DAVINZ: Data Valuation using Deep Neural Networks at Initialization

Zhaoxuan Wu [1 2] **Yao Shu** [3] **Bryan Kian Hsiang Low** [3]

Abstract

Recent years have witnessed a surge of interest in developing trustworthy methods to evaluate the value of data in many real-world applications (e.g., collaborative machine learning, data marketplaces). Existing data valuation methods typically valuate data using the generalization performance of converged machine learning models after their *long-term* model training, hence making data valuation on large complex *deep neural networks* (DNNs) unaffordable. To this end, we theoretically derive a *domain-aware generalization bound* to estimate the generalization performance of DNNs without model training. We then exploit this theoretically derived generalization bound to develop a novel *training-free* data valuation method named *data valuation at initialization* (DAVINZ) on DNNs, which consistently achieves remarkable effectiveness and efficiency in practice. Moreover, our training-free DAVINZ, surprisingly, can even theoretically and empirically enjoy the desirable properties that training-based data valuation methods usually attain, thus making it more trustworthy in practice.

1. Introduction

Data has been widely recognized as one of the most vital ingredients of learning high-performing *machine learning* (ML) models. Meanwhile, data with different qualities typically lead to diverse model performances in practice. Developing trustworthy data valuation methods that are explainable, fair, and robust is therefore extensively required to measure the value of data and also decide how to use them in real-world applications. For example, data valuation is essential to compensating the participants in data collection (Lo & DeMets, 2016), designing fair rewards in collaborative ML (Sim et al., 2020), developing a trustworthy data market for buyers and sellers (Agarwal et al., 2019; Han et al., 2021), among others. To meet these demands, a number of data valuation methods have been proposed (Ghorbani & Zou, 2019; Koh & Liang, 2017).

Unfortunately, it is prohibitively costly to deploy these conventional data valuation methods in real-world applications using large complex models. For example, both data *Shapley value* (SV) (Ghorbani & Zou, 2019) and *leave-one-out* (LOO) (Cook, 1977; Koh & Liang, 2017) are calculated using the validation performances of converged models. However, obtaining fully converged models is computationally costly for large complex models (e.g., *deep neural networks* (DNNs)) due to their inevitable long-term model training. As a result, developing efficient techniques to estimate the fully converged performances of large complex models is essential to making data valuation more applicable in practice. To the best of our knowledge, only a few efforts have been devoted to this direction (Jia et al., 2019a; Xu et al., 2021b). Nevertheless, these works impose strict restrictions on the choice of ML models or may introduce bias into data valuation when evaluating the value of learned data embedding rather than the value of original data.

While *statistical learning theory* (SLT) makes it possible to estimate the fully converged performances of DNNs without model training (Arora et al., 2019b; Cao & Gu, 2019), these works typically assume that the training and the validation datasets follow the same underlying distribution. Nonetheless, this assumption does not necessarily hold in the realm of data valuation. For instance, in the case of collaborative disease diagnosis among hospitals, data contributors (e.g., children's hospitals) usually collect data without any knowledge about the validation dataset (e.g., a dataset including the disease data across all age groups). In data markets, it is also difficult for data consumers to purchase datasets that can perfectly align with their validation tasks. As a result, the discrepancy between the training and validation datasets (i.e., domain discrepancy) also needs to be considered when applying SLT to estimate the performances of DNNs.

To this end, we theoretically derive a *domain-aware* gener-

[1]Institute of Data Science, National University of Singapore, Republic of Singapore [2]Integrative Sciences and Engineering Programme, NUSGS, Republic of Singapore [3]Department of Computer Science, National University of Singapore, Republic of Singapore. Correspondence to: Bryan Kian Hsiang Low <lowkh@comp.nus.edu.sg>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

alization bound to estimate the performances of DNNs in a principled way. Utilizing this bound, we develop a novel data valuation method named *data valuation at initialization* (DAVINZ) that completely avoids model training in data valuation. Specifically, our domain-aware generalization bound is derived by introducing domain discrepancy into the recent *neural tangent kernel* (NTK) theory (Jacot et al., 2018) (Sec. 4.1). While conventional data valuation methods typically use validation performance as their scoring function, in DAVINZ, we novelly employ our theoretically derived domain-aware generalization bound (i.e., an estimate of validation performance) as the scoring function (Sec. 4.2), followed by the widely adopted data valuation methods (e.g., LOO) as the valuation function (Sec. 4.3). Our training-free DAVINZ, surprisingly, is able to theoretically enjoy the properties that a trustworthy data valuation method should attain, such as the awareness of data preference, the awareness of data quantity, the stability to noise, and the robustness to model (Sec. 5). Finally, we perform extensive comparisons with both training-based and training-free data valuation baselines to justify the effectiveness and efficiency of our DAVINZ as well as the desirable properties it enjoys (Sec. 6).

2. Related Work

Valuation in latent space. Jia et al. (2019a) have proposed to use *K-nearest-neighbor* (KNN) models to estimate the exact SV of a KNN-specific performance metric in linear time. Though this method is shown to be efficient, it can only evaluate the value of *learned data embedding* in a latent space rather than the value of original data (Ghorbani et al., 2021), which may introduce the bias in the latent space into data valuation and thus would reduce the reliability of this method. In contrast, our data valuation method in this paper assesses the value of the *original dataset directly* while achieving improved effectiveness and efficiency.

Influence function. An *influence function* (IF), which is a method to estimate the variation of model performances when datasets are removed from the training set (Koh et al., 2019), has been considered by Jia et al. (2019b) to approximate the marginal contributions of datasets. Unfortunately, IF is only guaranteed to perform well on strongly convex and twice-differentiable models. In practice (e.g., in widely applied DNNs), these requirements are often violated. As a result, IF typically suffers a drastic performance degradation when it is applied to deep non-convex models (Basu et al., 2021). On the contrary, our method in this paper is able to achieve both effective and efficient data valuation on *complex DNNs*.

Volume-based valuation. Recently, Xu et al. (2021b) have proposed to use *robust volume* (RV) as a measure of dataset diversity to quantify data value. Though this method evaluates data value in a training-free manner, it not only suffers from exploding volumes in high-dimensional inputs but also entirely ignores the useful information in the validation dataset. In practice, data consumers usually have their preferences for datasets; for example, they prefer a dataset that is able to achieve better performance (measured on validation dataset) in their tasks. It is therefore more reasonable to correlate data value with validation performance (Ghorbani & Zou, 2019; Jia et al., 2019b), as followed by our *validation-based* training-free data valuation method.

3. Backgrounds and Notations

3.1. Data Valuation

This paper focuses on the data valuation problem in a supervised collaborative ML setting where multiple data contributors contribute their datasets to learn a single predictive model f from a predefined hypothesis set $\mathcal{F}$. We denote $S_{\mathcal{A}} = \{S_i\}_{i=1}^{K}$ as an aggregated dataset from K contributors where S_i denotes the dataset from contributor i. To measure the contribution (i.e., value) of different datasets to the final predictive function f, a scoring function $\nu : \mathcal{P}(S_{\mathcal{A}}) \to \mathbb{R}$ and a valuation function $\phi(S, S_{\mathcal{A}}, \nu)$ are conventionally defined where $\mathcal{P}(S_{\mathcal{A}})$ denotes the power set of $S_{\mathcal{A}}$. In practice, the validation performance (on the validation set T) of a predictive model f trained on dataset S is usually employed as the scoring function, while LOO and SV are adopted as the valuation function (Ghorbani & Zou, 2019; Koh et al., 2019).

In the literature, there are some empirically validated properties in existing data valuations (Agussurja et al., 2022; Ghorbani et al., 2020; Sim et al., 2022; Tay et al., 2022; Wang et al., 2021b; Xu et al., 2021a;b), which are shown to be essential to making data valuation methods more precise and practical. We summarize them below:

(i) *Awareness of Data Preference*: As outlined by IMDA (2019), the value of data should mainly depend on its usefulness in attaining the purpose of the data consumer. Datasets sharing different similarities to the validation dataset (i.e., a preferred dataset of the data consumer) hence should obtain distinguishable values.

(ii) *Awareness of Data Quantity*: Without considering any abnormal data (e.g., adversarial examples (Goodfellow et al., 2015)), ML models obtained using more data will typically achieve better performance in practice. Hence, datasets of varying sample quantities should enjoy different values in data valuation.

(iii) *Stability to Noise*: Random noise is commonly used to reduce overfitting in ML models (Bishop, 1995), which typically leads to their stable model performances, even in the presence of *small-scale* noises in a dataset. Data valuation using these ML models should thus pro-

duce stable values for datasets with small-scale noises.

(iv) *Robustness to Model*: As we have justified above, the value of data should mainly rely on its functionality in realizing the purpose of the data consumer. In practice, such a purpose sometimes can be irrelevant to the choice of ML models and instead data-driven (Sim et al., 2022). In this case, the value of datasets should be generally admitted by different ML models.

3.2. Neural Tangent Kernel

Following (Jacot et al., 2018), let $g^{(l)}(\boldsymbol{x},\boldsymbol{\theta})$ and $\tilde{g}^{(l)}(\boldsymbol{x},\boldsymbol{\theta})$ denote the n_l-dimensional pre-activations and activations of the l-th layer in an L-layer DNN model, respectively. Let $\sigma(\cdot)$ denote an activation function and $\mathbf{W}^{(l)} \in \mathbb{R}^{n_{l+1}\times n_l}$ denote the parameters of the l-th layer. Then, the L-layer DNN model can be represented recursively as

$$\begin{aligned} g^{(l+1)}(\boldsymbol{x},\boldsymbol{\theta}) &= \sqrt{1/n_l}\mathbf{W}^{(l)}\tilde{g}^{(l)}(\boldsymbol{x},\boldsymbol{\theta}) \\ \tilde{g}^{(l+1)}(\boldsymbol{x},\boldsymbol{\theta}) &= \sigma(g^{(l+1)}(\boldsymbol{x},\boldsymbol{\theta})) \end{aligned} \tag{1}$$

for $l = 0,\ldots,L-1$ where $\tilde{g}^{(0)}(\boldsymbol{x},\boldsymbol{\theta}) = \boldsymbol{x}$, $n_L = 1$, and $f(\boldsymbol{x},\boldsymbol{\theta}) = g^{(L)}(\boldsymbol{x},\boldsymbol{\theta})$ denotes the DNN output. The nonlinearity σ is applied entry-wise and model parameter $\boldsymbol{\theta}$ is a concatenation of all the DNN parameters. Besides, each element in $\mathbf{W}^{(l)}$ is initialized independently using the standard normal distribution. Following (Lee et al., 2019), we further set $n_1 = \ldots = n_{L-1} = n$ to simplify our analyses.

Based on the formulation above, Jacot et al. (2018) show that the training dynamics of DNNs with gradient descent can be characterized using a *neural tangent kernel* (NTK). Specifically, the NTK matrix $\boldsymbol{\Theta} \in \mathbb{R}^{m\times m}$ of a DNN model $f(\boldsymbol{x},\boldsymbol{\theta})$ on the dataset S of size m is defined as

$$\boldsymbol{\Theta}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x},\boldsymbol{\theta})^\top \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}',\boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{x}$ (or $\boldsymbol{x}'$) denotes any data point in dataset S. Interestingly, as $n_1,\ldots,n_{L-1} \to \infty$, the NTK matrix $\boldsymbol{\Theta}_0$ based on the initialized model parameters $\boldsymbol{\theta}_0$ will finally converge to a deterministic form $\boldsymbol{\Theta}_\infty$ (Jacot et al., 2018). More recently, Yang & Littwin (2021) further reveal that these conclusions hold for DNNs of any reasonable architecture. Moreover, Arora et al. (2019b) and Cao & Gu (2019) even prove that the generalization performance of DNNs can be theoretically bounded using $\boldsymbol{\Theta}_\infty$.

4. Data Valuation at Initialization (DAVINZ)

4.1. Domain-Aware Generalization Bound for DNNs

Recently, Arora et al. (2019b) and Cao & Gu (2019) have proven that the generalization errors of DNNs can be theoretically bounded using the NTK matrix with initialized model parameters, hence making the performance estimation of DNNs without model training possible in data valuation. These generalization bounds typically rely on the assumption that training dataset S and validation dataset T follow the same underlying distribution, which may not necessarily hold in practice. To overcome this limitation, we propose a novel *domain-aware* generalization bound based on the formulation of DNNs in Sec. 3.2 to estimate the generalization performance of DNNs more precisely, especially when S and T follow different underlying distributions. Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m_S}$ of size m_S be randomly sampled from a source domain $\mathcal{D}_S$ and $T = \{(\boldsymbol{x}'_i, y'_i)\}_{i=1}^{m_T}$ of size m_T be randomly sampled from a target domain $\mathcal{D}_T$. We firstly define domain discrepancy (as a measure of distribution divergence) between $\mathcal{D}_T$ and $\mathcal{D}_S$ in Definition 1, which will be used to derive our domain-aware generalization bound.

Definition 1 (Domain Discrepancy (Gretton et al., 2012a))**.** *Given any function space $\mathcal{H}$, the domain discrepancy between $\mathcal{D}_T$ and $\mathcal{D}_S$ is defined as*

$$d_{\mathcal{H}}(\mathcal{D}_T,\mathcal{D}_S) \triangleq \sup_{h\in\mathcal{H}} \left|\mathbb{E}_{\boldsymbol{x}'\sim\mathcal{D}_T}[h(\boldsymbol{x}')] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_S}[h(\boldsymbol{x})]\right|$$

which can be empirically estimated using samples S and T from the respective $\mathcal{D}_S$ and $\mathcal{D}_T$:

$$d_{\mathcal{H}}(T,S) \triangleq \sup_{h\in\mathcal{H}} \left|\frac{1}{m_T}\sum_{i=1}^{m_T} h(\boldsymbol{x}'_i) - \frac{1}{m_S}\sum_{i=1}^{m_S} h(\boldsymbol{x}_i)\right|.$$

Let the generalization error of function f on domain $\mathcal{D}$ be $\mathcal{L}_{\mathcal{D}}(f)$ and $f^* = \arg\min_f(\mathcal{L}_{\mathcal{D}_T}(f) + \mathcal{L}_{\mathcal{D}_S}(f))$ for a DNN model f. We assume that $f(\boldsymbol{x},\boldsymbol{\theta}) \in [0,1]$ and there exists a $h \in \mathcal{H}$ with $h(\boldsymbol{x}) \leq 1$ s.t. for any data point $\boldsymbol{x}$, $\left|f(\boldsymbol{x},\boldsymbol{\theta}) - f^*(\boldsymbol{x},\boldsymbol{\theta})\right| \leq h(\boldsymbol{x})$. Let $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalue of a matrix, respectively. When $\ell(f,y) = (f-y)^2/2$, the following domain-aware generalization bound can then be derived with $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_\infty$ being evaluated on S:

Theorem 1 (Domain-aware Generalization Bound)**.** *Assume $\lambda_{\min}(\boldsymbol{\Theta}_0) > 0$ and $\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x},\boldsymbol{\theta}_0)\|_2 \leq \rho$ for any $(\boldsymbol{x},y) \in S$ with $\|\boldsymbol{x}\|_2, y \in [0,1]$. There exist constants $c > 0$ and $N \in \mathbb{N}$ s.t. for every $n > N$, when applying gradient descent with learning rate $\eta < \min\{2n^{-1}(\lambda_{\min}(\boldsymbol{\Theta}_\infty) + \lambda_{\max}(\boldsymbol{\Theta}_\infty))^{-1}, m_S/\lambda_{\max}(\boldsymbol{\Theta}_0)\}$, for any f_t obtained at time $t > 0$, with probability at least $1-2\delta$,*

$$\mathcal{L}_{\mathcal{D}_T}(f_t) \leq \mathcal{L}_S(f_t) + 2\rho\sqrt{\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}/m_S} + d_{\mathcal{H}}(T,S) + \varepsilon$$

where each element in $\widehat{\boldsymbol{y}}$ is defined as $\widehat{y} \triangleq y - f(\boldsymbol{x},\boldsymbol{\theta}_0)$ and $\varepsilon \triangleq 2c/\sqrt{n} + 4\sqrt{\log(4/\delta)/(2m_S)} + \sqrt{\log(4/\delta)/(2m_T)} + \mathcal{L}_{\mathcal{D}_T}(f^*) + \mathcal{L}_{\mathcal{D}_S}(f^*)$.

