Accelerated Second-Order Methods for Convex and Nonconvex Optimization

MIIS, December 17, 2022

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today’s Talk

- Optimal Diagonal Precondition using SDP
- An Accelerated Second-Order Method Using Homogenized Descent Direction
- A Dimension Reduced Trust-Region Method for Unconstrained Optimization
- Potential Reduction Algorithm for Linear Programming

Optimal Diagonal Pre-Condition [QGHYZ 20]

Given $\boldsymbol{X}^\top \boldsymbol{X} \succ \mathbf{0}, \boldsymbol{X} \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}}$, iterative method (e.g., CG) is often applied to solve

$$(\boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{x} = \boldsymbol{b}$$

- **Convergence of iterative methods depends on condition number of** X
- **In practice we choose preconditioner** D_L, D_R **and solve** $(D_L X D_R)^\top (D_L X D_R) x' = b$
- **Diagonal** $D = \mathrm{diag}(d_1, \ldots, d_{\{m\ or\ n\}})$ **is called diagonal pre-conditioner**

More generally, we look for D_L, D_R **such that condition number of** $D_L X D_R$ **is minimized**

Is it possible to find optimal $\boldsymbol{D}_{\boldsymbol{L}}^*$ and $\boldsymbol{D}_{\boldsymbol{R}}^*$?

SDP works!

Optimal Diagonal Pre-Conditioner

$$\min_{D_R \succeq 0 \text{ diagonal}} \quad \kappa((XD_R)^\top (XD_R))$$

$$\begin{aligned} \min_{D_R \succeq 0 \text{ diagonal}, \kappa} \quad & \kappa \\ \text{subject to} \quad & I \preceq D_R X^\top X D_R \preceq \kappa I \end{aligned}$$

$$\begin{aligned} \min_{D_R \succeq 0 \text{ diagonal}, \kappa} \quad & \kappa \\ \text{subject to} \quad & D_R^{-2} \preceq X^\top X \preceq \kappa D_R^{-2} \end{aligned}$$

$$\begin{aligned} \min_{D_R \succeq 0 \text{ diagonal}, \kappa} \quad & \kappa \\ \text{subject to} \quad & D_R^{-2} \succeq \kappa^{-1} X^\top X \\ & D_R^{-2} \preceq X^\top X \end{aligned}$$

$$\begin{aligned} \max_{D_R^{-2} \succeq 0 \text{ diagonal}, \tau} \quad & \tau = \kappa^{-1} \\ \text{subject to} \quad & D_R^{-2} - \tau X^\top X \succeq 0 \\ & X^\top X - D_R^{-2} \succeq 0 \end{aligned}$$

- **Finding the optimal diagonal pre-conditioner is an SDP**
- **Two SDP blocks and sparse coefficient matrices**
- **Trivial dual interior-feasible solution**
- **An ideal formulation for dual SDP methods**
- **Similar trick applies to** $D_L X$

$$\begin{aligned} \min_{D_L \succeq 0 \text{ diagonal}, \kappa} \quad & \kappa \\ \text{subject to} \quad & I \preceq X^\top D_L^2 X \preceq \kappa I \end{aligned}$$

Two-Sided Optimal Pre-Conditioner

$$\min_{D_L, D_R \succeq 0 \text{ diagonal}} \quad \kappa(D_L X D_R)$$

- **Common in practice and popular heuristics exist**

  e.g. Ruiz-scaling, matrix equilibration & balancing
- **Not directly solvable using SDP**
- **Can be solved by *iteratively* fixing D_L, D_R, and optimizing the other side**

  Solving a sequence of SDPs
- **Computation cost of the preconditioner is often amortized by successive solves**
- **Answer a question: how far can diagonal pre-conditioners go**

Computational Results: Solving for the Optimal Pre-Conditioner

$$\begin{aligned} \min_{D,\kappa} \quad & \kappa \\ \text{subject to} \quad & D \preceq M \\ & \kappa D \succeq M \end{aligned}$$

$$\begin{aligned} \max_{\delta, d} \quad & \delta \\ \text{subject to} \quad & D - M \preceq 0 \\ & \delta M - D \preceq 0 \end{aligned}$$

SDP from optimal diagonal pre-conditioning problem

- **Perfectly in the dual form**
- **Trivial dual feasible interior point solution**
- **1 is an upper-bound for the optimal objective value**

HDSDP

- **A dual SDP algorithm (successor of DSDP5.8 by Benson)**
- **Support initial dual solution**
- **Customization for the diagonal pre-conditioner**

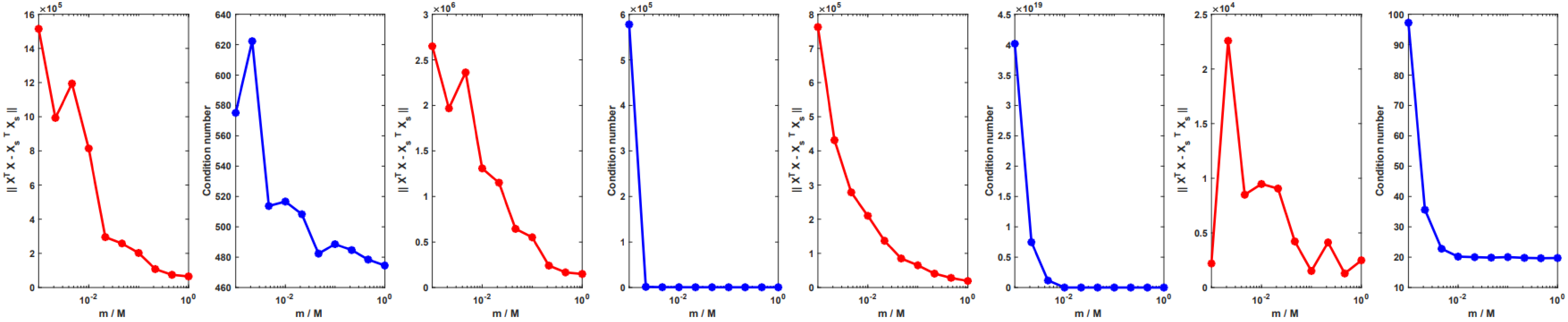
n	Sparsity	HDSDP (start from $(-10^6, 0)$)	COPT	Mosek	SDPT3
500	0.05	7.1	6.8	9.1	18.0
1000	0.09	44.5	53.9	54.2	327.0
2000	0.002	34.3	307.1	374.7	572.3
5000	0.0002	64.3	>1200	>1200	>1200

Computational Results: Build Preconditioner from Samples

- **Many matrices result from statistical datasets**
- $X^\top X$ **estimates the covariance matrix**
- **It suffices to use a few (row) samples to approximate**

How few?

As few as $O(\log(\#sample))$!



Experiment over regression datasets shows that

- **It generally takes 1% to 5% of the samples to approximate well**
- **Scales well with dimension and saves much time for matrix-matrix multiplication**

Computational Results: Optimal Diagonal Pre-Conditioner

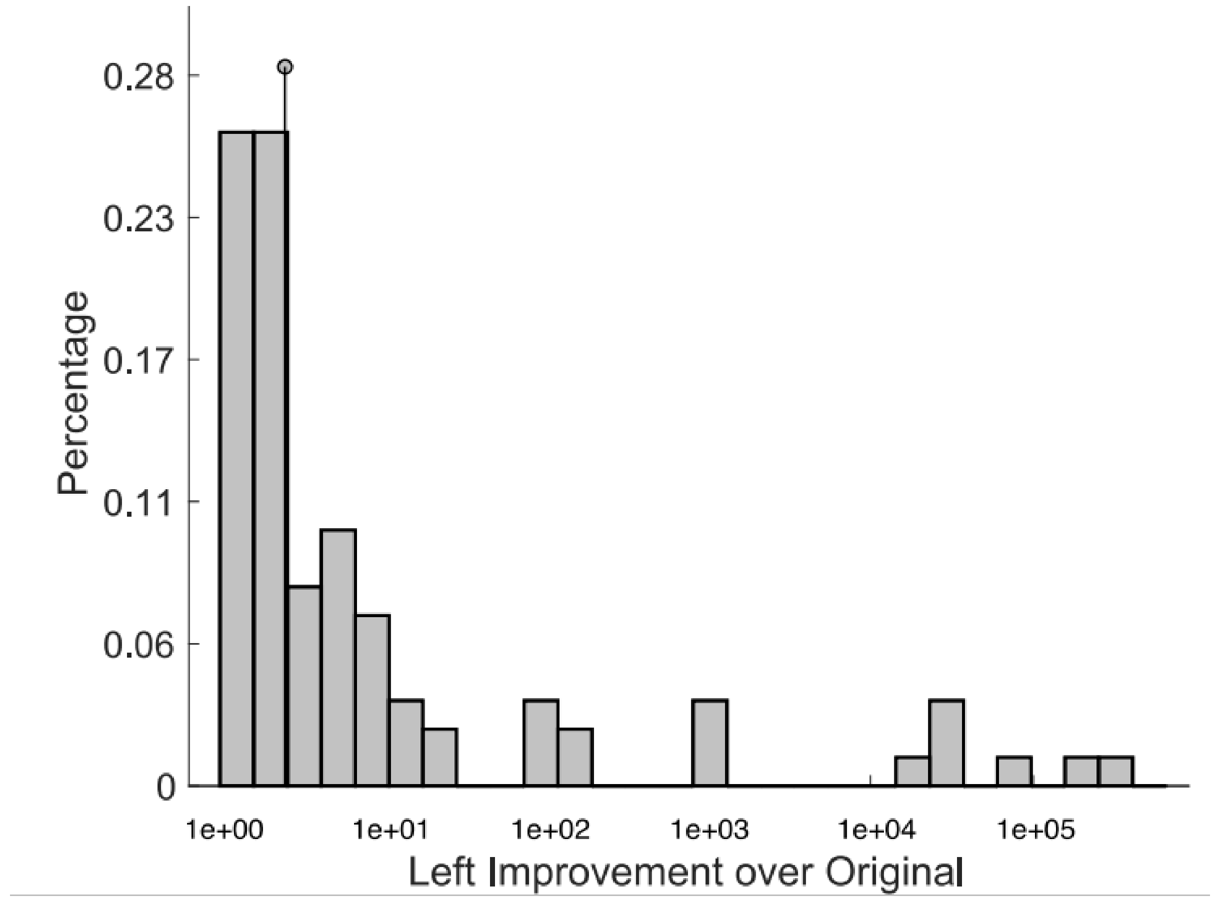
- **Test over 491 Suite Sparse Matrices of fewer than 1000 columns**

