ON THE LOW RANK THEORY OF DATA SHAPLEY: A KERNEL PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Data Shapley offers a principled framework for data valuation, but remains costly to compute at scale. Although FreeShap avoids repeated model fine-tuning through kernel regression, evaluating many sampled coalitions remains computationally demanding. Low-rank kernel approximation offers a natural acceleration strategy, but its effect on individual Data Shapley values is poorly understood. We investigate how kernel approximation relates to prediction and Data Shapley. Our analysis shows that kernel approximation error controls errors in individual Data Shapley values, whereas predictive agreement alone does not guarantee Shapley agreement. Together with our experiments, these results indicate that eigendecomposition is more reliable than Nyström approximation for Data Shapley. Guided by this analysis, we propose LRFShap, which accelerates coalition evaluation using shared low-dimensional kernel features. Across three downstream tasks, LRFShap achieves performance comparable to FreeShap at suitable ranks, with speedups of up to $22\times$ on data selection. Our code is available at `https://github.com/lrfshap/lrfshap`.

1 INTRODUCTION

The quality of training data strongly affects the performance of neural networks (Abdin et al., 2024; Penedo et al., 2024); for instance, repeated training on recursively generated synthetic data can degrade model performance (Shumailov et al., 2024; Alemohammad et al., 2024). Data valuation is therefore increasingly important, with applications such as data selection and data markets. Data Shapley (Ghorbani & Zou, 2019; Jia et al., 2019) provides a principled approach to data valuation based on the Shapley value (Shapley, 1953), a classical solution concept from cooperative game theory. It assigns each training sample a value based on its average marginal contribution to a given utility over coalitions of training data.

However, computing Data Shapley for large neural networks is prohibitively expensive. Although truncated Monte Carlo sampling (Ghorbani & Zou, 2019) mitigates the combinatorial cost, repeatedly training or fine-tuning the model for sampled coalitions remains the primary bottleneck. Methods based on the neural tangent kernel (NTK) (Jacot et al., 2018) avoid this repeated training (Wu et al., 2022; Wang et al., 2024). In particular, FreeShap (Wang et al., 2024) replaces coalition-wise fine-tuning with kernel regression based on the empirical NTK (eNTK), building on the connection between neural-network training and fixed-kernel learning in the kernel regime (Arora et al., 2019; Lee et al., 2019; Malladi et al., 2023). Nevertheless, FreeShap still requires solving kernel systems for many sampled coalitions. Low-rank kernel approximation is a natural remedy, but how such approximations affect individual Data Shapley values remains poorly understood. We therefore ask:

Does preserving predictions under low-rank kernel approximation also preserve individual Data Shapley values?

Figure 1 summarizes our answer. Classical results show that a small kernel approximation error controls the resulting prediction error (Cortes et al., 2010). Similarly, Proposition 3.2 shows that kernel approximation error also controls errors in individual Data Shapley values. For prediction alone, Bach (2013) showed that the Nyström method can preserve predictive performance without closely approximating the full kernel matrix. This prediction-oriented conclusion does not extend

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

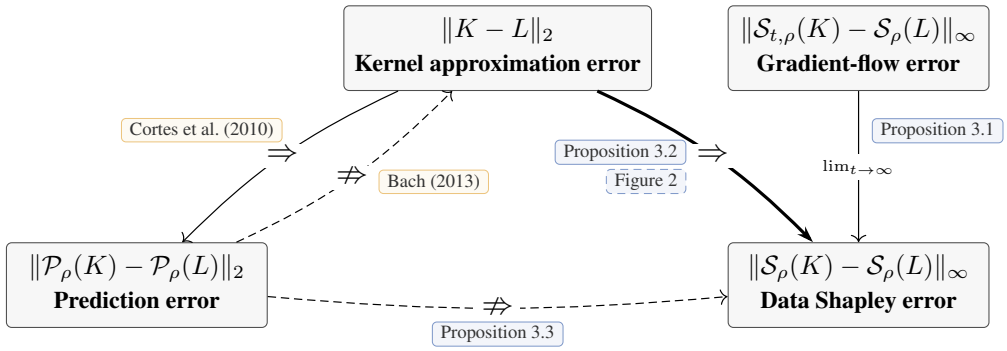

Figure 1: **KPS (kernel–prediction–Shapley) diagram.** Here, K denotes a kernel matrix and L its approximation, while $\mathcal{P}$ and $\mathcal{S}$ denote the prediction and Data Shapley maps, respectively. The symbol $\Rightarrow$ indicates a theoretical implication. Previously reported results are highlighted in orange, whereas our results are highlighted in blue. See Section 3.1 for details.

to Data Shapley. Proposition 3.3 establishes that predictive agreement alone does not guarantee agreement in Data Shapley values. Corollary 3.1 makes this separation concrete for uniform Nyström sampling by exhibiting instances in which, with high probability, the full-coalition prediction is preserved exactly, yet individual Data Shapley estimates incur constant-order relative errors.

Guided by this analysis, we propose Low-Rank Fine-tuning-free Shapley value (**LRFShap**), which accelerates FreeShap through low-rank kernel approximation. Importantly, the same pattern emerges in a practical pretrained-model fine-tuning setting, where Data Shapley is estimated using the eNTK: the Nyström method can yield unstable estimates, whereas eigendecomposition closely matches FreeShap while achieving a substantial speedup.

Our contributions are as follows:

- We establish a stability bound for individual Data Shapley values under kernel approximation and show that predictive agreement alone need not preserve these values, as summarized in the KPS diagram.
- We provide theoretical and empirical evidence that eigendecomposition is better suited to Data Shapley approximation than the Nyström method.
- LRFShap achieves performance comparable to FreeShap across three downstream tasks at suitable ranks, with speedups of up to $22\times$ in data selection.

2 Preliminaries

Notation. Let N, T, and C denote the training-sample, test-sample, and class-index sets, respectively. Let d and P denote the input and parameter dimensions, respectively. Let $D_N := \{(x_i, y_i)\}_{i\in N}$ and $D_T := \{(x_i, y_i)\}_{i\in T}$ be the training and test datasets, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0,1\}^{|C|}$ is the corresponding one-hot encoded class label. For each $S \subseteq N$, define $D_S := \{(x_i, y_i)\}_{i\in S}$. Given an arbitrary ordering $S = \{i_1, \ldots, i_{|S|}\}$, set $X_S := [x_{i_1}, \ldots, x_{i_{|S|}}]^\top \in \mathbb{R}^{|S|\times d}$ and $Y_S := [y_{i_1}, \ldots, y_{i_{|S|}}]^\top \in \{0,1\}^{|S|\times|C|}$. The choice of ordering is immaterial, as it only determines the row ordering of these matrices. The test matrices X_T and Y_T are defined analogously. Unless otherwise specified, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ denotes an arbitrary fixed scalar-valued positive-semidefinite kernel, and $\mathcal{H}_k$ denotes its associated reproducing kernel Hilbert space (RKHS). For index sets $A, B \subseteq N \cup T$, we write $K_{AB} := \big[k(x_i, x_j)\big]_{i\in A, j\in B} \in \mathbb{R}^{|A|\times|B|}$. When a particular kernel is used, its choice is stated explicitly.

Data Shapley. Data Shapley formulates data valuation as a cooperative game in which each training sample is treated as a player (Ghorbani & Zou, 2019; Jia et al., 2019). In the supervised learning setting considered here, for each coalition $S \subseteq N$, let f_S denote the model trained on the corresponding training subset D_S. A utility function $U : 2^N \to \mathbb{R}$ assigns to each coalition S a score

that measures the performance of f_S on the test dataset D_T. Common utility functions include test accuracy and negative test error, which we use for empirical evaluation and theoretical analysis, respectively. The Data Shapley value of a training sample $i \in N$ is defined by the following equivalent coalition-based and permutation-based formulas:

$$\phi_i(U) := \sum_{S \subseteq N \setminus \{i\}} w_{|N|,|S|} \left(U(S \cup \{i\}) - U(S)\right) = \frac{1}{|N|!} \sum_{\pi \in \Pi(N)} \left(U(P_i^\pi \cup \{i\}) - U(P_i^\pi)\right), \quad (1)$$

Here, $w_{|N|,|S|} := \frac{|S|!\left(|N|-|S|-1\right)!}{|N|!}$, $\Pi(N)$ is the set of all permutations of N, and P_i^π is the set of samples that appear before i in permutation π. Thus, $\phi_i(U)$ quantifies the contribution of training sample i to the predictive performance measured by U.

Computing Data Shapley has two main computational bottlenecks:

1. **Combinatorial complexity:** computing the exact average in equation 1 requires evaluating all $|N|!$ permutations of the training samples;
2. **Model retraining cost:** evaluating a marginal contribution $U(S \cup \{i\}) - U(S)$ generally requires training or retraining models on D_S and $D_{S\cup\{i\}}$.

The first bottleneck is commonly mitigated using Monte Carlo sampling based on equation 1, which yields an unbiased estimator of the Shapley value (Maleki et al., 2013). In this work, we focus on the second bottleneck, which remains particularly challenging for large neural networks.

Neural Tangent Kernels for Data Shapley. The Neural Tangent Kernel (NTK) provides a theoretical framework for describing the training dynamics of infinitely wide neural networks (Jacot et al., 2018). Under the kernel regime, neural network training can be characterized as kernel regression with a fixed NTK (Arora et al., 2019; Lee et al., 2019). For finite-width networks, the empirical NTK (eNTK) can be evaluated at pretrained weights, and prior work shows that the resulting eNTK regression can approximate prompt-based fine-tuning behavior in language and vision models(Wei et al., 2022; Malladi et al., 2023).

Let $F(x;\theta) = \left(f^1(x;\theta), \ldots, f^{|C|}(x;\theta)\right) \in \mathbb{R}^{|C|}$ be a differentiable model parameterized by $\theta \in \mathbb{R}^P$, and let θ_0 denote its pretrained parameter vector. Following Wei et al. (2022); Mohamadi et al. (2023), we replace the full multi-output eNTK with the scalar eNTK associated with the first output logit; see Appendix B for further details. For any pair of inputs $x_1, x_2 \in \mathbb{R}^d$,

$$\hat{\Theta}_{\theta_0}(x_1, x_2) := \left[\frac{\partial f^1(x_1;\theta_0)}{\partial \theta_0}\right] \left[\frac{\partial f^1(x_2;\theta_0)}{\partial \theta_0}\right]^\top. \quad (2)$$

For eNTK regression, we specialize the preceding kernel notation to $k = \hat{\Theta}_{\theta_0}$. Given a fixed ridge parameter $\gamma > 0$, for every nonempty coalition $S \subseteq N$, we define

$$f_S^{\mathrm{eNTK}}(X_T) = K_{TS}\left(K_{SS} + \gamma I_{|S|}\right)^{-1} Y_S. \quad (3)$$

Wang et al. (2024) propose Fine-Tuning-Free Shapley Value (FreeShap) to effectively address the model-retraining cost of Data Shapley. For large neural networks, fine-tuning the entire model for each sampled coalition during Monte Carlo estimation is computationally prohibitive. FreeShap replaces these repeated model updates with eNTK regression, so that, once the eNTK matrices are precomputed, coalition utilities can be evaluated without further parameter updates. This enables efficient computation of Data Shapley even for large-scale models.

Best rank-r and Nyström approximations. Let $K_{NN} \in \mathbb{R}^{|N|\times|N|}$ denote the training kernel matrix. Since K_{NN} is symmetric positive semidefinite, write its eigendecomposition as $K_{NN} = U\Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{|N|})$ and $\lambda_1 \geq \cdots \geq \lambda_{|N|} \geq 0$. Let U_r and Λ_r collect the leading r eigenvectors and eigenvalues, respectively. By the Eckart–Young theorem, truncation to these components yields a best rank-r approximation in spectral norm, with approximation error λ_{r+1}. We denote this approximation by

$$K_{NN}^{\mathrm{eig}} := U_r \Lambda_r U_r^\top. \quad (4)$$

As an alternative, the Nyström method (Drineas & Mahoney, 2005) constructs a low-rank approximation using a subset of columns. Let $M \subseteq N$ be a landmark set of size m, and denote the Moore–Penrose pseudoinverse of K_{MM} by $K_{MM}^{\dagger}$. The standard Nyström approximation is the following matrix of rank at most m:

$$K_{NN}^{\mathrm{nys}} := K_{NM} K_{MM}^{\dagger} K_{MN}. \tag{5}$$

3 Low-Rank Kernel Approximation for Data Shapley

3.1 Kernel–Prediction–Shapley Analysis

Throughout this subsection, k, $\mathcal{H}_k$, and K_{AB} retain the generic meanings assigned in the notation above. For the theoretical analysis, we specialize to scalar-response kernel regression rather than multiclass classification. With a slight abuse of notation, y_i denotes a scalar response satisfying $y_i \in \mathbb{R}$ and $|y_i| \leq 1$ for every $i \in N \cup T$, while $Y_S \in \mathbb{R}^{|S|}$ and $Y_T \in \mathbb{R}^{|T|}$ denote the corresponding response vectors. The multiclass convention introduced in Section 2 is retained elsewhere. We further assume that $\kappa := \sup_{x \in \mathbb{R}^d} \sqrt{k(x,x)} < \infty$.

We write $\mathrm{Sym}(|N|)^+$ for the open cone of real symmetric positive-definite $|N| \times |N|$ matrices, whose closure $\overline{\mathrm{Sym}(|N|)^+}$ is the cone of real symmetric positive-semidefinite matrices. We regard Y_N, Y_T, and K_{TN} as fixed, while G, representing either the exact training kernel matrix or one of its approximations, varies over $\overline{\mathrm{Sym}(|N|)^+}$. For $\rho > 0$, $t \geq 0$, and every nonempty $S \subseteq N$, set $A_{S,\rho}(G) := G_{SS} + \rho |S| I_{|S|}$[1]. Since $A_{S,\rho}(G) \succeq \rho |S| I_{|S|} \succ 0$ for every $G \in \overline{\mathrm{Sym}(|N|)^+}$, define

$$\mathcal{P}_{t,\rho}^{S} : \overline{\mathrm{Sym}(|N|)^+} \longrightarrow \mathbb{R}^{|T|}, \qquad \mathcal{P}_{t,\rho}^{S}(G) := K_{TS} \left(I_{|S|} - e^{-\frac{2t}{|S|} A_{S,\rho}(G)} \right) A_{S,\rho}(G)^{-1} Y_S.$$

For the empty coalition, set $\mathcal{P}_{t,\rho}^{\varnothing}(G) := 0$. In particular, when $G = K_{NN}$, $\mathcal{P}_{t,\rho}^{S}(K_{NN})$ is precisely the prediction at time t of the zero-initialized gradient-flow solution to the regularized squared-loss problem in $\mathcal{H}_k$; see Appendix C. We define the stationary prediction is $\mathcal{P}_{\rho}^{S}(G) := \mathcal{P}_{\infty,\rho}^{S}(G) := K_{TS} A_{S,\rho}(G)^{-1} Y_S$. For each $G \in \overline{\mathrm{Sym}(|N|)^+}$, define the utility function

$$U_{t,\rho}^{G} : 2^N \to \mathbb{R}, \qquad U_{t,\rho}^{G}(S) := -|T|^{-1} \| Y_T - \mathcal{P}_{t,\rho}^{S}(G) \|_2^2.$$

Using the Shapley-value definition in equation 1, define

$$\mathcal{S}_{t,\rho} : \overline{\mathrm{Sym}(|N|)^+} \longrightarrow \mathbb{R}^{|N|}, \qquad \mathcal{S}_{t,\rho}(G) := \left(\phi_i(U_{t,\rho}^{G}) \right)_{i \in N}.$$

Since the Shapley value is a finite linear combination of coalition utilities, $\mathcal{S}_{t,\rho}(G)$ also converges as $t \to \infty$. We therefore define the grand-coalition prediction map and the Data Shapley map by

$$\mathcal{P}_{\rho}(G) := \mathcal{P}_{\rho}^{N}(G), \qquad \mathcal{S}_{\rho}(G) := \mathcal{S}_{\infty,\rho}(G).$$

We show that the Shapley values induced by finite-time kernel gradient-flow predictors converge to their infinite-time counterparts. Thus, within the kernel regime, the proposition justifies replacing finite-time kernel training with its infinite-time limit when computing Shapley values, providing theoretical support for the stationary-predictor approximation employed by FreeShap.

Proposition 3.1 (Gradient-flow stability)**.** *Assume that $\rho > 0$. There exists a constant $C_{\kappa,\rho} > 0$, depending only on κ and ρ, such that, for every $t \geq 0$,*

$$\boxed{\| \mathcal{S}_{t,\rho}(K_{NN}) - \mathcal{S}_{\rho}(K_{NN}) \|_{\infty} \leq C_{\kappa,\rho} e^{-2\rho t}.}$$

Under the present assumptions, $\rho > 0$ provides the uniform spectral gap underlying the dataset-independent exponential rate in Proposition 3.1. In the ridgeless case, each fixed coalition still converges, but the bound contains $\lambda_{\min}^{+}(K_{SS})^{-1}$, which cannot be controlled uniformly because $\lambda_{\min}^{+}(K_{SS})$ may be arbitrarily small. Hence, no dataset-independent Data Shapley stability bound follows without a uniform positive lower bound; see Appendices C and D.1.

We next control the effect of replacing the exact training kernel matrix with a positive-semidefinite approximation. This formulation enables a unified comparison of low-rank approximation methods through their training-kernel approximation errors.

[1]Here, ρ denotes a fixed RKHS regularization coefficient under the averaged empirical loss, yielding the diagonal ridge $\rho|S|$. In our implementation, we instead use a coalition-independent ridge parameter γ.

Proposition 3.2 (Kernel-approximation stability)**.** *Assume that $\rho > 0$. Let K_{NN} be the exact training kernel matrix and let L_{NN} be a symmetric positive-semidefinite approximation. Then there exists a constant $\widetilde{C}_{\kappa,\rho} > 0$, depending only on κ and ρ, such that*

$$\boxed{\|\mathcal{S}_\rho(K_{NN}) - \mathcal{S}_\rho(L_{NN})\|_\infty \le \widetilde{C}_{\kappa,\rho} \frac{1+\log|N|}{|N|} \|K_{NN} - L_{NN}\|_2.}$$

Our implementation is different from the setting of Proposition 3.2 in several respects; Appendix E empirically examines the effect of each of these differences. The preceding proposition controls Data Shapley through a global training-kernel approximation error. By contrast, agreement of the grand-coalition predictions alone does not provide even a local control of Data Shapley. For the statement below, we restrict $\mathcal{P}_\rho$ and $\mathcal{S}_\rho$ to the open cone $\mathrm{Sym}(|N|)^+$.

Proposition 3.3 (Prediction does not determine Data Shapley)**.** *Let $G_0 \in \mathrm{Sym}(|N|)^+$, and suppose that $\ker D\mathcal{P}_\rho(G_0) \not\subseteq \ker D\mathcal{S}_\rho(G_0)$. Then there do not exist a neighborhood $\mathcal{V}$ of G_0 in $\mathrm{Sym}(|N|)^+$ and a function $\omega : [0,\infty) \to [0,\infty)$ satisfying $\omega(0) = 0$ such that, for all $K, L \in \mathcal{V}$,*

$$\boxed{\|\mathcal{S}_\rho(K) - \mathcal{S}_\rho(L)\|_2 \le \omega\left(\|\mathcal{P}_\rho(K) - \mathcal{P}_\rho(L)\|_2\right).}$$

Although Nyström approximations may preserve average predictive performance without closely approximating the full kernel matrix (Bach, 2013), Proposition 3.3 shows that predictive agreement alone does not guarantee Data Shapley accuracy. We therefore compare low-rank methods through the training-kernel approximation error controlled by Proposition 3.2.

