Going Beyond, Scaling and Tuning Microbial Simulations towards Real-world Systems

Stephen McGough
With thanks to: Denis Taniguchi, Daniel Elbrecht, Gavin Glenn, Ryn Gray, Grace Kim, Austin Li, Lino Valdovinos and Miguel Fuentes-Cabrera

Friday 14th May 2021

Wastewater Zero for Urban Sustainability and Health, Virtual

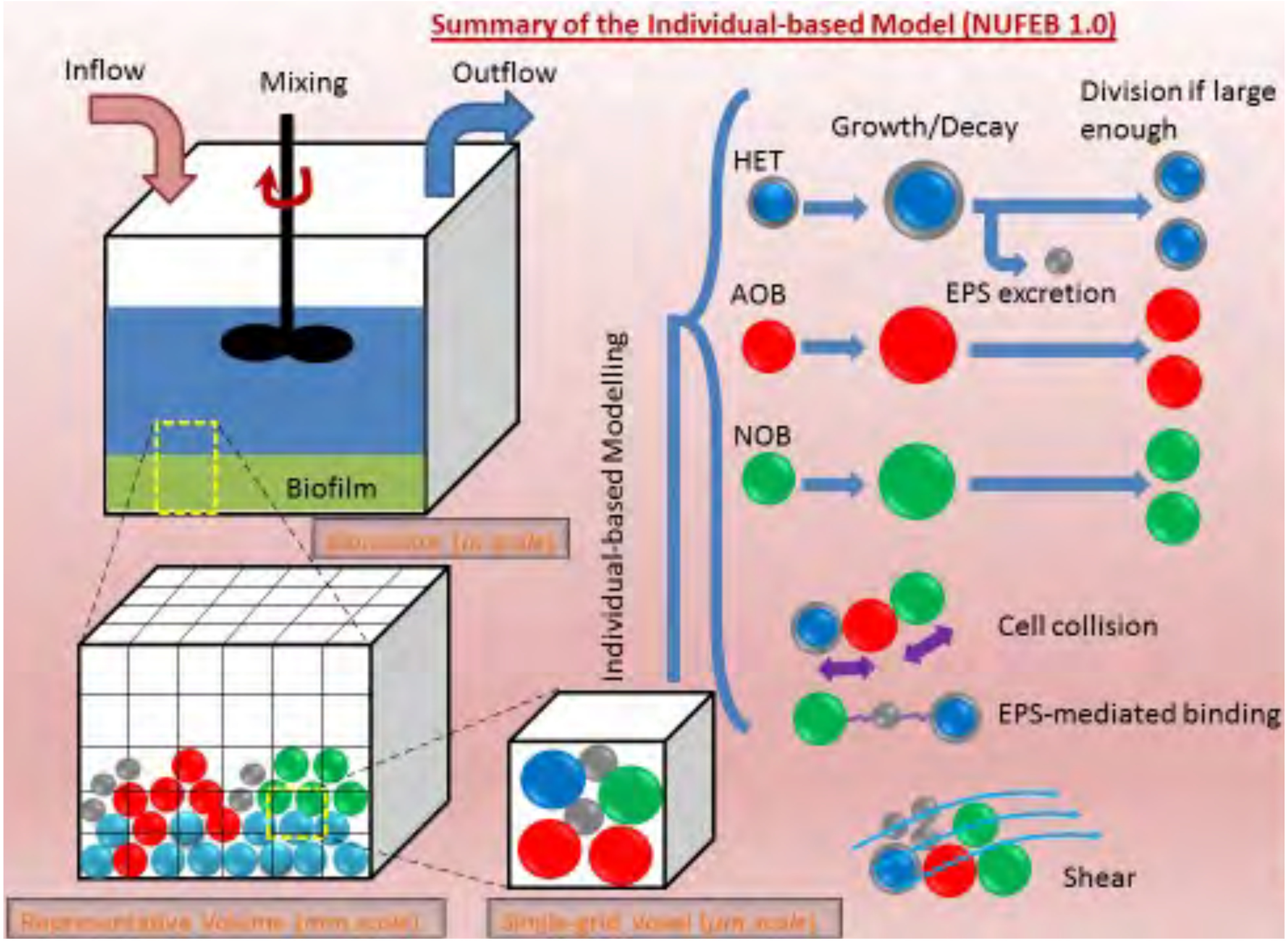
Model Overview

AOB - Ammonia Oxidizer Bacteria

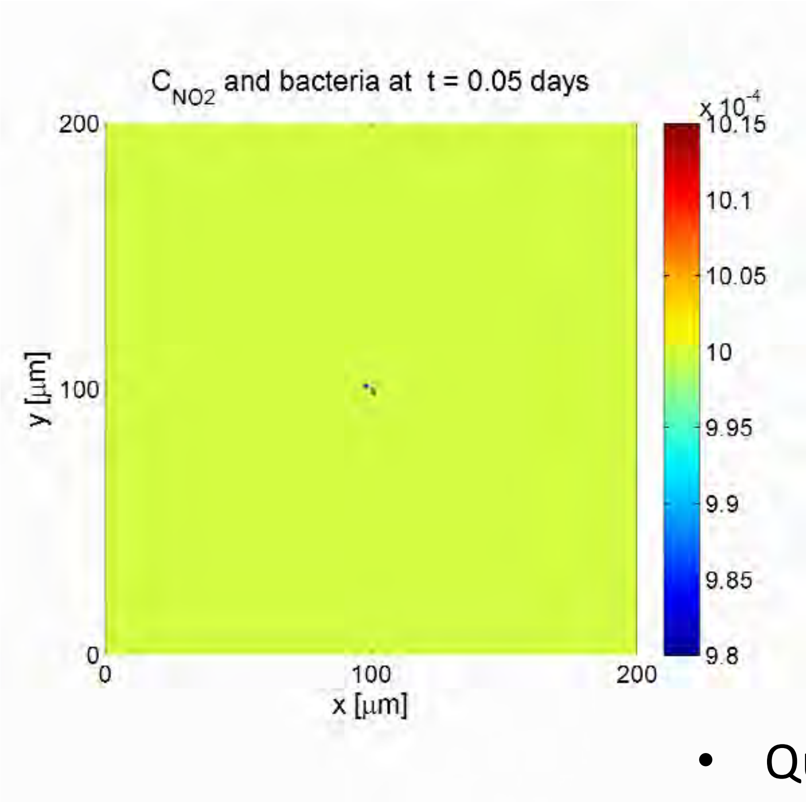
EPS - Extracellular Polymeric Substances

HET - HETerotrophs

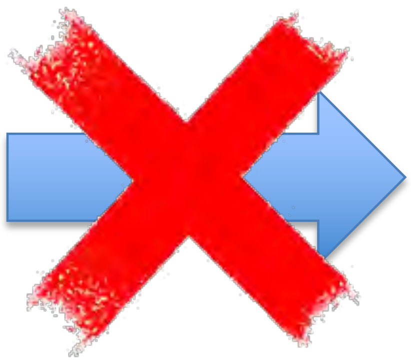
NOB - Nitrite Oxidizer Bacteria



