A. Proof of Theorem 2

Proof. The proof relies on dissecting the term $\mathbb{E}[U(T\cup\{z_i\})-U(T\cup\{z_j\})]$ and $\nu(z_i)-\nu(z_j)$ ($\nu=\nu_{\text{shap-knn}},\nu_{\text{LOO-knn}}$) in the definition of order-preserving property.

Consider any two points $z_i, z_{i+l}\in D$. We start by analyzing $\mathbb{E}[U(T\cup\{z_i\})-U(T\cup\{z_{i+l}\})]$. Let the kth nearest neighbor of x_{val} in T be denoted by $T_{(k)}=(x_{(k)},y_{(k)})$. Moreover, we will use $T_{(k)}\leq_d z_i$ to indicate that $x_{(k)}$ is closer to the validation point than x_i, i.e., $d(x_{(k)},x_{\text{val}})\leq d(x_i,x_{\text{val}})$. We first analyze the expectation of the above utility difference by considering the following cases:

(1) $T_{(K)}\leq_d z_i$. In this case, adding z_i or z_{i+l} into T will not change the K-nearest neighbors to z_{val} and therefore $U(T\cup\{z_i\})=U(T\cup\{z_{i+l}\})=U(T)$. Hence, $U(T\cup\{z_i\})-U(T\cup\{z_{i+l}\})=0$.

(2) $z_i<_d T_{(K)}\leq_d z_{i+l}$. In this case, including the point i into T can expel the Kth nearest neighbor from the original set of K nearest neighbors while including the point $i+1$ will not change the K nearest neighbors. In other words, $U(T\cup\{z_i\})-U(T)=\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}$ and $U(T\cup\{z_{i+l}\})-U(T)=0$. Hence, $U(T\cup\{z_i\})-U(T\cup\{z_{i+l}\})=\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}$.

(3) $T_{(K)}>_d z_{i+l}$. In this case, including the point i or $i+1$ will both change the original K nearest neighbors in T by excluding the Kth nearest neighbor. Thus, $U(T\cup\{z_i\})-U(T)=\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}$ and $U(T\cup\{z_{i+l}\})-U(T)=\frac{\mathbb{1}[y_{i+l}=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}$. It follows that $U(T\cup\{z_i\})-U(T\cup\{z_{i+l}\})=\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]}{K}$.

Combining the three cases discussed above, we have

$$\begin{aligned}
&\mathbb{E}[U(T\cup\{z_i\})-U(T\cup\{z_{i+l}\})] && (6)\\
&=P(T_{(K)}\leq_d z_i)\times 0+P(z_i<_d T_{(K)}\leq_d z_{i+l})\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}\\
&\quad+P(T_{(K)}>_d z_{i+l})\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]}{K} && (7)\\
&=P(z_i<_d T_{(K)}\leq_d z_{i+l})\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{(K)}=y_{\text{val}}]}{K}\\
&\quad+P(T_{(K)}>_d z_{i+l})\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]}{K} && (8)
\end{aligned}$$

Note that removing the first term in (8) cannot change the sign of the sum in (8). Hence, when analyzing the sign of (8), we only need to focus on the second term:

$$P(T_{(K)}>_d z_{i+1})\frac{\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]}{K} \tag{9}$$

Since $P(T_{(K)}>_d z_{i+1})=\sum_{k=N-K+1}^{N}P(Z>_d z_{i+1})^k$, the sign of (9) will be determined by the sign of $\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]$. Hence, we get

$$\big(\mathbb{E}[U(T\cup\{z_i\})-U(T\cup\{z_{i+1}\})]\big)\times\big(\mathbb{1}[y_i=y_{\text{val}}]-\mathbb{1}[y_{i+l}=y_{\text{val}}]\big)>0 \tag{10}$$

Now, we switch to the analysis of the value difference. By Theorem 1, it holds for *the KNN-Shapley value* that

$$\begin{aligned}
&\nu_{\text{shap-knn}}(z_i)-\nu_{\text{shap-knn}}(z_{i+l}) && (11)\\
&=\sum_{j=i}^{i+l-1}\frac{\min\{K,j\}}{jK}\big(\mathbb{1}[y_j=y_{\text{val}}]-\mathbb{1}[y_{j+1}=y_{\text{val}}]\big) && (12)\\
&=\frac{\min\{K,i\}}{iK}\mathbb{1}[y_i=y_{\text{val}}]+\sum_{j=i}^{i+l-2}\left(\frac{\min\{K,j+1\}}{(j+1)K}-\frac{\min\{K,j\}}{jK}\right)\mathbb{1}[y_{j+1}=y_{\text{val}}]\\
&\quad-\frac{\min\{K,i+l-1\}}{(i+l-1)K}\mathbb{1}[y_{i+l}=y_{\text{val}}] && (13)
\end{aligned}$$