Its proof is in Appendix A.1. As shown in Theorem 1, there exist two sources of generalization error: (a) in-domain generalization error characterized by $2\rho(\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}/m_S)^{1/2}$

and (b) out-of-domain generalization error characterized by $d_{\mathcal{H}}(T, S)$. Similar to (Arora et al., 2019b), the in-domain generalization error can be interpreted as a complexity measure of dataset S which is usually highly related to the value of a dataset (Mangalam & Prabhu, 2019). On the other hand, the out-of-domain generalization error measured by the domain discrepancy will become zero when S approaches T. Interestingly, when T is a preferred dataset of data consumers, this result aligns with the desirable *awareness of data preference* in data valuation, suggesting that the domain-aware generalization bound in Theorem 1 can be a good choice as the scoring function in data valuation.

4.2. Generalization Bound as Scoring Function

In conventional data valuation, validation performance (i.e., an estimate of generalization performance) is widely used as the scoring function, but is known to be computationally costly. Fortunately, our domain-aware generalization bound in Theorem 1 is able to estimate the generalization performance of DNNs without expensive model training. So, we propose to utilize it as the scoring function to achieve superior efficiency in data valuation. Specifically, by ignoring $\mathcal{L}_S(f_t)$ and ε in Theorem 1,[1] we set our scoring function as

$$\nu(S) = -\kappa\sqrt{\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}/m_S} - d_{\mathcal{H}}(T, S) \tag{3}$$

where $\widehat{\boldsymbol{y}}$ is evaluated on dataset S following its definition in Theorem 1 and $\kappa = 2\rho$. To evaluate (3) in a practical way, we need to choose a proper function space $\mathcal{H}$ and decide κ.

Choice of $\mathcal{H}$ in $d_{\mathcal{H}}(T, S)$. Note that $\mathcal{H}$ of a large capacity is required to meet the assumptions for Theorem 1. Meanwhile, the evaluation of any function $h \in \mathcal{H}$ should be computationally costly to preserve the efficiency of our training-free scoring function (3). As kernel methods based on *reproducing kernel Hilbert space* (RKHS) have been widely shown to be able to measure domain discrepancy effectively and efficiently (Sejdinovic et al., 2013; Long et al., 2015), we also choose to obtain our $\mathcal{H}$ from RKHS. Moreover, we further use a multiple kernel variant of MMD (i.e., MK-MMD) (Gretton et al., 2012b) in our scoring function to ensure a large capacity of $\mathcal{H}$.

Determination of κ. Notably, κ can be regarded as a hyper-parameter to trade off between in-domain and out-of-domain generalization errors to better characterize the generalization performance of DNNs. We usually decide κ by realizing similar averaged scales of the in-domain and out-of-domain generalization errors to balance their effects in practice. Let $\widehat{\boldsymbol{y}}_{S_i}$ and $\boldsymbol{\Theta}_{0,S_i}$ be evaluated on dataset S_i using the initialized model parameters $\boldsymbol{\theta}_0$. We set

$$\kappa = \frac{\sum_{i=1}^K d_{\mathcal{H}}(T, S_i)}{\sum_{i=1}^K \left(\widehat{\boldsymbol{y}}_{S_i}^\top \boldsymbol{\Theta}_{0,S_i}^{-1}\widehat{\boldsymbol{y}}_{S_i}/m_{S_i}\right)^{1/2}} \tag{4}$$

which can be further refined by averaging across different random initializations $\boldsymbol{\theta}_0$, if available. Our empirical experiments in Sec. 6 will validate that our scoring function (3) based on (4) can indeed perform well in practice.

[1] In practice, training error $\mathcal{L}_S(f_t)$ usually approaches zero for fully converged DNNs and can thus be ignored. Moreover, in data valuation, we are more interested in comparing the *relative* performances of ML models (Jia et al., 2019a; Xu et al., 2021b), i.e., the value of *my data* in the presence of *others' data*. So, the constant ε in Theorem 1 can be omitted from our data valuation.

Algorithm 1 **Da**ta **V**aluation at **In**itializ**a**tion (DAVINZ)

Input: Datasets $\{S_i\}_{i=1}^K$ from K data contributors, validation dataset T, DNN model f with initialized parameters $\boldsymbol{\theta}_0$, kernel k for $d_{\mathcal{H}}$, weighting factors $\alpha_{\mathcal{C}}$
for contributor $i = 1, \ldots, K$ **do**
  for coalition $\mathcal{C} \subseteq \mathcal{A} \setminus \{i\}$ **do**
    Evaluate the scores $\nu(S_{\mathcal{C}\cup\{i\}})$ and $\nu(S_{\mathcal{C}})$ by (3)
    Evaluate the marginal $\Delta_{i,\mathcal{C}} = \nu(S_{\mathcal{C}\cup\{i\}}) - \nu(S_{\mathcal{C}})$
  end for
  $\phi_i = \sum_{\mathcal{C}\subseteq\mathcal{A}\setminus\{i\}} \alpha_{\mathcal{C}} \times \Delta_{i,\mathcal{C}}$
end for

4.3. Training-free Data Valuation Algorithm

Our scoring function (3) can then be applied to the marginal contribution calculations commonly seen in data Shapley (Ghorbani & Zou, 2019) or LOO (Koh & Liang, 2017):

$$\Delta_{i,\mathcal{C}} \triangleq \nu(S_{\mathcal{C}\cup\{i\}}) - \nu(S_{\mathcal{C}}) \tag{5}$$

where $S_{\mathcal{C}} = \{S_i\}_{i\in\mathcal{C}}$ denotes the aggregated dataset from the coalition of parties $\mathcal{C} \subseteq \mathcal{A} \triangleq \{1, \ldots, K\}$. Finally, the value of any dataset S_i provided by contributor i can be evaluated as a weighted average of marginal contributions of S_i to all possible coalitions excluding i, as in the Shapley value (Shapley, 1953), Banzhaf value (Banzhaf, 1964; Dubey & Shapley, 1979), or leave-one-out (Cook, 1977):

$$\phi_i \triangleq \sum_{\mathcal{C}\subseteq\mathcal{A}\setminus\{i\}} \alpha_{\mathcal{C}} \times \Delta_{i,\mathcal{C}} \tag{6}$$

where $\alpha_{\mathcal{C}} \geq 0\ \forall\mathcal{C}$ are the weighting factors. In particular, for the commonly adopted Shapley value, $\alpha_{\mathcal{C}} = |\mathcal{C}|!(K - |\mathcal{C}| - 1)!/K!$. For Banzhaf value, $\alpha_{\mathcal{C}} = 1/2^{K-1}$. For LOO, $\alpha_{\mathcal{C}} = \mathbb{1}_{\mathcal{C}=\mathcal{A}\setminus\{i\}}$. Our *Data Valuation at Initialization* (DAVINZ) algorithm (Algorithm 1) is therefore completed.

5. Properties of DAVINZ

As summarized in Sec. 3.1, data valuation methods should enjoy the awareness of data preference and data quantity, the stability to noise, and the robustness to model in practice. Sec. 4.1 has already shown that the domain-aware

generalization bound in Theorem 1 is aware of data preference (i.e., validation dataset). Hence, DAVINZ using this generalization bound as the scoring function should also enjoy its awareness of data preference. In this section, we further show that our DAVINZ can also attain the other three properties theoretically.

5.1. Awareness of Data Quantity

We prove that our scoring function (3) is aware of data quantity in Proposition 1 by showing that it would distribute a higher score to the dataset with more data samples:

Proposition 1 (Awareness of Data Quantity)**.** *Following (Nguyen et al., 2021), suppose that $\mathcal{D}_S$ with zero mean satisfies Assumptions A.1 and A.2, and $\exists \alpha > 0$ s.t. $d = \Theta(m^\alpha)$ $\forall m \in \mathbb{N}^+$. Then, there is a constant $\beta > 0$ s.t. with a high probability,*

$$\nu(S) \geq -\kappa\beta m_S^{-\alpha/2} - d_{\mathcal{H}}(T, S)\ . \tag{7}$$

Its proof is in Appendix A.2. Proposition 1 suggests that when the size m_S of dataset S increases and $d_{\mathcal{H}}(T, S)$ only admits a minor change, our scoring function (3) would probably give a higher score to S. Notably, $d_{\mathcal{H}}(T, S)$ indeed only undergoes a minor change when sampling more data from the same source domain $\mathcal{D}_S$ for the dataset S that already achieves a large m_S. This is because the empirical expectation in $d_{\mathcal{H}}(T, S)$ (see Definition 1) would be approximately the same in this case according to the law of large numbers. These results imply that our scoring function and DAVINZ based on it indeed enjoy the awareness of data quantity property.

5.2. Stability to Noise

We prove that our scoring function (3) can also enjoy valuation stability to noise by showing that it gives similar scores to the original dataset and its counterpart with a small-scale noise. Given $\epsilon \in [0, 1]$, let $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_{0,\epsilon}$ be the NTK matrices of DNN model f evaluated on $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m_S}$ and its noisy counterpart $S_\epsilon = \{(\boldsymbol{x}_{i,\epsilon}, y_i)\}_{i=1}^{m_S}$, respectively. Let $\|\boldsymbol{x}_i - \boldsymbol{x}_{i,\epsilon}\|_2 \leq \epsilon$ for any i, each element in $\widehat{\boldsymbol{y}}_\epsilon$ be $\widehat{y}_\epsilon \triangleq y - f(\boldsymbol{x}_\epsilon, \boldsymbol{\theta}_0)$, and $\Delta(S, S_\epsilon) \triangleq |d_{\mathcal{H}}(T, S) - d_{\mathcal{H}}(T, S_\epsilon)|$. Then, the following result can be derived:

Proposition 2 (Stability to Noise)**.** *Assume $|f(\boldsymbol{x}, \boldsymbol{\theta}_0)| \leq \tau$ for any $\boldsymbol{x}_i$ or $\boldsymbol{x}_{i,\epsilon}$, $\min(\lambda_{\min}(\boldsymbol{\Theta}_0), \lambda_{\min}(\boldsymbol{\Theta}_{0,\epsilon})) > \lambda$, and $\min(\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1} \widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \widehat{\boldsymbol{y}}_\epsilon) \geq \gamma$ for any $\epsilon \in [0, 1]$. Then, there exists a constant $\beta > 0$ s.t. with a high probability,*

$$|\nu(S_\epsilon) - \nu(S)| \leq \frac{\kappa}{2\sqrt{\gamma}} \left(O(\tau) + \beta m_S^{3/2} \epsilon / \lambda^2\right) + \Delta(S, S_\epsilon).$$

Its proof is in Appendix A.3 and the definition of $O(\cdot)$ is in Appendix A.2. Proposition 2 suggests that our scoring function (3) would probably give comparable scores to the original dataset and also its counterpart with a small-scale noise (i.e., small ϵ) when $\Delta(S, S_\epsilon)$ is also small. Though it is non-trivial to show that $\Delta(S, S_\epsilon)$ is small theoretically, our empirical results in Sec. 6.5 validate that a small divergence between $\nu(S)$ and $\nu(S_\epsilon)$ can indeed be achieved when ϵ is small. Our scoring function and DAVINZ based on it can thus also enjoy the stability to noise property.

5.3. Robustness to Model

We prove that our scoring function (3) is robust to model choices by showing that it would distribute similar scores to a dataset even when distinct DNN models are used under certain conditions. Specifically, let $\boldsymbol{\Theta}_{0,f}$ and $\boldsymbol{\Theta}_{0,f'}$ be the NTK matrices of DNN models f and f' that are evaluated on the same dataset S of size m_S, respectively. Let each element in $\widehat{\boldsymbol{y}}_f$ be $\widehat{y}_f \triangleq y - f(\boldsymbol{x}, \boldsymbol{\theta}_0)$ with $(\boldsymbol{x}, y) \in S$; $\widehat{y}_{f'}$ enjoys a similar form. We derive the following result:

Proposition 3 (Robustness to Model)**.** *Suppose that $|f(\boldsymbol{x}, \boldsymbol{\theta}_0)| \leq \tau$ and $\left|f'(\boldsymbol{x}, \boldsymbol{\theta}_0)\right| \leq \tau$ for any $\boldsymbol{x}$ in S, $\min(\lambda_{\min}(\boldsymbol{\Theta}_{0,f}), \lambda_{\min}(\boldsymbol{\Theta}_{0,f'})) > \lambda$, and $\min(\widehat{\boldsymbol{y}}_f^\top \boldsymbol{\Theta}_{0,f}^{-1} \widehat{\boldsymbol{y}}_f, \widehat{\boldsymbol{y}}_{f'}^\top \boldsymbol{\Theta}_{0,f'}^{-1} \widehat{\boldsymbol{y}}_{f'}) \geq \gamma$. With a high probability, if $\|\boldsymbol{\Theta}_{0,f} - \boldsymbol{\Theta}_{0,f'}\|_2 \leq \epsilon$, then*

$$\left|\nu(S; f') - \nu(S; f)\right| \leq \frac{\kappa}{2\sqrt{\gamma}} \left(O(\tau) + \sqrt{m_S}\epsilon / \lambda^2\right).$$

Its proof is in Appendix A.4. Proposition 3 suggests that $\|\boldsymbol{\Theta}_{0,f} - \boldsymbol{\Theta}_{0,f'}\|_2$ can be interpreted as a measure to evaluate the robustness to model property of our scoring function (3). Specifically, when applying two different DNN models to evaluate the value of the same dataset, our scoring function would probably deliver comparable scores to this dataset if these two models share similar NTK matrices (i.e., small ϵ in Proposition 3). Surprisingly, even in the case of a large ϵ, our DAVINZ still produces a consistent data valuation when DNNs achieve large $\lambda_{\min}(\boldsymbol{\Theta}_0)$, as implied in Proposition 3. Therefore, under any one of the conditions above, our scoring function and DAVINZ based on it can enjoy the robustness to model property.

6. Experiments

6.1. Valid Scoring Function in Practice

To justify the validity of our scoring function (3) in practice, we examine the empirical gap and the Pearson correlation between the estimated score and the corresponding ground truth (i.e., validation accuracy of converged DNNs) of different datasets. In particular, we construct 200 datasets in this experiment and each dataset consists of up to 10K randomly bootstrapped MNIST images (Lecun et al., 1998). A DNN with two convolutional layers followed by a fully connected layer is employed to evaluate (3) on these datasets.

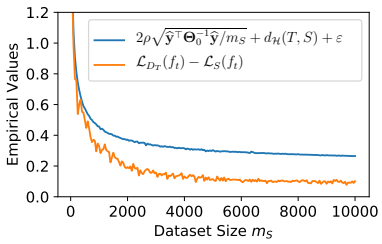

Figure 1. Empirical values of the generalization error and the estimated bound in Theorem 1. The constant $2c/\sqrt{n}$ in ε is ignored.

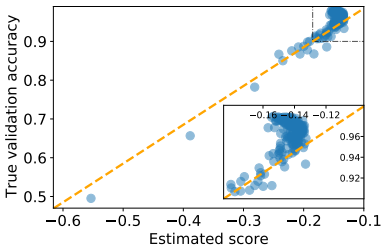

Figure 2. Correlation between the estimated score using (3) and the ground truth of 200 randomly bootstrapped MNIST datasets. The orange dotted line has a slope of 1.