Mapping this to the Real World

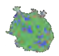
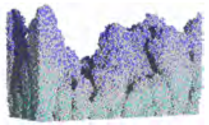
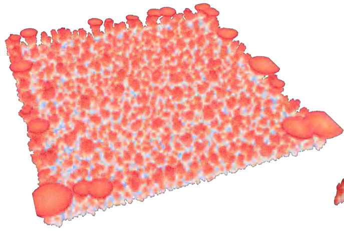
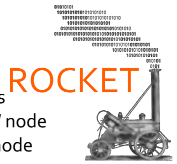


Ofiteru 2014



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

Scaling up

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

Scaling up

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

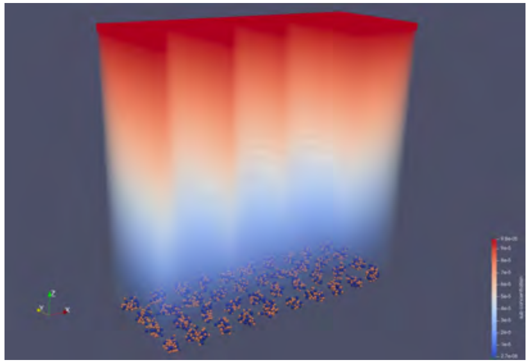
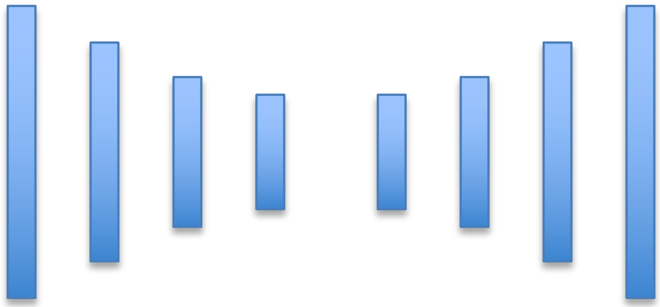
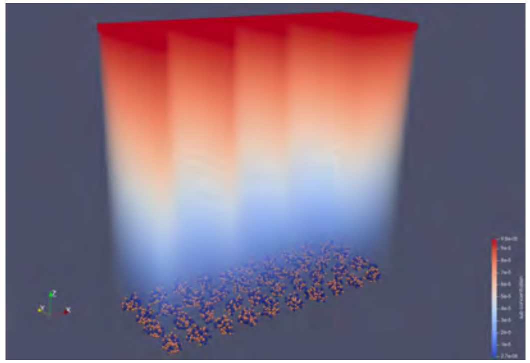
Building a Deep Learning Emulator

