Kernel-Ridge-Regression

Core idea

Procedure

- Get *i* data points with scalar property (label) $\{q_i\}$
  - E.g. atomisation energy
- Calculate all representations $\{\mathbf{M}_i\}$
  - typically ~1k
- Find distance and kernel matrices $\mathbf{D}, \mathbf{K}$
  - Symmetric
- Train model for predictions $\{\tilde{q}_i\}$

$$\arg\min_{\alpha} \sum_i (q_i - \tilde{q}_i)^2 + \lambda \sum_{ij} \alpha_i \alpha_j k_{ij}$$

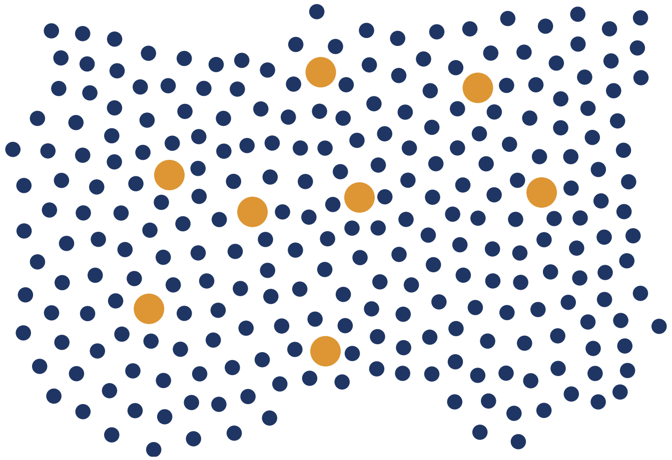
$$\Rightarrow \alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathrm{y} \qquad \tilde{q}(\mathbf{M}) = \sum_i \alpha_i k(\mathbf{M}, \mathbf{M}_i)$$

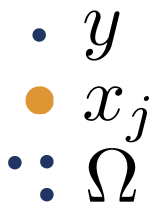
Gaussian kernel: $k(x, y) = \exp(-\gamma ||x - y||^2)$