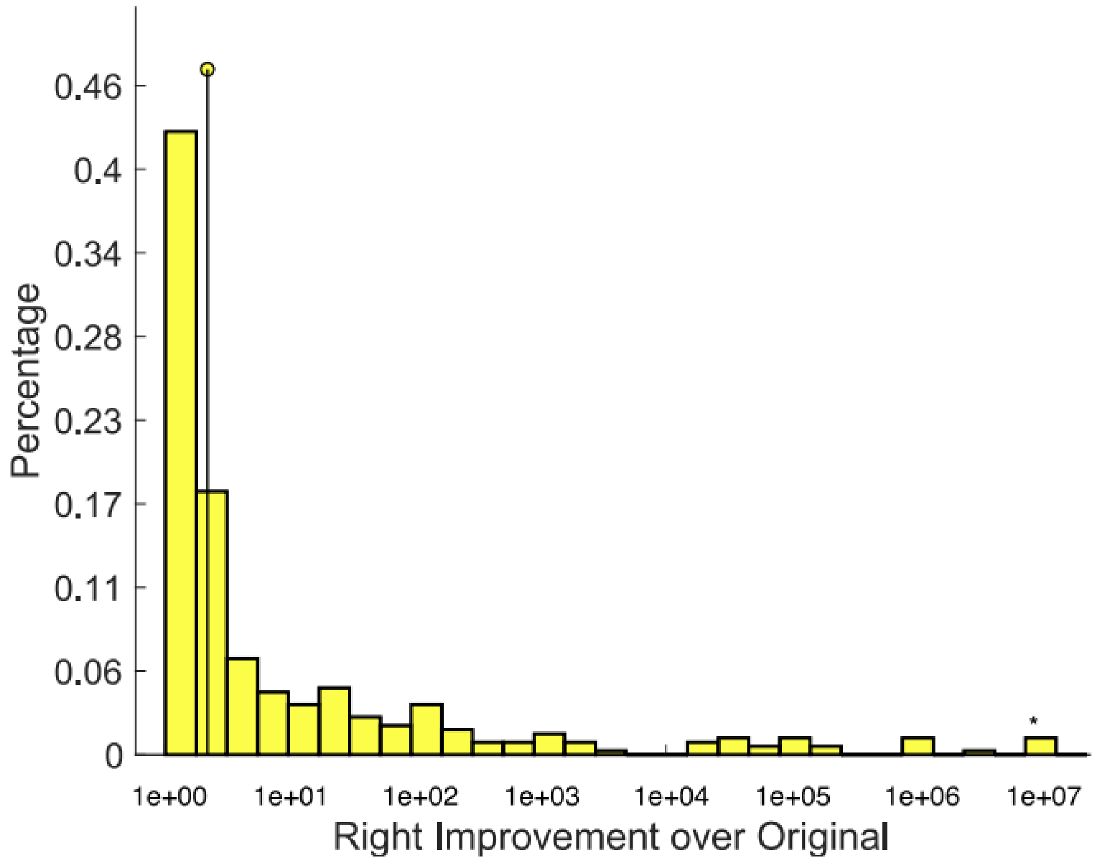
Reduction	Number
≥80%	121
≥50%	190
≥20%	261

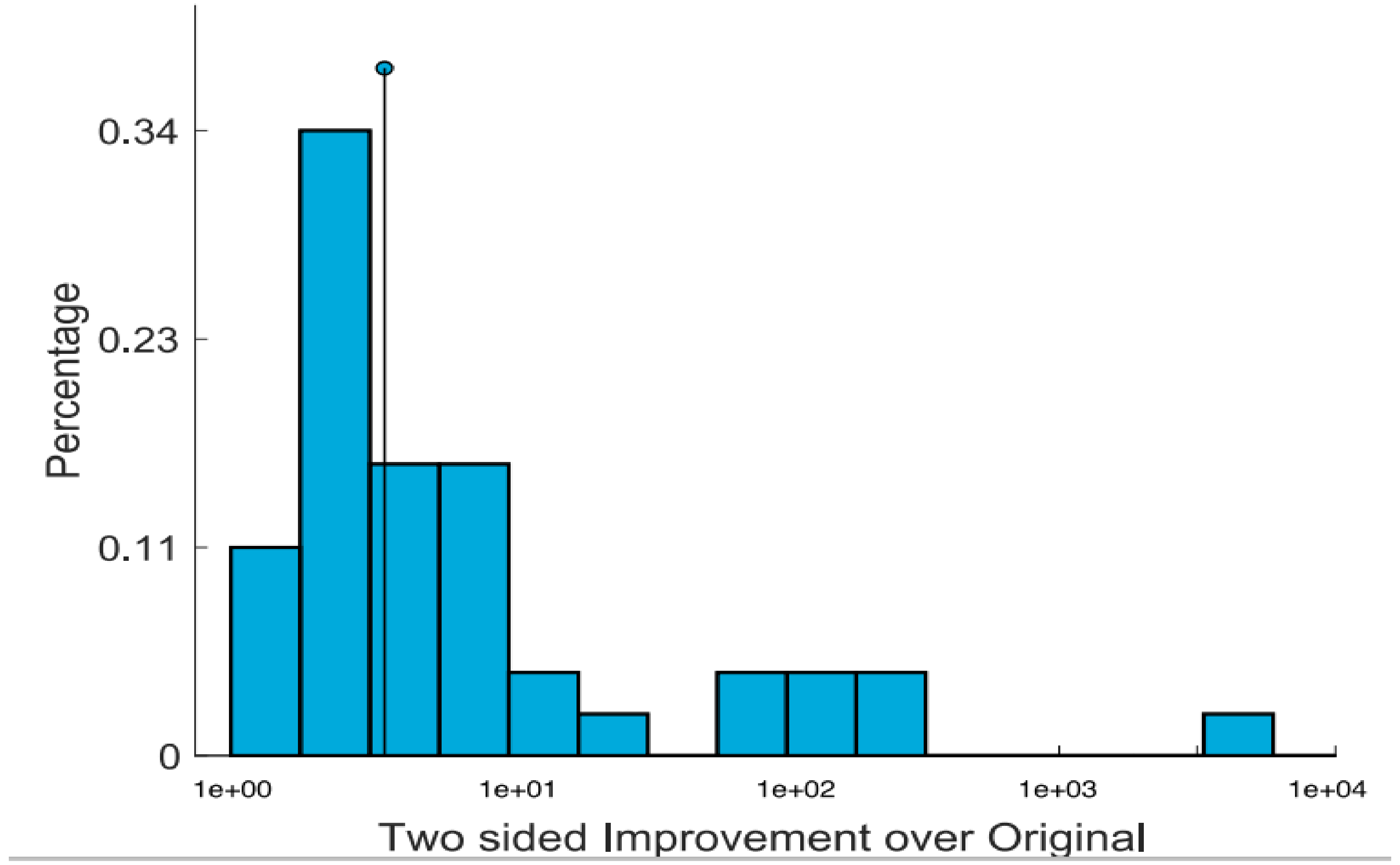
Average reduction	49.7%
Better than diagonal	36.0%
Average time	1.29

- **LIBSVM datasets**

Mat	Size	Cbef	Caft	Reduce
YearPredictionMSD	90	5233000.00	470.20	0.999910
YearPredictionMSD.t	90	5521000.00	359900.00	0.934816
abalone_scale.txt	8	2419.00	2038.00	0.157291
bodyfat_scale.txt	14	1281.00	669.10	0.477475
cadata.txt	8	8982000.00	7632.00	0.999150
cpusmall_scale.txt	12	20000.00	6325.00	0.683813
eunite2001.t	16	52450000.00	8530.00	0.999837
eunite2001.txt	16	67300000.00	3591.00	0.999947
housing_scale.txt	13	153.90	83.22	0.459371
mg_scale.txt	6	10.67	10.03	0.059988
mpg_scale.txt	7	142.50	107.20	0.247842
pyrim_scale.txt	27	49100000.00	3307.00	0.999933
space_ga_scale.txt	6	1061.00	729.60	0.312041
triazines_scale.txt	60	24580000.00	15460000.00	0.371034







Distribution of condition number reduction (Factor of improvement)

Summary

PCG is one of the most popular methods to accelerate SOM

- **Optimal Diagonal Precondition, either one-side or two-sides, is "polynomially" computable**
- **It would be efficient for solving systems with the stable left-hand matrix and variable right-hand vectors, such as in Regression and ADMM**
- **It establishes the bench-mark for evaluating other pre-conditioners based on heuristics and/or machine-learning**
- **HDSDP a general purpose SDP solver which using dual-scaling and simplified HSD**
- **It is developed with effective heuristics and computational tricks from DSDP**

Today's Talk

- Optimal Diagonal Precondition using SDP
- An Accelerated Second-Order Method using Homogenized Descent Direction
- A Dimension Reduced Trust-Region Method for Unconstrained Optimization
- Potential Reduction Algorithm for Linear Programming

Early Complexity Analyses for Nonconvex Optimization

$$\min f(x), x \in X \; in \; \mathbb{R}^n,$$

- where f is nonconvex and twice-differentiable,

  $g_k = \nabla f(x_k), H_k = \nabla^2 f(x_k)$

- Goal: find x_k such that:

  $\| \nabla f(x_k) \| \leq \epsilon$ (primary, first-order condition)

  $\lambda_{min}(H_k) \geq -\sqrt{\epsilon}$ (in active subspace, secondary, second-order condition)

- For the ball-constrained nonconvex QP: $\min \; c^T x + 0.5 x^T Q x \; \; s.t. \; \| x \|_2 \leq 1$

  O($\log\log(\epsilon^{-1})$); see Y (1989,93), Vavasis&Zippel (1990)

- For nonconvex QP with polyhedral constraints: O(ϵ^{-1}); see Y (1998), Vavasis (2001)

Standard methods for general nonconvex optimization I

First-order Method (FOM): Gradient-Type Methods

- Assume f has L-Lipschitz cont. gradient
- Global convergence by, e.g., linear-search (LS)
- No guarantee for the second-order condition
- Worst-case complexity, $O(\epsilon^{-2})$; see the textbook by Nesterov (2004)

Each iteration requires $O(n^2)$ operations

Classical Methods for General Nonconvex Optimization II

Second-order Method (SOM): Hessian-Type Methods

- Assume f has M-Lipschitz cont. Hessian
- Trust-region (More 70, Sorenson 80) with a fixed-radius strategy, $O(\epsilon^{-3/2})$, see the lecture notes by Y since 2005
- Cubic regularization, $O(\epsilon^{-3/2})$,see Nesterov and Polyak (2006), Cartis, Gould, and Toint (2011)
- An adaptive trust-region framework, $O(\epsilon^{-3/2})$,Curtis, Robinson, and Samadi (2017)

Each iteration requires $O(n^3)$ operations: **How to reduce it?**

An Integrated Descent Direction Using the Homogenized Quadratic Model I (Zhang at al. SHUFE)

- **Recall the fixed-radius trust-region method minimizes the Taylor quadratic model**

$$\min_{d \in \mathbb{R}^n} m_k(d) := g_k^T d + \frac{1}{2} d^T H_k d$$
$$\text{s.t.} \|d\| \leq \Delta_k .$$

- where $\Delta_k = \epsilon^{1/2}/M$ is the trust-ball radius.
- **$-g_k$ is the first-order steepest descent direction but ignores Hessian;**
- **the most-left eigenvector of H_k-would be a descent direction for the second order term but such direction may not exist if it becomes nearly convex…**
- **Could we construct a direction integrating both?**

Answer: Use the homogenized quadratic model of SDP relaxation

An Integrated Descent Direction Using the Homogenized Quadratic Model II

- **Using the homogenization trick by lifting with extra scalar t:**

$$\psi_k\left(\xi_0, t; \delta\right) := \frac{1}{2}\begin{bmatrix}\xi_0 \\ t\end{bmatrix}^T \begin{bmatrix}H_k & g_k \\ g_k^T & -\delta\end{bmatrix}\begin{bmatrix}\xi_0 \\ t\end{bmatrix} = \frac{t^2}{2}\begin{bmatrix}\xi_0/t \\ 1\end{bmatrix}^T \begin{bmatrix}H_k & g_k \\ g_k^T & -\delta\end{bmatrix}\begin{bmatrix}\xi_0/t \\ 1\end{bmatrix}$$