Corollary 3.1 (Informal, limitation of uniform Nyström sampling)**.** *Uniform Nyström sampling need not preserve individual Data Shapley values at the natural $O(|N|^{-1})$ scale. The resulting Shapley-error bound contains an additional term of order $(1+\log|N|)/\sqrt{m}$, which cannot certify error of order $1/|N|$ when $m < |N|$. Moreover, there exist instances for which, with probability $1 - O(|N|^{-1/3})$, the full-coalition prediction is preserved exactly, while the Nyström Data Shapley approximation has constant-order relative error on $\Theta(|N|^{1/3})$ samples.*

3.2 LRFSHAP: LOW-RANK DATA SHAPLEY COMPUTATION

We now present LRFShap, a low-rank approximation framework for accelerating kernel-based Data Shapley estimation. Let K_{NN} and K_{TN} denote the full train–train and test–train eNTK matrices, respectively. Let $\mathcal{A}$ denote a low-rank feature construction, and let ϑ denote its construction-specific configuration. We write $(\Phi_{\mathrm{tr}}, \Phi_{\mathrm{te}}) = \mathcal{A}(K_{NN}, K_{TN}; \vartheta)$, where $\Phi_{\mathrm{tr}} \in \mathbb{R}^{|N|\times q}$, $\Phi_{\mathrm{te}} \in \mathbb{R}^{|T|\times q}$, and $q \ll |N|$. Here, q denotes the resulting feature dimension, and both the meaning of ϑ and the value of q depend on the chosen construction. The feature matrices are constructed so that $K_{NN} \approx \Phi_{\mathrm{tr}}\Phi_{\mathrm{tr}}^\top$ and $K_{TN} \approx \Phi_{\mathrm{te}}\Phi_{\mathrm{tr}}^\top$. We instantiate $\mathcal{A}$ using eigen and Nyström features below.

For a nonempty subset $S \subseteq N$, let $\Phi_S \in \mathbb{R}^{|S|\times q}$ denote the rows of Φ_{tr} indexed by S. Replacing the kernel matrices in Equation 3 by their low-rank feature products and applying the push-through identity $\Phi_S^\top(\Phi_S\Phi_S^\top + \gamma I_{|S|})^{-1} = (\Phi_S^\top\Phi_S + \gamma I_q)^{-1}\Phi_S^\top$ gives

$$\boxed{f_S^{\mathrm{LRNTK}}(X_T) = \Phi_{\mathrm{te}}\left(\Phi_S^\top\Phi_S + \gamma I_q\right)^{-1}\Phi_S^\top Y_S.} \tag{6}$$

Thus, LRFShap replaces the $|S| \times |S|$ kernel system used by FreeShap with a $q \times q$ feature-space system. In the experiments, we evaluate each nonempty coalition using the accuracy utility

$$U_{\mathrm{acc}}(S) := \frac{1}{|T|}\sum_{j\in T} \mathbf{1}\left[\arg\max_{c\in C}\left[f_S^{\mathrm{LRNTK}}(X_T)\right]_{j,c} = \arg\max_{c\in C}[y_j]_c\right], \qquad S \neq \emptyset,$$

and set $U_{\mathrm{acc}}(\emptyset) := 0$. The feature matrices $(\Phi_{\mathrm{tr}}, \Phi_{\mathrm{te}})$ are constructed once and shared across all coalition evaluations. We then estimate the resulting Shapley values using truncated Monte Carlo (TMC), as in FreeShap. Algorithm 1 summarizes the resulting procedure.

Algorithm 1 LRFShap

Require: Training set D_N, test set D_T, number of MC iterations B, TMC tolerance $\tau_{\mathrm{TMC}} > 0$, precomputed eNTK matrices K_{NN} and K_{TN}, low-rank feature construction $\mathcal{A}$, construction-specific configuration ϑ, regularization $\gamma > 0$

Ensure: Shapley value estimates $\phi_1, \ldots, \phi_{|N|}$

```
Initialize φ_i ← 0 for every i ∈ N
Construct (Φ_tr, Φ_te) ← A(K_NN, K_TN; ϑ)
Evaluate U_acc(N) for the TMC truncation criterion
for b = 1 to B do
    Sample a uniformly random permutation π^b of N
    Set u_0^b ← U_acc(∅) = 0
    for p = 1 to |N| do
        Set S^(b,p) ← {π^b[1], ..., π^b[p]}
        if |U_acc(N) − u_{p−1}^b| < τ_TMC then
            Set u_p^b ← u_{p−1}^b
        else
            Set Φ_{S^(b,p)} ← Φ_tr[S^(b,p), :]
            Compute f^LRNTK_{S^(b,p)}(X_T) using Equation 6
            Evaluate u_p^b ← U_acc(S^(b,p))
        end if
        Update φ_{π^b[p]} ← ((b−1)/b) φ_{π^b[p]} + (1/b)(u_p^b − u_{p−1}^b)
    end for
end for
```

Eigen and Nyström features. We consider two specific choices of the low-rank feature construction $\mathcal{A}$. For eigen features, consider the best rank-r approximation in Equation 4, and assume that the retained eigenvalues are strictly positive. Taking $\vartheta = r$ gives $q = r$, and we define

$$\mathcal{A}^{\mathrm{eig}}(K_{NN}, K_{TN}; r) := \left(U_r \Lambda_r^{1/2},\ K_{TN} U_r \Lambda_r^{-1/2}\right). \tag{7}$$

For large kernel matrices, the leading r eigenvalues can be approximated using randomized subspace iteration (Halko et al., 2011), thereby avoiding a full dense eigendecomposition.

For Nyström features, consider the approximation in Equation 5. Write $K_{MM} = V_q \Sigma_q V_q^\top$ for the eigendecomposition of the landmark kernel matrix, where $q = \mathrm{rank}(K_{MM}) \le m$ and $\Sigma_q \succ 0$. Taking $\vartheta = M$, we define

$$\mathcal{A}^{\mathrm{nys}}(K_{NN}, K_{TN}; M) := \left(K_{NM} V_q \Sigma_q^{-1/2},\ K_{TM} V_q \Sigma_q^{-1/2}\right). \tag{8}$$

Computational complexity. For a prefix $S^{(b,p)}$ of size p, forming $\Phi_{S^{(b,p)}}^\top \Phi_{S^{(b,p)}}$ requires $\mathcal{O}(pq^2)$ operations, while factoring and solving the resulting $q \times q$ system costs $\mathcal{O}(q^3)$. Hence, the coalition-solve cost per prefix is $\mathcal{O}(pq^2 + q^3)$, and summing over all prefixes gives $\mathcal{O}(|N|^2 q^2 + |N| q^3)$ per permutation. Over B independent Monte Carlo permutations, the coalition-solve complexity is therefore $\mathcal{O}\big(B(|N|^2 q^2 + |N| q^3)\big)$. This bound excludes the one-time feature construction and the cost of test prediction and utility evaluation. Under the same direct-factorization convention, FreeShap requires $\mathcal{O}(B|N|^4)$ operations. Therefore, when $q \ll |N|$, LRFShap substantially reduces the dominant coalition-solve cost. A complete complexity analysis is provided in Appendix F.

4 Experimental results

Experimental settings We present our experiments in a Q&A format, first examining which decomposition is more suitable for approximating Data Shapley and then evaluating LRFShap on three downstream tasks: data selection, data removal, and wrong-label detection. We use FreeShap (Wang et al., 2024) as the baseline throughout. Our experiments use BERT and Llama-2 on SST-2, MR, MNLI, QQP, RTE, MRPC, and AG News, and ResNet-18 on CIFAR-10. Both methods use the same TMC estimator (Ghorbani & Zou, 2019) and first-logit eNTK approximation, with $\gamma = 10^{-2}$. All results are averaged over three random seeds. Model identifiers, hyperparameters, hardware specifications, and dataset details are provided in Appendix I.

4.1 RESULTS AND Q&A

Q. Eigen versus Nyström decompositions: which should we choose?

A. In our experiments, eigendecomposition provides more consistent approximations of Data Shapley values.

We compare eigendecomposition with uniform Nyström approximation through the KPS diagram. Figure 2 reports kernel approximation, prediction, and Data Shapley errors for these two methods, together with the distributions of individual Data Shapley errors. For the error curves, we use relative Frobenius errors for the kernel and prediction matrices and relative Euclidean errors for the Data Shapley vectors, normalizing each discrepancy by the norm of the corresponding full-kernel baseline.

We observe that both methods achieve similar prediction errors despite substantial differences in their kernel approximation errors. For eigendecomposition, improvements in kernel approximation are accompanied by decreasing Data Shapley errors. For uniform Nyström approximation, however, the Data Shapley error can plateau or increase at some ranks even as the kernel approximation error decreases. The histograms show that the individual Data Shapley errors are more concentrated around zero for eigendecomposition. Its higher Spearman and Pearson correlations further indicate closer agreement with the baseline, see Appendix G. Besides the uniform Nyström sampling used above, we also tested the leverage-score Nyström method (Alaoui & Mahoney, 2015) and found that the results were similar. For all results of this paragraph on all datasets and models, see Appendix G. Finally, as an ablation study, we show in Appendix H that the Data Shapley error of the Nyström method heavily depends on the choice of landmarks. We therefore conclude that eigendecomposition is more suitable for approximating Data Shapley.

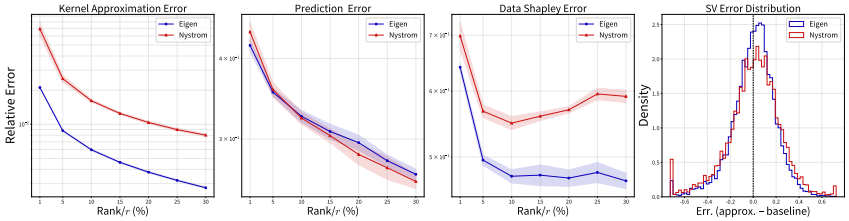

Figure 2: Eigen versus Nyström for BERT on MR at $n = 2000$: kernel approximation, prediction, and Data Shapley errors across ranks, data Shapley error distributions at rank 600 (30% of n). The prediction error can be low even when the kernel approximation error is not, and a low prediction error does not imply a low Data Shapley error; the eigen decomposition attains higher fidelity. In the error distribution graph, errors are clipped by $[-1, 1]$.

Q. Does LRFShap achieve performance comparable to the baseline?

A. Yes. We demonstrate this across three downstream tasks: data selection, data removal, and wrong-label detection.

First, we explain how the downstream tasks are conducted. Given $k \in \{1, \ldots, 100\}$, let $\mathcal{T}(k)$ and $\mathcal{B}(k)$ be the index sets of the training data with the top and bottom $k\%$ Shapley values, respectively, with $\mathcal{T}(0) = \emptyset$. In data selection (removal), we compute the predictor in Equation 3 from $D_{\mathcal{T}(k)}$ ($D_{N \setminus \mathcal{T}(k-1)}$) and report its accuracy on the evaluation set. In wrong label detection, we randomly change the labels of 10% of the training data, compute Data Shapley on this corrupted dataset, and measure the detection rate, the fraction of the wrongly labeled data contained in $D_{\mathcal{B}(k)}$. We compute Data Shapley using the validation set of each dataset as the utility set. For data selection and removal, the accuracy is evaluated on a separate held-out evaluation set, which is taken from the original training split and is disjoint from D_N.

Figure 3 shows the results for BERT on MR. Rather than reporting the performance at each individual percentage, we summarize each downstream task by the mean accuracy (mean detection rate for wrong label detection), defined as the performance averaged over $k \in \{1, 2, \ldots, 20\}$, where k denotes the selection, removal, or inspection percentage of the corresponding task. Since the most valuable or valueless data matter most in practice, we focus on selection, removal, and inspection

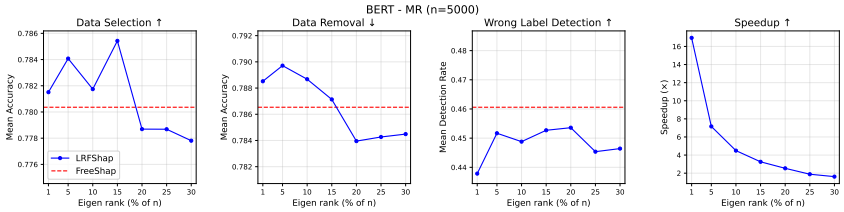

Figure 3: Mean accuracy, detection rate, and speedup of BERT on MR as the rank varies. Higher is better for data selection and wrong label detection; lower is better for data removal. LRFShap achieves performance comparable to FreeShap on all three tasks at $r = 5\%$ while being up to 7.15× faster.

percentages up to 20%. Tables 11–19 in Appendix L report the mean scores for all models and datasets and for figures representing raw accuracy, refer to Appendix K.

We regard LRFShap as comparable to the baseline if its mean accuracy (mean detection rate) over three seeds is worse than that of the baseline by at most 0.01; that is, at most 0.01 lower in data selection and wrong label detection, where higher is better, and at most 0.01 higher in data removal, where lower is better. Across all three tasks, LRFShap with a moderate eigen rank is comparable to or better than the baseline in most cases, which indicates that the top eigen-components capture representative features useful for identifying valuable and wrongly labeled data. Increasing the rank does not consistently improve downstream performance in our experiments. We guess that excessive information is not helpful for distinguishing such data.

Table 1: The largest speedup LRFShap achieves in data selection while remaining comparable to the baseline. For Llama-2, the speedup is small on a few datasets, such as MR, MRPC, QQP, and SST-2, because the convergence ratio of the baseline is low there (see Figure 15 and the third question).

Model	Rank (%) / **Speedup** (×)							
BERT/ResNet-18	AG News	MNLI	MR	MRPC	QQP	RTE	SST-2	CIFAR-10
	1 / **13.16**	1 / **22.32**	1 / **16.79**	1 / **9.81**	5 / **10.77**	5 / **2.39**	1 / **15.20**	1 / **13.23**
Llama-2	AG News	MNLI	MR	MRPC	QQP	RTE	SST-2	-
	1 / **10.90**	5 / **4.07**	1 / **5.07**	1 / **2.10**	1 / **1.41**	1 / **5.72**	1 / **3.75**	-

Q. How much faster is LRFShap when the performance is comparable to the baseline?

A. You can achieve up to 22.32× while maintaining comparable performance.

In Section 3.2, we saw that LRFShap achieves an algorithmic speedup; however, the speedup is meaningful only if a comparable performance is retained. Table 1 reports the largest speedup that LRFShap can achieve in data selection while remaining comparable to the baseline, in the sense defined in the previous question. LRFShap achieves up to a 22.32× speedup. Although the same set of ranks is used for all datasets, the speedup varies across datasets; in particular, it is small on MR, MRPC, QQP, and SST-2 for Llama-2, because the TMC iteration of the baseline terminates early there (see the next question). Tables 8–10 in Appendix J report the speedups for all cases, where we can see that, with too high a rank, LRFShap can even be slower than the baseline; this shows the importance of choosing an appropriate rank.

Q. Is the reduced computational complexity the only source of the speedup?

A. No. The number of coalition evaluations performed before TMC truncation also affects the runtime.

Both methods use the same fixed number of Monte Carlo permutations and the same TMC truncation rule. Within each permutation, coalition evaluations stop once the prefix utility is within the prescribed tolerance of the full-coalition utility, and the remaining samples receive zero marginal contributions. However, FreeShap evaluates utilities using the full kernel, whereas eigen-based LRFShap

uses only the retained eigen-components. This changes both the prefix utilities and the full-coalition utility used as the truncation target, so the two methods can truncate at different coalition sizes.

We quantify this behavior using the **TMC convergence ratio** $R_{\mathrm{TMC}} := \frac{100}{B|N|}\sum_{b=1}^{B}\ell_b\ (\%)$, where B is the number of sampled permutations and $\ell_b \le |N|$ is the number of nonempty prefixes whose utilities are evaluated in the b-th permutation before truncation. For example, $R_{\mathrm{TMC}} = 10\%$ means that TMC truncates each permutation after evaluating, on average, only 10% of its $|N|$ prefixes; the number of permutations B is the same for all methods.

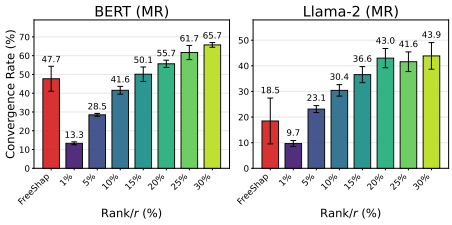

Figure 4: TMC convergence ratio for BERT and Llama-2 on MR.

Figure 4 shows that, at the smallest tested rank, LRFShap truncates at shorter prefixes than FreeShap for both BERT and Llama-2 on MR. In these settings, evaluating fewer coalitions provides additional runtime savings alongside the lower computational cost of each coalition evaluation. The effect of low-rank approximation on truncation depends on the model, dataset, and approximation rank. Additional results are provided in Appendix M.

Q. Does LRFShap work across models and datasets?

A. Yes. LRFShap can be used on diverse models and datasets.

Figures 12–14 in Appendix K show the results for BERT, Llama-2, and ResNet-18 on the diverse datasets. We observe the same tendency as in these figures: with a moderate rank, LRFShap is comparable to the baseline, and with a high rank, it surpasses the baseline in some cases. These results demonstrate the effectiveness of LRFShap across the evaluated models and datasets, with suitable ranks varying by setting.

5 Conclusion

We studied how low-rank kernel approximation affects individual Data Shapley values. In regularized kernel regression, we established stability bounds for finite-time training and kernel approximation, while showing that grand-coalition predictive agreement alone does not guarantee preservation of individual data values. Our counterexample for uniform Nyström sampling makes this distinction explicit, and our empirical comparisons show that eigendecomposition provides more faithful Shapley approximations in the settings studied. Guided by this analysis, we proposed LRFShap, which accelerates coalition evaluation by reusing low-dimensional kernel features. Experiments demonstrate its effectiveness for eNTK-based data valuation, where kernel regression serves as a surrogate for model fine-tuning. With suitable ranks, LRFShap maintains performance comparable to FreeShap on data selection, data removal, and wrong-label detection while substantially reducing valuation time. These results emphasize that low-rank approximations for data valuation should be assessed beyond their predictive performance on the full dataset.

Limitations and future work. The rank needed to retain downstream performance varies across datasets and valuation tasks, motivating principled methods for rank selection. Our theoretical guarantees concern squared-error utilities in regularized kernel regression with a fixed test–train kernel block. Extending these guarantees to classification utilities and joint approximations of the training and test–train kernel blocks remains an important next step. Moreover, the fixed-kernel formulation does not directly capture changes in feature representations during fine-tuning, motivating extensions to evolving kernels. Further directions include analyzing low-rank approximation in distributional Shapley (Ghorbani et al., 2020) and MMD-based data distribution valuation (Xu et al., 2024).

AI use statement

We used generative AI tools to assist with drafting and polishing the manuscript, identifying relevant literature, exploring research ideas, and developing mathematical arguments and proofs. The authors reviewed and revised all AI-assisted content, checked cited sources, and independently verified the

mathematical arguments included in the paper. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

Reproducibility statement

We describe the kernel regression setting and the LRFShap algorithm in the main text. The appendix provides the assumptions and proofs for our theoretical results. The experimental setup and supplementary details specify the datasets, pretrained models, kernel construction, approximation ranks, regularization parameters, and evaluation procedures used in our experiments.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *Microsoft tech blog*, 2024. URL https://www.microsoft.com/en-us/research/publication/phi-4-technical-report/.

