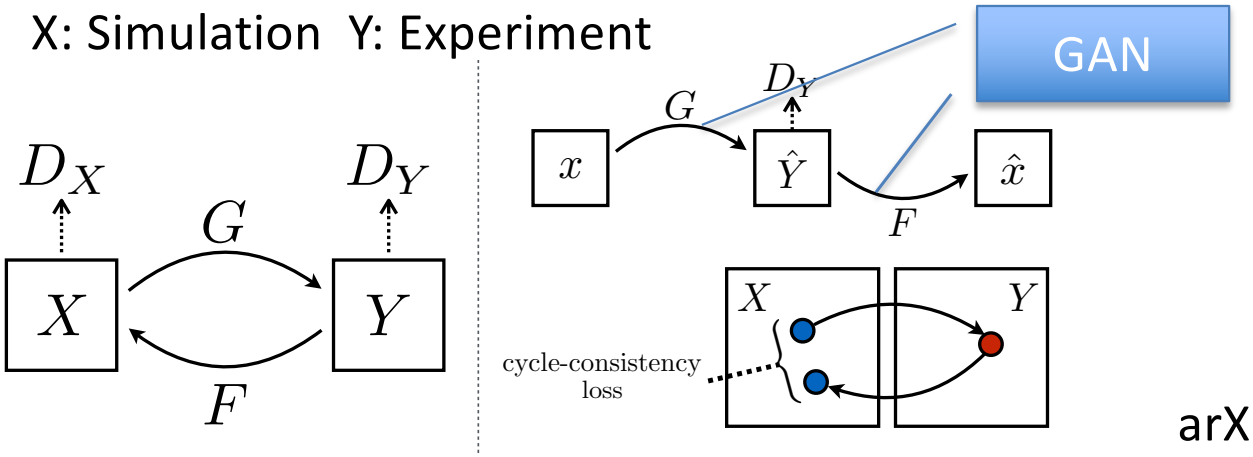
Style transfer of simulation data

- Unpaired Image-to-Image Translation
  - CycleGAN
- Pre-trained on images that resemble the distribution of the experiments
- Pad images to match size of experiment



