Spatial Evolutionary Generative Adversarial Networks

ANYSCALE LEARNING FOR ALL

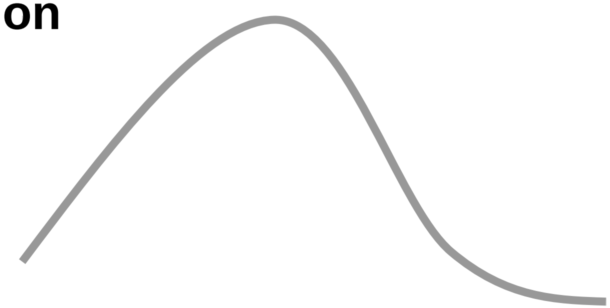
alfagroup.csail.mit.edu

MIT CSAIL

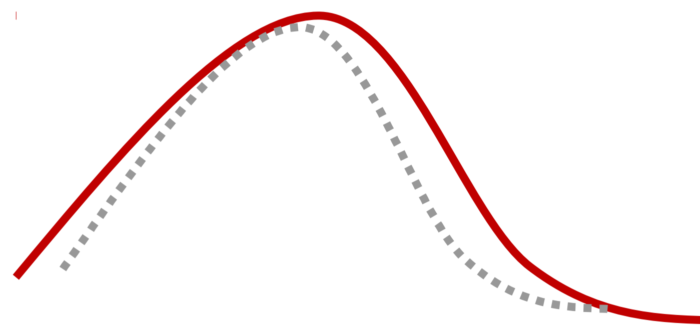
Generative Modeling

We have data samples that are drawn from a distribution



Can we learn a generative model that captures that distribution so we can generate new data samples?

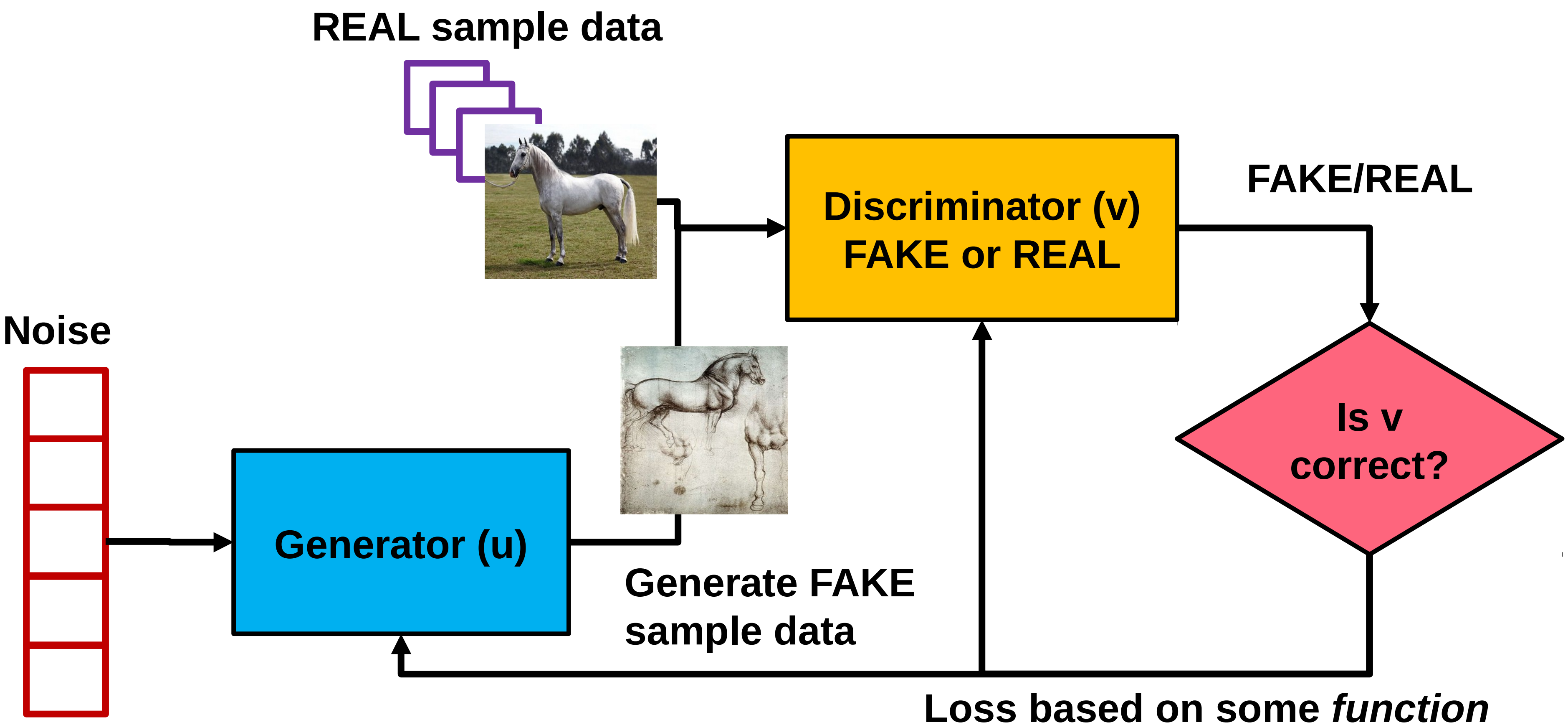
A current popular method to learn generative models is to use Generative Adversarial Networks

ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Generative Adversarial Network

Generative Adversarial Networks (GANs) create generative and discriminative models from data with an adversarial system



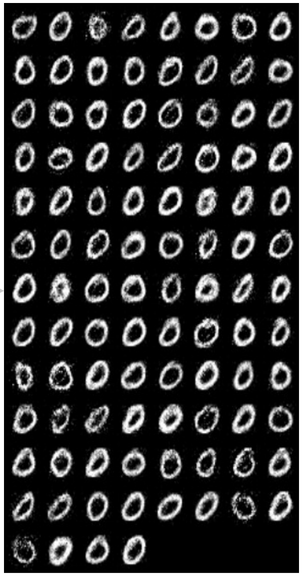
ALFA
ANYSCALE LEARNING FOR ALL

MIT CSAIL

GAN Pathologies

Generator 1

Generator 2



Mode collapse

Difficult to train due to pathologies, e.g. mode and discriminator collapse

Biological Arms Races

- **Biological arms races can generate mimicry**
- **Can coevolutionary algorithms help to improve robustness in GANs?**
  - **Our prior work solving GAN pathologies with Competitive Coevolution: Lipizzaner**
    - **ES GAN training on a theoretical GAN problem [Al-Dujaili, et al, 2018]**
    - **SGD GAN training with one loss function within a spatial separated evolving population [Schmiedelechner, et al, 2018]**
  - **E-GAN [Wang et al, 2018], 3 generator population, 3 loss functions, only 1 discriminator**

Mustangs

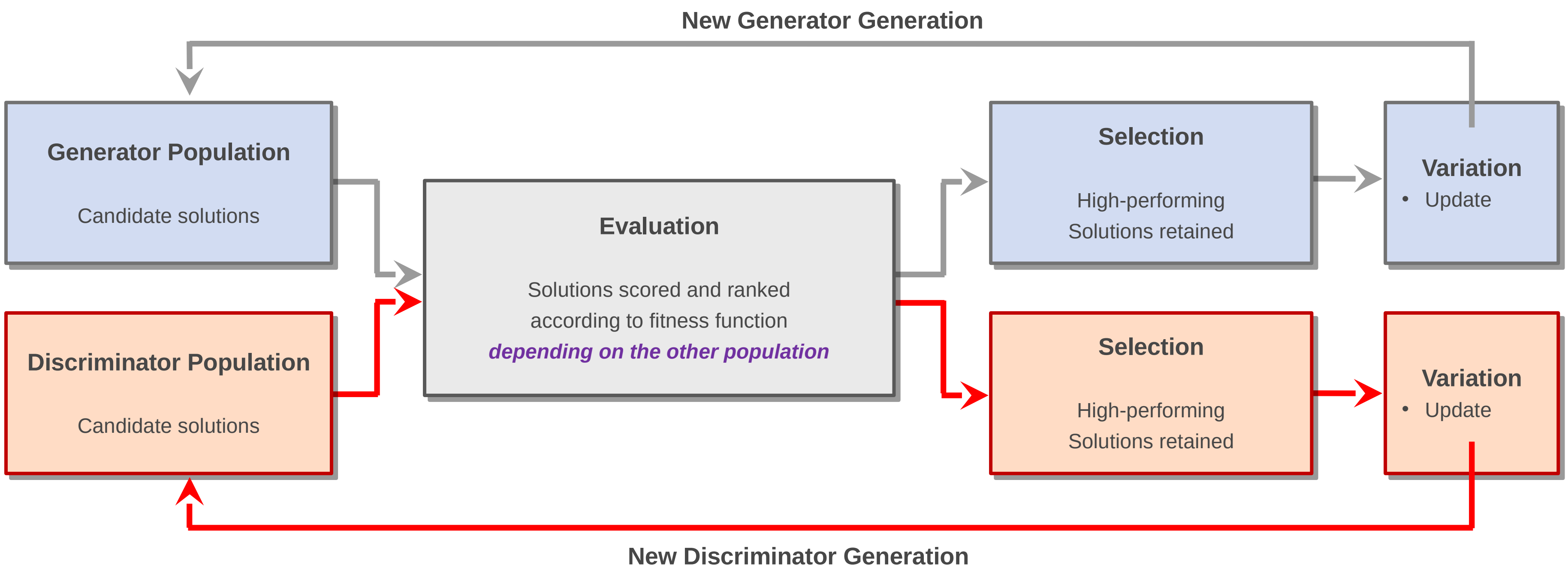
A distributed, coevolutionary framework to train GANs with gradient-based optimizers

