

Kernel-Ridge-Regression

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 $\{\tilde{q}_i\}$

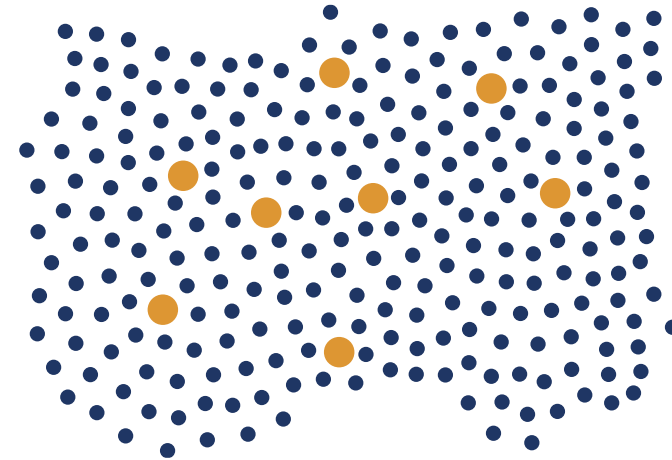
$$\arg \min_{\alpha} \sum_i (q_i - \tilde{q}_i)^2 + \lambda \sum_{ij} \alpha_i \alpha_j k_{ij}$$

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

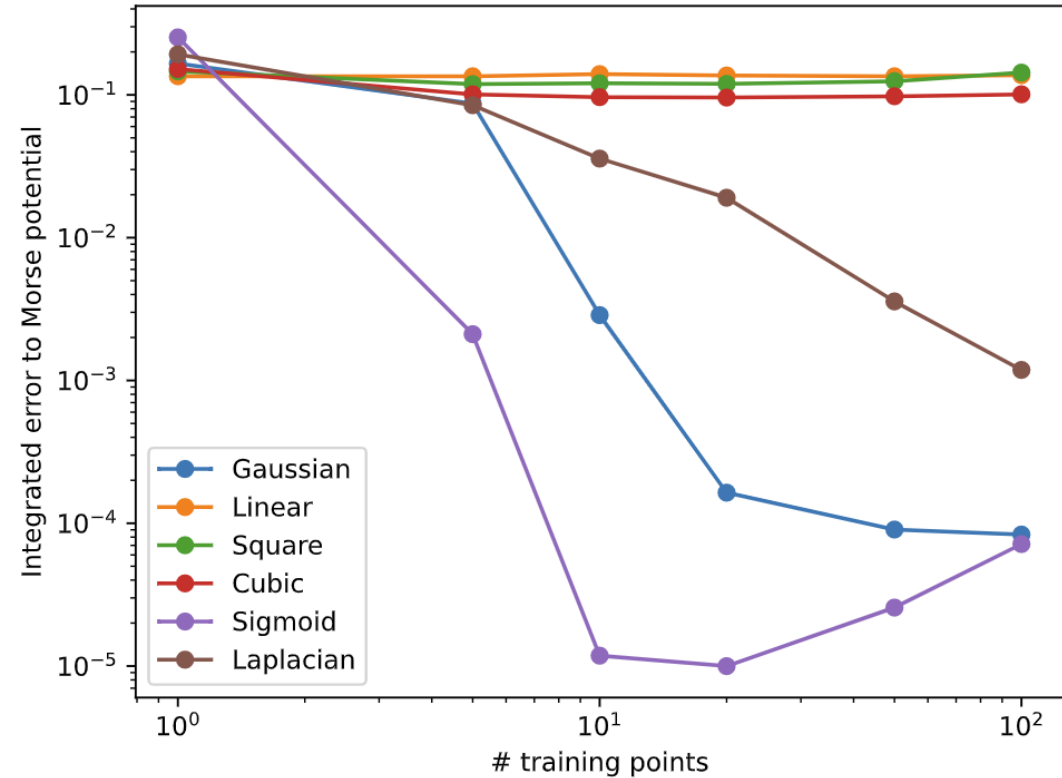
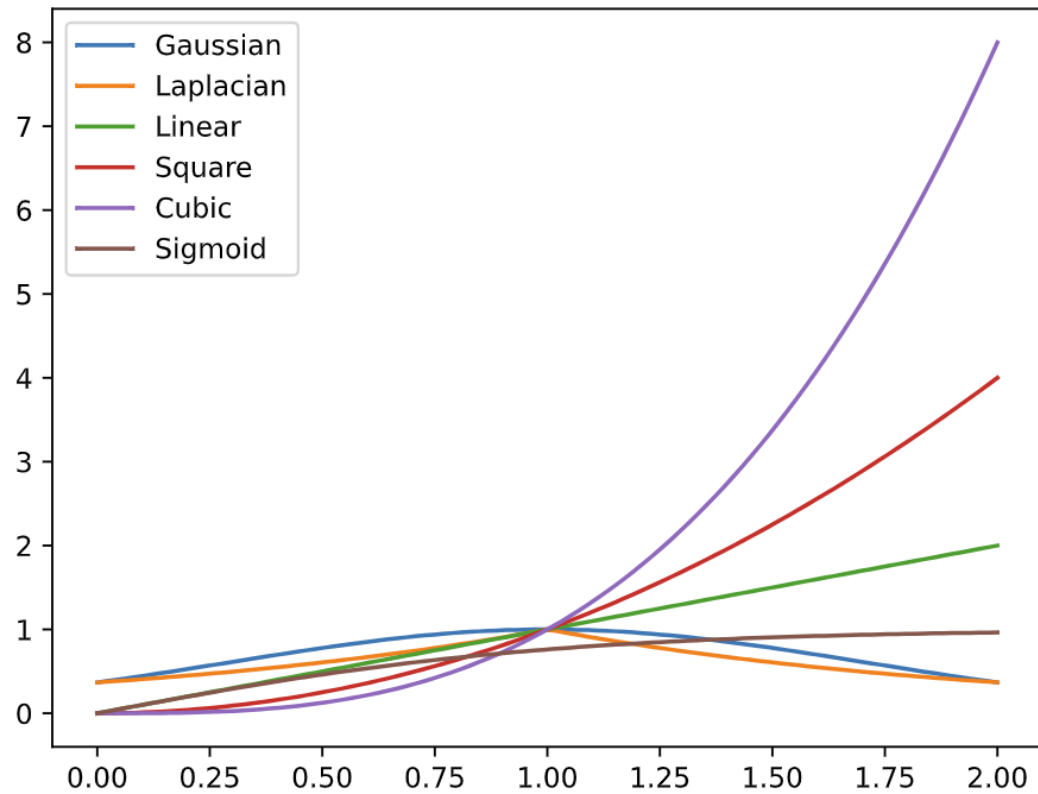
Gaussian kernel: $k(x, y) = \exp(-\gamma \|x - y\|^2)$