Fill Distance

$$h \equiv \sup_{y \in \Omega} \min_{x_j \in X} \|y - x_j\|_2$$

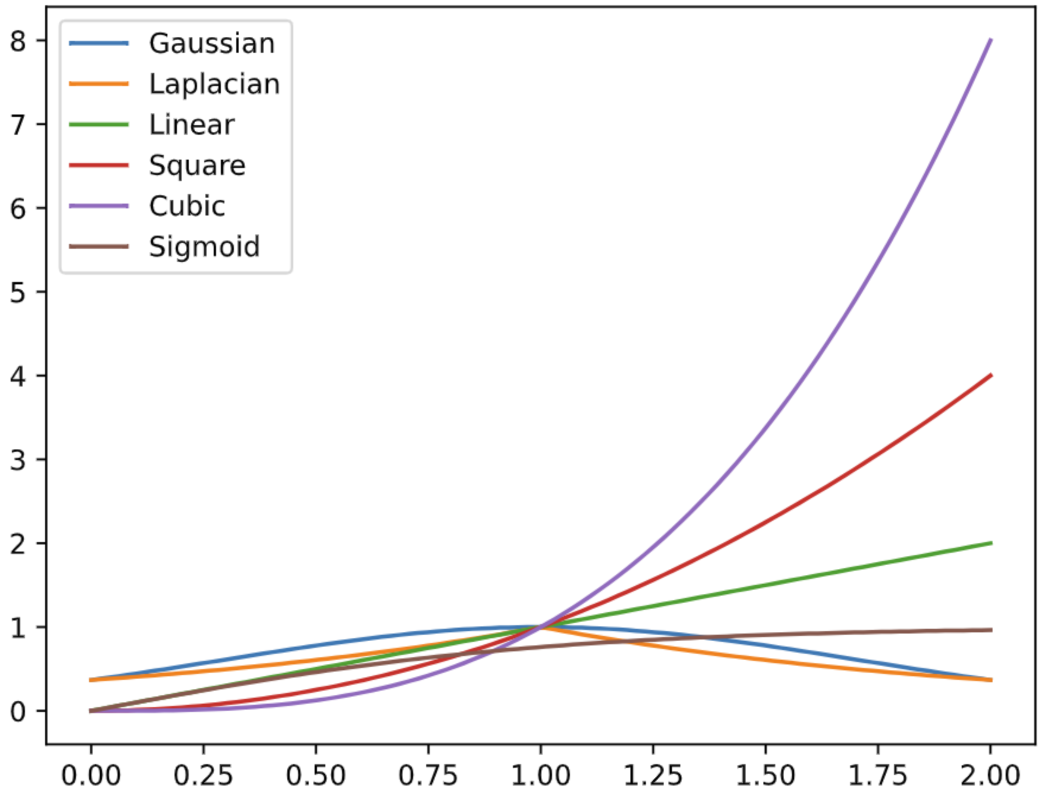
$$\text{Error} = a \exp(-c/h)$$

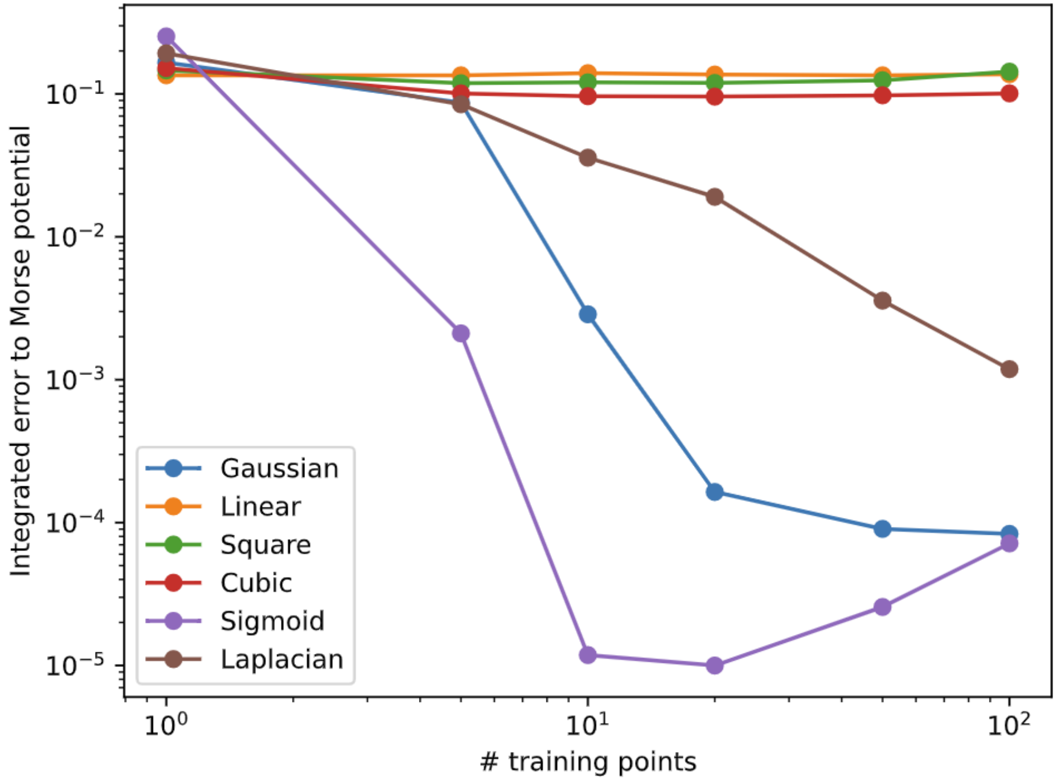




W.R. Madych, S.A. Nelson, *J Approx Theory*, **1992**.

