The Implicit Bias of Gradient Descent on Separable Multiclass Data

Hrithik Ravi[1] **Clayton Scott**[1] **Daniel Soudry**[2] **Yutong Wang**[3]
[1]University of Michigan [2]Technion - Israel Institute of Technology
[3]Illinois Institute of Technology
{hrithikr, clayscot}@umich.edu
daniel.soudry@gmail.com
ywang562@iit.edu

Abstract

Implicit bias describes the phenomenon where optimization-based training algorithms, without explicit regularization, show a preference for simple estimators even when more complex estimators have equal objective values. Multiple works have developed the theory of implicit bias for binary classification under the assumption that the loss satisfies an *exponential tail property*. However, there is a noticeable gap in analysis for multiclass classification, with only a handful of results which themselves are restricted to the cross-entropy loss. In this work, we employ the framework of Permutation Equivariant and Relative Margin-based (PERM) losses [Wang and Scott, 2024] to introduce a multiclass extension of the exponential tail property. This class of losses includes not only cross-entropy but also other losses. Using this framework, we extend the implicit bias result of Soudry et al. [2018] to multiclass classification. Furthermore, our proof techniques closely mirror those of the binary case, thus illustrating the power of the PERM framework for bridging the binary-multiclass gap.

1 Introduction

Overparameterized models such as neural networks have shown state-of-the-art performance in many applications, despite having the potential to overfit. Zhang et al. [2021] demonstrate that this potential is indeed realizable by training real-world models to fit random noise. In recent years, there have been several research efforts that aim to understand the impressive performance of overparametrized models despite this ability to overfit. Both the model architecture and the training algorithms for selecting the weights have been investigated in this regard.

Work on *implicit bias* [Soudry et al., 2018, Ji et al., 2020, Vardi, 2022] has focused on the latter factor. Implicit bias is the hypothesis that gradient-based methods have a built-in preference for models with low-complexity. This hypothesis is perhaps best understood in the setting of (unregularized) empirical risk minimization for learning a linear model under the assumption of linearly separable data. Soudry et al. [2018] showed that in binary classification, implicit bias holds when the loss has the exponential tail property [Soudry et al., 2018, Theorem 3]. The same work also demonstrated implicit bias in the multiclass setting for the cross-entropy loss, but implicit bias for a more broadly defined class of losses in the multiclass case is left open. In this work, we extend the notion of the exponential tail property to multiclass losses and prove that the property is sufficient for implicit bias to occur in the multiclass setting. Toward this end, we employ the framework of permutation equivariant and relative margin-based (PERM) losses [Wang and Scott, 2024].

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

1.1 Contributions

Multiclass extension of the exponential tail property (Definition 2.2) It is unclear how the exponential tail property for binary margin losses should be extended to the multiclass setting. By using the PERM framework, we provide a multiclass extension that generalizes the exponential tail property to multiclass (Definition 2.2 in Section 2.3). We further verify that this property holds for some common losses.

Sufficiency of the exponential tail property for implicit bias (Theorem 3.4) We prove that the proposed multiclass exponential tail property is sufficient for implicit bias. More precisely, we show in Theorem 3.4 that for almost all linearly separable multiclass datasets, given a convex, (β-smooth, strictly decreasing) PERM loss satisfying the exponential tail property in Definition 2.2, gradient descent exhibits directional convergence to the hard-margin multiclass SVM.

1.2 Related Work

Soudry et al. [2018] show that gradient descent, applied to *unregularized* empirical risk minimization, converges to the hard-margin SVM solution at a slow logarithmic rate, provided the loss satisfies the exponential tail property (defined below). Nacson et al. [2019] improve the convergence rate using a specific step-size schedule. Ji and Telgarsky [2019] extend implicit bias to the setting of *quasi-complete separation* [Candès and Sur, 2020], where the two classes are linearly separated but with a margin of zero. Many works have also considered gradient-based methods beyond gradient descent. For example, Gunasekar et al. [2018] examine the implicit bias effects of mirror descent [Beck and Teboulle, 2003], steepest descent [Boyd and Vandenberghe, 2004], and *adaptive* gradient descent [Duchi et al., 2011, Kingma and Ba, 2015]. Cotter et al. [2012], Clarkson et al. [2012], Ji et al. [2021] study first order methods that are designed specifically to approach the hard-margin SVM as quickly as possible.

Results for the multiclass setting are more scarce, and are *always* specific to cross-entropy. Soudry et al. [2018] establish implicit bias for cross-entropy loss. Lyu and Li [2019] focus on homogeneous predictors and prove convergence of GD on cross-entropy loss to a KKT point of the margin-maximization problem. Lyu and Li [2019] proves convergence of gradient flow to a generalized max-margin classifier for multiclass classification with cross-entropy loss using homogeneous models.[1] In the special case when the model are linear classifiers, the generalized max-margin classifier reduces to the classical hard-margin SVM. Lyu et al. [2021] consider two-layer neural networks and prove convergence of GD on cross-entropy loss to the max-margin solution under an additional assumption on the data, that both $\mathbf{x}$ and its negative counterpart $-\mathbf{x}$ must belong to the dataset. Wang et al. [2023] prove that in certain overparameterized regimes, gradient descent on squared loss leads to an equivalent solution to gradient descent on cross-entropy loss.

Beyond work establishing (rate of) convergence to the max-margin classifier, there is also a separate line of work [Shamir, 2021, Schliserman and Koren, 2022, 2023] focusing on the *generalization* aspect of implicit bias. These works examine the binary classification setting, with the exception of Schliserman and Koren [2022] who consider cross-entropy.

1.3 Notations

Let $K \geq 2$ and $d \geq 1$ denote the number of classes and feature space dimension, respectively. Let $[K] := \{1, 2, \ldots, K\}$. Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{v} \in \mathbb{R}^K$ whose entries are denoted by v_j for $j \in [K]$. Likewise, matrices are denoted by boldface uppercase letters, e.g., $\mathbf{W} \in \mathbb{R}^{d \times K}$. The columns of $\mathbf{W}$ are denoted $\mathbf{w}_1, \ldots, \mathbf{w}_K$. By $\mathbf{0}_n$ and $\mathbf{1}_n$ we denote the n-dimensional vectors of all 0's and all 1's respectively. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$.

By $\|\mathbf{v}\|$ we denote the Euclidean norm of vector $\mathbf{v}$. $\|\mathbf{A}\|_2$ is the spectral norm of matrix $\mathbf{A}$. Given two vectors $\mathbf{w}, \mathbf{v} \in \mathbb{R}^k$, we write $\mathbf{w} \succeq \mathbf{v}$ (resp. $\mathbf{w} \succ \mathbf{v}$) if $w_j \geq v_j$ (resp. $w_j > v_j$) for all $j \in [k]$; similarly we write $\mathbf{w} \preceq \mathbf{v}$ (resp. $\mathbf{w} \prec \mathbf{v}$) if $w_j \leq v_j$ (resp. $w_j < v_j$) for all $j \in [k]$. On the other

[1] Lyu and Li [2019] could be thought of as analyzing losses beyond CE, but the optimization problem would be non-convex so convergence might not be to a global minimum. See Appendix A for a more detailed discussion.

hand, if $\mathbf{A}$ and $\mathbf{B}$ are equally-sized *symmetric matrices*, then by $\mathbf{A} \succeq \mathbf{B}$ (resp. $\mathbf{A} \preceq \mathbf{B}$) we mean that $\mathbf{A} - \mathbf{B}$ (resp. $\mathbf{B} - \mathbf{A}$) is positive semi-definite, i.e. $\mathbf{A} - \mathbf{B} \succeq 0$ (resp. $\mathbf{B} - \mathbf{A} \succeq 0$).

A bijection from $[k]$ to itself is called a permutation on $[k]$. Denote by $\texttt{Sym}(k)$ the set of all permutations on $[k]$. For each $\sigma \in \texttt{Sym}(k)$, let $\mathbf{S}_\sigma$ denote the permutation matrix corresponding to σ. In other words, if $\mathbf{v} \in \mathbb{R}^k$ is a vector, then $[\mathbf{S}_\sigma \mathbf{v}]_j = \mathbf{v}_{\sigma(j)}$.

2 Multiclass Loss Functions

In multiclass classification, a classifier is typically represented in terms of a *class-score function* $f = (f_1, \ldots, f_K) : \mathbb{R}^d \to \mathbb{R}^K$, which maps an input $\mathbf{x} \in \mathbb{R}^d$ to a vector $\mathbf{v} := f(\mathbf{x})$ of class scores. For instance, f may be a feed-forward neural network and $\mathbf{v}$ in this context is sometimes referred to as the logits. The label set is $[K]$, and a label is predicted as $\operatorname{argmax}_j f_j(\mathbf{x})$. A K-ary multiclass loss function is a vector-valued function $\mathcal{L} = (\mathcal{L}_1, \ldots, \mathcal{L}_K) : \mathbb{R}^K \to \mathbb{R}^K$ where $\mathcal{L}_y(f(\mathbf{x}))$ is the loss incurred for outputting $f(\mathbf{x})$ when the ground truth label is y.

In binary classification, a classifier is typically represented using a function $g : \mathbb{R}^d \to \mathbb{R}$. The label set is $\{-1, 1\}$, and labels are predicted as $\mathbf{x} \mapsto \operatorname{sign}(g(\mathbf{x}))$. A *binary margin loss* is a function of the form $\psi : \mathbb{R} \to \mathbb{R}$ where $\psi(yg(\mathbf{x}))$ is the loss incurred for outputting $g(\mathbf{x})$ when the ground truth label is y. Margin losses have been central to the development of the theory of binary classification, and the lack of a multiclass counterpart to binary margin losses may have impaired the development of corresponding theory for multiclass classification. To address this issue, Wang and Scott [2024] introduce PERM losses as a bridge between binary and multiclass classification.

2.1 Permutation equivariant and relative margin-based (PERM) losses

Assume the label set is $[K]$. Define [2] the matrix $\mathbf{D} := [-\mathbf{I}_{K-1} \quad \mathbf{1}_{K-1}] \in \mathbb{R}^{(K-1)\times K}$. Observe that $\mathbf{D}\mathbf{v} = (v_K - v_1, v_K - v_2, \ldots, v_K - v_{K-1})^\top$ for all $\mathbf{v} \in \mathbb{R}^K$.

Definition 2.1 (PERM loss [Wang and Scott, 2024])**.** *Let $K \geq 2$ be an integer, and $\mathcal{L}$ be a K-ary multiclass loss function. We say that $\mathcal{L}$ is*

1. permutation equivariant *if $\mathcal{L}(\mathbf{S}_\sigma \mathbf{v}) = \mathbf{S}_\sigma \mathcal{L}(\mathbf{v})$ for all $\mathbf{v} \in \mathbb{R}^K$ and $\sigma \in \texttt{Sym}(K)$,*
2. relative margin-based *if for each $y \in [K]$ there exists a function $\ell_y : \mathbb{R}^{K-1} \to \mathbb{R}$ so that $\mathcal{L}_y(\mathbf{v}) = \ell_y(\mathbf{D}\mathbf{v}) = \ell_y(v_K - v_1, v_K - v_2, \ldots, v_K - v_{K-1})$, for all $\mathbf{v} \in \mathbb{R}^K$. We refer to the vector-valued function $\ell := (\ell_1, \ldots, \ell_K)$ as the* reduced form *of $\mathcal{L}$.*
3. PERM *if $\mathcal{L}$ is both permutation equivariant and relative margin-based. In this case, the function $\psi := \ell_K$ is referred to as the* template *of $\mathcal{L}$.*

Wang and Scott [2024] show that PERM losses are characterized by their template ψ. To show this, they introduce the *matrix label code*, an encoding of labels as matrices. Thus, for each $y \in [K-1]$, let $\mathbf{\Upsilon}_y$ be the $(K-1) \times (K-1)$ identity matrix, but with the y-th column replaced by all -1's. For $y = K$, let $\mathbf{\Upsilon}_y$ be the identity matrix. Note that when $K = 2$, this definition reduces to $\mathbf{\Upsilon}_y = (-1)^y$, the standard encoding of labels in the binary setting. Observe that (after permutation) $\mathbf{\Upsilon}_y \mathbf{D}\mathbf{v} = (v_y - v_1, v_y - v_2, \ldots, v_y - v_K)^\top \in \mathbb{R}^{K-1}$, where the $v_y - v_y = 0$ entry is omitted. Please see Wang and Scott [2024, Lemma B.2] for a simple proof.

Theorem 2.1 (Wang and Scott [2024])**.** *Let $\mathcal{L} : \mathbb{R}^K \to \mathbb{R}^K$ be a PERM loss with template ψ, and let $v \in \mathbb{R}^K$ and $y \in [K]$ be arbitrary. Then ψ is a symmetric function. Moreover,*

$$\mathcal{L}_y(\mathbf{v}) = \psi(\mathbf{\Upsilon}_y \mathbf{D}\mathbf{v}). \tag{1}$$

Conversely, let $\psi : \mathbb{R}^{K-1} \to \mathbb{R}$ be a symmetric function. Define a multiclass loss function $\mathcal{L} = (\mathcal{L}_1, \ldots, \mathcal{L}_k) : \mathbb{R}^K \to \mathbb{R}^K$ according to Eqn. (1)*. Then $\mathcal{L}$ is a PERM loss with template ψ.*

Theorem 2.1 shows that a PERM loss is characterized by its template ψ. The right hand side of Eqn. (1) is referred to as the *relative margin form* of the loss, which extends binary margin losses to multiclass. As noted by Wang and Scott [2024], an advantage of the relative margin form is that it

[2] Also see [Wang and Scott, 2024, Definition 2].

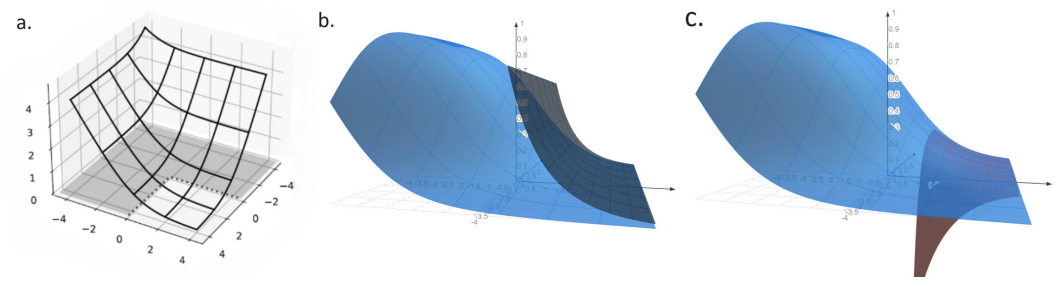

Figure 1: An illustration of the exponential tail property for the cross entropy/multinomial logistic loss when $K = 3$. *Panel a.* Plot of $\psi(\mathbf{u}) = \log(1 + \exp(-u_1) + \exp(-u_2))$, the template for the multinomial logistic loss. Note that the complement of the positive orthant in the domain $\mathbb{R}^2$ is shown in gray. *Panel b. and c.* Plot of the upper bound (shown in black) and lower bounds (red) of $-\frac{\partial \psi}{\partial u_1}$ (blue) respectively. These bounds are from Appendix C.1.3 where $u_\pm = 0$ and $c = 1$. Note that the lower bound is valid in the positive orthant, i.e., the red surface is below the blue one there.

decouples the labels from the predicted scores, which facilitates analysis. Our results below support this understanding.

Many losses in the literature are PERM losses, including the cross-entropy loss whose template is $\psi(\mathbf{u}) = \log(1 + \sum_{i=1}^{K-1} \exp(-u_i))$, the multiclass exponential loss [Mukherjee and Schapire, 2013] whose template is $\psi(\mathbf{u}) = \sum_{i=1}^{K-1} \exp(-u_i)$, and the PairLogLoss [Wang et al., 2022] whose template is = $\psi(\mathbf{u}) = \sum_{i=1}^{K-1} \log(1 + \exp(-u_i))$. See Wang and Scott [2024] for other examples.

