A NOTE ON QUANTIFYING THE INFLUENCE OF ENERGY REGULARIZATION FOR IMBALANCED CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

For classification problems where classifiers predict $\bar{p}(y|\mathbf{x})$, namely the probability of label y given data $\mathbf{x}$, an energy value can be defined (e.g. LogSumExp of the logits) and used to evaluate the estimated $\bar{p}(\mathbf{x})$ by learned model, which is widely used for generative learning. However, previous works overlook the relationship between the estimated $\bar{p}(\mathbf{x})$ and the testing accuracy of a classifier when shifts occur regarding $p(\mathbf{x})$ from the the training set to the testing set *e.g.* imbalanced dataset learning. In this paper, we propose to evaluate the influence of the energy value regarding $\bar{p}(\mathbf{x})$ on the testing accuracy via influence function which is a standard tool in robust statistics. In particular, we empirically show that the energy value could influence the testing accuracy of the model trained on the imbalanced dataset. Based on our findings, we further propose a technique that regularizes the energy value on the training set to improve imbalanced data learning. We theoretically prove that regularizing energy value could adjust the margin and re-weight the sample. Experimental results show the effectiveness of our method. In particular, when finetuning with our method for only a few epochs, the testing accuracy could be effectively boosted on popular imbalance classification benchmarks.

1 INTRODUCTION

Classification problems require the classifier to predict the conditional probability $\bar{p}(y|\mathbf{x})$ where y is the label and $\mathbf{x}$ is the given data. Based on the $\bar{p}(y|\mathbf{x})$, an energy value could be defined regarding to the predicted $\bar{p}(\mathbf{x})$, where it could be turned into $\bar{p}(\mathbf{x})$ with Gibbs distribution (LeCun et al., 2006; Grathwohl et al., 2020). It has been used for generative model (Grathwohl et al., 2020), out-of-distribution (OOD) detection (Liu et al., 2020), *etc*. Since the $\bar{p}(\mathbf{x})$ predicted by the classifier is not related to the predicted $\bar{p}(y|\mathbf{x})$, *e.g.* the energy of one data point could vary while the prediction for this data point stays the same, the influence of the energy value on the classification itself is either overlooked or simply treated by previous works.

Intuitively, the energy value reflects how the classifier models the data distribution and could influence the generalization performance of the classifier when the distribution of data $p(\mathbf{x})$ shifts between the training set and the testing set. A typical scenario is imbalanced dataset learning (*e.g.* long-tail) (Wang et al., 2017; Zhou et al., 2018; Buda et al., 2018; Liu et al., 2019; Zhong et al., 2019; He et al., 2021) where the distribution $p(\mathbf{x})$ shifts between the training set and the testing set. For imbalanced datasets like long-tail data distribution, the training set suffers from class imbalance *i.e.* some of the classes have way more samples than other classes. The class-imbalance in the training set leads to the poor performance of the classifier on less represented (or tail) classes. To improve performance in such scenarios, various methods have been proposed, such as re-sample the data or re-weights the data point (Cui et al., 2019; Cao et al., 2019; Kang et al., 2020) to devise a training loss that is close to the loss on the test data distribution.

However, to our best knowledge, there is not a thorough study about the influence of the energy value reflecting the predicted $\bar{p}(\mathbf{x})$ on the performance of the classifier as the data distribution is shifted. Therefore, we argue that the influence of the energy on the performance of the classifier needs to be quantified and taken a closer look. To quantify the influence of the energy value on the classifier's performance, we propose a metric based on the influence function. The influence function is a classic technique in robust statistics that evaluates the influence of a training data point on the

model's prediction (Cook & Weisberg, 1982). The seminal work (Koh & Liang, 2017) extends this method to deep learning models and proposes to approximate the influence function with stochastic estimation. The approximated influence function provides valuable information even in non-convex, non-convergent settings. Inspired by this, in this paper, we evaluate the influence of the energy value on the classifier's performance on the testing set via the approximated influence function. We show that the energy value could influence the classifier's performance on the testing set and the approximated influence function could reflect the influence of the energy value with experiment results under imbalanced dataset learning (long-tailed recognition).

Based on the quantification of the influence of the energy value, we propose a principled and generic method that finetunes the classifier with energy regularization determined by the influence of the energy value. We theoretically prove that when regularizing the energy value, it influences the model by re-weighting the loss and adjusting the margin. Experimental results have shown that our method could effectively improve the performance of classifiers on imbalance datasets including two typical settings: long-tail and step imbalance. **The highlights of this paper include:**

- To our best knowledge, this is the first work to quantify the influence of the energy regularization on the generalization behavior of a classifier, specifically under the imbalance setting. We empirically validate the correctness of our quantification of the influence of energy regularization.
- We propose a principled and generic technique as a plug-in to boost the classifier's performance with existing baselines. We theoretically prove that our method intrinsically re-weights the loss and affects the margin without hindering the optimization of the model. The effectiveness of our method is verified on imbalanced datasets with various baselines.
- We provide intriguing insights regarding the influence of energy regularization including: (1) the unregularized energy value may contribute to the overfitting of the model; (2) the influence of energy regularization is not correlated with the loss or the energy; (3) the influence of energy regularization may be the result of interfering the distance between the estimated distribution $\hat{p}(\mathbf{x})$ and the real testing distribution $p(\mathbf{x})$. Source code will be made publicly available to reproduce our results.

2 Preliminaries on Energy Function and Influence Function

In this section, we introduce notations and backgrounds related to this paper.

The Classification Problem. For a K-class classification problem, a parameterized classifier $f_\theta : \mathbb{R}^D \to \mathbb{R}^K$ maps data point $\mathbf{x} \in \mathbb{R}^D$ to K real-valued logits where θ is the trainable parameter. For a data point $\mathbf{x}$ and its corresponding label y, the loss for the parameter θ is defined as $\mathcal{L}(\mathbf{x}, y, \theta)$.

Energy-based Model for Data Distribution (LeCun et al., 2006; Grathwohl et al., 2020). Energy based model $E(\mathbf{x}) : \mathcal{R}^D \to \mathcal{R}$ maps each data point $\mathbf{x}$ to a single, non-probabilistic scalar called the energy, where the energy value could be turned to a probability through the Gibbs distribution as:

$$p(\mathbf{x}) = \frac{e^{-E(\mathbf{x})/T}}{\int_{\mathbf{x}'} e^{-E(\mathbf{x}')/T}} \tag{1}$$