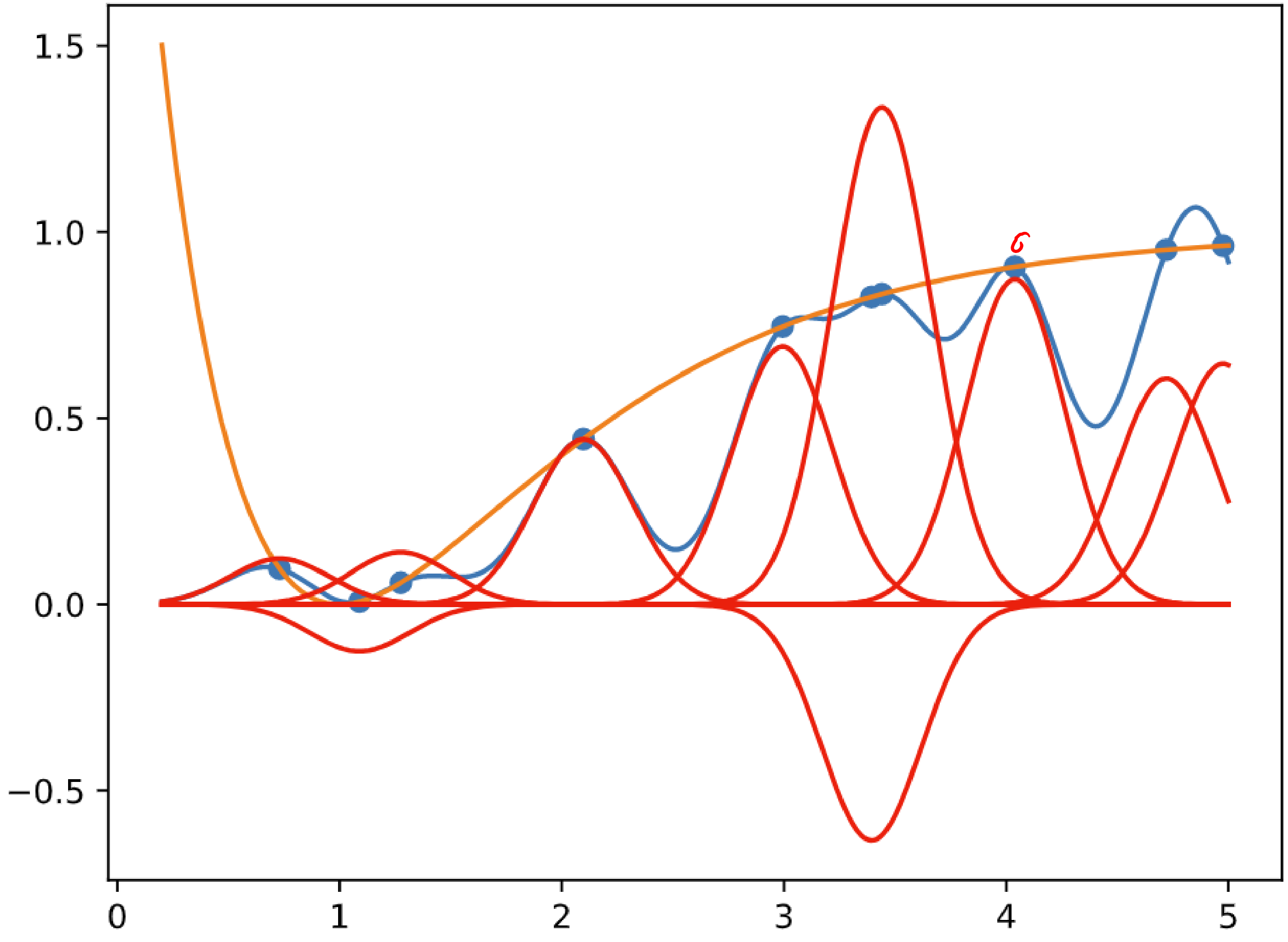
Kernel functions / Mercer‘s condition

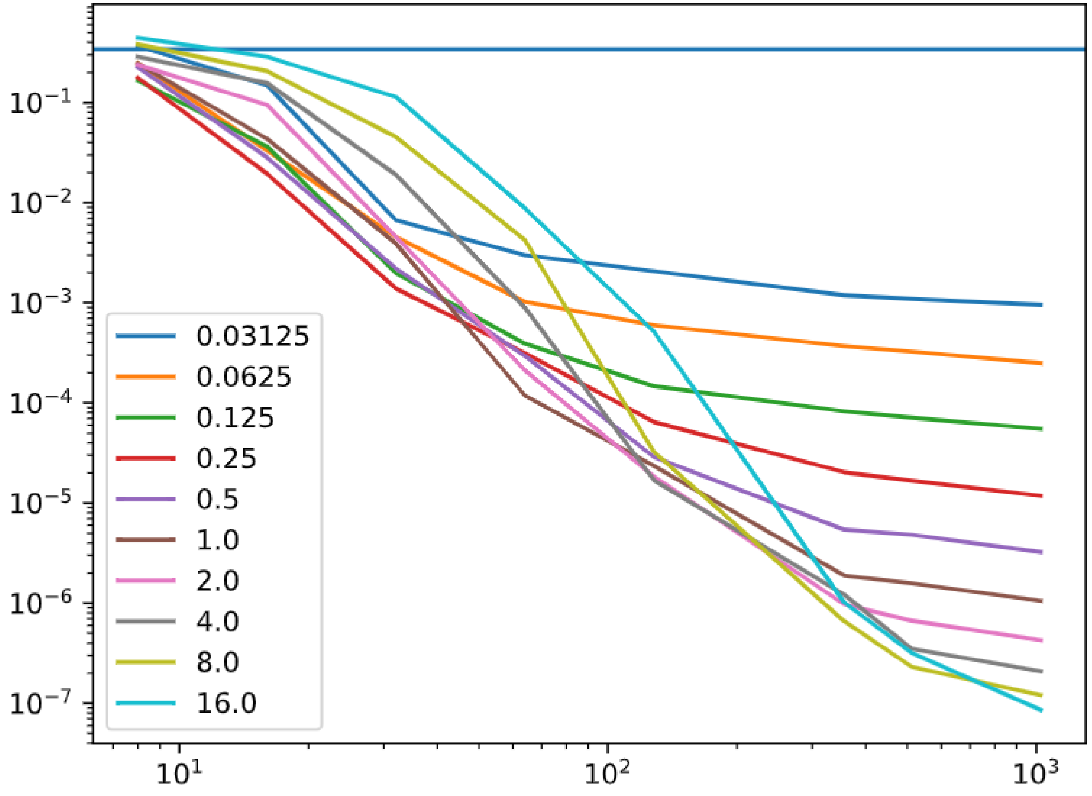