2.2 Regularity assumptions on loss functions

Let $\mathcal{L}$ be a PERM loss with differentiable template ψ. If

$$\frac{\partial \psi}{\partial u_i}(\mathbf{u}) < 0, \quad \text{for all } i \in \{1, 2 \ldots, K-1\}, \mathbf{u} \in \mathbb{R}^{K-1},$$

i.e., the gradient of the template is entrywise strictly negative, then we say that the PERM loss $\mathcal{L}$ is *strictly decreasing*. In this case, we write $\nabla\psi \prec \mathbf{0}$, where $\mathbf{0}$ is the 0-vector. If the template is differentiable, then it is convex if:

$$\psi(\mathbf{u_1}) \geq \psi(\mathbf{u_2}) + \nabla\psi(\mathbf{u_2})^\top(\mathbf{u_1} - \mathbf{u_2}), \quad \text{for all } \mathbf{u_1}, \mathbf{u_2} \in \mathbb{R}^{K-1}.$$

If ψ is twice-differentiable, this is equivalent to saying that the Hessian is positive-semidefinite:

$$\nabla^2\psi(\mathbf{u}) \succeq 0 \quad \text{for all } \mathbf{u} \in \mathbb{R}^{K-1}.$$

Finally, the template is said to be β-smooth if its gradient is β-Lipschitz:

$$\|\nabla\psi(\mathbf{u_1}) - \nabla\psi(\mathbf{u_2})\| \leq \beta\|\mathbf{u_1} - \mathbf{u_2}\|, \quad \text{for all } \mathbf{u_1}, \mathbf{u_2} \in \mathbb{R}^{K-1}.$$

If ψ is twice-differentiable, this is equivalent to saying that the maximum eigenvalue of its Hessian is bounded by β:

$$\|\nabla^2\psi(\mathbf{u})\|_2 \leq \beta \quad \text{for all } \mathbf{u} \in \mathbb{R}^{K-1},$$

where $\|\mathbf{A}\|_2$ is the spectral norm of matrix $\mathbf{A}$.

2.3 Multiclass analogue of exponential tail property

In the binary setting, the exponential tail property defined in prior work (Soudry et al. [2018], Nacson et al. [2019], Ji et al. [2020]) is assumed to hold for the negative *derivative* of the loss. Similarly, in the multiclass setting we are interested in bounding the negative *gradient* of the PERM loss template.

Definition 2.2 (Multiclass exponential tail property)**.** *A multiclass PERM loss with template* $\psi : \mathbb{R}^{K-1} \to \mathbb{R}$ *has the* exponential tail (ET) *property if there exist* $u_+, u_- \in \mathbb{R}$ *and positive* $c > 0$ *such*

that for all $i \in [K-1]$ the following holds:

$$\forall \mathbf{u} \text{ s.t. } \min_{j\in[K-1]} u_j > u_+, \text{ we have } -\frac{\partial \psi}{\partial u_i}(\mathbf{u}) \le c\exp(-u_i), \quad \text{and}$$

$$\forall \mathbf{u} \text{ s.t. } \min_{j\in[K-1]} u_j > u_-, \text{ we have } -\frac{\partial \psi}{\partial u_i}(\mathbf{u}) \ge c\left(1 - \sum_{j\in[K-1]} \exp(-u_j)\right)\exp(-u_i).$$

Remark 2.2. *We show in Appendix C that cross-entropy (CE), multiclass exponential loss, and PairLogLoss all have this property.*

3 Main Result

Consider a dataset $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, with $\mathbf{x}_n \in \mathbb{R}^d$ and class labels $y_n \in [K] := \{1, \dots, K\}$. The class score function for class k is $f_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x}$. Define $\mathbf{X} \in \mathbb{R}^{d\times N}$ to be the matrix whose nth column is $\mathbf{x}_n$. Define $\mathbf{W} \in \mathbb{R}^{d\times K}$ to be the matrix whose kth column is $\mathbf{w}_k$. The learning objective is

$$\mathcal{R}(\mathbf{W}) = \sum_{n=1}^N \mathcal{L}_{y_n}\left(\mathbf{W}^\top \mathbf{x}_n\right). \tag{2}$$

From Eqn. 1, if $\mathcal{L}$ is a PERM loss, then $\mathcal{L}_y(\mathbf{v}) = \psi(\mathbf{\Upsilon}_y \mathbf{D}\mathbf{v})$, and the learning objective becomes

$$\mathcal{R}(\mathbf{W}) = \sum_{i=1}^N \psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top \mathbf{x}_i\right). \tag{3}$$

Up to permuting the entries, $\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i$ is equal to the $(K-1)$-dimensional vector of relative-margins $\left[(\mathbf{w}_{y_i} - \mathbf{w}_1)^\top\mathbf{x}_i, (\mathbf{w}_{y_i} - \mathbf{w}_2)^\top\mathbf{x}_i, \dots, (\mathbf{w}_{y_i} - \mathbf{w}_K)^\top\mathbf{x}_i\right]^\top$, where the 0-valued entry $(\mathbf{w}_{y_i} - \mathbf{w}_{y_i})^\top\mathbf{x}_i$ is omitted. This follows from Wang and Scott [2024, Lemma B.2].

We are now ready to state our assumptions on the loss:

Assumption 3.1. *The PERM loss's template ψ is convex, β-smooth, strictly decreasing and non-negative.* [3] [4]

Assumption 3.2. *The PERM loss has exponential tail as defined in Definition 2.2.*

To optimize Eqn. (2) we employ gradient descent with fixed learning rate η. Define $\mathbf{w} := \mathsf{vec}(\mathbf{W})$ where vec denotes vectorization by column-stacking (See Definition B.1), and let the gradient descent iterate at time t be $\mathbf{w}(t)$. Then:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\nabla\mathcal{R}(\mathbf{w}(t)).$$

Define the "matrix-version" of the trajectory $\mathbf{W}(t) \in \mathbb{R}^{d\times K}$ such that $\mathbf{w}(t) = \mathsf{vec}(\mathbf{W}(t))$. Throughout this work, we frequently work with the risk as a *matrix*-input scalar-output function $\mathcal{R}(\mathbf{W})$, and as a *vector*-input scalar-output function $\mathcal{R}(\mathbf{w})$.

These two formulations will each be useful in different situations. For instances, adopting the matrix perspective can facilitate calculation of bounds, e.g., in Section 4.2. On the other hand, the vectorized formulation is easier for defining the Hessian of the risk $\nabla^2\mathcal{R}(\mathbf{w})$. See Appendix B for detail.

We focus on linearly separable datasets:

Assumption 3.3. *The dataset is linearly separable, i.e. there exists $\mathbf{w} \in \mathbb{R}^{dK}$ such that $\forall n \in [N], \forall k \in [K]\backslash\{y_n\} : \mathbf{w}_{y_n}^\top\mathbf{x}_n \ge \mathbf{w}_k^\top\mathbf{x}_n + 1$. Equivalently, there exists $\mathbf{W} \in \mathbb{R}^{d\times K}$ such that $\forall n \in [N]$, $\mathbf{\Upsilon}_{y_n}\mathbf{D}\mathbf{W}^\top\mathbf{x}_n \succeq \mathbf{1}$.*

[3] We note that in the binary case the implicit bias result in [Soudry et al., 2018, Theorem 3] does not require the loss to be convex. Closing this binary-multiclass gap is an open question.

[4] Note that multiclass exponential loss $\psi(\mathbf{u})$ does not have a global smoothness constant. However, we show in Appendix C.2.2 that any learning rate $\eta < 1/\left(B^2\mathcal{R}(\mathbf{w}(0))\right)$ is sufficient for the gradient descent iterates to achieve local smoothness, where $B = \sqrt{(2K-2)}\sum_{i=1}^N \|\mathbf{x}_i\|$.

Finally, let $\hat{\mathbf{w}}$ be the multiclass hard-margin SVM solution for the linearly separable dataset:

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \tfrac{1}{2}\|\mathbf{w}\|^2 \text{ s.t. } \forall n, \forall k \neq y_n : \mathbf{w}_{y_n}^\top \mathbf{x}_n \geq \mathbf{w}_k^\top \mathbf{x}_n + 1. \tag{4}$$

Now we state the main result of the paper:

Theorem 3.4. *For any PERM loss satisfying Assumptions 3.1 and 3.2, for all linearly separable datasets such that Assumption 4.1 holds, any sufficiently small learning rate $0 < \eta < 2\beta^{-1}\sigma_{max}^{-2}(\mathbf{X})$, and any initialization $\mathbf{w}(0)$, the iterates of gradient descent will behave as*

$$\mathbf{w}(t) = \hat{\mathbf{w}} \log(t) + \boldsymbol{\rho}(t)$$

where the norm of the residual, $\|\boldsymbol{\rho}(t)\|$, is bounded. This implies a directional convergence behavior:

$$\lim_{t\to\infty} \frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|}.$$

In Appendix I, we show experimental results demonstrating implicit bias towards the hard margin SVM when using the PairLogLoss, in line with Theorem 3.4.

4 Proof Sketch

In this section we will overview the proof of the result. Along the way, we prove lemmas that extend to the multiclass setting results from Soudry et al. [2018]. The extensions are facilitated by the PERM framework, in particular the relative margin from of the loss.

We adopt the notation of Soudry et al. [2018] where possible throughout this proof. Recalling the notation and definitions from the paper: let us define the standard basis $\mathbf{e}_k \in \mathbb{R}^K$ such that $(\mathbf{e}_k)_i = \delta_{ki}$ (where δ is the Kronecker-delta function), and the d-dimension identity matrix $\mathbf{I}_d$. Define $\mathbf{A}_k \in \mathbb{R}^{dK\times d}$ as the Kronecker product between $\mathbf{e}_k$ and $\mathbf{I}_d$, i.e. $\mathbf{A}_k = \mathbf{e}_k \otimes \mathbf{I}_d$. We can then relate the original k^{th}-class predictor $\mathbf{w}_k$ to the long column-vector $\mathbf{w}$ as follows: $\mathbf{A}_k^\top \mathbf{w} = \mathbf{w}_k$. Next define $\tilde{\mathbf{x}}_{n,k} := (\mathbf{A}_{y_n} - \mathbf{A}_k)\mathbf{x}_n$. Using this notation, the multiclass SVM becomes

$$\operatorname{argmin}_{\mathbf{w}} \tfrac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad \forall n, \forall k \neq y_n : \mathbf{w}^\top \tilde{\mathbf{x}}_{n,k} \geq 1 \tag{5}$$

For each $k \in [K]$, define $\mathcal{S}_k = \arg\min_n (\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k)^\top \mathbf{x}_n = \{n : (\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k)^\top \mathbf{x}_n = 1\}$, i.e., the k^{th} class support vectors. From the KKT optimality conditions for Eqn. (5), we have for some dual variables $\alpha_{n,k} > 0$ that

$$\hat{\mathbf{w}} = \sum_{n=1}^{N}\sum_{k=1}^{K} \alpha_{n,k}\tilde{\mathbf{x}}_{n,k}\mathbb{1}_{n\in\mathcal{S}_k}. \tag{6}$$

Finally, define

$$\mathbf{r}(t) = \mathbf{w}(t) - \log(t)\,\hat{\mathbf{w}} - \tilde{\mathbf{w}} \tag{7}$$

where $\tilde{\mathbf{w}}$ is a solution to

$$\forall k \in [K], \forall n \in \mathcal{S}_k : \eta \exp\left(-\mathbf{x}_n^\top (\tilde{\mathbf{w}}_{y_n} - \tilde{\mathbf{w}}_k)\right) = \alpha_{n,k}. \tag{8}$$

In Soudry et al. [2018], the existence of $\tilde{\mathbf{w}}$ is proven for the binary case for almost all datasets, and assumed in the multiclass case. Here, we also state the existence of $\tilde{\mathbf{w}}$ as an additional assumption:

Assumption 4.1. *Eqn. 8 has a solution, denoted $\tilde{\mathbf{w}}$.*

We pose the problem of proving Assumption 4.1 for almost all datasets as a conjecture in Appendix H, where we also show experimentally that on a large number (100 instances for each choice of $d \in \{2,3,4,5,6\}$ and $K \in \{3,4,5,6\}$) of synthetically generated linearly separable datasets, Assumption 4.1 indeed holds.

Note that $\mathbf{r}(t) = \boldsymbol{\rho}(t) - \tilde{\mathbf{w}}$, and $\tilde{\mathbf{w}}$ is independent of t, so bounding $\mathbf{r}(t)$ is equivalent to bounding $\boldsymbol{\rho}(t)$. Following the same steps as Soudry et al. [2018, Appendix E.3]:

$$\|\mathbf{r}(t+1)\|^2 - \|\mathbf{r}(t)\|^2 = \underbrace{\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2}_{\text{First Term}} + 2\underbrace{(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t)}_{\text{Second Term}} \tag{9}$$

The high-level approach is to bound the two terms of the above expansion for $\mathbf{r}(t)$ and then use a telescoping argument to bound $\mathbf{r}(t)$ for all $t > 0$. Below we provide the main arguments; for a complete proof of the second term's bound, please refer to Appendix F.

4.1 Bounding the First Term

Using $\log(1+x) \leq x$ for all $x > 0$, we expand the first term as follows:

$$\begin{aligned}\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 &\leq \eta^2 \|\nabla \mathcal{R}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 t^{-2} + 2\eta \hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t)) \log\left(1 + t^{-1}\right) \\ &\leq \eta^2 \|\nabla \mathcal{R}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 t^{-2}\end{aligned}$$

Obtaining the second inequality requires proving that

$$2\eta \hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t)) \log\left(1 + t^{-1}\right) \leq 0, \text{or equivalently, } \hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t)) < 0 \tag{10}$$

We will spend the rest of this subsection going over the complete proof of this inequality.

First we state the following lemma (derived in Appendix B.2) that gives us a useful expression for the gradient of the risk w.r.t. $\mathbf{W}$:

Lemma 4.2. *For any $\mathbf{W} \in \mathbb{R}^{d \times K}$, we have that $\nabla \mathcal{R}(\mathbf{W}) = \sum_{i=1}^N \mathbf{x}_i \nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D}$.*

This expression involves weight matrix $\mathbf{W}$. However the inequality we set out to prove (Eqn. (10)) is in terms of $\mathbf{w} = \mathsf{vec}(\mathbf{W})$. Throughout our main result proof, these two different forms – weight matrix versus vectorization of that matrix – will each be useful in different situations. Thus, to shuttle back and forth between these forms, the following well-known identity is useful:

Lemma 4.3. *For equally sized matrices $\mathbf{M}$ and $\mathbf{N}$, we have $\mathsf{vec}(\mathbf{M})^\top \mathsf{vec}(\mathbf{N}) = \mathsf{tr}(\mathbf{M}^\top \mathbf{N})$.*

Now we can prove our inequality of interest, i.e., Eqn. (10).

Lemma 4.4. *(Multiclass generalization of Soudry et al. [2018, Lemma 1]) For any PERM loss that is β-smooth, strictly decreasing, and non-negative, (Assumption 3.1) and Assumption 3.2, and for almost all linearly separable datasets (Assumption 3.3), we have $\hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t)) < 0$.*

Proof. Define matrix $\hat{\mathbf{W}}$ such that $\hat{\mathbf{w}} = \mathsf{vec}(\hat{\mathbf{W}})$. Since $\mathbf{w}(t) = \mathsf{vec}(\mathbf{W}(t))$, Lemma 4.3 implies

$$\hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t)) = \mathsf{tr}(\hat{\mathbf{W}}^\top \nabla \mathcal{R}(\mathbf{W}(t))) \tag{11}$$

To see how the PERM framework allows for a simple generalization of binary results, we will compare our multiclass proof side-by-side with the binary proof discussed in Soudry et al. [2018, Lemma 1]. In the binary case, we have $\mathcal{R}(\mathbf{w}) = \sum_{i=1}^N \psi\left(y_i \mathbf{w}^\top \mathbf{x}_i\right) \implies \nabla \mathcal{R}(\mathbf{w}) = \sum_{i=1}^N \psi'\left(y_i \mathbf{w}^\top \mathbf{x}_i\right) y_i \mathbf{x}_i$. Thus $\hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}) = \sum_{i=1}^N \psi'\left(y_i \mathbf{w}^\top \mathbf{x}_i\right) y_i \hat{\mathbf{w}}^\top \mathbf{x}_i$. In the multiclass case, the analogous quantity is $\mathsf{tr}(\hat{\mathbf{W}}^\top \nabla \mathcal{R}(\mathbf{W}(t)))$ which can be computed as

$$\sum_{i=1}^N \mathsf{tr}(\hat{\mathbf{W}}^\top \mathbf{x}_i \nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D}) = \sum_{i=1}^N \nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D} \hat{\mathbf{W}}^\top \mathbf{x}_i.$$

In the multiclass proof we used the risk gradient from Lemma 4.2 as well as the cyclic property of the trace operator. Then we dropped the trace because $\nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D} \hat{\mathbf{W}}^\top \mathbf{x}_i$ is a scalar (since $\nabla \psi(\cdot) \in \mathbb{R}^{K-1}$, $\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i \in \mathbb{R}^{K-1}$). For illustrative purpose, we place the rest of the proof, in both the binary and multiclass setting, side-by-side:

Binary: $\hat{\mathbf{w}}^\top \nabla \mathcal{R}(\mathbf{w}(t))$.
Focusing on just the i-th term of this sum:

$$\psi'\left(y_i \mathbf{w}(t)^\top \mathbf{x}_i\right) y_i \hat{\mathbf{w}}^\top \mathbf{x}_i$$

ψ is assumed to be strictly decreasing, i.e. $\psi'\left(y_i \mathbf{w}(t)^\top \mathbf{x}_i\right) < 0$. The dataset is linearly separable, so $y_i \hat{\mathbf{w}}^\top \mathbf{x}_i \geq 1$. Thus we obtain a sum (from $i = 1$ to N) of negative terms.

