Partial-Hessian Strategies for Fast Learning of Nonlinear Embeddings

Max Vladymyrov and **Miguel Á. Carreira-Perpiñán**

EECS, School of Engineering, University of California, Merced

1 Abstract

Stochastic neighbor embedding (SNE) and related nonlinear manifold learning algorithms achieve high-quality low-dimensional representations of similarity data, but are notoriously slow to train. We propose a generic formulation of embedding algorithms that includes SNE and other existing algorithms, and study their relation with spectral methods and graph Laplacians. This allows us to define several partial-Hessian optimization strategies, characterize their global and local convergence, and evaluate them empirically. We achieve up to two orders of magnitude speedup over existing training methods with a strategy (which we call the **spectral direction**) that adds nearly no overhead to the gradient and yet is simple, scalable and applicable to several existing and future embedding algorithms.

2 General Embedding Formulation

For $\mathbf{Y} \in \mathbb{R}^{D\times N}$ - high-dimensional data set and $\mathbf{X} \in \mathbb{R}^{d\times N}$ its its low-dimensional projection we can formulate several well-known dimensionality reduction techniques as:

$$E(\mathbf{X};\lambda) = E^+(\mathbf{X}) + \lambda E^-(\mathbf{X}) \qquad \lambda \geq 0$$

where E^+ is an attractive term, often quadratic psd and minimal with coincident points, and E^- is a repulsive term, often nonlinear and minimal when points separate infinitely. Special cases include:

- **Symm. Stochastic Neighbor Embedding (s-SNE)** and t**-SNE** define a posterior probability distributions P and Q in $\mathbf{X}$ and $\mathbf{Y}$ spaces resp. for a given kernel function $K(\|\mathbf{x}_n - \mathbf{x}_m\|^2)$. The objective function minimizes the KL divergence between the two (λ is equal to 1).
- **Elastic Embedding (EE)** goes without distributions and is simpler.
- **Laplacian Eigenmaps (LE)** and **Locally Linear Embedding (LLE)** minimize only attractive term (equivalent to $\lambda = 0$), but add quadratic constraints to eliminate the trivial solution $\mathbf{X} = \mathbf{0}$.

Call $d_{nm} = \|\mathbf{x}_n - \mathbf{x}_m\|^2$. Then, the objective functions can be reformulated as :

s-SNE: $E^+ = \sum_{n,m=1}^{N} p_{nm} d_{nm}$ $\quad E^- = \log \sum_{n,m=1}^{N} e^{-d_{nm}}$

t-SNE: $E^+ = \sum_{n,m=1}^{N} p_{nm} \log\left(1 + e^{-d_{nm}}\right)$ $\quad E^- = \log \sum_{n,m=1}^{N} \left(1 + e^{-d_{nm}}\right)^{-1}$

EE: $E^+ = \sum_{n,m=1}^{N} w^+_{nm} d_{nm}$ $\quad E^- = \sum_{n,m=1}^{N} w^-_{nm} e^{-d_{nm}}$

LE & LLE: $E^+ = \sum_{n,m=1}^{N} w^+_{nm} d_{nm}$ $\quad E^- = 0$

5 Experimental Evaluation

In the experiments we compared: gradient descent (GD), fixed-point diagonal iterations (FP), the diagonal of the full Hessian (DiagH), spectral direction (SD), partial Hessian (SD–), nonlinear Conj. Grad. (CG) and L-BFGS;

Method:	GD	FP	DiagH	SD	SD–
$\mathbf{B}_k$:	$\mathbf{I}$	$4\mathbf{D}^+$	$4\mathbf{D}^+ + 8\lambda\mathbf{D}^{xx}_{i*,i*}$	$4\mathbf{L}^+$	$4\mathbf{L}^+ + 8\lambda\mathbf{L}^{xx}_{i*,i*}$
$\mathbf{B}_k\mathbf{p}_k = -\mathbf{g}_k$:	–	Exact	Exact	trian. sys.	lin. conj. grad.

3 Partial-Hessian Strategies

We search for a descent search direction as a solution to $\mathbf{B}_k\mathbf{p}_k = -\mathbf{g}_k$, where $\mathbf{g}_k$ is the gradient at iteration k and $\mathbf{B}_k$ is a pd matrix. We want $\mathbf{B}_k$ to be a psd part of the Hessian such that it contains as much Hessian information as possible, it is fast to compute and it scales up to larger N.