$$\iint g(x)K(x,y)g(y)\,dx\,dy \geq 0$$

Example model



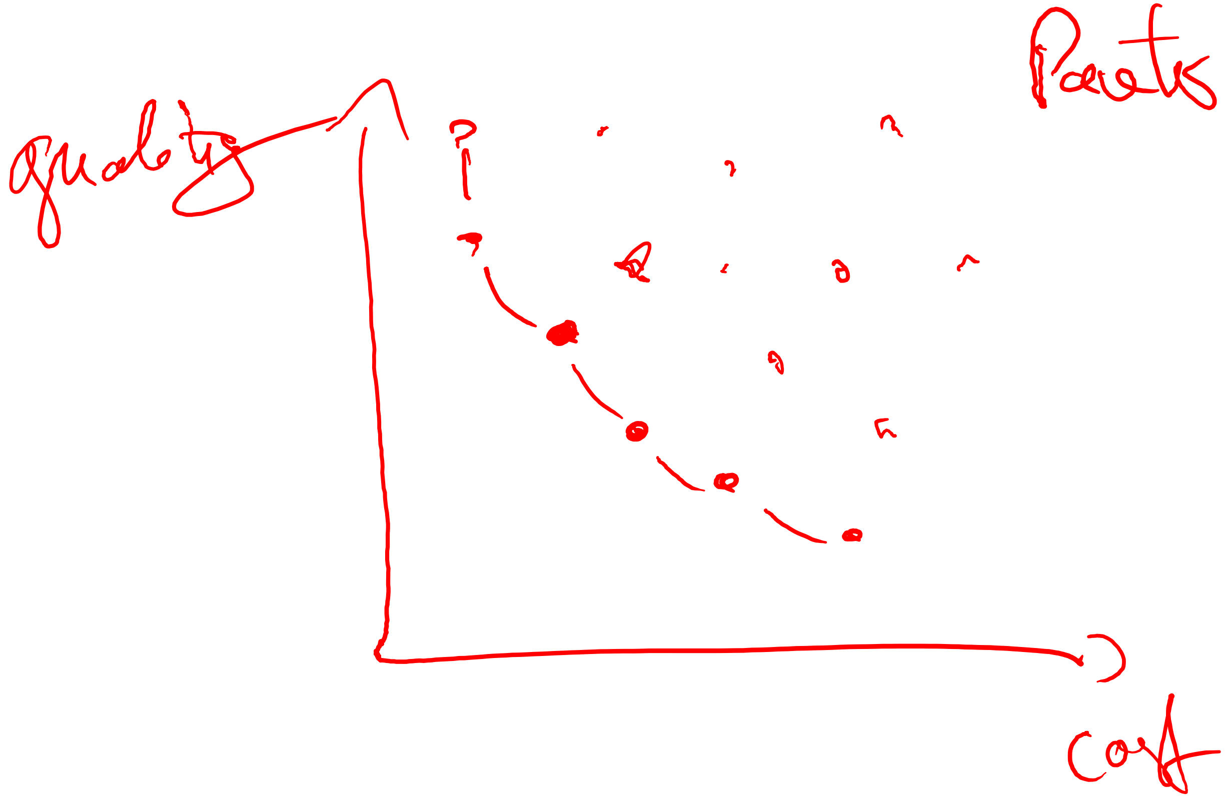


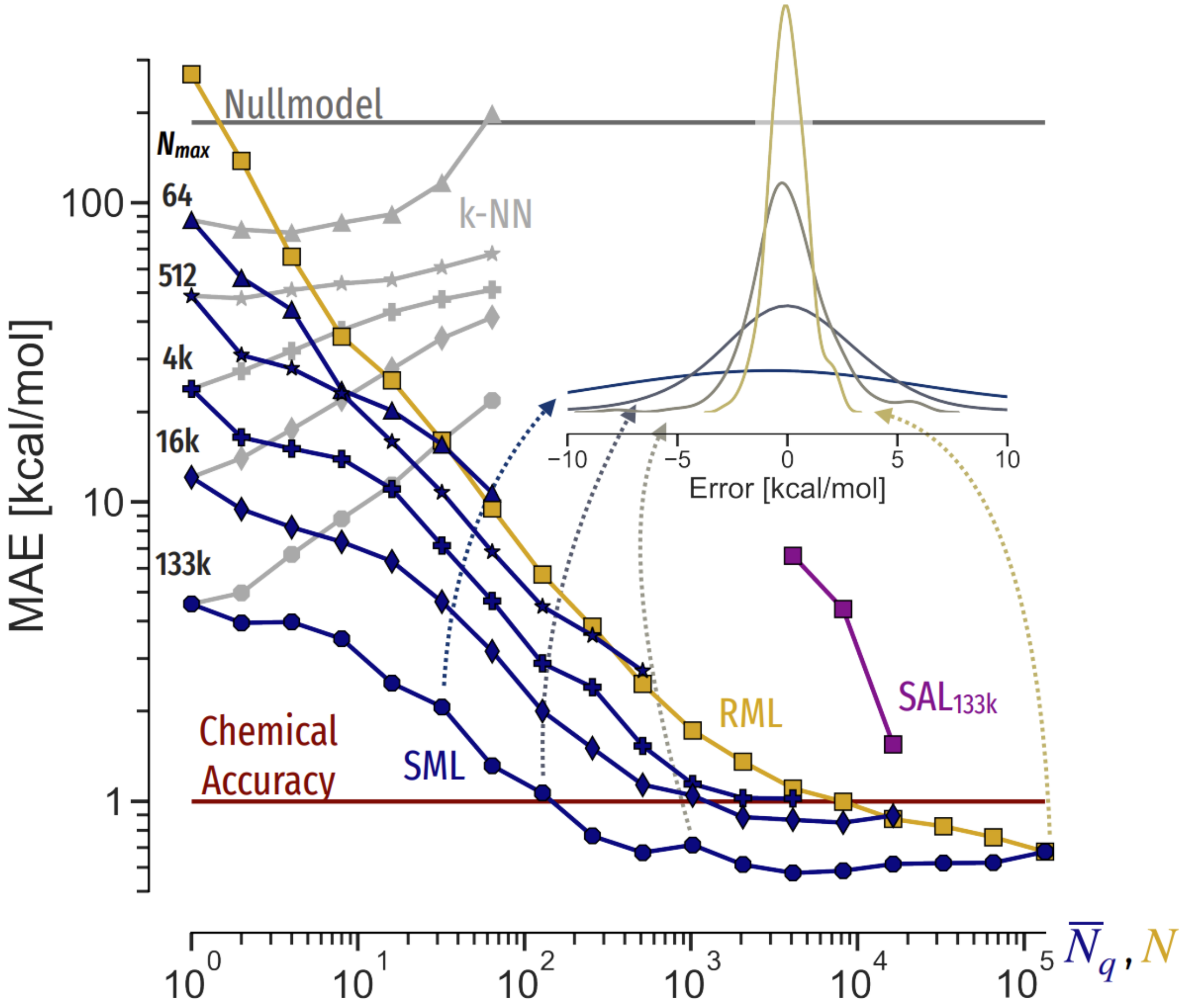