Multiclass: $\mathsf{tr}(\hat{\mathbf{W}}^\top \nabla \mathcal{R}(\mathbf{W}(t)))$.
Focusing on just the i-th term of this sum:

$$\nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D} \hat{\mathbf{W}}^\top \mathbf{x}_i$$

ψ is assumed to be strictly decreasing, i.e. $\nabla \psi\left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right) \prec \mathbf{0}$. The dataset is linearly separable, so $\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \hat{\mathbf{W}}^\top \mathbf{x}_i \succeq 1$. Thus we obtain a sum (from $i = 1$ to N) of negative terms.

Thus we see how the PERM framework allows us to essentially mirror the binary proof. In Remark 4.5, we elaborate more on the necessity of the relative margin form here. □

Lemma 4.4 directly implies the auxiliary inequality we set out to prove (see Eqn. (10)). Thus we obtain:

$$\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 \leq \eta^2 \|\nabla \mathcal{R}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 t^{-2} \tag{12}$$

Remark 4.5. *Let us see what happens to our proof if we just used the general risk form in Eqn.* (2) *without the PERM framework. First, we need an expression for the gradient of the risk:* $\nabla \mathcal{R}(\mathbf{W}) = \sum_{i=1}^{N} \mathbf{x}_i \nabla \mathcal{R}_{y_i} \left(\mathbf{W}^\top \mathbf{x_i}\right)^\top$. *Proceeding similarly to the binary case, we focus on just the* i*-th term of* $\mathsf{tr}\left(\hat{\mathbf{W}}^\top \nabla \mathcal{R}(\mathbf{W})\right)$*:*

$$\mathsf{tr}\left(\hat{\mathbf{W}}^\top \mathbf{x}_i \nabla \mathcal{R}_{y_i} \left(\mathbf{W}^\top \mathbf{x_i}\right)^\top\right) = \mathsf{tr}\left(\nabla \mathcal{R}_{y_i} \left(\mathbf{W}^\top \mathbf{x_i}\right)^\top \hat{\mathbf{W}}^\top \mathbf{x}_i\right) = \nabla \mathcal{R}_{y_i} \left(\mathbf{W}^\top \mathbf{x_i}\right)^\top \hat{\mathbf{W}}^\top \mathbf{x}_i$$

From here it is not clear how to proceed. The linear separability condition (Assumption 3.3) is not useful anymore- it does not make a statement about the scores in the vector $\hat{\mathbf{W}}^\top \mathbf{x}_i$*, but rather their* relative margins *(produced by the multiplication* $\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \hat{\mathbf{W}}^\top \mathbf{x}_i$*).*

4.2 Bounding the Second Term

In the previous subsection we established a bound on the first term of Eqn. (9). Here we sketch the main arguments required to bound the second term, i.e. $(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t)$. For more details please refer to Appendix F. We state our final bound below as a lemma:

Lemma 4.6. *(Generalization of Soudry et al. [2018, Lemma 20]) Define* θ *to be the minimum SVM margin across all datapoints and classes, i.e.* $\theta = \min_k \left[\min_{n \notin \mathcal{S}_k} \tilde{\mathbf{x}}_{n,k}^\top \hat{\mathbf{w}}\right] > 1$*. Then*

$$\exists C_1, C_2, t_1 : \forall t > t_1 : (\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq C_1 t^{-\theta} + C_2 t^{-2}. \tag{13}$$

A remark is in order on the difference of the above result to Soudry et al. [2018, Lemma 20]: on a high-level, we are able to generalize the argument of Soudry et al. [2018, Lemma 20] to account for *both binary and multiclass classification*, as well as general PERM ET losses beyond just CE.

We now proceed with the proof sketch. The first step is to rewrite $(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t)$ as

$$\begin{aligned}
&(-\eta \nabla \mathcal{R}(\mathbf{w}(t)) - \hat{\mathbf{w}} [\log(t+1) - \log(t)])^\top \mathbf{r}(t) \qquad \because \text{Definition of } \mathbf{r}(t) \text{ in Equation (7)} \\
&= \hat{\mathbf{w}}^\top \mathbf{r}(t) \left(t^{-1} - \log\left(1 + t^{-1}\right)\right) + \mathsf{tr}\left(\left(-\eta \sum_{i=1}^{N} \mathbf{x}_i \nabla \psi \left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D}\right)^\top \mathbf{R}(t)\right) \\
&\quad - t^{-1} \hat{\mathbf{w}}^\top \mathbf{r}(t) \qquad \because \text{Expression for } \nabla \mathcal{R}(\mathbf{w}(t)) \text{ from Lemma 4.4}
\end{aligned} \tag{14}$$

We defer the bound on the first term $\hat{\mathbf{w}}^\top \mathbf{r}(t) \left(t^{-1} - \log\left(1 + t^{-1}\right)\right)$ of Equation (14) to the appendix, and instead focus on the second two terms. Using the cyclic property of the trace, the term in the above final line involving the trace can be further simplified as:

$$\sum_{i=1}^{N} -\nabla \psi \left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{R}(t)^\top \mathbf{x}_i \tag{15}$$

Note that for each $i \in [N]$, a summand in Equation (15) is an inner product between two $(K-1)$-dimensional vectors, i.e., $-\nabla \psi \left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)$ and $\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{R}(t)^\top \mathbf{x}_i$. To proceed, to expand this inner product out as

$$\sum_{i=1}^{N} \sum_{k \in [K] \setminus \{y_i\}} [\![-\nabla \psi \left(\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k [\![\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{R}(t)^\top \mathbf{x}_i]\!]_k \tag{16}$$

Remark 4.7. *Here,* $[\![\cdot]\!]_k : \mathbb{R}^{K-1} \to \mathbb{R}$ *is defined as the coordinate projection such that* $[\![\boldsymbol{\Upsilon}_{y_i} \mathbf{D} \mathbf{W}^\top \mathbf{x}_i]\!]_k = \tilde{\mathbf{x}}_{i,k}^\top \mathbf{w}$*. Note that* $[\![\cdot]\!]_k$ *implicitly depends on* i *(the* $\tilde{\mathbf{x}}_{i,y_i}$ *0-entry is omitted). But we abuse notation for brevity. Please see Appendix D for a more precise definition.*

Using Equation (6) and Equation (8) we express the last two terms in Equation (14) as

$$\Big(\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_i\}}[\![-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i\right)]\!]_k[\![\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top\mathbf{x}_i]\!]_k\Big)-t^{-1}\hat{\mathbf{w}}^\top\mathbf{r}(t)$$

$$=\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_i\}}\big([\![-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i\right)]\!]_k-t^{-1}\exp(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{i,k})\mathbb{1}_{\{i\in S_k\}}\big)\tilde{\mathbf{x}}_{i,k}^\top\mathbf{r}(t)\qquad(17)$$

Finally, to upper bound the above expression, we consider a single tuple (i,k) case-by-case, depending on the sign of $\tilde{\mathbf{x}}_{i,k}^\top\mathbf{r}(t)$. This is the step where the upper and lower bounds in Definition 2.2 come in. Lemma D.2 in the appendix essentially applies Definition 2.2 to the relative margins to yield

$$[\![-\nabla\psi(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i)]\!]_k\le\exp(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w}(t)),\qquad\text{and}\qquad(18)$$

$$[\![-\nabla\psi(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i)]\!]_k\ge(1-\textstyle\sum_{r\in[K]\setminus\{y_i\}}\exp(-\tilde{\mathbf{x}}_{i,r}^\top\mathbf{w}(t)))\exp(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w}(t))\qquad(19)$$

for all $k\in[K]\setminus\{y_i\}$. We use Definition 2.2's exponential tail bounds by proving that the relative margins $\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w}(t)$ that appear in Lemma D.2 eventually become positive. This is true due to the following lemma (see Appendix E for the proof, which again mirrors the binary case):

Lemma 4.8. *(Multiclass generalization of Soudry et al. [2018, Lemma 1]) Consider any linearly separable dataset, and any PERM loss with template ψ that is convex, β-smooth, strictly decreasing, and non-negative. For all $k\in\{1,...,K\}$, let $\mathbf{w}_k(t)$ be the gradient descent iterates at iteration t for the k^{th} class. Then $\forall i\in\{1,...,N\},\forall j\in\{1,...,K\}\setminus\{y_i\}:\lim_{t\to\infty}(\mathbf{w}_{y_i}(t)-\mathbf{w}_j(t))^\top\mathbf{x}_i\to\infty$.*

This lemma lets us use the exponential tail bounds with any finite $u_\pm$. To conclude, we apply the upper (18) and lower bounds (19) to the summation in Equation (17), and reduce the problem to that of Soudry et al. [2018, Appendix E], thereby proving Lemma 4.6. See our Appendix F for details.

4.3 Tying It All Together

We use the logic of Soudry et al. [2018, Appendix A.2] to conclude the analysis. Define

$$C=\sum_{t=0}^{\infty}\left\|\mathbf{r}\left(t+1\right)-\mathbf{r}\left(t\right)\right\|^2\le\sum_{t=0}^{\infty}\eta^2\left\|\nabla\mathcal{R}\left(\mathbf{w}\left(t\right)\right)\right\|^2+\left\|\hat{\mathbf{w}}\right\|^2t^{-2}.$$

In the latter inequality we used Eqn. (12). Thus, C is bounded because from Soudry et al. [2018, Lemma 10], we know that $\sum_{t=0}^{\infty}\left\|\nabla\mathcal{R}\left(\mathbf{w}\left(t\right)\right)\right\|^2<\infty$. Here we note that Soudry et al. [2018, Lemma 10] requires the ERM objective $\mathcal{R}\left(\mathbf{w}\right)$ to be β'-smooth for some positive β'. It is easy to show that if the loss is β-smooth, then $\mathcal{R}\left(\mathbf{w}\right)$ is $\beta\sigma_{\max}^2\left(\mathbf{X}\right)$-smooth. This explains the learning rate condition $\eta<2/\left(\beta\sigma_{\max}^2\left(\mathbf{X}\right)\right)$ in our theorem. Also, a t^{-p} power series converges for any $p>1$.

Recalling the initial expansion of $\|\mathbf{r}(t+1)\|$ from Eqn. (9):

$$\left\|\mathbf{r}\left(t+1\right)\right\|^2=\left\|\mathbf{r}\left(t+1\right)-\mathbf{r}\left(t\right)\right\|^2+2\left(\mathbf{r}\left(t+1\right)-\mathbf{r}\left(t\right)\right)^\top\mathbf{r}\left(t\right)+\left\|\mathbf{r}\left(t\right)\right\|^2.\qquad(20)$$

Combining the bounds in Eqn. (12) and Lemma 4.6 into Eqn. (9), we find

$$\left\|\mathbf{r}\left(t\right)\right\|^2-\left\|\mathbf{r}\left(t_1\right)\right\|^2=\sum_{u=t_1}^{t-1}\left[\left\|\mathbf{r}\left(u+1\right)\right\|^2-\left\|\mathbf{r}\left(u\right)\right\|^2\right]\le C+2\sum_{u=t_1}^{t-1}\left[C_1u^{-\theta}+C_2u^{-2}\right].$$

Therefore, $\|\mathbf{r}\left(t\right)\|$ is bounded, which proves our main theorem.

5 Limitations

Here we describe some of our work's limitations/possible future research directions. We note that these questions have been analyzed for the binary classification setting, but not for multiclass.

Non-ET losses In our paper we only analyze multiclass implicit bias for losses with the ET property. Another possible line of future work is to analyze the gradient descent dynamics for non-ET losses. Nacson et al. [2019] and Ji et al. [2020] prove that in the binary setting, ET and well-behaved super-polynomial tailed losses ensure convergence to the maximum-margin direction, while other losses may converge to a different direction with poor margin. Is such a characterization possible in the multiclass setting?

Other gradient-based methods This paper only analyzes vanilla gradient descent. Another line of work involves exploring implicit bias effects of other gradient-based methods, such as those characterized in Gunasekar et al. [2018]. Nacson et al. [2022] uses similar proof techniques to prove results for SGD, which is prevalent in practice and often generalizes better than vanilla GD ([Amir et al., 2021]).

Non-asymptotic analysis Our result proves that the gradient descent predictors *asymptotically* do not overfit. However, in the binary classification case, Shamir [2021] goes one step further and proves that for gradient-based methods, throughout the entire training process (not just asymptotically), both the empirical risk and the generalization error decrease at an essentially optimal rate (or remain optimally constant). Does the same phenomenon occur in the multiclass setting?

6 Conclusion

We use the permutation equivariant and relative margin-based (PERM) loss framework to provide an multiclass extension of the binary ET property. On a high level, while the binary ET bounds the negative derivative of the loss, our multiclass ET bounds each negative partial derivative of the PERM template ψ. We demonstrate our definition's validity for multinomial logistic loss, multiclass exponential loss, and PairLogLoss. We develop new techniques for analyzing multiclass gradient descent, and apply these to generalize binary implicit bias results to the multiclass setting. Our main result is that for almost all linearly separable multiclass datasets and a suitable ET PERM loss, the gradient descent iterates directionally converge towards the hard-margin multiclass SVM solution.

Our proof techniques in this paper demonstrate the power of the PERM framework to facilitate extensions of known binary results to multiclass settings and provide a unified treatment of both binary and multiclass classification. Thus it is possible that the binary results discussed in the Limitations section can also be extended using the PERM loss framework. In the future we would like to consider more complex settings that have been analyzed primarily for the binary case, such as non-separable data (Ji and Telgarsky [2019]) and two-layer neural nets (Lyu et al. [2021]).

Acknowledgments and Disclosure of Funding

CS was supported in part by the National Science Foundation under award 2008074, and by the Department of Defense, Defense Threat Reduction Agency under award HDTRA1-20-2-0002. The research of DS was funded by the European Union (ERC, A-B-C-Deep, 101039436). Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency (ERCEA). Neither the European Union nor the granting authority can be held responsible for them. DS also acknowledges the support of the Schmidt Career Advancement Chair in AI. YW was supported in part by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Futures program.

References

Idan Amir, Tomer Koren, and Roi Livni. Sgd generalizes better than gd (and regularization doesn't help), 2021.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Emmanuel J. Candès and Pragya Sur. The phase transition for the existence of the maximum likelihood estimate in high-dimensional logistic regression. *The Annals of Statistics*, 48(1):27–42, 2020.

Kenneth L Clarkson, Elad Hazan, and David P Woodruff. Sublinear optimization for machine learning. *Journal of the ACM (JACM)*, 59(5):1–49, 2012.

Andrew Cotter, Shai Shalev-Shwartz, and Nathan Srebro. The kernelized stochastic batch perceptron. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 739–746, 2012.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798. PMLR, 2019.

Ziwei Ji, Miroslav Dudik, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, pages 2109–2136. PMLR, 2020.

Ziwei Ji, Nathan Srebro, and Matus Telgarsky. Fast margin maximization via dual acceleration. In *International Conference on Machine Learning*, pages 4860–4869. PMLR, 2021.

CG Khatri and C Radhakrishna Rao. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 167–180, 1968.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, volume 6, 2015.

Shuangzhe Liu. Matrix results on the khatri-rao and tracy-singh products. *Linear Algebra and its Applications*, 289(1-3):267–277, 1999.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: Margin maximization and simplicity bias. *Advances in Neural Information Processing Systems*, 34:12978–12991, 2021.

Jan R Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.

Indraneel Mukherjee and Robert E Schapire. A theory of multiclass boosting. *Journal of Machine Learning Research*, 2013.

Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3420–3428. PMLR, 2019.

Mor Shpigel Nacson, Nathan Srebro, and Daniel Soudry. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate, 2022.

Robert R Phelps. *Convex functions, monotone operators and differentiability*, volume 1364. Springer, 2009.