Fig. 1 illustrates the empirical values of the generalization gap (orange) and the theoretical upper bound (blue) in Theorem 1. The two values show strong correlations, thus implying that the bound can be useful for data valuation. Fig. 2 further demonstrates the strong correlation under the same experimental setting. The estimated scores provided by (3), surprisingly, achieve a nearly linear correlation with a coefficient of 0.954 to the ground truth, as shown in Fig. 2, which strongly confirms the validity of (3) in practice. Our scoring function (3) can thus be used as a promising alternative to the validation performance in marginal contribution-based data valuation methods (e.g., data Shapley (Ghorbani & Zou, 2019)). More interestingly, as shown in the zoomed-in plot of Fig. 2, our scoring function may slightly underestimate the true validation accuracy of DNNs. This observation can be well-explained by our Theorem 1, which provides an upper bound to the generalization error of DNNs and hence a lower bound to the true validation accuracy. Despite this marginal underestimation, it is still promising to apply our training-free scoring function to improve the efficiency of conventional data valuation pipelines (Ghorbani & Zou, 2019; Ghorbani et al., 2020; 2021; Jia et al., 2019b; Koh et al., 2019) and preserve compelling effectiveness, which we will demonstrate in the next section.

6.2. Effective and Efficient DAVINZ

We then compare our DAVINZ against other data valuation baselines (e.g., *validation performance* (VP), *influence function* (IF) (Koh et al., 2019), and *robust volume* (RV) (Xu et al., 2021b)) to demonstrate the effectiveness and efficiency of our DAVINZ. In VP, the validation performance after a model training of 300 epochs on a given training dataset is employed as the scoring function. For IF and RV, refer to Appendices D.4 and D.5 for more details. The ground truth is averaged over 5 independent evaluations using fully converged DNN models (i.e., a model training of $\gg 300$ epochs). To measure the effectiveness of different data valuation methods, we evaluate the Pearson and Spearman correlation between their estimated data values and the corresponding ground truth based on LOO. Meanwhile, the efficiency of these methods is measured by the wall-clock computational cost. The comparison is performed on both classification and regression tasks. For classification tasks, we use a particularly difficult dataset split on MNIST and CIFAR-10 (Krizhevsky et al., 2009) datasets by simulating 10 data contributors with each having a different number of images from a single class label (e.g., only digit 0 or only airplanes). For the regression task, we use the ising physical model dataset (Mills & Tamblyn, 2019), which is split via varying the number of samples more, as compared with that on the classification task. Refer to Appendix D.1 for more details about these two experiments.

Tables 1 and 2 summarize the results on the classification and regression tasks, respectively. Notably, our DAVINZ consistently achieves better correlations and lower evaluation costs than other training-free baselines (i.e., IF and RV).[2] Even when compared with the training-based VP method, our DAVINZ can achieve comparable correlations to the ground truth while incurring more than $30\times$ lower computational costs. Different from the compelling performance achieved by IF in (Jia et al., 2019b), IF generally performs poorly in our experiments, which may result from the highly non-convex nature of DNNs that violate the assumptions in IF. Moreover, the essential interdependence among samples in a dataset from a contributor in these two experiments may be ignored by IF since it evaluates the influence with a simple arithmetic summation over individual data samples. Surprisingly, VP achieves poor correlations on the ising physical model dataset when using CNN8. This may result from the difficulty in deciding proper hyper-parameters (e.g., number of epochs, learning rates) for the model training in VP. Fortunately, our training-free DAVINZ is independent of these hyper-parameters and can consequently circumvent the uncertainties introduced by these handcrafted design choices in model training, thus leading to more consistent results.

6.3. Awareness of Data Preference

Our training-free data valuation algorithm DAVINZ novelly uses a domain discrepancy between the training and valida-

[2]Though IF and RV require a one-shot model training on the grand coalition of datasets, they are regarded as training-free data valuation baselines in this paper since any further model re-training after the one-shot model training can be avoided in IF and RV.

Table 1. Comparison among DAVINZ and other baselines on classification tasks. Each correlation coefficient is reported with the mean and standard error over 5 independent evaluations. Each cost includes the evaluation of 11 scores for LOO over 10 different datasets.

Method	Model	MNIST			CIFAR-10		
		Pearson	Spearman	Cost (Min.)	Pearson	Spearman	Cost (Min.)
VP	VGG13	1.00±0.00	0.98±0.01	88.6	0.53±0.28	0.77±0.09	88.4
	ResNet18	0.99±0.00	0.97±0.01	185.9	0.63±0.17	0.70±0.09	211.8
IF	VGG13	0.17±0.04	0.30±0.07	11.0	0.55±0.04	0.57±0.03	11.0
	ResNet18	0.42±0.05	0.55±0.07	22.6	0.08±0.07	0.07±0.10	26.3
RV	VGG13	−0.01±0.05	−0.14±0.08	9.7	0.17±0.03	0.32±0.06	9.6
	ResNet18	−0.36±0.11	−0.30±0.05	18.8	0.18±0.05	0.22±0.07	21.6
DAVINZ	VGG13	0.84±0.01	0.52±0.02	**2.5**	0.46±0.10	0.44±0.12	**2.0**
	ResNet18	0.85±0.00	0.62±0.00	**3.3**	0.55±0.03	0.67±0.03	**3.2**

Table 2. Comparison among DAVINZ and other baselines on a regression task. Similarly, each correlation coefficient is reported with the mean and standard error over 5 independent evaluations.

Method	Model	Ising Physical Model Dataset		
		Pearson	Spearman	Cost (Min.)
VP	MLP10	0.998±0.001	0.978±0.007	17.1
	CNN8	0.317±0.169	0.273±0.137	34.4
IF	MLP10	0.095±0.250	−0.006±0.072	1.9
	CNN8	0.189±0.142	0.001±0.124	4.1
RV	MLP10	0.727±0.231	0.699±0.182	2.0
	CNN8	0.805±0.009	0.818±0.041	4.1
DAVINZ	MLP10	0.994±0.001	0.905±0.018	**1.7**
	CNN8	0.823±0.003	0.702±0.063	**2.0**

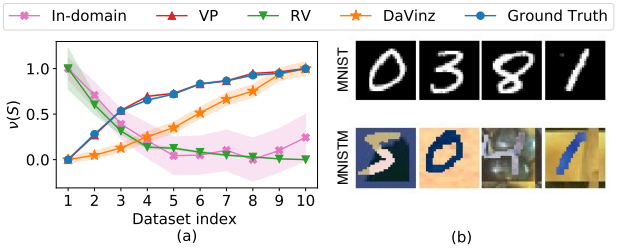

Figure 3. (a) Awareness of data preference achieved by different methods. Results (re-scaled to $[0, 1]$ using min-max normalization) are reported with the mean and standard error over 10 independent evaluations. (b) Examples of MNISTM and MNIST images. MNISTM is created by translating and adding randomly selected backgrounds to MNIST images, as detailed in (Ganin et al., 2016).

tion datasets to realize its awareness of data preference, as justified in Sec. 4.1. To verify this property, we compare DAVINZ with other data valuation baselines on MNIST and MNISTM (Ganin et al., 2016) datasets using the same DNN model in Sec. 6.1.[3] Specifically, we construct 10 training datasets of size 10K and each training dataset consists of a different mixture of MNISTM and MNIST images. For example, dataset S_1 contains 10% MNISTM images and 90% MNIST images, dataset S_2 contains 20% MNISTM images and 80% MNIST images, and so on. However, the validation dataset only contains MNISTM images to indicate the preference of the data consumer in practice. Thus, from dataset S_1 to dataset S_{10}, the domain discrepancy between the training and validation datasets becomes smaller.

As shown in Fig. 3a, the dataset scores provided by our scoring function (3) with a sufficiently large κ (i.e., with only the term related to the in-domain generalization error in Theorem 1) shows an inconsistent trend to the ground truth. This observation indicates that the in-domain generalization error term in (3) alone is not capable of capturing any potential divergence between the training and validation datasets. Fortunately, our scoring function (3) is able to mitigate this undesirable phenomenon by applying the domain discrepancy term within it (i.e., with an appropriate κ according to (4)) to capture the potential divergence between the training and validation datasets. Specifically, in Fig. 3a, our DAVINZ is shown to achieve a similar trend of dataset scores as the ground truth (with a strong Pearson correlation of 0.960) and VP. In contrast, RV shows an inconsistent trend to the ground truth in this scenario because RV is validation-free and may score datasets incorrectly especially when there exists significant domain divergence between the training and validation datasets.

Overall, our method has shown its awareness of data preference. We recognize this awareness of data preference as an important property in validation-based data valuation since it helps data consumers decide which dataset is more useful based on their application preferences and allows them to make more informed procurement decisions. Moreover, our DAVINZ advances other baselines by achieving higher flexibility in data valuation when data consumers have their preferences for κ. For example, data consumers can use κ

[3] We do not discuss IF here and in the following experiments about the behavior of $\nu(S)$ because IF requires defining an extra grand coalition dependent $\nu(S_{\mathcal{A}})$ term to evaluate individual $\nu(S)$.

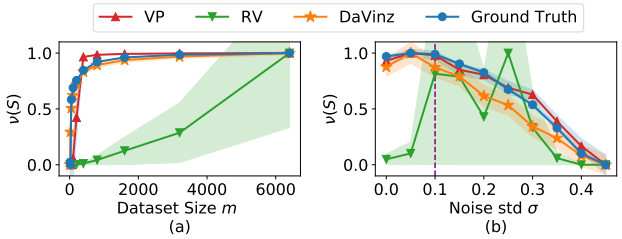

Figure 4. (a) Awareness of data quantity and (b) stability to noise achieved by different data valuation methods. Results of each method (re-scaled to $[0, 1]$ using min-max normalization) are reported with the mean and standard error over 10 evaluations.

to represent their trust in the data contributors in providing useful datasets close to their interested application (i.e., the validation set): Use a large κ in the collaboration among government agencies when there exists unflinching trust and a small κ among self-interested data sellers.

6.4. Awareness of Data Quantity

To justify that DAVINZ is aware of data quantity (Sec. 5.1), we compare it with other baselines using a 10-layer MLP model on the ising model dataset. The results are illustrated in Fig. 4a. Notably, both ground truth and VP demonstrate that a dataset of a larger sample quantity typically enjoys a higher score. This increasing sample quantity would eventually achieve a diminished marginal effect on the score of datasets in both ground truth and VP. Remarkably, nearly the same results can be achieved by DAVINZ, which conforms to our theoretical analysis in Sec. 5.1. On the other hand, RV has a distinct behavior compared to ground truth and VP. Overall, our DAVINZ is able to be aware of data quantity in practice. As an implication, data contributors can usually contribute more valuable datasets to data consumers by collecting more data samples.

6.5. Stability to Noise

We then investigate the effect of noises on data valuation across baselines (Sec. 5.2). A 10-layer MLP model and the ising model dataset are used. In particular, we construct 10 different datasets of size 500 by adding Gaussian noise of different scales into the original dataset. Specifically, the σ in the Gaussian noise $\mathcal{N}(0, \sigma^2)$ is increased uniformly from 0 to 0.45 for these 10 datasets. The results are illustrated in Fig. 4b. Notably, relatively stable dataset scores are achieved by ground truth, VP, and our DAVINZ, even after adding small-scale Gaussian noises (i.e., $\sigma < 0.1$ in this case) into the original datasets. This can be explained by the robustness of DNNs after their model training with random noise. Meanwhile, our DAVINZ can also enjoy a similar decreasing trend of dataset scores to ground truth and VP when noise increases to a large scale (i.e., $\sigma > 0.1$). Interestingly, this phenomenon can also be explained by Proposition 2. According to Proposition 2, the scores of the noisy and original datasets should experience a larger divergence given a larger-scale noise. Since large-scale noises would typically hurt the performances of ML models, the noisy dataset should receive a much lower score than the original dataset when ϵ is large in our Proposition 2. For RV, it produces the most unstable and imprecise results in Fig. 4b, which further implies that our DAVINZ is superior to it. Overall, DAVINZ has shown its stability to small-scale noises and also its consistency with training-based methods (e.g., VP) when large noises are present in datasets. As an implication, data contributors need to clean their noisy datasets to achieve their desirable higher dataset values.

Table 3. Data valuation using different DNNs (i.e., from f to f'). Each result is averaged over 10 datasets and 5 initializations.

Model	$f \to f'$	$\epsilon\%$ (%)	$\lambda_{\min,f}, \lambda_{\min,f'}$ ($\times 10^{-5}$)	$\Delta^{\text{DAVINZ}}_{\nu(S)}$ (%)	$\Delta^{\text{VP}}_{\nu(S)}$ (%)
VGG	11→13	97.0±0.0	56, 1.6	4.8±0.8	2.0±0.2
	11→16	99.8±0.0	56, 0.10	8.3±0.4	8.1±0.4
ResNet	18→21	38.8±2.6	1300, 1600	5.0±0.3	9.9±0.3
	18→34	101.6±3.5	1300, 2100	4.2±0.3	7.2±0.9

6.6. Robustness to Model

We finally examine the provable model-robust property of our data valuation method DAVINZ. We apply VGG and ResNet on 10 randomly sampled datasets containing 1000 CIFAR-10 images each to compare the DAVINZ scores under different DNNs. The results are summarized in Table 3 where $\epsilon\% = \|\mathbf{\Theta}_{0,f} - \mathbf{\Theta}_{0,f'}\|_2 / \|\mathbf{\Theta}_{0,f}\|_2$ measures the relative difference of NTK matrices, $\lambda_{\min,f}$ denotes the minimum eigenvalue of the NTK matrix on f, and $\Delta_{\nu(S)} = |(\nu(S; f) - \nu(S; f'))/\nu(S; f)|$. Surprisingly, although the NTK matrix (i.e., $\epsilon\%$) has considerable variations when the depth of VGG or ResNet increases, DAVINZ using these models is still able to give consistent dataset values, which aligns with the results of training-based method VP. Moreover, similar to the training-based method VP, we also observe a more consistent data valuation when using a DNN with a larger $\lambda_{\min}$ (e.g., the results of ResNet21 vs. ResNet34 in Table 3), which conforms to the conclusion in Proposition 3. More results about the model-robust property of our DAVINZ are provided in Appendix E.1. Overall, our DAVINZ enjoys robustness to models, which perfectly aligns with training-based methods. Therefore, our DAVINZ can be an efficient alternative to training-based data valuation methods, even when consistent data valuation on different DNNs is required.

6.7. Application: Large-scale Shapley Value

We showcase that DAVINZ makes Shapley value calculation of *large-scale* collaboration (Jia et al., 2019a) on complex deep neural networks feasible in practice. We construct 100

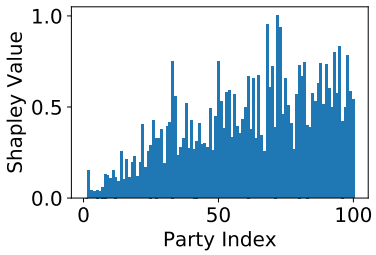

Figure 5. DAVINZ Shapley value of 100 data contributors with ascending size of CIFAR-10 images. Values are re-scaled to $[0, 1]$ using min-max normalization. The model used is ResNet18.

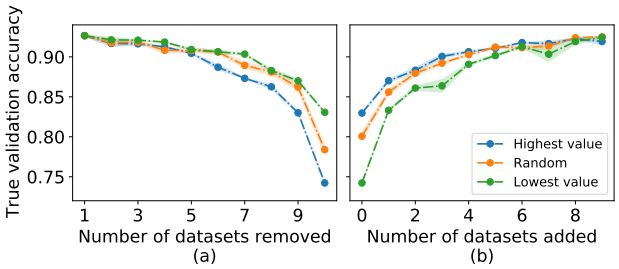

Figure 6. Effect on true validation accuracy when (a) removing or (b) adding datasets in the order of dataset values. The plots show average and standard error over 10 random initializations.

data contributors containing ascending numbers of CIFAR-10 images s.t. S_1 has 5 data samples and S_{100} has 500 samples. DAVINZ significantly reduces the computation time for each contribution evaluation to a matter of seconds. Using the truncated Monte Carlo Shapley algorithm (Ghorbani & Zou, 2019) for 1K permutations of 100 contributors' orderings (i.e., approximately equivalent to 20K model training in VP), we obtain the results in Fig. 5 in a few hours. The ascending trend of the Shapley value in Fig. 5 confirms our construction of ascending dataset sizes and thus the effectiveness of DAVINZ.

