Recent Developments on Optimization Algorithms and Applications

MAY 11, 2023

Yinyu Ye
Stanford University and CUHKSZ (Sabbatical Leave)

Today’s Talk

I. Accelerated Second-Order Methods and Applications

II. Pre-Trained Statistical Cut Generation for Mixed-Integer Linear Programming Solvers

I. 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)

Classic Methods for General Convex/Nonconvex Optimization

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

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, 2022)

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

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

with δ set to be O($\sqrt{\epsilon}$) !

- **Accessible at the cost of $O\left(\mathrm{n}^2 \epsilon^{-1/4}\right)$ via the randomized Lanczos**

Theoretical Guarantees of HSODM

- **Consider use the second-order homogenized direction, and the length of each step $\|\eta\xi\|$ is fixed: $\|\eta\xi\| \le \Delta_k = \frac{2\sqrt{\epsilon}}{M}$ 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, choose $\delta_k = O\left(\|g_k\|^{1/2}\right)$ 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, gradient-dominance, etc.
- Practical remarks: homogenized direction can be used with any Line-search (e.g., Hager-Zhang)

Application I: HSODM for Policy Optimization in Reinforcement Learning

- Consider policy optimization of linearized objective in reinforcement learning

$$\max_{\theta \in \mathbb{R}^d} L(\theta) := L(\pi_\theta),$$

$$\theta_{k+1} = \theta_k + \alpha_k \cdot M_k \nabla \eta(\theta_k),$$

- M_k is usually a preconditioning matrix.

- The Natural Policy Gradient (NPG) method (Kakade, 2001) uses the Fisher information matrix where M_k is the inverse of

$$F_k(\theta) = \mathbb{E}_{\rho_{\theta_k}, \pi_{\theta_k}} \left[\nabla \log \pi_{\theta_k}(s, a) \nabla \log \pi_{\theta_k}(s, a)^T \right]$$

- Based on KL divergence, TRPO (Schulman et al. 2015) uses KL divergence in the constraint:

$$\max_{\theta} \nabla L_{\theta_k}(\theta_k)^T (\theta - \theta_k)$$
$$\text{s.t. } \mathbb{E}_{s \sim \rho_{\theta_k}} [D_{KL}(\pi_{\theta_k}(\cdot \mid s); \pi_\theta(\cdot \mid s))] \leq \delta.$$

Homogeneous NPG:
Apply the homogenized model!

HSODM for Policy Optimization in RL I

- Consider **Homogeneous NPG** in reinforcement learning

$$\min_{\|[v;t]\|\le 1} \begin{bmatrix} v \\ t \end{bmatrix}^T \begin{bmatrix} F_k & g_k \\ g_k^T & -\delta \end{bmatrix} \begin{bmatrix} v \\ t \end{bmatrix}$$

- F_k is an estimation of Fisher matrix, see, Schulman et al. 2015, 2017

$$F_k(\theta) = \mathbb{E}_{\rho_{\theta_k}, \pi_{\theta_k}} \left[\nabla \log \pi_{\theta_k}(s, a) \nabla \log \pi_{\theta_k}(s, a)^T \right]$$

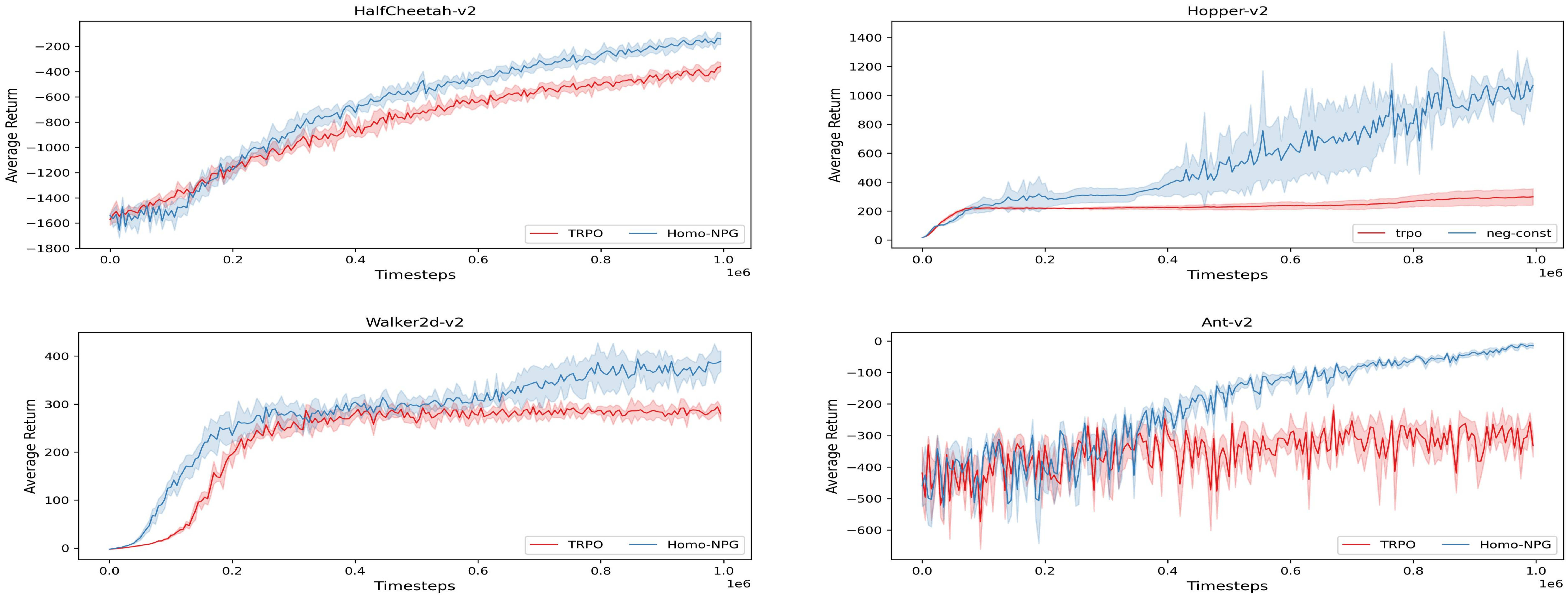
- We set a proper δ to work with "gradient dominance condition".

- After solving a direction d_k, similarly, apply a line-search in practice

- Ongoing: convergence analysis for HSODM in RL.

