Towards Generalisable Data Valuation: Learning an End-to-End Deep Model to Estimate Shapley Values

Zhiyu Chen **Jicong Fan***
School of Data Science
The Chinese University of Hong Kong, Shenzhen, China
zhiyuchen2@link.cuhk.edu.cn fanjicong@cuhk.edu.cn

Abstract

Data Shapley has emerged as a principled framework for data valuation, yet its practical utility is severely hampered by exponential computational complexity and the need for costly model retraining. Unlike existing methods that compute values independently for each dataset, we propose GenSHAP, a novel graph-based paradigm that reformulates data valuation as a transferable learning task. GenSHAP learns an amortised deep model capable of predicting Data Shapley values across previously unseen classification datasets through a single forward pass after preprocessing. Our framework utilises a relation-based dataset representation method and a hybrid GIN-Transformer architecture to capture local and global inter-sample dependencies. We further introduce GenSHAP-S, which augments these structural representations with low-budget marginal contribution probes to achieve high-fidelity valuation. Extensive experiments demonstrate that GenSHAP closely approximates high-budget permutation-Shapley reference estimates and excels in downstream tasks such as data removal and addition while reducing valuation time from hours to seconds. The source code is available in the GitHub repository: https://github.com/Zhiyu8765/GenSHAP.

1 Introduction

Data has emerged as a fundamental driving force behind the rapid advancement of modern artificial intelligence systems. Unlike traditional assets, data is inherently heterogeneous. Individual points can contribute unequally to model performance. Quantifying the value of data has therefore become a critical problem in data-centric AI. In particular, data valuation methods aim to measure the influence of individual training samples on model performance, enabling a wide range of applications such as data selection, noisy label detection, and fair compensation for data contributors [1].

The field of data valuation is expanding rapidly, with annual publications growing by over 50% [2]. Research is driven by two distinct paradigms: Information Economics, which focuses on qualitative, market-oriented mechanisms like pricing strategies, incentive design, and ownership fairness [3–5]; and Machine Learning, which emphasises quantitative methodologies. The latter leverages cooperative game theory and statistical learning to algorithmically measure the marginal contribution of individual data points to model performance [6–11].

Data Shapley is a quantitative valuation framework favoured for its equitable distribution and game-theoretic axioms [12]. However, exact computation is often intractable, requiring subset enumerations with exponential costs. To address this, research has bifurcated into two paradigms: structure-aware methods (e.g., KNN Shapley [13], Group-Testing Shapley [7]), which utilize analytical shortcuts to bypass sampling, and sampling-based approximations (e.g., TMC Data Shapley and G-Shapley [6], Delta-Shapley [14]), which estimate marginal contributions via heuristic coalition sampling.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Unlike the aforementioned methods, which compute Data Shapley values separately for each dataset, our method aims to learn a reusable data valuation model. A related approach is MLPbV [15], which trains a Multi-Layer Perceptron (MLP) to regress data Shapley values from engineered training dynamics and statistical features. However, its design primarily relies on tabular feature representations and does not explicitly model the relational structure among data points, which limits its ability to capture high-order interactions. Recent work such as SGUL [16] considers structural dependencies in GNN inference by learning a linear utility function from hand-crafted, domain-specific neighbourhood features. However, SGUL relies on linear optimisation rather than deep relational modelling, and its application is rigidly confined to GNN inference tasks, leaving the challenge of zero-shot, cross-domain training data valuation unresolved.

In contrast, we propose a graph-based paradigm that encodes local and global inter-sample relationships. To handle architectural heterogeneity, we adopt Relation-based Dataset Representation (RDR) to standardise raw features into structural relationships. We process these representations through a hybrid architecture of multi-branch Graph Isomorphism Networks (GIN) [17] and Transformers [18] to capture higher-order interactions. This shift from independent feature mapping to relational modelling facilitates transfer across heterogeneous datasets through inter-sample relationships.

To our knowledge, this is the first work to investigate generalisable Data Shapley estimation across heterogeneous datasets via structured relational modelling. Our contributions are:

- **Transferable learning paradigm (GenSHAP)**: We reformulate Data Shapley estimation as an amortised learning task. By introducing the RDR method, our model maps heterogeneous datasets into a unified space, enabling inference across unseen domains and varying feature dimensions.
- **Structural & stochastic fusion (GenSHAP-S)**: We propose a hybrid GIN-Transformer architecture to capture inter-sample dependencies. The enhanced GenSHAP-S variant integrates these structural embeddings with low-budget marginal contribution probes, providing high-fidelity valuation.
- **Superior efficiency & fidelity**: By replacing iterative retraining with a single neural forward pass, our framework reduces valuation latency from hours to seconds. Extensive results demonstrate that GenSHAP aligns closely with Permutation Data Shapley estimates and excels in downstream tasks.

2 Preliminary

Data Shapley The Shapley value, originally introduced in cooperative game theory [19], has been extended to machine learning through Data Shapley, where training samples are treated as players [6, 7]. Given a dataset $\mathcal{D} = \{\mathbf{z}_i\}_{i=1}^N$, let $V(S)$ denote the utility, e.g., test accuracy or loss, of a model or algorithm $\mathcal{A}$ trained on subset $S \subseteq \mathcal{D}$. The Data Shapley value of sample $\mathbf{z}_i$ is defined as

$$\phi_i = C \sum\nolimits_{S \subseteq \mathcal{D} \setminus \{\mathbf{z}_i\}} \binom{N-1}{|S|}^{-1} \Big(V(S \cup \{\mathbf{z}_i\}) - V(S) \Big), \tag{1}$$

where $C = \frac{1}{N}$ is the normalisation constant.. It measures the expected marginal contribution of each sample over all possible training subsets, providing a principled framework for data valuation. However, exact computation requires evaluating 2^N subsets and typically retraining a model for each utility evaluation, making it infeasible for realistic datasets and motivating efficient approximations.

Permutation Data Shapley To address the computational burden of exact Data Shapley, permutation-based sampling provides an efficient Monte Carlo approximation of the expectation form. For a random permutation π, define the prefix set as $P_i^\pi = \{\mathbf{z}_j \in \mathcal{D} : \pi(j) < \pi(i)\}$ and the marginal contribution as $\Delta_i^\pi = V(P_i^\pi \cup \{z_i\}) - V(P_i^\pi)$. Then an unbiased estimator is

$$\hat{\phi}_i = \frac{1}{m} \sum_{t=1}^{m} \Delta_i^{\pi_t}, \quad \text{where } \mathbb{E}\left[\hat{\phi}_i\right] = \phi_i. \tag{2}$$

This procedure can be interpreted as averaging the incremental contribution of a data point across random data acquisition orders [20]. Although statistically consistent, the method remains computationally expensive, as it requires repeated evaluations of the model on growing subsets, leading to a complexity of $O(m \cdot N \cdot T_{\text{train}})$, where m is the number of sampled permutations and T_{train} is the average training cost of $\mathcal{A}$.

In practice, we use a finite-budget permutation estimator as a reference valuation signal. Although the permutation estimator is unbiased for the exact Data Shapley value under deterministic utility

evaluation, finite sampling introduces Monte Carlo variance. We therefore refer to these labels as high-budget permutation reference estimates rather than exact ground truth.

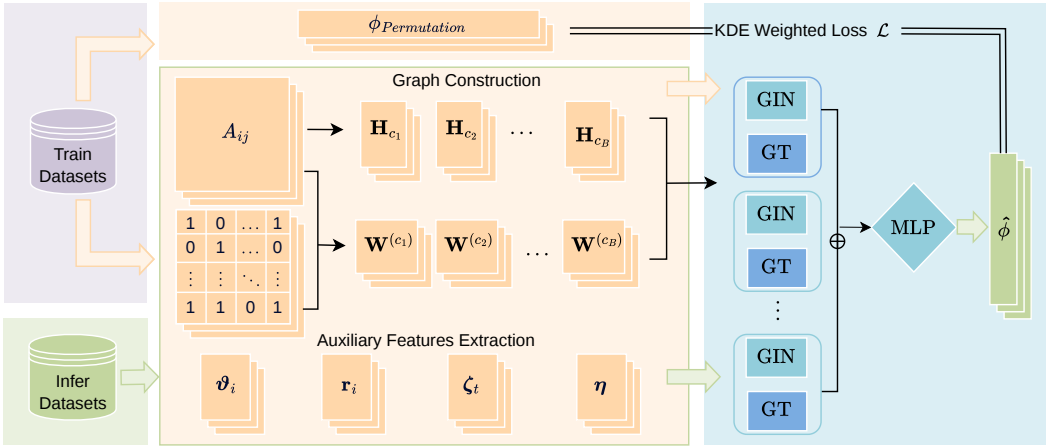


Figure 1: GenSHAP architecture. The trained model enables fast amortised evaluation on unseen datasets: relational and auxiliary representations are constructed once, and the learned predictor outputs values with a single neural forward pass, avoiding repeated subset evaluation and retraining.

3 Methodology

3.1 Motivation

Let $\mathcal{D} = \{\mathbf{z}_i\}_{i=1}^N$ denote a dataset of N samples, where each datum $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ consists of a feature vector $\mathbf{x}_i \in \mathbb{R}^d$ and an associated label y_i. Let $\mathcal{A}$ be a learning algorithm and $V(S)$ denote a utility function (e.g., accuracy) defined on any subset $S \subseteq \mathcal{D}$. The goal of data valuation is to assign a scalar importance score $\phi_i \in \mathbb{R}$ to each $\mathbf{z}_i$, quantifying its marginal contribution to the model trained by $\mathcal{A}$. A principled formulation is provided by Data Shapley [6], which is computed using (1). Despite its theoretical appeal, the exact computation of $\{\phi_i\}_{i=1}^N$ requires evaluating all possible subsets and thus incurs exponential complexity $\mathcal{O}(2^N)$, making it intractable even for moderately sized datasets. Although we may use Permutation Data Shapley, i.e. (2), the high complexity $\mathcal{O}(m \cdot N \cdot T_{\text{train}})$ limits its application to real scenarios.

To address the aforementioned limitation, we propose to use a number of historical datasets, denoted as $\mathcal{H} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$, to train a universal predictive model f_Θ for a given learning algorithm $\mathcal{A}$, such as logistic regression or support vector machines [21]. Once trained, f_Θ can directly estimate the data Shapley values of samples in a new dataset from any domain under $\mathcal{A}$, without requiring retraining. In other words, **for previously unseen classification datasets $\mathcal{D}$, regardless of its application domain, sample size, or feature dimension, f_Θ produces Shapley value estimates through preprocessing followed by a single neural forward pass.**

Rationale for using historical datasets The use of historical datasets is well justified because they constitute a large, reusable, and already labelled source of supervision. Since collecting and annotating data are often costly, ignoring existing open-source datasets would waste substantial human effort and domain knowledge. By aggregating many historical datasets, a large model can learn from diverse distributions, tasks, and feature spaces, thereby improving its potential generalisation ability to new datasets. Thus, historical datasets are not merely auxiliary resources; they are the natural training foundation for building universal or cross-domain machine learning models.

3.2 GenSHAP

It is non-trivial to construct f_Θ due to the following challenges:

- **Challenge I**. The datasets we want to evaluate often come from diverse domains, implying that the feature dimension and semantic meaning of two datasets are entirely different and incomparable.
- **Challenge II**. Shapley values are based on direct interactions between samples, which is difficult to explicitly model using standard machine learning techniques.

In the following, we show how our GenSHAP addresses these challenges to achieve an amortised model for data Shapley value prediction.

3.2.1 Relation-based dataset representation

To address Challenge I, we introduce a relation-based dataset representation (RDR) method. The core concept is to represent each dataset not by its raw feature coordinates, but by its internal sample-to-sample relational structure. This maps datasets with disparate input dimensions and feature meanings into a unified representation space.

Consider a collection of historical datasets $\mathcal{H} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$, where each dataset $\mathcal{D}_m = \{\mathbf{z}_i^{(m)}\}_{i=1}^{N_m}$ consists of samples $\mathbf{z}_i^{(m)} = (\mathbf{x}_i^{(m)}, y_i^{(m)})$, with features $\mathbf{x}_i^{(m)} \in \mathbb{R}^{d_m}$. Because both the sample size N_m and the feature dimension d_m vary across datasets, directly learning a universal predictor f_Θ from raw feature vectors is impractical. Specifically, $\mathbf{x}_i^{(m)}$ and $\mathbf{x}_i^{(m')}$ lie in different spaces, and their coordinates may carry entirely different semantic meanings even when $d_m = d_{m'}$.

For each dataset $\mathcal{D}_m$, we first compute a pairwise distance matrix by

$$D_{ij}^{(m)} = \|\mathbf{x}_i^{(m)} - \mathbf{x}_j^{(m)}\|_2, \quad 1 \le i, j \le N_m. \tag{3}$$

Let σ_m denote the average pairwise distance within $\mathcal{D}_m$. For each scale factor $c \in \mathcal{C}$, e.g., $\mathcal{C} = \{0.25, 0.50, 1.00, 2.00, 4.00\}$, we define a Gaussian-kernel adjacency matrix by

$$A_{ij}^{(m,c)} = \exp\Big(-\tfrac{(D_{ij}^{(m)})^2}{2(c\sigma_m)^2}\Big). \tag{4}$$

Remark 3.1. This adjacency matrix $\mathbf{A}^{(m,c)}$ captures the relative geometric relationships between samples within the same dataset $\mathcal{D}_m$, at scale $c \in \mathcal{C}$. Because it relies entirely on pairwise relations rather than the original coordinate system, this matrix is invariant to rotation and scaling. Consequently, it yields representations that are inherently comparable across heterogeneous domains.