For classifier f_θ, the logits are typically converted to a normalized probability distribution with the Softmax function: $\bar{p}_\theta(y|\mathbf{x}) = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])}$, where $f(\mathbf{x})[y]$ represents the logit corresponding to the y-th class. The joint distribution of data $\mathbf{x}$ and label y could be defined as $\bar{p}_\theta(\mathbf{x}, y) = \frac{\exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}$ where $Z(\theta)$ is unknown normalizing constant. By marginalizing out y, the unnormalized density model for $\mathbf{x}$ is $\bar{p}_\theta(\mathbf{x}) = \sum_y \bar{p}_\theta(y|\mathbf{x}) = \frac{\sum_y \exp(f_\theta(\mathbf{x})[y])}{Z(\theta)}$. Therefore as we set $T = 1$ the energy at data point $\mathbf{x}$ regarding to $\bar{p}_\theta(\mathbf{x})$ is defined as:

$$E_\theta(\mathbf{x}) = -\log \sum_y \exp(f_\theta(\mathbf{x})[y]) \tag{2}$$

Influence Function (Cook & Weisberg, 1982; Koh & Liang, 2017). Given a training set with n data points $D_{train} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_n, y_n)\}$, the optimal parameter for empirical risk is given by $\hat{\theta} \overset{def}{=} \arg\min_\theta \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\mathbf{x}_i, y_i, \theta)$. When the loss on a training

data point $(\mathbf{x}, y)$ is upweighted by a small ϵ, the new optimal parameter becomes $\hat{\theta}_{\epsilon,(\mathbf{x},y)} = \arg\min_\theta(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\mathbf{x}_i, y_i, \theta) + \epsilon\mathcal{L}(\mathbf{x}, y, \theta))$. Assume that the empirical risk is twice-differentiable and strictly convex w.r.t. θ, the influence function provides the influence of upweighting $(\mathbf{x}, y)$ on θ:

$$\mathcal{I}_{\hat{\theta}}(\mathbf{x}, y) \stackrel{def}{=} \frac{\mathrm{d}\hat{\theta}_{\epsilon,(\mathbf{x},y)}}{\mathrm{d}\epsilon}|_{\epsilon=0} = -H_{\hat{\theta}}^{-1}\nabla_\theta\mathcal{L}(\mathbf{x}, y, \hat{\theta}) \tag{3}$$

where $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta^2\mathcal{L}(\mathbf{x}_i, y_i, \hat{\theta})$ is the Hessian matrix. Using the chain rule, the influence of upweighting $(\mathbf{x}, y)$ on the loss at a testing point $(\mathbf{x}_{te}, y_{te})$ (Koh & Liang, 2017) is:

$$\mathcal{I}_{(\mathbf{x}_{te},y_{te})}(\mathbf{x}, y) \stackrel{def}{=} \nabla_\theta\mathcal{L}(\mathbf{x}_{te}, y_{te}, \hat{\theta})^\top\frac{\mathrm{d}\hat{\theta}_{\epsilon,z}}{\mathrm{d}\epsilon}|_{\epsilon=0} = -\nabla_\theta\mathcal{L}(z_{test}, \hat{\theta})^\top H_{\hat{\theta}}^{-1}\nabla_\theta\mathcal{L}(\mathbf{x}, y, \hat{\theta}) \tag{4}$$

3 Main Results and Approach

3.1 Most Methods for Classification Ignores the Energy Value

Since the energy value defined in Eq. 2 reflects the predicted $\bar{p}(\mathbf{x})$, the energy value is overlooked for classification. For a general case where the logits of the classifier are converted to a probability distribution with Softmax function and the classifier is optimized by negative log-likelihood loss, we have the following proposition (see proof in Appendix A).

Proposition 1. [Arbitrary Energy] Consider a data point $x \in \mathcal{R}^D$ and a classifier f_θ. For any $\mathcal{E} \in \mathcal{R}$, there exists a classifier g_η that satisfy

$$\begin{aligned}\bar{p}_\theta(y|\mathbf{x}) &= \bar{p}_\eta(y|\mathbf{x}),\\ E_\eta(\mathbf{x}) &= \mathcal{E}\end{aligned} \tag{5}$$

where $\bar{p}_\theta(y|\mathbf{x})$ and $\bar{p}_\eta(y|\mathbf{x})$ is the conditional probability predicted by f_θ and g_η respectively while $E_\eta(\mathbf{x})$ is the energy value of g_η on data point $\mathbf{x}$. Proposition 1 shows that for a data point $\mathbf{x}$, the energy value $E_\theta(\mathbf{x})$ is not related to the prediction $\bar{p}_\theta(y|\mathbf{x})$. Even two classifiers with identical predictions could have different energy values. Therefore the impact of energy value on the model's testing performance is often ignored. However, we argue that when the probability distribution $p(\mathbf{x})$ shifts *e.g.* the so-called imbalance classification setting (Buda et al., 2018; Cui et al., 2019) whereby the training data is imbalanced while testing data is balanced as mainly studied in this paper, the energy value of the classifier *i.e.* the predicted $\bar{p}(\mathbf{x})$ would also affect the classification performance.

Remark 1. [Uncertain Energy Shift] When optimizing the classifier f_θ with negative log-likelihood loss $\mathcal{L}_{ce}(\mathbf{x}, y, \theta)$, the gradient of the parameter θ is:

$$\frac{\partial\mathcal{L}_{ce}(\mathbf{x}, y, \theta)}{\partial\theta} = -\frac{\partial E_\theta(\mathbf{x})}{\partial\theta} - \frac{\partial f_\theta(\mathbf{x})[y]}{\partial\theta} \tag{6}$$

where $-\frac{\partial E_\theta(\mathbf{x})}{\partial\theta}$ pulls up the energy and $-\frac{\partial f_\theta(\mathbf{x})[y]}{\partial\theta}$ pushs down the energy. Therefore the energy $E_\theta(\mathbf{x})$ may fluctuate regarding the output of the model as the loss decreases.
Though well-classified samples would generally have lower energy, Remark 1 tells that optimizing the classifier would, in fact, leads to an unstable shift of the energy value. Thus, we argue that the energy value needs to be explicitly regularized. We provide more empirical results in Appendix C

3.2 Influence Function of Energy Value

To evaluate the influence of energy regularization on the model performance on the testing set, we propose a metric using the influence function. We first study the change in model parameters brought by energy regularization. Specifically, consider adding a small regularization on the energy of a training data point $(\mathbf{x}_{tr}, y_{tr}) \in D_{train}$, the new optimized parameter becomes $\hat{\theta}_{\epsilon,(\mathbf{x}_{tr},y_{tr})} = \arg\min_\theta(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\mathbf{x}_i, y, \theta) + \epsilon E_\theta(\mathbf{x}_{tr}))$. Similar to Eq. 3, the influence of regularizing energy $E_\theta(\mathbf{x}_{tr})$ on the parameter θ could be defined as:

$$\mathcal{I}_{\hat{\theta}}(\mathbf{x}_{tr}) \stackrel{def}{=} \frac{\mathrm{d}\hat{\theta}_{\epsilon,(\mathbf{x}_{tr},y_{tr})}}{\mathrm{d}\epsilon}|_{\epsilon=0} = -H_{\hat{\theta}}^{-1}\nabla_\theta E_\theta(\mathbf{x}_{tr}) \tag{7}$$

where $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta^2\mathcal{L}(\mathbf{x}_i, y_i, \hat{\theta})$ is the Hessian matrix. Using the chain rule, the influence of regularizing energy $E_\theta(\mathbf{x}_{tr})$ on the loss at the testing point $\mathbf{x}_{test}$ is:

$$\mathcal{I}_{\mathbf{x}_{test}}(\mathbf{x}_{tr}) \stackrel{def}{=} \nabla_\theta\mathcal{L}(\mathbf{x}_{test}, \hat{\theta})^\top\frac{\mathrm{d}\hat{\theta}_{\epsilon,(\mathbf{x}_{tr},y_{tr})}}{\mathrm{d}\epsilon}|_{\epsilon=0} = -\nabla_\theta\mathcal{L}(\mathbf{x}_{test}, \hat{\theta})^\top H_{\hat{\theta}}^{-1}\nabla_\theta E_\theta(\mathbf{x}_{tr}) \tag{8}$$

Algorithm 1: Finetune the classifier with energy regularization.

