Minimax Generalized Cross-Entropy

Kartheek Bondugula[1] **Santiago Mazuelas**[1,2] **Aritz Pérez**[1] **Anqi Liu**[3]
[1] Basque Center for Applied Mathematics, Bilbao, Spain
[2] Ikerbasque, Basque Foundation for Science
[3] Johns Hopkins University, Baltimore, USA
{kbondugula, smazuelas, aperez}@bcamath.org, aliu.cs@jhu.edu

Abstract

Loss functions play a central role in supervised classification. Cross-entropy (CE) is widely used, whereas the mean absolute error (MAE) loss can offer robustness but is difficult to optimize. Interpolating between the CE and MAE losses, generalized cross-entropy (GCE) has recently been introduced to provide a trade-off between optimization difficulty and robustness. Existing formulations of GCE result in a non-convex optimization over classification margins that is prone to underfitting, leading to poor performances with complex datasets. In this paper, we propose a minimax formulation of generalized cross-entropy (MGCE) that results in a convex optimization over classification margins. Moreover, we show that MGCEs can provide an upper bound on the classification error. The proposed bilevel convex optimization can be efficiently implemented using stochastic gradient computed via implicit differentiation. Using benchmark datasets, we show that MGCE achieves strong accuracy, faster convergence, and better calibration, especially in the presence of label noise.

1 INTRODUCTION

The learning process in supervised classification is strongly influenced by the choice of loss function. Cross-entropy (CE) has long been the standard choice for classification tasks in machine learning. On the other hand, the mean absolute error (MAE) loss is known to offer increased robustness to noise (Ghosh et al., 2017), but is difficult to optimize (Zhang and Sabuncu, 2018). To address this trade-off between optimization difficulty and robustness, a more general class of loss functions has been recently proposed, termed as generalized cross-entropy (GCE), which interpolate between the MAE and CE losses through a tunable parameter (Zhang and Sabuncu, 2018; Sypherd et al., 2022a). These losses have been shown to alleviate the optimization challenges of the MAE loss while achieving improved robustness over CE (Zhang and Sabuncu, 2018).

The GCE losses have been defined by leveraging probabilistic losses known as α-losses, which vary depending on the value of α (Ferrari and Yang, 2010; Liao et al., 2018; Sypherd et al., 2020, 2022a,b). Specifically, GCEs are obtained by first transforming classification margins into probabilities using a link function, and then using an α-loss over the resulting probabilities. In existing methods, the same link function (softmax) is used for all values of α, and the final classification loss becomes non-convex over margins. As a result, existing methods based on GCE are susceptible to underfitting, particularly when trained on complex datasets (Zhou et al., 2021; Staats et al., 2025).

Recently, several works have proposed a minimax approach for classification with the 0-1 loss that results in a convex optimization over the classification margins (Asif et al., 2015; Fathony et al., 2016, 2018; Mazuelas et al., 2023). Since the MAE loss coincides with the 0-1 loss of randomized classification rules, such approaches provide a convex formulation for the MAE loss. Convexity is a desirable property for a loss function (Zhang, 2004; Bartlett et al., 2006), and convex surrogates often provide favorable optimization landscapes that improve convergence rates in large-scale machine learning (Agarwal and Agarwal, 2015). However, it is still unclear whether it is possible to find a convex optimization solution to GCEs.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

In this paper, we propose a minimax approach for generalized cross-entropy (MGCE) that results in a bilevel convex optimization over classification margins. The presented methods achieve the benefits provided by general α-losses and avoid the underfitting issues of existing GCE methods. Specifically, the contributions of the paper can be summarized as follows:

- We present MGCE methods that minimize the worst-case expected α-loss over distributions in an uncertainty set.
- We show that the MGCE method can provide an upper bound on the classification error, and that the MGCE margin loss is classification-calibrated. Moreover, we characterize the relation between the worst-case distribution in the uncertainty set and the corresponding minimax classifier for general α-losses.
- We develop an efficient learning algorithm for the proposed bilevel convex optimization. Specifically, we derive the corresponding stochastic gradients using implicit differentiation, and propose a bisection-based algorithm to compute them efficiently, enabling scalability to complex models such as deep neural networks (DNNs).
- Using large benchmark datasets and DNNs, we demonstrate that MGCE methods achieve strong accuracy and faster convergence. Additionally, the MGCEs yield better calibrated models than GCEs, particularly under label noise.

Notations: For a set $\mathcal{S}$, we denote its cardinality as $|\mathcal{S}|$; bold lowercase and uppercase letters represent vectors and matrices, respectively; for a vector $\mathbf{b}$ and an index i, b_i denotes the component at index i; for a vector-valued function $f(x)$ and an index i, $f(x)_i$ denotes the i-th component of $f(x)$; $\mathbf{I}$ denotes the identity matrix; $\mathbb{1}\{\cdot\}$ denotes the indicator function; $\mathbf{1}$ denotes a vector of ones; for a vector $\mathbf{v}$, $|\mathbf{v}|$ and $(\mathbf{v})_+$, denote its component-wise absolute value and positive part, respectively; for a scalar v, $(\mathrm{v})_+^\beta$ denotes the positive part raised to the power of β; $\otimes$ denotes the Kronecker product; $\preceq$ and $\succeq$ denote vector inequalities; $\mathbb{E}_{\mathrm{p}}\{\cdot\}$ denotes the expectation of its argument with respect to distribution p.

2 RELATED WORK

This section introduces the main terminology, reviews existing approaches for GCE, and briefly outlines the minimax framework of Mazuelas et al. (2022, 2023).

2.1 Preliminaries

Let $\mathcal{X}$ and $\mathcal{Y}$ be the set of features and labels in a supervised classification problem with k classes. The goal is to find a classifier $\mathrm{h} : \mathcal{X} \to [0,1]^k$ that maps each feature vector $x \in \mathcal{X}$ to labels' probabilities. Specifically, given $x \in \mathcal{X}$, we denote by $\mathrm{h}(x)$ the vector of class probabilities given x, where the probability of label $y \in \mathcal{Y}$ is given by the y-th component of the vector, $\mathrm{h}(x)_y$. Commonly, the probability vector is obtained using a link function, such as softmax function, over the classifier margins. The classifier margins over different classes are represented by the vector

$$f(x, \boldsymbol{\mu}) = [\Phi(x,1)^\top \boldsymbol{\mu}, \Phi(x,2)^\top \boldsymbol{\mu}, \ldots, \Phi(x,k)^\top \boldsymbol{\mu}], \quad (1)$$

where $\Phi(x,y) \in \mathbb{R}^m$ is the feature mapping corresponding to the instance-label pair (x,y), e.g., one-hot encoding of the penultimate layer in a DNN, and $\boldsymbol{\mu} \in \mathbb{R}^m$ are coefficients learned by the classifier. The margin corresponding to class y is the y-th component of $f(x, \boldsymbol{\mu})$, i.e., $f(x, \boldsymbol{\mu})_y = \Phi(x,y)^{\mathrm{T}} \boldsymbol{\mu}$.

We denote by $\Delta(\mathcal{X} \times \mathcal{Y})$ the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$. If p^* denotes the underlying distribution of the instance-label pairs, the classification risk of a rule h under the loss ℓ is given by

$$\mathcal{R}_\ell(\mathrm{h}) = \mathbb{E}_{\mathrm{p}^*} \ell(\mathrm{h}, (x,y)). \quad (2)$$

The most commonly used probabilistic loss, especially in modern DNNs, is the CE that is defined as $\ell_{\mathrm{CE}}(\mathrm{h}, (x,y)) = -\log \mathrm{h}(x)_y$. However, such loss function can lead to overfitting and miscalibrated models (Guo et al., 2017). Besides CE, the MAE loss function defined as $\ell_{\mathrm{MAE}}(\mathrm{h}, (x,y)) = 1 - \mathrm{h}(x)_y$ is more robust to noise while may pose significant challenges for optimization (Zhang and Sabuncu, 2018). The MAE loss coincides with the 0-1 loss of randomized classification rules. Specifically, if $\mathrm{h}(x)_y$ is the probability with which rule h assigns label y to instance x, the value $1 - \mathrm{h}(x)_y$ is the probability of incorrectly labeling the example (x,y).

2.2 Generalized cross-entropy

The GCE losses have recently been proposed as a family of loss functions that interpolate between the CE and MAE losses, thereby offering a trade-off between robustness and optimization difficulty (Zhang and Sabuncu, 2018; Sypherd et al., 2020, 2022b). GCE losses are obtained using a family of probabilistic losses referred to as α-loss (Sypherd et al., 2020) and defined as

$$\ell_\alpha(\mathrm{h}, (x,y)) = \frac{\alpha}{\alpha - 1}\left(1 - \mathrm{h}(x)_y^{\frac{\alpha-1}{\alpha}}\right). \quad (3)$$

The minimum expected α-loss (Bayes risk) can be related to generalized notions of entropy as described in Mazuelas et al. (2022). In particular, the Bayes risk corresponding to an α-loss coincides with the Arimoto

conditional entropy, which is closely related to Tsallis entropy.

For notational convenience, we parametrize α-losses using $\beta = \alpha/(\alpha-1)$ leading to the following formulation

$$\ell_\beta(\mathrm{h}, (x, y)) = \beta(1 - \mathrm{h}(x)_y^{\frac{1}{\beta}}). \tag{4}$$

For $\beta \geq 1$, such probabilistic losses interpolate between the MAE loss and the CE loss: specifically, $\beta = 1$ recovers the MAE loss, while $\beta \to \infty$ yields the CE loss. Figure 1 shows the variation of the loss function with respect to β. For $\beta \in [1, \infty)$, the corresponding loss is noise-robust, since the sum of the losses over all classes is bounded as

$$\beta(k - k^{\frac{\beta-1}{\beta}}) \leq \sum_{y=1}^{k} \ell_\beta(\mathrm{h}, (x, y)) \leq \beta(k-1), \tag{5}$$

as shown in Zhang and Sabuncu (2018).

Existing GCE methods obtain classification losses from the α-losses in (4) by transforming the classifier's margins to probabilities using a link function that does not depend on the value of α (Zhang and Sabuncu, 2018; Sypherd et al., 2020). Such methods use the softmax link so that the probability of class y for a given margin $f(x, \boldsymbol{\mu})$ is obtained as $\mathrm{h}(x)_y = e^{f(x,\boldsymbol{\mu})_y} / \sum_{i=1}^{k} e^{f(x,\boldsymbol{\mu})_i}$. Therefore, existing GCE methods address the following minimization at training

$$\min_{\boldsymbol{\mu}} \sum_{i=1}^{n} \beta \left(1 - \frac{e^{f(x_i,\boldsymbol{\mu})_{y_i}/\beta}}{\left(\sum_{j=1}^{k} e^{f(x_i,\boldsymbol{\mu})_j}\right)^{1/\beta}} \right), \tag{6}$$

where $f(x_i, \boldsymbol{\mu})_j$ denotes the classification margin for instance x_i and class j as defined in (1).

