L-SR1:

Learned Symmetric-Rank-One Preconditioning

Gal Lifshitz Shahar Zuler Ori Fouks Dan Raviv

Tel Aviv University

Correspondence: **lifshitz@mail.tau.ac.il**

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

ICML International Conference On Machine Learning

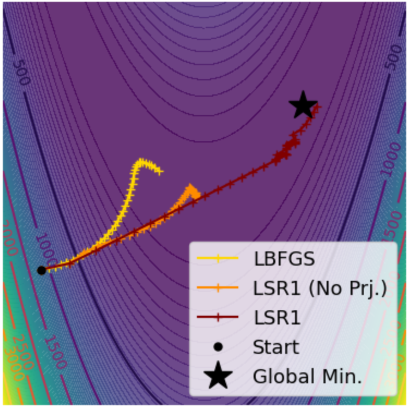
gallif.github.io/lsr1

Motivation: why a learned second-order optimizer?

- **End-to-end deep learning** is powerful, but data-hungry, heavy, and generalizes poorly.
- **Classical optimization** is lightweight and data-efficient, but **slow** to converge.
- **Learned optimizers** bridge the gap — yet they are almost all **first-order**.
- Learned **second**-order optimization (curvature-aware) is **largely unexplored**.

Our goal

A general, curvature-aware learned optimizer that needs **fewer iterations** than strong classical and learned baselines — on analytic benchmarks **and** real world tasks.



Optimizer trajectories on the Rosenbrock landscape.

Background: learned optimization

Idea. Replace the hand-designed update rule with a learned map

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \varphi_\Theta\big(\nabla f(\mathbf{x}_k), \mathbf{x}_k, \dots\big),$$

where the parameters Θ are shared across all optimization steps.

Meta-training. Learn Θ by unrolling the optimizer for K steps and minimizing a meta-loss

$$\mathcal{L}_{\text{meta}} = \sum_{k=1}^{K} f(\mathbf{x}_k).$$

The gap we target

Learned optimizers can outperform classical methods in iteration count, yet most remain **first-order**. Our results show that **curvature-aware learned optimization** yields faster convergence and stronger generalization.

Background: SR1 & quasi-Newton methods

Quasi-Newton step uses a preconditioner $\mathbf{B}_k \approx$ inverse Hessian:

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \alpha_k \mathbf{B}_k \mathbf{g}_k, \qquad \mathbf{g}_k = \nabla f(\mathbf{x}_k).$$

All QN methods enforce the **secant condition** (local curvature), with $\mathbf{p}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$, $\mathbf{q}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$:

$$\mathbf{B}_k \mathbf{q}_k = \mathbf{p}_k.$$

SR1 updates $\mathbf{B}$ with a single rank-one term — symmetric and limited-memory friendly:

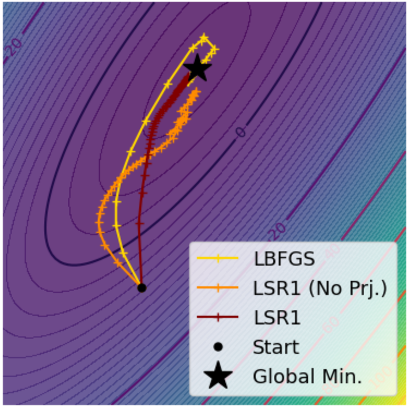
$$\mathbf{B}_k \leftarrow \mathbf{B}_{k-1} + \mathbf{u}_k \mathbf{v}_k^\top.$$

The catch

SR1 satisfies the secant equation but is **not guaranteed positive semi-definite (PSD)** — so steps can be **non-descent** directions and unstable. PSD $\Rightarrow$ guaranteed descent.

L-SR1: method highlights

- **Learned second-order optimization:** lightweight neural modules embedded within a principled SR1-inspired quasi-Newton update.
- **Guaranteed descent and curvature consistency:** learns a PSD preconditioner $\tilde{\mathbf{B}}_k \succeq \mathbf{0}$ via PGSM.
- **Practical and scalable:** dimension-invariant, limited-memory ($O(LN)$ storage), and self-supervised with only **10.4M** parameters.



Curvature-aware steps drive L-SR1 toward the optimum in few iterations.