Given a graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ with $\mathbf{D} = \operatorname{diag}\left(\sum_{n=1}^N w_{nm}\right)$ as a degree matrix ($\mathbf{L}$ is psd if the entries of $\mathbf{W}$ are non-negative). Then the Hessian of generalized embeddings is a $Nd \times Nd$ matrix given by:

$$\nabla^2 E = 4\mathbf{L} \otimes \mathbf{I}_d + 8\mathbf{L}^{xx} - 16\lambda \operatorname{vec}(\mathbf{X}\mathbf{L}^q)\operatorname{vec}(\mathbf{X}\mathbf{L}^q)^T$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix, and the weights of corresponding graph Laplacians depend on the particular method:

	w_{nm}	$w^{xx}_{in,jm}$	w^q_{nm}
s-SNE	$p_{nm} - \lambda q_{nm}$	$\lambda q_{nm}(x_{in} - x_{im})(x_{jn} - x_{jm})$	$-q_{nm}$
t**-SNE**	$K(p_{nm} - \lambda q_{nm})$	$K^2(2\lambda q_{nm} - p_{nm})(x_{in} - x_{im})(x_{jn} - x_{jm})$	$-K^2 q_{nm}$
EE	$w^+_{nm} - \lambda w^-_{nm} e^{-d_{nm}}$	$\lambda w^-_{nm} e^{-d_{nm}}(x_{in} - x_{im})(x_{jn} - x_{jm})$	0

Note that in both cases the weights p_{nm} and q_{nm} as well as w^+_{nm} and w^-_{nm} are always positive and $w^{xx}_{in,jm}$ has a constant sign for $i = j$.

4 The Spectral Direction

The partial Hessian constructed from the attractive Hessian $\mathbf{B}_k = \nabla^2 E^+(\mathbf{X}) = 4\mathbf{L}^+ \otimes \mathbf{I}_d$ compromises the best between deep descent and efficient computation, and yields what we call the spectral direction:

```
SpectralDirection($\mathbf{X}_0$, $\mathbf{W}^+$, $\kappa$)
$\mathbf{L}^+ \leftarrow \mathbf{D}^+ - \mathbf{W}^+$
Sparsify $\mathbf{L}^+$ with $\kappa$-NN graph
$\mathbf{R} \leftarrow \mathtt{chol}(\mathbf{L}^+)$
$k \leftarrow 1$
repeat
  Compute $\mathbf{g}_k$ and $E_k$
  $\mathbf{p}_k \leftarrow -\mathbf{R}^{-T}(\mathbf{R}^{-1}\mathbf{g}_k)$
  $\alpha \leftarrow$ backtracking line search
  $\mathbf{X}_k \leftarrow \mathbf{X}_{k-1} + \alpha\mathbf{p}_k$
$k \leftarrow k + 1$
until stop
return $\mathbf{X}$
```

- it guaranties to be globally convergent from any initialization.
- it is block-diagonal and consists of d identical blocks of $N \times N$ graph Laplacian $\mathbf{L}^+$;
- it is constant for Gaussians kernels and can be made constant for other kernels, thus it is computed just once for all iterations and values of homotopy parameter λ;
- we can further sparsify $\mathbf{L}^+$ through κ-nearest-neighbor graph:

$$\kappa = N; \mathbf{B}_k = \mathbf{L}^+ \xrightarrow{\text{more sparsity}} \kappa = 0; \mathbf{B}_k = \mathbf{D}^+$$

- we precompute the Cholesky factorization $4\mathbf{L}^+ = \mathbf{R}^T\mathbf{R}$ for $\mathcal{O}(\frac{1}{3}N^3)$ and then solve two triangular systems $\mathbf{R}^T(\mathbf{R}\mathbf{p}_k) = -\mathbf{g}_k$ for every iteration k $\left(\mathcal{O}(N^2 d)\right)$. This is much faster than solve the linear system $\left(\mathcal{O}(N^3 d)\right)$ for each iteration;
- we "bend" the exact gradient of the nonlinear E using the curvature of the spectral E^+.