For $1 \leq \beta < \infty$, the classification loss in (6) is non-convex over classification margins (Zhang and Sabuncu, 2018; Sypherd et al., 2022a), resulting in optimization difficulties especially in complex datasets (Zhou et al., 2021; Staats et al., 2025).

Convexity remains a highly desirable property for a loss function, as it can enable tractable optimization (Zhang, 2004; Bartlett et al., 2006). Minimax approaches have recently provided convex formulations for the 0-1 loss (and hence for the MAE loss) (Asif et al., 2015; Fathony et al., 2016, 2018; Mazuelas et al., 2020, 2022, 2023). In the following, we briefly describe the minimax framework that will be used in Section 3 to derive the proposed MGCE methods for general α-losses.

2.3 Minimax framework

The minimax framework obtains classification rules by minimizing the worst-case expected loss over distributions in an uncertainty set (Mazuelas et al., 2022, 2023). Specifically, for a given ℓ-loss function, the minimax classification rule h_ℓ is obtained by solving

$$\mathrm{V}_\ell = \min_{\mathrm{h}} \max_{\mathrm{p} \in \mathcal{U}} \mathbb{E}_{\mathrm{p}} \ell(\mathrm{h}, (x, y)), \tag{7}$$

where V_ℓ denotes the worst-case risk and $\mathcal{U}$ denotes an uncertainty set of distributions. The minimax classification rule h_ℓ minimizes the expected loss with respect to a worst-case distribution in $\mathcal{U}$. When the true underlying distribution p^* is in the uncertainty set, the worst-case risk upper bounds the classification risk of the minimax rule h_ℓ under the ℓ-loss, that is,

$$\mathcal{R}_\ell(\mathrm{h}_\ell) \leq \mathrm{V}_\ell. \tag{8}$$

The uncertainty set $\mathcal{U}$ in (7) is defined as

$$\mathcal{U} = \{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : |\mathbb{E}_{\mathrm{p}}\{\Phi(x, y)\} - \boldsymbol{\tau}| \preceq \boldsymbol{\lambda} \text{ and } \mathrm{p}(x) = \mathrm{p}^*(x)\}, \tag{9}$$

with $\mathrm{p}^*(x)$ denoting the underlying distribution over the instances. The vector $\boldsymbol{\tau}$ denotes the expectation estimates associated with the feature mapping Φ, and $\boldsymbol{\lambda} \succeq \mathbf{0}$ is a confidence vector that accounts for inaccuracies in these estimates. The mean and confidence vectors can be obtained from the training samples $\{(x_i, y_i)\}_{i=1}^n$ as

$$\boldsymbol{\tau} = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i, y_i), \ \boldsymbol{\lambda} = \lambda_0 \mathbf{s}, \tag{10}$$

where $\mathbf{s}$ denotes the vector formed by the component-wise sample standard deviations of Φ in the training samples and λ_0 is a regularization parameter. In particular, taking $\lambda_0 = \sqrt{(\log m + \log \frac{2}{\delta})/2}$, the underlying distribution p^* of the training samples lies in the

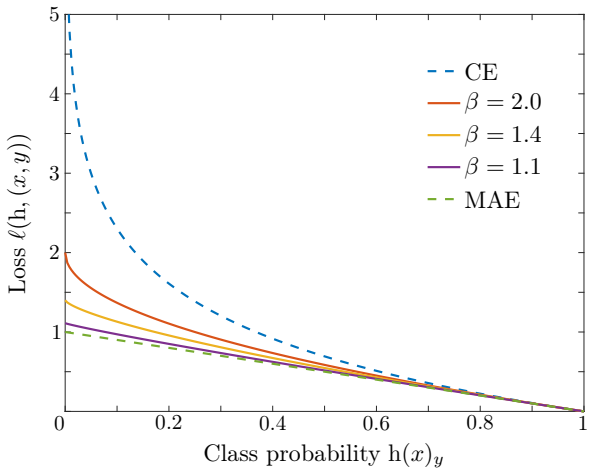


Figure 1: Relation between β and the resulting loss function. For $\beta = 1$, the loss corresponds to the MAE while for $\beta = \infty$, it corresponds to CE. For $\beta \in (1, \infty)$, the loss interpolates between the MAE and CE.

uncertainty set with probability $1-\delta$ (see e.g., Theorem 4 in Mazuelas et al. (2023)).

Previous works develop minimax methods as in (7) for the MAE and CE losses (Mazuelas et al., 2022, 2023). In addition, Mazuelas et al. (2022) explores the usage of α-losses under a minimax formulation that leads to an optimization that cannot be efficiently addressed in large-scale scenarios (not amenable for stochastic gradient descent (SGD) methods).

To address the limitations of the existing methods, we introduce a minimax formulation under the α-loss that results in a convex optimization over the classification margins, and effectively scales with complex datasets.

3 MINIMAX GENERALIZED CROSS-ENTROPY

In this section, we present the MGCE formulation based on the minimax framework in (7) and the α-loss in (4). In addition, we provide theoretical results that establish performance guarantees as well as an interpretation in terms of the worst-case distribution in $\mathcal{U}$.

The MGCE formulation minimizes the optimization problem in (7) over α-losses that results in the following bilevel convex optimization problem (see Appendix A for the detailed proof)

$$\mathrm{V}_\beta = \min_{\boldsymbol{\mu}\in\mathbb{R}^m} -\boldsymbol{\tau}^\top\boldsymbol{\mu} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}| - \mathbb{E}_{\mathrm{p}^*(x)}\{\varphi_\beta(x,\boldsymbol{\mu})\}, \tag{11}$$

where the function $\varphi_\beta(x,\boldsymbol{\mu})$ is obtained as

$$\begin{aligned}\varphi_\beta(x,\boldsymbol{\mu}) = \quad & \max_{\nu} \ \nu \\ & \text{s.t.} \sum_{y\in\mathcal{Y}} \Big(\tfrac{f(x,\boldsymbol{\mu})_y+\nu}{\beta}+1\Big)_+^\beta \leq 1.\end{aligned} \tag{12}$$

Since the function defining the constraint is monotonically increasing in ν, the function φ_β is implicitly defined over the parameters $\boldsymbol{\mu}$ as

$$F(x,\boldsymbol{\mu},\varphi_\beta(x,\boldsymbol{\mu})) = 1, \tag{13}$$

where $F(x,\boldsymbol{\mu},\varphi_\beta(x,\boldsymbol{\mu})) = \sum_{y\in\mathcal{Y}} \Big(\tfrac{f(x,\boldsymbol{\mu})_y+\varphi_\beta(x,\boldsymbol{\mu})}{\beta}+1\Big)_+^\beta$.

Note that the optimization in (11) is convex over $\boldsymbol{\mu}$ for $\beta\in[1,\infty)$ since the function $\varphi_\beta(x,\boldsymbol{\mu})$ is concave in $\boldsymbol{\mu}$. Specifically, if $\varphi_\beta(x,\boldsymbol{\mu}_1)=\nu_1$ and $\varphi_\beta(x,\boldsymbol{\mu}_2)=\nu_2$ we have that for any $t\in[0,1]$, $t\nu_1+(1-t)\nu_2$ is feasible for (12) taking $\boldsymbol{\mu}=t\boldsymbol{\mu}_1+(1-t)\boldsymbol{\mu}_2$, because the constraint in (12) is convex over $\boldsymbol{\mu}$ ($f(x,\boldsymbol{\mu})$ in (1) is linear and $\beta >= 1$). Then, the concavity of function φ_β follows because for any $t\in[0,1]$, we have

$$\begin{aligned}\varphi_\beta(x,t\boldsymbol{\mu}_1+(1-t)\boldsymbol{\mu}_2) &\geq t\nu_1+(1-t)\nu_2 \\ &\geq t\varphi_\beta(x,\boldsymbol{\mu}_1)+(1-t)\varphi_\beta(x,\boldsymbol{\mu}_2)\end{aligned}$$

as the maximum in the definition of $\varphi_\beta(x,t\boldsymbol{\mu}_1+(1-t)\boldsymbol{\mu}_2)$ is not lower than the value of any feasible solution.

The convex optimization problem in (11) is a bilevel optimization problem due to the definition of function $\varphi_\beta(x,\boldsymbol{\mu})$. For $\beta\in(1,\infty)$, a closed form expression of the function cannot be attained while for the special cases $\beta=1$ and $\beta=\infty$ is given by Mazuelas et al. (2022). However, in Section 4 we show that efficient learning can be achieved using stochastic gradients obtained by implicit differentiation.

The resulting classification rule h_β corresponding to the MGCE method is given by the solution $\boldsymbol{\mu}^*$ of (11). Specifically, for each instance $x\in\mathcal{X}$ such rule assigns the probability corresponding to label $y\in\mathcal{Y}$ as

$$\mathrm{h}_\beta(x)_y = \Big(\frac{f(x,\boldsymbol{\mu}^*)_y+\varphi_\beta(x,\boldsymbol{\mu}^*)}{\beta}+1\Big)_+^\beta, \tag{14}$$

where such an expression provides valid probabilities due to the definition of φ_β in (13).

The proposed MGCE formulation can also be viewed as obtaining classification rule h_β from the classifier margins using the link function in (14). This link is tailored to the loss parameter β, whereas in existing methods the softmax function is used as a fixed link, independent of β. For $\beta\to\infty$, the link function in (14) also reduces to the softmax function (see Appendix B), and in this case the loss coincides with CE.

The minimax formulation described above enables several theoretical guarantees. In the following, we establish a relation between the minimax probabilities and the worst-case distribution within the uncertainty set. Furthermore, we show that the proposed formulation not only bounds the classification risk associated with α-losses, but also provides an upper bound on the classification error of the MGCE classification rule.

3.1 Relation between worst-case distributions and minimax classifier

In this section, we characterize the worst-case distributions corresponding with the proposed MGCE formulation. The following theorem presents the general relation between the worst-case distribution and the minimax probabilities for any loss parameter β.

