DEEP K-NN FOR NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern machine learning models are often trained on examples with noisy labels that hurt performance and are hard to identify. In this paper, we provide an empirical study showing that a simple k-nearest neighbor-based filtering approach on the logit layer of a preliminary model can remove mislabeled training data and produce more accurate models than some recently proposed methods. We also provide new statistical guarantees into its efficacy.

1 INTRODUCTION

Machine learned models can only be as good as the data they were used to train on. With increasingly large modern datasets and automated and indirect labels like clicks, it is becoming ever more important to investigate and provide effective techniques to handle noisy labels.

We revisit the classical method of filtering out suspicious training examples using k-nearest neighbors (k-NN) (Wilson, 1972). Like Papernot & McDaniel (2018), we apply k-NN on the learned intermediate representation of a preliminary model, which adds robustness. In fact, the k-nearest neighbor approach has recently been receiving attention for its robustness properties (Wang et al., 2018; Reeve & Kaban, 2019) and as an auxiliary strategy for modern machine learning (Jiang et al., 2018).

The main contributions of this paper are:

- Experimentally showing that identifying mislabeled examples by k-NN executed on an intermediate layer of a preliminary deep model works well compared to state-of-art methods for handling noisy labels across noise levels, and is robust to the choice of k.
- Theoretically showing that k-NN's predictions will only identify a training example as clean if its label is the Bayes-optimal label. We also provide finite-sample analysis in terms of the margin and how spread out the corrupted labels are (Theorem 1), rates of convergence for the margin (Theorem 2) and rates under Tsybakov's noise condition (Theorem 3) with all rates matching minimax-optimal rates in the noiseless setting.

Our work shows that even though the preliminary neural network is trained with corrupted labels, it still yields intermediate representations that are useful for k-nearest neighbor filtering. Given labels which are in high disagreement, one can either automatically remove them and retrain on the remaining, or send to a human operator for further review. This strategy is also be useful in human-in-the-loop systems where one can warn the human annotator that a label is suspicious, and automatically propose new labels based on its nearest neighbors' labels.

In addition to strong empirical performance, deep k-NN filtering has a couple of advantages. Firstly, many methods require a clean set of samples whose labels can be trusted. Here we show that the k-NN based method is robust and does not require such a clean set of samples. Second, while k-NN does introduce the hyperparameter k, we will show that deep k-NN filtering is stable to the choice of k: such robustness to hyperparameters is highly desirable as optimal tuning for this problem is often not available in practice (i.e. when no clean validation set is available).

2 RELATED WORK

We review relevant prior work in training on noisy labels and related k-NN theory.

2.1 TRAINING WITH NOISY LABELS

Methods to handle label noise can be classified into two main strategies: (i) explicitly identify and remove the noisy examples, and (ii) indirectly handle the noise with robust training methods.

Data Cleaning This proposal fits into the broad family of *data cleaning* methods, in that our proposal detects and filters *dirty data* (see Chu et al. (2016) for a recent survey). The use of k-NN to "edit" training data has been popular since Wilson (1972) used it to throw away training examples that were not consistent with their $k = 3$ nearest neighbors. The idea of using a preliminary model to help identify mislabeled examples dates to at least Guyon et al. (1994), who proposed using the model to compute an information gain for each example and then suspecting ones with high gain. Other early work used a cross-validation set-up to train a classifier on part of the data, then use it to make a prediction on held-out training examples, and remove any examples if the prediction disagrees with the label; ensembles of models can also be used for the predictions. (Brodley & Freidl, 1999).

Noise Corruption Estimation For multi-class problems, a popular approach is to account for noisy labels by applying a confusion matrix after the model's softmax layer (Sukhbaatar et al., 2014). Such methods rely on a confusion matrix which is often unknown and must be estimated. Patrini et al. (2017) suggest deriving it from the softmax distribution of the model trained on noisy data while Goldberger & Ben-Reuven (2016); Jindal et al. (2016); Han et al. (2018) give alternatives. Accurate estimates are generally hard to attain when only untrusted data is available. Hendrycks et al. (2018) achieves more accurate estimates in the setting where some amount of known clean, trusted data is available. Xiao et al. (2015); Khetan et al. (2017); Vahdat (2017) use EM-type algorithms to estimate the clean label distribution.

Noise-Robust Training Loss Natarajan et al. (2013) propose a method to make any surrogate loss function noise-robust given knowledge of the corruption rates. Ghosh et al. (2017) proves that losses like mean absolute error (MAE) are inherently robust under symmetric or uniform label noise while Zhang & Sabuncu (2018) shows that training with MAE results in poor convergence and accuracy. They propose a new loss function based on the negative Box-Cox transformation that trades off the noise-robustness of MAE with the training efficiency of cross-entropy. Lastly, the ramp, unhinged, and savage losses have been proposed and theoretically justified to be noise-robust for support vector machines (Brooks, 2011; Van Rooyen et al., 2015; Masnadi-Shirazi & Vasconcelos, 2009).

Auxiliary Models Veit et al. (2017) propose learning a label cleaning network on trusted data by predicting the differences between clean and noisy labels. Li et al. (2017) suggests training on a weighted average between noisy labels and distilled predictions of an auxiliary model trained on trusted data.

Example Weighting Here we make a hard decision about whether to keep a training example, but one can also adapt the weights on training examples based on the confidence in their labels. Liu & Tao (2015) provides an importance-weighting scheme for binary classification. Ren et al. (2018) suggests upweighting examples whose loss gradient is aligned with those of trusted examples at every step in training. Jiang et al. (2017) investigates a recurrent network that learns a sample weighting scheme to give to the base model.

2.2 k-NEAREST NEIGHBOR THEORY

The theory of k-nearest neighbor classification has a long history, for example: Fix & Hodges Jr (1951); Cover (1968); Stone (1977); Devroye et al. (1994); Chaudhuri & Dasgupta (2014). Much of the prior work focuses on k-NN's statistical consistency properties. However, with the growing interest in adversarial examples and learning with noisy labels, there have recently been analyses of k-nearest neighbor methods in these settings. Wang et al. (2018) analyze the robustness of k-NN classification and provide a robust variant of 1-NN classification where their notion of robustness is that predictions of nearby points should be similar. Gao et al. (2016) provides an analysis of the k-NN classifier under noisy labels and like us, show that k-NN can attain similar rates in the noisy setting as in the noiseless setting. Gao et al. (2016) assumes a noise model where labels are corrupted uniformly at random, while we assume an arbitrary corruption pattern and provide results based on a notion of how spread out the corrupted points are. Moreover, we provide finite-sample bounds borrowing recent advances in k-NN convergence theory in the noiseless setting (Jiang, 2019)

while the guarantees of Gao et al. (2016) are asymptotic. Reeve & Kaban (2019) provide stronger guarantees on a robust modification of k-NN proposed by Gao et al. (2016). To the best of our knowledge, we provide the first finite-sample rates of consistency for the classical k-NN method in the noisy setting with very little assumptions on the label noise.

3 ALGORITHM

We first define the k-nearest neighbor classifier:

Definition 1 (k-NN)**.** *Let the k-NN radius of $x \in \mathcal{X}$ be $r_k(x) := \inf\{r : |B(x,r) \cap X| \geq k\}$ where $B(x,r) := \{x' \in \mathcal{X} : |x - x'| \leq r\}$ and the k-NN set of $x \in \mathcal{X}$ be $N_k(x) := B(x, r_k(x)) \cap X$. Then for all $x \in \mathcal{X}$, the k-NN classifier function w.r.t. X has discriminant function*

$$\eta_k(y; x) := \frac{1}{|N_k(x)|} \sum_{i=1}^{n} 1\left[y_i = y,\ x_i \in N_k(x)\right],$$

with prediction $\eta_k(x) := \arg\max_y \eta_k(y; x)$.

Our method Algorithm 1 assumes a dataset $\mathcal{D}_{\text{noisy}}$ with potentially noisy labels, along with a dataset $\mathcal{D}_{\text{clean}}$ consisting of clean or trusted labels. Note that we allow $\mathcal{D}_{\text{clean}}$ to be empty (i.e. in instances where no such trusted data is available). We have found that having $\mathcal{D}_{\text{clean}}$ becomes important when $\mathcal{D}_{\text{noisy}}$ has a high corruption rate; otherwise the representations learned by training on $\mathcal{D}_{\text{noisy}}$ alone are often reasonable enough. The procedure begins by training on either $\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{\text{clean}}$ or $\mathcal{D}_{\text{clean}}$. For our experiments, we partition $\mathcal{D}_{\text{clean}}$ into a training set $\mathcal{D}_{ct}$ and validation set $\mathcal{D}_{cv}$ and train models on $\mathcal{D}_{ct}$ and $\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{ct}$ and choose the one that performs better on $\mathcal{D}_{cv}$.

We then filter examples that disagree with the k-NN classifier prediction, where the k-NN is computed on the final logit layer of the trained model (i.e. the layer right before softmax).

Algorithm 1 Filtering datapoints via deep k-NN.