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

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

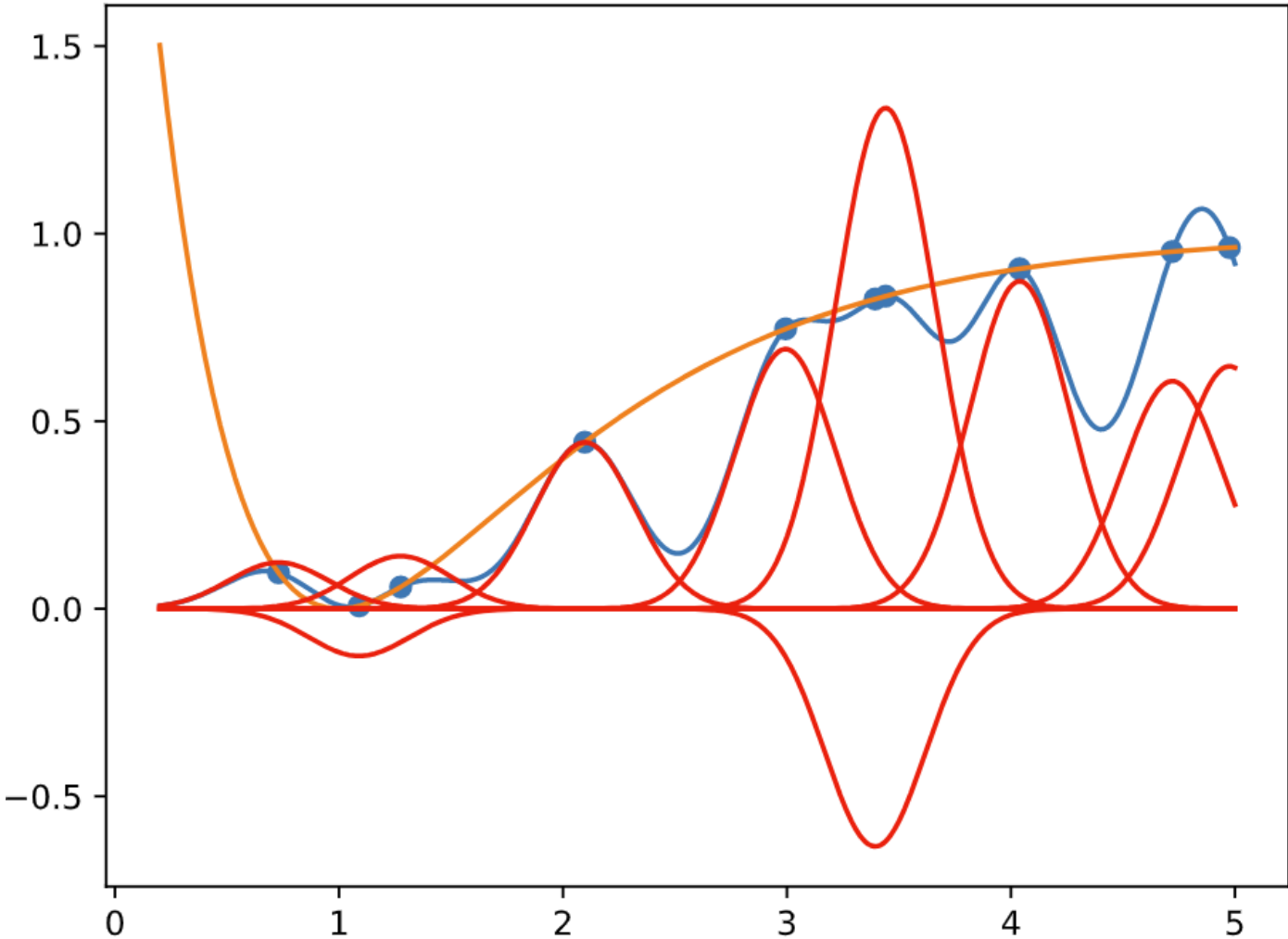


- y
- x_j