To extract fixed-dimensional data point representations from datasets with varying feature dimensions, we apply Singular Value Decomposition (SVD) to each scale-specific relational matrix $\mathbf{A}^{(m,c)}$ instead of the raw feature matrix. Specifically, for each relational matrix $\mathbf{A}^{(m,c)}$, we perform SVD:

$$\mathbf{A}^{(m,c)} = \mathbf{U}^{(m,c)}\mathbf{\Sigma}^{(m,c)}(\mathbf{V}^{(m,c)})^\top, \tag{5}$$

and extract the top d_s singular components to construct an initial embedding matrix as:

$$\mathbf{H}_{m,c}^{(0)} = \mathbf{U}_{:,1:d_s}^{(m,c)}(\mathbf{\Sigma}_{1:d_s,1:d_s}^{(m,c)})^{1/2} \in \mathbb{R}^{N_m \times d_s}. \tag{6}$$

Each row of $\mathbf{H}_{m,c}^{(0)}$, denoted as $\mathbf{h}_{i,m,c}^{(0)}$, corresponds to the d_s-dimensional embedding of sample $\mathbf{z}_i^{(m)}$, capturing its position within the relational structure of the dataset. Because d_s is fixed across all datasets in $\mathcal{H}$, this process maps every historical dataset $\mathcal{D}_m$ into a common relational representation space, irrespective of its original feature dimension d_m.

To further integrate label consistency, we define a label-relation matrix by

$$A_{ij}^{(m,\text{label})} = \mathbb{1}(y_i^{(m)} = y_j^{(m)}) \tag{7}$$

During GIN message passing, the geometric and label relations are then adaptively fused into edge weights using learnable parameters α and β:

$$W_{ij}^{(m,c)} = \frac{e^\alpha}{e^\alpha + e^\beta} A_{ij}^{(m,c)} + \frac{e^\beta}{e^\alpha + e^\beta} A_{ij}^{(m,\text{label})}. \tag{8}$$

Here, $\mathbf{W}^{(m,c)} \in \mathbb{R}^{N_m \times N_m}$ represents the learned edge-weight matrix used by the GIN branch of $\mathcal{D}_m$ at scale c. This learnable combination empowers GenSHAP to balance geometric proximity and label consistency during training automatically.

Ultimately, RDR addresses the cross-domain incompatibility outlined in Challenge I. Rather than requiring raw features to be strictly comparable across datasets, GenSHAP exploits informative relational structures among datasets. This framework allows the universal predictor f_Θ to be trained over the historical collection $\mathcal{H}$ and subsequently generalised to an unseen dataset $\mathcal{D}$ with varying feature dimensions and sample sizes.

3.2.2 Multi-branch GIN-Transformer architecture for modelling data interactions

To address Challenge II, we design a multi-branch GIN-Transformer architecture that operates over the previously constructed relational graphs. Because a sample's value is determined not only by its individual characteristics but also by its marginal contribution relative to others, the predictor f_Θ must explicitly model inter-sample dependencies. Consequently, we frame Data Shapley estimation as a node-wise regression task on each dataset graph.

For every historical dataset $\mathcal{D}_m \in \mathcal{H}$, let $\boldsymbol{\phi}^{(m)} = (\phi_1^{(m)}, \ldots, \phi_{N_m}^{(m)})$ denote the corresponding high-budget Permutation Data Shapley values computed under the learning algorithm $\mathcal{A}$. Given the multi-scale weighted graph $\{\mathbf{W}^{(m,c)} : c \in \mathcal{C}\}$ and their initial node embeddings $\{\mathbf{H}_{m,c}^{(0)} : c \in \mathcal{C}\}$, our objective is to learn a universal model f_Θ such that:

$$f_\Theta(\{\mathbf{W}^{(m,c)}, \mathbf{H}_{m,c}^{(0)}\}_{c\in\mathcal{C}}) \triangleq \hat{\boldsymbol{\phi}}^{(m)} \approx \boldsymbol{\phi}^{(m)} \tag{9}$$

for all $\mathcal{D}_m \in \mathcal{H}$. Once trained, the same model f_Θ can be applied to previously unseen datasets.

For each scale $c_b \in \mathcal{C}$, we employ a Graph Isomorphism Network (GIN) [17] branch to aggregate information across the weighted graph $\mathbf{W}^{(m,c_b)}$. Let $\mathbf{h}_{\text{GIN},i,m,b}^{(l)}$ denote the representation of node i in dataset $\mathcal{D}_m$, branch b, and layer l. The GIN update rule is defined as:

$$\mathbf{h}_{\text{GIN},i,m,b}^{(l)} = \text{MLP}_b^{(l)}\left((1+\epsilon_b^{(l)})\mathbf{h}_{\text{GIN},i,m,b}^{(l-1)} + \sum\nolimits_{j=1}^{N_m} W_{ij}^{(m,c_b)}\mathbf{h}_{\text{GIN},j,m,b}^{(l-1)}\right) \tag{10}$$

where $\mathbf{h}_{\text{GIN},i,m,b}^{(0)}$ is initialised from the corresponding SVD-based relational embedding $\mathbf{H}_{m,c_b}^{(0)}$. This aggregation mechanism allows each node to assimilate information from other samples based on their learned relational weights, effectively capturing both local and higher-order sample interactions (via multiple layers). The layer-l representation of dataset $\mathcal{D}_m$ in branch b is denoted as $\mathbf{H}_{\text{GIN},m,b}^{(l)}$.

While the GIN branches excel at modelling graph-based local neighbourhood aggregation, accurate Data Shapley estimation also relies on long-range and global interactions among samples. To capture these, we introduce a Transformer [18] encoder branch parallel to each GIN branch, operating on the scale-specific initial embedding matrix $\mathbf{H}_{m,c_b}^{(0)}$. For branch b at layer l, the Transformer representation is updated as follows:

$$\mathbf{Z}_{m,b}^{(l)} = \text{LayerNorm}(\mathbf{H}_{\text{GT},m,b}^{(l-1)} + \text{MHSA}_b^{(l)}(\mathbf{H}_{\text{GT},m,b}^{(l-1)})) \tag{11}$$

$$\mathbf{H}_{\text{GT},m,b}^{(l)} = \text{LayerNorm}(\mathbf{Z}_{m,b}^{(l)} + \text{FFN}_b^{(l)}(\mathbf{Z}_{m,b}^{(l)})) \tag{12}$$

The Multi-Head Self-Attention (MHSA) module enables all-to-all sample interactions, complementing the localised, weighted graph aggregation performed by the GIN branches.

In addition to the structural embeddings extracted by the GINs and Transformers from the RDR, we incorporate a suite of auxiliary features to capture sample-level optimisation behaviour and dataset-level context. These encompass gradient statistics $\boldsymbol{\vartheta}$, KNN-based neighbour reliability $\mathbf{r}$, temporal learning dynamics $\boldsymbol{\zeta}$, and global metadata $\boldsymbol{\eta}$. For instance, to explicitly encode local manifold reliability, the multi-scale KNN ratio $\mathbf{r}$ is computed by determining the fraction of a sample's k nearest neighbours – evaluated across a discrete set of neighbourhood sizes – that possess a conflicting label, thereby measuring local label heterogeneity. The exact mathematical formulations and standardisation procedures for all auxiliary features are detailed in Appendix B.2.

The final node representation is formed by concatenating outputs from all GIN and Transformer branches with injected auxiliary features. Suppose the GIN and Transformer have L_G and L_T layers, respectively. Let $\mathbf{h}_{\text{GIN},i,m,b}^{(L_G)}$ denote the final layer representation of node i from the GIN branch b, and $\mathbf{h}_{\text{GT},i,m,b}^{(L_T)}$ denote the corresponding i-th node representation extracted from the Transformer output matrix $\mathbf{H}_{\text{GT},m,b}$. We aggregate these structural embeddings across all scales $c_b \in \mathcal{C}$ and concatenate them with the standardised auxiliary features $\boldsymbol{\vartheta}_i$, $\mathbf{r}_i$, $\boldsymbol{\zeta}_i$, and $\boldsymbol{\eta}$. Formally, the final representation $\mathbf{u}_{i,m}$ for sample $\mathbf{z}_i$ in dataset $\mathcal{D}_m$ is defined as:

$$\mathbf{u}_{i,m} = \Big(\bigoplus_{b=1}^{|\mathcal{C}|}\left[\mathbf{h}_{\text{GIN},i,m,b}^{(L_G)} \oplus \mathbf{h}_{\text{GT},i,m,b}^{(L_T)}\right]\Big) \oplus \Big(\boldsymbol{\vartheta}_{i,m} \oplus \mathbf{r}_{i,m} \oplus \boldsymbol{\zeta}_{i,m} \oplus \boldsymbol{\eta}_m\Big) \tag{13}$$

where $\oplus$ denotes the concatenation operator. These representations are then passed through an MLP to produce the scaled final prediction:

$$\hat{\phi}_i^{(m,\text{scaled})} = \text{MLP}_{\text{out}}(\mathbf{u}_{i,m}) \tag{14}$$

Following (14), we normalise the training target as $\phi_i^{(m,\text{scaled})} = \phi_i^{(m)} \frac{N_m}{V(\mathcal{D}_m)}$. This normalization is important for learning a universal predictor across heterogeneous datasets. Assuming $V(\varnothing) = 0$, by the efficiency axiom of Shapley values, $\sum_i \phi_i^{(m)} = V(\mathcal{D}_m)$, so the average Data Shapley value in dataset $\mathcal{D}_m$ is $V(\mathcal{D}_m)/N_m$. Thus, raw Shapley values may differ substantially across datasets due to variations in dataset size and utility. The above scaling maps them to a scale-invariant space with an average value of 1, preventing datasets with large utilities or small sample sizes from dominating the regression loss, thereby improving optimisation stability. During inference, the scale-agnostic prediction is transformed back to the original Data Shapley scale by $\hat{\phi}_i^{(m)} = \hat{\phi}_i^{(m,\text{scaled})} \frac{V(\mathcal{D}_m)}{N_m}$.

Remark 3.2. Through this architecture, GenSHAP explicitly models the interaction structures underlying Data Shapley values. The GIN branches capture relation-weighted local and high-order dependencies, while the Transformer branches capture global dependencies across all samples. Since the parameters Θ are shared across all historical datasets in $\mathcal{H}$, the learned model f_Θ acts as a transferable data valuation function that generalises to unseen datasets.

3.2.3 KDE-weighted training objective

To reduce the influence of isolated Data Shapley values during training, we employ a kernel density estimation (KDE [22])-based weighting scheme where, for each historical dataset $\mathcal{D}_m \in \mathcal{H}$, the bandwidth $\sigma_{\text{kde}}^{(m)}$ is adaptively determined as the mean pairwise distance among its scaled target values $\boldsymbol{\phi}^{(m,\text{scaled})}$. The density at each point is estimated as:

$$\rho(\phi_i^{(m,\text{scaled})}) = \frac{1}{N_m} \sum_{j=1}^{N_m} K\left(\frac{\phi_i^{(m,\text{scaled})} - \phi_j^{(m,\text{scaled})}}{\sigma_{\text{kde}}^{(m)}} \right) \tag{15}$$

where K is the standard Gaussian kernel. The corresponding weight for each sample is assigned as: $w_i^{(m)} = \frac{\rho(\phi_i^{(m,\text{scaled})})}{\sum_j \rho(\phi_j^{(m,\text{scaled})})}$. Because GenSHAP is trained across highly heterogeneous historical datasets, extreme outliers in data values are often dataset-specific artifacts or label noise. By up-weighting the dense mass, we force the shared parameters Θ to learn fundamental, generalisable valuation patterns rather than overfitting to isolated anomalies. The final training objective is therefore given by:

$$\mathcal{J}(\Theta) = \sum_{m=1}^{M} \sum_{i=1}^{N_m} w_i^{(m)} \left(\hat{\phi}_i^{(m,\text{scaled})} - \phi_i^{(m,\text{scaled})} \right)^2. \tag{16}$$

Remark 3.3. By minimising this objective across a diverse corpus of historical data, GenSHAP successfully decouples the exponential cost of subset evaluation from the inference phase. Once Θ (involving the GIN, the Transformer, and the final MLP) is learned, the model can be deployed to provide efficient valuation on previously unseen datasets through representation construction followed by a single neural forward prediction.

3.3 GenSHAP-S: Injecting the power of sampling

Beyond the deterministic graph representation used in GenSHAP, we further introduce a sampling-augmented variant, denoted as GenSHAP-S, where "S" refers to sampled marginal contribution features. The key motivation is that Data Shapley is fundamentally defined through marginal utility differences over coalitions. Although exact enumeration is computationally intractable, a small number of sampled coalitions can provide informative local signals about whether a data point tends to improve or degrade model utility when inserted into different training contexts.

For each data point $\mathbf{z}_i^{(m)}$ in dataset $\mathcal{D}_m$, we draw a small budget of P subsets $S_{i,p} \subseteq \mathcal{D}_m \backslash \{\mathbf{z}_i^{(m)}\}$ of varying sizes. Given a proxy utility function $V(\cdot)$, we compute the sampled marginal contribution $\Delta_{i,p} = V(S_{i,p} \cup \{\mathbf{z}_i^{(m)}\}) - V(S_{i,p})$. Repeating this procedure P times yields a stochastic marginal

profile for each data point, denoted by $\mathbf{q}_{i,m} = [\Delta_{i,1}, \Delta_{i,2}, \ldots, \Delta_{i,P}] \in \mathbb{R}^P$. The detailed sampling strategy, subset constraints, and standardisation procedures are deferred to Appendix B.1.