HSODM for Policy Optimization in RL II

- A comparison of Homogeneous NPG and Trust-region Policy Optimization (Schultz, 2015)



- Homogeneous model provides significant improvements over TRPO
- Ongoing: second-order information?
- **Further reduce the computation cost per step**

Dimension Reduced Second-Order Method (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 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 2-D trust-region problem to decide “two step-sizes”:

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

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

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

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, terminates in n steps

Computing the two-dimensional quadratic model 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, 2022)

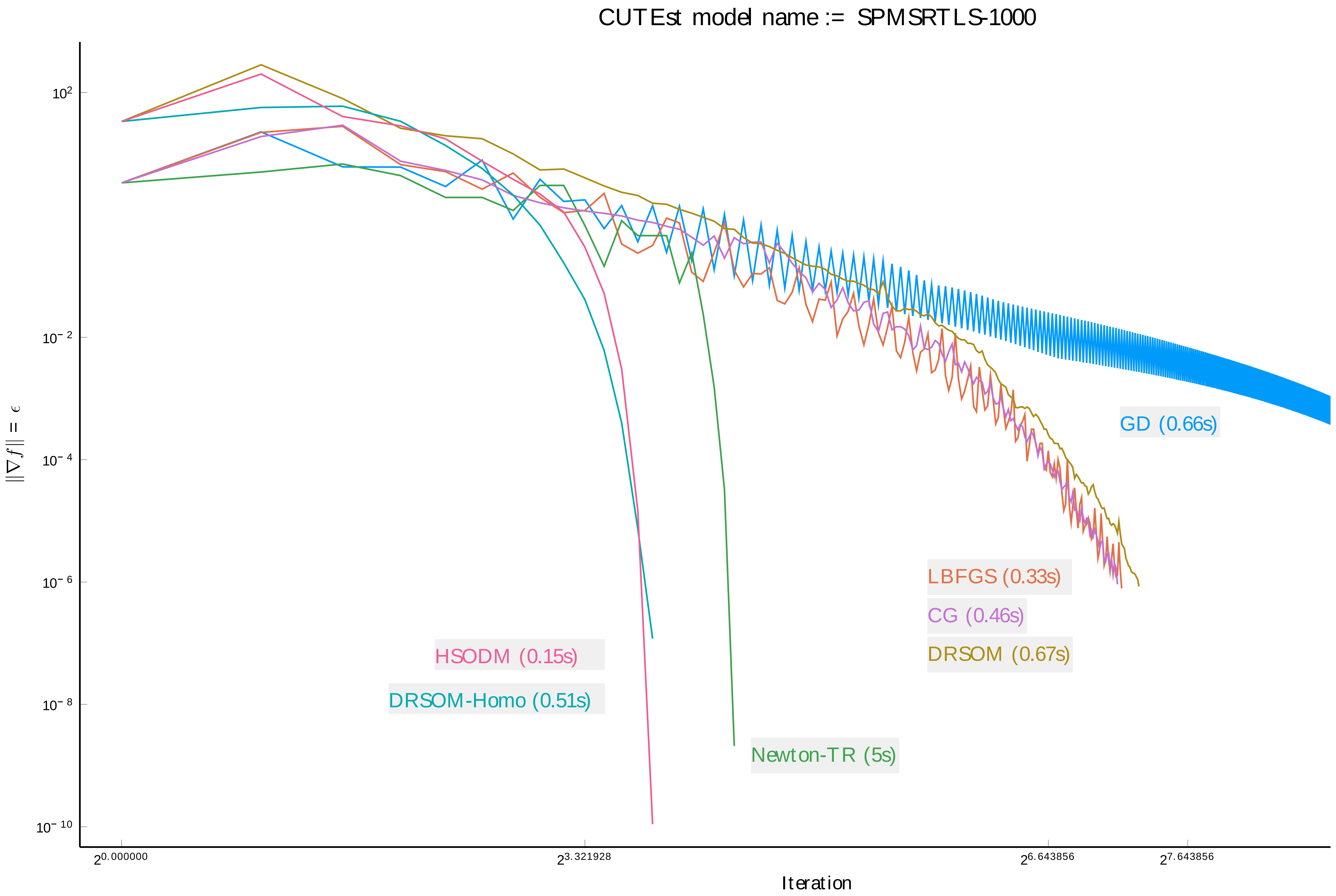
Assumption. (a) f has Lipschitz continuous Hessian. (b) **If the Lagrangian multiplier $\lambda_k < \sqrt{\epsilon}$, assume $\| (H_k - \widetilde{H}_k)d_{k+1} \| \leq C \| d_{k+1} \|^2$ (Cartis et al.),** where $\widetilde{H}_k$ is the projected Hessian in the subspace (commonly adopted for approximate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



