

Trust-Region Methods with Low-Fidelity Objective Models

Andrea Angino^[0009-0000-8525-375X],
Matteo Aurina^[0009-0002-2654-7488],
Alena Kopaničáková^[0000-0001-8388-5518],
Matthias Voigt^[0000-0001-8491-1861],
Marco Donatelli^[0000-0001-7958-9126],
Rolf Krause^[0000-0001-5408-5271]

1 Introduction

We consider large-scale unconstrained optimization problems arising from data-driven applications, particularly those encountered in supervised machine learning. Specifically, we focus on binary classification tasks, where the training dataset is given by

$$\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^n \times \mathcal{Y} \mid i = 1, \dots, q\},$$

with $x_i \in \mathbb{R}^n$ representing the i -th feature vector and $y_i \in \mathcal{Y} = \{-1, 1\}$ the corresponding labels. Furthermore, we collect the feature vectors in the data matrix

$$X = [x_1, x_2, \dots, x_q] \in \mathbb{R}^{n \times q},$$

which will be used to define low-fidelity directions in the methods presented below.

The learning task is formulated as an empirical risk minimization, where the objective is a finite sum of individual loss terms evaluated over the dataset, i.e.,

Andrea Angino
UniDistance Suisse, Switzerland, e-mail: andrea.angino@unidistance.ch

Matteo Aurina
Università dell'Insubria, Italy, e-mail: matteoaurina@gmail.com

Alena Kopaničáková
Toulouse INP-ENSEEIH, IRIT, ANITI, France, e-mail: alena.kopanicakova@toulouse-inp.fr

Matthias Voigt
UniDistance Suisse, Switzerland, e-mail: matthias.voigt@fernuni.ch

Marco Donatelli
Università dell'Insubria, Italy, e-mail: marco.donatelli@uninsubria.it

Rolf Krause
King Abdullah University of Science and Technology (KAUST), Saudi Arabia & UniDistance Suisse, Switzerland, e-mail: rolf.krause@kaust.edu.sa

$$\min_{w \in \mathbb{R}^n} f(w) = \frac{1}{q} \sum_{i=1}^q \ell(w; x_i, y_i), \quad (1)$$

where the optimization variable $w \in \mathbb{R}^n$ denotes classifier weights and $\ell(\cdot; x_i, y_i)$ is a loss function measuring the misfit between the model prediction and the label for the i -th data point. We make the standard assumption that ℓ (and hence, f) is bounded from below and twice continuously differentiable with respect to $w \in \mathbb{R}^n$ (e.g., logistic, squared hinge, cross-entropy loss). Both the parameter vector dimensionality n and the dataset size q can be large in modern applications.

While first-order methods, such as stochastic gradient descent (SGD) [13] and adaptive schemes like Adam [8], are widely used in large-scale machine learning due to their simplicity and low per-iteration computational cost, they often suffer from slow convergence and sensitivity to tuning of hyperparameters such as learning rate schedules, momentum parameters, and regularization coefficients, particularly in nonconvex settings. Among alternative methods, trust-region (TR) algorithms [3] construct a local model of the objective function at each iteration and solve a constrained subproblem to determine the search direction, guaranteeing global convergence to a first-order critical point [5]. Classical enhancements to TR include two-step variants tailored to structured problems [4].

In this work, building on ideas from multifidelity optimization [6, 11], we introduce two multifidelity trust-region methods inspired by the *magical* trust-region (MTR) framework [3, Section 10.4.1], which augment classical TR steps with an additional “magical” direction aimed at accelerating convergence, see the recent two-level TR method in the same framework [1]. Traditionally, the MTR framework assumes the availability of an oracle that provides enhanced directions, improving upon those obtained from the standard model.

Our first method, called Sketched Trust-Region (STR), constructs the secondary direction by sketching the data matrix X at every iteration, thereby reducing the dimensionality of the trust-region subproblem. In contrast to classical sketched optimization methods that rely entirely on the reduced model [12, 9, 2], STR employs the sketch matrix to only generate a corrective low-fidelity direction that enhances the full-space TR step.

The second method, named SVD Trust-Region (SVDTR), defines the magical direction via a truncated SVD of the data matrix X , retaining the leading t singular vectors to form the feature projector. This captures the dominant directions of variability in the dataset, which is particularly effective when the singular values decay rapidly.

Viewed through the lens of domain decomposition, STR provides an algebraic coarse correction via on-the-fly feature aggregation, whereas SVDTR supplies a spectral coarse space from the dominant singular vectors of X . Here, “coarse” refers to a low-dimensional subspace that captures dominant components of the solution, in analogy to classical coarse spaces in multilevel and domain decomposition methods. The full-space TR step acts as the fine-level update, while the low-fidelity direction plays the role of a coarse-grid correction.

