# Local Shapley: Model-Induced Locality and Optimal Reuse in Data Valuation

Xuan Yang
Duke University
Durham, NC, USA
xuan.yang@duke.edu

Hsi-Wen Chen
National Taiwan University
Taipei, Taiwan
hwchen@arbor.ee.ntu.edu.tw

Ming-Syan Chen
National Taiwan University
Taipei, Taiwan
mschen@ntu.edu.tw

Jian Pei
Duke University
Durham, NC, USA
j.pei@duke.edu

## ABSTRACT

The Shapley value provides a principled foundation for data valuation, but exact computation is #P-hard due to the exponential coalition space. Existing accelerations remain global and ignore a structural property of modern predictors: for a given test instance, only a small subset of training points influences the prediction. We formalize this **model-induced locality** through support sets defined by the model's computational pathway (e.g., neighbors in KNN, leaves in trees, receptive fields in GNNs), showing that Shapley computation can be projected onto these supports without loss when locality is exact. This reframes Shapley evaluation as a structured data processing problem over overlapping support-induced subset families rather than exhaustive coalition enumeration. We prove that the intrinsic complexity of Local Shapley is governed by the number of *distinct influential subsets*, establishing an information-theoretic lower bound on retraining operations. Guided by this result, we propose **LSMR** (**L**ocal **S**hapley via **M**odel **R**euse), an optimal subset-centric algorithm that trains each influential subset exactly once via support mapping and pivot scheduling. For larger supports, we develop **LSMR-A**, a reuse-aware Monte Carlo estimator that remains unbiased with exponential concentration, with runtime determined by the number of distinct sampled subsets rather than total draws. Experiments across multiple model families demonstrate substantial retraining reductions and speedups while preserving high valuation fidelity.

**PVLDB Reference Format:**
Xuan Yang, Hsi-Wen Chen, Ming-Syan Chen, and Jian Pei. Local Shapley: Model-Induced Locality and Optimal Reuse in Data Valuation. PVLDB, 19(11): 3330 - 3343, 2026.
doi:10.14778/3836663.3836692
**PVLDB Artifact Availability:**
The source code, data, and/or other artifacts have been made available at https://github.com/xuanyang19/Local_Shapley_Supplemental_Material.


Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836692

## 1 INTRODUCTION

Data has become a central resource for scientific discovery and commercial innovation [22, 29, 56]. As data commoditization accelerates, an increasing number of data marketplaces (e.g., Xignite, Snowflake, and Databricks) have emerged to facilitate the exchange and trading of datasets for model development across domains such as medicine, finance, and transportation [8, 15, 33, 36, 49]. Unlike traditional goods, however, the value of data is inherently context-dependent: a dataset is valuable only insofar as it improves downstream model performance. This creates a fundamental data management problem known as *data valuation*: how to rigorously quantify, attribute, and compute the contribution of individual data items to model performance in an efficient and fair manner when multiple parties jointly participate in model training [33, 49].

The Shapley value [53] provides a principled solution by assigning each data point its marginal contribution over all coalitions of the training data. It offers fairness guarantees and has become a standard foundation for data valuation [17][34]. Yet exact Shapley computation is #P-hard [13] because it requires evaluating an exponential number of subsets. Existing accelerations, including Monte Carlo sampling and truncated coalition evaluation [5, 17], influence- and gradient-based surrogates [1, 28], and per-trajectory gradient attribution [61], improve scalability but still rely on the *global* coalition space or a fixed training trajectory, implicitly treating every training point as potentially relevant to every test point and overlooking the structural sparsity of individual predictions.

This assumption is overly pessimistic. Modern predictors exhibit strong structural sparsity: for a fixed test instance, only a limited portion of the training data participates in the computational pathway that determines the prediction [28, 35, 47, 52, 66]. For example, KNN predictions depend on nearby neighbors; decision trees depend on leaf-level partitions; kernel and margin-based models depend on active supports; graph neural networks depend on receptive fields [11, 18, 37, 40]. For a given test point, a large portion of training points are effectively irrelevant for that prediction. From a data management perspective, this reveals a structural opportunity: Shapley computation should exploit model-induced locality to avoid evaluating coalitions that cannot affect the outcome.

We formalize this structural sparsity as **model-induced locality**. For each test point, we define a *support set* consisting of the training instances that influence its prediction through the model's computational graph. When this **locality** is exact, projecting the

coalition space onto the support region preserves the Shapley value. When locality is approximate, the deviation from global Shapley can be bounded by the aggregate influence of points outside the support. This reframes Shapley computation as a structured data processing problem: instead of enumerating $2^{|\mathcal{D}|}$ coalitions, we restrict attention to subsets of a much smaller support set.

However, locality alone is not sufficient. Even after restricting to supports, naïve computation remains exponential in the support size and suffers from extensive redundancy. The key insight of this paper is that the intrinsic complexity of Shapley valuation is governed not by the total number of coalitions, but by the number of *distinct subsets that influence at least one valuation*. We prove that any correct algorithm must evaluate each such subset at least once, establishing an information-theoretic lower bound on retraining complexity. This perspective shifts the problem from exponential coalition enumeration to optimal subset reuse.

Guided by this principle, we propose **LSMR** (for **L**ocal **S**hapley via **M**odel **R**euse), a subset-centric exact algorithm. LSMR builds a bipartite support-mapping graph that links subsets to the training and test points whose utilities depend on them, and applies a pivot-based scheduling rule so that each distinct subset is trained exactly once. The resulting utility is propagated to all dependent valuations. LSMR removes both intra-support and inter-support redundancy and attains the intrinsic lower bound on retraining cost.

For larger support sets where exact enumeration remains expensive, we introduce **LSMR-A**, a reuse-aware Monte Carlo estimator. Instead of treating each sampled coalition independently, LSMR-A shares every sampled subset across all compatible support sets. The estimator remains unbiased and enjoys *exponential concentration*, meaning that the probability of estimation error decreases exponentially fast with the number of samples. Importantly, its runtime depends on the number of *distinct sampled subsets* rather than the total number of draws. This decouples sampling complexity from retraining complexity and reduces variance through amortized reuse. Under *distribution shift*, when the test distribution differs from the training distribution and many training points become irrelevant to a given test instance, the variance advantage becomes *structural*: irrelevant points are never sampled, so unnecessary randomness is eliminated at its source rather than merely reduced statistically.

We evaluate our framework across four representative model families, including weighted KNN, RBF Kernel SVM, decision trees, and graph neural networks, and diverse datasets. Experiments validate the theoretical claims: support-induced locality preserves valuation fidelity; LSMR-A achieves faster convergence and dramatically reduces retraining, matching the intrinsic subset complexity.

In summary, this paper makes four main contributions. We introduce model-induced locality as a structural abstraction for data valuation, formalizing support sets and deriving bounds on the gap between global and local Shapley values. We characterize the intrinsic subset complexity induced by these supports and establish an information-theoretic lower bound on the number of retraining operations required by any correct algorithm. Guided by this principle, we propose **LSMR**, an optimal subset-centric algorithm that trains each influential subset exactly once using support mapping and pivot scheduling. Finally, we develop **LSMR-A**, a reuse-aware Monte Carlo estimator that decouples sampling from retraining, remains unbiased with exponential concentration, and achieves lower variance through structural reuse.

## 2 RELATED WORK

*Data Valuation and Scalable Shapley Approximations.* Data valuation [20, 25] quantifies the contribution of individual training examples to model performance, supporting applications such as data attribution [7], federated learning [9], model debugging [26], and dataset compression [69]. Early leave-one-out methods [28] measure the effect of removing one example at a time but cannot capture interactions among data points. Shapley-value-based methods [17, 54] address this limitation by aggregating marginal contributions over all coalitions, providing a cooperative-game-theoretic foundation for data valuation [53]. Since exact Shapley computation is #P-hard [13], prior work has developed scalable approximations [39, 46]. Monte Carlo estimators [5, 17, 41, 59] sample random permutations, reinforcement-learning methods [67] learn sampling policies, and structural shortcuts exploit convexity [24], submodularity [64], or influence-function reuse of gradients and Hessians [1, 50]. Recent methods also reduce or avoid retraining through amortization or trajectory-based approximation. Fast-DataShapley [57] and Learnable Data Shapley [31] train reusable neural predictors that map samples directly to Shapley values, while In-Run Data Shapley [61] accumulates gradient-based attribution along a single SGD trajectory using Taylor approximations. Despite these advances, existing methods either operate over the full global coalition space or bypass it through amortized approximations. They do not explicitly exploit model-induced sparsity in the prediction pathway of individual test points, nor do they characterize the intrinsic subset complexity induced by structural locality or establish lower bounds on the number of distinct retraining operations required for Shapley computation.

*Locality-Aware Shapley Computation.* To address the inefficiency of global coalition enumeration, several recent works explore locality-aware Shapley evaluation. For unweighted KNN classifiers, Jia et al. [23] observe that prediction at a test point depends primarily on its nearest neighbors and show that ignoring distant points preserves relative importance rankings, enabling efficient approximate local Shapley computation. Wang et al. [60] extend this idea to hard-label KNN via dynamic programming, and Zhang et al. [68] propose a near-linear-time weighted KNN-Shapley algorithm using a duplication method. While these methods highlight the practical benefits of restricting computation to local neighborhoods, their notion of locality is geometric and specific to KNN models. They do not generalize beyond distance-based neighborhoods, nor do they formalize locality as a structural property of the computational pathway of a model. In contrast, we introduce *model-induced support sets*, which abstract locality directly from the architecture and prediction mechanism of the model. This formulation applies broadly to margin-based models, decision trees, kernel methods, and graph neural networks. We further characterize the intrinsic family of distinct subsets induced by these supports and prove that any correct algorithm must evaluate each such subset at least once. By exploiting model-induced locality and subset reuse, our methods enable efficient Shapley valuation while preserving strong theoretical guarantees.

## 3 LOCAL SHAPLEY

We first review the standard Shapley value for data valuation, then introduce a localized variant that restricts computation to training points relevant to a given test prediction. Finally, we discuss how this locality notion extends across different model families. A complete notation reference is provided in Supplemental Material A.1.

### 3.1 Shapley Value for Data Valuation

Let $\mathcal{D}$ denote a training dataset, where each data point $z = (\mathbf{x}_z, y_z)$ consists of a feature vector $\mathbf{x}_z$ and a class label $y_z$. Data valuation [17] seeks to quantify how much each training point contributes to predictive performance on a testing set $\mathcal{T}$. Formally, we aim to construct a valuation function $\phi : \mathcal{D} \to \mathbb{R}$ that assigns an importance score to every $z \in \mathcal{D}$.