- **The homogeneous model is equivalent to m_k up to scaling:**

$$\psi_k(\xi_0, t; \delta) = t^2 \cdot (m_k(\xi_0/t) - \delta)$$

- **Find the direction $\xi = \xi_0/t$ (if *t = 0* then set *t=1*) by the leftmost eigenvector:**

$$\min_{\|[\xi_0;t]\| \leq 1} \psi_k\left(\xi_0, t; \delta\right)$$

- Fix δ **and compute the direction at the cost of $O(\epsilon^{-1/4}\log(1/\epsilon))$ via the randomized Lanczos method (Curvature computation of H_k was used in few hybrid $O(\epsilon^{-7/4}\log(1/\epsilon))$ methods of first and second orders; see Agarwal**

Global Convergence Rate: Outline of Analysis

- **A concise analysis using fixed radius Δ**

Let $x_{k+1} = x_k + \eta\xi$, $R(H_k, \xi) = \xi^T H_k \xi / \|\xi\|^2$, $\xi = \xi_0 / t$

- **(sufficient decrease in large step) If** $\|\xi\| \geq \Delta$, **we choose** $\eta = \Delta / \|\xi\|$
  - $f(x_{k+1}) - f(x_k) \leq -\frac{\delta \Delta^2}{2} + \frac{M}{6}\Delta^3$, **regardless of** $t = 0$ **or not**
  - δ **must be some** greater than $O(\sqrt{\epsilon})$ **to have** $O\left(\epsilon^{\frac{3}{2}}\right)$ decrease
- **(small step means convergence) Otherwise** $\|\xi\| < \Delta$, **then we choose step-size** $\eta = 1$ **and**
  - $\|g_{k+1}\| \leq 4(L + \delta)^2 \Delta^3 + \frac{M}{2}\Delta^2 + (2L\delta + 2\delta^2)\,\Delta$
  - δ **must be some** less than $O(\sqrt{\epsilon})$ **and converge**
- δ **should also be set in** $O(\sqrt{\epsilon})$ **!**

- **This results a single-looped (easy-to-implement)** $O(\epsilon^{-7/4}\log(1/\epsilon))$ **method**

Theoretical Guarantees of HSODM

- **Consider use the second-order homogenized direction, and the length of each step $\|\eta\xi\|$ is fixed: $\|\eta\xi\| \leq \Delta_k = \frac{2\sqrt{\epsilon}}{M}$ where $f(x)$ has L-Lipschitz gradient and M-Lipschitz Hessian.**
- **Theorem 1 (Global convergence rate) : if $f(x)$ satisfies the Lipchitz Assumption and $\delta = \sqrt{\varepsilon}$, the iterate moves along homogeneous vector ξ: $x_{k+1} = x_k + \eta_k \xi$, then, if we choose $\eta_k = \Delta_k / \|\xi\|$, and terminate at $\|\xi\| < \Delta_k$, then algorithm has $O(\epsilon^{-3/2})$ iteration complexity. Furthermore, x_{k+1} satisfies approximate first-order and second-order conditions.**
- **Theorem 2 (Local convergence rate): If the iterate x_k of HSODM converges to a strict local optimum x^* such that $H(x^*) \succ 0$,and then $\eta_k = 1$ if k is sufficiently large. If we do not terminate HSODM and set $\delta = 0$, then HSODM has a local superlinear (quadratic) speed of convergence, namely: $\| x_{k+1} - x^* \| = O(\| x_k$**

HSODM for Convex Optimization

- $f(x)$ is a *convex* function with M-Lipschitz Hessian.
- At every iteration, choos $\delta_k = O\left(\|g_k\|^{1/2}\right)$ and solve

$$\min_{\|[\xi_0;t]\|\leq 1} \begin{bmatrix} \xi_0 \\ t \end{bmatrix}^T \begin{bmatrix} H_k & g_k \\ g_k^T & -\delta_k \end{bmatrix} \begin{bmatrix} \xi_0 \\ t \end{bmatrix}$$

- Update $x_{k+1} = x_k + \xi$, $\xi = \xi_0/t$ ($t = 0$ won't happen when $f(x)$ is convex)
- Theorem 3 (Global convergence rate) : suppose the sublevel set $\{x: f(x) \leq f(x_0)\}$ is bounded, then the sequence $\{x_k\}$ satisfies

$$f(x_k) - f(x^*) \leq O\left(k^{-2}\right)$$

- Ongoing: improved bounds of accelerated HSODM
- Practical remarks: homogenized direction can be used with any Line-Search (e.g., Hager-Zhang)

Today's Talk

- Optimal Diagonal Precondition using SDP
- An Accelerated Second-Order Method using homogenized Descent Direction
- A Dimension Reduced Trust-Region Method for Unconstrained Optimization
- Potential Reduction Algorithm for Linear Programming

DRSOM I

- Motivation from Multi-Directional FOM and Subspace Method, DRSOM in general uses **reduced** m-independent directions $d(\alpha) := D_k\alpha$, $D_k \in R^{nm}$, $\alpha \in R^m$
- Plug the expression into the full-dimension Trust-Region quadratic minimization model, we minimize a m-dimension trust-region subproblem to decide “m stepsizes”:

$$\min\ m_k^{\alpha}(\alpha) := (c_k)^T\alpha + \frac{1}{2}\alpha^T Q_k \alpha$$

$$||\alpha||_{G_k} \leq \Delta_k$$

$$G_k = D_k^T D_k, Q_k = D_k^T H_k D_k, c_k = (g_k)^T D_k$$

How to choose D_k? Provable complexity result?

DRSOM II

- In following, as an example, DRSOM adopts one or two FOM directions

$$d = -\alpha^1 \nabla f(x_k) + \alpha^2 d_k := d(\alpha)$$

where $g_k = \nabla f(x_k), H_k = \nabla^2 f\left(x^k\right), d_k = x_k - x_{k-1}$

- Then we minimize a 1 or 2-D trust-region problem to decide “two step-sizes”:

$$\min\ m_k^{\alpha}(\alpha) := f(x_k) + (c_k)^T \alpha + \frac{1}{2}\alpha^T Q_k \alpha$$

$$||\alpha||_{G_k} \leq \Delta_k$$

$$G_k = \begin{bmatrix} g_k^T g_k & -g_k^T d_k \\ -g_k^T d_k & d_k^T d_k \end{bmatrix}, Q_k = \begin{bmatrix} g_k^T H_k g_k & -g_k^T H_k d_k \\ -g_k^T H_k d_k & d_k^T H_k d_k \end{bmatrix}, c_k = \begin{bmatrix} -||g_k||^2 \\ g_k^T d_k \end{bmatrix}$$

DRSOM III

DRSOM can be seen as:

- "Adaptive" **Accelerated Gradient Method** (Polyak's momentum 60)
- A second-order method minimizing quadratic model in the reduced 2-D subspace

  $$m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d, d \in \text{span}\{-g_k, d_k\}$$

  compare to, e.g., Dogleg method, 2-D Newton **Trust-Region Method**

  $d \in \text{span}\{g_k, [H(x_k)]^{-1} g_k\}$ (e.g., Powell 70, Byrd 88)
- A conjugate direction method for convex optimization exploring the **Krylov Subspace** (e.g., Barzilai&Borwein 88, Yuan&Stoer 95, Yuan 2014, Liu et al. 2021)
- For convex quadratic programming with no radius limit, it reduces to CG and BFGS terminating in n steps

Computing Hessian-Vector Product in DRSOM is the Key

In the DRSOM with two directions:

$$Q_k = \begin{bmatrix} g_k^T H_k g_k & -g_k^T H_k d_k \\ -g_k^T H_k d_k & d_k^T H_k d_k \end{bmatrix}, c_k = \begin{bmatrix} -||g_k||^2 \\ g_k^T d_k \end{bmatrix}$$

How to cheaply obtain Q? Compute $H_k g_k, H_k d_k$ first.

- Finite difference:

$$H_k \cdot v \approx \frac{1}{\epsilon}[g(x_k + \epsilon \cdot v) - g_k],$$

- Analytic approach to fit modern automatic differentiation,

$$H_k g_k = \nabla(\frac{1}{2} g_k^T g_k), H_k d_k = \nabla(d_k^T g_k),$$

- Use Hessian if readily available !
- **Three(-or more)-Point Interpolation: it is almost as fast as Polyak and CG!**

DRSOM: Key Assumptions and Theoretical Results (Zhang at al. SHUFE)

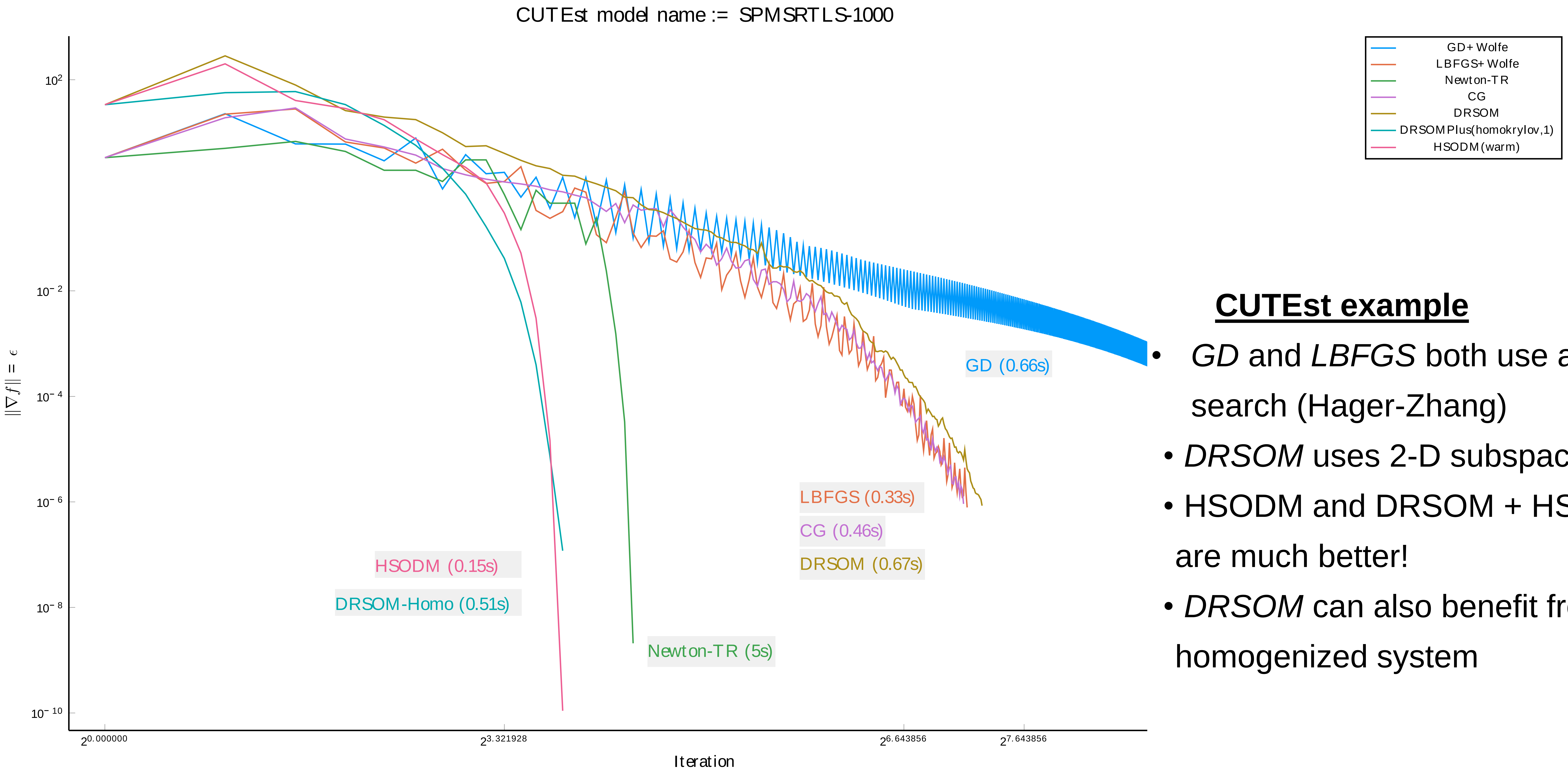
Assumption. (a) f has Lipschitz continuous Hessian. (b) **If the Lagrangian multiplier $\lambda_k < \sqrt{\epsilon}$, assume $\| (H_k - \widetilde{H}_k) d_{k+1} \| \le C \| d_{k+1} \|^2$ (Cartis et al.),** where $\widetilde{H}_k$ is the projected Hessian in the subspace (commonly adopted for approximate Hessian)

Theorem 1. If we apply DRSOM to QP, **then** the algorithms terminates in at most n steps to find a first-order stationary point

Theorem 2. (Global convergence rate) For *f* with second-order Lipschitz condition, let $\Delta_k = 2\epsilon^{1/2}/M$, then DRSOM terminates in $O(\epsilon^{-3/2})$ iterations. Furthermore, the iterate x_k satisfies the first-order condition, and the Hessian is positive semi-definite in the subspace spanned by the gradient and momentum.