To construct GenSHAP-S, we inject these sampled marginal contribution probes directly into our comprehensive node representation. Let $\mathbf{u}_{i,m}$ denote the final node representation for sample $\mathbf{z}_i^{(m)}$ established in (14). Specifically, GenSHAP-S extends (14) by

$$\hat{\phi}_i^{(m,\text{scaled})} = \text{MLP}_{\text{out}}(\mathbf{u}_{i,m} \oplus \mathbf{q}_{i,m}) \tag{17}$$

Importantly, this sampling module does not aim to approximate the full Shapley summation directly. In contrast to classical sampling-based methods such as TMC Shapley and Owen Shapley, which rely on a large number of sampled permutations to obtain accurate estimates, GenSHAP-S uses a limited budget of sampled subsets to construct informative predictive features. These low-budget samples are not used to reconstruct Shapley values explicitly, but instead serve as stochastic evidence of each data point's marginal behaviour under diverse training contexts. In this way, GenSHAP-S preserves the amortised graph-based prediction pipeline while efficiently injecting utility-aware marginal probes that remain closely aligned with the definition of Data Shapley.

3.4 Computational complexity analysis

Assume that the hidden dimensions of GINs and Transformers scale as $\mathcal{O}(d_s)$. Let $T_{\text{aux}}(N, d, K)$ denote the cost of auxiliary feature extraction, and let $T_{\text{train}}(k_{\max}, d)$ denote the cost of training and evaluating $\mathcal{A}$ on a subset of size at most $k_{\max} + 1$.

Theorem 3.4. *Under the assumptions detailed in Appendix A, preprocessing and graph construction require $\mathcal{O}(N^2 d + |\mathcal{C}|N^3 + T_{\text{aux}}(N, d, K))$ with full SVD. The neural training cost per epoch is $\mathcal{O}(M|\mathcal{C}|(L_G + L_T)N^2 d_s)$, and the neural inference cost is $\mathcal{O}(|\mathcal{C}|(L_G + L_T)N^2 d_s)$. End-to-end inference additionally includes preprocessing. For GenSHAP-S, marginal contribution probes add $\mathcal{O}(NP \cdot T_{\text{train}}(k_{\max}, d))$.*

4 Experiments

4.1 Experiment settings

Datasets and reference valuation We evaluate GenSHAP across diverse tabular and image classification tasks, including benchmarks such as *Iris* [23], *Characters* [24], *Vehicles* [25]. For multi-class datasets, we construct K-way classification tasks by randomly selecting subsets of classes, typically with $K \in [2, 5]$. The detailed train and unseen test dataset list is provided in Appendix D.1.

To manage the computational cost of generating reference values, we apply stratified subsampling to construct valuation tasks containing approximately 800 to 1000 training samples each. To train GenSHAP and GenSHAP-S, we compile a historical corpus of 100 valuation tasks from public repositories (e.g., *scikit-learn* [26], *TensorFlow* [27], *OpenML* [28], and *UCI* [29]). Five source-disjoint datasets are reserved for evaluation, with no data from these sources used for training.

We use high-budget Permutation Shapley [20] estimates as the reference valuation signal for supervision and evaluation. In our implementation, the estimator averages marginal contributions over a fixed number of random permutations ($R = 5000$) for most datasets. No early stopping is used in this reference-estimation routine.

Utility functions and base models To assess robustness across different model classes, we compute high-budget Permutation Data Shapley reference estimates under two distinct base learning algorithms $\mathcal{A}$: Logistic Regression and Support Vector Machine (with RBF kernel).

Baselines and Evaluation Metrics We benchmark GenSHAP and GenSHAP-S against a comprehensive set of representative data valuation methods:

- Sampling-based approaches: TMC Shapley [6], Delta-Shapley [14], Group-Testing Shapley [7], Owen-Shapley [30], Beta-Shapley [8], Data Banzhaf [31].
- Learning-based approaches: MLPbV [15].

Due to space limitations, we defer the implementation details to Appendix D.

4.2 Data removal and addition plots

Data removal and addition experiments evaluate ranking quality: prioritising high-value samples should induce rapid utility changes (rapid gains during addition, sharp drops during removal), while prioritising low-value samples should exert minimal impact (slower gains during addition, smaller drops during removal). Figure 2 demonstrates that on the *Characters* and *Volkert* datasets, both GenSHAP and GenSHAP-S closely follow this reference pattern: adding high-valued samples first leads to faster utility gains, while removing high-valued samples first causes a sharper utility drop than removing low-valued samples. Notably, GenSHAP-S aligns even more tightly with this baseline, confirming that injecting sampled marginal contribution probes enhances valuation fidelity.

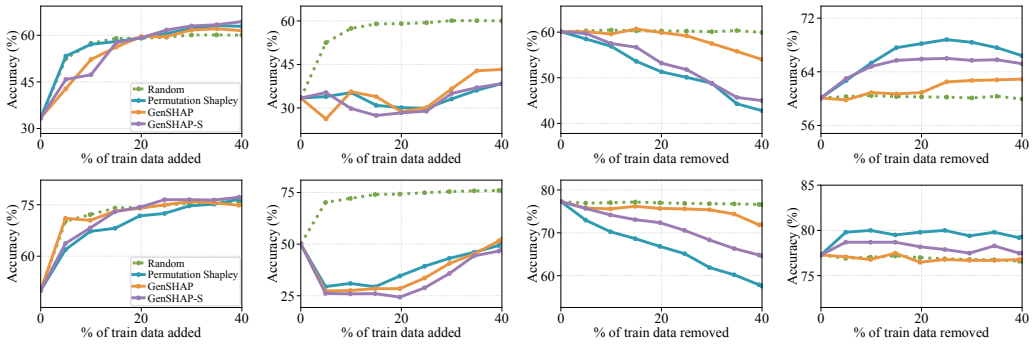


Figure 2: Data addition and removal evaluation on *Characters* [24] (row-1, multi-class classification) and *Volkert* [32](row-2, binary classification). These two datasets have around 1000 data points.

4.3 Running time and consistency

Table 1 compares the efficiency of GenSHAP and GenSHAP-S with baseline methods. Classical sampling methods such as TMC, Beta, and Owen Shapley are costly because they require model retraining; Owen Shapley even exceeds 10^5 seconds on large datasets such as *STL10*[33]. Delta and Group-Testing Shapley also remain computationally expensive. In contrast, learning-based methods amortise this cost into a one-time training phase. With our terminology, inference time denotes the full wall-clock time required to output values for a new dataset. For GenSHAP, this includes feature and graph construction plus neural forward prediction; for GenSHAP-S, it additionally includes sampled marginal contribution feature construction. GenSHAP finishes end-to-end inference in 0.14 to 1.45 seconds across the tested datasets, while the neural forward prediction alone is below 0.03 seconds (in GPU). GenSHAP-S is slower end-to-end because the marginal probes require repeated small-subset utility evaluations, but its final neural forward prediction remains millisecond-scale.

Table 1: Runtime comparison (in seconds) across datasets.

Method	**Iris**	**Vehicle**	**Characters**	**STL10**	**Volkert**
TMC Shapley	437.44	2440.63	5208.79	9895.02	5548.09
Delta Shapley	156.47	14199.68	26165.01	37081.73	26224.87
Group-Testing Shapley	6.04	364.51	6363.69	5627.85	6714.07
Owen Shapley	493.59	43892.85	88220.85	143653.55	87417.92
Beta Shapley	66.44	2681.15	5206.11	13422.76	5100.29
Data Banzhaf	65.91	2680.16	5074.26	10228.96	5035.65
GenSHAP	**0.14**	**0.54**	**1.12**	**1.45**	**1.39**
GenSHAP-S	60.12	551.81	793.67	1040.96	770.88

We evaluate approximation fidelity by measuring the MSE between each sampling-based method's estimated values and the high-budget Permutation Data Shapley values, which serves as the reference supervision signal. Owen Shapley achieves the lowest MSE overall, reflecting its high sampling fidelity but also its much larger computational cost. TMC Shapley also remains highly accurate, while Group-testing Shapley and especially MLPbV show larger errors on more complex datasets. GenSHAP achieves lower MSE than MLPbV on four of the five datasets, while GenSHAP-S outperforms MLPbV on all five, showing that graph-based relational modelling provides a more faithful learned approximation to Permutation Data Shapley. GenSHAP-S further improves over GenSHAP on every dataset, confirming the benefit of adding sampled marginal contribution features.

Table 2: Consistency evaluation in terms of mean squared error (MSE).

Method	Iris	Vehicle	Characters	STL10	Volkert
TMC Shapley	0.00005100	0.00000095	0.00000041	0.00000022	0.00000038
Delta Shapley	0.00013700	0.00000408	0.00000083	0.00000045	0.00000081
Group-Testing Shapley	0.03945400	0.00556213	0.00185878	0.00121310	0.00306043
Owen Shapley	0.00000300	0.00000013	0.00000004	0.00000002	0.00000007
MLPbV	0.00012900	0.00012055	0.00008788	0.00008770	0.00006617
GenSHAP	0.00018117	0.00000312	0.00000119	0.00000053	0.00000066
GenSHAP-S	0.00008299	0.00000221	0.00000041	0.00000033	0.00000037

4.4 Noisy label detection

We evaluate the framework's ability to identify corrupted examples by randomly flipping 10% of the training labels. Using low valuation scores as indicators of potentially harmful samples, we rank the training data in ascending order of their estimated values and track the percentage of noisy labels successfully identified.

As a representative example, we analyse a subset of *STL10* comprising 964 training points and 96 injected noisy labels. According to Figure 3, our method demonstrates a distinct advantage over random inspection and the majority of established baselines. At the 50% data inspection rate, GenSHAP successfully identifies approximately 75% of the noisy labels. While computationally intensive methods like TMC and Owen Shapley achieve slightly higher detection rates (around 80% at this threshold), our method remains highly competitive while operating at a mere fraction of their computational cost. Moreover, GenSHAP-S outperforms Owen at the 30% data inspection rate. This confirms that our model can identify corrupted samples early while avoiding the substantial computational overhead of high-budget marginal evaluations.

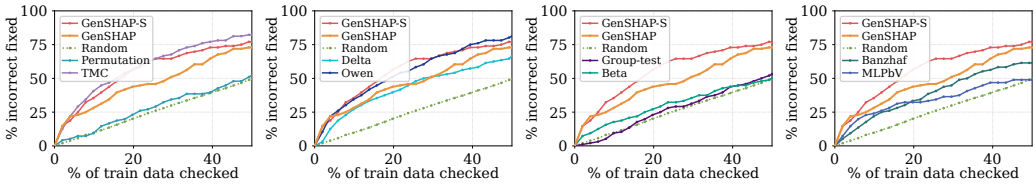


Figure 3: Noisy label detection on the flipped *STL10* dataset. To prevent the curves from becoming tangled, each plot compares our methods with two valuation baselines plus random inspection.

More results Appendices B-F provide detailed methodology, ablation studies, implementation details, additional experiments, and limitations. Appendix B describes the sampling module and auxiliary features, while Appendix C presents architecture, feature, and loss ablations. Appendix D provides implementation details and dataset configurations. Appendix E examines generalisation, repeated-run stability, robustness, reference uncertainty, sampling budgets, and scalability. Matched-hardware CPU-only evaluations confirm end-to-end GenSHAP inference times(Appendix E.7), complementing the GPU-assisted results in Table 1. Offline costs and amortisation are analysed in Appendix E.8. Finally, Appendix F discusses limitations and broader impact.

5 Conclusion

In this paper, we proposed GenSHAP, a graph-based framework for generalisable data valuation. By casting Data Shapley estimation as a node-wise regression problem, GenSHAP amortises the costly valuation process into a single training phase and enables efficient inference on new datasets. Its multi-branch architecture combines GINs and Transformer encoders to capture both local high-order interactions and global dependencies among samples. We further introduced GenSHAP-S, which incorporates low-budget sampled marginal probes to provide additional utility-aware features. Experiments on diverse datasets show that GenSHAP and GenSHAP-S closely approximate high-budget Permutation Shapley estimates, preserve downstream utility trends in data addition and removal tasks, while GenSHAP achieves substantial online efficiency gains and GenSHAP-S improves approximation fidelity at additional sampling cost. These results demonstrate that data valuation can be learned in a transferable and structurally informed manner. Limitations and social impact are discussed in the appendix.

Acknowledgments and Disclosure of Funding

The authors thank the reviewers and the area chair for reviewing the work and for their valuable comments.

This work was supported in part by the project "Quality Assessment of Data Assets in Public TCM Hospitals" (Grant No. MSRC-PFS-2026-005) of the Testing and Statistics Center, National Administration of Traditional Chinese Medicine.

References

[1] Junwei Deng, Ting-Wei Li, Shiyuan Zhang, Shixuan Liu, Yijun Pan, Hao Huang, Xinhe Wang, Pingbang Hu, and Xingjian Zhang. `dattri`: A library for efficient data attribution. *Advances in Neural Information Processing Systems*, 37:136763–136781, 2024.

[2] Malick Ebiele, Malika Bendechache, and Rob Brennan. Quantitative data valuation methods: A systematic review and taxonomy. *ACM Journal of Data and Information Quality*, 17(2):1–39, 2025.

[3] Julia Schweihoff, Anzelika Lipovetskaja, Ilka Jussen-Lengersdorf, and Frederik Möller. Stuck in the middle with you: Conceptualizing data intermediaries and data intermediation services. *Electronic Markets*, 34(1):48, 2024.

[4] Safwan Hossain and Yiling Chen. Equilibrium of data markets with externality. *arXiv preprint arXiv:2302.08012*, 2023.

[5] Chao Li, Daniel Yang Li, Gerome Miklau, and Dan Suciu. A theory of pricing private data. *ACM Transactions on Database Systems (TODS)*, 39(4):1–28, 2014.

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

[7] 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. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[8] Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

[9] Yongchan Kwon and James Zou. Data-oob: Out-of-bag estimate as a simple and efficient data value. In *International conference on machine learning*, pages 18135–18152. PMLR, 2023.