*Utility induced by retraining.* For each test point $t \in \mathcal{T}$, we define a utility function $v_t : 2^{\mathcal{D}} \to \mathbb{R}$, where $v_t(S)$ measures the performance of a model trained on a subset $S \subseteq \mathcal{D}$ when evaluated at $t$. Throughout the paper, we adopt the standard retraining-based formulation as follows

$$v_t(S) \;=\; g_t(\theta(S)), \tag{1}$$

where $\theta(S)$ denotes the model parameters obtained by training on $S$, and $g_t(\cdot)$ is a scalar evaluation functional at test point $t$. Examples of scalar evaluation functions include prediction confidence [23], signed margin [28], and loss reduction [17]. We assume that model hyperparameters and training procedures are fixed across coalitions, so that variation in $v_t(S)$ arises solely from the choice of training subset $S$ [17, 23, 63]. In the computational model considered throughout this work, the dominant cost of evaluating $v_t(S)$ is the training step required to obtain $\theta(S)$.

*Global Shapley value.* The Shapley value assigns to each training point its average marginal contribution over all possible coalitions [6, 53]. For a fixed test point $t$ and training point $z \in \mathcal{D}$, the Shapley value is

$$\phi_z(v_t) = \frac{1}{|\mathcal{D}|} \sum_{S \subseteq \mathcal{D}\setminus\{z\}} \frac{v_t(S \cup \{z\}) - v_t(S)}{\binom{|\mathcal{D}|-1}{|S|}}. \tag{2}$$

Equivalently, $\phi_z(v_t)$ is the expected marginal gain of $z$ under a uniformly random permutation of $\mathcal{D}$, and it satisfies the classical axioms of efficiency, symmetry, null player, and additivity [53].

When multiple test points are considered, we aggregate per-test-point values additively:

$$\phi_z = \sum_{t \in \mathcal{T}} \phi_z(v_t). \tag{3}$$

This summation corresponds to evaluating total contribution over $\mathcal{T}$. By linearity of the Shapley value [38], analysis can be conducted independently for each test point and combined afterward. Accordingly, we focus on a single fixed $t$ in the sequel.

*Computational barrier and structural redundancy.* Computing $\phi_z(v_t)$ requires evaluating marginal contributions over all $2^{|\mathcal{D}|}$ subsets of $\mathcal{D}$. Since each evaluation involves retraining the model on $S$ to obtain $\theta(S)$, exact computation is computationally prohibitive. In fact, computing the Shapley value of a single player in a general cooperative game is #P-hard [13].

Crucially, the classical formulation treats every training point as potentially influential for every test point across all coalitions. That is, the global coalition space $2^{\mathcal{D}}$ implicitly assumes that influence may propagate arbitrarily through retraining. However, modern learning architectures often exhibit strong structural sparsity: for a fixed test point $t$, only a small portion of the training data meaningfully participates in the computational pathway that determines $v_t(S)$. This observation suggests that the global coalition space contains substantial structural redundancy. The next subsection formalizes this idea by introducing a localized utility that restricts attention to the subset of training points relevant to $t$, leading to the notion of *Local Shapley value*.

### 3.2 Local Shapley Value

Modern learning systems often exhibit strong structural locality [28, 35, 47, 52, 66]: for a fixed test point $t$, only a limited portion of the training data participates in the computational pathway that determines $v_t(S)$. This observation suggests that the global coalition space $2^{\mathcal{D}}$ contains substantial redundancy when evaluating the contribution of training data to $t$.

*Support sets as structural locality.* To formalize this idea, we introduce a *support set* $\mathcal{N}(t) \subseteq \mathcal{D}$, defined as the subset of training points that can meaningfully influence the prediction or utility at $t$. The support set may be determined by architectural structure (e.g., receptive fields in GNNs, leaf membership in trees), margin sparsity (e.g., support vectors), kernel bandwidth, or other model-specific mechanisms. We treat $\mathcal{N}(t)$ as fixed with respect to a reference model trained once on the full dataset, so that locality reflects structural dependency rather than coalition-dependent drift. Its size can be controlled via model hyperparameters (e.g., $K$ in KNN [12], threshold in kernel methods [48], hop radius in GNN [27]), thereby trading fidelity for efficiency.

We define the *projected (local) utility* as follows.

$$v_t^{\mathcal{N}}(S) := v_t(S \cap \mathcal{N}(t)), \qquad \forall S \subseteq \mathcal{D}. \tag{4}$$

Since $v_t^{\mathcal{N}}(S)$ depends only on $S \cap \mathcal{N}(t)$, coalitions that agree on their intersection with $\mathcal{N}(t)$ are indistinguishable under the projected utility. Consequently, the cooperative game induced by $v_t^{\mathcal{N}}$ is effectively defined on the $2^{|\mathcal{N}(t)|}$ subsets of the support set, and players outside $\mathcal{N}(t)$ do not enlarge the intrinsic coalition space (formalized in Lemma 1).

**Definition 1** (Local Shapley Value)**.** *For a training point $z \in \mathcal{D}$, the local Shapley value with respect to test point $t$ is defined as the classical Shapley value computed under the projected utility $v_t^{\mathcal{N}}$:*

$$\phi_z^{\mathrm{loc}}(v_t) = \frac{1}{|\mathcal{D}|} \sum_{S \subseteq \mathcal{D}\setminus\{z\}} \frac{v_t^{\mathcal{N}}(S \cup \{z\}) - v_t^{\mathcal{N}}(S)}{\binom{|\mathcal{D}|-1}{|S|}}. \tag{5}$$

From Eq. (4), if $z \notin \mathcal{N}(t)$, then $v_t^{\mathcal{N}}(S \cup \{z\}) = v_t^{\mathcal{N}}(S)$ for all $S$, so $z$ acts as a null player and $\phi_z^{\mathrm{loc}}(v_t) = 0$. For $z \in \mathcal{N}(t)$, the valuation depends only on coalitions formed within the support set.

*When does locality approximate the global game?* Local projection modifies the cooperative game by removing non-support players. To quantify the resulting approximation error, we impose a stability condition controlling how non-local points affect marginal contributions.

**Assumption 1** (Additive Non-local Stability)**.** *Given a test point $t$, for any $z \in \mathcal{N}(t)$, there exist nonnegative constants $\{\ell_{t,z}(u)\}_{u \in \mathcal{D}\setminus\mathcal{N}(t)}$*

*such that for any* $S \subseteq \mathcal{N}(t) \setminus \{z\}$ *and any* $U \subseteq \mathcal{D} \setminus \mathcal{N}(t)$,

$$\left|\Delta_z v_t(S \cup U) - \Delta_z v_t(S)\right| \leq \sum_{u \in U} \ell_{t,z}(u),$$

*where* $\Delta_z v_t(S) := v_t(S \cup \{z\}) - v_t(S)$. □

This assumption formalizes a structural locality principle: non-support points influence the marginal effect of $z$ only weakly, and their aggregate impact scales at most additively [62]. This condition serves as the data-valuation counterpart to the classical algorithmic stability principle [3, 21]. For stable learning algorithms—such as regularized ERM, kernel methods, or SGD on smooth objectives—including a single data point perturbs the trained parameters by an amount that decays monotonically with the coalition size $|S|$. This parameter stability directly guarantees that adding non-support instances cannot cause volatile swings in a player's marginal contribution, thereby ensuring the variations are strictly bounded.

**Proposition 1** (Approximation of Global Shapley)**.** *Under Assumption 1, for any* $z \in \mathcal{N}(t)$,

$$\left|\phi_z(v_t) - \phi_z^{\mathrm{loc}}(v_t)\right| \leq \frac{1}{2} \sum_{u \in \mathcal{D} \setminus \mathcal{N}(t)} \ell_{t,z}(u),$$

*where* $\phi_z(v_t)$ *denotes the global Shapley value computed under the original utility* $v_t$. □

Using the permutation interpretation of the Shapley value, a non support point influences the marginal contribution of $z$ only when it appears before $z$ in a uniformly random permutation, which happens with probability one half. Invoking Assumption 1 together with the bound established in the proposition above, the deviation between the global and local valuations is bounded by one half of the total non local interaction mass. The approximation error therefore depends only on the aggregate strength of interactions outside the support, rather than on the size of the full training set. Hence, whenever such effects decay with distance, kernel bandwidth, or graph hop radius, a moderately sized support set is sufficient to maintain valuation fidelity.

*Example of regularized ERM.* Consider $\ell_2$-regularized empirical risk minimization $F_S(w) = \frac{1}{|S|}\sum_{i \in S} f_i(w) + \frac{\lambda}{2}\|w\|_2^2$, where each $f_i$ is convex and $L$-smooth. Then $F_S$ is $\lambda$-strongly convex, and classical uniform stability analysis [3] yields $\|w_{S \cup \{u\}} - w_S\|_2 = O\left(\frac{1}{\lambda |S|}\right)$. If the evaluation functional $g_t(w)$ is $\psi$-Lipschitz in $w$, then $|v_t(S \cup \{u\}) - v_t(S)| = O\left(\frac{\psi}{\lambda |S|}\right)$. Moreover, the interaction term $|\Delta_z v_t(S \cup \{u\}) - \Delta_z v_t(S)| = O\left(\frac{1}{\lambda |S|^2}\right)$ (up to alignment factors), which satisfies Assumption 1, where $\ell_{t,z}(u)$ decays rapidly as shown in Proposition 1. Consequently, the total non-local influence $\sum_{u \notin \mathcal{N}(t)} \ell_{t,z}(u)$ remains small when interactions are weak.

*Axiomatic properties.* An important question is whether restricting the game to the support set preserves the fundamental fairness guarantees of the Shapley framework.

**Proposition 2** (Properties of Local Shapley)**.** *The local Shapley value satisfies: (i)* ***Symmetry****: identically contributing data points receive equal value; (ii)* ***Null Player****: a data point with zero marginal contribution in all coalitions receives zero value; and (iii)* ***Additivity****: the valuation is linear in* $v_t$.

Since $\phi_z^{\mathrm{loc}}(v_t)$ is the classical Shapley value applied to the projected utility $v_t^{\mathcal{N}}$, it satisfies symmetry, null-player, and additivity. Efficiency holds with respect to the projected utility $v_t^{\mathcal{N}}$; relative to the original utility $v_t$, any deviation from global efficiency corresponds precisely to the total utility mass discarded by excluding points outside $\mathcal{N}(t)$. When locality is exact (e.g., Threshold KNN [63]), we have $v_t^{\mathcal{N}} = v_t$, and efficiency holds with respect to the original utility as well.

*Discussion and limitations.* The applicability of model-induced locality depends on the underlying architecture rather than being a guaranteed property of all learning systems. Highly non-convex or densely coupled architectures can exhibit global parameter dependencies where parameter shifts do not decay gracefully with coalition size, potentially widening the approximation gap in Proposition 1. Crucially, this limitation is manageable because the boundaries of the support set $\mathcal{N}(t)$ are inherently flexible. As detailed in Section 3.3, practitioners can systematically scale $\mathcal{N}(t)$ to capture long-range dependencies. Enlarging $\mathcal{N}(t)$ captures missing global interactions and tightens the theoretical approximation bound, offering a controllable trade-off between global fidelity and computational tractability.