Inputs: $\mathcal{D}_{\text{noisy}}$, $\mathcal{D}_{\text{clean}}$, k
Train model $\mathcal{M}$ on either $\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{\text{clean}}$ or $\mathcal{D}_{\text{clean}}$.
Let $\mathcal{N}$ be the activations of $\mathcal{D}_{\text{noisy}} \cup \mathcal{D}_{\text{clean}}$ on the logit layer of $\mathcal{M}$.
$\mathcal{D}_{\text{filtered}} := \{(x, y) \in \mathcal{N} : \eta_k(x) = y\}$, where η_k is computed w.r.t. $\mathcal{N}$.
Train final model on $\mathcal{D}_{\text{filtered}} \cup \mathcal{D}_c$.

4 THEORETICAL ANALYSIS

For the theoretical analysis, we assume the binary classification problem with the features defined on compact set $\mathcal{X} \subseteq \mathbb{R}^D$. We assume that points are drawn according to distribution $\mathcal{F}$ as follows: the features come from distribution $\mathbb{P}_{\mathcal{X}}$ on $\mathcal{X}$ and the labels are distributed according to the measurable conditional probability function $\eta : \mathcal{X} \to [0, 1]$. That is, a sample (X, Y) is drawn from $\mathcal{F}$ as follows: X is drawn according to $\mathbb{P}_{\mathcal{X}}$ and Y is chosen according to $\mathbb{P}(Y = 1|X = x) = \eta(x)$.

The goal will be to show that given corrupted examples, the k-NN disagreement method is still able to identify the examples whose labels do not match that of the Bayes-optimal label.

We will make a few regularity assumptions for our analysis to hold. The first regularity assumption ensures that the support $\mathcal{X}$ does not become arbitrarily thin anywhere. This is a standard non-parametric assumption (e.g. Singh et al. (2009); Jiang (2019)).

Assumption 1 (Support Regularity)**.** *There exists $\omega > 0$ and $r_0 > 0$ such that Vol$(\mathcal{X} \cap B(x,r)) \geq \omega \cdot$ Vol$(B(x,r))$ for all $x \in \mathcal{X}$ and $0 < r < r_0$, where $B(x,r) := \{x' \in \mathcal{X} : |x - x'| \leq r\}$.*

Let $p_{\mathcal{X}}$ be the density function corresponding to $\mathbb{P}_{\mathcal{X}}$. The next assumption ensures that with a sufficiently large sample, we will obtain a good covering of the input space.

Assumption 2 ($p_{\mathcal{X}}$ bounded from below)**.** $p_{X,0} := \inf_{x \in \mathcal{X}} p_X(x) > 0$.

Finally, we make a smoothness assumption on η, as done in other analyses of k-NN classification (e.g. Chaudhuri & Dasgupta (2014); Reeve & Kaban (2019))

Assumption 3 (η Hölder continuous)**.** *There exists $0 < \alpha \leq 1$ and $C_\alpha > 0$ such that $|\eta(x) - \eta(x')| \leq C_\alpha |x - x'|^\alpha$ for all $x, x' \in \mathcal{X}$.*

We propose a notion of how spread out a set of points is. This will allow us to characterize how difficult a configuration of incorrectly labeled examples will be to work with: for example, if such examples are spread out far apart, then there will be many correctly labeled examples nearby for the k-NN approach to identify the incorrectly labeled examples. On the other hand, if the corrupted examples are all close together, then it will be more difficult to identify them without many uncorrected examples in that region. To this end, we define the minimum k-NN spread:

Definition 2 (minimum k-NN spread)**.**

$$S_k(C) := \min_{x \in C} r_k(x, C),$$

where $r_k(x, C)$ denotes the distance from x to the k-th closest neighbor in C.

Also define the Δ-interior region of $\mathcal{X}$ where there is at least Δ margin in the probabilistic label:

Definition 3. *Let $\Delta \geq 0$. Define $\mathcal{X}^\Delta := \{x \in \mathcal{X} : \left|\frac{1}{2} - \eta(x)\right| \geq \Delta\}$.*

We now state the result, which says that with high probability *uniformly* on $\mathcal{X}^\Delta$ when $\Delta > 0$ is known, we have that the label disagrees with the k-NN classifier if and only if the label is not the Bayes-optimal prediction. Due to space, all of the proofs have been deferred to the Appendix.

Theorem 1 (Fixed Δ)**.** *Let $\Delta, \delta > 0$ and suppose Assumptions 1, 2, and 3 hold. There exists constants $K_l, K_u > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from the $\mathcal{F}$ and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range*

$$K_l \cdot \frac{1}{\Delta^2} \cdot \log^2(1/\delta) \cdot \log n \leq k \leq K_u \cdot \min\{S_2(C)^D, \Delta^{D/\alpha}\} \cdot n,$$

then the following holds uniformly over $x \in \mathcal{X}^\Delta$: the k-NN prediction computed w.r.t. X agrees with the label if and only if the label is the Bayes-optimal label $\eta^*(x) := 1[\eta(x) \geq \frac{1}{2}]$.

In the last result, we assumed that Δ was fixed. We next show how we can make a similar guarantee but show that we can take $\Delta \to 0$ as we choose $k, n \to \infty$ appropriately and provide rates of convergence.

Theorem 2 (Rates of convergence for Δ)**.** *Let $\delta > 0$ and suppose Assumptions 1, 2, and 3 hold. There exist constants $K_l, K_u, K > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from $\mathcal{F}$, and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range*

$$K_l \cdot \log^2(1/\delta) \cdot n^{\frac{\alpha}{\alpha+D}} \leq k \leq K_u \cdot S_2(C)^D \cdot n,$$

then the following holds uniformly over $x \in \mathcal{X}^\Delta$: the k-NN prediction computed w.r.t. X agrees with the label if and only if the label is the Bayes-optimal label $\eta^*(x) := 1[\eta(x) \geq \frac{1}{2}]$ where

$$\Delta = K \cdot \left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n}\right)^{\alpha/D} \right).$$

Remark 1. *Choosing $k = O(n^{2\alpha/(2\alpha+D)})$ in the above result gives us $\Delta = \widetilde{O}(n^{-\alpha/(2\alpha+D)})$. This rate for Δ is the minimax-optimal rate for k-nearest neighbor classification on $\mathcal{X}^\Delta$ given a sample of size n (Chaudhuri & Dasgupta, 2014) in the uncorrupted setting. Thus, our analysis is tight up to logarithmic factors.*

We next give results with an additional margin assumption, also known as Tsybakov's noise condition (Mammen et al., 1999; Tsybakov et al., 2004):

Assumption 4 (Tsybakov Noise Condition)**.** *The following holds for some C_β and β and all $\Delta > 0$:*

$$\mathbb{P}_{\mathcal{X}}(x \notin \mathcal{X}^\Delta) \leq C_\beta \cdot \Delta^\beta.$$

Theorem 3 (Rates under Tsybakov Noise Condition)**.** *Let $\delta > 0$ and suppose Assumptions 1, 2, 3 and 4 hold. There exists constants $K_l, K_u, K, K' > 0$ depending only on $\mathcal{F}$ such that the following holds with probability at least $1 - \delta$. Let $X_{[n]}$ be n (uncorrupted) examples drawn from the $\mathcal{F}$ and C be a set of points with corrupted labels and denote our sample $X := X_{[n]} \cup C$. Suppose k lies in the following range*

$$K_l \cdot \log^2(1/\delta) \cdot n^{\frac{\alpha}{\alpha+D}} \leq k \leq K_u \cdot S_2(C)^D \cdot n,$$

and define $\eta_k(x) := \arg\max_y \eta_k(y; x)$. Then,

$$\mathbb{P}\left(\eta_k(x) \neq \eta^*(x)\right) \leq K \cdot \left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n}\right)^{\alpha/D}\right)^{\beta},$$

$$R_X - R^* \leq K' \cdot \left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n}\right)^{\alpha/D}\right)^{\beta+1}$$

where $R_X := \mathbb{E}_{\mathcal{F}}[g_k(x) \neq y]$ and $R^* := \mathbb{E}_{\mathcal{F}}[g^*(x) \neq y]$ denote the risk of the k-NN method and Bayes optimal classifier, respectively.

Remark 2. *Choosing $k = O(n^{2\alpha/(2\alpha+D)})$ in the above gives us a rate of $\widetilde{O}(n^{-\alpha(\beta+1)/(2\alpha+D)})$ for the excess risk. This matches the lower bounds of Audibert et al. (2007) up to logarithmic factors.*

5 EXPERIMENTS

We evaluate the effectiveness of our algorithm as follows. We split each dataset's training set into two parts, $\mathcal{D}_{\text{clean}}$ and $\mathcal{D}_{\text{noisy}}$. We then corrupt the labels of some fraction of examples in $\mathcal{D}_{\text{noisy}}$ by applying a corruption matrix prescribed by one of the following methods.