6.8. Application: Data Summarization

Data summarization (Ghorbani & Zou, 2019; Wang et al., 2021b) is another application in the age of big data that an efficient data valuation algorithm like DAVINZ can enable. Specifically, we often have plenty of datasets collected in various means, but it can be too computationally costly to train a model using all of them. Which datasets should be removed first without significantly degrading the model's performance? On the contrary, we can also use data valuation to advise a data buyer under budget to choose a small yet representative subset of datasets for model training. We show promising results of DAVINZ's data summarization over a large number of datasets.

We propose a modified training-free data valuation algorithm suitable for large-scale data summarization. To improve the runtime efficiency, we change line 3 of Algorithm 1 to $\mathcal{C} = \emptyset$ s.t. we only consider the marginal contribution of a dataset w.r.t. the empty set.

With the modification proposed above, the time complexity of DAVINZ only scales linearly with the number of datasets. Thus, it has the ability to perform summarization over a large number of datasets. As an illustration, we experiment with 2 MNIST images randomly split into 10 datasets and investigate the effect on true validation accuracy when datasets are sequentially added or removed from the training set. As shown in Fig. 6a, we can still achieve a validation accuracy comparable to that achieved using the full training set after removing 7 out of 10 datasets of the lowest values. Therefore, we suggest that resource-constrained practitioners remove datasets with the lowest value first to save computational cost. Similarly, in Fig. 6b, we obtain a relatively high validation performance after adding only 3 datasets of the highest values. Thus, we suggest that budgeted data consumers buy datasets with the highest values first.

7. Conclusion & Discussion

In this paper, we have introduced the DAVINZ algorithm which is a novel training-free method for efficient and trustworthy data valuation in applications using large complex DNNs. In particular, we have novelly derived a *domain-aware* generalization bound for DNNs using the recent NTK theory to characterize the performances of DNNs without model training. By exploiting this generalization bound as the scoring function and using conventional data valuation techniques (e.g., SV and LOO) as the valuation function, DAVINZ is capable of valuating data effectively and efficiently. Interestingly, our *training-free* DAVINZ is able to enjoy the desirable properties that training-based data valuation methods usually attain (e.g., awareness of data preference and data quantity, stability to noise, and robustness to model). This further implies the reliability of our DAVINZ in practice. Moreover, since DAVINZ significantly reduces the computational costs of practical data valuation with DNNs, it even makes the calculation of Shapley value on *large-scale* collaborations and also data summarization over *a large number* of datasets affordable. Overall, thanks to the remarkable effectiveness and efficiency of our DAVINZ, it should be able to enjoy a wider applicability than conventional training-based methods.

Acknowledgements

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018).

References

Agarwal, A., Dahleh, M., and Sarkar, T. A marketplace for data: An algorithmic solution. In *Proc. ACM EC*, pp. 701–726, 2019.

Agussurja, L., Xu, X., and Low, B. K. H. On the convergence of the Shapley value in parametric Bayesian learning games. In *Proc. ICML*, 2022.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. In *Proc. NeurIPS*, pp. 8139–8148, 2019a.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *Proc. ICML*, pp. 322–332, 2019b.

Asif, A. and Moura, J. Block matrices with l-block-banded inverse: inversion algorithms. *IEEE Transactions on Signal Processing*, 53(2):630–642, 2005.

Banzhaf, J. F. I. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317, 1964.

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *Proc. ICLR*, 2021.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

Bishop, C. M. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Proc. NeurIPS*, pp. 10836–10846, 2019.

Cook, R. D. Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18, 1977.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, pp. 248–255, 2009.

Dubey, P. and Shapley, L. S. Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 17(1): 2096–2030, 2016.

Ghorbani, A. and Zou, J. Data Shapley: Equitable valuation of data for machine learning. In *Proc. ICML*, pp. 2242–2251, 2019.

Ghorbani, A., Kim, M., and Zou, J. A distributional framework for data valuation. In *Proc. ICML*, pp. 3535–3544, 2020.

Ghorbani, A., Zou, J., and Esteva, A. Data Shapley valuation for efficient batch active learning. arXiv:2104.08312, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *JMLR*, 13(25): 723–773, 2012a.

Gretton, A., Sriperumbudur, B. K., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., and Fukumizu, K. Optimal kernel choice for large-scale two-sample tests. In *Proc. NeurIPS*, pp. 1205–1213, 2012b.

Han, D., Wooldridge, M., Rogers, A., Tople, S., Ohrimenko, O., and Tschiatschek, S. Replication-robust payoff-allocation for machine learning data markets. arXiv:2006.14583, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proc. CVPR*, pp. 770–778, 2016.

IMDA. Guide to data valuation for data sharing. Technical report, The Infocomm Media Development Authority Singapore, 2019.

Jacot, A., Gabriel, F., and Hongler, C. Neural Tangent Kernel: Convergence and generalization in neural networks. In *Proc. NeurIPS*, pp. 8580–8589, 2018.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., Zhang, C., Spanos, C., and Song, D. Efficient task-specific data valuation for nearest neighbor algorithms. *Proc. VLDB Endowment*, 12(11):1610–1623, 2019a.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. Towards efficient data valuation based on the Shapley value. In *Proc. AISTATS*, pp. 1167–1176, 2019b.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *Proc. ICML*, pp. 1885–1894, 2017.

Koh, P. W., Ang, K.-S., Teo, H., and Liang, P. S. On the accuracy of influence functions for measuring group effects. In *Proc. NeurIPS*, pp. 5254–5264, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, 2009.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28:1302–1338, 2000.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Proc. NeurIPS*, pp. 8572–8583, 2019.

Lo, B. and DeMets, D. L. Incentives for clinical trialists to share data. *New England Journal of Medicine*, 375(12): 1112–1115, 2016.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. Learning transferable features with deep adaptation networks. In *Proc. ICML*, pp. 97–105, 2015.

Mangalam, K. and Prabhu, V. U. Do deep neural networks learn shallow learnable examples first? In *Proc. ICML Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.

Mills, K. and Tamblyn, I. Big graphene dataset. https://nrc-digital-repository.canada.ca/eng/view/object/?id=9f09901d-0736-4204-a35d-0c88ffb8da3b, 2019.

Nguyen, Q., Mondelli, M., and Montúfar, G. F. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep ReLU networks. In *Proc. ICML*, pp. 8119–8129, 2021.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41(5): 2263–2291, 2013.

Shapley, L. S. A value for n-person games. In *Contributions to the Theory of Games (AM-28), Volume II*, chapter 17, pp. 307–318. Princeton University Press, 1953.

Shu, Y., Dai, Z., Wu, Z., and Low, B. K. H. Unifying and boosting gradient-based training-free neural architecture search. arXiv:2201.09785, 2022.

Sim, R. H. L., Zhang, Y., Chan, M. C., and Low, B. K. H. Collaborative machine learning with incentive-aware model rewards. In *Proc. ICML*, pp. 8927–8936, 2020.

Sim, R. H. L., Xu, X., and Low, B. K. H. Data valuation in machine learning: "ingredients", strategies, and open challenges. In *Proc. IJCAI*, 2022.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.

Tay, S. S., Xu, X., Foo, C. S., and Low, B. K. H. Incentivizing collaboration in machine learning via synthetic data rewards. In *Proc. AAAI*, 2022.

Wang, T., Yang, Y., and Jia, R. Learnability of learning performance and its application to data valuation. arXiv:2107.06336, 2021a.

Wang, T., Zeng, Y., Jin, M., and Jia, R. A unified framework for task-driven data quality management. arXiv:2106.05484, 2021b.

Xu, X., Lyu, L., Ma, X., Miao, C., Foo, C. S., and Low, B. K. H. Gradient driven rewards to guarantee fairness in collaborative machine learning. In *Proc. NeurIPS*, pp. 16104–16117, 2021a.

Xu, X., Wu, Z., Foo, C. S., and Low, B. K. H. Validation free and replication robust volume-based data valuation. In *Proc. NeurIPS*, pp. 10837–10848, 2021b.

Yang, G. and Littwin, E. Tensor programs IIb: Architectural universality of neural tangent kernel training dynamics. In *Proc. ICML*, pp. 11762–11772, 2021.

Yoon, J., Arik, S. O., and Pfister, T. Data valuation using reinforcement learning. In *Proc. ICML*, pp. 10842–10851, 2020.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Proc. NeurIPS*, pp. 3394–3404, 2017.

A. Proofs

A.1. Proof of Theorem 1

We firstly introduce the following lemma, which is adapted from (34) based on $\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 \leq \rho$ and $\ell(\cdot, \cdot)$ being 1-Lipschitz continuous in the proof of Theorem 2 in (Shu et al., 2022).

Lemma A.1 (In-domain Generalization Bound using NTK)**.** *Assume that $\lambda_{\min}(\boldsymbol{\Theta}_0) > 0$ and $\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 \leq \rho$ for any $(\boldsymbol{x}, y) \in S$ sampled from $\mathcal{D}$ with $\|\boldsymbol{x}\|_2 \leq 1$ and $y \in [0, 1]$. Given the loss function $\ell(f, y) \triangleq (f - y)^2/2$ and define $\widehat{y} \triangleq y - f(\boldsymbol{x})$, there exist constants $c > 0$ and $N \in \mathbb{N}$ such that for every $n > N$, when applying gradient descent with learning rate $\eta < \min\{2n^{-1}(\lambda_{\min}(\boldsymbol{\Theta}_\infty) + \lambda_{\max}(\boldsymbol{\Theta}_\infty))^{-1}, m\lambda_{\max}^{-1}(\boldsymbol{\Theta}_0)\}$, for all the functions f_t obtained during the optimization, with a high probability $(1 - \delta)$ over the dataset S of size m, we have*

$$\mathcal{L}_{\mathcal{D}}(f_t) \leq \mathcal{L}_S(f_t) + 2\rho\sqrt{\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1} \widehat{\boldsymbol{y}}/m} + \varepsilon$$

where $\widehat{\boldsymbol{y}} = [\widehat{y}_1 \cdots \widehat{y}_m]^\top$, $\varepsilon \triangleq 2c/\sqrt{n} + 3\sqrt{\log(4/\delta)/(2m)}$ and $\lambda_{\min}(\cdot), \lambda_{\max}(\cdot)$ denotes the minimum and maximum eigenvalue of a matrix, respectively.

Remark. Note that the assumption of $\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 \leq \rho$ can be well-satisfied as shown in Lemma 1 of (Lee et al., 2019). Besides, as justified by Shu et al. (2022), $\lambda_{\min}(\boldsymbol{\Theta}_0) > 0$ can be satisfied by introducing zero-mean noise into the gradient of model parameters and this lemma will still hold with high probability in this case.

Define $\varepsilon_{\mathcal{D}_T} \triangleq \mathcal{L}_{\mathcal{D}_T}(f^*)$ and $\varepsilon_{\mathcal{D}_S} \triangleq \mathcal{L}_{\mathcal{D}_S}(f^*)$, we naturally have $\varepsilon_{\mathcal{D}} = \varepsilon_{\mathcal{D}_T} + \varepsilon_{\mathcal{D}_S}$ since $\varepsilon_{\mathcal{D}} \triangleq \mathcal{L}_{\mathcal{D}_T}(f^*) + \mathcal{L}_{\mathcal{D}_S}(f^*)$. Let ϕ_S and ϕ_T be the probability density function for data distribution $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively. Since $d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) \triangleq \sup_{h \in \mathcal{H}} \left|\mathbb{E}_{\mathcal{D}_S}[h(\cdot)] - \mathbb{E}_{\mathcal{D}_T}[h(\cdot)]\right|$, inspired by Ben-David et al. (2010), we have the following inequalities by assuming that loss function $\ell(\cdot, \cdot)$ is α-Lipschitz continuous in the first argument.

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_T}(f) - \mathcal{L}_{\mathcal{D}_T}(f^*) &\overset{(a)}{\leq} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_T}\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right| \\
&\overset{(b)}{\leq} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right| + \\
&\quad \left|\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right| - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_T}\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right|\right| \\
&\overset{(c)}{\leq} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right| + \left|\int \left(\phi_S(\boldsymbol{x}) - \phi_T(\boldsymbol{x})\right)\left|\ell(f(\boldsymbol{x}), y) - \ell(f^*(\boldsymbol{x}), y)\right| \mathrm{d}\boldsymbol{x}\right| \\
&\overset{(d)}{\leq} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\left(\ell(f(\boldsymbol{x}), y) + \ell(f^*(\boldsymbol{x}), y)\right) + \alpha\left|\int \left(\phi_S(\boldsymbol{x}) - \phi_T(\boldsymbol{x})\right)\left|f(\boldsymbol{x}) - f^*(\boldsymbol{x})\right| \mathrm{d}\boldsymbol{x}\right| \\
&\overset{(e)}{\leq} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\ell(f(\boldsymbol{x}), y) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_S}\ell(f^*(\boldsymbol{x}), y) + \alpha\left|\int \left(\phi_S(\boldsymbol{x}) - \phi_T(\boldsymbol{x})\right) h(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right| \\
&\overset{(f)}{\leq} \varepsilon_{\mathcal{D}_S} + \mathcal{L}_{\mathcal{D}_S}(f) + \alpha \sup_{h \in \mathcal{H}}\left|\mathbb{E}_{\mathcal{D}_S}[h(\boldsymbol{x})] - \mathbb{E}_{\mathcal{D}_T}[h(\boldsymbol{x})]\right| \\
&\overset{(g)}{\leq} \varepsilon_{\mathcal{D}_S} + \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S)
\end{aligned} \tag{8}$$

where (a) and (b) derive from the triangle inequality. Besides, (d) is based on the assumption that loss function $\ell(\cdot, \cdot)$ is α-Lipschitz continuous in the first argument and (e) relies on the assumption in Theorem 1 that there exists at least one $h \in \mathcal{H}$ such that for any $\boldsymbol{x}$, $\left|f(\boldsymbol{x}, \boldsymbol{\theta}) - f^*(\boldsymbol{x}, \boldsymbol{\theta})\right| \leq h(\boldsymbol{x})$. Finally, (g) derives from the definition of $d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S)$. The generalization performance on target domain $\mathcal{D}_T$ therefore can be bounded using the generalization performance on source domain $\mathcal{D}_S$ as below,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_T}(f) &\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) + \mathcal{L}_{\mathcal{D}_T}(f^*) + \varepsilon_{\mathcal{D}_S} \\
&\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) + \varepsilon_{\mathcal{D}_T} + \varepsilon_{\mathcal{D}_S} \\
&\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) + \varepsilon_{\mathcal{D}}\,.
\end{aligned} \tag{9}$$

In practice, $d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S)$ is non-trial to evaluate. We therefore approximate $d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S)$ using $d_{\mathcal{H}}(T, S)$ where T and S denote the datasets of size m_T and m_S that are randomly sampled from $\mathcal{D}_T$ and $\mathcal{D}_S$, respectively. Particularly, following

the Hoeffding inequality and the assumption that $h(\boldsymbol{x}) \leq 1$, we have that

$$\Pr\left(\left|\mathbb{E}_{\mathcal{D}}[h(\boldsymbol{x})] - \frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{x}_i)\right| > t\right) \leq 2\exp(-2mt^2)\,. \tag{10}$$

Let $\delta = 2\exp(-2mt^2)$, with probability at least $1-\delta$, the following inequality holds.

$$\left|\mathbb{E}_{\mathcal{D}}[h(\boldsymbol{x})] - \frac{1}{m}\sum_{i=1}^{m} h(\boldsymbol{x}_i)\right| \leq \sqrt{\frac{\log(2/\delta)}{2m}}\,. \tag{11}$$

Similarly, the following inequality holds with probability at least $1-\delta$.