Theorem 3. (Local convergence rate) If the iterate x_k converges to a strict local optimum x^* such that $H(x^*) \succ 0$, and if **Assumption (b)** is satisfied as soon as $\lambda_k \le C_\lambda \| d_{k+1} \|$, then DRSOM has a local superlinear (quadratic) speed of convergence, namely: $\| x_{k+1} - x^* \| = O(\| x_k - x^* \|^2)$

Preliminary Results: HSODM and DRSOM + HSODM



<u>CUTEst example</u>

- *GD* and *LBFGS* both use a Line-search (Hager-Zhang)
- *DRSOM* uses 2-D subspace
- HSODM and DRSOM + HSODM are much better!
- *DRSOM* can also benefit from the homogenized system

Sensor Network Location (SNL)

- Consider Sensor Network Location (SNL)

$$N_x = \{(i,j) : \|x_i - x_j\| = d_{ij} \leq r_d\}, N_a = \{(i,k) : \|x_i - a_k\| = d_{ik} \leq r_d\}$$

where r_d is a fixed parameter known as the radio range. The SNL problem considers the following QCQP feasibility problem,

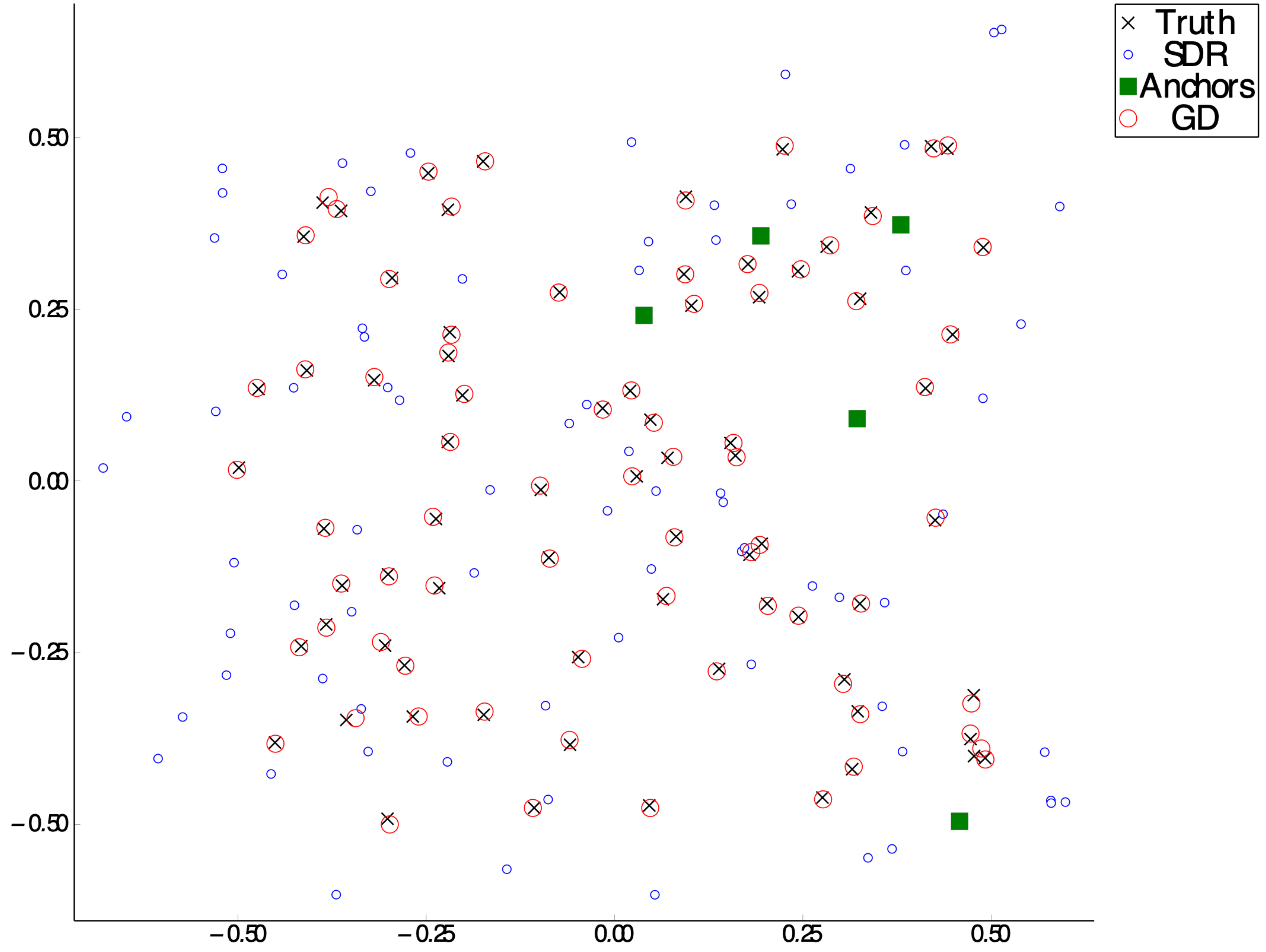
$$\|x_i - x_j\|^2 = d_{ij}^2, \forall (i,j) \in N_x$$
$$\|x_i - a_k\|^2 = \bar{d}_{ik}^2, \forall (i,k) \in N_a$$

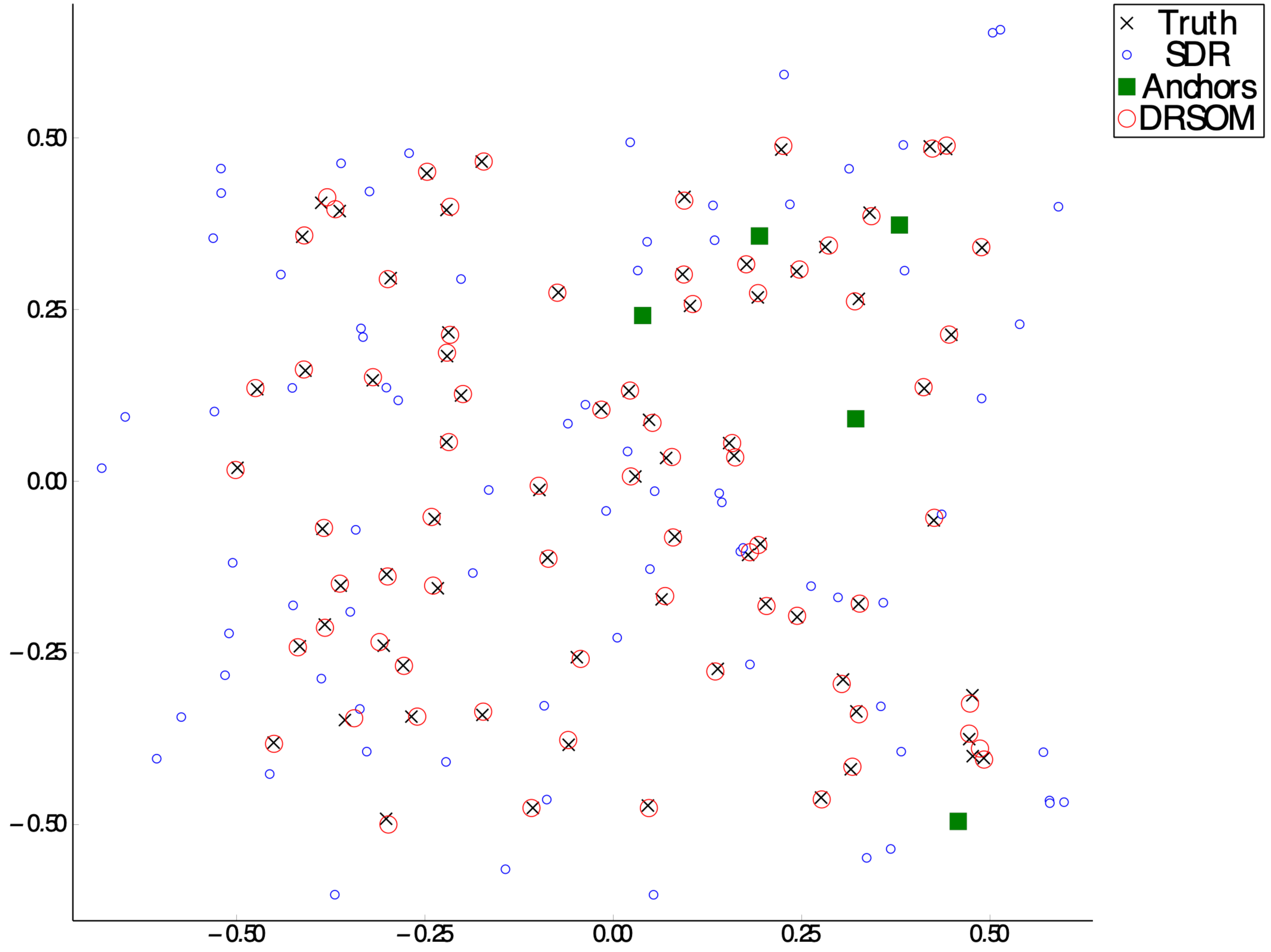
- We can solve SNL by the nonconvex nonlinear least square (NLS) problem

$$\min_X \sum_{(i<j,j)\in N_x} (\|x_i - x_j\|^2 - d_{ij}^2)^2 + \sum_{(k,j)\in N_a} (\|a_k - x_j\|^2 - \bar{d}_{kj}^2)^2.$$

Sensor Network Location (SNL)

