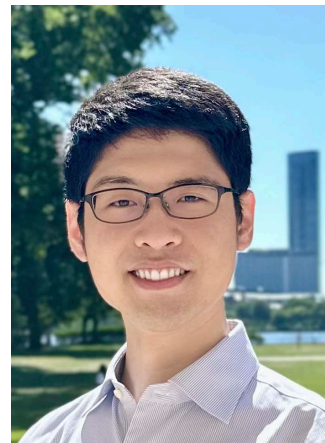


# Level-Set Geometry and the Performance of Restarted-PDHG for Conic LP

Zikai Xiong and Robert Freund



Zikai Xiong  
(MIT OR Center)



Robert Freund  
(MIT Sloan)

Paper available on [arXiv](#)

Shanghai Jiao Tong University

June 2025

# Huge-scale Optimization is “Everywhere”

**Manufacturing**



**Machine Learning**



**Energy**



**Transportation**



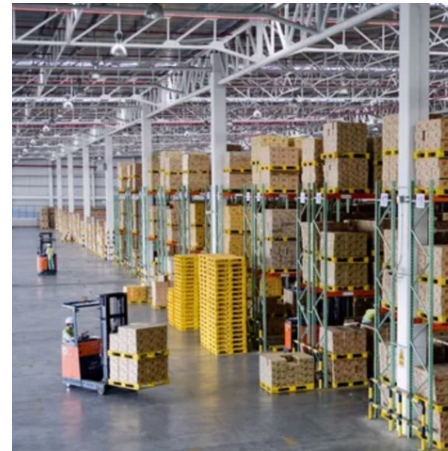
**Healthcare**



**Markets and Auctions**



**Supply Chains**



**Agriculture**

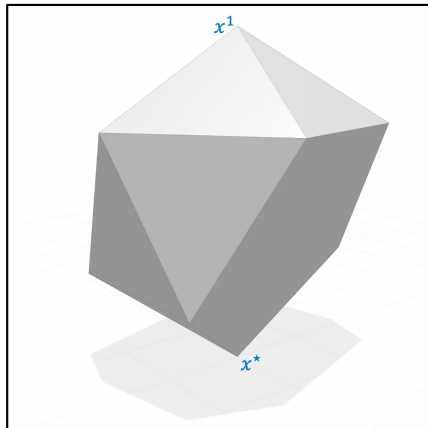


# History of Linear Optimization (“LO” or “LP”)

1947

**Simplex  
Method**

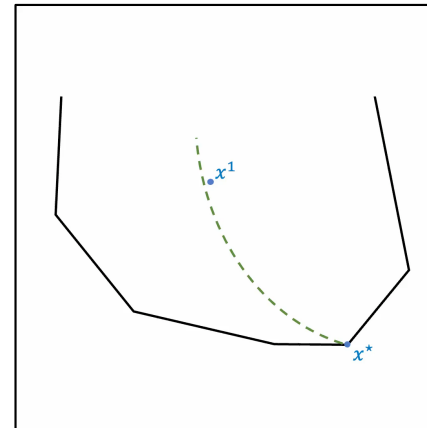
[George Dantzig, 1947]  
75+ years ago



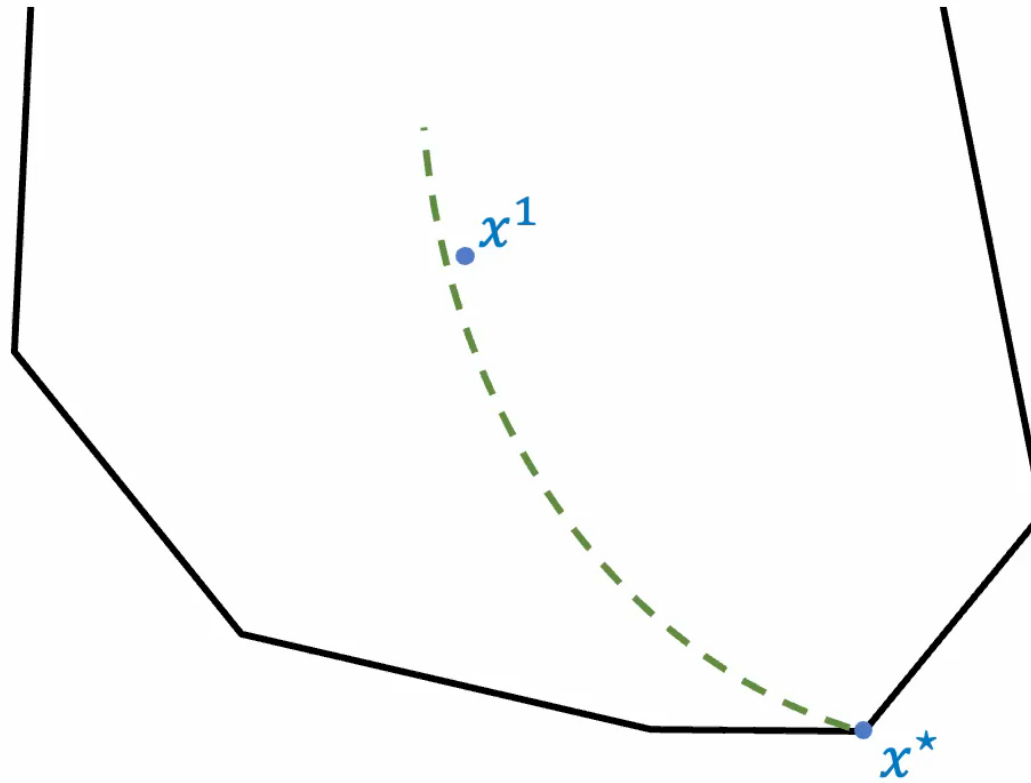
1984

**Interior Point  
Method**

[Narendra Karmarkar, 1984]  
40 years ago



# One slide on Interior-Point Methods (IPMs)



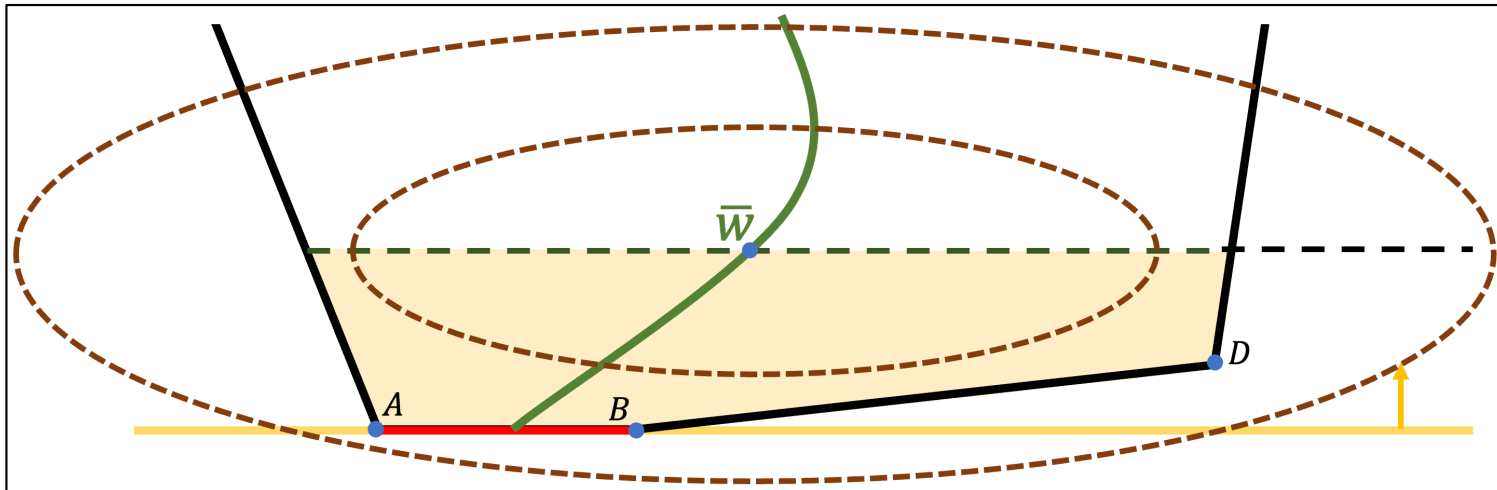
# Central-Path ellipsoids have remarkable properties

Central-path solutions are:

$$x_\mu := \arg \min_x c^\top x + \mu \cdot F(x) \\ \text{s. t. } Ax = b, x \in \mathcal{K}$$

$$y_\mu, s_\mu := \arg \max_{y,s} b^\top y - \mu \cdot F^*(s) \\ \text{s. t. } A^\top y + s = c, s \in \mathcal{K}^*$$

Example for LP:  $F(x) := -\sum_{i=1}^n \ln(x_i)$ ,  $\vartheta_F = n$



# Simplex and IPMs require expensive matrix factorizations

Consider an LP instance with  $n$  decision variables and  $\frac{n}{2}$  linear constraints, whose constraint matrix has sparsity = 0.05

**Number of  
variables ( $n$ )**

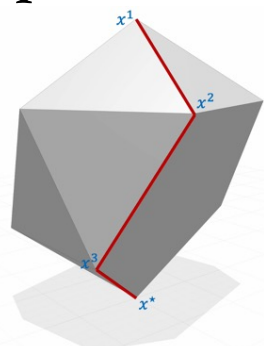
**Cost of one IPM  
iteration**

Hence the emergence of FOMs for solving huge (and also not-so-huge) LP instances

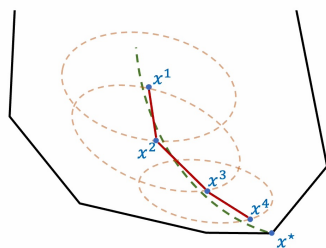
# Recent Advances on Huge-Scale LP Solvers in the Industry

## Classic methods

### Simplex method



### Interior-point method



## First-order methods

- Primal-Dual Hybrid Gradient (“PDHG”, “Chambolle-Pock method”)
- Tackles huge-scale problems
- Benefits from modern computational architecture (such as GPU)



2021



February, 2024



March, 2024



March, 2024



April, 2024



October, 2024

**We are witnessing a dramatic shift from classic methods to first-order methods**

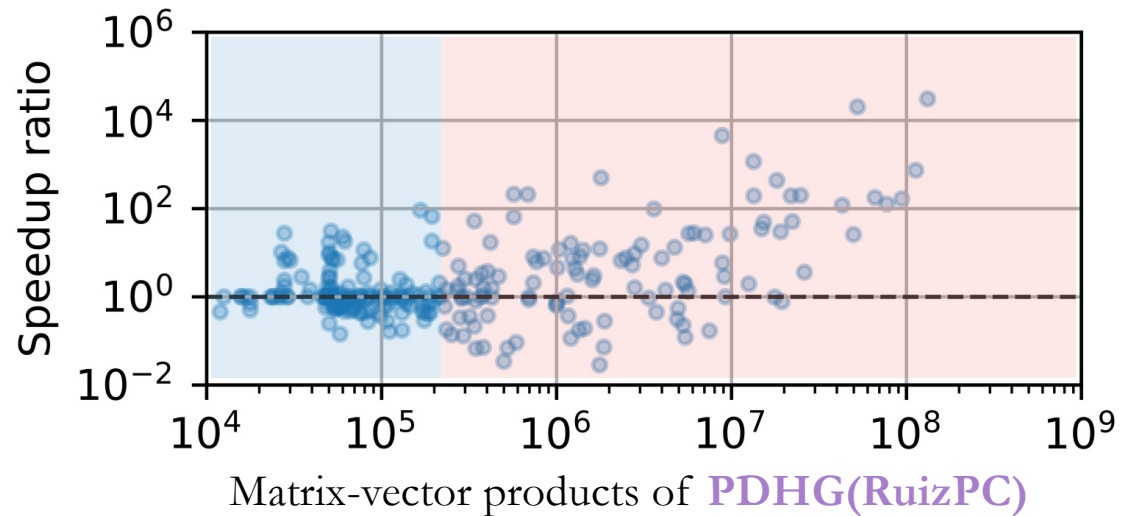
# Huge-Scale LP Research

- SCS: Operator splitting/ADMM [O'Donoghue, Chu, Parikh, Boyd, 2016]
- ABIP+: ADMM-based interior-point method  
[Lin, Ma, Ye, Zhang, 2021] & [Deng, et al., 2022]
- Semi-smooth Newton augmented Lagrangian [Li, Sun, Toh, 2020]
- **Primal-Dual Hybrid Gradient (PDHG)** with restarts, applied directly to the primal-dual saddle point problem  
[Applegate, Hinder, Lu, Lubin, 2023] & [Applegate, et al., 2021]
- **GPU implementations** of PDHG in Julia and C  
[Lu and Yang, 2023] & [Lu, Yang, Hu, Qi, Liu, Liu, Ye, Zhang, Ge, 2023]
- **Guarantees for PDHG for LP** using “Limiting Error Ratios” and LP Sharpness [Xiong and F 2023]
- **Guarantees for PDHG for CLP** – using level-set geometry  
[Xiong and F 2024]