Input: Training Set $D_{train} = \{(\mathbf{x}_i^{tr}, y_i^{tr})\}_{i=1}^n$. Validation Set $D_{val} = \{(\mathbf{x}_i^{val}, y_i^{val})\}_{i=1}^m$. A pretrained model f_θ, Hyperparmeter T, r, α, β

. **foreach** $(\mathbf{x}_i^{tr}, y_i^{tr})$ *in* D_{train} **do**

  $\mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr}) \leftarrow \frac{1}{m}\sum_{j=1}^m \mathcal{I}_{(\mathbf{x}_j^{val}, y_j^{val})}(\mathbf{x}_i^{tr}, y_i^{tr})$;

$\mathcal{I}_{val}^{max} \leftarrow \max(\{\|\mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr})\|\}_{i=1}^n)$

for $t = 1$ *to* T **do**

  $\mathcal{B} \leftarrow \text{SampleMiniBatch}(D_{train}, r)$ (a mini batch of r data points)

  $\mathcal{L} \leftarrow \frac{1}{r}\sum_{\mathbf{z}\in\mathcal{B}} (\mathcal{L}_{ce}(\mathbf{x}_i^{tr}, y_i^{tr}, \theta) - \beta \cdot E_\theta(\mathbf{x}_i^{tr}) \cdot \mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr})/\mathcal{I}_{val}^{max})$

  $f_\theta \leftarrow f_\theta - \alpha \cdot \nabla_\theta \mathcal{L}$ (one SGD step)

3.3 Finetune the Model with Energy Regularization

Based on our devised influence function of energy, we propose a principled method that introduces **I**nfluence **A**ware **E**nergy **R**egularization by **F**inetuning (we refer to it as **IAERF**) as shown in Algorithm 1. It firstly calculates the average influence of energy regularization on a validation set $D_{val} = \{(\mathbf{x}_i^{val}, y_i^{val})\}_{i=1}^m$ for each training data point.

$$\mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr}) = \frac{1}{m}\sum_{j=1}^m \mathcal{I}_{(\mathbf{x}_j^{val}, y_j^{val})}(\mathbf{x}_i^{tr}, y_i^{tr}) \tag{9}$$

To reduce the loss on the validation set, we should increase the energy on the data points with a positive influence of energy regularization and decrease the energy on those with a negative influence. Therefore, we finetune the model with energy penalties determined by the corresponding influence value. The loss on a training data point z with energy regularization is:

$$\mathcal{L}'(\mathbf{x}_i^{tr}, y_i^{tr}, \theta) = \frac{1}{r}\sum_{\mathbf{z}\in\mathcal{B}} \left(\mathcal{L}(\mathbf{x}_i^{tr}, y_i^{tr}, \theta) - \beta \cdot E_\theta(\mathbf{x}_i^{tr}) \cdot \mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr})/\mathcal{I}_{val}^{max}\right) \tag{10}$$

where $\mathcal{I}_{val}^{max}$ is the maximum absolute value of influence value over the train set D_{train} and β is a hyperparameter. Specifically, in our experiments, we fix the parameter of batch normalization layers since the running mean of the batch normalization layer would directly affect the energy and may hinder the energy regularization.

3.4 Energy Penalties Influence the Model as Reweighting and Margin Control

In this section, we theoretically analyze our proposed IAERF on the classifier optimized with cross entropy loss. For a data point $(\mathbf{x}, y)$, assume the loss for classifier f_θ is:

$$\mathcal{L}(\mathbf{x}, y, \theta) = \mathcal{L}_{ce}(\mathbf{x}, y, \theta) + \hat{\beta}_\mathbf{x} \cdot E_\theta(\mathbf{x}) \tag{11}$$

where $\mathcal{L}_{ce}(\mathbf{x}, y, \theta) = -\log \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'}\exp(f_\theta(\mathbf{x})[y'])}$ is the cross-entropy loss and $\hat{\beta}_\mathbf{x} \in \mathcal{R}$ is the coefficient for the energy regularization. Then the gradient of the loss is:

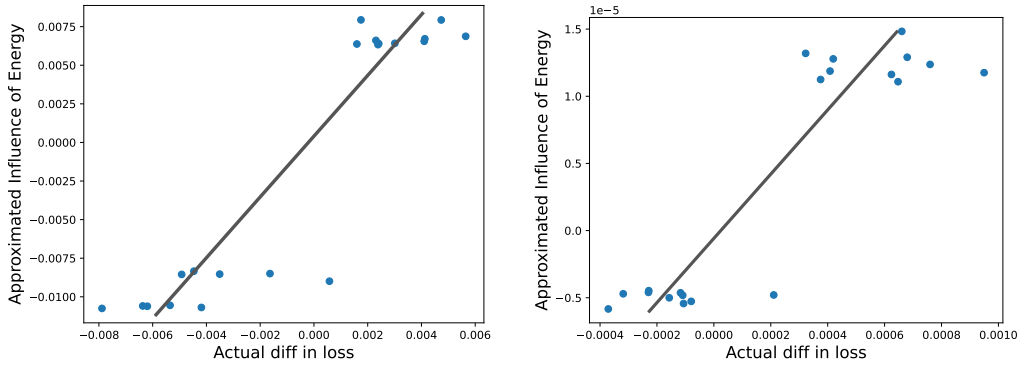
$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{x}, y, \theta)}{\partial\theta} &= \frac{\partial\mathcal{L}_{ce}(\mathbf{x}, y, \theta)}{\partial\theta} + \hat{\beta}_\mathbf{x} \cdot \frac{\partial E_\theta(\mathbf{x})}{\partial\theta} \\ &= -\frac{\partial f_\theta(\mathbf{x})[y]}{\partial\theta} - (1 - \hat{\beta}_\mathbf{x})\frac{\partial E_\theta(\mathbf{x})}{\partial\theta}\end{aligned} \tag{12}$$

For $\frac{\partial E_\theta(\mathbf{x})}{\partial\theta}$, we have:

$$\frac{\partial E_\theta(\mathbf{x})}{\partial\theta} = -\sum_{y'}\frac{\exp(f_\theta(\mathbf{x})[y'])}{\sum_i \exp(f_\theta(\mathbf{x})[i])} \cdot \frac{\partial f_\theta(\mathbf{x})[y']}{\partial\theta} = -\sum_{y'}\bar{p}(y'|\mathbf{x}) \cdot \frac{\partial f_\theta(\mathbf{x})[y']}{\partial\theta} \tag{13}$$

Combining Eq. 12 and Eq. 13, we get:

$$\frac{\partial\mathcal{L}(\mathbf{x}, y, \theta)}{\partial\theta} = \left[(1 - \hat{\beta}_\mathbf{x}) \cdot \bar{p}(y|\mathbf{x}) - 1\right]\frac{\partial f_\theta(\mathbf{x})[y]}{\partial\theta} + (1 - \hat{\beta}_\mathbf{x})\sum_{y'\neq y}\bar{p}(y'|\mathbf{x}) \cdot \frac{\partial f_\theta(\mathbf{x})[y']}{\partial\theta} \tag{14}$$



(a) ERM pretrained model (b) LDAM pretrained model

Figure 1: The positive relationship between the calculated influence function and the actual change in testing loss on pretrained ResNet-32. We plot the top 20 most influential data points.