- **Uniform**: The label is flipped to any one of the labels (including itself) with equal probability.
- **Flip**: The label is flipped to any *other* label with equal probability.
- **Hard Flip**: With probability $\frac{1}{2}$, we flip the label m to $\pi(m)$ where π is some predefined permutation of the labels.

We compare against the following baselines:

- **Gold Loss Correction (GLC)** (Hendrycks et al., 2018) estimates the corruption matrix by averaging the softmax outputs of the clean examples on a model trained on noisy data.
- **Distill** (Li et al., 2017) assigns each example in the combined dataset a "soft" label that is a convex combination of its label and its softmax output from a model trained solely on clean data.
- **Forward** (Patrini et al., 2017), similar in spirit to GLC, estimates the corruption matrix by training a model on noisy data and using the softmax output for prototype examples for each class. It does not require a clean dataset like other methods.
- **Control.** We define this as training on the clean data only.

We report test errors and show the average across multiple runs with standard error bands shaded. Errors are computed on 11 uniformly distributed noise rates between 0 and 1 inclusive. For the results shown in the main text, we have that $\mathcal{D}_{\text{clean}}$ is randomly selected and is 5% of the data. In the Appendix, we show results over different sizes of $\mathcal{D}_{\text{clean}}$. We implement all methods using the Tensorflow 2.0 Keras API and Scikit-Learn. We use the Adam optimizer with default learning rate 0.001 and a batch size of 128 across all experiments. For the UCI datasets, we set $k = 50$ and set $k = 500$ for all other datasets. We chose $k = 50$ for the UCI datasets because some of the datasets were of small size. However, we found that the k-NN method's performance was quite stable to the choice of k, which we show in Section 5.5. We describe the permutations used for hard flipping in the appendix.

5.1 UCI RESULTS

We show the results for UCI datasets in Figure 1. Due to space, we were only able to show two of the datasets here, the rest can be found in the Appendix. For all of the UCI datasets, we use a fully-connected neural network with a single hidden layer of dimension 100 with ReLU activations and train for 100 epochs.

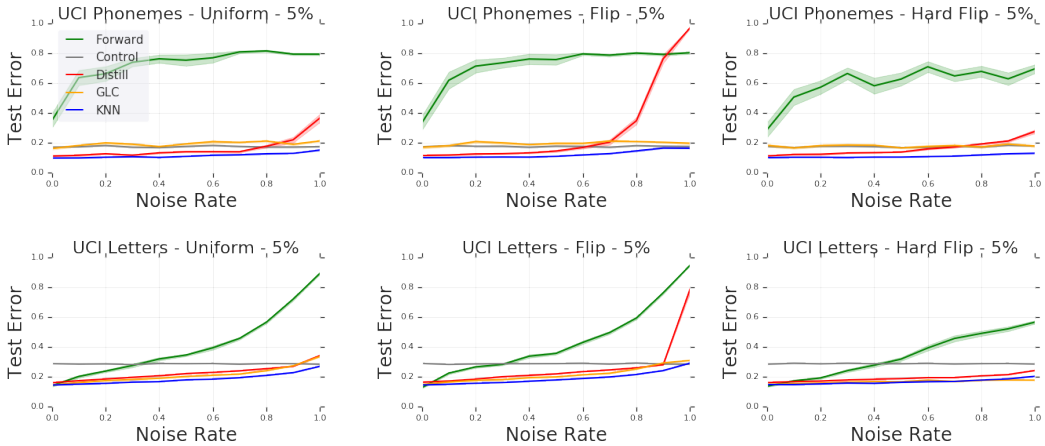

Figure 1: **UCI Results**: Error plots against amount of noise applied to the labels of $\mathcal{D}_{\text{noisy}}$. $\mathcal{D}_{\text{clean}}$ contains 5% of the data. Each column is a different corruption and each row is for a different dataset. We see that the k-NN method consistently chooses the best datapoints to filter leading to lower error. More results across different UCI datasets and clean dataset sizes can be found in the Appendix.

5.2 MNIST AND FASHION MNIST RESULTS

We show the results for MNIST and Fashion MNIST in Figure 2. For both of these datasets, the model we use is a two hidden-layer fully-connected neural network where each layer has 256 hidden units with ReLU activations. We train the model for 20 epochs. We see that the k-NN approach attains models with a low error rate across noise rates and either outperforms or is competitive with the next best method, GLC.

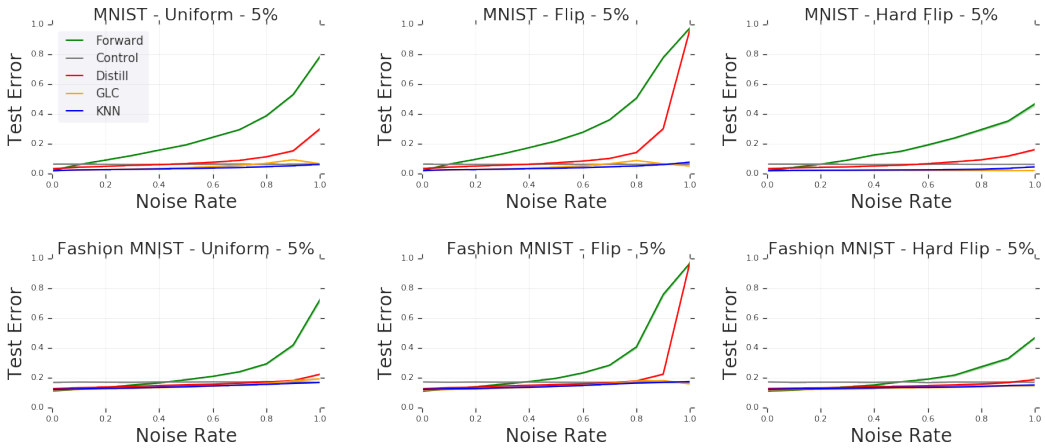

Figure 2: **Top: MNIST, bottom: Fashion MNIST**. Each column is a different corruption method. We see that the k-NN approach performs competitively. More results across different corruption rates can be found in the Appendix.

5.3 CIFAR RESULTS

For CIFAR10/100 we use ResNet-20, which we train from scratch on single NVIDIA P100 GPUs. We train CIFAR10 for 100 epochs and CIFAR100 for 150 epochs. We show the results in Figure 3. We see that the k-NN method performs competitively. It generally outperforms on the uniform and flip noise types but performs worse for the hard flip noise type. It is not too surprising that k-NN would be weaker in the presence of hard flip noise (i.e. where labels are mapped based on a pre-determined mapping between labels) as the noise is much more structured in that case making it more difficult to be filtered out by majority vote among the neighbors. In other words, unlike the uniform and flip noise types, we are no longer dealing with *white label noise* in the hard flip noise type.

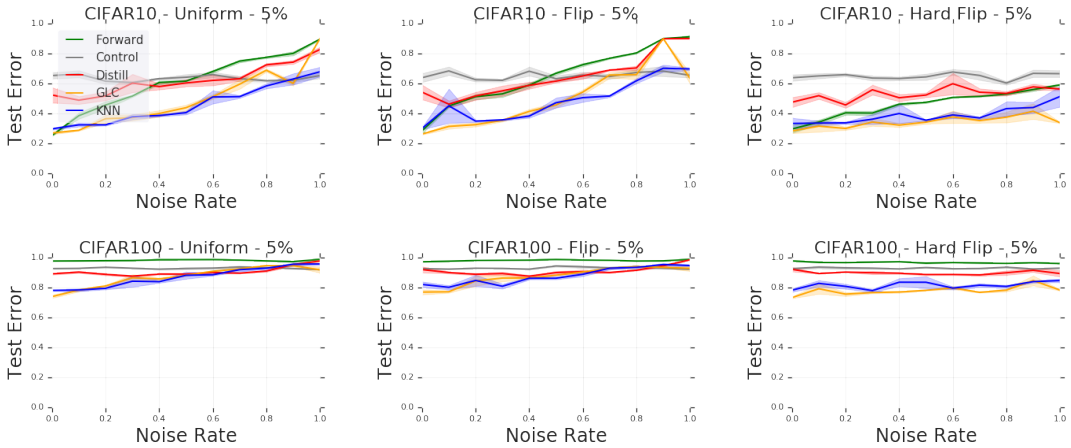

Figure 3: **Top: CIFAR10, bottom: CIFAR100**. Each column is a different corruption method. We see that our k-NN method performs competitively or outperforms on the uniform and flip noise types but performs worse for the hard flip noise type. More results across different corruption rates can be found in the Appendix.

5.4 SVHN RESULTS

We show the results in Figure 4. We train ResNet-20 from scratch on NVIDIA P100 GPUs for 100 epochs. As in the CIFAR experiments, we see that the k-NN method tends to be competitive in the uniform and flip noise types but does slightly worse in the hard flip.