Our goal is to apply these approaches to classification tasks in machine learning, where the balance between cost and accuracy is critical. By incorporating data-driven low-fidelity models into each TR step, we aim to improve the efficiency of training procedures.

2 Magical TR with low-fidelity directions

Following the MTR framework [3], both STR and SVDTR are initialized with an initial guess $w_0 \in \mathbb{R}^n$. At the k -th iteration, the algorithms first compute a high-fidelity step p_k^H by approximately solving the trust-region subproblem

$$\begin{aligned} \min_{p_k^H \in \mathbb{R}^n} m_k^H(p_k^H) &:= f(w_k) + \langle \nabla f(w_k), p_k^H \rangle + \frac{1}{2} \langle p_k^H, \nabla^2 f(w_k) p_k^H \rangle, \\ \text{subject to } \|p_k^H\| &\leq \Delta_k, \end{aligned} \quad (2)$$

where m_k^H is the quadratic model of the full objective around w_k and $\Delta_k > 0$ is the trust-region radius controlling the step size.

The secondary, low-fidelity direction is then computed around the intermediate iterate $w_{k+1/2} := w_k + p_k^H$. Both methods define a low-dimensional objective

$$f_k^L(\tilde{w}) := \frac{1}{q} \sum_{i=1}^q \tilde{\ell}(\tilde{w}; \tilde{x}_i, y_i),$$

where $\tilde{\ell}$ is the same loss function as in (1), now evaluated on the sketched features $\tilde{x}_i = S_k x_i$ and the reduced parameter vector $\tilde{w} \in \mathbb{R}^f$.

- In STR, S_k is a randomized sketching matrix (e.g., Gaussian) that compresses the dataset while approximately preserving its Euclidean geometry, i.e., norms and pairwise distances between data points up to a small distortion [7].
- In SVDTR, S_k is a fixed matrix across iterations (the subscript k is retained for consistency in the algorithm notation), can be pre-computed, and is constructed from the leading left singular vectors of X .

A second-order (Taylor) model m_k^L of f_k^L is built around $\tilde{w}_{k+1/2} = S_k w_{k+1/2}$, and the low-fidelity step $p_k^L \in \mathbb{R}^f$ is computed by solving the trust-region subproblem

$$\begin{aligned} \min_{p_k^L \in \mathbb{R}^f} m_k^L(p_k^L) &:= f_k^L(\tilde{w}_{k+1/2}) + \langle \nabla f_k^L(\tilde{w}_{k+1/2}), p_k^L \rangle + \frac{1}{2} \langle p_k^L, \nabla^2 f_k^L(\tilde{w}_{k+1/2}) p_k^L \rangle, \\ \text{subject to } \|p_k^L\| &\leq \Delta_k. \end{aligned} \quad (3)$$

The reduced step is then lifted to the full space via $S_k^\top p_k^L$ and incorporated into the update only if it decreases the original objective, i.e.,

Algorithm 1 Sketched Trust-Region Method**Input:** $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $w_0 \in \mathbb{R}^n$, $\Delta_0 \in \mathbb{R}^+$, $t < n \in \mathbb{N}$ **Output:** Minimizer w^* of f **Constants:** $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_1 \leq \gamma_2 < 1$

- 1: $k := 0$
 - 2: **while** not converged **do**
 - 3: $p_k^H := \underset{\|p\| \leq \Delta_k}{\operatorname{argmin}} m_k^H(p)$ ▷ Obtain full-space search direction
 - 4: $w_{k+1/2} := w_k + p_k^H$
 - 5: Construct S_k via sketching ▷ For SVDTR: S_k is precomputed
 - 6: $\tilde{X} := S_k X$
 - 7: $p_k^L := \underset{\|\tilde{p}\| \leq \Delta_k}{\operatorname{argmin}} m_k^L(\tilde{p})$ ▷ Obtain subspace search-direction
 - 8: $p_k^L := \begin{cases} p_k^L, & \text{if } f(w_k + p_k^H + \alpha_k S_k^\top p_k^L) < f(w_k + p_k^H) \\ 0, & \text{otherwise} \end{cases}$ ▷ Assess the quality of the subspace step
 - 9: Evaluate ϱ_k as in (4) ▷ Assess the quality of the composite trial step
- $$w_{k+1} := \begin{cases} w_k + p_k^H + \alpha_k S_k^\top p_k^L, & \text{if } \varrho_k > \eta_1, \\ w_k, & \text{otherwise,} \end{cases}$$
- $$\Delta_{k+1} := \begin{cases} [\Delta_k, \infty), & \text{if } \varrho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k], & \text{if } \varrho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k], & \text{if } \varrho_k < \eta_1 \end{cases}$$
- 10: $k := k + 1$
 - 11: **end while**
 - 12: **return** $w^* := w_k$

