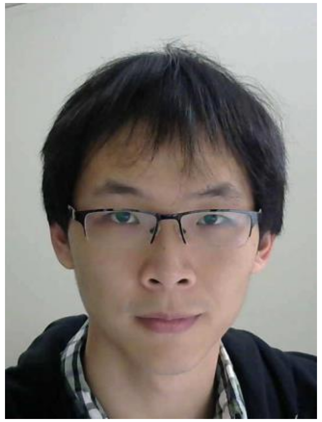
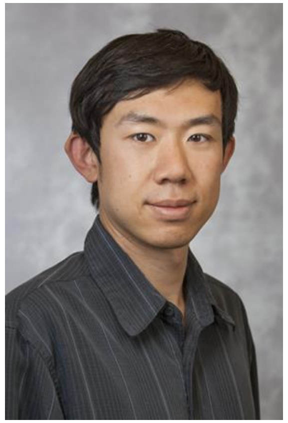
How to Escape Saddle Points Efficiently?

Praneeth Netrapalli

Microsoft Research India





Chi Jin
UC Berkeley

Michael I. Jordan
UC Berkeley

Rong Ge
Duke Univ.

Sham M. Kakade
U Washington

Non-convex optimization

Problem: $\min_{x} f(x)$ $f(\cdot)$: non-convex function

Applications: Neural networks, matrix/tensor factorization, unsupervised learning, …

Status: NP-hard in general

In practice

<u>Popular algorithms</u>

- Gradient descent [Cauchy 1847]
- Accelerated gradient descent [Nesterov 1983]

<u>Question</u>

How do they perform?

In practice

Popular algorithms

- Gradient descent [Cauchy 1847]
- Accelerated gradient descent [Nesterov 1983]

Question

How do they perform?

Answer

Converge to first order stationary points

In practice

Popular algorithms

- Gradient descent [Cauchy 1847]
- Accelerated gradient descent [Nesterov 1983]

Question

How do they perform?

Answer

Converge to first order stationary points

Definition

ϵ-First order stationary point (ϵ-FOSP) : $||\nabla f(x)|| \leq \epsilon$

In practice

Popular algorithms

- Gradient descent [Cauchy 1847]
- Accelerated gradient descent [Nesterov 1983]

Question

How do they perform?

Answer

Converge to first order stationary points

Definition

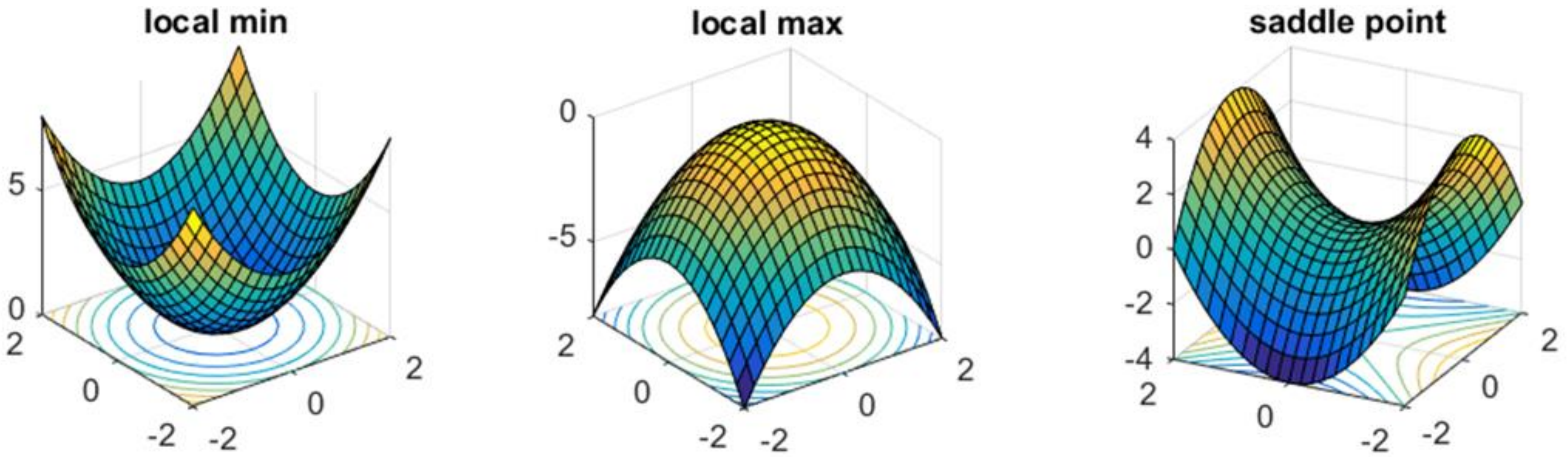
ϵ-First order stationary point (ϵ-FOSP) : $\|\nabla f(x)\| \leq \epsilon$

Concretely

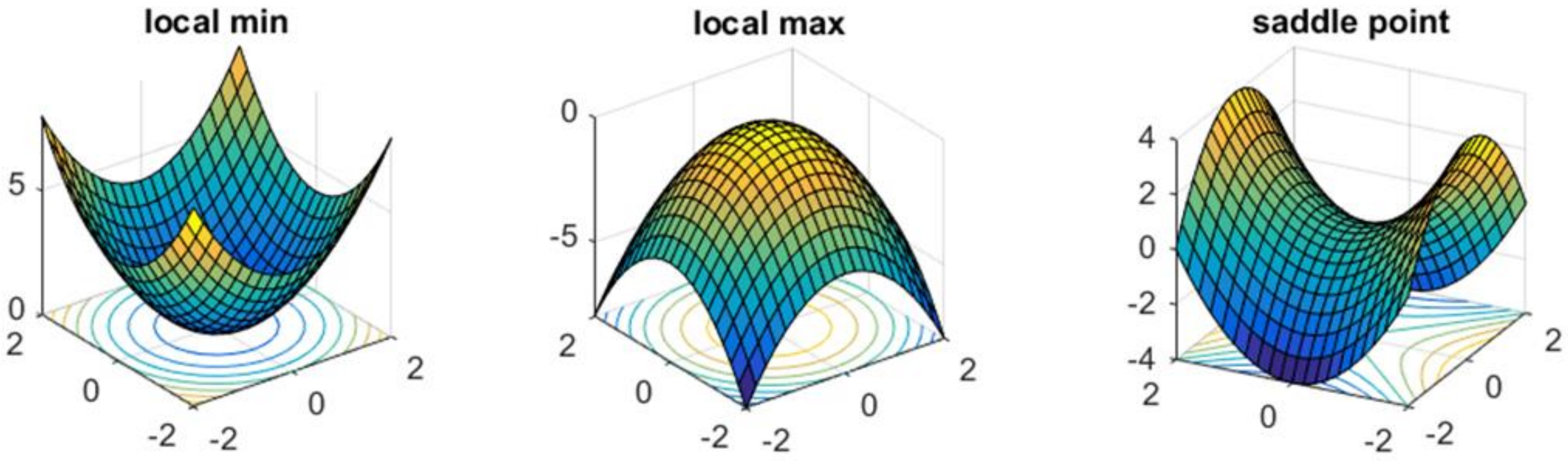
ϵ-FOSP in $O\left(\frac{1}{\epsilon^2}\right)$ iterations

[Folklore, Ghadimi & Lan 2013]

How do FOSPs look like?



How do FOSPs look like?