- 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

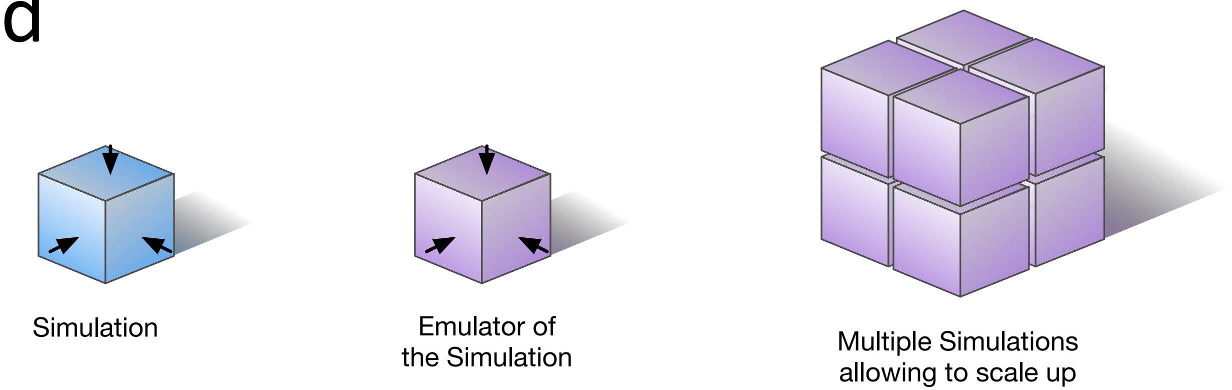
Building a Deep Learning Emulator

- 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



Fine-Tuning the Simulation

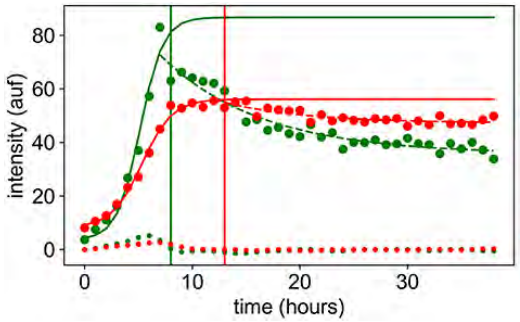
Why might the simulation need tuning?

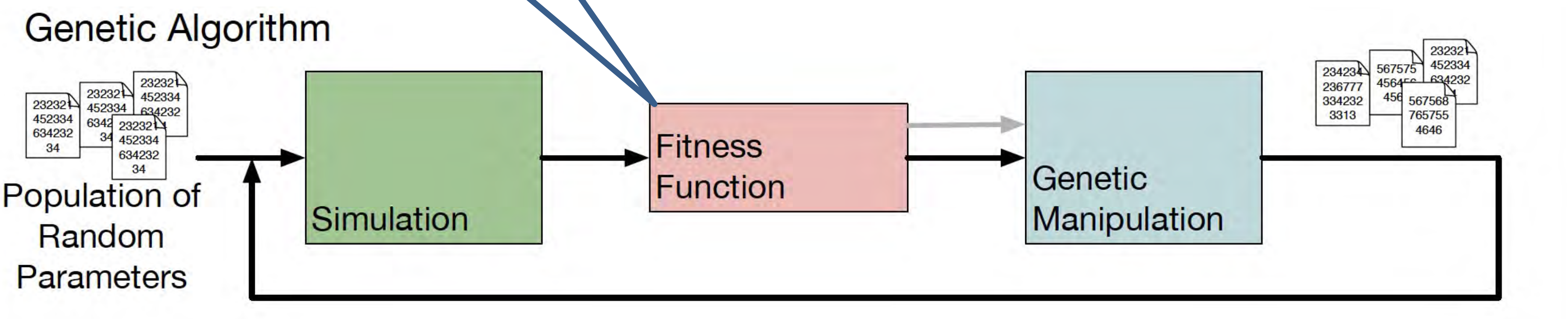
- Simulations are ‘best guesses’ as to how a system works
  – Parameters often based on results from papers/ books
  – Model is based on our understanding of how the system works
  – Can we match the output of the simulation to the real world?

Start with a fairly simple Genetic Algorithm approach

A, μ and τ

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





Hand-Crafted Fitness

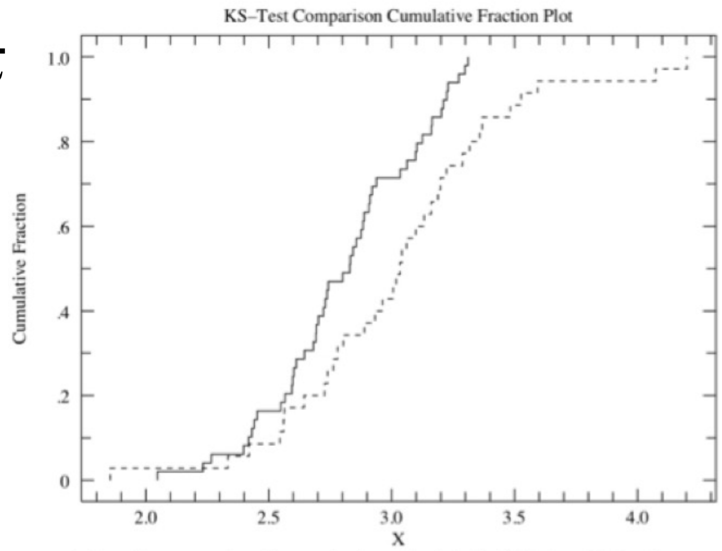
- Fitness function defined by comparing experimental and simulation data
- Multiple runs of simulation
- From each simulation compute A, μ and τ
- Compute empirical CDFs
- Compare with same for experiments