Matan Schliserman and Tomer Koren. Stability vs implicit bias of gradient methods on separable data and beyond. In *Conference on Learning Theory*, pages 3380–3394. PMLR, 2022.

Matan Schliserman and Tomer Koren. Tight risk bounds for gradient descent on separable data. *arXiv preprint arXiv:2303.01135*, 2023.

Ohad Shamir. Gradient methods never overfit on separable data. *The Journal of Machine Learning Research*, 22(1):3847–3866, 2021.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1): 2822–2878, 2018. See `arxiv.org/abs/1710.10345v7` for the most up-to-date version.

Gal Vardi. On the implicit bias in deep-learning algorithms. *arXiv preprint arXiv:2208.12591*, 2022.

Ke Wang, Vidya Muthukumar, and Christos Thrampoulidis. Benign overfitting in multiclass classification: All roads lead to interpolation, 2023.

Nan Wang, Zhen Qin, Le Yan, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. Rank4class: A ranking formulation for multiclass classification, 2022.

Yutong Wang and Clayton Scott. Unified binary and multiclass margin-based classification. Accepted to *Journal of Machine Learning Research,* arXiv:2311.17778, 2024.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A Discussion of Lyu and Li [2019]

Lyu and Li [2019] allow the class score functions to be linear classifiers, e.g., $\mathbf{w}_k^\top \mathbf{x}_i$, but also nonlinear, e.g., "cubed" linear classifier $(\mathbf{w}_k^\top \mathbf{x}_i)^3$. By shifting the cubing operation to the loss, we can view the implicit regularization result of Lyu and Li [2019] as a result for losses beyond the cross entropy. This resulting loss is rather exotic and we are not aware of it being used in the literature; it is interesting nevertheless. However, the optimization problem would become non-convex, so convergence would not necessarily be to a global minimum:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } \left(\mathbf{w}_{y_i}^\top \mathbf{x}_i\right)^3 - \left(\mathbf{w}_k^\top \mathbf{x}_i\right)^3 \geq 1 \text{ for all } i \in [N], j \in [K]\backslash\{y_i\}$$

Moreover, the decision region for the k-th class, i.e., the set of $\mathbf{x} \in \mathbb{R}^d$ such that $(\mathbf{w}_k^\top \mathbf{x})^3 > (\mathbf{w}_j^\top \mathbf{x})^3$ for all $j \neq k$, is an intersection of sets constructed via cubic hypersurfaces.

More precisely, the k-th decision region can be written as

$$\{\mathbf{x} \in \mathbb{R}^d : (\mathbf{w}_k^\top \mathbf{x})^3 = \text{argmax}_{j\in[K]}(\mathbf{w}_j^\top \mathbf{x})^3\} = \bigcap_{j\in[K]:j\neq k} \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{w}_k^\top \mathbf{x})^3 > (\mathbf{w}_j^\top \mathbf{x})^3\}$$

Let us define $\mathcal{H}_j := \{\mathbf{x} \in \mathbb{R}^d : (\mathbf{w}_k^\top \mathbf{x})^3 = (\mathbf{w}_j^\top \mathbf{x})^3\}$. Note that $\mathcal{H}_j \subseteq \mathbb{R}^d$ is the zero set of degree 3 polynomials with variables in $\mathbf{x}$, hence, a cubic hypersurface. Now, the set $\{\mathbf{x} \in \mathbb{R}^d : (\mathbf{w}_k^\top \mathbf{x})^3 > (\mathbf{w}_j^\top \mathbf{x})^3\}$ is a subset of the set-theoretic complement of $\mathcal{H}_j$ in $\mathbb{R}^d$. Thus, the decision regions are complicated geometric objects, compared to the classical hard-margin SVM.

B Matrix Calculus

This section of the appendix establishes matrix identities that will be useful for us to calculate the gradient/Hessian of the empirical risk objective $\mathcal{R}(\mathbf{w})$.

Vector-input scalar-output function. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. Let $\mathbf{x} = [x_1, \cdots, x_n]^\top \in \mathbb{R}^n$ be a vector of variables for differentiation. Define the (column) vector of partial derivatives w.r.t. $\mathbf{x}$: $\partial_{\mathbf{x}} := \left[\frac{\partial}{\partial_{x_i}}\right]_{i\in[n]}$. The *gradient* of f, denoted ∇f, is the function

$$\nabla f : \mathbb{R}^n \to \mathbb{R}^n, \quad \text{where} \quad \nabla f(\mathbf{x}) = \partial_{\mathbf{x}} f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}(\mathbf{x}) \quad \cdots \quad \frac{\partial f}{\partial x_n}(\mathbf{x})\right]^\top. \tag{21}$$

Suppose that f is twice continuously differentiable. The *Hessian* of f, denoted $\nabla^2 f$, is the function

$$\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n\times n}, \quad \text{where} \quad \nabla^2 f(\mathbf{x}) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x})\right]_{i,j\in[n]}. \tag{22}$$

Matrix-input scalar-output function. Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be a differentiable function. Let $\mathbf{X} = [x_{ij}]_{i\in[m],j\in[n]} \in \mathbb{R}^{m\times n}$ be an arbitrary matrix. Define the matrix of partial derivatives w.r.t. $\mathbf{X}$:

$$\partial_{\mathbf{X}} := \left[\frac{\partial}{\partial_{x_{ij}}}\right]_{i\in[m],j\in[n]}$$

Define the gradient of f, denoted ∇f, to be the function

$$\nabla f : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}, \quad \text{where} \quad \nabla(f)(\mathbf{X}) := \partial_{\mathbf{X}} f(\mathbf{X}) = \left[\frac{\partial}{\partial_{x_{ij}}} f(\mathbf{X})\right]_{i\in[m],j\in[n]}. \tag{23}$$

We do not define the Hessian of a matrix-input scalar-output function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$. Instead, we will define the Hessian for its *vectorization* $\mathsf{vec}(f) : \mathbb{R}^{mn} \to \mathbb{R}$.

Definition B.1 (Vectorization operator)**.** *Let* vec *denote the vectorization operator by stacking the columns of a vector. In other words, if* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *is a matrix with columns* $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^m$*, then*

$$\mathsf{vec}(\mathbf{A}) := \begin{bmatrix}\mathbf{a}_1^\top & \mathbf{a}_2^\top & \cdots & \mathbf{a}_n^\top\end{bmatrix}^\top.$$

Definition B.2 (Vectorization of a matrix-input function)**.** *Let* $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ *be a matrix-input function, we define* $\mathsf{vec}(f) : \mathbb{R}^{mn} \to \mathbb{R}$ *to be the vector-input function such that*

$$f(\mathbf{A}) = \mathsf{vec}(f)(\mathsf{vec}(\mathbf{A})).$$

In particular, if f *is already a vector-input function, then* $\mathsf{vec}(f) = f$*.*

See [Magnus and Neudecker, 2019, Ch.5-§15]. Below in Lemma B.4, we give a convenient formula to calculate the Hessian of $\mathcal{R}(\mathbf{w})$, the vectorization of $\mathcal{R}(\mathbf{W})$.

The following relates the vectorization operator with the Kronecker product:

Lemma B.1. *Let* $\mathbf{A} \in \mathbb{R}^{p\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}, \mathbf{C} \in \mathbb{R}^{m\times q}$ *be matrices. Then*

$$\mathsf{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\mathsf{vec}(\mathbf{B}).$$

Proof. This is [Magnus and Neudecker, 2019, Theorem 2.2]. □

B.1 Special case of the chain rule for linear functions

Proposition B.2. *Let* $\mathbf{M} \in \mathbb{R}^{m\times n}$ *be a matrix. Let* $f : \mathbb{R}^m \to \mathbb{R}$ *be a continuously differentiable function and define* $g : \mathbb{R}^n \to \mathbb{R}$ *by* $g(\mathbf{x}) := f(\mathbf{Mx})$*. Then*

$$\nabla g(\mathbf{x}) = \mathbf{M}^\top \nabla f(\mathbf{Mx}), \quad \textit{and} \quad \nabla^2 g(\mathbf{x}) = \mathbf{M}^\top \nabla^2 f(\mathbf{Mx})\mathbf{M}.$$

Proof. See [Magnus and Neudecker, 2019, Ch.9-§13] for the first identity and [Magnus and Neudecker, 2019, Ch.10-§8] for the second identity. □

The next two results will be referred to as the "gradient formula" and the "Hessian formula", respectively, for the function $g(\mathbf{X}) := f(\mathbf{AX}^\top\mathbf{B})$.

Lemma B.3. *Let* $f : \mathbb{R}^{p\times q} \to \mathbb{R}$ *be a matrix-input scalar-output differentiable function with Jacobian denoted* $\nabla f : \mathbb{R}^{p\times q} \to \mathbb{R}^{p\times q}$*. Let* $\mathbf{A} \in \mathbb{R}^{p\times n}$*,* $\mathbf{X} \in \mathbb{R}^{m\times n}$*, and* $\mathbf{B} \in \mathbb{R}^{m\times q}$*. Define a function* $g : \mathbb{R}^{m\times n} \to \mathbb{R}$ *by* $g(\mathbf{X}) := f(\mathbf{AX}^\top\mathbf{B})$*. Then*

$$\nabla g(\mathbf{X}) = \partial_{\mathbf{X}} f(\mathbf{AX}^\top\mathbf{B}) = \mathbf{B}\nabla f(\mathbf{AX}^\top\mathbf{B})^\top\mathbf{A}.$$

Lemma B.4. *Let* $f : \mathbb{R}^p \to \mathbb{R}$ *be a vector-input scalar-output twice differentiable function. Let* $\mathbf{A} \in \mathbb{R}^{p\times n}, \mathbf{X} \in \mathbb{R}^{m\times n}$ *be matrices and* $\mathbf{b} \in \mathbb{R}^m$ *be a (column) vector. Let* $\mathbf{V}$ *be another matrix with the same shape as* $\mathbf{X}$*. Let* $\mathbf{x} := \mathsf{vec}(\mathbf{X})$ *and* $\mathbf{v} := \mathsf{vec}(\mathbf{V})$*. Define* $g(\mathbf{X}) := f(\mathbf{AX}^\top\mathbf{b})$ *and let* $\overline{g} = \mathsf{vec}(g)$ *be the vectorization of* g*. Then we have the following formula for computing* $\mathbf{v}^\top\nabla^2\overline{g}(\mathbf{x})\mathbf{v}$*:*

$$\mathbf{v}^\top\nabla^2\overline{g}(\mathbf{x})\mathbf{v} = (\mathbf{AV}^\top\mathbf{b})^\top\nabla^2 f(\mathbf{AX}^\top\mathbf{b})\mathbf{AV}^\top\mathbf{b}^\top.$$

B.2 Proof of the gradient formula: Lemma 4.2

In the notation of Section 2.8.1 of the Matrix Cookbook, define matrix $\mathbf{U} \in \mathbb{R}^{p\times q}$ by $\mathbf{U} := \mathbf{A}\mathbf{X}^\top\mathbf{B}$. Note that $\mathbf{U}$ is a function of $\mathbf{X}$. Then by Eqn. (137) of the Matrix Cookbook, we have for each $(i,j) \in [m] \times [n]$

$$\frac{\partial}{\partial X_{ij}} f(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = \frac{\partial}{\partial X_{ij}} f(\mathbf{U}) = \mathrm{Tr}\left[\left(\frac{\partial f(\mathbf{U})}{\partial \mathbf{U}}\right)^\top \frac{\partial \mathbf{U}}{\partial X_{ij}}\right]$$

Note that by definition, we have $\frac{\partial f(\mathbf{U})}{\partial \mathbf{U}} = \nabla f(\mathbf{U})$. Therefore

$$\frac{\partial}{\partial X_{ij}} f(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = \mathrm{Tr}\left[\nabla f(\mathbf{U})^\top \frac{\partial \mathbf{U}}{\partial X_{ij}}\right]$$

Next, write $\mathbf{U} = [U_{k\ell}]_{k\in[p],\ell\in[q]}$ in the "matrix-comprehension" notation. Recall that $U_{k\ell}$, i.e., the (k,ℓ)-th entry of $\mathbf{U}$, is precisely computed by $\mathbf{A}[k,:](\mathbf{X}^\top\mathbf{B})[:,\ell] = \mathbf{A}[k,:]\mathbf{X}^\top\mathbf{B}[:,\ell]$. For each $k,\ell \in [p]\times[q]$, we have

$$\frac{\partial U_{k\ell}}{\partial X_{ij}} = \frac{\partial(\mathbf{A}[k,:]\mathbf{X}^\top\mathbf{B}[:,\ell])}{\partial X_{ij}}$$

where "$[k,:]$" and " $[:,\ell]$ " denote taking the k-th row vector and ℓ-th column vector, respectively. Now, by Eqn. (71) of the Matrix Cookbook, we have the following expression of the matrix-partial derivative as an outer product

$$\frac{\partial U_{k\ell}}{\partial \mathbf{X}} = \frac{\partial \mathbf{A}[k,:]\mathbf{X}^\top\mathbf{B}[:,\ell]}{\partial \mathbf{X}} = \mathbf{B}[:,\ell]\mathbf{A}[k,:].$$

From this, it follows that computing the entry-wise partial derivative at X_{ij} is simply obtained by indexing at (i,j), i.e., $\frac{\partial U_{k\ell}}{\partial X_{ij}} = \mathbf{B}[i,\ell]\mathbf{A}[k,j] = \mathbf{A}[k,j]\mathbf{B}[i,\ell]$ (we emphasize that this is just a product of two scalars). Thus, $\frac{\partial \mathbf{U}}{\partial X_{ij}} = \mathbf{A}[:,j]\mathbf{B}[i,:]$. Consequently,

$$\frac{\partial}{\partial X_{ij}} f(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = \mathrm{Tr}\left[\nabla f(\mathbf{U})^\top\mathbf{A}[:,j]\mathbf{B}[i,:]\right] = \mathbf{B}[i,:]\nabla f(\mathbf{U})^\top\mathbf{A}[:,j].$$

In other words, $\frac{\partial}{\partial \mathbf{X}} f(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = \mathbf{B}\nabla f(\mathbf{U})^\top\mathbf{A}$.

For our purposes, we replace f with ψ, $\mathbf{A}$ with $\boldsymbol{\Upsilon}_{y_i}\mathbf{D}$, $\mathbf{X}$ with $\mathbf{W}$, and $\mathbf{B}$ with $\mathbf{x}_i$. Thus we obtain

$$\nabla\mathcal{R}(\mathbf{W}) = \sum_{i=1}^N \frac{\partial}{\partial \mathbf{W}} \psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right) = \sum_{i=1}^N \mathbf{x}_i \nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right)^\top \boldsymbol{\Upsilon}_{y_i}\mathbf{D}$$

as desired. □

B.3 Proof of Hessian formula: Lemma B.4

Our goal is to calculate the Hessian of $\mathsf{vec}(g)$. First, we note that by definition

$$\mathsf{vec}(g)(\mathbf{x}) = \mathsf{vec}(g)(\mathsf{vec}(\mathbf{X})) = \mathsf{vec}(f)(\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{b}))$$

Note that the last equality is simply $\mathsf{vec}(f)(\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{b})) = f(\mathbf{A}\mathbf{X}^\top\mathbf{b})$, but we work in the more general case of a matrix $\mathbf{B}$ right now. We will need to simplify $\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{b})$. It is more convenient during the first phase of the proof viewing $\mathbf{b}$ as a $m \times 1$ matrix and denote it using uppercase letter $\mathbf{B}$. First, applying Lemma B.1 to $\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B})$, we get

$$\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathsf{vec}(\mathbf{X}^\top)$$

However, $\mathsf{vec}(\mathbf{X}^\top) \neq \mathsf{vec}(\mathbf{X})$ in general. However, these two expressions are related using the *commutation matrix*:

Definition B.3 (Commutation matrix)**.** *Define $\mathbf{K}_{m,n}$ to be the permutation matrix in $\mathbb{R}^{mn\times mn}$ such that $\mathbf{K}_{m,n}\mathsf{vec}(\mathbf{A}) = \mathsf{vec}(\mathbf{A}^\top)$ for all matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$.*

See [Magnus and Neudecker, 2019, Ch.3-§7]. Below, we drop the subscripts in Definition B.3 and simply write $\mathbf{K} := \mathbf{K}_{m,n}$. Now, we have

$$\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathsf{vec}(\mathbf{X}) = (\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{x}$$

Thus

$$\mathsf{vec}(g)(\mathbf{x}) = \mathsf{vec}(g)(\mathsf{vec}(\mathbf{X})) = \mathsf{vec}(f)((\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{x})$$