[10] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023.

[11] Ryuichiro Hataya and Makoto Yamada. Nyström method for accurate and scalable implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 4643–4654. PMLR, 2023.

[12] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In *The 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence*, pages 5572–5579. International Joint Conferences on Artificial Intelligence Organization, 2022.

[13] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.

[14] Lauren Watson, Zeno Kujawa, Rayna Andreeva, Hao-Tsung Yang, Tariq Elahi, and Rik Sarkar. Accelerated shapley value approximation for data evaluation. *arXiv preprint arXiv:2311.05346*, 2023.

[15] Ou Wu, Weiyao Zhu, and Mengyang Li. Is data valuation learnable and interpretable? *arXiv preprint arXiv:2406.02612*, 2024.

[16] Hongliang Chi, Qiong Wu, Zhengyi Zhou, and Yao Ma. Shapley-guided utility learning for effective graph inference data valuation. *arXiv preprint arXiv:2503.18195*, 2025.

[17] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[19] Lloyd S. Shapley. Notes on the n-person game—ii: The value of an n-person game. Research memorandum, RAND Corporation, 1951.

[20] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & operations research*, 36(5):1726–1730, 2009.

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

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

[23] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[24] H. A. Guvenir, Burak Acar, and Haldun Muderrisoglu. Artificial Characters. UCI Machine Learning Repository, 1992. DOI: `https://doi.org/10.24432/C5303Z`.

[25] Pete Mowforth and Barry Shepherd. Statlog (Vehicle Silhouettes). UCI Machine Learning Repository, 1997. DOI: https://doi.org/10.24432/C5HG6N.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[27] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[28] Bernd Bischl, Giuseppe Casalicchio, Taniya Das, Matthias Feurer, Sebastian Fischer, Pieter Gijsbers, Subhaditya Mukherjee, Andreas C Müller, László Németh, and Luis Oala. Openml: Insights from 10 years and more than a thousand papers. *Patterns*, 6(7), 2025.

[29] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository. `https://archive.ics.uci.edu`, 2023. Accessed: 2026-05-06.

[30] Ramin Okhrati and Aldo Lipani. A multilinear sampling algorithm to estimate shapley values. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7992–7999. IEEE, 2020.

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

[32] Isabelle Guyon, Lisheng Sun-Hosoya, Marc Boullé, Hugo Jair Escalante, Sergio Escalera, Zhengying Liu, Damir Jajetic, Bisakha Ray, Mehreen Saeed, Michéle Sebag, Alexander Statnikov, WeiWei Tu, and Evelyne Viegas. Analysis of the automl challenge series 2015-2018. In *AutoML*, Springer series on Challenges in Machine Learning, 2019.

[33] Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*, 2011. `https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf`.

A Proof for Theorem 3.4

Proof. We decompose the computational complexity into four distinct phases: preprocessing and graph construction, model training, inference, and the sampling module in GenSHAP-S.

Preprocessing and graph construction: Constructing the pairwise Euclidean distance matrix for $\mathcal{D}$ requires computing distances between all pairs of samples, which costs $\mathcal{O}(N^2 d)$. Given this matrix, each Gaussian-kernel adjacency matrix $\mathbf{A}^{(c)}$ for $c \in \mathcal{C}$ can be constructed in $\mathcal{O}(N^2)$ time, and the label-relation matrix is also computed in $\mathcal{O}(N^2)$ time. These terms are dominated by the pairwise distance computation when $d \geq 1$. For each of the $B = |\mathcal{C}|$ scales, GenSHAP applies full SVD to an $N \times N$ relational matrix, which costs $\mathcal{O}(N^3)$ per scale, hence $\mathcal{O}(BN^3)$ in total. Including the auxiliary feature extraction cost $T_{\text{aux}}(N, d, K)$, the overall preprocessing complexity is $\mathcal{O}(N^2 d + BN^3 + T_{\text{aux}}(N, d, K))$.

Model training: During the amortised training phase over M historical datasets, the neural network computes representations across the relational graphs. Each graph is fully connected, meaning the number of directed edges is $\mathcal{O}(N^2)$. Since the hidden dimensions of both GINs and Transformers scale as $\mathcal{O}(d_s)$, in each GIN layer, message passing over the graph costs $\mathcal{O}(N^2 d_s)$. Across B branches and L_G layers, the GIN component therefore costs $\mathcal{O}(BL_G N^2 d_s)$. Each Transformer branch performs all-to-all self-attention over N samples, costing $\mathcal{O}(N^2 d_s)$ per layer, yielding $\mathcal{O}(BL_T N^2 d_s)$ across all Transformer branches. Assuming $d_s = \mathcal{O}(N)$ and fixed MLP depth, the node-wise operations are dominated by the quadratic graph and attention computations. Aggregating these dominant terms across the M datasets yields a per-epoch training complexity of $\mathcal{O}(MB(L_G + L_T)N^2 d_s)$.

Inference: For a new, unseen dataset, end-to-end inference requires executing both the preprocessing pipeline and a single neural forward prediction. The preprocessing cost is $\mathcal{O}(N^2 d + BN^3 + T_{\text{aux}}(N, d, K))$, while the neural forward pass costs $\mathcal{O}(B(L_G + L_T)N^2 d_s)$. Therefore, the overall end-to-end inference complexity is $\mathcal{O}(N^2 d + BN^3 + T_{\text{aux}}(N, d, K) + B(L_G + L_T)N^2 d_s)$.

Sampling module in GenSHAP-S: GenSHAP-S augments the base framework with sampled marginal contribution probes during preprocessing. For each data point $\mathbf{z}_i$, it samples P coalitions $S_{i,p}$ of size at most $k_{\max}$, and evaluates $V(S_{i,p} \cup \{\mathbf{z}_i\}) - V(S_{i,p})$. Each probe requires a constant number of train-and-evaluate calls of $\mathcal{A}$ on at most $k_{\max} + 1$ samples, which is $\mathcal{O}(T_{\text{train}}(k_{\max}, d))$. Over all N data points and P probes per point, this contributes an additional cost of $\mathcal{O}(NP \cdot T_{\text{train}}(k_{\max}, d))$. Adding this term to the GenSHAP cost yields the stated GenSHAP-S complexity. Since P and $k_{\max}$ are user-controlled constants in practice, GenSHAP-S remains polynomial and avoids the exponential $\mathcal{O}(2^N)$ complexity of exact Data Shapley computation.

□

B Detailed methodology of GenSHAP and GenSHAP-S

B.1 Sampling module in GenSHAP-S

This section provides the exact implementation details and hyperparameter configurations for the marginal contribution sampling module in GenSHAP-S.

For each training point $\mathbf{z}_i^{(m)}$ in dataset $\mathcal{D}_m$, we sample P subsets from $\mathcal{D}_m \backslash \{\mathbf{z}_i^{(m)}\}$. For the p-th probe, we first draw a subset size $k_{i,p}$ uniformly from $\{k_{\min}, \ldots, k_{\max}\}$. To ensure the target utility model is trainable and captures meaningful signals, we set $k_{\min} = \max(5, K + 1)$. We then sample a coalition $S_{i,p} \subseteq \mathcal{D}_m \backslash \{\mathbf{z}_i^{(m)}\}$ with $|S_{i,p}| = k_{i,p}$, enforcing the constraint that the sampled subset contains at least two distinct classes.

The proxy utility function $V(\cdot)$ is implemented as the holdout accuracy of a Logistic Regression model. After computing the P marginal contributions $\Delta_{i,p} = V(S_{i,p} \cup \{\mathbf{z}_i^{(m)}\}) - V(S_{i,p})$ for each sample, we obtain the raw marginal profile $\mathbf{q}_{i,m} = [\Delta_{i,1}, \Delta_{i,2}, \ldots, \Delta_{i,P}] \in \mathbb{R}^P$.

To harmonise these stochastic signals with the structural graph embeddings and ensure stable gradients during the training of the $\mathrm{MLP}_{\text{out}}$ regression head, each of the P marginal dimensions is z-score standardised across all N_m data points within the dataset $\mathcal{D}_m$. The standardised profile $\mathbf{q}_{i,m}$ is then concatenated with the deterministic representation $\mathbf{u}_{i,m}$.

B.2 Auxiliary feature extraction

In the main text, we briefly note the incorporation of auxiliary signals to complement the structural embeddings derived from our multi-branch GIN-Transformer architecture. This section provides the comprehensive mathematical definitions of these auxiliary features.

Gradient and prediction statistics To capture sample-level optimisation behaviour, let $\ell(\mathbf{z}_i;\theta)$ be the loss of sample $\mathbf{z}_i$ and $g_i = \nabla_\theta \ell(\mathbf{z}_i;\theta)$ its sample-wise gradient. We use the gradient norm $\|g_i\|_2$ to measure the update magnitude induced by $\mathbf{z}_i$. We additionally define the prediction error as

$$e_i = 1 - p_\theta(y_i \mid \mathbf{x}_i). \tag{18}$$

This quantity measures sample-level prediction difficulty, with larger values indicating lower model confidence in the true class. Both features are standardised across the dataset: $\widetilde{g}_i = (\|g_i\|_2 - \mu_g)/\sigma_g$ and $\widetilde{e}_i = (e_i - \mu_e)/\sigma_e$, where the corresponding means and standard deviations are computed over all samples. Their concatenation, $\boldsymbol{\vartheta}_i = [\widetilde{g}_i, \widetilde{e}_i]$, forms the final optimisation-statistics vector.

This quantity measures how atypical a sample's gradient is relative to the dataset, with larger values indicating potential outliers or conflicting training signals. Finally, both features are standardised across the dataset: $\tilde{g}_i = \frac{\|\mathbf{g}_i\|_2 - \mu_g}{\sigma_g}$, $\tilde{e}_i = \frac{e_i - \mu_e}{\sigma_e}$, where μ_g, σ_g and μ_e, σ_e denote the empirical mean and standard deviation computed over all samples. These standardised metrics are then concatenated to form the final gradient statistics vector for the sample, denoted as $\boldsymbol{\vartheta}_i = [\tilde{g}_i, \tilde{e}_i]$.

KNN-based reliability Structural consistency and local manifold reliability are encoded through the injection of multi-scale K-nearest neighbour (KNN) ratios as auxiliary node features. For each training sample $\mathbf{z}_i = (\mathbf{x}_i, y_i)$, the ratio $r_k(\mathbf{z}_i)$ characterises the local neighbourhood heterogeneity by calculating the fraction of its k nearest neighbours that possess a label distinct from y_i. Formally, for a given neighbourhood size k, the ratio is defined as:

$$r_k(\mathbf{z}_i) = \frac{1}{k} \sum_{\mathbf{z}_j \in \mathcal{N}_k(\mathbf{z}_i)} \mathbb{1}(y_j \neq y_i) \tag{19}$$

where $\mathcal{N}_k(\mathbf{z}_i)$ represents the set of k nearest neighbours of $\mathbf{z}_i$ based on Euclidean distance, and $\mathbb{1}(\cdot)$ denotes the indicator function. Following the technical implementation in our pipeline, we compute these ratios across a discrete set of scales $k \in \{5, 7, 9, 10, 15, 20, 30\}$ to provide the model with a multi-resolution characterisation of local label noise.

Learning dynamics The temporal optimisation behaviour of individual samples is characterised through learning dynamics features, which represent the evolution of pointwise loss during the initial training phase. For each sample $\mathbf{z}_i$, we record the negative log-likelihood at 21 discrete intervals, specifically at the first iteration and every subsequent 10 iterations up to 200. The loss at iteration t is formally defined as:

$$\ell_i^{(t)} = -\log(P(y_i \mid \mathbf{x}_i; \boldsymbol{\theta}_t)) \tag{20}$$

where $\boldsymbol{\theta}_t$ denotes the model parameters at the specific iteration. Beyond absolute loss values, we compute sequential loss differences $\Delta\ell_i^{(t)} = \ell_i^{(t)} - \ell_i^{(t+10)}$ to provide a granular measure of convergence velocity and stability. These 41-dimensional temporal signatures are standardised to form the temporal learning dynamics vector, denoted as $\boldsymbol{\zeta}_i$. This design enables the model to effectively differentiate between high-utility samples that converge rapidly and "hard" or noisy samples that exhibit stagnant or volatile loss profiles throughout the optimisation process.

Metadata To provide the model with global context regarding the performance baseline of the learning task, we incorporate metadata features that characterise the dataset's overall utility and complexity. In our framework, the total utility $V(\mathcal{D})$ is specifically defined as the classification accuracy of a Logistic Regression model trained on the full dataset $\mathcal{D}$ and evaluated on a held-out test

set. To ensure the regression head can account for different performance scales across heterogeneous datasets, we construct a global utility vector:

$$\mathbf{v}_{\text{feat}} = [V(\mathcal{D}), 1/\max(V(\mathcal{D}), \epsilon)] \tag{21}$$

where $\epsilon = 10^{-12}$ is a small constant for numerical stability. This vector is concatenated with class-count statistics, specifically the number of classes K, its reciprocal $1/K$, and its logarithm $\log K$, to form the dataset-level global metadata vector, denoted as $\boldsymbol{\eta} = [\mathbf{v}_{\text{feat}}, K, 1/K, \log K]$. This global vector is then tiled across all N samples in the dataset. These descriptors serve as essential normalisation signals, allowing the architecture to calibrate the predicted Data Shapley values relative to the difficulty and scale of the specific classification task.

C Ablation studies

C.1 Impact of architecture blocks

We first conduct an architecture-level ablation study to examine the contribution of the main representation blocks in GenSHAP. This experiment isolates the modelling capacity of the architecture itself by comparing the full GIN-Transformer model with three specific variants: an MLP-only model, an GIN-only model, and a Transformer-only model. All variants use the same preprocessing pipeline, graph construction, SVD-based node features, label-related adjacency information, auxiliary feature injection, density-weighted objective, and evaluation protocol. Therefore, the performance differences mainly reflect the representational ability of each architecture block rather than changes in input features, training objective, or evaluation setup.

