HETEROSKEDASTIC AND IMBALANCED DEEP LEARNING WITH ADAPTIVE REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world large-scale datasets are heteroskedastic and imbalanced – labels have varying levels of uncertainty and label distributions are long-tailed. Heteroskedasticity and imbalance challenge deep learning algorithms due to the difficulty of distinguishing among mislabeled, ambiguous, and rare examples. Addressing heteroskedasticity and imbalance simultaneously is under-explored. We propose a data-dependent regularization technique for heteroskedastic datasets that regularizes different regions of the input space differently. Inspired by the theoretical derivation of the optimal regularization strength in a one-dimensional nonparametric classification setting, our approach adaptively regularizes the data points in higher-uncertainty, lower-density regions more heavily. We test our method on several benchmark tasks, including a real-world heteroskedastic and imbalanced dataset, WebVision. Our experiments corroborate our theory and demonstrate a significant improvement over other methods in noise-robust deep learning.

1 INTRODUCTION

In real-world machine learning applications, even well-curated training datasets have various types of heterogeneity. Two main types of heterogeneity are: (1) data imbalance: the input or label distribution often has a long-tailed density, and (2) heteroskedasticity: the labels given inputs have varying levels of uncertainties across subsets of data stemming from various sources such as the intrinsic ambiguity of the data or annotation errors. Many deep learning algorithms have been proposed for imbalanced datasets (e.g., see (Wang et al., 2017; Cao et al., 2019; Cui et al., 2019; Liu et al., 2019) and the reference therein). However, heteroskedasticity, a classical notion studied extensively in the statistical community (Pintore et al., 2006; Wang et al., 2013; Tibshirani et al., 2014), has so far been under-explored in deep learning. This paper focuses on addressing heteroskedasticity and its interaction with data imbalance in deep learning.

Heteroskedasticity is often studied in regression analysis and refers to the property that the distribution of the error varies across inputs. In this work, we mostly focus on classification, though the developed technique also applies to regression. Here, heteroskedasticity reflects how the uncertainty in the conditional distribution $y \mid x$, or the entropy of $y \mid x$, varies as a function of x. Real-world datasets are often heteroskedastic. For example, Li et al. (2017) shows that the WebVision dataset has a varying number of ambiguous or true noisy examples across classes.[1]

Conversely, we consider a dataset to be *homoscedastic* if every example is mislabeled with a fixed probably ϵ, as assumed by many prior theoretical and empirical works on label corruption (Ghosh et al., 2017; Han et al., 2018; Jiang et al., 2018). We note that varying uncertainty in $y \mid x$ can come from at least two sources: the intrinsic semantic ambiguity of the input, and the (data-dependent) mislabeling introduced by the annotation process. Our approach can handle both types of noisy examples in a unified way, but for the sake of comparisons with past methods, we call them "ambiguous examples" and "mislabeled examples" respectively, and refer to both of them as "noisy examples".

Overparameterized deep learning models tend to overfit more to the noisy examples (Arpit et al., 2017; Frénay & Verleysen, 2013; Zhang et al., 2016). To address this issue, a common approach is to detect noisy examples by selecting those with large training losses, and then remove them

[1] See Figure 4 of (Li et al., 2017), the number of votes for each example indicates the level of uncertainty of that example.

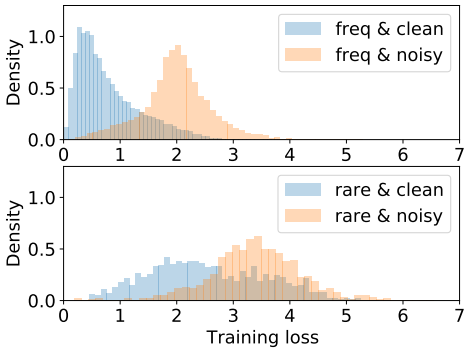

Figure 1: Histogram of the distributions of losses on an imbalanced and noisy CIFAR-10 dataset. Clean but rare examples tend to have larger losses, similar to the noisy examples in frequent classes.

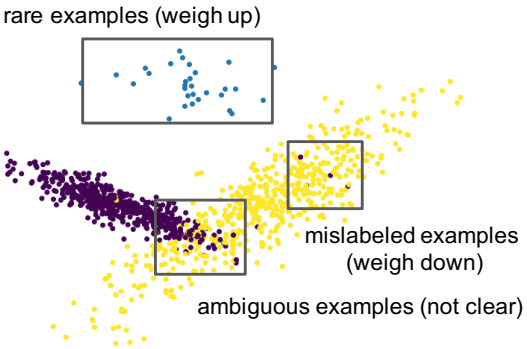

Figure 2: Real-world datasets have various sources of heterogeneity and it could be hard to distinguish one from another. They require mutually-exclusive reweighting strategy, but they all benefit from stronger regularization.

from the (re-)training process. However, an input's training loss can also be big because it is rare or ambiguous (Hacohen & Weinshall, 2019; Wang et al., 2019), as shown in Figure 1. Noise-cleaning methods could fail to distinguish mislabeled from rare/ambiguous examples (see Section 3.1 for empirical proofs). Though dropping the former is desirable, dropping the latter loses important information. Another popular approach is reweighting methods that reduce the contribution of noisy examples in optimization. However, failing to distinguish between mislabeled and rare/ambiguous examples makes the decision of the weights tricky – mislabeled examples require small weights, whereas rare / ambiguous examples benefit from larger weights (Cao et al., 2019; Shu et al., 2019).

We propose a regularization method that deals with noisy and rare examples in a unified way. We observe that mislabeled, ambiguous, and rare examples all benefit from stronger regularization (Hu et al., 2020; Cao et al., 2019). We apply a Lipschitz regularizer (Wei & Ma, 2019a;b) with varying regularization strength depending on the particular data point. Through theoretical analysis in the one-dimensional setting, we derive the *optimal* regularization strength for each training example. The optimal strength is larger for rarer and noisier examples. Our proposed algorithm, heteroskedastic adaptive regularization (HAR), first estimates the noise level and density of each example, and then optimizes a Lipschitz-regularized objective with input-dependent regularization with strength provided by the theoretical formula.

In summary, our main contributions are: (i) we propose to learn heteroskedastic imbalanced datasets under a unified framework, and theoretically study the optimal regularization strength on one-dimensional data. (ii) we propose an algorithm, heteroskedastic adaptive regularization (HAR), which applies stronger regularization to data points with high uncertainty and low density. (iii) we experimentally show that HAR achieves significant improvements over other noise-robust deep learning methods on simulated vision and language datasets with controllable degrees of data noise and data imbalance, as well as a real-world heteroskedastic and imbalanced dataset, WebVision.

2 Adaptive Regularization for Heteroskedastic Datasets

2.1 Backgrounds

We first introduce general nonparametric tools that we use in our analysis, and review the dependency of optimal regularization strength on the sample size and noise level.