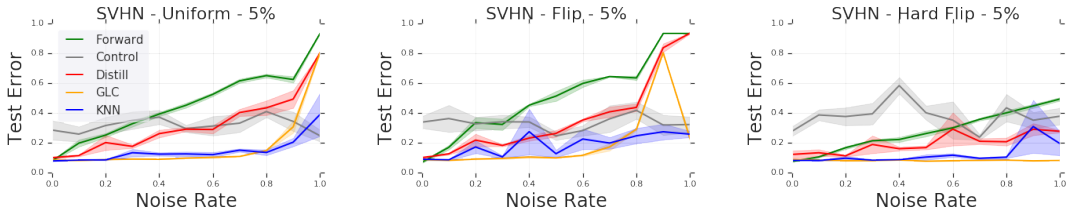

Figure 4: We see that the k-NN method performs competitively on the uniform and flip noise types but performs worse for the hard flip noise type. More results in the Appendix.

5.5 ROBUSTNESS TO k

In this section, we show that our procedure is stable in its hyperparameter k. The theoretical results suggest that a wide range of k can give us statistical consistency guarantees and we show that in practice a wide range of k gives us similar results for Algorithm 1 (Figure 5). Such robustness in

hyperparameter is highly desirable because optimal tuning is often not available, especially when no sufficient clean validation set is available.

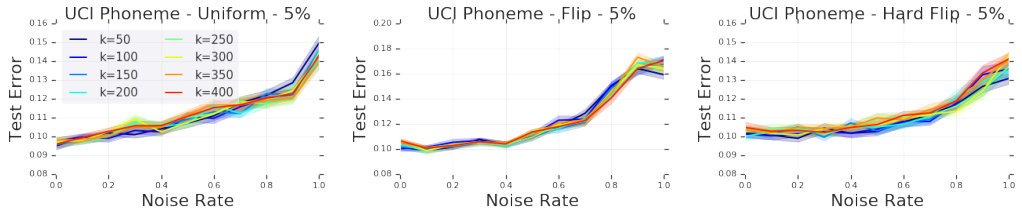

Figure 5: **Performance across different values of** k Here we show that on a UCI dataset, the performance of Algorithm 1 is stable when varying its hyperparameter k. Note that the y-axis has been zoomed in to better see the differences between the curves.

5.6 AREA UNDER ERROR VS NOISE LEVEL CURVE ACROSS DATASETS

In the figures shown so far, it may be difficult to compare the curves in some cases so we report an area under the curve metric in Table 1.

		Uniform					**Flip**					**Hard Flip**				
Dataset	**% Clean**	Forward	Control	Distill	GLC	k-NN	Forward	Control	Distill	GLC	k-NN	Forward	Control	Distill	GLC	k-NN
Letters	5	4.55	3.16	2.48	2.33	**2.05**	4.83	3.16	2.96	2.35	**2.10**	3.77	3.17	2.10	1.83	**1.82**
	10	4.28	2.51	2.05	1.91	**1.78**	4.45	2.52	2.12	1.92	**1.80**	3.63	2.52	1.85	1.60	**1.60**
	20	3.77	2.06	1.76	1.57	**1.56**	3.85	2.07	1.78	**1.56**	1.58	3.35	2.05	1.62	**1.38**	1.41
Phonemes	5	7.89	1.91	1.79	2.12	**1.26**	7.91	1.93	3.21	2.16	**1.34**	6.60	1.92	1.77	1.96	**1.20**
	10	7.86	1.54	1.53	1.67	**1.16**	7.95	1.54	2.75	1.69	**1.22**	6.73	1.55	1.61	1.60	**1.15**
	20	7.72	1.34	1.33	1.35	**1.13**	7.89	1.34	1.97	1.35	**1.16**	6.36	1.33	1.45	1.30	**1.14**
Wilt	5	5.18	0.56	0.85	0.54	**0.39**	5.27	0.56	3.89	0.53	**0.52**	4.60	0.55	0.73	0.58	**0.39**
	10	4.68	0.43	0.75	0.45	**0.32**	5.63	0.44	3.14	0.43	**0.41**	4.78	0.43	0.66	0.43	**0.31**
	20	4.31	0.36	0.86	0.35	**0.32**	5.18	0.34	2.67	0.35	**0.34**	5.63	0.34	0.61	0.36	**0.30**
Seeds	5	3.29	4.22	3.71	4.20	**3.08**	5.13	4.33	5.56	4.39	**3.64**	2.94	4.06	3.63	3.91	**2.99**
	10	3.43	3.04	2.84	2.99	**2.14**	5.02	3.38	4.58	3.19	**2.65**	2.75	3.42	2.96	3.14	**2.25**
	20	2.99	2.69	2.27	2.74	**1.72**	4.41	2.56	3.57	2.65	**2.09**	2.85	2.53	2.25	2.57	**1.62**
Iris	5	2.97	3.23	3.48	3.97	**2.46**	5.25	3.38	5.15	4.03	**3.02**	2.90	3.29	3.13	4.10	**2.32**
	10	2.89	2.48	2.59	2.25	**1.32**	5.06	2.53	4.58	2.24	**2.17**	2.60	2.34	2.02	1.86	**1.15**
	20	2.46	1.85	1.97	1.52	**0.60**	4.46	1.51	3.45	**1.38**	1.46	2.51	1.84	1.48	1.34	**0.58**
Parkinsons	5	5.00	3.46	**3.26**	4.22	3.40	5.17	3.55	4.49	4.10	**3.36**	4.98	3.71	**3.56**	4.59	3.68
	10	5.35	3.26	3.22	3.45	**3.21**	5.43	3.38	4.25	3.44	**3.32**	5.24	3.26	**3.10**	3.37	3.11
	20	4.88	3.01	3.08	3.10	**2.98**	5.19	3.02	3.94	3.05	**2.95**	5.10	2.98	3.01	2.98	**2.96**
MNIST	5	2.88	0.69	1.03	0.50	**0.40**	3.60	0.69	1.91	0.50	**0.44**	2.03	0.69	0.78	**0.22**	0.29
	10	2.57	0.50	0.85	0.41	**0.33**	3.22	0.50	1.50	0.42	**0.35**	1.86	0.50	0.67	**0.21**	0.26
	20	2.07	0.35	0.69	0.34	**0.27**	2.48	0.35	0.86	0.34	**0.27**	1.54	0.35	0.53	**0.20**	0.22
Fashion MNIST	5	2.76	1.88	1.73	1.59	**1.56**	3.55	1.87	2.55	**1.59**	1.60	2.30	1.87	1.62	**1.44**	1.48
	10	2.47	1.71	1.60	1.52	**1.52**	3.14	1.71	2.13	**1.53**	1.54	2.14	1.71	1.52	**1.41**	1.46
	20	2.07	1.56	1.48	1.45	**1.44**	2.38	1.56	1.54	1.46	**1.46**	1.92	1.56	1.43	**1.38**	1.41
CIFAR10	5	6.74	7.00	6.86	5.43	**5.03**	7.14	7.20	7.12	5.52	**5.35**	5.08	7.13	5.85	**3.76**	4.27
	10	6.58	6.58	6.32	5.39	**5.27**	6.82	6.56	6.72	5.62	**5.32**	4.89	6.53	5.30	**3.91**	4.40
	20	6.40	5.52	5.66	5.11	**4.57**	6.62	5.59	5.90	5.16	**4.85**	4.77	5.45	4.68	**3.51**	3.82
CIFAR100	5	10.80	10.22	9.98	9.59	**9.57**	10.79	10.24	10.03	**9.64**	9.66	10.64	10.23	9.88	**8.58**	8.98
	10	10.79	9.94	9.70	**9.42**	9.63	10.81	9.89	9.68	**9.46**	9.63	10.65	9.89	9.38	**8.56**	9.19
	20	10.78	9.38	9.15	**9.06**	9.23	10.80	9.44	9.13	**8.97**	9.17	10.66	9.41	8.65	**8.07**	8.81
SVHN	5	5.04	3.52	3.56	1.99	**1.62**	5.60	3.65	4.09	2.17	**2.05**	3.04	4.17	2.16	**0.89**	1.35
	10	4.98	2.20	3.20	2.27	**1.34**	5.50	2.28	3.72	2.29	**1.48**	2.98	2.41	2.01	**0.88**	1.04
	20	4.32	1.83	2.67	2.14	**1.20**	4.85	1.83	3.09	1.90	**1.19**	2.70	1.85	1.54	0.98	**0.98**

Table 1: **Area under the test error vs noise rate curve**: Each row corresponds to a dataset and size of clean dataset $\mathcal{D}_{\text{clean}}$ pair, where the size is a percentage of the total training set (5%, 10%, 20%). Each column shows the area under the error curve across noise rates for a particular method and noise type (Uniform, Flip, Hard Flip). We see that the k-NN method consistently outperforms the other methods for Uniform and Flip and outperforms the other methods on Hard Flip on the smaller datasets.