Note that $\frac{\min\{K,j+1\}}{(j+1)K}-\frac{\min\{K,j\}}{jK}<0$ for all $j=i,\ldots,i+l-2$. Thus, if $\mathbb{1}[y_i=y_{\text{val}}]=1$ and $\mathbb{1}[y_{i+l}=y_{\text{val}}]=0$, the minimum of (13) is achieved when $\mathbb{1}[y_{j+1}=y_{\text{val}}]=1$ for all $j=i,\ldots,i+l-2$ and the minimum value is $\frac{\min\{K,i+l-1\}}{(i+l-1)K}$,

which is greater than zero. On the other hand, if $\mathbb{1}[y_i = y_{\text{val}}] = 0$ and $\mathbb{1}[y_{i+l} = y_{\text{val}}] = 1$, then the maximum of (13) is achieved when $\mathbb{1}[y_{j+1} = y_{\text{val}}] = 0$ for all $j = i, \ldots, i+l-2$ and the maximum value is $-\frac{\min\{K, i+l-1\}}{(i+l-1)K}$, which is less than zero.

Summarizing the above analysis, we get that $\nu_{\text{shap-knn}}(z_i) - \nu_{\text{shap-knn}}(z_{i+l})$ has the same sign as $\mathbb{1}[y_i = y_{\text{val}}] - \mathbb{1}[y_{i+l} = y_{\text{val}}]$. By (10), it follows that $\nu_{\text{shap-knn}}(z_i) - \nu_{\text{shap-knn}}(z_{i+l})$ also shares the same sign as $\mathbb{E}[U(T \cup \{z_i\}) - U(T \cup \{z_{i+1}\})]$.

To analyze the sign of *the KNN-LOO value* difference, we first write out the expression for *the KNN-LOO value* difference:

$$\nu_{\text{loo-knn}}(z_i) - \nu_{\text{loo-knn}}(z_{i+l}) = \begin{cases} \frac{1}{K}(\mathbb{1}[y_i = y_{\text{val}}] - \mathbb{1}[y_{i+l} = y_{\text{val}}]) & \text{if } i + l \leq K \\ \frac{1}{K}(\mathbb{1}[y_i = y_{\text{val}}] - \mathbb{1}[y_{K+1} = y_{\text{val}}]) & \text{if } i \leq K < i + l \\ 0 & \text{if } i > K \end{cases} \tag{14}$$

Therefore, $\nu_{\text{loo-knn}}(z_i) - \nu_{\text{loo-knn}}(z_{i+l})$ has the same sign as $\mathbb{1}[y_i = y_{\text{val}}] - \mathbb{1}[y_{i+l} = y_{\text{val}}]$ and $\mathbb{E}[U(T \cup \{z_i\}) - U(T \cup \{z_{i+1}\})]$ only when $i + l \leq K$.

□

B. Proof of Theorem 3

We will need the following lemmas on group differential privacy for the proof of Theorem 3.

Lemma 2. *If $\mathcal{A}$ is (ϵ, δ)-differentially private with respect to one change in the database, then $\mathcal{A}$ is $(c\epsilon, ce^{c\epsilon}\delta)$-differentially private with respect to c changes in the database.*

Lemma 3 ([12])**.** *For any $z_i, z_j \in D$, the difference in Shapley values between z_i and z_j is*

$$\nu_{shap}(z_i) - \nu_{shap}(z_j) = \frac{1}{N-1} \sum_{T \subseteq D \setminus \{z_i, z_j\}} \frac{U(T \cup \{z_i\}) - U(T \cup \{z_j\})}{\binom{N-2}{|T|}} \tag{15}$$

Proof. Let S' be the set with one element in S replaced by a different value. Let the probability density/mass defined by $\mathcal{A}(S')$ and $\mathcal{A}(S)$ be $p(h)$ and $p'(h)$, respectively. Using Lemma 2, for any z_{val} we have

$$\begin{aligned} \mathbb{E}_{h \sim \mathcal{A}(S)} l(h, z_{\text{val}}) &= \int_0^1 P_{h \sim \mathcal{A}(S)}[l(h, z_{\text{val}}) > t] dt & (16) \\ &\leq \int_0^1 (e^{c\epsilon} P_{h \sim \mathcal{A}(S')}[l(h, z_{\text{val}}) > t] + ce^{c\epsilon}\delta) dt & (17) \\ &= e^{c\epsilon} \mathbb{E}_{h \sim \mathcal{A}(S')}[l(h, z_{\text{val}})] + ce^{c\epsilon}\delta & (18) \end{aligned}$$

