Connecting Parameter Magnitudes and Hessian Eigenspaces at Scale using Sketched Methods

Andres Fernandez, Frank Schneider, Maren Mahreseci, Philipp Hennig
{a.fernandez, f.schneider, maren.mahsereci, philipp.hennig}@uni-tuebingen.de

May 9, 2025

Faculty of Science
Department of Computer Science
Methods of Machine Learning

Early crystallization in deep learning

Setup:

- **Dataset** of pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$
- **Neural Network** $\hat{\boldsymbol{y}}_i = f(\boldsymbol{x}_i, \boldsymbol{\theta})$ with **parameters** $\boldsymbol{\theta} \in \mathbb{R}^D$
- **Loss** $\mathcal{L}(\boldsymbol{\theta}) = \sum_i \ell(\boldsymbol{y}_i, f(\boldsymbol{x}_i, \boldsymbol{\theta})) \in \mathbb{R}_{\geq 0}$ with **gradient** $\boldsymbol{g} \in \mathbb{R}^D$ and **Hessian** $\boldsymbol{H} \in \mathbb{R}^{D \times D}$

Early crystallization of **parameters**:

- Parameters can be *pruned* (Blalock et al. 2020)
- Pruning masks $\boldsymbol{\theta} \odot \boldsymbol{m}$ appear *early* in training (Frankle et al. 2019)
- Magnitude pruning masks *don't change much* during training! (You et al. 2020)

Early crystallization of **loss landscape**:

- $\boldsymbol{H}$ is rank-defficient, i.e. $\boldsymbol{H} \approx \boldsymbol{U}_{top} \boldsymbol{\Lambda}_{top} \boldsymbol{U}_{top}^\top$ (e.g. Sagun et al. 2018)
- $\boldsymbol{g}$ resides mostly in $\boldsymbol{U}_{top}$ (Gur-Ari et al. 2019)
- span($\boldsymbol{U}_{top}$) *doesn't change much* during training! (Gur-Ari et al. 2019)

Are those connected?

Research questions and contributions

Questions:

- Can this similarity be measured? If so, how?
- What similarity can be considered high? What are the implications?

Contributions:

- Methodology to compare arbitrary k-parameter masks to *top-k* Hessian eigenspaces
- Algorithm and code to perform said measurements at scale $\rightarrow$ Hessian eigendecompositions
- **In DL, connection is orders of magnitude larger than random**
- Potential implications for pruning, optimization, UQ and loss landscape analysis

Comparing parameters with Hessian subspaces

Top-k parameter pruning is a projection onto $\boldsymbol{I}_{D,k}$:

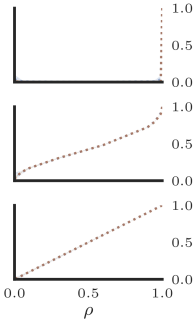
$$\boldsymbol{P}^\top(\boldsymbol{m}_k \odot \boldsymbol{\theta}) = \tilde{\boldsymbol{m}}_k \odot \tilde{\boldsymbol{\theta}} = \begin{pmatrix} \boldsymbol{I}_k & 0 \\ 0 & 0 \end{pmatrix} \tilde{\boldsymbol{\theta}} =: \boldsymbol{I}_{D,k}\boldsymbol{I}_{D,k}^\top \tilde{\boldsymbol{\theta}}$$

Also recall the *top-k* eigenbasis $\boldsymbol{U}_{top}$:

$$\boldsymbol{H} = \left(\begin{array}{c:c} \boldsymbol{U}_{top} & \boldsymbol{U}_{bulk} \end{array} \right) \left(\begin{array}{c:c} \boldsymbol{D}_{top} & \\ \hdashline & \boldsymbol{D}_{bulk} \approx 0 \end{array} \right) \left(\begin{array}{c} \boldsymbol{U}_{top}^\top \\ \hdashline \boldsymbol{U}_{bulk}^\top \end{array} \right)$$