Conclusions and Open Questions We conclude from our experiments and theory that the k-NN based method (Algorithm 1) is a relatively safe method to remove problematic training examples before training. While k-NN methods can be sensitive to the choice of k when used with small datasets (Garcia et al., 2009), we hypothesize that with today's large datasets one can blithely set k to a fixed practically medium-sized value (e.g. $k = 500$) as done here and expect reasonable performance. Theoretically we provided some new results for how well k-NN can identify clean versus corrupted labels. Open theoretical questions are whether there are alternate notions of how to characterize the difficulty of a particular configuration of corrupted examples and whether we can provide both upper and lower learning bounds under these noise conditions.

REFERENCES

Jean-Yves Audibert, Alexandre B Tsybakov, et al. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633, 2007.

Carla E. Brodley and Mark A. Freidl. Identifying mislabeled training data. *Journal Artificial Intelligence Research*, 1999.

J Paul Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59(2):467–479, 2011.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pp. 3437–3445, 2014.

Xu Chu, Ilhab F. Ilyas, Sanjay Krishnan, and Jiannan Wan. Data cleaning: Overview and emerging challenges. In *SIGMOD*, 2016.

Thomas M Cover. Rates of convergence for nearest neighbor procedures. In *Proceedings of the Hawaii International Conference on Systems Sciences*, pp. 413–415, 1968.

Luc Devroye, Laszlo Gyorfi, Adam Krzyzak, Gábor Lugosi, et al. On the strong universal consistency of nearest neighbor regression function estimates. *The Annals of Statistics*, 22(3):1371–1385, 1994.

Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley, 1951.

Wei Gao, Xin-Yi Niu, and Zhi-Hua Zhou. On the consistency of exact and approximate nearest neighbor with noisy data. *arXiv preprint arXiv:1607.07526*, 2016.

E. K. Garcia, S. Feldman, M. R. Gupta, and S. Srivastava. Completely lazy learning. *IEEE Trans. on Knowledge and DataEngineering*, 2009.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

I. Guyon, N. Matic, and V. Vapnik. Discovering informative patterns and data cleaning. In *AAAI Workshop on Knowledge Discovery in Databases*, 1994.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *Advances in Neural Information Processing Systems*, pp. 5836–5846, 2018.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pp. 10456–10465, 2018.

H. Jiang, B. Kim, M. Y. Guan, and M. R. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Heinrich Jiang. Non-asymptotic uniform rates of consistency for k-nn regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3999–4006, 2019.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 4, 2017.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 967–972. IEEE, 2016.

Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1910–1918, 2017.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015.

Enno Mammen, Alexandre B Tsybakov, et al. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in Neural Information Processing Systems*, pp. 1049–1056, 2009.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pp. 1196–1204, 2013.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

Henry WJ Reeve and Ata Kaban. Fast rates for a kNN classifier robust to unknown asymmetric label noise. *arXiv preprint arXiv:1906.04542*, 2019.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

Aarti Singh, Clayton Scott, Robert Nowak, et al. Adaptive Hausdorff estimation of density level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.

Charles J Stone. Consistent nonparametric regression. *The Annals of Statistics*, pp. 595–620, 1977.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

Alexander B Tsybakov et al. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pp. 5596–5605, 2017.

Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pp. 10–18, 2015.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 839–847, 2017.

Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *International Conference on Machine Learning*, pp. 5120–5129, 2018.

D. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. on Systems, Man and Cybernetics*, 1972.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2691–2699, 2015.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8778–8788, 2018.

A PROOFS

A.1 SUPPORTING THEORETICAL RESULTS

The following bounds $r_k(x)$ uniformly in $x \in \mathcal{X}$.

Lemma 1 (Lemma 2 of Jiang (2019))**.** *The following holds with probability at least* $1 - \delta/2$*. If*

$$2^8 \cdot D \log^2(4/\delta) \cdot \log n \leq k \leq \frac{1}{2} \cdot \omega \cdot p_{X,0} \cdot v_D \cdot r_0^D \cdot n,$$

then $\sup_{x \in \mathcal{X}} r_k(x) \leq \left(\frac{2k}{\omega \cdot v_D \cdot n \cdot p_{X,0}}\right)^{1/D}$*, where* v_D *is the volume of the unit ball in* $\mathbb{R}^D$*.*

The next result bounds the number of distinct k-NN sets over $\mathcal{X}$.

Lemma 2 (Lemma 3 of Jiang (2019))**.** *Let* M *be the number of distinct* k*-NN sets over* $\mathcal{X}$*, that is,* $M := |\{N_k(x) : x \in \mathcal{X}\}|$*. Then* $M \leq D \cdot n^D$*.*

A.2 PROOF OF THEOREM 1

Proof of Theorem 1. Let $\tau, \gamma, \epsilon > 0$ be quantities that will be determined later. Suppose that for some $x \in \mathcal{X}^\Delta$, we have $r_k(x) \leq \tau$ and $S_{\lfloor(\frac{1}{2}-\gamma)\cdot k\rfloor}(C) \geq \tau$. Then, at least $\frac{1}{2} + \gamma$ fraction of the points within x's k-nearest neighbors are not in the corrupted set C. Let $A_x := N_k(x) \backslash C$, that is, the k-nearest neighbors of x that are not in C. Then it is clear that A_x is a k_0-nearest neighbor set of x relative to $X \backslash C$ for some $k_0 \geq \lceil(\frac{1}{2} + \gamma) \cdot k\rceil$. We have that $A_x \subseteq \mathcal{X}^\Delta \oplus \tau$ where $A \oplus r := \{x \in \mathcal{X} : \inf_{a \in A} |x - a| \leq r\}$. Let us consider without loss of generality that $\eta(x) \geq \frac{1}{2} + \Delta$ (call this set $\mathcal{X}^{\Delta,+}$). The case $\mathcal{X}^{\Delta,-} := \{x \in \mathcal{X}^\Delta : \eta(x) \leq \frac{1}{2} - \Delta\}$ follows by symmetry. Thus, we have $\eta(x') \geq \frac{1}{2} + \Delta - C_\alpha \tau^\alpha$ for all $x' \in A_x$. By Hoeffing's inequality, we have

$$\mathbb{P}\left(\frac{1}{|A_x|} \sum_{x' \in A_x} y(x') < \frac{1}{2} + \Delta - C_\alpha \tau^\alpha - \epsilon\right) \leq \exp(-2\epsilon^2 \cdot k_0),$$

where $y(x)$ is the label corresponding to sample x. By Lemma 2, we have that there are at most $D \cdot n^D$ such k_0-nearest neighbor sets across all k_0 in $X \backslash C$. That is, this is also a bound on the number of distinct A_x for $x \in \mathcal{X}$. Therefore, if we set

$$\epsilon = \sqrt{\frac{D \log n + \log(4D/\delta)}{(1 + 2\gamma) \cdot k}},$$

then by union bound, we have that

$$\mathbb{P}\left(\inf_{x \in \mathcal{X}^{\Delta,+}} \frac{1}{|A_x|} \sum_{x' \in A_x} y(x') < \frac{1}{2} + \Delta - C_\alpha \tau^\alpha - \epsilon\right) \leq \frac{\delta}{4}.$$

and thus, with probability at least $1 - \delta/4$, we have that $\frac{1}{|A_x|} \sum_{x' \in A_x} y(x') \geq \frac{1}{2} + \Delta - C_\alpha \tau^\alpha - \epsilon$ *uniformly* over $x \in \mathcal{X}^{\Delta,+}$. Similarly, with probability at least $1 - \delta/4$ we have that $\frac{1}{|A_x|} \sum_{x' \in A_x} y(x') \leq \frac{1}{2} - \Delta + C_\alpha \tau^\alpha + \epsilon$ *uniformly* over $x \in \mathcal{X}^{\Delta,-}$.

Hence, in order for k-nearest neighbor prediction to predict the Bayes-optimal label on $\mathcal{X}^\Delta$, it suffices that

$$k_0 \left(\frac{1}{2} + \Delta - C_\alpha \tau^\alpha - \epsilon\right) \geq \frac{k}{2}.$$

Since $k_0 \geq (\frac{1}{2} + \gamma) \cdot k$, we have that the above holds if

$$\Delta \geq C_\alpha \tau^\alpha + \epsilon + \frac{1 - 2\gamma}{2(1 + 2\gamma)}.$$

We now choose the values of τ, γ, ϵ to upper bound each of the terms on the R.H.S. by $\Delta/3$ so that the above holds.