- **Fast convergence due to gradient-based steps**
- **Robustness due to coevolution**
- **Improved convergence due to hyperparameter evolution**
- **Diverse solutions due to mixture evolution and multiple loss functions**
- **Scalability due to spatial distribution topology**

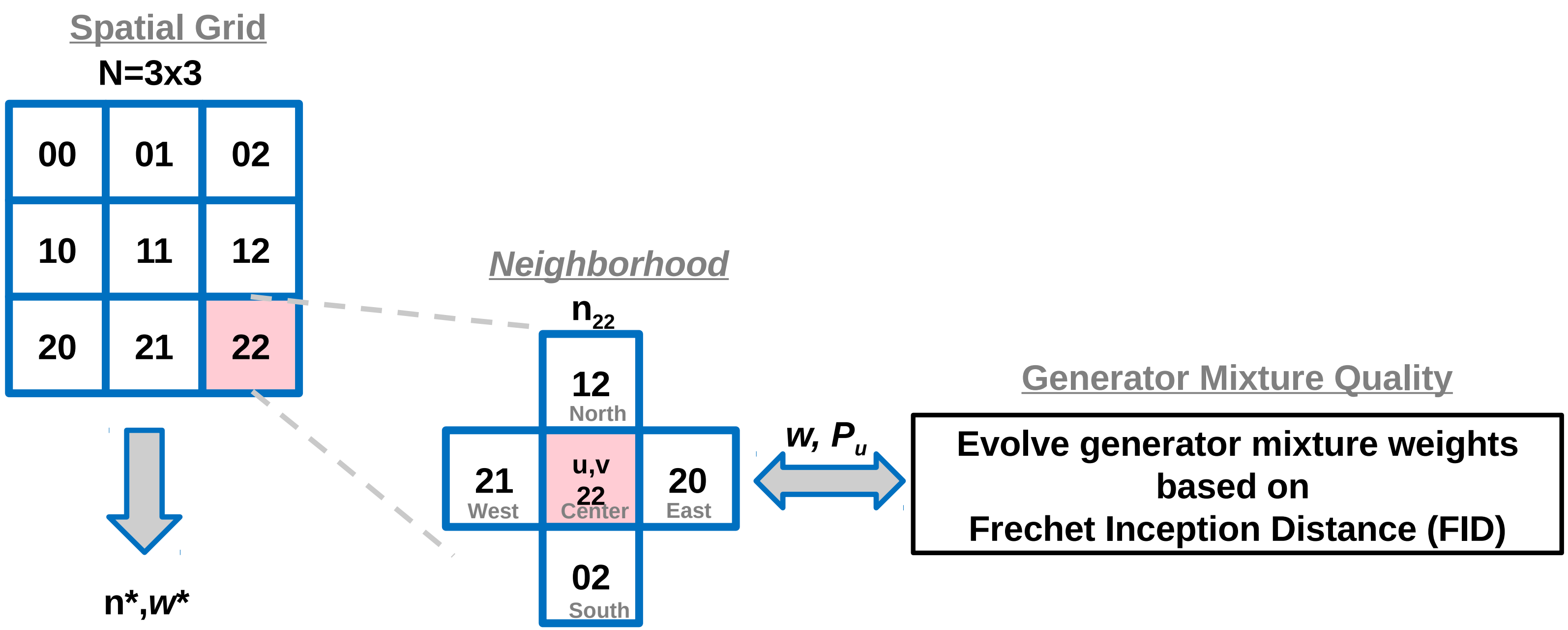
ANYSCALE LEARNING FOR ALL

MIT CSAIL

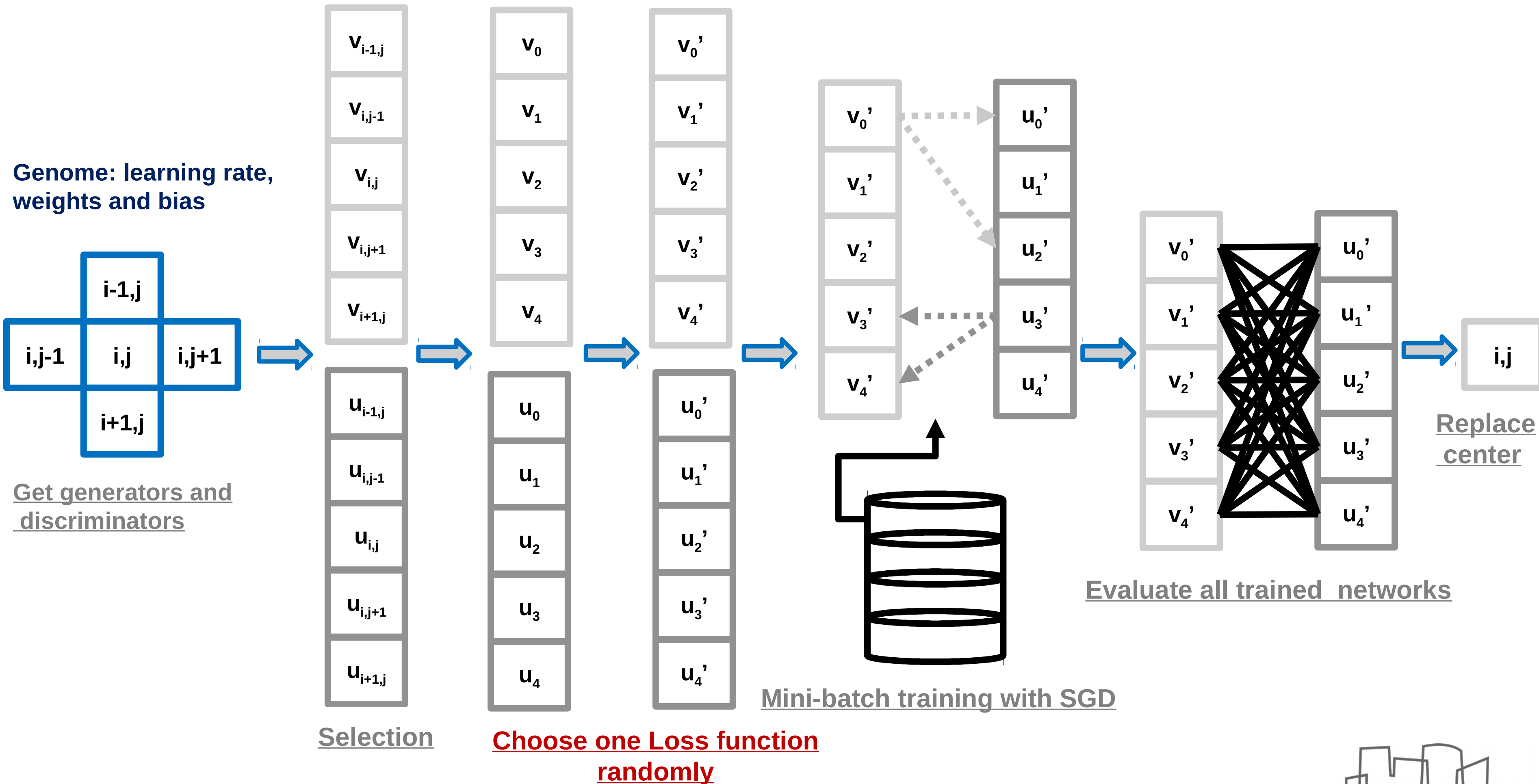
Competitive Coevolution



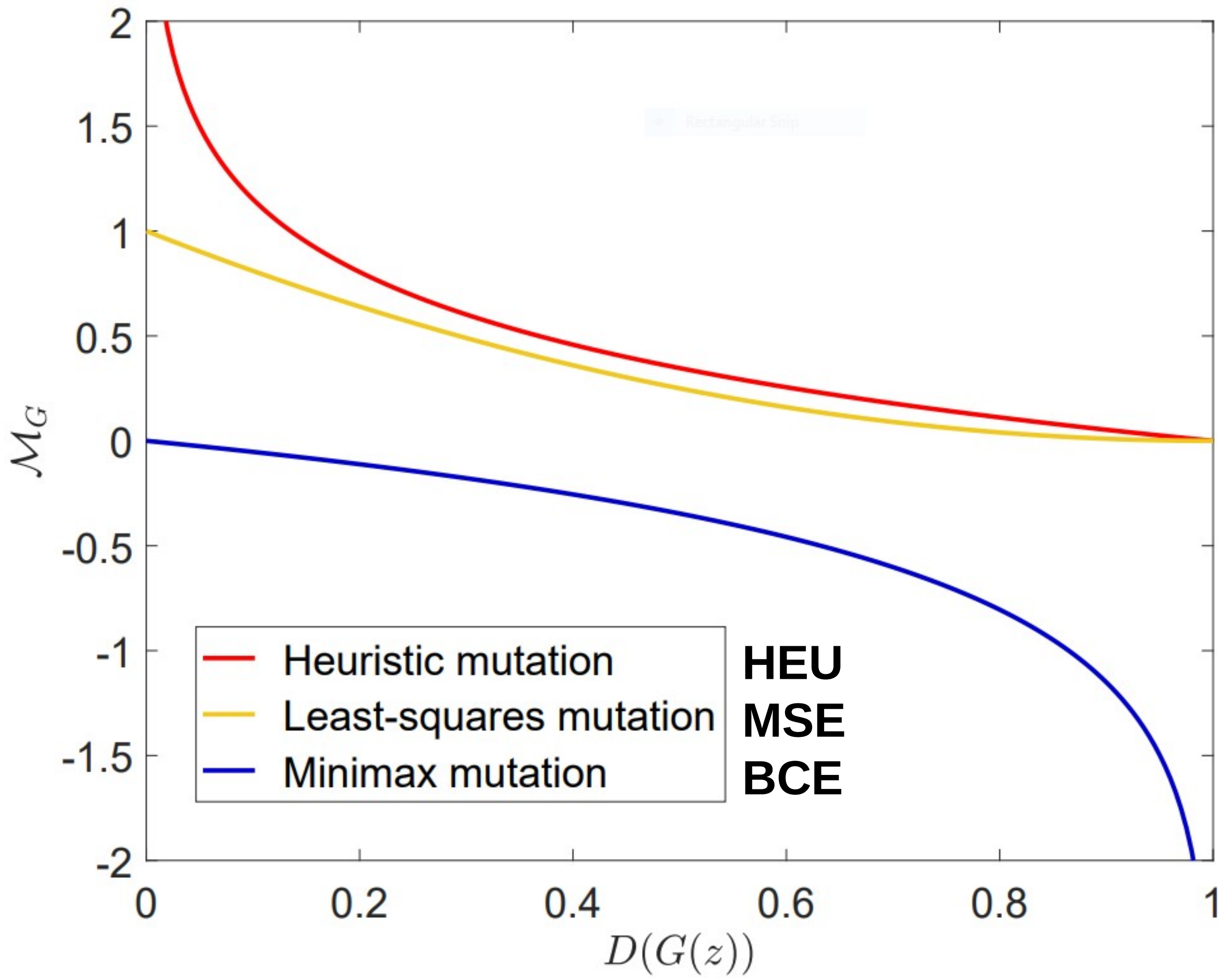
Mustang – Spatial separation and generator mixtures