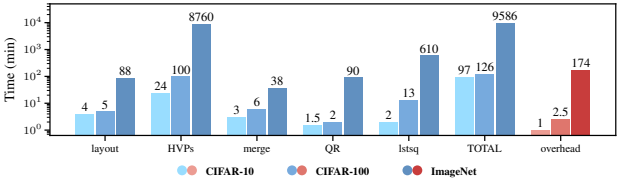
- We have **same-shape, orthogonal matrices** $\boldsymbol{I}_{D,k}$ and $\boldsymbol{U}_{top}$
- **Grassmannian metrics** measure the distance between their **spaces**
- Theoretical and empirical analysis of several Grassmannian metrics
- The **overlap** metric is stable and has a **random baseline value** of $\frac{k}{D}$:

$$\frac{1}{k}\|\boldsymbol{I}_{D,k}^\top \boldsymbol{U}_{top}\|_F^2 \in [0,1] \quad \text{(higher} \iff \text{more similar)}$$



Computing the EIGH via sketched methods

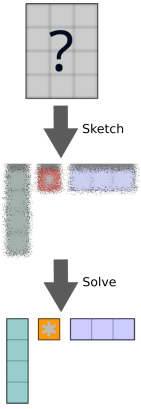
- Computing *overlap* **requires top-k Hessian eigendecomposition** $\frac{1}{k}\|\boldsymbol{I}_{D,k}^\top \boldsymbol{U}_{top}\|_F^2$
- **Intractable:** $\mathcal{O}(D^2)$ memory, $\mathcal{O}(D^3)$ arithmetic (Golub et al. 2013)
- **Expensive measurements:** Each $\boldsymbol{w} = \boldsymbol{Hv}$ costs 2 forward+backpropagations (Pearlmutter 1994)
- Sketched methods: $\mathcal{O}(k)$ *parallel* measurements, $\mathcal{O}(Dk)$ memory (Halko et al. 2011)
- **PyTorch library**: `skerch`



`pip install skerch`

Sketched SVD: Intuition

- Consider a LinOp, **very large and matrix-free**, with **expensive measurements** but some **simpler sub-structure** (low-rank in this case)
- We'll see how to **sketch-and-solve**:
  - Draw a few random measurements *in a way that captures the sub-structure*
  - Project measurements back into original space
- We will cover:
  - *Sketching* step: random measurements with guarantees
  - *Solving* step: cheap, traiditonal linear algebra
  - How this may help **approximating full DL Hessians**



The Randomized Range Finder (RRF) (Halko et al. 2011)

Let's revisit our low-rank operator $\boldsymbol{A} \in^{m \times n}$:

$$\boldsymbol{A} := \left(\begin{array}{c:c} \boldsymbol{U} & \bar{\boldsymbol{U}} \end{array} \right) \left(\begin{array}{c:c} \boldsymbol{D} & \\ \hdashline & \bar{\boldsymbol{E}} \end{array} \right) \left(\begin{array}{c} \boldsymbol{V}^* \\ \hdashline \bar{\boldsymbol{V}}^* \end{array} \right) \approx \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^*$$

Consider $\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{V}}$ as *rotations* of $\boldsymbol{U}, \boldsymbol{V}$, such that $\tilde{\boldsymbol{U}} := \boldsymbol{U}\boldsymbol{Z}$ for unitary $\boldsymbol{Z} \in^{k \times k}$. The key observation is that:

$$\boldsymbol{A} \approx \boldsymbol{U}\boldsymbol{U}^*\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^*\boldsymbol{A}\tilde{\boldsymbol{V}}\tilde{\boldsymbol{V}}^*$$

And the "magic" is that **$\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{V}}$ can be obtained from $\mathcal{O}(k)$ random measurements!** (Halko et al. 2011). Intuition:

- If we draw $\{\boldsymbol{A}\boldsymbol{\omega}_1, \boldsymbol{A}\boldsymbol{\omega}_2, \dots\}$ measurements, they are all likely to be **linearly independent**
- Furthermore, if they are random, they all likely to **land on the top subspace**
- The two conditions above mean that we are **fully covering the top subspace**
- QR orthogonalization of our measurements yields $\tilde{\boldsymbol{U}}$
- Measurements from quasi-orthogonal iid noise are **numerically stable** and **parallelizable**

