DeepMind

Multilevel Optimization for Generative Models, Games and Robotics

David Pfau

pfau@google.com

@pfau

CERN IML Workshop
9 April 2018

Part I

A Whirlwind Tour of Generative Models

Why Generative Models

Move beyond associating inputs to outputs

Understand and imagine how the world evolves

Recognize objects in the world and their factors of variation

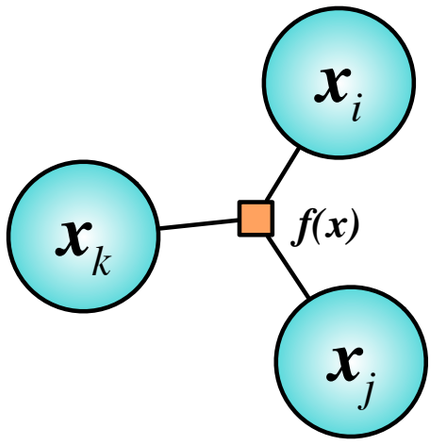
Detect surprising events in the world

Establish concepts as useful for reasoning and decision making

Imagine and generate rich plans for the future

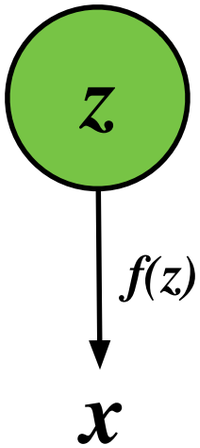
Part of a suite of complementary learning systems

Types of Generative Models



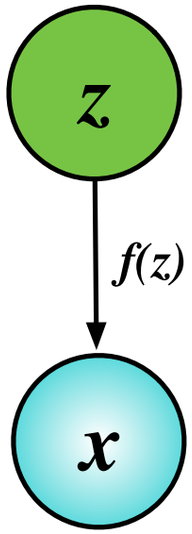
Fully-observed models

Model observed data directly without introducing any new unobserved local variables.



Implicit Models

Simulator-based models; models data as a transformation of an unobserved noise source using a parameterized function.



Latent variable models

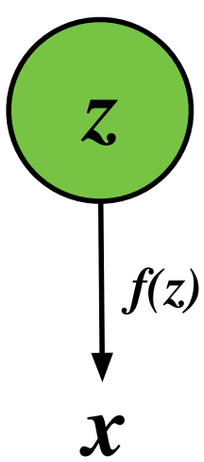
Introduce an unobserved random variable for every observed data point to explain hidden causes.

Generative Adversarial Networks

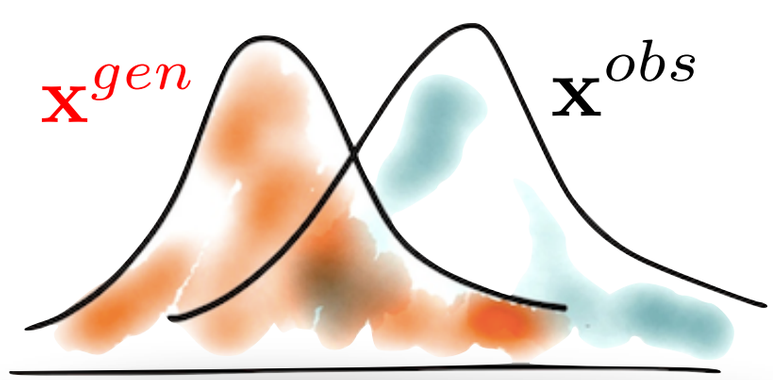
GAN Value Function

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$$

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Wade-Farley, S. Ozair, A. Courville, Y. Bengio (2014)



$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x}^{gen} = f_\phi(\mathbf{z})$$



Minimax optimisation

$$\min_\theta \max_\phi \mathcal{L}(\theta, \phi)$$

Optimal Discriminator

$$D_{\theta^*}(x) = \frac{p^*(x)}{p^*(x) + q_\phi(x)}$$

$$\mathcal{L}(\theta^*, \phi) = 2JS(p^* || q_\phi) - \log(4)$$

Generative Adversarial Networks

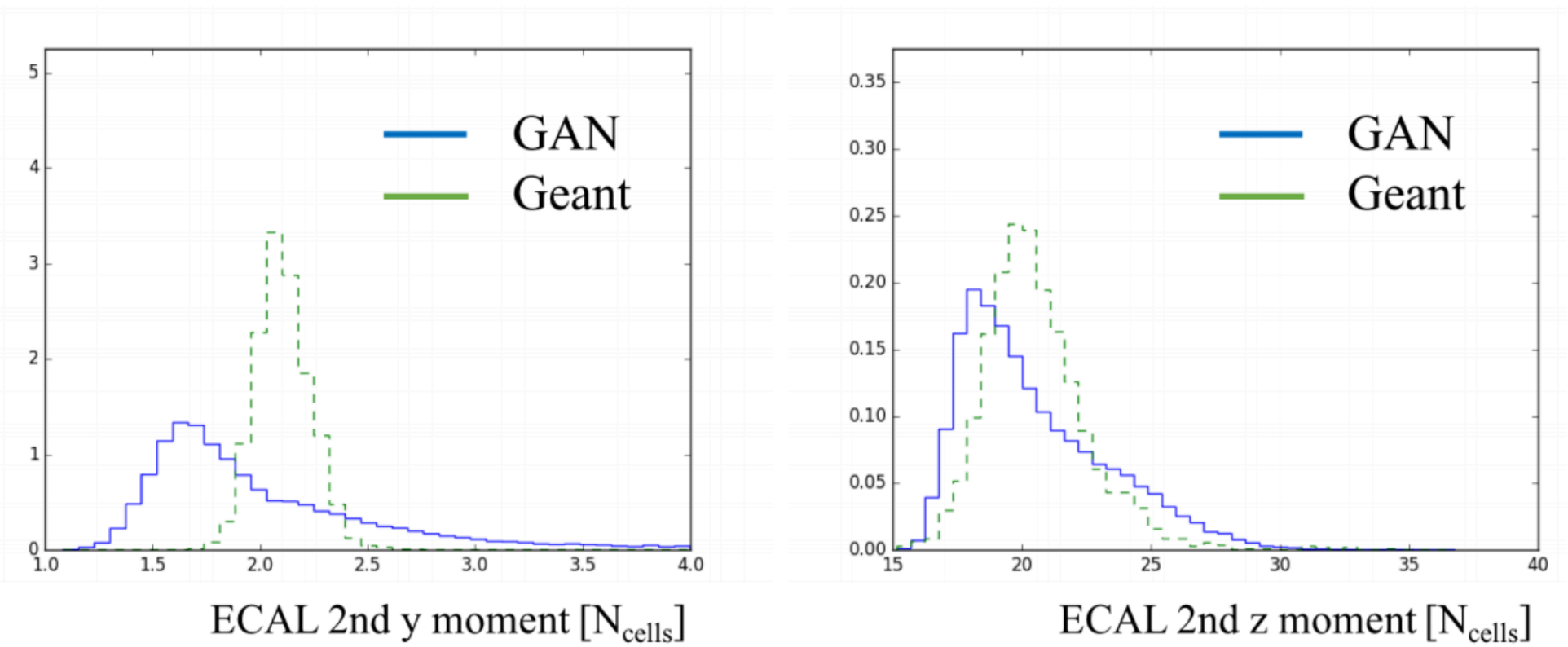
Progressively-Growing Generative Adversarial Networks

POTTEDPLANT HORSE SOFA

T. Karras, T. Aila, S. Laine, J. Lehtinen (2018)

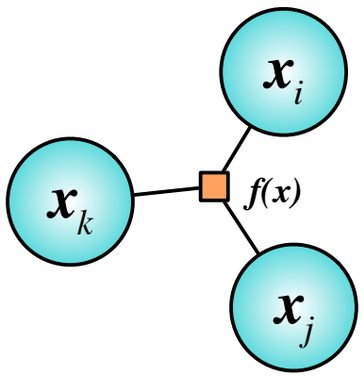
Generative Adversarial Networks

GANs are Poor at Matching Statistics of True Distribution



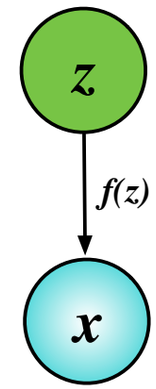
F. Carminati et al (2017)

Generative Models Beyond GANs



Fully-observed models

Model observed data directly without introducing any new unobserved local variables.



Latent variable models

Introduce an unobserved random variable for every observed data point to explain hidden causes.

- Boltzmann Machines
  D. Ackley, G. Hinton and T. Sejnowski (1985)
- PixelCNN
  A. van den Oord, N. Kalchbrenner and K. Kavukcuoglu (2016)
- WaveNet
  A. van den Oord et al (2016)
- Normalizing Flows
  D. Rezende and S. Mohamed (2015)
- Real Non-volume-preserving Flows
  L. Dinh, J. Sohl-Dickstein and S. Bengio (2017)

- Helmholtz Machines
  P. Dayan, G. Hinton, R. Neal and R. Zemel (1995)
- Variational Autoencoders
  D. Rezende and S. Mohamed (2014)
  D. Kingma and M. Welling (2014)
- Importance Weighted Autoencoders
  Y. Burda, R. Grosse, R. Salakhutdinov (2016)
- Variational Inference for Monte Carlo Objectives
  A. Mnih and D. Rezende (2016)
- Memory, RL and Inference Networks
  G. Wayne et al. (2018)

Part II

Generative Adversarial Networks and Bilevel Optimization

Adversarial Learning

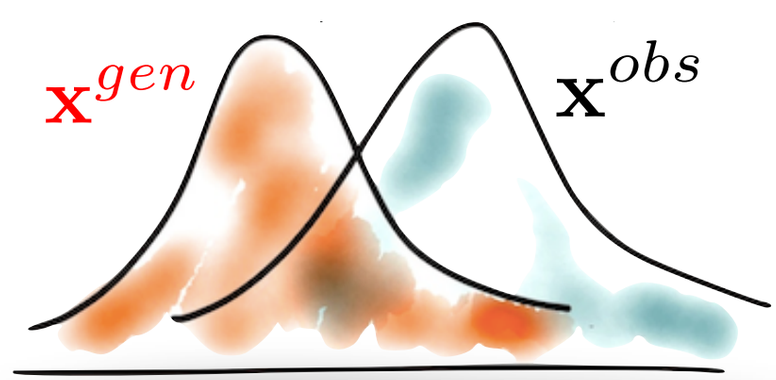
Generative Adversarial Networks

GAN Value Function

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$$

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Wade-Farley, S. Ozair, A. Courville, Y. Bengio (2014)