Ahmed Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in Neural Information Processing Systems*, 2015.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. *International Conference on Learning Representations*, 2024.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 2019.

Francis Bach. Sharp analysis of low-rank kernel matrix approximations. *Conference on Learning Theory*, 2013.

Mark Alexander Burgess and Archie C Chapman. Approximating the shapley value using stratified empirical bernstein sampling. *International Joint Conferences on Artificial Intelligence*, 2021.

Yuanyuan Chen, Boyang Li, Han Yu, Pengcheng Wu, and Chunyan Miao. Hydra: Hypergradient data relevance analysis for interpreting deep neural networks. *Association for the Advancement of Artificial Intelligence*, 2021.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. *International Conference on Artificial Intelligence and Statistics*, 2010.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. *International Workshop on Paraphrasing*, 2005.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 2005.

Carl Eckart and Gale Young. *The approximation of one matrix by another of lower rank*. Psychometrika, 1936.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. *International Conference on Machine Learning*, 2019.

Amirata Ghorbani, Michael Kim, and James Zou. A distributional framework for data valuation. *International Conference on Machine Learning*, 2020.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

Moritz Haas, David Holzmüller, Ulrike Luxburg, and Ingo Steinwart. Mind the spikes: Benign overfitting of kernels and neural networks in fixed dimension. *Advances in Neural Information Processing Systems*, 2023.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE conference on Computer Vision and Pattern Recognition*, 2016.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 2018.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. *International Conference on Artificial Intelligence and Statistics*, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *International conference on machine learning*, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *International Conference on Artificial Intelligence and Statistics*, 2022.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in Neural Information Processing Systems*, 2019.

John M Lee. *Introduction to smooth manifolds*. Springer, 2013.

Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast dpp sampling for nystrom with application to kernel methods. *International Conference on Machine Learning*, 2016.

Weida Li and Yaoliang Yu. Faster approximation of probabilistic and distributional values via least squares. *International Conference on Learning Representations*, 2024.

Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv:1306.4265*, 2013.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. *International Conference on Machine Learning*, 2023.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for Shapley value estimation. *Journal of Machine Learning Research*, 2022.

Mohamad Amin Mohamadi, Wonho Bae, and Danica J Sutherland. A fast, well-founded approximation to the empirical neural tangent kernel. *International Conference on Machine Learning*, 2023.

Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. *Advances in Neural Information Processing Systems*, 2017.

Ramin Okhrati and Aldo Lipani. A multilinear sampling algorithm to estimate shapley values. *International Conference on Pattern Recognition*, 2021.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Annual Meeting of the Association for Computational Linguistics*, 2005.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 2024.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 2020.

Lloyd S. Shapley. *A value for n-person games*. Princeton University Press, 1953.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 2024.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Conference on Empirical Methods in Natural Language Processing*, 2013.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *International Conference on Learning Representations*, 2019.

Jiachen Tianhao Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. *International Conference on Learning Representations*, 2025.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. *International Conference on Machine Learning*, 2024.

Shusen Wang and Zhihua Zhang. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 2013.

Alexander Wei, Wei Hu, and Jacob Steinhardt. More than a toy: Random matrix models predict how real-world neural representations generalize. *International Conference on Machine Learning*, 2022.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *Conference of the North American Chapter of the Association for Computational Linguistics*, 2018.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 2000.

Zhaoxuan Wu, Yao Shu, and Bryan Kian Hsiang Low. Davinz: Data valuation using deep neural networks at initialization. *International Conference on Machine Learning*, 2022.

Xinyi Xu, Shuaiqi Wang, Chuan-Sheng Foo, Bryan K Low, and Giulia Fanti. Data distribution valuation. *Advances in Neural Information Processing Systems*, 2024.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems*, 2012.

Kai Zhang, Ivor W Tsang, and James T Kwok. Improved nyström low-rank approximation and error analysis. *International Conference on Machine Learning*, 2008.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 2015.

APPENDIX CONTENTS

A Related work **14**

B Neural Tangent Kernel **15**

C Solutions by kernel regression **16**

C.1 Spectral analysis for the finite-time error of gradient flow 16

D Theoretical understanding **18**

D.1 Proof of Proposition 3.1 18

D.2 Proof of Proposition 3.2 19

D.3 Proof of Proposition 3.3 21

D.4 proof of Corollary 3.1 23

E Empirical Analysis of the Gap Between Proposition 3.2 and the Implemented Predictor **26**

F Computational complexity of FreeShap and LRFShap **29**

G KPS errors and fidelity scores **31**

H Sensitivity of the Nyström Method to the Choice of Landmarks **33**

I Experimental details **34**

J Computation time **35**

K Raw accuracies **37**

L Mean accuracy results **38**

M TMC convergence ratios for all cases **40**

A Related work

Reducing the combinatorial cost of Data Shapley. Data Shapley (Ghorbani & Zou, 2019; Jia et al., 2019) formulates data valuation as a cooperative game, treating each training sample as a player and using the Shapley value to quantify its contribution. To mitigate the combinatorial cost of Shapley value computation, existing methods improve estimation efficiency through group-testing-based estimation (Jia et al., 2019), multilinear-extension-based sampling (Okhrati & Lipani, 2021), adaptive stratified sampling (Burgess & Chapman, 2021), and structured permutation sampling (Mitchell et al., 2022). Li & Yu (2024) develop least-squares-based estimators for probabilistic values, including the Shapley value, providing another approach to efficient data valuation.

Reducing model retraining costs. Classical leave-one-out (LOO) attribution evaluates marginal contributions to the remaining training set, whereas Data Shapley averages them across all subsets of that set. LOO scores can have low signal-to-noise ratios and unstable signs under dataset resampling (Kwon & Zou, 2022; Wang et al., 2024). Influence functions (Koh & Liang, 2017) approximate LOO attribution. Dynamic methods such as TracIn (Pruthi et al., 2020) and HyDRA (Chen et al., 2021) estimate data influence along training trajectories, while In-Run Data Shapley (Wang et al., 2025) defines local Shapley games within a single training run. Among NTK-based approaches, DAVINZ (Wu et al., 2022) uses a generalization bound incorporating distributional differences as its utility. FreeShap (Wang et al., 2024) approximates coalition-wise fine-tuning through empirical NTK regression and evaluates utility using predictive accuracy. We adopt FreeShap as our baseline because its kernel regression formulation facilitates low-rank analysis of Data Shapley and supports attribution for large pretrained models.

Low-rank approximation of kernel matrices. Classical approaches include truncated eigendecomposition (Eckart & Young, 1936) and Nyström approximation (Williams & Seeger, 2000; Drineas & Mahoney, 2005). CUR decomposition generalizes the Nyström framework to arbitrary matrices through column and row selection; see Wang & Zhang (2013) for an overview of these methods and their variants. Landmark-selection strategies include clustering-based selection (Zhang et al., 2008), ridge leverage-score sampling (Alaoui & Mahoney, 2015; Musco & Musco, 2017), and determinantal point process sampling (Li et al., 2016). The relationship between kernel approximation and predictive performance has also been studied. Yang et al. (2012) compare Nyström and random Fourier features, showing that Nyström can achieve sharper generalization bounds under suitable spectral conditions. Bach (2013) directly analyzes the fixed-design prediction risk of Nyström kernel ridge regression, obtaining relative-risk guarantees without relying on a small kernel-matrix approximation error. Our work investigates how kernel approximation affects individual Data Shapley values and whether predictive agreement suffices to preserve these values.

B NEURAL TANGENT KERNEL

We consider a parametric model $f(x;\theta) : \mathbb{R}^d \to \mathbb{R}^{|C|}$ that is differentiable with respect to the parameter vector $\theta \in \mathbb{R}^P$. Let $\theta_0 \in \mathbb{R}^P$ denote an initialization of the model parameters. The empirical neural tangent kernel (eNTK) associated with the model at θ_0 is defined, for any pair of inputs $x_1, x_2 \in \mathbb{R}^d$, as

$$\Theta_{\theta_0}(x_1, x_2) := \underbrace{\left[\frac{\partial f(x_1;\theta_0)}{\partial \theta_0}\right]}_{|C|\times P} \underbrace{\left[\frac{\partial f(x_2;\theta_0)}{\partial \theta_0}\right]^\top}_{P\times|C|} \in \mathbb{R}^{|C|\times|C|}. \tag{9}$$

Here, $\frac{\partial f(x;\theta_0)}{\partial \theta_0} \in \mathbb{R}^{|C|\times P}$ denotes the Jacobian of the model output with respect to the flattened parameter vector θ_0.

In the *kernel regime*, the network parameters remain sufficiently close to their initialization throughout training, so that the network is well approximated by its first-order linearization $f(x;\theta_t) \approx f(x;\theta_0) + \frac{\partial f(x;\theta_0)}{\partial \theta_0}(\theta_t - \theta_0)$ for every $t \geq 0$. At the same time, the empirical neural tangent kernel remains approximately constant along the training trajectory: $\Theta_{\theta_t}(x_1, x_2) \approx \Theta_{\theta_0}(x_1, x_2)$ for every $x_1, x_2 \in \mathbb{R}^d$. Consequently, the evolution of the network outputs is governed by the fixed kernel determined at initialization. Under the kernel approximation, parameter-space gradient flow reduces to kernel gradient flow in function space with respect to the fixed eNTK Θ_{θ_0}. For the squared loss, its infinite-time limit corresponds to kernel regression associated with Θ_{θ_0}.

Let the training dataset be denoted by D_S with index set $S := \{1, \ldots, |S|\}$. We define the input matrix as $X_S := [x_1, \ldots, x_{|S|}]^\top$ and the corresponding one-hot encoded label matrix as $Y_S := [y_1, \ldots, y_{|S|}]^\top \in \{0,1\}^{|S|\times|C|}$. We further use $\mathrm{vec}(Y_S) \in \{0,1\}^{|S||C|}$ to denote the vectorized form of the label matrix.

Given a test input x_t, the prediction of the eNTK regression model is given by

$$f_S^{\mathrm{eNTK}}(x_t) = \underbrace{f(x_t;\theta_0)}_{|C|\times 1} + \underbrace{K(x_t, X_S)^\top}_{|C|\times|S||C|} \underbrace{K(X_S, X_S)^\dagger}_{|S||C|\times|S||C|} \underbrace{\mathrm{vec}(Y_S - f(X_S;\theta_0))}_{|S||C|\times 1}, \tag{10}$$

where $K(x_t, X_S) = \left[\Theta_{\theta_0}(x_t, x_j)\right]_{j=1}^{|S|}$ and $K(X_S, X_S) = \left[\Theta_{\theta_0}(x_i, x_j)\right]_{i,j=1}^{|S|}$.

We make two remarks regarding the practical use of the empirical neural tangent kernel.

First, one may ask whether it is possible to use pretrained weights, rather than random initialization, as the reference parameter θ_0 for defining the empirical neural tangent kernel. In the context of natural language processing, this question was investigated by Malladi et al. (2023), while related results for computer vision tasks were reported in Wei et al. (2022).

Second, directly inverting the kernel matrix $K(X_S, X_S)$ in equation 10 requires $\mathcal{O}((|S||C|)^3)$ computational cost, which becomes prohibitive when the number of classes $|C|$ is large. To mitigate this class-dependent inefficiency, various approximation techniques have been proposed, including the first-logit approximation.

From a theoretical perspective, Mohamadi et al. (2023) provides a principled justification for such approximations. Specifically, they consider a multi-output model $f(x;\theta) = [f^1(x;\theta), \ldots, f^{|C|}(x;\theta)]$ and introduces a pseudo neural tangent kernel (pNTK) defined by

$$\hat{\Theta}_{\theta_0}(x_1, x_2) := \underbrace{\left[\frac{\partial}{\partial \theta_0}\left(\frac{1}{\sqrt{|C|}}\sum_{i=1}^{|C|} f^i(x_1;\theta_0)\right)\right]}_{1\times P} \underbrace{\left[\frac{\partial}{\partial \theta_0}\left(\frac{1}{\sqrt{|C|}}\sum_{i=1}^{|C|} f^i(x_2;\theta_0)\right)\right]^\top}_{P\times 1}. \tag{11}$$

Using this kernel, they show that the regression predictor in equation 10 admits the following closed-form approximation:

$$f_S^{\mathrm{pNTK}}(x_t) = \underbrace{f(x_t;\theta_0)}_{1\times|C|} + \underbrace{\hat{K}(x_t, X_S)^\top}_{1\times|S|} \underbrace{\hat{K}(X_S, X_S)^\dagger}_{|S|\times|S|} \underbrace{(Y_S - f(X_S;\theta_0))}_{|S|\times|C|}, \tag{12}$$

where $\hat{K}(x_t, X_S) = \big[\hat{\Theta}_{\theta_0}(x_t, x_j)\big]_{j=1}^{|S|} \in \mathbb{R}^{|S|}$ and $\hat{K}(X_S, X_S) = \big[\hat{\Theta}_{\theta_0}(x_i, x_j)\big]_{i,j=1}^{|S|} \in \mathbb{R}^{|S|\times|S|}$.

Importantly, this approximation has been shown empirically to remain effective when θ_0 is chosen as pretrained weights, rather than random initialization. To facilitate subsequent analysis, we rewrite equation 12 in vectorized form over a test set X_T and introduce an explicit ridge regularization parameter $\gamma > 0$. This yields

$$F_S^{\mathrm{pNTK}}(X_T) = \underbrace{f(X_T;\theta_0)}_{|T|\times|C|} + \underbrace{\hat{K}(X_T, X_S)^\top}_{|T|\times|S|} \underbrace{(\hat{K}(X_S, X_S) + \gamma I_{|S|})^{-1}}_{|S|\times|S|} \underbrace{(Y_S - f(X_S;\theta_0))}_{|S|\times|C|}. \quad (13)$$

Define the residual targets $\tilde{Y}_S := Y_S - f(X_S;\theta_0)$ and the residual predictor $\tilde{F}_S^{\mathrm{pNTK}}(X_T) := F_S^{\mathrm{pNTK}}(X_T) - f(X_T;\theta_0)$. Then equation 13 can be equivalently expressed as

$$\tilde{F}_S^{\mathrm{pNTK}}(X_T) = k(X_T, X_S)^\top \big(k(X_S, X_S) + \gamma I_{|S|}\big)^{-1} \tilde{Y}_S, \quad (14)$$

where we write $k = \hat{K}$ for simplicity.

At independent Gaussian initialization, the first-logit and sum-of-logits scalarizations agree in distribution. Consequently, the random-initialization approximation guarantee of Mohamadi et al. (2023) applies to both constructions. In this work, however, we use the first-logit kernel evaluated at pretrained parameters; we do not assert equality between the two kernels for a fixed pretrained network.

C Solutions by kernel regression

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive semi-definite kernel function, and let $\mathcal{H}_k$ denote the associated reproducing kernel Hilbert space (RKHS). Given a training dataset $D_S = \{(x_i, y_i)\}_{i=1}^{|S|}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we consider the following empirical risk functional:

$$\mathcal{L}_\rho^S(f) := \frac{1}{|S|} \sum_{i=1}^{|S|} \big(f(x_i) - y_i\big)^2 + \rho \|f\|_{\mathcal{H}_k}^2 \quad (15)$$

where $\rho \geq 0$ is a regularization parameter.

Let $f_{t,\rho}^S$ denote the function obtained by running gradient flow on the regularized objective $\mathcal{L}_\rho^S$ in the RKHS $\mathcal{H}_k$, initialized at $f_0 = 0$ and evolved up to time $t \in [0, \infty)$ with a fixed regularization parameter $\rho > 0$. According to Appendix C of Haas et al. (2023), the solution admits the explicit representation

$$f_{t,\rho}^S(x) = k(x, X_S)\Big(I_{|S|} - e^{-\frac{2}{|S|}t\left(k(X_S,X_S)+\rho|S|I_{|S|}\right)}\Big)\big(k(X_S, X_S) + \rho|S|I_{|S|}\big)^{-1} Y_S, \quad (16)$$

where $k(x, X_S) := [k(x, x_i)]_{i\in S}$ and $k(X_S, X_S) := [k(x_i, x_j)]_{i,j\in S}$ denotes the kernel matrix.

This expression provides a unified perspective on several classical kernel estimators:

1. kernel ridge regression in the limit $t \to \infty$;
2. gradient flow on the unregularized objective as $\rho \downarrow 0$;
3. ridgeless kernel regression $f_{\infty,0}(x) = k(x, X_S)\, k(X_S, X_S)^\dagger y$ in the joint limit $\rho \to 0$ and $t \to \infty$.

C.1 Spectral analysis for the finite-time error of gradient flow

Our goal is to analyze the finite-time error $\|f_{\infty,\rho}^S - f_{t,\rho}^S\|_{\mathcal{H}_k}^2$ of the gradient flow solution for each subset $S \subseteq N$. To this end, we express $f_{t,\rho}^S(x) = k(x, X_S)\, \alpha_{t,\rho}^S$ where the coefficient vector $\alpha_{t,\rho}^S \in \mathbb{R}^{|S|}$ is given by

$$\alpha_{t,\rho}^S = \Big(I_{|S|} - e^{-\frac{2}{|S|}t\left(k(X_S,X_S)+\rho|S|I_{|S|}\right)}\Big)\big(k(X_S, X_S) + \rho|S|I_{|S|}\big)^{-1} Y_S. \quad (17)$$

Then, the limiting coefficient vector is $\alpha^S_{\infty,\rho} = \big(k(X_S, X_S) + \rho|S|I_{|S|}\big)^{-1} Y_S$, and the corresponding finite-time residual is $\Delta\alpha^S_\rho := \alpha^S_{t,\rho} - \alpha^S_{\infty,\rho} = -e^{-\frac{2}{|S|}t\left(k(X_S,X_S)+\rho|S|I_{|S|}\right)}\big(k(X_S, X_S) + \rho|S|I_{|S|}\big)^{-1} Y_S$. From this, since $f^S_{\infty,\rho} - f^S_{t,\rho} = k(\cdot, X_S)\,\Delta\alpha^S_\rho$, the RKHS norm satisfies

$$\|f^S_{\infty,\rho} - f^S_{t,\rho}\|^2_{\mathcal{H}} = (\Delta\alpha^S_\rho)^\top k(X_S, X_S)(\Delta\alpha^S_\rho).$$

Using the eigen decomposition we have $k(X_S, X_S) = V_S \Lambda_S V_S^\top$ where $\Lambda_S = \mathrm{diag}(\lambda^S_1, \ldots, \lambda^S_{|S|})$ is diagonal and $V_S \in \mathbb{R}^{|S|\times|S|}$ is orthonormal. Then we can represent the residual as

$$\Delta\alpha^S_\rho = V_S e^{-\frac{2}{|S|}t(\Lambda_S+\rho|S|I_{|S|})}(\Lambda_S + \rho|S|I_{|S|})^{-1} V_S^\top Y_S.$$

Defining rotated coordinates $\tilde{Y}_S := V_S^\top Y_S$, a direct computation yields the exact spectral identity

$$\|f^S_{\infty,\rho} - f^S_{t,\rho}\|^2_{\mathcal{H}} = (V_S^\top \Delta\alpha^S_\rho)^\top \Lambda_S (V_S^\top \Delta\alpha^S_\rho) \tag{18}$$

$$= \sum_{i=1}^{|S|} \lambda_i^{|S|} (V_S^\top \Delta\alpha^S_\rho)^2_i = \sum_{i=1}^{|S|} \lambda^S_i \left(\frac{e^{-\frac{2t}{|S|}(\lambda^S_i+\rho|S|)}}{\lambda^S_i + \rho|S|}\right)^2 (\tilde{Y}_{S,i})^2. \tag{19}$$

Case 1: $\rho = 0$ (ridgeless). When $\rho = 0$, only the strictly positive eigenvalues of $k(X_S, X_S)$ contribute:

$$\|f^S_{\infty,0} - f^S_{t,0}\|^2_{\mathcal{H}_k} = \sum_{i:\lambda^S_i>0} \frac{e^{-\frac{4t}{|S|}\lambda^S_i}}{\lambda^S_i} (\tilde{Y}_{S,i})^2.$$

Since $e^{-x} \le \frac{1}{x}$ for $x > 0$, we have

$$\|f^S_{\infty,0} - f^S_{t,0}\|^2_{\mathcal{H}_k} \le \frac{|S|}{4t}\|k(X_S, X_S)^\dagger Y_S\|_2^2 \le \frac{|S|\|Y_S\|_2^2}{4t(\lambda^+_{\min}(k(X_S, X_S)))^2}, \tag{20}$$

where $\lambda^+_{\min}(k(X_S, X_S)) := \min\{\lambda^S_i : \lambda^S_i > 0\}$.