$$f(w_k + p_k^H + \alpha_k S_k^\top p_k^L) < f(w_k + p_k^H),$$

where $\alpha_k > 0$ may be fixed or chosen via a line search strategy; otherwise we set $p_k^L = 0$. When $p_k^L = 0$, the algorithm reduces to a classical trust-region method. Overall, this ensures global convergence of the method [3].

The effectiveness of the composite step $p_k := p_k^H + \alpha_k S_k^\top p_k^L$ is measured by a trust-region ratio

$$\varrho_k := \frac{f(w_k) - f(w_k + p_k)}{m_k^H(w_k) - m_k^H(w_k + p_k^H) + f(w_k + p_k^H) - f(w_k + p_k)}, \quad (4)$$

which determines step acceptance and trust-region radius updates. Thus, the low-fidelity step may improve acceptance of steps that would be rejected by standard TR, potentially accelerating the objective function decrease. The complete procedure for both methods is summarized in Algorithm 1, where the only difference lies in the construction of the low-fidelity feature projection.

In both methods, the high-fidelity step p_k^H is computed approximately (e.g., via a few Steihaug-Toint CG iterations or using the Cauchy point). The additional com-

putational effort compared to classical trust-region methods arises from two main tasks: constructing the low-fidelity model and solving the corresponding reduced trust-region subproblem.

3 Numerical examples

We evaluate the proposed algorithms, STR and SVDTR, on binary classification problems for the empirical-risk formulation (1), and compare them against the classical full-space TR baseline. We consider two objective functions, namely

$$f_{LL}(w) := \frac{1}{q} \sum_{i=1}^q \log \left(1 + e^{-y_i \langle w, x_i \rangle} \right) + \frac{\lambda}{2} \|w\|_2^2,$$

$$f_{LS}(w) := \frac{1}{q} \sum_{i=1}^q \left(y_i - \frac{e^{\langle w, x_i \rangle}}{1 + e^{\langle w, x_i \rangle}} \right)^2 + \frac{\lambda}{2} \|w\|_2^2,$$

with the number of training samples q and the regularization parameter $\lambda = 1/q$.

Unless stated otherwise, each run is terminated when the Euclidean norm of the full gradient satisfies $\|\nabla f(w)\|_2 \leq 10^{-6}$ or when a predefined problem-dependent iteration budget is reached. We use fixed trust-region parameters $\eta_1 = 0.2$, $\eta_2 = 0.75$, $\gamma_1 = \frac{1}{2}$, and $\gamma_2 = 2$ with initial radius $\Delta_0 = \|\nabla f(w_0)\|_2$. The datasets are drawn from the LIBSVM repository (Australian (621 samples, 14 features), Mushroom (6,499 samples, 112 features), Gisette (6,000 samples, 5,000 features)).¹ The methods are implemented in Python using PyTorch 2.8.0 [10]. All reported results were obtained on Windows 64-bit with an AMD Ryzen 7 5700G CPU (3.80 GHz) and 16 GB RAM (CPU-only).

The high-fidelity TR subproblems in (2) are approximately solved with the Steihaug-Toint conjugate gradient (ST-CG) method with two inner iterations for the Australian and Mushroom datasets. For the Gisette dataset, we either use ST-CG with 25 inner iterations or the Cauchy-point (CP) solver.

For STR and SVDTR, the low-fidelity subproblems in (3), posed in a reduced space of dimension t , are solved by ST-CG with at most t inner iterations, so the reduced directions are accurate in the reduced space and if the projection is of sufficient quality, then they are also good directions once lifted to the full space. In STR, the sketch matrix $S \in \mathbb{R}^{t \times n}$ has i.i.d. entries drawn from $\mathcal{N}(0, t^{-1})$. In SVDTR, the reduced space is the span of the top t left singular vectors of X .