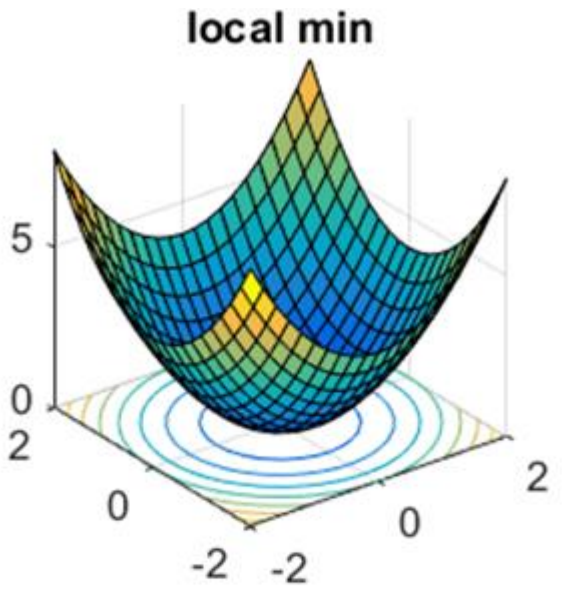


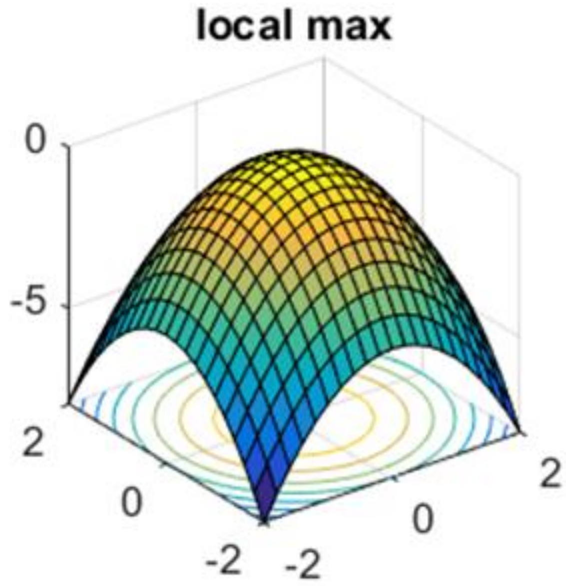
Hessian PSD

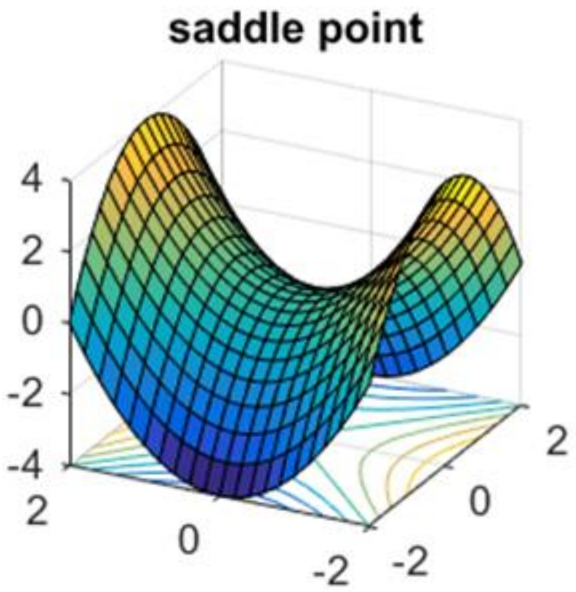
$$\nabla^2 f(x) \succcurlyeq 0$$

Second order stationary points (SOSP)

How do FOSPs look like?







Hessian PSD

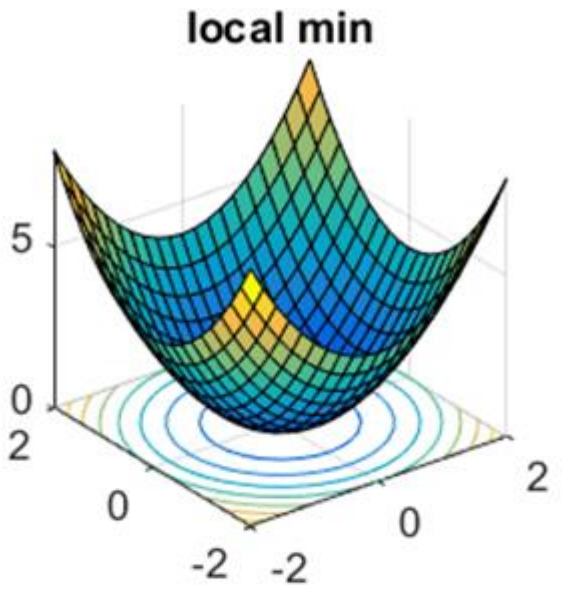
$\nabla^2 f(x) \succcurlyeq 0$

Second order stationary points (SOSP)

Hessian NSD

$\nabla^2 f(x) \preccurlyeq 0$

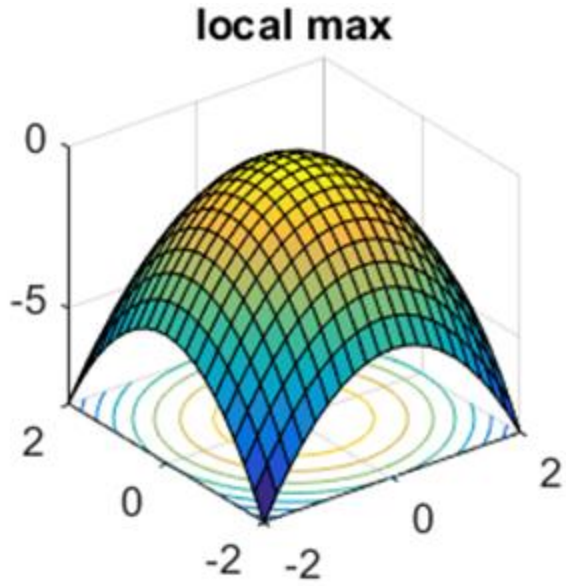
How do FOSPs look like?



Hessian PSD

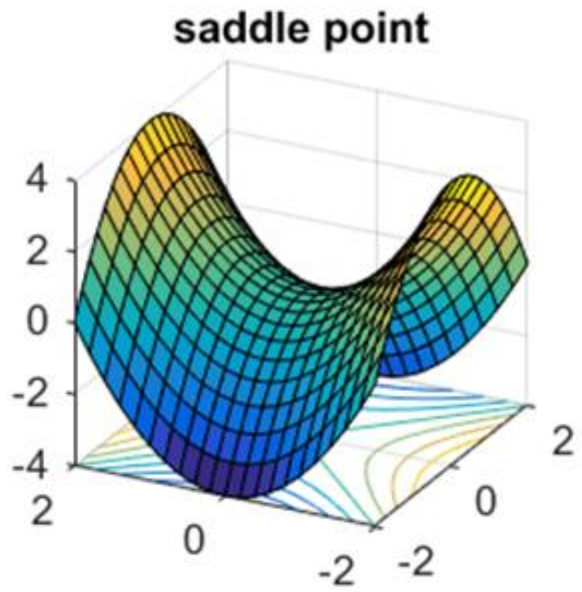
$\nabla^2 f(x) \succcurlyeq 0$

Second order stationary points (SOSP)



Hessian NSD

$\nabla^2 f(x) \preccurlyeq 0$



Hessian indefinite

$\lambda_{\min}(\nabla^2 f(x)) \leq 0$

$\lambda_{\max}(\nabla^2 f(x)) \geq 0$

FOSPs in popular problems

- Very well studied
  - Neural networks [Dauphin et al. 2014, Choromanska et al. 2014, Kawaguchi 2016]
  - Matrix sensing [Bhojanapalli et al. 2016]
  - Matrix completion [Ge et al. 2016]
  - Robust PCA [Ge et al. 2017]
  - Tensor factorization [Ge et al. 2015, Ge & Ma 2017]
  - Smooth semidefinite programs [Boumal et al. 2016]
  - Synchronization & community detection [Bandeira et al. 2016, Mei et al. 2017]

Two major observations

- FOSPs: proliferation (exponential #) of saddle points
  - Recall FOSP $\triangleq \nabla f(x) = 0$
  - Gradient descent can get stuck near them

- SOSPs: not just local minima; as good as global minima
  - Recall SOSP $\triangleq \nabla f(x) = 0$ & $\nabla^2 f(x) \succcurlyeq 0$

Upshot

1. FOSP not good enough
2. Finding SOSP sufficient

How to find SOSPs?