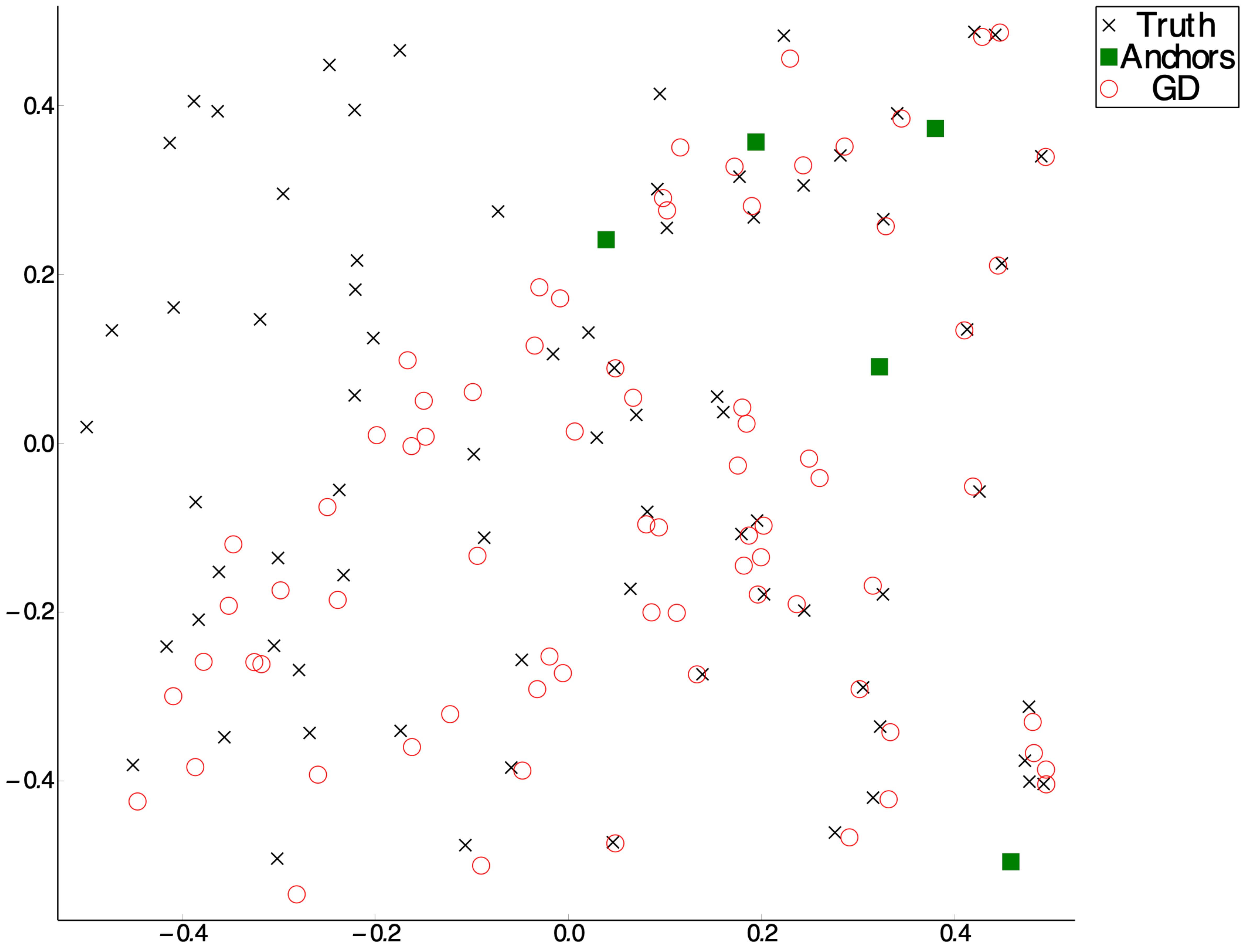
- Graphical results using SDP relaxation to initialize the NLS
- n = 80, m = 5 (anchors), radio range = 0.5, degree = 25, noise factor = 0.05
- Both Gradient Descent and DRSOM can find good solutions !

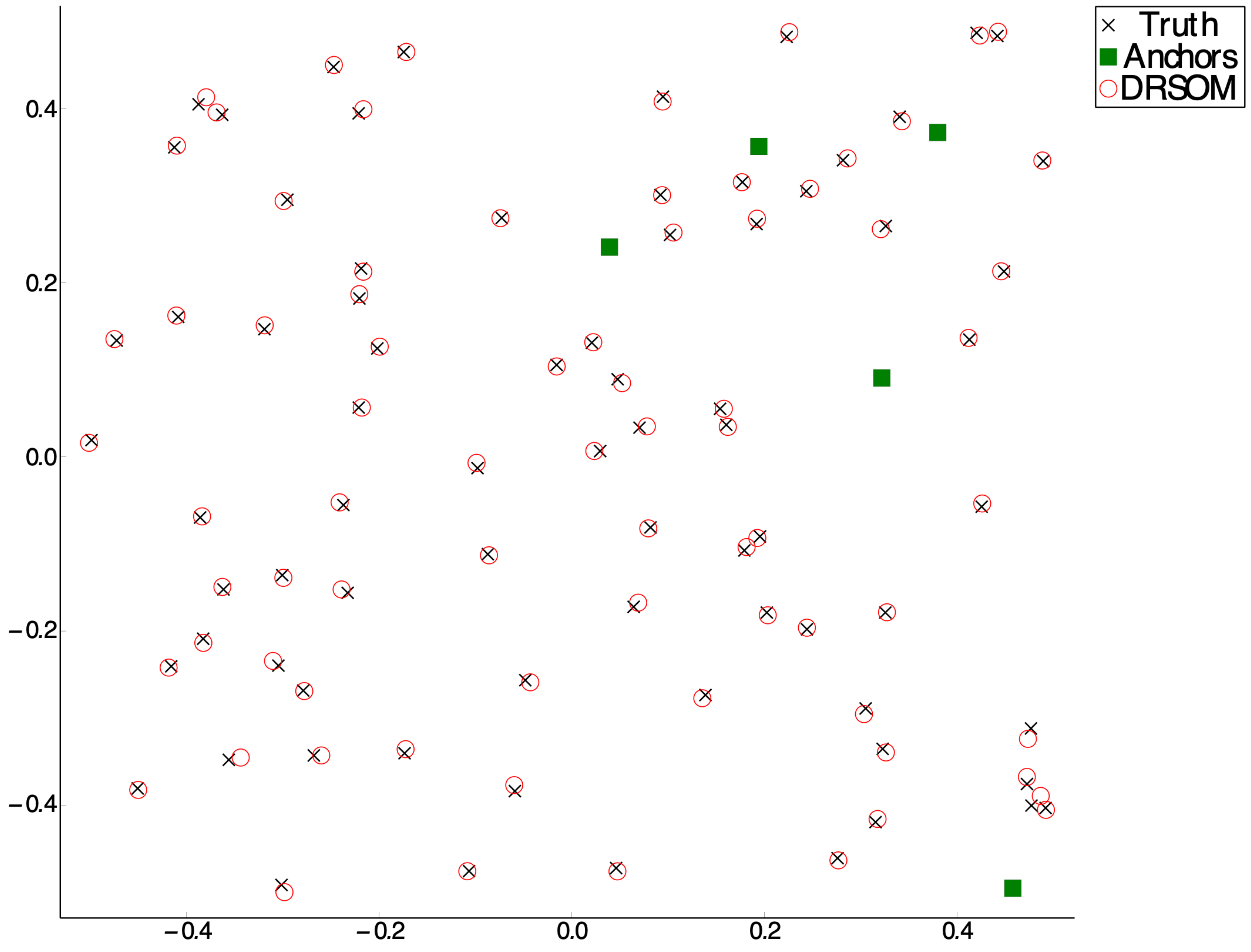




Sensor Network Location (SNL)

- Graphical results without SDP relaxation
- DRSOM can still converge to optimal solutions





Sensor Network Location, Large-scale instances

- Test large SNL instances (terminate at 3,000s and $|\boldsymbol{g_k}| \leq \mathbf{1}\boldsymbol{e}^{-\mathbf{5}}$)
- Compare GD, CG, and DRSOM. (GD and CG use Hager-Zhang Linesearch)

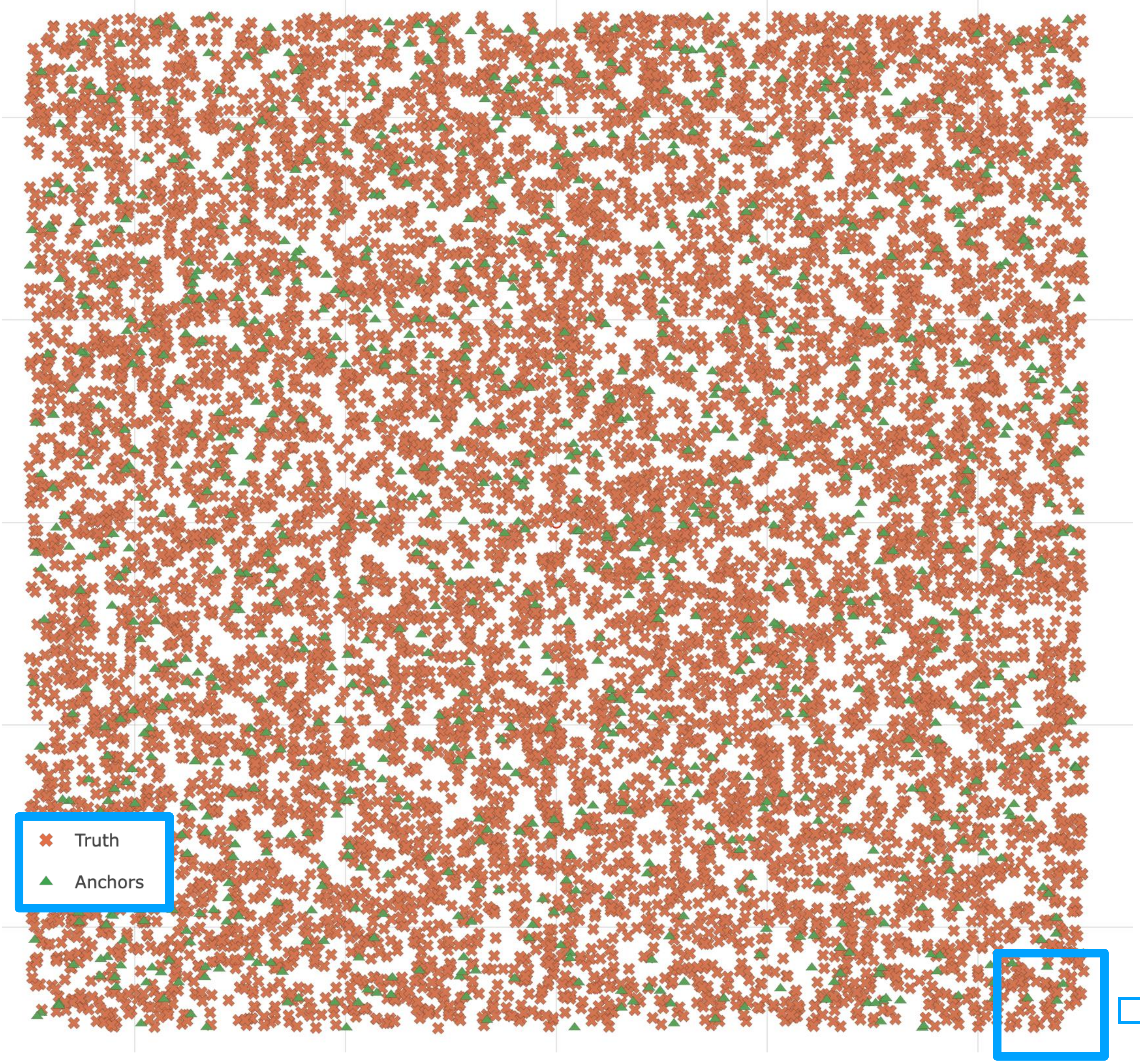
n	m	$\|E\|$	t CG	t DRSOM	t GD
500	50	2.2e+04	1.7e+01	1.1e+01	2.3e+01
1000	80	4.6e+04	7.3e+01	3.9e+01	1.8e+02
2000	120	9.4e+04	2.5e+02	1.4e+02	1.1e+03
3000	150	1.4e+05	6.5e+02	1.4e+02	-
4000	400	1.8e+05	1.3e+03	5.0e+02	-
6000	600	2.7e+05	2.0e+03	1.1e+03	-
10000	1000	4.5e+05	-	2.2e+03	-

