LEAD: Least Action Dynamics for Min-Max Optimization




Reyhane Askari* Amartya Mitra* Guillaume Lajoie Ioannis Mitliagkas

Mila Université de Montréal UC Riverside University of California

Min-Max Optimization

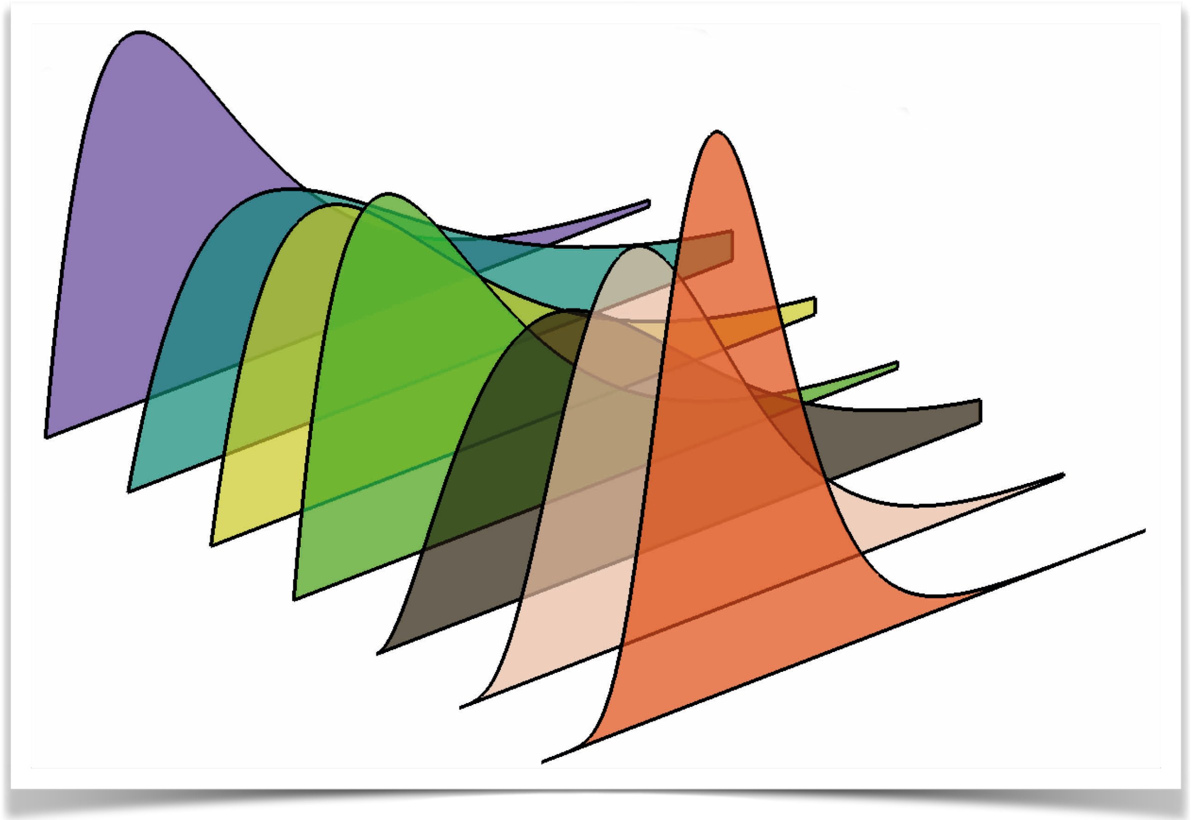
- Optimize with respect to two players,

$$x^* \in \underset{x}{\operatorname{argmin}} \; f(x, y^*)$$
$$y^* \in \underset{y}{\operatorname{argmax}} \; g(x^*, y)$$

- Examples include,

GANs

Out-of-distribution generalization



Adversarial Training

Single-objective Optimization

- Minimization on a smooth, differentiable loss,

$$x^* \in \underset{x}{\text{argmin}} \; f(x)$$

- Gradient Descent,

Vector-field

$$v(x) := \nabla_x f(x)$$

$$x_{t+1} = x_t - \eta v(x_t)$$

- Update operator,

Gets x_t
Returns x_{t+1}

$$F_\eta(x_t) := x_t - \eta v(x_t)$$

Stable fixed point of F_η is a local **minimum**.

$$F_\eta(x^*) = x^*$$

Min-Max Optimization

Single-objective Optimization

- Minimization on a smooth, differentiable loss,

$$x^* \in \underset{x}{\text{argmin}} \; f(x)$$

- Gradient Descent,

Vector-field

$$v(x) := \nabla_x f(x)$$

$$x_{t+1} = x_t - \eta v(x_t)$$

- Update operator,

Gets x_t
Returns x_{t+1}

$$F_\eta(x_t) := x_t - \eta v(x_t)$$

Stable fixed point of F_η is a local **minimum**.

$$F_\eta(x^*) = x^*$$

Min-Max Optimization

- Optimize $f(x, y)$ with respect to two players,

$$x^* \in \underset{x}{\text{argmin}} \; f(x, y^*)$$
$$y^* \in \underset{y}{\text{argmax}} \; f(x^*, y)$$

- Gradient Descent-Ascent,

$$v(x, y) := \begin{bmatrix} \nabla_x f(x, y) \\ -\nabla_y f(x, y) \end{bmatrix}$$

- Update operator on gradient descent-ascent,

$$F_\eta(x_t, y_t) := \begin{bmatrix} x_t \\ y_t \end{bmatrix} - \eta v(x_t, y_t)$$

Stable fixed point of F_η is a local **Nash equilibrium**.

F_η is convergent if the spectral radius of ∇F_η at the fixed point, is smaller than one.

The largest eigenvalue

F_η is convergent if the spectral radius of ∇F_η at the fixed point, is smaller than one.

The largest eigenvalue

$$\nabla F_\eta = I - \eta \nabla v$$

F_η is convergent if the spectral radius of ∇F_η at the fixed point, is smaller than one.

The largest eigenvalue

$$\nabla F_\eta = I - \eta \nabla v$$

Single-objective Optimization

∇v is the Hessian

Symmetric

All real eigenvalues

Well-known dynamics

Min-Max Optimization

F_η is convergent if the spectral radius of ∇F_η at the fixed point, is smaller than one.