As shown in Table 3, each individual block captures some useful information, but none of the single-block variants consistently matches the full GIN-Transformer model across datasets. Although the MLP-only variant obtains a low error on *Iris*, it performs worse than the full model on *Vehicle*, *Characters*, *STL10*, and *Volkert*, showing that SVD features and auxiliary statistics alone are not sufficient for robust data valuation. The GIN-only variant achieves a low error on *Characters*, indicating that local graph aggregation is useful, but it degrades substantially on *Vehicle*, *STL10*, and *Volkert*. Similarly, the Transformer-only variant performs competitively on *Vehicle* and *Volkert*, but it is clearly worse on *Iris* and *STL10*. In contrast, the full GIN-Transformer model achieves the best performance on *Vehicle* and *STL10*, and remains competitive on the remaining datasets, demonstrating that combining local graph propagation with global sample interaction leads to a more robust architecture.

Figure 4 provides a distribution-level comparison between Permutation Data Shapley values and predictions from the single-block architecture variants. The MLP-only model often produces distributions that are overly concentrated, suggesting that SVD features and auxiliary statistics along cannot fully recover the target value distribution. The GIN-only model better captures some local relational patterns, especially when neighbourhood structure is informative, but it can still produce shifted distributions because local graph aggregation alone does not model global sample interactions. The Transformer-only model captures broader dependencies among samples, yet its predictions may deviate from the target distribution when explicit graph propagation is needed. Overall, these distribution plots are consistent with the quantitative architecture ablation results: each individual block captures part of the signal, but none of them provides the same balanced distributional alignment as the full GIN-Transformer architecture.

Table 3: Ablation study of architecture blocks in GenSHAP. We report the MSE between predicted values and Permutation Data Shapley values. All variants use the same preprocessing, graph construction, auxiliary features, density-weighted objective, and evaluation protocol.

Architecture	Iris	Vehicle	Characters	STL10	Volkert
GIN+Transformer	1.81×10^{-4}	3.12×10^{-6}	1.19×10^{-6}	5.27×10^{-7}	6.55×10^{-7}
MLP	6.76×10^{-5}	5.54×10^{-6}	1.24×10^{-6}	5.30×10^{-7}	6.75×10^{-7}
GIN	3.00×10^{-5}	7.77×10^{-6}	9.20×10^{-7}	7.16×10^{-7}	9.12×10^{-7}
Transformer	2.00×10^{-4}	3.39×10^{-6}	1.16×10^{-6}	8.42×10^{-7}	5.91×10^{-7}

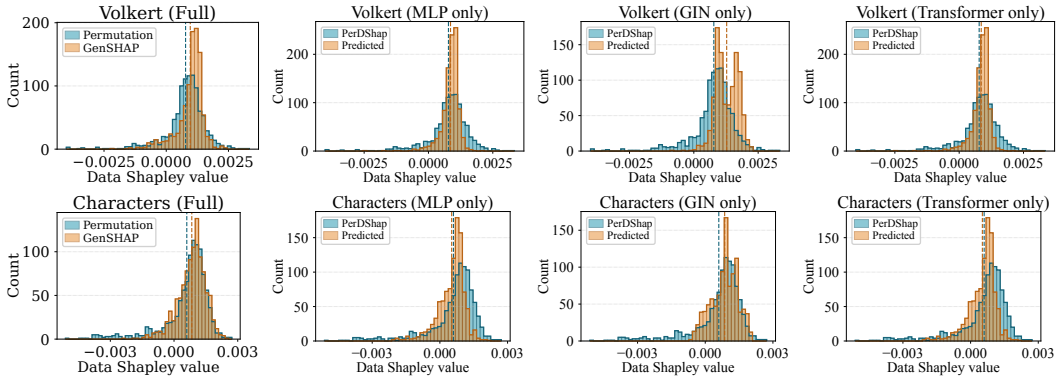


Figure 4: Distribution comparison between high-budget Permutation Data Shapley values and GenSHAP predictions under architecture-block ablations. We compare single-block variants using only MLP, only GIN, or only Transformer branches with our GIN-Transformer architecture on *Volkert* and *Characters*. The plots show that individual blocks capture partial signals but can produce shifted or over-concentrated predicted distributions, supporting the full GIN-Transformer design that combines local graph propagation with global sample interactions.

C.2 Impact of auxiliary features

Beyond the relational representations learned by the multi-branch GIN-Transformer architecture, GenSHAP injects four groups of auxiliary features into the final prediction head: gradient statistics $\boldsymbol{\vartheta}_{i,m}$, KNN-based neighbour reliability $\mathbf{r}_{i,m}$, learning dynamics $\boldsymbol{\zeta}_{i,m}$, and dataset-level metadata $\boldsymbol{\eta}_m$. Although these features are motivated by different aspects of data valuation, it is important to verify whether each group contributes useful information, or whether some of them become redundant once the relational graph representations have been learned. We therefore conduct two complementary ablation studies. First, we remove one auxiliary feature group at a time while keeping the graph construction, GIN-Transformer architecture, training objective, and evaluation protocol unchanged. In each removal-based variant, the corresponding feature block is replaced by zero-valued placeholders of the same dimension, so that the regression head remains identical to that of the full GenSHAP model. Second, we perform a stricter inclusion-only ablation, where only one auxiliary feature group is retained at a time, together with a variant that removes all auxiliary features.

The results in Table 4 show that the four auxiliary feature groups provide complementary information. Removing gradient statistics substantially increases the MSE on *Iris*, *Vehicle*, and *STL10*, suggesting that sample-level optimisation difficulty is useful for calibrating data values. Removing KNN reliability features worsens performance on *Characters*, *STL10*, and *Volkert*, indicating that local label heterogeneity provides information not fully captured by the relational graph alone. Removing learning dynamics causes a large degradation on *Iris* and also increases the error on *STL10*, confirming that temporal loss behaviour contains useful evidence about whether a sample is easy, hard, or unstable during optimisation. Finally, removing metadata weakens performance especially on *Iris*, showing that dataset-level utility and class-count information help calibrate predictions across heterogeneous tasks.

The inclusion-only ablation further clarifies the role of each auxiliary signal. As suggested by Table 4, retaining only one feature group is generally insufficient to recover the behaviour of the full model across all datasets. For example, using only gradient statistics $\boldsymbol{\vartheta}_{i,m}$ gives competitive error on *Characters* and *Volkert*, but leads to clear degradation on *Iris*, *Vehicle*, and *STL10*. Keeping only KNN reliability $\mathbf{r}_{i,m}$ substantially worsens the error on *Iris* and *Characters*, while using only learning dynamics $\boldsymbol{\zeta}_{i,m}$ improves over some single-feature variants but remains weak on *Volkert*. Metadata alone $\boldsymbol{\eta}_m$ is helpful on *Iris*, yet it cannot consistently replace sample-level signals on the remaining datasets. The no-auxiliary variant provides an additional reference point: although it occasionally achieves a small MSE on a specific dataset, such as *Iris*, this behaviour is not robust across the benchmark and does not imply that the auxiliary signals are redundant.

Figure 5 provides distribution-level evidence consistent with the quantitative results. In representative datasets such as *Characters* and *Volkert*, the removal-based ablations show that dropping individual feature groups can shift or concentrate the predicted distributions relative to the Permutation Data Shapley values. The inclusion-only and no-auxiliary variants in the last two rows show an even clearer

loss of distributional fidelity, indicating that a single auxiliary signal is not sufficient to reproduce the overall shape of the target distribution. Taken together, the table and distribution plots justify retaining all four auxiliary feature groups in GenSHAP, since they encode distinct and complementary signals: optimisation behaviour, local neighbourhood reliability, temporal learning dynamics, and global dataset context.

Table 4: Ablation study of auxiliary features in GenSHAP. We report the MSE between predicted values and the Permutation Data Shapley values. The full model uses gradient statistics $\boldsymbol{\vartheta}_{i,m}$, KNN reliability $\mathbf{r}_{i,m}$, learning dynamics $\boldsymbol{\zeta}_{i,m}$, and metadata $\boldsymbol{\eta}_m$.

Auxiliary features	**Iris**	**Vehicle**	**Characters**	**STL10**	**Volkert**
No auxiliary features	4.49×10^{-5}	4.51×10^{-6}	1.31×10^{-6}	5.73×10^{-7}	7.35×10^{-7}
Only $\boldsymbol{\vartheta}_{i,m}$	4.08×10^{-4}	5.91×10^{-6}	9.79×10^{-7}	7.53×10^{-7}	6.26×10^{-7}
Only $\mathbf{r}_{i,m}$	6.23×10^{-4}	3.44×10^{-6}	1.42×10^{-6}	5.61×10^{-7}	6.69×10^{-7}
Only $\boldsymbol{\zeta}_{i,m}$	3.35×10^{-4}	4.23×10^{-6}	1.04×10^{-6}	5.39×10^{-7}	1.03×10^{-6}
Only $\boldsymbol{\eta}_m$	1.11×10^{-4}	4.31×10^{-6}	1.31×10^{-6}	7.57×10^{-7}	9.17×10^{-7}
$\mathbf{r}_{i,m} \oplus \boldsymbol{\zeta}_{i,m} \oplus \boldsymbol{\eta}_m$	4.43×10^{-4}	3.61×10^{-6}	1.03×10^{-6}	6.62×10^{-7}	5.73×10^{-7}
$\boldsymbol{\vartheta}_{i,m} \oplus \boldsymbol{\zeta}_{i,m} \oplus \boldsymbol{\eta}_m$	2.97×10^{-4}	2.80×10^{-6}	1.30×10^{-6}	6.20×10^{-7}	8.27×10^{-7}
$\boldsymbol{\vartheta}_{i,m} \oplus \mathbf{r}_{i,m} \oplus \boldsymbol{\eta}_m$	4.27×10^{-4}	2.62×10^{-6}	1.05×10^{-6}	5.64×10^{-7}	6.54×10^{-7}
$\boldsymbol{\vartheta}_{i,m} \oplus \mathbf{r}_{i,m} \oplus \boldsymbol{\zeta}_{i,m}$	3.74×10^{-4}	2.85×10^{-6}	1.08×10^{-6}	5.05×10^{-7}	5.98×10^{-7}
$\boldsymbol{\vartheta}_{i,m} \oplus \mathbf{r}_{i,m} \oplus \boldsymbol{\zeta}_{i,m} \oplus \boldsymbol{\eta}_m$	1.81×10^{-4}	3.12×10^{-6}	1.19×10^{-6}	5.27×10^{-7}	6.55×10^{-7}

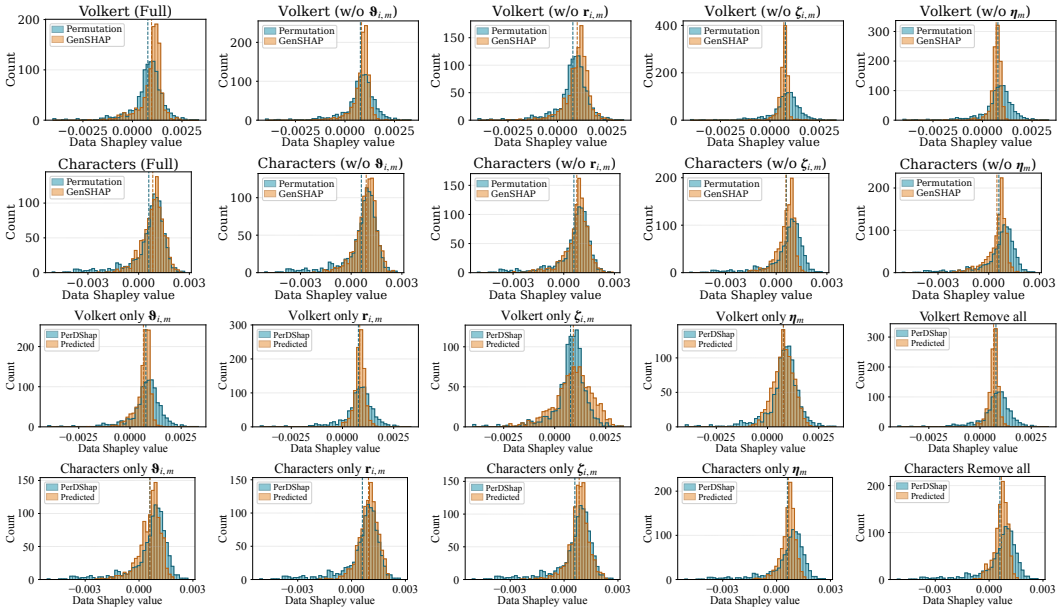


Figure 5: Distribution comparison between Permutation Data Shapley and GenSHAP predictions under auxiliary feature ablations. The first two rows show removal-based ablations on *Volkert* and *Characters*, where the full model is compared with variants that remove gradient statistics $\boldsymbol{\vartheta}_{i,m}$, KNN reliability $\mathbf{r}_{i,m}$, learning dynamics $\boldsymbol{\zeta}_{i,m}$, or metadata $\boldsymbol{\eta}_m$. The last two rows show inclusion-only ablations, where GenSHAP retains only one auxiliary feature group or removes all auxiliary features.

C.3 Impact of density weights in loss function

In addition to the auxiliary feature ablations, we further study the effect of the density-weighted loss used to train GenSHAP. The motivation of this ablation is that data Shapley values are often unevenly distributed. Many samples concentrate in high-density value regions, while a smaller number of samples lie in low-density but important tail regions. A standard MSE objective may therefore be dominated by the most common value ranges and under-emphasise rare but informative Shapley values. The density-weighted objective instead emphasises high-density regions of the Shapley value

distribution, reducing the influence of isolated observations. To verify whether this objective is beneficial, we compare the full GenSHAP model trained with density weights against an otherwise identical model trained without density weights.