When $\hat{\beta}_{\mathbf{x}} \neq 1$, we could further derive the gradient as:

$$\frac{\partial \mathcal{L}(\mathbf{x}, y, \theta)}{\partial \theta} = (1 - \hat{\beta}_{\mathbf{x}}) \cdot \left(\left[\bar{p}(y|\mathbf{x}) - \frac{1}{1 - \hat{\beta}_{\mathbf{x}}} \right] \frac{\partial f_\theta(\mathbf{x})[y]}{\partial \theta} + \sum_{y' \neq y} \bar{p}(y'|\mathbf{x}) \cdot \frac{\partial f_\theta(\mathbf{x})[y']}{\partial \theta} \right) \tag{15}$$

As demonstrated in Eq. 15, the influence of energy regularization is two folds: adjusting the margin and reweighting data points. When $\hat{\beta}_{\mathbf{x}}$ is positive, the energy regularization first enlarges the margin by pushing down the coefficient of $\frac{\partial f_\theta(\mathbf{x})[y]}{\partial \theta}$ and then down-weights the data point $\mathbf{x}$ with coefficient $(1 - \hat{\beta}_{\mathbf{x}})$.

Beyond that, we further highlight that adding energy regularization would not interfere the optimal classifier.

Remark 2. [Unaffected Optimal] If there exist an optimal model f_θ^* that minimizes the negative log likelihood loss. Then for any model g_υ, there always exits a model h_ξ that satisfies:

$$\forall \mathbf{x} \in \mathcal{R}^D, h_\xi(\mathbf{x}) = f_\theta^*(\mathbf{x}), E_\upsilon(\mathbf{x}) = E_\xi(\mathbf{x}) \tag{16}$$

Remark 2 indicates that if there is an optimal classifier f_θ^* for the loss with energy regularization $\mathcal{L}(\mathbf{x}, y, \theta)$, it must be the optimal classifier for negative log likelihood loss $\mathcal{L}_{ce}(\mathbf{x}, y, \theta)$.

4 Experiment

4.1 Validate the Estimated influence of energy regularization

In this section, we validate that the energy value could influence the testing performance of the classifier and the influence function could reflect the influence. Taking pretrained models, we approximate the influence function of the energy value using stochastic approximation following (Koh & Liang, 2017) over the whole testing set. Then we finetune the pretrained model with energy regularization on one of the data points and evaluate the change in testing loss compared to finetuning the model without any energy penalties. Because batch normalization may affect the energy, we fix the parameter for batch normalization during the finetune.

Specifically, we take ResNet-32 trained on long-tailed CIFAR10 (Cui et al., 2019) with ERM and LDAM (Cao et al., 2019) as the pretrained model and the model is finetuned for 50 epochs. As shown in Fig. 1, the influence function is highly correlated to the actual change in testing loss (Pearson's R $= 0.9154$ for ResNet32 pretrained with ERM and Pearson's R $= 0.9037$ for ResNet32 pretrained with LDAM). It shows that energy regularization could influence the testing loss of the classifier and that the influence function could reflect the influence. For more details and results, see Appendix B

Table 1: Average testing error (%) on Imbalanced CIFAR10 and Imbalanced CIFAR100.

Dataset	Imbalanced CIFAR10				Imbalanced CIFAR100			
Imbalance type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.39	36.70	15.73	61.68	43.41	61.45	45.30
LDAM-DRW (Cao et al., 2019)	22.84	12.38	24.64	12.58	57.96	43.33	**54.64**	43.16
ERM + IAERF	24.17	12.98	28.67	14.38	60.40	**42.41**	60.88	44.90
LDAM-DRW + IAERF	**21.63**	**12.28**	**24.11**	**12.30**	**57.19**	43.33	55.39	**43.10**

4.2 Finetune the Classifier for Imbalance Data Learning

In this section, we evaluate IAERF on the imbalanced version of CIFAR10, CIFAR100 (Cui et al., 2019) and ImageNet-LT (Liu et al., 2019), which are artificially created with class imbalance. We follow (Cao et al., 2019) to conduct experiments on imbalanced CIFAR and follow (Kang et al., 2020) to conduct experiments on ImageNet-LT.

4.2.1 Experimental Results on CIFAR

Baseline We evaluate IAERF (Algorithm 1) with ResNet-32 pretrained by: (1). Empirical risk minimization (ERM): each training data have the same weight, and the model is trained to minimize the average cross-entropy over the training set. (2). LDAM-DRW (Cao et al., 2019): LDAM introduces a label-distribution aware margin loss which enlarges the decision while DRW applies re-weighting or re-sampling after the last learning rate decay.

Dataset CIFAR10 and CIFAR100 both contain $50,000$ images in training set and $10,000$ images in testing set with 10 and 100 classes, respectively. For their imbalanced version, the number of images is reduced for each class. Specifically, we construct the imbalanced version of CIFAR10 and CIFAR100. Two types of imbalance are considered: long-tailed imbalance (Cui et al., 2019) and step imbalance (Buda et al., 2018). For long-tailed imbalance, the number of data points follows an exponential decay across different classes. For step imbalance, data points in half of the classes are reduced to the same number while the number of data points in the other classes remains the same. The imbalance ratio of the imbalanced dataset is defined as the ratio between the maximum number of data points in one class and the minimum number of data points in one class.

Implementation Note that IAERF requires calculating the influence function with a validation set. To fairly compare with the baselines, we randomly sample a class-balanced subset from the training set to compose the validation set where the number of data points per class is determined by the minimum number of data points per class in the training set. We train each model for 200 epochs and finetune it for 5 epochs. The error rate at the last epoch is reported. Refer to Appendix B for more details.

As shown in Table 1, our method could effectively boost the testing performance after only 5 epochs. The more imbalanced the more effective is IAERF and IAERF could greatly improve the ERM pretrained model. For instance, IAERF reduces the testing error of ERM pretrained model for 5.47% (from 29.64% to 24.17%) on long-tailed CIFAR10 with the imbalance ratio at 100. For LDAM-DRW pretrained model, IAERF can also improve the testing performance on CIFAR10.

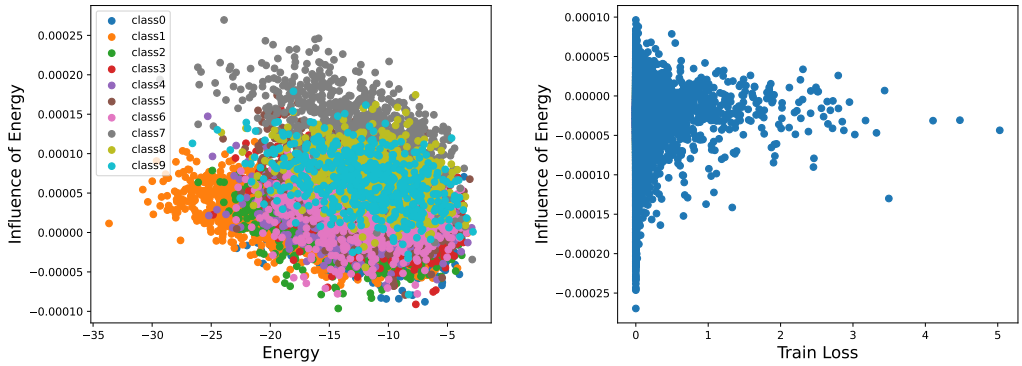
For CIFAR100, IAERF also improves the testing performance for ERM pretrained model and improves the LDAM-DRW pretrained model on long-tailed CIFAR100 with the imbalance ratio at 100 and step imbalanced CIFAR100 with the imbalance ratio at 10. However, the improvement on imbalanced CIFAR100 brought by our IAERF is much smaller than that on imbalanced CIFAR10. We conjecture that the calculated influence of energy regularization on our sampled validation set for CIFAR100 is less accurate since the number of images per class in CIFAR100 is much smaller than that of CIFAR10 *e.g.* only 5 images for the least frequent class when imbalance ratio is 100.