- Methods using full Hessian
  - Cubic regularization [Nesterov & Polyak 2006]
    Trust region [Curtis et al. 2014]
  - Infeasible for high dimensional problems

- Methods using Hessian-vector products
  - Carmon et al. 2016, Agarwal et al. 2017
    Royer & Wright 2017

- Pure gradient based methods
  - Ge et al. 2015, Levy 2016

How to find SOSPs?

- Methods using full Hessian
  - Cubic regularization [Nesterov & Polyak 2006]
    Trust region [Curtis et al. 2014]
  - Infeasible for high dimensional problems

1. Query $\nabla f(x_t)$ & $\nabla^2 f(x_t)$
2. $x_{t+1} \leftarrow \text{Update}\left(x_t, \nabla f(x_t), \nabla^2 f(x_t)\right)$

- Methods using Hessian-vector products
  - Carmon et al. 2016, Agarwal et al. 2017
    Royer & Wright 2017

- Pure gradient based methods
  - Ge et al. 2015, Levy 2016

How to find SOSPs?

- Methods using full Hessian
  - Cubic regularization [Nesterov & Polyak 2006]
    Trust region [Curtis et al. 2014]
  - Infeasible for high dimensional problems

1. Query $\nabla f(x_t)$ & $\nabla^2 f(x_t)$
2. $x_{t+1} \leftarrow \text{Update}\left(x_t, \nabla f(x_t), \nabla^2 f(x_t)\right)$

- Methods using Hessian-vector products
  - Carmon et al. 2016, Agarwal et al. 2017
    Royer & Wright 2017

1. Solve $\text{Update}\left(x_t, \nabla f(x_t), \nabla^2 f(x_t)\right)$ using GD/AGD
2. Query $\nabla f(x_t)$ & $\nabla^2 f(x_t) \cdot v$ for vectors v

- Pure gradient based methods
  - Ge et al. 2015, Levy 2016

How to find SOSPs?

- Methods using full Hessian
  - Cubic regularization [Nesterov & Polyak 2006]
    Trust region [Curtis et al. 2014]
  - Infeasible for high dimensional problems

1. Query $\nabla f(x_t)$ & $\nabla^2 f(x_t)$
2. $x_{t+1} \leftarrow \mathrm{Update}\left(x_t, \nabla f(x_t), \nabla^2 f(x_t)\right)$

- Methods using Hessian-vector products
  - Carmon et al. 2016, Agarwal et al. 2017
    Royer & Wright 2017

1. Solve $\mathrm{Update}\left(x_t, \nabla f(x_t), \nabla^2 f(x_t)\right)$ using GD/AGD
2. Query $\nabla f(x_t)$ & $\nabla^2 f(x_t) \cdot v$ for vectors v

- Pure gradient based methods
  - Ge et al. 2015, Levy 2016

Noisy GD [Ge et al. 2015]

$$x_{t+1} = x_t - \eta[\nabla f(x_t) + \zeta_t]$$

Gradient: $\nabla f(x_t)$; Random perturbation: ζ_t

State of the art

ϵ-SOSP [Nesterov & Polyak 2006]

$\|\nabla f(x)\| \leq \epsilon$ & $\lambda_{\min}(\nabla^2 f(x)) \gtrsim -\sqrt{\epsilon}$

Oracle	Paper	# Iterations	Simplicity
Full Hessian	Nesterov & Polyak 2006 Curtis et al. 2014	$O\left(\frac{1}{\epsilon^{1.5}}\right)$	Single loop
Hessian-vector product	Carmon et al. 2016 Agarwal et al. 2017	$\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)$	Nested loop
Gradient	Ge et al. 2015 Levy 2016	$O\left(\mathrm{poly}\left(\frac{d}{\epsilon}\right)\right)$	Single loop

State of the art

ϵ-SOSP [Nesterov & Polyak 2006]
$\|\nabla f(x)\| \leq \epsilon$ & $\lambda_{\min}\left(\nabla^2 f(x)\right) \gtrsim -\sqrt{\epsilon}$

Oracle	Paper	# Iterations	Simplicity
Full Hessian	Nesterov & Polyak 2006 Curtis et al. 2014	$O\left(\frac{1}{\epsilon^{1.5}}\right)$	Single loop
Hessian-vector product	Carmon et al. 2016 Agarwal et al. 2017	$\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)$	Nested loop
Gradient	Ge et al. 2015 Levy 2016	$O\left(\mathrm{poly}\left(\frac{d}{\epsilon}\right)\right)$	Single loop

Question 1
Does **essentially pure GD** converge to SOSP efficiently?
In particular, independent of d?

Yes (almost)!

ϵ-SOSP [Nesterov & Polyak 2006]

$\|\nabla f(x)\| \leq \epsilon$ & $\lambda_{\min}(\nabla^2 f(x)) \gtrsim -\sqrt{\epsilon}$

Oracle	Paper	# Iterations	Simplicity
Full Hessian	Nesterov & Polyak 2006 Curtis et al. 2014	$O\left(\frac{1}{\epsilon^{1.5}}\right)$	Single loop
Hessian-vector product	Carmon et al. 2016 Agarwal et al. 2017	$\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)$	Nested loop
Gradient	Ge et al. 2015 Levy 2016	$O\left(\text{poly}\left(\frac{d}{\epsilon}\right)\right)$	Single loop
Gradient	**Jin, Ge, N., Kakade, Jordan 2017**	$\boldsymbol{\tilde{O}\left(\frac{1}{\epsilon^{2}}\right)}$	**Single loop**

Yes (almost)!

ϵ-SOSP [Nesterov & Polyak 2006]
$\|\nabla f(x)\| \le \epsilon$ & $\lambda_{\min}(\nabla^2 f(x)) \gtrsim -\sqrt{\epsilon}$

Oracle	Paper	# Iterations	Simplicity
Full Hessian	Nesterov & Polyak 2006 Curtis et al. 2014	$O\left(\frac{1}{\epsilon^{1.5}}\right)$	Single loop
Hessian-vector product	Carmon et al. 2016 Agarwal et al. 2017	$\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)$	Nested loop
Gradient	Ge et al. 2015 Levy 2016	$O\left(\mathrm{poly}\left(\frac{d}{\epsilon}\right)\right)$	Single loop
Gradient	**Jin, Ge, N., Kakade, Jordan 2017**	$\tilde{O}\left(\frac{1}{\epsilon^2}\right)$	**Single loop**

Question 2
Does **essentially pure AGD** converge to SOSP faster than essentially pure GD?

Yes!