- Ω



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

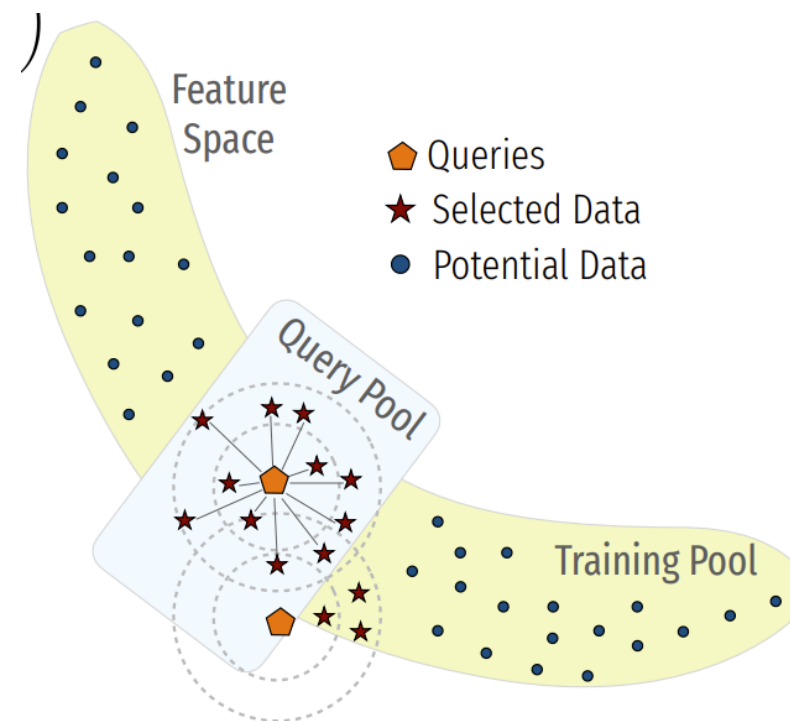
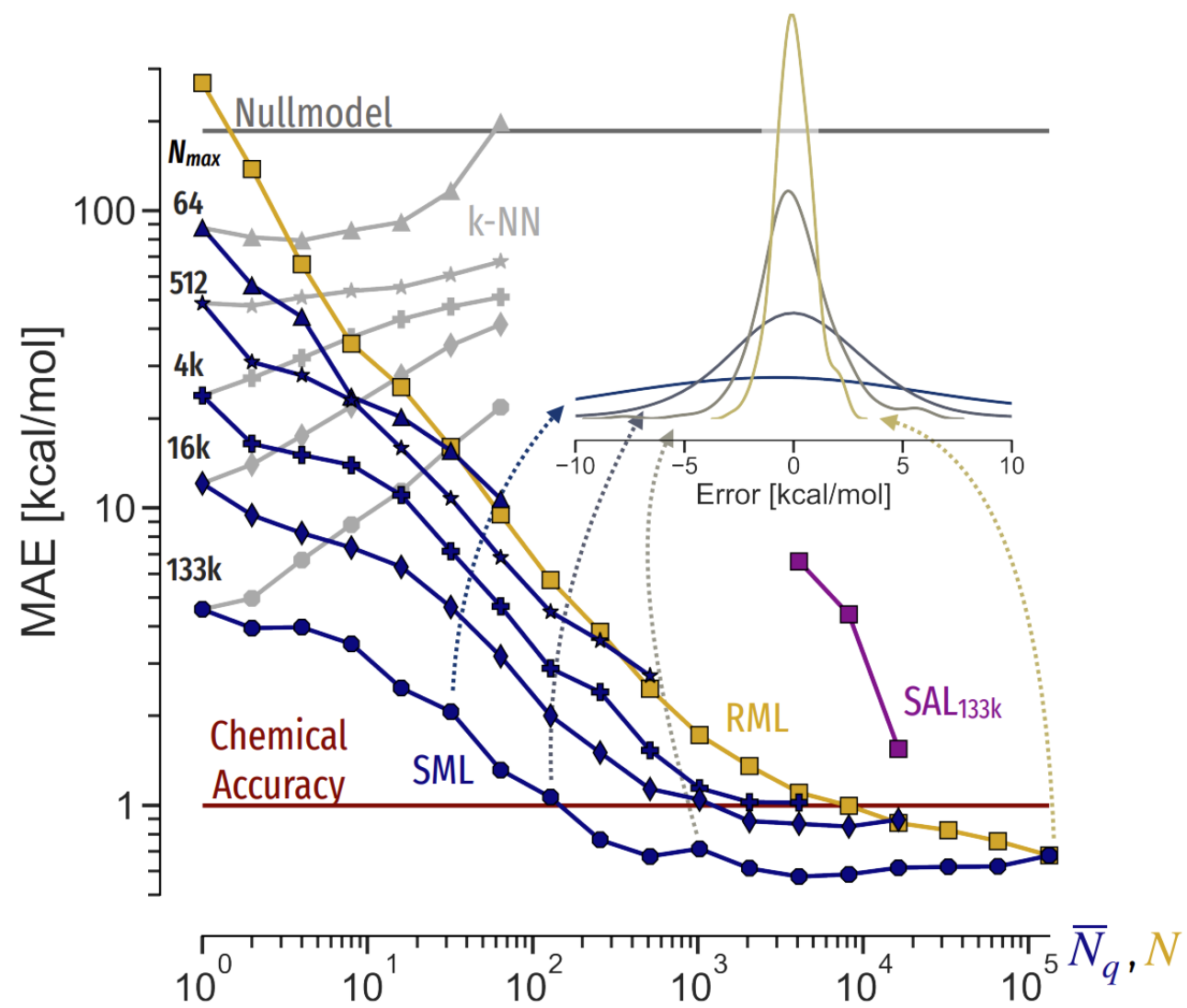
Example model