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

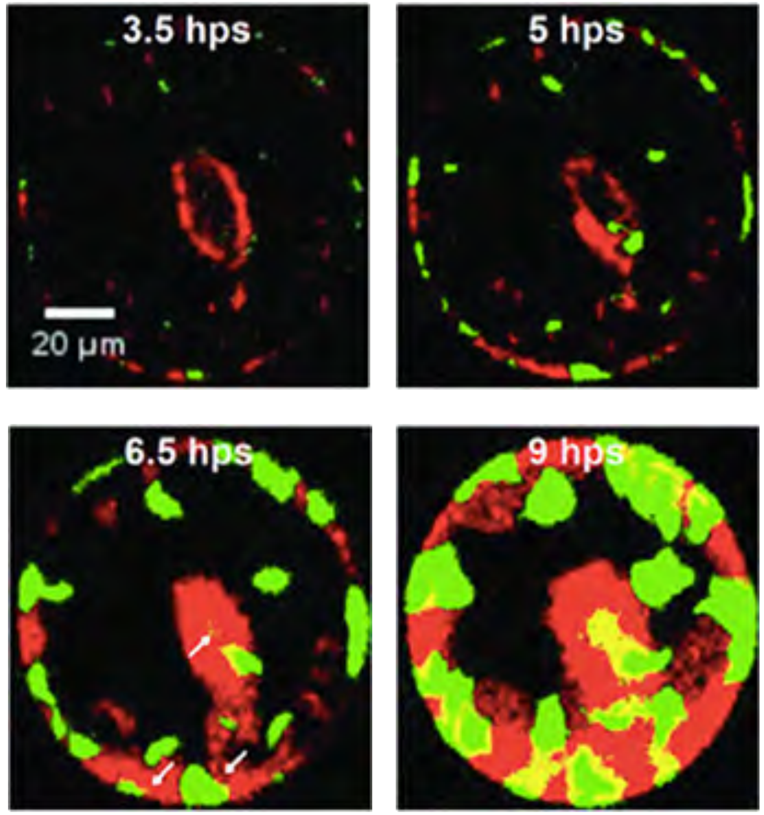
$$f = \frac{1}{0.1 + L}$$



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

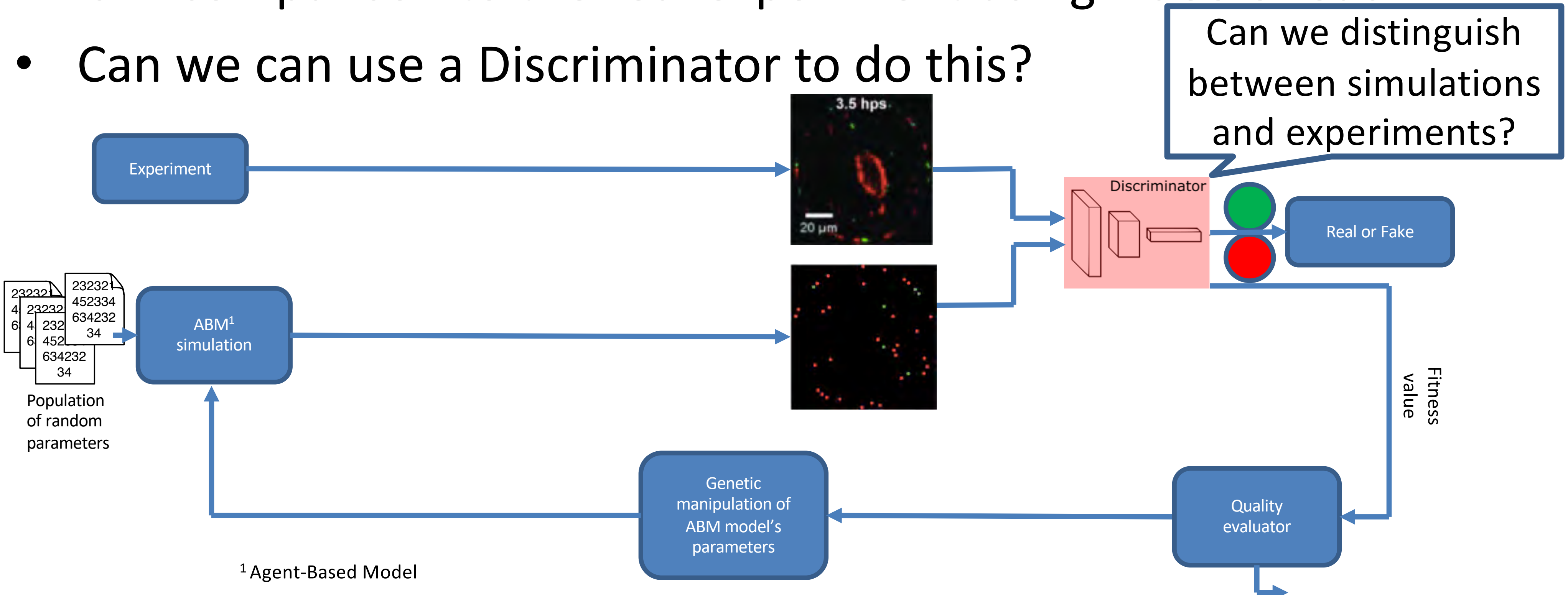
Brain storm – what else is there?

- 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’

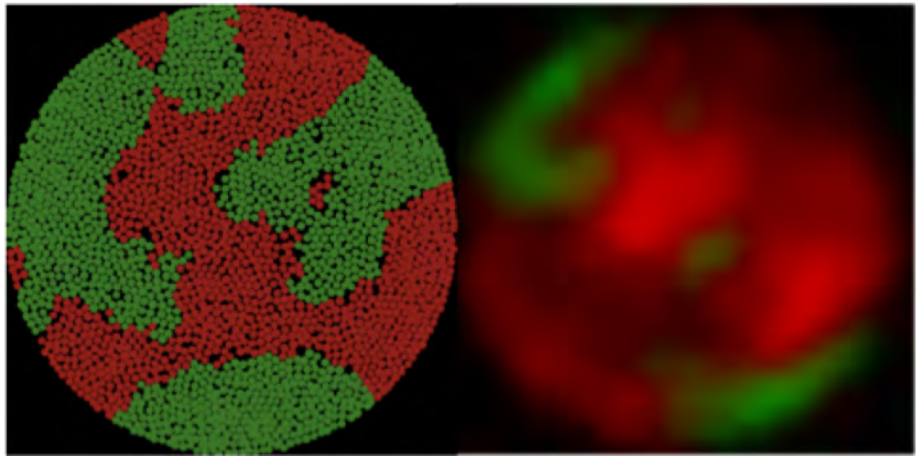
- Can we get Deep Learning to tell us how good our simulation is in comparison to the real experiment using video of each?
- Can we can use a Discriminator to do this?