<u>ϵ-SOSP [Nesterov & Polyak 2006]</u>

$\|\nabla f(x)\| \leq \epsilon$ & $\lambda_{\min}(\nabla^2 f(x)) \gtrsim -\sqrt{\epsilon}$

Oracle	Paper	# Iterations	Simplicity
Full Hessian	Nesterov & Polyak 2006 Curtis et al. 2014	$O\left(\frac{1}{\epsilon^{1.5}}\right)$	Single loop
Hessian-vector product	Carmon et al. 2016 Agarwal et al. 2017	$\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)$	Nested loop
Gradient	Ge et al. 2015 Levy 2016	$O\left(\mathrm{poly}\left(\frac{d}{\epsilon}\right)\right)$	Single loop
Gradient	**Jin, Ge, N., Kakade, Jordan 2017**	$\boldsymbol{\tilde{O}\left(\frac{1}{\epsilon^{2}}\right)}$	**Single loop**
Gradient	**Jin, N., Jordan 2017**	$\boldsymbol{\tilde{O}\left(\frac{1}{\epsilon^{1.75}}\right)}$	**Single loop**

Summary of results

- Convergence to SOSPs very important in practice
- Pure GD and AGD can get stuck near FOSPs (saddle points)
- Small modifications (such as adding perturbation) to GD and AGD helps them escape saddle points efficiently
- Do not need complicated nested loop algorithms

Main Ideas of the Proof of Gradient Descent

Setting

- **Gradient Lipschitz**: $\|\nabla f(x) - \nabla f(y)\| \lesssim \|x - y\|$
- **Hessian Lipschitz**: $\|\nabla^2 f(x) - \nabla^2 f(y)\| \lesssim \|x - y\|$
- **Lower bounded**: $\min_x f(x) > -\infty$

How does GD behave?

GD step

$x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

Recall

FOSP: $\nabla f(x)$ small

SOSP: $\nabla f(x)$ small &

$\lambda_{\min}\left(\nabla^2 f(x)\right) \gtrsim 0$

How does GD behave?

Recall

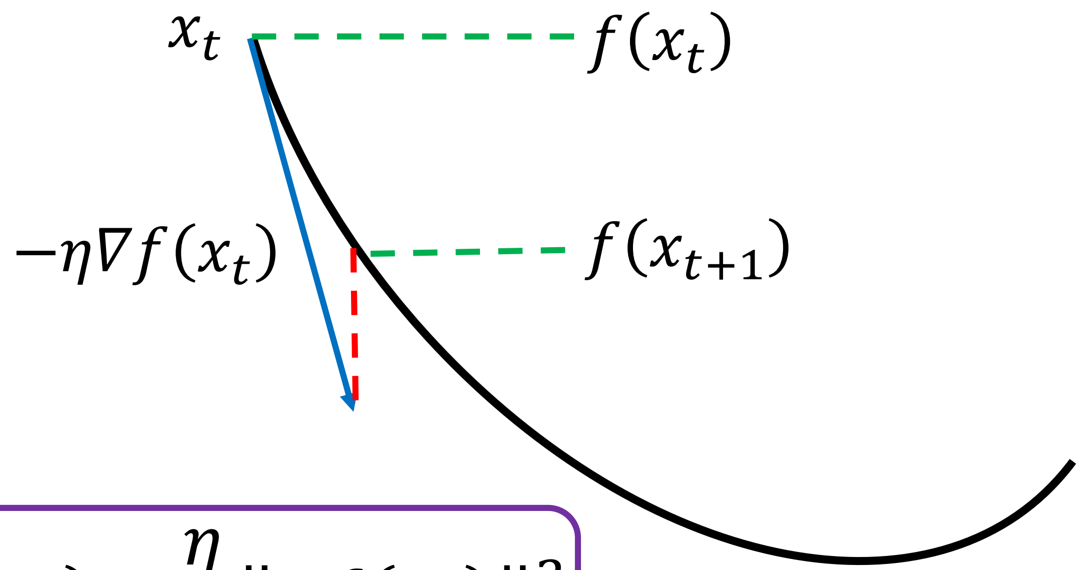
FOSP: $\nabla f(x)$ small

SOSP: $\nabla f(x)$ small & $\lambda_{\min}\left(\nabla^2 f(x)\right) \gtrsim 0$

GD step

$x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

$\|\nabla f(x_t)\|$ large



$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2$$

✓

How does GD behave?

Recall

FOSP: $\nabla f(x)$ small

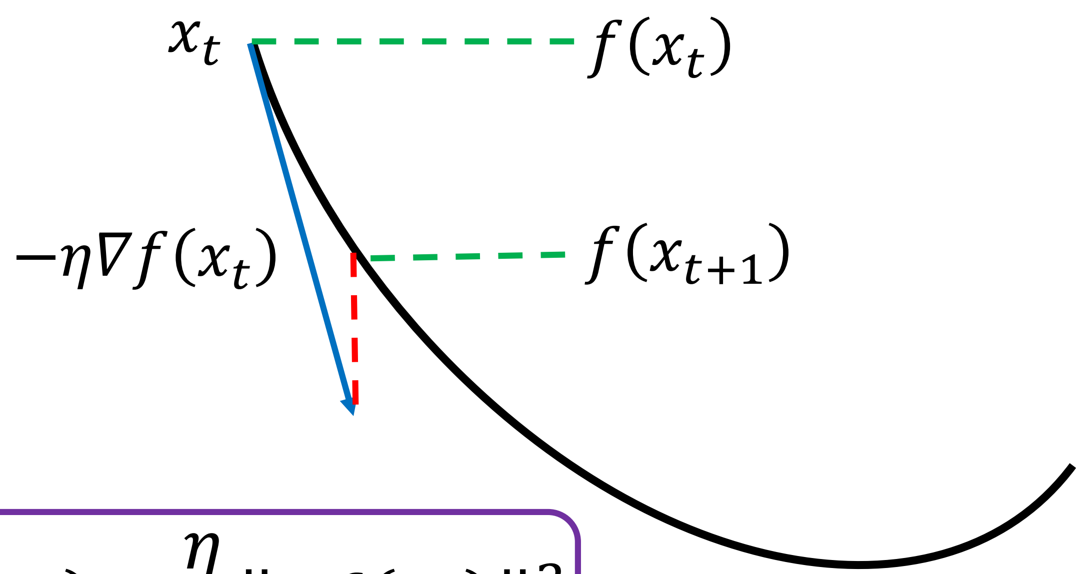
SOSP: $\nabla f(x)$ small & $\lambda_{\min}\left(\nabla^2 f(x)\right) \gtrsim 0$

GD step

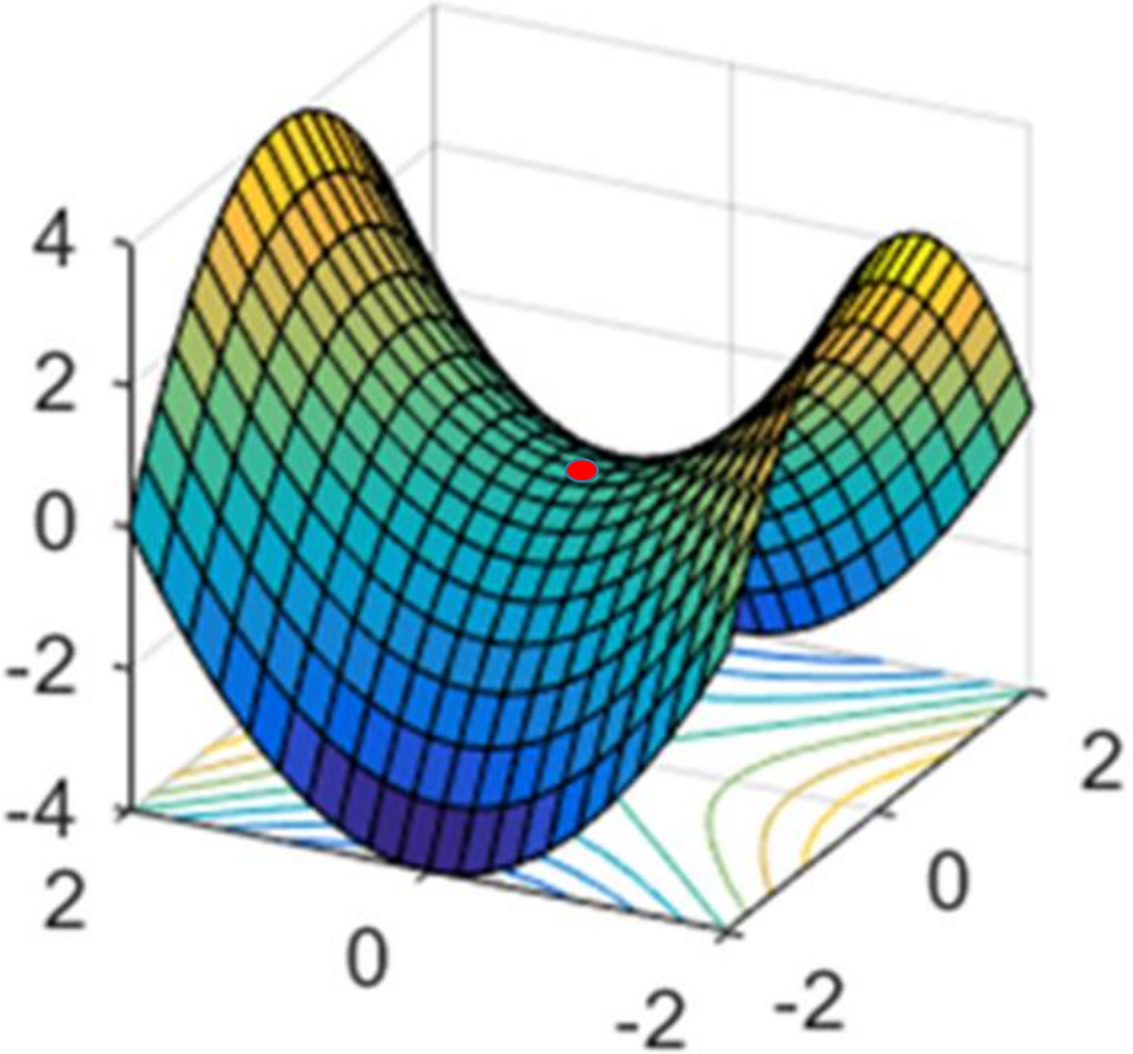
$x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