$$\begin{aligned}
&\left|\left|\mathbb{E}_{\mathcal{D}_S}[h(\boldsymbol{x})] - \mathbb{E}_{\mathcal{D}_T}[h(\boldsymbol{x})]\right| - \left|\frac{1}{m_T}\sum_{i=1}^{m_T} h(\boldsymbol{x}'_i) - \frac{1}{m_S}\sum_{i=1}^{m_S} h(\boldsymbol{x}_i)\right|\right| \\
=&\left|\left(\mathbb{E}_{\mathcal{D}_S}[h(\boldsymbol{x})] - \frac{1}{m_S}\sum_{i=1}^{m_S} h(\boldsymbol{x}_i)\right) - \left(\mathbb{E}_{\mathcal{D}_T}[h(\boldsymbol{x})] - \frac{1}{m_T}\sum_{i=1}^{m_T} h(\boldsymbol{x}'_i)\right)\right| \\
\leq&\left|\mathbb{E}_{\mathcal{D}_S}[h(\boldsymbol{x})] - \frac{1}{m_S}\sum_{i=1}^{m_S} h(\boldsymbol{x}_i)\right| + \left|\mathbb{E}_{\mathcal{D}_T}[h(\boldsymbol{x})] - \frac{1}{m_T}\sum_{i=1}^{m_T} h(\boldsymbol{x}'_i)\right| \\
\leq&\sqrt{\frac{\log(4/\delta)}{2m_S}} + \sqrt{\frac{\log(4/\delta)}{2m_T}}\,.
\end{aligned} \tag{12}$$

Based on the inequality above, we can approximate $d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S)$ using $d_{\mathcal{H}}(T, S)$ as below with probability at least $1-\delta$.

$$\begin{aligned}
d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) &= \sup_{h\in\mathcal{H}} \left|\mathbb{E}_{\mathcal{D}_S}[h(\boldsymbol{x})] - \mathbb{E}_{\mathcal{D}_T}[h(\boldsymbol{x})]\right| \\
&\leq \sup_{h\in\mathcal{H}} \left|\frac{1}{m_S}\sum_{i=1}^{m_S} h(\boldsymbol{x}_i) - \frac{1}{m_T}\sum_{i=1}^{m_T} h(\boldsymbol{x}'_i)\right| + \sqrt{\frac{\log(4/\delta)}{2m_S}} + \sqrt{\frac{\log(4/\delta)}{2m_T}} \\
&\leq d_{\mathcal{H}}(T, S) + \sqrt{\frac{\log(4/\delta)}{2m_S}} + \sqrt{\frac{\log(4/\delta)}{2m_T}}\,.
\end{aligned} \tag{13}$$

Combining the results above, with probability at least $1-\delta$, we have

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_T}(f) &\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(\mathcal{D}_T, \mathcal{D}_S) + \varepsilon_{\mathcal{D}} \\
&\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(T, S) + \alpha\sqrt{\frac{\log(4/\delta)}{2m_S}} + \alpha\sqrt{\frac{\log(4/\delta)}{2m_T}} + \varepsilon_{\mathcal{D}}\,.
\end{aligned} \tag{14}$$

Note that $\alpha = 1$ for loss function $\ell(f, y) = (f-y)^2/2$ when $f, y \in [0, 1]$. Therefore, by integrating the conclusion in Lemma A.1 with the results above, the following inequality holds with probability at least $1-2\delta$.

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}_T}(f) &\leq \mathcal{L}_{\mathcal{D}_S}(f) + \alpha d_{\mathcal{H}}(T, S) + \alpha\sqrt{\frac{\log(4/\delta)}{2m_S}} + \alpha\sqrt{\frac{\log(4/\delta)}{2m_T}} + \varepsilon_{\mathcal{D}} \\
&\leq \mathcal{L}_S(f) + 2\rho\sqrt{\frac{\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}}{m_S}} + d_{\mathcal{H}}(T, S) + \varepsilon
\end{aligned} \tag{15}$$

where $\varepsilon \triangleq 2c/\sqrt{n} + 4\sqrt{\log(4/\delta)/(2m_S)} + \sqrt{\log(4/\delta)/(2m_T)} + \varepsilon_{\mathcal{D}}$. The proof hence is concluded.

A.2. Proof of Proposition 1

We firstly give definitions for symbol $O(\cdot)$, $\Omega(\cdot)$ and $\Theta(\cdot)$.

Definition A.1 (Definition of $O(\cdot)$)**.** *If there exists constants $c, \delta > 0$ such that for all x with $0 < |x - a| < \delta$,*

$$|f(x)| \leq c \cdot g(x) \ ,$$

we can say

$$f(x) = O(g(x)) \qquad \text{as } x \to a \ .$$

Definition A.2 (Definition of $\Omega(\cdot)$)**.** *If there exists constants $c, \delta > 0$ such that for all x with $0 < |x - a| < \delta$,*

$$c \cdot g(x) \leq |f(x)| \ ,$$

we can say

$$f(x) = \Omega(g(x)) \qquad \text{as } x \to a \ .$$

Definition A.3 (Definition of $\Theta(\cdot)$)**.** *If there exists constant $c_1, c_2, \delta > 0$ such that for all x with $0 < |x - a| < \delta$,*

$$c_1 \cdot g(x) \leq |f(x)| \leq c_2 \cdot g(x) \ ,$$

we can say

$$f(x) = \Theta(g(x)) \qquad \text{as } x \to a \ .$$

To prove our Proposition 1, we then introduce Lemma A.2 based on the following assumptions. We refer to Nguyen et al. (2021) for more details.

Assumption A.1 (Data Scaling (Nguyen et al., 2021))**.** *The data distribution $\mathcal{D}(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^d$ satisfies the following properties:*

$$\int \|\boldsymbol{x}\|_2 \mathrm{d}\mathcal{D}(\boldsymbol{x}) = \Theta(\sqrt{d}) \ , \quad \int \|\boldsymbol{x}\|_2^2 \mathrm{d}\mathcal{D}(\boldsymbol{x}) = \Theta(d) \ ,$$
$$\int \left\| \boldsymbol{x} - \int \boldsymbol{x}' \mathrm{d}\mathcal{D}\left(\boldsymbol{x}'\right) \right\|_2^2 \mathrm{d}\mathcal{D}(\boldsymbol{x}) = \Omega(d) \ .$$

Assumption A.2 (Lipschitz Concentration (Nguyen et al., 2021))**.** *The data distribution $\mathcal{D}(\boldsymbol{x})$ satisfies the Lipschitz concentration property. Namely, for every Lipschitz continuous function $f : \mathbb{R}^d \to \mathbb{R}$, there exists an absolute constant $c > 0$ such that, for all $t > 0$,*

$$\mathbb{P}\left(\left| f(\boldsymbol{x}) - \int f\left(\boldsymbol{x}'\right) d\mathcal{D}\left(\boldsymbol{x}'\right) \right| > t \right) \leq 2e^{-ct^2/\|f\|_{\text{Lip}}^2}$$

Lemma A.2 (Corollary of Theorem 3.2 in (Nguyen et al., 2021))**.** *Let $\{\boldsymbol{x}_i\}_{i=1}^m$ be a set of data points randomly sampled from $\mathcal{D}$, where $\mathcal{D}$ has zero mean and satisfies the assumptions above. Let $\boldsymbol{\Theta}_0$ be the NTK matrix. Then, for any even integer constant $r \geq 2$, with probability $1 - me^{-\Omega(d)} - m^2 e^{-\Omega(dm^{-2/(r-0.5)})}$, we have*

$$\lambda_{\min}(\boldsymbol{\Theta}_0) = \Theta(d)$$

where $\lambda_{\min}(\boldsymbol{\Theta}_0)$ denotes the minimum eigenvalue of $\boldsymbol{\Theta}_0$.

For Lemma A.2 to hold with high probability, m cannot grow super-polynomially in d. We introduce an additional assumption.

Assumption A.3. *Assume that $\exists \alpha > 0$ such that $d = \Theta(m^\alpha), \forall m \in \mathbb{N}^+$.*

Then, by introducing the conclusion in Lemma A.2 into our utility function $\nu(\cdot)$, there exists a constant $\beta > 0$ such that the following holds with high probability.

$$\begin{aligned}\nu(S) &= -\kappa\sqrt{\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}/m_S} - d_{\mathcal{H}}(T,S)\\ &\geq -\kappa\sqrt{\|\widehat{\boldsymbol{y}}\|_2 \left\|\boldsymbol{\Theta}_0^{-1}\right\|_2 \|\widehat{\boldsymbol{y}}\|_2 /m_S} - d_{\mathcal{H}}(T,S)\\ &\geq -\kappa\sqrt{\lambda_{\min}(\boldsymbol{\Theta}_0)^{-1}} - d_{\mathcal{H}}(T,S)\\ &= -\kappa\Theta(d^{-1/2}) - d_{\mathcal{H}}(T,S)\\ &\geq -\kappa\beta m_S^{-\alpha/2} - d_{\mathcal{H}}(T,S)\end{aligned} \tag{16}$$

where the second inequality derives from $\|\widehat{\boldsymbol{y}}\|_2^2 \leq m_S$ with $\widehat{\boldsymbol{y}} \in [-1,1]_S^m$ based on the assumption in Theorem 1. Our proof hence is concluded.

Remark. Note that a zero-mean data distribution can typically be satisfied by subtracting the original data distribution with its expectation. Meanwhile, as justified in (Nguyen et al., 2021), Assumption A.1 are scaling conditions on the data vector $\boldsymbol{x}$, which can be easily satisfied by normalizing the whole data distribution with a certain constant. Moreover, Nguyen et al. (2021) show that many real-world distributions are able to satisfy Assumption A.2, such as the standard Gaussian distribution, the uniform distribution on the sphere, etc. The Assumption A.3 is satisfied in the high-dimentional regime. Nevertheless, we provide empirical validations in Section 6.4.

A.3. Proof of Proposition 2

We introduce the following lemma, where the first result in Lemma A.4 is adapted from (Lee et al., 2019).

Lemma A.3 (Lemma 1 of (Laurent & Massart, 2000))**.** *If* $\mathrm{x}_1, \cdots, \mathrm{x}_k$ *are independent standard normal random variables, for* $\mathrm{y} = \sum_{i=1}^k \mathrm{x}_i^2$ *and any* ϵ*,*

$$\Pr(\mathrm{y} - k \geq 2\sqrt{k\epsilon} + 2\epsilon) \leq \exp(-\epsilon)\ .$$

Lemma A.4. *Assume* $\|\boldsymbol{x}\|_2, \|\boldsymbol{x}'\|_2 \leq 1$ *and* σ *is* 1*-Lipschitz continuous and* β*-Lipschitz smooth with* $\sigma(0) = 0$*, there is a* $\rho_1, \rho_2 > 0$ *such that with a high probability, we have*

$$\begin{aligned}\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 &\leq \rho_1\\ \left\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0) - \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}', \boldsymbol{\theta}_0)\right\|_2 &\leq \rho_2 \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2\ .\end{aligned}$$

Proof. Note that $\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 \leq \rho_1$ can be adapted from the Lemma 1 in (Lee et al., 2019). We therefore only provide the proof of our second result. To ease notations, we hide the model parameter $\boldsymbol{\theta}_0$ in the following proofs and use $\nabla_g f(\boldsymbol{x})$ to denote the gradient w.r.t the output of g if g is a function. Specifically, based on the formulation of DNNs in Sec. 3.2,

$$\begin{aligned}&\left\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}) - \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}')\right\|_2^2\\ =& \sum_{l=0}^{L-1} \left\|\nabla_{\mathbf{W}^{(l)}} f(\boldsymbol{x}) - \nabla_{\mathbf{W}^{(l)}} f(\boldsymbol{x}')\right\|_{\mathrm{F}}^2\\ =& \frac{1}{\sqrt{n}} \sum_{l=0}^{L-1} \left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) \tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}') \tilde{g}^{(l)}(\boldsymbol{x}')^\top\right\|_{\mathrm{F}}^2\\ =& \frac{1}{\sqrt{n}} \sum_{l=0}^{L-1} \Big\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) \tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}') \tilde{g}^{(l)}(\boldsymbol{x})^\top +\\ &\left(\nabla_{g^{(l+1)}} f(\boldsymbol{x}') \tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}') \tilde{g}^{(l)}(\boldsymbol{x}')^\top\right)\Big\|_{\mathrm{F}}^2\\ \leq& \frac{1}{\sqrt{n}} \sum_{l=0}^{L-1} \left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) \tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}') \tilde{g}^{(l)}(\boldsymbol{x})^\top\right\|_{\mathrm{F}}^2 +\end{aligned}$$

$$
\begin{aligned}
&\left\|\left(\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\tilde{g}^{(l)}(\boldsymbol{x}')^\top\right)\right\|_{\mathrm{F}}^2 + \\
&2\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x})\tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\tilde{g}^{(l)}(\boldsymbol{x})^\top\right\|_{\mathrm{F}}\left\|\left(\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\tilde{g}^{(l)}(\boldsymbol{x})^\top - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\tilde{g}^{(l)}(\boldsymbol{x}')^\top\right)\right\|_{\mathrm{F}} \\
\leq\ & \sqrt{n}\sum_{l=0}^{L-1}\left\|\left(\nabla_{g^{(l+1)}} f(\boldsymbol{x}) - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right)\tilde{g}^{(l)}(\boldsymbol{x})^\top\right\|_2^2 + \left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\left(\tilde{g}^{(l)}(\boldsymbol{x}) - \tilde{g}^{(l)}(\boldsymbol{x}')\right)^\top\right\|_2^2 + \\
&2\left\|\left(\nabla_{g^{(l+1)}} f(\boldsymbol{x}) - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right)\tilde{g}^{(l)}(\boldsymbol{x})^\top\right\|_2\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\left(\tilde{g}^{(l)}(\boldsymbol{x}) - \tilde{g}^{(l)}(\boldsymbol{x}')\right)^\top\right\|_2 \\
\leq\ & \sqrt{n}\sum_{l=0}^{L-1}\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right\|_2^2\left\|\tilde{g}^{(l)}(\boldsymbol{x})\right\|_2^2 + \left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right\|_2^2\left\|\tilde{g}^{(l)}(\boldsymbol{x}) - \tilde{g}^{(l)}(\boldsymbol{x}')\right\|_2^2 + \\
&2\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right\|_2^2\left\|\tilde{g}^{(l)}(\boldsymbol{x})\right\|_2\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right\|_2^2\left\|\tilde{g}^{(l)}(\boldsymbol{x}) - \tilde{g}^{(l)}(\boldsymbol{x}')\right\|_2
\end{aligned} \tag{17}
$$

where the second inequality derives from $\|\cdot\|_{\mathrm{F}} \leq \sqrt{n}\|\cdot\|_2$ and triangle inequality of matrix norm. Consequently, we only need to bound each term in the last inequality above.