It follows that

$$\begin{aligned} \mathbb{E}_{h \sim \mathcal{A}(S)} l(h, z_{\text{val}}) - \mathbb{E}_{h \sim \mathcal{A}(S')}[l(h, z_{\text{val}})] &\leq (e^{c\epsilon} - 1)\mathbb{E}_{h \sim \mathcal{A}(S')}[l(h, z_{\text{val}})] + ce^{c\epsilon}\delta & (19) \\ &\leq e^{c\epsilon} - 1 + ce^{c\epsilon}\delta & (20) \end{aligned}$$

By symmetry, it also holds that

$$\begin{aligned} \mathbb{E}_{h \sim \mathcal{A}(S')} l(h, z_{\text{val}}) - \mathbb{E}_{h \sim \mathcal{A}(S)}[l(h, z_{\text{val}})] &\leq (e^{c\epsilon} - 1)\mathbb{E}_{h \sim \mathcal{A}(S)}[l(h, z_{\text{val}})] + ce^{c\epsilon}\delta & (21) \\ &\leq e^{c\epsilon} - 1 + ce^{c\epsilon}\delta & (22) \end{aligned}$$

Thus, we have the following bound:

$$|\mathbb{E}_{h \sim \mathcal{A}(S)} l(h, z_{\text{val}}) - \mathbb{E}_{h \sim \mathcal{A}(S')}[l(h, z_{\text{val}})]| \leq e^{c\epsilon} - 1 + ce^{c\epsilon}\delta \tag{23}$$

Denoting $\epsilon' = e^{c\epsilon} - 1 + ce^{c\epsilon}\delta$. For the performance measure that evaluate the loss averaged across multiple validation points $U(S) = -\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{h \sim \mathcal{A}(S)} l(h, z_{\text{val},i})$, we have

$$|U(S) - U(S')| \leq \epsilon' \tag{24}$$

Making the dependence on the training set size explicit, we can re-write the above equation as

$$\max_{z_i,z_j\in D,T\subseteq D\setminus\{z_i,z_j\}} |U(T\cup z_i)-U(T\cup z_j)|\leq \epsilon'(|T|+1) \tag{25}$$

By Lemma 3, we have for all $z_i, z_j \in D$,

$$\nu_{\text{shap}}(z_i)-\nu_{\text{shap}}(z_j)\leq \frac{1}{N-1}\sum_{k=0}^{N-2}\sum_{T\subseteq D\setminus\{z_i,z_j\},|T|=k}\frac{\epsilon'(k+1)}{\binom{N-2}{k}} \tag{26}$$

$$=\frac{1}{N-1}\sum_{k=0}^{N-2}\epsilon'(k+1) \tag{27}$$

$$=\frac{1}{N-1}\sum_{k=1}^{N-1}\epsilon'(k) \tag{28}$$

As for the LOO value, we have

$$\nu_{loo}(z_i)-\nu_{loo}(z_j)=U(D\setminus\{z_j\})-U(D\setminus\{z_i\}) \tag{29}$$

$$\leq \epsilon'(N-1) \tag{30}$$

□

C. Comparing the LOO and the Shapley Value for Stable Learning algorithms

An algorithm G has uniform stability γ with respect to the loss function l if $\|l(G(S),\cdot)-l(G(S^{\setminus i}),\cdot)\|_\infty\leq\gamma$ for all $i\in\{1,\cdots,|S|\}$, where S denotes the training set and $S^{\setminus i}$ denotes the one by removing ith element of S.

Theorem 4. *For a learning algorithm $\mathcal{A}(\cdot)$ with uniform stability $\beta=\frac{C_{stab}}{|S|}$, where $|S|$ is the size of the training set and C_{stab} is some constant. Let the performance measure be $U(S)=-\frac{1}{M}\sum_{i=1}^{M}l(A(S),z_{val,i})$. Then,*

$$\max_{z_i\in D}\nu_{loo}(z_i)-\nu_{loo}(z^*)\leq\frac{C_{stab}}{N-1} \tag{31}$$

and

$$\max_{z_i\in D}\nu_{shap}(z_i)-\nu_{shap}(z^*)\leq\frac{C_{stab}(1+\log(N-1))}{N-1} \tag{32}$$

Proof. By the definition of uniform stability, it holds that

$$\max_{z,z_j\in D,T\subseteq D\setminus\{z_i,z_j\}}|U(T\cup\{z_i\})-U(T\cup\{z_j\})|\leq\frac{C_{\text{stab}}}{|T|+1} \tag{33}$$