Case 2: $\rho > 0$ (regularized). When $\rho > 0$, we have $\lambda^S_i + \rho|S| \ge \rho|S| > 0$. Hence from equation 19, we have

$$\|f^S_{\infty,\rho} - f^S_{t,\rho}\|^2_{\mathcal{H}_k} \le e^{-4t\rho} \sum_{i=1}^{|S|} \frac{(\tilde{Y}_{S,i})^2}{\lambda^S_i + \rho|S|} = Y_S^\top (k(X_S, X_S) + \rho|S|I_{|S|})^{-1} Y_S \le \frac{e^{-4t\rho}}{\rho|S|}\|Y_S\|_2^2. \tag{21}$$

D THEORETICAL UNDERSTANDING

This appendix follows the scalar-response setting of Section 3.1, with notation summarized in Table 2. The notation below is recalled only for convenience; no new assumptions are introduced. When the training kernel matrix varies, Y_N, Y_T, and K_{TN} remain fixed. Thus, in the kernel-approximation analysis, only the train–train kernel matrix is perturbed, while the test–train kernel blocks remain unchanged.

Table 2: Notation used throughout Appendix C.

Symbol	Meaning in Appendix C
N, T, S	Nonempty finite training- and test-index sets and a coalition $S \subseteq N$.
D_S, X_S, Y_S	The coalition dataset, its input matrix, and its scalar-response vector. In particular, $Y_S \in \mathbb{R}^{\lvert S\rvert}$.
y_i, κ	Scalar responses satisfying $\lvert y_i\rvert \le 1$, and the kernel bound $\kappa := \sup_{x\in\mathbb{R}^d} \sqrt{k(x,x)} < \infty$.
$k, \mathcal{H}_k, K_{AB}$	A fixed scalar-valued positive-semidefinite kernel, its RKHS, and the kernel matrix indexed by $A, B \subseteq N \cup T$.
K_{NN}, L_{NN}, G	The exact training kernel matrix, its positive-semidefinite approximation, and a generic training-matrix argument, respectively.
$\rho, t, A_{S,\rho}(G)$	The theoretical regularization parameter, gradient-flow time, and the regularized coalition matrix $A_{S,\rho}(G) := G_{SS} + \rho\lvert S\rvert I_{\lvert S\rvert}$. The main propositions assume $\rho > 0$; the case $\rho = 0$ is considered only where explicitly stated.
$\mathcal{P}^S_{t,\rho}(G), \mathcal{P}^S_\rho(G), \mathcal{P}_\rho(G)$	The finite-time coalition prediction, stationary coalition prediction, and grand-coalition prediction, respectively, with $\mathcal{P}^S_\rho(G) := \mathcal{P}^S_{\infty,\rho}(G)$, $\mathcal{P}_\rho(G) := \mathcal{P}^N_\rho(G)$.
$U^G_{t,\rho}(S)$	The negative squared-test-error utility $U^G_{t,\rho}(S) := -\lvert T\rvert^{-1}\lVert Y_T - \mathcal{P}^S_{t,\rho}(G)\rVert_2^2$.
$\mathcal{S}_{t,\rho}(G), \mathcal{S}_\rho(G)$	The finite-time and stationary Data Shapley maps, with $\mathcal{S}_{t,\rho}(G) := (\phi_i(U^G_{t,\rho}))_{i\in N}$ and $\mathcal{S}_\rho := \mathcal{S}_{\infty,\rho}$.

For the empty coalition, we use the conventions $\mathcal{P}^\emptyset_{t,\rho}(G) = \mathcal{P}^\emptyset_\rho(G) = 0$, and $U^G_{t,\rho}(\emptyset) = U^G_{\infty,\rho}(\emptyset) = -\frac{1}{\lvert T\rvert}\lVert Y_T\rVert_2^2$. The implementation parameter γ is a coalition-independent ridge parameter and is distinct from the size-normalized theoretical parameter ρ. Symbols used only within one proof are introduced locally when they first appear.

Lemma D.1 (Size-dependent perturbation bound for Shapley value)**.** *Let $N = \{1, \dots, n\}$ be the player set, and let $U, V : 2^N \to \mathbb{R}$ be two cooperative games. Assume that there exists a function $g : \{0, 1, \dots, n\} \to \mathbb{R}_{\ge 0}$ such that for every $l = 0, \dots, n$ and every $S \subseteq N$ with $\lvert S\rvert = l$, $\lvert U(S) - V(S)\rvert \le g(l)$. Then for every $i \in N$,*

$$\lvert\phi_i(U) - \phi_i(V)\rvert \le \frac{1}{n}\Big(g(0) + g(n) + 2\sum_{l=1}^{n-1} g(l)\Big).$$

The above result follows directly from the size-grouped representation of the Shapley value equation 1 together with the additivity axiom of the Shapley value. Moreover, if $g(0) = 0$, we obtain the following simplified bound: $\lvert\phi_i(U) - \phi_i(V)\rvert \le \frac{2}{n}\sum_{l=1}^{n} g(l)$. In particular, if the perturbation bound does not depend on the coalition size, that is, if there exists a constant $\varepsilon \ge 0$ such that $\lvert U(S) - V(S)\rvert \le \varepsilon$ for all $S \subseteq N$, then taking $g(l) \equiv \varepsilon$ yields $\lvert\phi_i(U) - \phi_i(V)\rvert \le 2\varepsilon$, which coincides with the bound obtained by directly controlling the supremum $\sup_{S\subseteq N} \lvert U(S) - V(S)\rvert$.

D.1 PROOF OF PROPOSITION 3.1

Proof. Fix a nonempty coalition $S \subseteq N$. Let $f^S_{t,\rho} \in \mathcal{H}_k$ denote the zero-initialized gradient-flow solution associated with D_S, and let $f^S_{\infty,\rho}$ denote its stationary limit. By the definitions in Appendix C, $\mathcal{P}^S_{t,\rho}(K_{NN}) = f^S_{t,\rho}(X_T)$ and $\mathcal{P}^S_\rho(K_{NN}) = f^S_{\infty,\rho}(X_T)$.

For every $f \in \mathcal{H}_k$, the reproducing property and the definition of κ give

$$\|f(X_T)\|_2^2 = \sum_{i \in T} |f(x_i)|^2 \le \|f\|_{\mathcal{H}_k}^2 \operatorname{tr}(K_{TT}) \le \kappa^2 |T| \|f\|_{\mathcal{H}_k}^2.$$

By Proposition 2 of Haas et al. (2023), $\|f_{t,\rho}^S\|_{\mathcal{H}_k} \le \|f_{\infty,\rho}^S\|_{\mathcal{H}_k} \le \|f_{\infty,0}^S\|_{\mathcal{H}_k}$. Since $f_{\infty,\rho}^S$ minimizes the regularized risk in Equation equation 15, comparison with the zero function gives $\rho \|f_{\infty,\rho}^S\|_{\mathcal{H}_k}^2 \le \mathcal{L}_\rho^S(f_{\infty,\rho}^S) \le \mathcal{L}_\rho^S(0) = |S|^{-1}\|Y_S\|_2^2 \le 1$. Consequently, $\|f_{t,\rho}^S\|_{\mathcal{H}_k} \le \|f_{\infty,\rho}^S\|_{\mathcal{H}_k} \le \rho^{-1/2}$.

Moreover, the spectral decay estimate in Equation equation 21 and the bound $\|Y_S\|_2 \le \sqrt{|S|}$ imply $\|f_{t,\rho}^S - f_{\infty,\rho}^S\|_{\mathcal{H}_k} \le \rho^{-1/2} e^{-2\rho t}$. Therefore,

$$\begin{aligned}
&\left| U_{t,\rho}^{K_{NN}}(S) - U_{\infty,\rho}^{K_{NN}}(S) \right| \\
&\le \frac{1}{|T|} \left(\left\| Y_T - \mathcal{P}_{t,\rho}^S(K_{NN}) \right\|_2 + \left\| Y_T - \mathcal{P}_\rho^S(K_{NN}) \right\|_2 \right) \left\| \mathcal{P}_{t,\rho}^S(K_{NN}) - \mathcal{P}_\rho^S(K_{NN}) \right\|_2 \\
&\le 2\kappa \left(1 + \kappa \|f_{\infty,\rho}^S\|_{\mathcal{H}_k} \right) \left\| f_{t,\rho}^S - f_{\infty,\rho}^S \right\|_{\mathcal{H}_k} \\
&\le \frac{2\kappa}{\sqrt{\rho}} \left(1 + \frac{\kappa}{\sqrt{\rho}} \right) e^{-2\rho t} =: \varepsilon_t.
\end{aligned}$$

Together with the standing convention for the empty coalition, the preceding estimate shows that the utility difference is bounded by $g(|S|)$, where $g(0) := 0$ and $g(\ell) := \varepsilon_t$ for $1 \le \ell \le |N|$. Applying Lemma D.1 gives $\left| \phi_i(U_{t,\rho}^{K_{NN}}) - \phi_i(U_{\infty,\rho}^{K_{NN}}) \right| \le 2\varepsilon_t$ for every $i \in N$. Taking the maximum over $i \in N$, we obtain

$$\|\mathcal{S}_{t,\rho}(K_{NN}) - \mathcal{S}_\rho(K_{NN})\|_\infty \le C_{\kappa,\rho} e^{-2\rho t}, \qquad C_{\kappa,\rho} := \frac{4\kappa}{\sqrt{\rho}} \left(1 + \frac{\kappa}{\sqrt{\rho}} \right).$$

□

Ridgeless setting. When $\rho = 0$, the preceding utility estimate together with equation 20 yields, for every nonempty coalition $S \subseteq N$ and every $t > 0$,

$$\left| U_{t,0}^{K_{NN}}(S) - U_{\infty,0}^{K_{NN}}(S) \right| \le \kappa \left(1 + \kappa \left\| f_{\infty,0}^S \right\|_{\mathcal{H}_k} \right) \frac{\sqrt{|S|}\ \|Y_S\|_2}{\sqrt{t}\, \lambda_{\min}^+(K_{SS})}.$$

However, $\lambda_{\min}^+(K_{SS})$ can be arbitrarily small as the dataset and coalition vary. Consequently, because Data Shapley depends on all coalitions $S \subseteq N$, the coalition-wise estimate above does not yield a uniform stability bound for the ridgeless gradient flow without an additional spectral-gap assumption.

D.2 Proof of Proposition 3.2

Proof. Fix a nonempty coalition $S \subseteq N$. Recall that, for any training-matrix G, $A_{S,\rho}(G) = G_{SS} + \rho|S| I_{|S|}$, $\mathcal{P}_\rho^S(G) = K_{TS} A_{S,\rho}(G)^{-1} Y_S$, and $U_{\infty,\rho}^G(S) = -|T|^{-1}\|Y_T - \mathcal{P}_\rho^S(G)\|_2^2$.

Since K_{NN} and L_{NN} are positive semidefinite, for every $G \in \{K_{NN}, L_{NN}\}$, we have $A_{S,\rho}(G) \succeq \rho|S| I_{|S|} \succ 0$, and hence $\|A_{S,\rho}(G)^{-1}\|_2 \le (\rho|S|)^{-1}$. By the definition of κ, $\|K_{TS}\|_2 \le \|K_{TS}\|_{\mathrm{F}} \le \kappa^2 \sqrt{|T||S|}$. Together with $\|Y_T\|_2 \le \sqrt{|T|}$ and $\|Y_S\|_2 \le \sqrt{|S|}$, this gives, for $G \in \{K_{NN}, L_{NN}\}$,

$$\left\| \mathcal{P}_\rho^S(G) \right\|_2 \le \frac{\kappa^2}{\rho} \sqrt{|T|}, \qquad \left\| Y_T - \mathcal{P}_\rho^S(G) \right\|_2 \le \left(1 + \frac{\kappa^2}{\rho} \right) \sqrt{|T|}.$$

We first bound the prediction difference. Since $K_{SS} - L_{SS}$ is a principal submatrix of $K_{NN} - L_{NN}$, we have $\|K_{SS} - L_{SS}\|_2 \le \|K_{NN} - L_{NN}\|_2$. Combining this observation with the resolvent identity $A^{-1} - B^{-1} = A^{-1}(B - A)B^{-1}$ and the preceding bounds, we obtain

$$\begin{aligned}
&\left\| \mathcal{P}_\rho^S(K_{NN}) - \mathcal{P}_\rho^S(L_{NN}) \right\|_2 \\
&\le \|K_{TS}\|_2 \|A_{S,\rho}(K_{NN})^{-1}\|_2 \|K_{SS} - L_{SS}\|_2 \|A_{S,\rho}(L_{NN})^{-1}\|_2 \|Y_S\|_2 \\
&\le \frac{\kappa^2 \sqrt{|T|}}{\rho^2 |S|} \|K_{NN} - L_{NN}\|_2.
\end{aligned}$$

We next bound the corresponding utility difference. Applying $|\|a\|_2^2 - \|b\|_2^2| \le (\|a\|_2 + \|b\|_2)\|a-b\|_2$ and using the preceding prediction and residual bounds yields

$$\begin{aligned}
&\left|U^{K_{NN}}_{\infty,\rho}(S) - U^{L_{NN}}_{\infty,\rho}(S)\right| \\
&\le \frac{1}{|T|}\left(\left\|Y_T - \mathcal{P}^S_\rho(K_{NN})\right\|_2 + \left\|Y_T - \mathcal{P}^S_\rho(L_{NN})\right\|_2\right)\left\|\mathcal{P}^S_\rho(K_{NN}) - \mathcal{P}^S_\rho(L_{NN})\right\|_2 \\
&\le 2\kappa^2\rho^{-2}\left(1+\frac{\kappa^2}{\rho}\right)\frac{\|K_{NN}-L_{NN}\|_2}{|S|}.
\end{aligned}$$

For $S = \emptyset$, the utility difference is zero by the standing convention. Applying Lemma D.1 to the preceding estimate and using $\sum_{\ell=1}^{|N|}\ell^{-1} \le 1 + \log|N|$, we obtain, for every $i \in N$,

$$\begin{aligned}
&\left|\phi_i\big(U^{K_{NN}}_{\infty,\rho}\big) - \phi_i\big(U^{L_{NN}}_{\infty,\rho}\big)\right| \\
&\le \frac{4\kappa^2}{\rho^2|N|}\left(1+\frac{\kappa^2}{\rho}\right)\|K_{NN}-L_{NN}\|_2\sum_{\ell=1}^{|N|}\frac{1}{\ell} \\
&\le 4\kappa^2\rho^{-2}\left(1+\frac{\kappa^2}{\rho}\right)\frac{1+\log|N|}{|N|}\|K_{NN}-L_{NN}\|_2.
\end{aligned}$$

Taking the maximum over $i \in N$ proves the claimed bound, with $\widetilde{C}_{\kappa,\rho} = 4\kappa^2\rho^{-2}(1+\kappa^2/\rho)$. □

Scale invariance. The right-hand side of the bound in Proposition 3.2 is invariant under the simultaneous rescaling $K_{NN} \mapsto cK_{NN}$, $L_{NN} \mapsto cL_{NN}$, and $\rho \mapsto c\rho$ for $c > 0$. Indeed, the corresponding kernel rescaling sends κ to $\sqrt{c}\,\kappa$, so that $\widetilde{C}_{\sqrt{c}\kappa,c\rho} = c^{-1}\widetilde{C}_{\kappa,\rho}$, while $\|cK_{NN} - cL_{NN}\|_2 = c\|K_{NN} - L_{NN}\|_2$.

Best rank-r approximation. For the best rank-r approximation K^{eig}_{NN} introduced in equation 4 $\|K_{NN} - K^{\mathrm{eig}}_{NN}\|_2 = \lambda_{r+1}(K_{NN})$. Hence, if $\lambda_j(K_{NN}) \le C_\lambda j^{-\alpha}$ for some $C_\lambda > 0$ and $\alpha > 0$, Proposition 3.2 gives

$$\left\|\mathcal{S}_\rho(K_{NN}) - \mathcal{S}_\rho(K^{\mathrm{eig}}_{NN})\right\|_\infty \le \widetilde{C}_{\kappa,\rho}\frac{1+\log|N|}{|N|}\lambda_{r+1}(K_{NN}) \le C_\lambda\widetilde{C}_{\kappa,\rho}\frac{1+\log|N|}{|N|}(r+1)^{-\alpha}.$$

By the efficiency axiom, the average signed Shapley value is of order $1/|N|$ whenever $U^{K_{NN}}_{\infty,\rho}(N) - U^{K_{NN}}_{\infty,\rho}(\emptyset) = O(1)$. To make the preceding approximation error no larger than this scale, it suffices to choose r such that

$$r \ge \left[C_\lambda\widetilde{C}_{\kappa,\rho}(1+\log|N|)\right]^{1/\alpha} - 1.$$

Using the explicit expression for $\widetilde{C}_{\kappa,\rho}$, when $\rho \ll \kappa^2$, it suffices to take

$$r \gtrsim \left(C_\lambda\kappa^4\rho^{-3}(1+\log|N|)\right)^{1/\alpha}.$$

Thus, faster eigenvalue decay reduces the sufficient rank, whereas weaker regularization increases it.