$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x}^{gen} = f_\phi(\mathbf{z})$$



Minimax optimisation

$$\min_\theta \max_\phi \mathcal{L}(\theta, \phi)$$

Optimal Discriminator

$$D_{\theta^*}(x) = \frac{p^*(x)}{p^*(x) + q_\phi(x)}$$

$$\mathcal{L}(\theta^*, \phi) = 2JS(p^* || q_\phi) - \log(4)$$

Adversarial Learning

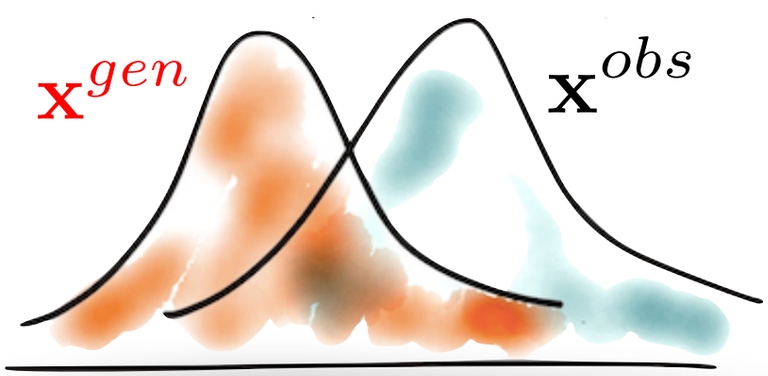
Generative Adversarial Networks

Discriminator Loss $\mathcal{L}_D(\theta, \phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$

Generator Loss $\mathcal{L}_G(\theta, \phi) = -\mathbb{E}_{q_\phi(\mathbf{x})}[\log(D_\theta(\mathbf{x}))]$

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Wade-Farley, S. Ozair, A. Courville, Y. Bengio (2014)

$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x}^{gen} = f_\phi(\mathbf{z})$$



Multilevel optimisation

$$\min_\phi \mathcal{L}_G(\theta^*, \phi)$$
$$\theta^*(\phi) = \arg\min_\theta \mathcal{L}_D(\theta, \phi)$$

Where else does this appear?

Optimal Discriminator

$$D_{\theta^*}(x) = \frac{p^*(x)}{p^*(x) + q_\phi(x)}$$

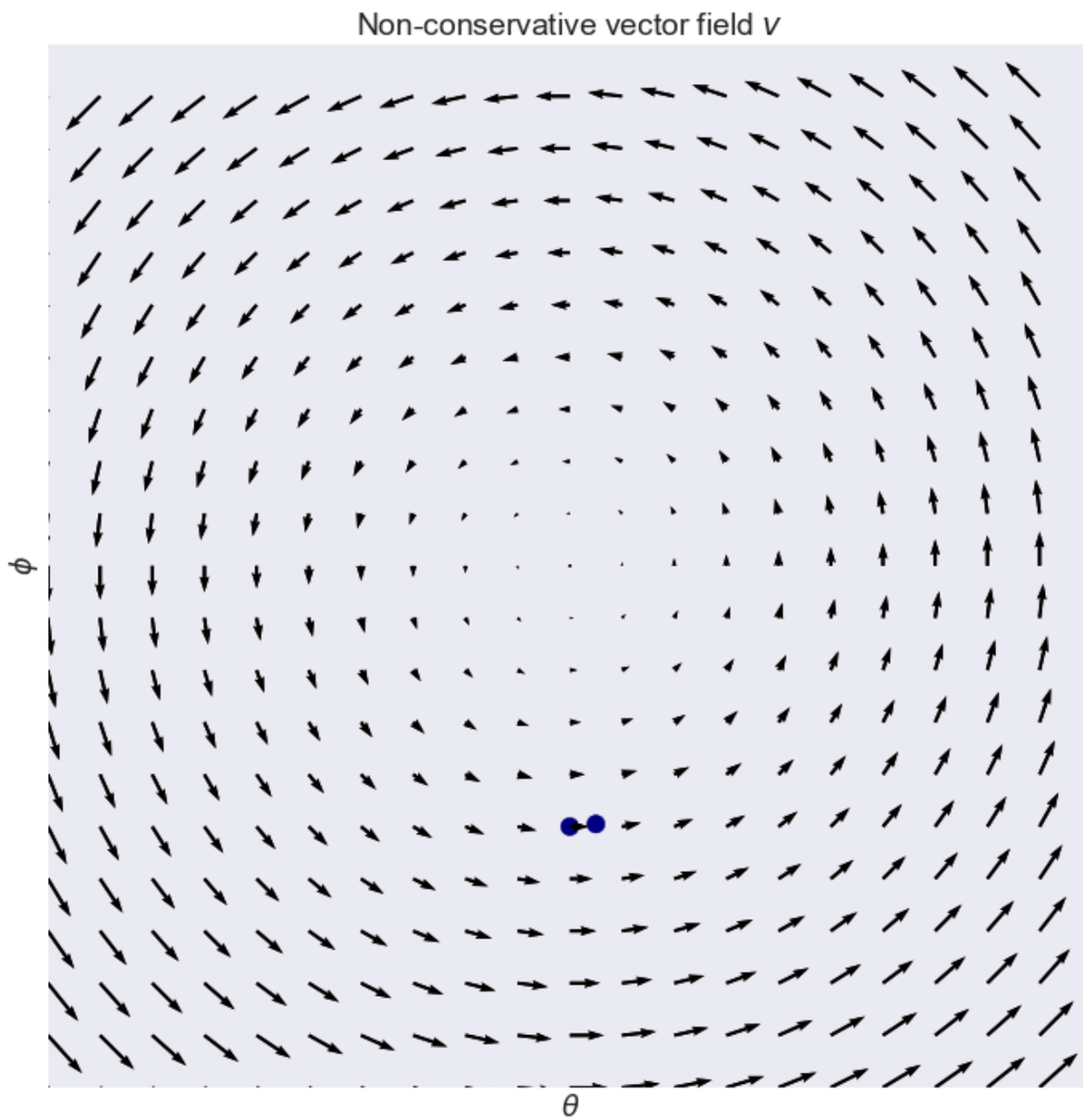
$$\mathcal{L}(\theta^*, \phi) = 2JS(p^* || q_\phi) - \log(4)$$

Bilevel Optimization

$$\phi^* = \arg\min_{\phi} F(\theta^*, \phi)$$
$$\theta^*(\phi) = \arg\min_{\theta} f(\theta, \phi)$$

- Also appears under the name "two time scale optimization"
- Naturally arises in adversarial settings in operations research (B. Colson, P. Marcotte and G. Savard 2005)
- Appearance of parameters in both problems couples solutions
- NP-Hard even for bilevel linear programming (P. Hansen, B. Jaumard and G. Savard 1992)
- Little formal study of optimization for deep learning

Bilevel Optimization



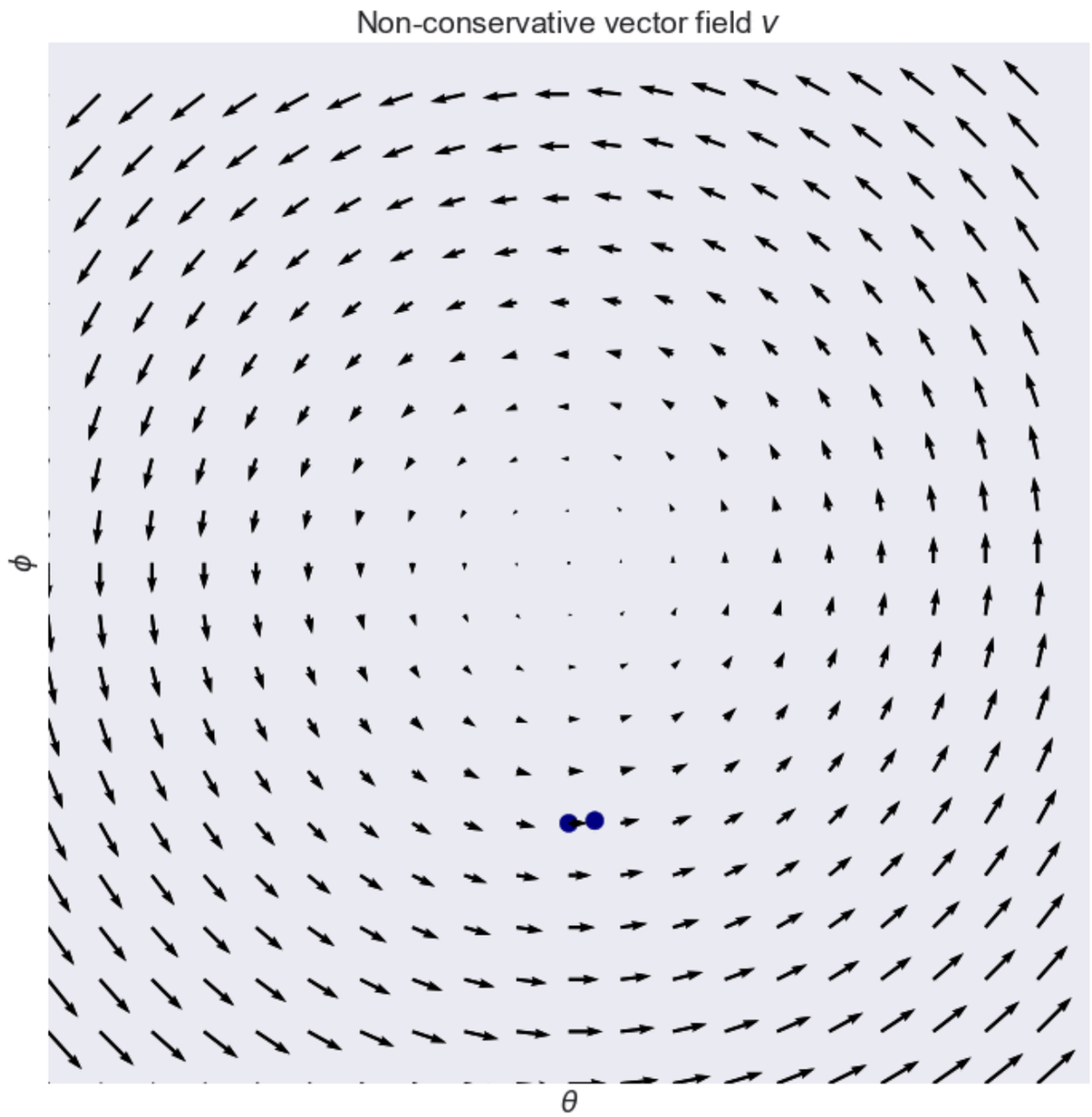
$$\phi^* = \arg\min_{\phi} F(\theta^*, \phi)$$
$$\theta^*(\phi) = \arg\min_{\theta} f(\theta, \phi)$$

$$\vec{v}(\phi, \theta) = \begin{pmatrix} \nabla_\phi F(\theta, \phi) \\ \nabla_\theta f(\theta, \phi) \end{pmatrix}$$

$$\vec{v}(\phi, \theta) \neq \nabla \rho(\phi, \theta)$$

No scalar function whose gradient is the vector field.