Theorem 1. *Given a loss function ℓ_β, if h_β is the minimax classifier in* (14)*, the worst-case distribution $\mathrm{p}_\beta\in\arg\max_{\mathrm{p}\in\mathcal{U}} \mathbb{E}_{\mathrm{p}}\ell_\beta(\mathrm{h}_\beta,(x,y))$ is given by*

$$\mathrm{p}_\beta(y|x) = \frac{\mathrm{h}_\beta(x)_y^{\frac{\beta-1}{\beta}}}{\sum_{y=1}^k \mathrm{h}_\beta(x)_y^{\frac{\beta-1}{\beta}}}, \quad \forall x\in\mathcal{X},\ y\in\mathcal{Y}. \tag{15}$$

Reciprocally, if p_β *is the worst-case distribution corresponding to the minimax problem in* (7)*, that is,* $\mathrm{p}_\beta \in \arg\max_{\mathrm{p}\in\mathcal{U}} \min_{\mathrm{h}} \mathbb{E}_{\mathrm{p}} \ell_\beta(\mathrm{h}, (x, y))$*, the minimax classifier in* (14) *satisfies* $\mathrm{h}_\beta \in \arg\min_{\mathrm{h}} \mathbb{E}_{\mathrm{p}_\beta} \ell_\beta(\mathrm{h}, (x, y))$ *and is given by*

$$\mathrm{h}_\beta(x)_y = \frac{\mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}}{\sum_{y=1}^{k} \mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}}. \tag{16}$$

Proof. See Appendix C. □

Similar relationships between probabilities and classification rules have been observed in other works for composite losses (Bao and Charoenphakdee, 2025). Theorem 1 characterizes the relation between the worst-case distribution p_β and the classifier probabilities h_β corresponding with different values of β. In particular, the worst-case distribution for a given solution $\boldsymbol{\mu}^*$ corresponding with the minimax formulation (11) is obtained as

$$\mathrm{p}_\beta(y|x) = \frac{\Big(\Big(f(x, \boldsymbol{\mu}^*)_y + \varphi_\beta(x, \boldsymbol{\mu}^*)\Big)/\beta + 1\Big)_+^{\beta-1}}{\sum_{j=1}^{k} \Big(\Big(f(x, \boldsymbol{\mu}^*)_j + \varphi_\beta(x, \boldsymbol{\mu}^*)\Big)/\beta + 1\Big)_+^{\beta-1}}. \tag{17}$$

The results in Theorem 1 show how the relationship between worst-case distribution and classification rule changes with the values of β. Specifically, for $\beta \in (1, \infty)$, the worst-case probability satisfies $\mathrm{p}_\beta(x)_y < \mathrm{h}_\beta(x)_y$ for the high confidence classes y, that is, those for which $\mathrm{h}_\beta(x)_y > \frac{1}{|\mathcal{Y}|}$. On the other hand, the worst-case probability satisfies $\mathrm{p}_\beta(x)_y \geq \mathrm{h}_\beta(x)_y$ for the low confidence classes y, that is, those for which $\mathrm{h}_\beta(x)_y \leq \frac{1}{|\mathcal{Y}|}$ (see also Figure 2 corresponding with 2 classes). This behavior ensures that the minimax formulation penalizes overconfident predictions while preserving uncertainty, thereby performing a form of distributional smoothing on the classifier's output. For the extreme cases MAE ($\beta = 1$) and CE ($\beta = \infty$), the worst-case distribution shows a different behavior. In case of MAE, the worst-case distribution uniformly weighs all classes for which the classifier's probability is non-zero, irrespectively of its value (an extremely cautious behavior). However, when $\mathrm{h}_\beta(x)_y = 0$ or $\mathrm{h}_\beta(x)_y = 1$, the worst-case distribution also agrees with the classifier. On the other hand, in case of CE, the worst-case distribution coincides with the classification probabilities, that is, $\mathrm{h}_{\mathrm{CE}}(x)_y = \mathrm{p}_{\mathrm{CE}}(x)_y$.

The relationship established above between the classifier probabilities and the worst-case distribution elucidates the optimization dynamics of the proposed MGCE methods. In Section 4 we will show that the stochastic gradients used at learning are given by the worst-case distribution.

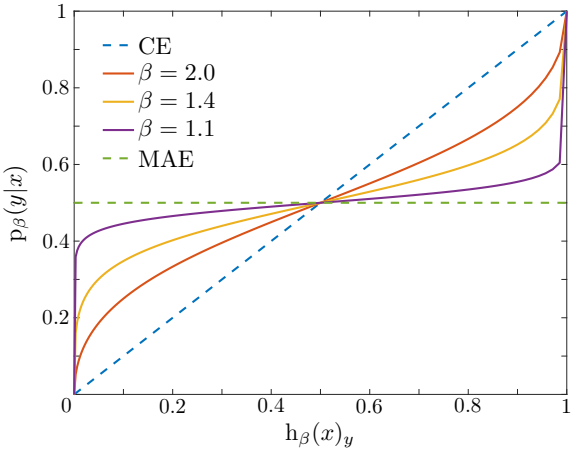


Figure 2: Relation between the minimax classifier $\mathrm{h}_\beta(x)_y$ and the worst-case probability $\mathrm{p}_\beta(y|x)$ corresponding with 2 classes. For $\beta \in (1, \infty)$, the worst-case probabilities take a cautious stance, avoiding the extremes of MAE ($\beta = 1$) and CE ($\beta = \infty$) losses.

The MGCE formulation can also provide an upper bound on the MAE classification risk of the classifier as shown in the following section.

3.2 Performance guarantees

The proposed MGCE formulation can serve as a surrogate for the MAE loss. In the following, we characterize the closeness of the MGCE to the MAE loss in terms of the loss parameter β. In addition, we show that our proposed formulation can provide an upper bound on the classification error.

Theorem 2. *The probabilistic* α*-losses in* (4) *satisfy*

$$\frac{1}{\beta}\ell_\beta(\mathrm{h}, (x, y)) \leq \ell_{\mathrm{MAE}}(\mathrm{h}, (x, y)) \leq \ell_\beta(\mathrm{h}, (x, y)). \tag{18}$$

Moreover, for $\beta \in (1, \infty)$*, we have*

$$\ell_\beta(\mathrm{h}, (x, y)) - \ell_{\mathrm{MAE}}(\mathrm{h}, (x, y)) \leq \beta - 1. \tag{19}$$

Proof. See Appendix D. □

Theorem 2 shows that the α-losses can provide upper bounds for MAE loss. Using this relation, we also obtain performance guarantees in terms of an upper bound on the MAE classification risk $\mathcal{R}_{\mathrm{MAE}}(\mathrm{h})$ for the minimax classifier h_β derived from the MGCE formulation.

Remark 1. *As a consequence of inequality* (18)*,* $\mathcal{R}_{\mathrm{MAE}}(\mathrm{h}_\beta) \leq \mathcal{R}_\beta(\mathrm{h}_\beta)$ *for a minimax classification rule* h_β *corresponding with* (11)*. Moreover, using inequality* (8)*, we have that*

$$\mathcal{R}_{\mathrm{MAE}}(\mathrm{h}_\beta) \leq \mathrm{V}_\beta \tag{20}$$

in cases where the underlying distribution p^* *lies in the uncertainty set* $\mathcal{U}$ *in* (9).

The remark above shows that the optimal value V_β obtained by the proposed MGCE formulation in (11) can provide an upper bound on the classification error. Notice that the error probability of a classification rule coincides with the MAE risk for randomized rules and is at most twice the MAE risk for deterministic rules (Mazuelas et al., 2023). Moreover, inequality (19) in Theorem 2 shows that the upper bound in (20) is tighter as β decreases.

As described above, the proposed MGCE provides performance guarantees together with a convex optimization over margins. However, the bilevel optimization in (8) prevents the usage of conventional methods for large-scale learning. In the following, we present an efficient learning algorithm for MGCE methods based on implicit differentiation.

4 OPTIMIZATION

In this section, we detail the training procedure for the convex optimization problem of MGCE defined in (11). Such optimization is amenable to stochastic gradient descent (SGD) as the objective is given in terms of expectations. The main technical difficulty lies in the fact that the function $\varphi_\beta(x, \boldsymbol{\mu})$ in (12) is only implicitly defined as in (13). In the following, we provide an efficient algorithm to compute the stochastic gradients in (22) using implicit differentiation (Dontchev and Rockafellar, 2009; Agrawal et al., 2019).

4.1 Stochastic gradients

A stochastic gradient $g_\beta(x, y)$ of the objective in (11) for training sample (x, y) is given by

$$g_\beta(x, y) = \boldsymbol{\lambda} \odot \operatorname{sign}(\boldsymbol{\mu}) - \Phi(x, y) - \frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}}, \quad (21)$$

where $\odot$ denotes the Hadamard product, $\operatorname{sign}(\boldsymbol{\mu})$ denotes the vector given by the signs of the components of vector $\boldsymbol{\mu}$. Such gradient is directly derived from the expression $\boldsymbol{\tau}$ in (10), but requires to evaluate the derivative of function $\varphi_\beta(x, \boldsymbol{\mu})$ that does not have a closed-form expression.

The following theorem derives the gradient of function $\varphi_\beta(x, \boldsymbol{\mu})$ using implicit differentiation. Interestingly, the gradient is determined by the worst-case distribution of the classifier corresponding with $\boldsymbol{\mu}$.

Theorem 3. *The stochastic gradient of the function* $\varphi(x, \boldsymbol{\mu})$ *corresponding with sample* x *is given as*

$$\frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = -\sum_{y=1}^{k} \mathrm{p}_\beta(y|x) \Phi(x, y), \quad (22)$$

where $\mathrm{p}_\beta(y|x)$ *is the worst-case distribution corresponding to the parameters* $\boldsymbol{\mu}$ *and sample* x *given in* (17).

Proof. See Appendix E. □

Theorem 3 shows that the gradient used by the proposed MGCE formulation varies as a function of the loss parameter β. In particular, the gradient of φ_β corresponds to an expectation with respect to the worst-case distribution p_β.

The results above show that the proposed MGCE formulation is amenable to SGD, similarly as methods based on margin losses. Indeed, the proposed approach corresponds with the margin loss defined as

$$\ell_\beta(\boldsymbol{\mu}, (x, y)) = -f(x, \boldsymbol{\mu})_y - \varphi_\beta(x, \boldsymbol{\mu}), \quad (23)$$

which is convex on $\boldsymbol{\mu}$. Furthermore, the following corollary shows that such MGCE margin loss is classification-calibrated.

Corollary 1. *Let* ℓ_β *be the margin loss defined in* (23). *For any* x *and* $\beta > 1$, *if*

$$\boldsymbol{\mu}^* \in \arg\min_{\boldsymbol{\mu}} \sum_y \mathrm{p}^*(y|x) \ell_\beta(\boldsymbol{\mu}, (x, y)) \quad (24)$$

then

$$\arg\max_y f(x, \boldsymbol{\mu}^*)_y = \arg\max_y \mathrm{p}^*(y|x). \quad (25)$$

Proof. See Appendix F. □

Beyond classification-calibration, the composite loss corresponding with MGCE leads to a proper loss since MGCE losses are "calm" (Bao and Charoenphakdee, 2025). Specifically, a composite loss for probabilities $\tilde{\ell}_\beta(\mathrm{p}, (x, y))$ can be defined using the margin loss in (23) as $\tilde{\ell}_\beta(\mathrm{p}, (x, y)) = \ell_\beta(\boldsymbol{\mu}_\mathrm{p}, (x, y))$, where $\boldsymbol{\mu}_\mathrm{p}$ is the parameter vector associated with probability p via (17). Then, an argument similar to that in the proof of Corollary 1 shows that the loss $\tilde{\ell}_\beta$ is proper for any $\beta > 1$.