Compute requirements

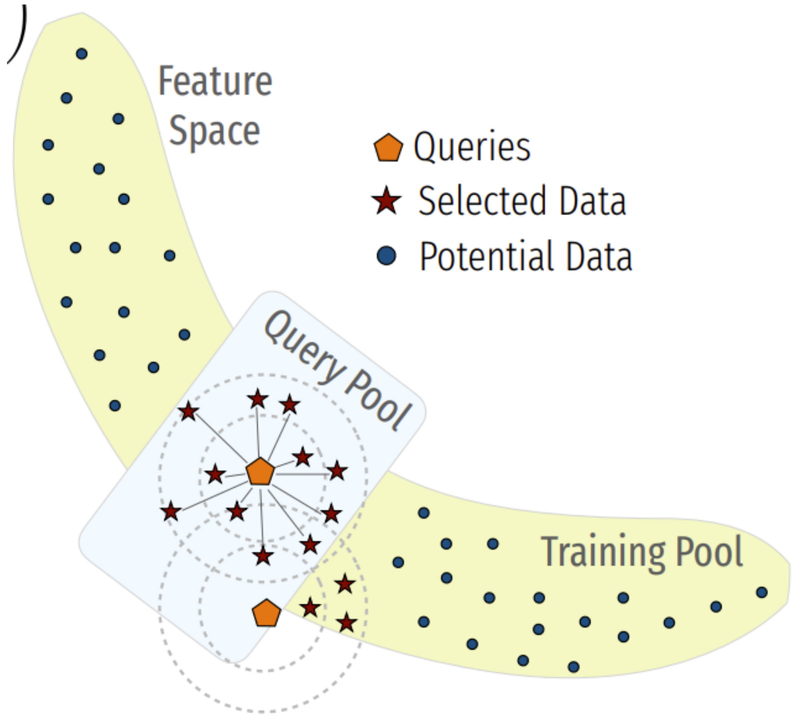
$$\Rightarrow \alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathrm{y}$$