Bilevel Optimization



M. C. Escher

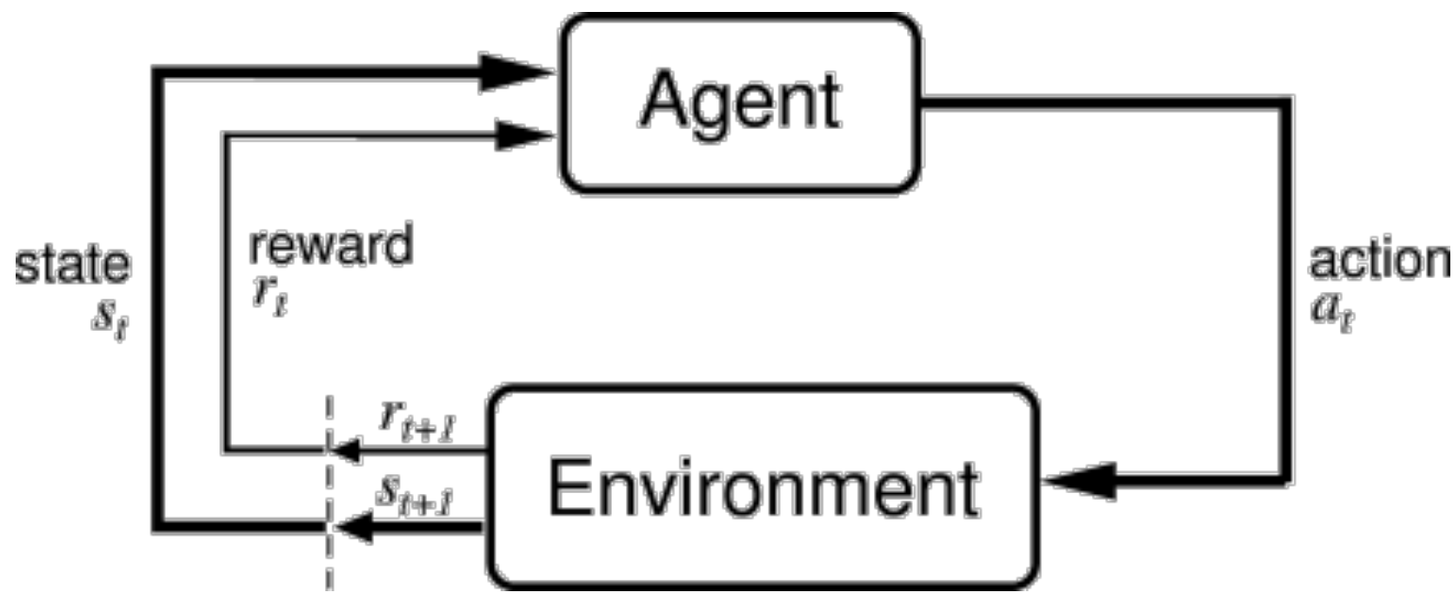
Bilevel Optimization

- More than just GANs!
- Many reinforcement learning architectures have similar issues -
  - Actor-critic methods: Deep Deterministic Policy Gradients
  - Inverse reinforcement learning: Guided Cost Learning
  - Imitation learning: Generative Adversarial Imitation Learning
- All these models have close architectural similarities with GANs

Part III

Reinforcement Learning and Actor-Critic Methods

Reinforcement Learning



Key Idea: *Rather than passively observe data, goal is to* ***pick an action*** *that maximizes the* ***long-term reward*** *when only the short term reward can be observed at each time step*

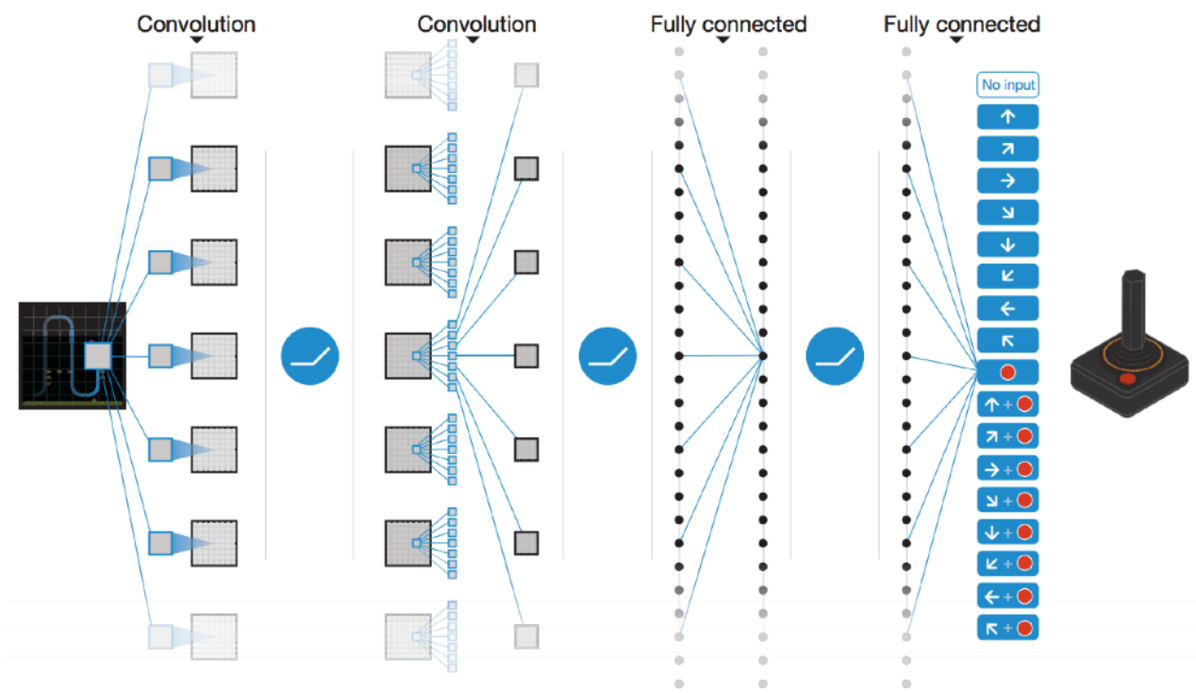
Policy: *The probability distribution over actions given a state*

Value: *The expected long-term reward of following a policy from a given state*

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} p(s'|s,a) V^*(s') \right]$$

Reinforcement Learning

Human-level performance on Atari Games



V. Mnih, K. Kavukcuoglu, D. Silver et al (2015)

Reinforcement Learning

Policy Gradient Theorem

Expected Discounted Reward

$$\eta(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_0\right]$$

Policy Gradient

$$\frac{\partial \eta}{\partial \theta} = \sum_s d^{\pi_\theta}(s) \sum_a \frac{\partial \pi_\theta(s,a)}{\partial \theta} Q^{\pi_\theta}(s,a)$$

Action-State Value Function

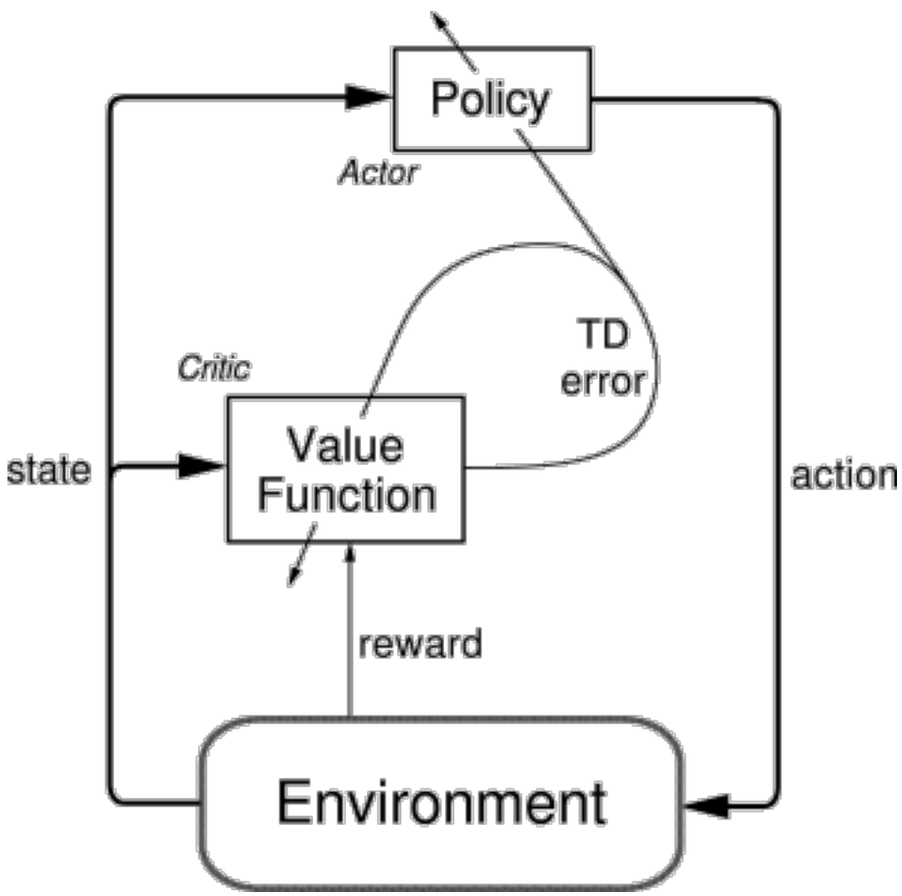
$$Q^{\pi_\theta}(s,a) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \middle| s_t = s, a_t = a\right]$$