$\|\nabla f(x_t)\|$ small → SOSP ✓ / Saddle point ?

$\|\nabla f(x_t)\|$ large → $f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2$ ✓



How to escape saddle points?



Perturbed gradient descent

1. **For** $t = 0,1,\cdots$ **do**
2. **if** perturbation_condition_holds **then**
3. $x_t \leftarrow x_t + \xi_t$ where $\xi_t \sim Unif\big(B_0(\epsilon)\big)$
4. $x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

Perturbed gradient descent

1. **For** $t = 0,1,\cdots$ **do**
2. **if** perturbation_condition_holds **then**
3. $x_t \leftarrow x_t + \xi_t$ where $\xi_t \sim Unif\left(B_0(\epsilon)\right)$
4. $x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

Between two perturbations, just run GD!

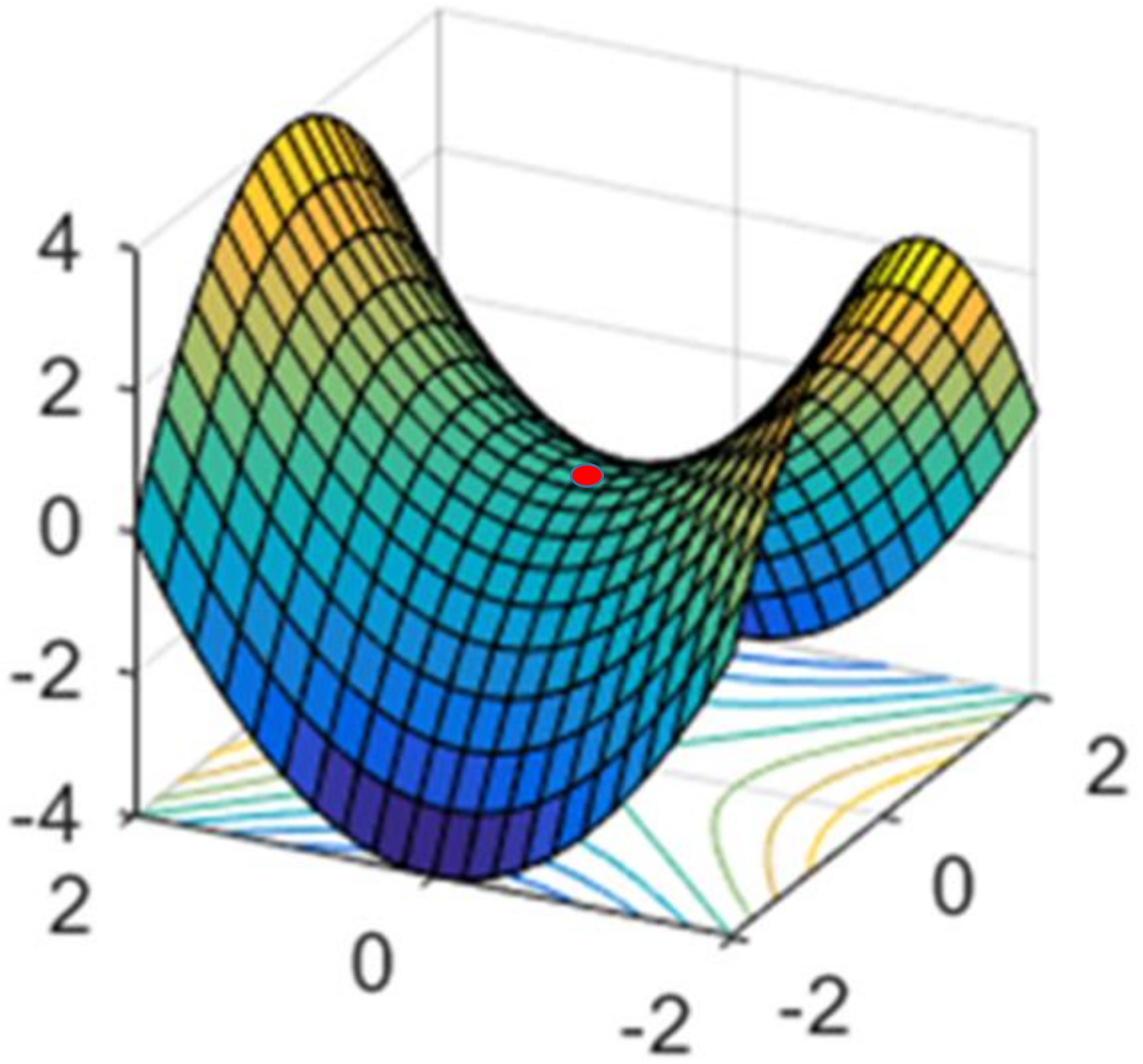
Perturbed gradient descent

1. $\nabla f(x_t)$ is small
2. No perturbation in last several iterations

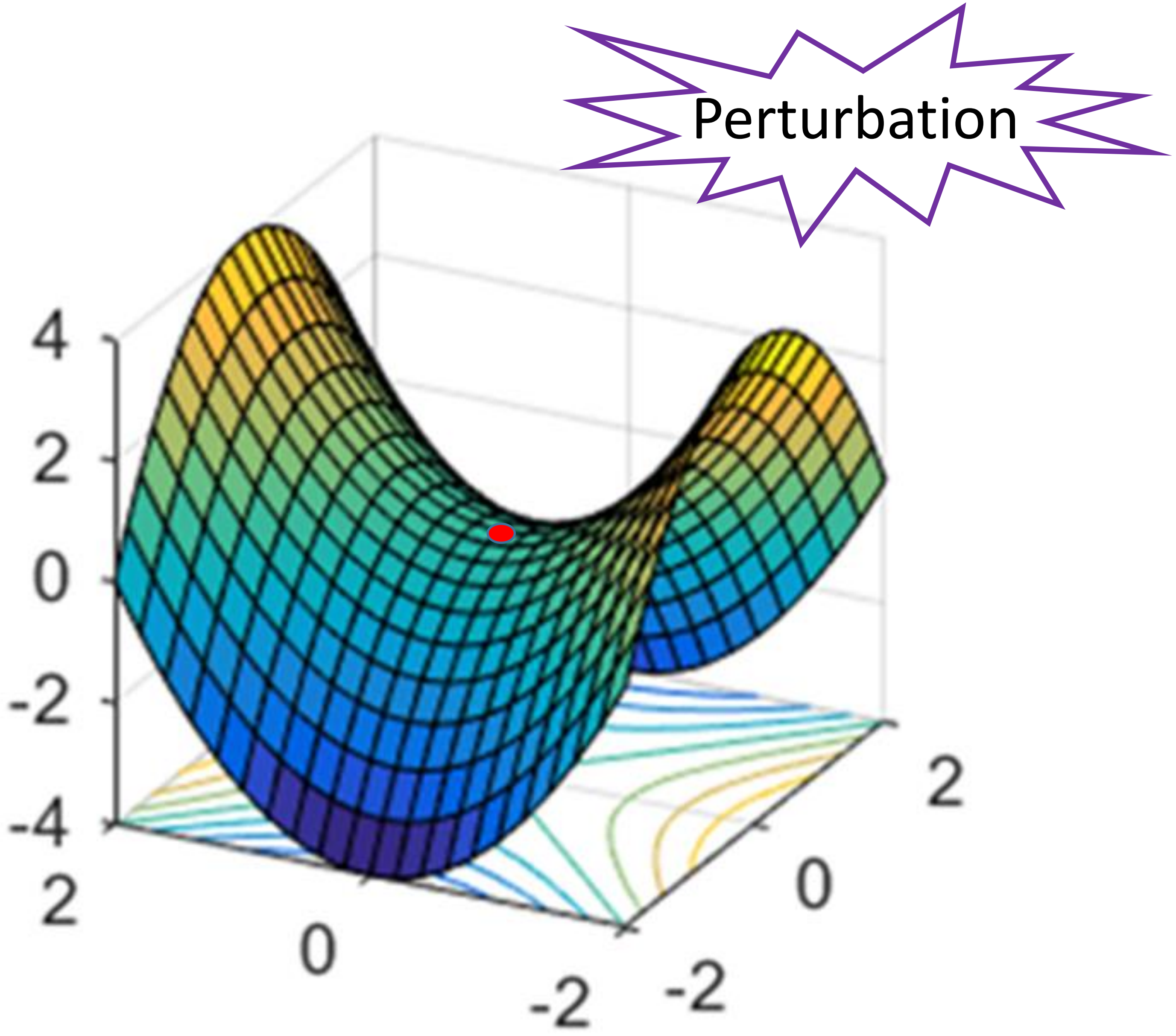
1. **For** $t = 0,1,\cdots$ **do**
2. **if** perturbation_condition_holds **then**
3. $x_t \leftarrow x_t + \xi_t$ where $\xi_t \sim Unif\left(B_0(\epsilon)\right)$
4. $x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$

Between two perturbations, just run GD!

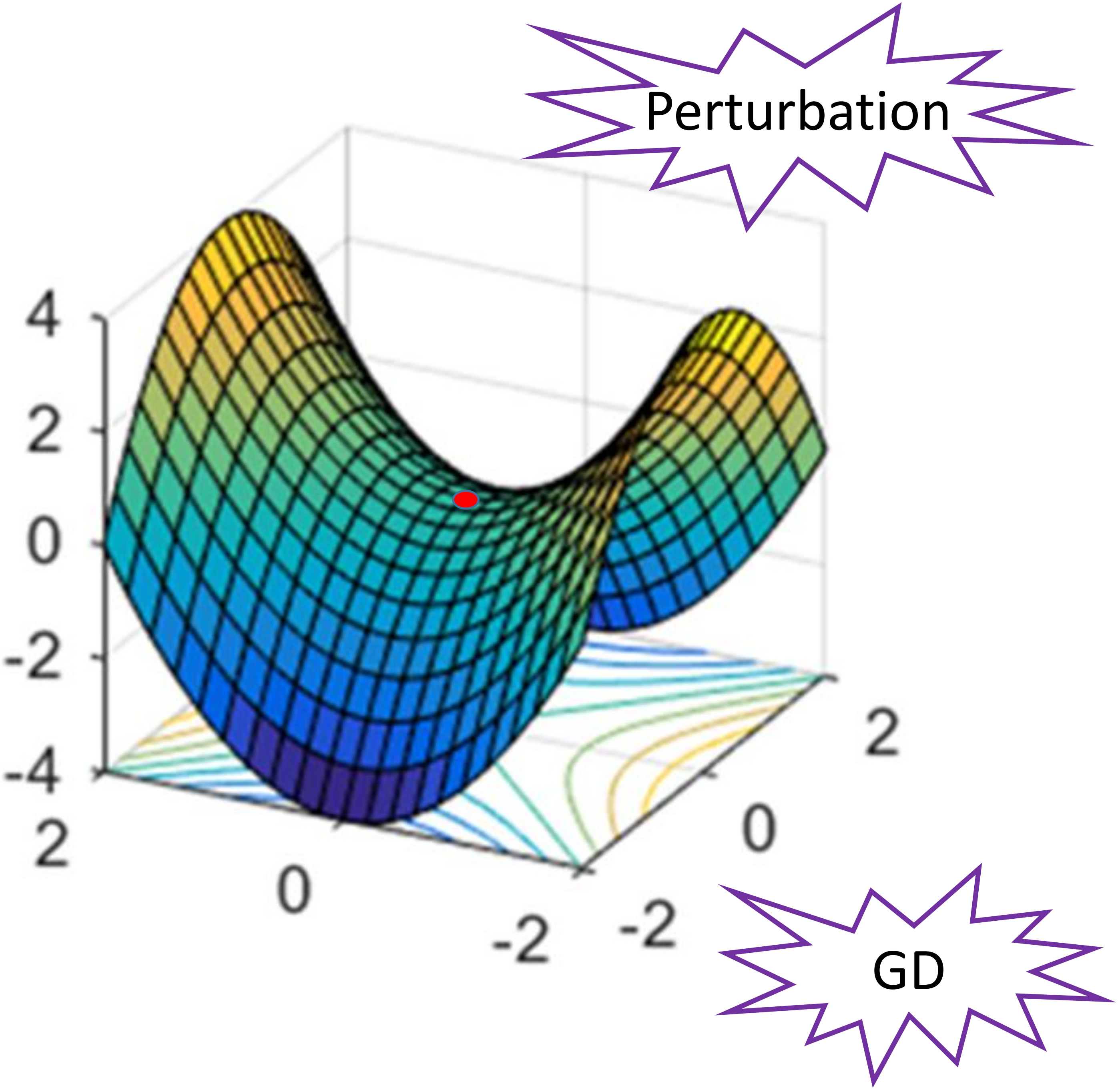
How can perturbation help?



How can perturbation help?



How can perturbation help?