Quadratic scaling in memory
$O(n^{2.8})$ in compute

1k points:	8 MB
10k points:	760 MB
100k points:	75 GB

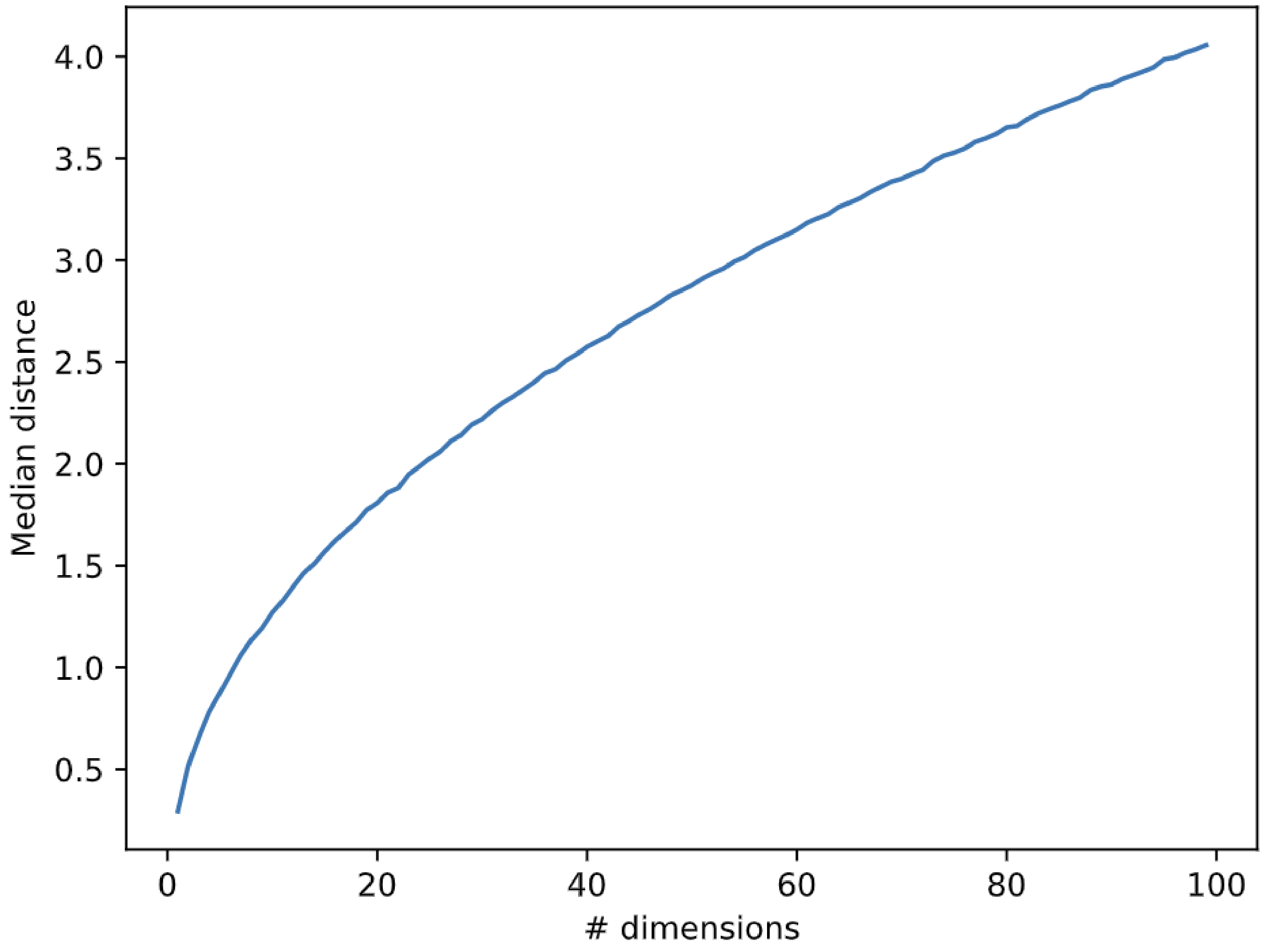






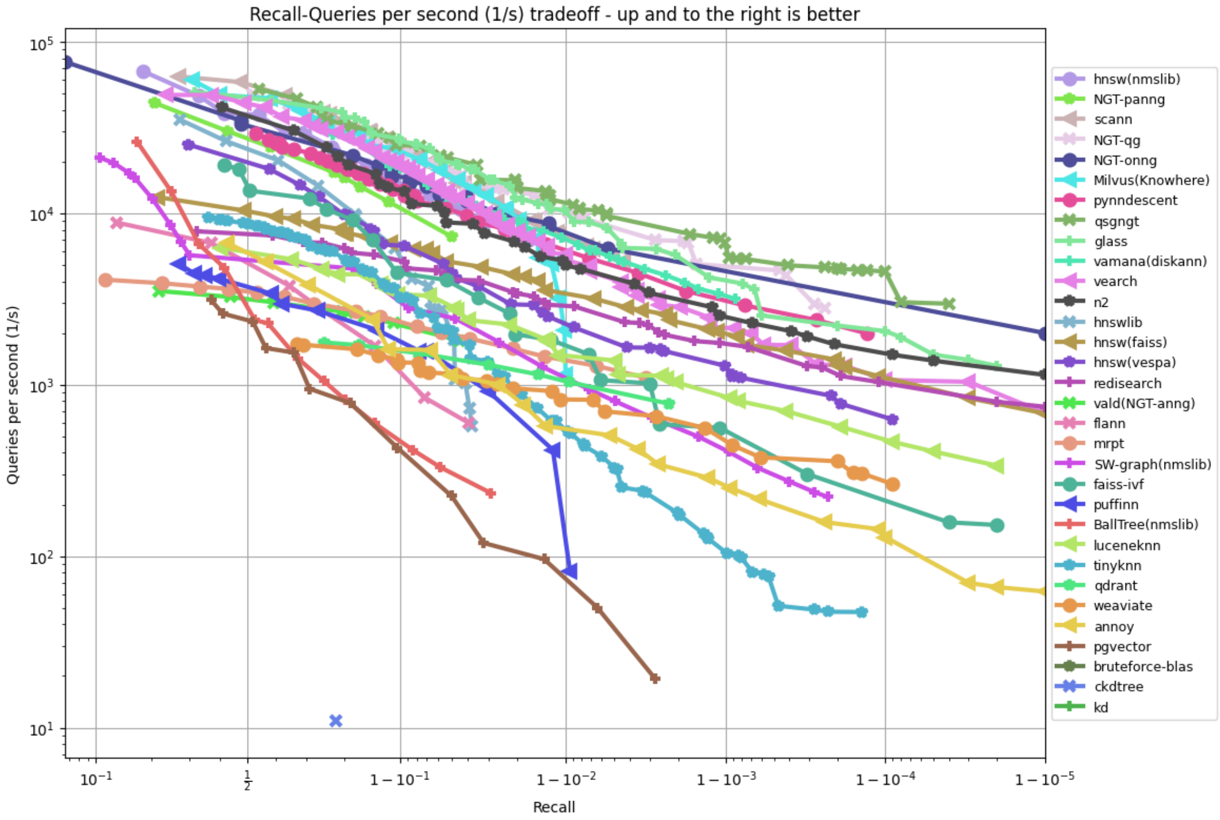
D. Lemm, **GFvR**, OA. von Lilienfeld, *arXiv*, **2205.05633**.