We can bound the last term by $\Delta/3$ by setting:

$$\gamma = \frac{1}{2} \cdot \frac{3 - 2\Delta}{3 + 2\Delta}.$$

Next, taking

$$k \geq \frac{3(3 + 2\Delta)}{2\Delta^2} \left(D \log n + \log(4D/\delta) \right),$$

we have that $\epsilon \leq \Delta/3$. Now, by Lemma 1, we have that setting

$$\tau = \left(\frac{2k}{\omega \cdot v_D \cdot n \cdot p_{X,0}} \right)^{1/D}$$

gives us that $r_k(x) \leq \tau$ for all $x \in \mathcal{X}$ with probability at least $1 - \delta/2$. It thus suffices to take

$$k \leq \frac{1}{2} \left(\frac{\Delta}{3 \cdot C_\alpha} \right)^{D/\alpha} \cdot \omega \cdot v_D \cdot p_{X,0} \cdot n$$

so that $C_\alpha \tau^\alpha \leq \Delta/3$. Now in order for $S_{\lfloor (\frac{1}{2} - \gamma) \cdot k \rfloor}(C) \geq \tau$, it suffices to have $S_2(C) \geq \tau$. This can be accomplished by having the following hold:

$$k \leq \frac{1}{2} \cdot S_2(C)^D \cdot \omega \cdot v_D \cdot p_{X,0} \cdot n.$$

Thus, there exists positive constants K_l and K_u depending only on $\mathcal{F}$ such that if

$$K_u \cdot \frac{1}{\Delta^2} \cdot (\log^2(1/\delta) \cdot \log n) \leq k \leq K_u \cdot \min\{S_2(C)^D, \Delta^{D/\alpha}\} \cdot n,$$

then the desired conditions hold. □

A.3 Proof of Theorem 2

Proof of Theorem 2. The proof begins in the same way as the proof of Theorem 1. As before, let $\tau, \gamma, \epsilon > 0$ be quantities that will be determined later. Like before, we are reduced to showing

$$\Delta \geq C_\alpha \tau^\alpha + \epsilon + \frac{1 - 2\gamma}{2(1 + 2\gamma)},$$

as long as the conditions for Lemma 1 and 2 hold and $S_2(x) \geq \tau$ and $r_k(x) \leq \tau$ where we choose

$$\epsilon = \sqrt{\frac{D \log n + \log(4D/\delta)}{(1 + 2\gamma) \cdot k}}, \quad \gamma = \frac{1}{2} \cdot \frac{3 - 2\Delta}{3 + 2\Delta}, \quad \tau = \left(\frac{2k}{\omega \cdot v_D \cdot n \cdot p_{X,0}} \right)^{1/D}.$$

These conditions hold for some K_u and K_l depending on $\mathcal{F}$. Then we are reduced to having

$$\frac{2}{3}\Delta \geq \sqrt{\frac{D \log n + \log(4D/\delta)}{(1 + 2\gamma) \cdot k}} + C_\alpha \left(\frac{2k}{\omega \cdot v_D \cdot n \cdot p_{X,0}} \right)^{\alpha/D}.$$

Since $\gamma \geq 0$, it suffices to have

$$\frac{2}{3}\Delta \geq \sqrt{\frac{D \log n + \log(4D/\delta)}{k}} + C_\alpha \left(\frac{2k}{\omega \cdot v_D \cdot n \cdot p_{X,0}} \right)^{\alpha/D}.$$

The desired form for Δ clearly follows for some choice of K depending only on $D, \omega, p_{X,0}, C_\alpha$, all of which depend only on $\mathcal{F}$.

Finally, we must ensure that $\lfloor (\frac{1}{2} - \gamma) \cdot k \rfloor \geq 2$ so that $S_{\lfloor (\frac{1}{2} - \gamma) \cdot k \rfloor}(C) \geq S_2(x)$. Given the expression for γ, it is equivalent to have $\lfloor \left(\frac{2\Delta}{3 + 2\Delta} \right) \cdot k \rfloor \geq 2$. It suffices to show that $k \geq \frac{3(3+2\Delta)}{2\Delta}$. Given the form of Δ in terms of n and k, we see that it suffices to have that

$$k \geq \frac{9}{2} \cdot K \cdot \left(\left(\sqrt{\frac{\log n + \log(1/\delta)}{k}} + \left(\frac{k}{n} \right)^{\alpha/D} \right) \right)^{-1} + 3,$$

which holds when $k \geq K_0 \cdot n^{2\alpha/(2\alpha + D)}$ for some K_0 depending only on $\mathcal{F}$, as desired. □

A.4 PROOF OF THEOREM 3

Proof of Theorem 3. The first part follows from Theorem 2. For the second part, we have by Theorem 2 that if $x \in \mathcal{X}^{\Delta}$, then the k-NN classifier and the Bayes-optimal classifier match with probability $1-\delta$ uniformly. Thus, we have

$$\begin{aligned} R_X - R^* &\leq \mathbb{P}(x \notin \mathcal{X}^{\Delta}) \left(\mathbb{E}_{\mathcal{F}}[g_k(x) \neq y | x \notin \mathcal{X}^{\Delta}] - \mathbb{E}_{\mathcal{F}}[g^*(x) \neq y | x \notin \mathcal{X}^{\Delta}]\right) \\ &\leq C_\beta \cdot \Delta^\beta \cdot \left(\mathbb{E}_{\mathcal{F}}[g_k(x) \neq y | x \notin \mathcal{X}^{\Delta}] - \mathbb{E}_{\mathcal{F}}[g^*(x) \neq y | x \notin \mathcal{X}^{\Delta}]\right) \leq C_\beta \cdot \Delta^\beta \cdot 2\Delta. \end{aligned}$$

The result follows immediately from Theorem 2. □

B HARD FLIP PERMUTATIONS

For Fashion MNIST we hard flip by swapping the following classes: TSHIRT ↔ SHIRT, TROUSER ↔ DRESS, PULLOVER ↔ COAT, SANDAL ↔ BAG, SNEAKER ↔ ANKLEBOOT. For CIFAR10 we swap the pairs: TRUCK ↔ AUTOMOBILE, BIRD ↔ AIRPLANE, DEER ↔ HORSE, CAT ↔ DOG, FROG ↔ SHIP. For CIFAR100, we hard flip circularly (i.e. $\pi(i) = (i+1) \mod K$) within each of the 20 superclasses. For all other datasets, we hard flip circularly.

C ADDITIONAL PLOTS

We provide the plots that were ommitted from the main text due to space constraints.

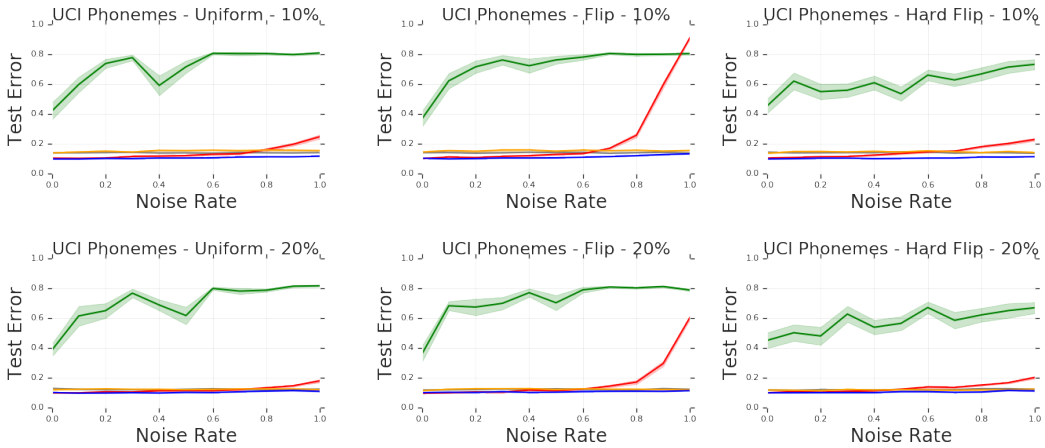

Figure 6: Plots for UCI Phonemes dataset at 10, 20% clean data and all corruption types.

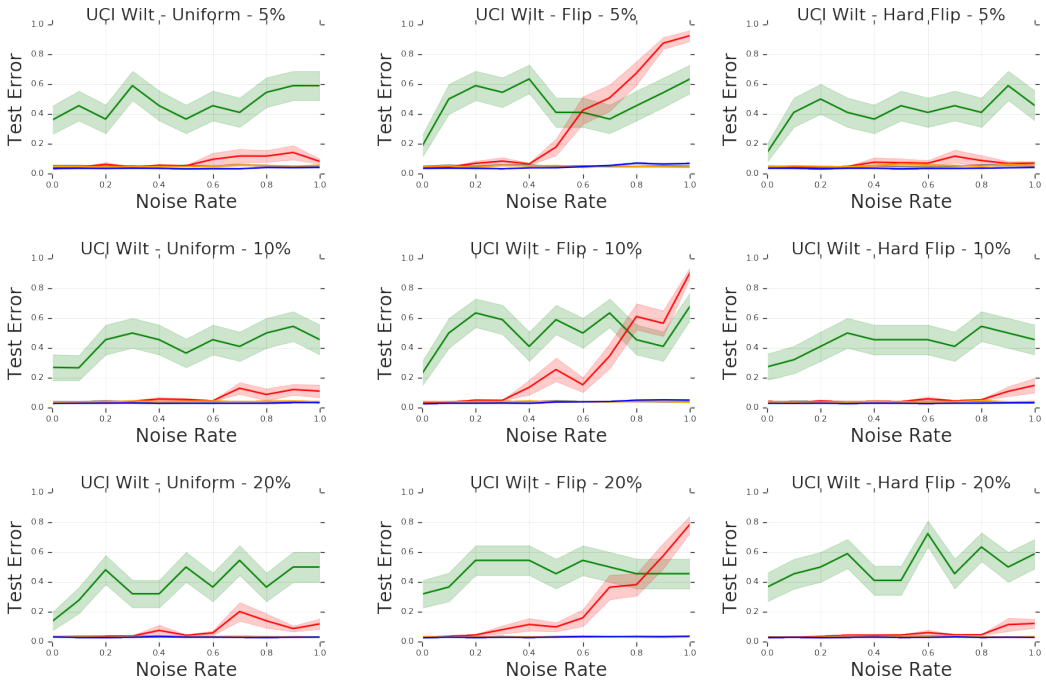

Figure 8: Plots for UCI Wilt dataset at 5, 10, 20% clean data and all corruption types.