Uniform Nyström approximation. For the following analysis, we impose an additional sampling assumption on the Nyström approximation K^{nys}_{NN} introduced in equation 5: M is formed by sampling m landmark indices independently and uniformly with replacement from N, with repeated indices retained, as in Theorem 1 of Cortes et al. (2010). For $r \le \mathrm{rank}(K_{MM})$, let $(K_{MM})_r$ be the best rank-r approximation of K_{MM} and define the truncated approximation $K^{\mathrm{nys},r}_{NN} := K_{NM}(K_{MM})^\dagger_r K_{MN}$. Since $K^{\mathrm{nys},r}_{NN} \preceq K^{\mathrm{nys}}_{NN} \preceq K_{NN}$, Theorem 1 of Cortes et al. (2010) gives, with probability at least $1-\delta$,

$$\|K_{NN} - K^{\mathrm{nys}}_{NN}\|_2 \le \|K_{NN} - K^{\mathrm{nys},r}_{NN}\|_2 \le \lambda_{r+1}(K_{NN}) + \frac{|N|\kappa^2}{\sqrt{m}}\left(2+\log\frac{1}{\delta}\right).$$

Combining this estimate with Proposition 3.2 yields, with the same probability,

$$\|\mathcal{S}_\rho(K_{NN}) - \mathcal{S}_\rho(K^{\mathrm{nys}}_{NN})\|_\infty \le \widetilde{C}_{\kappa,\rho}\left[\frac{1+\log|N|}{|N|}\lambda_{r+1}(K_{NN}) + \frac{\kappa^2(1+\log|N|)}{\sqrt{m}}\left(2+\log\frac{1}{\delta}\right)\right].$$

For fixed κ, ρ, and δ, the sampling contribution in this upper bound is of order $(1+\log|N|)/\sqrt{m}$. Requiring this contribution to be of order $1/|N|$ would require $m \gtrsim |N|^2(1+\log|N|)^2$, which is incompatible with the intended low-rank regime $m < |N|$. Thus, this particular uniform-sampling bound does not certify individual Shapley accuracy at the $1/|N|$ scale.

D.3 Proof of Proposition 3.3

Open-domain extension of the prediction and Data Shapley maps. We use the scalar-response notation of Section 3.1. Throughout this subsection, Y_N, Y_T, K_{TN}, and $\rho > 0$ are fixed, while G varies in $\mathrm{Sym}(|N|)$. Here, $\mathrm{Sym}(|N|)^+$ denotes the open cone of real symmetric positive-definite matrices, and $\overline{\mathrm{Sym}(|N|)^+}$ denotes its closure, namely, the cone of real symmetric positive-semidefinite matrices. To differentiate the coalition prediction and Data Shapley maps, define the open set $\mathcal{D}_\rho := \left\{G \in \mathrm{Sym}(|N|) : G + \rho I_{|N|} \in \mathrm{Sym}(|N|)^+\right\}$. Since $\mathcal{D}_\rho$ and $\mathrm{Sym}(|N|)^+$ are open subsets of $\mathrm{Sym}(|N|)$, we have

$$T_G\mathcal{D}_\rho = \mathrm{Sym}(|N|), \qquad T_G\,\mathrm{Sym}(|N|)^+ = \mathrm{Sym}(|N|).$$

Moreover, since $\rho > 0$, every $G \in \overline{\mathrm{Sym}(|N|)^+}$ satisfies $G + \rho I_{|N|} \succ 0$, and hence $\overline{\mathrm{Sym}(|N|)^+} \subseteq \mathcal{D}_\rho$. Therefore, for every nonempty $S \subseteq N$, the coalition prediction map defined in Section 3.1 extends to $\mathcal{D}_\rho$ by

$$A_{S,\rho}(G) := G_{SS} + \rho|S|I_{|S|}, \qquad \mathcal{P}_\rho^S(G) := K_{TS}A_{S,\rho}(G)^{-1}Y_S,$$

with $\mathcal{P}_\rho^{\varnothing}(G) := 0$. For every $G \in \mathcal{D}_\rho$ and every nonempty $S \subseteq N$,

$$A_{S,\rho}(G) = (G + \rho I_{|N|})_{SS} + \rho(|S|-1)I_{|S|} \succ 0,$$

because every principal submatrix of $G + \rho I_{|N|} \succ 0$ is positive definite. Hence $A_{S,\rho}(G)^{-1}$ exists, and the extended coalition prediction map is well-defined. Using these prediction maps, define

$$\mathcal{P}_\rho(G) := \mathcal{P}_\rho^N(G), \qquad U_{\infty,\rho}^G(S) := -\frac{1}{|T|}\left\|Y_T - \mathcal{P}_\rho^S(G)\right\|_2^2, \qquad \mathcal{S}_\rho(G) := \left(\phi_i(U_\rho^G)\right)_{i\in N}.$$

These definitions agree with the corresponding maps from Section 3.1 on $\overline{\mathrm{Sym}(|N|)^+}$. Matrices in $\mathcal{D}_\rho \setminus \overline{\mathrm{Sym}(|N|)^+}$ are used only to provide an open extension on which Fréchet differentiation can be performed. The differentials computed below have the types

$$D\mathcal{P}_\rho(G) : \mathrm{Sym}(|N|) \longrightarrow \mathbb{R}^{|T|}, \qquad D\mathcal{S}_\rho(G) : \mathrm{Sym}(|N|) \longrightarrow \mathbb{R}^{|N|}.$$

Lemma D.2 (Smoothness of prediction and Data Shapley)**.** *For every nonempty $S \subseteq N$, the maps $G \mapsto \mathcal{P}_\rho^S(G)$ and $G \mapsto U_\rho^G(S)$ are C^∞ on $\mathcal{D}_\rho$. More explicitly, for $G \in \mathcal{D}_\rho$, $H \in T_G\mathcal{D}_\rho = \mathrm{Sym}(|N|)$, and every nonempty $S \subseteq N$, one has*

$$D\mathcal{P}_\rho^S(G)[H] = -K_{TS}A_{S,\rho}(G)^{-1}H_{SS}A_{S,\rho}(G)^{-1}Y_S,$$
$$D_G U_{\infty,\rho}^G(S)[H] = -\frac{2}{|T|}\left\langle Y_T - \mathcal{P}_\rho^S(G), K_{TS}A_{S,\rho}(G)^{-1}H_{SS}A_{S,\rho}(G)^{-1}Y_S\right\rangle.$$

Consequently, $\mathcal{P}_\rho$ and $\mathcal{S}_\rho$ are C^∞ on $\mathcal{D}_\rho$.

Proof. For every nonempty $S \subseteq N$, the map $G \mapsto A_{S,\rho}(G)$ is affine with derivative $H \mapsto H_{SS}$, and it takes values in $\mathrm{Sym}(|S|)^+$. Matrix inversion is C^∞ on the invertible matrices and satisfies $D(A^{-1})[B] = -A^{-1}BA^{-1}$. The first formula therefore follows from the chain rule.

Applying the chain rule to $U_{\infty,\rho}^G(S) = -|T|^{-1}\|Y_T - \mathcal{P}_\rho^S(G)\|_2^2$ gives $D_G U_{\infty,\rho}^G(S)[H] = 2|T|^{-1}\langle Y_T - \mathcal{P}_\rho^S(G), D\mathcal{P}_\rho^S(G)[H]\rangle$. Substituting the prediction derivative gives the second formula. Finally, $\mathcal{P}_\rho = \mathcal{P}_\rho^N$, and every component of $\mathcal{S}_\rho$ is a finite linear combination of coalition utilities. Hence both maps are C^∞. □

Lemma D.3 (Constant-rank geometry of the prediction fibers)**.** *Define $\Lambda : \mathrm{Sym}(|N|) \to \mathbb{R}^{|T|}$ by $\Lambda(B) := K_{TN}BY_N$. Then*

$$\operatorname{rank} D\mathcal{P}_\rho(G) = \operatorname{rank}\Lambda \qquad \textit{for every } G \in \mathcal{D}_\rho.$$

Consequently, $\mathcal{P}_\rho$ has constant rank on $\mathcal{D}_\rho$. For every $G_0 \in \mathrm{Sym}(|N|)^+$, the prediction fiber

$$\mathcal{M}_0 := \left\{G \in \mathrm{Sym}(|N|)^+ : \mathcal{P}_\rho(G) = \mathcal{P}_\rho(G_0)\right\}$$

is a properly embedded smooth submanifold of $\mathrm{Sym}(|N|)^+$, and

$$T_{G_0}\mathcal{M}_0 = \ker D\mathcal{P}_\rho(G_0) \subseteq T_{G_0}\,\mathrm{Sym}(|N|)^+ = \mathrm{Sym}(|N|).$$

Consequently, for every $H \in \ker D\mathcal{P}_\rho(G_0)$, there exist $\varepsilon > 0$ and a smooth curve $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}_0$ satisfying $\gamma(0) = G_0$ and $\gamma'(0) = H$.

Proof. For $G \in \mathcal{D}_\rho$, set $Q_G := (G + \rho|N|I_{|N|})^{-1}$ and define $\Gamma_G : \mathrm{Sym}(|N|) \to \mathrm{Sym}(|N|)$ by $\Gamma_G(H) := Q_G H Q_G$. Since Q_G is symmetric and invertible, Γ_G is a linear automorphism with inverse $\Gamma_G^{-1}(B) = Q_G^{-1} B Q_G^{-1}$. Moreover, we have $D\mathcal{P}_\rho(G) = -\Lambda \circ \Gamma_G$. Composition with a linear automorphism does not change rank, proving the constant-rank assertion.

Since $\mathrm{Sym}(|N|)^+$ is open in $\mathrm{Sym}(|N|)$, the differential of the restriction $\mathcal{P}_\rho|_{\mathrm{Sym}(|N|)^+}$ agrees with $D\mathcal{P}_\rho(G)$ under the identification $T_G\,\mathrm{Sym}(|N|)^+ = \mathrm{Sym}(|N|)$. Hence this restriction also has constant rank. By the constant-rank level-set theorem of Lee (2013, Theorem 5.12), $\mathcal{M}_0$ is a properly embedded submanifold of $\mathrm{Sym}(|N|)^+$, whose codimension equals $\mathrm{rank}\,\Lambda$.

Let $\iota : \mathcal{M}_0 \hookrightarrow \mathrm{Sym}(|N|)^+$ denote the inclusion. Since $\mathcal{P}_\rho \circ \iota$ is constant, the chain rule gives $T_{G_0}\mathcal{M}_0 \subseteq \ker D\mathcal{P}_\rho(G_0)$. The constant-rank level-set theorem and rank–nullity give $\dim T_{G_0}\mathcal{M}_0 = \dim \mathrm{Sym}(|N|) - \mathrm{rank}\,\Lambda = \dim \ker D\mathcal{P}_\rho(G_0)$. The inclusion is therefore an equality. The curve assertion follows from the standard characterization of tangent vectors as velocities of smooth curves. □

We now combine the preceding two lemmas to prove Proposition 3.3.

Proof. Suppose, toward a contradiction, that there exist a neighborhood $\mathcal{V}$ of G_0 in $\mathrm{Sym}(|N|)^+$ and a function $\omega : [0, \infty) \to [0, \infty)$ satisfying $\omega(0) = 0$ such that the asserted bound holds for every $K, L \in \mathcal{V}$.

The kernel condition gives $H \in T_{G_0}\,\mathrm{Sym}(|N|)^+ = \mathrm{Sym}(|N|)$ such that $D\mathcal{P}_\rho(G_0)[H] = 0$ but $D\mathcal{S}_\rho(G_0)[H] \neq 0$. By Lemma D.3, there exist $\varepsilon > 0$ and a smooth curve $\gamma : (-\varepsilon, \varepsilon) \to \mathcal{M}_0$ such that $\gamma(0) = G_0$, $\gamma'(0) = H$, and $\mathcal{P}_\rho(\gamma(t)) = \mathcal{P}_\rho(G_0)$ for every $t \in (-\varepsilon, \varepsilon)$.

Fréchet differentiability of $\mathcal{S}_\rho$ gives

$$\mathcal{S}_\rho(\gamma(t)) = \mathcal{S}_\rho(G_0) + tD\mathcal{S}_\rho(G_0)[H] + o(|t|).$$

Since $D\mathcal{S}_\rho(G_0)[H] \neq 0$, this expansion implies $\mathcal{S}_\rho(\gamma(t)) \neq \mathcal{S}_\rho(G_0)$ for every sufficiently small nonzero t. Since $\gamma(t) \to G_0$ and $\mathcal{V}$ is a neighborhood of G_0, we may choose such a $t \neq 0$ with $\gamma(t) \in \mathcal{V}$. Taking $K = G_0$ and $L = \gamma(t)$ in the assumed bound gives

$$0 < \|\mathcal{S}_\rho(\gamma(t)) - \mathcal{S}_\rho(G_0)\|_2 \leq \omega\left(\|\mathcal{P}_\rho(\gamma(t)) - \mathcal{P}_\rho(G_0)\|_2\right) = \omega(0) = 0,$$

which is impossible. □

Remark D.1 (Equivalent Jacobian rank condition)**.** *Let $d := |N|(|N| + 1)/2$, and let* $\mathrm{svec} : \mathrm{Sym}(|N|) \to \mathbb{R}^d$ *denote the linear isomorphism obtained by stacking the upper-triangular entries, including the diagonal, of a symmetric matrix in a fixed order. Let $J_{\mathcal{P}_\rho}(G_0) \in \mathbb{R}^{|T| \times d}$ and $J_{\mathcal{S}_\rho}(G_0) \in \mathbb{R}^{|N| \times d}$ be the corresponding Jacobian matrices. Then*

$$\ker D\mathcal{P}_\rho(G_0) \not\subseteq \ker D\mathcal{S}_\rho(G_0) \quad \Longleftrightarrow \quad \mathrm{rank}\begin{bmatrix} J_{\mathcal{P}_\rho}(G_0) \\ J_{\mathcal{S}_\rho}(G_0) \end{bmatrix} > \mathrm{rank}\, J_{\mathcal{P}_\rho}(G_0).$$

Thus the hypothesis of Proposition 3.3 can be checked through a Jacobian rank comparison without explicitly constructing $\ker D\mathcal{P}_\rho(G_0)$.

D.4 PROOF OF COROLLARY 3.1

The upper-bound analysis in Appendix D.2 highlights limitations in certifying the accuracy of Data Shapley approximation via uniform Nyström sampling. Wang & Zhang (2013) establish lower bounds on Nyström matrix approximation errors through explicit constructions. Motivated by their approach, we present an explicit example demonstrating how uniform Nyström sampling can fail to preserve individual Data Shapley values.

Fix $\rho > 0$. For all sufficiently large $|N|$, partition $N = A \sqcup B$, where $|B| := 2\lfloor |N|^{1/3}/2 \rfloor$ and $|A| := |N| - |B|$. Further partition $B = B_+ \sqcup B_-$, where $|B_+| = |B_-| = |B|/2$. Let $J_k := \mathbf{1}_k \mathbf{1}_k^\top$, and define

$$K_{NN} := \begin{pmatrix} J_{|A|} & 0 \\ 0 & J_{|B|} \end{pmatrix}, \qquad K_{TN} := \begin{pmatrix} \mathbf{0}_{|A|}^\top & \mathbf{1}_{|B|}^\top \\ \mathbf{0}_{|A|}^\top & -\mathbf{1}_{|B|}^\top \end{pmatrix}.$$

Set $y_i = 0$ for $i \in A$, $y_i = 1$ for $i \in B_+$, and $y_i = -1$ for $i \in B_-$, and let $Y_T = (1, 1)^\top$. Thus, $\sum_{i \in B} y_i = 0$ and $|y_i| \le 1$ for every $i \in N$. Let $K_{NN}^{\mathrm{eig},2}$ denote the eigen approximation in equation 4 specialized to $r = 2$. Let K_{NN}^{nys} be the Nyström approximation in equation 5, using $|B|$ landmark indices sampled independently and uniformly with replacement from N. Throughout, K_{TN}, Y_N, and Y_T are held fixed. Thus, for $G \in \{K_{NN}, K_{NN}^{\mathrm{eig},2}, K_{NN}^{\mathrm{nys}}\}$, the quantities $\mathcal{P}_\rho(G)$ and $\mathcal{S}_\rho(G)$ vary only through the training–training kernel block G.

This construction is realized by a bounded two-cluster feature map. Indeed, for orthonormal vectors e_A, e_B, take $\psi(x_i) = e_A$ for $i \in A$, $\psi(x_i) = e_B$ for $i \in B$, $\psi(x_{t_+}) = e_B$, and $\psi(x_{t_-}) = -e_B$. Hence the bounded-kernel assumption holds uniformly with $\kappa = 1$. Moreover, each admissible training block G above is positive semidefinite. Therefore, every nonempty coalition $S \subseteq N$ satisfies

$$G_{SS} + \rho |S| I_{|S|} \succeq \rho |S| I_{|S|} \succ 0.$$

Proposition D.1 (Exact eigen recovery and uniform Nyström failure)**.** *Under the preceding setting, the rank-two eigen approximation recovers K_{NN} exactly and hence preserves its Data Shapley values:*

$$K_{NN}^{\mathrm{eig},2} = K_{NN}, \qquad \mathcal{S}_\rho(K_{NN}^{\mathrm{eig},2}) = \mathcal{S}_\rho(K_{NN}).$$

For every $i \in B$, $|[\mathcal{S}_\rho(K_{NN})]_i| = \Theta_\rho(|N|^{-1})$. Moreover, with probability $1 - O(|N|^{-1/3})$, the Nyström approximation preserves the grand-coalition prediction, $\mathcal{P}_\rho(K_{NN}^{\mathrm{nys}}) = \mathcal{P}_\rho(K_{NN}) = 0$, but, simultaneously for every $i \in B$,

$$|[\mathcal{S}_\rho(K_{NN}^{\mathrm{nys}})]_i - [\mathcal{S}_\rho(K_{NN})]_i| = \Theta_\rho(|N|^{-1}).$$

Thus, the rank-two eigen approximation preserves $\mathcal{S}_\rho(K_{NN})$ exactly, whereas, with high probability, uniform Nyström preserves the grand-coalition prediction but incurs a Shapley error on B of the same order as the corresponding coordinates of $\mathcal{S}_\rho(K_{NN})$.

Proof. Define $u_A := \mathbf{1}_A / \sqrt{|A|}$ and $u_B := \mathbf{1}_B / \sqrt{|B|}$. Since $A \cap B = \emptyset$, these vectors are orthonormal, and

$$K_{NN} = |A| u_A u_A^\top + |B| u_B u_B^\top.$$

This is the eigendecomposition of K_{NN}. Its only nonzero eigenvalues are $|A|$ and $|B|$, while every vector orthogonal to u_A and u_B belongs to the kernel of K_{NN}. Hence K_{NN} has rank two and $K_{NN}^{\mathrm{eig},2} = K_{NN}$. The rows of K_{TN} are $\mathbf{1}_B^\top$ and $-\mathbf{1}_B^\top$, and therefore belong to $\mathrm{span}\{u_A, u_B\}$. Thus, if U_2 contains the two nonzero eigenvectors, then $K_{TN} U_2 U_2^\top = K_{TN}$. Consequently, the rank-two eigen construction preserves both K_{NN} and K_{TN} exactly.