Figure 1 displays the evolution of $\|\nabla f(w_k)\|_2$ as a function of outer iterations. The top/bottom row reports the results for the Australian/Mushroom dataset with the f_{LS}/f_{LL} loss using CP (left) and ST-CG (right). Wall-clock times are not reported, as for these small-scale datasets, all solvers complete within negligible runtime. We present the iteration count to illustrate that the STR and SVDTR can effectively

¹ <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>

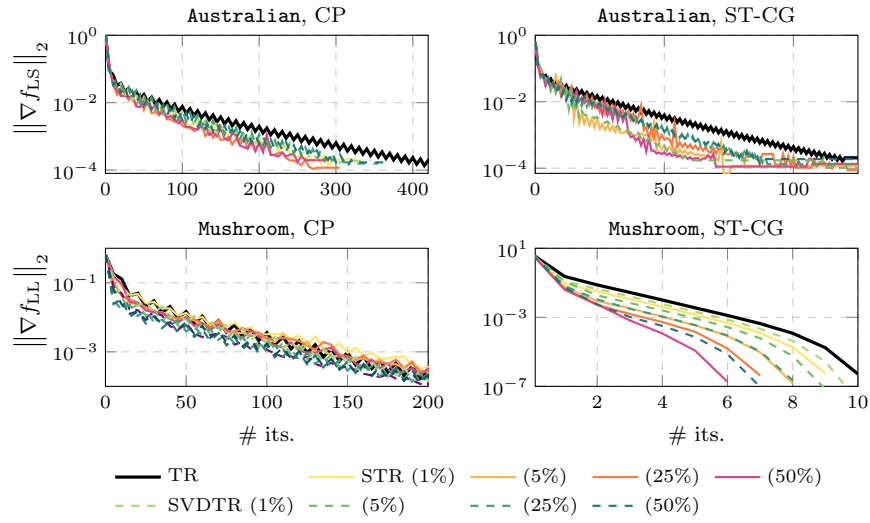


Fig. 1 Convergence histories of TR (solid black), STR (solid), and SVDTR (dashed) for solving (1). Top: Australian with f_{LS} using CP (left) and ST-CG (right). Bottom: Mushroom with f_{LL} under the same full-space solvers. Legend entries for STR/SVDTR indicate the reduced dimension t as a percentage of the feature dimension n .

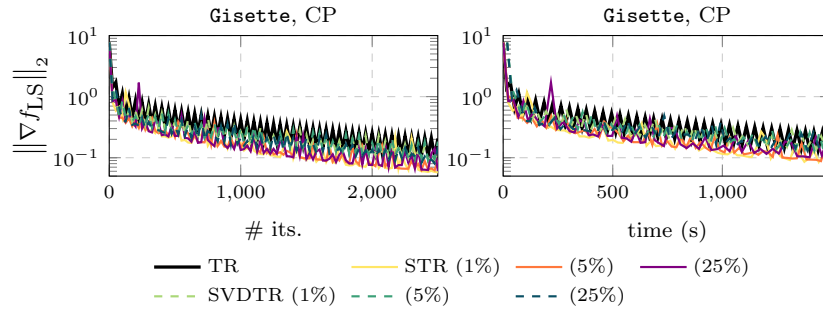


Fig. 2 Convergence histories of TR (solid black), STR (solid), and SVDTR (dashed) for solving (1) with f_{LS} , all using CP. Left: $\|\nabla f_{LL}\|_2$ versus iteration count; right: $\|\nabla f_{LS}\|_2$ versus wall-clock time (s). Legend entries for STR/SVDTR indicate the reduced dimension t as a percentage of n .

reduce the number of iterations compared to TR. Across all configurations, augmenting the full-space step with a reduced-space direction yields a systematic reduction in the number of outer iterations required to attain comparable gradient norms. The improvement exhibits a monotone trend with t and is most pronounced when the full-space subproblems are solved by ST-CG; CP exhibits the same qualitative behavior, albeit with smaller margins. These observations substantiate the effectiveness of the proposed two-direction TR framework in accelerating convergence.