Using Lemma 3, we have we have for all $z_i, z_j \in D$,

$$\nu_{\text{shap}}(z_i)-\nu_{\text{shap}}(z_j) \tag{34}$$

$$\leq\frac{1}{N-1}\sum_{k=0}^{N-2}\sum_{T\subseteq D\setminus\{z_i,z_j\},|T|=k}\frac{C_{\text{stab}}}{\binom{N-2}{k}(k+1)} \tag{35}$$

$$=\frac{1}{N-1}\sum_{k=0}^{N-2}\frac{C_{\text{stab}}}{k+1} \tag{36}$$

Recall the bound on the harmonic sequences

$$\sum_{k=1}^{N}\frac{1}{k}\leq 1+\log(N)$$

which gives us

$$\nu_{\text{shap}}(z_i) - \nu_{\text{shap}}(z_j) \leq \frac{C_{\text{stab}}(1 + \log(N-1))}{N-1}$$

As for the LOO value, we have

$$\nu_{loo}(z_i) - \nu_{loo}(z_j) = U(D \setminus \{z_j\}) - U(D \setminus \{z_i\}) \leq \frac{C_{\text{stab}}}{N-1} \tag{37}$$

□

D. Additional Experiments

D.1. Rank Correlation with Ground Truth Shapley Value

We perform experiments to compare the ground truth Shapley value of raw data and the value estimates produced by different heuristics. The ground truth Shapley value is computed using the group testing algorithm in [12], which can approximate the Shapley value with provable error bounds. We use a fully-connected neural network with three hidden layers as the target model. Following the setting in [12], we construct a size-1000 training set using MNIST, which contains both benign and adversarial examples, as well as a size-100 validation set with pure adversarial examples. The adversarial examples are generated by the Fast Gradient Sign Method [8]. This construction is meant to simulate data with different levels of usefulness. In the above setting, the adversarial examples in the training set should be more valuable than the benign data because they can improve the prediction on adversarial examples. Note that the KNN-Shapley computes the Shapley value of deep features extracted from the penultimate layer.

The rank correlation of KNN-Shapley and G-Shapley with the ground truth Shapley value is 0.08 and 0.024 with p-value 0.0046 and 0.4466, respectively. It shows that both heuristics may not be able to preserve the exact rank of the ground truth Shapley value. Since TMC-Shapley cannot finish in a week for this model and data size, we omit it from comparison. We further apply some local smoothing to the scores and check whether these heuristics can produce large scores for data groups with large Shapley values. Specifically, we compute 1 to 100 percentiles of the Shapley values, find the group of data points within each percentile interval, and compute the average Shapley value as well as the average heuristic scores for each group. The rank correlation of the average KNN-Shapley and the average G-Shapley with the average ground truth Shapley value for these data groups are 0.22 and -0.002 with p-value 0.0293, 0.9843, respectively. We can see that although ignoring the data contribution for feature learning, KNN-Shapley can better preserve the rank of the Shapley value in a macroscopic level than G-Shapley.

E. Experiment Details and Results on More Datasets

In this section, we present experiment details and results on more datasets corresponding to the applications introduced in the main body (see Section 4).

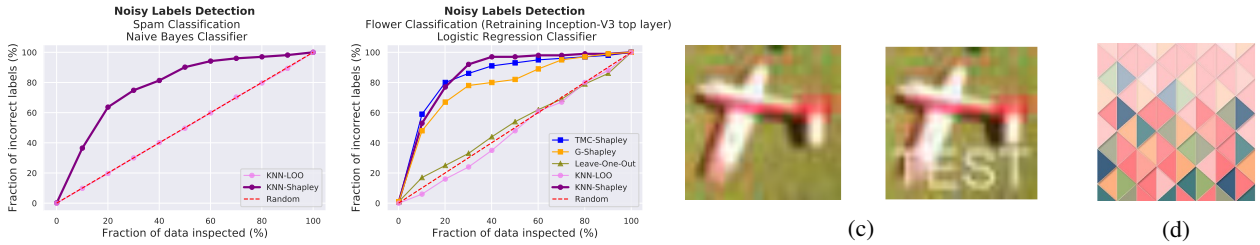
E.1. Detecting Noisy Labels

Following Ghorbani *et al.* [7], we conducted another two experiments: a Naive Bayes model trained on a spam classification dataset and a logistic regression model trained on Inception-V3 features of a flower classification dataset. The noise flipping ratio is 20% and 10% respectively for these two datasets. The performance

The performance of different data importance measures is illustrated in Fig. 6a and Fig. 6b. We examine the label of the training instances that have the lowest scores, and plot the change of the fraction of detected mislabeled data with the fraction of the checked training data. We can see that *the KNN-Shapley value* outperforms all other methods. Also, the Shapley value–based measures, including TMC-Shapley, G-Shapley, and our KNN-Shapley, are more effective than the LOO-based measures.