Mustang – GAN training with SGD and different loss functions



Loss Functions



ANYSCALE LEARNING FOR ALL

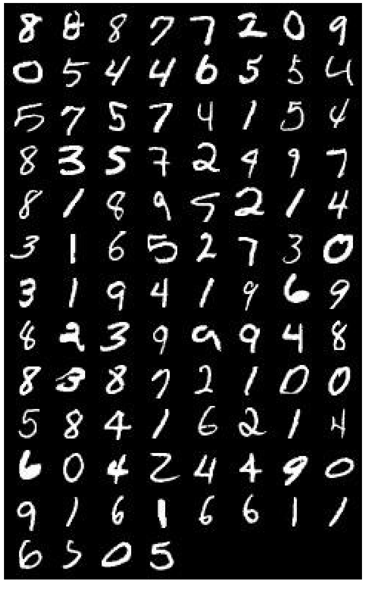
MIT CSAIL

Experiments

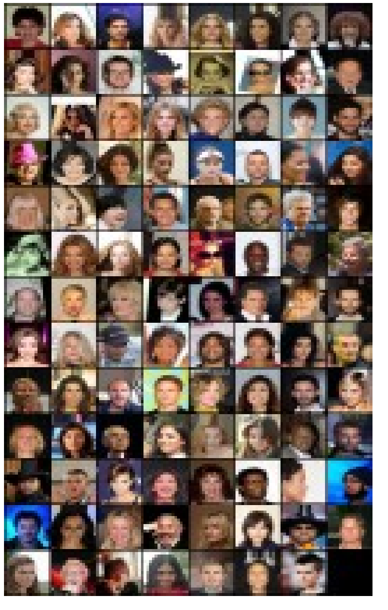
Evaluated methods (9h of compute):