## 3.3 Locality Induced by Model Architecture

Section 3.2 introduced the notion of support set $\mathcal{N}(t)$ and showed that, whenever the prediction at $t$ depends only on this subset (exactly or approximately), the effective cooperative game collapses from $2^{|\mathcal{D}|}$ to $2^{|\mathcal{N}(t)|}$. We now demonstrate that such support sets arise naturally in common model families as a consequence of their computational structure. In some cases locality is exact and Local Shapley coincides with the global Shapley value; in others locality is approximate but satisfies Assumption 1, yielding bounded error.

**Remark 1** (Sources of Locality)**.** *In many widely used predictors, the prediction at* $t$ *primarily depends on a structurally determined subset* $\mathcal{N}(t) \subseteq \mathcal{D}$*:*

1. ***KNN models*** *[12, 14]: The prediction is computed solely from the* $K$ *nearest neighbors of* $t$*, so* $\mathcal{N}(t)$ *is naturally defined as this* $K$*-point neighborhood. Locality is determined by Euclidean distance and is exact under the threshold setting [63].*
2. ***Support vector machines*** *[10, 11]: The prediction admits a sparse dual form involving only support vectors, i.e.,* $\mathcal{N}(t) = \{z \in \mathcal{D} : \alpha_z \neq 0\}$*. Locality is representational: although parameters are trained globally, only support vectors influence predictions.*
3. ***Kernel methods with compact support*** *[44, 48]: For kernels with finite bandwidth, only training points satisfying* $\|x_z - x_t\| \leq h$ *contribute, yielding* $\mathcal{N}(t) = \{z \in \mathcal{D} : \|x_z - x_t\| \leq h\}$*. Locality is governed by bandwidth.*
4. ***Decision trees*** *[4, 51]: The prediction at* $t$ *is determined by the leaf reached along its decision path. The relevant training points are those passing through the same parent node, and we define* $\mathcal{N}(t) = \{z \in \mathcal{D} : \ z \text{ reaches the same parent node as } t\}$*. Locality arises from rule-based partitioning of the input space.*
5. ***Graph neural networks*** *[18, 19, 27]: In an* $L$*-layer message-passing GNN, only nodes within the* $L$*-hop receptive field of* $t$ *directly influence its embedding, so* $\mathcal{N}(t)$ *is the* $L$*-hop neighborhood of* $t$*. Locality is structural and induced by graph topology.*

(6) ***Deep neural networks*** *[2, 47]: The learned encoder maps inputs into a representation space where semantically similar examples cluster together (e.g., via neural collapse). Consequently, $\mathcal{N}(t)$ is defined as the $K$-nearest training points to $t$ within this embedding space [65].*

Across these families, locality emerges through distinct mechanisms: geometric proximity (KNN), dual sparsity (SVM), finite bandwidth (compact kernels), rule-based partitioning (trees), graph message passing (GNNs), and representational clustering (deep networks). In each case, $\mathcal{N}(t)$ is determined by the pathway—explicit in the architecture or implicit in the learned representation—that produces the prediction at $t$, and typically satisfies $|\mathcal{N}(t)| \ll |\mathcal{D}|$. When this dependency is strictly finite, Local Shapley recovers the global Shapley value exactly; when the influence decays but is nonzero, the approximation gap is governed by the aggregate non-local interaction mass characterized in Proposition 1.

## 4 AN EXACT ALGORITHM FOR LOCAL SHAPLEY

Building on the Local Shapley formulation in Section 3.2, we now turn to exact computation of local Shapley values. Although Definition 1 replaces the original utility with the projected utility $v_t^{\mathcal{N}}$, its combinatorial form still ranges over all $2^{|\mathcal{D}|}$ coalitions. We first prove that the projected utility induces an equivalent cooperative game whose effective player set is the support set $\mathcal{N}(t)$, thereby collapsing the intrinsic coalition space from $2^{|\mathcal{D}|}$ to $2^{|\mathcal{N}(t)|}$.

This reduction alone, however, does not eliminate computational redundancy. A naïve local evaluation still enumerates all subsets of $\mathcal{N}(t)$ separately for each training and test point. As shown in Figure 1, two redundancies and inefficiencies remain: *intra-support redundancy*, where training points within the same support set require repeated evaluation of identical subsets, and *inter-support redundancy*, where overlapping support sets across test points trigger duplicate subset retraining. In this section, we eliminate these redundancies in a structured manner: first by reorganizing the Shapley computation around subsets rather than players, and then by introducing a global reuse mechanism that guarantees each distinct subset is trained once while preserving exactness.

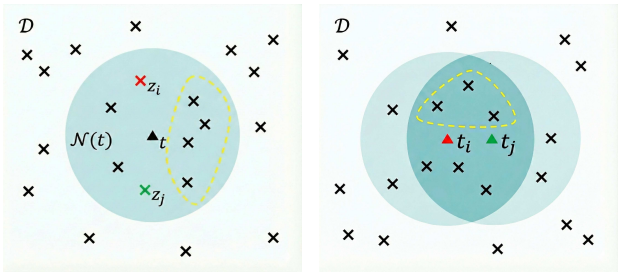
### 4.1 Reducing the Game to the Support Set

We begin by formalizing the intrinsic dimensional reduction induced by the projected utility. Although Definition 1 is written over subsets of the full dataset $\mathcal{D}$, the projected utility satisfies $v_t^{\mathcal{N}}(S) = v_t(S \cap \mathcal{N}(t))$, implying that coalitions that agree on their intersection with $\mathcal{N}(t)$ are indistinguishable. Consequently, the Shapley computation for test point $t$ depends only on subsets of the support set $\mathcal{N}(t)$.

**Lemma 1** (Equivalence Under Projected Utility)**.** *The local Shapley value in Definition 1, can be equivalently expressed as*

$$\phi_z^{\mathrm{loc}}(v_t) = \frac{1}{|\mathcal{N}(t)|} \sum_{S \subseteq \mathcal{N}(t) \setminus \{z\}} \frac{v_t(S \cup \{z\}) - v_t(S)}{\binom{|\mathcal{N}(t)|-1}{|S|}}. \qquad \square$$

Lemma 1 shows that the Local Shapley value for $t$ can be computed by treating $\mathcal{N}(t)$ as the effective player set. Thus, the coalition space collapses from $2^{|\mathcal{D}|}$ to $2^{|\mathcal{N}(t)|}$. For subsets of size $k$, there are $\binom{|\mathcal{N}(t)|}{k}$ coalitions. As each marginal contribution requires evaluating $v(S)$ and $v(S \cup \{z\})$, enumeration over the reduced space yields the training cost as $2\sum_{k=1}^{|\mathcal{N}(t)|} k\binom{|\mathcal{N}(t)|}{k} = |\mathcal{N}(t)| \cdot 2^{|\mathcal{N}(t)|}$. Aggregating over all test points, the total number of evaluations becomes $\sum_{t \in \mathcal{T}} |\mathcal{N}(t)| \cdot 2^{|\mathcal{N}(t)|}$, which is much smaller than the $|\mathcal{T}| \cdot |\mathcal{D}| \cdot 2^{|\mathcal{D}|}$ cost of global Shapley when $|\mathcal{N}(t)| \ll |\mathcal{D}|$. This serves as the natural local baseline and is summarized in Algorithm 1.




(a) Intra-support redundancy (b) Inter-support redundancy

**Figure 1: (a) *Intra-support redundancy*: the blue area denotes the support set $\mathcal{N}(t)$. For training points $z_i$ and $z_j$, many training subsets are shared in their Shapley value computations. (b) *Inter-support redundancy*: test points $t_i$ and $t_j$ have overlapping supports (darker blue). Within this overlap, subsets are shared across their Shapley computations. The yellow dotted subset illustrates one such shared training subset.**

**Algorithm 1** Baseline Algorithm

**Require:** Training set $\mathcal{D}$, test set $\mathcal{T}$
**Ensure:** Local Shapley value $\phi_z$ for each $z \in \mathcal{D}$

```
for each z ∈ D do
    initialize φ_z ← 0
for each test point t ∈ T do
    Retrieve support set N(t)
    for each training point z ∈ N(t) do
        for each subset S ⊆ N(t) \ {z} do
            Train model θ(S) and evaluate v_t(S);
            Train model θ(S ∪ {z}) and evaluate v_t(S ∪ {z});
            φ_z ← φ_z + (v_t(S∪{z}) − v_t(S)) / (|N(t)| · C(|N(t)|−1, |S|))
return {φ_z : z ∈ D}
```

### 4.2 Subset-Centric Reformulation: Eliminating Intra-Support Redundancy

The baseline method (Algorithm 1) evaluates marginal contributions independently for each training point within the support set $\mathcal{N}(t)$. Thus, identical training subsets are repeatedly retrained when computing the marginal gains of different players within the same support set. To eliminate this intra-support redundancy, we break away from traditional player-centric loops and reorganize the valuation framework entirely around subsets. The key intuition is that a single utility evaluation $v_t(S)$ contains complete combinatorial information that can be distributed to all players in $\mathcal{N}(t)$ simultaneously, bypassing the need for separate marginal queries.

