Determinant Estimation under Memory Constraints and Neural Scaling Laws

Siavash Ameli[1, 2], Chris van der Heide[3], Liam Hodgkinson[3], Fred Roosta[4], Michael W. Mahoney[1,2,5]
[1]UC Berkeley, [2]ICSI, [3]University of Melbourne, [4]University of Queensland, [5]LBNL

Motivation

Log-determinants: widely encountered in linear algebra and statistics, often **the most difficult term** to compute in applications: Gaussian processes (kernel methods), determinantal point processes, rank estimation, regularization, robust PCA...

Neural Tangent Kernel (NTK) is our motivating example:

- Neural network $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}^d$, with parameters $\boldsymbol{\theta}$ and Jacobian $\mathbf{J}_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(x))$
- (Empirical) NTK is the Gramian of $\mathbf{J}_{\boldsymbol{\theta}}$: $\kappa_{\boldsymbol{\theta}}(x, x') \coloneqq \mathbf{J}_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(x))\mathbf{J}_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(x'))^{\intercal}$

Challenge I. Forming NTK

- Prohibitively expensive: ResNet101 w/ CIFAR-10 is **50 days on H100**; ImageNet1K needs **13,131,111 Terabytes** to store
- **Precision loss** when forming NTK: **positive-definiteness** requires **double precision**

Challenge II. Computing LogDet

- **Cubic** complexity $\mathcal{O}(n^3 d^3)$, but this requires the matrix to be loaded into **memory**
- NTK is nearly **singular:** condition number $\sim 10^{16}$ to 10^{30} in our experiments, with 10% of eigenvalues near zero

Exact Computation with MEMDET

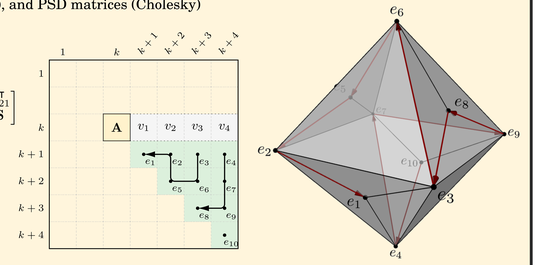
- Out-of-core algorithm **eliminates the memory wall** for matrices of **any scale**
- Leverages hierarchical block decompositions within Schur complements
- Obtains log-determinants for **all nested submatrices**
- Versions for general (LU), symmetric (LDL), and PSD matrices (Cholesky)

Block decompositions:

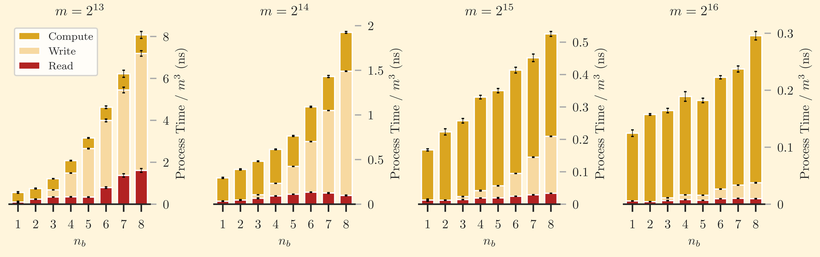
- Example: LU decomposition becomes

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{11}^{\intercal} & \mathbf{L}_{21}^{\intercal} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}$$

- Repeat decomposition on block S.
- Only **four blocks** required in memory
- Others written to **scratch space**
- **Optimal block processing** traverses a Hamiltonian path of the line graph

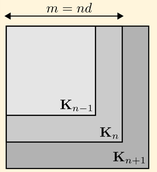


MEMDET Performance



- **Compute time** remains **consistent** across varying number of blocks.
- For small matrices, most of the process time is spent reading and writing to the scratch space.
- For large matrices, **data transfer time** becomes **negligible** compared to compute.

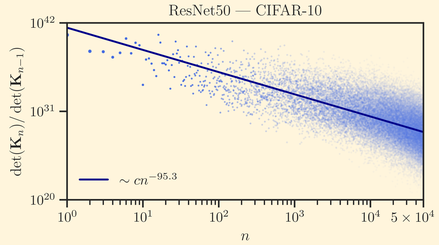
Neural Scaling Law



$$\frac{\det(\mathbf{K}_n)}{\det(\mathbf{K}_{n-1})} \sim n^{\nu}$$

- n: examples in dataset
- d: number of classes
- $m = nd$: matrix size

Lemma 1. *Let $f : \mathcal{X} \to \mathbb{R}^d$ be a zero-mean Gaussian process with kernel κ. For each $n \geq 2$, let $E(n) \coloneqq \mathbb{E}[d^{-\frac{1}{2}}\|f(x_n)\|^2 \mid f(x_i) = 0 \text{ for } i < n]$, denote the MSE of fitting f to the zero function. Then* $\frac{\mathrm{pdet}(\mathbf{K}_n)}{\mathrm{pdet}(\mathbf{K}_{n-1})} \leq E(n)^d, \quad \forall n > 1,$ *with equality if $d = 1$.*