By Proposition B.2, we have

$$\begin{aligned}
&\nabla^2\mathsf{vec}(g)(\mathbf{x}) \\
&= ((\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top)^\top\nabla^2\mathsf{vec}(f)((\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{x})(\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top \\
&= ((\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top)^\top\nabla^2\mathsf{vec}(f)(\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B}))(\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top.
\end{aligned}$$

From this, we see that (recall that $\mathbf{v} = \mathsf{vec}(\mathbf{V})$)

$$\mathbf{v}^\top\nabla^2\mathsf{vec}(g)(\mathbf{x})\mathbf{v} = ((\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{v})^\top\nabla^2\mathsf{vec}(f)(\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B}))(\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{v}.$$

Now, by Lemma B.1, we have

$$(\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathbf{v} = (\mathbf{B}^\top \otimes \mathbf{A})\mathbf{K}^\top\mathsf{vec}(\mathbf{V}) = \mathsf{vec}(\mathbf{A}\mathbf{V}^\top\mathbf{B}).$$

Now, since $\mathbf{B} = \mathbf{b}$ is just a vector, we have

$$\mathsf{vec}(\mathbf{A}\mathbf{V}^\top\mathbf{B}) = \mathbf{A}\mathbf{V}^\top\mathbf{b}.$$

and

$$\nabla^2\mathsf{vec}(f)(\mathsf{vec}(\mathbf{A}\mathbf{X}^\top\mathbf{B})) = \nabla^2 f(\mathbf{A}\mathbf{X}^\top\mathbf{b})$$

Putting it all together, we get the desired equality. □

C PERM Losses That Satisfy Assumptions 3.1 and 3.2

C.1 Cross-Entropy

By Wang and Scott [2024, Example 1], the cross-entropy loss $\mathcal{L}_y(\mathbf{v}) = -\log\left(\frac{\exp(v_y)}{\sum_{k=1}^K \exp(v_k)}\right)$ has template $\psi(\mathbf{u}) = \log\left(1 + \sum_{k=1}^{K-1}\exp(-u_k)\right)$. We calculate the partial derivatives:

$$\begin{aligned}
&\frac{\partial\psi}{\partial u_i}(\mathbf{u}) \\
&= -\frac{\exp(-u_i)}{1 + \sum_{k=1}^{K-1}\exp(-u_k)} && (24) \\
&= -\frac{1}{1 + (C_i+1)\exp(u_i)} \quad \text{where } C_i = \textstyle\sum_{k\in[K-1]:k\neq i}\exp(-u_k). && (25)
\end{aligned}$$

C.1.1 Convexity

Let us analyze the entries of the Hessian of the template, i.e. $\nabla^2\psi(\mathbf{u})$. Let $[\mathbf{A}]_{l,m}$ denote the element of $\mathbf{A}$ at the l-th row and m-th column. We get for all $i, j \in [K-1]$ where $j \neq i$:

$$\begin{aligned}
\left[\nabla^2\psi(\mathbf{u})\right]_{i,i} &= \frac{\partial^2\psi(\mathbf{u})}{\partial u_i^2} = \frac{(C_i+1)\,e^{-u_i}}{\left(1 + \sum_{k=1}^{K-1} e^{-u_k}\right)^2} \\
\left[\nabla^2\psi(\mathbf{u})\right]_{i,j} &= \left[\nabla^2\psi(\mathbf{u})\right]_{j,i} = \frac{\partial^2\psi(\mathbf{u})}{\partial u_i u_j} = \frac{-e^{-u_i-u_j}}{\left(1 + \sum_{k=1}^{K-1} e^{-u_k}\right)^2}
\end{aligned}$$

From the definition of C_i, this implies that:

$$\left[\nabla^2\psi(\mathbf{u})\right]_{i,i} = \sum_{j\in[K-1],j\neq i}\left|\left[\nabla^2\psi(\mathbf{u})\right]_{i,j}\right| + \frac{e^{-u_i}}{\left(1 + \sum_{k=1}^{K-1} e^{-u_k}\right)^2}$$

Thus, the Hessian is a symmetric *diagonally dominant* matrix, and hence is positive semi-definite.

C.1.2 β-smoothness

For any diagonally dominant matrix $\mathbf{B}$, let $|\mathbf{B}|$ be the matrix obtained by taking the absolute value of each element of $\mathbf{B}$, that is:

$$[|\mathbf{B}|]_{l,m} = \left|[\mathbf{B}]_{l,m}\right| \quad \text{for all } l, m.$$

Additionally, let $\mathsf{diag}(\cdot) : \mathbb{R}^p \to \mathbb{R}^{p\times p}$ (for any $p \in \mathbb{N}$) be the function that maps a vector to a diagonal matrix in the obvious way.

Then we have the following lemma:

Lemma C.1. *Let* $\mathbf{B}' := \mathsf{diag}\left(|\mathbf{B}|\,\mathbf{1}\right)$ *where* $\mathbf{1}$ *is the appropriately-sized vector of all-1's. Then* $\mathbf{B} \preceq \mathbf{B}'$.

Proof. This can be proven simply by observing that $\mathbf{B}' - \mathbf{B}$ is symmetric and diagonally dominant (and thus positive semi-definite). □

Lemma C.1 can be directly applied to analyze the Hessian (and eventually bound its maximum eigenvalue). Define $\mathbf{H}' = \mathsf{diag}\left(\left|\nabla^2\psi\left(\mathbf{u}\right)\right|\mathbf{1}\right)$. In other words, from Eqn. (25):

$$[\mathbf{H}']_{i,i} = \sum_{k=1}^{K-1}\left|\nabla^2\psi\left(\mathbf{u}\right)\right|_{i,k} = \frac{\left(2C_i+1\right)e^{-u_i}}{\left(1+\sum_{k=1}^{K-1}e^{-u_k}\right)^2} \tag{26}$$

$$[\mathbf{H}']_{i,j} = 0$$

Thus, by directly applying Lemma C.1, we obtain

$$\nabla^2\psi\left(\mathbf{u}\right) \preceq \mathbf{H}'$$

So now since $\mathbf{H}'$ is defined to be a diagonal matrix, all that's left to do is bound the diagonal entries by a positive constant. First note that from the definition of C_i, it follows that $1+\sum_{k=1}^{K-1}e^{-u_k} = C_i + 1 + e^{-u_i}$. Combining this with Eqn. (26), we get:

$$\frac{\left(2C_i+1\right)e^{-u_i}}{\left(\left(C_i+1\right)+e^{-u_i}\right)^2} = \frac{\left(2C_i+1\right)}{\left(\left(C_i+1\right)+e^{-u_i}\right)\left(\left(C_i+1\right)e^{u_i}+1\right)}$$

We can find a global minimum of the denominator of the above expression and thus arrive at an upper bound for the expression. Differentiating with respect to u_i and setting to 0 yields a single critical point at $u_i = -\log\left(C_i+1\right)$, which produces a value of $4\left(C_i+1\right)$ when substituted in the denominator (this is a global minimum of the denominator expression). Thus, we get

$$[\mathbf{H}']_{i,i} \leq \frac{\left(2C_i+1\right)}{4\left(C_i+1\right)} = \frac{1}{2} - \frac{1}{4\left(C_i+1\right)}$$

In the binary i.e. $K=2$ case, $C_i = 0$, so our bound is exactly $1/4$. However, in the multiclass case (i.e. $K > 2$), C_i can be arbitrarily large. Setting $C_i = \infty$ yields a final upper bound of $1/2$.

So our final bound can be summarized as follows:

$$\|\nabla^2\psi\left(\mathbf{u}\right)\|_2 \leq [\mathbf{H}']_{i,i} \leq \begin{cases} 1/4 & \text{, if } K=2 \\ 1/2 & \text{, if } K>2. \end{cases}$$

Thus, $\beta = 1/4$ for binary cross-entropy (logistic loss), but $\beta = 1/2$ for K-class cross-entropy.

C.1.3 Exponential Tail

We claim that for the cross-entropy Definition 2.2 holds with $u_\pm = 0$ and $c = a = 1$. We are interested in analyzing the (negative) gradient of the template. From Eqn. 24:

$$\begin{aligned} -\frac{\partial\psi}{\partial u_i}\left(\mathbf{u}\right) &= \frac{e^{-u_i}}{1+\sum_{k=1}^{K-1}e^{-u_k}} \\ &\leq e^{-u_i} \\ &\geq e^{-u_i}\left(1-\sum_{k=1}^{K-1}e^{-u_k}\right) \quad \because \forall x \geq 0, \tfrac{1}{1+x} \geq 1-x \end{aligned}$$

This proves that the cross-entropy loss satisfies Definition 2.2 with $u_\pm = 0$ and $c = 1$.

C.2 Multiclass Exponential Loss [Mukherjee and Schapire, 2013]

The multiclass exponential loss $\mathcal{L} : \mathbb{R}^K \to \mathbb{R}$ can be written as $\mathcal{L}_y(\mathbf{v}) = \sum_{k\in[K]:k\neq y} \exp(-(v_y - v_k))$. Thus, the template function $\psi : \mathbb{R}^{K-1} \to \mathbb{R}$ can be expressed as $\psi(\mathbf{u}) = \sum_{i\in[K-1]} e^{-u_i}$. The partial derivatives of the template are then simply:

$$\frac{\partial}{\partial u_i}\psi(\mathbf{u}) = -e^{-u_i} \quad \text{for all } i \in [K-1]. \tag{27}$$

C.2.1 Convexity

We have

$$\nabla^2\psi(\mathbf{u}) = \mathsf{diag}(\exp(-u_i) : i = 1, \dots, K-1). \tag{28}$$

The Hessian is a diagonal matrix with all diagonal entries positive. Hence it is positive definite.

C.2.2 β-"smoothness"

Recall the identity derived in Lemma B.4:

$$\mathbf{v}^\top\nabla^2\overline{g}(\mathbf{x})\mathbf{v} = (\mathbf{A}\mathbf{V}^\top\mathbf{b})^\top\nabla^2 f(\mathbf{A}\mathbf{X}^\top\mathbf{b})\mathbf{A}\mathbf{V}^\top\mathbf{b}^\top.$$

We are interested in the special case where $\mathbf{X} \leftarrow \mathbf{W}$ is a linear classifier (represented as a matrix) and $\mathbf{x} \leftarrow \mathsf{vec}(\mathbf{W}) = \mathbf{w}$ is its vectorization as in Section 3. Moreover, $g(\mathbf{X})$ represents the risk $\mathcal{R}(\mathbf{W})$, viewed as a *matrix-input scalar-output* function (defined in Appendix B), while $\overline{g}(\mathbf{x})$ represents the *vectorized* risk $\mathcal{R}(\mathbf{w})$, viewed as a *vector-input scalar-output* function (defined in Appendix B). We will use the formula in Lemma B.4 to calculate $\mathbf{v}^\top\nabla^2\mathcal{R}(\mathbf{w})\mathbf{v}$, where we substitute in

$$\mathbf{A} \leftarrow \mathbf{\Upsilon}_{y_i}\mathbf{D} \in \mathbb{R}^{(K-1)\times K}, \qquad \mathbf{b} \leftarrow \mathbf{x}_i \in \mathbb{R}^d, \qquad f \leftarrow \psi.$$

where $(\mathbf{x}_i, y_i)$ is a training sample and ψ is the template of a PERM loss. Since ∇^2 is linear (i.e., distributive over additions), we have by Lemma B.4 that

$$\begin{aligned}
&\mathbf{v}^\top\nabla^2\mathcal{R}(\mathbf{w})\mathbf{v} = \mathsf{vec}(\mathbf{V})^\top\nabla^2\mathcal{R}(\mathbf{w})\mathsf{vec}(\mathbf{V}) \\
&= \sum_{i=1}^N (\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{V}\mathbf{x}_i)^\top\nabla^2\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}\mathbf{x}_i)\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{V}\mathbf{x}_i.
\end{aligned}$$

Note that $\|\mathbf{v}\| = \|\mathsf{vec}(\mathbf{V})\| = \|\mathbf{V}\|_F$ by the definitions of the Frobenius norm and vectorization. Thus,

$$\max_{\mathbf{v}\in\mathbb{R}^{dK}:\|\mathbf{v}\|=1} \mathbf{v}^\top\nabla^2\mathcal{R}(\mathbf{w})\mathbf{v} = \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \mathsf{vec}(\mathbf{V})^\top\nabla^2\mathcal{R}(\mathbf{w})\mathsf{vec}(\mathbf{V}). \tag{29}$$

We note that we never defined the Hessian of a *matrix*-input function, i.e., we do not work with $\nabla^2\mathcal{R}(\mathbf{W})$. Combining the two previous identities, we have proven

Corollary C.2. *Let $\mathcal{R}(\mathbf{W})$ be the risk viewed as a matrix-input scalar-output function defined in Equation* (3)*. Let $\mathcal{R}(\mathbf{w})$ be the vectorization of $\mathcal{R}(\mathbf{W})$. Then we have*

$$\begin{aligned}
&\max_{\mathbf{v}\in\mathbb{R}^{dK}:\|\mathbf{v}\|=1} \mathbf{v}^\top\nabla^2\mathcal{R}(\mathbf{w})\mathbf{v} \\
&= \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \sum_{i=1}^N (\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{V}\mathbf{x}_i)^\top\nabla^2\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}\mathbf{x}_i)\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{V}\mathbf{x}_i.
\end{aligned}$$

We have

$$\nabla^2\psi(\mathbf{u}) = \mathsf{diag}(\exp(-u_i) : i = 1, \dots, K-1). \tag{30}$$

Let $\mathsf{vdiag}(\cdot)$ be the "inverse" of $\mathsf{diag}(\cdot)$, i.e., $\mathsf{vdiag}(\cdot)$ takes a diagonal matrix and returns the vector of the diagonal elements.

The max eigenvalue of the Hessian of $\mathcal{R}(\mathbf{w})$, i.e., Equation (29), is computed below:

$$
\begin{aligned}
&\max_{\mathbf{v}\in\mathbb{R}^{dK}:\|\mathbf{v}\|=1} \mathbf{v}^\top \nabla^2 \mathcal{R}(\mathbf{w})\mathbf{v} \\
&\overset{1}{=} \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \sum_{i=1}^N (\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^\top \nabla^2\psi(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i)\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i \quad \because \text{ Corollary C.2} \\
&\overset{2}{=} \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \sum_{i=1}^N \mathsf{tr}\Big(\nabla^2\psi(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i)\ \underbrace{\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^\top}_{(K-1)\text{-by-}(K-1)\text{ outer product}}\Big) \\
&\overset{3}{=} \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \sum_{i=1}^N \underbrace{\mathsf{vdiag}(\nabla^2\psi(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i))^\top}_{\text{Vector of diagonal elements of }\nabla^2\psi}\underbrace{(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^{\odot 2}}_{\text{entrywise square}} \\
&\leq \max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1} \sum_{i=1}^N \mathcal{R}(\mathbf{W})\mathbf{1}^\top(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^{\odot 2} \quad \because \text{ replace each entry of vdiag with risk}
\end{aligned}
$$

In equality 2 we took trace of a scalar (the expression in equality 1 is a scalar, so taking the trace of it will not change the value) and used the cyclic property. For equality 3: as per Equation (28), $\nabla^2\psi$ is a diagonal matrix. Finally, in the last inequality, we bound each element of the diagonal vector (i.e. $\exp(-u_i)$ for all $i \in [K-1]$). Dropping the $\mathcal{R}(\mathbf{W})$ and "$\max_{\mathbf{V}\in\mathbb{R}^{d\times K}:\|\mathbf{V}\|_F=1}$" from the front:

$$
\begin{aligned}
&\sum_{i=1}^N \mathbf{1}^\top(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^{\odot 2} \\
&= \sum_{i=1}^N (\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i = \sum_{i=1}^N \mathsf{tr}\left((\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i)^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i\right) \\
&= \sum_{i=1}^N \mathsf{tr}\Big(\mathbf{D}^\top\boldsymbol{\Upsilon}_{y_i}^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{V}^\top\mathbf{x}_i\mathbf{x}_i^\top\mathbf{V}\Big) \quad \text{Note that } \mathbf{V}^\top\mathbf{x}_i \in \mathbb{R}^K \\
&\leq \sum_{i=1}^N \|\mathbf{D}^\top\boldsymbol{\Upsilon}_{y_i}^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\|_F\|\mathbf{V}^\top\mathbf{x}_i\mathbf{x}_i^\top\mathbf{V}\|_F \quad \because \text{Cauchy-Schwarz inequality}
\end{aligned}
$$