**Algorithm 2** LSMR

**Input:** Bipartite support-mapping structure $G = (\mathcal{D}, \mathcal{T}, \mathcal{N}, \mathcal{R})$, ordered test set $\Pi_{\mathcal{T}}$
**Output:** Local Shapley values $\phi_z$ for each $z \in \mathcal{D}$
Initialize $\phi_z \leftarrow 0$ for all $z \in \mathcal{D}$
**for all** test point $t \in \mathcal{T}$ in order $\Pi_{\mathcal{T}}$ **do**
  Retrieve support set $\mathcal{N}(t)$ from $G$
  **for all** subsets $S \subseteq \mathcal{N}(t)$ **do**
    **if** $S = \emptyset$ **then**
      $\mathcal{R}_S \leftarrow \mathcal{T}$
    **else**
      $\mathcal{R}_S \leftarrow \bigcap_{z \in S} \mathcal{R}(z)$
    $t^* \leftarrow$ first test point in $\mathcal{R}_S$ under $\Pi_{\mathcal{T}}$
    **if** $t = t^*$ **then**
      Train model $\theta(S)$ and evaluate $v_{t'}(S)$ for all $t' \in \mathcal{R}_S$
      **for all** $t' \in \mathcal{R}_S$ **do**
        **for all** $z \in \mathcal{N}(t')$ **do**
          **if** $z \in S$ **then**
            $\phi_z \mathrel{+}= \frac{v_{t'}(S)}{|\mathcal{N}(t')| \cdot \binom{|\mathcal{N}(t')|-1}{|S|-1}}$
          **else**
            $\phi_z \mathrel{-}= \frac{v_{t'}(S)}{|\mathcal{N}(t')| \cdot \binom{|\mathcal{N}(t')|-1}{|S|}}$
**return** $\{\phi_z : z \in \mathcal{D}\}$

**Lemma 2** (Subset-Centric Reformulation)**.** *For any test point $t$ and training point $z \in \mathcal{N}(t)$, the local Shapley value admits the following equivalent expression:*

$$\phi_z^{\text{loc}}(v_t) = \frac{1}{|\mathcal{N}(t)|} \sum_{S \subseteq \mathcal{N}(t)} \frac{(-1)^{\mathbb{I}(z \notin S)}\, v_t(S)}{\binom{|\mathcal{N}(t)|-1}{|S|-\mathbb{I}(z \in S)}}. \qquad \square$$

Instead of computing differences ($v(S \cup \{z\}) - v(S)$), Lemma 2 reformulates Local Shapley as a structured, weighted sum directly over the power set of the support region. The role of each subset $S$ is completely determined by whether the training point $z$ is contained within it. There are two cases: (i) **Positive Contributions ($z \in S$)**: Subsets containing $z$ correspond to valid coalitions **after** including the player, receiving a positive sign; and (ii) **Negative Contributions ($z \notin S$)**: Subsets lacking $z$ correspond to base coalitions **prior** to its addition, receiving a negative sign.

Crucially, evaluating $v_t(S)$ once allows us to update the values of all players in $\mathcal{N}(t)$ concurrently using these closed-form weights. This algorithmic shift yields an immediate computational dividend: instead of demanding $|\mathcal{N}(t)| \cdot 2^{|\mathcal{N}(t)|}$ utility evaluations per test point via player-centric loops, we evaluate only $2^{|\mathcal{N}(t)|}$ distinct local subsets. Intra-support redundancy is thus completely eradicated, dropping the computational overhead by an exact factor of $|\mathcal{N}(t)|$.

At this point, the computation is localized to $\mathcal{N}(t)$ and reorganized in a subset-centric manner, eliminating redundancy within each support set. Nevertheless, subsets that appear in overlapping supports of different test points are still evaluated independently. The next subsection addresses this remaining inefficiency by introducing a global reuse mechanism that ensures each distinct subset is trained only once across all test points.

## 4.3 Global Subset Reuse Across Test Points

While subset-centric processing clears redundancies within an individual local game, nearby test points sampled from a shared data distribution still often exhibit heavily overlapping neighborhoods (supports). This overlap triggers duplicate evaluations of identical subsets across independent test iterations, inflating runtime because each unique evaluation requires an expensive model retraining step ($\theta(S)$).

To eliminate this remaining redundancy, we introduce **Local Shapley via Model Reuse (LSMR)**, which ensures that every distinct subset $S$ is trained exactly once across the entire computation and reused wherever it is valid. The key principle is global subset canonicalization: each subset has a unique designated evaluator, and all other occurrences reuse its result. LSMR achieves this through three components: a bipartite support-mapping graph, reverse support-set indexing, and pivot-based scheduling.

*4.3.1 Global Support Structure.* We represent training–test dependencies using a bipartite structure $G = (\mathcal{D}, \mathcal{T}, \mathcal{N}, \mathcal{R})$, where $\mathcal{N}(t) \subseteq \mathcal{D}$ is the support of test point $t$, and

$$\mathcal{R}(z) = \{\, t \in \mathcal{T} : z \in \mathcal{N}(t) \,\} \tag{6}$$

is the reverse map listing the test points whose supports contain training point $z$. The mappings $\mathcal{N}$ and $\mathcal{R}$ are relational inverses, and together they characterize all potential reuse opportunities across test points.

*4.3.2 Reverse Support-Set Indexing.* For any subset $S \subseteq \mathcal{D}$, only test points whose supports contain $S$ can reuse its evaluation. These test points are exactly

$$\mathcal{R}_S = \bigcap_{z \in S} \mathcal{R}(z), \qquad \mathcal{R}_\emptyset = \mathcal{T}. \tag{7}$$

$\mathcal{R}_S$ contains all test points for which $S \subseteq \mathcal{N}(t)$. Computing $\mathcal{R}_S$ requires intersecting adjacency lists, whose cost is negligible compared to model retraining as supports are typically small and sparse.

*4.3.3 Pivot-Based Scheduling.* To eliminate redundant trainings, LSMR assigns each subset $S$ a canonical evaluator. Intuitively, the first test point (under a fixed ordering) whose support contains $S$ "owns" the training of $S$; all later test points reuse the result. Fix a global ordering $\Pi_{\mathcal{T}}$ over test points and define the pivot

$$t^*(S) = \arg\min_{t' \in \mathcal{R}_S} \Pi_{\mathcal{T}}(t'). \tag{8}$$

Subset $S$ is trained only when processing its pivot $t^*(S)$. For any other $t' \in \mathcal{R}_S$, the subset has already been evaluated at its pivot and the result is reused. This rule ensures that each distinct subset is trained exactly once, eliminating all inter-support redundancy.

*4.3.4 Complete LSMR Procedure.* Combining subset-centric reformulation with global reuse yields a unified pipeline. For each test point $t$, LSMR enumerates all subsets $S \subseteq \mathcal{N}(t)$. When encountering $S$, the algorithm computes $\mathcal{R}_S$ and checks whether $t$ is the pivot $t^*(S)$. If so, it trains $\theta(S)$, evaluates $v_{t'}(S)$ for all $t' \in \mathcal{R}_S$ and redistributes the utility values to all $z \in \mathcal{N}(t')$ using the closed-form weights from Lemma 2; otherwise, the subset has already been processed and no retraining occurs.

By construction, LSMR removes both intra-support redundancy (via subset-centric reformulation) and inter-support redundancy

(via global pivot reuse), ensuring that each distinct subset $S$ is trained exactly once while preserving exact Local Shapley values.

### 4.4 Optimality and Complexity Analysis

We now quantify the computational savings of global subset reuse. Since the dominant cost is model retraining, we measure complexity by counting the number of distinct subsets $S$ for which $\theta(S)$ must be trained. Let

$$\mathcal{S} = \bigcup_{t \in \mathcal{T}} \{\, S \subseteq \mathcal{N}(t) \,\} \tag{9}$$

denote the family of all distinct subsets that appear across support sets. The size $|\mathcal{S}|$ thus reflects the intrinsic retraining complexity of the Local Shapley problem under a given support mapping.

**Theorem 1** (Optimal Reuse). *Let $\mathcal{R}_S$ be the set of test points whose support sets contain $S$. Any algorithm computing $\{v_t(S)\}_{t \in \mathcal{R}_S}$ must evaluate $v(S)$ at least once for every $S \in \mathcal{S}$.*

Theorem 1 shows that any correct algorithm must evaluate each subset in $\mathcal{S}$ at least once. Thus $|\mathcal{S}|$ is an information-theoretic lower bound on the number of model trainings. By construction, LSMR attains this bound exactly, performing one training per distinct subset and no more. By definition, we have

$$|\mathcal{S}| \le \min\left\{ \sum_{t \in \mathcal{T}} 2^{|\mathcal{N}(t)|},\ 2^{|\mathcal{D}|} \right\}. \tag{10}$$

The first term corresponds to independently enumerating all subsets within each support set, while the second reflects the trivial upper bound over the entire dataset. In contrast, LSMR performs exactly $|\mathcal{S}|$ trainings, which can be substantially smaller when supports overlap. We therefore conclude across four levels of evaluation:

- **Global Shapley:** $|\mathcal{T}| \cdot |\mathcal{D}| \cdot 2^{|\mathcal{D}|}$ trainings.
- **Local baseline:** $\sum_{t \in \mathcal{T}} |\mathcal{N}(t)| \cdot 2^{|\mathcal{N}(t)|}$ trainings.
- **Subset-centric:** $\sum_{t \in \mathcal{T}} 2^{|\mathcal{N}(t)|}$ trainings.
- **LSMR:** $|\mathcal{S}|$ trainings.

The final quantity depends only on the number of *distinct* subsets induced by all supports, not on the number of test points or the number of players per support.

In practical regimes where test points are drawn from a fixed distribution, supports induced by nearby test points tend to overlap significantly. As $|\mathcal{T}|$ grows, many new test points introduce no new subsets, and the growth of $|\mathcal{S}|$ is sublinear relative to $\sum_{t \in \mathcal{T}} 2^{|\mathcal{N}(t)|}$. Consequently, the amortized number of model trainings per test point decreases with scale, making LSMR particularly effective in large evaluation settings.

**Corollary 1** (Amortized Reuse). *Assume the support mapping $\mathcal{N}(\cdot)$ has a finite range, i.e., only finitely many distinct support sets can appear across test points. Let $\mathcal{S}_T = \bigcup_{t \in \mathcal{T}_T} \{\, S \subseteq \mathcal{N}(t) \,\}$ denote the family of distinct subsets induced by the first $T$ test points. Then $|\mathcal{S}_T|$ is bounded as $T$ grows, and consequently $\frac{|\mathcal{S}_T|}{T} \to 0$ as $T \to \infty$.* □

Therefore, the amortized number of model trainings per test point under LSMR vanishes as $T$ grows.

## 5 MONTE CARLO APPROXIMATION FOR LOCAL SHAPLEY

While LSMR achieves the intrinsic retraining lower bound and serves as the optimality reference for our analysis, its $2^{|\mathcal{N}(t)|}$ enumeration is impractical when supports are large. To tackle the scenarios with large supports, we therefore propose **LSMR-A**, the reuse-aware Monte Carlo estimator developed in this section. For stochastic approximation, classical Monte Carlo methods [43] estimate Shapley values by sampling permutations and averaging marginal contributions to avoid exhaustive enumeration. However, standard MC is *player-centric*, retraining each sampled coalition independently and repeatedly, failing to exploit the structural reuse opportunities identified in Section 4.3. This insight motivates a reuse-aware Monte Carlo strategy.

### 5.1 From Exact Reuse to Reuse-Aware Sampling

We therefore develop **LSMR-A**, a stochastic estimator that integrates the subset-centric formulation and pivot-based canonicalization of LSMR into the Monte Carlo setting. The resulting algorithm preserves unbiasedness and concentration guarantees while ensuring that each distinct subset is trained at most once across all samples and test points.