Sketched SVD (Halko et al. 2011; Tropp et al. 2017)

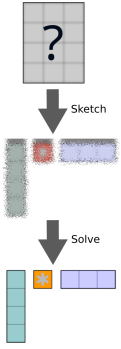
Naively, all is left is to perform a $k \times k$ SVD:

$$\boldsymbol{A} \approx \tilde{\boldsymbol{U}} \underbrace{\tilde{\boldsymbol{U}}^* \boldsymbol{A} \tilde{\boldsymbol{V}}}_{\boldsymbol{C} = \boldsymbol{Z}_1 \boldsymbol{\Sigma} \boldsymbol{Z}_2^T} \tilde{\boldsymbol{V}}^*$$

Nice! But this requires us to do a **second, expensive pass** over $\boldsymbol{A}\tilde{\boldsymbol{V}}$, which is **unnecesary!**

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\Omega} = \boldsymbol{A}\tilde{\boldsymbol{V}}\tilde{\boldsymbol{V}}^*\boldsymbol{\Omega}$$

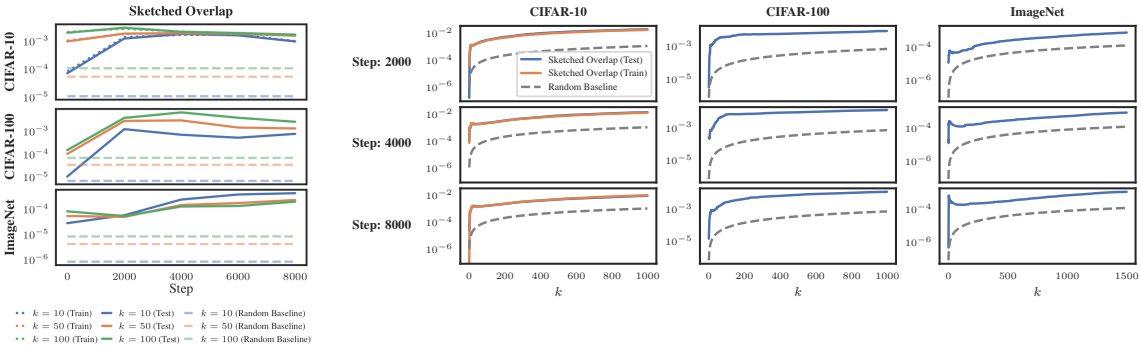
$$\iff \underbrace{\boldsymbol{A}\tilde{\boldsymbol{V}}}_{\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{Z}^*} = \boldsymbol{Y}(\tilde{\boldsymbol{V}}^*\boldsymbol{\Omega})^\dagger \iff \boldsymbol{A} = \boldsymbol{A}\tilde{\boldsymbol{V}}\tilde{\boldsymbol{V}}^* = \boldsymbol{U}\boldsymbol{\Sigma}\underbrace{(\boldsymbol{Z}^*\tilde{\boldsymbol{V}}^*)}_{\boldsymbol{V}^*}$$



This is very good!

- First we obtain *thin outer matrices* $\tilde{\boldsymbol{U}} \in^{m \times \alpha}, \tilde{\boldsymbol{V}} \in^{n \times \alpha}$ from random measurements and QR
- Then, decompose a *small inner matrix* $\tilde{\boldsymbol{C}} := \boldsymbol{\Phi}^* \boldsymbol{A} \boldsymbol{\Psi} \in^{\beta \times \beta}$ and regroup matrices
- For **single-pass**, solve a well-conditioned least squares problem

High overlap in DL



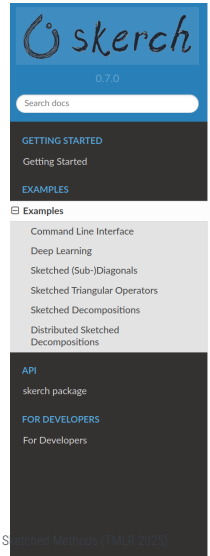
- **Scalable**: Rank-1500 eigendecompositions on 12M-parameter networks
- **Orders of magnitude higher for all observed splits, steps, rank sizes and problems**
- Parameter inspection cheaply informs about curvature → training, pruning, UQ, analysis
- Still, spaces are far from identical ($\frac{k}{D}$ is small), so no direct mapping