Note that we applied Cauchy-Schwarz to the inner product space $\mathbb{R}^{K\times K}$ with inner product $\langle\mathbf{A},\mathbf{B}\rangle := \mathsf{tr}(\mathbf{A}^\top\mathbf{B})$. Now, continuing with the calculation:

$$
\begin{aligned}
&\sum_{i=1}^N \|\mathbf{D}^\top\boldsymbol{\Upsilon}_{y_i}^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\|_F\|\mathbf{V}^\top\mathbf{x}_i\mathbf{x}_i^\top\mathbf{V}\|_F \\
&\leq \sum_{i=1}^N \|\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\|_F^2\|\mathbf{V}^\top\|_F\|\mathbf{x}_i\mathbf{x}_i^\top\|_F\|\mathbf{V}\|_F \quad \because \|\mathbf{A}\mathbf{B}\|_F \leq \|\mathbf{A}\|_F\|\mathbf{B}\|_F \\
&\leq (2K-2)\sum_{i=1}^N \|\mathbf{x}_i\|^2 \quad \because \text{Lemma C.3, } \|\mathbf{x}_i\mathbf{x}_i^\top\|_F = \|\mathbf{x}_i\|^2,\ \|\mathbf{V}\|_F = 1 \\
&\leq (2K-2)\left(\sum_{i=1}^N \|\mathbf{x}_i\|\right)^2 \quad \because \text{for all } a_j > 0,\ M \geq 1,\ \textstyle\sum_{i=1}^M a_j^2 \leq \left(\sum_{i=1}^M a_j\right)^2
\end{aligned}
$$

Therefore we have proven that $\|\nabla^2\mathcal{R}(\mathbf{w})\|_2 \leq B^2\mathcal{R}(\mathbf{w})$, where

$$
B = \sqrt{(2K-2)}\sum_{i=1}^N \|\mathbf{x}_i\|. \tag{31}
$$

Now we will also analyze the Euclidean norm of the gradient, for reasons that will become clear later.

$$
\begin{aligned}
&\|\nabla\mathcal{R}(\mathbf{w})\| = \|\nabla\mathcal{R}(\mathbf{W})\|_F \\
&= \left\| \sum_{i=1}^{N} \mathbf{x}_i \nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right)^\top \mathbf{\Upsilon}_{y_i}\mathbf{D} \right\|_F \\
&\leq \sum_{i=1}^{N} \left\| \mathbf{x}_i \nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right)^\top \mathbf{\Upsilon}_{y_i}\mathbf{D} \right\|_F \quad \because \text{triangle inequality} \\
&\leq \sum_{i=1}^{N} \|\mathbf{x}_i\|_2 \left\|\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right)\right\|_2 \|\mathbf{\Upsilon}_{y_i}\mathbf{D}\|_F \quad \because \|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F\|\mathbf{B}\|_F \\
&= \sqrt{(2K-2)}\sum_{i=1}^{N} \|\mathbf{x}_i\|_2 \left\|\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i\right)\right\|_2 \quad \because \text{Lemma C.3} \\
&\leq \sqrt{(2K-2)}\sum_{i=1}^{N} \|\mathbf{x}_i\| \mathcal{R}(\mathbf{w}) \quad \because \text{for all } a_j > 0,\ M \geq 1,\ \sqrt{\textstyle\sum_{i=1}^{M} a_j^2} \leq \textstyle\sum_{i=1}^{M} a_j
\end{aligned}
$$

Gradient descent is a special case of steepest descent with the Euclidean norm [Boyd and Vandenberghe, 2004]. Thus, we can apply Gunasekar et al. [2018, Lemmas 11 & 12] to see that $\nabla\mathcal{R}(\mathbf{w}) \to 0$ even for multiclass exponential loss. Elaborating on this: these lemmas from Gunasekar et al. [2018] assume a convex risk objective (which we have in the case of multiclass exponential loss). Additionally, they assume that $\|\nabla\mathcal{R}(\mathbf{w})\| \leq B\mathcal{R}(\mathbf{w})$ and $\left\|\nabla^2\mathcal{R}(\mathbf{w})\right\|_2 \leq B^2\mathcal{R}(\mathbf{w})$. In the above section we prove these exact results with B as defined in Eqn. 31. Finally, Lemma C.3 below proves that $\|\mathbf{\Upsilon}_k\mathbf{D}\|_F = \sqrt{2K-2}$ for all $k \in [K-1]$.

In conclusion, by Gunasekar et al. [2018, Lemma 11], if our learning rate $\eta < \frac{1}{B^2\mathcal{R}(\mathbf{w}(0))}$, we can use Soudry et al. [2018, Lemma 10].

Finally, we calculate $\|\mathbf{\Upsilon}_k\mathbf{D}\|_F$ which was used in several places above:

Lemma C.3. *For each $k \in [K]$, we have $\|\mathbf{\Upsilon}_k\mathbf{D}\|_F = \sqrt{2(K-1)}$.*

Proof. First, if $k = K$, then $\mathbf{\Upsilon}_K$ is the identity matrix. In this case, we have $\mathbf{\Upsilon}_k\mathbf{D} = \mathbf{D} = [-\mathbf{I}_{K-1} \quad \mathbf{1}_{K-1}]$ is the negative $(K-1)$-by-$(K-1)$ identity matrix concatenated with the all-ones vector. Thus,

$$\|\mathbf{\Upsilon}_k\mathbf{D}\|_F^2 = \|\mathbf{D}\|_F^2 = \|\mathbf{I}_{K-1}\|_F^2 + \|\mathbf{1}_{K-1}\|^2 = 2(K-1)$$

If $k \neq K$, then

$$\mathbf{\Upsilon}_k\mathbf{D} = [-\mathbf{\Upsilon}_k \quad \mathbf{\Upsilon}_k\mathbf{1}_{K-1}]$$

and so

$$\|\mathbf{\Upsilon}_k\mathbf{D}\|_F^2 = \|\mathbf{\Upsilon}_k\|_F^2 + \|\mathbf{\Upsilon}_k\mathbf{1}_{K-1}\|^2$$

Now, we recall from the definition of $\mathbf{\Upsilon}_k$ (Definition 2.4 of Wang and Scott [2024]) that $\mathbf{\Upsilon}_k$ is obtained by replacing the k-th column of the identity matrix by the "all-negative-ones" vector. Thus

$$\|\mathbf{\Upsilon}_k\|_F^2 = 2(K-1) - 1, \quad \text{and} \quad \|\mathbf{\Upsilon}_k\mathbf{1}_{K-1}\|_F^2 = \|-\mathbf{e}_k\|_F^2 = 1.$$

This proves Lemma C.3, as desired. □

C.2.3 Exponential Tail

From Eqn. (27), the negative partial derivative of the template is clearly always positive:

$$-\frac{\partial}{\partial u_i}\psi(\mathbf{u}) = e^{-u_i} \geq 0.$$

From the above, it is clear that the upper and lower bounds in Definition 2.2 hold when $u_\pm = 0$ and $c = 1$.

C.3 PairLogLoss [Wang et al., 2022]

Recall that the template of the PairLogLoss is $\psi(\mathbf{u}) = \sum_{k=1}^{K-1} \log(1 + \exp(-u_k))$. By elementary calculus, we see that

$$\frac{\partial \psi(u)}{\partial u_k} = -\frac{e^{-u_k}}{1 + e^{-u_k}}.$$

C.3.1 Convexity

We have

$$\nabla^2 \psi(\mathbf{u}) = \mathsf{diag}\left(e^{u_i}/\left(1 + e^{u_i}\right)^2 : i = 1, \ldots, K-1\right).$$

The Hessian is a diagonal matrix with all diagonal entries positive. Hence it is positive definite.

C.3.2 β-smoothness

Notice that the partial derivative of the template is exactly the same expression as the derivative of the logistic loss (i.e. binary cross-entropy). Thus, the exact same proof as logistic loss can be used to prove β-smoothness for the PairLogLoss as well. Thus, from Appendix C.1.2, $\beta = 1/4$ (logistic loss is simply the $K = 2$ case for cross-entropy).

C.3.3 Exponential Tail

$$-\frac{\partial \psi(u)}{\partial u_k} = \frac{e^{-u_k}}{1 + e^{-u_k}} \leq e^{-u_k}$$

This gives us the desired upper tail. As for the lower tail:

$$\begin{aligned}
-\frac{\partial \psi(u)}{\partial u_k} &= \frac{e^{-u_k}}{1 + e^{-u_k}} \\
&\geq e^{-u_k}\left(1 - e^{-u_k}\right) \quad \because \tfrac{1}{1+x} \geq 1 - x \text{ for all } x \geq 0 \\
&\geq e^{-u_k}\left(1 - \sum_{i=1}^{K-1} e^{-u_i}\right)
\end{aligned}$$

Thus, PairLogLoss satisfies Definition 2.2 with $u_\pm = 0$ and $c = a = 1$.

D Pseudo-index

Note that $\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i$ produces a $(K-1)$-dimensional vector with entries of the form of $(\mathbf{r}_{y_i}(t) - \mathbf{r}_k(t))^\top \mathbf{x}_i, \forall k \in [K] \backslash \{y_i\}$. For $k \in [K] \backslash \{y_i\}$, let us represent the corresponding entry of the vector as $[\![\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i]\!]_k$. Note that this indexing is not the same as the k^{th} entry of the vectors, since the ${y_i}^{th}$ entry $(\mathbf{r}_{y_i}(t) - \mathbf{r}_{y_i}(t))^\top \mathbf{x}_i$ is not present in the vector. Similarly, let us define $[\![-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k$ to be the corresponding entry of $-\nabla\psi$.

This section makes this indexing trick rigorous.

Lemma D.1. *Let $\mathbf{W} \in \mathbb{R}^{d \times K}$ be arbitrary and $\mathbf{w} := \mathsf{vec}(\mathbf{W})$ be its vectorization. Let $(\mathbf{x}_i, y_i)$ be a training sample. Then there exists a bijection that depends only on y_i that maps the entries of*

$$\boldsymbol{\Upsilon}_y \mathbf{D}\mathbf{W}^\top \mathbf{x}_i \in \mathbb{R}^{K-1}$$

to the elements of the set of "y_i-versus-k" relative margins, i.e., $\{\tilde{\mathbf{x}}_{i,k}^\top \mathbf{w} \in \mathbb{R} : k \in [K] \setminus \{y_i\}\}$.

The following definition makes the bijection from Lemma D.1 concrete.

Definition D.1 (Pseudo-index)**.** *In the situation of Lemma D.1, define $[\![\cdot]\!]_{i,k} : \mathbb{R}^{K-1} \to \mathbb{R}$ to be the coordinate projection such that $[\![\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}\mathbf{x}_i]\!]_{i,k} = \tilde{\mathbf{x}}_{i,k}^\top \mathbf{w}$. In other words, $[\![\cdot]\!]_{i,k}$ selects the y_i-versus-k relative margin. When the sample index i is clear from context, we drop i from the subscript and simply write $[\![\cdot]\!]_k$.*

The pseudo-index is useful for working with the exponential tail bounds:

Lemma D.2. *In the situation of Lemma D.1, consider $\nabla\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i)$ which is the $(K-1)$-dimensional vector of partial derivatives of the template evaluated at $\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i$. If ψ satisfies Definition 2.2, then*

$$[\![-\nabla\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i)]\!]_k \leq \exp(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w})$$

and

$$[\![-\nabla\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}^\top\mathbf{x}_i)]\!]_k \geq (1-\textstyle\sum_{r\in[K]\setminus\{y_i\}}\exp(-\tilde{\mathbf{x}}_{i,r}^\top\mathbf{w}))\exp(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w})$$

for all $k\in[K]\setminus\{y_i\}$.

D.1 Proofs of Lemma D.1 and Lemma D.2

In both lemmas, we work with a fixed sample, i.e., the index i does not change. As such, we simply drop the index and write $y\leftarrow y_i$, $\mathbf{x}\leftarrow\mathbf{x}_i$, $\tilde{\mathbf{x}}_k\leftarrow\tilde{\mathbf{x}}_{i,k}$, and $[\![\cdot]\!]_k\leftarrow[\![\cdot]\!]_{i,k}$.

Below, we fix $k\in[K-1]$ throughout the proof. Let $\mathbf{v}:=\mathbf{w}^\top\mathbf{x}=[v_1,\,\ldots,\,v_K]^\top$. Note that

$$\mathbf{\Upsilon}_y\mathbf{D}^\top\mathbf{W}^\top\mathbf{x}=\mathbf{\Upsilon}_y\begin{bmatrix}v_K-v_1\\ \vdots\\ v_K-v_{K-1}\end{bmatrix} \tag{32}$$

We prove Lemma D.1 by considering the case $y=K$ and $y\neq K$ separately. First, let us consider the case when $y=K$. Then $\mathbf{\Upsilon}_y$ is the identity matrix and so

$$k\text{-th component of Equation (32) is } = v_K-v_k=(\mathbf{w}_K-\mathbf{w}_k)^\top\mathbf{x}=\tilde{\mathbf{x}}_k^\top\mathbf{w}.$$

Thus, we've proven Lemma D.1 when $y=K$. In this case, $[\![\cdot]\!]_k$ simply picks out the k-th entry of the input $(K-1)$-dimensional) vector. In other words,

$$[\![\mathbf{z}]\!]_k=z_k,\quad \text{for all } \mathbf{z}=[z_1,\,\ldots,z_{K-1}]^\top\in\mathbb{R}^{K-1}\text{ when } y=K. \tag{33}$$

By definition, we note that the k-th row of $\mathbf{\Upsilon}_y$ is

$$\mathbf{\Upsilon}_y[k,:]=\begin{cases}\mathbf{e}_k-\mathbf{e}_y & :k\neq y,\\ -\mathbf{e}_y & :\text{otherwise.}\end{cases}$$

Thus, when $y\neq K$

$$\begin{aligned}k\text{-th component of Equation (32) } &= \begin{cases}(v_K-v_n) & :k\neq y,\\ -(v_K-v_y) & :k=y.\end{cases}\\ &= \begin{cases}(\mathbf{w}_y-\mathbf{w}_k)^\top\mathbf{x}=\tilde{\mathbf{x}}_k & :k\neq y,\\ (\mathbf{w}_y-\mathbf{w}_K)^\top\mathbf{x}=\tilde{\mathbf{x}}_K & :k=y.\end{cases}\end{aligned}$$

Thus, we've proven Lemma D.1 when $y\neq K$ as well. In this case, $[\![\cdot]\!]_k$ picks out the k-th element of the input $(K-1)$-dimensional) vector when $k\neq y$. Otherwise when $k=y$, we have that $[\![\cdot]\!]_k$ picks out the y-th element. More explicitly,

$$[\![\mathbf{z}]\!]_k=\begin{cases}z_k & :k\neq y\\ z_y & :k=y\end{cases},\quad \text{for all } \mathbf{z}=[z_1,\,\ldots,z_{K-1}]^\top\in\mathbb{R}^{K-1}\text{ when } y\neq K. \tag{34}$$

Next, we prove Lemma D.2 by considering the case $y=K$ and $y\neq K$ separately. First, assume that we are in the $y=K$ case. From Equation (33), we get that

$$[\![-\nabla\psi(\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}^\top\mathbf{x}_i)]\!]_k=-\frac{\partial\psi}{\partial u_k}(\mathbf{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}^\top\mathbf{x}_i)$$

Now, we have that for $\mathbf{u}=[u_1,\,\ldots,\,u_{K-1}]^\top\in\mathbb{R}^{K-1}$, the upper and lower exponential tail bounds are

$$-\frac{\partial\psi}{\partial u_k}(\mathbf{u})\leq c\exp(-u_k),\quad\text{and}\quad -\frac{\partial\psi}{\partial u_k}(\mathbf{u})\geq c\left(1-\sum_{r\in[K-1]}\exp(-u_r)\right)\exp(-u_k).$$

Letting $\mathbf{u} := \boldsymbol{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}^\top\mathbf{x}_i$, using Lemma D.1 and Equation (33), we immediately prove Lemma D.2 in the case when $y = K$. This is because we have

$$\exp(-u_k) = \exp(-[\![\boldsymbol{\Upsilon}_{y_i}\mathbf{D}^\top\mathbf{W}^\top\mathbf{x}_i]\!]_k) = \exp(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w})$$

and

$$\sum_{r\in[K-1]}\exp\left(-u_r\right) = \sum_{r\in[K]\backslash\{y_i\}}\exp\left(-u_r\right) = \sum_{k\in[K]\backslash\{y_i\}}\exp\left(-\tilde{\mathbf{x}}_{i,k}^\top\mathbf{w}\right)$$

When $y = K$, a similar argument proves Lemma D.2 using Equation (33) for the pseudo-index $[\![\cdot]\!]_k$.