4.2.2 Experimental Results on ImageNet-LT and iNatrualist 2018

Baseline We evaluate IAERF with ResNeXt-50 (Xie et al., 2017) pretrained by the techniques and protocols proposed in (Kang et al., 2020) where each model is divided into two parts: backbone and linear classifier. The protocol include (1) Joint: jointly train the backbone and the linear classifier

Table 2: Testing accuracy(%) on ImageNet-LT and iNatrualist.

Dataset	ImageNet-LT				iNaturalist			
Method	Many	Median	Few	All	Many	Median	Few	All
Joint (Kang et al., 2020)	**65.9**	37.5	7.7	44.4	**78.2**	70.6	64.7	69.0
Joint + IAERF	65.5	38.8	9.0	45.0	76.8	68.1	66.8	68.8
cRT (Kang et al., 2020)	64.7	48.6	28.6	52.1	75.9	71.9	69.1	71.2
cRT + IAERF	64.7	49.3	29.0	52.4	76.1	71.6	69.5	71.2
LWS (Kang et al., 2020)	63.4	**50.0**	33.3	**52.9**	74.3	**72.4**	71.2	72.1
LWS + IAERF	63.1	**50.0**	34.1	**52.9**	74.4	**72.4**	**71.6**	**72.3**



(a) Influence of energy regularization against the energy (b) Influence of energy regularization against the loss

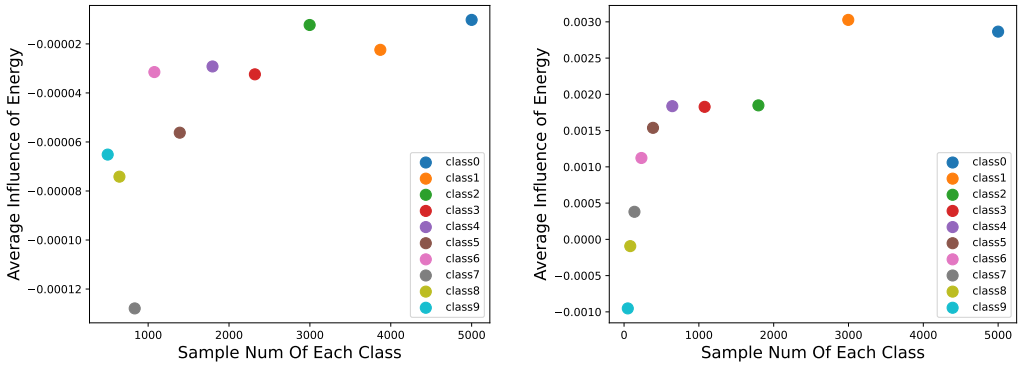
Figure 2: We calculate the influence of energy regularization, the loss, and the energy at each training data point in long-tailed CIFAR10 with the imbalnaced ratio at 10 for the ResNet-32 trained with ERM. We plot the influence of energy regularization against the loss and energy. Each point denotes a training data point.

with conventional cross-entropy loss and instance balance which equals Empirical risk minimization (ERM). (2) Classifier Re-training (cRT): employ the backbone trained with ERM and retrain the linear classifier with the class balance sampling method. (3) Learnable Weight Scaling (LWS): Rescale the weight of the classifier for each class by a rescale factor learned with the class balance sampling method as in cRT.

Dataset We conduct our experiments on ImageNet-LT (Liu et al., 2019) and iNaturalist 2018 (Van Horn et al., 2018). ImageNet-LT is artificially truncated from ImageNet (Deng et al., 2009), where the label distribution follows a long-tailed distribution. It has 1000 classes and the number of images per class ranges from 1280 to 5 images. iNatualist 2018 is a real-world, naturally long-tailed dataset with 8142 classes. We follow (Liu et al., 2019) and report the testing accuracy on three kind of the set of classes: *Many-shot* (more than 100 images), *Medium-shot* (20 $\sim$ 100 images) and *Few-shot* (less than 20 images). The testing accuracy on all classes is denoted as *All*

Implementation We employ the backbone provided by (Kang et al., 2020) and finetune or retrain the classifier. For ImageNet-LT, we calculate the influence of energy regularization on the validation set of the ResNeXt-50 pretrained for 90 epochs and retrain the classifiers with energy regularization as proposed in Algorithm 1 for 10 epochs. Specifically, for joint training, we jointly finetune the backbone and the classifier without class balance sampling. For iNaturalist 2018, we take the subset of the training set that contains the images of *few*-shot classes to calculate the influence of energy regularization of the ResNet-152 pretrained for 200 epochs. The classifiers on the iNaturalist are retrained for 30 epochs with energy regularization. For more details, please refer to Appendix B.

As shown in Table 2, for ImageNet-LT, IAERF could boost the accuracy of Median-shot classes and Few-shot classes at the cost of decreasing the accuracy of Many-shot classes. For iNatrualist 2018, due to the lack of validation set and the influence function is calculated on the *Few*-shot classes of the training set, the accuracy on *Few*-shot classes is improved while the accuracy for *Many*-shot and *Median*-shot is decreased.



(a) long-tailed CIFAR10 with imbalance ratio at 10 (b) long-tailed CIFAR10 with imbalance ratio at 100

Figure 3: We average the influence of energy regularization over the training data points of each class, and plot it against the number of data points for each class in long-tail CIFAR10.

4.3 A Closer Look on the Influence of Energy Regularization

In this section, we take a closer look on the influence of energy regularization and provide some insights. We conduct experiment on the ResNet-32 trained on the imbalanced CIFAR10. For more details and more results on imbalanced CIFAR100, please refer to Appendix B.