6 Conclusions

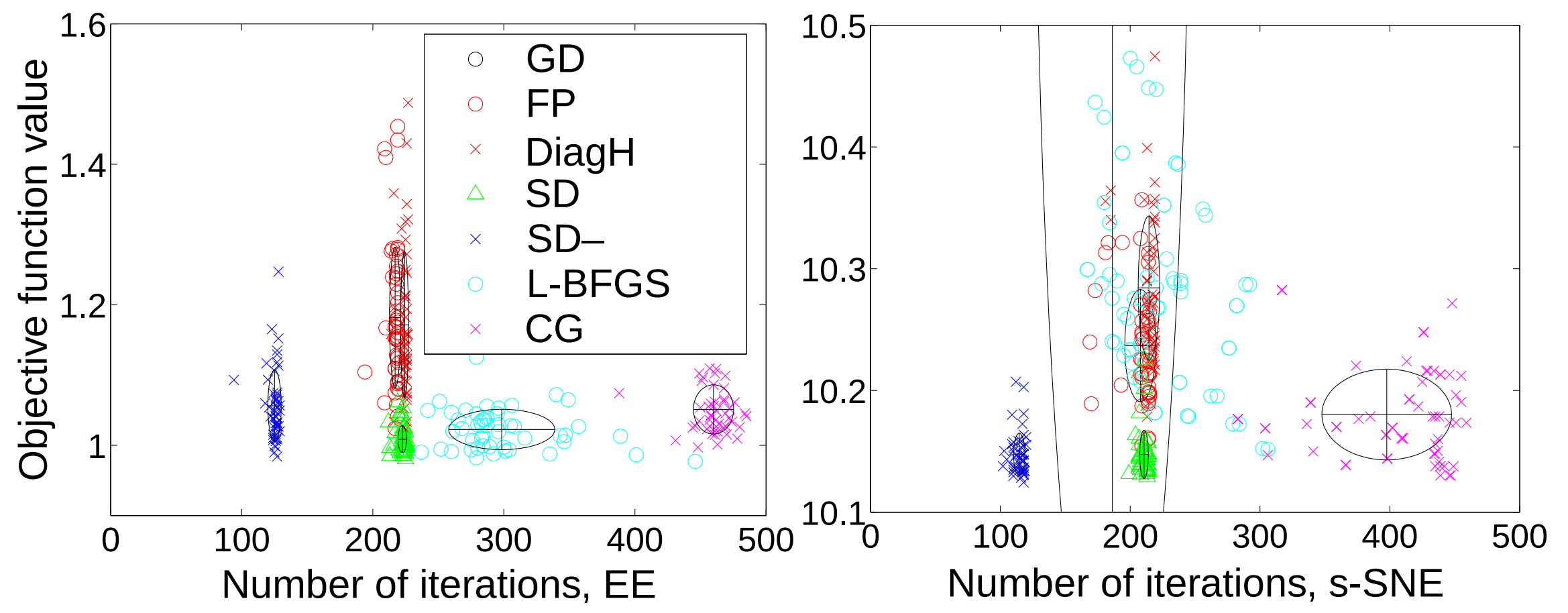
- We presented general formulation of such methods as **SNE**, **s-SNE**, t**-SNE**, **EE**, **LE** and **LLE**, and also suggest new ones.
- We showed the role of graph Laplacians in the gradient and Hessian, and derived several partial-Hessian optimization strategies.
- We presented a new simple, generic and scalable optimization strategy based on the Cholesky factors of the (sparsified) attractive Laplacian. The preferred method is able to achieve 1–2 orders of magnitude speed-up compared to traditional methods.
- Matlab implementation is available online at authors' websites.

Partially supported by NSF CAREER award IIS–0754089.