The largest eigenvalue

$$\nabla F_\eta = I - \eta \nabla v$$

Single-objective Optimization

↓

∇v is the Hessian

↓

Symmetric

↓

All real eigenvalues

↓

Well-known dynamics

Min-Max Optimization

↓

∇v is a Jacobian

↓

Might be non-symmetric

↓

Imaginary eigenvalues

↓

Rotational Dynamics

Motivation

- Understand game optimization from a physical perspective.

- Model games as a physical system and manipulate the system to curb the rotational dynamics.

Physics Perspective, recipe from minimization

Common intuition in optimization:
Polyak momentum (heavy ball) is described as an object moving inside a potential well and is accelerating while moving towards the minimum.

Physics Perspective, recipe from minimization

Common intuition in optimization:
Polyak momentum (heavy ball) is described as an object moving inside a potential well and is accelerating while moving towards the minimum.

Model our system using all the forces $\vec{F} = m\ a$ → Path of the particle in continuous time → Replace the net force with the gradients → $-\nabla f(x) = m\ddot{x}$

Discretize the continuous-time system → $x_{k+1} = x_k + \beta(x_k - x_{k-1}) - \eta \nabla_x f(x_k)$ → Polyak momentum!

Physics Perspective, intuition from minimization

Model min-max optimization as a physical system with rotational dynamics → Introduce relevant forces in the system to curb the rotations → Determine the equations of motion

Continuous time Equation of motion using all the forces in the system → Discretize the continuous-time system → New optimization algorithm!

Step1 - Model Min-Max optimization as a physical system

Continuous-time dynamics of GDA with momentum in a 2-D plane

$$\ddot{x} = -\nabla_x f(x, y)$$

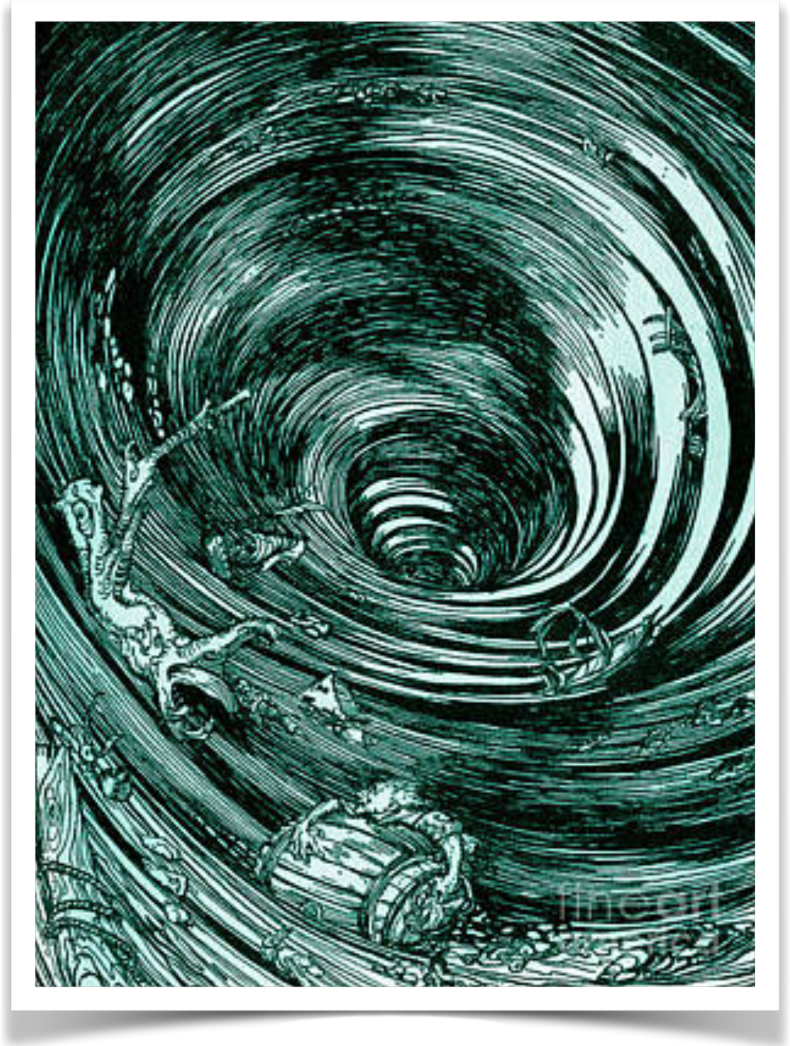
$$\ddot{y} = \nabla_y f(x, y)$$

Step1 - Model Min-Max optimization as a physical system

Continuous-time dynamics of GDA with momentum in a 2-D plane

$$\ddot{x} = -\nabla_x f(x, y) \qquad \xrightarrow[\text{nature}]{\text{similar force in}} \qquad m\ddot{x} = F_{vortex} = -\nabla_x f(x, y)$$

$$\ddot{y} = \nabla_y f(x, y) \qquad \qquad m\ddot{y} = F_{vortex} = \nabla_y f(x, y)$$