Over-parameterized neural networks as nonparametric methods. We use nonparametric method as a surrogate for neural networks because they have been shown to be closely related. Recent work (Savarese et al., 2019) shows that the minimum norm two-layer ReLU network that fits the training data is in fact a linear spline interpolation. Parhi & Nowak (2019) extend this result to a broader family of neural networks with a broader family of activations.

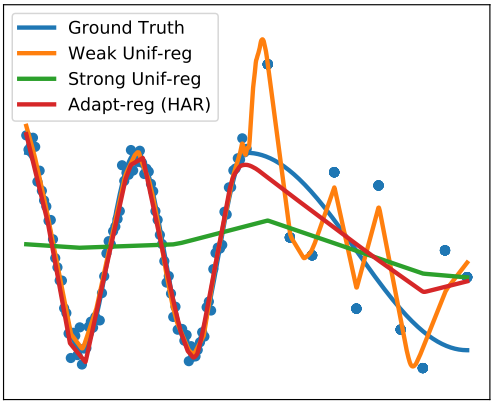

Figure 3: A one-dimensional example with a three-layer neural network in heteroskedastic and imbalanced regression setting. The curve in blue is the underlying ground truth and the dots are observations with heteroskedastic noise. This example shows that uniformly weak regularization overfits on noisy and rare data (on the right half), whereas uniformly strong regularization causes underfitting on the frequent and oscillating data (on the left half). The adaptive regularization does not underfit the oscillating data but still denoise the noisy data. We note that standard nonparametric methods such as cubic spline do not work here because they also use uniform regularization.

Given a training dataset $\{(x_i, y_i)\}_{i=1}^n$, nonparametric method with penalty works as follows. Let $\mathcal{F} : \mathbb{R} \to \mathbb{R}$ be a twice-differentiable model family. We aim to fit the data with smoothness penalty

$$\min_f \ \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) + \lambda \int (f'(x))^2 dx \tag{1}$$

Lipschitz regularization for neural networks. Lipschitz regularization has been shown to be effective for deep neural networks as well. Wei & Ma (2019a) proves a generalization bound of neural networks dependent on the Lipschitzness of each layer with respect to all intermediate layers on the training data, and show that, empirically, regularizing the Lipschitzness improve the generalization. Sokolić et al. (2017) shows similar results in data-limited settings. In Section 2.3, we extend the Lipschitz regularization technique to heteroskedastic setting.

Regularization strength as a function of noise level and sample size. Finally, we briefly review existing theoretical insights on the optimal choice of regularization strength. Generally, the optimal regularization strength for a given model family increases with the label noise level and decreases in the sample size. As a simple example, consider linear ridge regression $\min_\theta \frac{1}{n}\sum_{i=1}^n (x_i^\top \theta - y_i)^2 + \lambda\|\theta\|^2$, where $x_i, \theta \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We assume $y_i = x_i^\top \theta^* + \xi$ for some ground truth parameter θ^*, and $\xi \sim \mathcal{N}(0, \sigma^2)$. Then the optimal regularization strength $\lambda_{opt} = d\sigma^2/n\|\theta^*\|_2^2$. Results of similar nature can also be found in nonparametric statistics (Wang et al., 2013; Tibshirani et al., 2014).

2.2 Heteroskedastic Nonparametric Classification on One-Dimensional Data

We consider a one-dimensional binary classification problem where $\mathcal{X} = [0, 1] \subset \mathbb{R}$ and $\mathcal{Y} = \{-1, 1\}$. We assume Y given X follows a logistic model with ground-truth function $f^\star$, i.e.

$$\Pr[Y = y|X = x] = \frac{1}{1 + \exp(-yf^\star(x))}. \tag{2}$$

The training objective is cross-entropy loss plus Lipschitz regularization, i.e.

$$\hat{f} = \text{argmin}_f \ \ \widehat{L}(f) \triangleq \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) + \lambda \int_0^1 \rho(x)(f'(x))^2 dx, \tag{3}$$

where $\ell(a, y) = -\log(1 + \exp(-ya))$, and $\rho(x)$ is a smoothing parameter as a function of the noise level and density of x. Let $I(x)$ be the fisher information matrix conditioned on the input, i.e. $I(x) \triangleq \mathbb{E}[\nabla_a^2 \ell(a, Y)|_{a=f^\star(X)}|X = x]$. When (X, Y) follows the logistic model in equation 2,

$$I(x) = \frac{1}{(1 + \exp(f^\star(x))(1 + \exp(-f^\star(x))} = \text{Var}(Y|X = x).$$

Therefore, $I(x)$ captures the aleatoric uncertainty of x. For example, when Y is deterministic conditioned on $X = x$, we have $I(x) = 0$, indicating perfect certainty.

Define the test metric as the mean-squared-error on the test set $\{(x_i, y_i)\}_{i=1}^n$, i.e.,[1]

$$\text{MSE}(\hat{f}) \triangleq \mathop{\mathbb{E}}_{\{(x_i,y_i)\}_{i=1}^n} \int_0^1 (\hat{f}(t) - f^\star(t))^2 dt \tag{4}$$

Our main goal is to derive the optimal choice of $\rho(x)$ that minimizes the MSE. We start with an analytical characterization of the test error. Let $W_2^2 = \{f' \text{ is absolute continuous and } f'' \in L^2[0,1]\}$. We denote the density of X as $q(x)$. The following theorem analytically computes the MSE under the regularization strength $\rho(\cdot)$, building upon (Wang et al., 2013) for regression problems. The proof of the Theorem is deferred to Appendix A.

Theorem 1. *Assume that $f^\star, q, I \in W_2^2$. Let $r(t) = -1/(q(t)I(t))$ and $L_0 = \int_{-\infty}^{\infty} \frac{1}{4}\exp(-2|t|)dt$. If we choose $\lambda = C_0 n^{-2/5}$ for some constant $C_0 > 0$, the asymptotic mean squared error is*

$$\lim_{n\to\infty} \textit{MSE}(\hat{f}) = C_n \int_0^1 \lambda^2 r^2(t) \left[\frac{d}{dt}(\rho(t)(f^*)'(t))\right]^2 + L_0 r(t)^{1/2}\rho(t)^{-1/2} dt$$

in probability, where C_n is a scalar that only depends on n.