We can establish a connection between the classifier outputs and the gradients obtained during the SGD iterations for $\beta \in [1, \infty)$ using the relation between the worst-case distribution and the classifier outputs in Section 3.1. Relative to the MAE loss ($\beta = 1$), the gradient of MGCE associated with larger values of β places more emphasis on the high confidence classes, i.e., the labels with classifier probability exceeding $\frac{1}{|\mathcal{Y}|}$ at the current iteration. Conversely, relative to the CE loss ($\beta = \infty$), the gradient of MGCE for smaller values of β places less emphasis on high confidence

classes, while allowing for more uncertainty by emphasizing the low confidence classes. As a result, for clean samples, larger β values accelerate learning from confident and correctly classified examples. In contrast, for noisy samples, smaller β values mitigate the influence of confident but potentially mislabeled examples.

4.2 Effective bisection method

The worst-case distribution $\mathrm{p}_\beta(y|x)$ is defined in terms of the implicit function $\varphi_\beta(x, \boldsymbol{\mu})$, so that, the gradient in (22) requires evaluation of $\varphi_\beta(x, \boldsymbol{\mu})$. For a given $\boldsymbol{\mu}$ and x, this value can be computed by solving the equation $F(x, \boldsymbol{\mu}, \varphi_\beta^i) = 1$ in (13) for φ_β^i using the bisection method as detailed below.

A unique root for equation (13) exists since $F(x, \boldsymbol{\mu}, \varphi_\beta^i)$ is continuous and monotonically increasing in φ_β^i. In addition, such a root is included in the interval $[\varphi_\beta^1, \varphi_\beta^2]$ given by

$$\begin{aligned} \varphi_\beta^1 &= C_\beta - \max_{i \in \mathcal{Y}} f(x, \boldsymbol{\mu})_i, \\ \varphi_\beta^2 &= C_\beta - \min_{i \in \mathcal{Y}} f(x, \boldsymbol{\mu})_i, \end{aligned} \tag{26}$$

where $C_\beta = \beta(\frac{1}{k^{1/\beta}} - 1)$. This interval contains a root because $F(x, \boldsymbol{\mu}, \varphi_\beta^1) \leq 1$ and $F(x, \boldsymbol{\mu}, \varphi_\beta^2) \geq 1$. Specifically, these inequalities follow because for each $i \in \mathcal{Y}$, we have

$$\left(\frac{f(x,\boldsymbol{\mu})_i + \varphi_\beta^1}{\beta} + 1\right)^\beta \leq \frac{1}{|\mathcal{Y}|}$$

and

$$\left(\frac{f(x,\boldsymbol{\mu})_i + \varphi_\beta^2}{\beta} + 1\right)^\beta \geq \frac{1}{|\mathcal{Y}|}.$$

Therefore, $\varphi_\beta(x, \boldsymbol{\mu})$ can be efficiently evaluated by applying the bisection method over such an interval.

Algorithm 1 details the gradient computation step for SGD, which has a computational complexity of $O\Big(\log\Big((\varphi_\beta^2 - \varphi_\beta^1)/\epsilon\Big)\Big)$ for ϵ level of precision in the bisection routine. In the following, we evaluate the performance of the proposed MGCE formulation using the efficient learning Algorithm 1 on DNNs.

Algorithm 1 Implicit gradient for MGCE via bisection.

Input: Instance x, model parameters $\boldsymbol{\mu}$, loss parameter β, tolerance ϵ
Output: Gradient $\frac{\partial \varphi_\beta(x,\boldsymbol{\mu})}{\partial \boldsymbol{\mu}}$

Compute φ_β^1 and φ_β^2 as in (26)
while $|\varphi_\beta^1 - \varphi_\beta^2| \geq \epsilon$ **do**
  $\varphi_\beta^i \leftarrow (\varphi_\beta^1 + \varphi_\beta^2)/2$
  if $F(\boldsymbol{x}, \boldsymbol{\mu}, \varphi_\beta^i) > 1$ **then**
    $\varphi_\beta^2 \leftarrow \varphi_\beta^i$
  else
    $\varphi_\beta^1 \leftarrow \varphi_\beta^i$
  end if
end while
$\varphi_\beta(x, \boldsymbol{\mu}) \leftarrow \varphi_\beta^i$
Compute $\mathrm{p}_\beta(y|x)$ using $\varphi_\beta(x, \boldsymbol{\mu})$ as in (17)
$\frac{\partial \varphi_\beta(x,\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \leftarrow -\sum_{y=1}^k \mathrm{p}_\beta(y|x)\Phi(x, y)$

5 EXPERIMENTAL RESULTS

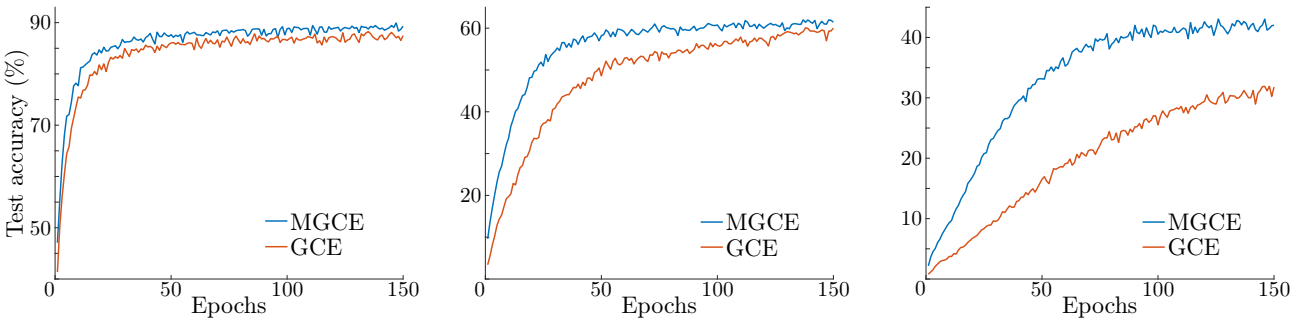
In this section, we present numerical results evaluating the performance of the proposed MGCE formulation in terms of test accuracy, iteration complexity, and calibration error using DNNs corresponding with ResNet architectures (He et al., 2016). The experimental results are carried out using the datasets FashionMNIST, CIFAR-10, SVHN, CIFAR-100, and Tiny ImageNet that are commonly used as benchmarks. In addition, we also assess the performance on the large benchmark dataset WebVision affected by real-world label noise (Li et al., 2017). For CIFAR-10 and CIFAR-100, we employ the standard ResNet-34 architecture, while for FashionMNIST, Tiny ImageNet, and SVHN, we use ResNet-18 architecture. For the WebVision dataset, we use the standard ResNet-50 architecture.

We compare the proposed MGCE method with the GCE method as presented in Zhang and Sabuncu (2018). Additionally, we include comparisons with the standard CE loss and the convex MAE loss based on the minimax approach proposed in Mazuelas et al. (2023). Additional results are presented in Appendix G. The implementation of the proposed MGCE formulation is available in the python library MRCpy (Bondugula et al., 2024).[1]

5.1 Experimental setup

The following setup applies to all the experiments conducted. As part of data preprocessing and augmentation procedure, we apply per-pixel mean subtraction, horizontal random flipping, and random cropping to 32×32 after padding the images with 4 pixels on each side. For training, we use SGD with a momentum of 0.9 and a fixed learning rate of 0.01. We choose the regularization parameter $\lambda_0 = 10^{-5}$ based on grid search among common values as detailed in Appendix G.1. Algorithm 1 with tolerance $\epsilon = 10^{-4}$ is used to compute the gradients for the proposed MGCE. For all settings, we clip the gradient norm to 5.0. We train the entire network for 150 epochs using mini-batch of size 128, and we use 10% of the entire training data for validation. For the experiments corresponding with noisy

[1] https://github.com/MachineLearningBCAM/MRCpy/tree/main/MRCpy/pytorch/mgce



(a) CIFAR10 **(b)** CIFAR100 **(c)** Tiny ImageNet

Figure 3: Average test accuracy under clean training data obtained for multiple complex datasets. The value of loss parameter β is set to 1.4. The figure shows the fast convergence of the proposed MGCE in comparison to the GCE.

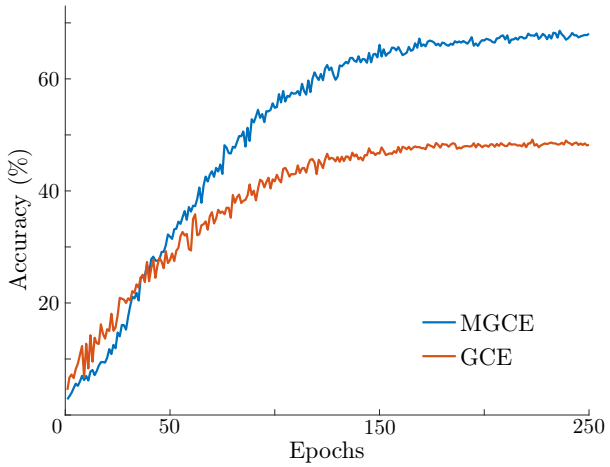


Figure 4: Top-1 validation accuracy on the real-world noisy dataset WebVision. The figure shows that the proposed MGCE outperforms GCE, which significantly underfits on this complex dataset due to its non-convexity.

settings, both training and validation set are contaminated with symmetric label noise, and all the experiments are conducted for five times with randomization.

5.2 Comparison between GCE and MGCE in terms of accuracy and convergence

We compare the proposed MGCE method with GCE using DNNs on multiple datasets CIFAR10 (10 classes), CIFAR100 (100 classes), and Tiny ImageNet (200 classes). Figure 3 shows the test accuracy across training epochs for $\beta = 1.4$, which is the default value suggested in Zhang and Sabuncu (2018) for the GCE method.

The figure illustrates that the proposed MGCE converges to a better accuracy faster than the GCE method. Specifically, the difference is significant for complex datasets with many classes. For instance, in CIFAR-100, the MGCE method achieves the same accuracy as GCE at epoch 50 for a significantly lower number of epochs.

5.3 Evaluation on real-world noisy label

We further assess the performance of the proposed MGCE formulation using the real-world noisy dataset WebVision (Li et al., 2017) that contains more than 2.4 million web images crawled from the internet. Here, we follow the "Mini" setting proposed in Jiang et al. (2018) that only takes the first 50 classes of the Google resized image subset as the training dataset. A ResNet-50 architecture is trained under different methods and evaluated on the corresponding clean WebVision validation set. We use the same experimental setup as in Ma et al. (2020); Zhou et al. (2021), which is detailed in the Appendix G.2.