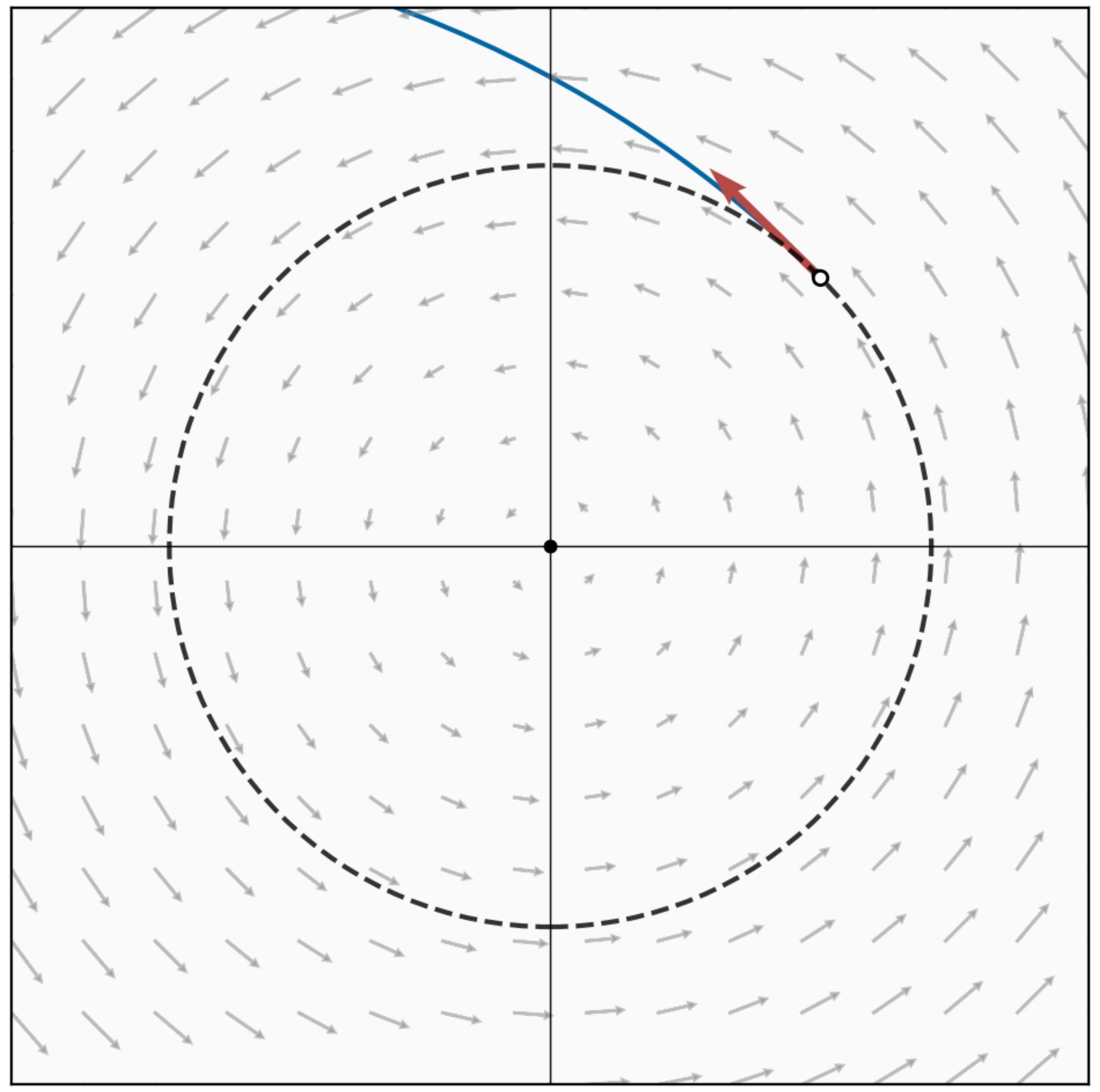


Step1 - Model Min-Max optimization as a physical system

Continuous-time dynamics of GDA with momentum in a 2-D plane

$$m\ \ddot{x} = F_{vortex} = -\nabla_x f(x, y)$$

$$m\ \ddot{y} = F_{vortex} = \nabla_y f(x, y)$$



Step1 - Model Min-Max optimization as a physical system

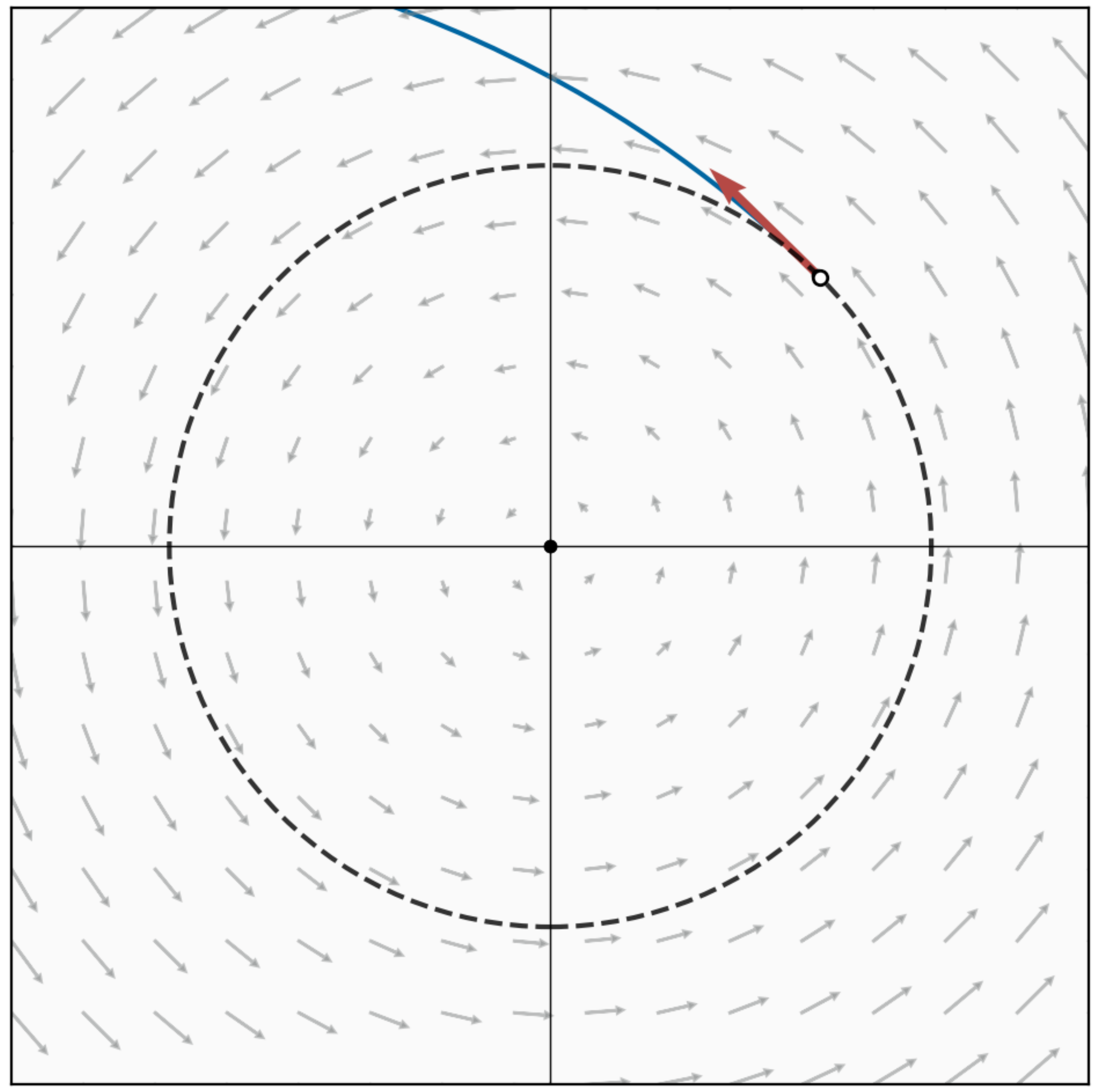
Continuous-time dynamics of GDA with momentum in a 2-D plane

$$m\ \ddot{x} = F_{vortex} = -\nabla_x f(x, y)$$

$$m\ \ddot{y} = F_{vortex} = \nabla_y f(x, y)$$

Vortex Force:

1. Exhibit rotational dynamics
2. Increases the velocity



Step 2: Introduce relevant forces to curb the rotations

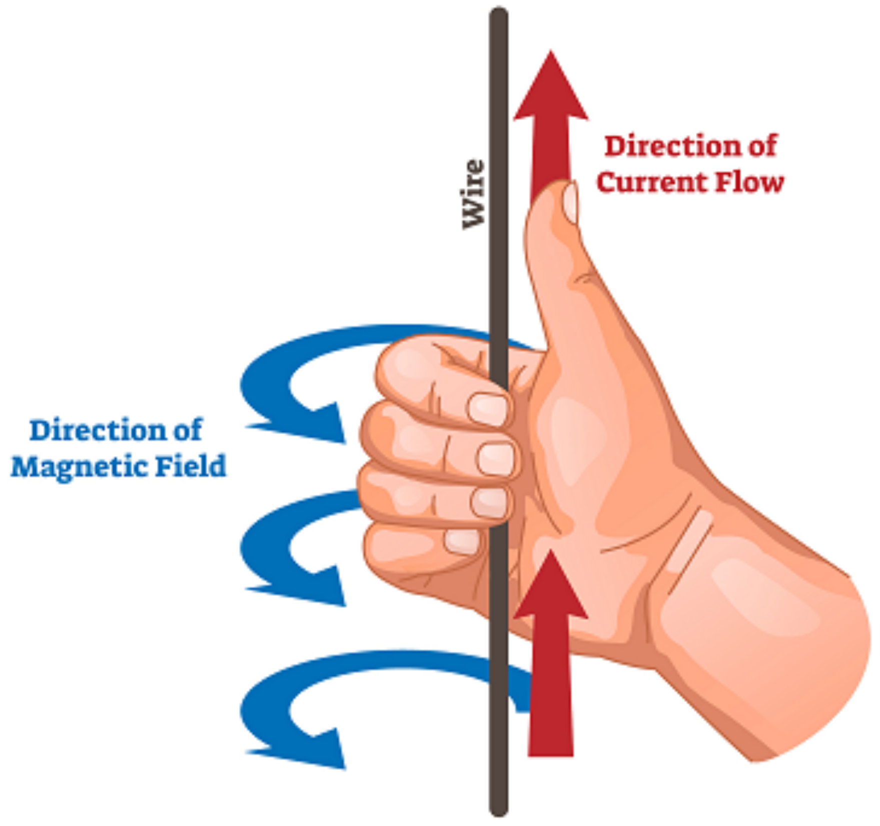
Step 2: Introduce relevant forces to curb the rotations

Magnetic force is also known to produce rotations,

$$m\ \ddot{x} = {\color{red}F_{vortex}} + {\color{green}F_{magnetic}} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y}$$

$$m\ \ddot{y} = {\color{red}F_{vortex}} + {\color{green}F_{magnetic}} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x}$$

charge



https://www.pasco.com/products/guides/right-hand-rule

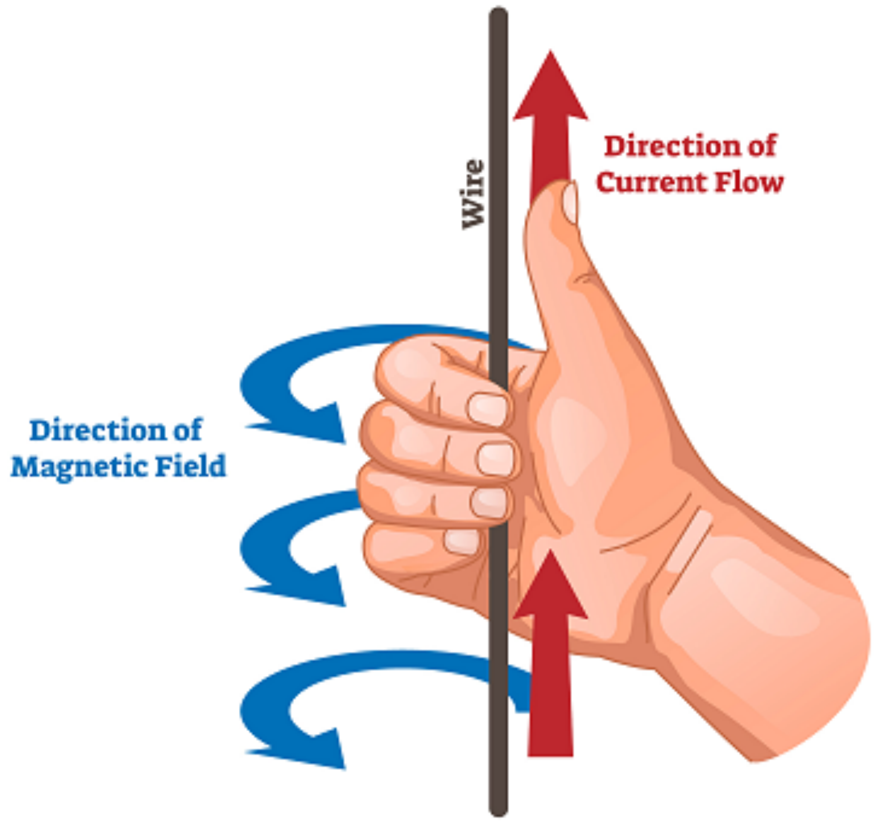
Step 2: Introduce relevant forces to curb the rotations