E Proof of Lemma 4.8

Re-stating the lemma:

Lemma 4.8. *(Multiclass generalization of Soudry et al. [2018, Lemma 1]) Consider any linearly separable dataset, and any PERM loss with template ψ that is convex, β-smooth, strictly decreasing, and non-negative. For all $k \in \{1, ..., K\}$, let $\mathbf{w}_k(t)$ be the gradient descent iterates at iteration t for the k^{th} class. Then $\forall i \in \{1, ..., N\}, \forall j \in \{1, ..., K\}\backslash\{y_i\} : \lim_{t\to\infty}(\mathbf{w}_{y_i}(t) - \mathbf{w}_j(t))^\top\mathbf{x}_i \to \infty$.*

Proof. We know that $\lim_{t\to\infty}\nabla\mathcal{R}\left(\mathbf{w}\left(t\right)\right) = \mathbf{0}$ by Soudry et al. [2018, Lemma 10].

This implies that $\hat{\mathbf{w}}^\top\nabla\mathcal{R}(\mathbf{w}(t)) \to \mathbf{0}$. Following the same steps as in the proof of Lemma 4.4, this is equivalent to saying:

$$\hat{\mathbf{w}}^\top\nabla\mathcal{R}(\mathbf{w}(t)) = \mathsf{tr}(\nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i\right)^\top\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\hat{\mathbf{W}}^\top\mathbf{x}_i) \to 0.$$

However, for linearly separable data we know that $\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\hat{\mathbf{W}}^\top\mathbf{x}_i \succeq \mathbf{1}$ (since $\hat{\mathbf{W}}$ here is the hard-margin SVM solution). Thus for the above limit to be true, the limit

$$\lim_{t\to\infty}\nabla\psi\left(\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i\right) = \mathbf{0}$$

must hold. By Proposition G.1, we have

$$\lim_{t\to\infty}\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i = \boldsymbol{\infty} \qquad \forall i \in [N]$$

where $\boldsymbol{\infty}$ is the "vector" whose entries are all equal to infinity. This is equivalent to

$$\lim_{t\to\infty}(\mathbf{w}_{y_i}(t) - \mathbf{w}_j(t))^\top\mathbf{x}_i = \infty \qquad \forall i \in [N], \forall j \in [K]\backslash\{y_i\}$$

(since $[\![\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i]\!]_j = (\mathbf{w}_{y_i}(t) - \mathbf{w}_j(t))^\top\mathbf{x}_i$). □

Note that in the binary case, the above "convergence-to-infinity" condition is for a scalar quanity, where the assumption that the loss be strictly decreasing and non-negative suffices. In the multiclass setting, we must ensure that all entries of the vector $\boldsymbol{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top\mathbf{x}_i$ converges to infinity. This is a nontrivial result and is addressed by our Proposition G.1.

F Proof of Lemma 4.6

Let us first re-state the lemma we want to prove.

Lemma 4.6. *(Generalization of Soudry et al. [2018, Lemma 20]) Define θ to be the minimum SVM margin across all data points and classes, i.e., $\theta = \min_k\left[\min_{n\notin\mathcal{S}_k}\tilde{\mathbf{x}}_{n,k}^\top\hat{\mathbf{w}}\right] > 1$. Then:*

$$\exists C_1, C_2, t_1 : \forall t > t_1 : \left(\mathbf{r}\left(t+1\right) - \mathbf{r}\left(t\right)\right)^\top\mathbf{r}\left(t\right) \leq C_1 t^{-\theta} + C_2 t^{-2}$$

Proof. Proceeding the same way as Soudry et al. [2018], we have

$$\begin{aligned}
&\left(\mathbf{r}\left(t+1\right)-\mathbf{r}\left(t\right)\right)^{\top}\mathbf{r}\left(t\right)\\
&=\left(-\eta\nabla\mathcal{R}\left(\mathbf{w}\left(t\right)\right)-\hat{\mathbf{w}}\left[\log\left(t+1\right)-\log\left(t\right)\right]\right)^{\top}\mathbf{r}\left(t\right)\\
&=\left(-\eta\nabla\mathcal{R}\left(\mathbf{w}\left(t\right)\right)\right)^{\top}\mathbf{r}(t)-\hat{\mathbf{w}}^{\top}\mathbf{r}(t)\log\left(1+t^{-1}\right)\\
&=\hat{\mathbf{w}}^{\top}\mathbf{r}(t)\left(t^{-1}-\log\left(1+t^{-1}\right)\right)+\mathsf{tr}\left(\left(-\eta\sum_{i=1}^{N}\mathbf{x}_{i}\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)^{\top}\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\right)^{\top}\mathbf{R}(t)\right)\\
&\quad-t^{-1}\hat{\mathbf{w}}^{\top}\mathbf{r}(t)
\end{aligned}\tag{35}$$

The last equality is a new step required for our multiclass generalization, in which we used Lemma 4.2 and introduced the matrices $\mathbf{W}(t)$ and $\mathbf{R}(t)$, where $\mathsf{vec}(\mathbf{W}(t))=\mathbf{w}(t)$ and $\mathsf{vec}(\mathbf{R}(t))=\mathbf{r}(t)$. Let us focus just on the second term of this expansion.

$$\begin{aligned}
&\mathsf{tr}\left(\left(-\eta\sum_{i=1}^{N}\mathbf{x}_{i}\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)^{\top}\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\right)^{\top}\mathbf{R}(t)\right)\\
&\overset{(1)}{=}\eta\mathsf{tr}\left(\sum_{i=1}^{N}\mathbf{R}(t)^{\top}\mathbf{x}_{i}\left(-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)\right)^{\top}\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\right)\\
&\overset{(2)}{=}\eta\mathsf{tr}\left(\sum_{i=1}^{N}\left(-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)\right)^{\top}\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{R}(t)^{\top}\mathbf{x}_{i}\right)
\end{aligned}\tag{36}$$

In step (1) we used the fact that for any square matrix $\mathbf{M}$, $\mathsf{tr}(\mathbf{M})=\mathsf{tr}\left(\mathbf{M}^{\top}\right)$. In step (2) we used the cyclic property of the trace.

Similar to in the proof of Lemma 4.4, the trace's cyclic property has enabled us to convert a matrix-product into a simple dot product. Since dot products are scalars, we can now drop the trace and rewrite our expression in Eqn. (36) as a dot product:

$$\eta\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_{i}\}}[\![-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)]\!]_{k}[\![\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{R}(t)^{\top}\mathbf{x}_{i}]\!]_{k}$$

Using this form, we can rewrite Eqn. (35):

$$\begin{aligned}
\left(\mathbf{r}\left(t+1\right)-\mathbf{r}\left(t\right)\right)^{\top}\mathbf{r}\left(t\right)=\hat{\mathbf{w}}^{\top}\mathbf{r}(t)\left(t^{-1}-\log\left(1+t^{-1}\right)\right)\\
+\eta\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_{i}\}}[\![-\nabla\psi\left(\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{W}(t)^{\top}\mathbf{x}_{i}\right)]\!]_{k}[\![\boldsymbol{\Upsilon}_{y_{i}}\mathbf{D}\mathbf{R}(t)^{\top}\mathbf{x}_{i}]\!]_{k}\\
-t^{-1}\hat{\mathbf{w}}^{\top}\mathbf{r}(t)
\end{aligned}\tag{37}$$

The first term $\hat{\mathbf{w}}^{\top}\mathbf{r}(t)\left(t^{-1}-\log\left(t^{-1}\right)\right)$ is bounded in [Soudry et al., 2018, Eqn. (139)]. We will focus on the second and third terms. Recall by Equation (6) and Equation (8) that

$$\hat{\mathbf{w}}=\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_{i}\}}\alpha_{i,k}\mathbb{1}_{\{i\in S_{k}\}}\tilde{\mathbf{x}}_{i,k}=\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_{i}\}}\eta\exp(-\tilde{\mathbf{w}}^{\top}\tilde{\mathbf{x}}_{i,k})\mathbb{1}_{\{i\in S_{k}\}}\tilde{\mathbf{x}}_{i,k}$$

Thus, the third term on the RHS of Equation (37) can be written as

$$t^{-1}\hat{\mathbf{w}}^{\top}\mathbf{r}(t)=\eta\sum_{i=1}^{N}\sum_{k\in[K]\setminus\{y_{i}\}}t^{-1}\exp(-\tilde{\mathbf{w}}^{\top}\tilde{\mathbf{x}}_{i,k})\tilde{\mathbf{x}}_{i,k}^{\top}\mathbf{r}(t)\mathbb{1}_{\{i\in S_{k}\}}$$

Therefore, the last two terms on the RHS of Equation (37) can be written as

$$\eta \sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} [\![-\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k [\![\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i]\!]_k - t^{-1}\hat{\mathbf{w}}^\top \mathbf{r}(t)$$
$$= \eta\Big(\sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} [\![-\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k [\![\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i]\!]_k$$
$$- t^{-1}\exp(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{i,k})\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)\mathbb{1}_{\{i\in S_k\}}\Big)$$

Since $\eta > 0$ is constant, we ignore it below and consider only the term inside the parenthesis:

$$\sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} [\![-\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k [\![\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i]\!]_k$$
$$- t^{-1}\exp(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{i,k})\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)\mathbb{1}_{\{i\in S_k\}}$$
$$\overset{(1)}{=} \sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} \big([\![-\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k$$
$$- t^{-1}\exp(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{i,k})\mathbb{1}_{\{i\in S_k\}}\big)\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)$$
$$\overset{(2)}{\leq} \sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} \big(\exp\left(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k}\right)$$
$$- t^{-1}\exp(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{i,k})\mathbb{1}_{\{i\in S_k\}}\big)\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)\mathbb{1}_{\{\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)\geq 0\}}$$
$$+ \sum_{i=1}^{N} \sum_{k\in[K]\setminus\{y_i\}} \Big(\exp\left(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k}\right)\left(1 - \sum_{k\in[K]} \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k})\right)$$
$$- t^{-1}\exp(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{i,k})\mathbb{1}_{\{i\in S_k\}}\Big)\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)\mathbb{1}_{\{\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t)<0\}}. \tag{38}$$

In (1) we used Lemma D.1, which implies that $\tilde{\mathbf{x}}_{i,k}^\top \mathbf{r}(t) = [\![\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{R}(t)^\top \mathbf{x}_i]\!]_k$. For (2), from the exponential tail upper/lower bound and Lemma D.2, we have that

$$\exp\left(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k}\right) \geq [\![-\nabla\psi\left(\mathbf{\Upsilon}_{y_i}\mathbf{D}\mathbf{W}(t)^\top \mathbf{x}_i\right)]\!]_k$$
$$\geq \exp\left(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k}\right)\Big(1 - \sum_{k\in[K]\setminus\{y_i\}} \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{i,k})\Big).$$

We note that Eqn. (38) above is identical to the right hand side of inequality (1) in [Soudry et al., 2018, Eqn. (141)]. Thus, the remainder of the analysis proceeds identically as in Soudry et al. [2018, Lemma 20]. □

G A structural result on symmetric and convex functions

Proposition G.1. *Let $\psi : \mathbb{R}^{K-1} \to \mathbb{R}$ be the template of a PERM loss that satisfies our Theorem 3.4. Let $\mathbf{u}^t \in \mathbb{R}^{K-1}$ be any sequence, where $t = 1, 2, \ldots$, such that*

$$\lim_{t\to\infty} \nabla\psi(\mathbf{u}^t) = \mathbf{0}$$

is the zero vector. Then $\lim_{t\to\infty} u_j^t = \infty$ for every $j \in [K-1]$.

We prove Proposition G.1 by first proving a structural result (Theorem G.2) concerning symmetric and convex function $f : \mathbb{R}^n \to \mathbb{R}$. The proof of Proposition G.1 will be presented in Appendix G.2 as an application of the structural result, where we take $f = \psi$, the template of a PERM loss, and $n = K - 1$, number of classes minus one.

Given a vector $\mathbf{x} \in \mathbb{R}^n$ and a real number $C \in \mathbb{R}$, define $\mathbf{x} \vee C \in \mathbb{R}^n$ to be the vector such that

$$[\mathbf{x} \vee C]_i := \max\{x_i, C\}, \quad \text{for all } i \in [n].$$

In other words, $\mathbf{x} \vee C$ "boosts" entries of $\mathbf{x}$ up to C if those entries are smaller than C. Entries of $\mathbf{x}$ larger than C are kept as-is.

Define $\min(\mathbf{x}) = \min_{j \in [n]} x_j$ and $\operatorname{argmin}(\mathbf{x}) := \{i \in [n] : x_i = \min(\mathbf{x})\}$. We note the following easy-to-prove properties of the "$\vee$" operation:

1. $\min(\mathbf{x} \vee C) \geq C$ with equality if $\min(\mathbf{x}) \leq C$,
2. $\operatorname{argmin}(\mathbf{x} \vee C) \supseteq \operatorname{argmin}(\mathbf{x})$.

Theorem G.2. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a symmetric, convex, and differentiable function. Then for any real number $C \in \mathbb{R}$ and any $\mathbf{x} \in \mathbb{R}^n$, we have*

$$\tfrac{\partial f}{\partial x_i}(\mathbf{x}) \leq \tfrac{\partial f}{\partial x_i}(\mathbf{x} \vee C), \quad \textit{for any } i \in \operatorname{argmin}(\mathbf{x}).$$

Before proceeding with the proof (which is in Appendix G.1), we first introduce some necessary preliminary notations and facts. Given a vector $\mathbf{x} \in \mathbb{R}^n$, we define

$$\mathsf{val}(\mathbf{x}) := \{x_i : i = 1, \ldots, n\}$$

to be the *set* of values consistings of the entries of $\mathbf{x}$. For example, if $\mathbf{x}$ is the all-ones vector, then $\mathsf{val}(\mathbf{x}) = \{1\}$. Given $v \in \mathsf{val}(\mathbf{x})$, we let $\mathsf{idx}(v, \mathbf{x}) = \{i \in \mathbf{x} : x_i = v\}$ be the set of indices that attains the value v.

Fact 1: For a convex and differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we have that

$$\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq 0, \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n. \tag{39}$$

This is a simple and well-known consequence of convexity. See this stackexchange answer for a short proof. When $n = 1$, Ineq. (39) is the fact that a convex differentiable function has nondecreasing derivative. Ineq. (39) is also a consequence of [Phelps, 2009, Theorem 3.24].

Fact 2: For a symmetric and differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we have that

$$\tfrac{\partial f}{\partial x_i}(\mathbf{x}) = \tfrac{\partial f}{\partial x_j}(\mathbf{x}), \quad \text{whenever } x_i = x_j. \tag{40}$$

This fact follows from the chain rule and the definition of a symmetric function. To be precise, let $\mathbf{T} : \mathbb{R}^n \to \mathbb{R}^n$ be the permutation matrix that switches the i and j-th coordinate. Then $f(\mathbf{x}) = f(\mathbf{T}\mathbf{x})$ and moreover $\frac{\partial f}{\partial x_i}(\mathbf{x}) = \frac{\partial f}{\partial x_i}(\mathbf{T}\mathbf{x}) = [\mathbf{T}\nabla f(\mathbf{T}\mathbf{x})]_i = [\nabla f(\mathbf{T}\mathbf{x})]_j = [\nabla f(\mathbf{x})]_j = \frac{\partial f}{\partial x_j}(\mathbf{x})$.

G.1 Proof of Theorem G.2

Now, to prove the above theorem, we will use induction on "m" in the following lemma, which is simply a "stratification" of Theorem G.2 into cases indexed by the "parameter" m:

Lemma G.3. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is a convex, symmetric, and differentiable function. Let $m \in \{0, 1, \ldots, n\}$. Then for any real number $C \in \mathbb{R}$ and any $\mathbf{x} \in \mathbb{R}^n$ with the property that $|\{v \in \mathsf{val}(\mathbf{x}) : v < C\}| = m$, we have*

$$\tfrac{\partial f}{\partial x_i}(\mathbf{x}) \leq \tfrac{\partial f}{\partial x_i}(\mathbf{x} \vee C), \quad \textit{for any } i \in \operatorname{argmin}(\mathbf{x}).$$

Note that if we have proved Lemma G.3 for each $m \in \{0, 1, \ldots, n\}$, then Theorem G.2 holds.

The base step: we prove Lemma G.3 when $m = 0$ and $m = 1$. Strictly speaking, the proof-by-induction technique typically only involve *only* the base case, which would be the $m = 0$ case in this instance. But below, we will see that in the induction step, the $m = 1$ case is helpful.