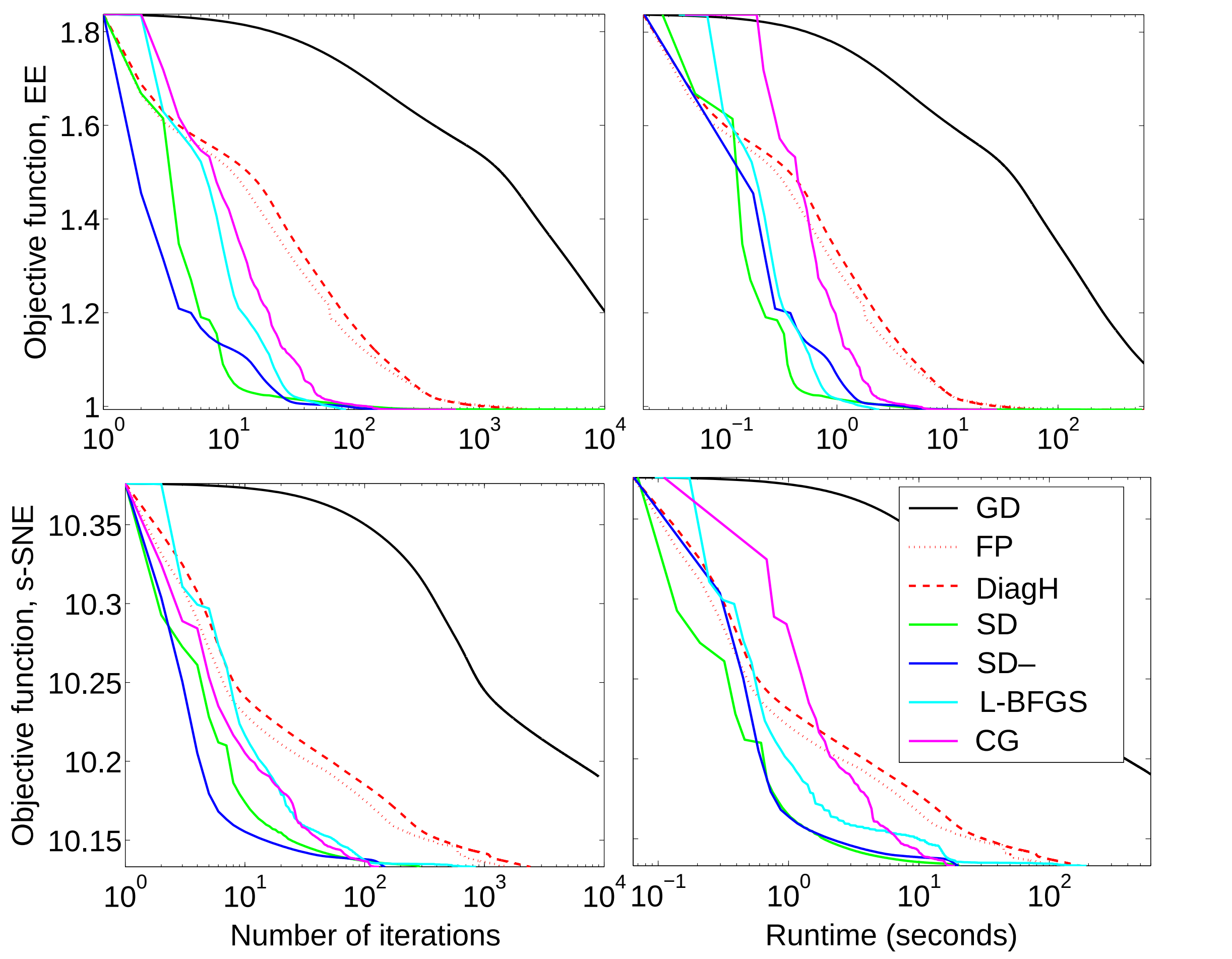
1. COIL-20.

Rotation sequences of 10 objects every 5 degrees; each data point is a greyscale image of 128×128, so $\mathbf{Y}$ has $N = 720$ points in $D = 16\,384$ dimensions. We used SNE affinities with perplexity $k = 20$.

Convergence from random $\mathbf{X}_0$ to possibly different minima. Run from 50 different random points for 20 seconds each.