Magnetic force is also known to produce rotations,

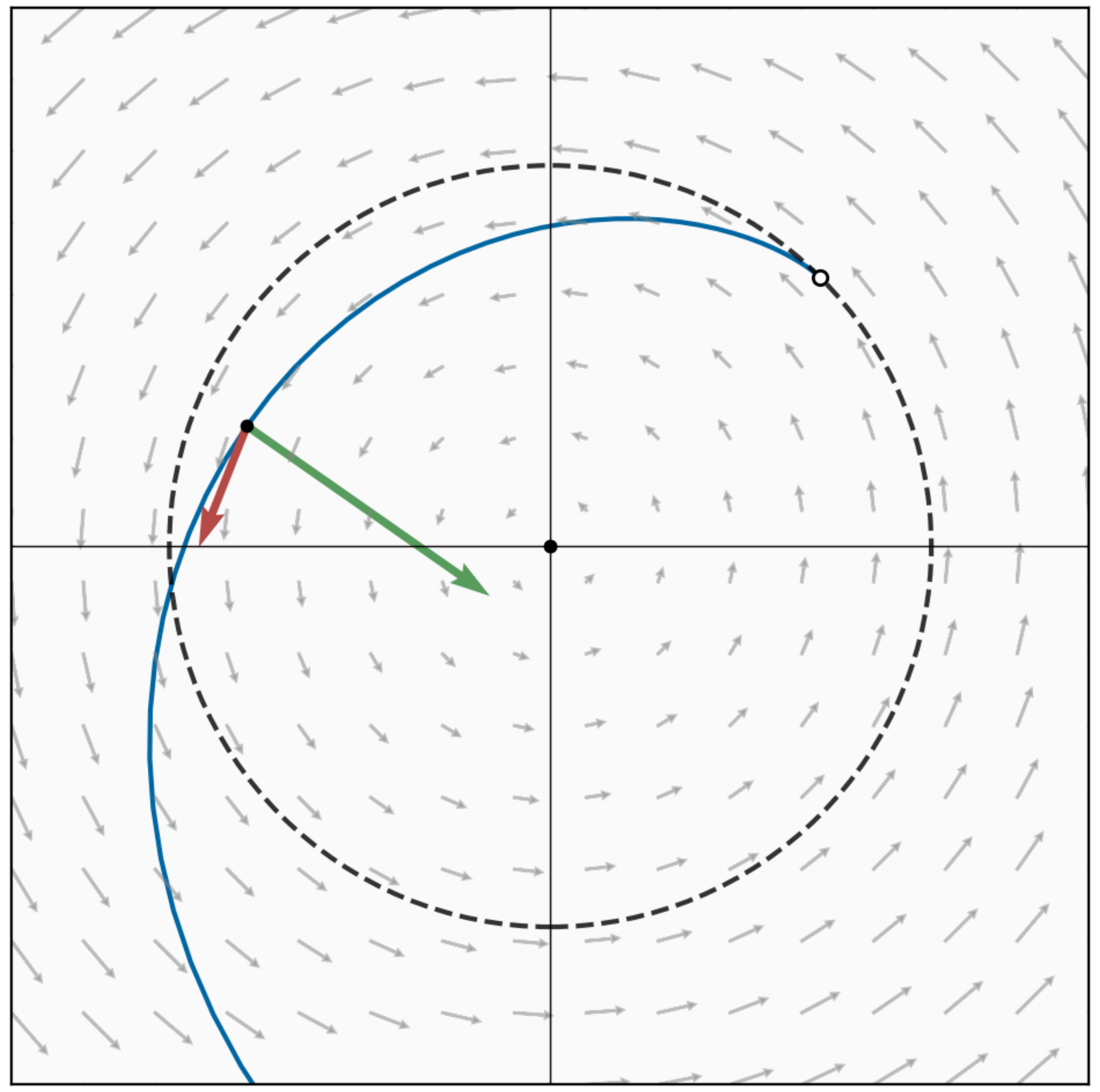
$$m\,\ddot{x} = {\color{red}F_{vortex}} + {\color{green}F_{magnetic}} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y}$$

$$m\,\ddot{y} = {\color{red}F_{vortex}} + {\color{green}F_{magnetic}} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x}$$

charge



https://www.pasco.com/products/guides/right-hand-rule



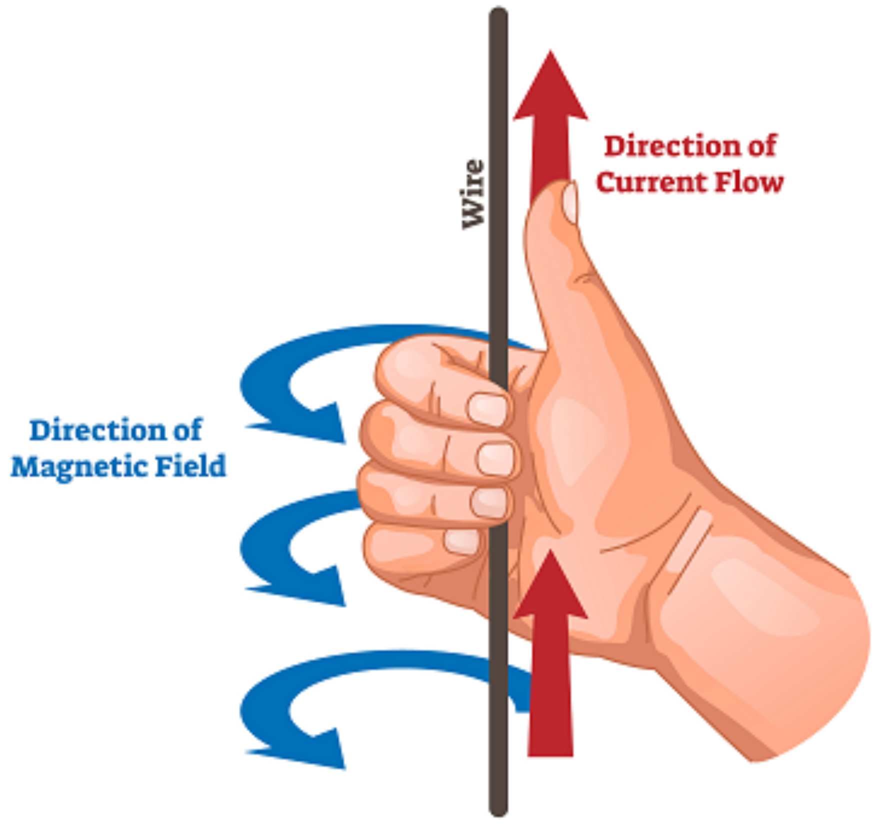
Step 2: Introduce relevant forces to curb the rotations