Using the analytical formula of the test error above, we want to derive an approximately optimal choice of $\rho(x)$. A precise computation is infeasible, so we restrict ourselves to consider only $\rho(x)$ that is constant within groups of examples. We introduce an additional structure – we assume the data can be divided into k groups $[a_0, a_1), [a_1, a_2), \cdots, [a_{k-1}, a_k)$. Each group $[a_j, a_{j+1})$ consists of an interval of data with approximately the same aleatoric uncertainty. We approximate $\rho(t)$ is constant on each of the group $[a_i, a_{i+1})$ with value ρ_i. Plugging this piece-wise constant ρ into the asymptotic MSE in Theorem 1, we obtain

$$\lim_{n\to\infty} \text{MSE}(\hat{f}) = \sum_j \left[\rho_j^2 \int_{a_j}^{a_{j+1}} r^2(t) \left[\frac{d^2}{dt^2} f^\star(t)\right]^2 dt + \rho_j^{-1/2} L_0 \int_{a_j}^{a_{j+1}} r^{1/2}(t) dt\right].$$

Minimizing the above formula over $\rho_1, \ldots, \rho_k$ separately, we derive the optimal weights, $\rho_j = \left[\frac{L_0 \int_{a_j}^{a_{j+1}} r(t)^{1/2} dt}{4\int_{a_j}^{a_{j+1}} r^2(t)\left[\frac{d^2}{dt^2} f^\star(t)\right]^2 dt}\right]^{2/5}$. In practice, we do not know $f^\star$ and $q(x)$, so we make the following simplifications. We assume that $q(t)$ and $I(t)$ are constant on each interval $[a_j, a_{j+1}]$. In other words, we assume that $q(t) = q_j$ and $I(t) = I_j$ for all $t \in [a_j, a_{j+1}]$. We further assume that $\frac{d^2}{dt^2} f^\star(t)$ is close to a constant on the entire space, because estimating the curvature in high dimension is difficult. This simplification yields $\rho_j \propto \left[\frac{q_j^{-1/2} I_j^{-1/2}}{q_j^{-2} I_j^{-2}}\right]^{2/5} = q_j^{3/5} I_j^{3/5}$. We find the simplification works well in practice.

Adaptive regularization with importance sampling. It is practically infeasible to implement the integration in equation 3 for high-dimensional data. We use importance sampling to approximate the integral:

$$\text{minimize}_f \quad L(f) \triangleq \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i) + \lambda \sum_{i=1}^n \tau_i f'(x_i)^2 \tag{5}$$

Suppose $x_i \in [a_j, a_{j+1})$, we have that τ_i should satisfy that $\tau_i q_j = \rho_j$ so that the expectation of the regularization term in equation 5 is equal to that in equation 3. Hence,

$$\tau_i = I_j^{3/5} q_j^{-2/5} = I(x_i)^{3/5} q(x_i)^{-2/5}.$$

Adaptive regularization for multi-class classification and regression. In fact, the proof of Theorem 1 is proved for general loss $\ell(a, y)$. Therefore, we can directly generalize it to multi-class classification and regression problems. For a regression problem, $\ell(a, y)$ is the square loss: $\ell(y, a) = 0.5(y - a)^2$, the Fisher information $I(x) = 1$. Therefore, for a regression problem, we can choose regularization weight $\tau_i = q(x_i)^{-2/5}$.

[1] Note that we integrate the error without weighting because we are interested in the balanced test performance.

2.3 PRACTICAL IMPLEMENTATION ON NEURAL NETWORKS WITH HIGH-DIMENSIONAL DATA

We heuristically extend the Lipschitz regularization technique discussed in Section 2.2 from non-parametric models to over-parameterized deep neural networks. Let (x, y) be an example and f_θ be an r-layer neural network. We denote by $h^{(j)}$ the j-th hidden layer of the network, by $J^{(j)}(x) \triangleq \frac{\partial}{\partial h^{(j)}} \mathcal{L}(f(x), y)$, i.e., the Jacobian of the loss w.r.t $h^{(j)}$. We replace the regularization term $f'(x)^2$ in equation 5 by $R(x) = \left(\sum_{j=1}^{r} \|J^{(j)}(x)\|_F^2\right)^{1/2}$, which was proposed by (Wei & Ma, 2019a). As a proof of concept, we visualize the behavior of our algorithm in Figure 3, where we observe that the rare and noisy examples have significantly improved error due to stronger regularization. In contrast, a uniform regularization either overfits or underfits different subsets.

Note that the differences from the 1-D case include the following three aspects. 1. The derivative is taken w.r.t to all the hidden layers for deep models, which has been shown to have superior generalization guarantees for neural networks by (Wei & Ma, 2019a;b). 2. An additional square root is taken in computing $R(x)$. This modified version may have milder curvature and be easier to tune. 3. We take the derivative of the loss instead of the derivative of the model, which outputs k numbers for multi-class classification. This is because the derivative of the model requires k times more time to compute. The regularized training objective is consequently

$$\text{minimize}_f \quad L(f) \triangleq \frac{1}{n} \sum_{i=1}^{n} \left(\ell(f(x_i), y_i) + \lambda \tau_i R(x_i)\right), \tag{6}$$

where τ_i is chosen to be $\tau_i = I(x_i)^{3/5}/q(x_i)^{2/5}$ following the formula equation 5 in Section 2.2 and λ is a hyperparameter to control the overall scale of the regularization strength. We note that we do not expect this choice of τ_i to be optimal for the high-dimensional case with all the modifications above – the optimal choice does depend on the nuances. However, we also observe that the empirical performance is not sensitive to the form of τ as long as it's increasing in $I(x)$ and decreasing in $q(x)$. That is, the more uncertain or rare an example is, the stronger regularization should be applied.

In order to estimate the relative regularization strength τ_i, the key difficulty lies in the estimation of uncertainty $I(x)$. As in the 1-D setting, we divide the examples into k groups $G_1, \ldots, G_k$ (e.g., each group can correspond to a class), and estimate the uncertainty on each group. In the binary setting, $I(x) = \text{Var}(Y|X = x) = \Pr[Y = 1 \mid X] \cdot \Pr[Y = 0 \mid X]$ can be approximated by $\tilde{I}(x) = 1 - \max_{k \in \{0,1\}} \Pr[Y = k \mid X = x]$ up to a factor of at most 2. We use the same formula for multi-class setting as the approximation of the uncertainty. (As a sanity check, when Y is concentrated on a single outcome, the uncertainty is 0.) Note that $\tilde{I}(x)$ is essentially the minimum possible error of any deterministic prediction on the data point x. Assume that we have a sufficiently accurate pre-trained model, we can use its validation error to estimate $\tilde{I}(x)$:

Then for all $x \in G_j$, we estimate $q(x)$ and $I(x)$ by

$$\forall x \in G_j, q(x) \propto |G_j|, \; I(x) \propto \text{average validation error of a pre-trained model } f_{\tilde{\theta}} \text{ on } G_j \tag{7}$$

The whole training pipeline is summarized in Algorithm 1.