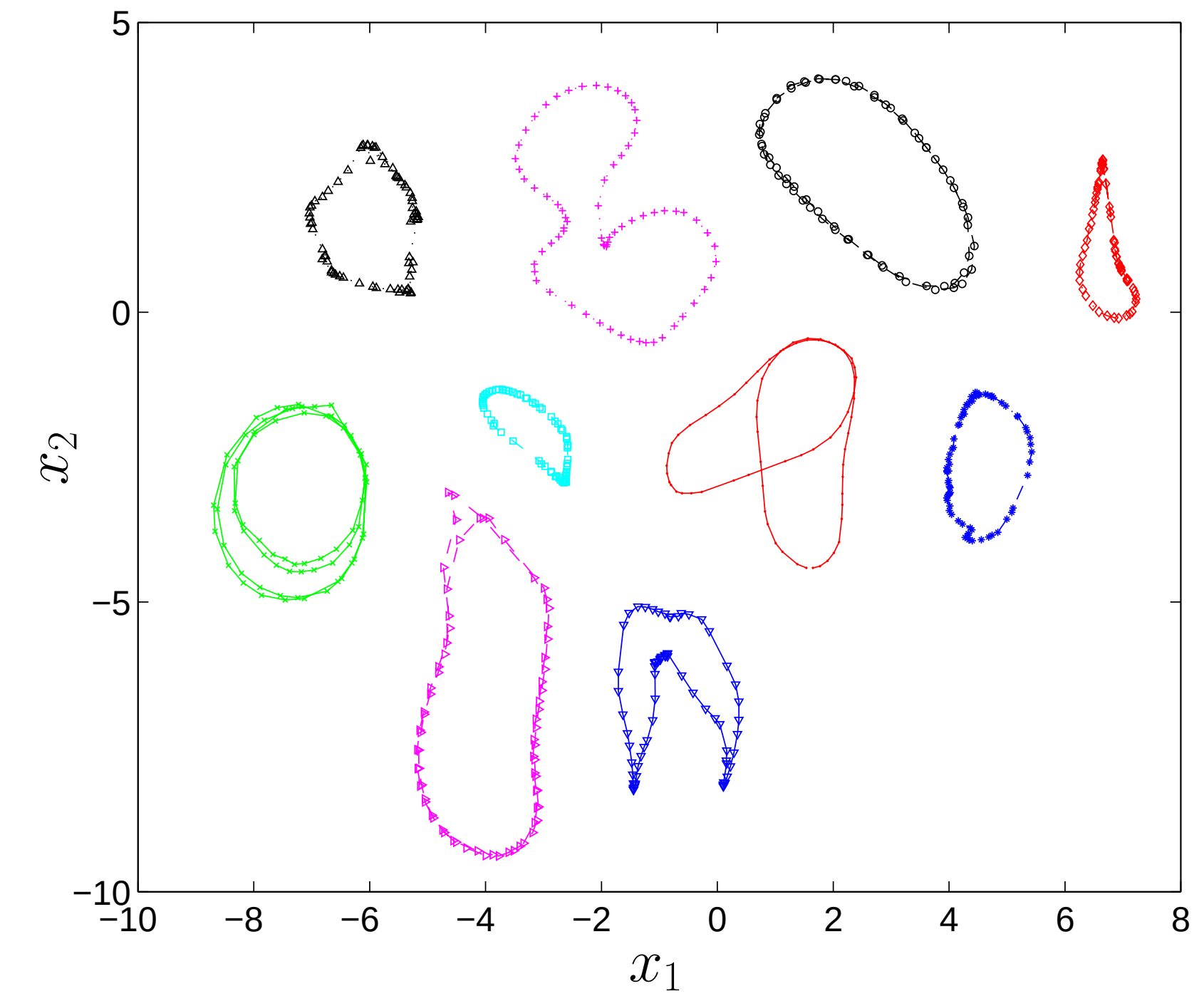


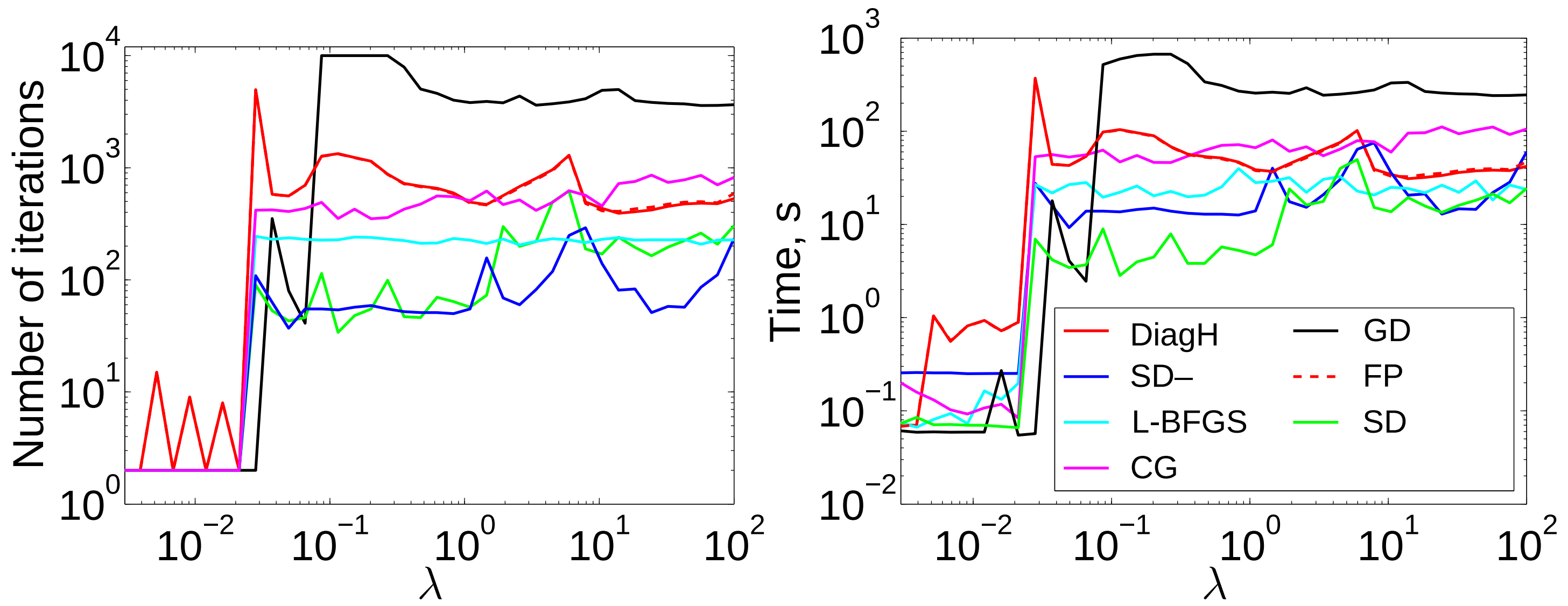
Convergence to the same minimum from the same initial $\mathbf{X}$. Initialize $\mathbf{X}_0$ close to $\mathbf{X}_\infty$ so all methods have the same final point.



Homotopy optimization for EE. Used 50 log-spaced values of λ from 10^{-4} to 10^2 and minimized E at each λ value until the relative error decrease was less than 10^{-6} or we reached 10^4 iterations.

Method:	GD	FP	DiagH	SD	SD–	L-BFGS	CG