Expected Discounted State Distribution

$$d^{\pi_\theta}(s) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\theta}\left[s_t | s_0\right]$$

Actor-Critic Methods

Classic Policy Gradient Algorithms



Sutton and Barto (1998)

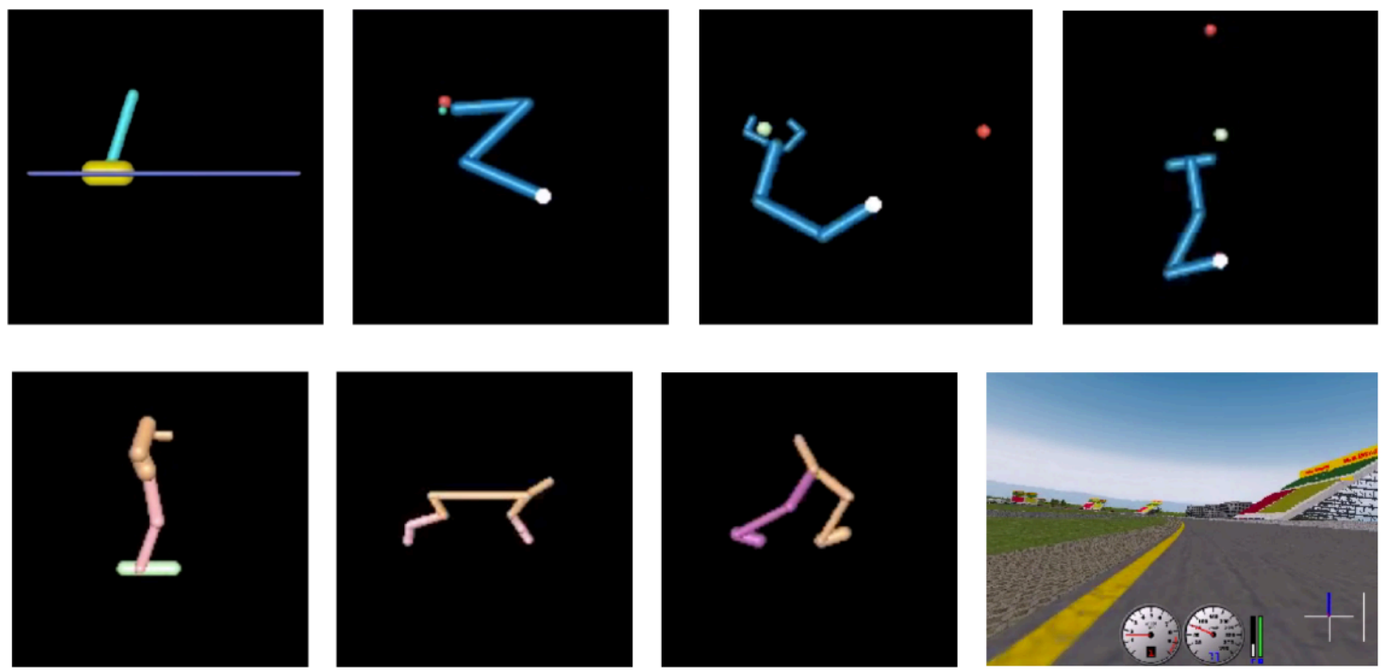
REINFORCE (*R. J. Williams 1992*):
Use real returns to approximate Q(s,a)
-Unbiased but high variance
-Can approximate Q(s,a) to reduce variance
-Scale policy gradient update by TD error

Actor-Critic Methods:
Directly plug Q(s,a) estimate into policy gradient
-Simple but often unstable if Q(s,a) not accurate - highly biased gradients
-Can provide unbiased gradients if Q is linear function of gradients of policy *(R. S. Sutton et al 1999)*

Actor-Critic Methods

Deep Deterministic Policy Gradients

$$\frac{\partial \eta}{\partial \theta} = \mathbb{E}_{\pi_\theta}\left[\frac{\partial Q_\phi}{\partial a}\frac{\partial \pi_\theta}{\partial \theta}\right]$$

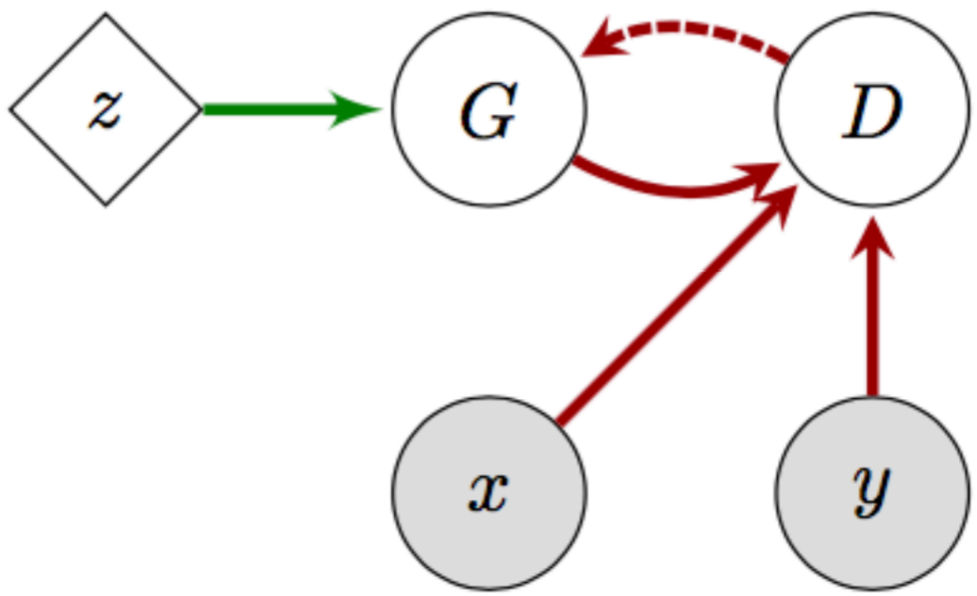


-Deterministic policy network
-Alternate between updates to Q by TD learning and policy by gradient descent
-Policy network only sees gradients of TD error wrt action, not rewards
-Similar architecture to NFQCA, SVG(0)

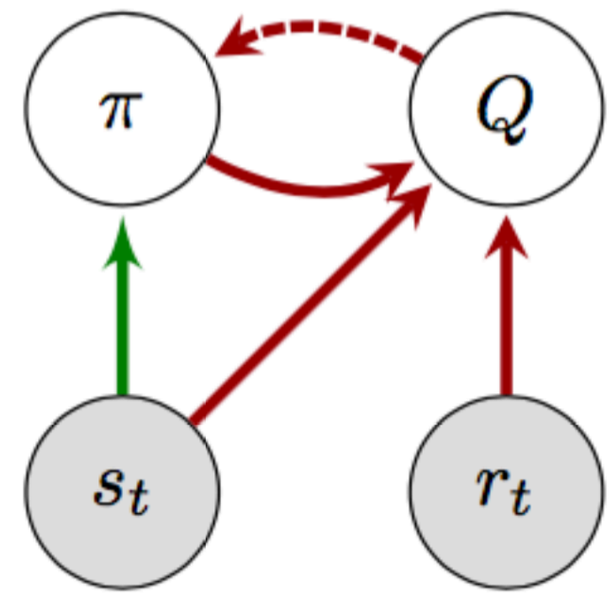
D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller (2014)
T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra (2015)

GANs as Actor-Critic

Generative Adversarial Networks



Deep Deterministic Policy Gradients



$$\mathcal{L}_Q(\theta,\phi) = \mathbb{E}_{\pi_\theta}\left[||\mathbb{E}_{\pi_\theta}[r_t + \gamma Q_\phi(s_{t+1}, \pi_\theta(s_{t+1}))] - Q_\phi(s_t, \pi_\theta(s_t))||^2\right]$$

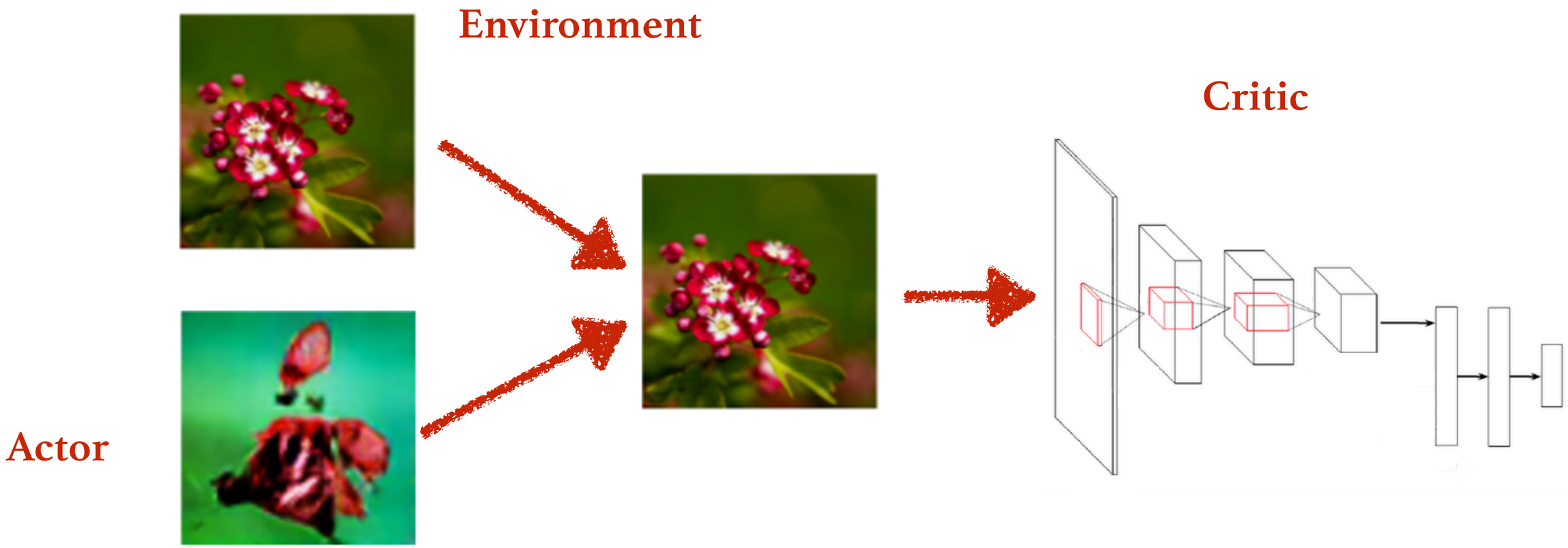
$$\mathcal{L}_\pi(\theta,\phi) = -\mathbb{E}_{\pi_\theta}\left[Q_\phi(s_0, \pi_\theta(s_0))\right]$$

$$\mathcal{L}_D(\theta,\phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$$

$$\mathcal{L}_G(\theta,\phi) = -\mathbb{E}_{q_\phi(\mathbf{x})}[\log(D_\theta(\mathbf{x}))]$$