Algorithm 1 Heteroskedastic Adaptive Regularization (HAR)

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. A parameterized model f_θ

1: Split training set $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$
2: $f_{\tilde{\theta}} \leftarrow$ Standard SGD Training on $\mathcal{D}_{\text{train}}$
3: Estimate $I(x), q(x)$ with equation 7 using $f_{\tilde{\theta}}$ on $\mathcal{D}_{\text{val}}$, and compute $\tau_i = I(x_i)^{3/5}/q(x_i)^{2/5}$
4:
5: Initialize the model parameters θ randomly
6: $f_\theta \leftarrow$ SGD with the regularized objective as in equation 6 on the full dataset $\mathcal{D}$

3 EXPERIMENTS

We experimentally show that our proposed algorithm HAR(Algorithm 1) improves the test performance of the noisier and rarer groups of examples (by stronger regularization) without negatively

Table 1: Top-1 validation accuracy (averaged over 3 runs) of ResNet-32 on heteroskedastic and imbalanced CIFAR-10. HAR significantly improves noisy and rare classes, while keeping the accuracy on other classes almost unaffected.

Imbalance ratio	10		100	
Method	Noisy&Rare Cls.	Clean Cls.	Noisy&Rare Cls.	Clean Cls.
ERM	52.9 ± 1.2	94.4 ± 0.1	18.9 ± 1.0	94.2 ± 0.1
Co-teaching	30.2 ± 2.3	88.9 ± 0.3	15.4 ± 2.8	86.4 ± 0.7
INCV	48.9 ± 1.7	94.0 ± 0.2	25.8 ± 1.8	93.8 ± 0.2
MentorNet	54.1 ± 1.0	90.3 ± 0.5	28.3 ± 1.5	90.2 ± 0.4
L2RW	44.3 ± 2.0	90.1 ± 0.5	31.2 ± 1.9	89.7 ± 0.7
MW-Net	55.4 ± 1.1	91.7 ± 0.5	35.6 ± 1.6	92.3 ± 0.5
GCE	48.2 ± 0.6	91.6 ± 0.3	14.1 ± 2.0	91.7 ± 0.4
DMI	44.7 ± 2.3	90.7 ± 0.8	14.0 ± 2.1	91.8 ± 0.6
Unif-reg (optimal)	53.9 ± 0.9	92.1 ± 0.2	36.7 ± 1.0	92.4 ± 0.3
Ours (HAR)	$\mathbf{63.5 \pm 0.8}$	$\mathbf{94.3 \pm 0.2}$	$\mathbf{42.4 \pm 0.7}$	$\mathbf{94.0 \pm 0.2}$

affecting the training and test performance of the other groups. We evaluate our algorithms on three vision datasets and one NLP dataset: CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), IMDB-review (Maas et al., 2011) (see Appendix C.3), and WebVision (Li et al., 2017), a real-world heteroskedastic and imbalanced dataset. Please refer to Appendix B for low-level implementation details. Code will be made publicly available at the time of the publication.

Baselines. We compare our proposed HAR with the following baselines. The simplest one is (1) Empirical risk minimization (ERM): the vanilla cross-entropy loss with all examples having the same weights of losses. We select two representatives from the **noise-cleaning** line of approach. (2) Co-teaching (Han et al., 2018): two deep networks are trained simultaneously. Each network aims to identify clean data points that have small losses and use them to guide the training of the other network. (3) INCV (Chen et al., 2019): it extends Co-teacing to an interative version to estimate the noise ratio and select data. We consider three representatives from the **reweighting-based methods**, including two that learn the weighting using **meta-learning**. (4) MentorNet (Jiang et al., 2018): it pretrains a teacher network that outputs weights for examples that are used to train the student network with reweighting. (5) L2RW (Ren et al., 2018): it directly optimizes weights of each example in the training set by minimizing its corresponding loss on a small meta validation set. (6) MW-Net (Shu et al., 2019): it extends L2RW by explicitly defining a weighting function which depends only on the loss of the example. We also compare against two representatives from the **robust loss function**. (7) GCE (Zhang & Sabuncu, 2018): it generalizes mean average error and cross-entropy loss to obtain a new loss function. (8) DMI (Xu et al., 2019): it designs a new loss function based on generalized mutual information. In addition, as an essential ablation study, we consider vanilla uniform regularization. (9) Unif-reg: we apply the Jacobian regularizer on all examples with equal strength, and tune the strength to get the best possible validation accuracy.

3.1 Simulating heteroskedastic and imbalanced datasets on CIFAR

Setting. Unlike previous works that test on uniform random or asymmetric noise, which is often not the case in reality, in this paper we test our method on more realistic noisy settings, as suggested by Patrini et al. (2017); Zhang & Sabuncu (2018). In order to simulate heteroskedasticity, we only corrupt semantically-similar classes. For CIFAR-10, we exchange 40% of the labels between classes 'cat' and 'dog', and between 'truck' and 'automobile'. CIFAR-100 has 100 classes grouped into 20 super classes. For each class of the 5 classes under the super class 'vehicles 1' and 'vehicles 2', we corrupt the labels with 40% probability uniformly randomly to the rest of four classes under the same super class. As a result, the 10 classes under super class 'vehicle 1' and 'vehicle 2' have high label noise level and the corruption are only within the same super class. Heteroskedasticity of the labels and imbalance of the inputs commonly coexist in the real world settings. HAR can take both of them into account. To understand the challenge imposed by the entanglements of heteroskedasticity and imbalance, and compare HAR with the aforementioned baselines, we inject data imbalance concurrently with the heteroskedastic noise. We remove samples from the corrupted classes to simulate the most difficult scenario — the rare and noisy groups overfit significantly. (A more benign

Table 2: Validation accuracy of ResNet-50 when tuning the regularization strength on mini WebVision. HAR stands out of the trade-off constraint of fitting and generalization.

	Train Acc		Val Acc	
Reg Strength	Top 1	Top 5	Top 1	Top 5
0	69.01	88.64	59.40	80.84
Unif-reg ($\lambda = 0.01$)	68.96	88.54	64.32	86.11
Unif-reg ($\lambda = 0.02$)	67.02	87.51	64.40	85.92
Unif-reg ($\lambda = 0.05$)	65.11	86.33	65.80	86.84
Unif-reg ($\lambda = 0.1$)	63.35	84.98	65.04	86.56
Adaptive (HAR)	69.12	88.41	**69.20**	**88.96**

interaction between the noises and imbalance is that the rare classes have lower noise level, we defer it to Appendix C.5.) We use the imbalance ratio to denote the frequency ratio between the frequent (and clean) classes to the rare (and corrupted) classes. We consider imbalance ratio to be 10 and 100.

Result. Table 1 summarizes the results. Since examples from rare classes tend to have larger training and validation loss regardless of whether the labels are correct or not, noise-cleaning based methods might drop excessive examples with correct labels. We examined the noise ratio of dropped samples for INCV under the setting of imbalance ratio equals 10. Among all dropped examples, there is only 19.2% of true noise examples. In addition, the rare class examples selected still have 29.8% of label noise. This explains that the significant decrease of accuracies of Co-teaching and INCV on corrupted and rare classes. Reweighting-based methods tend to suffer from the loss of accuracy in other more frequent classes, which is aligned with the findings in Cao et al. (2019). While the aforementioned baselines struggle to deal with heteroskedasticity and imbalance together, HAR is able to put them under the same regularization framework and achieve significant improvements. Notably, HAR also shows improvement over uniform regularization with optimally tuned strength. This clearly demonstrates the importance of introducing adaptive regularization among all examples for a better trade-off. A more detailed ablation study on the trade-off between training accuracy and validation accuracy can be found in Section 3.2.