As shown in Table 5, density weighting improves performance on four out of five datasets. The gain is especially clear on *Iris* and *Vehicle*, where removing density weights increases the MSE from 1.81×10^{-4} to 2.37×10^{-4}, and from 3.12×10^{-6} to 6.37×10^{-6}, respectively. The density-weighted objective also reduces the error on *STL10* and *Volkert*, suggesting that it improves robustness across heterogeneous datasets. The only exception is *Characters*, where training without density weights gives a slightly smaller MSE. Overall, however, the results suggest that density weighting provides a useful regularising effect and improves cross-dataset calibration of predicted data Shapley values.

Figure 6 further illustrates this behaviour at the distribution level. For both *Characters* and *Volkert*, the density-weighted model better preserves the spread and shape of the Permutation Data Shapley distribution, while the model trained without density weights tends to produce more concentrated or shifted predictions. This is consistent with the emphasis on common valuation patterns while reducing the influence of isolated observations.

Table 5: Ablation study of the density-weighted training objective in GenSHAP. We report the MSE between predicted values and Permutation Data Shapley values.

Training objective	**Iris**	**Vehicle**	**Characters**	**STL10**	**Volkert**
With density weights	1.81×10^{-4}	3.12×10^{-6}	1.19×10^{-6}	5.27×10^{-7}	6.55×10^{-7}
Without density weights	2.37×10^{-4}	6.37×10^{-6}	1.14×10^{-6}	6.42×10^{-7}	6.93×10^{-7}

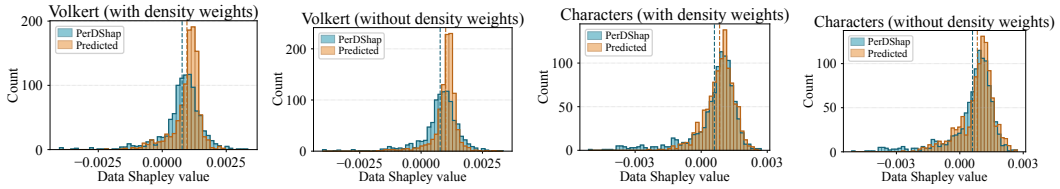


Figure 6: Distribution comparison between high-budget Permutation Data Shapley values and GenSHAP predictions under the density-weighted objective ablation. For each dataset, we compare the full model trained with density weights against the same model trained without density weights.

C.4 Repeated Architecture Ablation

We independently trained the full GenSHAP model and GIN-only, Transformer-only, and MLP-only variants with three random seeds on the same source-disjoint training corpus. All variants use the same data and training protocol, and no held-out result is used for checkpoint selection. Table 6 reports coordinate-wise MSE. No architecture minimizes this metric on every dataset, which is expected because MSE is sensitive to the scale and concentration of each dataset's reference values. In data valuation, however, the operational goal is commonly to identify and rank the most harmful or beneficial samples, rather than to minimize error on the many values concentrated near zero.

We therefore add two ranking-oriented evaluations. Let $B_q(\mathbf{v})$ denote the lowest-valued $q\%$ of samples under valuation vector $\mathbf{v}$. We define

$$\text{BottomRecall@}\,q = \frac{|B_q(\widehat{\boldsymbol{\phi}}) \cap B_q(\boldsymbol{\phi}^{\text{ref}})|}{|B_q(\boldsymbol{\phi}^{\text{ref}})|}. \tag{22}$$

Removal SeparationAUC is the area between the utility curves obtained by removing samples from lowest to highest predicted value and from highest to lowest predicted value. A larger value indicates better separation between harmful and beneficial samples. Table 7 macro-averages these metrics over the five held-out sources and three seeds.

The full hybrid is best on all four aggregate operational metrics. At BottomRecall@10, it improves over GIN-only, Transformer-only, and MLP-only by 20.4%, 19.1%, and 19.6%, respectively. Its Removal SeparationAUC is $2.58\times$, $3.98\times$, and $2.12\times$ that of the corresponding ablations, and it is the best architecture on four of five individual sources for this metric. These results support the hybrid

Table 6: Repeated architecture ablation. Entries are MSE in units of 10^{-6}, reported as mean $\pm$ sample standard deviation over three seeds.

Architecture	Iris	Vehicle	Characters	STL10	Volkert
Full GenSHAP	232.67 ± 66.36	4.49 ± 0.36	1.22 ± 0.18	$\mathbf{0.525 \pm 0.042}$	0.631 ± 0.050
GIN-only	64.52 ± 33.68	3.15 ± 0.34	$\mathbf{0.92 \pm 0.22}$	0.621 ± 0.033	0.72 ± 0.05
Transformer-only	133.57 ± 76.50	$\mathbf{2.87 \pm 0.24}$	1.09 ± 0.11	0.55 ± 0.07	0.65 ± 0.05
MLP-only	$\mathbf{57.17 \pm 16.56}$	3.30 ± 0.15	1.05 ± 0.04	0.53 ± 0.02	$\mathbf{0.57 \pm 0.05}$

Table 7: Aggregate ranking and removal metrics for the repeated architecture ablation. Higher is better for every metric.

Metric	Full GenSHAP	GIN-only	Transformer-only	MLP-only
BottomRecall@5	**0.2767**	0.2510	0.2418	0.2580
BottomRecall@10	**0.3284**	0.2728	0.2758	0.2745
BottomRecall@20	**0.4303**	0.3605	0.4044	0.3843
Removal SeparationAUC	**0.0174**	0.0067	0.0044	0.0082

architecture as the final design: local graph aggregation, global interactions, and auxiliary optimization information are complementary for identifying operationally important samples. Because there are only three seeds, we report sample standard deviations and do not overstate the power of an asymptotic significance test.

D Implementation details

For all model-based baselines, Logistic Regression is used as the base learner, and accuracy serves as the utility function. The Monte-Carlo sampling budget is fixed at $3N$, where N denotes the number of training samples. Hyperparameters are set following standard configurations suggested by their papers: TMC-Shapley (tolerance 0.05, error threshold 0.1), Delta-Shapley (subset sizes being $N/3$), Group-Testing Shapley ($\epsilon = 0.05$), Owen-Shapley (number of inner samples = 2, number of outer samples $Q = \max(2000, 38.4 * N)$, Beta-Shapley ($\alpha = 1, \beta = 16$), and KNN-Shapley ($k = \min(10, N - 1)$).

All baseline computations and preprocessing steps are executed on a remote cluster equipped with dual Intel Xeon E5-2678 v3 CPUs (24 cores, 48 threads) and 251 GiB RAM. Training and inference of GenSHAP framework, along with other GPU-accelerated components, are performed on an NVIDIA GeForce RTX 3090 GPU with 24 GiB VRAM.

D.1 Datasets

For each base-model family, we organize the data as paired balanced train/evaluation subsets. The train side of each pair contains the samples whose high-budget Permutation Data Shapley values are computed, and the matched evaluation side is used to evaluate the utility of candidate subsets. Importantly, the GenSHAP training process is performed only on the datasets labelled as *Training* in Tables 8 and 9. The datasets labelled as *Unseen test* are held out at the dataset level: no samples, labels, graphs, or Shapley values from these datasets are used to optimize GenSHAP parameters. They are used only after training to evaluate cross-dataset generalization on previously unseen datasets. Unless otherwise noted, preprocessing follows the same pipeline used in the calculation of Permutation Data Shapley: standardization, PCA, and a second standardization step; the tables report the observed number of saved principal components in the local paired files.

In total, we construct 100 training pairs for each base-model setting. These pairs are generated by extracting task-specific subsets from the training datasets listed in Tables 8 and 9. We use sampling because computing reference data values on the full datasets is computationally expensive, especially for permutation-based Shapley estimation. Sampling also allows us to form multiple classification tasks from the same source dataset by varying the selected classes and instances, thereby increasing the diversity of training graphs available to GenSHAP.

For most generated training pairs, the subset size is randomly chosen between 800 and 1000 samples whenever the source dataset contains enough examples. This randomization is intentional: instead of training only on fixed-size datasets, GenSHAP observes valuation problems with different sample sizes, class compositions, and data distributions. Such variation helps test and strengthen the model's generalization ability, making the learned predictor less dependent on a particular dataset scale and more robust when transferred to the unseen test datasets.

Table 8: Dataset manifest for the fixed-PCA LR base-model setting. Rows marked *Source-disjoint test* are held out at the source-dataset level: no task, sample, label, graph, reference estimate, or checkpoint from these sources is used for GenSHAP training or model selection. The training corpus contains 100 valuation tasks from 33 source datasets.

Dataset	Use	# subsets	Feature processing
Autos	Training	3	PCA (20 dimensions)
Bank Marketing	Training	3	PCA (20 dimensions)
Breast Cancer	Training	2	PCA (20 dimensions)
Caltech101	Training	3	PCA (20 dimensions)
CIFAR-10	Training	3	PCA (20 dimensions)
CIFAR-100	Training	3	PCA (20 dimensions)
Connect-4	Training	3	PCA (20 dimensions)
Covertype	Training	3	PCA (20 dimensions)
CovPokElec	Training	3	PCA (20 dimensions)
Credit Score	Training	3	PCA (20 dimensions)
Devnagari-Script	Training	3	PCA (20 dimensions)
Dilbert	Training	3	PCA (20 dimensions)
Dry Bean	Training	3	No PCA (16 dimensions)
EEG Eye State	Training	3	No PCA (14 dimensions)
FashionMNIST	Training	3	PCA (20 dimensions)
Gas Drift	Training	3	PCA (20 dimensions)
HAR	Training	3	PCA (20 dimensions)
hls4ml LHC Jets HLF	Training	3	No PCA (16 dimensions)
ImageNet-A	Training	3	PCA (20 dimensions)
Jannis	Training	3	PCA (20 dimensions)
Kuzushiji-MNIST	Training	3	PCA (20 dimensions)
LED(50000)	Training	3	PCA (20 dimensions)
Letter	Training	3	No PCA (16 dimensions)
MNIST	Training	5	PCA (20 dimensions)
Mozilla4	Training	3	No PCA (5 dimensions)
Nomao	Training	3	PCA (20 dimensions)
Pendigits	Training	3	No PCA (16 dimensions)
Pokerhand	Training	3	No PCA (10 dimensions)
RandomRBF 0 0	Training	3	No PCA (10 dimensions)
RT-IoT2022	Training	3	PCA (20 dimensions)
SDSS17	Training	3	PCA (20 dimensions)
Shuttle	Training	3	No PCA (9 dimensions)
Spoken Arabic Digit	Training	3	No PCA (14 dimensions)
Artificial Characters	Source-disjoint test	1	No PCA (7 dimensions)
Iris	Source-disjoint test	1	No PCA (4 dimensions)
STL10	Source-disjoint test	1	PCA (20 dimensions)
Vehicle	Source-disjoint test	1	No PCA (18 dimensions)
Volkert	Source-disjoint test	2	PCA (20 dimensions)

Table 9: Dataset manifest for the SVM base-model setting. This experiment evaluates whether the GenSHAP framework can be retrained for a different base learner and utility. Artificial Characters and Breast Cancer contribute disjoint task subsets to both training and evaluation and are therefore marked as shared-source evaluations; the SVM experiment is not used as evidence for strict source-level generalization. Strict source-level transfer is evaluated by the fixed-PCA LR protocol in Table 8. PCA dimensions in the SVM training corpus are task-specific.

Dataset	Use	# subsets	Feature processing
Artificial Characters	Training	3	No PCA (7 dimensions)
Autos	Training	3	PCA
Bank Marketing	Training	3	PCA
Breast Cancer	Training	1	PCA
Caltech101	Training	3	PCA
CIFAR-10	Training	3	PCA
CIFAR-100	Training	3	PCA
Connect-4	Training	3	PCA
Covertype	Training	3	PCA

Continued on next page

Dataset	Use	# subsets	Feature processing
CovPokElec	Training	3	PCA
Credit Score	Training	3	PCA
Devnagari-Script	Training	3	PCA
Dilbert	Training	3	PCA
Dry Bean	Training	3	No PCA (16 dimensions)
FashionMNIST	Training	3	PCA
Financial Risk	Training	3	PCA
Gas Drift	Training	3	PCA
HAR	Training	3	PCA
hls4ml LHC Jets HLF	Training	3	No PCA (16 dimensions)
ImageNet-A	Training	3	PCA
Jannis	Training	3	PCA
Kuzushiji-MNIST	Training	3	PCA
LED(50000)	Training	3	PCA
Letter	Training	3	No PCA (16 dimensions)
MNIST	Training	3	PCA
Nomao	Training	3	PCA
Pendigits	Training	3	No PCA (16 dimensions)
Pokerhand	Training	3	No PCA (10 dimensions)
RandomRBF 0 0	Training	3	No PCA (10 dimensions)
RT-IoT2022	Training	3	PCA
SDSS17	Training	3	PCA
Shuttle	Training	3	No PCA (9 dimensions)
Spoken Arabic Digit	Training	3	No PCA (14 dimensions)
WBCAtt	Training	3	PCA
Artificial Characters	Evaluation (shared source)	1	No PCA (7 dimensions)
Breast Cancer	Evaluation (shared source)	2	PCA
Iris	Evaluation (source-disjoint)	1	No PCA (4 dimensions)
STL10	Evaluation (source-disjoint)	3	PCA
Vehicle	Evaluation (source-disjoint)	1	No PCA (18 dimensions)
Volkert	Evaluation (source-disjoint)	3	PCA

E Additional Robustness and Efficiency Analyses

E.1 Interpretability

To qualitatively verify whether the learned valuation model captures meaningful sample-level utility, we examine images from the extremes of the predicted value on *STL10*. As expected, high-value samples should be visually representative, while low-value samples should be ambiguous or atypical.