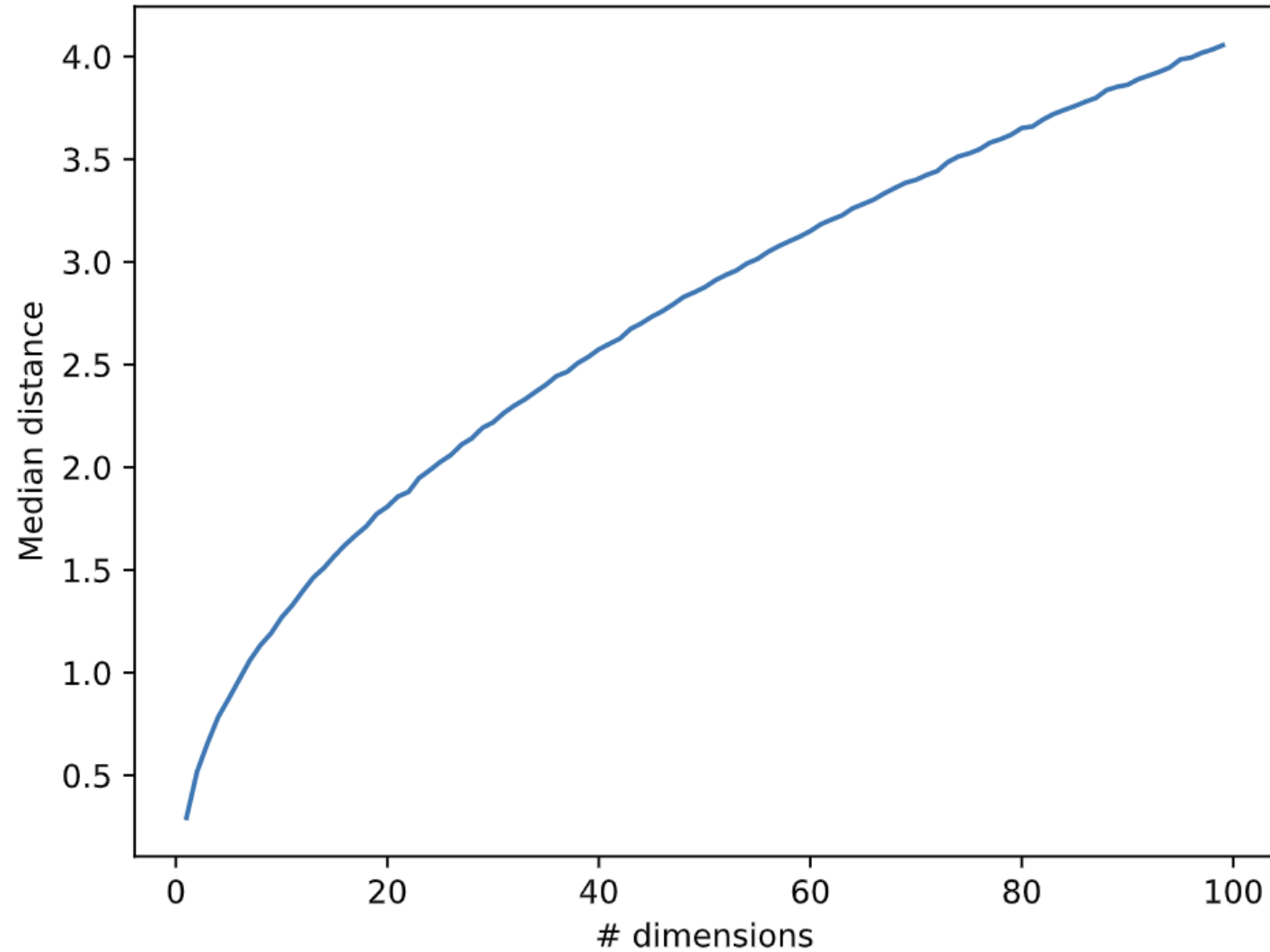
Quadratic scaling in memory
 $O(n^{2.8})$ in compute