Based on Lemma A.3, with probability at least $1 - L\delta$, for all $l \in [0, L-1]$, the following inequality holds.

$$
\begin{aligned}
\left\|\tilde{g}^{(l)}(\boldsymbol{x})\right\|_2 = \left\|\sigma(g^{(l)}(\boldsymbol{x}))\right\|_2 &\leq \left\|g^{(l)}(\boldsymbol{x})\right\|_2 \leq \frac{1}{\sqrt{n}}\left\|\mathbf{W}^{(l-1)}\right\|_{\mathrm{F}}\left\|\tilde{g}^{(l-1)}(\boldsymbol{x})\right\|_2 \\
&\leq \frac{1}{\sqrt{n}}\sqrt{n^2 + 2\sqrt{n^2}\log(1/\delta) + 2\log(1/\delta)}\left\|\tilde{g}^{(l-1)}(\boldsymbol{x})\right\|_2 \\
&= \sqrt{n + 2\log(1/\delta) + 2\log(1/\delta)/n}\left\|\tilde{g}^{(l-1)}(\boldsymbol{x})\right\|_2 \\
&\leq \left(\sqrt{n + 2\log(1/\delta) + 2\log(1/\delta)/n}\right)^l \|\boldsymbol{x}\|_2 \\
&\leq \left(\sqrt{n + 2\log(1/\delta) + 2\log(1/\delta)/n}\right)^l
\end{aligned} \tag{18}
$$

Similarly, with probability at least $1 - L\delta$, for all $l \in [0, L-1]$, the following inequality holds.

$$
\begin{aligned}
\left\|\tilde{g}^{(l)}(\boldsymbol{x}) - \tilde{g}^{(l)}(\boldsymbol{x}')\right\|_2 &= \left\|\sigma(g^{(l)}(\boldsymbol{x})) - \sigma(g^{(l)}(\boldsymbol{x}'))\right\|_2 \\
&\leq \left\|g^{(l)}(\boldsymbol{x}) - g^{(l)}(\boldsymbol{x}')\right\|_2 \\
&= \frac{1}{\sqrt{n}}\left\|\mathbf{W}^{(l-1)}\left(\tilde{g}^{(l-1)}(\boldsymbol{x}) - \tilde{g}^{(l-1)}(\boldsymbol{x}')\right)\right\|_2 \\
&\leq \frac{1}{\sqrt{n}}\left\|\mathbf{W}^{(l-1)}\right\|_{\mathrm{F}}\left\|\tilde{g}^{(l-1)}(\boldsymbol{x}) - \tilde{g}^{(l-1)}(\boldsymbol{x}')\right\|_2 \\
&\leq \left(\sqrt{n + 2\log(1/\delta) + 2\log(1/\delta)/n}\right)^l \|\boldsymbol{x} - \boldsymbol{x}'\|_2
\end{aligned} \tag{19}
$$

Let $\odot$ denote the element-wise multiplication of two vectors and $\dot{\sigma}(\cdot)$ denote the derivative of activation function $\sigma(\cdot)$. With probability at least $1 - L\delta$, for all $l \in [0, L-1]$, the following inequalities hold.

$$
\begin{aligned}
\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x})\right\|_2 &= \left\|\frac{1}{\sqrt{n}}\left(\mathbf{W}^{(l+1)}\right)^\top\left(\nabla_{g^{(l+2)}} f(\boldsymbol{x}) \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x}))\right)\right\|_2 \\
&\leq \left(\sqrt{n + 2\log(1/\delta) + 2\log(1/\delta)/n}\right)^{L-l-1}
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
&\left\|\nabla_{g^{(l+1)}} f(\boldsymbol{x}) - \nabla_{g^{(l+1)}} f(\boldsymbol{x}')\right\|_2 \\
&= \left\| \frac{1}{\sqrt{n}} \left(\mathbf{W}^{(l+1)}\right)^\top \left(\nabla_{g^{(l+2)}} f(\boldsymbol{x}) \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x}')) \right) \right\|_2 \\
&= \left\| \frac{1}{\sqrt{n}} \left(\mathbf{W}^{(l+1)}\right)^\top \left(\nabla_{g^{(l+2)}} f(\boldsymbol{x}) \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) \right) + \right. \\
&\quad \left. \frac{1}{\sqrt{n}} \left(\mathbf{W}^{(l+1)}\right)^\top \left(\nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x}')) \right) \right\|_2 \\
&\leq \frac{1}{\sqrt{n}} \left\|\mathbf{W}^{(l+1)}\right\|_{\mathrm{F}} \left\| \nabla_{g^{(l+2)}} f(\boldsymbol{x}) \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) \right\|_2 + \\
&\quad \frac{1}{\sqrt{n}} \left\|\mathbf{W}^{(l+1)}\right\|_{\mathrm{F}} \left\| \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x})) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \odot \dot{\sigma}(g^{(l+2)}(\boldsymbol{x}') \right\|_2 \\
&\leq \frac{1}{\sqrt{n}} \left\|\mathbf{W}^{(l+1)}\right\|_{\mathrm{F}} \left(\left\| \nabla_{g^{(l+2)}} f(\boldsymbol{x}) - \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \right\|_2 + \beta \left\| \nabla_{g^{(l+2)}} f(\boldsymbol{x}') \right\|_2 \left\| g^{(l+2)}(\boldsymbol{x}) - g^{(l+2)}(\boldsymbol{x}') \right\|_2 \right) \\
&= O(\left\|\boldsymbol{x} - \boldsymbol{x}'\right\|)
\end{aligned}
\tag{21}
$$

where the last inequality can be derived by using (18), (19) and (20).

Finally, by introducing (18), (19), (20) and (21) into (17), there exist a constant ρ_2, with a high probability, we have

$$
\left\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0) - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}')\right\|_2 \leq \rho_2 \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2 . \tag{22}
$$

□

We can now prove our Proposition 2. To ease notations, we also hide the model parameter $\boldsymbol{\theta}_0$ in the following proofs. Specifically, in the case of noisy data $\boldsymbol{x}_\epsilon$ (or $\boldsymbol{x}'_\epsilon$) based on $\boldsymbol{x}$ (or $\boldsymbol{x}$), we have $\|\boldsymbol{x} - \boldsymbol{x}_\epsilon\|_2 \leq \epsilon$ (or $\left\|\boldsymbol{x}' - \boldsymbol{x}'_\epsilon\right\|_2 \leq \epsilon$). Let $\boldsymbol{\Theta}_{0,\epsilon}$ be the NTK matrix induced by dataset with noise ϵ and $\widehat{y}_\epsilon = y - f(\boldsymbol{x}_\epsilon, \boldsymbol{\theta}_0)$, we hence have

$$
\begin{aligned}
\left| \widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1} \widehat{\boldsymbol{y}} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \widehat{\boldsymbol{y}}_\epsilon \right| &= \left| \widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1} \widehat{\boldsymbol{y}} - \boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1} \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1} \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \boldsymbol{y} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \widehat{\boldsymbol{y}}_\epsilon \right| \\
&\leq \left| \widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1} \widehat{\boldsymbol{y}} - \boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1} \boldsymbol{y} \right| + \left| \boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1} \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \boldsymbol{y} \right| + \left| \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \boldsymbol{y} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1} \widehat{\boldsymbol{y}}_\epsilon \right| .
\end{aligned}
\tag{23}
$$

With Lemma A.4, we have

$$
\begin{aligned}
\left| \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\Theta}_0(\boldsymbol{x}_\epsilon, \boldsymbol{x}'_\epsilon) \right| &= \left| \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}'_\epsilon) + \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}'_\epsilon) - \boldsymbol{\Theta}_0(\boldsymbol{x}_\epsilon, \boldsymbol{x}'_\epsilon) \right| \\
&\leq \left| \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}'_\epsilon) \right| + \left| \boldsymbol{\Theta}_0(\boldsymbol{x}, \boldsymbol{x}'_\epsilon) - \boldsymbol{\Theta}_0(\boldsymbol{x}_\epsilon, \boldsymbol{x}'_\epsilon) \right| \\
&= \left| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)^\top \left(\nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}') - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}'_\epsilon) \right) \right| + \left| \left(\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0) - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}_\epsilon) \right)^\top \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}'_\epsilon) \right| \\
&\leq \|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)\|_2 \left\| \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}') - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}'_\epsilon) \right\|_2 + \left\| \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0) - \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}_\epsilon) \right\|_2 \left\| \nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x}'_\epsilon) \right\|_2 \\
&= \rho_1 \rho_2 \left(\|\boldsymbol{x} - \boldsymbol{x}_\epsilon\|_2 + \left\|\boldsymbol{x}' - \boldsymbol{x}'_\epsilon\right\|_2 \right) \\
&= 2\rho_1 \rho_2 \epsilon .
\end{aligned}
\tag{24}
$$

Let $\lambda_{\min}$ and $\lambda_{\min,\epsilon}$ be the smallest eigenvalues of $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_{0,\epsilon}$, respectively. Consequently, for dataset S of size m_S with

$\boldsymbol{y} \in [0,1]^{m_S}$,

$$\begin{aligned}
\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_{0,\epsilon}\|_2 &\leq \|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_{0,\epsilon}\|_{\mathrm{F}} \leq 2\rho_1\rho_2 m_S \epsilon \\
\left|\boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1}\boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\boldsymbol{y}\right| &= \left|\boldsymbol{y}^\top \left(\boldsymbol{\Theta}_0^{-1} - \boldsymbol{\Theta}_{0,\epsilon}^{-1}\right)\boldsymbol{y}\right|_2 \\
&\leq \left\|\boldsymbol{\Theta}_0^{-1} - \boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_2 \|\boldsymbol{y}\|_2^2 \\
&\leq \left\|\boldsymbol{\Theta}_0^{-1}\left(\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_{0,\epsilon}\right)\boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_2 \|\boldsymbol{y}\|_2^2 \\
&\leq \left\|\boldsymbol{\Theta}_0^{-1}\right\|_2 \left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_{0,\epsilon}\right\|_2 \left\|\boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_2 \|\boldsymbol{y}\|_2^2 \\
&\leq 2\rho_1\rho_2 m_S \|\boldsymbol{y}\|_2^2 \epsilon/(\lambda_{\min}\lambda_{\min,\epsilon}) \\
&\leq 2\rho_1\rho_2 m_S^2 \epsilon/(\lambda_{\min}\lambda_{\min,\epsilon})\,.
\end{aligned} \tag{25}$$

Since $\boldsymbol{\Theta}_0$ is symmetric, we can also represent $\boldsymbol{\Theta}_0$ as $\boldsymbol{\Theta}_0 = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$ using principal component analysis (PCA), where $\mathbf{V}$ and $\boldsymbol{\Lambda}$ denotes the matrix of eigenvectors $\{\boldsymbol{v}_i\}_{i=1}^m$ and eigenvalues $\{\lambda_i\}_{i=1}^m$, respectively. Based on the assumption that $|f(\boldsymbol{x},\boldsymbol{\theta}_0)| \leq \tau$ for any $\boldsymbol{x}$ and $\lambda_i > 0$ we have the following inequality:

$$\begin{aligned}
\left|\boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1}\boldsymbol{y} - \widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}\right| &= \left|\sum_{i=1}^{m_S} \lambda_i^{-1}(\boldsymbol{v}_i^\top \boldsymbol{y})^2 - \sum_{i=1}^{m_S} \lambda_i^{-1}(\boldsymbol{v}_i^\top \widehat{\boldsymbol{y}})^2\right| \\
&\leq \sum_{i=1}^{m_S} \lambda_i^{-1}\left|(\boldsymbol{v}_i^\top \boldsymbol{y})^2 - (\boldsymbol{v}_i^\top \widehat{\boldsymbol{y}})^2\right| \\
&\leq \sum_{i=1}^{m_S} \lambda_i^{-1}\left|\boldsymbol{v}_i^\top(\boldsymbol{y}\boldsymbol{y}^\top - \widehat{\boldsymbol{y}}\widehat{\boldsymbol{y}}^\top)\boldsymbol{v}_i\right| \\
&\leq \sum_{i=1}^{m_S} \lambda_i^{-1}\|\boldsymbol{v}_i\|_2^2 \left\|\boldsymbol{y}\boldsymbol{y}^\top - \widehat{\boldsymbol{y}}\widehat{\boldsymbol{y}}^\top\right\|_{\mathrm{F}} \\
&\leq \sum_{i=1}^{m_S} \lambda_i^{-1}\left\|\boldsymbol{y}\boldsymbol{y}^\top - \widehat{\boldsymbol{y}}\widehat{\boldsymbol{y}}^\top\right\|_{\mathrm{tr}} \\
&\leq m_S \left\|\boldsymbol{\Theta}_0^{-1}\right\|_{\mathrm{tr}} \tau(2+\tau)
\end{aligned} \tag{26}$$

where the last inequality can be derived by exploiting the fact that $\left\|\boldsymbol{\Theta}_0^{-1}\right\|_{\mathrm{tr}} = \sum_{i=1}^m \lambda_i^{-1}$ and

$$\left|y^2 - \widehat{y}^2\right| = \left|y^2 - (y - f(\boldsymbol{x},\boldsymbol{\theta}_0))^2\right| \leq |f(\boldsymbol{x},\boldsymbol{\theta}_0)|\,|2y - f(\boldsymbol{x},\boldsymbol{\theta}_0)| \leq \tau(2+\tau)\,. \tag{27}$$

Similarly, we also have

$$\left|\boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\boldsymbol{y} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon\right| \leq m_S \left\|\boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_{\mathrm{tr}} \tau(2+\tau)\,. \tag{28}$$

By introduce the results above into (23), we have

$$\begin{aligned}
\left|\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon\right| &\leq \left|\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}} - \boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1}\boldsymbol{y}\right| + \left|\boldsymbol{y}^\top \boldsymbol{\Theta}_0^{-1}\boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\boldsymbol{y}\right| + \left|\boldsymbol{y}^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\boldsymbol{y} - \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon\right| \\
&\leq m_S\left(\left\|\boldsymbol{\Theta}_0^{-1}\right\|_{\mathrm{tr}} + \left\|\boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_{\mathrm{tr}}\right)\tau(2+\tau) + 2\rho_1\rho_2 m_S^2 \epsilon/(\lambda_{\min}\lambda_{\min,\epsilon})\,.
\end{aligned} \tag{29}$$

Since $\min(\widehat{\boldsymbol{y}}^\top \boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}, \widehat{\boldsymbol{y}}_\epsilon^\top \boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon) \geq \gamma$ and $\Delta(S, S_\epsilon) \triangleq |d_{\mathcal{H}}(T,S) - d_{\mathcal{H}}(T,S_\epsilon)|$, let $\beta = 2\rho_1\rho_2$, we then have the following

inequalities:

$$
\begin{aligned}
|\nu(S)-\nu(S_\epsilon)| &\le \kappa\left|\sqrt{\widehat{\boldsymbol{y}}_\epsilon^\top\boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon/m_S}-\sqrt{\widehat{\boldsymbol{y}}^\top\boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}/m_S}\right|+\Delta(S,S_\epsilon)\\
&\le \frac{\kappa}{2\sqrt{m_S\gamma}}\left|\widehat{\boldsymbol{y}}^\top\boldsymbol{\Theta}_0^{-1}\widehat{\boldsymbol{y}}-\widehat{\boldsymbol{y}}_\epsilon^\top\boldsymbol{\Theta}_{0,\epsilon}^{-1}\widehat{\boldsymbol{y}}_\epsilon\right|+\Delta(S,S_\epsilon)\\
&\le \frac{\kappa}{2\sqrt{\gamma}}\left(\sqrt{m_S}\left(\left\|\boldsymbol{\Theta}_0^{-1}\right\|_{\text{tr}}+\left\|\boldsymbol{\Theta}_{0,\epsilon}^{-1}\right\|_{\text{tr}}\right)\tau(2+\tau)+2\rho_1\rho_2 m_S\sqrt{m_S}\epsilon/(\lambda_{\min}\lambda_{\min,\epsilon})\right)+\Delta(S,S_\epsilon)\\
&\le \frac{\kappa}{2\sqrt{\gamma}}\left(O(\tau)+\beta m_S\sqrt{m_S}\epsilon/\lambda^2\right)+\Delta(S,S_\epsilon)
\end{aligned}
\tag{30}
$$

where the second inequality derives from the Lipschitz continuity of function $\sqrt{x}$ when $x>\gamma$. The last equality is based on the definition of $O(\cdot)$ since τ is small and the assumption that $\lambda_{\min},\lambda_{\min,\epsilon}>\lambda$. The proof hence is finally concluded.

Remark. Note that τ indeed will be small in practice as validated in Appendix E.2.