E evals	143 237	26 219	26 235	5 183	**2 775**	6 816	16 600
Time	9 291	2 015	2 016	**402**	703	756	2 154

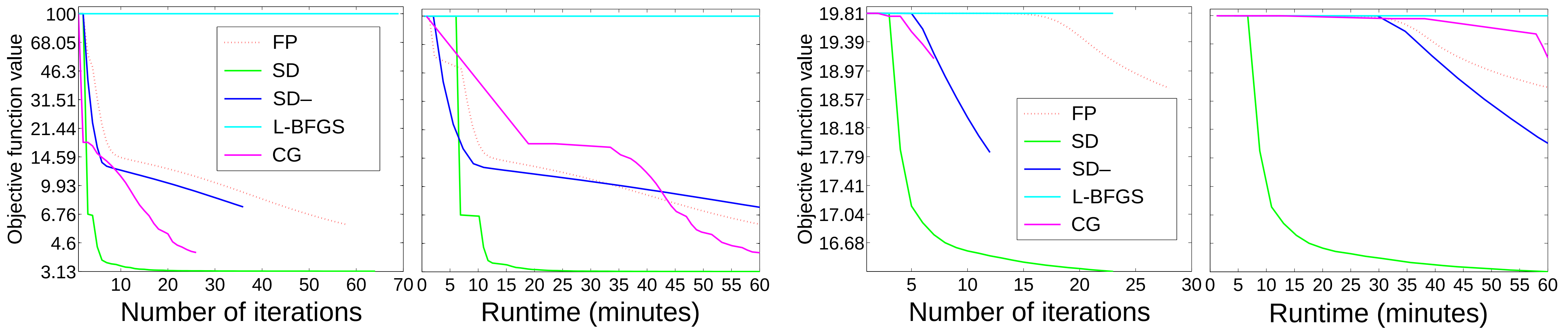




2. MNIST.

$N = 20\,000$ MNIST handwritten digits (each a 28×28 pixel grayscale image, i.e., $D = 784$). Perplexity $k = 50$. Run the EE and t-SNE optimization methods for 1 hour each. For the SD we used $\kappa = 7$.

Elastic Embedding (EE) $\qquad$ t-Stochastic Neighbor Embedding (t-SNE)



Resulting embedding after 20 min $\qquad$ Resulting embedding after 1 hour