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

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



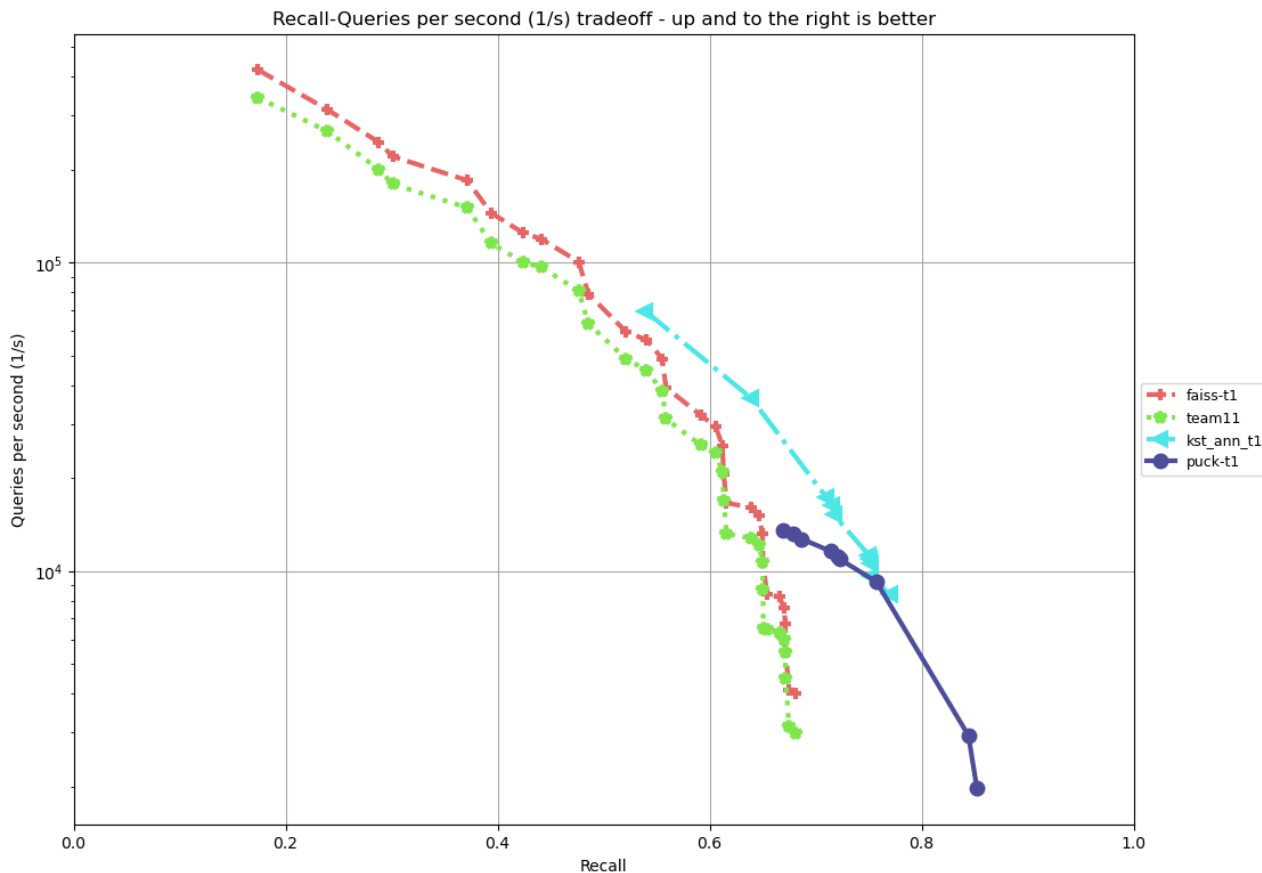
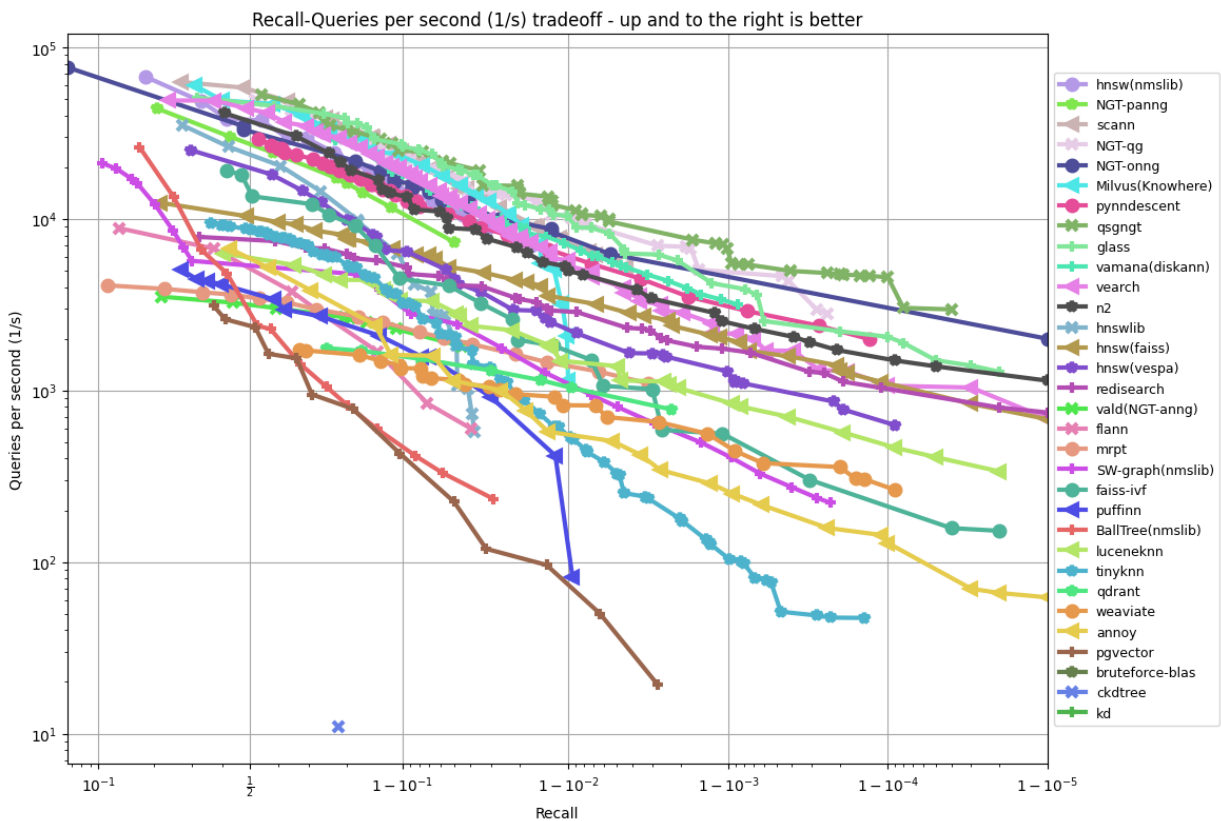
Median distance in the unit cube