Figure 4 reports the top-1 validation accuracy on the clean validation set. The results show that the proposed MGCE method significantly outperforms GCE using the same value $\beta = 1.4$. Specifically, results demonstrate that the GCE method may underfit with such complex dataset inherent with real-world noise.

5.4 Cross-validation over general values of β

We further evaluate the performance of MGCE and GCE across general values of the parameter β. In particular, we compare with the baselines provided by CE and the convex MAE loss introduced in Mazuelas et al. (2023). We assess the performance in terms of the test accuracy and model calibration under both clean and noisy label conditions. For the noisy setting, we consider symmetric noise with rate $\eta = 0.2$ and $\eta = 0.4$ introduced synthetically to both the training and validation sets. We consider static calibration error (SCE) that takes into account the probabilities corresponding with all the classes rather just the maximum prediction as defined in Nixon et al. (2019). The loss parameter β is selected via cross-validation over 8 values in the interval $(1.05, 11)$. This range is sufficient in practice, as values below 1.05 yield results similar to the MAE loss, while values above 11 yield results similar to CE.

The results in Table 1 show that the proposed MGCE

Dataset	Method	Test Accuracy (%)			SCE (%)		
		η=0.0	η=0.2	η=0.4	η=0.0	η=0.2	η=0.4
FashionMNIST	MGCE	**92.94 ± 0.17**	90.11 ± 0.38	88.65 ± 0.15	**05.20 ± 0.23**	**02.44 ± 0.28**	**02.54 ± 1.30**
	GCE	92.93 ± 0.34	**91.11 ± 0.22**	**89.74 ± 0.29**	05.52 ± 0.31	07.14 ± 0.22	08.63 ± 0.35
	CE	92.73 ± 0.32	89.62 ± 0.15	87.40 ± 0.13	05.33 ± 0.24	17.32 ± 0.85	33.52 ± 1.86
	MAE	91.84 ± 0.24	89.86 ± 0.26	88.44 ± 0.38	73.47 ± 0.57	74.88 ± 0.46	74.23 ± 0.28
CIFAR10	MGCE	**91.01 ± 0.43**	86.91 ± 0.36	83.49 ± 0.45	**05.53 ± 0.31**	**10.26 ± 0.37**	**12.15 ± 0.5**
	GCE	90.55 ± 0.25	**86.95 ± 0.43**	**83.80 ± 0.57**	06.14 ± 0.13	10.88 ± 0.48	12.95 ± 0.66
	CE	90.80 ± 0.27	83.73 ± 0.37	77.31 ± 0.93	05.71 ± 0.44	10.24 ± 1.86	25.31 ± 2.40
	MAE	89.31 ± 0.22	86.01 ± 0.46	82.64 ± 0.61	73.00 ± 0.24	71.54 ± 0.46	68.29 ± 0.59
SVHN	MGCE	**95.04 ± 0.21**	**94.03 ± 0.31**	**92.92 ± 0.80**	03.19 ± 0.33	**04.30 ± 0.24**	**03.96 ± 0.60**
	GCE	94.68 ± 0.24	93.43 ± 0.45	92.84 ± 0.26	**02.58 ± 0.27**	12.27 ± 1.75	05.27 ± 0.27
	CE	93.49 ± 1.23	93.15 ± 0.25	90.88 ± 0.94	04.22 ± 1.57	15.01 ± 1.07	32.61 ± 4.63
	MAE	92.89 ± 1.45	93.71 ± 0.41	92.32 ± 0.41	75.76 ± 2.83	79.15 ± 0.32	77.91 ± 0.36
CIFAR100	MGCE	64.35 ± 0.45	**58.88 ± 0.76**	**52.65 ± 0.85**	**22.22 ± 0.09**	24.50 ± 1.05	23.54 ± 0.65
	GCE	**64.82 ± 0.38**	56.66 ± 0.85	51.37 ± 0.68	22.89 ± 0.34	33.77 ± 0.65	37.16 ± 0.81
	CE	64.66 ± 0.39	52.51 ± 1.10	42.32 ± 0.78	22.19 ± 0.13	**04.30 ± 1.80**	**06.06 ± 2.24**
	MAE	62.17 ± 0.59	57.57 ± 0.49	41.99 ± 1.45	60.47 ± 0.59	56.15 ± 0.49	60.46 ± 0.59

Table 1: Average test accuracy and SCE for clean and noisy data. The ratio η represents the fraction of samples corrupted by symmetric label noise. We report accuracies of the epoch where validation accuracy is maximum, and the loss parameter β is selected using cross-validation. Our proposed MGCE method obtains strong accuracy together with improved calibration.

method attains strong accuracy while yielding improved calibration, particularly in the presence of label noise. Without noise, MGCE and GCE obtain similar accuracy as that with CE. On the other hand, in noisy scenarios, both methods demonstrate an increased robustness surpassing the MAE and CE baselines, especially on complex datasets such as CIFAR100. Beyond accuracy, MGCE results in an enhanced model calibration, especially in noisy scenarios. Although CE achieves better calibration on CIFAR-100, it does so at the expense of accuracy.

6 CONCLUSION

In this paper, we introduce a minimax formulation for generalized cross-entropy (MGCE). While existing formulations of generalized cross-entropy lead to non-convex optimization over the classifier margins, our formulation casts learning as a convex bilevel optimization problem. Specifically, such formulation minimizes the worst-case expected α-loss with respect to an uncertainty set of distributions. We show that the MGCE formulation can provide an upper bound on the classification error, and leads to classification-calibrated margin losses. Moreover, the paper establishes a relationship between the MGCE classifier and the worst-case distribution, which provides insights into the optimization dynamics.

Using implicit differentiation, we show that the stochastic gradients for MGCE are determined by worst-case distributions, and can be efficiently computed by means of a bisection method. Experiments on benchmark datasets with deep neural networks demonstrate that MGCE improves accuracy and convergence over existing methods while producing better calibrated models, especially under label noise. Overall, the presented results show that the methodology proposed can enable to generalize cross-entropy losses in a more effective manner than existing approaches.

Acknowledgements

Funding in direct support of this work has been provided by projects PID2022-137063NBI00, PLEC2024-011247, and CEX2021-001142-S funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR, and programs BERC-2022-2025 and ELKARTEK funded by the Basque Government. Anqi Liu is also supported by a grant from Open Philanthropy. Kartheek Bondugula also holds a predoctoral grant (EJ-GV 2022) from the Basque Government and developed part of this research during a research stay at Johns Hopkins University.

References

Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *In Advances in Neural Information Processing Systems*, 31, 2018.

Tyler Sypherd, Mario Diaz, John Kevin Cava, Gautam Dasarathy, Peter Kairouz, and Lalitha Sankar. A tunable loss function for robust classification: Calibration, landscape, and generalization. *IEEE Transactions on Information Theory*, 68:6027–6047, 2022a.

Davide Ferrari and Yuhong Yang. Maximum lq-likelihood estimation. *The Annals of Statistics*, 38 (2):753–783, 2010.

Jiachun Liao, Oliver Kosut, Lalitha Sankar, and Flavio P. Calmon. A tunable measure for information leakage. In *Proceedings of the IEEE International Symposium on Information Theory*, 2018.

Tyler Sypherd, Mario Diaz, Lalitha Sankar, and Gautam Dasarathy. On the α-loss landscape in the logistic model. In *IEEE International Symposium on Information Theory*. IEEE, 2020.

Tyler Sypherd, Richard Nock, and Lalitha Sankar. Being properly improper. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, 2022b.

Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 2021.

Max Staats, Matthias Thamm, and Bernd Rosenow. Enhancing noise-robust losses for large-scale noisy data learning. In *AAAI Conference on Artificial Intelligence*, volume 39, 2025.

Kaiser Asif, Wei Xing, Sima Behpour, and Brian D. Ziebart. Adversarial cost-sensitive classification. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, volume 31, 2015.

Rizal Fathony, Anqi Liu, Kaiser Asif, and Brian Ziebart. Adversarial multiclass classification: A risk minimization perspective. *In Advances in Neural Information Processing Systems*, 29, 2016.

Rizal Fathony, Kaiser Asif, Anqi Liu, Mohammad Ali Bashiri, Wei Xing, Sima Behpour, Xinhua Zhang, and Brian D. Ziebart. Consistent robust adversarial prediction for general multiclass classification. *arXiv*, 2018.

Santiago Mazuelas, Mauricio Romero, and Peter Grunwald. Minimax risk classifiers with 0-1 loss. *Journal of Machine Learning Research*, 24, 2023.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–85, 2004.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101, 2006.

Arpit Agarwal and Shivani Agarwal. On consistent surrogate risk minimization and property elicitation. In *Conference on Learning Theory*, volume 40, 2015.

Santiago Mazuelas, Yuan Shen, and Aritz Pérez. Generalized maximum entropy for supervised classification. *IEEE Transactions on Information Theory*, 68, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 2017.

Santiago Mazuelas, Andrea Zanoni, and Aritz Perez. Minimax classification with 0-1 loss and performance guarantees. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Han Bao and Nontawat Charoenphakdee. Calm composite losses: Being improper yet proper composite. In *The 28th International Conference on Artificial Intelligence and Statistics*, volume 258, 2025.

Asen L Dontchev and R Tyrrell Rockafellar. *Implicit Functions and Solution Mappings*, volume 543. Springer, 2009.

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *In Advances in Neural Information Processing Systems*, 32, 2019.

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*, 2016.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data, 2017.

Kartheek Bondugula, Verónica Álvarez, Jose I. Segovia-Martín, Aritz Pérez, and Santiago Mazuelas. MRCpy: A library for minimax risk classifiers. *arXiv*, 2024.

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 *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 2020.