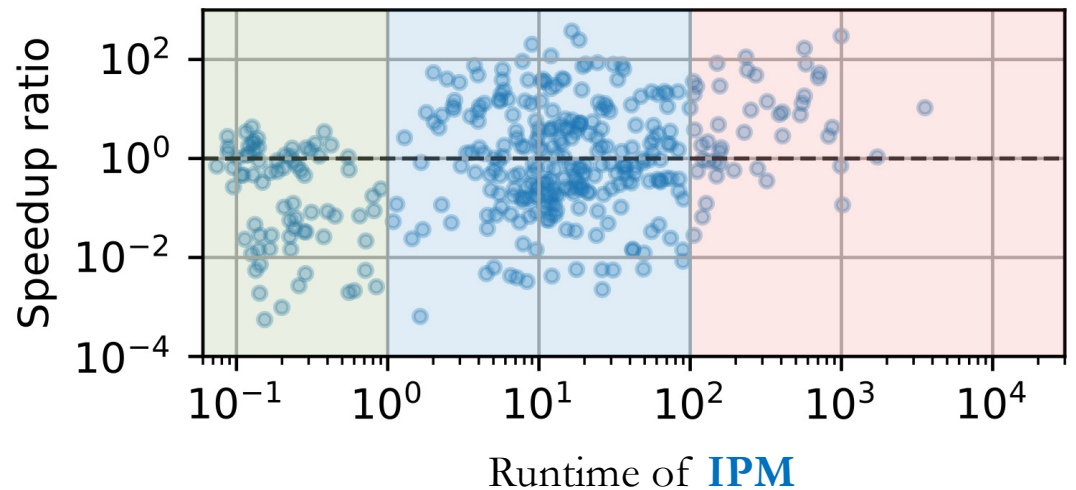
Sneak Preview:

# Distribution of Speedups of our method **PDHG-AHR**

Speedups compared with  
**PDHG(RuizPC)**

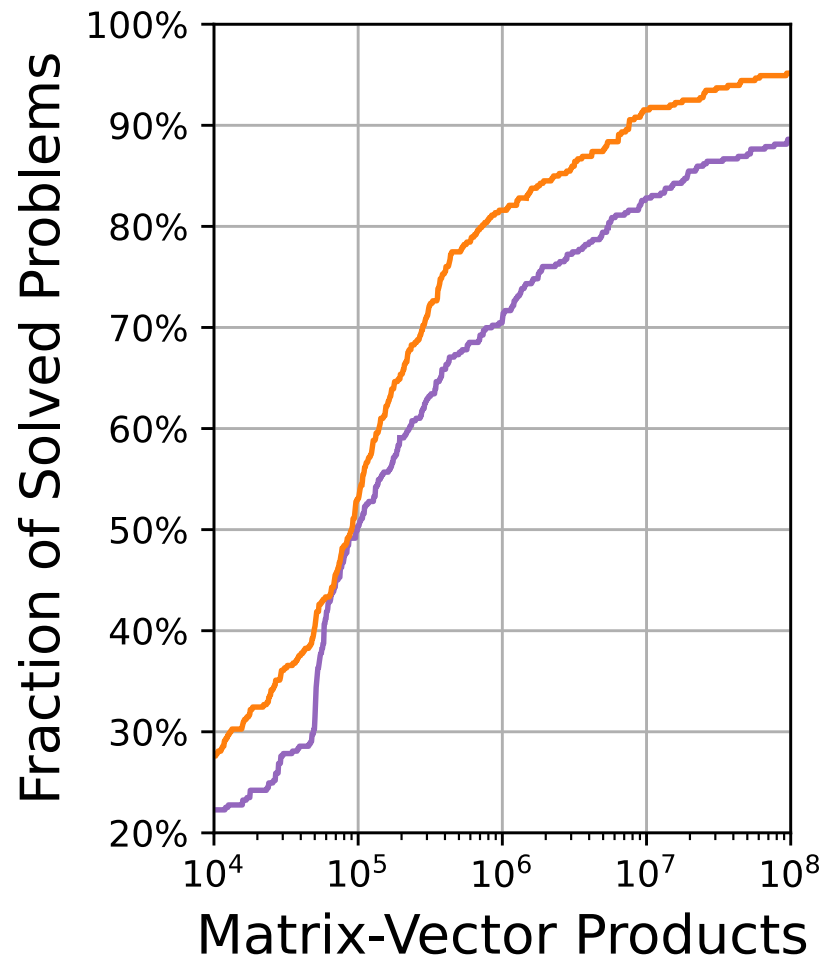


Speedups compared with a  
“home-grown” **IPM**  
(Predictor-corrector path-  
following interior-point method in  
Nocedal and Wright *Numerical  
Optimization* (2006))



Sneak Preview:

# Performance Comparison on MIPLIB 2017



## PDHG-AHR

rPDHG with our adaptive rescaling

## PDHG (RuizPC)

rPDHG with heuristic Ruiz/PC rescaling  
(same with “PDLP”)

# Conic Linear Optimization (“CLO” or “CLP”)

## CLP in standard form

(primal)

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathcal{K} \end{aligned}$$

(dual)

$$\begin{aligned} \max \quad & b^\top y \\ \text{s.t.} \quad & c - A^\top y \in \mathcal{K}^* \end{aligned}$$

Decision variables

- $x \in R^n$  (for primal problem)
- $y \in R^m$  (for dual problem)

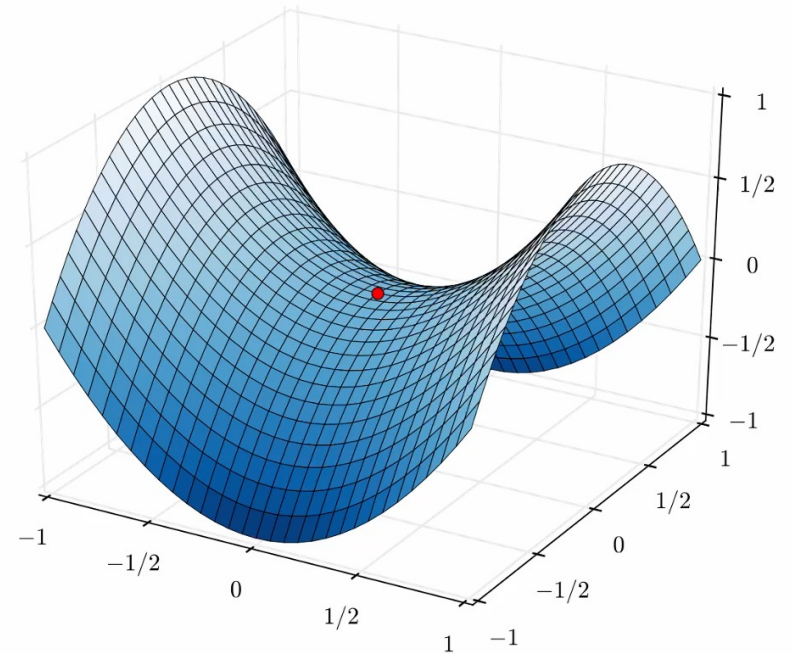
## CLP saddlepoint formulation

$$\min_{x \in \mathcal{K}} \max_y c^\top x - y^\top Ax + b^\top y$$

# Primal-Dual Hybrid Gradient Method (PDHG)

Conic Optimization in  
Saddlepoint Form

$$\min_{x \in \mathcal{K}} \max_y c^\top x + b^\top y - y^\top A x$$



PDHG

$$x^{k+1} \leftarrow \text{Proj}_{\mathcal{K}} \left( x^k - \tau (c - A^\top y^k) \right)$$

Gradient w.r.t.  $x^k$

$$y^{k+1} \leftarrow y^k + \sigma (b - A x^{k+1}) - \sigma A (x^{k+1} - x^k)$$

Gradient w.r.t.  $y^k$                       Momentum Term

# Primal-Dual Hybrid Gradient for Conic Optimization

## PDHG

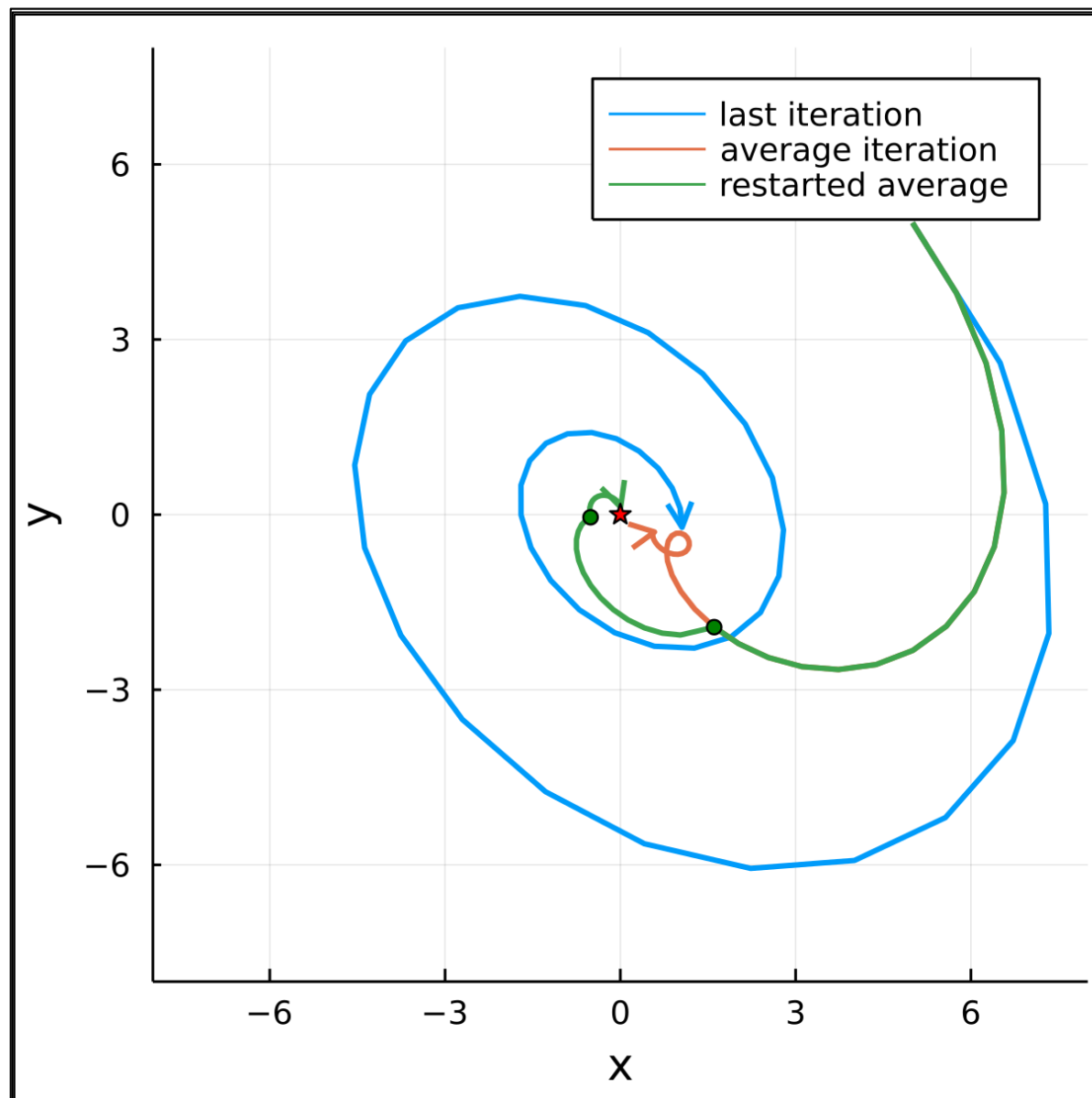
$$\begin{aligned}x^{k+1} &\leftarrow \text{Proj}_{\mathcal{K}} \left( x^k - \tau(c - A^\top y^k) \right) \\y^{k+1} &\leftarrow y^k + \sigma(b - Ax^{k+1}) - \sigma A(x^{k+1} - x^k)\end{aligned}$$

- **Inexpensive iterations:**  
Only requires matrix-vector multiplications
- **“Fast” convergence rates:**  
Adaptive restarts based on average iterates yield global linear convergence on LP [Applegate, Hinder, Lu, Lubin, 2023]

**We use “PDHG” to denote “PDHG with adaptive restarts”**

# Motivation for Restarts for PDHG: “Visualization”

$$\min_x \max_y x \cdot y$$

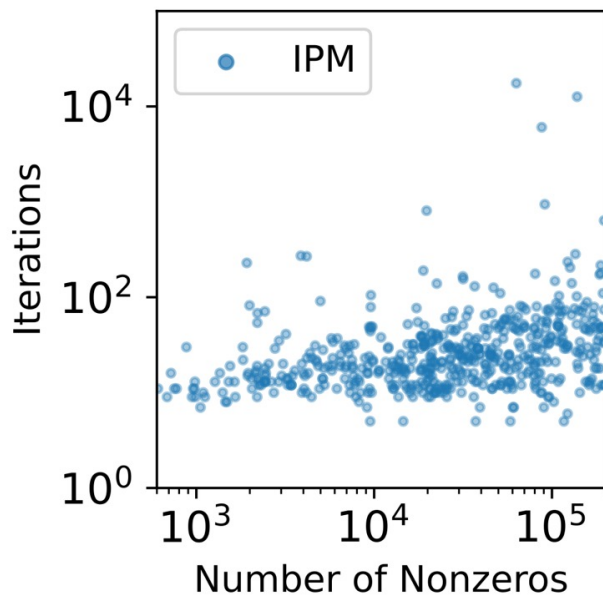


\*figure courtesy Haihao Lu

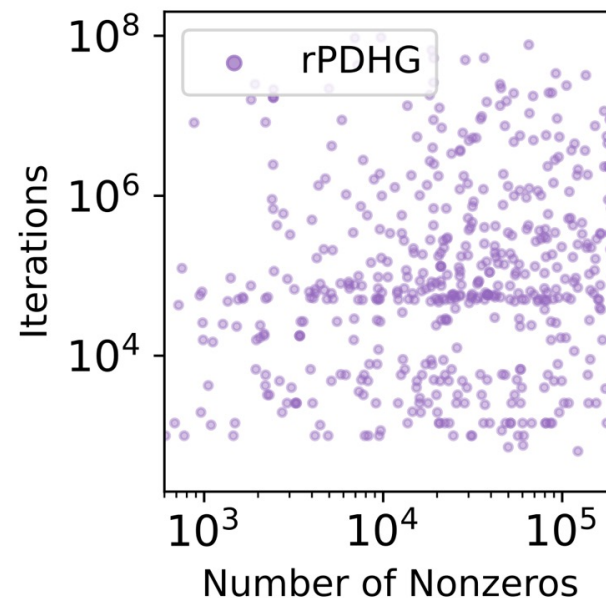
# Challenge I: Variability in the Performance of PDHG