- **GAN-BCE:** BCE loss function, 1 generator, 1 discriminator
- **E-GAN**: 3 loss functions, 3 generators, 1 discriminator
- **Lip-BCE**: BCE loss function, 3x3 grid
- **Lip-MSE**: MSE loss function, 3x3 grid
- **Lip-HEU**: HEU loss function, 3x3 grid
- **Mustangs**: 3 loss functions, 3x3 grid

MNIST

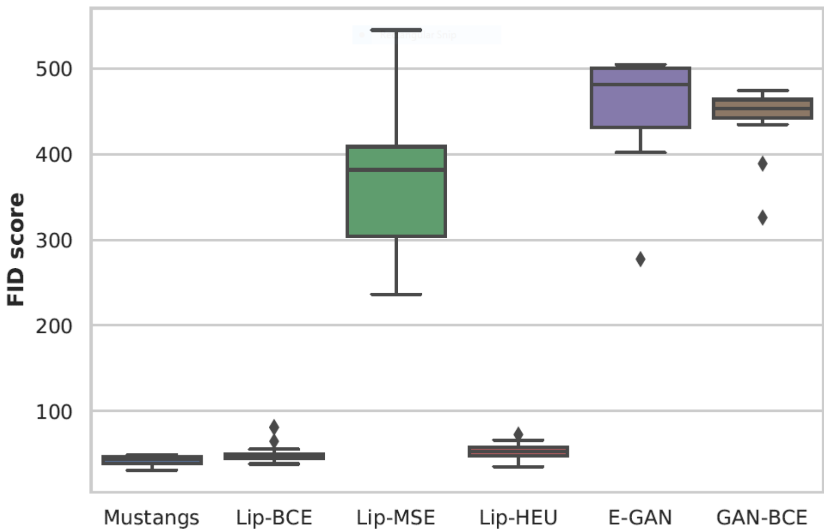


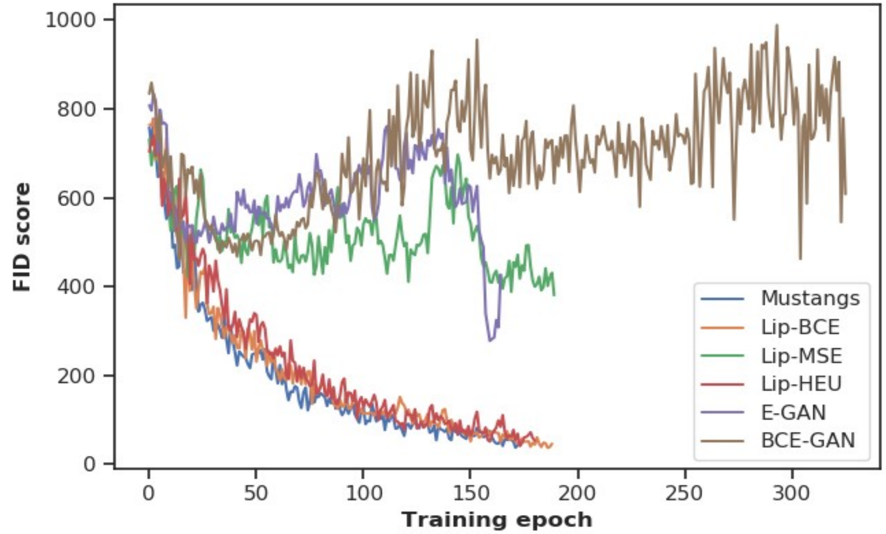
CelebA



Mustang: Mixture of generators from a spatially separated evolutionary GAN training with SGD and multiple loss functions