Key question

- $S \stackrel{\text{def}}{=}$ set of points around saddle point from where gradient descent does not escape quickly

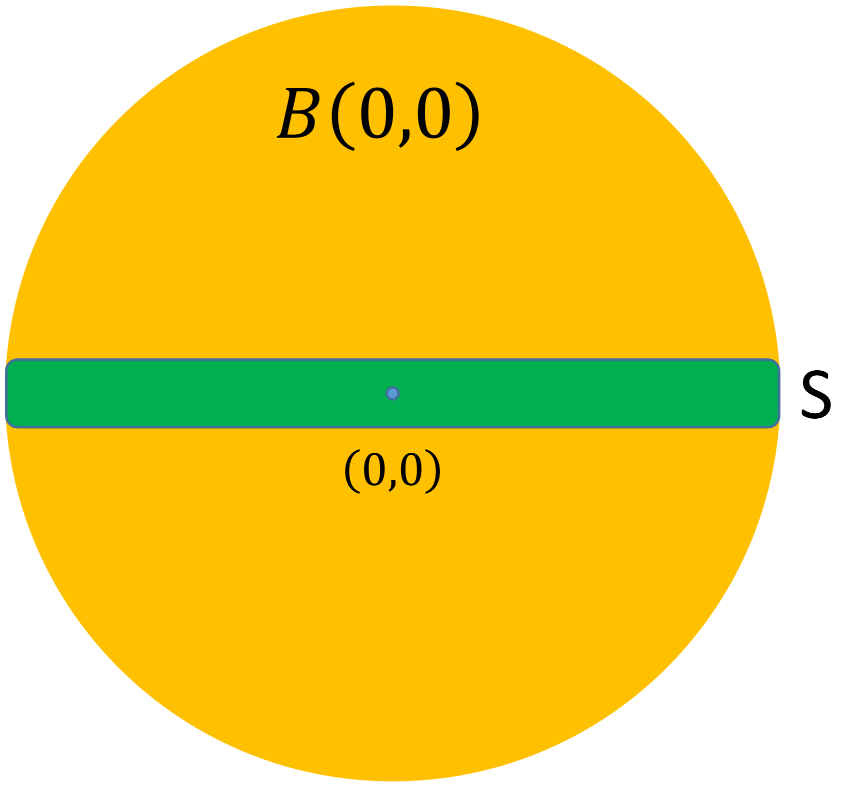
- Escape $\stackrel{\text{def}}{=}$ function value decreases significantly

- How much is $\text{Vol}(S)$?

- $\text{Vol}(S)$ small $\Rightarrow$ perturbed GD escapes saddle points efficiently

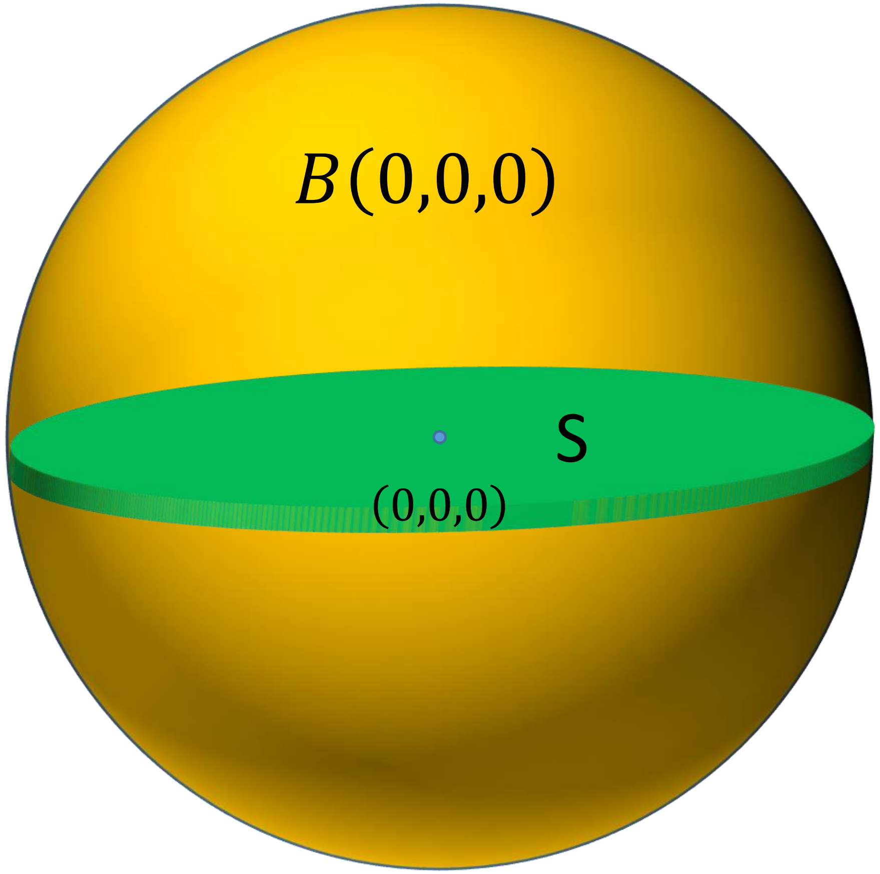
Two dimensional quadratic case

- $f(x) = \frac{1}{2} x^\top \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} x$
- $\lambda_{\min}(H) = -1 < 0$
- $(0,0)$ is a saddle point
- GD: $x_{t+1} = \begin{bmatrix} 1 - \eta & 0 \\ 0 & 1 + \eta \end{bmatrix} x_t$
- S is a thin strip, $\mathrm{Vol}(S)$ is small



Three dimensional quadratic case

- $f(x) = \frac{1}{2}x^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x$
- $(0,0,0)$ is a saddle point
- GD: $x_{t+1} = \begin{bmatrix} 1-\eta & 0 & 0 \\ 0 & 1-\eta & 0 \\ 0 & 0 & 1+\eta \end{bmatrix} x_t$
- S is a thin disc, $\mathrm{Vol}(S)$ is small

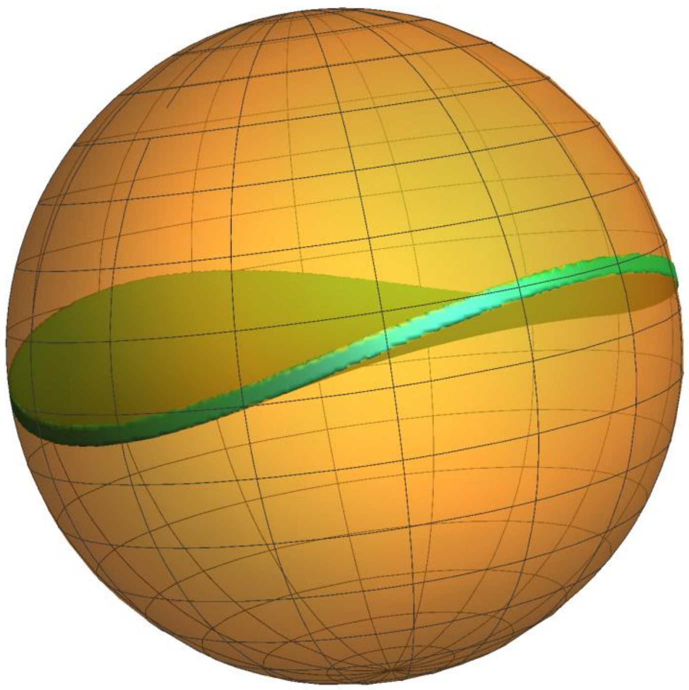


General case

Key technical results

S ~ thin deformed disc

$\mathrm{Vol}(S)$ is small



Two key ingredients of the proof

Two key ingredients of the proof

Improve or localize

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2}\|\nabla f(x_t)\|^2$$
$$= f(x_t) - \frac{\eta}{2}\left\|\frac{x_t - x_{t+1}}{\eta}\right\|^2$$

$$\|x_t - x_{t+1}\|^2 \leq 2\eta(f(x_t) - f(x_{t+1}))$$

$$\|x_0 - x_t\|^2 \leq t\sum_{i=0}^{t-1}\|x_i - x_{i+1}\|^2$$
$$\leq 2\eta t\big(f(x_0) - f(x_t)\big)$$