Now consider the event $\mathcal{E}_N := \{M \subseteq A\}$, where multiplicities are retained. Since the $|M| = |B|$ landmarks are sampled independently and uniformly with replacement,

$$\mathbb{P}(\mathcal{E}_N) = \left(1 - \frac{|B|}{|N|}\right)^{|B|} \ge 1 - \frac{|B|^2}{|N|} = 1 - O(|N|^{-1/3}).$$

On $\mathcal{E}_N$, every sampled column of K_{NN} equals $\mathbf{1}_A$. Hence $K_{NM} = \mathbf{1}_A \mathbf{1}_{|M|}^\top$, $K_{MM} = J_{|M|}$, and $J_{|M|}^\dagger = |M|^{-2} J_{|M|}$. It follows that

$$K_{NN}^{\mathrm{nys}} = \mathbf{1}_A \mathbf{1}_A^\top = \begin{pmatrix} J_{|A|} & 0 \\ 0 & 0_{|B|} \end{pmatrix}, \qquad \|K_{NN} - K_{NN}^{\mathrm{nys}}\|_2 = |B| = \Theta(|N|^{1/3}).$$

Fix a nonempty coalition $S \subseteq N$, order its indices so that those in $S \cap A$ precede those in $S \cap B$, and define $z_S := \sum_{j \in S \cap B} y_j$. Under this ordering,

$$K_{SS} = \begin{pmatrix} J_{|S\cap A|} & 0 \\ 0 & J_{|S\cap B|} \end{pmatrix}, \quad Y_S = \begin{pmatrix} \mathbf{0}_{|S\cap A|} \\ Y_{S\cap B} \end{pmatrix}, \quad K_{TS} = \begin{pmatrix} \mathbf{0}^\top & \mathbf{1}^\top \\ \mathbf{0}^\top & -\mathbf{1}^\top \end{pmatrix}.$$

The regularized coalition matrix is block diagonal. Since Y_S vanishes on $S \cap A$, only its B-block contributes to the prediction. If $S \cap B \neq \emptyset$, then $\mathbf{1}_{|S\cap B|}^\top$ is a left eigenvector of $J_{|S\cap B|} + \rho|S|I_{|S\cap B|}$ with eigenvalue $|S \cap B| + \rho|S|$. Therefore, $\mathbf{1}_{|S\cap B|}^\top (J_{|S\cap B|} + \rho|S|I_{|S\cap B|})^{-1} = \mathbf{1}_{|S\cap B|}^\top / (|S \cap B| + \rho|S|)$. Using $\mathbf{1}_{|S\cap B|}^\top Y_{S\cap B} = z_S$, we obtain

$$\mathcal{P}_\rho^S(K_{NN}) = \begin{pmatrix} p_S \\ -p_S \end{pmatrix}, \quad p_S = \frac{z_S}{|S \cap B| + \rho|S|}, \quad \mathcal{P}_\rho^S(K_{NN}^{\mathrm{nys}}) = \begin{pmatrix} q_S \\ -q_S \end{pmatrix}, \quad q_S = \frac{z_S}{\rho|S|}.$$

Indeed, on $\mathcal{E}_N$, the B-block of $(K_{NN}^{\mathrm{nys}})_{SS}$ is zero, so its regularized B-block is $\rho|S|I_{|S\cap B|}$. Here K_{TS} is retained without approximation. If $S \cap B = \emptyset$, then $z_S = 0$, so both formulas still give the zero prediction. Since $Y_T = (1, 1)^\top$ and $|T| = 2$, the two utilities are $U_{\infty,\rho}^{K_{NN}}(S) = -1 - p_S^2$ and $U_{\infty,\rho}^{K_{NN}^{\mathrm{nys}}}(S) = -1 - q_S^2$. The balanced-label condition gives $z_N = 0$. Hence

$$\mathcal{P}_\rho(K_{NN}) = \mathcal{P}_\rho(K_{NN}^{\mathrm{nys}}) = 0, \qquad U_{\infty,\rho}^{K_{NN}}(N) = U_{\infty,\rho}^{K_{NN}^{\mathrm{nys}}}(N) = U_{\infty,\rho}^{K_{NN}}(\emptyset) = U_{\infty,\rho}^{K_{NN}^{\mathrm{nys}}}(\emptyset) = -1.$$

Fix $i \in A$, and let $S \subseteq N \setminus \{i\}$ be its predecessor coalition. Define $\Delta_i^K(S) := U_{\infty,\rho}^{K_{NN}}(S \cup \{i\}) - U_{\infty,\rho}^{K_{NN}}(S)$ and $\Delta_i^{\mathrm{nys}}(S) := U_{\infty,\rho}^{K_{NN}^{\mathrm{nys}}}(S \cup \{i\}) - U_{\infty,\rho}^{K_{NN}^{\mathrm{nys}}}(S)$. If $S = \emptyset$, then both marginal contributions are zero.

Suppose that $S \neq \emptyset$. Adding $i \in A$ leaves z_S and $|S \cap B|$ unchanged and replaces $|S|$ by $|S| + 1$. Define $g_\rho(u) := u^{-2} - (u + \rho)^{-2}$ for $u > 0$. Then

$$\Delta_i^K(S) = z_S^2 \left[\frac{1}{(|S \cap B| + \rho|S|)^2} - \frac{1}{(|S \cap B| + \rho|S| + \rho)^2} \right] = z_S^2 g_\rho(|S \cap B| + \rho|S|),$$

$$\Delta_i^{\mathrm{nys}}(S) = z_S^2 \left[\frac{1}{(\rho|S|)^2} - \frac{1}{(\rho|S| + \rho)^2} \right] = z_S^2 g_\rho(\rho|S|),$$

$$\Delta_i^{\mathrm{nys}}(S) - \Delta_i^K(S) = z_S^2 \left[g_\rho(\rho|S|) - g_\rho(|S \cap B| + \rho|S|) \right] \geq 0.$$

The last inequality follows because $g_\rho'(u) = -2u^{-3} + 2(u + \rho)^{-3} < 0$.

Let δ_A denote the common Nyström-minus-exact Shapley difference for samples in A. In the permutation representation, the event that S consists of one sample from B has probability $|B|/[|N|(|N| - 1)]$. On this event, $|S| = |S \cap B| = 1$ and $z_S^2 = 1$, giving the required lower bound.

For the upper bound, conditional on $|S| = k$, the set S is uniformly distributed over the k-subsets of $N \setminus \{i\}$. Since $i \in A$, we have $\sum_{j \in N\setminus\{i\}} y_j = 0$, $\sum_{j \in N\setminus\{i\}} y_j^2 = |B|$, and $\sum_{j \neq \ell} y_j y_\ell = -|B|$. Expanding z_S^2 and using the corresponding inclusion probabilities gives

$$\mathbb{E}\big[z_S^2 \,\big|\, |S| = k\big] = \frac{|B|k(|N| - 1 - k)}{(|N| - 1)(|N| - 2)} \leq \frac{|B|k}{|N| - 2}, \qquad g_\rho(\rho k) \leq \frac{2}{\rho^2 k^3}.$$

Since the predecessor cardinality is uniform on $\{0, \ldots, |N| - 1\}$, these estimates imply

$$\frac{|B|}{|N|(|N| - 1)} \left[g_\rho(\rho) - g_\rho(1 + \rho) \right] \leq \delta_A \leq \frac{2|B|}{\rho^2 |N|(|N| - 2)} \sum_{k=1}^{|N|-1} \frac{1}{k^2} \leq \frac{\pi^2 |B|}{3\rho^2 |N|(|N| - 2)}.$$

Therefore, $\delta_A = \Theta_\rho(|B|/|N|^2)$.

All samples in A have the same Shapley value. Samples in B also have the same Shapley value: exchanging B_+ and B_- changes z_S to $-z_S$, while both utilities depend on z_S only through z_S^2.

Let δ_B denote the common Nyström-minus-exact Shapley difference for samples in B. Since the two games have the same empty- and grand-coalition utilities, Shapley efficiency gives

$$|A|\delta_A + |B|\delta_B = 0,$$
$$\frac{|A|}{|N|(|N|-1)}\left[g_\rho(\rho) - g_\rho(1+\rho)\right] \le |\delta_B| \le \frac{\pi^2|A|}{3\rho^2|N|(|N|-2)}, \qquad |\delta_B| = \Theta_\rho(|N|^{-1}).$$

It remains to determine the exact Shapley values on B. Let α_A denote the common exact Shapley value of the samples in A. Applying the same singleton lower bound and moment upper bound to $\Delta_i^K(S)$ gives

$$\frac{|B|}{|N|(|N|-1)} g_\rho(1+\rho) \le \alpha_A \le \frac{\pi^2|B|}{3\rho^2|N|(|N|-2)}, \qquad \alpha_A = \Theta_\rho\left(\frac{|B|}{|N|^2}\right).$$

Let α_B denote the common exact Shapley value of the samples in B. Shapley efficiency for the exact game gives

$$|A|\alpha_A + |B|\alpha_B = 0, \qquad \alpha_B = -\frac{|A|}{|B|}\alpha_A = -\Theta_\rho(|N|^{-1}).$$

Finally, $K_{NN}^{\mathrm{eig},2} = K_{NN}$, so the eigen approximation has zero Data Shapley error. On $\mathcal{E}_N$,

$$[\mathcal{S}_\rho(K_{NN})]_i = -\Theta_\rho(|N|^{-1}), \quad |[\mathcal{S}_\rho(K_{NN}^{\mathrm{nys}})]_i - [\mathcal{S}_\rho(K_{NN})]_i| = \Theta_\rho(|N|^{-1}) \quad (i \in B).$$

This completes the proof. □

E Empirical Analysis of the Gap Between Proposition 3.2 and the Implemented Predictor

Proposition 3.2 bounds the change of Data Shapley values when the training kernel is replaced by an approximation. The setting of the proposition is not the setting of the predictor implemented in LRFShap. There are three differences: (i) the proposition approximates only the training block K_{SS} and keeps the test–train block K_{TS} exact, whereas LRFShap approximates both blocks through the low-rank features Φ; (ii) the ridge in the proposition is $\rho|S|$, which comes from a fixed regularization coefficient under the averaged empirical loss, whereas the implementation uses a coalition-independent ridge γ; (iii) the utility in the proposition is the negative squared test error of a scalar-response regression, whereas the implementation uses test accuracy. We do not extend the proposition to the implemented predictor here. Instead, we introduce the three differences one at a time and measure how the Data Shapley error changes at each step. The purpose is to see which of the assumptions matters in practice, not to show that the bound holds in the implemented setting.

Setup. We use the eNTK of `bert-base-uncased` on SST-2 and MR, and draw a random subset of $n = 500$ training samples and $|T| = 500$ test samples. We write ϕ for the exact Data Shapley values of the full-kernel game and ϕ_r for those of the game with the rank-r eigen-truncation K_{NN}^{eig}, $r \in \{1, 5, 10, 15, 20, 25, 30\}\%$ of n. Neither vector can be computed exactly, so we use Monte Carlo estimates $\widehat{\phi}$ and $\widehat{\phi}_r$ over $B = 300$ random permutations. The same permutations are used for both games and no truncation is applied, so the estimate of the difference, $\widehat{\phi}_r - \widehat{\phi}$, is a paired estimate: for each permutation we take the difference of the two marginal contributions and average over permutations. This estimate is unbiased for $\phi_r - \phi = \mathcal{S}(K_{NN}^{\text{eig}}) - \mathcal{S}(K_{NN})$, the quantity bounded in Proposition 3.2, and the pairing removes the variation that the two games share. Monte Carlo uncertainty from the finite number of permutations remains, and the reported $\|\widehat{\phi}_r - \widehat{\phi}\|_\infty$ is not an unbiased estimate of $\|\phi_r - \phi\|_\infty$. We report $\|\widehat{\phi}_r - \widehat{\phi}\|_\infty$ and the Spearman rank correlation between $\widehat{\phi}_r$ and $\widehat{\phi}$ over the n training samples. The regularization parameter is $\rho = 10^{-2}$ for the size-scaled ridge. For the two variants with a fixed ridge we report $\gamma = 10^{-2}$ and $\gamma = 10^{-1}$. Starting from the setting of the proposition, we switch on the three differences cumulatively (Table 3); the last variant is the predictor used by LRFShap.

Variant	Approximated blocks	Ridge	Utility
Prop. 3.2 setting	K_{SS}	$\rho\|S\|$	$-$MSE, $y \in \{-1, +1\}$
+ approximate K_{TS}	K_{SS}, K_{TS}	$\rho\|S\|$	$-$MSE, $y \in \{-1, +1\}$
+ fixed ridge	K_{SS}, K_{TS}	γ	$-$MSE, $y \in \{-1, +1\}$
+ accuracy utility (LRFShap)	K_{SS}, K_{TS}	γ	accuracy, one-hot Y

Table 3: Variants used in Appendix E. Each row adds one difference between the setting of Proposition 3.2 and the implemented predictor; the last row is the predictor used by LRFShap.

	SST-2				MR			
Variant	5%	10%	20%	30%	5%	10%	20%	30%
Prop. 3.2 setting	0.93	0.96	0.98	0.99	0.95	0.97	0.99	0.99
+ approximate K_{TS}	0.92	0.95	0.97	0.98	0.94	0.97	0.99	0.99
+ fixed ridge ($\gamma = 10^{-2}$)	0.58	0.70	0.78	0.79	0.67	0.74	0.80	0.85
+ fixed ridge ($\gamma = 10^{-1}$)	0.85	0.89	0.93	0.94	0.88	0.92	0.95	0.97
+ accuracy utility (LRFShap) ($\gamma = 10^{-2}$)	0.69	0.76	0.75	0.76	0.72	0.76	0.80	0.81
+ accuracy utility (LRFShap) ($\gamma = 10^{-1}$)	0.79	0.86	0.89	0.91	0.79	0.85	0.88	0.90

Table 4: Spearman rank correlation between $\widehat{\phi}_r$ and $\widehat{\phi}$ for each variant and rank (% of n), $n = 500$, 300 shared permutations. Both vectors are Monte Carlo estimates. For the fixed-ridge variants, the value of γ is given in parentheses.

Rank dependence in the setting of the proposition. In the setting of Proposition 3.2, the estimated error decreases with the rank. Up to rank about 10% its decrease follows the shape of λ_{r+1}

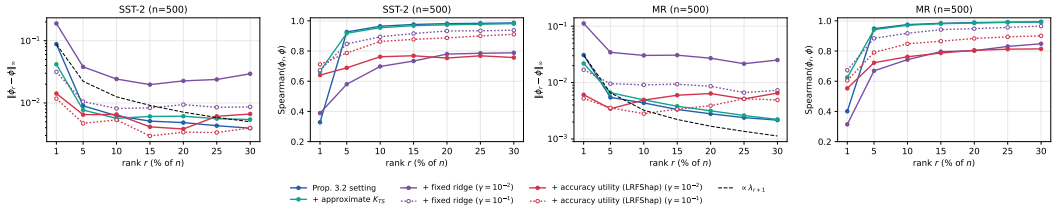

Figure 5: Paired Monte Carlo estimate of the Data Shapley difference between the rank-r eigen-truncated game and the full-kernel game, using the same 300 permutations, on SST-2 (left two panels) and MR (right two panels) with $n = 500$. Left of each pair: $\|\widehat{\phi}_r - \widehat{\phi}\|_\infty$ on a log scale. The dashed line is a reference curve proportional to λ_{r+1}, scaled to the first point of the Proposition 3.2 setting; it shows the shape of the spectral tail and is not a quantitative bound. Right of each pair: Spearman rank correlation between $\widehat{\phi}_r$ and $\widehat{\phi}$. For the fixed-ridge variants, solid lines with filled markers use $\gamma = 10^{-2}$ and dotted lines with open markers use $\gamma = 10^{-1}$.

(dashed line in Figure 5); beyond that the curve flattens, which we attribute to the resolution of the Monte Carlo estimate rather than to the approximation. The rank correlation with the full-kernel estimate is 0.93 (SST-2) and 0.95 (MR) at rank 5% and 0.99 at rank 30% on both datasets. We note that this compares the observed rank dependence with the decay of the spectral tail; we did not evaluate the constant $\widetilde{C}_{\kappa,\rho}$ of the bound, so we make no claim about how tight the bound is numerically.

Approximating the test–train block. Approximating K_{TS} in addition to K_{SS} changes the reported quantities little on these two datasets. For ranks of 5% and above, the Spearman correlations of the two variants differ by at most 0.02 and the ℓ_∞ curves are close; at rank 1% the two ℓ_∞ errors do differ visibly (Figure 5). A plausible reason for the small effect at moderate ranks is that the low-rank test–train block equals $K_{TN}U_rU_r^\top$ restricted to S, so its error is driven by the same tail of the training spectrum. We only compare each variant against the full-kernel estimate here; we did not measure the distance between the two low-rank estimates themselves.

Fixed ridge. Replacing $\rho|S|$ by a fixed γ is the one change with a clear effect under the squared-error utility. With $\gamma = 10^{-2}$, the ℓ_∞ error stops decreasing with λ_{r+1}, and the rank correlation plateaus at 0.79 (SST-2) and 0.85 (MR). A fixed ridge corresponds to an effective coefficient $\gamma/|S|$ under the averaged loss, which becomes small for large coalitions; this is consistent with the constant $\widetilde{C}_{\kappa,\rho}$ growing as ρ decreases. In line with this explanation, a larger fixed ridge reduces the gap. With $\gamma = 10^{-1}$, the rank correlation rises to 0.94 (SST-2) and 0.97 (MR) at rank 30%, close to the setting of the proposition, and the ℓ_∞ error decreases by a factor of about two to four at ranks of 5% and above, although it does not follow the decay of λ_{r+1}. These results suggest that, on these two datasets, the gap in rank agreement observed with the smaller fixed ridge is largely a matter of regularization strength and can be substantially reduced through the choice of γ.

Accuracy utility. Using accuracy as the utility, as in LRFShap, yields rank correlations between the low-rank and full-kernel estimates of 0.76 (SST-2) and 0.81 (MR) at rank 30% for $\gamma = 10^{-2}$. The correlations remain roughly constant at ranks of 10% and above, while the ℓ_∞ error is of a similar magnitude to that in the setting of the proposition. With $\gamma = 10^{-1}$, the rank correlation continues to increase with the rank and reaches 0.91 (SST-2) and 0.90 (MR) at rank 30%, while the ℓ_∞ error remains of a similar magnitude. At the same fixed ridge, the rank correlations under accuracy and negative squared-test-error utilities are relatively close. These results suggest that, in the experimental settings considered, rank agreement is more sensitive to the choice of the fixed ridge parameter than to the choice between these two utilities.

Summary. On two eNTK datasets, we went from the setting of Proposition 3.2 to the implemented predictor in three steps. Approximating the test–train block changed the reported fidelity measures little at moderate ranks. Replacing the size-scaled ridge by a fixed one clearly changed the sensitivity of the squared-error game to the approximation. However, a larger fixed ridge brought the rank

agreement close to that observed in the setting of the proposition. These results suggest that, when replacing coalition-size-scaled regularization with a fixed ridge, choosing an appropriate regularization strength can substantially reduce the resulting loss of rank agreement. With the accuracy utility, a larger fixed ridge likewise improved rank agreement, while switching between the two utilities at the same fixed ridge had a smaller effect. This section examines the experimental role of the Proposition 3.2 assumptions; it does not extend the theoretical guarantee to the implemented predictor.

F COMPUTATIONAL COMPLEXITY OF FREESHAP AND LRFSHAP

Table 5 summarizes the computational costs of FreeShap and LRFShap. We denote the training-set size by $|N|$, the coalition size by $|S|$, the number of classes by $|C|$, the test-set size by $|T|$, and the number of Monte Carlo permutations by B. The feature dimension is $q = r$ for eigen features. For Nyström features, q is the number of retained positive eigenvalues of the landmark matrix W, with $q \leq m$, where m is the number of landmarks. We assume $|C| \leq q \leq |N|$ when simplifying the bounds, and use the coalition-independent ridge parameter $\gamma > 0$.

Both implementations share the class-independent factorization or inverse across all label columns. In particular, the FreeShap baseline computes the factorization or inverse of $K_{SS} + \gamma I_{|S|}$ once and reuses it for all $|C|$ classes. LRFShap similarly shares its system matrix across classes. Consequently, factorization and inverse-update costs do not carry an additional factor of $|C|$. Label-dependent solves, matrix products, and test predictions are counted separately.

The table reports linear-algebra costs once the required kernel entries are available. The main text focuses on coalition-solve costs, whereas the table also includes label-dependent computations and test predictions.