- PDHG uses many more iterations than an IPM  
makes sense, it is a first-order method ... IPM iterations are hugely expensive while PDHG iterations are very cheap
- Some small problem instances require a very large number of PDHG iterations  
a real challenge for PDHG

IPM Iterations needed for  
LP relaxations from MIPLIB 2017



PDHG iterations  
LP relaxations from MIPLIB 2017



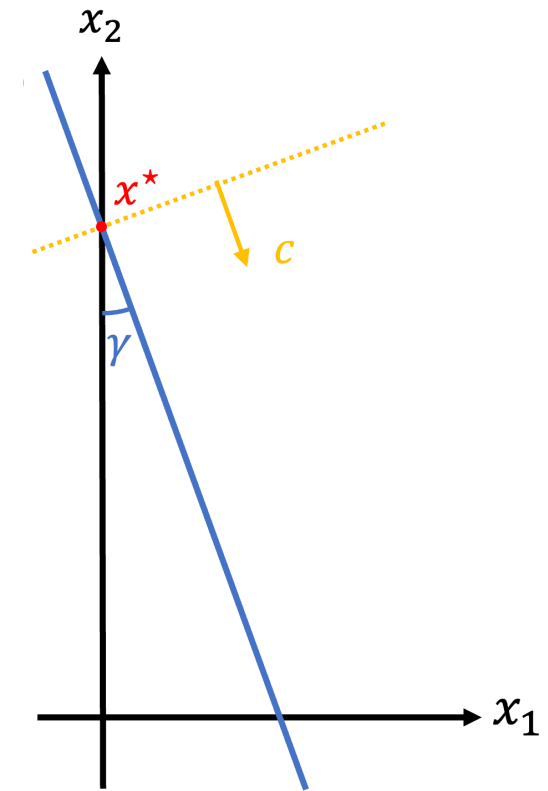
# A seemingly easy LP instance

For  $\gamma \in (0, \frac{\pi}{2})$  define:

$$P(\gamma): \quad \begin{array}{ll} \min_{x_1, x_2} & \sin(\gamma) x_1 - \cos(\gamma) x_2 \\ \text{s. t.} & \sin(\gamma) x_1 + \cos(\gamma) x_2 = 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

$P(\gamma)$  is always easy for the simplex method and interior-point methods

However, when  $\gamma$  is small, PDHG requires at least 1,000,000 iterations. What **conditions** of  $P(\gamma)$  make it so hard for PDHG?



# Challenge II: Loose/unworkable computational guarantees

Existing computational guarantees:

**Theorem** [Applegate, Hinder, Lu, Lubin, 2023] PDHG computes an  $\varepsilon$ -optimal solution within

$$O\left((\|x^*\| + \|y^*\|) \cdot \|A\| \cdot H(K) \cdot \log\left(\frac{\|x^*\| + \|y^*\|}{\varepsilon}\right)\right)$$

iterations.

$H(K)$  is the global Hoffman constant of the matrix  $K$  of the KKT system

$$K = \begin{pmatrix} A & & & \\ -A & & & \\ & A^\top & & \\ c^\top & & -b^\top & \end{pmatrix} \quad H(K) \approx \frac{1}{\min_{J \subseteq \{1, \dots, 2m+n+1\}} \sigma_{\min}^+(K_J)}$$

Likely way too loose, and very hard to compute/validate/analyze.

## Challenge II: Loose/unworkable computational guarantees

Existing computational guarantees:

**Theorem** [Applegate, Hinder, Lu, Lubin, 2023] PDHG computes an  $\varepsilon$ -optimal solution within

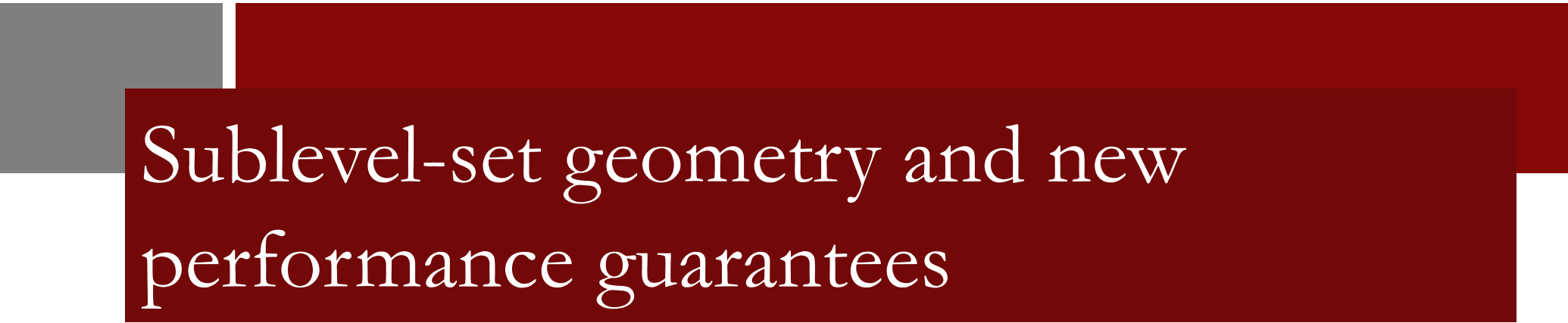
$$O\left(\left(\|x^*\| + \|y^*\|\right) \cdot \|A\| \cdot \mathbf{H}(\mathbf{K}) \cdot \log\left(\frac{\|x^*\| + \|y^*\|}{\varepsilon}\right)\right)$$

iterations.

Key questions:

- What conditions of the problem actually drive the performance of PDHG? **Sublevel-set Geometry**
- Can we improve these condition numbers and so improve computational performance in theory/practice?

**Yes, we will improve the geometry using Hessian rescaling**



# Sublevel-set geometry and new performance guarantees

# Primal-Dual Slack Space

Primal

$$\min_{\boldsymbol{x}} c^{\top} \boldsymbol{x}$$

$$\text{s. t. } A\boldsymbol{x} = \boldsymbol{b}$$

$$\boldsymbol{x} \in \mathcal{K}$$

Dual

$$\max_{\boldsymbol{y}, \boldsymbol{s}} b^{\top} \boldsymbol{y}$$

$$\text{s. t. } A^{\top} \boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}$$

$$\boldsymbol{s} \in \mathcal{K}^*$$

The “primal-dual slack-space variable” is  $\boldsymbol{w}$  :

$\boldsymbol{w} := (\boldsymbol{x}, \boldsymbol{s})$  are primal/dual feasible slacks

Duality gap:  $\text{Gap}(\boldsymbol{x}, \boldsymbol{s}) = c^{\top} \boldsymbol{x} - b^{\top} \boldsymbol{y}$

(which is a linear function of  $\boldsymbol{x}$  and  $\boldsymbol{s}$ )

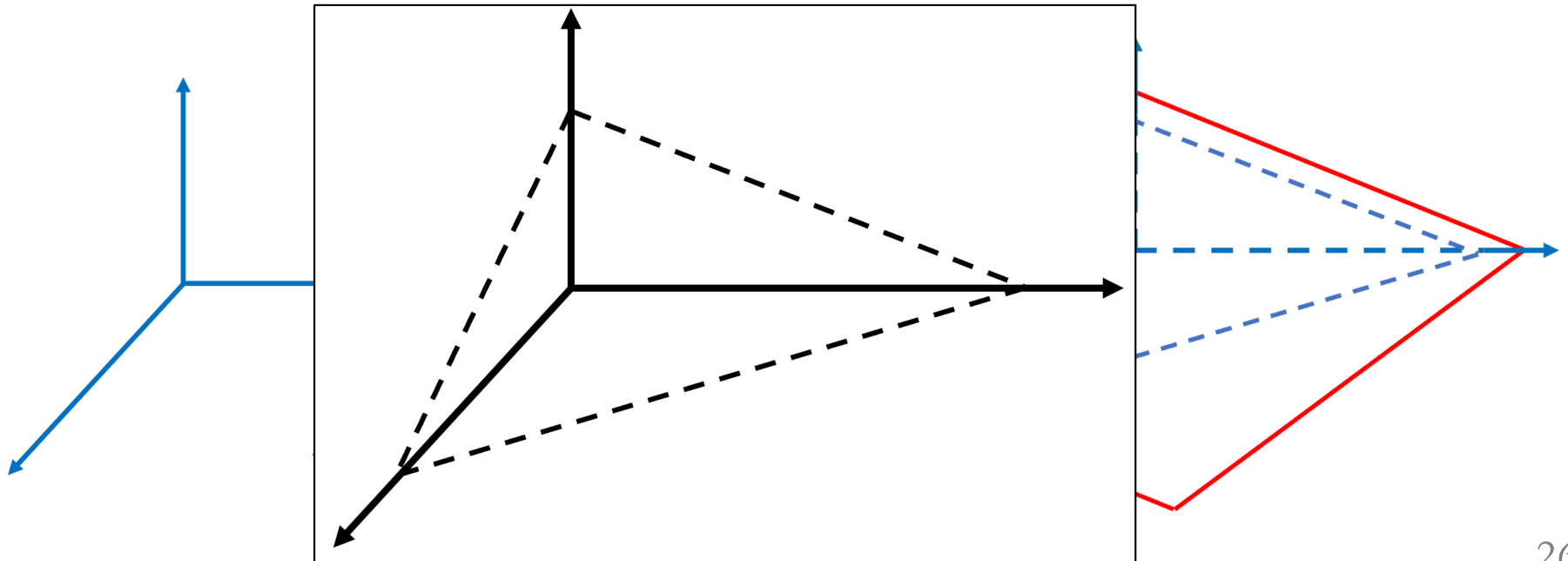
# The feasible primal-dual slack-space variables

$$\begin{array}{ll} \min_x & c^\top x \\ \text{s. t.} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

$$\begin{array}{ll} \max_{y,s} & b^\top y \\ \text{s. t.} & A^\top y + s = c \\ & s \in \mathcal{K}^* \end{array}$$

$(x, s)$  in the primal and dual cone for CLP

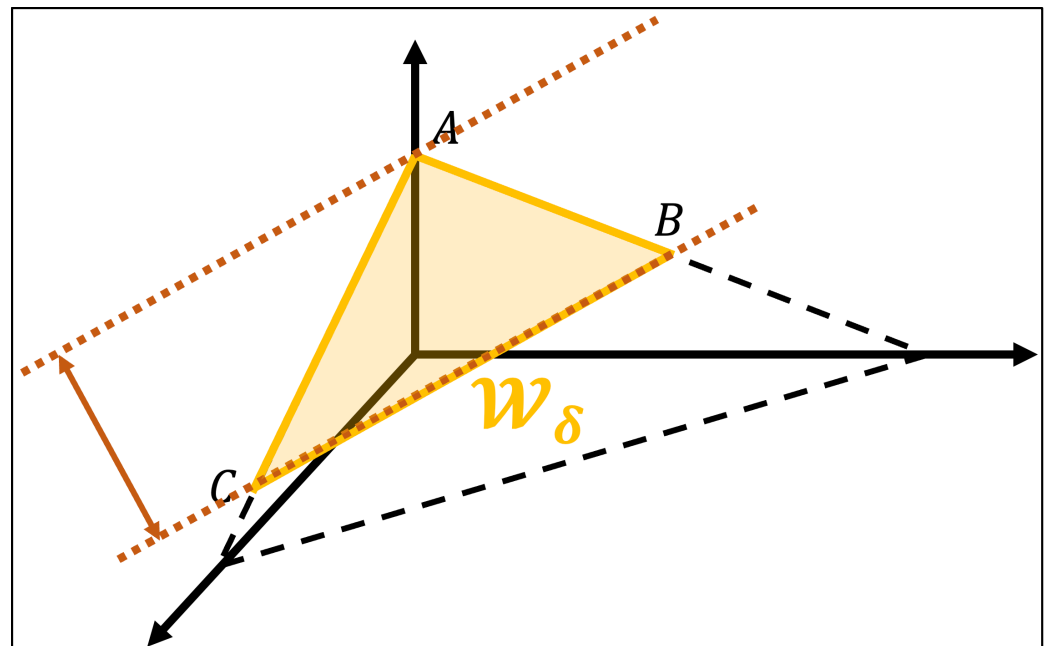
$(x, s)$  lies in an affine subspace



# Primal-Dual Slack Sublevel Set

$$\mathcal{W}_\delta := \left\{ w := (x, s) \mid \begin{array}{l} w \text{ is primal/dual feasible} \\ \mathbf{Gap}(w) \leq \delta \end{array} \right\}$$

Note:  $\mathcal{W}_0 = \mathcal{W}^*$



# Worst-case complexity of PDHG (under unique optima)

**Theorem** [Xiong and F 2024]: Suppose  $w^*$  is unique. PDHG computes an  $\varepsilon$ -optimal solution within

$$\tilde{O} \left( \kappa \cdot \lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta} \cdot \ln \left( \frac{1}{\varepsilon} \right) \right)$$

iterations.