D. Pfau and O. Vinyals (2016)

GANs as Actor-Critic

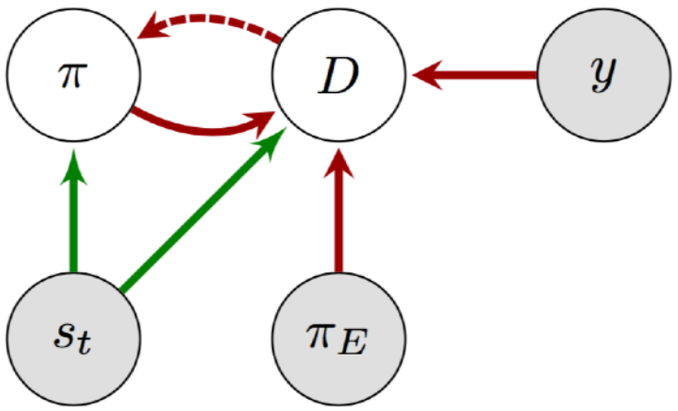


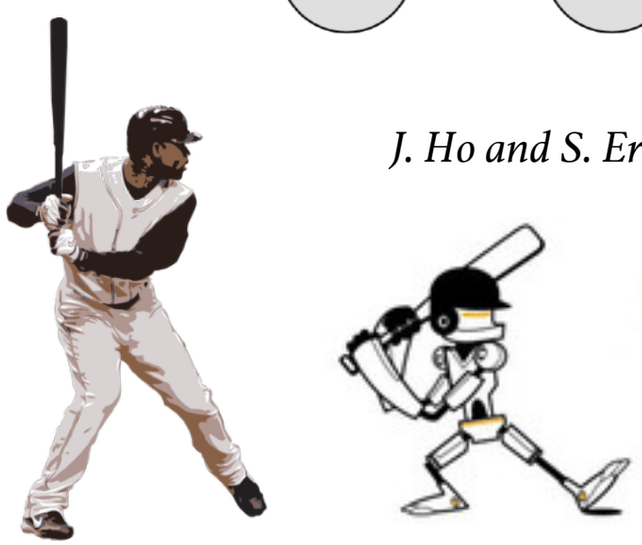
- Actor cannot see state - otherwise it could just copy environment
- Reward is not a function of action - actor and critic become adversarial

D. Pfau and O. Vinyals (2016)

*Tri*level Problems

Generative Adversarial Imitation Learning

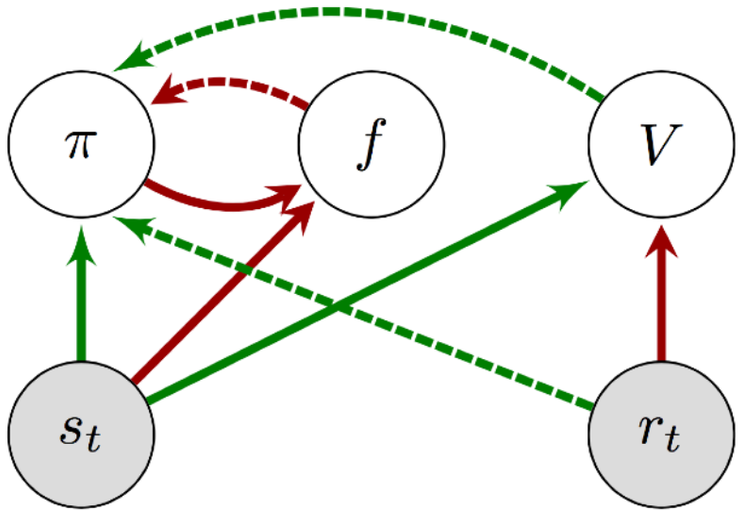




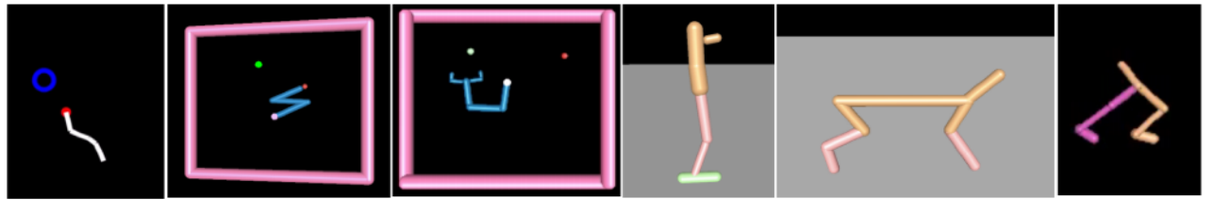
J. Ho and S. Ermon (2016)

- Replace generator with policy that acts on environment
- Can optimize policy directly (eg with TRPO) or ***add actor-critic inside,*** making it trilevel optimization problem

Stochastic Value Gradients



N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, Y. Tassa (2015)



- Learn policy, value function ***and model*** simultaneously
- Policy update combines 3 gradients simultaneously

Part IV

Stabilizing Multilevel Optimization

Stabilizing Strategies

Generative Adversarial Networks

- Batch normalization
- Historical averaging (*Salimans et al 2016*)
- Minibatch discrimination (*Salimans et al 2016*)
- Instance noise (*Sønderby et al 2016*)
- Weight clipping or approximate Wasserstein loss (*Arjovsky et al 2017*)
- Unrolling SGD (*Metz et al 2017*)
- Consensus optimization (*Mescheder et al 2017*)
- Symplectic Gradient Adjustment (*Balduzzi et al 2018*)

Actor-Critic and Inverse RL

- Compatibility (*Sutton et al 1999*)
- Polyak averaging (*Konda and Tsitsiklis 2003*)
- Entropy regularization
- Target networks (*Lillicrap et al 2015*) n/a
- Replay buffers (*Lillicrap et al 2015*)

Fixing the GAN Loss

Generative Adversarial Networks

Discriminator Loss $\mathcal{L}_D(\theta, \phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$

Generator Loss $\mathcal{L}_G(\theta, \phi) = -\mathbb{E}_{q_\phi(\mathbf{x})}[\log(D_\theta(\mathbf{x}))]$

$$\min_\phi \mathcal{L}_G(\theta^*, \phi)$$

$$\theta^*(\phi) = \arg\min_\theta \mathcal{L}_D(\theta, \phi)$$

Can we directly optimize wrt the optimal value?

Fixing the GAN Loss

Generative Adversarial Networks

Discriminator Loss $\mathcal{L}_D(\theta, \phi) = \mathbb{E}_{p^*(\mathbf{x})}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{x})}[\log(1 - D_\theta(\mathbf{x}))]$

Generator Loss $\mathcal{L}_G(\theta, \phi) = -\mathbb{E}_{q_\phi(\mathbf{x})}[\log(D_\theta(\mathbf{x}))]$

$$\min_\phi \mathcal{L}_G(\theta^*, \phi)$$

$$\theta^*(\phi) = \arg\min_\theta \mathcal{L}_D(\theta, \phi)$$

Approximate with a few steps of SGD!

Unrolled Optimization

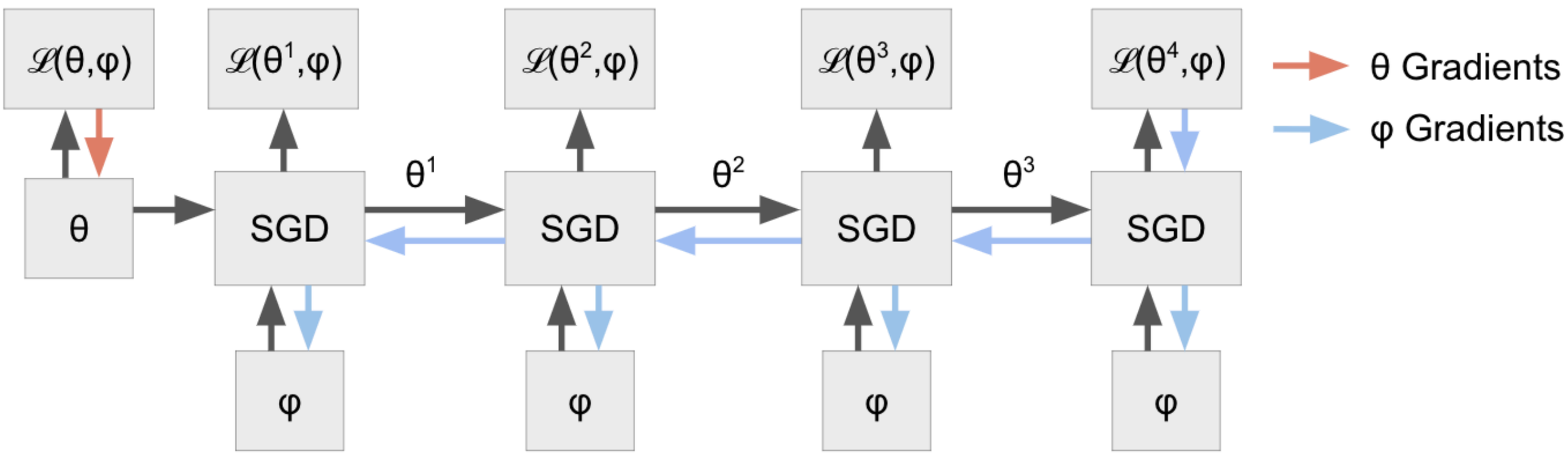
$$\theta^*(\phi) \approx \theta^K(\phi, \theta)$$

$$\theta^K(\phi, \theta) = \theta^{K-1}(\phi, \theta) - \epsilon \nabla_\theta \mathcal{L}_D(\theta^{K-1}, \phi)$$