*5.1.1 Subset-Centric Monte Carlo Estimation.* Standard permutation-based MC estimates the Shapley value of a player $z$ by averaging marginal gains $v_t(S \cup \{z\}) - v_t(S)$ over sampled subsets $S$. This formulation ties each sample to a specific player, making cross-player reuse impossible. In contrast, Lemma 2 expresses the Local Shapley value as a weighted expectation over subsets $S \subseteq \mathcal{N}(t)$. This observation enables a conceptual shift from a marginal-gain estimator to a *subset-centric weighted utility estimator*. Instead of associating each sample with a single player, LSMR-A samples subsets $S$ and updates the Shapley values of *all* players in $\mathcal{N}(t)$ simultaneously using the closed-form weights derived in Lemma 2.

Under this formulation, a single model training on $S$ becomes a shared computational resource for the entire support set, eliminating intra-support redundancy in the stochastic setting.

*5.1.2 Pivoted Sampling for Cross-Test Reuse.* To extend reuse across test points, we integrate the pivot mechanism introduced in Section 4.3. For each test point $t$, LSMR-A draws $M$ random permutations of $\mathcal{N}(t)$ and extracts the prefix-before-$t$ subset $S$ in the standard Shapley sampling manner.

For a sampled subset $S$, only test points whose supports contain $S$ can reuse its evaluation. These are precisely the elements of $\mathcal{R}_S$. To keep the estimator unbiased, we assign each subset a canonical evaluator via the pivot rule as follows.

$$t^*(S) = \arg\min_{t' \in \mathcal{R}_S} \Pi_{\mathcal{T}}(t'). \tag{11}$$

The sampling procedure follows two cases: (i) **Pivot hit:** If the current test point $t$ equals $t^*(S)$, the subset is accepted. The model $\theta(S)$ is trained once and reused to evaluate $v_{t'}(S)$ for all $t' \in \mathcal{R}_S$. These utilities are distributed to all $z \in \mathcal{N}(t')$ using the weighted subset-centric update rule; and (ii) **Pivot miss:** If $t \neq t^*(S)$, the sample is discarded. Since $S$ is assigned exclusively to its pivot $t^*(S)$, discarding avoids duplicate retraining without altering the sampling distribution.

**Algorithm 3** LSMR-A

**Input:** Bipartite support-mapping structure $G = (\mathcal{D}, \mathcal{T}, \mathcal{N}, \mathcal{R})$, ordered test set $\Pi_{\mathcal{T}}$, number of Monte Carlo samples $M$
**Output:** Local Shapley values $\phi_z$ for each $z \in \mathcal{D}$
Initialize $\phi_z \leftarrow 0$ for all $z \in \mathcal{D}$
**for all** test point $t \in \mathcal{T}$ in order $\Pi_{\mathcal{T}}$ **do**
  Retrieve support set $\mathcal{N}(t)$ from $G$
  **for** $m = 1$ to $M$ **do**
    Sample a random permutation $\pi$ of $\mathcal{N}(t) \cup \{t\}$
    $S \leftarrow \{\pi_1, \ldots, \pi_{i-1}\}$ where $i = \pi^{-1}(t)$
    Compute $\mathcal{R}_S$ and pivot $t^*$ as in Algorithm 2, lines 7–11
    **if** $t = t^*$ **then**
      Train model $\theta(S)$ and evaluate $v_{t'}(S)$ for all $t' \in \mathcal{R}_S$
      **for** all $t' \in \mathcal{R}_S$, $z \in \mathcal{N}(t')$ **do**
        **if** $z \in S$ **then**
          $\phi_z \mathrel{+}= \dfrac{(|\mathcal{N}(t')| + 1) \cdot v_{t'}(S)}{|S| \cdot M}$
        **else**
          $\phi_z \mathrel{-}= \dfrac{(|\mathcal{N}(t')| + 1) \cdot v_{t'}(S)}{(|\mathcal{N}(t')| - |S|) \cdot M}$
**return** $\{\phi_z : z \in \mathcal{D}\}$

This pivoted sampling scheme maximizes the reuse of trained models while preserving the unbiasedness of the estimator. The complete procedure is summarized in Algorithm 3.

## 5.2 Statistical Guarantees

We first analyze the statistical validity and convergence properties of LSMR-A. The resulting guarantees mirror classical Monte Carlo analysis, while incorporating the effect of pivot-based reuse. We begin by establishing unbiasedness in expectation.

**Theorem 2** (Unbiasedness of LSMR-A)**.** *In each Monte Carlo round, LSMR-A draws a uniform permutation of $\mathcal{N}(t) \cup \{t\}$ and takes the prefix-before-$t$ subset $S$. The pivot associated with $S$ is the first test point in the fixed ordering $\Pi_{\mathcal{T}}$, denoted $t^*(S)$. The model is trained on $S$ only when processing $t = t^*(S)$, and the resulting utilities $v_{t'}(S)$ are reused for all $t'$ such that $S \subseteq \mathcal{N}(t')$. Under this reuse mechanism, a MC estimator satisfies*

$$\mathbb{E}\Big[\widehat{\phi}(z)\Big] = \sum_{t \in \mathcal{T}} \frac{\mathbb{I}(z \in \mathcal{N}(t))}{|\mathcal{N}(t)|} \sum_{S \subseteq \mathcal{N}(t)} \frac{(-1)^{\mathbb{I}(z \notin S)}\, v_t(S)}{\binom{|\mathcal{N}(t)|-1}{|S|-\mathbb{I}(z \in S)}},$$

*which coincides exactly with the closed-form Local Shapley value. Thus, LSMR-A is an unbiased estimator.* □

The estimator remains unbiased because each subset $S$ is drawn from the same permutation-induced distribution as in classical MC, and the pivot rule merely canonicalizes which test point evaluates it. The weighted subset-centric update from Lemma 2 preserves the exact expectation of the Local Shapley value.

We then prove exponential concentration of the estimator around the true Local Shapley value. The guarantee follows from independent permutation sampling together with standard bounded-difference inequalities.

**Theorem 3** (Concentration of LSMR-A)**.** *Assume $|v_t(S)| \le B$ for all $t$ and all subsets $S$. Let $\widehat{\phi}(z)$ be the estimator obtained from $M$ samples. Then for any $\epsilon > 0$,*

$$\Pr\Big(\big|\widehat{\phi}(z) - \phi_z^{\mathrm{loc}}\big| > \epsilon\Big) \le 2\exp\Big(-\frac{2M\epsilon^2}{B^2}\Big).$$ □

The concentration bound immediately yields a sample complexity guarantee.

**Corollary 2** (Sample Complexity of LSMR-A)**.** *To ensure*

$$\Pr\Big(\big|\widehat{\phi}(z) - \phi_z^{\mathrm{loc}}\big| > \epsilon\Big) \le \delta,$$

*it suffices to take*

$$M \;\ge\; \frac{B^2}{2\epsilon^2} \log \frac{2}{\delta}.$$

*Moreover, due to subset reuse across test points, the effective number of required model trainings satisfies $M_{\mathrm{eff}} = \frac{M}{\mathbb{E}[\,|\mathcal{T}_S|\,]}$, where the expectation is taken over sampled subsets $S$.* □

Importantly, the number of samples $M$ required for $(\epsilon, \delta)$-accuracy matches classical MC. The distinction between LSMR-A and standard MC lies not in statistical efficiency, but in retraining efficiency.

## 5.3 Computational Implications of Reuse

While $M$ determines statistical accuracy, runtime is governed by the number of *distinct* subsets encountered during sampling, since each such subset is trained at most once.

**Theorem 4** (Expected Number of Distinct Sampled Subsets)**.** *Fix a test point $t$ with support set $\mathcal{N}(t)$. Let $M$ rounds sample random permutations of $\mathcal{N}(t) \cup \{t\}$, and let $U_t$ be the collection of distinct prefix-before-$t$ subsets. Then, $\mathbb{E}\,|U_t| = O\Big(\min\{2^{|\mathcal{N}(t)|}, \sqrt{M}\}\Big)$.* □

The sublinear growth in Theorem 4 reflects a saturation effect. Although many permutations are sampled, a large portion lead to overlapped subsets, so retraining increases far more slowly than the total sampling effort. Consequently, the effective number of model trainings is governed by the number of distinct subsets rather than by $M|\mathcal{T}|$. When supports overlap substantially, reuse dramatically reduces retraining cost while preserving statistical guarantees.

## 5.4 Variance Reduction and Distribution Shift

Beyond correctness and runtime, reuse also improves estimator stability. Because utilities are shared across overlapping support regions, randomness that would otherwise be repeatedly injected in classical MC is amortized.

**Theorem 5** (Variance Reduction)**.** *Let $\widehat{\phi}^{\mathrm{MC}}(z)$ be the classical Monte Carlo estimator and $\widehat{\phi}^{\text{LSMR-A}}(z)$ be the LSMR-A estimator. Then*

$$\begin{aligned}\mathrm{Var}\Big[\widehat{\phi}^{\text{LSMR-A}}(z)\Big] \;&\le\; \mathrm{Var}\Big[\widehat{\phi}^{\mathrm{MC}}(z)\Big] - \mathbb{E}[\mathrm{Var}(v_t(S) \mid S)] \\ &\le\; \mathrm{Var}\Big[\widehat{\phi}^{\mathrm{MC}}(z)\Big].\end{aligned}$$ □

The variance reduction stems from removing conditional variance caused by redundant marginal evaluations. For the same sampling budget $M$, LSMR-A attains lower variance than classical MC.

This advantage becomes more pronounced under distribution shift. When the test distribution diverges from the training distribution, many training points lie outside the support set $\mathcal{N}(t)$ and have no influence on $v_t(\cdot)$. Classical MC nonetheless samples coalitions containing such irrelevant points, introducing unnecessary conditional variance. In contrast, LSMR-A restricts sampling to subsets of $\mathcal{N}(t)$ and therefore structurally removes this source of variance.

**Corollary 3** (Distribution Shift). *For a test point $t$ with support set $\mathcal{N}(t)$ and irrelevant points $\mathcal{D}_{\text{irr}} = \mathcal{D} \setminus \mathcal{N}(t)$, the variance of the Monte Carlo estimator decomposes as*

$$\begin{aligned}\text{Var}\left[\widehat{\phi}^{\text{MC}}(z)\right] &= \text{Var}\left[\widehat{\phi}^{\text{MC}}(z) \mid S \subseteq \mathcal{N}(t)\right] \\ &\quad + \text{Var}\left(v_t(S \cup \{z\}) - v_t(S) \,\middle|\, S \cap \mathcal{D}_{\text{irr}} \neq \varnothing\right).\end{aligned}$$

*Since LSMR-A samples only subsets of $\mathcal{N}(t)$, the second term is always zero for $\widehat{\phi}^{\text{LSMR-A}}(z)$.* □

In summary, LSMR-A provides unbiased estimation with exponential convergence, retraining complexity determined by distinct subsets rather than the total number of samples, and strictly reduced variance under overlapping supports and distribution shift.