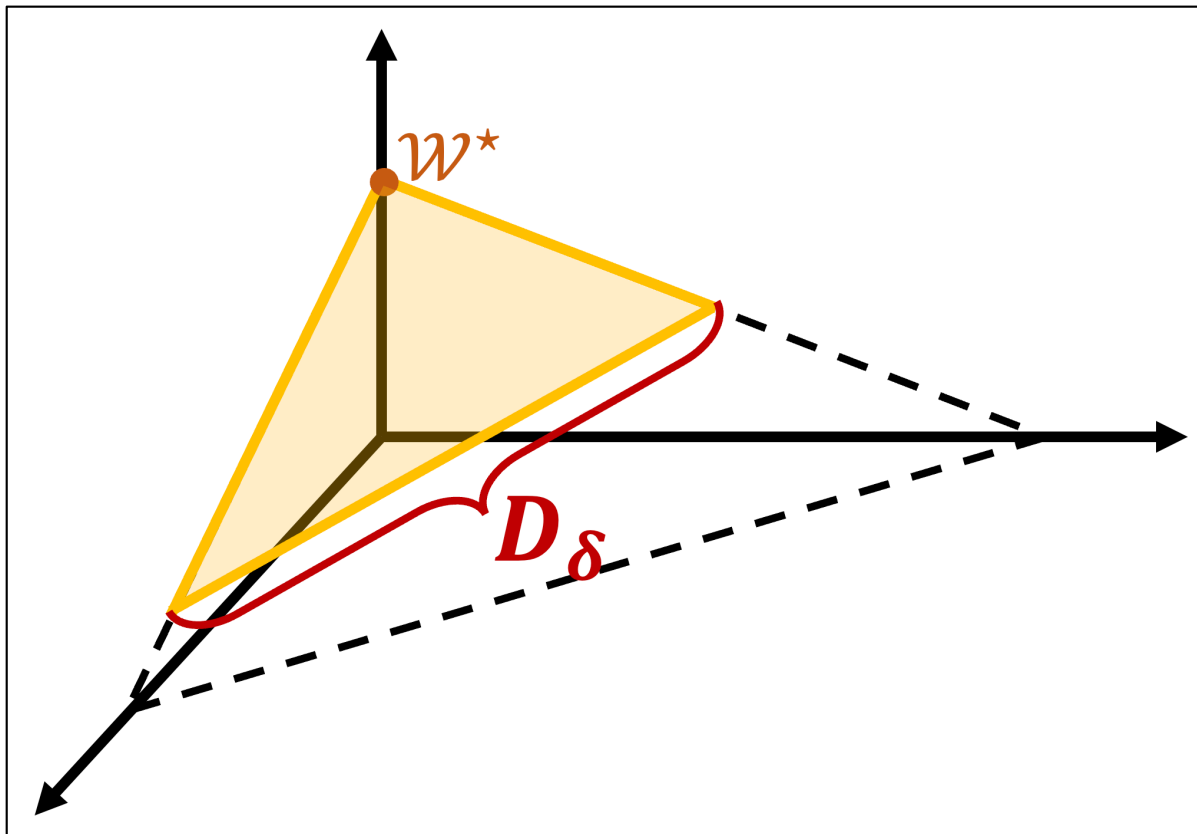
**Matrix** condition number of  $A$ :

$$\kappa := \sigma_{\max}^+(A) / \sigma_{\min}^+(A)$$

“Sublevel-set geometry”

$D_\delta$ : Diameter of  $\delta$ -sublevel set  $\mathcal{W}_\delta$

$$D_\delta := \max_{\bar{w}, \hat{w} \in \mathcal{W}_\delta} \|\bar{w} - \hat{w}\|$$

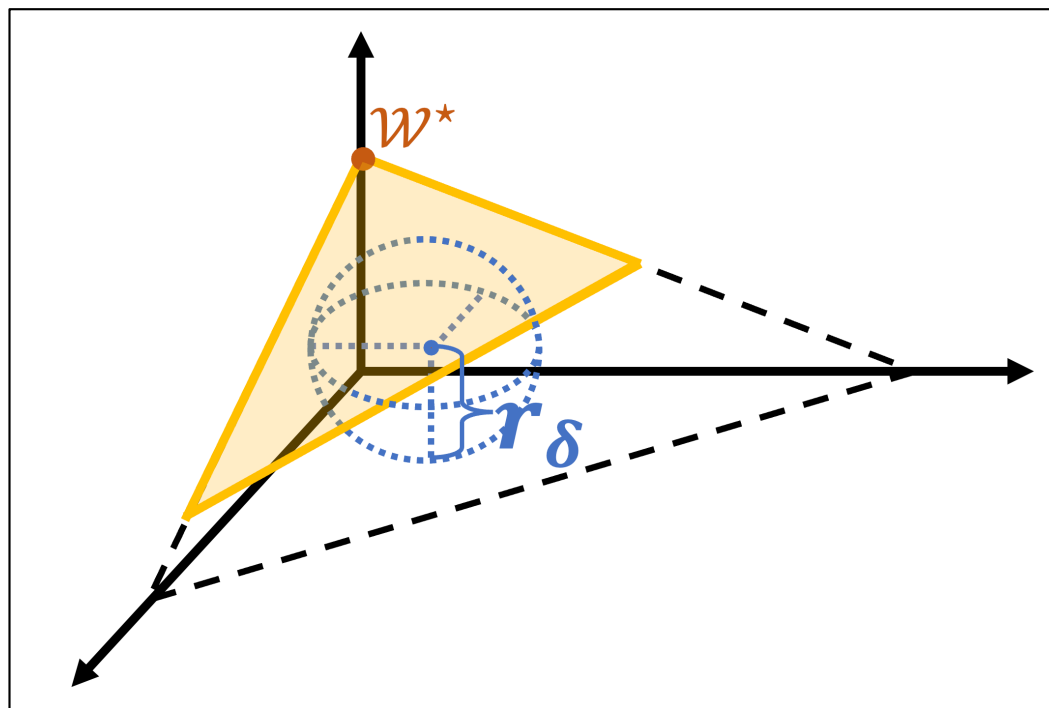


# $r_\delta$ : “Conic Radius” of $\mathcal{W}_\delta$

$$r_\delta := \max_{r \geq 0, w \in \mathcal{W}_\delta} r$$

s.t.  $B_w(r) \subset R_+^{2n}$

$r_\delta$  is the radius of the maximum ball inscribed in  $R_+^{2n}$  and centered at a point in  $\mathcal{W}_\delta$



# Target: $\varepsilon$ -optimal solution

$(x, s)$  is an  $\varepsilon$ -optimal solution if:

- distance to each type of constraint is no larger than  $\varepsilon$ , and
- the duality gap is not larger than  $\varepsilon$

$(x, s)$  is an  $\varepsilon$ -optimal solution if:

- $\text{Dist}(x, \{x \mid Ax = b\}) \leq \varepsilon$
- $\text{Dist}(x, \mathcal{K}) \leq \varepsilon$
- $\text{Dist}(s, \{s \mid \exists y \text{ s. t. } A^\top y + s = c\}) \leq \varepsilon$
- $\text{Dist}(s, \mathcal{K}^*) \leq \varepsilon$
- $c^\top x - b^\top (AA^\top)^{-1} A(c - s) \leq \varepsilon$

# Worst-case complexity of PDHG (under unique optima)

**Theorem** [Xiong and F 2024]: Suppose  $w^*$  is unique. PDHG computes an  $\varepsilon$ -optimal solution within

$$\tilde{O} \left( \kappa \cdot \lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta} \cdot \ln \left( \frac{1}{\varepsilon} \right) \right)$$

iterations.

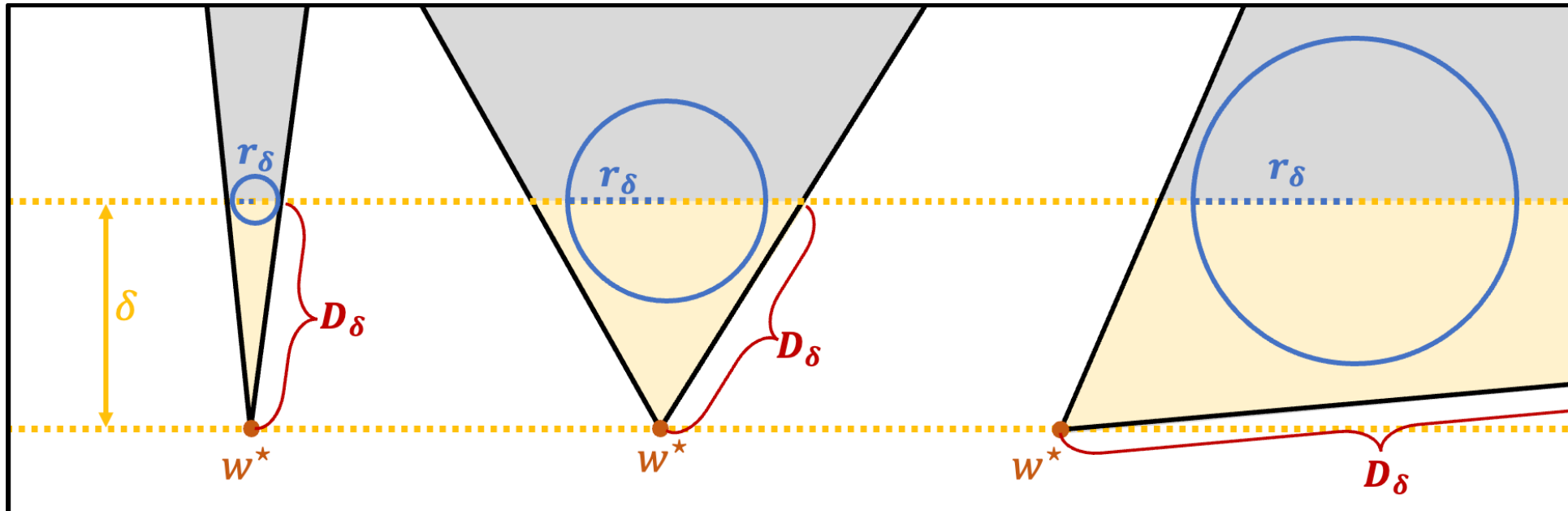
**Matrix** condition number of  $A$ :

$$\kappa := \sigma_{\max}^+(A) / \sigma_{\min}^+(A)$$

“Sublevel-set geometry”

# Local Geometry of and $\lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta}$ in the case of LP

When  $w^*$  is unique and  $\delta$  is sufficiently small,  $\mathcal{W}_\delta$  is a slice of a pointed cone at  $w^*$ .



Very small  $r_\delta$   
Intermediate  $D_\delta$



Intermediate  $r_\delta$   
Intermediate  $D_\delta$



Intermediate  $r_\delta$   
Very large  $D_\delta$

# Worst-case complexity of PDHG (under unique optima)

Matrix condition number of  $A$ :  $\kappa = \sigma_{\max}^+(A)/\sigma_{\min}^+(A)$

**Theorem** [Xiong and F 2024]: Suppose  $w^*$  is unique. PDHG computes an  $\varepsilon$ -optimal solution within

$$\tilde{O} \left( \kappa \cdot \lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta} \cdot \ln \left( \frac{1}{\varepsilon} \right) \right)$$

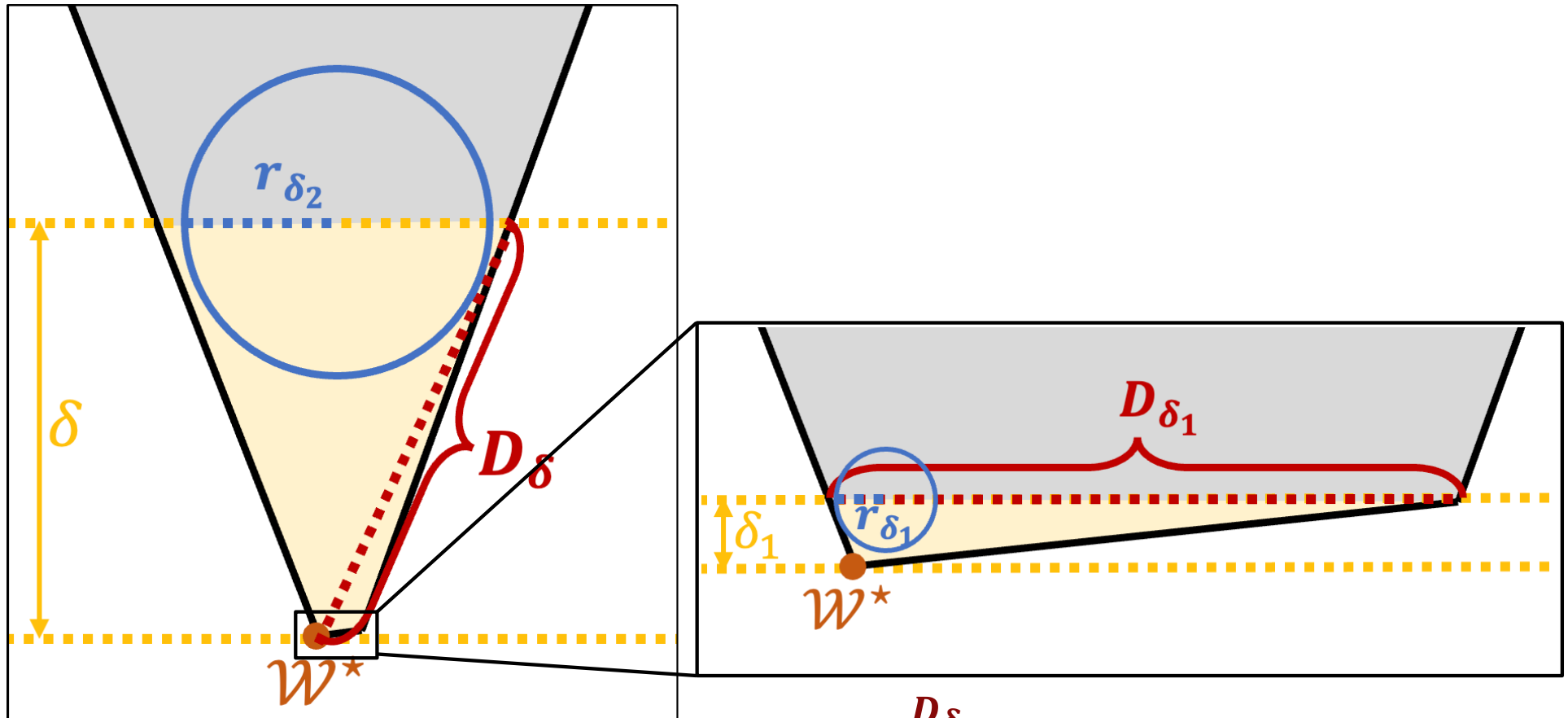
iterations.