A.4. Proof of Proposition 3

Let $\boldsymbol{\Theta}_{0,f}$ be the NTK matrix induced by the dataset on model f and $\widehat{y}_f=y-f(\boldsymbol{x}_\epsilon,\boldsymbol{\theta}_0)$. Following the same principle shown in the proof of Proposition 2, we have

$$
\begin{aligned}
\left|\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f-\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}\right| &= \left|\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f-\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f}^{-1}\boldsymbol{y}+\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f}^{-1}\boldsymbol{y}-\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\boldsymbol{y}+\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\boldsymbol{y}-\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}\right|\\
&\le \left|\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f-\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f}^{-1}\boldsymbol{y}\right|+\left|\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f}^{-1}\boldsymbol{y}-\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\boldsymbol{y}\right|+\left|\boldsymbol{y}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\boldsymbol{y}-\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}\right|\\
&\le m_S\left(\left\|\boldsymbol{\Theta}_{0,f}^{-1}\right\|_{\text{tr}}+\left\|\boldsymbol{\Theta}_{0,f'}^{-1}\right\|_{\text{tr}}\right)\tau(2+\tau)+\left\|\boldsymbol{\Theta}_{0,f}^{-1}\right\|_2\left\|\boldsymbol{\Theta}_{0,f}-\boldsymbol{\Theta}_{0,f'}\right\|_2\left\|\boldsymbol{\Theta}_{0,f'}^{-1}\right\|_2\|\boldsymbol{y}\|_2^2\\
&\le m_S\left(\left\|\boldsymbol{\Theta}_{0,f}^{-1}\right\|_{\text{tr}}+\left\|\boldsymbol{\Theta}_{0,f'}^{-1}\right\|_{\text{tr}}\right)\tau(2+\tau)+m_S\epsilon/(\lambda_{\min,f}\lambda_{\min,f'})\,.
\end{aligned}
\tag{31}
$$

Based on the assumption that $\min\left(\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f,\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}\right)\ge\gamma$, $\min(\lambda_{\min,f},\lambda_{\min,f'})>\lambda$ and the fact that $d_{\mathcal{H}}(T,S)$ is independent from f and f', we have

$$
\begin{aligned}
|\nu(S;f)-\nu(S;f')| &= \left|\sqrt{\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f/m_S}-\sqrt{\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}/m_S}\right|\\
&\le \frac{\kappa}{2\sqrt{m_S\gamma}}\left|\widehat{\boldsymbol{y}}_f^\top\boldsymbol{\Theta}_{0,f}^{-1}\widehat{\boldsymbol{y}}_f-\widehat{\boldsymbol{y}}_{f'}^\top\boldsymbol{\Theta}_{0,f'}^{-1}\widehat{\boldsymbol{y}}_{f'}\right|\\
&\le \frac{\kappa}{2\sqrt{\gamma}}\left(\sqrt{m_S}\left(\left\|\boldsymbol{\Theta}_{0,f}^{-1}\right\|_{\text{tr}}+\left\|\boldsymbol{\Theta}_{0,f'}^{-1}\right\|_{\text{tr}}\right)\tau(2+\tau)+\sqrt{m_S}\epsilon/(\lambda_{\min}\lambda_{\min,\epsilon})\right)\\
&\le \frac{\kappa}{2\sqrt{\gamma}}\left(O(\tau)+\sqrt{m_S}\epsilon/\lambda^2\right)\,,
\end{aligned}
\tag{32}
$$

which concludes our proof.

B. Efficient Approximations of $\boldsymbol{\Theta}_0$

The NTK matrix $\boldsymbol{\Theta}_0\in\mathbb{R}_{m\times m}$ enjoys an elegant definition of $\boldsymbol{\Theta}(\boldsymbol{x},\boldsymbol{x}')=\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x})\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x}')^\top$ (Equation 2). However, we face several difficulties in implementing it in practice. In this section, we mainly address the empirical approximations of the NTK due to memory and time constraints. The invertibility guarantee of $\boldsymbol{\Theta}_0$ is given in Appendix B.4.

Generally, computing $\boldsymbol{\Theta}_0$ for a dataset S requires pairwise products for the set of gradients $\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x})$ for all $\boldsymbol{x}\in\{\boldsymbol{x}_i,\ldots,\boldsymbol{x}_m\}$ data samples in the dataset S. Each gradient term $\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x})$ with respect to the parameters $\boldsymbol{\theta}\in\mathbb{R}^p$ is a vector of length p.

A straightforward implementation involves on-the-fly evaluations of $\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x})$ and $\nabla_{\boldsymbol{\theta}}f(\boldsymbol{x}')$ for each NTK entry $\boldsymbol{\Theta}(\boldsymbol{x},\boldsymbol{x}')$. This requires $2m^2$ gradient evaluations naively, which reduces to m^2 evaluations by calculating in a row-wise manner. Since $\boldsymbol{\Theta}_0$ is symmetric, we effectively require $\frac{1}{2}m^2$ evaluations. However, this method still contains repeated evaluations and yet

could not make use of potential efficient parallelism implementations in popular machine learning platforms like PyTorch and TensorFlow. In practice, each gradient evaluation takes a fraction of seconds, even computing the NTK for a dataset with 1K samples could easily take up to 1 GPU day and thus become clearly infeasible.

A potential solution would be calculating each gradient once and storing the gradients in memory to avoid repeated evaluations. This potentially reduces the computational requirement to m gradient evaluations. However, video RAM of an ordinary GPU could pose constraints for time-efficient exact NTK evaluation because it requires storing an $m \times p$ gradients matrix. In our baselines experiments, for example, $S_{\mathcal{A}}$ has size $m = 10750$ and ResNet18 has $p \approx 11.2 \times 10^6$ parameters, such a gradients matrix easily exceeds the limit of commercial GPUs.

One workaround would be using a fast Solid-State Drive (SSD) for storage. However, we usually consider I/O operations like reading from and writing to disks as slow operations unless specialized hardware is used. We do not use this method in our implementation. We instead consider a batched evaluation technique that circumvents the memory constraint.

B.1. Diagonal Block Approximation

We approximate $\mathbf{\Theta}_0$ with a matrix containing only the diagonal $l \times l$-sized blocks, where each block only involves a batch of l data samples. For simplicity, we assume l is a factor of m. We first define such block matrices,

$$B_{i,i}^{(l)} = \begin{pmatrix} \mathbf{\Theta}(\boldsymbol{x}_{(i-1)l+1}, \boldsymbol{x}_{(i-1)l+1}) & \dots & \mathbf{\Theta}(\boldsymbol{x}_{(i-1)l+1}, \boldsymbol{x}_{il}) \\ \vdots & \ddots & \vdots \\ \mathbf{\Theta}(\boldsymbol{x}_{il}, \boldsymbol{x}_{(i-1)l+1}) & \dots & \mathbf{\Theta}(\boldsymbol{x}_{il}, \boldsymbol{x}_{il}) \end{pmatrix} \in \mathbb{R}^{l \times l}.$$

Then, filling the off-diagonal blocks with zeros, the diagonal block approximation of $\mathbf{\Theta}_0$ is defined as follows,

$$\mathbf{\Theta}_0^{(l),blocked} = \begin{pmatrix} B_{1,1}^{(l)} & & & \mathbf{0} \\ & B_{2,2}^{(l)} & & \\ & & \ddots & \\ \mathbf{0} & & & B_{m/l,m/l}^{(l)} \end{pmatrix}.$$

For example, by setting the batch size to 2, the diagonal block approximation of $\mathbf{\Theta}_0$ is

$$\mathbf{\Theta}_0^{(2),blocked} = \begin{pmatrix} \underbrace{\boxed{\begin{matrix} \mathbf{\Theta}(\boldsymbol{x}_1, \boldsymbol{x}_1) & \mathbf{\Theta}(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ \mathbf{\Theta}(\boldsymbol{x}_2, \boldsymbol{x}_1) & \mathbf{\Theta}(\boldsymbol{x}_2, \boldsymbol{x}_2) \end{matrix}}}_{\text{block size } l=2} & & & \mathbf{0} \\ & \boxed{\begin{matrix} \mathbf{\Theta}(\boldsymbol{x}_3, \boldsymbol{x}_3) & \mathbf{\Theta}(\boldsymbol{x}_3, \boldsymbol{x}_4) \\ \mathbf{\Theta}(\boldsymbol{x}_4, \boldsymbol{x}_3) & \mathbf{\Theta}(\boldsymbol{x}_4, \boldsymbol{x}_4) \end{matrix}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\begin{matrix} \mathbf{\Theta}(\boldsymbol{x}_{m-1}, \boldsymbol{x}_{m-1}) & \mathbf{\Theta}(\boldsymbol{x}_{m-1}, \boldsymbol{x}_m) \\ \mathbf{\Theta}(\boldsymbol{x}_m, \boldsymbol{x}_{m-1}) & \mathbf{\Theta}(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{matrix}} \end{pmatrix}.$$

Evaluation of each block now only requires storing a total of l sample gradients. Also, inverting a block diagonal matrix can be easily performed by inverting each block separately,

$$\left[\mathbf{\Theta}_0^{(l),blocked}\right]^{-1} = \begin{pmatrix} \left[B_{1,1}^{(l)}\right]^{-1} & & & \mathbf{0} \\ & \left[B_{2,2}^{(l)}\right]^{-1} & & \\ & & \ddots & \\ \mathbf{0} & & & \left[B_{m/l,m/l}^{(l)}\right]^{-1} \end{pmatrix}.$$

Therefore, the inverse of the analytic NTK matrix in Equation 3 can be efficiently computed even when the dataset size m is large.

Remark. Fully parallelizing the batch in computing the gradients could gain us significant speedups. It is possible because there is no data dependency between the gradients of each sample. PyTorch *autograd_hacks* library offers such efficient per-sample gradient implementation for networks containing only linear and convolutional layers. PyTorch recently introduced *vmap* that officially supports per-sample gradient computation, which can potentially further improve our results shown in Table 1. Tensorflow *vectorized_map* method also provides us with similar features. We adopt the PyTorch *autograd_hacks* implementation in this paper.

Empirical investigation. We show the effectiveness and efficiency of diagonal block approximations. Using the ising model regression baseline dataset in Section 6, we vary the number of blocks used in the method and find that this method gives very good approximations. As seen in Table 4, increasing the number of blocks does not degrade the overall correlation with the ground truth much. As for efficiency, using a moderate batch size that fits into the GPU memory better utilizes the parallel computations of samples in a batch while reducing the number of batch evaluations required. Therefore, we recommend using a large batch size that fits into the GPU memory when using diagonal block approximations.

Table 4. Correlation with ground truth and efficiency of DAVINZ approximated using the diagonal block approximation. Each correlation coefficient is reported with the mean and standard error over 5 independent evaluations. Each cost includes the evaluation of 11 scores for LOO over 10 different datasets.

No. of Blocks	**Ising Physical Model Dataset**		
	Pearson	Spearman	Comp. Cost
m/l	r	ρ	(Min.)
1	0.996±0.001	0.905±0.015	1.7
50	0.994±0.001	0.908±0.014	0.5
500	0.992±0.001	0.901±0.019	1.1

B.2. Diagonal Block Approximation with Permutations

Diagonal block approximation preserves sample correlations within a batch but disregards any inter-batch sample correlations. We propose to permute the data sample orders in a dataset before performing diagonal block approximations and average the results over several trials. In doing so, we incorporate sample correlations at the expense of computational cost. The total computational cost scales linearly with the number of permutations we consider.

Empirical investigation. We again use the ising model regression baseline dataset in Section 6 and vary the number of permutations. As shown in Table 5, the computational cost scales linearly with the number of permutations as expected and both the Pearson's and Spearman's correlation stay relatively unchanged with permutations. This suggests that diagonal block approximation without permutations is sufficient for our purpose of data valuation using the NTK generalization lower bound.

Table 5. Correlation with ground truth and efficiency of DAVINZ approximated using the diagonal block approximation with permutations. Each correlation coefficient is reported with the mean and standard error over 5 independent evaluations. Each cost includes the evaluation of 11 scores for LOO over 10 different datasets.

No. of Permutations	**No. of Blocks**	**Ising Physical Model Dataset**		
		Pearson	Spearman	Comp. Cost
	m/l	r	ρ	(Min.)
1	50	0.994±0.001	0.908±0.014	0.5
10	50	0.995±0.001	0.925±0.015	1.9
50	50	0.994±0.001	0.922±0.020	15.6

B.3. L-block-banded Block Matrix Approximation

A natural extension of the diagonal block approximation is to include off-diagonal blocks. With more exactly evaluated blocks in the matrix, we hope to better approximate $\mathbf{\Theta}_0$ when computational resources allow. We define the L-block-banded matrix as

$$\mathbf{M} = \begin{pmatrix} & & \mathbf{0} \\ & M_{ij} \neq \mathbf{0} & \\ & |i-j| \leq L & \\ \mathbf{0} & & \end{pmatrix},$$

where M_{ij} is a square block matrix on the i-th row and j-th column. However, assuming a positive definite matrix $\mathbf{P}$ and an L-block-banded matrix $\mathbf{M}$ such that all L-block-banded entries $M_{ij} = P_{ij} : |i-j| \leq L$, then this L-block-banded matrix $\mathbf{M}$ is not necessarily still positive definite. Therefore, naively using an L-block-banded version of $\mathbf{\Theta}_0$ in the data valuation score calculation is not appropriate.

As a resolution, Asif & Moura (2005) study the properties of a positive definite L-block-banded matrix and proves recursive structures in its dense inverse. Specifically, in Theorem 3 of (Asif & Moura, 2005), the blocks outside the L-block band are fully specified by blocks within the L-block band if the matrix inverse is a positive definite L-block-banded matrix. Therefore, assuming a positive definite $\bar{\mathbf{\Theta}}_0^{-1}$, we are only required to evaluate the entries in the L-block band of $\bar{\mathbf{\Theta}}_0$ such that $[\bar{\mathbf{\Theta}}_0]_{ij} = [\mathbf{\Theta}_0]_{ij}$. Then, a dense $\bar{\mathbf{\Theta}}_0$ can be fully specified and positive definiteness is preserved. Additionally, getting the inverse $\bar{\mathbf{\Theta}}_0^{-1}$ does not require inverting the whole matrix even when the dataset size m is large. The details of the fast inversion algorithm could be found in Algorithm 1 of (Asif & Moura, 2005). This method also potentially provides an accuracy and computation trade-off in specifying the length of the band L to evaluate. We leave this method as a future exploration direction for finer approximations because the diagonal block approximation we used in the implementation of this paper is accurate enough for our purposes.

B.4. Invertibility of $\mathbf{\Theta}_0$

To ensure an invertible NTK matrix $\mathbf{\Theta}_0$, we replace it with the expectation of its noisy counterpart in practice. In particular, given a sufficiently small constant $\zeta > 0$ and a DNN model f, for each sample $(\boldsymbol{x}, y) \in S$, we apply an independent Gaussian noise $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \zeta\mathbf{I})$ to the gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta}_0)$ and then obtain a corresponding noisy NTK matrix based on these noisy gradients. Interestingly, such a noise usually will be introduced into the model training of DNNs by the widely adopted *stochastic gradient descent* (SGD) algorithm in practice. Finally, denote $\mathbf{Z}$ as our introduced noise to NTK and define $\widehat{\mathbf{\Theta}}_0 \triangleq \mathbb{E}_{\mathbf{Z}}[\mathbf{\Theta}_0 + \mathbf{Z}]$, we have that

$$\widehat{\mathbf{\Theta}}_0 = \mathbb{E}_{\mathbf{Z}}[\mathbf{\Theta}_0 + \mathbf{Z}] = \mathbf{\Theta}_0 + \zeta\mathbf{I}\,, \tag{33}$$

which is therefore guaranteed to be invertible. Moreover, as $\zeta \to 0$, we have $\widehat{\mathbf{\Theta}}_0 \to \mathbf{\Theta}_0$. Consequently, $\widehat{\mathbf{\Theta}}_0$ is reasonably good to be applied to approximate the true generalization error in (3) as long as ζ is sufficiently small.

C. Additional Literature

There have been attempts to directly model the scoring function using advances in deep learning and reinforcement learning. We outline these approaches and describe the difference between them and our work below.

Data valuation using reinforcement learning (DVRL). DVRL (Yoon et al., 2020) integrates data valuation with the training process of the target predictive model and utilizes reinforcement signals to train a network for data valuation. DVRL is relatively efficient as it only requires one training of the valuation network. However, the data value outputted by the network now measures how likely a datum will be used in training the predictive model, which is not directly related to the contributions of data points in achieving a high-performing predictive model. As a result, the paper only conducted experiments using the data values in the form of ranks, instead of relative dataset values. Methods to be discussed in this paper should produce more informative real value scores that make relative-contribution-based value comparisons.