Figure 7 confirms this pattern. The two highest-value images feature prominent, easily recognisable subjects (horse) clearly separated from their backgrounds. Conversely, the lowest-value images are uninformative, suffering from non-frontal viewpoints, low contrast, and complex backgrounds. Notably, one low-value sample is labelled as a 'cat' but visually resembles a tiger or a wildcat, highlighting the model's ability to penalise confusing or potentially mislabelled data. This demonstrates that our framework successfully aligns with the intuitive semantics of Data Shapley, correctly assigning higher values to clean, informative samples and lower values to noisy ones.



Figure 7: *STL10* [33], an image classification dataset containing 10 object categories, including animals and vehicles. We visualise images with the highest and lowest predicted data Shapley values. The ground-truth label for these four figures (from left to right) are: dog, cat, horse, horse. High-value samples show clear, representative horse images with salient object structure, while low-value samples are less recognisable with non-frontal viewpoints or complex backgrounds.

Table 10: MSE between predicted data Shapley values and Permutation Data Shapley on datasets generated with random dimensions of PCA.

Method	Iris_1	Vehicle_01_1	characters_04	STL10_3568x	volkert_15_2
GenSHAP	3.02×10^{-5}	3.81×10^{-6}	1.13×10^{-6}	5.64×10^{-7}	6.20×10^{-7}
GenSHAP-S	1.75×10^{-4}	1.52×10^{-6}	5.98×10^{-7}	3.24×10^{-7}	3.33×10^{-7}

E.2 Effect of PCA Dimensionality on Cross-Dataset Generalisation

In the previous experiments mentioned in main text, all PCA-transformed datasets used a fixed 20-dimensional PCA representation. Here, we consider a more challenging random PCA setting, where the number of principal components is sampled randomly from 5 to 100 for each dataset. This setting tests whether GenSHAP and GenSHAP-S remain reliable when the downstream valuation model is exposed to PCA embeddings with varying dimensionality, rather than a fixed representation size.

Data removal and addition plots Again, we use Permutation Data Shapley as the reference valuation method. The removal and addition plots compare four ranking strategies: random ordering, Permutation Data Shapley, GenSHAP, and GenSHAP-S. Compared with the fixed-PCA setting, random-PCA setting introduces additional variation in the feature representation. As a result, the ranking quality of GenSHAP can deteriorate slightly. As shown in Figure 8, small ranking errors may accumulate as more training samples are removed or added. Nevertheless, the curves of GenSHAP-S still track the performance of Permutation Data Shapley well.

Consistency Table 10 shows that both GenSHAP and GenSHAP-S achieve low MSE under the random-PCA setting, despite the added variability from using a random number of PCA components. GenSHAP-S obtains lower MSE on all test sets except *Iris*. Overall, these results indicate that although random-PCA setting can slightly weaken the ranking behaviour observed in removal and addition plots, the predicted data Shapley values remain close to Permutation Data Shapley in absolute error.

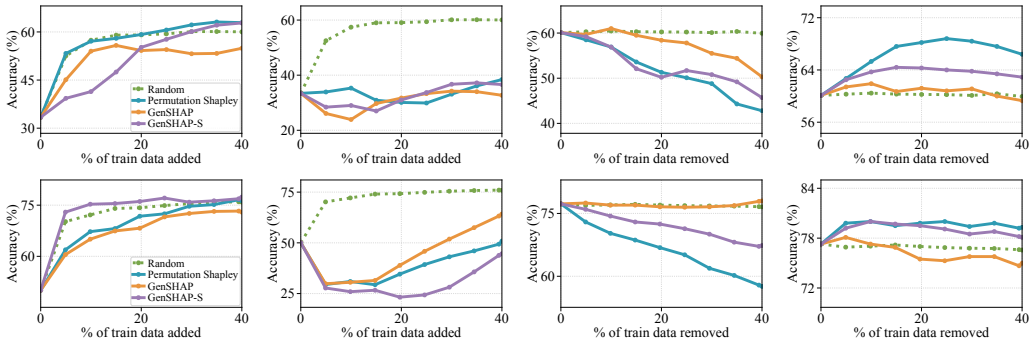


Figure 8: Data addition and removal evaluation under random-PCA setting on: *Characters* [24] (multi-class classification) and *Volkert* [32](binary classification). Each dataset contains around 1000 training points and is first transformed by PCA with the number of principal components randomly sampled from 5 to 100, rather than fixed at 20. For each dataset, we compare performance curves when adding or removing samples based on their estimated values.

E.3 Experiments with SVM as the base model

To further examine whether GenSHAP is tied to a specific algorithm, we conduct additional experiments using Support Vector Machine (SVM) as the base learning algorithm. Logistic Regression and SVM represent substantially different model classes. Logistic Regression is a linear probabilistic model governed by cross-entropy loss and class-probability estimates, whereas SVM is a margin-based classifier. With the RBF kernel used in our experiments, SVM can induce nonlinear decision

boundaries, and its utility is shaped by support vectors and margin violations rather than calibrated probabilities. Therefore, SVM provides a meaningful test case for evaluating whether GenSHAP can generalise beyond the linear probabilistic setting.

For the SVM setting, we recompute the Permutation Data Shapley values using an RBF-kernel SVM as the utility model. Accordingly, when constructing auxiliary features, we replace Logistic-Regression-based probability and loss statistics with SVM margin- and hinge-loss-based signals. The resulting SVM-based GenSHAP model is then evaluated on held-out evaluation tasks using the same data addition and removal protocols as in the main experiments.

Data removal and addition plots The addition and removal plots are generated for each dataset under four settings: adding high-valued samples first, adding low-valued samples first, removing high-valued samples first, and removing low-valued samples first. The evaluated datasets include *STL10* and *Volkert*. As suggested by Figure 9, the SVM-based GenSHAP preserves meaningful ranking behaviour under a non-linear margin-based utility model. On *STL10*, removing samples ranked highly by Permutation Shapley causes the SVM accuracy to drop sharply from 71.6% to 46.0% after removing 40% of the data, while GenSHAP also produces a clear degradation to 57.1%. In the corresponding addition experiment, adding high-valued samples first yields substantially faster accuracy improvement than adding low-valued samples first. For *Volkert*, high-valued samples selected by GenSHAP quickly improve accuracy and approach the Permutation Shapley trend in the addition plots, whereas low-valued samples lead to much slower gains, confirming that the learned values capture useful ranking information under the SVM base model.

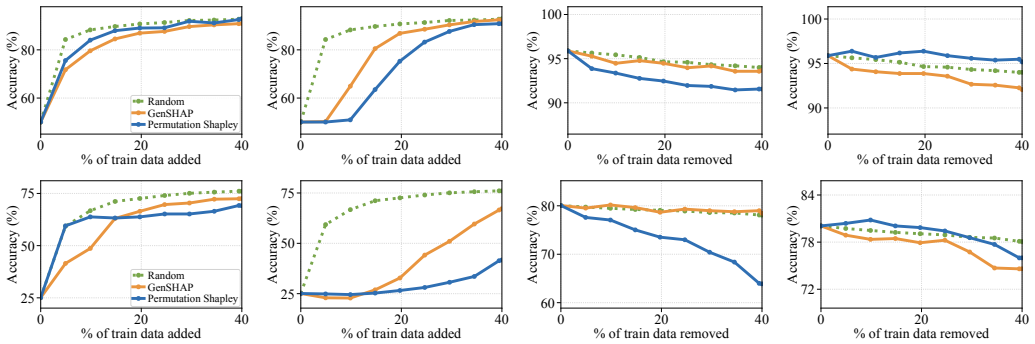


Figure 9: Data addition and removal evaluation on *STL10*(row-1, binary classification) and *Volkert* [32](row-2, multi-class classification) with SVM as the base models.

Interpretability We further conduct an interpretability experiment on *STL10* using GenSHAP trained and evaluated with SVM as the base model. After predicting SVM-based data Shapley values for the sampled training images, we visualise examples with the lowest and highest predicted values. As shown by Figure 10, the low-valued examples include a bird image and an airplane image whose visual structures are relatively similar: both contain a compact foreground object against a sky-like background. Such ambiguity makes these samples less informative for an RBF-kernel SVM, since they may lie close to non-linear decision boundaries or behave like margin-violating points rather than clean representatives of their classes. In contrast, the high-valued truck examples show clearer object identity, stronger foreground structure, and more class-specific visual cues. These qualitative results suggest that SVM based GenSHAP assigns lower values to ambiguous or cross-class-similar samples and higher values to visually representative samples that better support the non-linear margin-based classifier.

Runtime and consistency Table 11 compares the computational cost of different valuation methods under the SVM base-model setting. Classical sampling-based methods remain expensive, since they require repeatedly training SVMs on many sampled coalitions. Among the baselines, Owen Shapley incurs the highest cost, taking more than 4000 seconds on *Volkert*, while TMC Shapley also requires hundreds to over one thousand seconds on larger datasets. By contrast, GenSHAP completes valuation in less than 0.5 seconds on all five datasets. This efficiency is achieved by amortising the valuation process into a trained neural predictor, so that each unseen dataset only requires representation construction followed by a single forward pass.



Figure 10: *STL10*, an image classification dataset containing 10 object categories, including animals and vehicles. We visualise images with the highest and lowest predicted data Shapley values. The ground-truth label for these four figures (from left to right) are: bird, airplane, truck, truck . High-value samples show clear, representative truck images with salient object structure, while low-value samples are less recognisable with non-frontal viewpoints or complex backgrounds.

Consistency with high-budget Permutation Data Shapley under the same SVM utility model is reported in Table 12. Owen Shapley attains the lowest MSE overall, which is consistent with its large sampling budget, and TMC Shapley also remains highly accurate. GenSHAP achieves small absolute MSE values across all datasets and is substantially more accurate than Group-Testing Shapley. Notably, the SVM training tasks contain approximately 1,000 points, whereas the held-out evaluation tasks contain at most 200 points. These results support GenSHAP's portability to an RBF-SVM utility, transfer to held-out tasks, and robustness to a substantial dataset-size shift. Strict source-level generalisation is evaluated separately under the fixed-PCA LR protocol in Table 8.

Table 11: Runtime comparison (in seconds) under the SVM base-model setting.

Method	**Iris**	**Vehicle**	**Characters**	**Breast Cancer**	**Volkert**
TMC Shapley	78.24	626.47	1131.21	538.52	1212.46
Delta Shapley	21.84	249.88	294.85	220.50	306.65
Group-Testing Shapley	5.42	42.10	41.71	40.20	42.71
Owen Shapley	251.66	3245.11	3878.26	2801.88	4067.36
Beta Shapley	10.52	52.73	62.97	46.40	66.16
Data Banzhaf	10.67	53.52	64.17	46.09	65.95
GenSHAP	**0.12**	**0.32**	**0.37**	**0.49**	**0.38**

Table 12: Consistency evaluation under the SVM base-model setting. We report the MSE between each method and the high-budget Permutation Data Shapley reference values.

Method	**Iris**	**Vehicle**	**Characters**	**Breast Cancer**	**Volkert**
TMC Shapley	2.82×10^{-5}	2.08×10^{-6}	7.20×10^{-7}	7.19×10^{-6}	7.90×10^{-7}
Delta Shapley	1.40×10^{-4}	7.76×10^{-6}	1.49×10^{-6}	1.71×10^{-5}	1.18×10^{-6}
Group-Testing Shapley	2.70×10^{-2}	6.48×10^{-3}	2.73×10^{-3}	1.52×10^{-2}	1.85×10^{-3}
Owen Shapley	3.11×10^{-6}	2.08×10^{-6}	7.00×10^{-8}	7.10×10^{-7}	7.00×10^{-8}
GenSHAP	4.00×10^{-5}	8.13×10^{-6}	1.28×10^{-6}	1.24×10^{-5}	1.26×10^{-6}

E.4 Repeated Evaluation on the Main Source-Disjoint Split

To quantify optimization variability beyond the representative run in the main text, we retrained GenSHAP from scratch with three random seeds (23, 42, and 2026) on the same fixed-PCA corpus. The five evaluation sources are entirely absent from training, and no held-out result is used for checkpoint selection. Table 13 reports the original baseline results together with the three-run mean and sample standard deviation for GenSHAP. Values are MSE in units of 10^{-5}.

High-budget samplers can achieve lower coordinate-wise MSE by repeatedly evaluating the base learner on the target dataset. GenSHAP instead amortizes this computation across datasets and requires no target-specific coalition sampling. Among learned estimators, GenSHAP outperforms

Table 13: MSE on the main source-disjoint LR evaluation. GenSHAP is reported as mean ± sample standard deviation over three independent training runs; the baseline rows are the original target-specific or learned-estimator results.

Method	Iris	Vehicle	Characters	STL10	Volkert
TMC-Shapley	5.100	0.095	0.041	0.022	0.038
Delta-Shapley	13.700	0.408	0.083	0.045	0.081
Group-Testing	3945.400	556.213	185.878	121.310	306.043
Owen-Shapley	0.300	0.013	0.004	0.002	0.007
MLPbV	12.900	12.055	8.788	8.770	6.617
GenSHAP (3 runs)	23.267 ± 6.636	0.449 ± 0.036	0.122 ± 0.018	0.053 ± 0.004	0.063 ± 0.005

Table 14: Additional source-disjoint partitions. Macro MSE is averaged over the five test tasks in each partition.