Ablation study. We disentangle the problem setting to show the effectiveness of our unified framework. In Appendix C.1 , we demonstrate state-of-the-art results on a balanced and heteroskedastic dataset where only the uncertainty part is effective. In Appendix C.2, we show state-of-the-art results on an imbalanced and hemosekdeastic dataset where only the density part is effective.

3.2 Evaluation on WebVision with real-world heterogeneity

WebVision (Li et al., 2017) contains 2.4 million images crawled from Google and Flickr using 1,000 labels shared with the ImageNet dataset. Its training set is both heteroskedastic and imbalanced (detailed statistics can be found in (Li et al., 2017)), and it is considered as a popular benchmark for noise robust learning. As the full dataset is very large, we follow (Jiang et al., 2018) to use a mini version, which contains the first 50 classes of the Google subset of the data. Following the standard protocol (Jiang et al., 2018), we test the trained model on the WebVision validation set and the ImageNet validation set. We use ResNet-50 for ablation study and InceptionResNet-v2 for a fair comparison with the baselines. We report results comparing against other state-of-the-art approaches in Table 3. Strikingly, HAR achieves significant improvement.

Ablation study. We demonstrate the trade-off between training accuracy and validation accuracy on mini WebVision with various uniform regularization strength and HAR in Table 2. It's evident that when we gradually increase the overall uniform regularization strength, the training accuracy continues to decrease, and the validation accuracy reaches its peak at 5e-2. While a strong regularization could improve generalization, it reduces deep networks' capacity to fit the training data. However, with our proposed HAR, we only enforce strong regularization on a subset so that we improve the generalization on noisier groups while maintaining the overall training accuracy not affected.

4 Related Work

Our work is closely related to the following methods and directions.

Table 3: Validation accuracy of InceptionResNet-v2 on WebVision and ImageNet validation sets. HAR demonstrates significant improvements over the previous state-of-the-arts.

Train	mini WebVision				full WebVision			
Test	WebVision		ImageNet		WebVision		ImageNet	
Method	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
ERM	62.5	80.8	58.5	81.8	69.7	87.0	62.9	83.6
Co-teaching	63.6	85.2	61.5	84.7	-	-	-	-
INCV	65.2	85.3	61.6	85.0	-	-	-	-
MentorNet	63.0	81.4	57.8	79.9	70.8	88.0	62.5	83.0
Ours (HAR)	**75.5**	**90.7**	**70.3**	**90.0**	**75.0**	**90.6**	**67.1**	**86.7**

Noise-cleaning. The key idea of noise-cleaning is to identify and remove (or re-label) examples with wrong annotations. The general procedure for identifying mislabeled instances has a long history (Brodley & Friedl, 1999; Wilson & Martinez, 1997; Zhao & Nishida, 1995). Some recent works tailored this idea for deep neural networks. Veit et al. (2017) trains a label cleaning network on a small set of data with clean labels, and uses this model to identify noises in large datasets. To circumvent the requirement of a clean subset, Malach & Shalev-Shwartz (2017) train two networks simultaneously and perform update steps only in case of disagreement. Similarly, in co-teaching (Han et al., 2018), each network selects a certain number of small-loss samples and feeds them to its peer network. Chen et al. (2019) further extends the co-training strategy and comes up with an iterative version that performs even better empirically. Recently Song et al. (2020) discovers that it is not necessary to maintain two networks. Removing examples whose training loss exceeds a certain threshold before learning rate decay can also get robust performance.

Reweighting. Reweighting training data has shown its effectiveness on noisy data (Liu & Tao, 2015). Its challenge lies in the difficulty of weights estimation. Ren et al. (2018) proposes a meta-learning algorithm to assign weights to training examples based on its gradient direction with the one on a clean validation set. Recently, Shu et al. (2019) proposes to learn an explicit loss-weight function to mitigate the optimizing issue of (Ren et al., 2018). Another line of work resorts to curriculum learning by either designing an easy-to-hard strategy of training (Guo et al., 2018) or introducing an extra network (Jiang et al., 2018) to assign weights.

Noise-cleaning and reweighting methods usually rely on the empirical loss to determine if a sample is noisy. However, when the dataset is heteroskedastic, each example's training/validation loss no longer correlates well with its noise level. In such cases, we argue that changing the strength of regularization is a more conservative adaption and suffers less from uncertain estimation, compared to changing the weights of losses (Please refer to Section C.6 for empirical justifications).

Robust loss function. Another line of works has attempted to design robust loss functions (Ghosh et al., 2017; Xu et al., 2019; Zhang & Sabuncu, 2018; Patrini et al., 2017; Cheng et al., 2017; Menon et al., 2016). They usually rely on prior assumption about latent transition matrix that might not hold in practice. On the contrary, we focus on more realistic settings.

Regularization. Regularization based techniques have also been explored to combat label noise. Li et al. (2019) proves that SGD with early stopping is robust to label noise. Hu et al. (2020) provides theoretical analysis of two additional regularization methods. While these methods consider a uniform regularization on all training examples, our work emphasizes on adjusting the weights of regularizers in search of a better generalization than uniform assignment.

5 Conclusion

We propose a unified framework (HAR) for training on heteroskedastic and imbalanced datasets. Our method achieves significant improvements over the previous state-of-the-arts on a variety of benchmark vision and language tasks. We provide theoretical results as well as empirical justifications by showing that ambiguous, mislabeled, and rare examples all benefit from stronger regularization. We further provide the formula for optimal weighting of regularization. Heteroskedasticity of datasets is a fascinating direction worth exploring, and it is an important step towards a better understanding of real-world scenarios in the wild.

REFERENCES

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pp. 1565–1576, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070, 2019.

Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance-and label-dependent label noise. *arXiv preprint arXiv:1709.03768*, 2017.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

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.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 135–150, 2018.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pp. 2535–2544, 2019.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

W Hu, Z Li, and D Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations*, 2020.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv preprint arXiv:1903.11680*, 2019.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 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.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150. Association for Computational Linguistics, 2011.

Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*, pp. 960–970, 2017.

Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751*, 2016.

Rahul Parhi and Robert D Nowak. Minimum" norm" neural networks are splines. *arXiv preprint arXiv:1910.02333*, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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.

Alexandre Pintore, Paul Speckman, and Chris C Holmes. Spatially adaptive smoothing splines. *Biometrika*, 93(1):113–125, 2006.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343, 2018.

Pedro H. P. Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *COLT*, 2019.

Zuofeng Shang, Guang Cheng, et al. Local and global asymptotic inference in smoothing spline models. *The Annals of Statistics*, 41(5):2608–2638, 2013.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pp. 1917–1928, 2019.