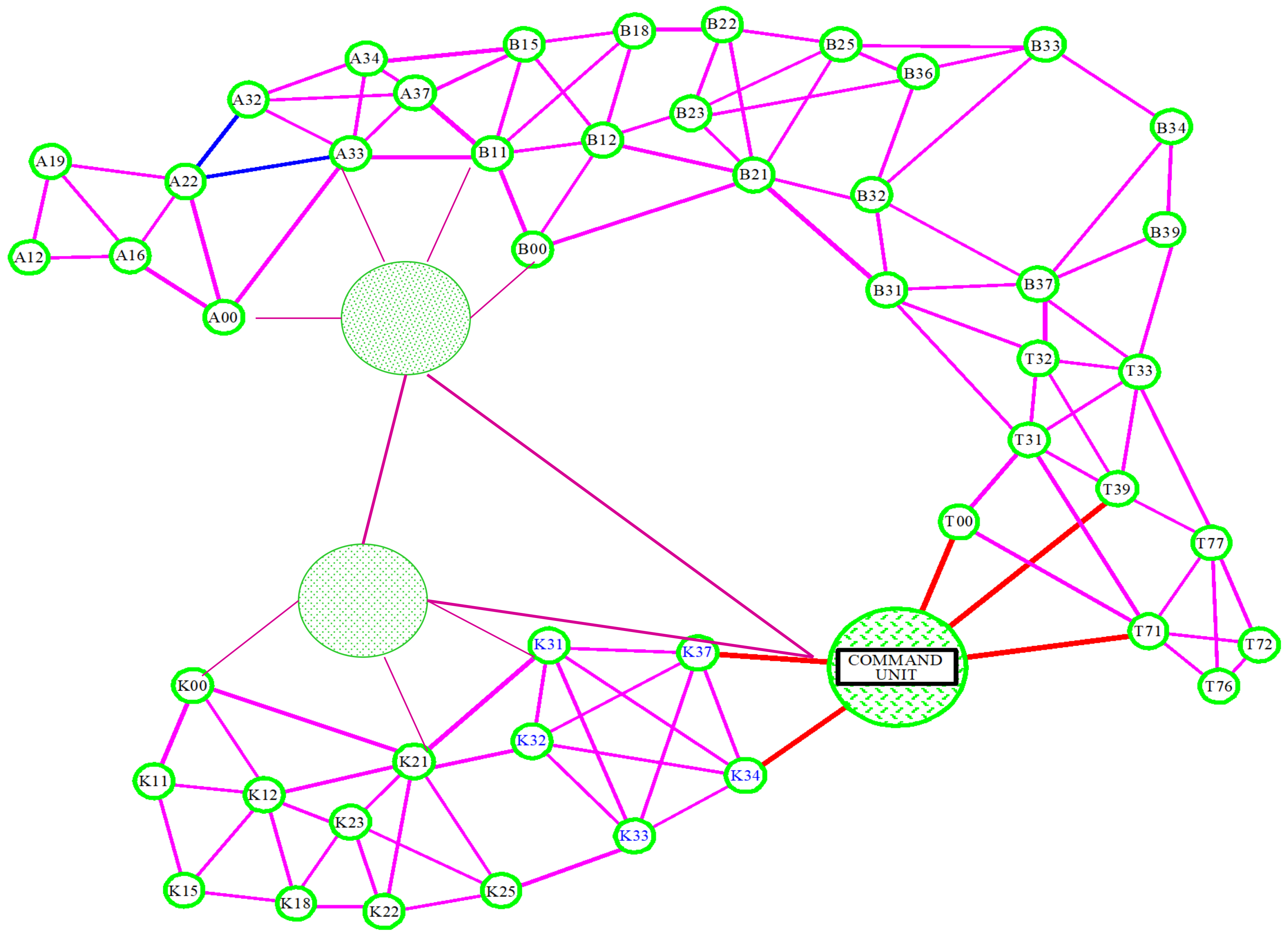
CUTEst example

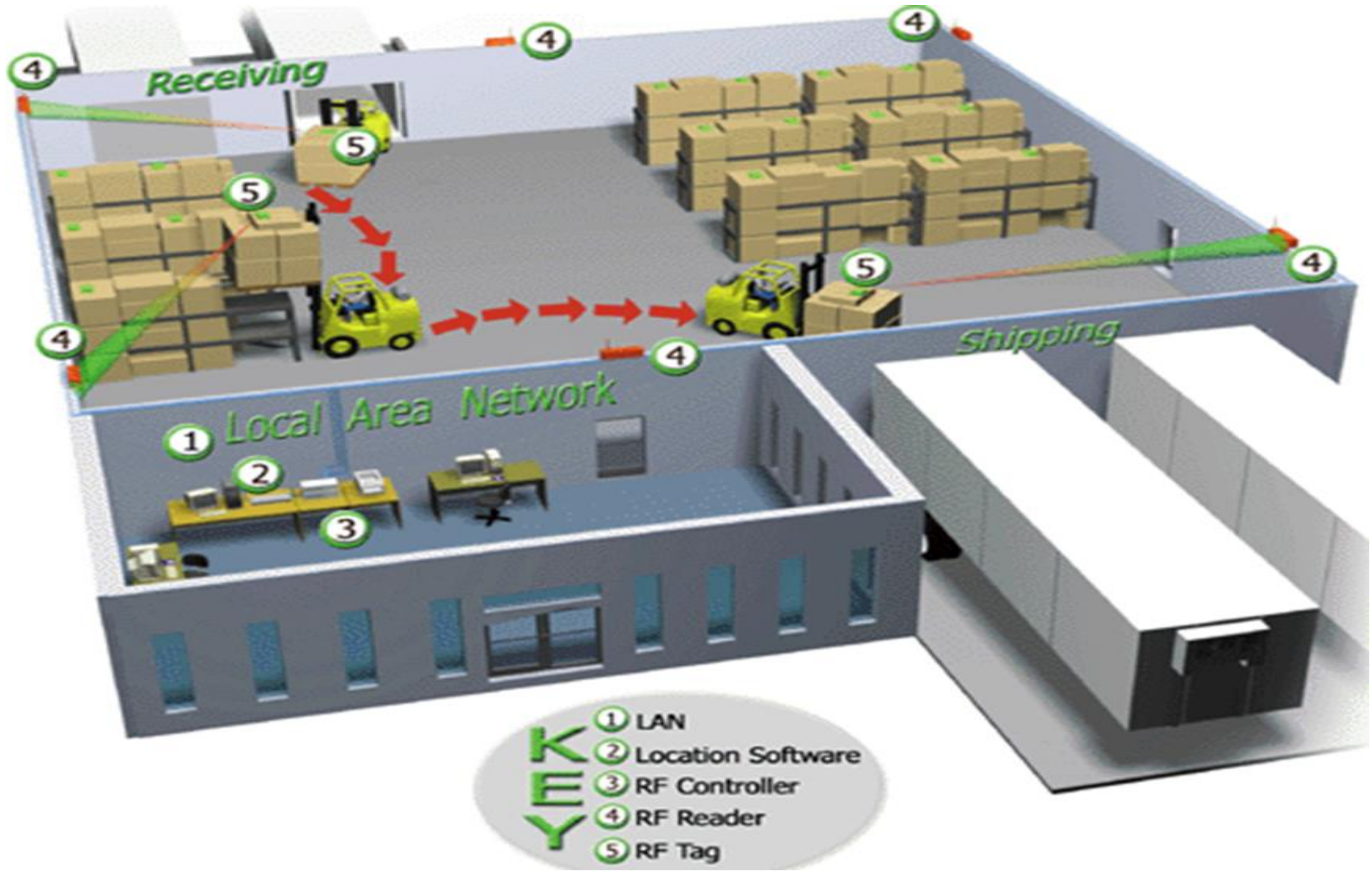
- *GD* and *LBFGS* both use a Line-search (Hager-Zhang)
- *DRSOM* uses 2-D subspace
- HSODM and DRSOM + HSODM are much better!
- *DRSOM* can also benefit from the homogenized system

Application II: Sensor Network Location (SNL)

- Localization
  - –Given partial pair-wise measured distance values
  - –Given some anchors' positions
  - –Find locations of all other sensors that fit the measured distance values

This is also called graph realization on a fixed dimension Euclidean space





Mathematical Formulation of Sensor Network Location (SNL)

- Consider Sensor Network Location (SNL)

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

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

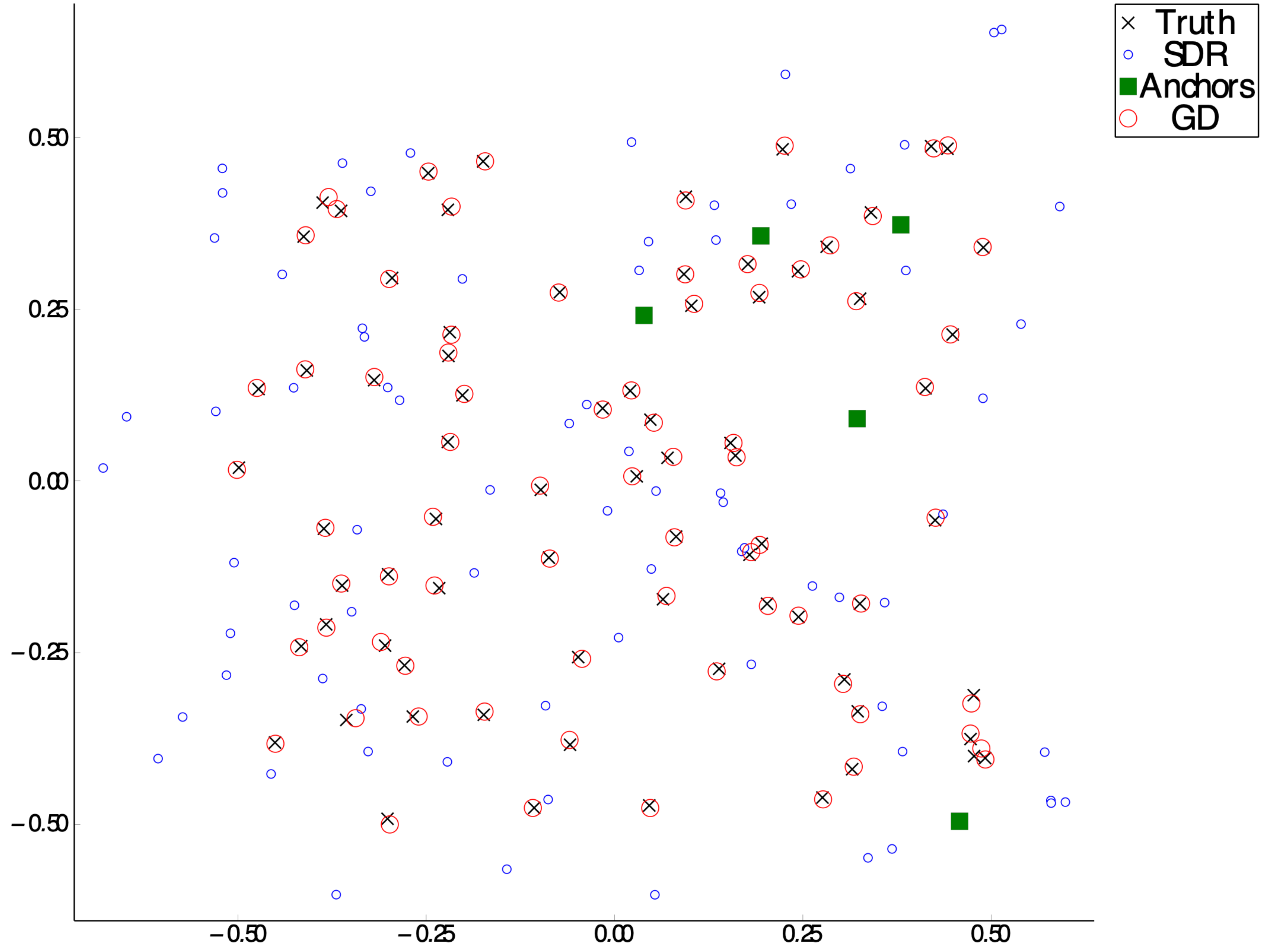
$$\|x_i - x_j\|^2 = d_{ij}^2, \forall (i,j) \in N_x$$
$$\|x_i - a_k\|^2 = \bar{d}_{ik}^2, \forall (i,k) \in N_a$$