[1] Agent-Based Model

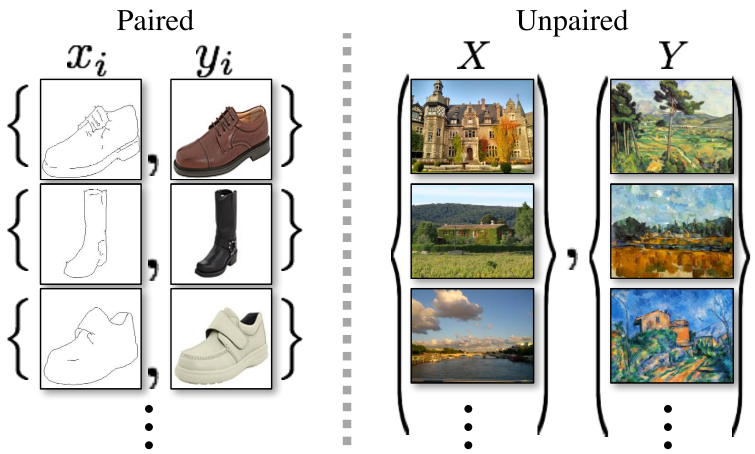
This will fail: Simulation looks nothing like Experiment

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

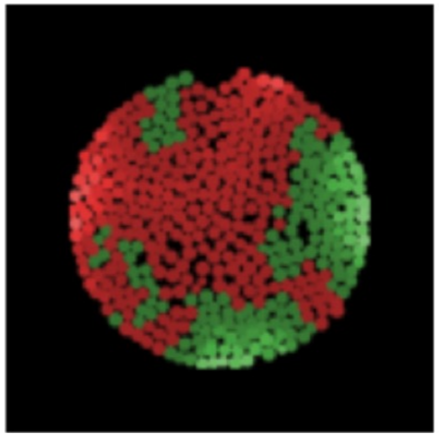


Making Simulations look more 'Real'

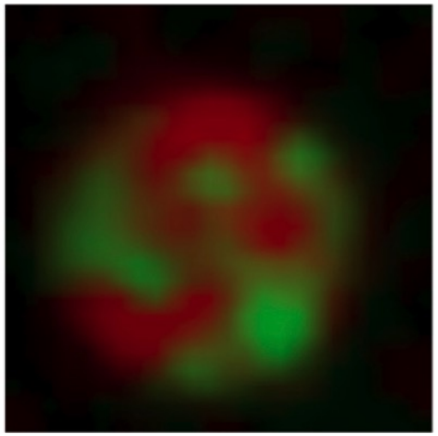
- Style Transfer
- Using CycleGAN
- No need for paired images



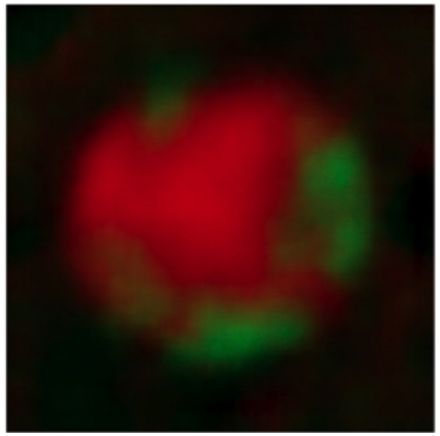
Input (simulation)



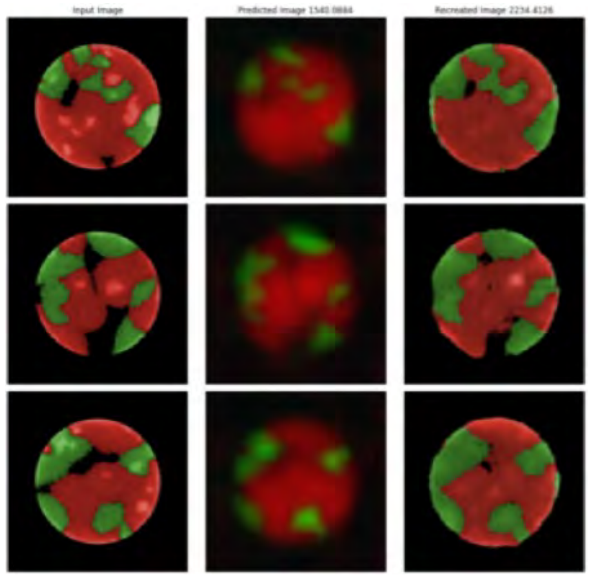
Source (experiment)