Partition	Fully isolated test sources	GenSHAP macro MSE	MLPbV macro MSE	GenSHAP wins
1	Autos, CIFAR-100, hls4ml, ImageNet-A, Iris	2.333×10^{-5}	7.454×10^{-5}	5/5
2	Bank Marketing, Covertype, Gas Drift, HAR, Kuzushiji-MNIST	3.792×10^{-7}	6.686×10^{-5}	5/5
3	Caltech101, EEG Eye State, Mozilla4, RT-IoT2022, Volkert	7.465×10^{-7}	8.800×10^{-5}	5/5
All	**15 distinct held-out sources**	$\mathbf{8.151 \times 10^{-6}}$	$\mathbf{7.647 \times 10^{-5}}$	**15/15**

MLPbV on four of the five held-out sources. We therefore interpret the results as an accuracy–efficiency trade-off rather than claiming universal MSE superiority over target-specific, high-budget estimators.

E.5 Sensitivity to the Choice of Held-Out Sources

We further constructed three additional partitions to test whether the result depends on a particular choice of held-out source datasets. Each partition contains 80 training tasks and five test tasks from five fully isolated sources. Every task derived from a test source is removed from that partition's training corpus; the 15 held-out sources are distinct across the three partitions. These experiments evaluate split sensitivity rather than seed variability and use one training run per partition. MLPbV is included as the closest amortized learning-based baseline that could be retrained on all partitions under the same protocol.

Across all 15 source-disjoint evaluations, GenSHAP obtains a $9.38\times$ lower task-macro MSE than MLPbV and wins every task-level comparison. Together with Table 13, this supports transfer to previously unseen source datasets within the evaluated classification, preprocessing, dataset-size, and LR-utility regime.

E.6 Monte Carlo Uncertainty of the Reference Estimates

GenSHAP is supervised by finite-budget Permutation Data Shapley estimates, not exact Shapley values. To quantify the associated Monte Carlo uncertainty, we saved all $R = 5000$ untruncated marginal contributions for every point and estimated the average variance of the per-point reference mean as

$$\widehat{\mathcal{E}}_{\mathrm{MC}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\widehat{\mathrm{Var}}(\Delta_i)}{R}, \qquad R = 5000. \tag{23}$$

We evaluate Iris and STL10, which represent the small/no-PCA and image-derived/PCA regimes. We additionally use 2000 block-bootstrap replicates, two disjoint 2500-permutation halves, and an independently generated 5000-permutation reference. The analytical, bootstrap, split-half, and independent-reference estimates agree closely.

Reference uncertainty explains only 0.51% and 3.77% of GenSHAP's MSE on Iris and STL10, respectively. GenSHAP remains 26.55–$197.08\times$ above the estimated reference-noise floor; its error therefore cannot be explained by fitting or reaching the Monte Carlo noise level. Throughout the paper, we accordingly call the targets *finite-budget, 5000-permutation reference estimates* rather than exact ground truth.

Table 15: Estimated uncertainty of the 5000-permutation supervision. GenSHAP MSE is the mean over three independent training runs. The noise share is the analytical floor divided by GenSHAP MSE.

Dataset	Analytical floor	Bootstrap floor [95% interval]	GenSHAP MSE	MSE/floor	Noise share
Iris	1.181×10^{-6}	1.155×10^{-6} $[1.116, 1.195] \times 10^{-6}$	2.327×10^{-4}	197.08	0.51%
STL10	1.977×10^{-8}	1.974×10^{-8} $[1.943, 1.986] \times 10^{-8}$	5.250×10^{-7}	26.55	3.77%

Table 16: CPU-only end-to-end online runtime on matched hardware.

Dataset	Preprocessing + representation (s)	Neural forward (s)	Total (s)
Iris	0.140 ± 0.023	0.019 ± 0.003	0.159 ± 0.023
Vehicle	0.523 ± 0.006	0.050 ± 0.001	0.573 ± 0.006
Characters	1.101 ± 0.055	0.173 ± 0.020	1.274 ± 0.060
STL10	1.136 ± 0.019	0.175 ± 0.010	1.311 ± 0.025
Volkert	1.246 ± 0.036	0.181 ± 0.010	1.427 ± 0.043

E.7 Matched-Hardware End-to-End Online Runtime

We measure the complete target-side GenSHAP pipeline on the same dual-Xeon E5-2678 v3 server used for the classical baselines, with GPU access disabled. Timing begins before data loading and includes pairwise distances, KNN features, one full-data LR fit, gradient statistics, 21 learning-dynamics LR fits, metadata, five graph/SVD constructions, and CPU neural prediction. We perform one warm-up followed by ten timed runs and report mean $\pm$ sample standard deviation.

Thus, the complete CPU-only online time is 0.159–1.427 seconds; the optimization features and graph/SVD construction are included rather than omitted. On these datasets, complete GenSHAP is $38\times$ faster than Group-Testing on Iris and 636–$4994\times$ faster on the other four datasets. We use *single neural forward pass* only for the neural stage and *online end-to-end time* for the complete target pipeline.

E.8 Offline Cost and Amortization Break-Even

GenSHAP shifts computation from repeated target-specific coalition evaluation to a one-time offline stage. Table 17 accounts for this stage explicitly. The reference-generation value is a conservative serial-equivalent range because historical per-task timing logs were not retained. Once the 5000-permutation reference labels are available, deterministic preprocessing and neural optimization together require 8831.711 seconds, or 2.453 hours.

Let C_{off} be the relevant offline cost, $\overline{T}_b$ the comparator's mean per-target runtime, and $\overline{T}_G = 0.949$ seconds the mean GenSHAP online time. The number of future target datasets required to amortize the offline stage is

$$K_b^* = \left\lceil \frac{C_{\text{off}}}{\overline{T}_b - \overline{T}_G} \right\rceil. \tag{24}$$

Table 18 distinguishes three deployment settings. From scratch includes reference generation, preprocessing, and optimization; labels available amortizes only the measured 2.453-hour post-label cost; and a released checkpoint incurs no offline cost for the downstream user.

The from-scratch break-even is 155–232 future tasks against TMC-Shapley, 192–286 against Group-Testing, and 11–15 against Owen-Shapley. When labels already exist, the corresponding counts are 2, 3, and 1. A user starting from a released checkpoint saves computation from the first target dataset. We release trained checkpoints and preprocessing code in the accompanying repository.

E.9 GenSHAP-S under Matched Sampling Budgets

GenSHAP-S uses 100 sampled coalitions per valued point and evaluates two utilities for each coalition, corresponding to at most $200N$ target-side utility fits. We compare it with TMC-Shapley using 200 permutations and Owen-Shapley using 100 paired marginal samples per point, giving

Table 17: One-time cost for 100 historical tasks containing 95,107 valued points.

One-time component	Cost
Reference generation (5000 permutations)	approximately 200–300 serial-equivalent h
Deterministic training preprocessing	129.252 s (0.036 h)
Neural optimization	8702.459 s (2.417 h)
Complete from-scratch cost	**approximately 202.453–302.453 h**

Table 18: Amortization break-even in number of future target datasets.

Comparator	Mean online time (s)	From scratch	Labels available
TMC-Shapley	4705.994	155–232	2
Group-Testing	3815.232	192–286	3
Owen-Shapley	72735.744	11–15	1

the same nominal $200N$ utility-fit budget. This is conservative for GenSHAP-S because truncation allows TMC-Shapley to use only 35.49–51.78% of its nominal fits, while GenSHAP-S uses its probes to provide point-specific, utility-aware features.

At the matched nominal budget, GenSHAP-S has lower MSE than TMC-Shapley on three of five sources and lower MSE than Owen-Shapley on two of five, including the best results on Characters and Volkert. Relative to zero-probe GenSHAP, the probes reduce MSE on all five sources by 54.2%, 29.2%, 65.5%, 37.7%, and 43.9%, respectively, for a 53.8% macro reduction. Thus, GenSHAP and GenSHAP-S occupy distinct operating points: the former prioritizes sub-second amortized valuation, while the latter spends an explicit and controllable target-side budget to increase fidelity.

E.10 Scaling Beyond the Main Experimental Regime

We benchmark the complete pipeline on STL10-derived tasks containing 250 to 3000 valued points while holding the 20-dimensional feature representation and four-class structure fixed. Larger tasks use class-balanced resampling with small deterministic feature jitter. This controlled experiment isolates computational scaling and is not presented as a new accuracy evaluation.

The exact-SVD implementation processes 3000 points in 30.519 ± 0.062 seconds, with peak CPU and GPU memory of 1.633 and 1.176 GiB, respectively. The empirical end-to-end scaling exponent over 250–3000 points is 2.00 (and 2.68 over 1000–3000 points), although the dense implementation retains the conservative worst-case complexity of $O(BN^3)$ time and $O(BN^2)$ memory for B graph scales. We clarify that the lower complexity in the theorem applies when a truncated decomposition is used; the reference implementation evaluated in the main experiments uses full SVD.

As a diagnostic, rank-100 randomized SVD reduces the time for the five decompositions from 84.279 to 1.711 seconds ($49.3\times$) in a matched large-N experiment. Even when inserted into a frozen exact-SVD checkpoint without retraining, the resulting predictions retain Spearman correlation 0.930, Pearson correlation 0.949, top-decile overlap 0.777, and bottom-decile overlap 0.850 with the exact-SVD predictions. This deliberately stringent zero-shot test shows that low-rank decomposition preserves most of the valuation signal. Representation-consistent retraining, sparse graphs, and Nyström approximations provide direct paths to substantially larger datasets.

F Limitations and Broader Impact

While GenSHAP demonstrates strong generalisability and computational efficiency, several limitations remain. First, our framework relies on Permutation Shapley as the supervision signal during training. As a result, the accuracy of GenSHAP is inherently bounded by the fidelity of this approximation and cannot correct systematic bias or variance inherited from the target values. Second, the main computational bottleneck of our preprocessing pipelines lies in the full Singular Value Decomposition (SVD) applied to the scale-specific adjacency matrices, which scales as $O(BN^3)$. Although this represents a substantial improvement over the exponential complexity of exact Data Shapley computation, it may still limit scalability for very large datasets without additional approx-

Table 19: Matched-budget comparison for GenSHAP-S. MSE is measured against the 5000-permutation reference. Runtime is averaged over the five LR test sources.

Method	Mean runtime (s)	Iris	Vehicle	Characters	STL10	Volkert
GenSHAP	0.949	1.8117×10^{-4}	3.12×10^{-6}	1.19×10^{-6}	5.3×10^{-7}	6.6×10^{-7}
TMC-Shapley (200 permutations)	347.561	3.295×10^{-5}	1.62×10^{-6}	9.5×10^{-7}	5.4×10^{-7}	1.76×10^{-6}
Owen-Shapley (100 paired samples)	861.638	5.695×10^{-5}	1.47×10^{-6}	4.5×10^{-7}	2.9×10^{-7}	5.3×10^{-7}
GenSHAP-S (100 probes)	643.886	8.299×10^{-5}	2.21×10^{-6}	$\mathbf{4.1 \times 10^{-7}}$	3.3×10^{-7}	$\mathbf{3.7 \times 10^{-7}}$

Table 20: Scaling of the dense, full-SVD implementation. Total time is mean $\pm$ sample standard deviation.

N	Five full SVDs (s)	Online end-to-end total (s)
250	0.059	0.214 ± 0.002
500	0.229	0.436 ± 0.003
1000	1.271	1.625 ± 0.010
1500	3.697	4.308 ± 0.015
2000	8.304	9.249 ± 0.029
3000	28.504	30.519 ± 0.062

imation techniques such as randomised SVD. Third, while GenSHAP-S improves approximation fidelity by injecting utility-aware marginal probes, its sampling design is currently heuristic: both the sampling budget P and the subset size distribution $k_{i,m}$ are selected empirically. A more principled or adaptive sampling strategy may further improve the trade-off between predictive accuracy and preprocessing cost. Finally, our current framework is evaluated mainly in supervised classification settings with accuracy as the utility functions and permutation-based Shapley supervision. Extending the framework to broader learning scenarios, such as regression, unsupervised learning, or alternative utility definitions, remains an important direction for future work.

Beyond these technical constraints, the deployment of scalable data valuation carries significant societal implications. On a positive note, GenSHAP lowers the computational barrier for equitable data valuation, democratising access to fair data attribution mechanisms. By reducing valuation latency from hours to seconds, it enables practical, large-scale data auditing. This can profoundly enhance the safety and reliability of AI systems by allowing practitioners to efficiently identify and remove noisy, corrupted, or malicious training samples. Furthermore, scalable valuation facilitates fair compensation models for data creators, fostering a more transparent and equitable data economy. Additionally, it is crucial to recognise that while these methods calculate fair compensation based on marginal utility, they inherently cannot evaluate or offset the intangible risks and inherent rights associated with personal data privacy. However, practitioners must apply these tools thoughtfully. If the underlying model utility metrics inherently favour majority classes, the valuation framework may systematically assign lower values to rare but crucial data from under-represented groups. Blindly relying on these scores for data pruning could inadvertently exacerbate algorithmic biases, highlighting the critical need to pair efficient valuation frameworks with utility metrics that explicitly account for fairness and minority representation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction are supported by the experiment results we provided in main text and appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In appendix F, we have discussed the limitations of our work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Most theory results are put in the main text while the assumptions and proof of them are clearly stated in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the repository to reproduce the experiments results listed in our paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets are publicly available, and we release the implementation, preprocessing code, and experiment scripts in the accompanying repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: In appendix D, we provided detailed description of experimental settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report mean and sample standard deviation over three independent training runs for the main evaluation and architecture ablations. We also report repeated runtime measurements and Monte Carlo reference uncertainty in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided sufficient information on computer resources in appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirmed that the code of ethics are strictly followed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The article has discussed the negative and positive social impact in appendix F.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have provided a detailed reference list.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this article does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.