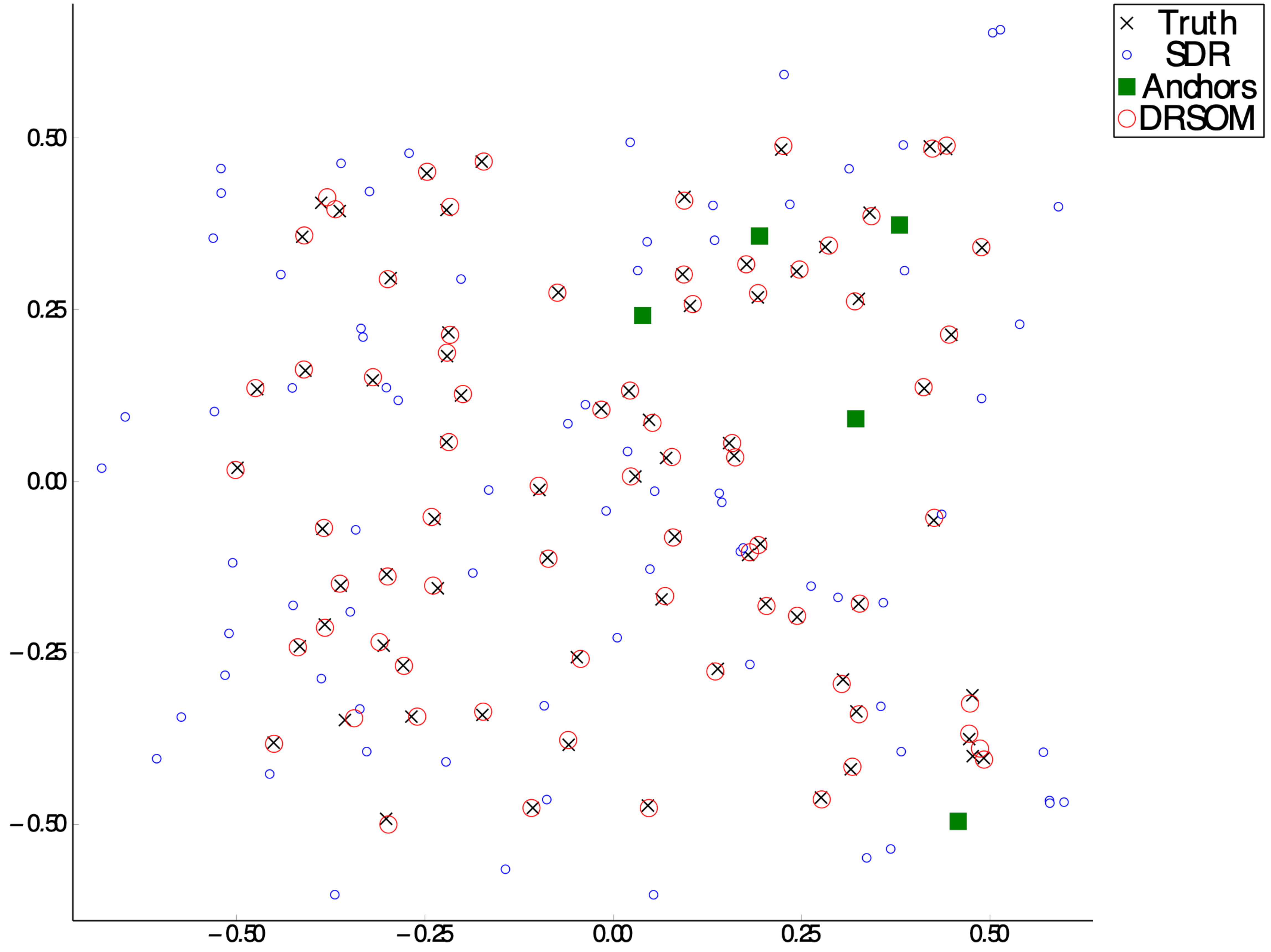
- We can solve SNL by the nonconvex nonlinear least square (NLS) problem

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

Sensor Network Location (SNL)