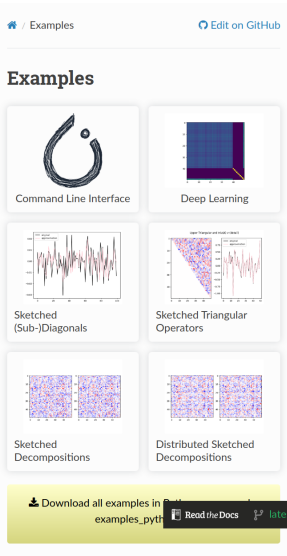
skerch: Sketched matrix decompositions for PyTorch

"I love it, 5/5!" — Diederik P. Kingma, probably talking about the Adam optimizer

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

https://github.com/andres-fr/skerch





Thank you!

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Conclusions:

- Grassmannian metrics to compare arbitrary parameters and Hessian eigenspaces
- Sketched eigendecompositions to measure *overlap* at scale $\rightarrow$ `skerch`
- DL *overlap* orders-of-magnitude larger than baseline (albeit far from identical)
- Connecting expensive Hessian quantities with cheap parameter observations

Future work:

- Scalability: We also explore faster alternatives like perturbation-based and GGN
- Explaining why do we observe high overlap
- Leveraging this effect in downstream applications

How can this be useful to you?

- Insights on comparing spaces of seemingly unrelated quantities
- Scalable linear algebra
- DL Hessians: Optimization, pruning, UQ, learning theory

References I

Blalock, Davis et al. (2020). "What is the State of Neural Network Pruning?" In: *Proceedings of Machine Learning and Systems (MLSys)*.

Candès, Emmanuel J. (2008). "The restricted isometry property and its implications for compressed sensing". In: *Comptes Rendus Mathematique* 346.9, pp. 589–592. ISSN: 1631-073X. DOI: https://doi.org/10.1016/j.crma.2008.03.014. URL: https://www.sciencedirect.com/science/article/pii/S1631073X08000964.

Frankle, Jonathan and Michael Carbin (2019). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJl-b3RcF7.

Golub, Gene H. and Charles F. Van Loan (2013). *Matrix Computations*. Fourth. The Johns Hopkins University Press.

Gur-Ari, Guy, Daniel A. Roberts, and Ethan Dyer (2019). *Gradient Descent Happens in a Tiny Subspace*. URL: https://openreview.net/forum?id=ByeTHsAqtX.

Halko, N., P. G. Martinsson, and J. A. Tropp (2011). "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In.

Hardt, Moritz, Benjamin Recht, and Yoram Singer (2016). "Train Faster, Generalize Better: Stability of Stochastic Gradient Descent". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 1225–1234.

Pearlmutter, Barak A. (1994). "Fast Exact Multiplication by the Hessian". In.

Sagun, Levent et al. (2018). *Empirical Analysis of the Hessian of Over-Parametrized Neural Networks*. ICLR 2018 Workshop. URL: https://openreview.net/forum?id=rJrTwxbCb.

Tropp, Joel A. et al. (2017). "Practical Sketching Algorithms for Low-Rank Matrix Approximation". In: *SIAM Journal on Matrix Analysis and Applications*.

— (2019). "Streaming Low-Rank Matrix Approximation with an Application to Scientific Simulation". In: *SIAM Journal on Scientific Computing*.

You, Haoran et al. (2020). "Drawing Early-Bird Tickets: Toward More Efficient Training of Deep Networks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=BJxsrgStvr.