Matrix condition number

Local geometric condition

- This iteration bound is “superior” to the Hoffman constant
- For LP, this bound is  $\tilde{O} \left( n^{2.5} \cdot \ln \left( \frac{1}{\varepsilon} \right) \right)$  with high probability [Xiong, 2024]
- For LP, this bound has a closed-form expression [Xiong, 2024]

# Another example

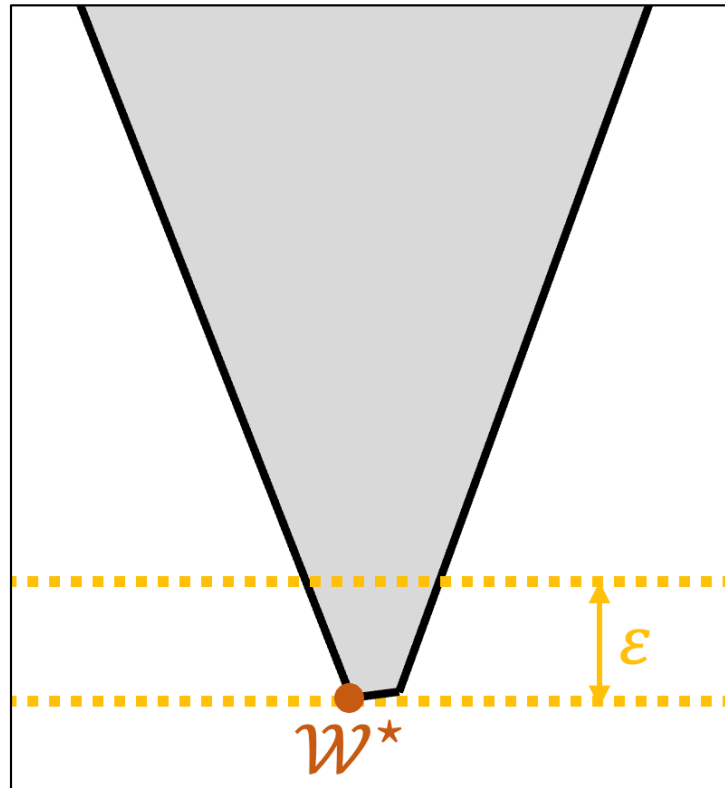


For  $\delta_2 > \delta_1$ ,  $\frac{D_{\delta_2}}{r_{\delta_2}}$  becomes smaller/better

$\frac{D_{\delta_1}}{r_{\delta_1}}$  is very large/bad  
(due to the small  $r_{\delta_1}$ )

Is  $\lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta}$  the only geometric condition?

Suppose we want an  $\varepsilon$ -optimal solution:



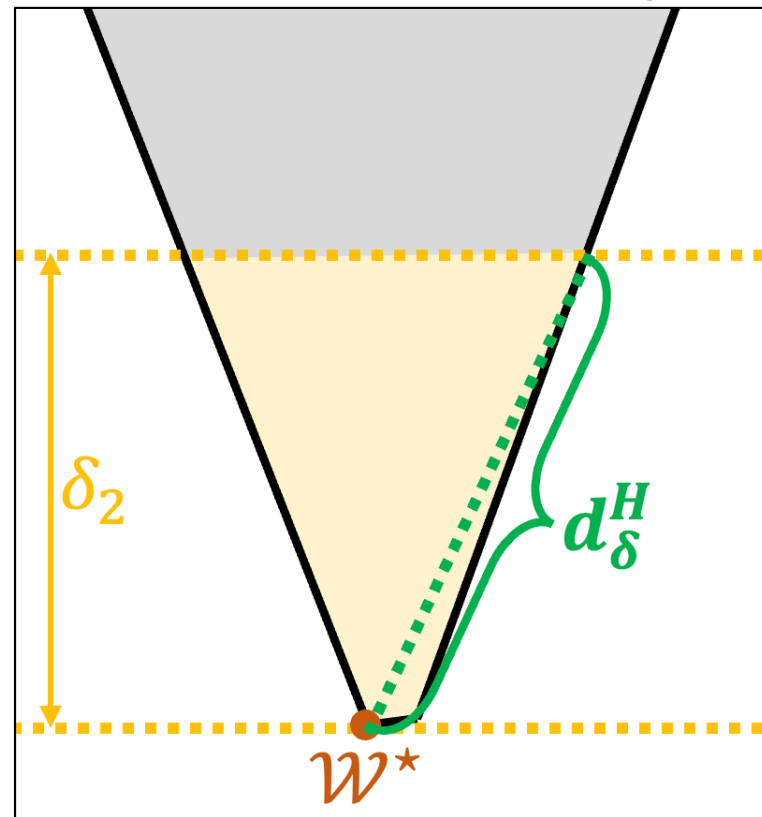
Intuition: The very-local bad geometry should not have a significant impact when the iterates of the algorithm have not yet reached the local neighborhood.

Is  $\lim_{\delta \rightarrow 0} \frac{D_\delta}{r_\delta}$  the only geometric condition?

We will add a third geometric measure to define “being close to  $\mathcal{W}^*$ ”

$$d_\delta^H := \max_{w \in \mathcal{W}_\delta} \text{Dist}(w, \mathcal{W}^*)$$

Hausdorff distance from  $\mathcal{W}_\delta$  to  $\mathcal{W}^*$



# Our General Conic Optimization Computational Guarantee

**Theorem** [Xiong and F 2024]: The number of PDHG iterations required to compute an  $\varepsilon$ -optimal solution is upper bounded by:

$$T_\delta := \kappa \cdot \max \left\{ \frac{D_\delta}{r_\delta} \cdot \ln \left( \frac{1}{\varepsilon} \right), \frac{d_\delta^H}{\varepsilon} (1 + \text{Dist}(0, \mathcal{W}^*)) \right\}$$

for each  $\delta > 0$ .

How good the geometry of  $\mathcal{W}_\delta$  is

How close  $\mathcal{W}_\delta$  is to  $\mathcal{W}^*$

Remark: This result holds under multiple optima and general conic optimization.

$D_\delta$ : Diameter of  $\mathcal{W}_\delta$

$r_\delta$ : Conic radius of  $\mathcal{W}_\delta$

$d_\delta^H$ : Hausdorff distance from  $\mathcal{W}_\delta$  to  $\mathcal{W}^*$

$\kappa = \sigma_{\max}^+(A) / \sigma_{\min}^+(A)$  (the standard condition number of  $A$ )

# Our General Conic Optimization Computational Guarantee

**Theorem** [Xiong and F 2024]: The number of PDHG iterations required to compute an  $\varepsilon$ -optimal solution is upper bounded by:

$$\tilde{O} \left( \inf_{\delta > 0} T_\delta := \kappa \cdot \max \left\{ \frac{D_\delta}{r_\delta} \cdot \ln \left( \frac{1}{\varepsilon} \right), \frac{d_\delta^H}{\varepsilon} (1 + \text{Dist}(0, \mathcal{W}^*)) \right\} \right)$$

- Linear convergence part
- Note that  $D_\delta/r_\delta$  might/not be large when  $\delta$  is small

The tightest bound is given by the  $T_\delta$  that minimizes the bound 😊

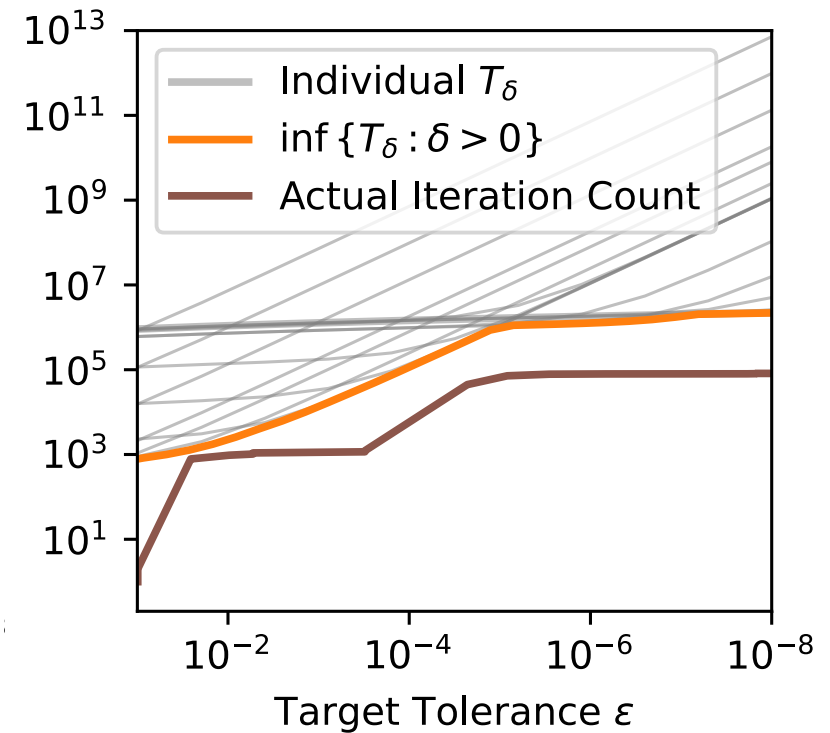
- Sublinear convergence part
- Note  $d_\delta^H$  is small when  $\delta$  is small

$D_\delta$ : Diameter of  $\mathcal{W}_\delta$   
 $r_\delta$ : Conic radius of  $\mathcal{W}_\delta$   
 $d_\delta^H$ : Hausdorff distance between  $\mathcal{W}_\delta$  and  $\mathcal{W}^*$   
 $\kappa = \sigma_{\max}^+(A)/\sigma_{\min}^+(A)$

# $T_\delta$ and $\inf\{T_\delta : \delta > 0\}$

$$\begin{aligned} \min_{x=(x_1, x_2, x_3)} \quad & 0.20001 \cdot x_1 + x_2 + 1.0001 \cdot x_3 \\ \text{s.t.} \quad & -10x_1 + x_2 + x_3 = 1 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{aligned}$$

- Different  $\delta$  yield different bounds  $T_\delta$
- For each tolerance  $\varepsilon$ ,  $\inf\{T_\delta : \delta > 0\}$  provides the best bound
- In the beginning, PDHG converges sublinearly
- Since  $\lim_{\delta \searrow 0} \frac{D_\delta}{r_\delta} < \infty$  and  $\lim_{\delta \searrow 0} d_\delta^H = 0$ , we eventually obtain linear convergence
- Validated by actual iteration count



# Our General Conic Optimization Computational Guarantee

**Theorem** [Xiong and F 2024]: The number of PDHG iterations required to compute an  $\varepsilon$ -optimal solution is upper bounded by:

$$\tilde{O} \left( \inf_{\delta > 0} T_{\delta} := \kappa \cdot \max \left\{ \frac{D_{\delta}}{r_{\delta}} \cdot \ln \left( \frac{1}{\varepsilon} \right), \frac{d_{\delta}^H}{\varepsilon} (1 + \text{Dist}(0, \mathcal{W}^*)) \right\} \right) .$$

**Corollary:** If there exists  $\delta > 0$  whose  $\delta$ -sublevel set satisfies:

- $\frac{D_{\delta}}{r_{\delta}}$  is small ( $\mathcal{W}_{\delta}$  has good geometry), and
- $d_{\delta}^H$  is small ( $\mathcal{W}_{\delta}$  is close to  $\mathcal{W}^*$ ),

then PDHG will converge faster. (If not, rPDHG might be slow...)



Q: Can we possibly improve  $\frac{D_{\delta}}{r_{\delta}}$  and  $d_{\delta}^H$  ?

**A: Yes, we can do so by using Hessian Rescaling**

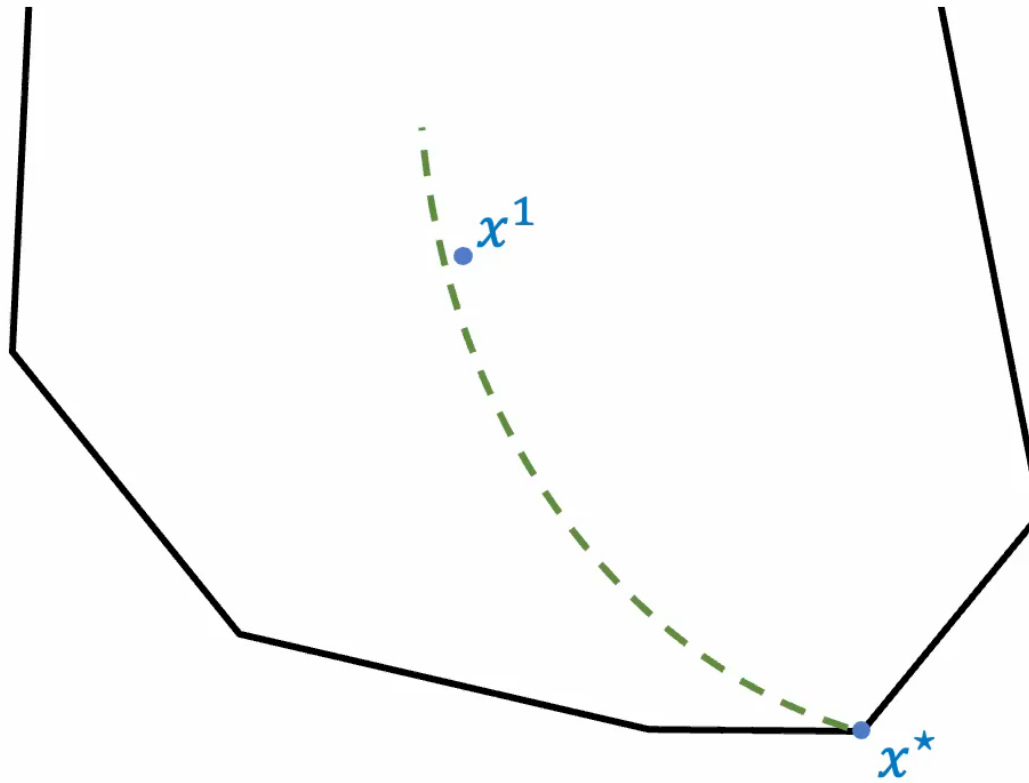
$D_{\delta}$ : Diameter of ....