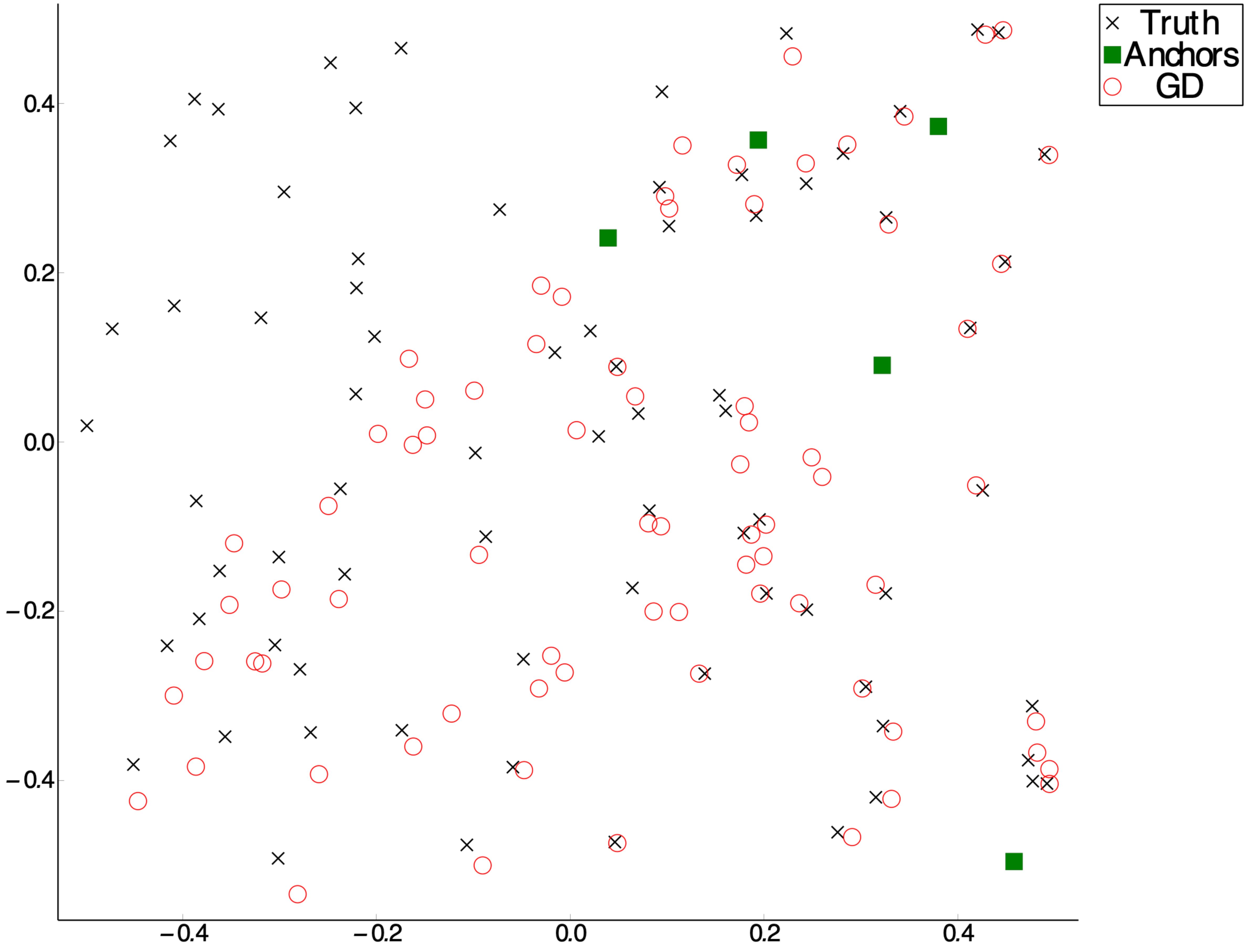
- Graphical results using SDP relaxation (Biswas&Y 2004, SO&Y 2007) to initialize the NLS
- n = 80, m = 5 (anchors), radio range = 0.5, degree = 25, noise factor = 0.05
- Both Gradient Descent and DRSOM can find good solutions !

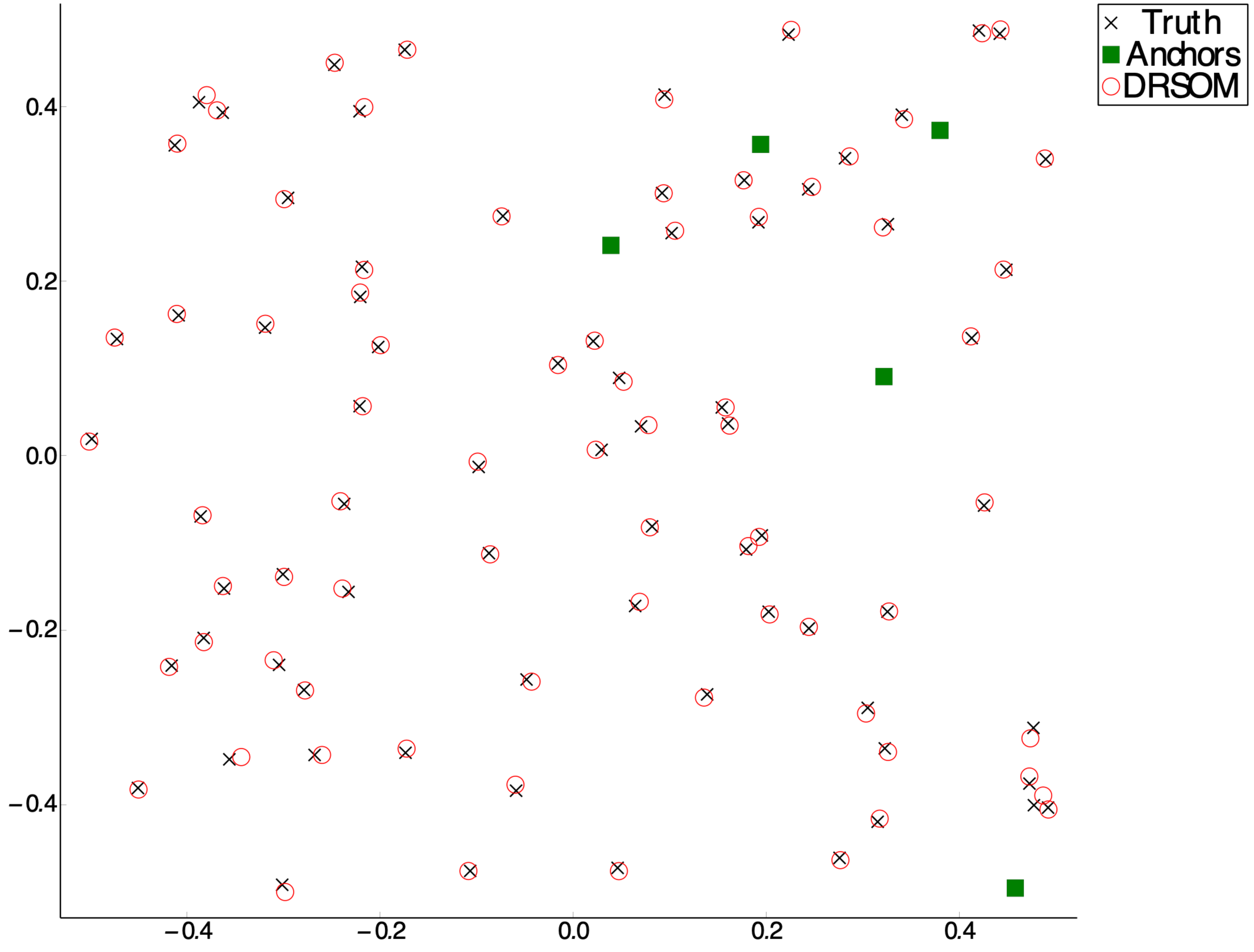




Sensor Network Location (SNL)

- 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 $|\boldsymbol{g_k}| \leq \mathbf{1}\boldsymbol{e}^{-5}$)
- Compare GD, CG, and DRSOM. (GD and CG use Hager-Zhang Linesearch)