Median distance in the unit cube

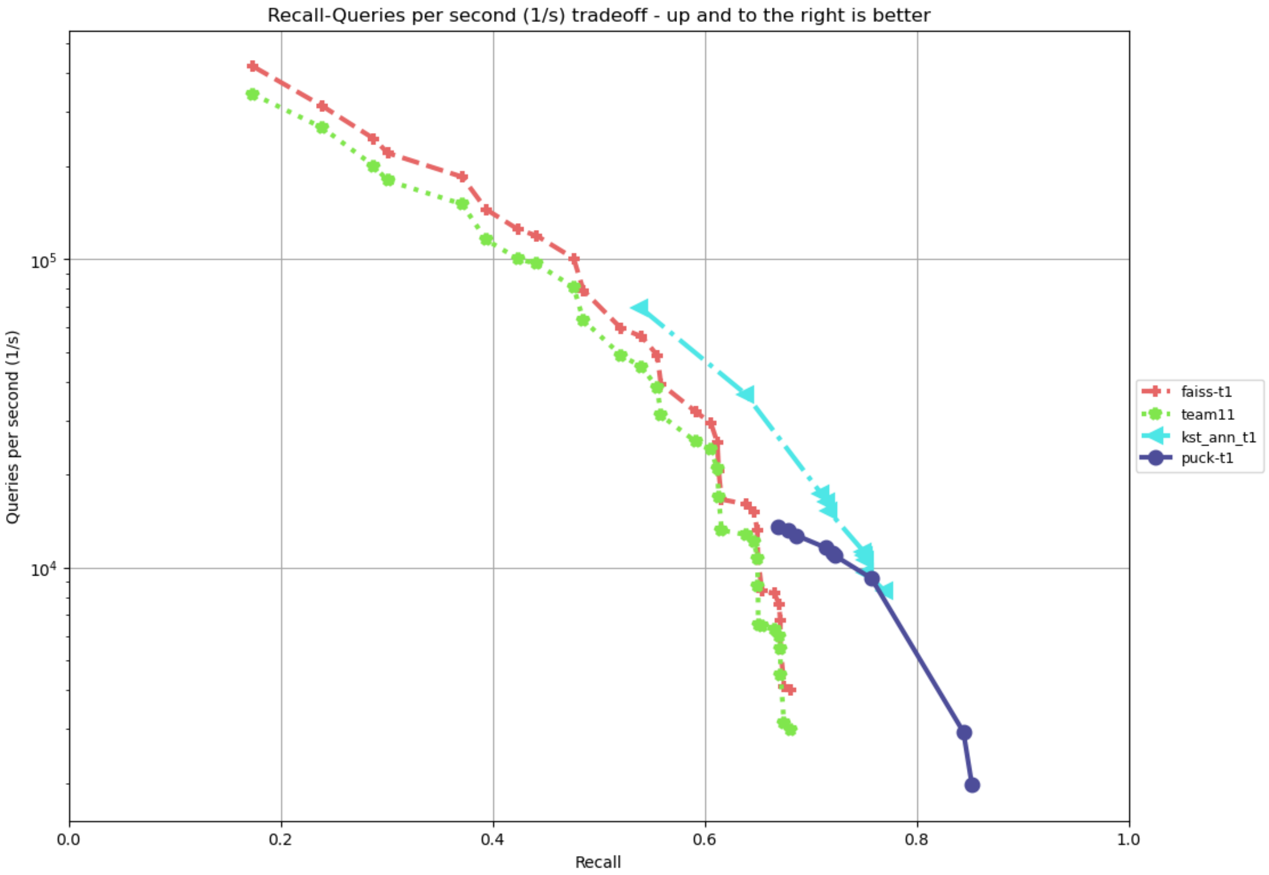


Approximate nearest neighbors

1M data points
25 features

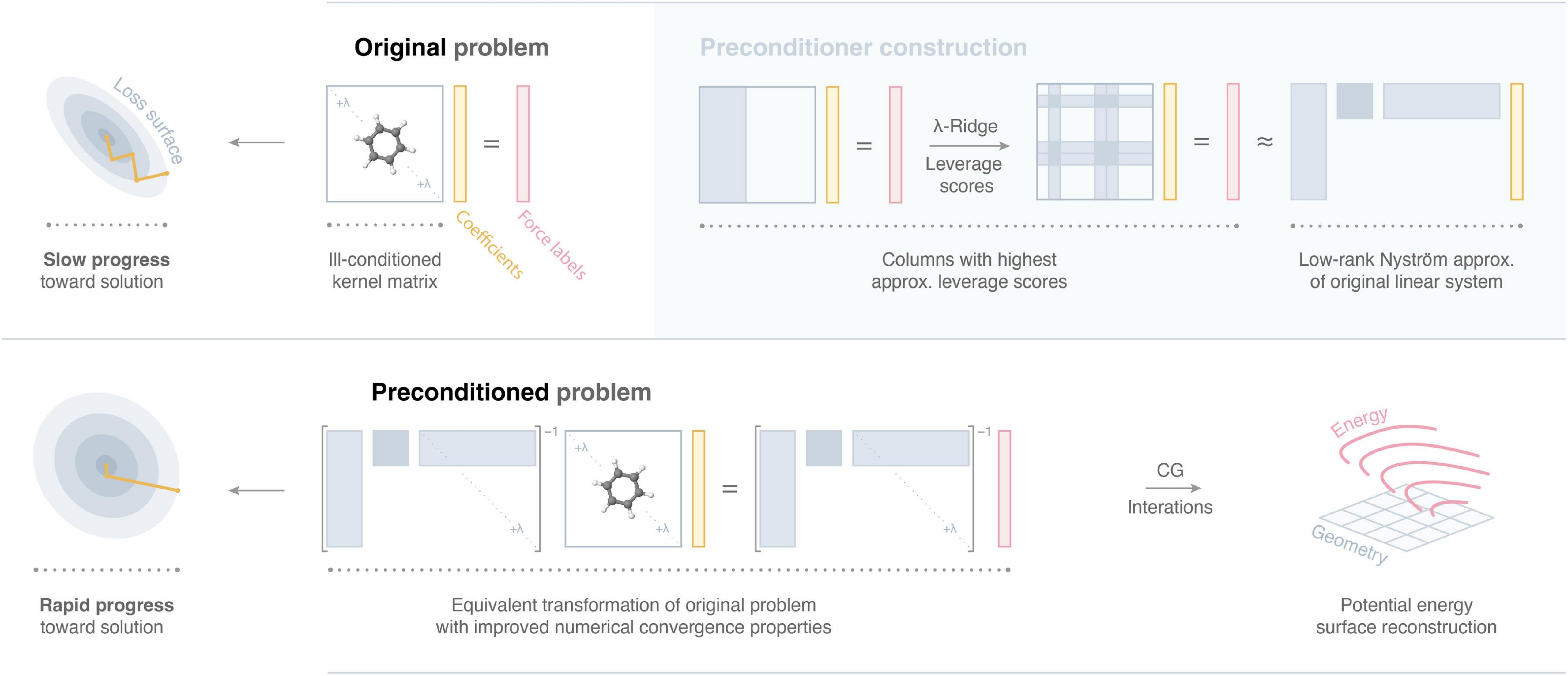


1B data points
96 features



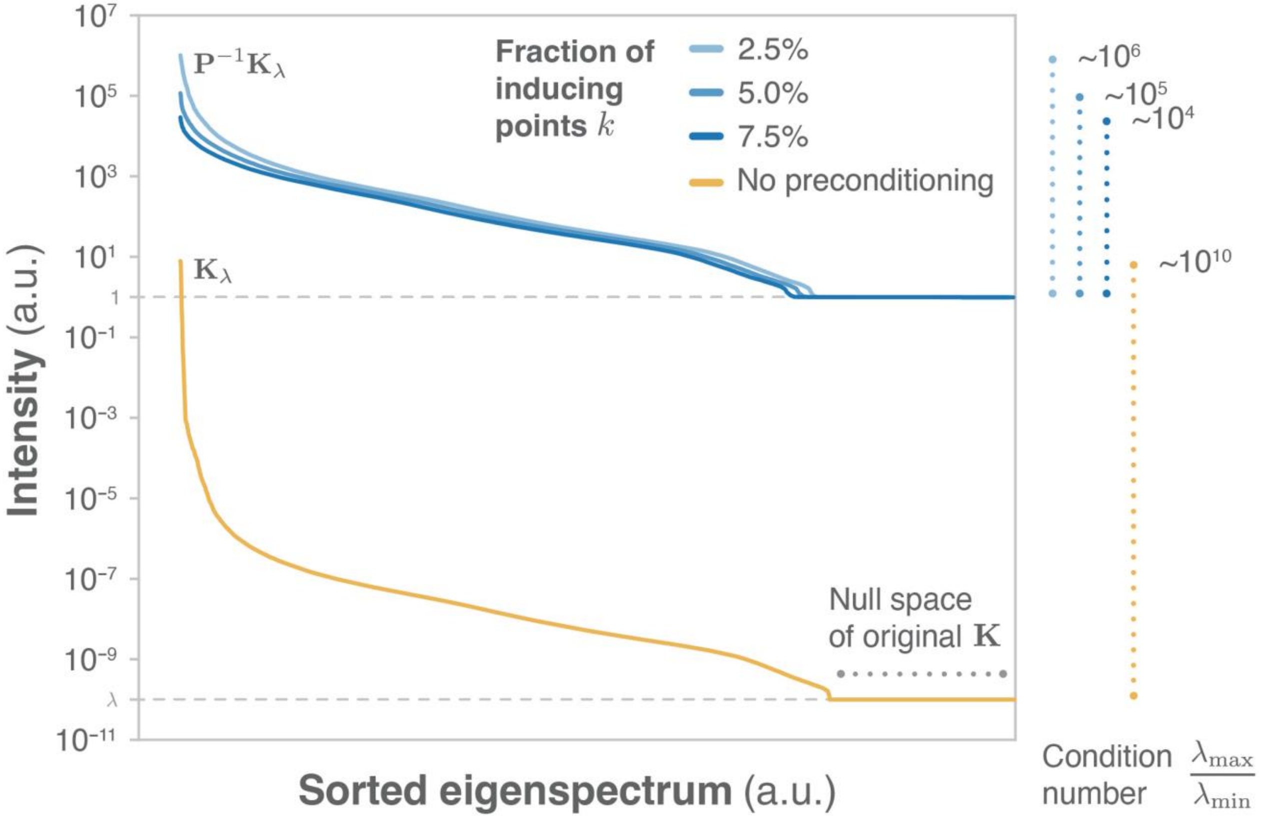
ann-benchmarks.com, big-ann-benchmarks.com

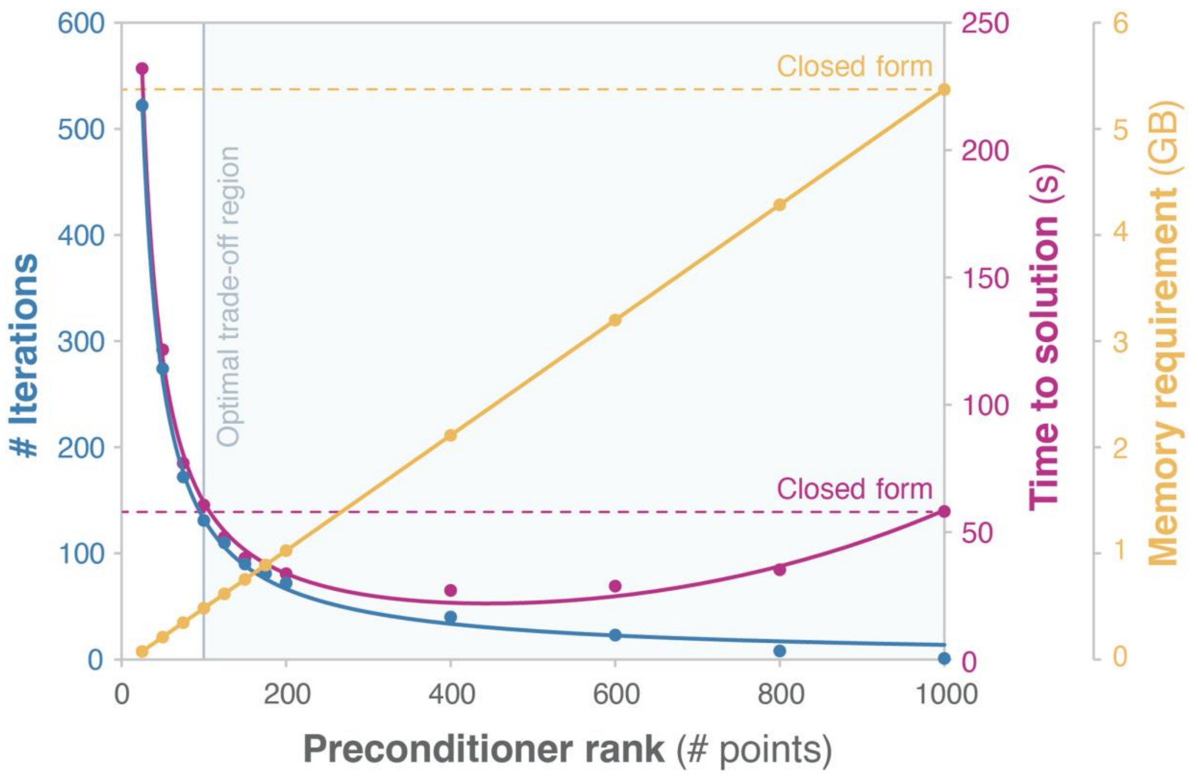
Preconditioners



S. Chimela, et al., *Sci. Adv.* , 2023.

Preconditioners





S. Chimela, et al., *Sci. Adv.* , 2023.

When to use

Pros
- One-Step learning
- Explainable model
- Easy ablatation studies (=„what if certain data was not known“)
- Efficient cross-validation

Cons
- Need to have features
- Memory requirements can be challenging
- Plenty of hyperparameters
- Instable for large data sets

How to fix
Make them overcomplete
Consider nearest neighbors only
More compute
Regularize

Summary Kernel Ridge Regression

- Explainable method
- Requires kernel function
- Can be learned without optimization
- Requires hyperparameter scans
- Highly susceptible to kernel function choice
- Regularization required especially for noisy data
- Easy to implement, hard to implement efficiently