Learning the scoring function. Two concurrent works (Wang et al., 2021b;a) propose to directly learn the scoring function using a utility ML model, often implemented using a 3-layer MLP for individual data samples or the DeepSets (Zaheer et al., 2017) for groups of data samples. However, learning the utility ML model itself requires abundant utility (i.e., score) samples obtained through multiple complete re-trainings of the predictive model, which is prohibitively costly for complex

models like deep neural networks. Consequently, the authors use a small proxy model (usually a logistic regression classifier) in place of the neural networks as the predictive model. Therefore, these works learn the scoring function on the logistic regression proxy and compare the relative contributions of data to learning a logistic regression model. Our method instead focuses on estimating the dataset values using relative contributions of data in learning the original deep neural network predictive model. We focus on comparing with training-free data valuation methods in this paper.

D. Experimental Setup

D.1. Dataset, Model and Ground Truth

We offer a comprehensive comparison between our method and the existing training-based and training-free baselines using common benchmark datasets. All experiments have been run on a server with Intel(R) Xeon(R)@ 2.20GHz processors and 512GB RAM. One Tesla V100 GPU is used for the experiments. We use MNIST and CIFAR-10 for classification tasks and the ising physical model dataset (Mills & Tamblyn, 2019) for regression tasks.

All MNIST and CIFAR-10 images are pre-processed by re-scaling the pixel value to the $[0, 1]$ range. We split the images into 10 datasets each containing images of a single label class. The number of data samples in each dataset also varies from 1000 to 1250. The validation dataset contains 10K images from all class labels. This baseline setup mimics the practical scenario where a particular agent only have access to a specific type of data (e.g., bank credit records from a single geographic region or MRI medical scan images from a machine of a specific brand) and each agent's dataset size varies slightly to further increase the difficulty of this data valuation baseline task. The validation objective is, however, learning a grand model performs well on all label classes of this classification problem (e.g., a model that works for banks in different regions and MRI scans from machines of different brands included in the training data). Details of the data split can be found in Table 6.

Table 6. Dataset split details for classification baseline comparisons.

Dataset	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
No. Samples	1000	1000	1000	1000	1000	1050	1100	1150	1200	1250
Labels Class	0	1	2	3	4	5	6	7	8	9

The ising physical model dataset aims to predict system energy based on an ising array of atomic spin states. The labels are also pre-processed by re-scaling the energy values to the $[0, 1]$ range. We split the input data into 10 datasets to simulate another practical scenario where agents of varying capability collaborate on a machine learning task (e.g., large companies could provide more data samples whereas the smaller companies contribute less). All agents have a similar data distribution. Details of the data split can be found in Table 7.

Table 7. Dataset split details for regression baseline comparisons.

Dataset	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
No. Samples	12	25	50	100	200	400	800	1600	3200	6400

To facilitate reproducible baseline comparisons, we select commonly used model architecture to evaluate. For classification tasks, we choose (1) the ResNet18 (He et al., 2016) and (2) the VGG13 (Simonyan & Zisserman, 2015). For regression tasks, we choose (1) an MLP with 10 layers and (2) a CNN with 6 convolutional layers and 2 fully connected layers. We remove the bias term for all layers. For DAVINZ calculations, we additionally turn off the Batch Normalization layer to remove its effect on NTK evaluations.

For ground truth, we use leave-one-out validation performance as the metric. In practice, we experience problems finding the true validation performance given a model architecture and a dataset, since the choice of hyperparameters (e.g., batch sizes, number of training epochs, learning rates) and randomness in the training procedure (e.g., random initializations) could all affect the final results. In our experiments, we fix batch size to 128 throughout and train all models with a learning rate of 0.01 until convergence for both MNIST and CIFAR-10 classification tasks. We train all models with a learning rate of 0.1 until convergence for the ising model regression task. Convergence is assumed when the training loss over two consecutive epochs is below a very small threshold of 10^{-8}. The only exception is for training the MLP10 model on the

Table 8. Computation time for the ground truth and DAVINZ, in GPU minutes.

Method	MNIST		CIFAR-10	
	VGG13	ResNet18	VGG13	ResNet18
Ground Truth	189.5	464.4	227.5	620.8
DAVINZ	2.5	3.3	2.0	3.2

ising physical model dataset, we use a smaller threshold of 10^{-10} to ensure convergence. The ground truth is then obtained using the average over 5 randomized trials. Overall, DAVINZ is able to achieve high correlations with the ground truth.

D.2. Validation Performance

Computing the ground truth for validation performances upon model convergence as described in Appendix D.1 is usually too computationally expensive to carry out in practice. We show Table 8 that ground truth baselines typically require 3 to 10 hours to evaluate while DAVINZ only takes 2 minutes. For practical considerations, we train all the models for 300 epochs and assume that convergence is achieved thereafter. We call this method validation performance (VP) in Sec. 6.2. The model details and hyperparameters are stated in Appendix D.1.

D.3. DAVINZ Setups

When the number of parameters in a network is small, such as the case in MLP10, we could afford exact computations of the NTK matrix $\mathbf{\Theta}_0$. For larger models such as VGG13 and ResNet18, the GPU VRAM could not hold the parameter gradients of the whole training dataset and thus the diagonal block NTK approximation is used, detailed in Appendix B. The number of diagonal blocks is set to 100. The balancing hyper-parameter κ is tuned following (4), as detailed in Sec.4.2.

D.4. Influence Functions

Originally designed for identifying influential training data, influence functions (IF) (Koh & Liang, 2017; Koh et al., 2019) can also be applied to data valuation problems by drawing this analogy: the training data that increase the validation loss the most has the least value. IF provides us with a way to quantify the change of validation loss when a data subset is removed without retraining the model. This fits the LOO setting and requires a fully-trained model on the complete dataset. The literature has also been discussed as a related work in Sec. 2.

Despite a training-free method, its evaluation comprises inverting a Hessian on the training loss of size $p \times p$, where p is the number of parameters of the model. This operations times time $O(np^2 + p^3)$ which is infeasible for complex neural networks with millions of parameters. Even after approximations such as Hessian vector products (HVP) and Conjugate Gradient (CG) techniques are adopted, the computation complexity is still $O(np)$.

For our baseline comparison purposes, we set batch size to 128, the recursion depth to 100 and the number of evaluations to average to 10 for the HVP calculation. This setup follows the original paper's suggestion that stochastic samples of a size similar to the training set size should be used for HVP calculation and we average over 10 evaluation for a stable HVP estimate.

A more important shortcoming is that IF requires convex and twice-differentiable models to produce accurate influence estimates. Although excellent results are shown in logistic regression, it fails on deep complex networks (Basu et al., 2021). Our results in Table 1 also show such evidence.

D.5. Robust Volume

Robust Volume (RV) is a diversity-based data valuation method proposed by Xu et al. (2021b). It has many appealing characteristics such as being training-free, replication robust, model- and task-agnostic. Its theoretical guarantee is, however, only proven for regression problems.

In this baseline comparison, we use an extension of the method mentioned in the original paper: We train a neural network as the feature extractor and use the learned features to calculate RV. This aligns with the approach of evaluating the value of

data (Jia et al., 2019a; Ghorbani et al., 2021) described in Sec. 2. In all experiments, we use a standardized 10-dimensional feature embedding and set the discretization coefficient $\omega = 0.1$, as recommended by the original paper.

E. More Results

E.1. Robustness to Different Initializations

Our data valuation is computed on a specific initialization $\boldsymbol{\theta}_0$ of the DNN. We show in this section that the valuation is robust under random initializations. As shown in Proposition 3, a smaller $\|\boldsymbol{\Theta}_{0,f} - \boldsymbol{\Theta}_{0,f'}\|_2$ (i.e., ϵ) provides a tighter bound on the variations of the valuations. We show in Proposition E.1 that we can theoretically guarantee a small $\|\boldsymbol{\Theta}_{0,f} - \boldsymbol{\Theta}_{0,f'}\|_2$ for a DNN model with two different random initializations and hence robustness under initializations.

Proposition E.1. *Choose $\epsilon > 0$ and $\delta \in (0,1)$. If the width of a L-layer DNN model satisfies $n = \Omega(L^6/\epsilon^4 \log\left(m_S^2 L/\delta\right))$, then for $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_0'$ using two different initialization $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0'$, the following holds with probability at least $1-\delta$ on dataset S of size m_S with $\|\boldsymbol{x}\|_2 \leq 1$ for any $\boldsymbol{x}$ in S,*

$$\left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_0'\right\|_2 \leq 2m_S^2 L\epsilon\,.$$

Proof. We firstly introduce the following lemmas, where $\boldsymbol{\theta}_0^{\leq l}$ and $\boldsymbol{\Theta}^{(l)}$ denote the model parameters from the first layer to l-th layer and the corresponding NTK matrix.

Lemma E.1 (An extension of Theorem 3.1 in (Arora et al., 2019a))**.** *Choose $\epsilon > 0$ and $\delta \in (0,1)$. If the width of every layer satisfied $n = \Omega\left(\frac{L^6}{\epsilon^4}\log\left(L/\delta\right)\right)$, then $\forall \boldsymbol{x}, \boldsymbol{x}'$ in dataset S with $\|\boldsymbol{x}\|_2, \|\boldsymbol{x}'\|_2 \leq 1$ and $\forall l \in [L]$, with probability at least $1-\delta$, we have that*

$$\left|\left\langle \nabla_{\boldsymbol{\theta}^{\leq l}} f(\boldsymbol{x}, \boldsymbol{\theta}_0), \nabla_{\boldsymbol{\theta}^{\leq l}} f(\boldsymbol{x}', \boldsymbol{\theta}_0)\right\rangle - \boldsymbol{\Theta}_\infty^{(l)}(\boldsymbol{x}, \boldsymbol{x}')\right| \leq l\epsilon\,.$$

Consequently, for any two different $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0'$ initialized using the standard normal distribution, the following inequality holds based on the result in Lemma E.1 and the definition of NTK matrix.

$$\begin{aligned}\left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_0'\right\|_2 &= \left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_\infty + \boldsymbol{\Theta}_\infty - \boldsymbol{\Theta}_0'\right\|_2 \\ &\leq \left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_\infty\right\|_2 + \left\|\boldsymbol{\Theta}_\infty - \boldsymbol{\Theta}_0'\right\|_2 \\ &\leq \left\|\boldsymbol{\Theta}_0 - \boldsymbol{\Theta}_\infty\right\|_{\mathrm{F}} + \left\|\boldsymbol{\Theta}_\infty - \boldsymbol{\Theta}_0'\right\|_{\mathrm{F}} \\ &\leq 2m_S^2 L\epsilon\,,\end{aligned} \tag{34}$$

which concludes our proof. □

Based on the proposition above, the conditional model-robust bound in Proposition 3 can be well-satisfied under different initializations. We next perform empirical investigations of such robustness. Similar to our observations in Section 6.6 and Table 3, ResNet with higher λ's typically demonstrate a higher consistency in data value over random initializations as compared to VGG. The relative changes in the valuations across initializations also agree with those obtained using the training-based VP method. Furthermore, we are able to compute the *coefficient of variance* (CV) under different initializations here and demonstrate that the CV of DAVINZ and VP are similar to the same order of magnitude. Therefore, our valuation method has similar robustness to retraining under different network initializations.

Table 9. The tables shows the change in $\nu(S)$ as model initialization changes from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}_0'$, averaged over 10 datasets S and all combinations of 5 initializations. $\lambda_{\min,\theta_0}$ is minimum eigenvalue of the NTK matrix when the model f is initialized with $\boldsymbol{\theta}_0$ and $\Delta_{\nu(S)} = |(\nu(S;\boldsymbol{\theta}_0) - \nu(S;\boldsymbol{\theta}_0'))/(\nu(S;\boldsymbol{\theta}_0)|$.

Model	Params (M)	$\lambda_{\min,\boldsymbol{\theta}_0}, \lambda_{\min,\boldsymbol{\theta}_0'}$ ($\times 10^{-5}$)	$\Delta^{\text{DAVINZ}}_{\nu(S)}$ (%)	$\Delta^{\text{VP}}_{\nu(S)}$ (%)	DAVINZ's CV (Coef. of Variance)	VP's CV (Coef. of Variance)
VGG11	9.2	56.4, 56.4	8.25±0.43	1.57±0.10	0.077±0.000	0.017±0.003
VGG13	9.4	1.58, 1.58	11.50±0.60	1.72±0.10	0.103±0.000	0.016±0.001
VGG16	14.7	0.104, 0.104	6.29±0.35	2.17±0.14	0.058±0.000	0.022±0.003
ResNet18	11.2	1310, 1310	5.11±0.26	2.72±0.18	0.046±0.000	0.024±0.003
ResNet21	17.4	1590, 1590	7.53±0.37	7.44±1.39	0.067±0.001	0.078±0.042
ResNet34	21.3	2140, 2140	3.44±0.23	5.23±0.51	0.036±0.000	0.049±0.014

E.2. $|f(\boldsymbol{x}, \boldsymbol{\theta}_0)|$ at Initialization

The proof of Proposition 2 is based on the assumption that $|f(\boldsymbol{x}, \boldsymbol{\theta}_0)| \leq \tau$ for any $\boldsymbol{x}$. We empirically verify that τ usually small, which approaches zero. Using the MLP10 network (in baseline comparisons for regression), we compute the absolute value of the network outputs for 10K data points on a randomly initialized network and find that the assumption is well satisfied. We obtain a mean output value of 0.000187 and a maximum output value of 0.000679. The details for all outputs are shown as a heat map in Fig. 7.

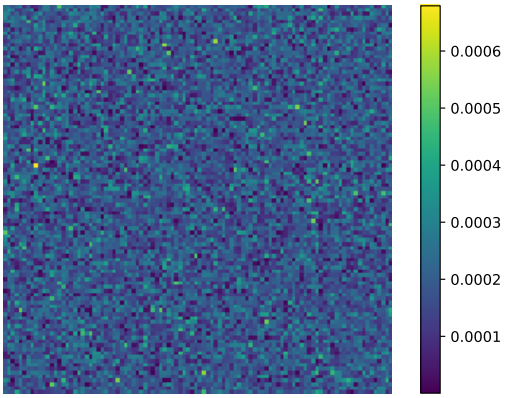

Figure 7. Function outputs $|f(\boldsymbol{x}, \boldsymbol{\theta}_0)|$ of 10K data points on an initialized MLP10 network. The length 10K output vector is reshaped into a 100×100 matrix for the convenience of viewing.

E.3. DAVINZ on Large-scale Datasets

With efficient approximations of $\nu(S)$ computation (detailed in Appendix B), our method does not suffer scalability problems with the size of dataset m_S. The inversion of the NTK matrix $\boldsymbol{\Theta}$ can be performed in a batch-wise manner. In this section, we demonstrate the effectiveness of DAVINZ on Tiny ImageNet (Deng et al., 2009), which includes 100K downsized 64×64 colored images for a 200-way classification task.

Fig. 1(b) shows $\nu(S)$ on differently sized datasets from Tiny Imagenet using ResNet-18. We follow the setting in Section 6.4 and the data split in Table 7. DAVINZ closely resembles the trend of VP when the quantity of data samples in the dataset increases. Note that we omit groud truth due to the model's long convergence time on Tiny ImageNet. We also follow the experimental setting of Table 2 and achieve a Pearson's correlation of 0.900 with VP.

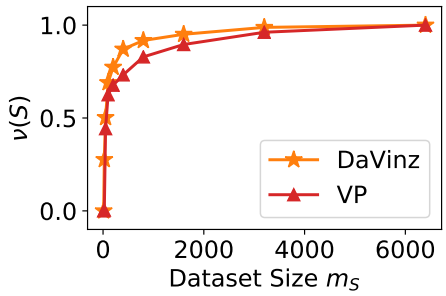

Figure 8. Scores achieved by differently sized datasets sampled from the Tiny Imagenet for a 200-way classification task using ResNet-18. We compare DAVINZ and VP, the results are re-scaled to $[0, 1]$ using min-max normalization.