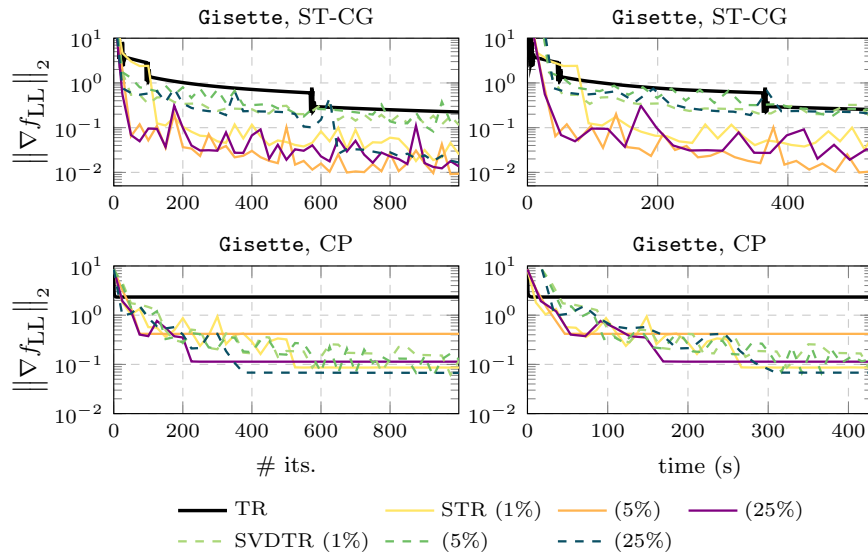


Fig. 3 Convergence histories on the *Gisette* dataset of TR (solid black), STR (solid), and SVDTR (dashed) for solving (1) with f_{LL} . Top: ST-CG full-space solver. Bottom: CP full-space solver. Legend entries for STR/SVDTR indicate the reduced dimension t as a percentage of n .

We proceed by testing our methods on the high-dimensional *Gisette* dataset. Figures 2 and 3 report the decay of the full-gradient norm $\|\nabla f(w_k)\|_2$ versus outer iterations and wall-clock time in seconds for TR, STR, and SVDTR at multiple reduced dimensions t . In all cases, augmenting the full-space step with a reduced-space direction markedly lowers the iteration counts relative to TR, with a monotone trend as t increases. Overall, SVDTR tends to excel for f_{LL} once the reduced dimension t is sufficiently large for the subspace to capture the dataset structure, whereas STR provides robust preprocessing-free improvements.

Code and data availability

The code and data used to produce the numerical results is available at

<https://doi.org/10.5281/zenodo.17473878>.

Acknowledgements The work of A.K. benefited from the AI Interdisciplinary Institute ANITI, funded by the France 2030 program under Grant Agreement No. ANR-23-IACL-0002. The research of A.A. and M.V. was funded in part by the Swiss National Science Foundation (SNSF) grant No. 224943.

References

1. Angino, A., Kopaničáková, A., Krause, R.: Two-level trust-region method with random subspaces. In: P. Bjorstad, X.C. Cai, V. Dolean, D. Keyes, R. Kornhuber, J. Xu (eds.) *Domain Decomposition Methods in Science and Engineering XXVIII*, Lect. Notes Comput. Sci. Eng. Springer, Cham, Switzerland (2025). Lect. Notes Comput. Sci. Eng. Springer, Cham (2025). Available as arXiv preprint arXiv:2409.05479
2. Cartis, C., Fiala, J., Shao, Z.: Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares. arXiv preprint arXiv:2211.09873 (2022)
3. Conn, A.R., Gould, N.I.M., Toint, P.L.: *Trust Region Methods*. MOS-SIAM Ser. Optim. SIAM, Philadelphia, PA, USA (2000)
4. Conn, A.R., Vicente, L.N., Visweswariah, C.A.R., Gould, N.I.M., Toint, P.L.: Two-step algorithms for nonlinear optimization with structured applications. *SIAM J. Optim.* **9**(4), 924–947 (1999)
5. Curtis, F.E., Robinson, D.P., Samadi, M.: A trust-region algorithm with a worst-case iteration complexity of $O(\epsilon^{-3/2})$ for nonconvex optimization. *Math. Program.* **162**(1), 1–32 (2017)
6. Forrester, A.I.J., Sóbester, A., Keane, A.J.: Multi-fidelity optimization via surrogate modelling. *Proc. Roy. Soc. A* **463**, 3251–3269 (2007)
7. Johnson, W.B., Lindenstrauss, J., Schechtman, G.: Extensions of Lipschitz maps into Banach spaces. *Israel J. Math.* **54**, 129–138 (1986)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kozak, D., Becker, S., Doostan, A., Tenorio, L.: A stochastic subspace approach to gradient-free optimization in high dimensions. *Comput. Optim. Appl.* **79**, 339–368 (2021)
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: an imperative style, high-performance deep learning library. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (2019)
11. Peherstorfer, B., Willcox, K., Gunzburger, M.: Survey of multi-fidelity methods in uncertainty propagation, inference, and optimization. *SIAM Rev.* **60**(3), 550–591 (2018)
12. Pilanci, M., Wainwright, M.J.: Newton Sketch: A near linear-time optimization algorithm with linear–quadratic convergence. *SIAM J. Optim.* **27**(1), 205–245 (2017)
13. Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**(3), 400–407 (1951)