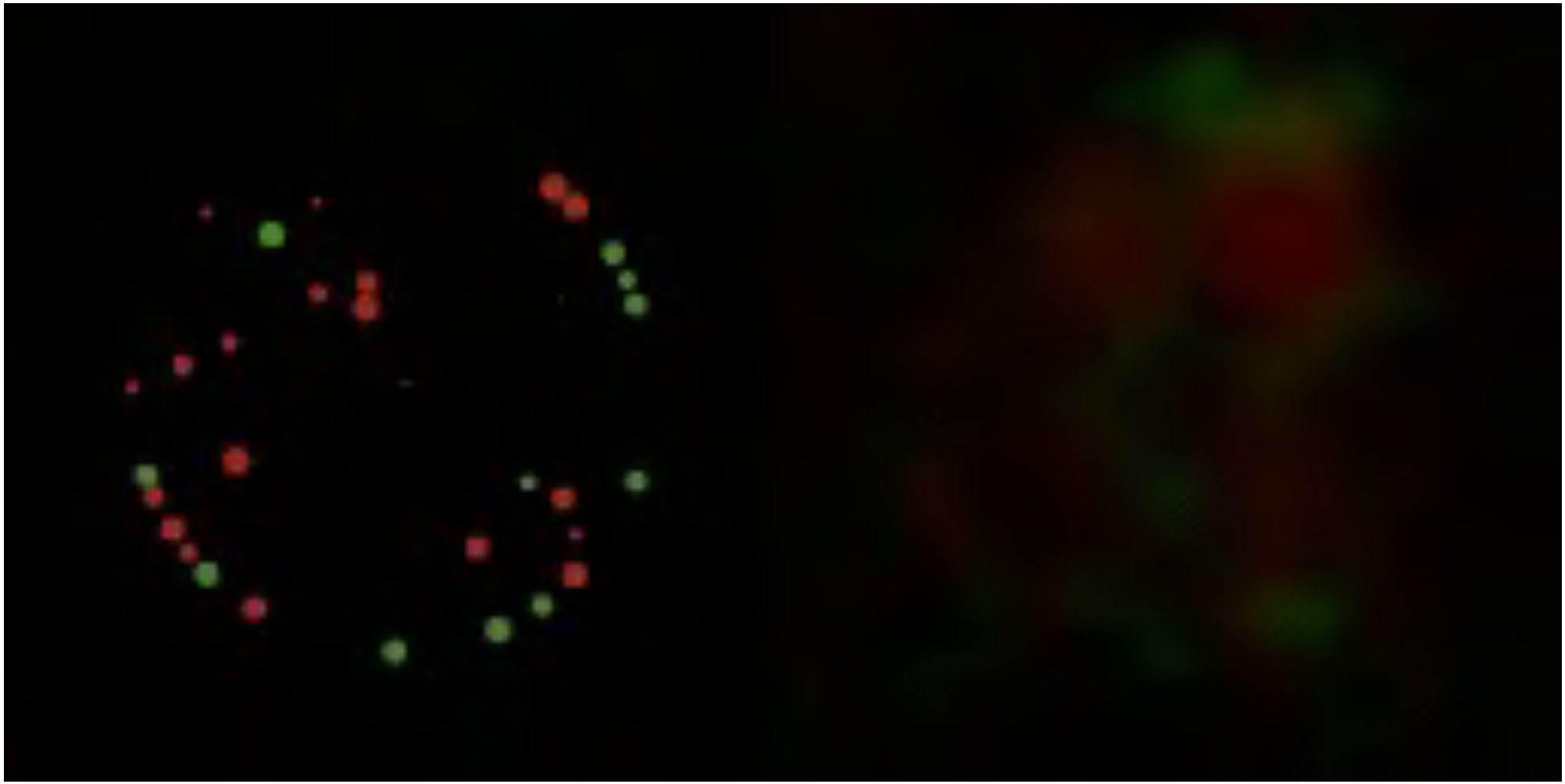


X: Simulation Y: Experiment

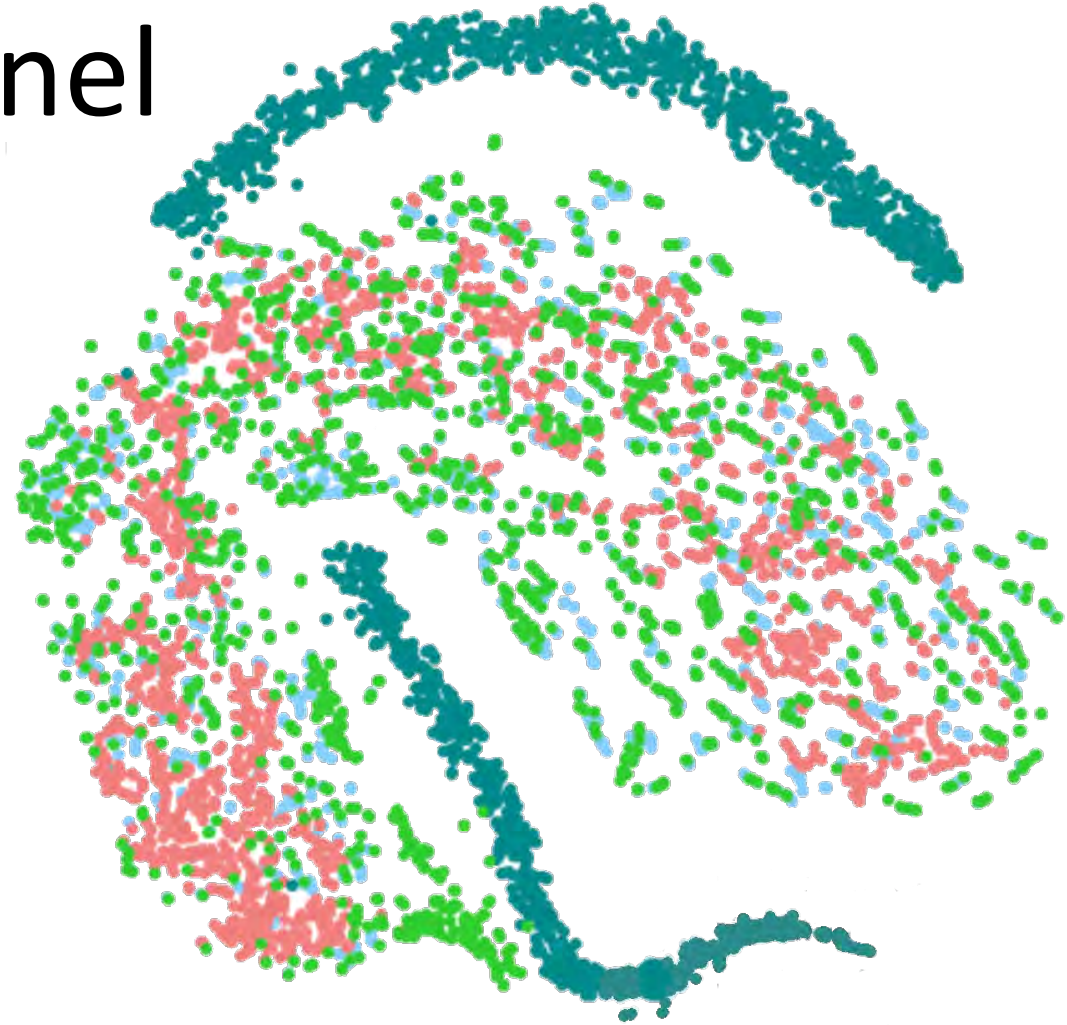


arXiv:1703.10593v6 [cs.CV] 15 Nov 2018

The finished simulation



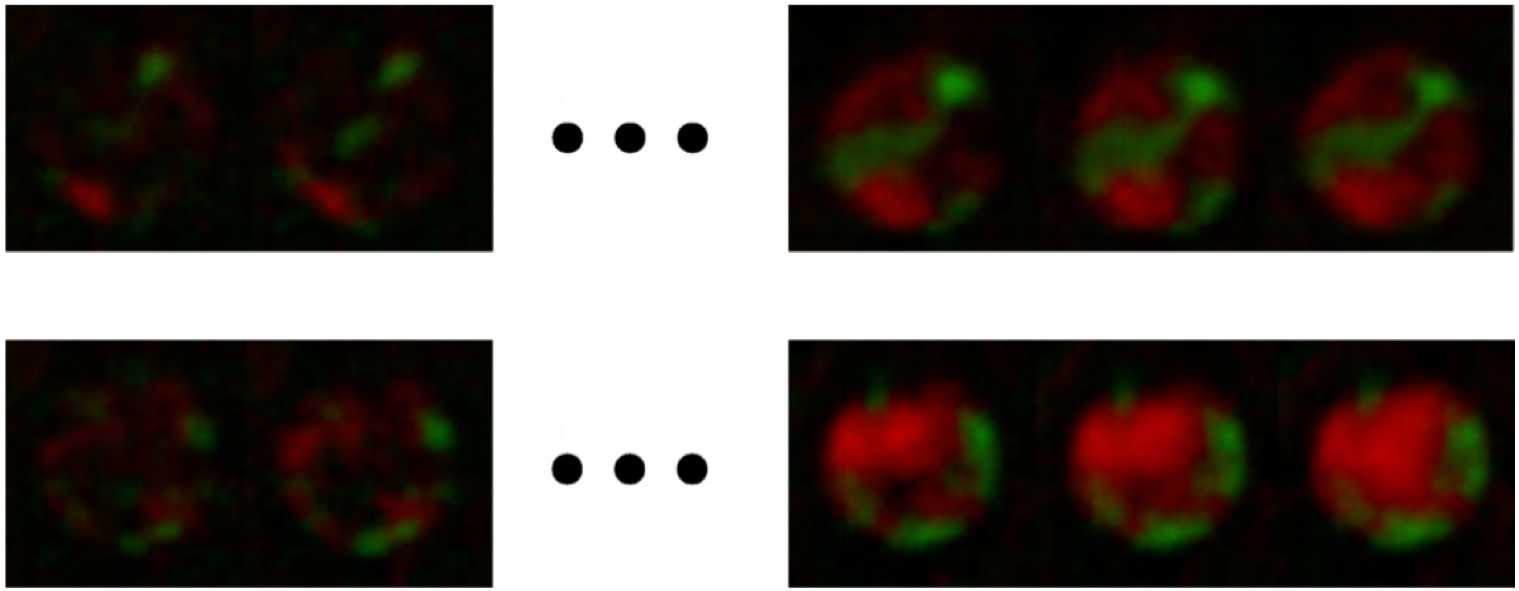
T-SNE Liner Kernel



- Experiment
- Experiment Artifact
- Simulation
- Simulation Style Transfer

Discriminator

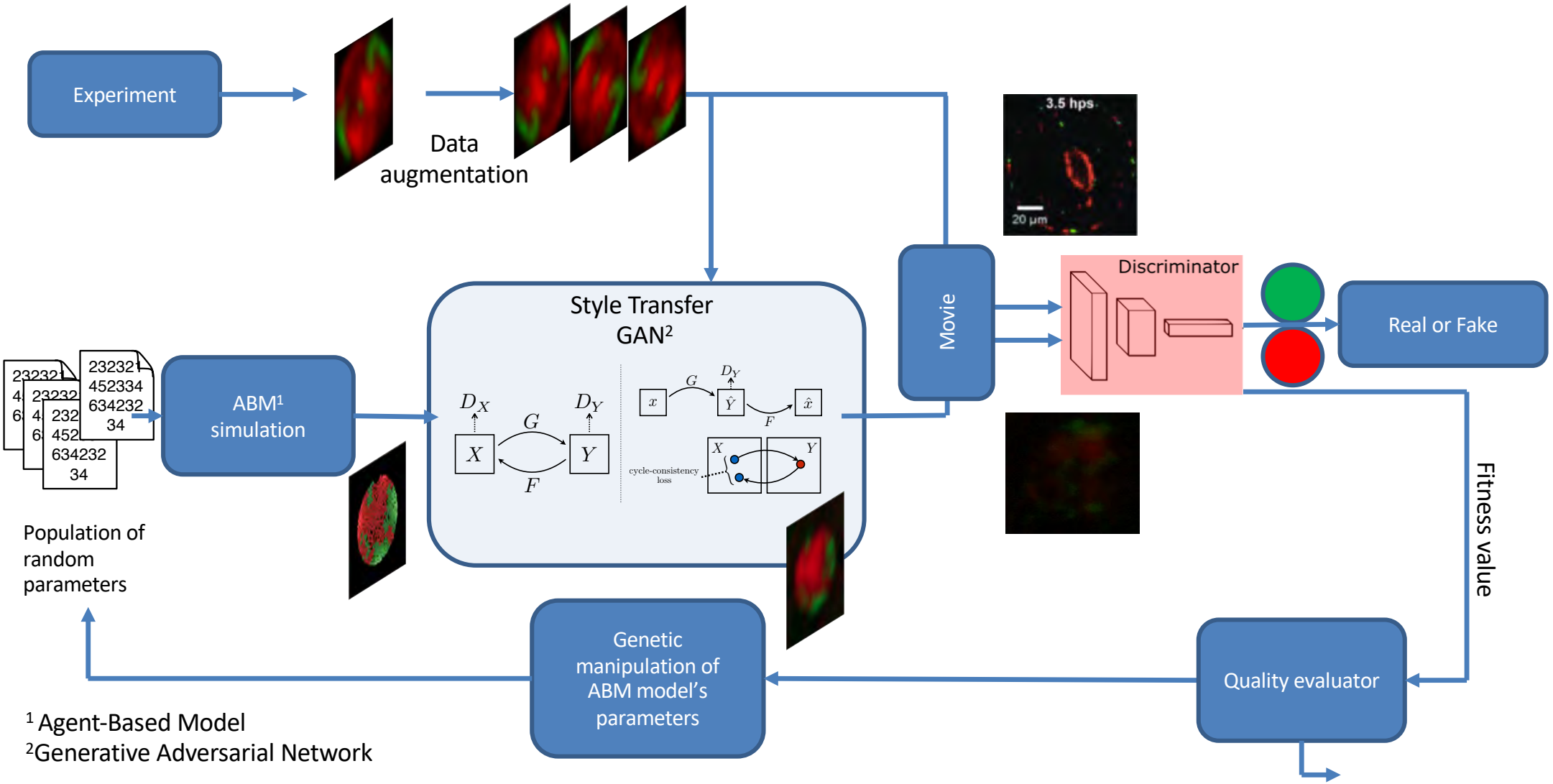
Input: 15 images of the growth phase



Experiment

Simulation

AI Framework for Creating Accurate Agent-Based Models of Microbial Populations



[1] Agent-Based Model
[2] Generative Adversarial Network

Stephen.McGough@Newcastle.ac.uk

Summary

- To make better simulations we need:
  - Larger Scale -> observe emergent properties
  - More accurate simulations -> fine-tune
- Larger simulations
  - Scale up with emulators
- More accurate simulations
  - Tune parameters / agents to experiments

Many thanks to: Denis Taniguchi, Miguel Fuentes-Cabrera, Daniel Elbrecht, Gavin Glenn, Ryn Gray, Grace Kim, Austin Li, Lino Valdovinos and the NUFEB team

UKRI Engineering and Physical Sciences Research Council
OAK RIDGE National Laboratory
CENTER FOR NANOPHASE MATERIALS SCIENCES