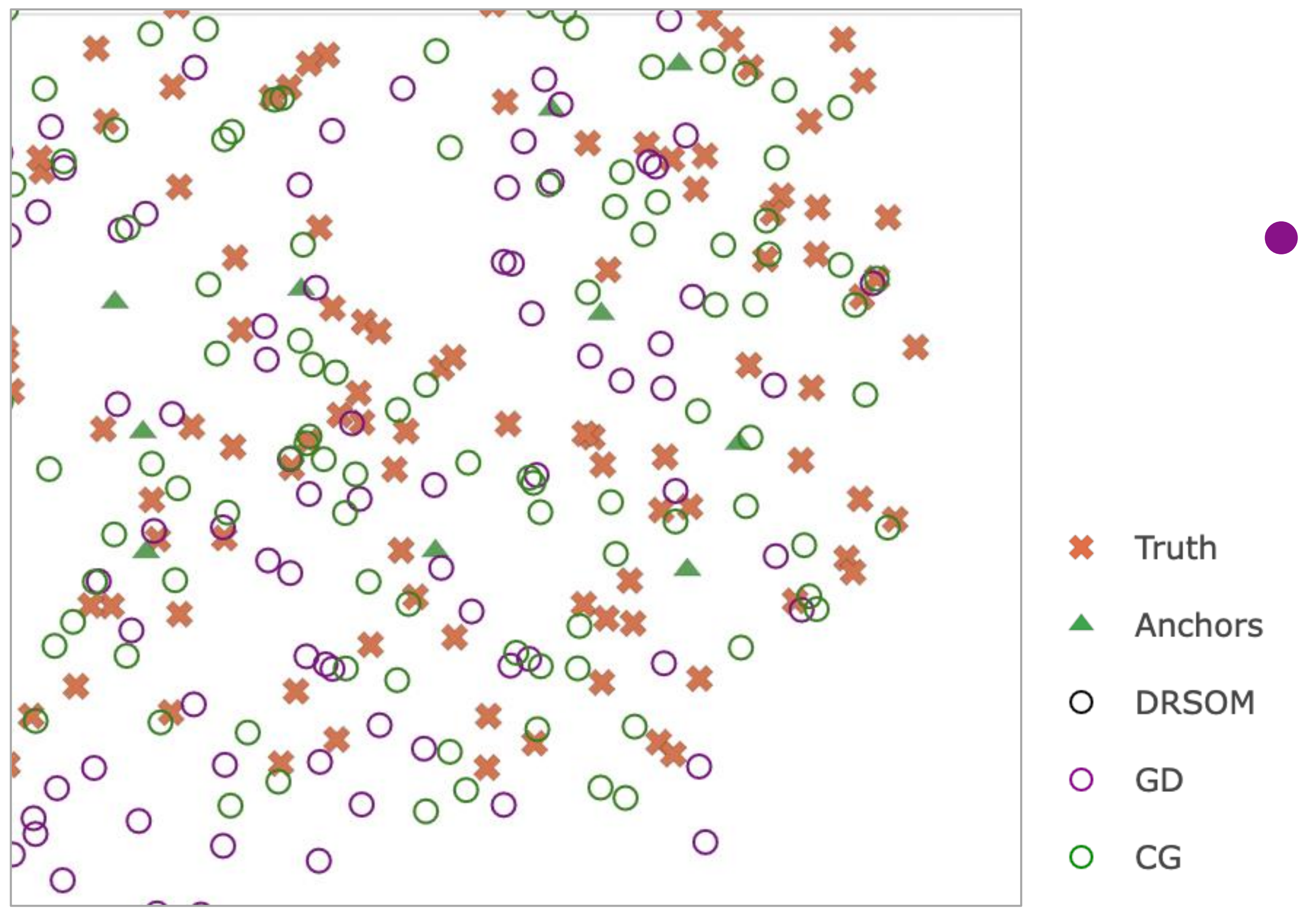
Table 2: Running time of CG, DRSOM, and GD on SNL instances of different problem size, $|E|$ denotes the number of QCQP constraints. "-" means the algorithm exceeds 3,000s.

- DRSOM has the best running time (benefits of 2nd order info and interpolation!)

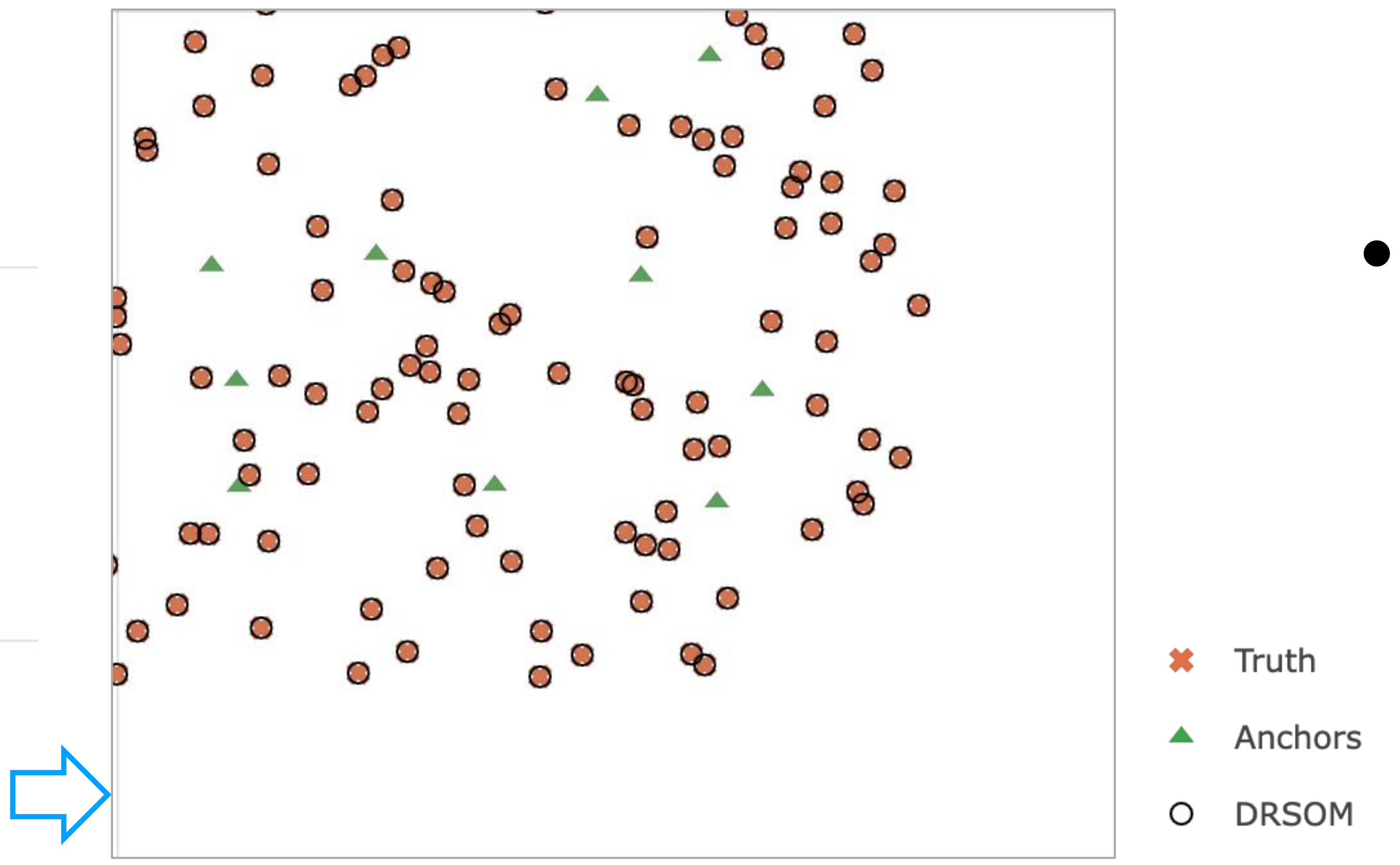
Sensor Network Location, Large-scale instances

- Graphical results with 10,000 nodes and 1000 anchors (no noise) **within 3,000 seconds**