Table 5: Computational complexity of FreeShap and LRFShap with factorization and inverse reuse across classes. Preparation costs are incurred once; other costs are per evaluated coalition. For direct solves, LRFShap uses the $q \times q$ feature-space system when $|S| \geq q$ and the $|S| \times |S|$ kernel-space system when $|S| < q$. Incremental inverse updates use blockwise inversion for FreeShap and Sherman–Morrison updates in feature space for LRFShap.

Operation	FreeShap	LRFShap
One-time preparation		
Eigen features	—	$\mathcal{O}(\|N\|^3 + \|T\|\|N\|q)$
Nyström features	—	$\mathcal{O}(m^3 + (\|N\| + \|T\|)mq)$
Per coalition, direct, $\|S\| \geq q$		
Matrix construction and factorization	$\mathcal{O}(\|S\|^3)$	$\mathcal{O}(\|S\|q^2 + q^3)$
Label solves and predictions	$\mathcal{O}(\|C\|\|S\|(\|S\| + \|T\|))$	$\mathcal{O}(\|C\|q(\|S\| + q + \|T\|))$
Total	$\mathcal{O}(\|S\|^3 + \|C\|\|S\|(\|S\| + \|T\|))$	$\mathcal{O}(\|S\|q^2 + q^3 + \|C\|\|T\|q)$
Per coalition, direct, $\|S\| < q$		
Total	$\mathcal{O}(\|S\|^3 + \|C\|\|S\|(\|S\| + \|T\|))$	$\mathcal{O}(\|S\|q(\|S\| + \|T\|))$
Per coalition, incremental		
Shared inverse update	$\mathcal{O}(\|S\|^2)$	$\mathcal{O}(q^2)$
Label products and predictions	$\mathcal{O}(\|C\|\|S\|(\|S\| + \|T\|))$	$\mathcal{O}(\|C\|q(\|S\| + q + \|T\|))$
Total	$\mathcal{O}(\|C\|\|S\|(\|S\| + \|T\|))$	$\mathcal{O}(\|C\|q(\|S\| + q + \|T\|))$

For a direct feature-space solve, LRFShap forms $G_S = \Phi_S^\top \Phi_S + \gamma I_q$ and $R_S = \Phi_S^\top Y_S$, solves $G_S A_S = R_S$, and computes $\Phi_{\text{te}} A_S$. For $|S| < q$, the table assumes that LRFShap explicitly forms $\widetilde{K}_{SS} = \Phi_S \Phi_S^\top$ and $\widetilde{K}_{TS} = \Phi_{\text{te}} \Phi_S^\top$ before solving the smaller kernel-space system. The incremental label-product bounds allow these products to be recomputed at each evaluated prefix; sharing the inverse does not eliminate their dependence on $|C|$.

The comparisons below sum over all $|N|$ nonempty prefixes of each permutation. They therefore provide upper bounds when TMC truncation is enabled. Both methods use the same fixed number B of permutations and the same truncation rule. However, their prefix utilities and full-coalition utilities can differ, so they can evaluate different numbers of prefixes before truncation. These differences also contribute to the observed wall-clock runtimes.

- **Direct factorization.** Evaluating every prefix with a direct solve costs

$$\mathcal{O}\big(B\big(|N|^4 + |C||N|^3 + |C||T||N|^2\big)\big)$$

for FreeShap and $\mathcal{O}\big(B\big(|N|^2q^2 + |N|q^3 + |C||T||N|q + |T|q^3\big)\big)$ for LRFShap. Excluding prediction and preparation costs, the leading direct-solve terms are $\mathcal{O}(B|N|^4)$ and $\mathcal{O}(B|N|^2q^2)$, respectively. Their ratio is $(|N|/q)^2$. This comparison concerns direct-solve costs; practical runtimes also depend on incremental updates, inverse refreshes, preparation, and TMC truncation. FreeShap incurs no additional class factor in its shared factorization cost.

- **Small coalitions.** When $|S| < q$, LRFShap solves a $|S| \times |S|$ system, so the low-rank representation does not reduce the system dimension at these prefixes. It also incurs the cost of constructing $\widetilde{K}_{SS}$ and $\widetilde{K}_{TS}$ from the features, whereas FreeShap uses the corresponding full-kernel blocks. Thus, a runtime advantage is not guaranteed for small coalitions. These prefixes contribute $\mathcal{O}(q^4 + |T|q^3)$ per permutation to LRFShap. Since $q \leq |N|$, the q^4 term is absorbed by $\mathcal{O}(|N|^2q^2)$. The $|T|q^3$ term is also absorbed when $|T| \lesssim |N|^2/q$; it is retained explicitly in the general bound above.
- **Incremental updates and inverse refreshes.** Both implementations begin caching after a coalition-size threshold h ($h = 500$ in our code). FreeShap additionally rebuilds its shared inverse at every multiple of h to limit accumulated round-off. Each rebuild is performed once and reused across classes.

  Summing the incremental terms over a full permutation gives $\mathcal{O}\big(|C||N|^2(|N| + |T|)\big)$ for FreeShap and $\mathcal{O}(|C||N|q(|N| + q + |T|))$ for LRFShap. These expressions include the label-dependent products and predictions; the inverse updates themselves are shared across classes.

  For $h \leq |N|$, FreeShap's initial direct factorizations and periodic inverse rebuilds add $\mathcal{O}(h^4 + |N|^4/h)$ per permutation. For LRFShap, let $s = \min\{h, q\}$. Its initial direct evaluations add $\mathcal{O}(h^2qs + |T|qs^2)$ for matrix construction, factorization, and small-coalition cross-kernel construction. The initial feature-space inverse construction is covered by the stated bounds. These startup and refresh costs must be added to the incremental terms, and all per-permutation costs are multiplied by B. Their contribution to actual runtime depends on the problem dimensions and the prefixes reached before truncation.
- **Preparation.** Feature construction is a one-time cost shared by all permutations and is included in our reported wall-clock speedups. For eigen features, the table gives the dense eigendecomposition cost $\mathcal{O}(|N|^3 + |T||N|q)$. Our implementation uses a dense eigendecomposition for $|N| \leq 5000$ and a partial Lanczos decomposition above this threshold. For larger problems, randomized subspace iteration (Halko et al., 2011) provides an alternative with $\mathcal{O}(|N|^2q)$ decomposition cost for a fixed number of iterations and oversampling proportional to q, plus $\mathcal{O}(|T||N|q)$ for test features.

  For Nyström features, dense eigendecomposition of the $m \times m$ landmark matrix and feature construction cost $\mathcal{O}(m^3 + (|N| + |T|)mq)$. Keeping m and q distinct accounts for the case in which only $q < m$ positive eigenvalues are retained.

G KPS ERRORS AND FIDELITY SCORES

In this section, we report the kernel approximation error, the prediction error, and the Data Shapley error for all models and datasets at $n = 2000$, together with the Spearman and Pearson correlations with the baseline mentioned in Section 4.1.

In addition to the uniform Nyström sampling used in the main text, the figures include the leverage-score Nyström method (Alaoui & Mahoney, 2015).

For both Nyström variants, the pseudoinverse $K_{MM}^{\dagger}$ in Equation equation 5 is computed from the eigendecomposition $K_{MM} = Q\Sigma Q^{\top}$ by discarding the eigenvalues below $\varepsilon = 10^{-8}$; writing Q_s and Σ_s for the retained eigenvectors and eigenvalues, the features are $\Phi = K_{NM} Q_s \Sigma_s^{-1/2}$, so that $\Phi\Phi^{\top} = K_{NM} K_{MM}^{\dagger} K_{MN}$ on the retained spectrum. Uniform Nyström draws the m landmarks uniformly at random. Leverage-score Nyström draws them with probability proportional to the ridge leverage scores $\left[K_{NN}(K_{NN} + \bar{\gamma} I)^{-1}\right]_{ii}$ with $\bar{\gamma} = \gamma n$, computed exactly from the eigendecomposition of K_{NN}; no importance reweighting is applied, and the same landmark seed is used for both variants. The two Nyström variants behave similarly on all datasets.

The fidelity scores point in the same direction as the errors: the eigen decomposition achieves higher correlations than both Nyström variants at every rank, and in particular its higher Spearman correlation means that it better preserves the ranking of valuable data.

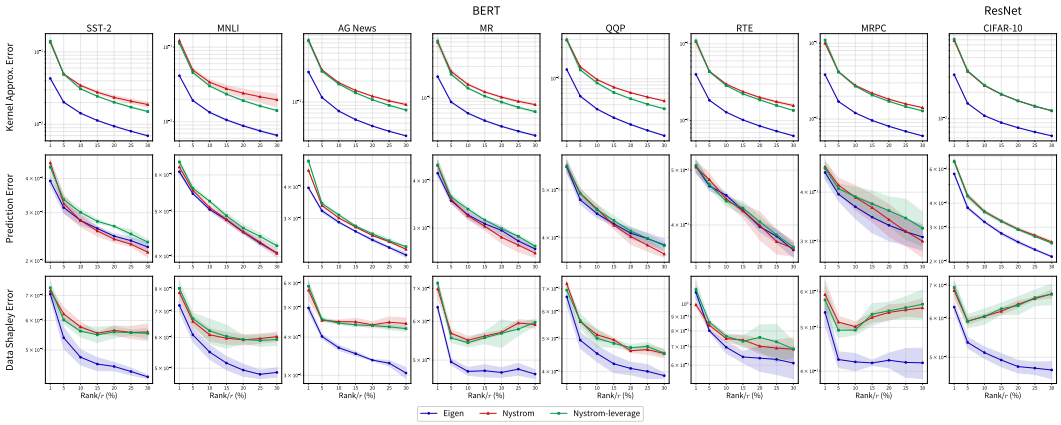

Figure 6: Kernel approximation error, prediction error, and data Shapley errors for BERT and ResNet-18.

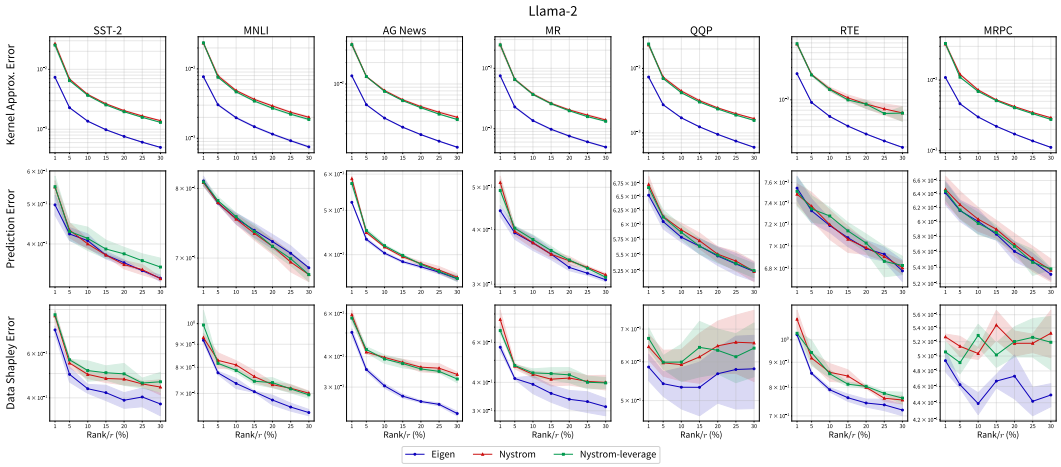

Figure 7: Kernel approximation error, prediction error, and data Shapley errors for Llama-2.

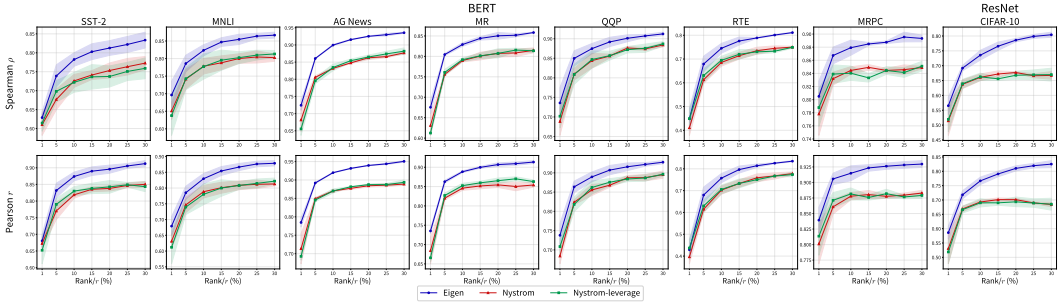

Figure 8: the Spearman and Pearson correlations for BERT and ResNet-18.

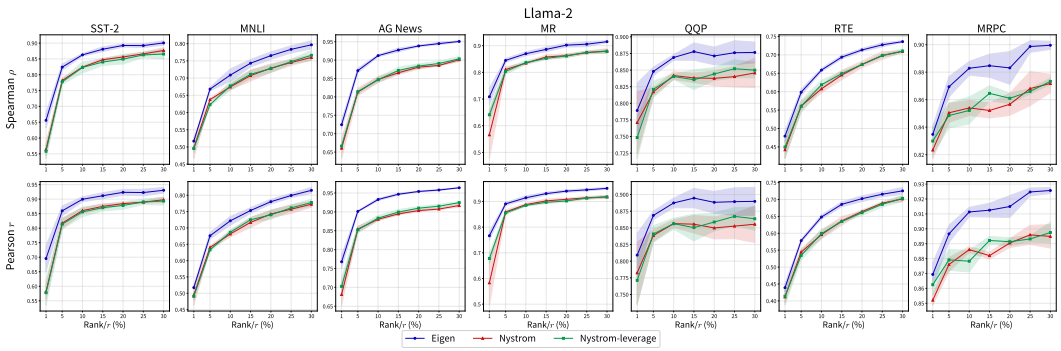

Figure 9: the Spearman and Pearson correlations for Llama-2.

H SENSITIVITY OF THE NYSTRÖM METHOD TO THE CHOICE OF LANDMARKS

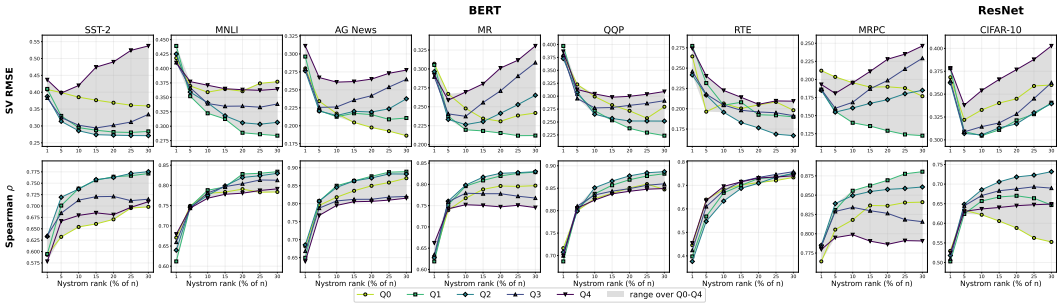

Figure 10: Kernel relative error, root mean square error (RMSE) of data Shapley with respect to the baseline, and Spearman correlation for the Nyström landmark settings Q0–Q4. The model is BERT and ResNet-18. With increasing rank, the gap across settings shrinks for the kernel error but widens for RMSE and Spearman correlation.

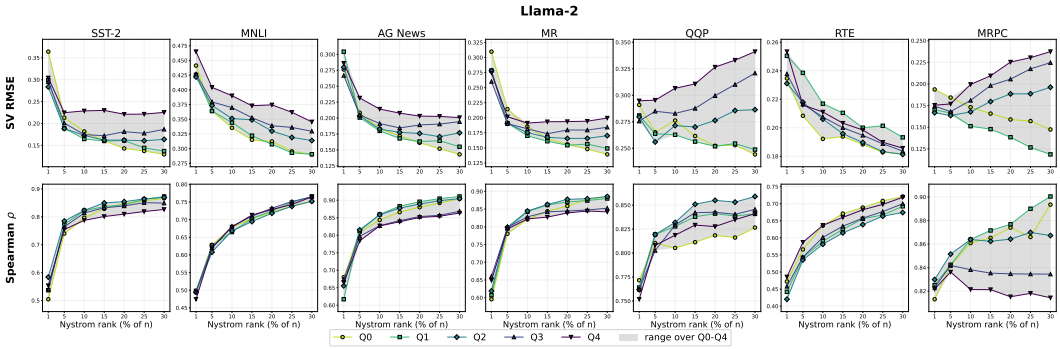

Figure 11: Kernel relative error, root mean square error (RMSE) of data Shapley with respect to the baseline, and Spearman correlation for the Nyström landmark settings Q0–Q4. The model is Llama-2. With increasing rank, the gap across settings shrinks for the kernel error but widens for RMSE and Spearman correlation.

We show that the Data Shapley error of the Nyström method depends heavily on the choice of landmarks, and that this dependence grows with the rank. For a dataset with $n = 2000$ samples and a model, we first compute Shapley values with the baseline. At each rank r, we construct five landmark sets of equal size along this ordering. Q0 and Q4 contain the r samples with the smallest and largest baseline Shapley values, respectively, while Q1, Q2, and Q3 contain r consecutive samples centered at the 25th, 50th, and 75th percentiles, respectively. We then compute Shapley values for each of the Q0–Q4 settings. Figure 10 shows that, as r increases, the variation across Q0–Q4 decreases for the kernel approximation error but increases for the RMSE and Spearman correlation.

I EXPERIMENTAL DETAILS

The models which are used in the experiments are BERT (`bert-base-uncased`; Devlin et al., 2019), Llama-2 (`llama-2-7b-hf`; Touvron et al., 2023), and ResNet-18 (`resnet-18-pretrained`; He et al., 2016). Table 6 lists the settings shared by all experiments, which follow FreeShap, and Table 7 summarizes the datasets.

Table 6: Experimental settings. The values of γ and ε showed the best performance and fidelity. For each dataset, both methods are timed on the same device.

Shapley estimator	TMC sampling (Ghorbani & Zou, 2019), replacing the Monte Carlo loop in Algorithm 1
Permutations / tolerance	500 / 0.05
eNTK approximation	first-logit approximation (Wei et al., 2022) (Section 2), for both methods
Regularization γ / ε	10^{-2} / 10^{-8}
Seeds	3 (all results are averaged)
GPU	NVIDIA GeForce RTX 3090 (CIFAR-10); NVIDIA RTX A6000 (all other datasets)

The construction of the Nyström features, including the pseudoinverse cutoff and the leverage-score sampling, is described in Appendix G.

Table 7: Datasets. n is the number of training samples used, $|C|$ the number of classes, $|T|$ the size of the test set D_T (the official validation split), and $|E|$ the size of the evaluation dataset (a pre-defined test set for MR and a held-out dataset from the train dataset for other datasets).

Dataset	Task	$\lvert C\rvert$	n	$\lvert T\rvert$	$\lvert E\rvert$	Models
AG News (Zhang et al., 2015)	topic classification	4	5000	1000	1000	BERT, Llama-2
MNLI (Williams et al., 2018)	NLI	3	5000	1000	1000	BERT, Llama-2
MR (Pang & Lee, 2005)	sentiment	2	5000	1000	1000	BERT, Llama-2
MRPC (Dolan & Brockett, 2005)	paraphrase	2	3000	408	668	BERT, Llama-2
QQP (Wang et al., 2019)	paraphrase	2	5000	1000	1000	BERT, Llama-2
RTE (Giampiccolo et al., 2007)	NLI	2	1500	277	990	BERT, Llama-2
SST-2 (Socher et al., 2013)	sentiment	2	5000	872	1000	BERT, Llama-2
CIFAR-10 (Krizhevsky, 2009)	image classification	10	5000	1000	1000	ResNet-18

J COMPUTATION TIME

In this section, we report the computation time of FreeShap and LRFShap across all models and datasets. In the tables, we regard LRFShap as comparable to FreeShap when its data selection score is within 0.01 of FreeShap.