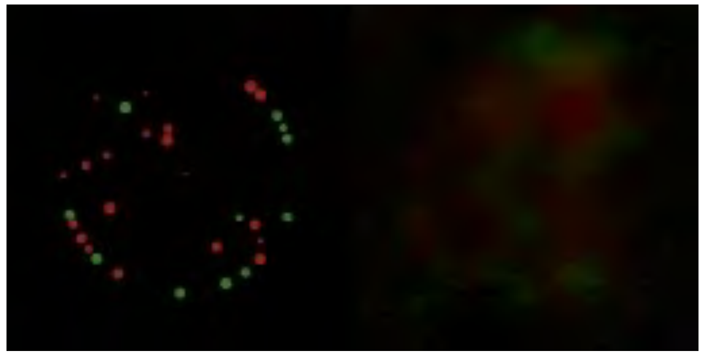
Output (experiment-like sim)

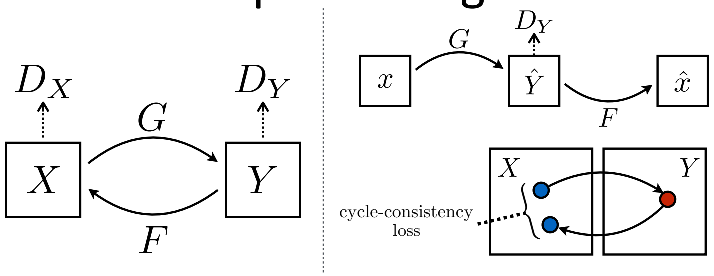


Style transfer of simulation data



- Unpaired Image-to-Image Translation
  - CycleGAN
- Pre-trained on images that resemble the distribution of the experiments
- Pad images to match size of experiments
- Lambda of 50 (not 10)
  - More importance on the cycle-consistency loss preserving the information