Unrolled GANs

$$\phi \leftarrow \phi - \epsilon \nabla_\phi \mathcal{L}_G(\theta^K, \phi)$$

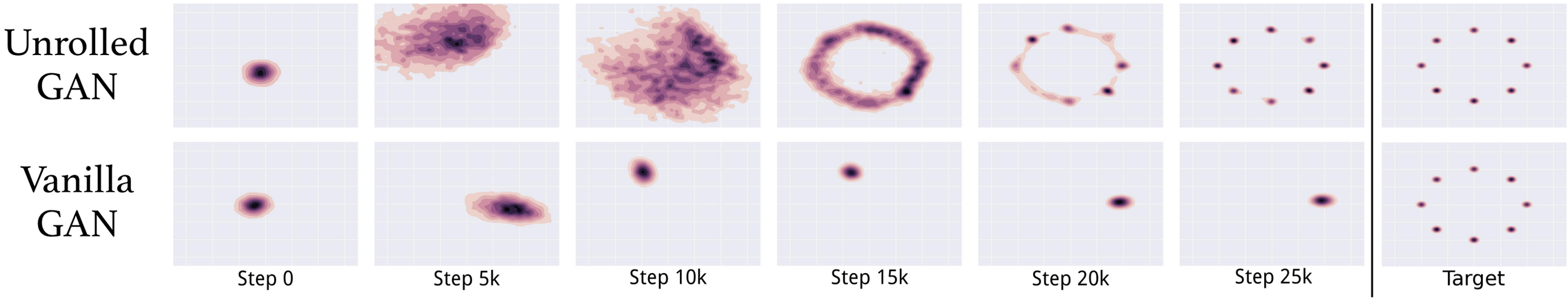
$$\theta \leftarrow \theta - \epsilon \nabla_\theta \mathcal{L}_D(\theta, \phi)$$



L. Metz, B. Poole, D. Pfau and J. Sohl-Dickstein (2017)

https://github.com/poolio/unrolled_gan

Mode Collapse



Recurrent Generator

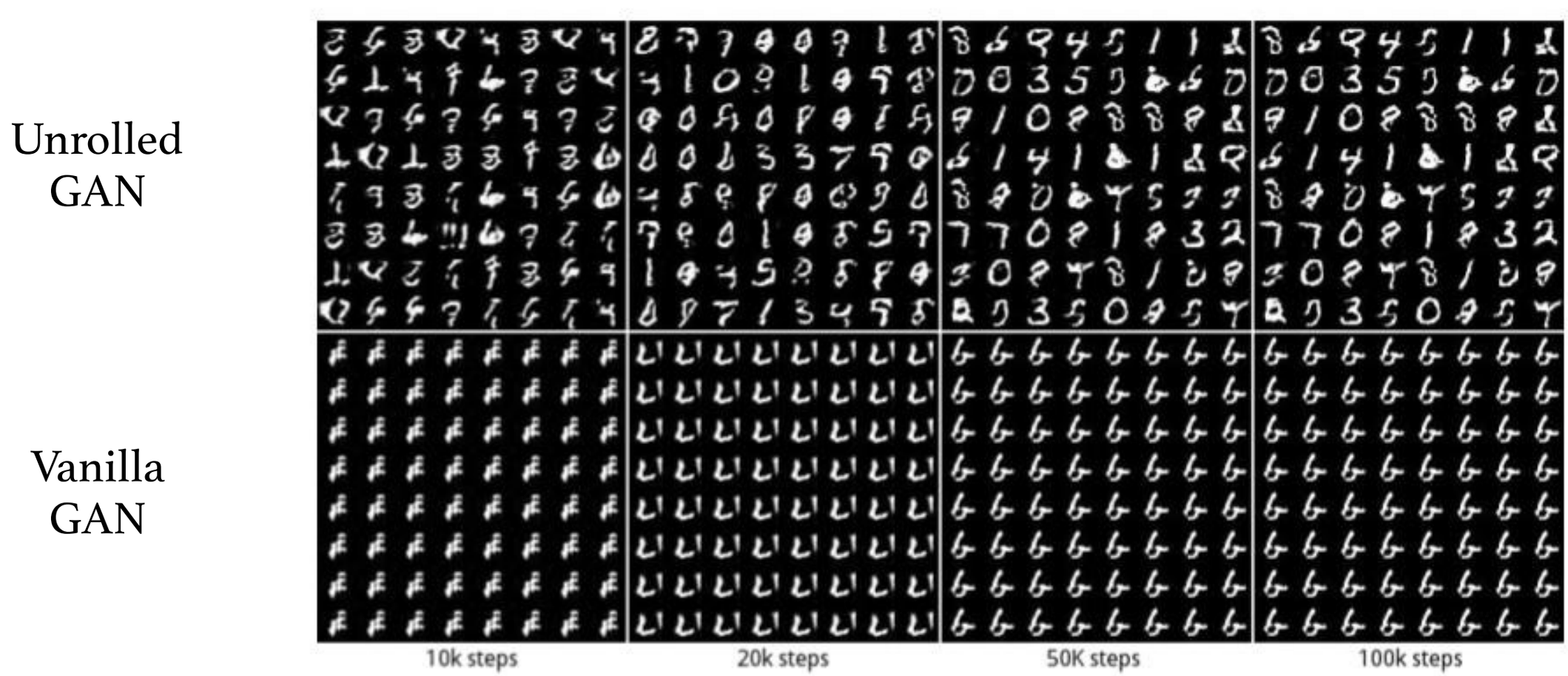
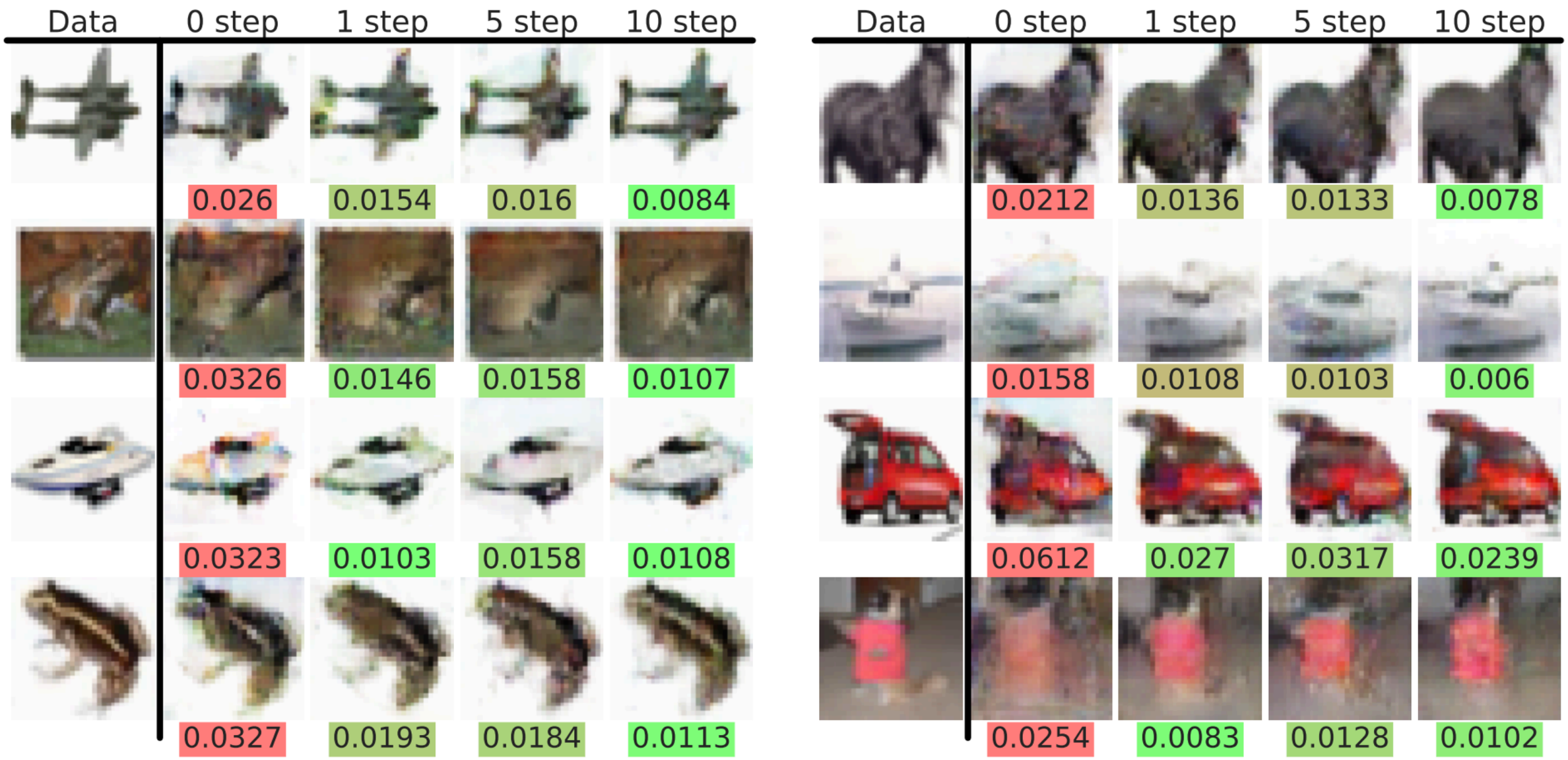
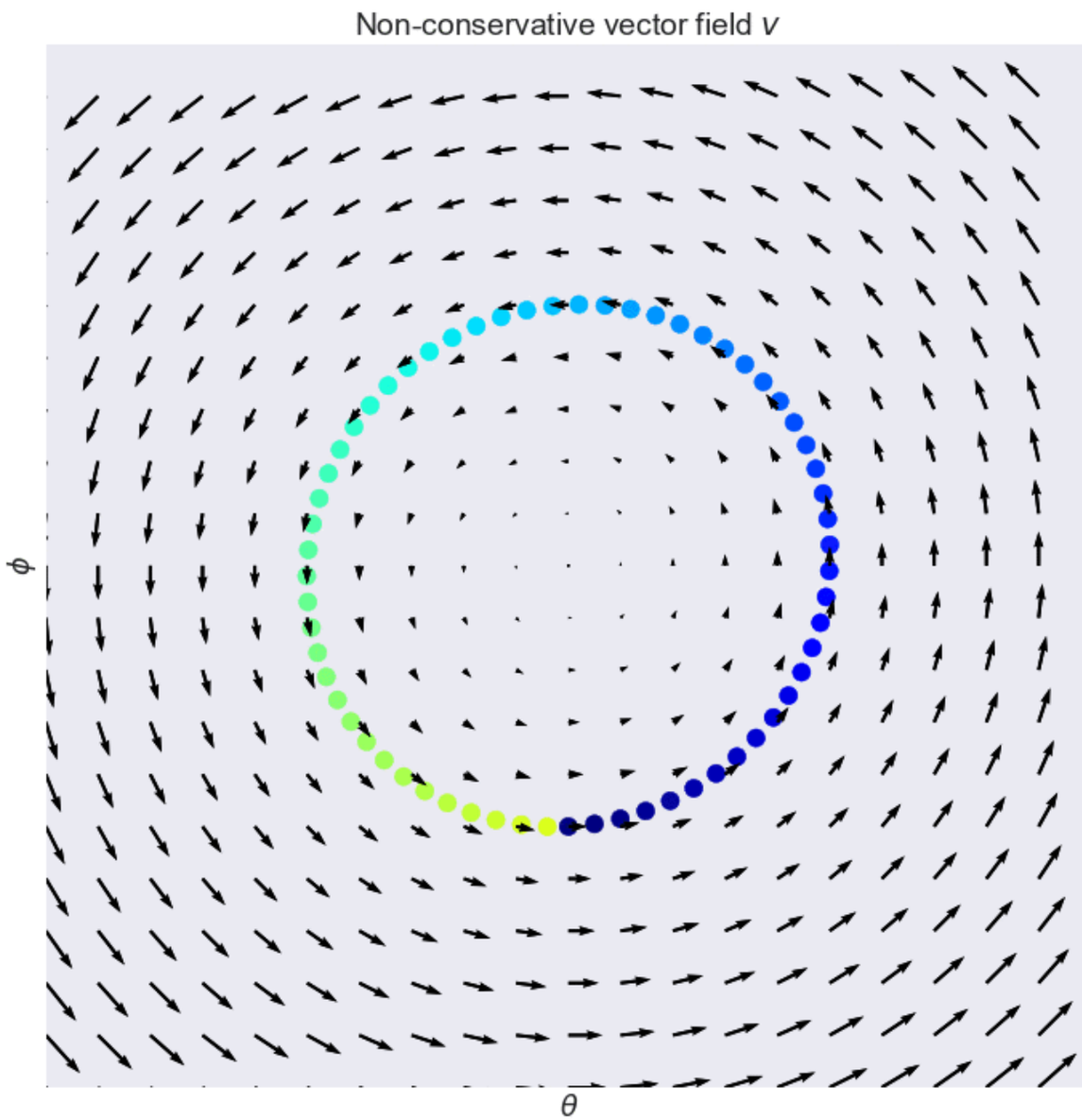