4.3.1 The Influence of Energy Regularization on Different Data Point

The influence of energy regularization is not correlated to the energy value and the training loss As shown in Fig. 2(a), we find that generally the influence of energy regularization on the training data point is not related to the energy value of the data point (Pearson's R is -0.21). As for the training loss, Fig. 2(b) shows that the influence of energy regularization on the data points of similar training loss ranges from positive to negative. The influence of energy regularization is also not related to the training loss (Pearson's R is -0.04). *It indicates that one could not predict the influence of energy regularization on the data point based on the training loss or the energy value of the data.*

Energy regularization have large influence mainly on well-classified data points As shown in Fig. 2(b), the range of the influence of energy regularization expands as the training loss decreases. Therefore, data points where energy regularization have a large influence are generally well-classified data points with low loss. As pointed out in Remark 1, the energy value is unstable during the training, *we conjecture that the unregularized energy value of well-classified data points is one of the possible reasons for the overfitting of the classifier.*

4.3.2 Relationship between The Influence of Energy regularization & Classification

The influence of energy regularization correlates with data distribution As shown in Fig. 3, the average influence of energy regularization of different class is positively correated to the number of data points of the corresponding class (The Pearson's R is 0.70 for long-tailed CIFAR10 with imbalance ratio at 10 and the Pearson's R is 0.77 for long-tailed CIFAR10 with imbalance ratio at 100). The lower the influence of the energy regularization means that the testing loss of the classifier would be lower after adding a positive energy regularization. It indicates that pushing down the energy value of data points of less frequent class and pulling up the energy value of data points of more frequent class would boost the testing performance *i.e.* pull up the predicted $\bar{p}(\mathbf{x})$ for less frequent class and push down the predicted $\bar{p}(\mathbf{x})$ for more frequent class. Since the probability density $p(\mathbf{x})$ of data points of less frequent class is much lower and the $p(\mathbf{x})$ of data points of more frequent class is much higher in the imbalanced training set compared to the testing set, it shows that pushing the predicted $\bar{p}(\mathbf{x})$ closer to the real $p(\mathbf{x})$ of the testing set would boost the testing performance. *We conjecture that energy regularization changes the distance between the predicted data distribution and the real testing data distribution and thus influence the generalization ability of the model.*

5 Related Works and Further Discussion

Energy Based Learning Energy-based models (EBMs) (LeCun et al., 2006; Ranzato et al., 2006; 2007) provide a unified theoretical framework for various learning models. EBMs associate a scalar energy value to each configuration of the variables where the energy value is lower for observed configurations than the unobserved ones. Many recent works employ the energy function defined on discriminative models for other tasks *e.g.* generative learning or OOD detection. (Xie et al., 2016) show that a generative random field model can be derived from a discriminative neural network. While (Grathwohl et al., 2020) find that neural classifiers are also an energy-based model for joint distribution and propose a hybrid model that acts as both a discriminative model and a generative model. (Liu et al., 2020) propose to use the energy value to detect out-of-distribution (OOD) samples, which has been theoretically proved (Bitterwolf et al., 2022) to be equal to training an additional binary discriminator. The influence of this specific energy on the discriminative model itself is often simply assumed as the lower energy value on the training set the better (Zhao et al., 2022). While recent work (Xie et al., 2022) minimizes the distance of energy distribution between the source domain and target domain to enhance the model performance in domain adaptation (see (Wang & Deng, 2018) and reference therein), the influence of the energy needs a closer look.

Imbalanced Dataset Learning Imbalanced dataset learning has drawn increasing attention *i.e.* the long-tail recognition (Wang et al., 2017; Zhou et al., 2018; Liu et al., 2019; Zhong et al., 2019; He et al., 2021) due to the pervasiveness of the imbalanced data in real-world scenarios. Most methods could be divided into three categories: re-sampling the data, re-weighting the loss, and transfer learning. For re-sampling, various methods have been proposed to re-sample the dataset to achieve a more balanced data distribution (Chawla et al., 2002; Estabrooks et al., 2004; Han et al., 2005; Liu et al., 2009; Shen et al., 2016; Wang et al., 2020; Zhang & Pfister, 2021). Specifically, (Liu et al., 2019) employs a two-stage scheme where the model learns the representation in the first stage without balancing and is finetuned in the second stage with a memory module while (Kang et al., 2020) proposes to decouple the representation learning and the classifier training where it trains the backbone to learn the representation without any balancing method and only adjust the classifier for good performance. As for re-weighting, re-weighting methods assign different losses to different classes (Zhang et al., 2018; Zhao et al., 2019; Ye et al., 2020; Hsieh et al., 2021) or different data samples (Lin et al., 2017; Ren et al., 2018; Shu et al., 2019) to achieve a more balanced performance on each class. Specifically, LDAM (Cao et al., 2019) proposes a distribution aware loss that enlarges the margin to less frequent (tail) classes. Another direction views the imbalanced data as a label shift problem. Methods have been proposed to transfer the knowledge from the more frequent (head) classes to the less frequent (tail) classes (Wang et al., 2017; Chu et al., 2020; Wang et al., 2021).

Compared to an imbalanced dataset where the target distribution is clearly known, the distribution shift in domain adaptation is implicit and only could be estimated, which makes it more challenging. As shown in Sec. 4.3.2, we have the assumption that the influence of energy regularization comes from interfering with the distance between the predicted data distribution corresponding to the energy value and the real testing data distribution. Therefore, our method probably also provides a way to estimate the distribution under scenarios with implicit distribution shift *e.g.* domain adaptation.

6 Conclusion

In this paper, we propose to study the influence of energy regularization on classifiers under an imbalanced dataset learning setting. Based on the influence function, a classic method in robust statistics, we manage to quantify the influence and further propose an influence-aware energy regularization method that regularizes the energy value on training data point according to the quantified influence. Theoretically, we prove that our method could re-weight the loss and affect the margin without hindering the optimization. Empirically, we show that the proposed quantification correctly reflects the influence of energy regularization and we verify the effectiveness on various imbalanced datasets. The main limitation of our method is that our method requires a validation set to calculate the influence. For most of the imbalanced datasets, we have to re-sample the training set to form a validation set, which may interfere with the effectiveness of our method. While this work mainly focuses on imbalanced dataset learning, the influence-aware energy regularization may be applicable for other data distribution shift settings *e.g.* domain adaptation which we leave for future work. We also hope the insights of how the energy influence the classifier could provide inspiration for future works.

REFERENCES

Julian Bitterwolf, Alexander Meinke, Maximilian Augustin, and Matthias Hein. Breaking down out-of-distribution detection: Many methods based on OOD training data estimate a combination of the same core quantities. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2041–2074. PMLR, 2022.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pp. 1565–1576, 2019.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *ECCV (29)*, volume 12374 of *Lecture Notes in Computer Science*, pp. 694–710. Springer, 2020.

R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pp. 9268–9277. Computer Vision Foundation / IEEE, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. IEEE Computer Society, 2009.

Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Comput. Intell.*, 20(1):18–36, 2004.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*. OpenReview.net, 2020.

Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *ICIC (1)*, volume 3644 of *Lecture Notes in Computer Science*, pp. 878–887. Springer, 2005.

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.

Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *ICCV*, pp. 235–244. IEEE, 2021.

Ting-I Hsieh, Esther Robb, Hwann-Tzong Chen, and Jia-Bin Huang. Droploss for long-tail instance segmentation. In *AAAI*, pp. 1549–1557. AAAI Press, 2021.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*. OpenReview.net, 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2999–3007. IEEE Computer Society, 2017.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. Part B*, 39(2):539–550, 2009.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pp. 2537–2546. Computer Vision Foundation / IEEE, 2019.

Alston Lo and Juhan Bae. torch-influence, 2022. URL https://github.com/alstonlo/torch-influence.

Marc'Aurelio Ranzato, Christopher S. Poultney, Sumit Chopra, and Yann LeCun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, pp. 1137–1144. MIT Press, 2006.

Marc'Aurelio Ranzato, Y-Lan Boureau, Sumit Chopra, and Yann LeCun. A unified energy-based framework for unsupervised learning. In *AISTATS*, volume 2 of *JMLR Proceedings*, pp. 371–379. JMLR.org, 2007.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4331–4340. PMLR, 2018.

Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV (7)*, volume 9911 of *Lecture Notes in Computer Science*, pp. 467–482. Springer, 2016.

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 *NeurIPS*, pp. 1917–1928, 2019.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. RSG: A simple but effective module for learning imbalanced datasets. In *CVPR*, pp. 3784–3793. Computer Vision Foundation / IEEE, 2021.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV (14)*, volume 12359 of *Lecture Notes in Computer Science*, pp. 728–744. Springer, 2020.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NIPS*, pp. 7029–7039, 2017.

Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, Xinjing Cheng, and Guoren Wang. Active learning for domain adaptation: An energy-based approach. In *AAAI*, pp. 8708–8716. AAAI Press, 2022.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 2635–2644. JMLR.org, 2016.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pp. 5987–5995. IEEE Computer Society, 2017.

Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *CoRR*, abs/2001.01385, 2020.

Yifan Zhang, Peilin Zhao, Jiezhang Cao, Wenye Ma, Junzhou Huang, Qingyao Wu, and Mingkui Tan. Online adaptive asymmetric active learning for budgeted imbalanced data. In *KDD*, pp. 2768–2777. ACM, 2018.

Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *ICCV*, pp. 705–714. IEEE, 2021.

Guangxiang Zhao, Wenkai Yang, Xuancheng Ren, Lei Li, Yunfang Wu, and Xu Sun. Well-classified examples are underestimated in classification with deep neural networks. In *AAAI*, pp. 9180–9189. AAAI Press, 2022.

Peilin Zhao, Yifan Zhang, Min Wu, Steven C. H. Hoi, Mingkui Tan, and Junzhou Huang. Adaptive cost-sensitive online classification. *IEEE Trans. Knowl. Data Eng.*, 31(2):214–228, 2019.

Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, pp. 7812–7821. Computer Vision Foundation / IEEE, 2019.

Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6): 1452–1464, 2018.

A PROOFS

A.1 PROOF FOR PROPOSITION 1

Proof. For the classifier $f_\theta : \mathcal{R}^D \to R^K$, assume the energy on data point $\mathbf{x} \in \mathcal{R}^D$ is $E_\theta(\mathbf{x})$. For $\forall \mathcal{E} \in \mathcal{R}$, a classifier $g_\eta : \mathcal{R}^D \to R^K$ could be defined that satisfies:

$$\forall i \in \{1, 2, \cdots, K\}, \quad g_\eta(\mathbf{x})[i] = f_\theta(\mathbf{x})[i]) - \mathcal{E} + E_\theta(\mathbf{x}) \tag{17}$$

Then for the predicted $\bar{p}_\eta(y|\mathbf{x})$ we have:

$$\begin{aligned} \bar{p}_\eta(y|\mathbf{x}) &= \frac{\exp\left[f_\theta(\mathbf{x})[y]) - \mathcal{E} + E_\theta(\mathbf{x})\right]}{\sum_i \exp\left[f_\theta(\mathbf{x})[i]) - \mathcal{E} + E_\theta(\mathbf{x})\right]} \\ &= \frac{\exp\left[f_\theta(\mathbf{x})[y])\right]}{\sum_i \exp\left[f_\theta(\mathbf{x})[i])\right]} \\ &= \bar{p}_\theta(y|\mathbf{x}) \end{aligned} \tag{18}$$

For the energy $E_\eta(\mathbf{x})$ on the g_η, we have:

$$\begin{aligned} E_\eta(\mathbf{x}) &= -\log \sum_i \exp\left[g_\eta(\mathbf{x})[i]\right] \\ &= -\log \sum_i \exp\left[f_\theta(\mathbf{x})[i] - \mathcal{E} + E_\theta(\mathbf{x})\right] \\ &= -\log \left(\exp[E_\theta(\mathbf{x}) - \mathcal{E}] \cdot \sum_i \exp\left[f_\theta(\mathbf{x})[i]\right] \right) \\ &= \mathcal{E} - E_\theta(\mathbf{x}) - -\log \sum_i \exp\left[f_\theta(\mathbf{x})[i]\right] \\ &= \mathcal{E} - E_\theta(\mathbf{x}) + E_\theta(\mathbf{x}) \\ &= \mathcal{E} \end{aligned} \tag{19}$$

□

A.2 PROOF FOR REMARK 1

Proof. The negative log likelihodd loss is:

$$\begin{aligned}\mathcal{L}_{nll}(\mathbf{x}, y, \theta) &= -\log \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_i \exp(f_\theta(\mathbf{x})[i])} \\ &= \log \sum_i \exp(f_\theta(\mathbf{x})[i]) - f_\theta(\mathbf{x})[y]) \\ &= -E_\theta(\mathbf{x}) - f_\theta(\mathbf{x})[y])\end{aligned} \tag{20}$$

Then gradient of negative log likelihood loss *w.r.t* parameter θ is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{nll}(\mathbf{x}, y, \theta)}{\partial \theta} &= \frac{\partial[-E_\theta(\mathbf{x}) - f_\theta(\mathbf{x})[y])]}{\partial \theta} \\ &= -\frac{\partial E_\theta(\mathbf{x})}{\partial \theta} - \frac{\partial f_\theta(\mathbf{x})[y])}{\partial \theta}\end{aligned} \tag{21}$$

As the loss decreasing, $-\frac{\partial E_\theta(\mathbf{x})}{\partial \theta}$ will increase the energy while $-\frac{\partial f_\theta(\mathbf{x})[y])}{\partial \theta}$ will decrease the the energy by increasing $f_\theta(\mathbf{x})[y])$.

□

A.3 PROOF FOR REMARK 2

Proof. Similar to the proof for Proposition 1, for the optimal classifier f_θ^* and an arbitrary classifier g_υ, we could define a classifier h_ξ as:

$$h_\xi(\mathbf{x}) = f_\theta^*(\mathbf{x}) - E_\upsilon(\mathbf{x}) + E_\theta(\mathbf{x}) \tag{22}$$

Then for the predicted probability:

$$\begin{aligned}\bar{p}_\xi(y|\mathbf{x}) &= \frac{\exp\left[f_\theta^*(\mathbf{x})[y]) - E_\upsilon(\mathbf{x}) + E_\theta(\mathbf{x})\right]}{\sum_i \exp\left[f_\theta^*(\mathbf{x})[i]) - E_\upsilon(\mathbf{x}) + E_\theta(\mathbf{x})\right]} \\ &= \frac{\exp\left[f_\theta^*(\mathbf{x})[y])\right]}{\sum_i \exp\left[f_\theta^*(\mathbf{x})[i])\right]} \\ &= \bar{p}_\theta(y|\mathbf{x})\end{aligned} \tag{23}$$

For the energy $E_\xi(\mathbf{x})$ on the h_ξ, we have:

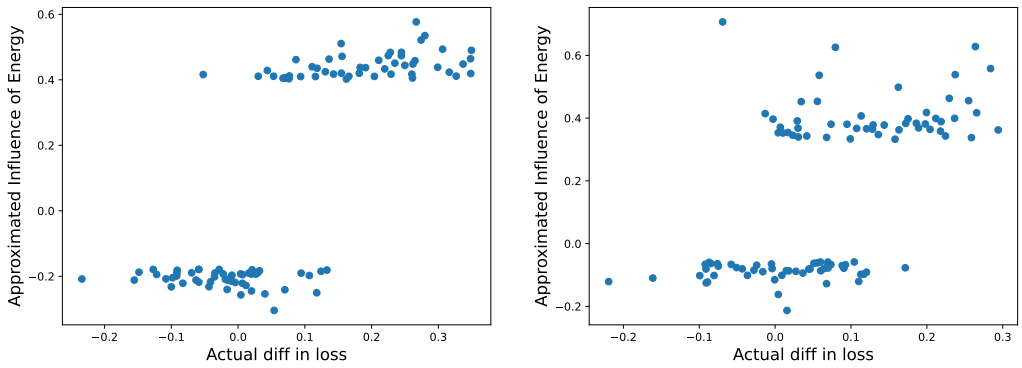
$$\begin{aligned}E_\xi(\mathbf{x}) &= -\log \sum_i \exp\left[h_\xi(\mathbf{x})[i]\right] \\ &= -\log \sum_i \exp\left[f_\theta^*(\mathbf{x})[i] - E_\upsilon(\mathbf{x}) + E_\theta(\mathbf{x})\right] \\ &= -\log \left(\exp[E_\theta(\mathbf{x}) - E_\upsilon(\mathbf{x})] \cdot \sum_i \exp\left[f_\theta^*(\mathbf{x})[i]\right] \right) \\ &= E_\upsilon(\mathbf{x}) - E_\theta(\mathbf{x}) - -\log \sum_i \exp\left[f_\theta^*(\mathbf{x})[i]\right] \\ &= E_\upsilon(\mathbf{x})\end{aligned} \tag{24}$$

□

B EXPERIMENT DETAILS

B.1 DETAILS FOR THE CALCULATION OF INFLUENCE FUNCTION

The calculation of influence function requires a validation set. For imbalanced CIFAR10 and CIFAR100, we sample data points of each class from the class-imbalanced training set to compose



(a) long-tailed CIFAR10 (b) long-tailed CIFAR100

Figure 4: The relationship between the influence function and the actual change in testing loss on one testing point. We plot the top 100 most influential data points.

the validation set. While the number of data points per class is determined by the minimum number of data points per class in the training set. For ImageNet-LT, since it have a val split, we use the val split to calculate the influence function. For iNaturalist, since the minimum number of data points per class is small (2 images per class), we take the few-shot classes of the trainig set as the validation set.

We calculate the influence function with stochastic estimation (Cook & Weisberg, 1982) following (Koh & Liang, 2017). We implement the calculation of influence function based on the Python package for calculating influence function (Lo & Bae, 2022). For imbalanced CIFAR10 and imbalanced CIFAR100, the influence function is calculated with stochastic estimation for 5000 iteration and averaged over 10 trails. For ImageNet-LT and iNatrualist, we calculate the influence function only on the classifier, and theinfluence function is calculated with stochastic estimation for 2500 iteration and averaged over 10 trails.

B.2 Details for the experiments on imbalanced dataset

We follow the setting in (Cao et al., 2019) to train ResNet-32 on imbalanced CIFAR dataset and report the performance of the model at the final epoch. The model is trained for 200 epochs with SGD optimizer where learning rate at 0.1, momentum at 0.9 and weight decay at $2e-4$. The learning rate is decayed with factor 0.01 at 160-th epoch and 180-th epoch. For IAERF, we finetune the model for 5 epochs with batch size at 128 and learning rate at $1e-4$.

For ImageNet-LT and iNaturalist 2018, we employ the pretrained model provided in (Kang et al., 2020) and follow the setting in it to finetune the ResNeXt-50 (Xie et al., 2017) on ImgaeNet-LT and the ResNet-152 (He et al., 2016) on iNaturalist 2018. For ImageNet-LT, the classifier is finetuned for 10 epochs with batch size at 512 and learning rate at 0.2. For iNaturalist, the classifier is finetuned for 30 epochs with batch size at 512 and learning rate at 0.2.

The β in Algorithm 1 is searched in $\{0.1, 0.5, 1, 10\}$ and when the abosolute value of energy regularization is bigger than the cross-entropy loss the β is set to be $\| \frac{\mathcal{L}_{ce}(\mathbf{x}_i^{tr}, y_i^{tr}, \theta)}{E_\theta(\mathbf{x}_i^{tr}) \cdot \mathcal{I}_{val}(\mathbf{x}_i^{tr}, y_i^{tr}) / \mathcal{I}_{val}^{max}} \|$

C Additional Experiment Results

C.1 Additional Results to Validate the Influence Function of Energy Regularization

In addition to Fig. 1 where we calculate the average influence over the whole testing set, we arbitrary pick a wrongly classified data point and calculate the influence function for this data point following (Koh & Liang, 2017).

We calculate the influence function for the ResNet-32 trained with ERM on the long-tail CIFAR10 and long-tail CIFAR100 where the imbalance ratio is set to be 100. We plot the influence of energy

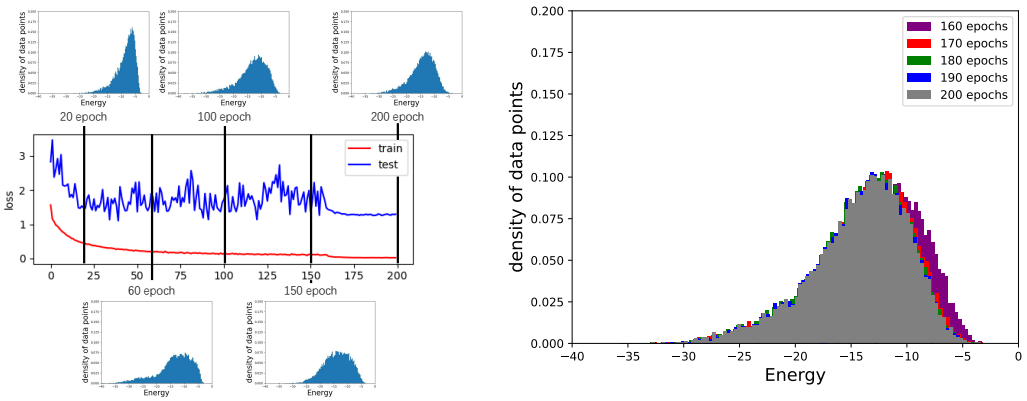


Figure 5: The energy distribution of a ResNet-32 trained on the long-tailed CIFAR10. (a): the energy distribution of the training set on different epochs during the taining (b): the energy distribution of the training set at 160, 170, 180, 190 and 200 epoch.

regularization and the actual change in testing loss after finetune the model with energy regularization for 50 epochs on 100 most influential data point. As shown in Fig. 4, the calculated influence function has a positive relation to the actual change in loss (Pearson's R is 0.7875 on long-tail CIFAR10 and is 0.5744 on long-tail CIFAR100)

C.2 Energy Distribution Shifts During Training

We plot the energy distribution of the training set during the training of ResNet-32 by ERM on the long-tailed CIFAR10 with imbalance ratio at 100. As shown in Fig. 5, the energy distribution keeps changing during the training even thought the training loss is stable *e.g.* from 100-th epoch to 150-th epoch. We further plot the distribution at 160, 170, 180, 190 and 200 epoch after the learning rate have decayed. As shown in Fig. 5(b), the energy distribution of the training set still changes when the learning rate is decayed and the model is converged.