- **GD with Line-search and Hager-Zhang CG both timeout**

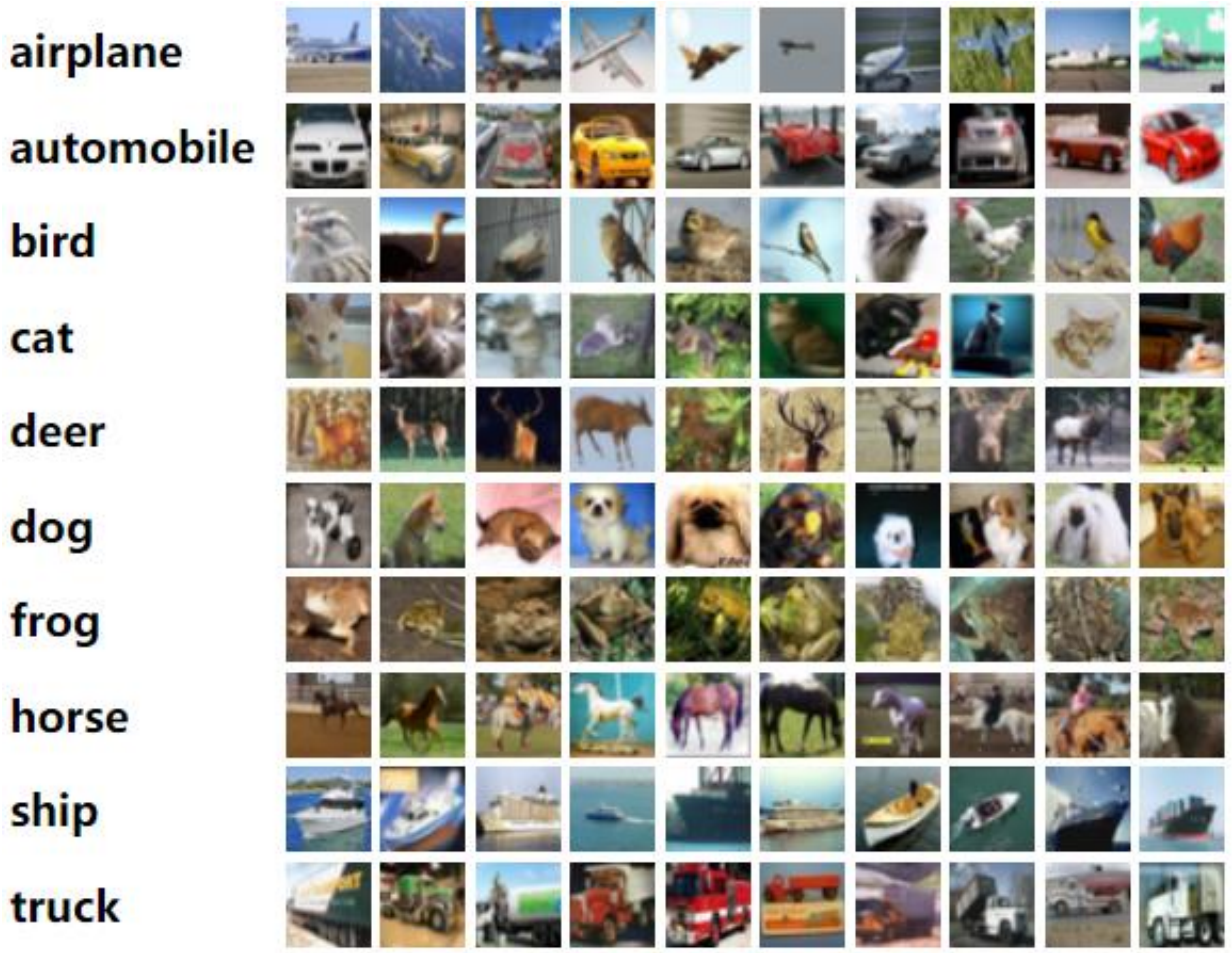


- **DRSOM can converge to $|g_k| \le 1e^{-5}$ in 2,200s**

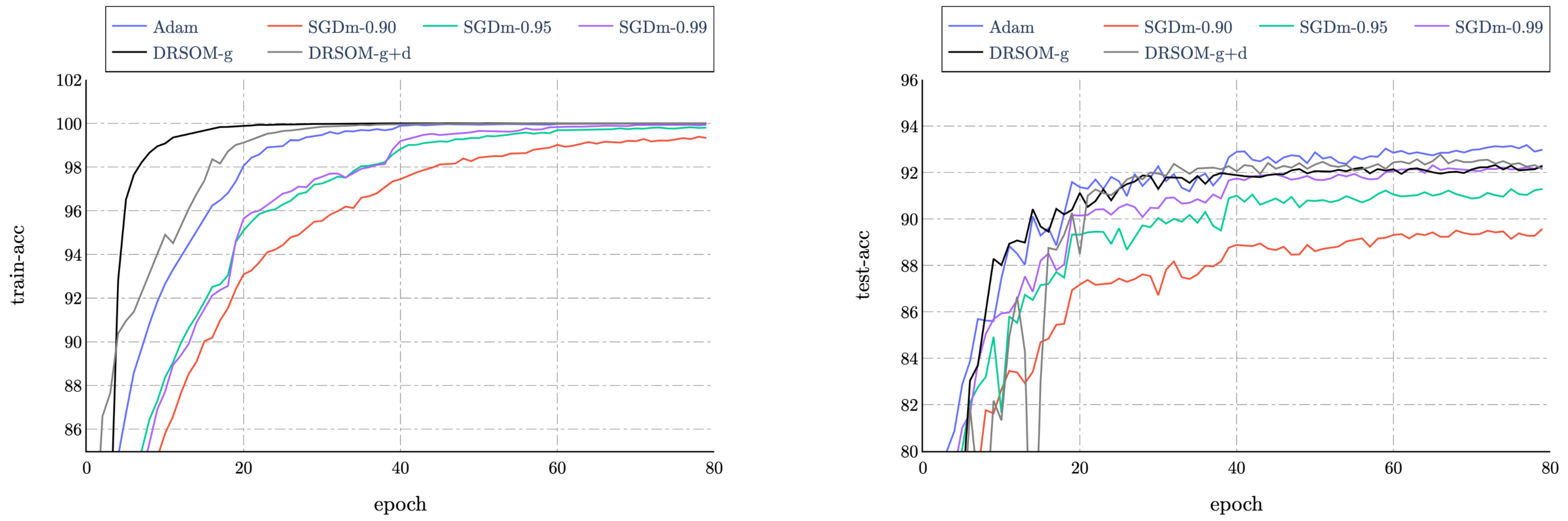
Neural Networks and Deep Learning

To use DRSOM in machine learning problems

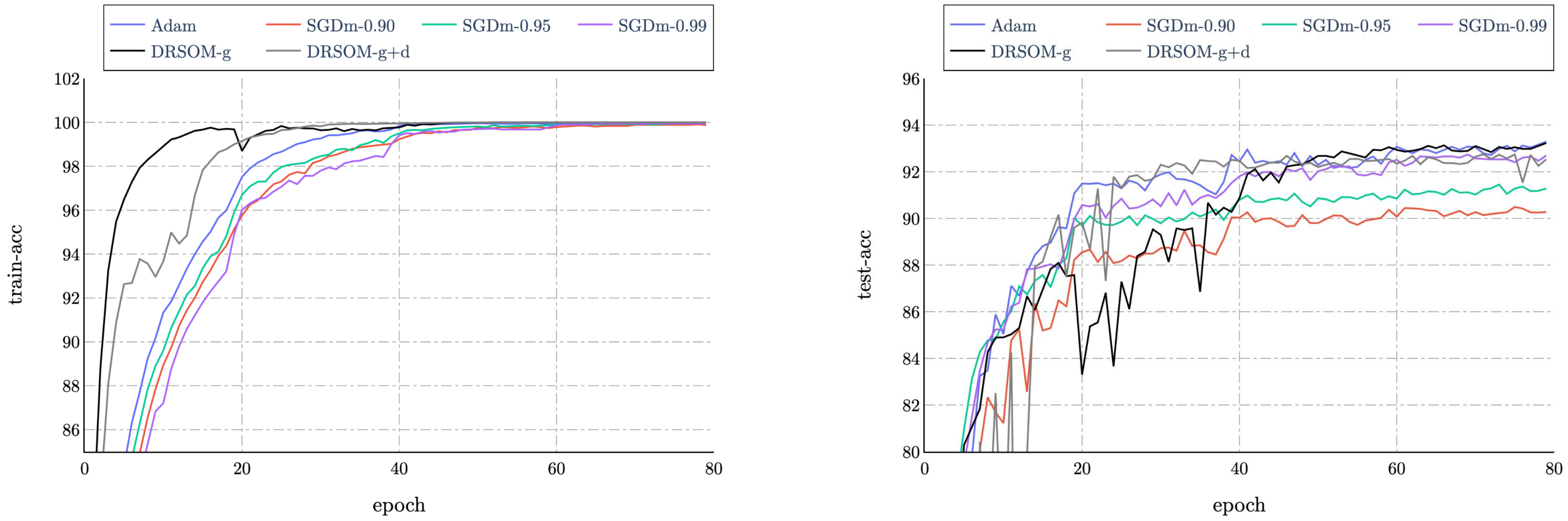
- We apply the mini-batch strategy to a vanilla DRSOM
- Use Automatic Differentiation to compute gradients
- Train ResNet18/Resnet34 Model with CIFAR 10
- Set Adam with initial learning rate 1e-3



Neural Networks and Deep Learning



Training and test results for ResNet18 with DRSOM and Adam



Training and test results for ResNet34 with DRSOM and Adam

Pros

- DRSOM has rapid convergence (30 epochs)
- DRSOM needs little tuning

Cons

- DRSOM may over-fit the models
- Running time can benefit from Interpolation
- Single direction DRSOM is also good

Good potential to be a standard optimizer for deep learning!

DRSOM for Riemannian Optimization (Tang et al. NUS)

$$\min_{x \in \mathcal{M}} \quad f(x) \qquad \text{(ROP)}$$

- $\mathcal{M}$ is a Riemannian manifold embeded in Euclidean space $\mathbb{R}^n$.
- $f : \mathbb{R}^n \to \mathbb{R}$ is a second-order continuously differentiable function that is lower bounded in $\mathcal{M}$.

R-DRSOM: Choose an initial point $x_0 \in \mathcal{M}$, set $k = 0$, $p_{-1} = 0$;
for $k = 0, 1, \dots, T$ **do**
Step 1. Compute $g_k = \mathrm{grad} f(x_k)$, $d_k = \mathrm{T}_{x_k \leftarrow x_{k-1}}(p_{k-1})$, $H_k g_k = \mathrm{Hess} f(x_k)[g_k]$ and $H_k d_k = \mathrm{Hess} f(x_k)[d_k]$;
Step 2. Compute the vector $c_k = \begin{bmatrix} -\langle g_k, g_k \rangle_{x_k} \\ \langle g_k, d_k \rangle_{x_k} \end{bmatrix}$ and the following matrices

$$Q_k = \begin{bmatrix} \langle g_k, H_k g_k \rangle_{x_k} & \langle -d_k, H_k g_k \rangle_k \\ \langle -d_k, H_k g_k \rangle_{x_k} & \langle d_k, H_k d_k \rangle_{x_k} \end{bmatrix}, \ , \ G_k := \begin{bmatrix} \langle g_k, g_k \rangle_{x_k} & -\langle d_k, g_k \rangle_{x_k} \\ -\langle d_k, g_k \rangle_{x_k} & \langle d_k, d_k \rangle_{x_k} \end{bmatrix}.$$

Step 3. Solve the following 2 by 2 trust region subproblem with radius $\triangle_k > 0$

$$\alpha_k := \arg \min_{\|\alpha_k\|_{G_k} \le \triangle_k} f(x_k) + c_k^\top \alpha + \frac{1}{2} \alpha^\top Q_k \alpha;$$

Step 4. $x_{k+1} := \mathcal{R}_{x_k}\left(x_k - \alpha_k^1 g_k + \alpha_k^2 d_k\right)$;
end
Return x_k.

1D-Kohn-Sham Equation

$$\min\left\{\frac{1}{2}\mathrm{tr}(R^\top LR)+\frac{\alpha}{4}\mathrm{diag}(RR^\top)^\top L^{-1}\mathrm{diag}(RR^\top):\ R^\top R=I_p,\ R\in\mathbb{R}^{n\times r}\right\}, \quad (3)$$

where L is a tri-diagonal matrix with 2 on its diagonal and -1 on its subdiagonal and $\alpha > 0$ is a parameter. We terminate algorithms when $\|\mathrm{grad} f(R)\| < 10^{-4}$.