Magnetic force is also known to produce rotations,

$$m\,\ddot{x} = F_{vortex} + F_{magnetic} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y}$$

$$m\,\ddot{y} = F_{vortex} + F_{magnetic} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x}$$

charge

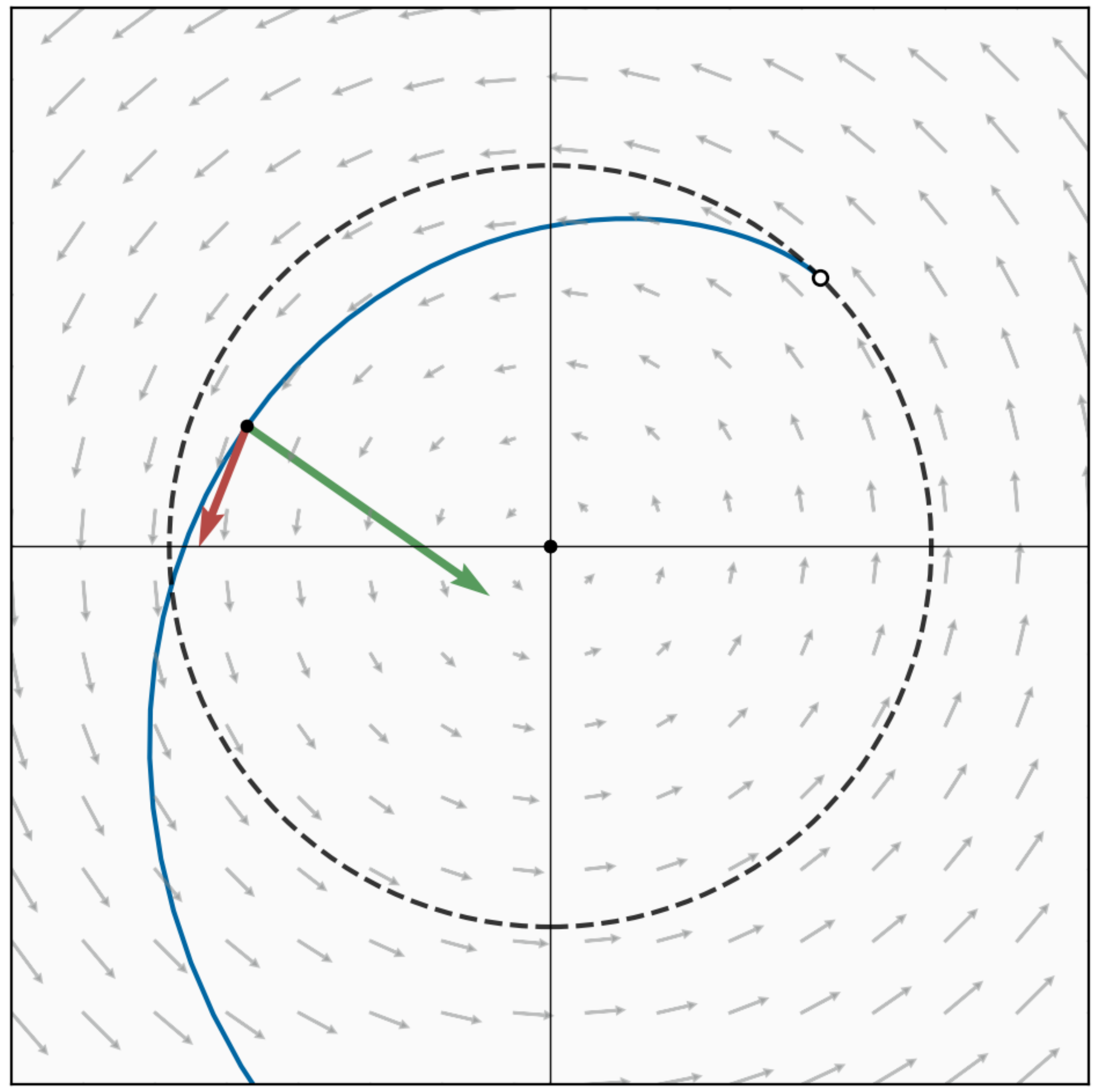


https://www.pasco.com/products/guides/right-hand-rule

Vortex Force:

1. Exhibit rotational dynamics ✓
2. Increases the velocity

But, magnetic force only changes the direction of movement and not the velocity.



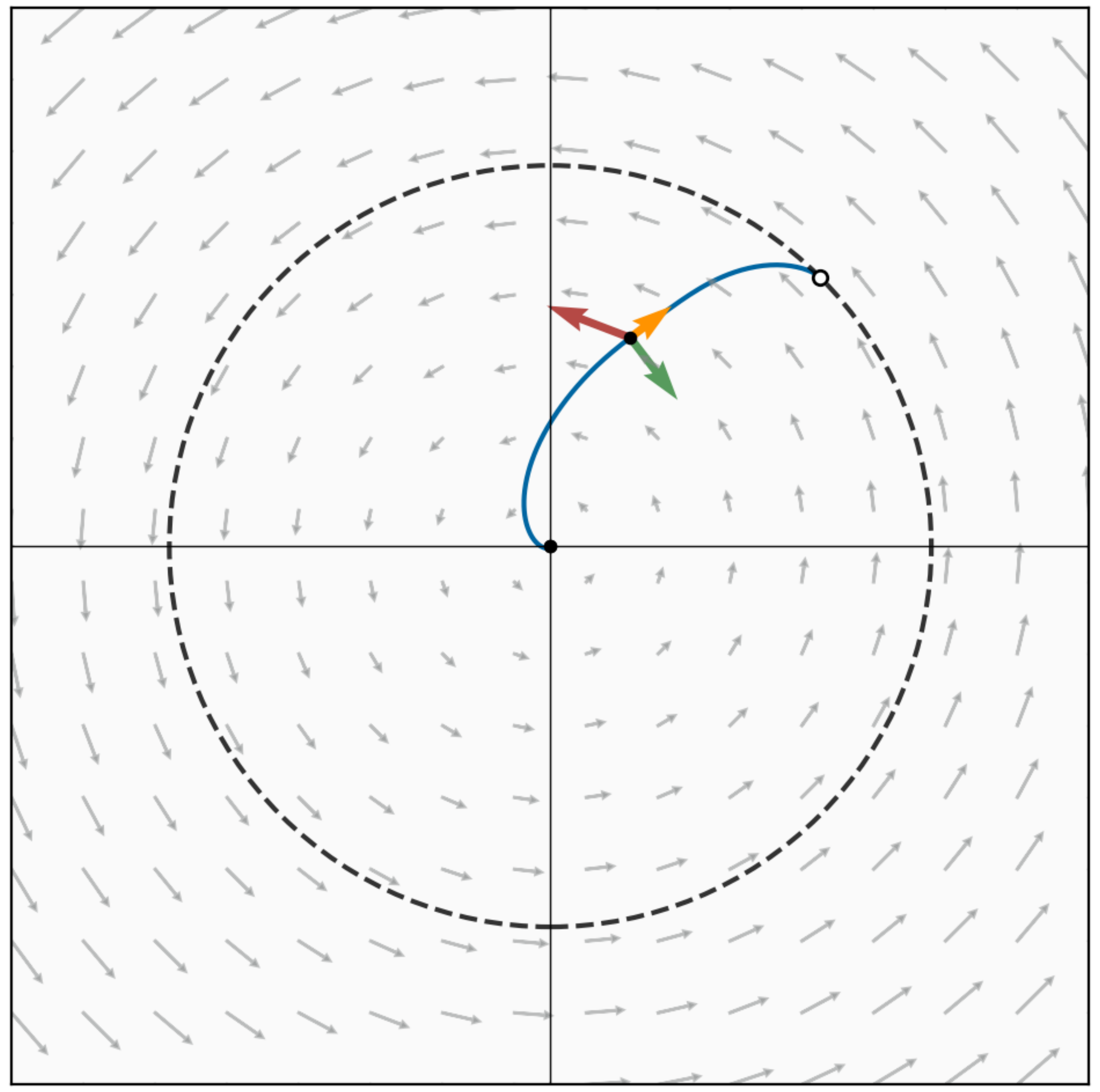
Step 3: Introduce relevant forces to reduce the velocity