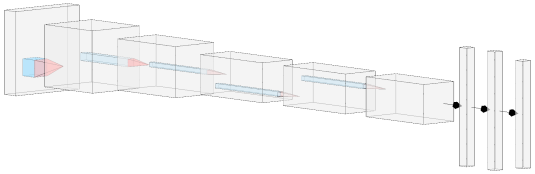
Backup

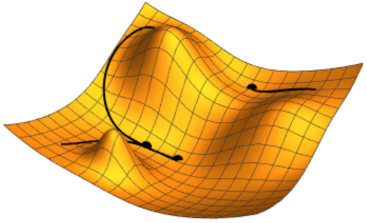
Deep Learning and The Hessian

A tale as old as Geoffrey Hinton

images from WikiMedia

- Dataset $\mathcal{D} := \{(x_d, y_d)\}_{n=1}^{D} \in (\mathcal{X} \times \mathcal{Y})^D$, model $f_\theta : \mathcal{X} \ni x_n \mapsto \hat{y}_n \in \mathcal{Y}$ and loss function $l(y_d, \hat{y}_d) : (\mathcal{Y} \times \mathcal{Y}) \mapsto \mathbb{R}_{\geq 0}$
- Local loss landscape: $\mathcal{L}(\theta+\delta) \approx \hat{\mathcal{L}}(\theta+\delta) = \frac{1}{2}\delta^\top H_\theta \delta + \nabla_\theta^\top \delta + \mathcal{L}(\theta)$
- Empirical risk minimization on training data leads to update $\theta_{t+1} = \theta_t - H_{\theta_t}^{-1} \nabla_{\theta_t}$





Often, D and $dim(\theta) = N$ are very large $\rightarrow$ exact updates unfeasible!

Speedup techniques also generalize well (Hardt et al. 2016), but optimizers often present **slow convergence** and **hyperparameter instability**.

Fernandez, Schneider, Mahsereci, Hennig – Connecting Parameter Magnitudes and Hessian Eigenspaces at Scale using Sketched Methods (TMLR 2025)

Note: Efficient and stable measurements

Recall we want to do measurements $\boldsymbol{A}\boldsymbol{\Upsilon}$ with random $\boldsymbol{\Upsilon}$. If we e.g. draw Gaussian noise, *this is prohibitively slow and inefficient*! And what we *really* want is $\boldsymbol{\Upsilon}$ to be *uncorrelated* with $\boldsymbol{A}$, but ideally also *orthogonal* to ensure numerical stability (Halko et al. 2011, p. 6.2). This is well characterized via the *Restricted Isometry Property* (Candès 2008).

Luckily, there is a fast, competitive, stable and matrix-free way of doing random measurements! The Scrambled Subsampled Randomized Fourier Transform (SSRFT) (Tropp et al. 2019, p. 3.2), related to the *Fast Johnson-Lindenstrauss Transform (FJLT)*, achieves exactly this:

SSRFT: $\mathcal{R}\mathcal{F}\Pi\mathcal{F}\Pi'$

Here, $\mathcal{F}$ is the Fourier transform, Π are signed random permutations and $\mathcal{R}$ is a random index picker. Furthermore:

- It is matrix-free and leverages the FFT, requiring only $\mathcal{O}(n)$ memory and $\mathcal{O}(n \log n)$ time
- It is composed by unitary projections, hence it is unitary
- As a bonus, the adjoint operation is also trivial
- Empirically, it has been shown to outperform Gaussian noise

Matrix-Freedom

The *forced* ammendment

- **Double-edged sword:** Arbitrarily large, but become implicit
- Satisfy linearity via mat-vec products:
  $\boldsymbol{A}(c_1\boldsymbol{v_1} + c_2\boldsymbol{v_2}) = c_1\boldsymbol{A}\boldsymbol{v_1} + c_2\boldsymbol{A}\boldsymbol{v_2}$
- Examples: differentiation, Laplace, Fourier, convolution…
- **This restricts the type of algorithms and analysis available**

```
# here is a matrix, and a matrix-free Hessian
mat = torch.randn(5, 5)
H = HessianLinearOperator(
        dnn, loss_fn, params, dataloader)

# both support matrix-vector multiplications
w1 = mat @ v1
w2 = H @ v2

# and both are linear
w12 = H @ (a*v1 + b*v2)
assert w12 == a*(H @ v1) + b*(H @ v2)

# But H is not explicit in memory!
x1 = mat[3, 4:]  # good
x2 = H[3, 4:]  # not supported!

# instead, we have to run an HVP:
e3 = one_hot_vector(idx=3)
x2 = (e3 @ H)[4:]
```