J. Sokolić, R. Giryes, G. Sapiro, and M. R. D. Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Jiaming Song, Lunjia Hu, Yann Dauphin, Michael Auli, and Tengyu Ma. Robust and on-the-fly dataset denoising for image classification. *arXiv preprint arXiv:2003.10647*, 2020.

Ryan J Tibshirani et al. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 2014.

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.

Xiao Wang, Pang Du, and Jinglai Shen. Smoothing splines with varying smoothing parameter. *Biometrika*, 100(4):955–970, 2013.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5017–5026, 2019.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7029–7039, 2017.

Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. In *Advances in Neural Information Processing Systems*, pp. 9722–9733, 2019a.

Colin Wei and Tengyu Ma. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*, 2019b.

D Randall Wilson and Tony R Martinez. Instance pruning techniques. In *ICML*, volume 97, pp. 400–411, 1997.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, pp. 6222–6233, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

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.

Qi Zhao and Toyoaki Nishida. Using qualitative hypotheses to identify inaccurate data. *Journal of Artificial Intelligence Research*, 3:119–145, 1995.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9719–9728, 2020.

A PROOFS OF THEOREM 1

We prove a general theorem here. In particular, we have the general theorem below.

Theorem 2. *Assume that $f^\star, q, I \in W_2^2$. Suppose (1) $\ell(a, y)$ is convex and three times continously differentiable with respect to a, (2) there exist constants $0 < c < C < \infty$ such that $c \leq I(X) \leq C$ almost surely, and $\epsilon \triangleq \nabla_a \ell(f^\star(X), Y)$ satisfies $\mathbb{E}[\epsilon|X] = 0$, $\mathbb{E}[\epsilon^2|X] = I(X)$ and $\mathbb{E}[\epsilon^4|X] < \infty$ almost surely. Let $r(t) = -1/(q(t)I(t))$ and $L_0 = \int_{-\infty}^{\infty} \frac{1}{4}\exp(-2|t|)dt$. If we choose $\lambda = C_0 n^{-2/5}$ for some constant $C_0 > 0$, the asymptotic mean squared error of $\hat{f}$ by equation 4 is*

$$\lim_{n\to\infty} MSE(\hat{f}) = C_n \int_0^1 \lambda^2 r^2(t) \left[\frac{d}{dt}(\rho(t)(f^*)'(t))\right]^2 + L_0 r(t)^{1/2}\rho(t)^{-1/2}dt$$

in probability, where C_n is a scalar that only depends on n.

It is easy to check that the logistic loss satisfies the condition of the theorem.

The proof strategy of Theorem 2 is adopted from the proof of Theorem 2 of (Wang et al., 2013) by generalizing it from the least square loss to logistic loss.

The high level idea is to reformulate $\hat{f}$ as solutions to ordinary differential equations. Let (γ_v, h_v) be the (normalized) solution of the following equation

$$-\rho(t)h_v^{''}(t) = \gamma_v I(t)q(t)h_v(t), \tag{8}$$

$$h_v^{'}(0) = h_v^{'}(1) = 0, \text{ and } h_v^{''}(0) = h_v^{''}(1) = 0. \tag{9}$$

We define the the leading term of $\hat{f} - f^\star$ as $S_{n,\lambda}(f^\star)$ as

$$S_{n,\lambda}(f^\star) = \frac{1}{n}\epsilon_i K_{X_i} - W_\lambda f^\star, \text{where}$$

$$K_t(\cdot) = \sum_v \frac{h_v(t)}{1+\lambda\gamma_v} h_v(\cdot) \text{ and } W_\lambda h_v(\cdot) = \frac{\lambda\gamma_v}{1+\lambda\gamma_v} h_v(\cdot).$$

By Proposition 2.1 and Theorem 3.4 of Shang et al. (2013), we have

$$\sup_x |\hat{f}(x) - f^\star(x) - S_{n,\lambda}(f^\star)(x)| = o_P(n^{-1/3}). \tag{10}$$

Following the same proof of Theorem 2 of (Wang et al., 2013), we can simplify the definition of K_t and W^λ as

$$K_t(x) = \frac{I(t)}{q(t)} J(t,x) \text{ and } W_\lambda f^\star(t) = \lambda r(t)\left[\frac{d}{dt}(\rho(t)(f^*)'(t))\right], \tag{11}$$

where $J(t,s) = \beta\rho(s)Q'_\beta(s)L_0(\beta|Q_\beta(t) - Q_\beta(s)|)$ and $Q_\beta(t,s) = \int_0^t (r(s)\rho(s))^{-1/2}(1 + O(\beta^{-1}))ds$ and $\beta = 1/\sqrt{\lambda}$. Plugging equation 11 into equation 10, we then have

$$\lim_{n\to\infty} \text{MSE}(\hat{f}) = C_n \int_0^1 \lambda^2 r^2(t) \left[\frac{d}{dt}(\rho(t)(f^*)'(t))\right]^2 + L_0 r(t)^{1/2}\rho(t)^{-1/2}dt$$

B IMPLEMENTATION DETAILS

We develop our core algorithm in PyTorch (Paszke et al., 2017).

Implementation details for CIFAR. We follow the simple data augmentation used in (He et al., 2016) with only random crop and horizontal flip. We use ResNet-32 as our base network and repeat all experiments for 3 runs. We use standard SGD with momentum of 0.9, weight decay of 1×10^{-4} for training. The model is trained with a batch size of 128 for 120 epochs. We anneal the learning rate by a factor of 10 at 80 and 100 epochs. We group the data by class labels, and by default we split $\mathcal{D}$ equally and randomly into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. As for the Jacobian regularizer, we sum over the frobenius norm of the gradients of all normalization layers' (BN by default) activations with respect to the classification loss. For experiments of HAR, we tune λ so that the largest enforced regularization strength ($\lambda\tau_i$) is 0.1. We train each model with 1 NVIDIA GeForce RTX 2080 Ti.

Implementation details for IMDB-review. We train a two-layer bidirectional LSTM (Huang et al., 2015) with 256 units followed with 0.5 dropout before the linear classifier. The network is trained for 20 epochs with Adam optimizer (Kingma & Ba, 2014). For HAR, we tune λ so that the largest enforced regularization strength ($\lambda\tau_i$) is 0.1. We train each model with 1 NVIDIA GeForce RTX 2080 Ti.

Implementation details for WebVision. We use the standard data augmentation same as (He et al., 2016) including random crop and horizontal flip. For mini WebVision, We train the network for 90 epochs using standard SGD with a batch size of 128. The initial learning rate is 0.1 and is annealed by a factor of 10 at epoch 60 and 90. For full WebVision, We train the network for 50 epochs using standard SGD with a batch size of 256. The initial learning rate is 0.1 and is annealed by a factor of 10 at epoch 30 and 40. For experiments of HAR, we tune λ so that the largest enforced regularization strength ($\lambda\tau_i$) is 0.1. We train each model with 8 NVIDIA Tesla V100 GPUs.