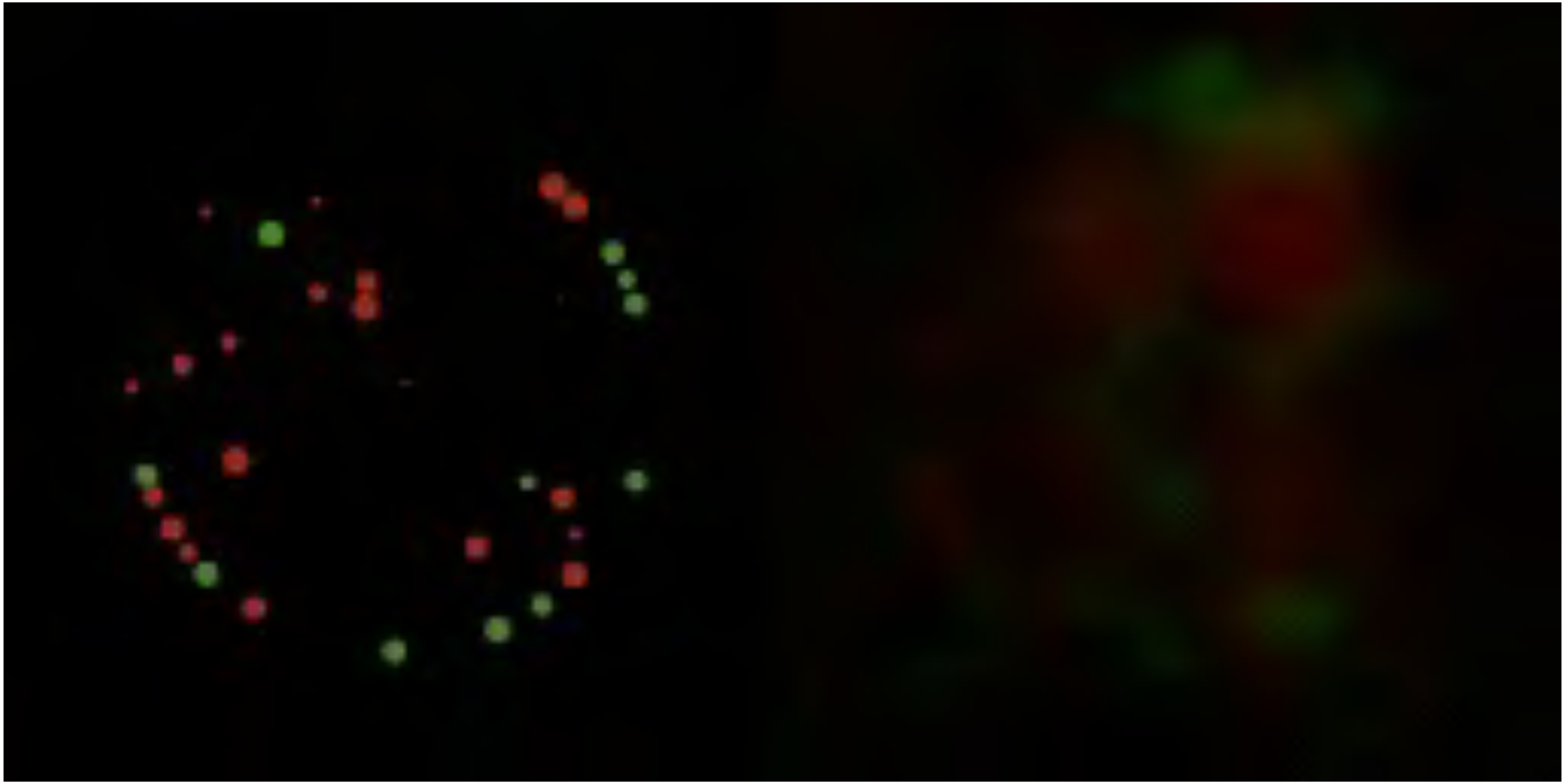




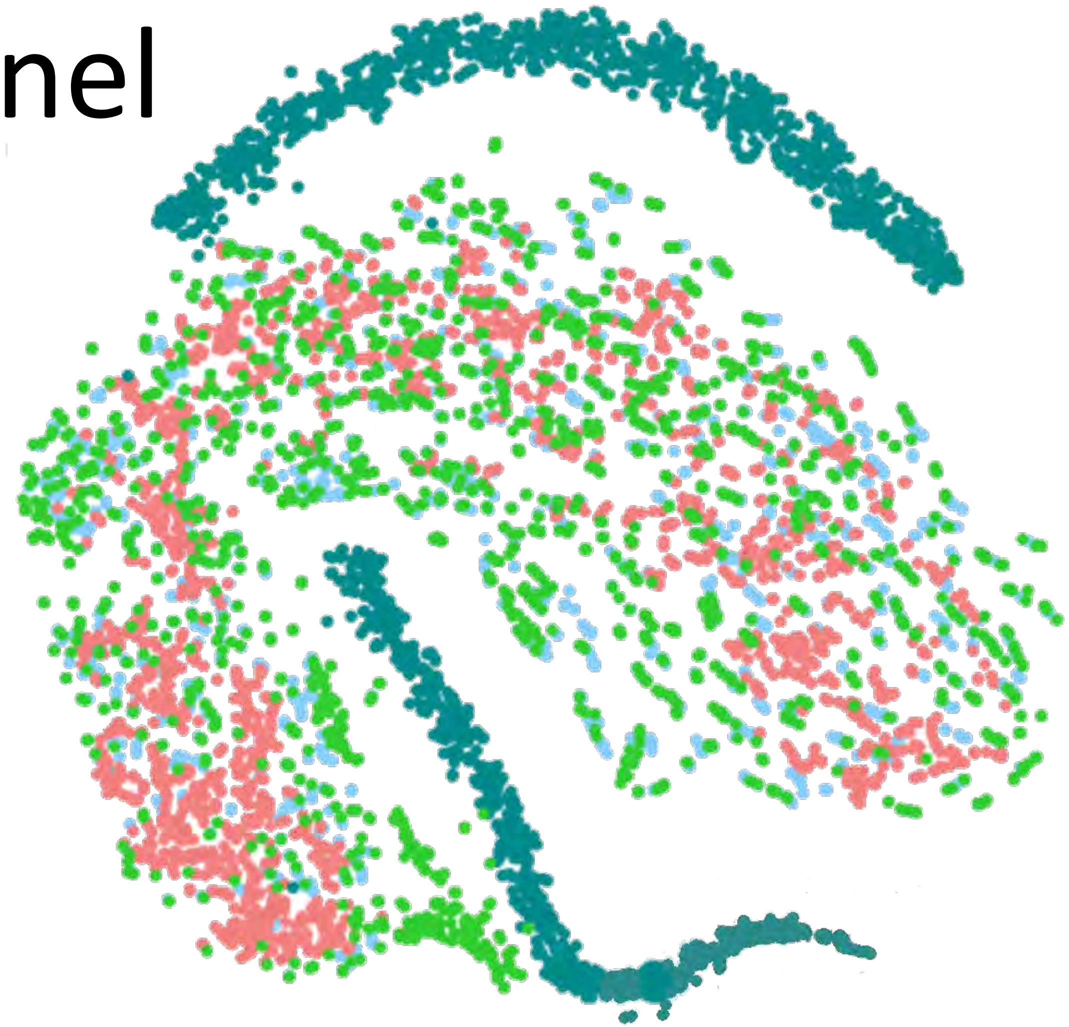
X: Simulation Y: Experiment

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

The finished simulation

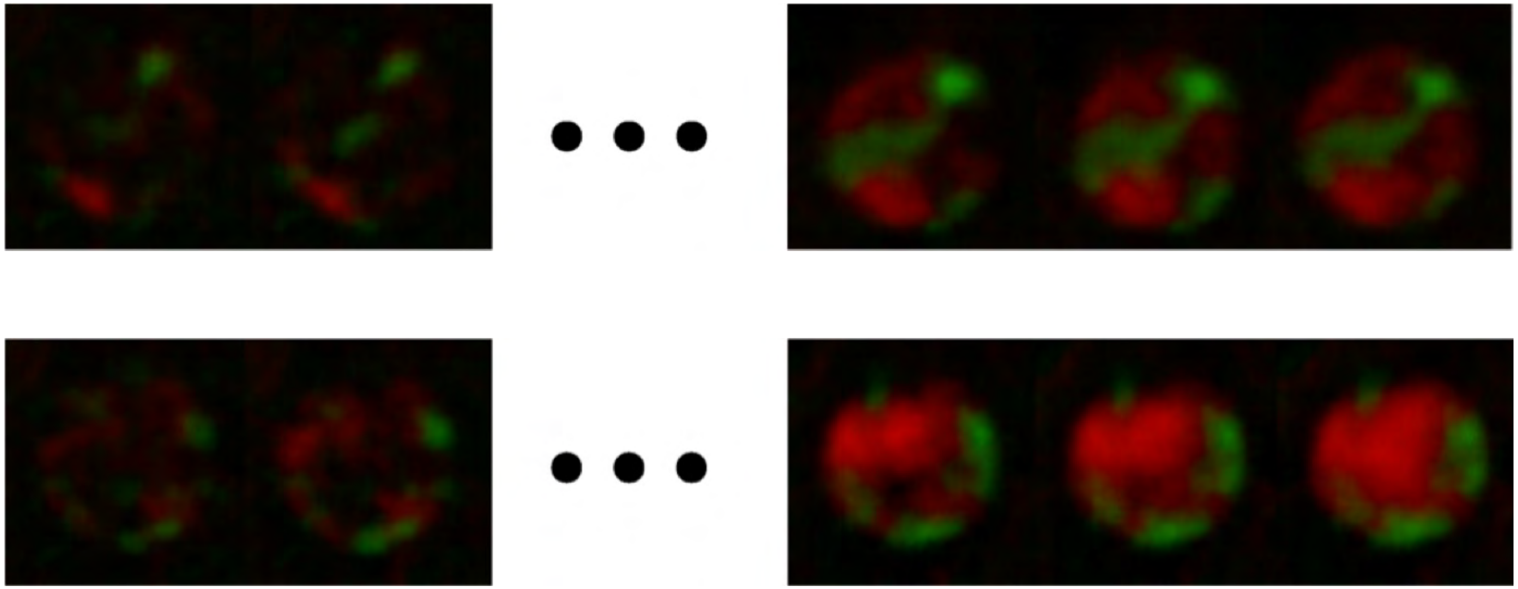


T-SNE Liner Kernel



Discriminator