- NTK of ResNet50 on CIFAR-10 (**largest ever computed**)
- Dataset: $n = 50$K images with $d = 10$ classes
- Matrix size: $m = 500$K, requires **2TB at 64-bit**

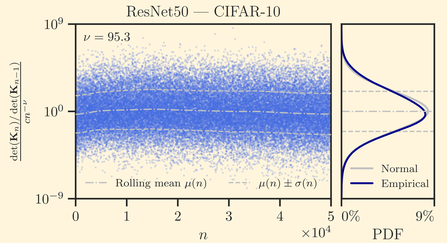
Log-Determinants as a Stochastic Process

Proposition 2. *Let $L_n \coloneqq \frac{1}{n}\mathrm{logdet}(\mathbf{K}_n)$. Then*

$$\hat{L}_n \approx L_1 + \left(1 - \frac{1}{n}\right) c_0 - \nu \frac{\log(n!)}{n}.$$

- **Law of large numbers**: $L_n = \hat{L}_n + o_p(1)$.
- **CLT**: $\frac{n}{\sqrt{n-1}}(L_n - \hat{L}_n) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma^2)$.

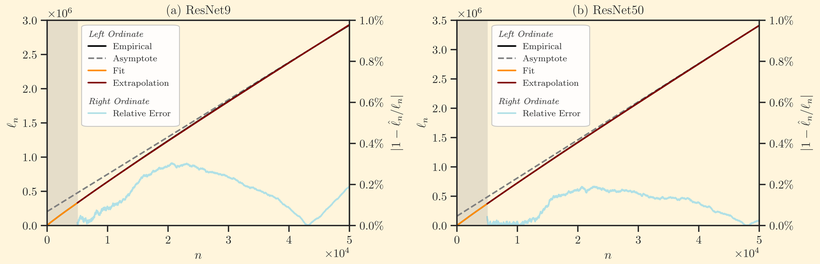
Assumptions: The log of the stochastic process $\frac{\det(\mathbf{K}_n)\,/\,\det(\mathbf{K}_{n-1})}{cn^{\nu}}$ is **stationary** and **ergodic**.



FLODANCE:

- **Fit** $\hat{L}_n$ for $n = 1, \ldots, n_s \ll n$
- **Extrapolate** to larger $n \gg n_s$

Estimating NTK Log-Dets with FLODANCE



- **Fit:** FLODANCE trained on 10% of total matrix (shaded gray region, yellow curve)
- **Extrapolation:** in much larger interval (red curve)
- MEMDET baseline (black curve) is indistinguishable from our approximations
- Error relative to MEMDET: (blue curve right axis in each panel), **0.2%** (left), **0.02%** (right).

Comparison with the Current State of the Art

Method					
Name	Settings	TFLOPs	Rel. Error	Est. Cost	Wall Time
SLQ	$l = 100$, $s = 104$	5203	55%	$83	1.8 days
MEMDET	LDL, $n_b = 32$	41,667	**0%**	$601	13.8 days
FLODANCE	$n_s = 500$, $q = 0$	**0.04**	4%	$0.04	1 min
FLODANCE	$n_s = 5000$, $q = 4$	41.7	**0.02%**	$4	1.5 hr

✘ SLQ slow, expensive, and inaccurate
✓ MEMDET enables exact benchmarking
✓ FLODANCE fast, cheap, and accurate

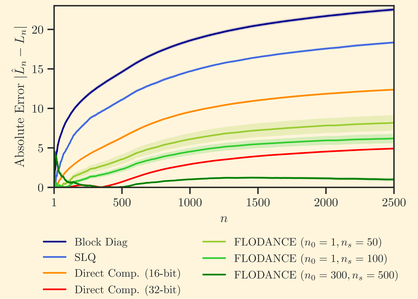
FLODANCE Beats Exact Methods

Experiment:

- ResNet9 with CIFAR-10
- Smaller matrices to compare with other methods
- Uncertainty quantification: submatrix samples
- Shaded region: standard deviation
- Benchamrk: MEMDET in double precision

Results:

- FLODANCE gives best approximation
- FLODANCE trained on 64-bit data is comparable to 32-bit exact computation



Resources

References: Ameli, S., van der Heide, C., Hodgkinson, L., Roosta, F., Mahoney, M.W., (2025). *Determinant Estimation under Memory Constraints and Neural Scaling Laws*, The 42nd International Conference on Machine Learning.

Related Work: Ameli, S., van der Heide, C., Hodgkinson, L., Mahoney, M.W., (2025). *Spectral Estimation with Free Decompression*, arXiv:2506.11994.

Software:

Package	Documentation	Install	Implements
detkit	`ameli.github.io/detkit`	`pip install detkit`	MEMDET FLODANCE
imate	`ameli.github.io/imate`	`pip install imate`	SLQ
freealg	`ameli.github.io/freealg`	`pip install freealg`	(Related work)