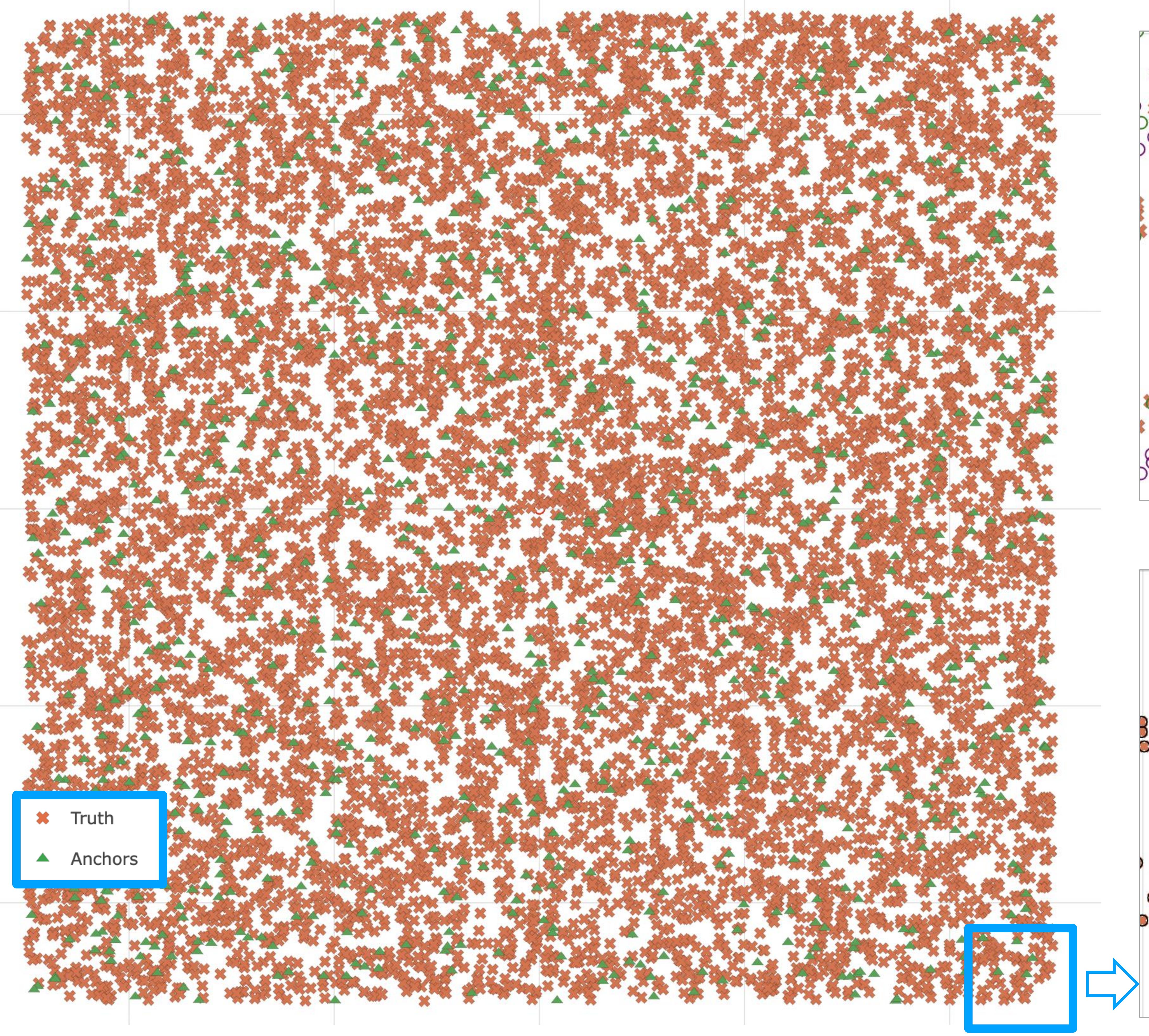
n	m	$\|E\|$	t		
			CG	DRSOM	GD
500	50	2.2e+04	1.7e+01	1.1e+01	2.3e+01
1000	80	4.6e+04	7.3e+01	3.9e+01	1.8e+02
2000	120	9.4e+04	2.5e+02	1.4e+02	1.1e+03
3000	150	1.4e+05	6.5e+02	1.4e+02	-
4000	400	1.8e+05	1.3e+03	5.0e+02	-
6000	600	2.7e+05	2.0e+03	1.1e+03	-
10000	1000	4.5e+05	-	2.2e+03	-

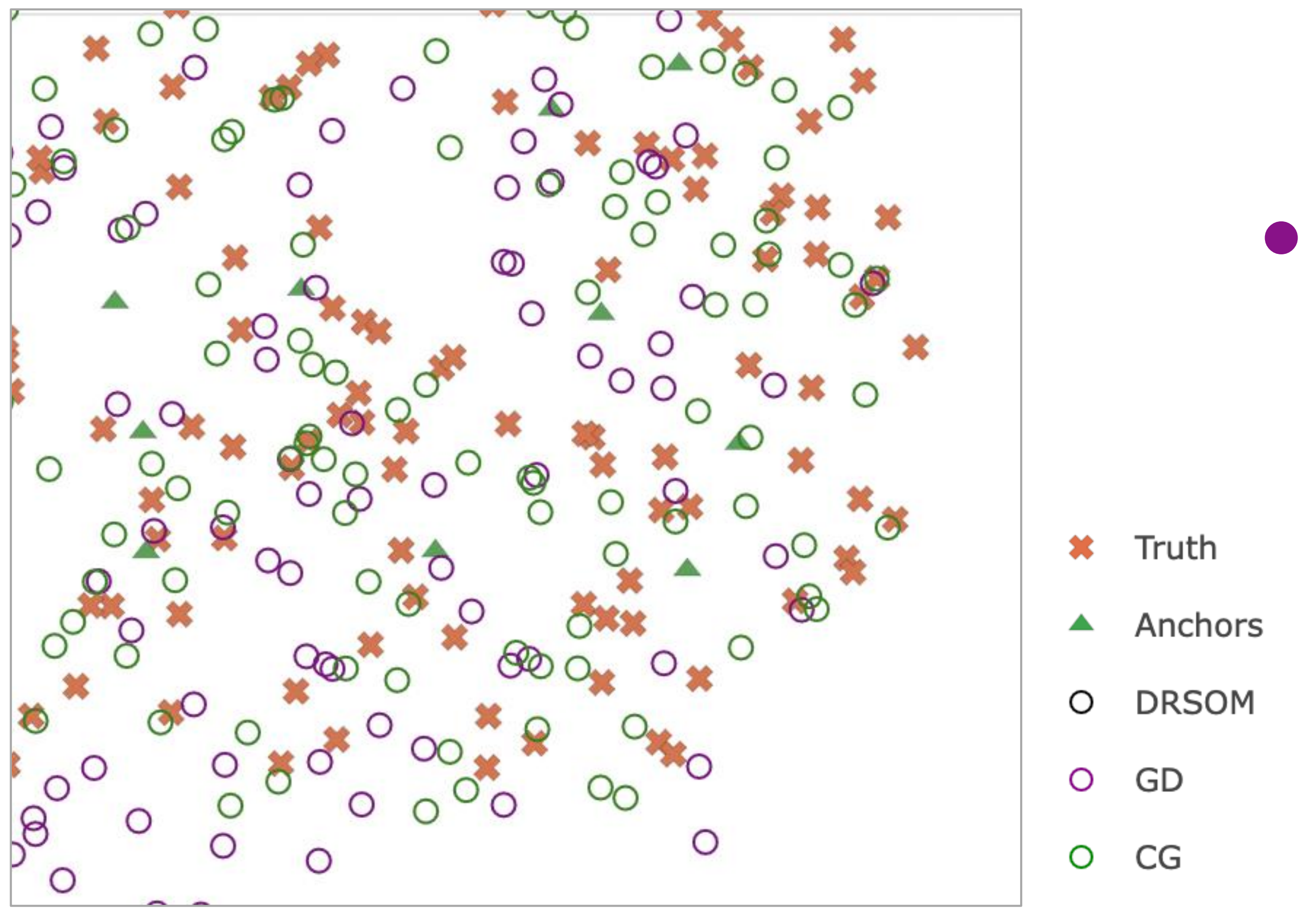
Table 2: Running time of CG, DRSOM, and GD on SNL instances of different problem size, $|E|$ denotes the number of QCQP constraints. "-" means the algorithm exceeds 3,000s.