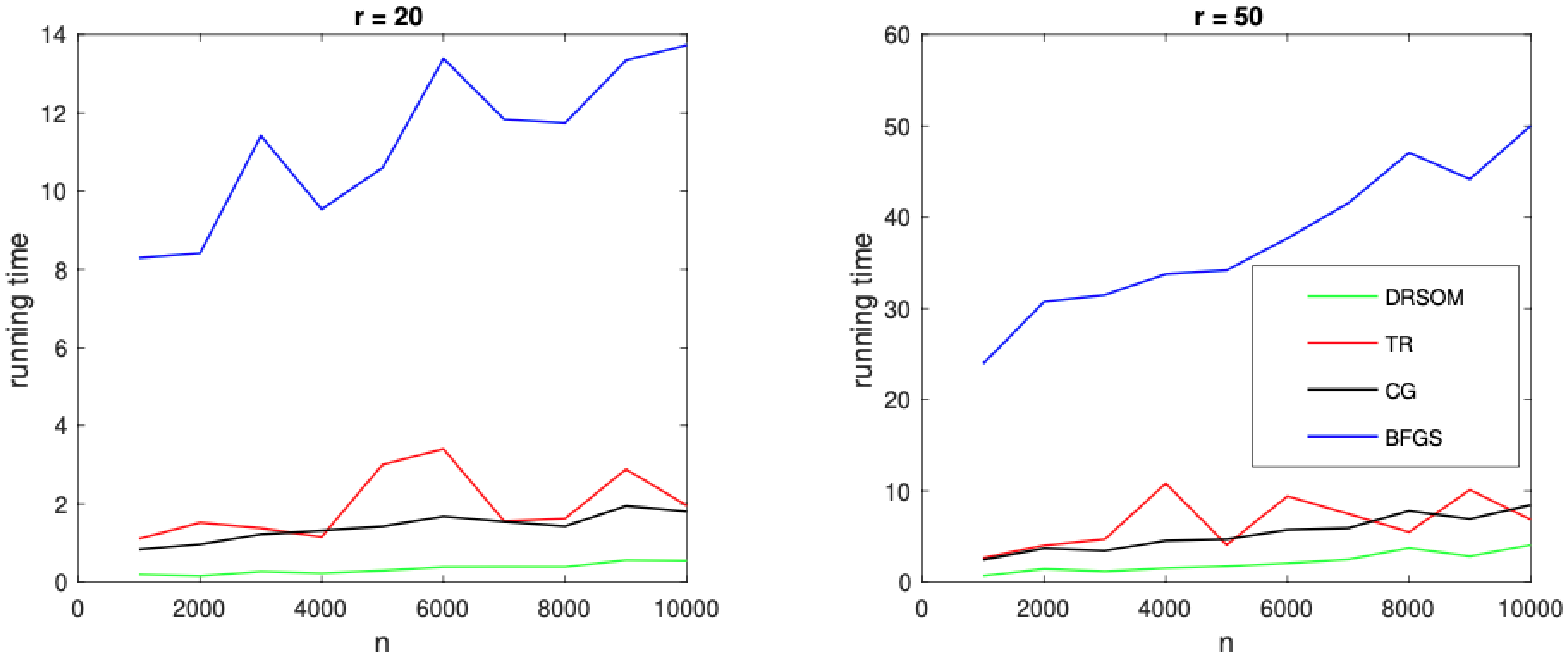


Figure 1: Results for Discretized 1D Kohn-Sham Equation. $\alpha = 1$.

Ongoing Research and Future Directions on HSODM/DRSOM

- **Rigorous DRSOM analyses, that is, removing Assumption (b)?**
- **Low-rank approximation of the homogenized matrix** $\begin{bmatrix} H_k & g_k \\ {g_k}^T & * \end{bmatrix}$**, and “Hot-Start” eigenvector computing by Power Methods (linear convergence of Liu et al. 2017)?**
- **Indefinite and Randomized Hessian rank-one updating via BFGS/SR1**
- **Dimension Reduced Non-Smooth/Semi-Smooth Newton**

Today’s Talk

- **Optimal Diagonal Precondition using SDP**
- **An Accelerated Second-Order Method using homogenized Descent Direction**
- **A Dimension Reduced Trust-Region Method for Unconstrained Optimization**
- **Potential Reduction Algorithm for Linear Programming**

DRSOM for LP Potential Reduction

We consider a simplex-constrained QP model

$$\begin{aligned} \min_x \quad & \frac{1}{2}\|Ax\|^2 =: f(x) \\ \textit{subject to} \quad & e^\top x = 1 \\ & x \geq 0 \end{aligned}$$

⇦

$$\begin{aligned} Ax - b\tau &= 0 \\ -A^\top y - s + c\tau &= 0 \\ b^\top y - c^\top x - \kappa &= 0 \\ e_n^\top x + e_n^\top s + \kappa + \tau &= 1 \end{aligned}$$

⇦

We wish to solve a standard LP (and its dual)

$$\begin{aligned} \min_x \quad & c^\top x \\ \textit{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

$$\begin{aligned} \max_{y,s} \quad & b^\top y \\ \textit{subject to} \quad & A^\top y + s = c \\ & s \geq 0 \end{aligned}$$

The self-dual embedding builds a bridge

- **The homogeneous QP seems so restrictive!**
- **How to solve much more general LPs?**

Then we define the (nonconvex) potential function and apply DRSOM to it

$$\phi(x) := \rho \log(f(x)) - \sum_{i=1}^{n} \log x_i$$

$$\nabla \phi(x) = \frac{\rho \nabla f(x)}{f(x)} - X^{-1}e, \qquad \nabla^2 \phi(x) = -\frac{\rho \nabla f(x) \nabla f(x)^\top}{f(x)^2} + \rho \frac{A^\top A}{f(x)} + X^{-2}$$

Combined with scaled gradient(Hessian) projection, the method solves LPs.

First-order Potential Reduction Algorithm

The first order steepest descent potential reduction algorithm would update x by solving

$$\begin{array}{ll} \min_{d} & \nabla\phi(x)^T d \\ \text{subject to} & e^T d = 0, \|X^{-1} d\| \leq \beta \end{array}$$

- $\beta < 1$ **to guarantee the update** $x + d > 0$
- **It admits a close-form solution** $d^* = p(x)$ **which is the scaled gradient projection vector**
- **By choosing** $\beta = O(f(x))$**, it is guaranteed to generate a solution** $f(x) \leq \epsilon$ **in** $O\left(\epsilon^{-1}\log(\epsilon^{-1})\right)$ **iterates, see Ye (2015).**

Question: Can we achieve a faster convergence by including second order information?

DRSOM for LP Potential Reduction

Recall the DRSOM is to minimize a 2-D trust-region problem

$$\begin{aligned} \min_{d} \quad & \nabla\phi(x)^T d + \frac{1}{2} \mathrm{d}^{\mathrm{T}} \nabla^2 \phi(x) d \\ \text{subject to} \quad & e^T d = 0, \|X^{-1} d\| \le \beta \\ & d \in span\{ p(x), \mathrm{m}(x)\} \end{aligned}$$

- $p(x)$ **is the scaled gradient projection vector and** $m(x)$ **is the moment vector**
- **If the assumption** $\| (\boldsymbol{H}_k - \tilde{\boldsymbol{H}}_k)\boldsymbol{d}_{k+1} \| \le \boldsymbol{C} \| \boldsymbol{d}_{k+1} \|^2$ **(Cartis et al.) still holds,**

  then a faster convergence rate $O(\epsilon^{-3/4} \log(\epsilon^{-1}))$ **can be guaranteed**

Question: Can we remove this assumption?

DRSOM + Negative Curvature

Once DRSOM gets stuck at some local region:

- we can compute the smallest eigenvector (usually negative curvature) of the projected Hessian matrix $\mathrm{H(x)} = (I - \frac{ee^T}{n})\nabla^2\phi(x)(I - \frac{ee^T}{n})$, which could help to escape local
- **Theorem 3:** For any point x satisfying $\min\{x_i\} \geq c_0 f(x)^{1/2}$ for some $c_0 > 0$, the smallest eigenvalue of $H(x)$ satisfies $\lambda_{min}(\mathrm{H(x)}) \leq \frac{-c_0^2 H(A)^2\rho+1}{c_0^2 f(x)}$, where $\mathrm{H(A)}$ is the Hoffman constant for LP. Besides, let d be the smallest eigenvector of $\mathrm{Q(x)}$, then we have $\nabla\phi(x)^T d \leq -\frac{H(A)\rho}{f(x)}$.
- A simple corrector step can be applied to guarantee the condition $\min\{x_i\} \geq c_0 f(x)^{1/2}$ holds:

$$x_l^+ = 2x_l \;\; and \;\; x_m^+ = x_m - x_l, \qquad where \;\; l = argmin_{1\leq i\leq n}\, x_i \;\; and \;\; m = argma_{1\leq i\leq n}\, x_i$$

If $\min\{x_i\} \geq c_0 f(x)^{1/2}$, then $\phi(x^+) - \phi(x) \leq -0.15$.

DRSOM-Potential Reduction

Repeat until stopping rule holds

- (Corrector step if necessary) If $\min\{x_i\} \geq c_0 f(x)^{1/2}$, then apply the corrector step
- (DRSOM step) Choose $\beta = O(f(x)^{-3/4})$ and take the DRSOM step
- (Negative curvature step if the decrease is slow) If $\phi(x^+) - \phi(x) \leq -cf(x)^{-3/4}$ do not holds, go alone with the smallest eigenvector of $\mathrm{H(x)} = (I - \frac{ee^T}{n})\nabla^2\phi(x)(I - \frac{ee^T}{n})$, and apply the linear search.

Theorem 4: By choosing $\beta = O(f(x)^{-3/4})$, the algorithm is guaranteed to generate a solution $f(x) \leq \epsilon$ in $O\left(\epsilon^{-3/4}\log(\epsilon^{-1})\right)$ iterates.

- The theorem holds without using the Assumption on Hessian projection.
- The results can be extended to general function satisfying local error bound condition.

DR-Potential Reduction: Computational Techniques

Several computational techniques have been applied to accelerate

Scaling and matrix equilibration

- **Solving** $f(x) := 1/2 \parallel D_l A D_r x \parallel^2$ **with diagonal** $D.' s$
- **Using Ruiz, PC,** l_2 **scaling to equilibrate the matrix**
- **Adaptively adjust** $D's$ **during algorithm iteration**

Line-search

- **Given direction** d**, line-search reduces potential** $\phi(x + \alpha d)$

Other techniques

- **Iteration averaging**
- **Restart by projective transformation**
- **Curvature filtering**
- **Interior point cleanup**
- **…**

Computational techniques: Averaging and Restart

Iteration averaging

- **maintains a window of past iterates** $X = [x^k, \ldots, x^{k+w}]$
- **finds affine combination** $\alpha = (\alpha_1, \ldots, \alpha_w)$ **to minimize** $1/2 \parallel AX\alpha \parallel^2$
- **similar spirits to Anderson acceleration**
- **the QP is solved using primal-dual interior point method at cost** $O(nw^3)$

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\|AX\alpha\|^2 \\ \text{subject to} \quad & X\alpha \geq 0 \\ & e^\top \alpha = 1 \end{aligned}$$

After averaging, we restart via Kamarkar's projective transformation

$$\frac{1}{2}\|Ax\|^2 = \frac{1}{2}\|(AX)e\|^2 = \frac{1}{2}\|\hat{A}e\|^2$$

- **restart from center of simplex**
- **improve numerical stability**

Numerical Experiments: Netlib and Large Instances

- **114 Netlib LP instances**
- **Solving to 10^{-4} relative tolerance**
- **600 seconds per-instance**
- **Allow final interior point iterations for cleanup**

Method	#Solved
Raw algorithm	$\sim$20
+Scaling	$\sim$50
+Negative curvature	$\sim$70
+Averaging and restart	$\sim$90
+Newton cleanup	114/114

Table 1. Techniques vs. performance

Pure first-order methods work particularly well on LPs with matrix coefficients $\{1, -1, 0\}$

- **Controlled tabular adjustment**
- **Set partitioning**
- **PageRank**

Instance	Iteration to 1e-04	Instance	Iteration to 1e-04
L2CTA3D	320	CTA-15-15-10	< 320
CTA-15-15-10	< 320	CTA-15-15-25	< 320
CTA-15-15-25	< 320	scpm1	Around 4096
CTA-25-25-25	< 320	scpn2	Around 4096
CTA-15-15-10	< 320	scpk4	Around 4096
CTA-15-15-25	< 320	scpn2	Around 4096

Summary of DRSOM for LP

- Able to make use of dual information.
- Provide estimation of both primal and dual solutions.
- Faster speed in a few problems.
- Robust under noise.
- QP sub-problem solver

Overall Takeaways

Second-Order Information matters and simple accelerated SOMs, with various computation tricks, work as faster as FOMs!

Algorithm customization/individualization is very helpful

- **THANK YOU**