Need to add some form of dissipation; simplest form is friction

$$m\,\ddot{x} = F_{vortex} + F_{magnetic} + F_{friction} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y} - \mu\dot{x}$$

$$m\,\ddot{y} = F_{vortex} + F_{magnetic} + F_{friction} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x} - \mu\dot{y}$$

friction coefficient



Step 3: Introduce relevant forces to reduce the velocity

Need to add some form of dissipation; simplest form is friction

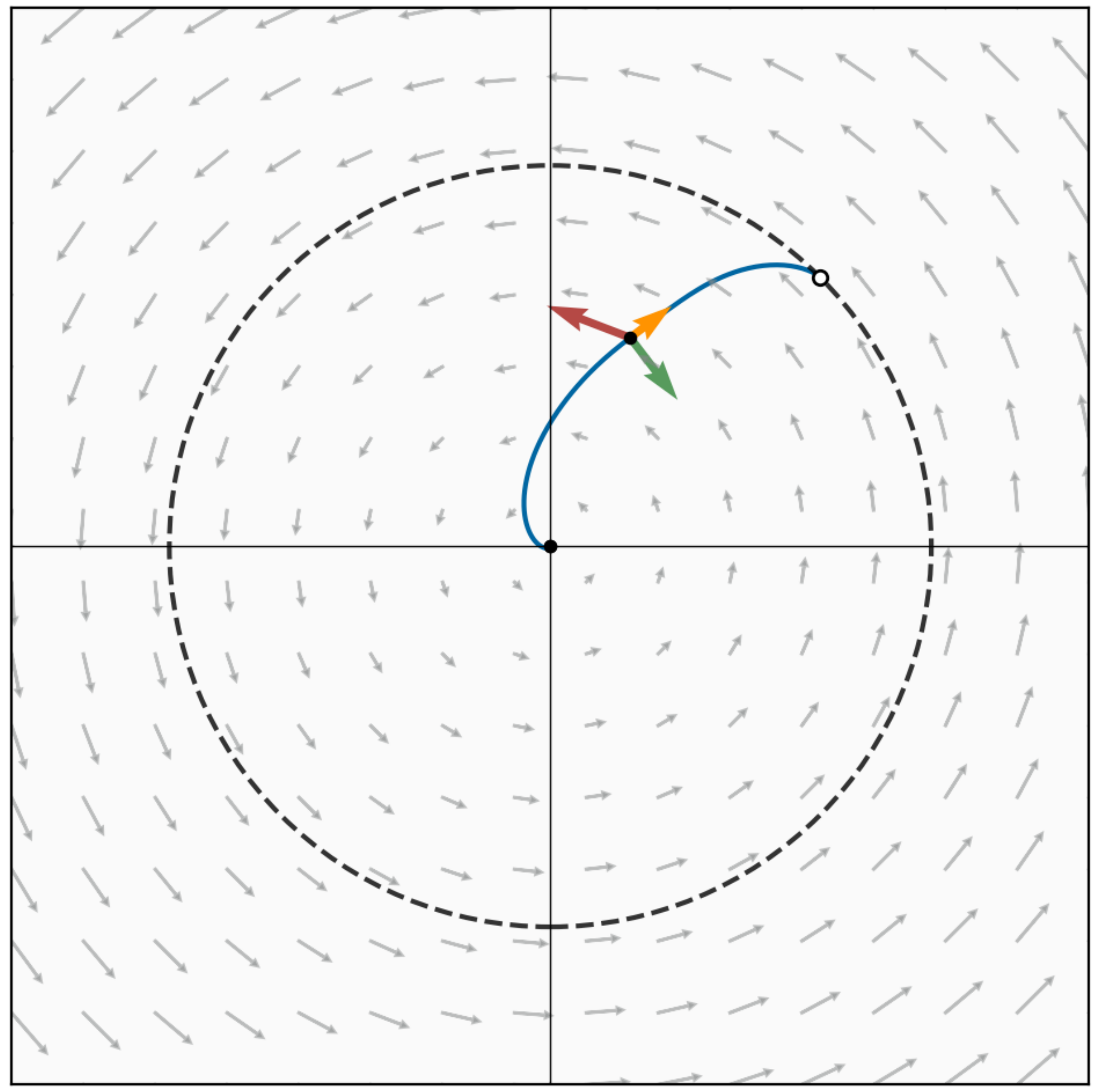
$$m\ \ddot{x} = F_{vortex} + F_{magnetic} + F_{friction} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y} - \mu\dot{x}$$

$$m\ \ddot{y} = F_{vortex} + F_{magnetic} + F_{friction} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x} - \mu\dot{y}$$

friction coefficient

Vortex Force:

1. Exhibit rotational dynamics ✓
2. Increases the velocity ✓



The friction causes the particle to lose speed and converge!