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

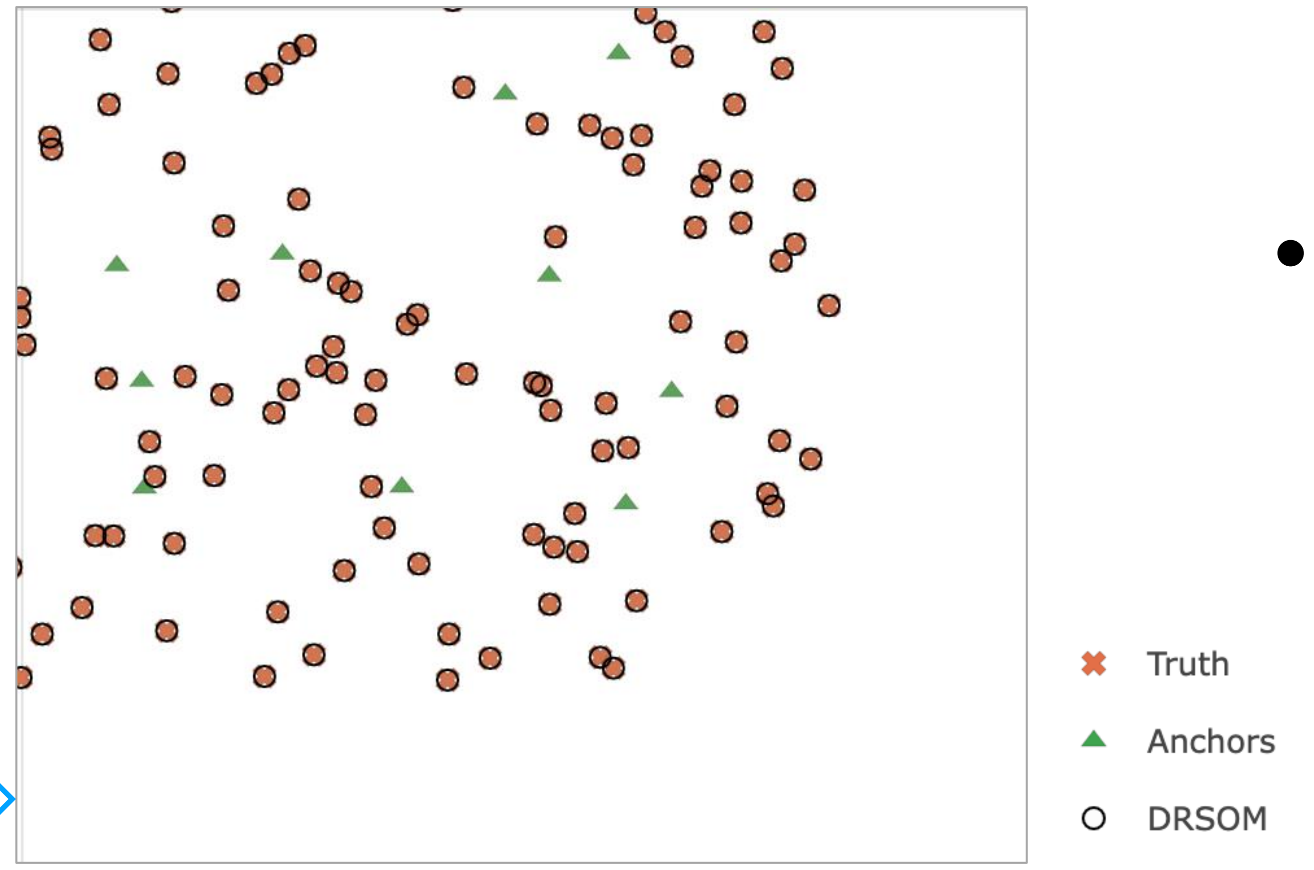
Sensor Network Location, Large-scale instances

- 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**



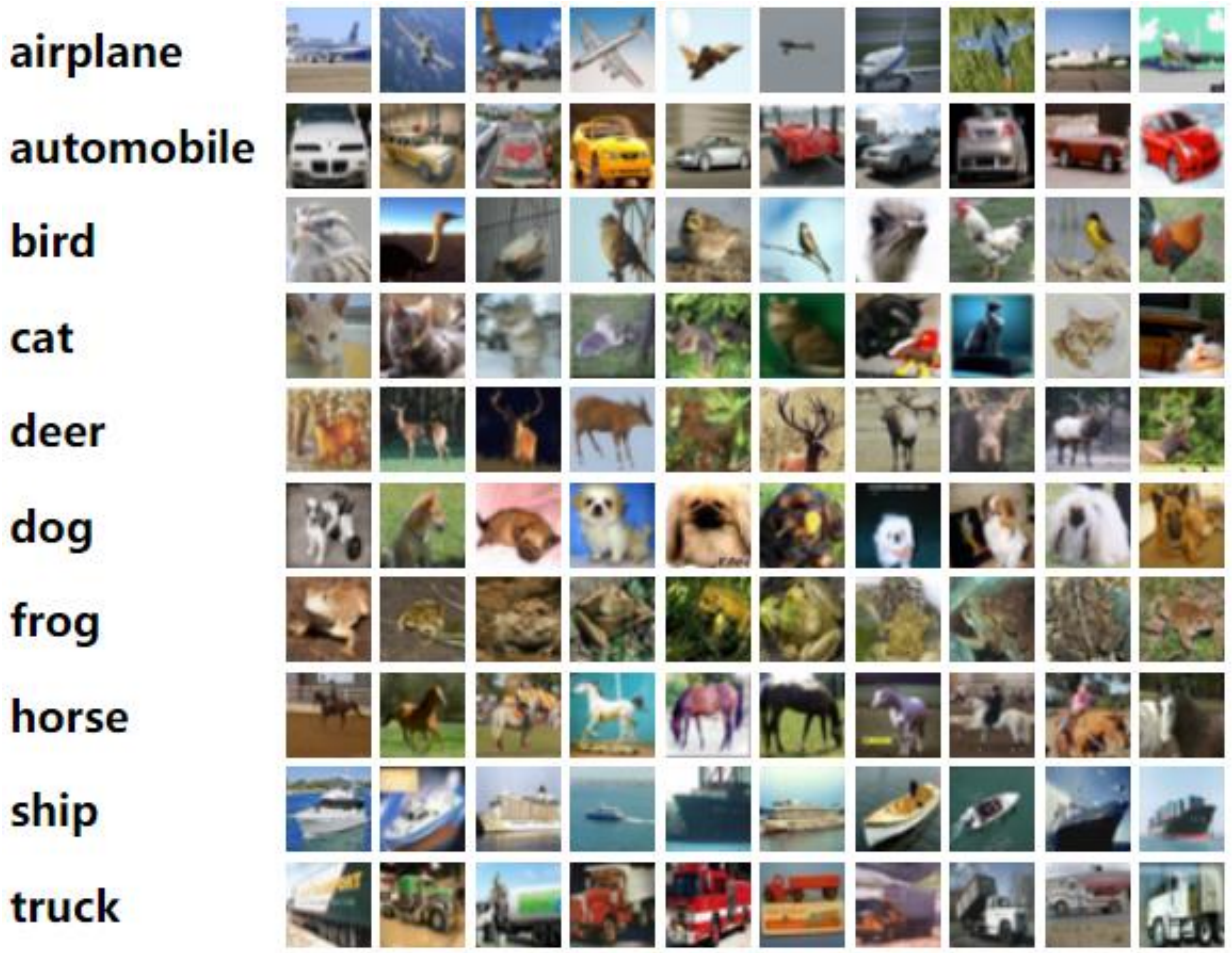
- **DRSOM can converge to $|g_k| \le 1e^{-5}$ in 2,200s**