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Dongmin Park, Seola Choi, Doyoung Kim, Hwanjun Song, and Jae-Gil Lee. Robust data pruning under label noise via maximizing re-labeling accuracy. *In Advances in Neural Information Processing Systems*, 36, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A DERIVATIONS FOR THE CONVEX FORMULATION OF MGCE IN (11)

The minimax risk optimization problem (7) corresponding with the α-loss is given as follows

$$\begin{aligned}\mathrm{V}_\beta = \min_{\mathrm{h}} \max_{\mathrm{p}\in\mathcal{U}} \mathbb{E}_{\mathrm{p}}\ell_\beta(\mathrm{h},(x,y)) = \min_{\mathbf{h}} \max_{\mathbf{p}} \quad & \beta\big(1-(\mathbf{h}^{1/\beta})^\top\mathbf{p}\big) - I_+(\mathbf{p}) \\ \text{s.t.} \quad & \textstyle\sum_{y\in\mathcal{Y}} \mathrm{p}(x,y) = \mathrm{p}^*(x)\ \forall\ x\in\mathcal{X}, \\ & \boldsymbol{\tau}-\boldsymbol{\lambda} \preceq \boldsymbol{\Phi}^\top\mathbf{p} \preceq \boldsymbol{\tau}+\boldsymbol{\lambda},\end{aligned}$$

where each row of the vectors $\mathbf{h}\in\mathbb{R}^{|\mathcal{X}||\mathcal{Y}|}$ and $\mathbf{p}\in\mathbb{R}^{|\mathcal{X}||\mathcal{Y}|}$ represents a classification probability $\mathrm{h}(x)_y$ and a probability $\mathrm{p}(x,y)$ for distribution $\mathrm{p}\in\mathcal{U}$, corresponding with a sample $x\in\mathcal{X}$ and label $y\in\mathcal{Y}$. Each row of the matrix $\boldsymbol{\Phi}\in\mathbb{R}^{|\mathcal{X}||\mathcal{Y}|\times m}$ represents the feature mapping vector $\Phi(x,y)\in\mathbb{R}^m$. Moreover,

$$I_+(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p}\succeq 0 \\ \infty & \text{otherwise.}\end{cases}$$

Taking the Lagrange dual of the maximization over $\mathbf{p}$, we obtain

$$\begin{aligned}\min_{\mathbf{h},\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\boldsymbol{\nu}} \quad & \beta - (\boldsymbol{\tau}-\boldsymbol{\lambda})^\top\boldsymbol{\mu}_1 + (\boldsymbol{\tau}+\boldsymbol{\lambda})^\top\boldsymbol{\mu}_2 - \mathbb{E}_{\mathrm{p}^*}\nu_x + f^*\big(\boldsymbol{\Phi}^\top(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)+\bar{\boldsymbol{\nu}}\big) \\ s.t. \quad & \boldsymbol{\mu}_1,\boldsymbol{\mu}_2 \succeq 0,\end{aligned}$$

where each row of the vector $\boldsymbol{\nu}\in\mathbb{R}^{|\mathcal{X}|}$ corresponds to an $x\in\mathcal{X}$ denoted by ν_x, while each row of the vector $\bar{\boldsymbol{\nu}}\in\mathbb{R}^{|\mathcal{X}||\mathcal{Y}|}$ corresponds to a pair (x,y) such that the value for an x and any $y\in\mathcal{Y}$ is equal to ν_x. The function f^* is the conjugate of $f(\mathbf{p}) = \beta(\mathbf{h}^{1/\beta})^\top\mathbf{p} + I_+(\mathbf{p})$ given by

$$f^*(\mathbf{w}) = \sup_{\mathbf{p}\succeq 0}\big(\mathbf{w}-\beta\mathbf{h}^{1/\beta}\big)^\top\mathbf{p} = \begin{cases} 0 & \text{if } \mathbf{w}\preceq\beta\mathbf{h}^{1/\beta} \\ \infty & \text{otherwise.}\end{cases}$$

Then, taking $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ and simplifying, we have

$$\begin{aligned}\min_{\mathbf{h},\boldsymbol{\mu},\boldsymbol{\nu}} \quad & -\boldsymbol{\tau}^\top\boldsymbol{\mu} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}| - \mathbb{E}_{\mathrm{p}^*}\nu_x \\ \text{s.t.} \quad & \Phi(x,y)^\top\boldsymbol{\mu} + \nu_x + \beta \le \beta\mathrm{h}(x)_y^{1/\beta}\ \ \forall y\in\mathcal{Y}, x\in\mathcal{X}.\end{aligned}$$

Since $\mathrm{h}(x)_y$ represents a conditional probability of $y\in\mathcal{Y}$ for a given $x\in\mathcal{X}$, we have $\sum_y \mathrm{h}(x)_y = 1$. By incorporating this condition into the constraints of the previous minimization problem, we obtain

$$\begin{aligned}\min_{\boldsymbol{\mu},\boldsymbol{\nu}} \quad & -\boldsymbol{\tau}^\top\boldsymbol{\mu} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}| - \mathbb{E}_{\mathrm{p}^*}\nu_x \\ \text{s.t.} \quad & \textstyle\sum_{y\in\mathcal{Y}}\left(\frac{\Phi(x,y)^\top\boldsymbol{\mu}+\nu_x}{\beta}+1\right)_+^\beta \le 1, \quad x\in\mathcal{X}.\end{aligned}$$

Because the variable ν_x is implicitly defined by $\boldsymbol{\mu}$, the optimization becomes equivalent to

$$\min_{\boldsymbol{\mu}} \ -\boldsymbol{\tau}^\top\boldsymbol{\mu} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}| - \mathbb{E}_{\mathrm{p}^*}\{\varphi_\beta(x,\boldsymbol{\mu})\},$$

where

$$\begin{aligned}\varphi_\beta(x,\boldsymbol{\mu}) = \max_{\nu} \quad & \nu \\ \text{s.t.} \quad & \sum_{y\in\mathcal{Y}}\left(\tfrac{f(x,\boldsymbol{\mu})_y+\nu}{\beta}+1\right)_+^\beta \le 1.\end{aligned}$$

This corresponds to the optimization problem defined by (11) in the main paper.

B CONVERGENCE OF THE LINK FUNCTION IN (14) TO SOFTMAX FUNCTION FOR $\beta \to \infty$

We begin by applying the limit $\lim_{\beta\to\infty}(1+\frac{a}{\beta})^\beta = e^a$ to the link function in (14), which gives

$$\lim_{\beta\to\infty} \mathrm{h}_\beta(x)_y = \lim_{\beta\to\infty} \Big(\frac{f(x,\boldsymbol{\mu}^*)_y + \varphi_\beta(x,\boldsymbol{\mu}^*)}{\beta} + 1\Big)_+^\beta = \mathrm{e}^{f(x,\boldsymbol{\mu}^*)_y + \varphi_\infty(x,\boldsymbol{\mu}^*)}. \tag{27}$$

Using the result in (27), taking the the limit $\beta \to \infty$ of the implicit function defined in (13), we obtain

$$\sum_{y\in\mathcal{Y}} \mathrm{e}^{f(x,\boldsymbol{\mu}^*)_y + \varphi_\infty(x,\boldsymbol{\mu}^*)} = 1.$$

Solving for φ_∞ gives the log-sum-exp normalization

$$\varphi_\infty(x,\boldsymbol{\mu}^*) = -\log\{\sum_{y\in\mathcal{Y}} \mathrm{e}^{f(x,\boldsymbol{\mu}^*)_y}\}. \tag{28}$$

Finally, substituting (28) into (27), we obtain

$$\lim_{\beta\to\infty} \mathrm{h}_\beta(x)_y = \frac{\mathrm{e}^{f(x,\boldsymbol{\mu}^*)_y}}{\sum_{y\in\mathcal{Y}} \mathrm{e}^{f(x,\boldsymbol{\mu}^*)_y}}.$$

This corresponds to the softmax function over the classification margins $f(x,\boldsymbol{\mu}^*)$.

C PROOF OF THEOREM 1

We first prove the relation (16) and then establish (15) using it. For $\beta \geq 1$, and given the worst-case distribution $\mathrm{p}_\beta \in \arg\max_{\mathrm{p}\in\mathcal{U}} \min_{\mathrm{h}} \ell_\beta(\mathrm{h},\mathrm{p})$, the minimax classifier h_β is defined as a solution to the following optimization problem

$$\min_{\mathrm{h}} \ell_\beta(\mathrm{h},\mathrm{p}_\beta) = \min_{\mathrm{h}} \sum_{x,y} \mathrm{p}_\beta(x,y)\beta(1-\mathrm{h}(x)_y^{\frac{1}{\beta}}) = \beta + \beta\sum_x \mathrm{p}_\beta(x) \min_{\mathrm{h}} \sum_y \{-\mathrm{p}_\beta(y|x)\mathrm{h}(x)_y^{\frac{1}{\beta}}\}.$$

Therefore, this minimization problem can be solved independently for each instance $x \in \mathcal{X}$ as

$$\begin{aligned} \min_{\mathbf{h}} \quad & -\mathbf{p}_\beta^\top \mathbf{h}^{\frac{1}{\beta}} \\ \text{s.t.} \quad & \mathbf{h} \succeq 0,\ \mathbf{1}^\top\mathbf{h} = 1, \end{aligned} \tag{29}$$

where each row of the vectors $\mathbf{h} \in \mathbb{R}^k$ and $\mathbf{p}_\beta \in \mathbb{R}^k$ represents $\mathrm{h}(x)_y$ and $\mathrm{p}_\beta(y|x)$, respectively, and $\mathbf{1}$ denotes a vector of ones.

The optimization problem in (29) is convex and we can utilize the Karush-Kuhn-Tucker (KKT) optimality conditions to solve it. The KKT conditions corresponding with the primal solution $\mathbf{h}_\beta$ and the dual solution $\boldsymbol{\lambda}^*, \nu^*$ are given by

$$\begin{aligned} \mathbf{h}_\beta \succeq 0,\ \mathbf{1}^\top\mathbf{h}_\beta = 1,\ \boldsymbol{\lambda}^* \succeq 0 \qquad & \text{(feasibility)}, \\ \lambda_i^* \mathrm{h}_{\beta_i} = 0 \qquad & \text{(complementary slackness)}, \\ -\tfrac{1}{\beta}\mathrm{p}_{\beta_i}\mathrm{h}_{\beta_i}^{\frac{1-\beta}{\beta}} - \lambda_i^* + \nu^* = 0\ \forall\ i \in \mathcal{Y} \qquad & \text{(stationary condition)}, \end{aligned} \tag{30}$$

where p_{β_i} and h_{β_i} represent $\mathrm{p}_\beta(i|x)$ and $\mathrm{h}_\beta(x)_i$, respectively, for $i \in \mathcal{Y}$.

From the stationary condition in (30), we obtain

$$\lambda_i^* = -\frac{1}{\beta}\mathrm{p}_{\beta_i}\mathrm{h}_{\beta_i}^{\frac{1-\beta}{\beta}} + \nu^*\ \forall\ i.$$

Multiplying with h_{β_i} on both sides and using the complementary slackness in (30), we obtain

$$0 = \Big(-\frac{1}{\beta}\mathrm{p}_{\beta_i}\mathrm{h}_{\beta_i}^{\frac{1-\beta}{\beta}} + \nu^* \Big)\mathrm{h}_{\beta_i}$$

$$\implies \nu^* = \frac{1}{\beta}\mathrm{p}_{\beta_i}\mathrm{h}_{\beta_i}^{\frac{1-\beta}{\beta}} \implies \mathrm{h}_{\beta_i} = \Big(\frac{\mathrm{p}_{\beta_i}}{\nu^*\beta}\Big)^{\frac{\beta}{\beta-1}}. \tag{31}$$

Since the probabilities h_β satisfy $\sum_y \mathrm{h}_{\beta_i} = 1$, we have

$$\sum_{i\in\mathcal{Y}} \Big(\frac{\nu^*\beta}{\mathrm{p}_{\beta_i}}\Big)^{\frac{\beta}{1-\beta}} = 1 \implies \nu^{*\frac{\beta}{1-\beta}} = \frac{1}{\beta^{\frac{\beta}{1-\beta}} \sum_i \mathrm{p}_{\beta_i}{}^{\frac{\beta}{\beta-1}}}.$$

Finally, substituting the above expression ν^* into (31), we obtain, for each $x \in \mathcal{X}$,

$$\mathrm{h}_{\beta_i} = \frac{\mathrm{p}_{\beta_i}{}^{\frac{\beta}{\beta-1}}}{\sum_i \mathrm{p}_{\beta_i}{}^{\frac{\beta}{\beta-1}}} \implies \mathrm{h}_\beta(x)_y = \frac{\mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}}{\sum_y \mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}}, \tag{32}$$

which corresponds to (16) in the main paper.