Image Reconstruction



Consensus Optimization

L. Mescheder, S. Nowozin, A. Geiger (2017)

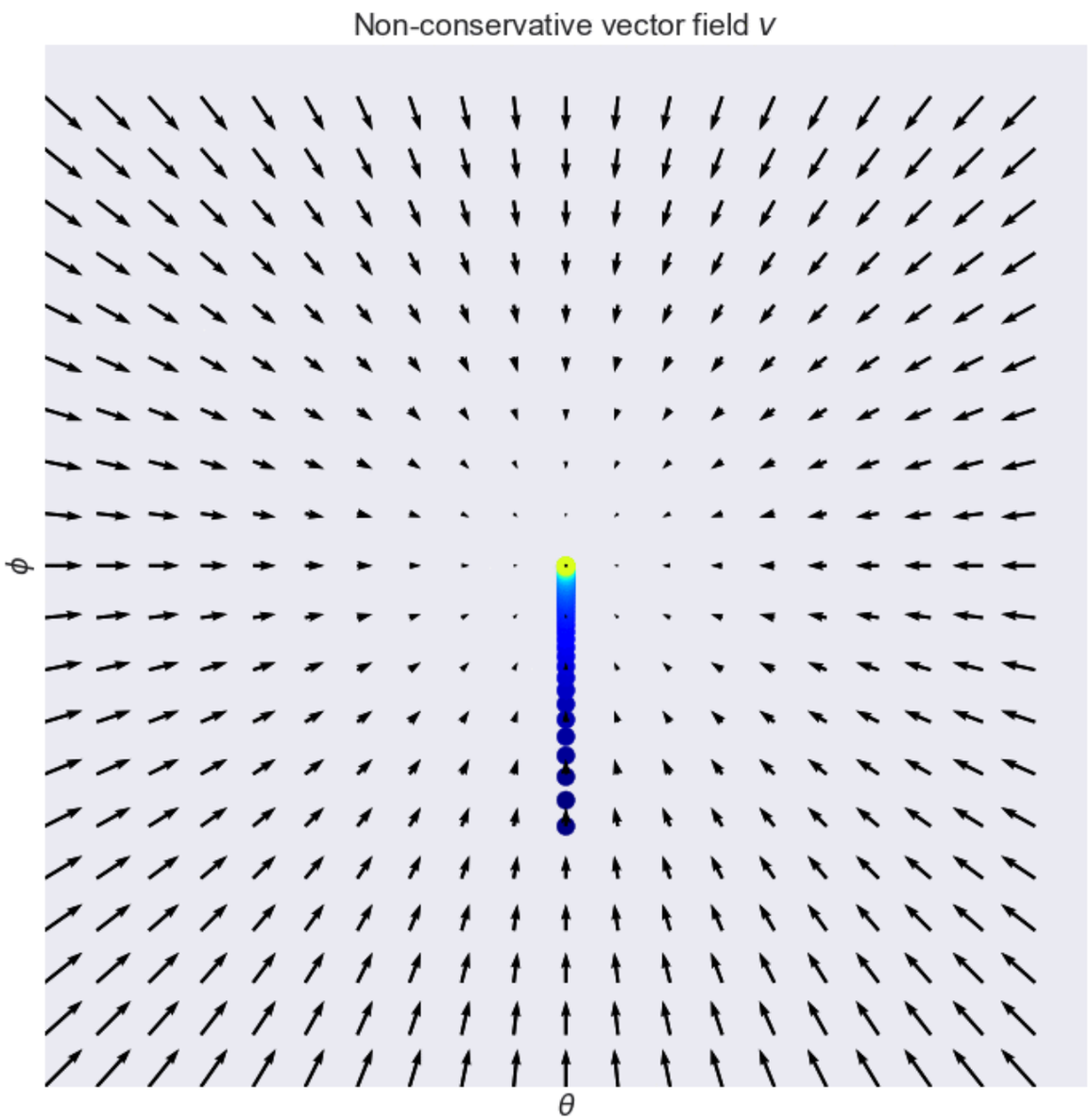


$$\vec{v}(\phi, \theta) = \begin{pmatrix} \nabla_\phi F(\theta, \phi) \\ \nabla_\theta f(\theta, \phi) \end{pmatrix}$$

Courtesy Ferenc Huszar

Consensus Optimization

L. Mescheder, S. Nowozin, A. Geiger (2017)



$$\vec{v}(\phi, \theta) = \begin{pmatrix} \nabla_\phi F(\theta, \phi) \\ \nabla_\theta f(\theta, \phi) \end{pmatrix}$$

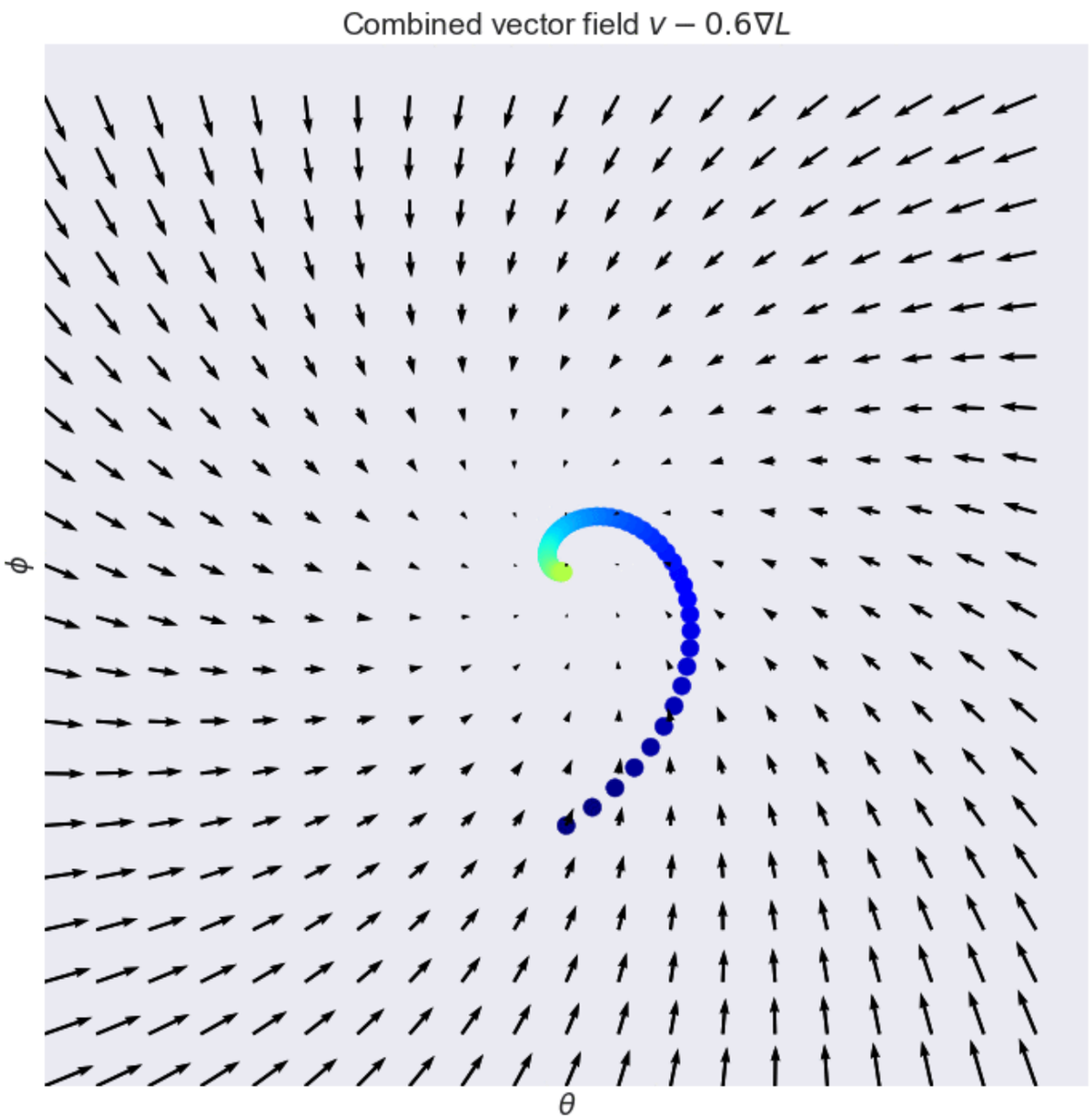
$$L(\phi, \theta) = \frac{1}{2}||\vec{v}||^2$$

All extrema become minima!