Table 8: Wall-clock cost of data Shapley estimation of FreeShap and LRFShap, on `bert-base-uncased` at $n = 5000$ (1500 for RTE and 3000 for MRPC). **DS** is the data Shapley computation time and **ED** the eigendecomposition time. **Speedup** is $\mathrm{DS}_{\mathrm{FreeShap}}/(\mathrm{ED}_{\mathrm{Eigen}} + \mathrm{DS}_{\mathrm{Eigen}})$. Times are in seconds, averaged over three seeds. Magenta speedups mark the cases where LRFShap is comparable to or better than FreeShap on data selection.

Dataset	FreeShap	Eigen						
	DS (s)	ED (s) / DS (s) / **Speedup** ($\times$)						
		Rank 1%	5%	10%	15%	20%	25%	30%
AG News	3513.90	7.01 260.07 **13.16**	7.19 716.93 **4.85**	7.07 1095.03 **3.19**	7.42 1527.65 **2.29**	7.36 2008.99 **1.74**	7.37 2428.07 **1.44**	7.44 2633.04 **1.33**
MNLI	28898.75	7.30 1287.57 **22.32**	7.16 2853.85 **10.10**	7.46 4072.05 **7.08**	7.41 5995.83 **4.81**	7.07 6266.81 **4.61**	7.31 7438.42 **3.88**	7.41 8259.17 **3.50**
MR	10515.90	7.45 618.82 **16.79**	7.18 1463.88 **7.15**	7.49 2337.29 **4.48**	7.31 3256.70 **3.22**	7.42 4206.85 **2.50**	7.47 5526.64 **1.90**	7.18 6482.37 **1.62**
MRPC	3231.15	2.08 327.45 **9.81**	2.05 771.08 **4.18**	2.13 1128.21 **2.86**	2.09 1652.71 **1.95**	2.13 2016.76 **1.60**	2.14 1968.87 **1.64**	2.17 2284.23 **1.41**
QQP	22666.92	7.41 669.17 **33.50**	7.29 2097.78 **10.77**	7.39 3981.74 **5.68**	7.33 5071.86 **4.46**	7.38 5694.62 **3.98**	7.35 6796.78 **3.33**	7.31 8265.90 **2.74**
RTE	1355.60	0.52 654.71 **2.07**	0.52 565.80 **2.39**	0.51 626.81 **2.16**	0.53 708.65 **1.91**	0.53 786.63 **1.72**	0.52 852.03 **1.59**	0.53 815.26 **1.66**
SST-2	6824.66	7.24 441.75 **15.20**	7.29 958.94 **7.06**	7.14 1528.11 **4.45**	7.25 2178.87 **3.12**	7.28 2751.13 **2.47**	7.45 3504.77 **1.94**	7.50 4176.16 **1.63**

Table 9: Wall-clock cost of data Shapley estimation of FreeShap and LRFShap, on `Llama-2-7b-hf` at $n = 5000$ (1500 for RTE and 3000 for MRPC). **DS** is the data Shapley computation time and **ED** the eigendecomposition time. **Speedup** is $\mathrm{DS}_{\mathrm{FreeShap}}/(\mathrm{ED}_{\mathrm{Eigen}} + \mathrm{DS}_{\mathrm{Eigen}})$. Times are in seconds, averaged over three seeds. Magenta speedups mark the cases where LRFShap is comparable to or better than FreeShap on data selection. On MR, QQP and SST-2, TMC truncates after only a few iterations, so FreeShap itself is cheap and the speedup is correspondingly low; in these cases, however, FreeShap also attains substantially lower accuracy (see Section 4.1).

Dataset	FreeShap	Eigen						
	DS (s)	ED (s) / DS (s) / **Speedup** ($\times$)						
		Rank 1%	5%	10%	15%	20%	25%	30%
AG News	2698.49	7.25 240.29 **10.90**	7.41 575.79 **4.63**	7.12 878.23 **3.05**	7.07 1235.78 **2.17**	7.50 1578.36 **1.70**	7.06 1790.97 **1.50**	7.40 1903.16 **1.41**
MNLI	18189.62	7.09 2340.91 **7.75**	7.08 4458.57 **4.07**	7.27 7654.89 **2.37**	7.26 7440.18 **2.44**	7.37 8795.92 **2.07**	7.63 11475.75 **1.58**	7.38 11472.57 **1.58**
MR	2277.68	6.95 442.25 **5.07**	6.91 1130.66 **2.00**	7.28 1528.31 **1.48**	7.33 2084.85 **1.09**	7.10 2792.51 **0.81**	7.10 3037.48 **0.75**	7.35 3546.07 **0.64**
MRPC	1062.20	2.13 503.83 **2.10**	2.11 1195.96 **0.89**	2.08 1398.12 **0.76**	2.13 1517.16 **0.70**	2.11 1851.80 **0.57**	2.13 1988.92 **0.53**	2.10 2280.07 **0.47**
QQP	1269.64	7.17 894.37 **1.41**	7.36 1819.15 **0.70**	7.31 3150.27 **0.40**	7.26 4322.61 **0.29**	7.38 4679.51 **0.27**	7.30 6216.22 **0.20**	7.29 7324.34 **0.17**
RTE	1934.86	0.52 337.98 **5.72**	0.53 733.20 **2.64**	0.52 685.37 **2.82**	0.53 747.77 **2.59**	0.53 836.38 **2.31**	0.54 844.94 **2.29**	0.53 860.18 **2.25**
SST-2	1536.96	7.17 402.17 **3.75**	7.25 930.85 **1.64**	7.20 1864.09 **0.82**	7.33 2260.33 **0.68**	7.23 3042.92 **0.50**	7.14 3294.88 **0.47**	7.37 3905.43 **0.39**

Table 10: Wall-clock cost of data Shapley estimation of FreeShap and LRFShap, on `ResNet-18` at $n = 5000$. **DS** is the data Shapley computation time and **ED** the eigendecomposition time. **Speedup** is $\mathrm{DS}_{\mathrm{FreeShap}}/(\mathrm{ED}_{\mathrm{Eigen}} + \mathrm{DS}_{\mathrm{Eigen}})$. Times are in seconds, averaged over three seeds. Magenta speedups mark the cases where LRFShap is comparable to or better than FreeShap on data selection.

Dataset	FreeShap	Eigen						
	DS (s)	ED (s) / DS (s) / **Speedup** ($\times$)						
		Rank 1%	5%	10%	15%	20%	25%	30%
CIFAR-10	2929.20	15.37 206.08 **13.23**	15.58 569.09 **5.01**	15.82 898.23 **3.20**	15.85 1283.69 **2.25**	15.85 1595.59 **1.82**	15.94 1925.10 **1.51**	15.46 2469.52 **1.18**

K RAW ACCURACIES

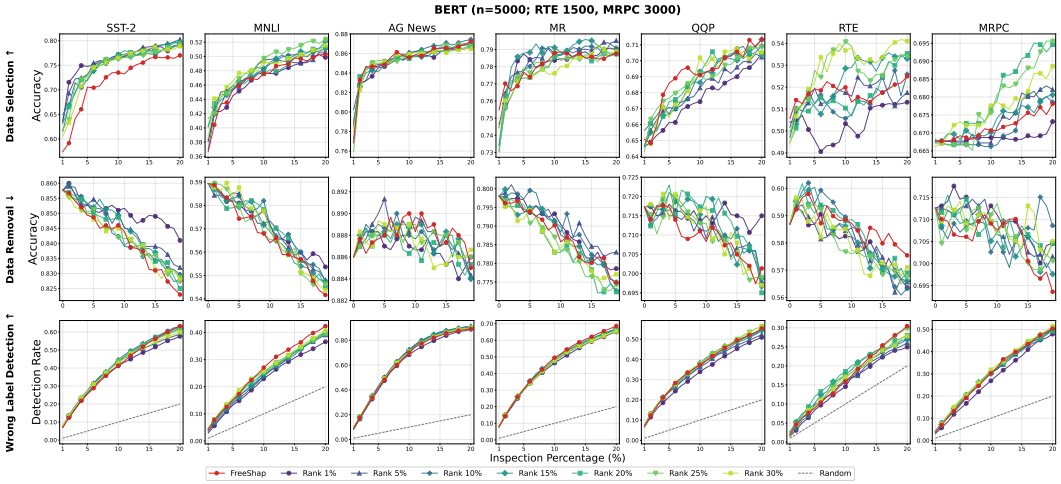

Figure 12: Data selection, removal, and wrong label detection results for 1-20% selection percentages for `bert-base-uncased`.

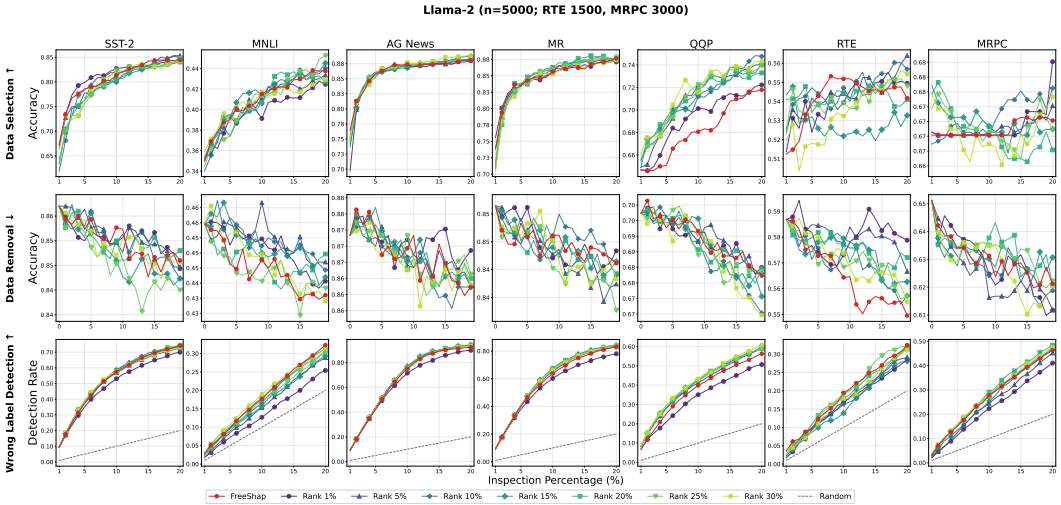

Figure 13: Data selection, removal, and wrong label detection results for 1-20% selection percentages for `llama-2-7b-hf`.

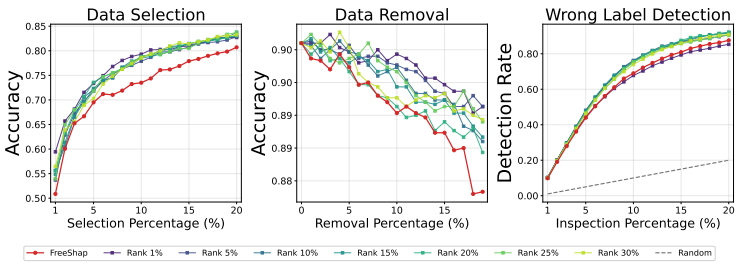

Figure 14: The results of all tasks in CIFAR-10 for `resnet-18-pretrained`.

L MEAN ACCURACY RESULTS

In this section, we report the mean accuracy (detection rate for wrong label detection) across three seeds for all datasets and downstream tasks. For data selection and wrong label detection, higher is better; for data removal, lower is better, since the most valuable data are removed first. In each table, scores equal to or better than the baseline are shown in **bold**, and scores worse than the baseline by less than 0.01 are underlined.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.8534	0.8503	**0.8546**	**0.8552**	0.8519	0.8533	0.8522	0.8523
MNLI	0.4684	0.4659	**0.4752**	**0.4747**	**0.4793**	**0.4754**	**0.4862**	**0.4801**
MR	0.7804	**0.7815**	**0.7841**	**0.7817**	**0.7854**	0.7787	0.7787	0.7778
MRPC	0.6715	0.6686	**0.6730**	0.6711	**0.6731**	**0.6800**	**0.6783**	**0.6767**
QQP	0.6908	0.6761	0.6818	0.6876	0.6856	0.6879	0.6876	0.6872
RTE	0.5173	0.5036	0.5160	0.5136	**0.5224**	0.5156	**0.5267**	**0.5293**
SST-2	0.7179	**0.7599**	**0.7593**	**0.7486**	**0.7530**	**0.7521**	**0.7487**	**0.7464**

Table 11: The mean accuracy of `bert-base-uncased` when the selection percentage is 1-20%. Higher is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.8607	0.8561	0.8593	**0.8615**	0.8588	**0.8610**	**0.8613**	**0.8630**
MNLI	0.4094	0.3986	0.4068	0.4059	**0.4133**	**0.4095**	0.4082	0.4072
MR	0.8469	**0.8474**	**0.8482**	**0.8488**	0.8462	**0.8510**	**0.8483**	0.8429
MRPC	0.6687	**0.6691**	0.6687	**0.6707**	**0.6698**	0.6683	**0.6692**	0.6681
QQP	0.6819	**0.6925**	**0.7100**	**0.7098**	**0.7088**	**0.7100**	**0.7118**	**0.7163**
RTE	0.5417	0.5403	**0.5439**	**0.5466**	0.5281	**0.5423**	0.5411	0.5342
SST-2	0.8054	**0.8103**	0.8046	0.7938	0.7946	0.7924	0.7940	0.7976

Table 12: The mean accuracy of `meta-llama/Llama-2-7b-hf` when the selection percentage is 1-20%. Higher is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
CIFAR-10	0.7260	**0.7699**	**0.7575**	**0.7603**	**0.7597**	**0.7636**	**0.7618**	**0.7609**

Table 13: The mean accuracy of `resnet-18-pretrained` when the selection percentage is 1-20%. Higher is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.8880	**0.8867**	**0.8877**	**0.8874**	**0.8874**	**0.8870**	**0.8879**	**0.8870**
MNLI	0.5645	0.5694	0.5682	0.5693	0.5673	0.5681	0.5657	0.5673
MR	0.7865	0.7885	0.7897	0.7887	0.7871	**0.7840**	**0.7843**	**0.7845**
MRPC	0.7047	0.7095	0.7076	0.7075	0.7056	0.7080	0.7064	0.7081
QQP	0.7085	0.7150	0.7102	0.7107	0.7122	0.7133	0.7095	0.7101
RTE	0.5860	**0.5789**	**0.5794**	**0.5828**	**0.5809**	**0.5832**	**0.5787**	**0.5790**
SST-2	0.8394	0.8495	0.8449	0.8420	0.8435	0.8418	0.8412	0.8407

Table 14: The mean accuracy of `bert-base-uncased` when the removal percentage is 1-20%. Lower is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.8684	0.8701	0.8684	**0.8671**	**0.8683**	0.8685	**0.8681**	**0.8669**
MNLI	0.4438	0.4505	0.4537	0.4537	0.4520	0.4472	**0.4425**	0.4470
MR	0.8492	0.8493	**0.8470**	0.8500	**0.8490**	**0.8476**	**0.8465**	**0.8481**
MRPC	0.6291	**0.6278**	**0.6239**	**0.6261**	0.6305	0.6321	0.6317	**0.6267**
QQP	0.6884	**0.6882**	0.6887	**0.6838**	**0.6862**	**0.6871**	**0.6862**	**0.6838**
RTE	0.5636	0.5804	0.5795	0.5692	0.5730	0.5747	0.5697	0.5701
SST-2	0.8494	0.8499	0.8505	0.8495	**0.8468**	0.8494	**0.8448**	**0.8489**

Table 15: The mean accuracy of `meta-llama/Llama-2-7b-hf` when the removal percentage is 1-20%. Lower is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
CIFAR-10	0.8904	0.8971	0.8964	0.8947	0.8944	0.8919	0.8954	0.8944

Table 16: The mean accuracy of `resnet-18-pretrained` when the removal percentage is 1-20%. Lower is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.6314	0.6175	**0.6491**	**0.6480**	**0.6449**	**0.6392**	**0.6361**	**0.6378**
MNLI	0.2604	0.2221	0.2357	0.2395	0.2421	0.2448	0.2488	0.2525
MR	0.4606	0.4378	0.4517	0.4488	0.4527	0.4536	0.4454	0.4464
MRPC	0.2992	0.2724	0.2959	0.2964	**0.3021**	0.2975	**0.3027**	**0.3085**
QQP	0.3627	0.3296	0.3489	0.3557	**0.3645**	0.3621	**0.3678**	**0.3677**
RTE	0.1689	0.1493	0.1569	0.1624	**0.1733**	**0.1780**	**0.1717**	0.1664
SST-2	0.4031	0.3863	**0.4036**	**0.4189**	**0.4122**	**0.4150**	**0.4102**	**0.4038**

Table 17: The mean detection rate of `bert-base-uncased` when the inspection percentage is 1-20%. Higher is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
AG News	0.6661	0.6370	**0.6729**	**0.6756**	**0.6769**	**0.6756**	**0.6722**	**0.6683**
MNLI	0.1830	0.1364	0.1597	0.1774	0.1647	0.1812	0.1730	**0.1864**
MR	0.5842	0.5500	**0.5931**	**0.6016**	**0.5982**	**0.5990**	**0.5980**	**0.5965**
MRPC	0.2720	0.2238	0.2467	0.2614	**0.2746**	**0.2852**	**0.2755**	**0.2788**
QQP	0.3743	0.3319	**0.3965**	**0.4032**	**0.4062**	**0.4060**	**0.4063**	**0.4121**
RTE	0.1888	0.1639	0.1708	0.1651	0.1638	0.1843	**0.1945**	0.1766
SST-2	0.5270	0.4917	0.5209	**0.5350**	**0.5315**	**0.5295**	**0.5317**	**0.5291**

Table 18: The mean detection rate of `meta-llama/Llama-2-7b-hf` when the inspection percentage is 1-20%. Higher is better.

Dataset	FreeShap	Rank 1%	5%	10%	15%	20%	25%	30%
CIFAR-10	0.6255	0.6158	**0.6694**	**0.6742**	**0.6734**	**0.6706**	**0.6627**	**0.6614**

Table 19: The mean detection rate of `resnet-18-pretrained` when the inspection percentage is 1-20%. Higher is better.

M TMC CONVERGENCE RATIOS FOR ALL CASES

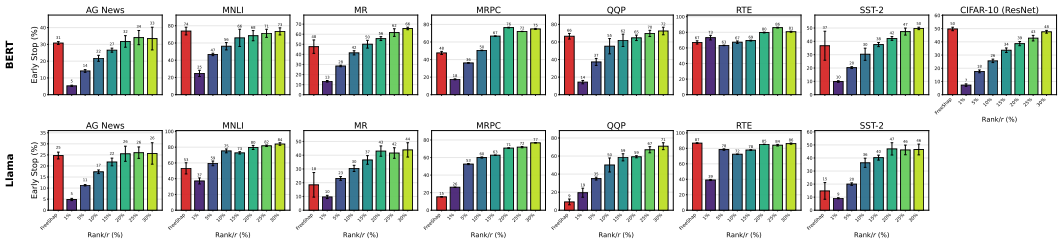

Figure 15: TMC convergence ratios for all datasets and models.