$r_{\delta}$ : Conic radius of ...

$d_{\delta}^H$ : Hausdorff distance...

Using IPM theory to develop practical  
computational speed-ups of PDHG

# Recall the two slides on IPMs



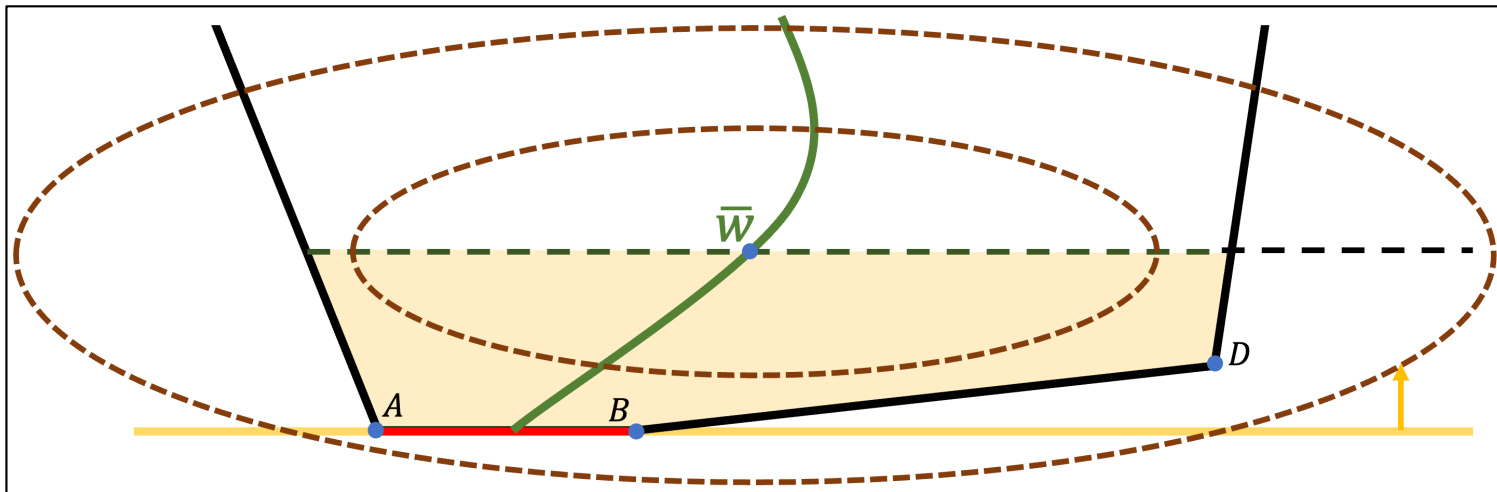
# Central-Path ellipsoids have remarkable properties

Central-path solutions are:

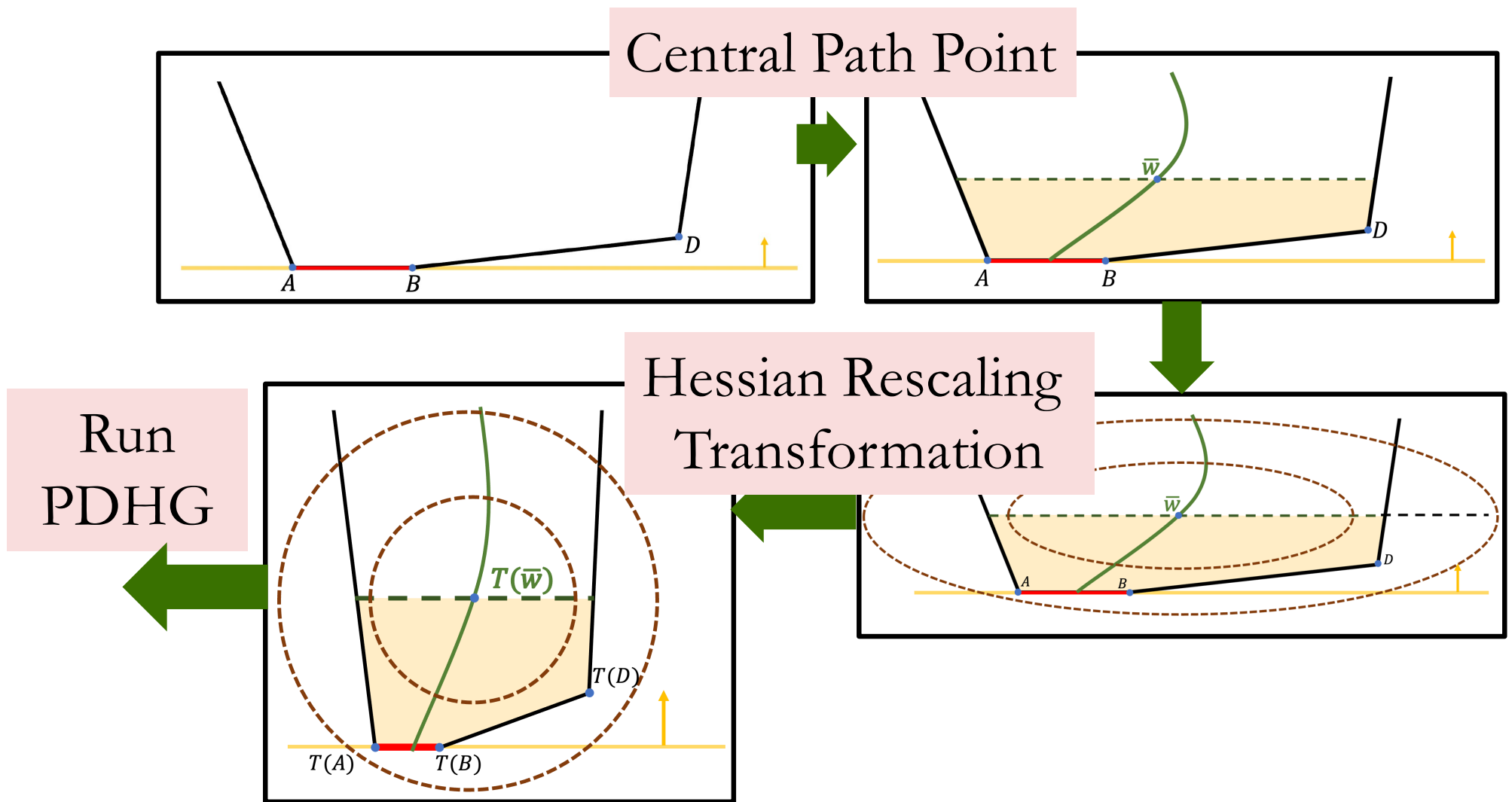
$$x_\mu := \arg \min_x c^\top x + \mu \cdot F(x) \\ \text{s. t. } Ax = b, x \in \mathcal{K}$$

$$y_\mu, s_\mu := \arg \max_{y,s} b^\top y - \mu \cdot F^*(s) \\ \text{s. t. } A^\top y + s = c, s \in \mathcal{K}^*$$

Example for LP:  $F(x) := -\sum_{i=1}^n \ln(x_i)$ ,  $\vartheta_F = n$



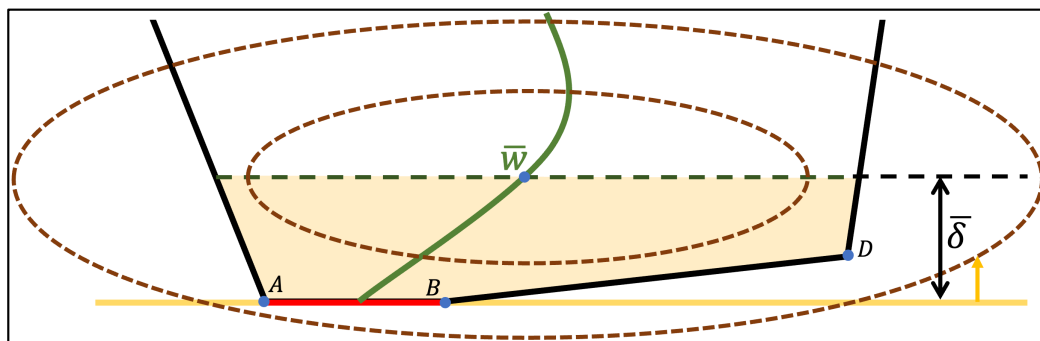
This suggests the following strategy:



# Transformation based on a central-path solution

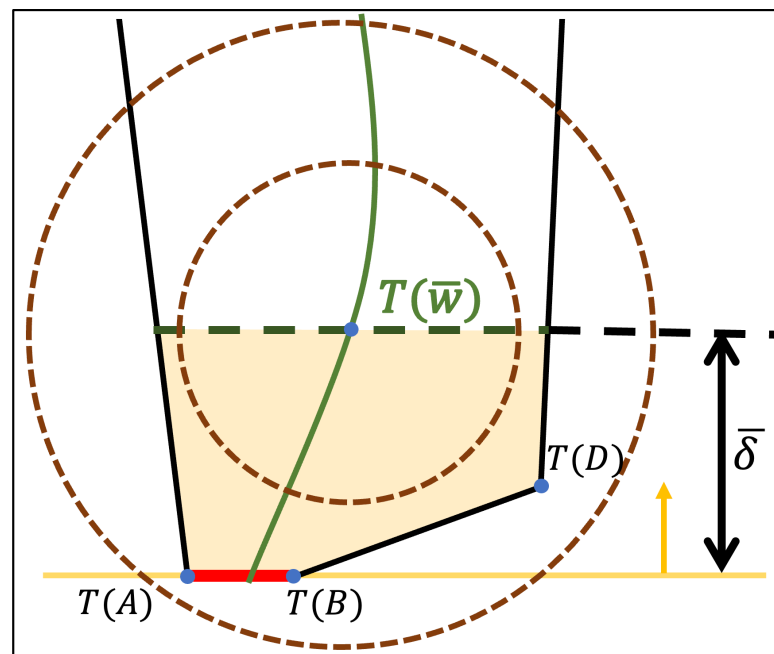
Let  $H := \mu \cdot \nabla^2 F(x_\mu)$ , then

- $\bar{\delta} := n\mu = \text{Gap}(\bar{w})$
- $D_{\bar{\delta}} \leq 2n \cdot \sigma_{\min}^+(H)^{-1/2}$
- $r_{\bar{\delta}} \geq \sigma_{\max}^+(H)^{-1/2}$
- $d_{\bar{\delta}}^H \leq 2n \cdot \sigma_{\min}^+(H)^{-1/2}$

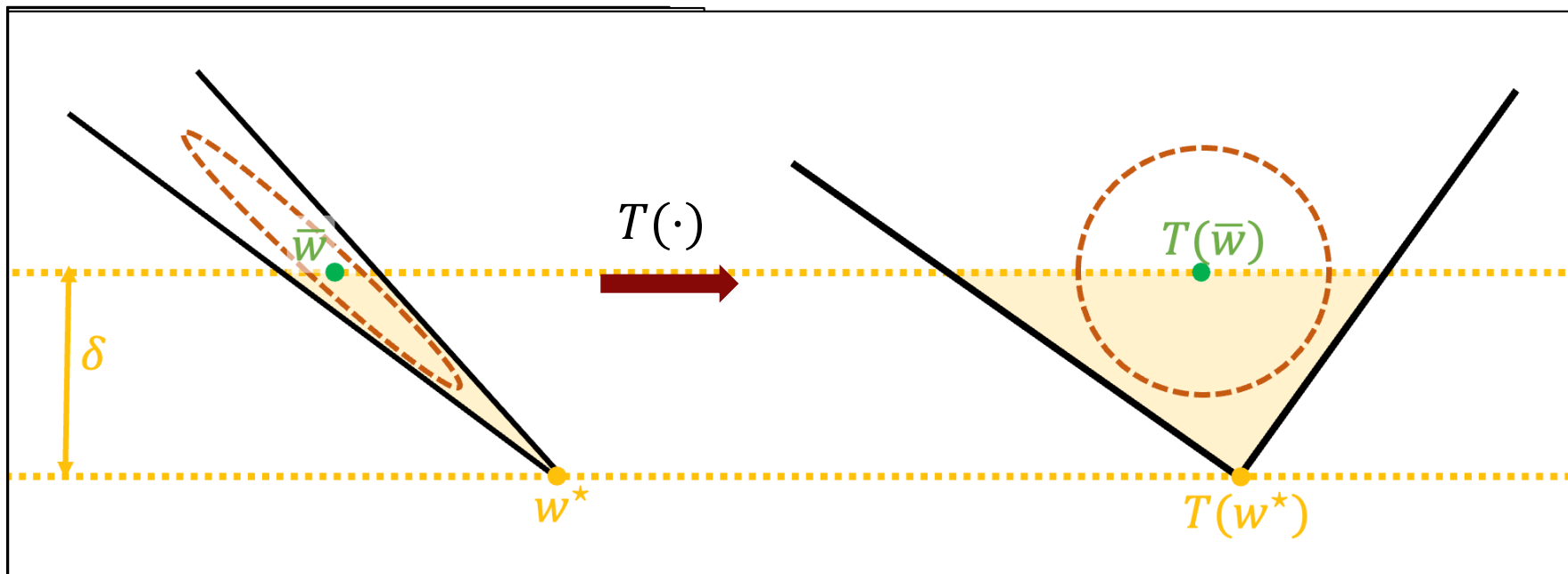


After a rescaling transformation (that maps the local-norm ball to a Euclidean norm ball) we have:

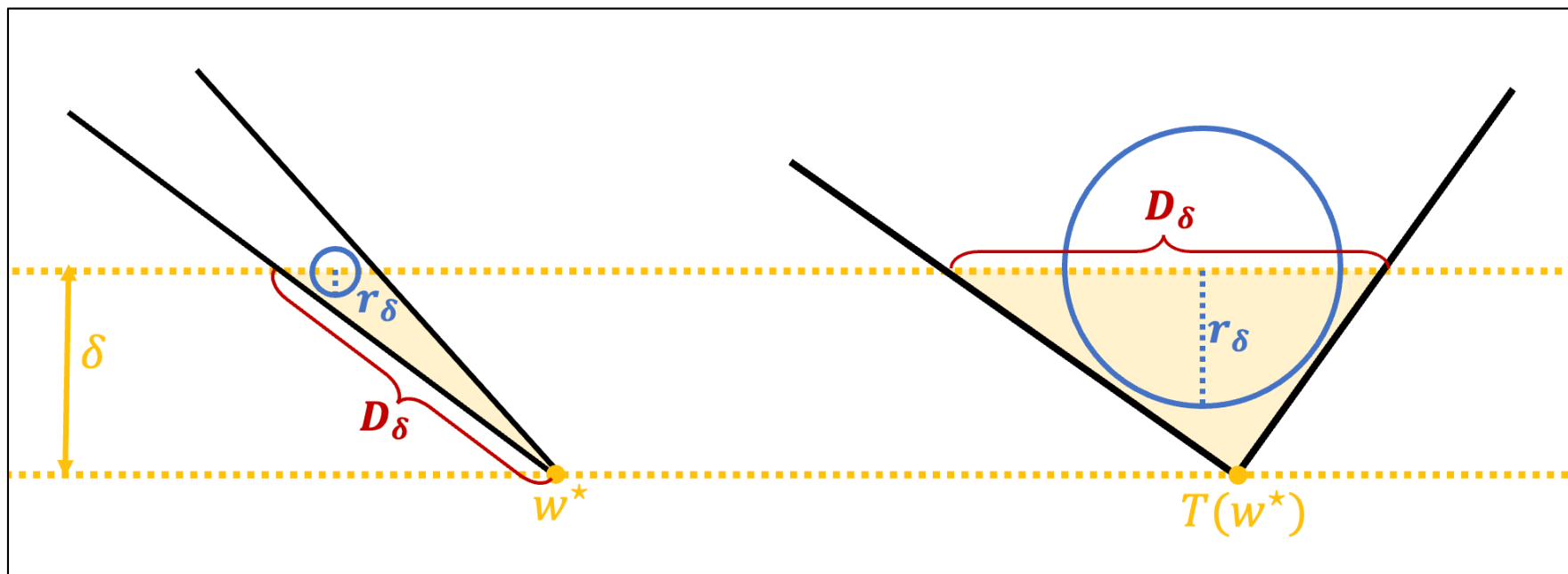
- $D_{\bar{\delta}} \leq 2\sqrt{n} \cdot \sqrt{\bar{\delta}}$
- $r_{\bar{\delta}} \geq \sqrt{\frac{1}{n}} \cdot \sqrt{\bar{\delta}}$
- $d_{\bar{\delta}}^H \leq 2\sqrt{n} \cdot \sqrt{\bar{\delta}}$
- $\frac{D_{\bar{\delta}}}{r_{\bar{\delta}}} \leq 2n$
- $d_{\bar{\delta}}^H$  is small if  $\bar{\delta}$  is small
- “Very nice theory”



# How does Hessian rescaling improve the geometry?



# How Hessian rescaling improves the geometry?



Very Small  $r_\delta$   
Intermediate  $D_\delta$   
Large  $\frac{D_\delta}{r_\delta}$



Intermediate  $r_\delta$   
Intermediate  $D_\delta$   
Small  $\frac{D_\delta}{r_\delta}$

# Complexity guarantee after central-path Hessian rescaling

Suppose we do the following:

1. rescaling transformation using a central-path solution **with duality gap  $\bar{\delta}$**
2. row transformation to try to decrease  $\kappa$  closer to  $\kappa = 1$

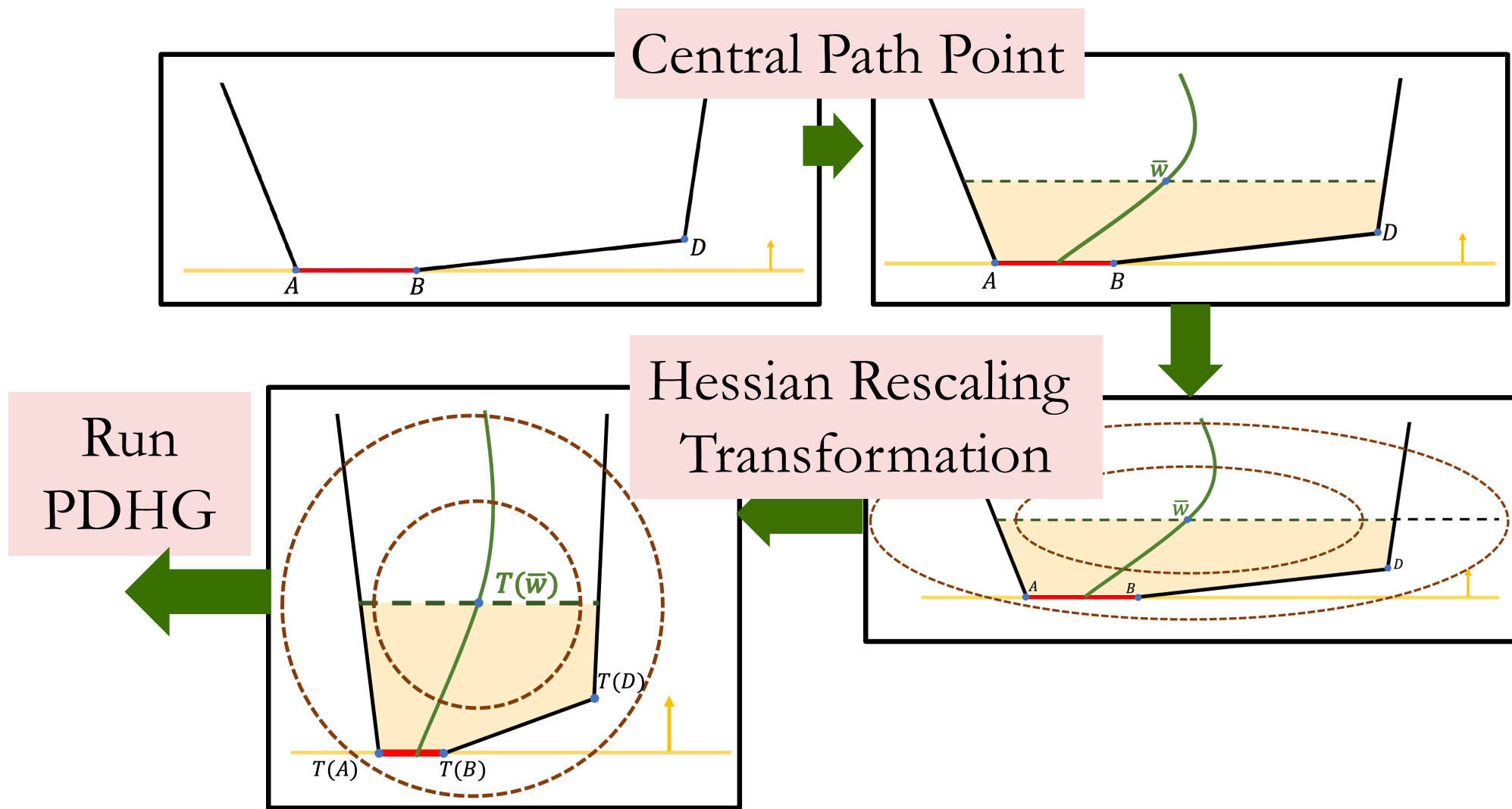
Then the number of PDHG iterations required to compute an  $\varepsilon$ -optimal solution of the original problem is upper bounded by:

$$\tilde{O} \left( n \cdot \left( \ln \left( \frac{1}{\varepsilon} \right) + \frac{D_{\bar{\delta}} + \bar{\delta}}{\varepsilon} \right) \right)$$

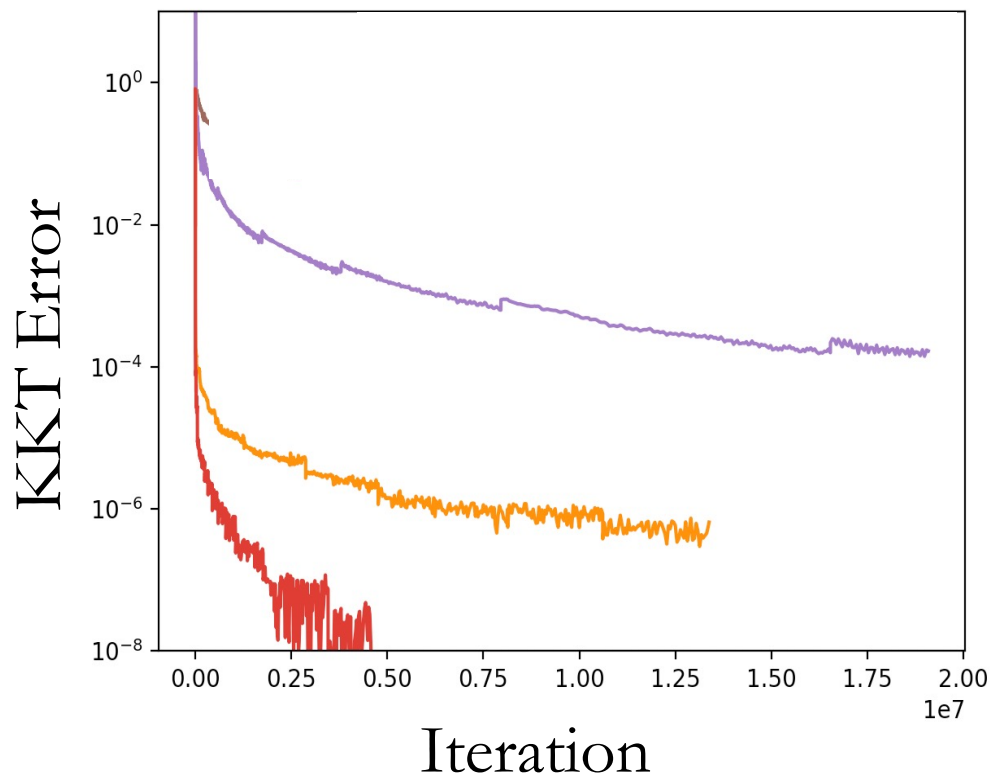
We have replaced  $\frac{D_{\delta}}{r_{\delta}}$  by  $n$

The smaller  $\bar{\delta}$  is, the faster the convergence

# Summary



# “Proof of concept” applied to problem instance **bmoipr2**



Here we pre-multiply  $A$  by  $(AA^T)^{-1/2}$  to yield  $\kappa = 1$

## PDHG-EasyColumn

Column rescaling to **normalize**  $L_\infty$  column norms

## PDHG-Central $\delta=0.1$

Column rescaling using **central-path solution** with KKT error **0.1**

## PDHG-Central $\delta=0.01$

Similar as above, with KKT error **0.01**

- Here the central path solutions were computed by MOSEK
- This is an admittedly “unfair” comparison, but it validates the potential of the overall approach...