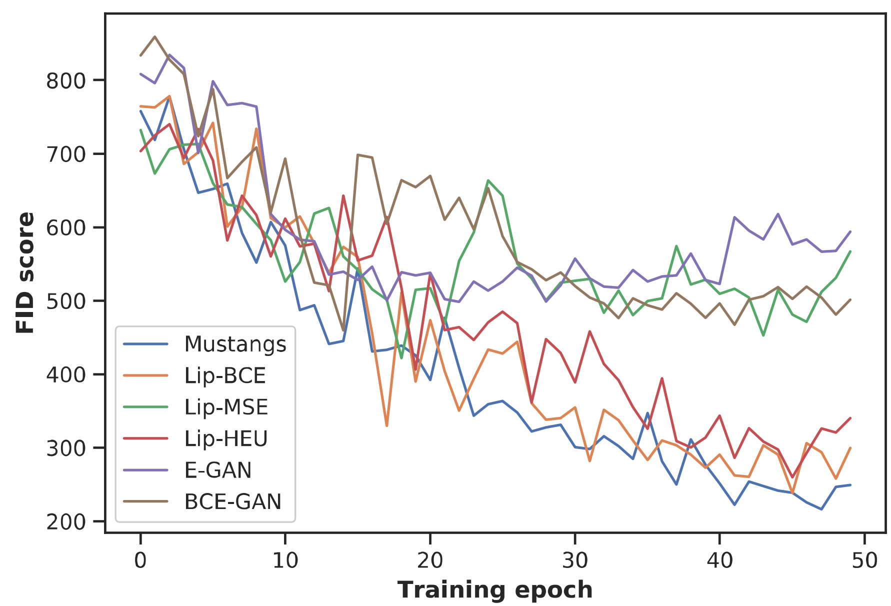
MNIST Results





Population improves FID score

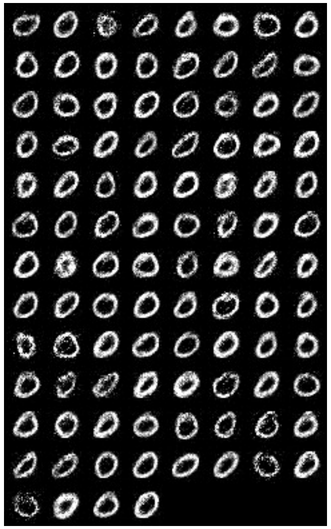
More training epochs does not guarantee improved FID