E.2. Watermark Removal

We discuss two main types of techniques for injecting watermarks. The pattern-based techniques inject a set of samples that are blended with the same pattern and labeled with one certain class into the training set; the data contributor can later verify the data source of the trained model by checking the output of the model for an input with the pattern. The instance-based



(a) (b) (c) (d)

Figure 6: (a-b) Results of noisy label detection on Spam Dataset and Flower Dataset; (c-d) Examples of watermarks generated by pattern-based techniques and instance-based techniques.

techniques, by contrast, inject individual training samples labeled with a specific class as watermarks and the verification can be done by inputting the same samples into the trained model.

In pattern-based watermark removal, we adopted two types of patterns: one is to change the pixel values at the corner of an image [5], another is to blend a specific word (like "TEST") into an image, as shown in Figure 6c. Specifically, after an image is blended with the "TEST" pattern, there is high chance that it is classified as the target label, e.g, an "automobile" on CIFAR-10. The first pattern is used in the experiments on fashion MNIST and MNIST, which is composed of single channel images. The second pattern is applied to Pubfig-83 which contains multi-channel images.

In instance-based watermark removal, we used the same watermarks as [2], which contains a set of abstract images with specific assigned labels. The example of a trigger image is shown in Figure 6d. This type of watermarks are typically chosen from out-of-distribution data.

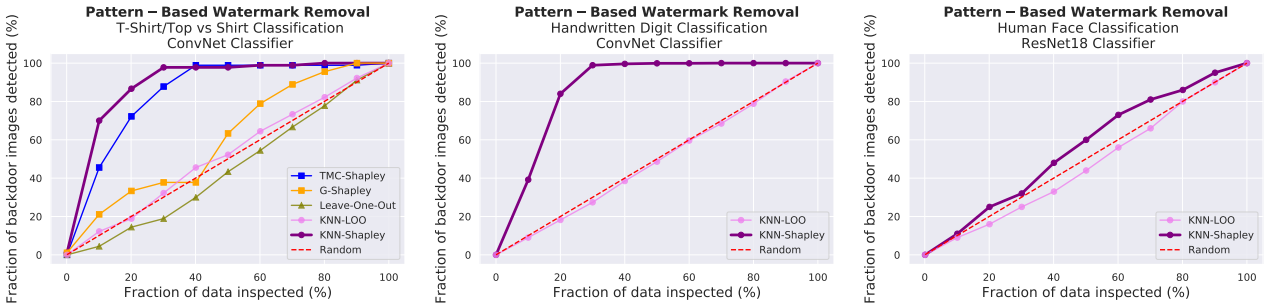
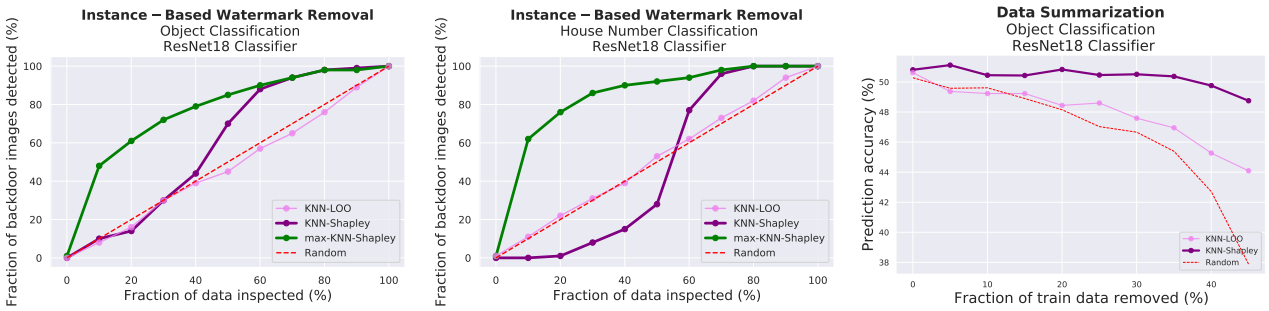


Figure 7: Results of pattern-based watermark removal tasks on (a) Fashion-MNIST Dataset [29], (b) MNIST Dataset [21], and (c) PubFig-83 Dataset [19].