PGSM: Projection-Guided Secant Mechanism

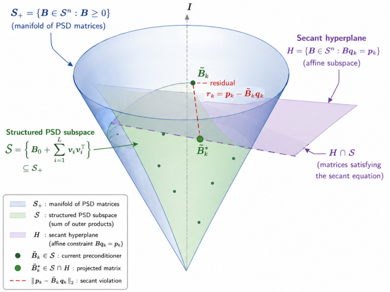
Challenge. Stay PSD **and** respect curvature (the secant relation).

1. PSD by construction — a sum of rank-one outer products:

$$\tilde{\mathbf{B}}_k = \mathbf{I} + \sum_{i=1}^{L} \mathbf{v}_i \mathbf{v}_i^\top \succeq \mathbf{0}.$$

2. Secant guidance — penalize secant mismatch in the **meta-loss**:

$$\mathcal{L}_{\text{meta}} = \frac{1}{K} \sum_{k=1}^{K} \Big(f(\mathbf{x}_k) + \lambda_{\text{sec}} \, \|\mathbf{p}_k - \tilde{\mathbf{B}}_k \mathbf{q}_k\|_2^2 \Big).$$



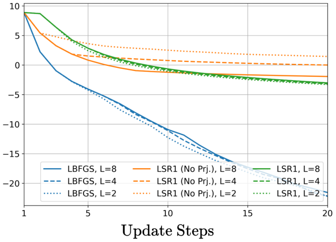
PSD projection guided toward the secant relation.

Key idea

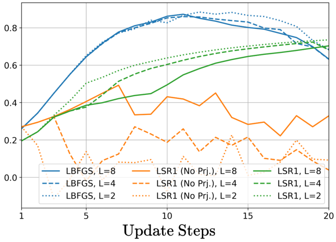
PSD preconditioners **for free**, with curvature consistency learned during training — and **no projection subproblem at inference**.

Results I — analytic functions

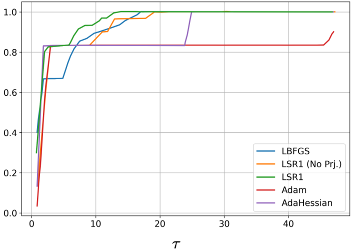
Objective vs. iter.



Newton alignment



Performance profiles



Quadratic functions

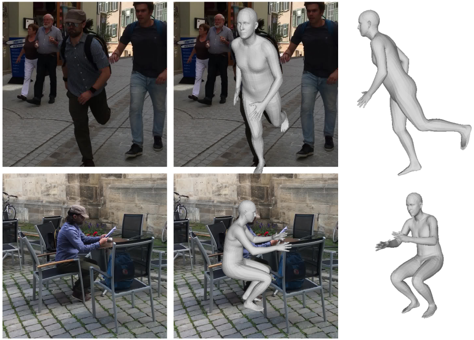
- Train on random quadratics; test on higher-dimensional ones
- PGSM aligns updates with the **Newton direction**
- Generalizes to higher dimensions without retraining

Performance profiles

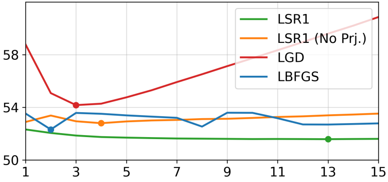
- Evaluated across varying-dimensional objective functions
- Compared against no-PGSM, L-BFGS, Adam, AdaHessian
- Achieves **best performance profile**

Results II — a real-world task: Human Mesh Recovery

- **Task:** Recover a 3D human mesh from a single RGB image (ill-posed inverse problem).
- **Setup:** Trained on AMASS; evaluated on in-the-wild 3DPW **without fine-tuning**.
- **Accuracy:** Achieves the **lowest reconstruction error** vs. strong optimization-based baselines.
- **Optimization behavior:** Shows **faster convergence** in inner-loop refinement.



Input → L-SR1 overlay → recovered mesh



PA-MPJPE vs. iteration

Method	Error
LGD (4 steps)	55.90
L-SR1 (best)	**51.58**

Thank you!

L-SR1: Learned Symmetric-Rank-One Preconditioning

Project page: **gallif.github.io/lsr1**

Correspondence: **lifshitz@mail.tau.ac.il**

Project page