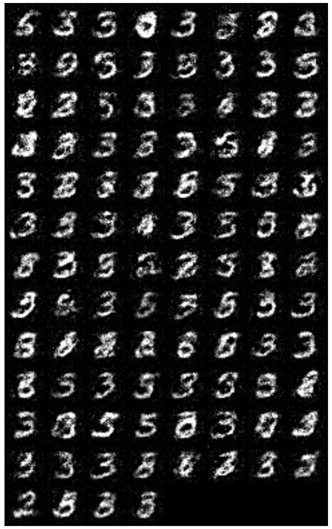
ANYSCALE LEARNING FOR ALL

MIT CSAIL

Generator output diversity for MNIST

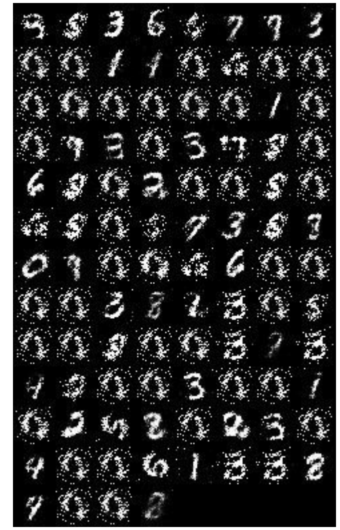


Mode collapse



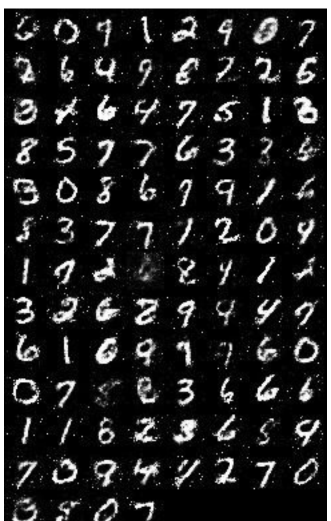
GAN-BCE, TVD: 0.51

E-GAN, TVD: 0.53



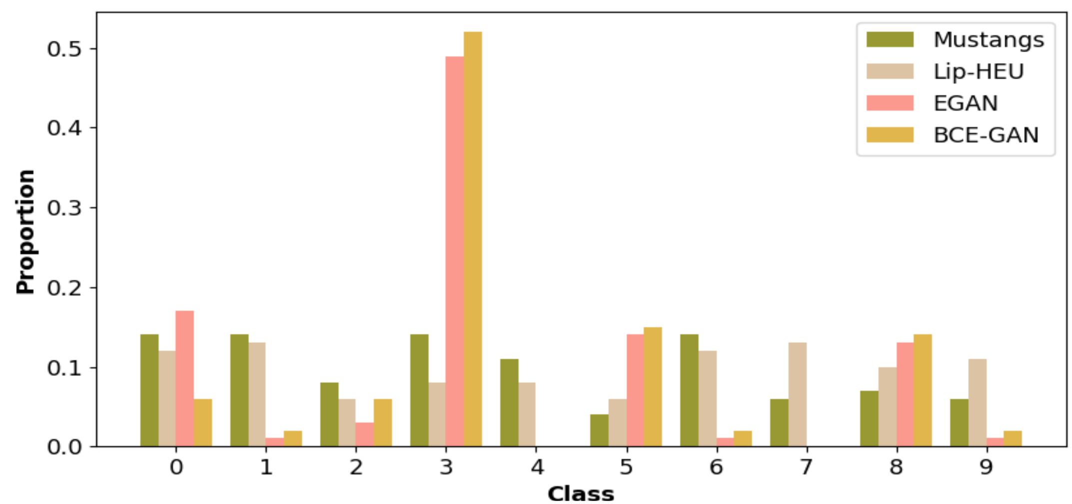
Lip-MSE, TVD: 0.37

Lip-BCE, TVD: 0.17



Lip-HEU, TVD: 0.12

Mustangs, TVD: 0.18

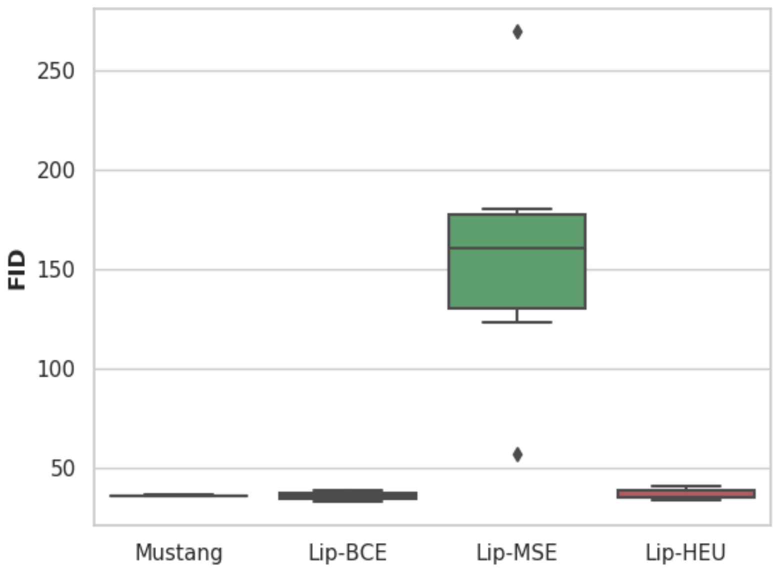


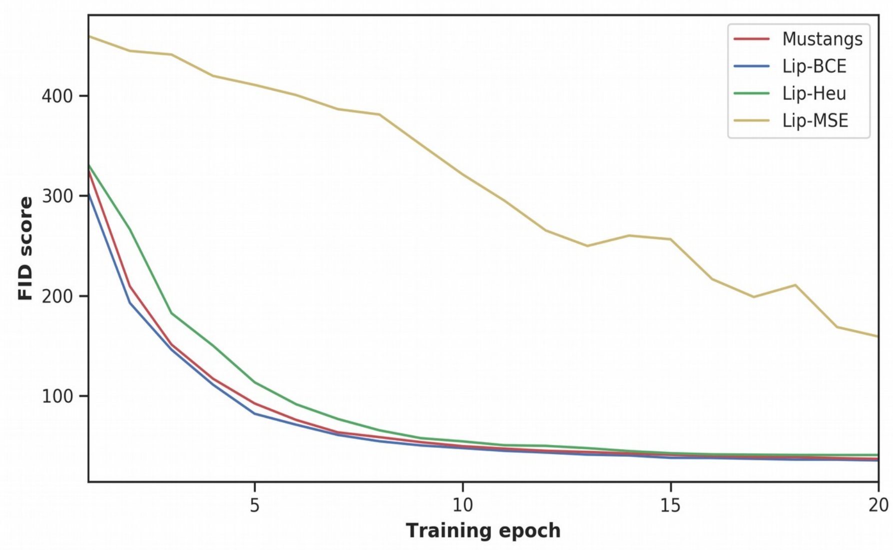
Spatial separation improves generator output diversity

ANYSCALE LEARNING FOR ALL

MIT CSAIL

Results CelebA





Choosing from multiple loss functions does not degrade performance

Mustangs

Lip-BCE

ANYSCALE LEARNING FOR ALL

MIT CSAIL

Summary

- **Mustang**: Evolves a mixture of generators by a spatially separated evolutionary GAN training with stochastic gradient descent and multiple loss functions
- Empirically showed that GAN training can be improved by boosting diversity
- Enhanced an existing spatial evolutionary GAN training framework by it choosing one of three loss functions
- Released an open source, distributed Python framework with Pytorch that use Docker
  - https://github.com/mustang-gan
- Future work to investigate scaling, larger populations, diversity and more problems