For the pattern-based watermark removal experiment, we consider three settings: two convolutional networks trained on 1000 images from fashion MNIST and 10000 images from MNIST, respectively, and a ResNet18 [9] model trained on 1000 images from the face recognition dataset Pubfig-83. The watermark ratio is 10% for all three settings. Since for the last two settings, TMC-Shapley, G-Shapley, and Leave-one-out all fail to produce importance estimates in 3 hours either due to large data size or model size, we compare our algorithm only with the rest of the baselines. In plotting Fig. 7, we examine the label of the training instances that have the lowest scores and plot the change of the faction of the detected watermarks (in percentage) with the fraction of the checked training data (in percentage). Although TMC-Shapley can achieve similar performance to KNN-Shapley, its time complexity is actually much higher than KNN-Shapley. Compared with all other baselines, our KNN-Shapley outperforms achieves the best performance.

For the instance-based watermark removal experiment, we consider the following two settings: a convolution network trained on 3000 images from CIFAR-10 [18], and ResNet18 trained on 3000 images from SVHN [22]. The watermark ratio is 3% in both settings. The results of our experiment are displayed in Fig. 8a and Fig. 8b. We plot the change of the fraction of the detected watermarks (in percentage) with the fraction of the checked training data (in percentage).

As discussed in Section 4.3, we propose a novel measure max-KNN-Shapley to tackle the instance-based watermark removal task specifically. As shown in Fig. 8a and Fig. 8b, the max-KNN-Shapley is a more effective measure to detect



(a) (b) (c)

Figure 8: Results of (a-b) instance-based watermark removal tasks on CIFAR-10 [18] and SVHN; (c) Data summarization on Tiny ImageNet [20].

instance-based watermarks than all other baselines.

Table 3: **Instance-based watermark removal.** Prediction accuracy on different types of data.

Data Type	Handwritten Digit (Logistic Regression)	Object (ResNet18)	House Number (ResNet18)
Benign Data	0.998	0.981	1.000
Watermark Data	0.980	1.000	1.000

We additionally measure the prediction accuracy of the watermarked model on both benign and watermark instances and provide the results in Table 3. The results indicate that the amount of watermarks we added satisfies our purpose of claiming the ownership of the data source.

E.3. Data Summarization

For the experiment on UCI Adult Census dataset [15] introduced in Section 4.3, we train the same multilayer perceptron model as [6].

We consider another setting for this application: a ResNet-18 trained on Tiny ImageNet. In this setting, we use 95000 points as the training set, 5000 points to calculate the scores, and another 10000 points as the held-out validation set. In Fig. 8c, we plot the change of prediction accuracy (in percentage) with the change of the fraction of data removed (in percentage). As it reveals, KNN-Shapley is able to maintain model performance even after removing 40% of the whole training set. However, TMC-Shapley, G-Shapley, and LOO cannot finish in 24 hours and hence are omitted from the figure.

In this experiment, we fine-tune the pretrained ResNet18 from He *et al.*. We train the ResNet18 with 15 epochs and learning rate 0.001 with SGD optimizer [13] and the model achieves an accuracy of 77.95% on the training set. Then, we extract the deep features of the training set and calculate their Shapley values. When evaluating the model performance on the summarized dataset, we re-train the ResNet18 with 30 epochs and learning rate 0.01.

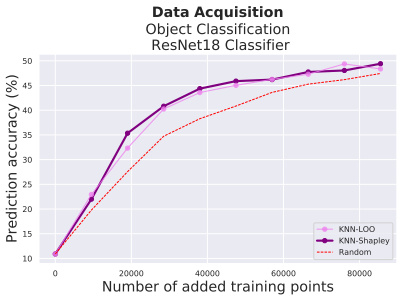
E.4. Data Acquisition

We follow the same protocols as in Section 4.3 to conduct the experiments on Tiny ImageNet, which in nature has realistic variation of data quality. We separate the training set into two parts with 5000 training points and 95000 new points. We calculate importannce of 2500 data points in the training set based on the other 2500 points. In Fig. 9a we plot the change of prediction accuracy with the number of added training points. Evidently, new data selected based on KNN-Shapley value improves model accuracy faster than all other methods.