In addition, as a consequence of (32), we obtain relation (15) as follows. Using the fact that $\sum_{y\in\mathcal{Y}} \mathrm{p}_\beta(y|x) = 1$, and

$$\mathrm{p}_\beta(y|x) = \Big(\mathrm{h}_\beta(x)_y \sum_y \mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}\Big)^{\frac{\beta-1}{\beta}}$$

from (32), we have

$$\sum_y \mathrm{h}_\beta(x)_y^{\frac{\beta-1}{\beta}} = \frac{1}{\big(\sum_y \mathrm{p}_\beta(y|x)^{\frac{\beta}{\beta-1}}\big)^{\frac{\beta-1}{\beta}}}. \tag{33}$$

Then, combining the relations in (32) and (33), we achieve

$$\mathrm{p}_\beta(y|x) = \frac{\mathrm{h}_\beta(x)_y^{\frac{\beta-1}{\beta}}}{\sum_y \mathrm{h}_\beta(x)_y^{\frac{\beta-1}{\beta}}}.$$

D PROOF OF THEOREM 2

Since $\mathrm{h}(x)_y$ is a probability, we have that $0 \le \mathrm{h}(x)_y \le 1$ and $\sum_{y\in\mathcal{Y}} \mathrm{h}(x)_y = 1$. Therefore, for $\beta \ge 1$, the first inequality in (18) is achieved as follows

$$\mathrm{h}(x)_y \le \mathrm{h}(x)_y^{\frac{1}{\beta}} \iff 1 - \mathrm{h}(x)_y^{\frac{1}{\beta}} \le 1 - \mathrm{h}(x)_y \iff \frac{\ell_\beta(\mathrm{h},(x,y))}{\beta} \le \ell_{\mathrm{MAE}}(\mathrm{h},(x,y)).$$

In the following, we prove the second inequality in (18). Consider the function $g(\mathrm{h}(x)_y) = \beta(1-\mathrm{h}(x)_y^{\frac{1}{\beta}}) - (1-\mathrm{h}(x)_y)$ that defines the difference between the α-loss and the MAE loss for given probability $\mathrm{h}(x)_y$. Such function is decreasing in $\mathrm{h}(x)_y$ since the derivative $g'(\mathrm{h}(x)_y) = 1 - \mathrm{h}(x)_y^{\frac{1}{\beta}-1} \le 0$ since $0 \le \mathrm{h}(x)_y \le 1$ and $\frac{1}{\beta} - 1 \le 0$, so that the minimum value of the function is achieved at $\mathrm{h}(x)_y = 1$. Therefore, we have that $g(\mathrm{h}(x)_y) \ge g(1) = 0$ for all $\mathrm{h}(x)_y \in [0,1]$ and $\beta \ge 1$ which proves the second inequality in (18).

We prove inequality (19) using the function $g(\mathrm{h}(x)_y)$ as follows. As described above, the function is decreasing in $\mathrm{h}(x)_y$ in the interval $[0,1]$. Then, the maximum value of the function is given by $g(0) = \beta - 1$ for $\beta \ge 1$. Therefore, we have that $g(\mathrm{h}(x)_y) \le \beta - 1 \implies \ell_\beta(\mathrm{h},(x,y)) - \ell_{\mathrm{MAE}}(\mathrm{h},(x,y)) \le \beta - 1$.

E PROOF OF THEOREM 3

The function $\varphi_\beta(x, \boldsymbol{\mu})$ is implicitly defined as

$$F(x, \boldsymbol{\mu}, \varphi_\beta(x, \boldsymbol{\mu})) = \sum_{y \in \mathcal{Y}} \Big(\frac{f(x, \boldsymbol{\mu})_y + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^\beta = 1, \tag{34}$$

and the gradient $\frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}}$ can be obtained using implicit differentiation (Dontchev and Rockafellar, 2009) as follows.

For any instance $x \in \mathcal{X}$ and parameters $\boldsymbol{\mu}$, the function φ_β satisfies (34). Therefore, differentiating both sides in (34) with respect to $\boldsymbol{\mu}$, we have

$$\sum_{y \in \mathcal{Y}} \Big(\frac{f(x, \boldsymbol{\mu})_y + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^{\beta - 1} \Big(\Phi(x, y) + \frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \Big) = 0.$$

Thus, rearranging the terms, we have

$$\frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = - \frac{\sum\limits_{y \in \mathcal{Y}} \Big(\frac{f(x, \boldsymbol{\mu})_y + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^{\beta - 1} \Phi(x, y)}{\sum\limits_{j \in \mathcal{Y}} \Big(\frac{f(x, \boldsymbol{\mu})_j + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^{\beta - 1}}. \tag{35}$$

Moreover, using Theorem 1, we have that the worst-case distribution is

$$\mathrm{p}_\beta(y|x) = \frac{\Big(\frac{f(x, \boldsymbol{\mu})_y + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^{\beta - 1}}{\sum\limits_{j \in \mathcal{Y}} \Big(\frac{f(x, \boldsymbol{\mu})_j + \varphi_\beta(x, \boldsymbol{\mu})}{\beta} + 1 \Big)_+^{\beta - 1}} \tag{36}$$

corresponding with a classifier given by parameters $\boldsymbol{\mu}$. Combining (35) and (36), we obtain the gradient

$$\frac{\partial \varphi_\beta(x, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = - \sum_{y \in \mathcal{Y}} \mathrm{p}_\beta(y|x) \Phi(x, y),$$

which corresponds to (22).

F PROOF OF COROLLARY 1

In the following, we show that the MGCE margin loss ℓ_β in (23) is classification-calibrated, that is, for any x and $\beta > 1$, if p^* denotes the underlying distribution of the data and

$$\boldsymbol{\mu}^* \in \arg\min_{\boldsymbol{\mu}} \sum_y \mathrm{p}^*(y|x) \ell_\beta(\boldsymbol{\mu}, (x, y)), \tag{37}$$

then

$$\arg\max_y f(x, \boldsymbol{\mu}^*)_y = \arg\max_y \mathrm{p}^*(y|x). \tag{38}$$

The minimization in (37) corresponding with the MGCE margin loss is given as

$$\arg\min_{\boldsymbol{\mu}} \sum_y \Big(- f(x, \boldsymbol{\mu})_y - \varphi_\beta(x, \boldsymbol{\mu}) \Big) \mathrm{p}^*(y|x). \tag{39}$$

The stationary condition for this minimization problem can be derived using the gradients in (21) and (22). In particular, such condition is given by

$$\sum_{y=1}^k (\mathrm{p}^*(y|x) - \mathrm{p}_\beta(y|x)) \Phi(x, y) = 0, \tag{40}$$

where p_β is the worst-case distribution corresponding to $\boldsymbol{\mu}^*$ as given in (17). The above equality implies that

$$\mathrm{p}^*(y|x) = \mathrm{p}_\beta(y|x), \tag{41}$$

since the one-hot encoded vectors $\Phi(x,1), \Phi(x,2), \ldots, \Phi(x,k)$ are linearly independent. Therefore, using the relation in (16), the corresponding minimax classifier $\mathrm{h}_\beta(x)_y$ satisfies

$$\mathrm{h}_\beta(x)_y \propto \mathrm{p}^*(y|x)^{\frac{\beta}{\beta-1}}, \tag{42}$$

which implies that $\arg\max_y f(x,\boldsymbol{\mu}^*)_y = \arg\max_y \mathrm{p}^*(y|x)$ since $\arg\max_y \mathrm{h}_\beta(x)_y = \arg\max_y f(x,\boldsymbol{\mu}^*)_y$, and $\frac{\beta}{\beta-1} > 1$.

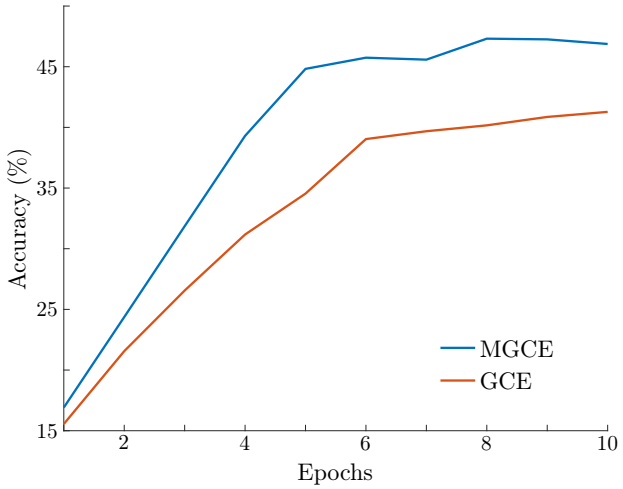


Figure 5: Top-1 test accuracy on the real-world noisy dataset Clothing-1M. The figure shows that the proposed MGCE outperforms GCE, which underfits on this complex dataset with 1 million training samples with noisy labels.