Sensor Network Online Tracking, 2D and 3D

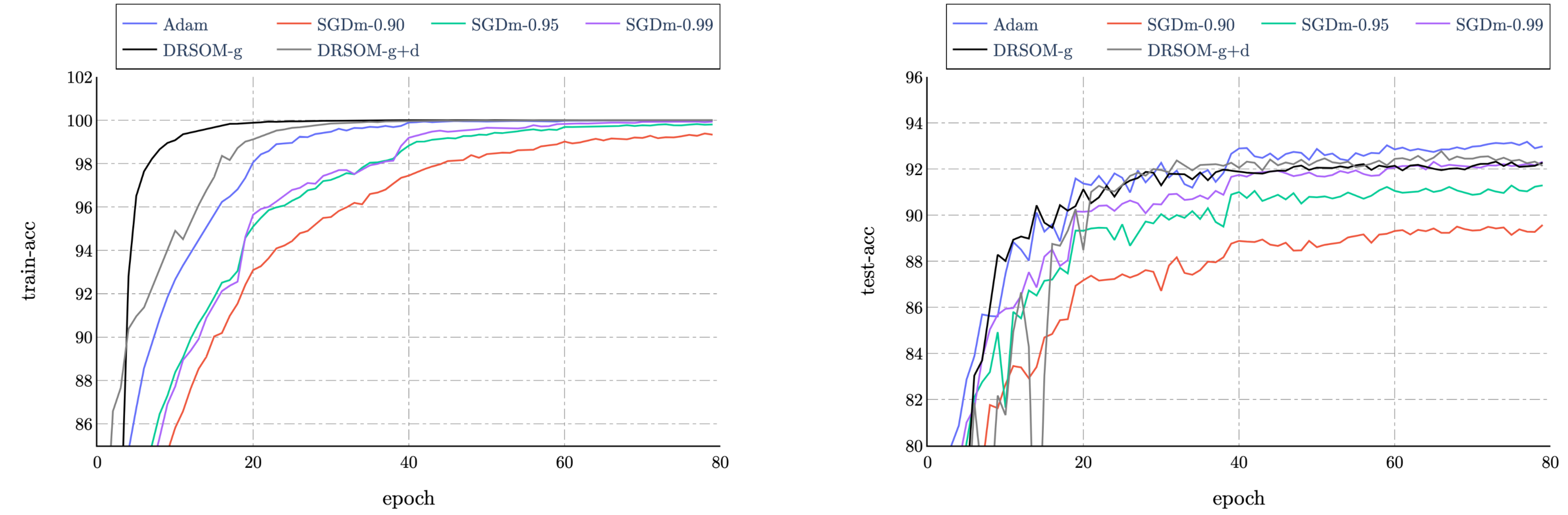
Application III: Neural Networks and Deep Learning

To use DRSOM in machine learning problems

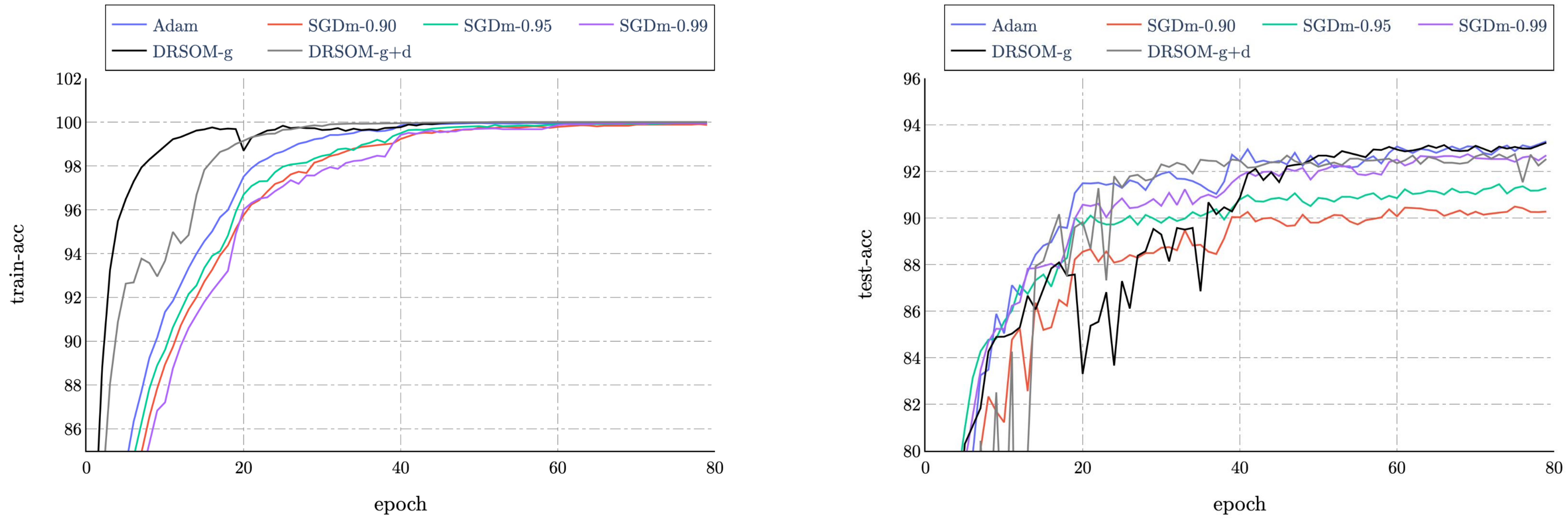
- We apply the mini-batch strategy to a vanilla DRSOM
- Use Automatic Differentiation to compute gradients
- Train ResNet18/Resnet34 Model with CIFAR 10
- Set Adam with initial learning rate 1e-3



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!

Overall Takeaways

Second-Order Derivative information matters and better to integrate FOM and SOM for nonlinear optimization!

It is possible to train Mixed-Integer Linear Programming Solvers and add Statistical Confidence Cuts to significantly accelerate the solution process.

- THANK YOU