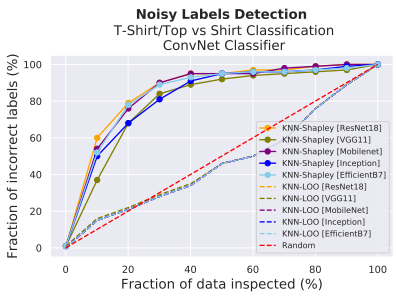
E.5. Domain Adaptation

In section 4.3 we elaborated on the transfer between MNIST and USPS[2], where we trained a multinomial logistic regression classifier. Here, we introduce another experiment on transferring from SVHN to MNIST. In this experiment, we train a

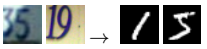
[2]from https://www.kaggle.com/bistaumanga/usps-dataset



(a)



(b)

Method	SVHN → MNIST
	
KNN-Shapley	**9.65% → 14.1%**
KNN-LOO	9.65% → 9.3%
TMC-Shapley	—
LOO	—

(c)

Figure 9: (a) Data acquisition on Tiny ImageNet; (b) results of different embeddings of noisy label detection on Fashion-MNIST; (c) Domain adaptation on SVHN→MNIST

ResNet18 model using 15 epochs and learning rate 0.001 with SGD optimizer on SVHN, since multinomial logistic regression is too simple to perform well in this setting. We pick 2000 training data from SVHN, train a ResNet-18 model, and evaluate the performance on the whole test set of MNIST. KNN-Shapley is able to work on data of this scale efficiently while TMC-Shapley algorithm simply cannot finish in 48 hours. As shown in Table 9c, our KNN-Shapley achieves better performance than KNN-LOO.

F. Impact of Different Embeddings

In Section 4.3 we provide the result corresponding to the embedding extracted by one single feature extractor for each dataset. In this section, for all the aforementioned experiments, we tried different embeddings extracted using five pre-trained classifiers including ResNet18, VGG11 [25], MobileNet [10], Inception-V3 [26], and EfficientNet B7 [27].

Illustrated in Fig. 9b is the comparison of two data importance measures: KNN-Shapley and KNN-LOO, each applied to five different embeddings. This experiment is carried out on the Fashion-MNIST dataset for the task of detecting noisy labels. Notably, the five curves of KNN-Shapley are close to each other, and the same trend can also be observed for the five curves of KNN-LOO. Apart from this observation, the scores given by KNN-LOO are roughly the same as random, while our KNN-Shapley are all much higher. As a conclusion, the influence induced by using different embeddings is marginal compared to using different measures. Furthermore, our KNN-Shapley data importance measure can achieve terrific performance without the need of carefully selecting embeddings. We provide a comprehensive set of results in Fig. 10, where similar conclusions can be drawn.

As a supplement to Fig. 3 in the main body, similarly, in Fig. 11, we provide the top 20 images with highest Shapley value, as well as the top 50 classes after the summarization step for each of the following embeddings: Resnet18, Inception-V3, and EfficientNet B7. As can be observed, there is a large range of overlap among the top classes for all these embeddings, which we believe is an intriguing phenomenon to study and will inspire future research.