Approximate nearest neighbors

1M data points
25 features

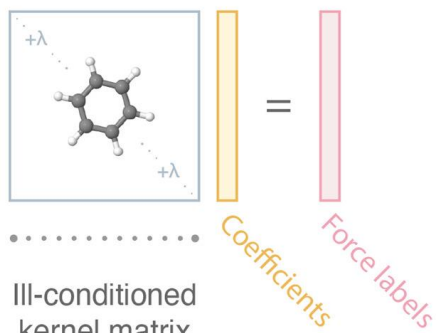
1B data points
96 features





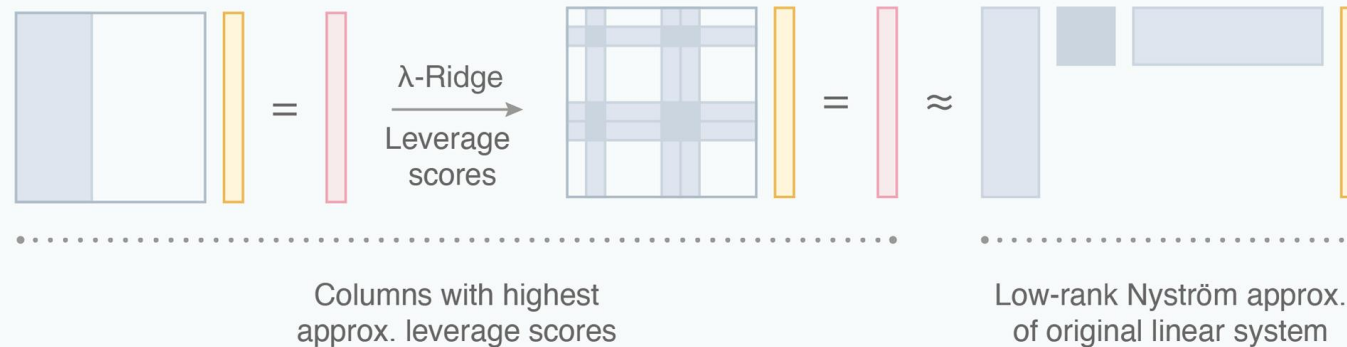
.....
Slow progress
toward solution

Original problem



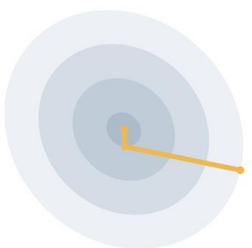
.....
Ill-conditioned
kernel matrix

Preconditioner construction



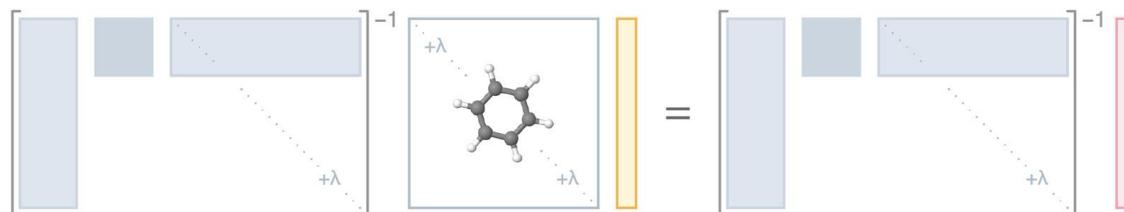
.....
Columns with highest
approx. leverage scores