# 6 EMPIRICAL EVALUATION

We conduct a comprehensive empirical evaluation of the Local Shapley framework across diverse model families. Our experiments are designed not only to assess predictive fidelity, but also to validate the structural and computational claims established in Sections 3–5, including intrinsic subset complexity, optimal reuse and retraining efficiency. We aim to answer five central research questions:

- **RQ1 (Approximation Fidelity):** How accurately does Local Shapley approximate the global Shapley value across different model families, and under what conditions does structural locality preserve valuation quality?
- **RQ2 (Downstream Data Selection Utility):** Do valuations derived from Local Shapley remain effective for downstream tasks such as data selection, compared with global and alternative Shapley estimators?
- **RQ3 (Optimal Reuse for Efficiency):** How much does LSMR-A reduce model trainings and runtime compared to baselines, and how closely does its behavior match the intrinsic subset complexity in Theorem 1?
- **RQ4 (Sensitivity to Support Set Size):** How do Local Shapley fidelity and runtime change as the support size varies, and do the observed fidelity–runtime trade-offs agree with the theoretical analysis in Sections 4 and 5?
- **RQ5 (Model-Induced Locality):** Does the support set need to align with the evaluated model? We construct support sets using one architecture (e.g., GNN), and evaluate utility under another (e.g., KNN), as discussed in Remark 1.

## 6.1 Experimental Setup

*6.1.1 Locality Across Model Families and Datasets.* To examine the generality of model-induced support sets, we instantiate locality across four representative model classes, each paired with a structurally aligned dataset: i) **Weighted $K$-Nearest Neighbors (WKNN)** [14] on **MNIST** [30], with $\mathcal{N}(t)$ defined as the $2K$ nearest neighbors in feature space; ii) **Decision Tree (DT)** [4] on **Iris** [16], with $\mathcal{N}(t)$ consisting of training instances that share the same parent node as the leaf reached by $t$; iii) **RBF Kernel SVM (RBF-SVM)** [11] on **Breast Cancer** [55], with $\mathcal{N}(t) = \{z \in \mathcal{D} : K(x_z, x_t) \geq 0.5\}$; iv) **Graph Neural Network (GNN)** [27] on **Cora** [42], with $\mathcal{N}(t)$ defined as the two-hop neighbors of $t$ under a two-layer GCN. These models span geometric proximity, kernel-induced decay, rule-based partitioning, and graph topology. For each model, the support set is constructed directly from its computational pathway, enabling measurement of support size and the associated retraining complexity.

*6.1.2 Baselines.* We compare LSMR-A against four representative Shapley estimation strategies spanning global evaluation, locality restriction, truncation, and complementary reformulation: i) **Global-MC** [24], which performs standard Monte Carlo estimation; ii) **Local-MC**, which restricts Monte Carlo sampling to the support set $\mathcal{N}(t)$ without reuse; iii) **TMC-S** [17], which applies truncated Monte Carlo with early stopping once predictions stabilize; iv) **Comple-S** [58], which estimates Shapley values via complementary contribution reformulation. These baselines do not incorporate model-induced locality or reuse-based retraining reduction.

*6.1.3 Evaluation Metrics and Implementation Details.* We evaluate two foundational aspects of Shapley value computation: i) **Fidelity** (correlation with global Shapley values and downstream task performance), ii) **Retraining Cost** (total running time and number of model trainings). Following [17], all Monte Carlo methods use the same stopping rule: every 100 samples, we test whether

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\phi_i^{(m)} - \phi_i^{(m-100)}|}{|\phi_i^{(m)}| + \epsilon} < \tau, \tag{12}$$

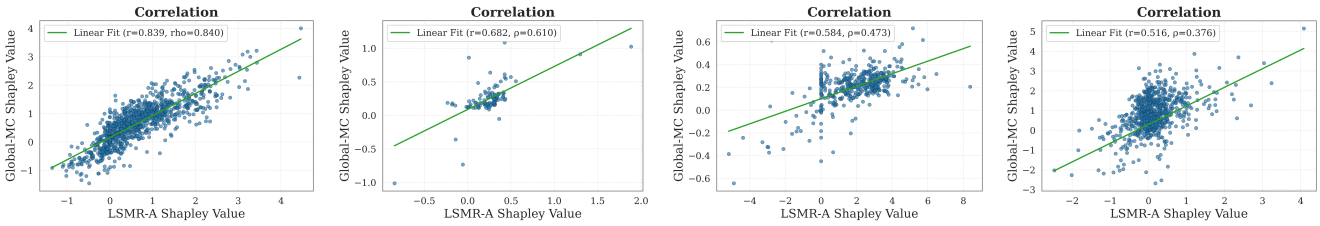
where $\phi_i^{(m)}$ denotes the estimate for training point $i$ after $m$ samples and $\epsilon = 10^{-12}$. $\tau = 0.05$ controls convergence and serves as the stopping threshold for all models. Experiments were run on an Intel Xeon E5-2640 v4 CPU with 32 GB RAM. For GNN workloads, we used a single machine with 64 CPU cores and 8 NVIDIA H20 GPUs, running 64 jobs in parallel. Experimental details are provided in Supplemental Material B.

## 6.2 RQ1: Approximation Fidelity

We first evaluate whether Local Shapley faithfully approximates the global Shapley value across model families. For each model–dataset pair, we compute global Shapley values using **Global-MC** and Local Shapley values using **LSMR-A** both run to convergence (Eq. (12)). Agreement is quantified using Pearson's $r$ for linear consistency and Spearman's $\rho$ for rank-order consistency. Across all four models, Local Shapley exhibits strong positive correlation with Global Shapley, with Pearson $r$ ranging from 0.516 to 0.839, indicating substantial agreement between local and global valuation.

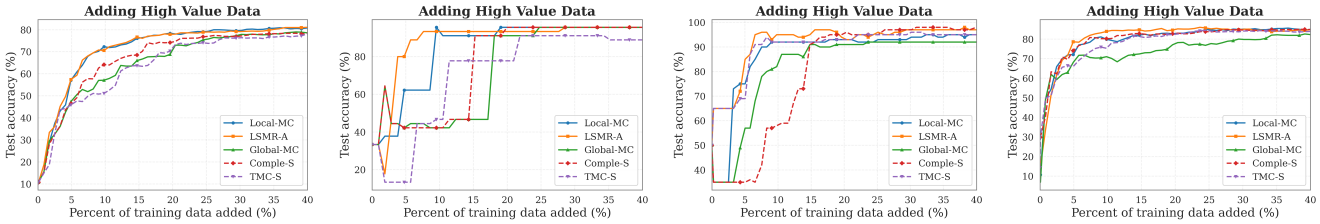
The strongest alignment occurs for **WKNN** (Figure 2(a)), where predictions are fully determined by the $K$ nearest neighbors and locality is highly concentrated. The strong correlation confirms that the support set captures the dominant influence pathway, consistent with the tight error bound in Proposition 1 when non-local interaction mass is negligible. For **Decision Tree** (Figure 2(b)) and **RBF-SVM** (Figure 2(c)), correlations remain consistently positive, reflecting that rule-based partitioning and kernel-induced decay capture the primary influence pathways. The slight reduction in correlation relative to WKNN aligns with approximate locality, as influence outside the support is nonzero but controlled.

For **GNN** (Figure 2(d)), correlation is weaker but still clearly positive. This is expected: GNNs exhibit approximate locality, as nodes outside the $L$-hop receptive field may still affect learned parameters through shared gradients during training. Additionally, deep learning training introduces stochasticity via random initialization, mini-batch sampling, and non-convex optimization, which




(a) WKNN, MNIST (b) DT, Iris (c) RBF-SVM, Breast Cancer (d) GNN, Cora

**Figure 2: Scatter plots of Local Shapley (x-axis) versus Global Shapley (y-axis). Green lines indicate linear regression fits.**




(a) WKNN, MNIST (b) DT, Iris (c) RBF-SVM, Breast Cancer (d) GNN, Cora

**Figure 3: Test accuracy versus the percentage of training data added in descending Shapley order.**

inflates variance in both the global reference and the local estimates. This empirical trend is consistent with the approximation bound in Proposition 1: when non-local interaction mass is small but nonzero, correlation degrades gradually rather than collapsing. We additionally compare against In-Run Data Shapley [61] on GNN, the only applicable setting due to its reliance on gradient-based training (Supplemental Material B.3). While [61] completes valuation in a single training run, it yields lower approximation fidelity and data-selection accuracy. Conversely, LSMR-A preserves exact Shapley semantics while maintaining bounded error.

## 6.3 RQ2: Downstream Data Selection Utility

High correlation with global Shapley does not automatically imply downstream effectiveness. Since data valuation is commonly used for training set pruning and prioritization [17, 24], we evaluate whether Local Shapley preserves *selection utility*. Specifically, we rank training points in descending order of their Shapley scores. Starting from an empty training set, we incrementally add top-ranked samples and retrain the model at each step. Test accuracy is plotted against the fraction of training data incorporated. A stronger valuation method should achieve high accuracy with fewer samples, producing a steeper and dominating selection curve. Across all four model pairs, LSMR-A consistently matches or exceeds the data selection performance of global estimators.

For **WKNN** (Figure 3(a)), where locality is strong, LSMR-A achieves the most improvement: approximately 10% of locally selected data attains accuracy comparable to 20% selected by Global-MC. This behavior is consistent with RQ1, where correlation is strongest under exact locality. For **Decision Tree** (Figure 3(b)), LSMR-A achieves the highest accuracy across most operating points, indicating that leaf-induced support sets preserve the ranking of influential samples even in small datasets. For **RBF-SVM** (Figure 3(c)), LSMR-A and Local-MC perform comparably to Comple-S and outperform Global-MC and TMC-S in the low-data regime ($< 10\%$). Although RQ1 shows only moderate linear correlation in this setting, the identification of the top-ranked samples remains robust—highlighting that data selection depends primarily on accurately identifying high-impact samples rather than perfectly matching global magnitudes. For **GNN** (Figure 3(d)), LSMR-A yields the best selection performance. Global-MC performs worst, likely due to high retraining variance across coalitions in deep learning settings. In contrast, restricting valuation to 2-hop structural supports stabilizes ranking and better captures dominant influence pathways.

Overall, model-induced locality enables efficient analysis of data selection utility. The support-restricted approach identifies influential samples while substantially reducing retraining cost. As established in Theorem 2 and 3, LSMR-A provides unbiased estimation with exponential concentration guarantees under permutation sampling. Moreover, Theorem 5 shows that structured reuse further reduces estimator variance by amortizing redundant randomness across overlapping supports. Together, these results demonstrate that model-induced locality improves computational efficiency while preserving statistical stability.