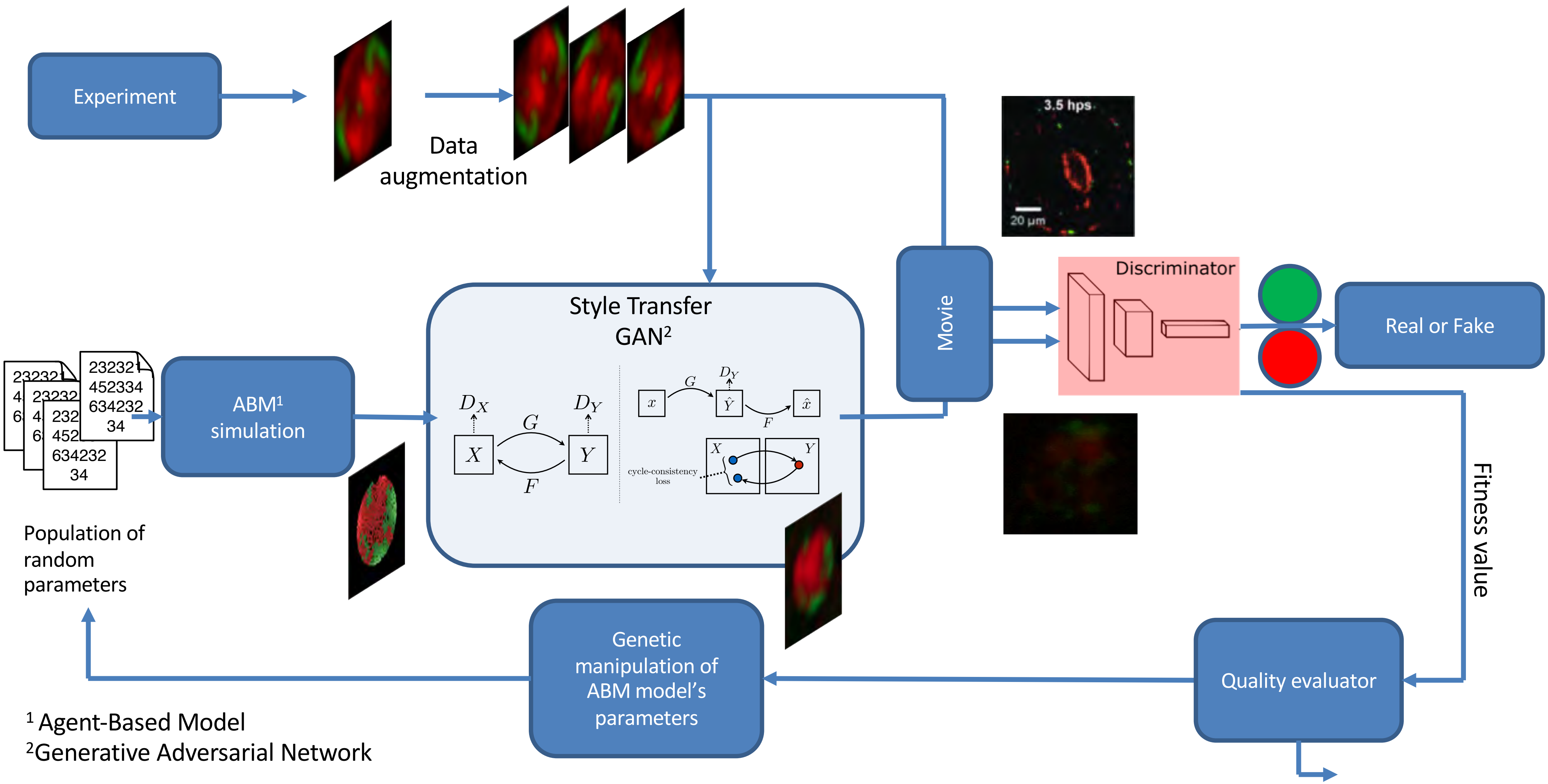
Input: 15 images of the growth phase



Experiment

Simulation

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



¹ Agent-Based Model
²Generative Adversarial Network

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