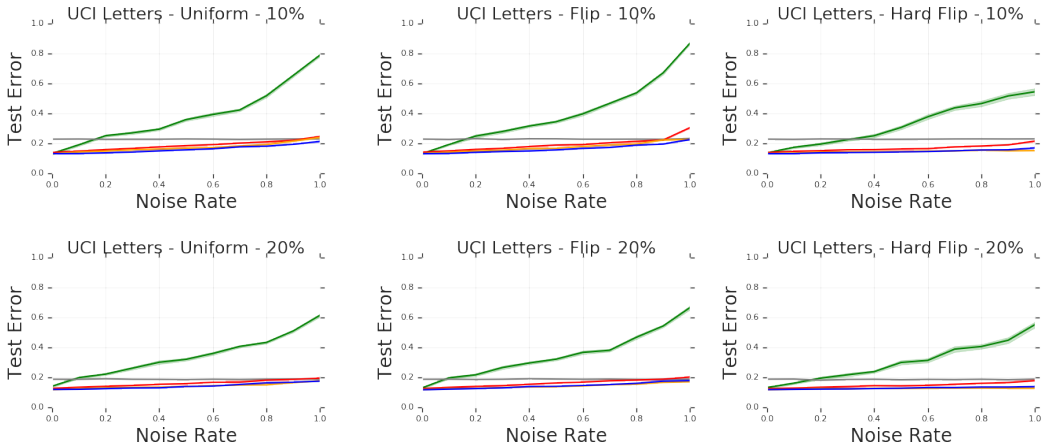

Figure 7: Plots for UCI Letters dataset at 10, 20% clean data and all corruption types.

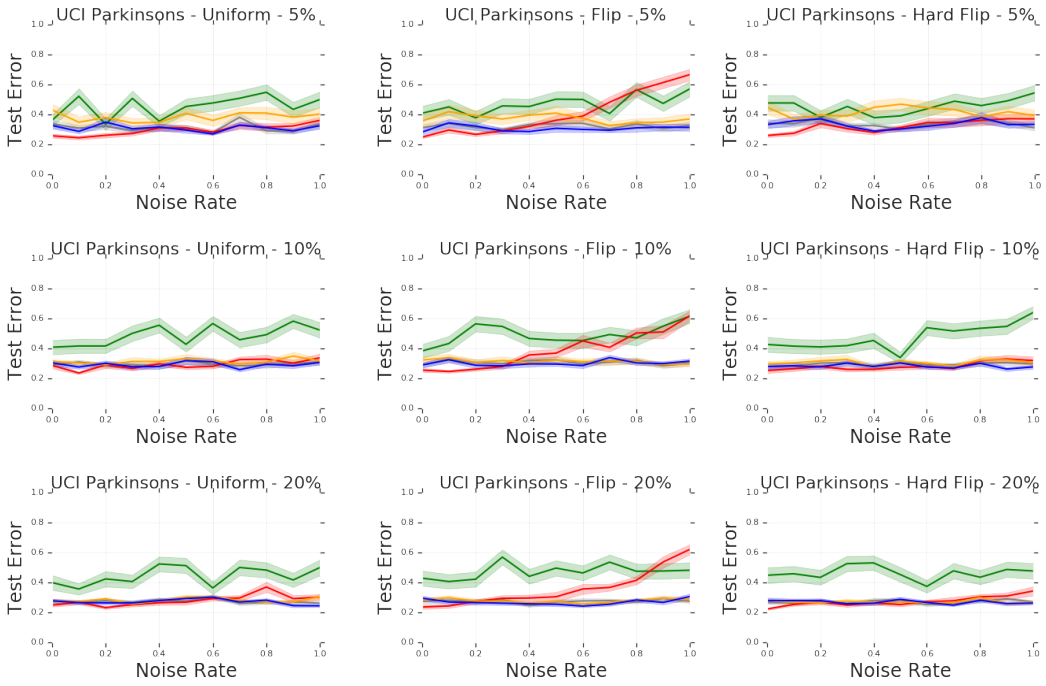

Figure 9: Plots for UCI Parkinsons dataset at 5, 10, 20% clean data and all corruption types.

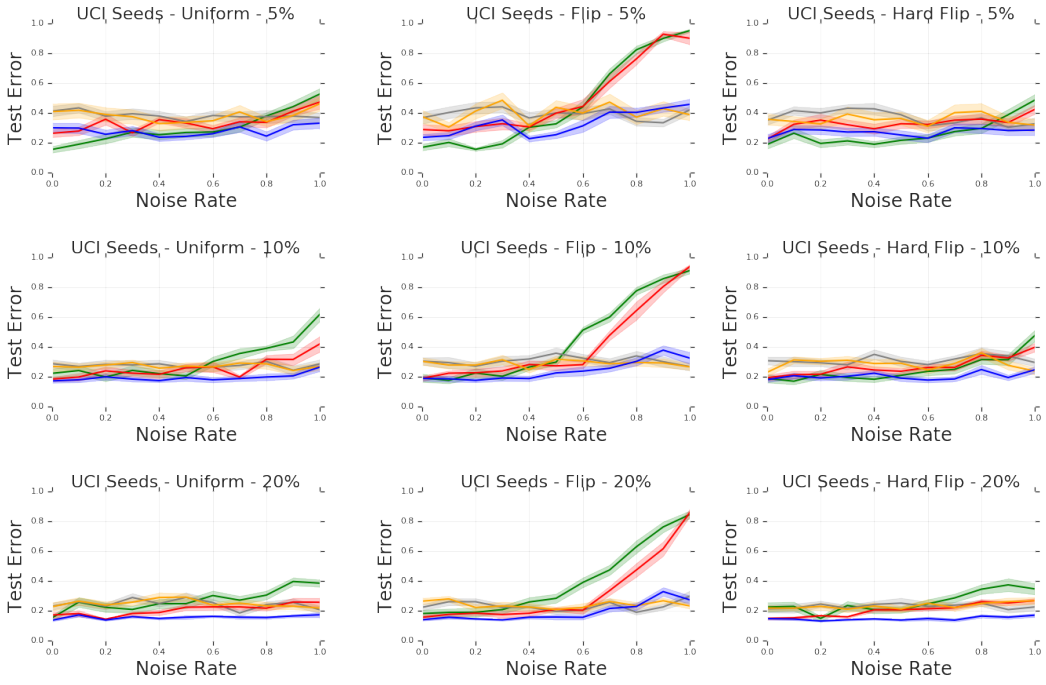

Figure 10: Plots for UCI Seeds dataset at 5, 10, 20% clean data and all corruption types.

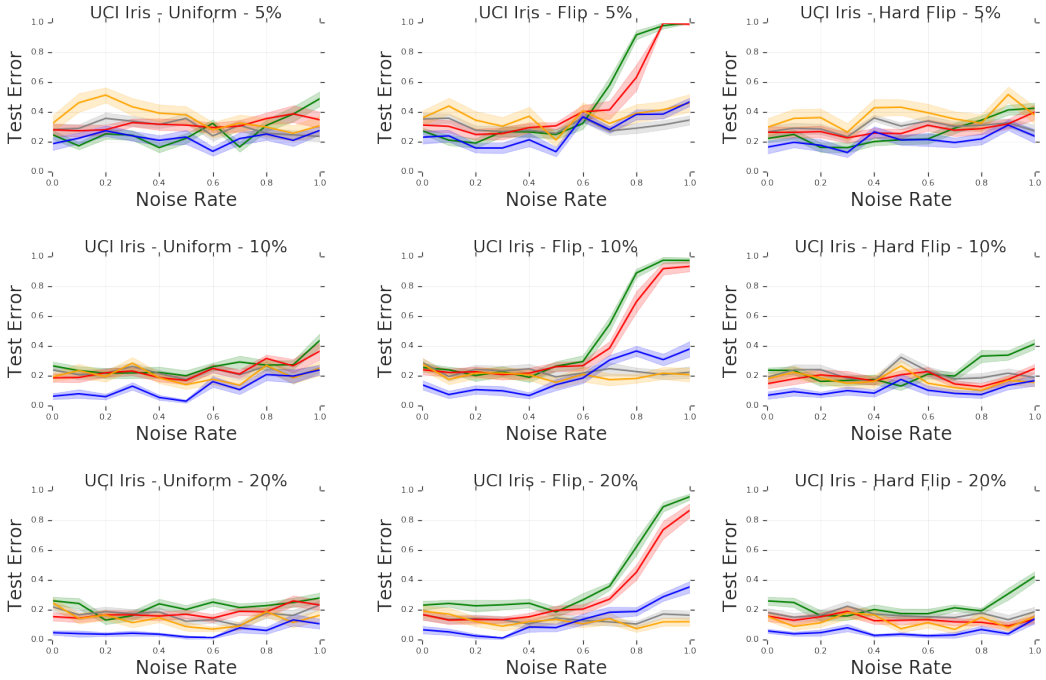

Figure 11: Plots for UCI Iris dataset at 5, 10, 20% clean data and all corruption types.