Runtime analysis. Because the pre-trained model only trains on half of the training data and is only done once, the run-time of HAR is at most twice of the time for ERM. Many baselines in our paper use sophisticated pipelines and are slower than HAR. For example, INCV trains 2 models simultaneously for 4 times from random initialization to get a clean training set. MW-Net has a very slow convergence rate, which is a common issue for meta-learning.

C ADDITIONAL RESULTS

C.1 SIMULATING HETEROSKEDASTIC NOISE ON CIFAR

Table 4: Top-1 validation accuracy (averaged over 3 runs) of ResNet-32 on heteroskedastic CIFAR-10 and CIFAR-100 for the noisy classes and the clean classes.

Dataset	CIFAR-10		CIFAR-100	
Method	Avg. Noisy Cls.	Avg. Clean Cls.	Avg. Noisy Cls.	Avg. Clean Cls.
ERM	68.6 ± 0.2	93.6 ± 0.2	65.3 ± 0.3	67.8 ± 0.2
Co-teaching	64.7 ± 0.4	89.1 ± 0.3	59.8 ± 0.4	65.3 ± 0.3
INCV	76.7 ± 0.6	93.0 ± 0.2	66.2 ± 0.3	68.6 ± 0.2
MentorNet	71.1 ± 0.4	93.7 ± 0.2	65.9 ± 0.3	67.5 ± 0.3
L2RW	70.1 ± 0.3	92.5 ± 0.3	65.1 ± 0.5	67.0 ± 0.3
MW-Net	75.0 ± 0.3	94.4 ± 0.2	65.7 ± 0.3	69.1 ± 0.2
GCE	62.6 ± 1.1	90.2 ± 0.2	61.2 ± 0.6	66.9 ± 0.2
DMI	73.2 ± 0.7	90.8 ± 0.2	64.8 ± 0.5	67.1 ± 0.2
Unif-reg (λ = 0.1)	77.5 ± 0.6	92.3 ± 0.2	69.3 ± 0.5	66.6 ± 0.3
Unif-reg (optimal)	75.3 ± 0.3	94.1 ± 0.2	68.5 ± 0.3	68.6 ± 0.2
Ours (HAR)	$\mathbf{80.7 \pm 0.3}$	$\mathbf{94.5 \pm 0.2}$	$\mathbf{74.2 \pm 0.3}$	$\mathbf{69.3 \pm 0.2}$

We study the uncertainty part of HAR by testing under the setting with only heteroskedastic noise. The type of noise injection is the same as Section 3.1.

We report the top-1 validation accuracy of various methods in Table 4. Aligned with our analysis in Section 4, we observe that both noise-cleaning and reweighting based methods don't get a comparable accuracy on noisy classes with applying strong regularization ($\lambda = 0.1$) under this heteroskedastic setting. We observe the behavior that too strong regularization impede the model from fitting informative samples, thus it could lead to a decrease on clean classes' accuracy. On the contrary, too weak regularization leads to overfitting the noisy examples thus the accuracy on noisy classes do not reach the optimal.

Interestingly, we find that even the well-studied CIFAR-100 dataset has intrinsic heteroskedasticity and HAR can improve over uniform regularization to some extent. Please refer to Appendix C.4 for the results on CIFAR-100 and Appendix C.3 for results on IMDB-review.

C.2 SIMULATING DATA IMBALANCE ON CIFAR

We study the density part of HAR by testing under the setting with only data imbalance. We follow the same setting as Cao et al. (2019) to create imbalanced CIFAR. Long-tailed imbalance follows an exponential decay in sample sizes across different classes. For step imbalance setting, all rare classes have the same sample size, as do all frequent classes. HAR achieves better results than LDAM-DRW and is comparable to recent state-of-the-art methods under the imbalanced setting.

Table 5: Top-1 validation errors of ResNet-32 on imbalanced CIFAR-10 and CIFAR-100.

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.61	36.70	17.50	61.68	44.30	61.45	45.37
Focal	29.62	13.34	36.09	16.36	61.59	44.22	61.43	46.54
CB Focal	25.43	12.90	39.73	16.54	63.98	42.01	80.24	49.98
LDAM-DRW	22.97	11.84	23.08	12.19	57.96	41.29	54.64	40.54
BBN (Zhou et al., 2020)	**20.18**	11.68	21.64	11.99	57.44	40.88	57.44	40.36
HAR	20.46	**10.62**	**20.27**	**11.58**	**55.35**	**38.98**	**51.73**	**37.54**

C.3 SIMULATING HETEROSKEDASTIC NOISE ON IMDB-REVIEW.

IMDB-review dataset has a total of 50,000 (25,000 positive and 25,000 negative reviews) movie reviews for binary sentiment classification (Maas et al., 2011). To simulate heteroskedastic noise for this binary classification problem, we project 5% of the labels of negative reviews to positive, and 40% in the reverse direction. Table 6 summarizes the results. The proposed HAR outperforms the ERM baseline with various strength of uniform regularization.

Table 6: Top-1 validation accuracy (averaged over 3 runs) on heteroskedastic IMDB-review dataset.

Reg Strength	Acc. of neg. reviews	Acc. of pos. reviews	Mean Acc
0	91.9 ± 2.0	50.9 ± 1.8	71.4 ± 0.5
Unif-reg ($\lambda = 0.01$)	94.3 ± 1.8	51.9 ± 2.0	73.1 ± 0.3
Unif-reg ($\lambda = 0.1$)	91.5 ± 1.9	64.3 ± 1.6	77.9 ± 0.4
Ours (HAR)	93.1 ± 1.5	72.8 ± 1.7	83.0 ± 0.3

C.4 EVALUATION ON CIFAR-100 WITH REAL-WORLD HETEROSKEDASTICITY

It is acknowledged that CIFAR-100 training set contains noisy examples. For instance, some "tiger" examples are labeled as "leopard" ("tiger" is a defined class as well). There are also noisy examples that contain multiple objects, or are more ambiguous in terms of indentity (Song et al., 2020). We find that HAR can improve over uniform regularization on the well-studied CIFAR-100 due to its heteroskedasticity and the results are reported in Table 7.

Table 7: Top-1 validation accuracy (average over 3 runs) of ResNet-32 on the original CIFAR-100.

Reg Strength	Train Acc	Val Acc
0	96.0 ± 0.1	69.8 ± 0.2
Unif-reg ($\lambda = 0.001$)	96.4 ± 0.1	70.0 ± 0.2
Unif-reg ($\lambda = 0.01$)	95.7 ± 0.1	70.6 ± 0.1
Unif-reg ($\lambda = 0.1$)	88.8 ± 0.1	70.5 ± 0.1
Adaptive (HAR)	96.2 ± 0.1	$\mathbf{71.4 \pm 0.1}$

C.5 SIMULATING HETEROSKEDASTIC AND IMBALANCED DATASETS ON CIFAR

As mentioned in Section 3.1, we consider another variant of heteroskedastic and imbalanced dataset such that the rare classes have low noise level. To simulate this setting, we make the clean classes have fewer labels than the corrupted classes on the heteroskedastic CIFAR-10 we created in Section C.1.