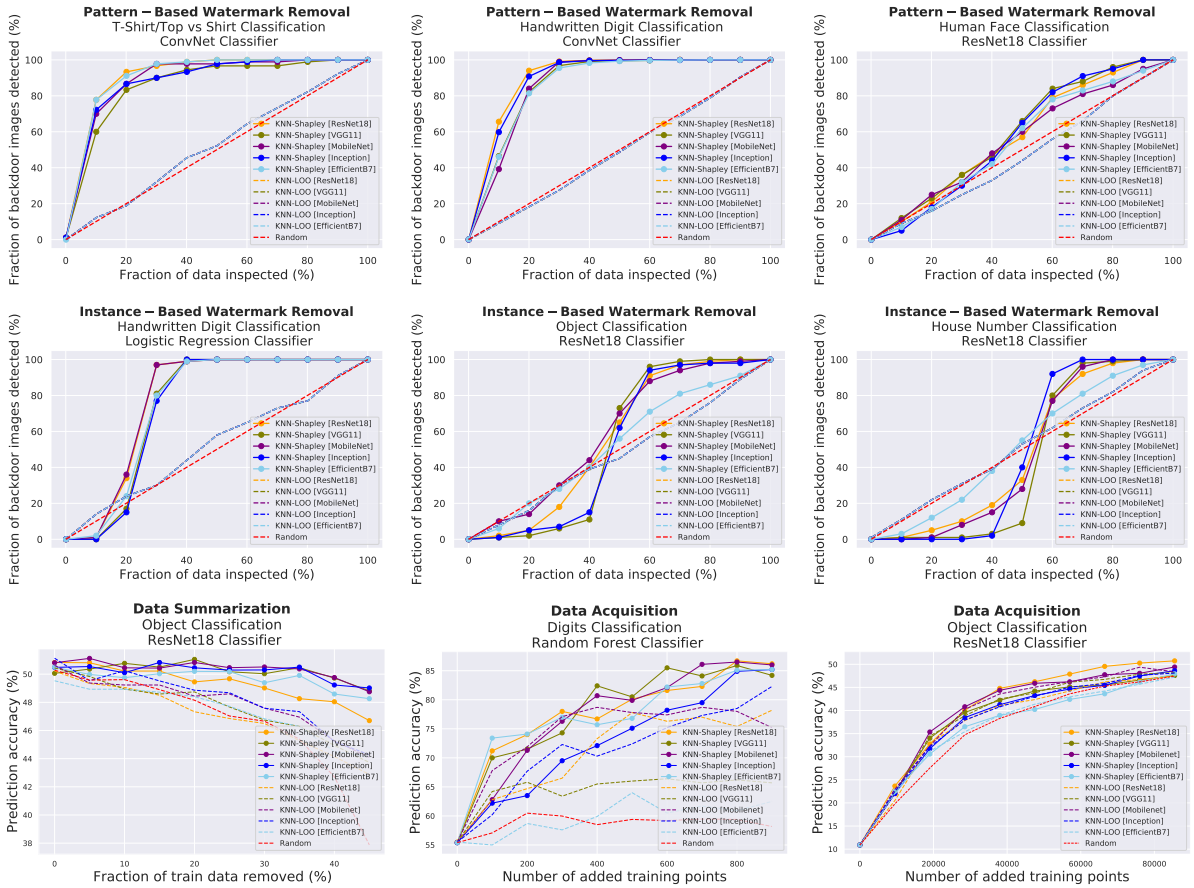


Figure 10: Comparisons of different embeddings on different datasets and different applications

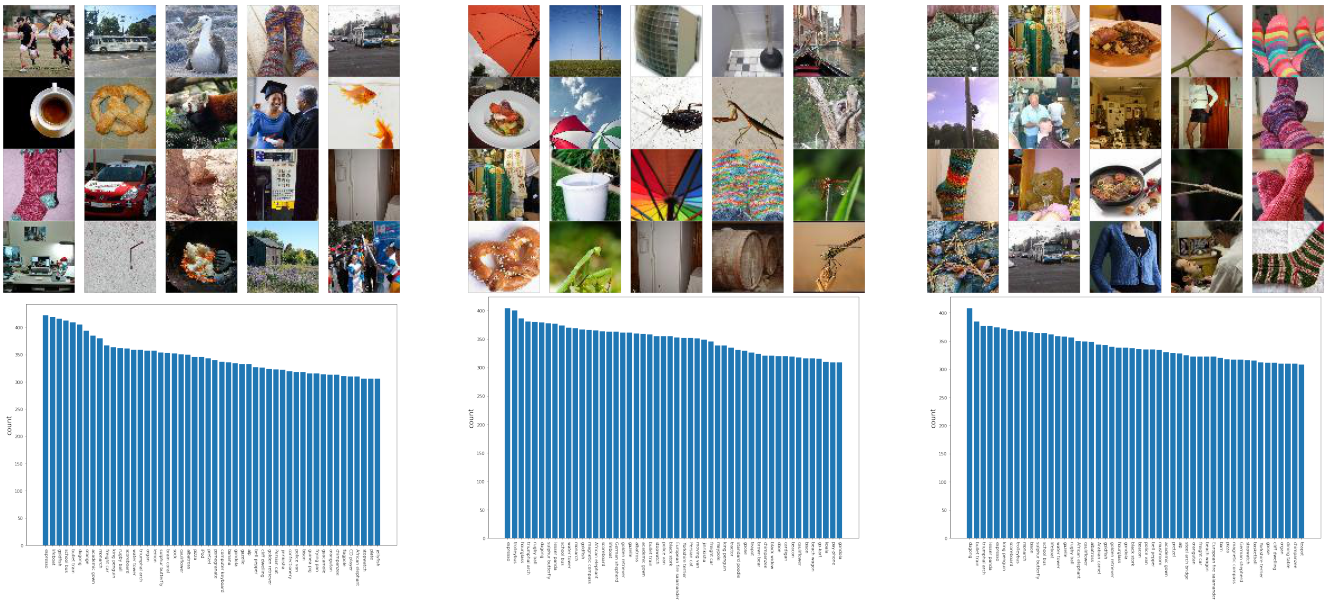


Figure 11: First row: top 20 selected images with the highest Shapley values in Tiny ImageNet for Resnet18, Inception-V3, and EfficientNet B7 embeddings, respectively; Second row: counts of images in top 50 classes after the summarization step (sorted by the count in a decreasing manner). There are many overlapped classes among different embeddings.