.....
Low-rank Nyström approx.
of original linear system



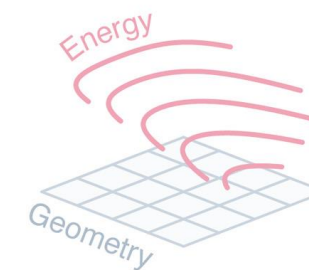
.....
Rapid progress
toward solution

Preconditioned problem

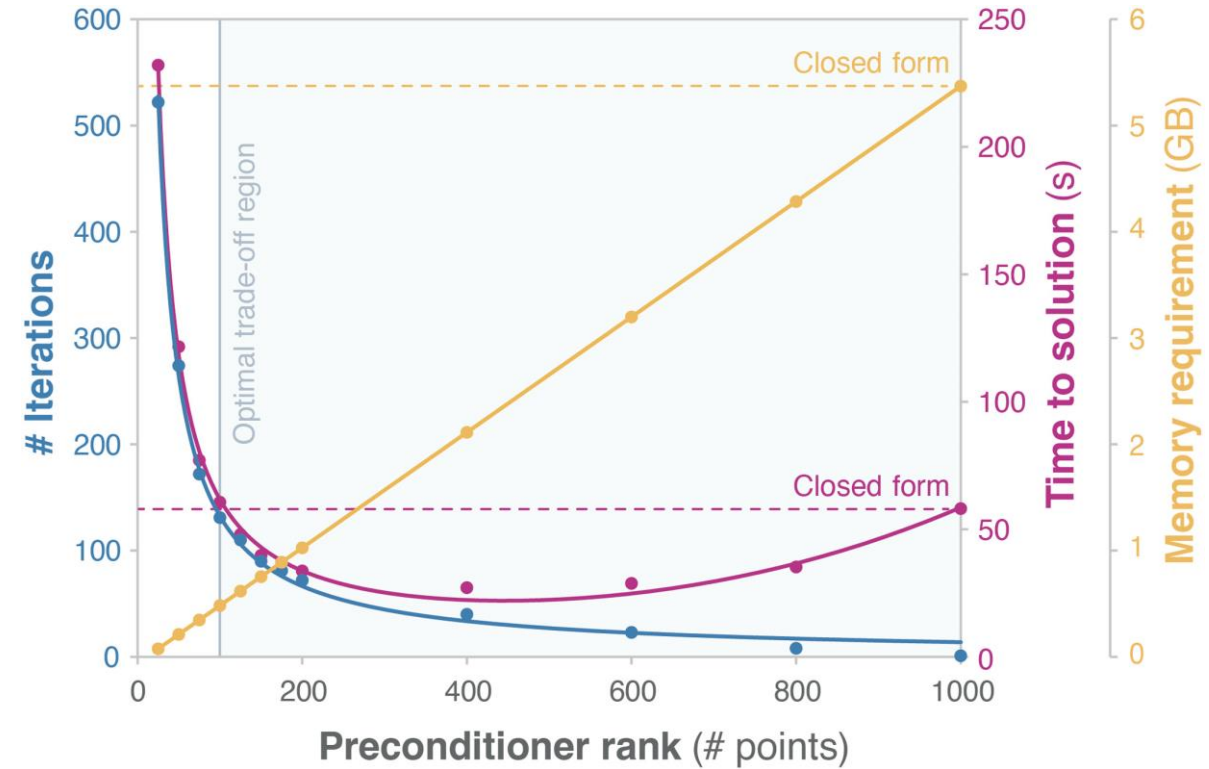
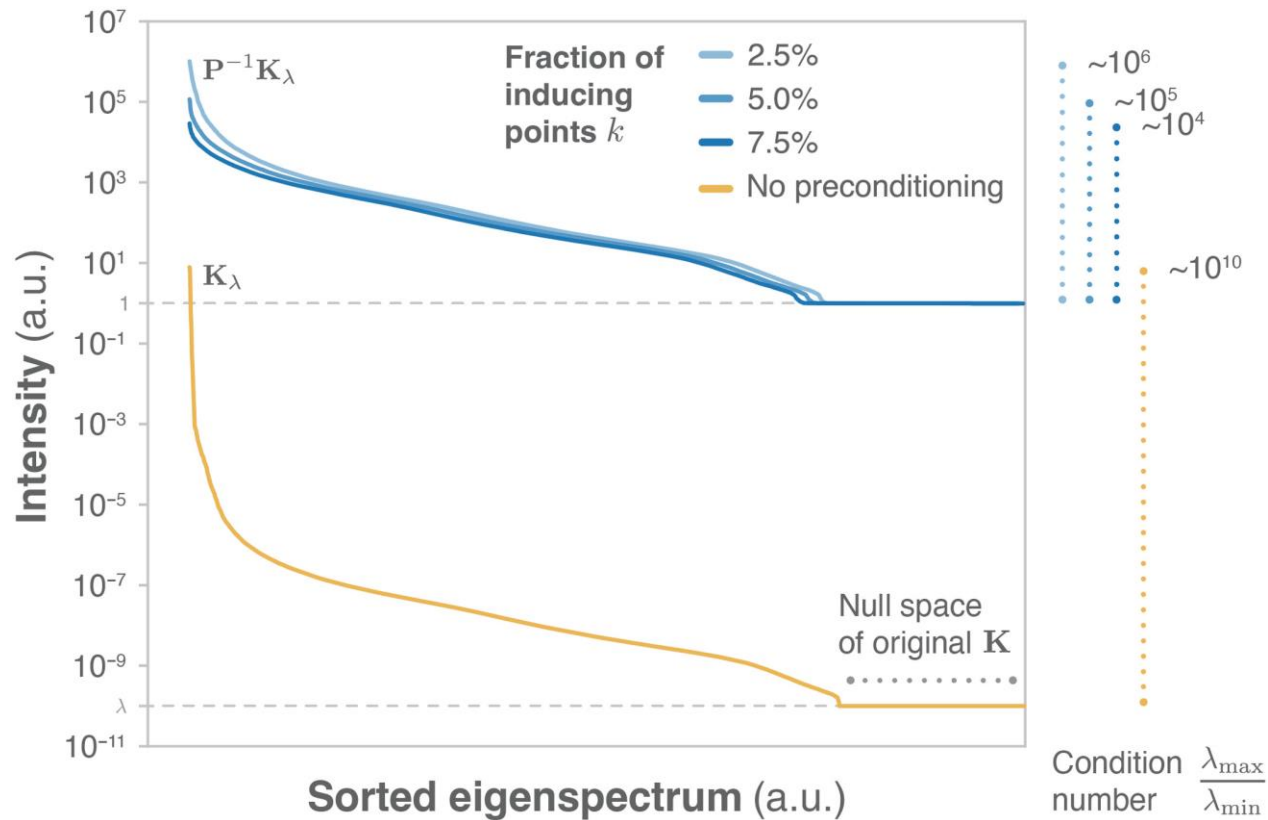


.....
Equivalent transformation of original problem
with improved numerical convergence properties

CG
Iterations



.....
Potential energy
surface reconstruction



Pros

- One-Step learning
- Explainable model
- Easy ablation 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