Two key ingredients of the proof

Improve or localize

Upshot

Either function value decreases significantly or iterates do not move much

$$\|x_0 - x_t\|^2 \leq t \sum_{i=0}^{t-1} \|x_i - x_{i+1}\|^2$$

$$\leq 2\eta t \big(f(x_0) - f(x_t)\big)$$

Two key ingredients of the proof

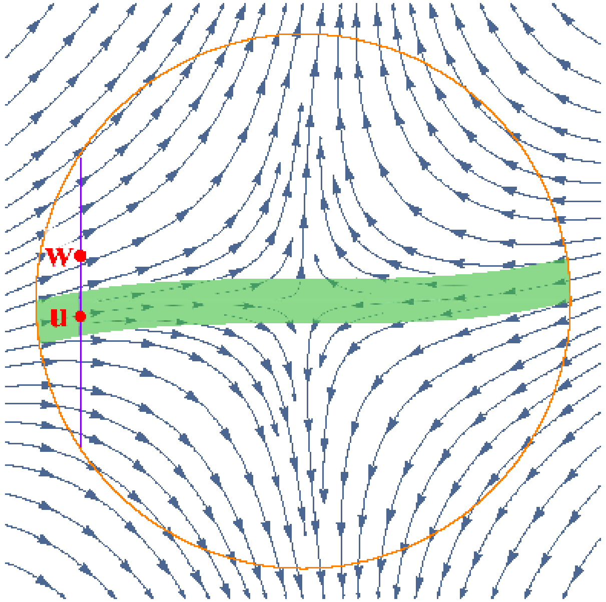
Improve or localize

Upshot

Either function value decreases significantly or iterates do not move much

$$\|x_0 - x_t\|^2 \le t \sum_{i=0}^{t-1} \|x_i - x_{i+1}\|^2 \le 2\eta t\big(f(x_0) - f(x_t)\big)$$

Coupling



Either GD from u escapes

Or GD from w escapes

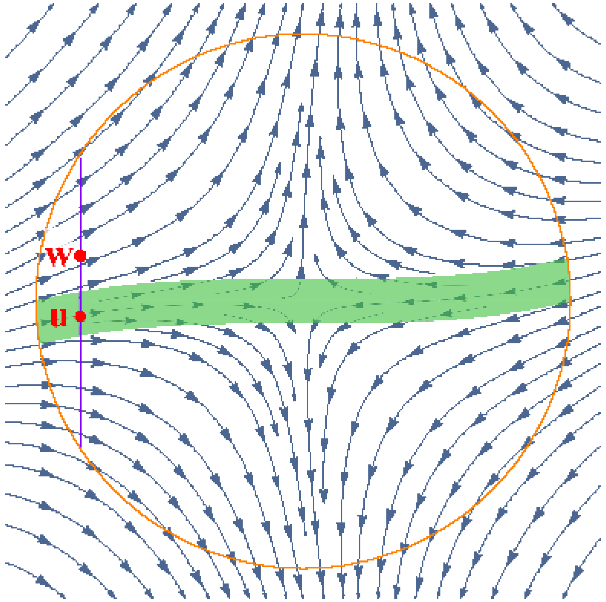
Proof idea

- If GD from either u or w goes outside a small ball, it escapes (function value $\Downarrow$)

- If GD from both u and w lie in a small ball, use local quadratic approximation of $f(\cdot)$

- Show the claim for exact quadratic, and bound approximation error using Hessian Lipschitz property

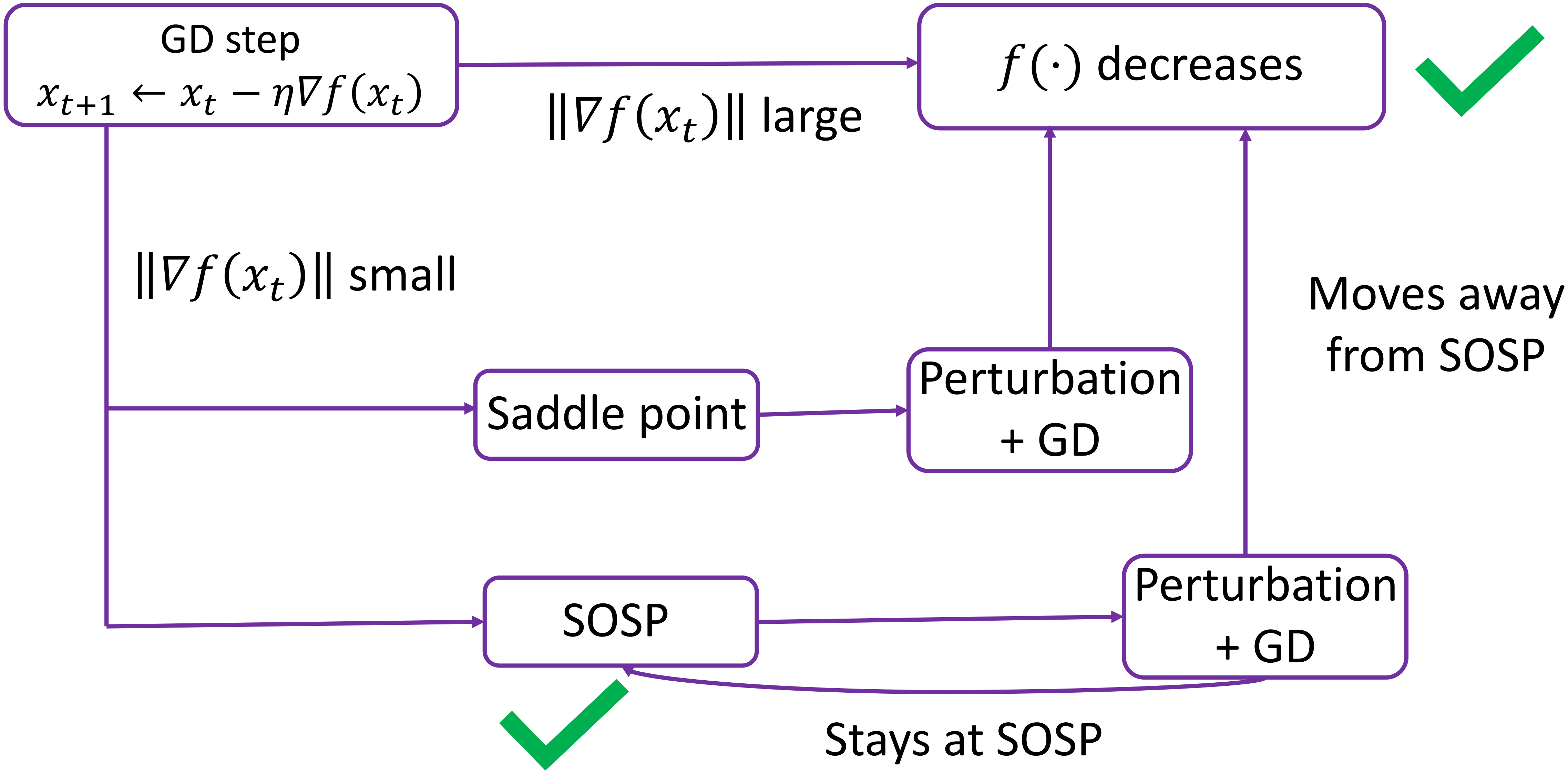
<u>Coupling</u>



Either GD from u escapes

Or GD from w escapes

Putting everything together



Not in today's talk

- "Essentially pure AGD escapes saddle points faster than essentially pure GD"
- Key tool: New Hamiltonian (potential function in CS parlance) for AGD
- Inspired by differential equation view of AGD [Su et al. 2015]
- See https://arxiv.org/abs/1711.10456 for details

Summary

- Simple variations to GD/AGD ensure efficient escape from saddle points

- Fine understanding of geometric structure around saddle points

- Novel techniques of independent interest

- Some extensions to stochastic setting

Open questions

- Lower bounds – recent work by Carmon et al. 2017, but gaps between upper and lower bounds

- Extensions to stochastic setting

- Nonconvex optimization for faster algorithms

Thank you!

Questions?