G EXPERIMENTATION DETAILS AND ADDITIONAL RESULTS

G.1 Selection of regularization parameter

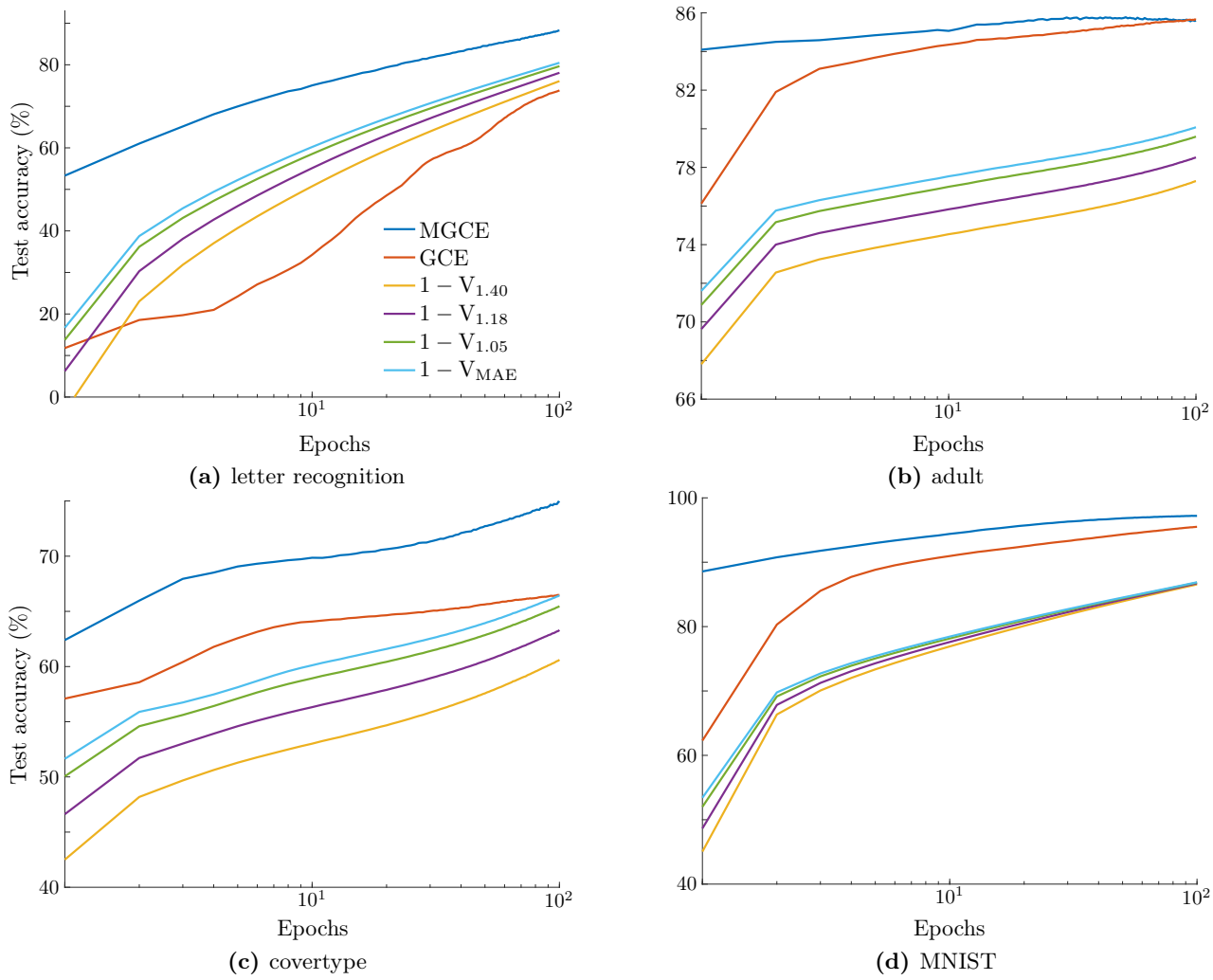
Table 2 shows the validation accuracies for multiple common values of regularization parameter λ_0 that appear in the literature (see e.g., Zhang and Sabuncu (2018); Ma et al. (2020); Zhou et al. (2021)). The table shows that the effect of the regularization parameter λ_0 is similar in the proposed method MGCE and existing techniques GCE. Based on the comparison, we use the same value of 10^{-5} for λ_0 in all datasets and methods.

Dataset	$\lambda_0 = 10^{-3}$		$\lambda_0 = 10^{-4}$		$\lambda_0 = 10^{-5}$	
	MGCE	GCE	MGCE	GCE	MGCE	GCE
SVHN	80.42 ± 0.57	85.00 ± 1.06	92.41 ± 0.28	93.53 ± 0.13	94.58 ± 0.02	93.32 ± 0.15
CIFAR10	67.40 ± 1.27	69.32 ± 0.93	87.98 ± 0.53	88.06 ± 0.39	91.49 ± 0.53	91.11 ± 0.09
FashionMNIST	89.56 ± 0.53	89.51 ± 0.33	93.09 ± 0.55	92.74 ± 0.43	92.79 ± 0.50	92.71 ± 0.31

Table 2: Average validation accuracies for different λ_0

G.2 Experimental setup for the real-world noisy WebVision dataset

The training details follow Ma et al. (2020), where for each method, we train a ResNet-50 architecture (He et al., 2016) using SGD for 250 epochs with initial learning rate 0.4, momentum 0.9, and L1 weight 10^{-5} and batch size 512. The learning rate is multiplied by 0.97 after every epoch of training. All the images are resized to 224 × 224. We also use the common data augmentations of random width/height shift, color jittering, and random horizontal flip are applied.



(a) letter recognition **(b)** adult **(c)** covertype **(d)** MNIST

Figure 6: Average test accuracy under clean data training data obtained for multiple tabular datasets. The value of loss parameter β is set to 1.4 for both MGCE and GCE. In addition, the figure reports the objective values $V_{1.40}$, $V_{1.18}$, $V_{1.05}$, and V_{MAE} corresponding with $\beta = 1.40$, $\beta = 1.18$, $\beta = 1.05$, and $\beta = 1.00$, respectively, that provide a lower bound on the test accuracy. The figure shows the fast convergence of the proposed MGCE in comparison to GCE, and that the bounds can provide a minimum estimate on the classification accuracy.

G.3 Evaluation on real-world noisy Clothing-1M dataset

Clothing-1M (Xiao et al., 2015) is a large-scale dataset with real-world noisy labels, whose images are clawed from the online shopping websites, and labels are generated based on surrounding texts. It contains 1 million training images with noisy labels, and 15k validation images, and 10k test images with clean labels. We train a ResNet-50 architecture (He et al., 2016) on the noisy training dataset and then validate its performance using the cleas test images. Following Park et al. (2023), we train the architecture using SGD for 10 epochs with initial learning rate 0.002, momentum 0.9, and L1 weight 10^{-5} and batch size 512. The learning rate is dropped by a factor of 10 at the halfway point of the training epochs.

Figure 5 reports the accuracy across the epochs for the test images having clean labels. The results show that the proposed MGCE method outperforms GCE method using the same value $\beta = 1.4$ as in the main paper.

G.4 Additional experimental results using tabular datasets

A simple multi-layer perceptron (MLP) classifier with 1024 hidden units was used for training. The model was optimized using SGD with a learning rate of 0.001, following the same training strategy as in the main experiments

(see Section 5.1). We evaluate the model on four tabular datasets: MNIST (70K samples, 10 classes), Letter Recognition (20K samples, 26 classes), Covertype (110K samples, 7 classes), and Adult (48K samples, 2 classes).

We compare the MGCE and GCE methods in terms of test accuracy achieved along the epochs, using multiple tabular datasets and a simple MLP classifier. In addition, we assess the objective value V_β of the proposed MGCE method in as an upper bound on the MAE classification risk of the classifier as discussed in Section 3.2 of the main paper.

Figure 6 reports the test accuracy as well as the lower bound it (or upper bound on the classification error) across the epochs corresponding with different values of β. The results are reported for $\beta = 1.4$, that is, the same value used for all the figures in the main paper. The results show that the proposed MGCE method achieves faster convergence than the GCE method, and results in improved accuracy. Moreover, the upper bound on the classification error V_β obtained by the MGCE method provides an estimate on the minimum test accuracy achieved by the classifier that is tighter for smaller values of β as shown in Theorem 2.

We also evaluate the performance of MGCE and GCE across general values of the parameter $\beta \in (1.05, 11)$. We use similar setup and evaluation metrics as in Table 1 of main paper that assesses complex DNNs. In particular, we compare with the baselines provided by CE ($\beta = \infty$) and the convex 0–1 loss introduced in Mazuelas et al. (2023). The results in Table 3 show that the proposed MGCE method attains strong accuracy improving on the existing methods especially in multi-class datasets.

Dataset	Loss	Test Accuracy (%)			SCE (%)		
		$\eta = 0.0$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.0$	$\eta = 0.2$	$\eta = 0.4$
MNIST	MGCE	**97.28 ± 0.07**	**96.15 ± 0.05**	**94.99 ± 0.05**	0.79 ± 0.06	**6.52 ± 0.14**	10.53 ± 0.15
	GCE	97.14 ± 0.04	95.74 ± 0.07	94.19 ± 0.23	**0.37 ± 0.03**	17.3 ± 0.3	**1.43 ± 0.07**
	CE	97.17 ± 0.04	95.59 ± 0.1	94.02 ± 0.09	0.4 ± 0.07	21.18 ± 0.76	39.46 ± 0.87
	MAE	97.24 ± 0.03	96.1 ± 0.06	94.95 ± 0.13	80.03 ± 0.12	81.18 ± 0.22	80.55 ± 0.23
covertype	MGCE	**77.24 ± 0.09**	**75.83 ± 0.21**	**74.08 ± 0.07**	1.74 ± 0.17	**12.58 ± 0.48**	**25.55 ± 0.2**
	GCE	76.93 ± 0.07	75.54 ± 0.14	73.99 ± 0.08	**1.43 ± 0.14**	12.89 ± 0.33	25.7 ± 0.74
	CE	77.28 ± 0.15	75.8 ± 0.08	74.11 ± 0.17	3.16 ± 0.2	16.41 ± 0.36	29.07 ± 0.58
	MAE	75.58 ± 0.09	73.13 ± 0.08	71.43 ± 0.22	54.96 ± 0.32	52.94 ± 0.1	51.38 ± 0.24
letter	MGCE	**90.63 ± 0.19**	**87.29 ± 0.19**	**83.94 ± 0.22**	**3.28 ± 0.3**	**9.97 ± 0.28**	40.03 ± 0.33
	GCE	86.64 ± 0.13	85.19 ± 0.29	82.67 ± 0.22	8.44 ± 0.15	25.03 ± 0.19	**38.9 ± 0.26**
	CE	87.59 ± 0.09	86.19 ± 0.21	83.66 ± 0.19	9.25 ± 0.14	30.72 ± 0.28	44.42 ± 0.18
	MAE	90.58 ± 0.16	87.15 ± 0.29	83.09 ± 0.52	83.99 ± 0.15	81.5 ± 0.28	77.76 ± 0.49
adult	MGCE	**85.68 ± 0.13**	84.56 ± 0.19	82.63 ± 0.39	7.85 ± 0.37	12.13 ± 0.57	6.02 ± 2.47
	GCE	**85.68 ± 0.09**	**85.33 ± 0.16**	**83.51 ± 0.6**	1.2 ± 0.24	**5.69 ± 0.1**	**5.21 ± 1.7**
	CE	**85.68 ± 0.08**	84.56 ± 0.21	81.38 ± 0.69	**0.76 ± 0.1**	14.3 ± 0.78	23.48 ± 1.73
	MAE	85.49 ± 0.17	83.85 ± 0.24	82.01 ± 0.38	16.72 ± 1.37	19.36 ± 0.21	18.69 ± 1.41

Table 3: Average test accuracy and static calibration error (SCE) for clean and noisy tabular datasets. The ratio η represents the fraction of samples corrupted by symmetric label noise. We report accuracies of the epoch where validation accuracy is maximum. Our proposed MGCE method obtains strong accuracy.