# Proof of concept, continued

## PDHG-EasyColumn

Column rescaling to **normalize**  $L_\infty$  column norms

## PDHG-Central $\delta=0.1$

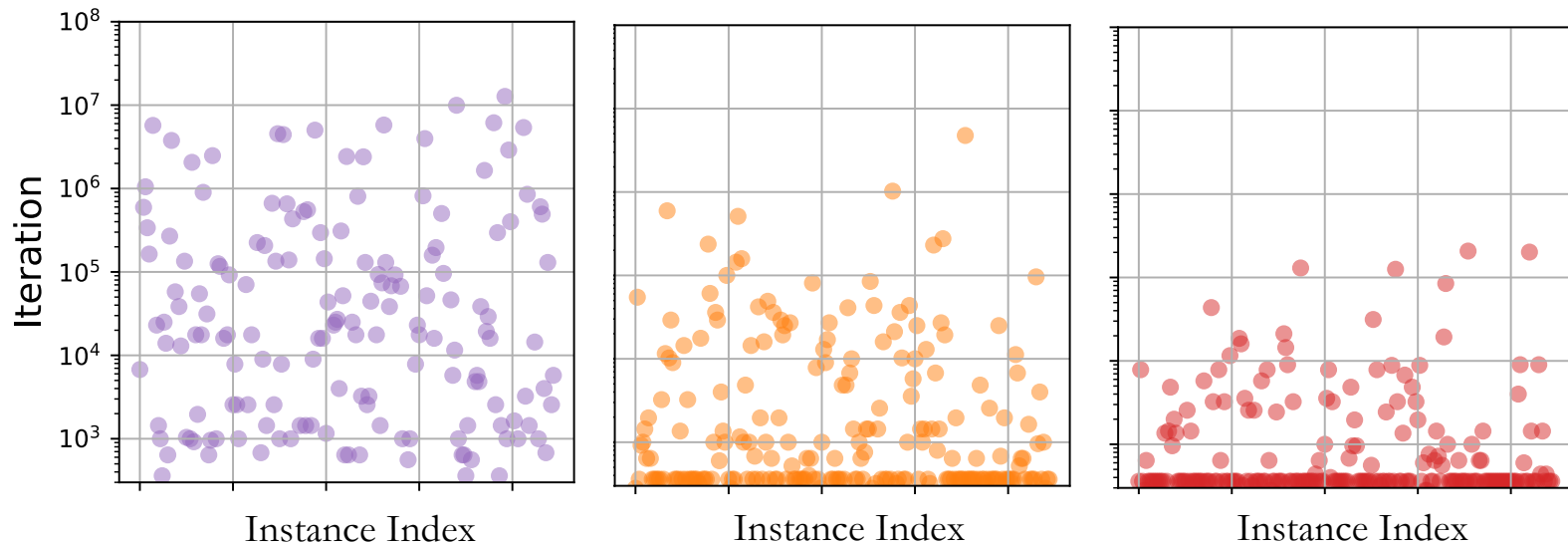
Column rescaling using **central-path solution** with KKT error **0.1**

## PDHG-Central $\delta=0.01$

Column rescaling using **central-path solution** with KKT error **0.01**

Note: we pre-multiply  $A$  by  $(AA^\top)^{-1/2}$  to yield  $\kappa = 1$  for all rescalings

PDHG iterations needed to compute a solution with KKT error  $10^{-8}$

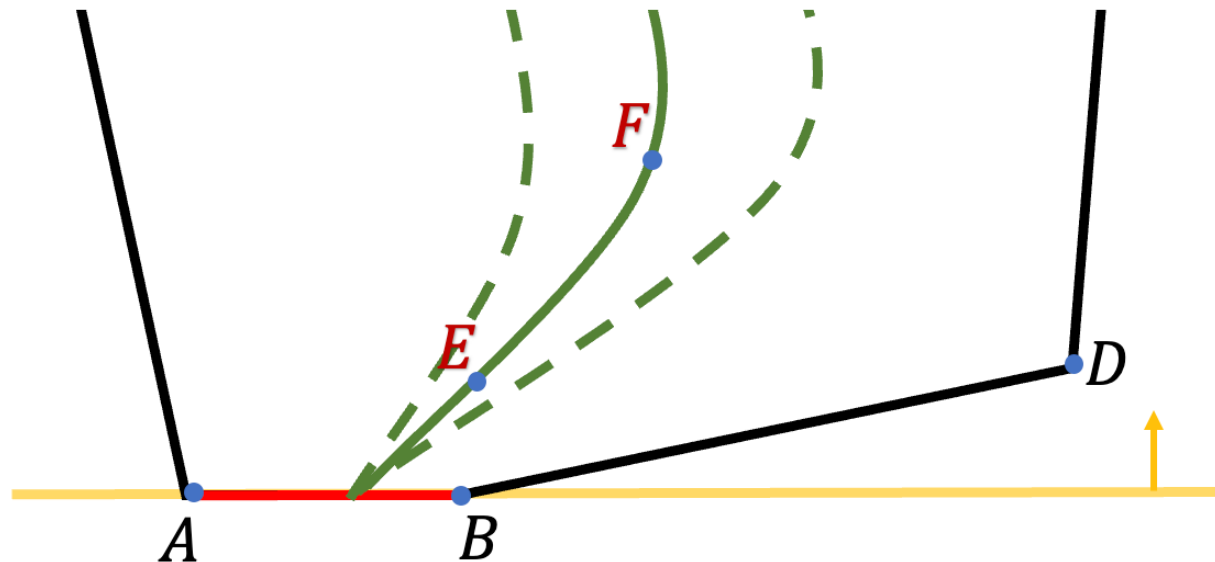


222 LP relaxation instances from the MIPLIB 2017 dataset

PDHG-AHR (“Adaptive Hessian Rescaling”)  
and  
Computational Experiments

# Main Strategy

Main strategy: use a “CG-IPM” to compute a low-accuracy central-path solution to obtain a good rescaling, and then use PDHG

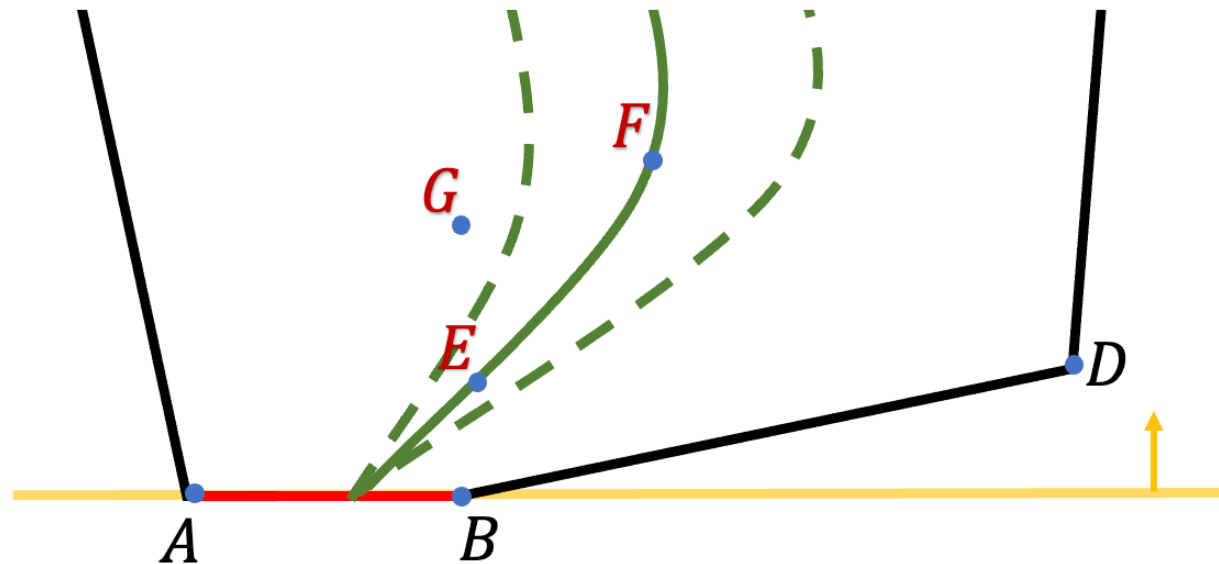


# Main Strategy, continued

Main strategy: use a “CG-IPM” to compute a low-accuracy central-path solution to obtain a good rescaling, and then use PDHG

- **We use a first-order method to compute the Newton steps of a “central path” solution**
  - we use a conjugate-gradient-method-based IPM (**CG-IPM**)
  - otherwise, implementation exactly follows Nocedal and Wright *Numerical Optimization* (2006)
  - but we solve the normal equations using conjugate gradient method
- **We employ only diagonal row rescaling to try to improve  $\kappa$** 
  - Column rescaling uses the central-path Hessian of  $w_{int}$  followed by PDLP’s rescaling (Ruiz rescaling and Pock-Chambolle rescaling), which we call the “ **$w_{int}$ -rescaled problem**”

# Using Adaptive Hessian Rescaling



Motivating concepts of Adaptive Hessian Rescaling:

- Adaptively balance the cost of computing the rescaling (CG-IPM) with the savings from running PDHG on the rescaled instance
- Try to identify a “good-enough” rescaling as early as possible and use the good-enough rescaling

# Computational Experiments with **PDHG-AHR**

We compare:

1. **PDHG-AHR**: PDHG with Adaptive central-path Hessian Rescaling (using CG-IPM for the central-path computations)
2. **PDHG(RuizPC)**: use heuristic Ruiz rescaling on  $\mathbf{A}$ , followed by Pock-Chambolle rescaling. (This is the same as the rescaling used in PDLP.)
3. **IPM**: a home-grown standard primal-dual predictor-corrector interior point method, straight from Nocedal and Wright *Numerical Optimization* (2006).

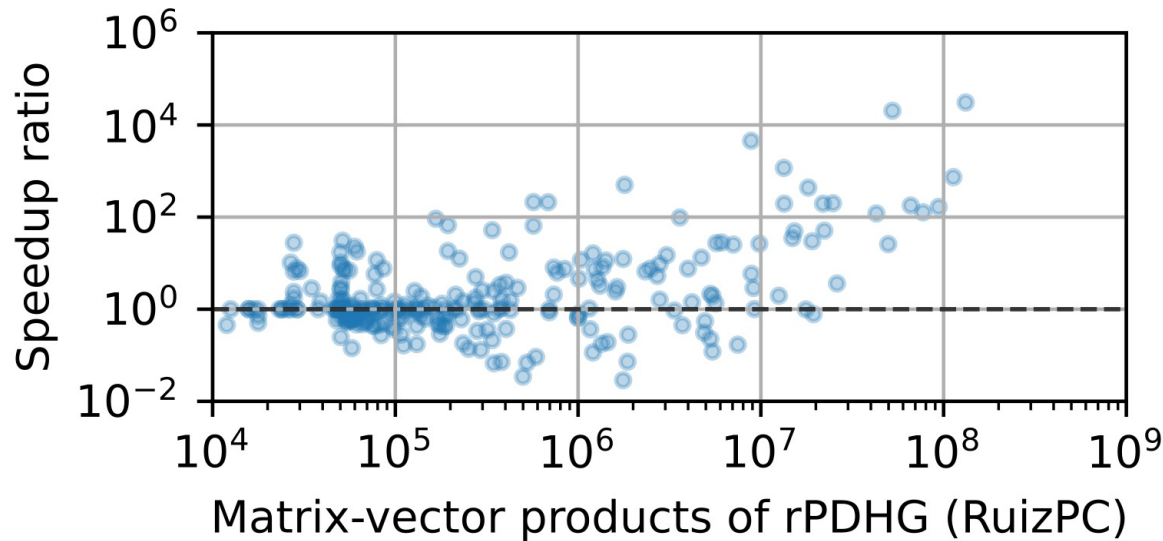
We performed tests on all the LP relaxations from MIPLIB 2017 dataset that are:

- large enough ( $m \times n > 10^6$ )
- but not too large for the IPM (number of non-zeros  $< 10^5$ )
- This yielded 413 instances in total

# Computational Comparison: PDHG-AHR and PDHG(RuizPC)

Speedup Ratio:

$$\frac{\text{Matrix-vector products PDHG(RuizPC)}}{\text{Matrix-vector products PDHG-AHR}}$$



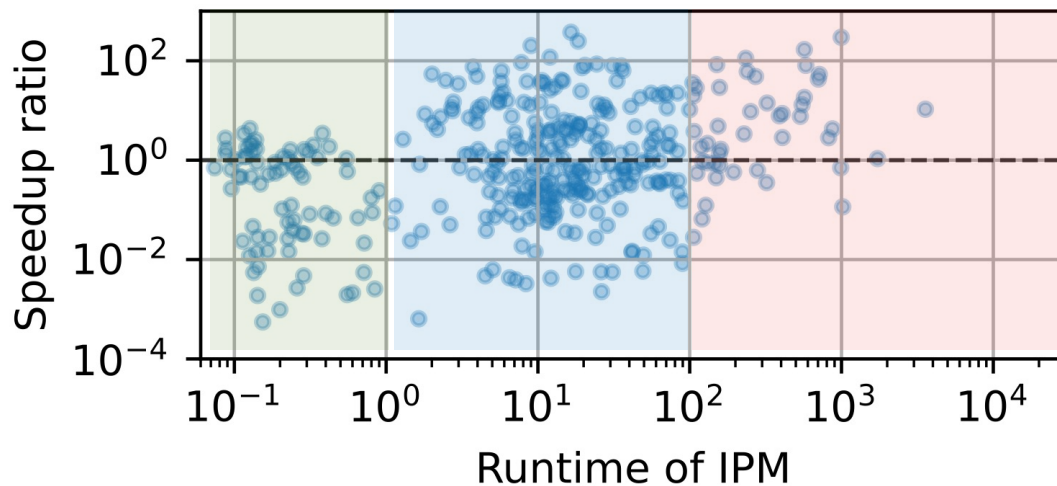
		Fraction solved by PDHG-AHR	
		S	N
Fraction solved by PDHG (RuizPC)	S	87.7%	1.7%
	N	7.5%	3.1%

- In general, the harder the problem is for PDHG(RuizPC), the larger the speedups from using PDHG-AHR
- PDHG-AHR solves about 95.2% of problems and is more reliable than PDHG(RuizPC) (solves about 89.4% of problems)

# Computational Comparison: PDHG-AHR and IPM

Speedup Ratio:

$$\frac{\text{Runtime IPM}}{\text{Runtime PDHG-AHR}}$$



		Fraction solved by PDHG-AHR	
		S	N
Fraction solved by IPM	S	93.0%	1.5%
	N	2.2%	3.4%

- Generally speaking, the harder the problem is for IPM, the larger the speedups from using PDHG-AHR.
- PDHG-AHR is also (slightly) more reliable than IPM

# Recap, Takeaways, and Remarks

Recap and takeaways:

- The convergence rate of PDHG on CLP is related to the geometry of primal-dual sublevel sets measured with  $D_\delta, r_\delta, d_\delta^H$
- Rescaling using a central-path solution can improve the geometry of the primal-dual sublevel sets
- Our strategy: **PDHG-AHR** uses a “CG-IPM” to compute a low-accuracy central-path solution to obtain a good rescaling, and then use PDHG
- FOMs can compete and outperform IPMs

Remarks:

- Our results relied only on PDHG’s average iterate convergence and non-expansiveness properties. Similar results might also hold for other FOMs, in particular ADMM, EGM, ...



Thank you!