Table 8 summarizes the results. For the setting of imbalance ratio equals 10, INCV automatically drops 34.1% of examples from the clean and rare classes, which results in a decrease of mean accuracy on the rare and clean classes. HAR is able to achieve improvements on both noisy classes and rare classes by enforcing the optimal regularization.

Table 8: Top-1 validation accuracy (averaged over 3 runs) of ResNet-32 on heteroskedastic and imbalanced CIFAR-10.

Imbalance ratio	10		100	
Method	Noisy Cls.	Rare Cls.	Noisy Cls.	Rare Cls.
ERM	59.9 ± 1.1	65.2 ± 0.7	60.4 ± 0.9	7.9 ± 1.3
Co-teaching	63.1 ± 2.3	59.4 ± 1.4	53.8 ± 1.9	4.4 ± 1.8
INCV	74.5 ± 1.2	63.7 ± 0.8	68.3 ± 1.8	2.1 ± 1.3
MentorNet	67.3 ± 1.6	65.5 ± 1.2	63.3 ± 1.5	10.8 ± 1.9
L2RW	65.8 ± 1.4	66.3 ± 1.2	62.4 ± 2.1	11.3 ± 2.9
MW-Net	71.4 ± 0.6	67.7 ± 0.6	65.0 ± 1.6	13.5 ± 2.4
GCE	64.6 ± 1.1	60.2 ± 1.3	71.2 ± 1.9	2.6 ± 1.4
DMI	72.3 ± 1.5	63.3 ± 1.2	70.8 ± 1.7	6.2 ± 1.9
Ours (HAR)	$\mathbf{76.1 \pm 0.8}$	$\mathbf{72.1 \pm 1.0}$	$\mathbf{73.0 \pm 1.6}$	$\mathbf{26.1 \pm 0.8}$

C.6 COMPARING THE EFFECT OF WEIGHTS ON LOSSES AND REGULARIZERS

As discussed in Section 4, we train several classifiers with alternative weights selection scheme which are not optimal. We consider the following two alternatives. (1) **random**: we draw the weights from a uniform distribution with the same mean as the weights of MW-Net and HAR. (2) **inverse**: we take the inverse of the weights learned by MW-Net and HAR and then normalize them to ensure the average reweighting/regularization strength remains the same.

We conducted experiments on the heteroskedastic CIFAR-10 introduced in Section C.1 and the results are summarized in Table 9. We could conclude that changing the weights of the regularizer is a more conservative adaption and less susceptible to uncertain estimation than reweighting.

Table 9: Top-1 validation accuracy (averaged over 3 runs) of ResNet-32 on heteroskedastic CIFAR-10 by changing the weighting scheme.

Method	Avg. Noisy Cls.	Avg. Clean Cls.
ERM	68.6 ± 0.2	93.6 ± 0.2
Reweight (MW-Net)	75.0 ± 0.3	94.4 ± 0.2
Reweight (random)	62.1 ± 0.5	92.9 ± 0.5
Reweight (inverse)	13.1 ± 0.9	90.9 ± 0.3
Unif-reg (optimal)	75.3 ± 0.3	94.1 ± 0.2
Adap-reg (HAR)	80.7 ± 0.3	94.5 ± 0.2
Adap-reg (random)	74.8 ± 0.4	94.2 ± 0.2
Adap-reg (inverse)	73.2 ± 0.5	94.0 ± 0.2

C.7 VISUALIZATION

In order to better understand how the proposed HAR works on real-world heteroskedastic datasets, we plot the per-class key statistics used by HAR and validation errors in Figure 4. We observe that HAR outperforms the tuned uniform regularization baseline on the majority of the classes.

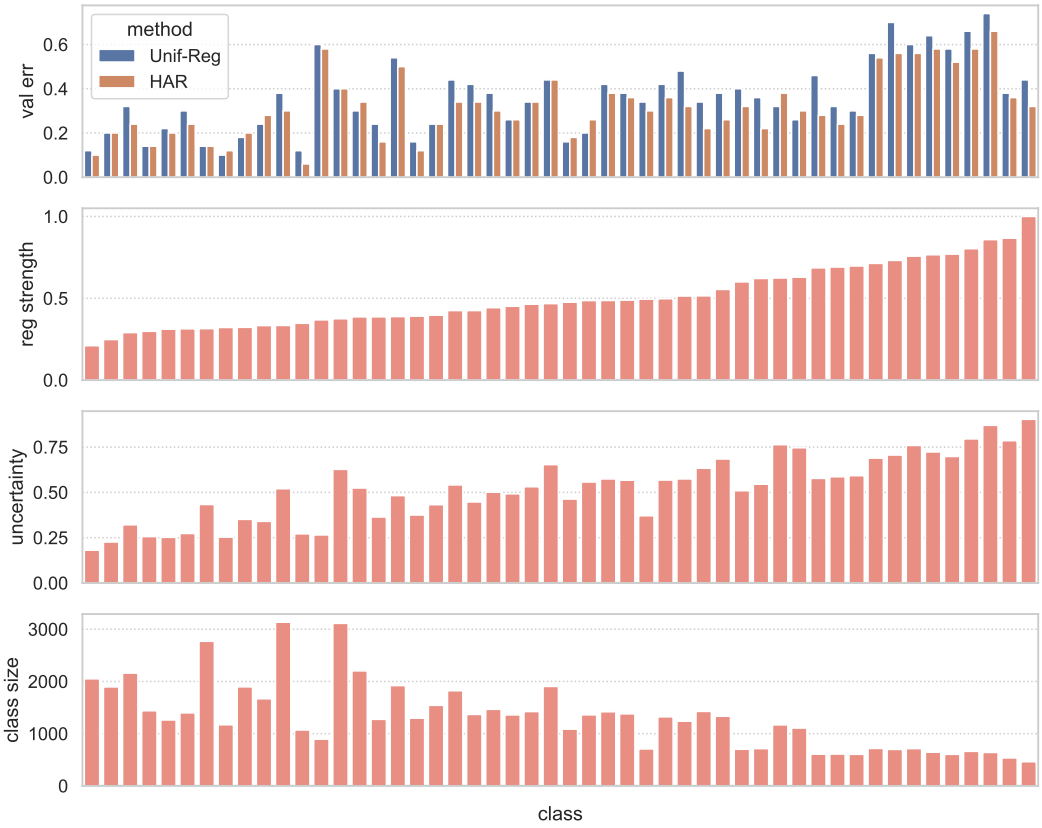

Figure 4: Visualizations of per-class top-1 error and regularization strength of HAR on mini WebVision dataset. The classes are sorted in the ascending order of applied regularization strength.