## 6.4 RQ3: Optimal Reuse for Efficiency

In this section, we assess computational efficiency by measuring runtime and the number of model trainings required to satisfy the convergence criterion (Eq. (12)). Shown in Table 1, LSMR-A achieves the fastest convergence across all model families. The retraining

**Table 1: Total running time and number of model trainings.**

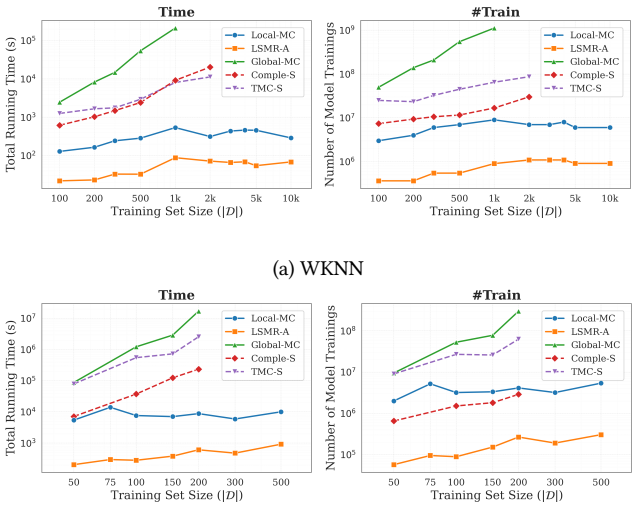
| Model | Metric | Global | TMC-S | Comple-S | Local-MC | LSMR-A |
|---|---|---|---|---|---|---|
| WKNN | Time | 212,408 | 8,085 | 9,181 | 537 | **88** |
| | # Train | 1,126M | 65.1M | 16.8M | 9.0M | **0.9M** |
| DT | Time | 1,446 | 250 | 206 | 243 | **111** |
| | # Train | 2.8M | 0.5M | 0.4M | 0.5M | **0.2M** |
| RBF-SVM | Time | 101,903 | 2,790 | 3,087 | 16,013 | **251** |
| | # Train | 81.7M | 4.9M | 5.1M | 32M | **0.4M** |
| GNN | Time | 200,860 | 122,264 | 71,768 | 45,378 | **7,241** |
| | # Train | 28.2M | 23.8M | 9.1M | 12.0M | **1.7M** |

counts show a consistent trend: on WKNN, LSMR-A reduces required trainings by more than three orders of magnitude compared to Global-MC, while on other models it delivers over 10× speedups. The efficiency gains of LSMR-A arise from three mechanisms that eliminate redundant computation in Shapley evaluation: (i) coalition space reduction by restricting evaluation to $\mathcal{N}(t)$ (Lemma 1); (ii) subset-centric reformulation to enable reuse within each support set (Lemma 2); and (iii) pivot scheduling to enable reuse across support sets (Theorem 1), achieving theoretically optimal reuse.

When supports are tight, as in **WKNN** where $|\mathcal{N}(t)| \ll |D|$, coalition space reduction alone already yields substantial gains: Local-MC achieves a significant speedup over Global-MC simply by restricting computation to the support set. LSMR-A further eliminates redundant evaluations during computation, reducing model trainings from 9.0M to 0.9M. In contrast, for **RBF-SVM**, applying a kernel threshold of 0.5 yields support sets that retain roughly 30% of the training data, reducing the effectiveness of coalition space reduction through model-induced locality. Compared to Local-MC, which also restricts computation to the support set, LSMR-A substantially reduces training cost and converges more rapidly. The performance gap between the two highlights the importance of reuse: as support sets grow, limiting the coalition space alone becomes insufficient, and efficiency is primarily driven by sharing within and across support sets. Without carefully eliminating redundant subset evaluations through structured reuse, Local-MC even underperforms TMC-S and Comple-S in this setting. Detailed analyses of the efficiency contributions of each mechanism in LSMR-A are provided in the Supplemental Material B.4. For **Decision Tree** and **GNN**, the speedup over Global-MC is relatively modest. Nevertheless, LSMR-A still achieves the lowest retraining count, indicating that the reuse mechanisms remain effective even under these less favorable structural conditions.

To further examine the scalability of LSMR-A, Figure 4 reports the runtime and number of model trainings on the MNIST dataset as the training set size $|\mathcal{D}|$ increases, for two model families: WKNN ($|\mathcal{D}|$ = 100 to 10,000, with test set size $|\mathcal{T}|$ = 1,000) and Decision Tree ($|\mathcal{D}|$ = 50 to 500, $|\mathcal{T}|$ = 100). Note that the results of certain baselines are not reported once their runtime becomes prohibitively expensive, i.e., when they require more than 24 hours to converge.

On **WKNN** (Figure 4(a)), all global baselines exhibit steep growth in both runtime and training count over their observable ranges. Local-MC flattens earlier due to coalition space reduction but still requires substantial model trainings. In contrast, LSMR-A remains nearly flat across the entire range. At $|\mathcal{D}|$ = 10,000, LSMR-A is more




(a) WKNN

(b) DT

**Figure 4: Scalability with respect to the training set size $|\mathcal{D}|$ on MNIST, plotted on logarithmic scales.**

**Table 2: Sensitivity to the support set size $\mathcal{N}(t)$.**

| Method | $\lvert\mathcal{N}(t)\rvert$ | $r$ | $\rho$ | Acc@10% | Acc@20% | Time | # Train |
|---|---|---|---|---|---|---|---|
| Global | — | — | — | 60.0 | 68.9 | 212k | 1,126M |
| LSMR-A | 1 | 0.443 | 0.460 | 68.3 | 74.7 | 20.2 | 0.4M |
| LSMR-A | 3 | 0.619 | 0.613 | 71.7 | 76.5 | 55.7 | 0.8M |
| LSMR-A | 5 | 0.721 | 0.717 | 70.5 | 76.9 | 65.1 | 0.8M |
| LSMR-A | 10 | 0.839 | 0.840 | 70.4 | 78.6 | 87.6 | 0.9M |
| LSMR-A | 20 | 0.901 | 0.898 | 70.5 | 77.2 | 89.2 | 0.9M |

than five orders of magnitude faster than Global-MC, converging in under two minutes where global methods are infeasible to run. On **Decision Tree** (Figure 4(b)), LSMR-A stays nearly constant on both metrics across the entire range, whereas Global-MC increases by more than two orders of magnitude as $|\mathcal{D}|$ grows from 50 to 200. Local-MC also levels off, but at a scale about one order of magnitude higher than LSMR-A, further demonstrating that the subset reuse mechanism consistently yields additional computational savings.

This scaling trend is consistent with Theorems 3 and 4: when the support size is fixed, the number of distinct subsets is bounded independently of $|\mathcal{D}|$, so the amortized retraining cost per test point decreases as the dataset grows, with concentration guarantees preserving estimation stability under reuse. The widening gap between LSMR-A and global baselines across both model families further demonstrates that the framework remains effective in regimes where global methods are computationally infeasible.

## 6.5 RQ4: Sensitivity to Support Set Size

Table 2 examines how the support size $|\mathcal{N}(t)|$ mediates the trade-off between approximation fidelity and retraining time cost. We use WKNN on MNIST and vary the size of $|\mathcal{N}(t)|$. All methods

are evaluated at the common convergence point (Eq. (12)). To assess downstream utility, we introduce Acc@$x$%, denoting the test accuracy achieved when training on the top-$x$% of selected data.

As $|\mathcal{N}(t)|$ increases, approximation fidelity improves, with Pearson correlation rising by approximately 103% from the smallest to the largest support. Most gains are realized before $|\mathcal{N}(t)| = 10$, after which improvements plateau, consistent with Proposition 1: the primary contributors enter the support early, and further expansion mainly reduces residual non-local effects with diminishing returns. Across all settings, LSMR-A remains more than three orders of magnitude faster than Global-MC. Although the coalition space scales as $2^{|\mathcal{N}(t)|}$, the number of model trainings and wall-clock time grow sublinearly, as both intra-support and inter-support reuse amortize computation across training and test points respectively.

Even with $|\mathcal{N}(t)| = 3$, LSMR-A outperforms Global-MC in data selection by concentrating computation on outcome-relevant coalitions and producing more stable rankings. As $|\mathcal{N}(t)|$ grows further, correlation with the global estimator continues to improve, but selection accuracy remains stable. This is expected: including more training points in the support set reduces the non-local interaction mass outside $\mathcal{N}(t)$, tightening the approximation bound in Proposition 1 and bringing the local game closer to the global game in magnitude. Data selection accuracy does not benefit that much from this continued improvement. This suggests that even a relatively small support set already captures the dominant influence pathway that determines the prediction at each test point, and thus suffices to identify the most valuable training instances. Enlarging the support beyond this point refines numerical agreement with the global estimator but does not change which training points are ranked highest, as the additional nodes contribute only marginally to the prediction and receive correspondingly small Shapley values.

## 6.6 RQ5: Model-Induced Locality

Table 3 examines whether locality must align with the utility evaluation model. While earlier experiments construct $\mathcal{N}(t)$ and compute utility under the same model architecture, we introduce cross-model settings to isolate the effect of architectural alignment. We build $\mathcal{N}(t)$ using one model's computational pathway (e.g., $K$-nearest neighbors for WKNN or same-leaf membership for Decision Tree) but evaluate utility under a different model, keeping the convergence criterion and average support size fixed. We also include a **Random** baseline that samples $|\mathcal{N}(t)|$ training points uniformly.

For WKNN, model-aligned supports deliver the highest correlation and strongest selection performance. Substituting Decision Tree pathways leads to a moderate decrease, indicating partial consistency between the two locality definitions, since feature-based splits can still cluster nearby samples. The reduction is greater with RBF-SVM supports, where a stringent kernel threshold produces sparse neighborhoods that misses WKNN's locality structure. Even so, both cross-model supports markedly surpass the random baseline, with Pearson correlations more than five times higher. This suggests LSMR-A remains robust to architectural mismatch as long as the support set preserves a partial model prediction pathway.

For GNN, misalignment has a substantially stronger impact. Replacing graph-structural supports with WKNN or Decision Tree neighborhoods nearly eliminates correlation ($r$ and $\rho \approx 0.1$), even worse than random performance. This stems from a severe structural mismatch: GNN predictions depend on message passing over graph topology, while WKNN and Decision Tree supports are defined in feature space. Nodes that are close in feature space may be far apart in the graph and thus have minimal influence on GNN predictions, and vice versa. Consequently, the misaligned support set violates the assumption that $\mathcal{N}(t)$ contains the most relevant training points for $v_t$. In this case, the approximation bound in Proposition 1 no longer provides meaningful guarantees, as interactions outside the intended local region dominate the valuation.