Step 4 - Discretize the system: LEAD

$$m\,\ddot{x} = F_{vortex} + F_{magnetic} + F_{friction} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y} - \mu\dot{x}$$

$$m\,\ddot{y} = F_{vortex} + F_{magnetic} + F_{friction} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x} - \mu\dot{y}$$

Step 4 - Discretize the system: LEAD

$$m\,\ddot{x} = F_{vortex} + F_{magnetic} + F_{friction} = -\nabla_x f(x,y) - 2q(\nabla_{xy} f)\dot{y} - \mu\dot{x}$$

$$m\,\ddot{y} = F_{vortex} + F_{magnetic} + F_{friction} = \nabla_y f(x,y) + 2q(\nabla_{xy} f)\dot{x} - \mu\dot{y}$$

discretization ↓

$$x_{k+1} = x_k + \beta(x_k - x_{k-1}) - \eta\nabla_x f(x_k, y_k) - \alpha\nabla_{xy} f(x_k, y_k)(y_k - y_{k-1})$$

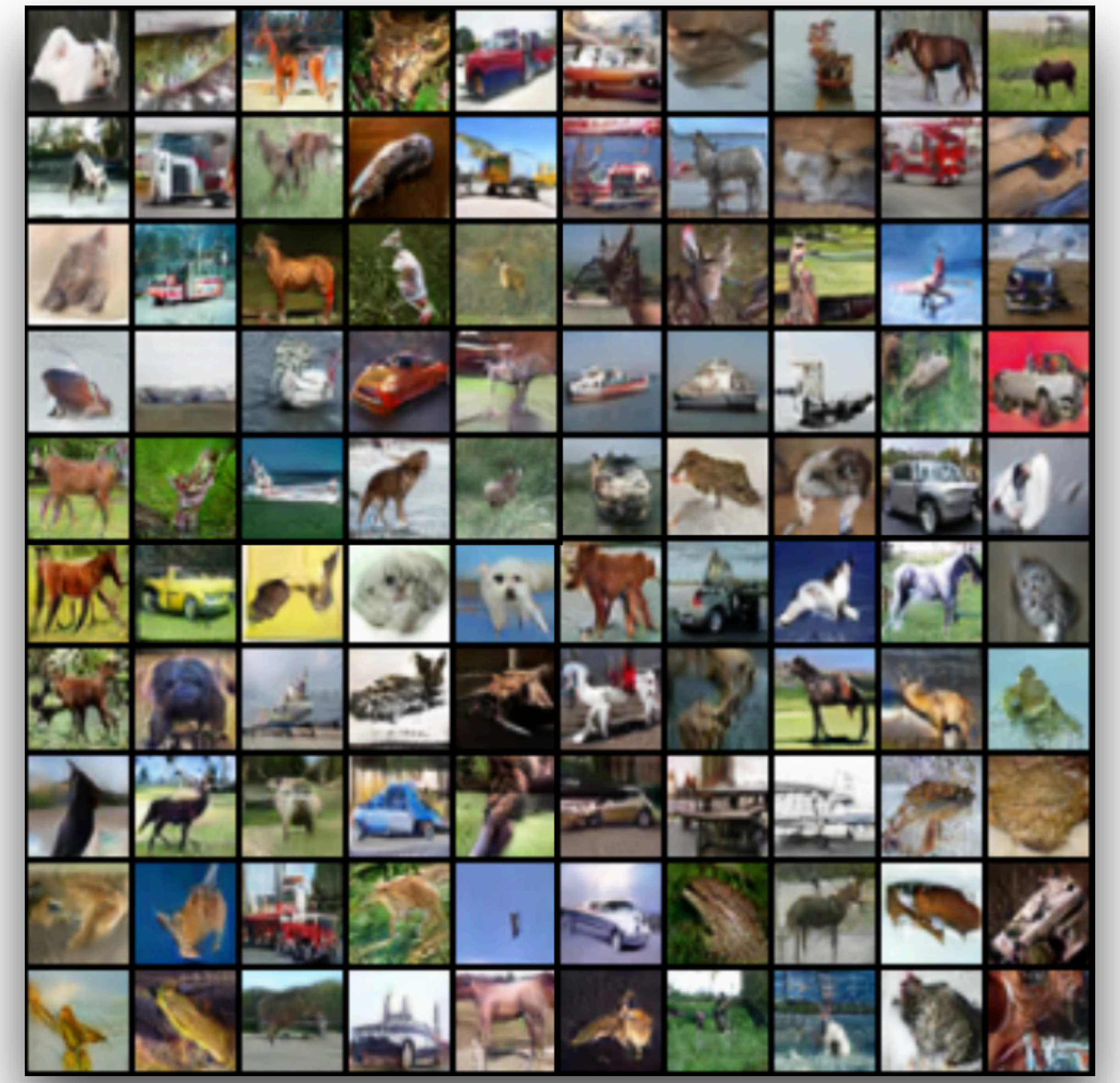
$$y_{k+1} = y_k + \beta(y_k - y_{k-1}) + \eta\nabla_y f(x_k, y_k) + \alpha\nabla_{xy} f(x_k, y_k)(x_k - x_{k-1})$$

LEAD - Experiments

- LEAD outperforms BigGAN in terms of FID with an architecture that is 30 times smaller.

ResNet	FID	IS
LEAD-Adam	10.49 ± 0.20	8.82 ± 0.05
Spectral Normalization*	12.1 ± 0.31	8.58 ± 0.39
ExtraAdam**	16.78 ± 0.21	8.47 ± 0.10
ODE-GAN***	11.85 ± 0.21	8.61 ± 0.06

* SN from Miyato et al, 2018
** ExtraAdam from Gidel et al, 2018
*** ODE-GAN from Qin et al, 2020



Sample Generated output on CIFAR 10 with a ResNet Architecture

Thank you!

LEAD: Least Action Dynamics for Min-Max Optimization

Blogpost: **https://reyhaneaskari.github.io/LEAD.html**

Paper: **https://openreview.net/forum?id=vXSsTYs6ZB**

Code: **https://github.com/ReyhaneAskari/Least_action_dynamics_minmax**