Courtesy Ferenc Huszar

Consensus Optimization

L. Mescheder, S. Nowozin, A. Geiger (2017)



$$\vec{v}(\phi, \theta) = \begin{pmatrix} \nabla_\phi F(\theta, \phi) \\ \nabla_\theta f(\theta, \phi) \end{pmatrix}$$

$$L(\phi, \theta) = \frac{1}{2}||\vec{v}||^2$$

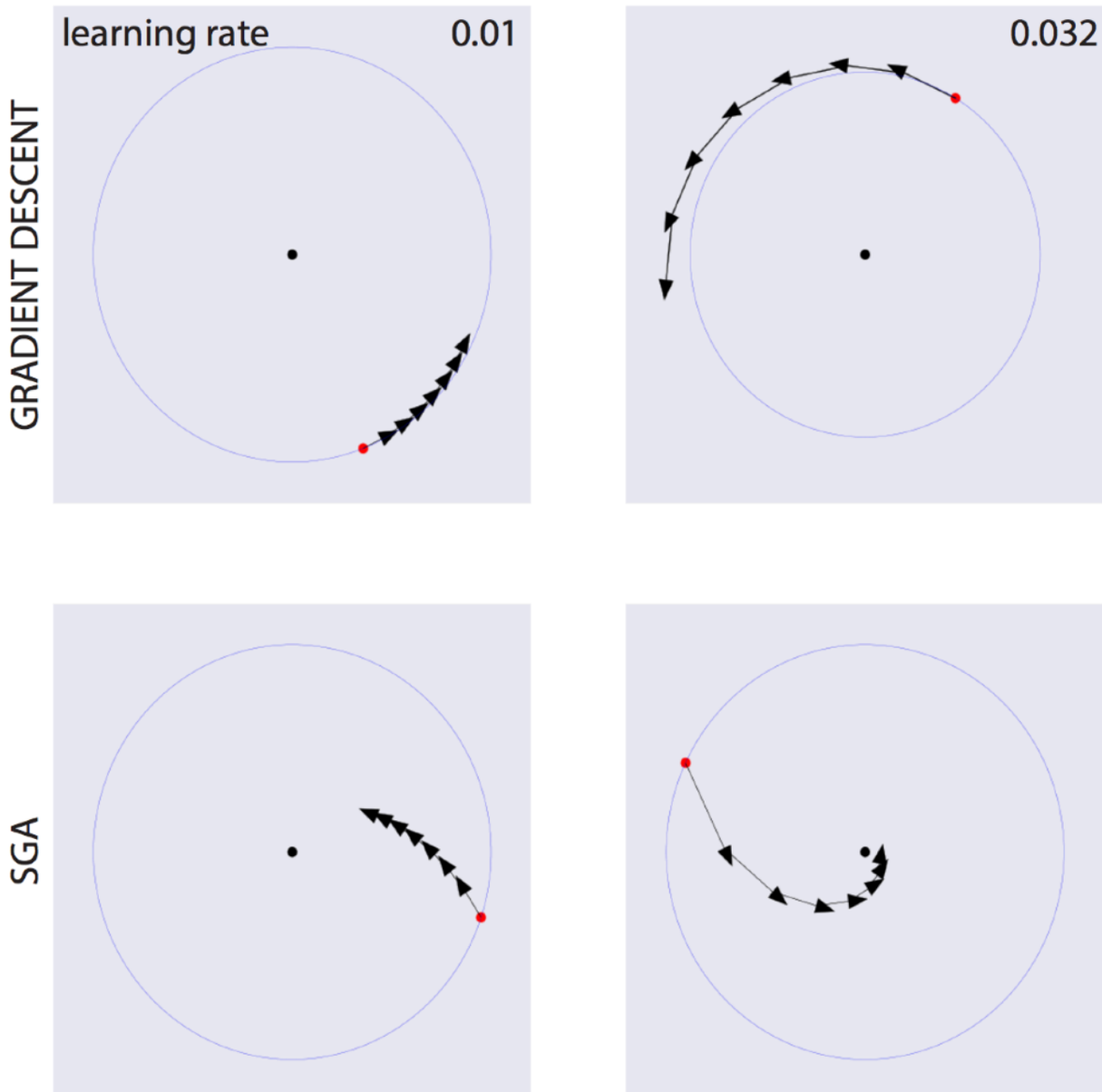
Update:

$$-\vec{v} - \gamma \nabla L$$

Courtesy Ferenc Huszar

Symplectic Gradient Adjustment

D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls and T. Graepel (2018)



$$\vec{v}(\phi,\theta) = \begin{pmatrix} \nabla_\phi F(\theta,\phi) \\ \nabla_\theta f(\theta,\phi) \end{pmatrix}$$

$$H(\phi,\theta) = \begin{pmatrix} \nabla_\phi \nabla_\phi F & \nabla_\phi \nabla_\theta F \\ \nabla_\phi \nabla_\phi f & \nabla_\phi \nabla_\theta f \end{pmatrix}$$

$$A = \frac{1}{2}\left(H - H^T\right)$$

Update:

$$-\vec{v} - \gamma A^T \vec{v}$$

Conclusions

- GANs are an instance of *multilevel* deep learning, which also include actor-critic methods, imitation learning and multi-agent games.

- Multilevel optimization is a significantly harder problem than normal optimization, and new algorithms are being developed to handle this complexity

- Unrolled optimization, consensus optimization and symplectic gradient adjustment are all recently-developed algorithms for multilevel problems

- If you just want a generative model, there are better models than GANs. If you can fix multilevel optimization, the horizon of possible models becomes much larger than just GANs.

Conclusions

Many thanks to the IML Organizers, as well as David Balduzzi, Ferenc Huszar, Shakir Mohamed, Oriol Vinyals, Josh Merel, Luke Metz, Ben Poole and Jascha Sohl-Dickstein

DeepMind

pfau@google.com

@pfau

References

- I. Goodfellow et al (2014). Generative Adversarial Networks. *Advances in Neural Information Processing Systems.*
- T. Karras, T. Aila, S. Laine, J. Lehtinen (2018). Progressive Growing of GANs for Improved Quality, Stability and Variation. *6th International Conference on Learning Representations.*
- F. Carminati et al (2017). Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics. *NIPS Workshop on Deep Learning for Physical Science.*
- D. Ackley, G. Hinton and T. Sejnowski (1985). A Learning Algorithm for Boltzmann Machines. *Cognitive Science.*
- A. van den Oord, N. Kalchbrenner, K. Kavukcuoglu (2016). Pixel Recurrent Neural Networks. *Proceedings of the 33rd International Conference on Machine Learning*
- A. van den Oord et al (2016). WaveNet: A Generative Model for Raw Audio. arXiv: 1609.03499
- D. Rezende and S. Mohamed (2015). Variational Inference with Normalizing Flows. *Proceedings of the 32nd International Conference on Machine Learning*
- P. Dayan, G. Hinton, R. Neal and R. Zemel (1995). The Helmholtz Machine. *Neural Computation.*
- L. Dinh, J. Sohl-Dickstein, S. Bengio (2017). Density Estimation Using Real NVP. *5th International Conference on Learning Representations.*
- D. Rezende and S. Mohamed (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *Proceedings of the 31st International Conference on Machine Learning.*
- D. Kingma and M. Welling (2014). Auto-encoding Variational Bayes. *Second International Conference on Learning Representations.*
- Y. Burda, R. Grosse and R. Salakhutdinov (2016). Importance Weighted Autoencoders. *4th International Conference on Learning Representations.*
- A. Mnih and D. Rezende (2016). Variational Inference for Monte Carlo Objectives. *Proceedings of the 33rd International Conference on Machine Learning.*
- G. Wayne, C. Hung, D. Amos et al (2018). Unsupervised Predictive Memory in a Goal-Directed Agent. arXiv: 1803.10760.
- B. Colson, P. Marcotte and G. Savard (2005). Bilevel Programming: A Survey. *4OR-Q J Oper Res.* 3: 87-107
- P. Hansen, B. Jaumard and G. Savard (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing.* 13(5): 1194-1217
- R. Sutton and A. Barto (1998). *Reinforcement Learning*
- V. Mnih, K. Kavukcuoglu, D. Silver et al (2015). Human-level control through deep reinforcement learning. *Nature* 518: 529-533
- R. S. Sutton et al (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems.* 99: 1057-1063
- D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller (2014). Deterministic Policy Gradient Algorithms. *Proceedings of the 31st International Conference on Machine Learning.*

References

- T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra (2015). Continuous control with deep reinforcement learning. arXiv: 1509:02971
- J. Ho and S. Ermon (2016). Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems*.
- N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, Y. Tassa (2015). Learning Continuous Control Policies by Stochastic Value Gradients. *Advances in Neural Information Processing Systems*. arXiv: 1510:09142
- C. K. Sønderby, J. Caballero, L. Theis, W. Shi and F. Huszár (2016). Amortised MAP Inference for Image Super-Resolution. arXiv: 1610:04490
- V. R. Konda and J. N. Tsitsiklis (2003). On Actor-Critic Algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143-1166.
- D. Pfau and O. Vinyals (2016). Connecting Generative Adversarial Networks and Actor-Critic Methods. *NIPS Workshop on Adversarial Training*. arXiv: 1610:01945
- C. Finn, P. Christiano, P. Abbeel, S. Levine (2016). A Connection Between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models. *NIPS Workshop on Adversarial Training*. arXiv: 1611:03852
- T. Salimans et al (2016). Improved Techniques for Training GANs. arXiv: 1606:03498
- L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein (2017). Unrolled Generative Adversarial Networks. *5th International Conference on Learning Representations*. arXiv: 1611:02163
- M. Arjovsky, S. Chintala and L. Bottou (2017). Wasserstein GANs. arXiv: 1701.07875
- L. Mescheder, S. Nowozin, A. Geiger (2017). The Numerics of GANs. arXiv: 1705.10461
- D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, T. Graepel (2018). The Mechanics of n-Player Differentiable Games. arXiv: 1802.05642