**Table 3: Effect of alignment between model architecture and support set construction**

| Model | Support Set | $r$ | $\rho$ | Acc@10% | Acc@20% |
|---|---|---|---|---|---|
| WKNN | WKNN | 0.839 | 0.840 | 70.4 | 78.6 |
| WKNN | DT | 0.713 | 0.722 | 66.7 | 74.8 |
| WKNN | RBF-SVM | 0.596 | 0.575 | 68.0 | 75.3 |
| WKNN | Random | 0.118 | 0.120 | 64.1 | 68.0 |
| Model | Support Set | $r$ | $\rho$ | Acc@5% | Acc@10% |
| GNN | GNN | 0.516 | 0.376 | 78.6 | 83.9 |
| GNN | DT | 0.136 | 0.139 | 73.8 | 76.5 |
| GNN | WKNN | 0.112 | 0.113 | 75.1 | 79.3 |
| GNN | Random | 0.164 | 0.188 | 74.6 | 79.6 |

These results show that locality is model-dependent. Aligning support set construction with the evaluation model's computational pathway is essential for preserving valuation fidelity, and the penalty for misalignment grows with structural differences in their definitions of locality as shown in Remark 1.

## 7 CONCLUSION

This paper reframes Shapley-based data valuation through the lens of model-induced locality and intrinsic subset complexity. We show that, for modern predictors, the true computational bottleneck is not the exponential global coalition space, but the number of distinct subsets that can influence at least one valuation. This perspective leads to an information-theoretic lower bound on retraining complexity and motivates LSMR, an optimal subset-centric algorithm that evaluates each influential subset exactly once via structured support mapping and reuse. For larger support sets, LSMR-A extends this principle to Monte Carlo estimation, decoupling sampling complexity from retraining complexity while preserving unbiasedness, exponential concentration, and lower variance through structural reuse. These results demonstrate that Shapley computation can be treated as a structured data management problem—where exploiting locality and eliminating redundancy yield both theoretical optimality and practical scalability, opening new opportunities for efficient and principled data valuation at scale.

## ACKNOWLEDGMENTS

Xuan Yang and Jian Pei's research is supported in part by the NSF Project MSPA-2434666. Ming-Syan Chen was supported by the Ministry of Science and Technology (MOST), Taiwan, under Grant 115-2223-E-002-001. All opinions, Findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

# REFERENCES


[1] S Basu, P Pope, and S Feizi. 2021. Influence Functions in Deep Learning Are Fragile. In *International Conference on Learning Representations (ICLR)*.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[3] Olivier Bousquet and André Elisseeff. 2002. Stability and generalization. *Journal of machine learning research* 2, Mar (2002), 499–526.

[4] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. 1984. *Classification and Regression Trees*. Wadsworth International Group.

[5] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research* 36, 5 (2009), 1726–1730.

[6] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2011. *Computational aspects of cooperative game theory*. Morgan & Claypool Publishers.

[7] Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. 2023. Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence* 5, 6 (2023), 590–601.

[8] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards model-based pricing for machine learning in a data marketplace. In *Proceedings of the 2019 international conference on management of data*. 1535–1552.

[9] Yi-Chung Chen, Hsi-Wen Chen, Shun-Gui Wang, and Ming-Syan Chen. 2023. Space: Single-round participant amalgamation for contribution evaluation in federated learning. *Advances in Neural Information Processing Systems* 36 (2023), 6422–6441.

[10] Yan-Cheng Chen and Chao-Ton Su. 2016. Distance-based margin support vector machine for classification. *Appl. Math. Comput.* 283 (2016), 141–152.

[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[12] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13, 1 (1967), 21–27.

[13] Xiaotie Deng and Christos H Papadimitriou. 1994. On the complexity of cooperative solution concepts. *Mathematics of operations research* 19, 2 (1994), 257–266.

[14] S.A. Dudani. 1976. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6, 4 (1976), 325–327.

[15] Raul Castro Fernandez, Pranav Subramaniam, and Michael J. Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proceedings of the VLDB Endowment* 13, 11 (2020), 1933–1947. https://doi.org/10.14778/3407790.3407800

[16] Ronald A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 2 (1936), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[17] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.

[18] Justin Gilmer, Samuel Schoenholz, Patrick Riley, Oriol Vinyals, and George Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*.

[19] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[20] Zayd Hammoudeh and Daniel Lowd. 2024. Training data influence analysis and estimation: A survey. *Machine Learning* 113, 5 (2024), 2351–2403.

[21] Moritz Hardt, Ben Recht, and Yoram Singer. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*. PMLR, 1225–1234.

[22] Tony Hey, Stewart Tansley, Kristin Michele Tolle, et al. 2009. *The fourth paradigm: data-intensive scientific discovery*. Vol. 1. Microsoft research Redmond, WA.

[23] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, and Dawn Song. 2019. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment* 12, 11 (2019), 1610–1623.

[24] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1167–1176.

[25] Kevin Jiang, Weixin Liang, James Y Zou, and Yongchan Kwon. 2023. Opendataval: a unified benchmark for data valuation. *Advances in Neural Information Processing Systems* 36 (2023), 28624–28647.

[26] Bojan Karlaš, David Dao, Matteo Interlandi, Sebastian Schelter, Wentao Wu, and Ce Zhang. 2023. Data debugging with shapley importance over machine learning pipelines. In *The Twelfth International Conference on Learning Representations*.

[27] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[28] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. PMLR, 1885–1894.

[29] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. 2009. Computational Social Science. *Science* 323, 5915 (2009), 721–723. https://doi.org/10.1126/science.1167742

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[31] Mengyang Li, Weiyao Zhu, and Ou Wu. 2025. Toward learnable and interpretable data Shapley valuation for deep learning. *Knowledge-Based Systems* 325 (2025), 114002.

[32] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[33] Fan Liang, Wei Yu, Dou An, Qingyu Yang, Xinwen Fu, and Wei Zhao. 2018. A survey on big data market: Pricing, trading and protection. *Ieee Access* 6 (2018), 15132–15154.

[34] Hong Lin, Shixin Wan, Zhongle Xie, Ke Chen, Meihui Zhang, Lidan Shou, and Gang Chen. 2025. A Comprehensive Study of Shapley Value in Data Analytics. *Proceedings of the VLDB Endowment* 18, 9 (2025), 3077–3092.

[35] Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. 2022. Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*. PMLR, 13468–13504.

[36] Jinfei Liu, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Dealer: An end-to-end model marketplace with differential privacy. *Proceedings of the VLDB Endowment* 14, 6 (2021).

[37] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* 2, 1 (2020), 56–67.

[38] Xuan Luo, Jian Pei, Zicun Cong, and Cheng Xu. 2022. On shapley value in data assemblage under independent utility. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2761–2773.

[39] Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast shapley value computation in data assemblage tasks as cooperative simple games. *Proceedings of the ACM on Management of Data* 2, 1 (2024), 1–28.

[40] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. 2008. Supervised dictionary learning. *Advances in neural information processing systems* 21 (2008).

[41] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the estimation error of sampling-based Shapley value approximation. *arXiv preprint arXiv:1306.4265* (2013).

[42] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. In *Information Retrieval*, Vol. 3. Springer, 127–163.

[43] Tomasz P Michalak, Karthik V Aadithya, Piotr L Szczepanski, Balaraman Ravindran, and Nicholas R Jennings. 2013. Efficient computation of the Shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research* 46 (2013), 607–650.

[44] Elizbar A Nadaraya. 1964. On estimating regression. *Theory of Probability & Its Applications* 9, 1 (1964), 141–142.

[45] Kenta Oono and Taiji Suzuki. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations (ICLR)*.

[46] Junyuan Pang, Jian Pei, Haocheng Xia, Xiang Li, and Jinfei Liu. 2025. Shapley value estimation based on differential matrix. *Proceedings of the ACM on Management of Data* 3, 1 (2025), 1–28.

[47] Vardan Papyan, XY Han, and David L Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences* 117, 40 (2020), 24652–24663.

[48] Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics* 33, 3 (1962), 1065–1076.

[49] Jian Pei. 2020. A survey on data pricing: from economics to data science. *IEEE Transactions on knowledge and Data Engineering* 34, 10 (2020), 4586–4608.

[50] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems* 33 (2020), 19920–19930.

[51] J Ross Quinlan. 2014. *C4. 5: programs for machine learning*. Elsevier.

[52] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You? Explaining the Predictions of Any Classifier. In *KDD*.

[53] Lloyd S Shapley. 1953. A value for n-Person Games: contributions to the Theory of Games (AM 28), Volume II. , 307–317 pages.

[54] Michelle Si and Jian Pei. 2024. Counterfactual Explanation of Shapley Value in Data Coalitions. *Proceedings of the VLDB Endowment* 17, 11 (2024), 3332–3345.

[55] W. N. Street, W. H. Wolberg, and O. L. Mangasarian. 1993. Nuclear feature extraction for breast tumor diagnosis. *IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology* 1905 (1993), 861–870.

[56] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*. 843–852.

[57] Haifeng Sun, Yu Xiong, Runze Wu, Xinyu Cai, Changjie Fan, Lan Zhang, and Xiang-Yang Li. 2026. Fast-DataShapley: Neural Modeling for Training Data Valuation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*. 607–617.

[58] Qiheng Sun, Jiayao Zhang, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2024. Shapley value approximation based on complementary contribution. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[59] Jiachen T Wang and Ruoxi Jia. 2023. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 6388–6421.

[60] Jiachen T Wang, Prateek Mittal, and Ruoxi Jia. 2024. Efficient data shapley for weighted nearest neighbor algorithms. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2557–2565.

[61] Jiachen Tianhao Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2025. Data shapley in one training run. In *International conference on learning representations*, Vol. 2025. 12358–12395.

[62] Jiachen T Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. 2024. Rethinking data shapley for data selection tasks: misleads and merits. In *Proceedings of the 41st International Conference on Machine Learning*. 52033–52063.

[63] Jiachen T. Wang, Yuqing Zhu, Yu-Xiang Wang, Ruoxi Jia, and Prateek Mittal. 2023. Threshold KNN-Shapley: A Linear-Time and Privacy-Friendly Approach to Data Valuation. arXiv:2308.15709 [cs.LG] https://arxiv.org/abs/2308.15709

[64] Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International conference on machine learning*. PMLR, 1954–1963.

[65] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3733–3742.

[66] Jinghan Yang, Sarthak Jain, and Byron C Wallace. 2023. How Many and Which Training Points Would Need to be Removed to Flip this Prediction?. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2571–2584.

[67] Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data valuation using reinforcement learning. In *International Conference on Machine Learning*. PMLR, 10842–10851.

[68] Guangyi Zhang, Qiyu Liu, and Aristides Gionis. 2025. Shapley-Based Data Valuation for Weighted $k$-Nearest Neighbors. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[69] Yansen Zhang, Xiaokun Zhang, Ziqiang Cui, and Chen Ma. 2025. Shapley value-driven data pruning for recommender systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 3879–3888.