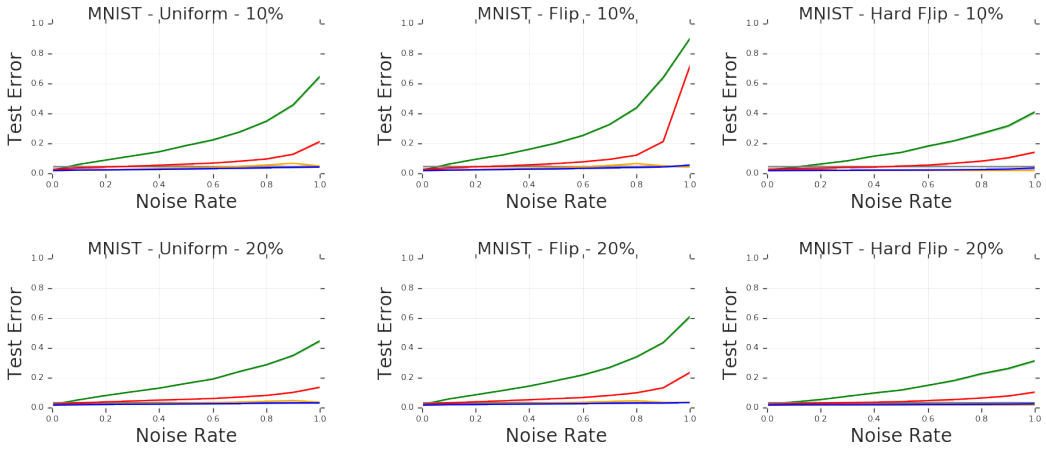

Figure 12: Plots for MNIST at 10, 20% clean data and all corruption types.

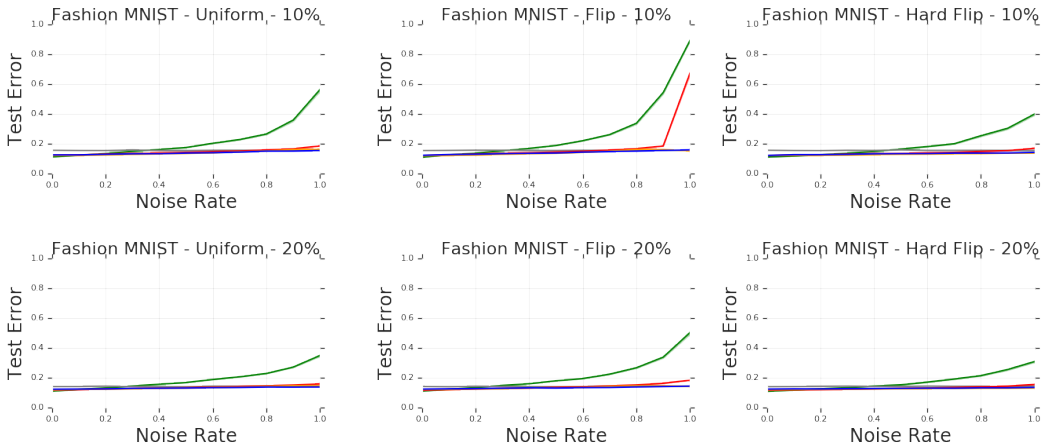

Figure 13: Plots for Fashion MNIST at 10, 20% clean data and all corruption types.

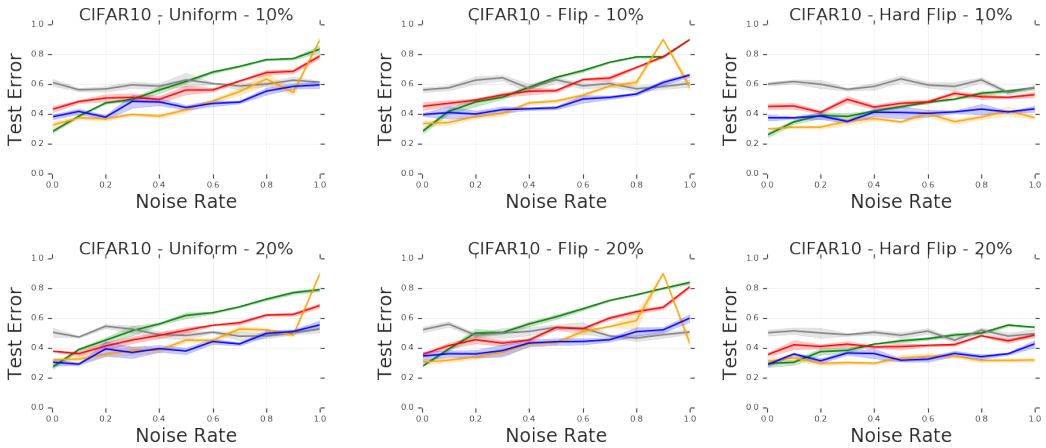

Figure 14: Plots for CIFAR10 at 10, 20% clean data and all corruption types.

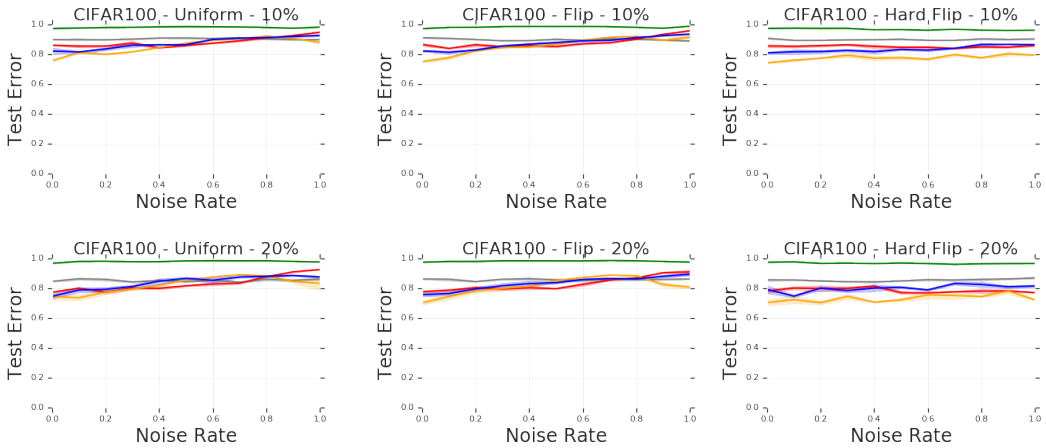

Figure 15: Plots for CIFAR100 at 10, 20% clean data and all corruption types.

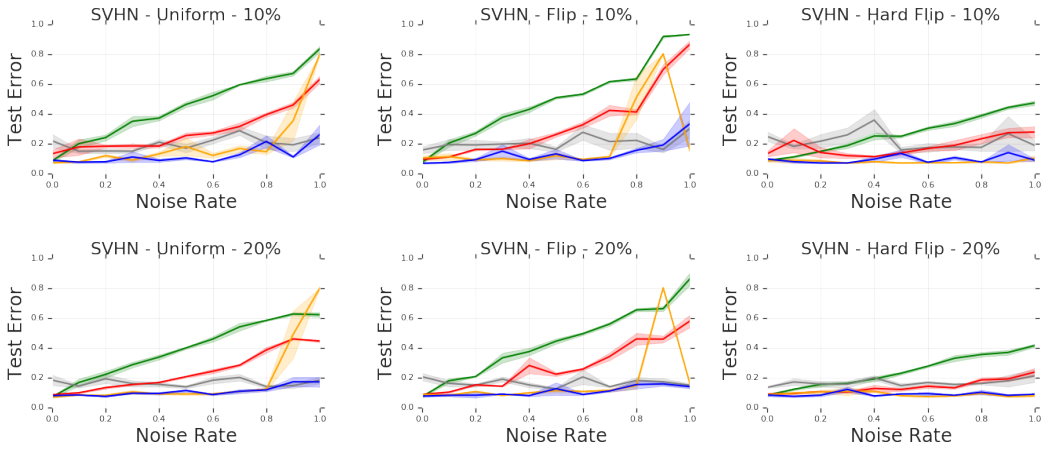

Figure 16: Plots for SVHN at 10, 20% clean data and all corruption types.