Note that the $m = 0$ case holds vacuously, since $\mathbf{x} \vee C = \mathbf{x}$. Below, we focus on the $m = 1$ case, where there exists a unique $v \in \mathsf{val}(\mathbf{x})$ such that $v \leq C$. Let $i \in \operatorname{argmin}(\mathbf{x})$. Note that we have $\mathsf{idx}(v, \mathbf{x}) = \operatorname{argmin}(\mathbf{x})$. Using Equation (39) (Fact 1), we have that

$$\langle \nabla f(\mathbf{x} \vee C) - \nabla f(\mathbf{x}), (\mathbf{x} \vee C) - \mathbf{x} \rangle \geq 0.$$

Definition G.1. *Given any set $S \subseteq [n]$, we let $\chi_S \in \mathbb{R}^n$ denote the* characteristic vector on S: *χ_S is the vector whose jth entry is $= 1$ if $j \in S$ and $= 0$ otherwise.*

By construction, we have

$$(\mathbf{x} \vee C) - \mathbf{x} = (C - v)\chi_{\mathsf{idx}(v,\mathbf{x})} = (C - v)\chi_{\mathrm{argmin}(\mathbf{x})}.$$

The "$\frac{\partial f}{\partial x_i}(\cdot)$" notation for partial derivatives is a bit cumbersome. Instead, we will write "$[\nabla f(\cdot)]_i$" from now on. By Equation (40) (Fact 2), we have

$$[\nabla f(\mathbf{x})]_j = [\nabla f(\mathbf{x})]_{j'}, \quad \text{for all } j, j' \in \mathsf{idx}(v, \mathbf{x})$$

and likewise

$$[\nabla f(\mathbf{x} \vee C)]_j = [\nabla f(\mathbf{x} \vee C)]_{j'}, \quad \text{for all } j, j' \in \mathsf{idx}(v, \mathbf{x}).$$

Thus, by Equation (40), we have

$$\langle \nabla f(\mathbf{x} \vee C) - \nabla f(\mathbf{x}), (\mathbf{x} \vee C) - \mathbf{x} \rangle = |\mathrm{argmin}(\mathbf{x})| \cdot (C - v)([\nabla f(\mathbf{x} \vee C)]_i - [\nabla f(\mathbf{x})]_i).$$

Now, since $C > v$ and $|\mathrm{argmin}(\mathbf{x})| > 0$, we must have that $[\nabla f(\mathbf{x} \vee C)]_i - [\nabla f(\mathbf{x})]_i \geq 0$, as desired. This proves the base step.

Induction step: Suppose Lemma G.3 holds for every integer m where $0 \leq m < n$, we must show that Lemma G.3 also holds for $m + 1$. To this end, let $\mathbf{x} \in \mathbb{R}^n$ and $C \in \mathbb{R}$ be such that $|\{v \in \mathsf{val}(\mathbf{x}) : v < C\}| = m+1$. Let $v_1, \ldots, v_{m+1} \in \mathbb{R}$ be all the elements of $\{v \in \mathsf{val}(\mathbf{x}) \mid v < C\}$ enumerated in increasing order, i.e., $v_1 < \cdots < v_{m+1}$.

Note by construction, we have that $\{v \in \mathsf{val}(\mathbf{x}) \mid v < v_{m+1}\} = \{v_1, \ldots, v_m\}$ and so we immediately get that $|\{v \in \mathsf{val}(\mathbf{x}) \mid v < v_{m+1}\}| = m$. By the m-th case of Lemma G.3 (i.e., the induction hypothesis) using v_{m+1} as C, we get

$$[\nabla f(\mathbf{x} \vee v_{m+1})]_i \geq [\nabla f(\mathbf{x})]_i \quad \text{for any } i \in \mathrm{argmin}(\mathbf{x}). \tag{41}$$

Below fix some $i \in \mathrm{argmin}(\mathbf{x})$ arbitrarily. Let $\mathbf{x}' := \mathbf{x} \vee v_{m+1}$. We note that by construction, all the entries of $\mathbf{x}'$ that are less than C are set to equal to v_{m+1}. In other words,

$$\{v \in \mathsf{val}(\mathbf{x}') : v < C\} = \{v_{m+1}\}$$

is a singleton set. Thus, by the $m = 1$ case of Lemma G.3 applied to $\mathbf{x}'$, we get that

$$[\nabla f(\mathbf{x}' \vee C)]_{i'} \geq [\nabla f(\mathbf{x}')]_{i'}, \quad \text{for any } i' \in \mathrm{argmin}(\mathbf{x}').$$

Since $\mathrm{argmin}(\mathbf{x}') \supseteq \mathrm{argmin}(\mathbf{x})$, we have that $i \in \mathrm{argmin}(\mathbf{x}')$ as well (recall that i was chosen earlier from $\mathrm{argmin}(\mathbf{x})$ arbitrarily). Thus the above inequality implies in particular that

$$[\nabla f(\mathbf{x}' \vee C)]_i \geq [\nabla f(\mathbf{x}')]_i = [\nabla f(\mathbf{x} \vee v_{m+1})]_i.$$

Combined with Equation (41), we get

$$[\nabla f(\mathbf{x}' \vee C)]_i \geq [\nabla f(\mathbf{x})]_i.$$

Finally, we note that $\mathbf{x}' \vee C = (\mathbf{x} \vee v_{m+1}) \vee C = \mathbf{x} \vee C$. Thus, the above implies

$$[\nabla f(\mathbf{x} \vee C)]_i \geq [\nabla f(\mathbf{x})]_i \quad \text{for any } i \in \mathrm{argmin}(\mathbf{x})$$

since the choice of $i \in \mathrm{argmin}(\mathbf{x})$ was arbitrary.

G.2 Application to our setting

The condition $\lim_{t\to\infty} u_j^t = \infty$ by definition means that for every real number $M \in \mathbb{R}$, there exists T such that for all $t \geq T$ we have $u_j^t > M$. Thus, suppose that there exists $j \in [K-1]$ such that $\lim_{t\to\infty} u_j^t \neq \infty$, then there exists a real number $M \in \mathbb{R}$ such that for all $T = 1, 2, \ldots$ there exists some $t \geq T$ such that $u_j^t \leq M$. Passing to a subsequence, we assume that $u_j^t \leq M$ (and so $\min(\mathbf{u}^t) \leq M$) for all $t = 1, 2, \ldots$. Note that $\lim_{t\to\infty} \nabla\psi(\mathbf{u}^t) = \mathbf{0}$ continues to hold.

Below, whenever we say "for all/every t", we mean "for all/every $t = 1, 2, \ldots$".

Onto the proof. First recall the lower bound portion of the exponential tail property:

$$\text{For all } \mathbf{u} \in \mathbb{R}^{K-1} \text{ such that } \min(\mathbf{u}) > u_-, \text{ we have}$$
$$-[\nabla\psi(\mathbf{u})]_i \geq c\Big(1 - \sum_{j=1}^{K-1} \exp(-u_j)\Big)\exp(-u_i), \quad \text{for all } i \in [K-1]. \tag{42}$$

Let $C := \max\{\ 2|u_-|,\ M,\ -\log(\frac{1}{2(K-1)})\ \}$ and $\mathbf{v}^t := \mathbf{u}^t \vee C$ for all t. This choice of C (and $\mathbf{v}^t$) has the following consequences:

1. The fact that $C \geq -\log(\frac{1}{2(K-1)})$ implies $\Big(1 - \sum_{j=1}^{K-1}\exp(-v_j)\Big) \geq \frac{1}{2}$.
2. If $\mathbf{v} \in \mathbb{R}^{K-1}$ is such that $\min(\mathbf{v}) \geq C$, then we have by Equation (42) that
$$-[\nabla\psi(\mathbf{v})]_i \geq \frac{1}{2}c\exp(-v_i). \tag{43}$$
3. $\min(\mathbf{u}^t) \leq C$. This is true since $\min(\mathbf{u}^t) \leq M$.
4. Choose $i_t \in \operatorname{argmin}(\mathbf{u}^t)$ for every t. Then $v^t_{i_t} = \min(\mathbf{v}^t) = C$ for every t. This is simply a consequence of the fact that $\operatorname{argmin}(\mathbf{u}^t) \subseteq \operatorname{argmin}(\mathbf{v}^t)$.
5. We have $\lim_{t\to\infty}[\nabla\psi(\mathbf{u}^t)]_{i_t} = 0$. This follows from the assumption that $\lim_{t\to\infty}\nabla\psi(\mathbf{u}^t) = \mathbf{0}$.

By Theorem G.2, we have for every t that

$$[\nabla\psi(\mathbf{u}^t)]_{i_t} \leq [\nabla\psi(\mathbf{v}^t)]_{i_t}.$$

By plugging in $\mathbf{v}^t$ for $\mathbf{u}$ in Equation (43) above and the fact that $v^t_{i_t} = C$, we have

$$[\nabla\psi(\mathbf{v}^t)]_{i_t} \leq -\tfrac{1}{2}c\exp(-v_{i_t}) = -\tfrac{1}{2}c\exp(-C) < 0.$$

Since $-\frac{1}{2}c\exp(-C)$ is a constant that doesn't depend on t, it is impossible for $\lim_{t\to\infty}[\nabla\psi(\mathbf{v}^t)]_{i_t} = 0$ to hold. This proves Proposition G.1.

H On the existence of $\tilde{\mathbf{w}}$

The goal of this section is to explain the challenge and the current gap in the proof of the existence of $\tilde{\mathbf{w}}$ that satisfies the condition in Equation (8) for almost all linearly separable datasets. To this end, recall Equation (4), the hard-margin SVM formulated as a constrained optimization:

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \text{ s.t. } \forall n, \forall k \neq y_n : \mathbf{w}_{y_n}^\top \mathbf{x}_n \geq \mathbf{w}_k^\top \mathbf{x}_n + 1. \tag{44}$$

Moreover, recall that $\mathcal{S}_k$, the set of support vectors for each $k \in [K]$, is defined by

$$\mathcal{S}_k := \{n : (\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k)^\top \mathbf{x}_n = 1\}.$$

The Lagrangian of the objective in Equation (44) is

$$L(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2}\sum_{r=1}^{K}\|\mathbf{w}_r\|^2 + \sum_{n=1}^{N}\sum_{r\neq y_n}\alpha_{n,r}\left(\mathbf{w}_{y_n} - \mathbf{w}_r\right)^\top \mathbf{x}_n \tag{45}$$

where $\alpha_{n,r}$ are the dual variables. Let $\delta_{i,j}$ denote the Kronecker delta, i.e., $\delta_{i,j} = 1$ if $i = j$ and $\delta_{i,j} = 0$ otherwise. Taking the gradient of $L(\mathbf{w}, \boldsymbol{\alpha})$ with respect to $\mathbf{w}_k$, we get

$$\mathbf{w}_k + \sum_{n=1}^{N}\sum_{r\neq y_n}\alpha_{n,k}\left(\delta_{r,y_n} - \delta_{r,k}\right)\mathbf{x}_n = \mathbf{w}_k + \sum_{n=1}^{N}\left(\delta_{k,y_n}\sum_{r\neq y_n}\alpha_{n,r} - \alpha_{n,k}\right)\mathbf{x}_n.$$

So the KKT conditions satisfied by a stationary point $\hat{\mathbf{w}}$ (hence globally optimal for Equation (44)) are

$$\forall k \in [K] : \hat{\mathbf{w}}_k = \sum_{n=1}^{N} \left(\alpha_{n,k} - \delta_{k,y_n} \sum_{r \neq k} \alpha_{n,r} \right) \mathbf{x}_n \tag{46}$$

$$\forall k \in [K] : \forall n : \text{one of the following holds} \begin{cases} \alpha_{n,k} \geq 0 \text{ and } \left(\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k\right)^\top \mathbf{x}_n = 1 \\ \alpha_{n,k} = 0 \text{ and } \left(\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k\right)^\top \mathbf{x}_n > 1 \end{cases} \tag{47}$$

where Eqn. (47) (the second line) above is the complementary slackness condition.

The goal of this section is to prove the following result regarding the existence of $\tilde{\mathbf{w}}$ that satisfies the condition in Equation (8), which we restate below:

$$\forall k \in [K], \forall n \in \mathcal{S}_k : \eta \exp\left(-\mathbf{x}_n^\top \left(\tilde{\mathbf{w}}_{y_n} - \tilde{\mathbf{w}}_k\right)\right) = \alpha_{n,k}. \tag{48}$$

Conjecture H.1. *For almost all linearly separable multiclass datasets, Assumption 4.1 holds, i.e., Eqn.* (48) *has a solution* $\tilde{\mathbf{w}}$.

Below, we use the word "generically" to mean "for linearly separable datasets outside of a set of Lebesgue measure zero". In order for (48) to have a solution in $\tilde{\mathbf{w}}$ generically, two conditions need to hold (generically).

Condition 1. $\alpha_{n,k} > 0$ for all k and n such that $n \in \mathcal{S}_k := \{n : (\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k)^\top \mathbf{x}_n = 1\}$.

Condition 1 is already nontrivial and a gap in proving the Conjecture, as we will see below. For the sake of explaining Condition 2 below, let us assume Condition 1 holds. Then we can rewrite (48) as

$$\forall k \in [K], \forall n \in \mathcal{S}_k : \mathbf{x}_n^\top \left(\tilde{\mathbf{w}}_{y_n} - \tilde{\mathbf{w}}_k\right) = \log\left(\frac{\eta}{\alpha_{n,k}}\right). \tag{49}$$

Define the vector $\mathbf{m}_{n,k}$ obtained by taking the difference between the k-th and y_n-th elementary basis vector in $\mathbb{R}^K$, i.e.,

$$\mathbf{m}_{n,k} := \mathbf{e}_k - \mathbf{e}_{y_n} \in \mathbb{R}^K.$$

Then we can further rewrite (49) as

$$\forall k \in [K], \forall n \in \mathcal{S}_k : \left(\mathbf{m}_{n,k} \otimes \mathbf{x}_n\right)^\top \tilde{\mathbf{w}} = \log\left(\frac{\eta}{\alpha_{n,k}}\right). \tag{50}$$

It is more convenient to pool all the class-specific support vectors $\mathcal{S}_k$ into a single set: $\mathcal{S} \triangleq \left\{(n,k) : \left(\hat{\mathbf{w}}_{y_n} - \hat{\mathbf{w}}_k\right)^\top \mathbf{x}_n = 1\right\}$. For readability, we linearly order the tuples in $\mathcal{S}$, i.e., we assign to each $(n,k) \in \mathcal{S}$ a unique index $i \in \{1, ..., |\mathcal{S}|\}$. In other words, we define $n(1), \ldots, n(|\mathcal{S}|)$ and $k(1), \ldots, k(|\mathcal{S}|)$ such that

$$\mathcal{S} = \{(n(1), k(1)),\ (n(2), k(2)),\ \ldots,\ (n(|\mathcal{S}|), k(|\mathcal{S}|))\}.$$

To reduce notational clutter in the subscript, define $\overline{\mathbf{x}}_i \triangleq \mathbf{x}_{n(i)}$ and $\overline{\mathbf{m}}_i \triangleq \mathbf{m}_{n(i),k(i)}$. Finally, define

$$\overline{\mathbf{M}} \triangleq \left[\overline{\mathbf{m}}_1, \ldots, \overline{\mathbf{m}}_{|\mathcal{S}|}\right] \mathbb{R}^{K \times |\mathcal{S}|},\ \overline{\mathbf{X}} \triangleq \left[\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_{|\mathcal{S}|}\right] \in \mathbb{R}^{d \times |\mathcal{S}|},\ \text{and } \mathbf{G} \triangleq (\mathbf{M} \circ \mathbf{X}) \in \mathbb{R}^{dK \times |\mathcal{S}|}.$$

with $\circ$ denoting the Khatri-Rao product, which is, by definition, the matrix obtained by taking the Kronecker product of corresponding columns [Khatri and Rao, 1968]. Note that the Khatri-Rao product is only defined for two matrices that have the same number of columns. See Liu [1999] for a reference. We now state

Condition 2. $\mathrm{rank}(\mathbf{G}) = |\mathcal{S}|$ generically.

Note that given Condition 2, Eqn. (50) has a solution in $\tilde{\mathbf{w}}$, while Condition 1 is necessary for the logarithm in (50) to be valid in the first place.

The challenge in proving Condition 2 in the multiclass case is that the column vectors of $\overline{\mathbf{X}}$ may have repeats, i.e., it is possible for $n(i) = n(i')$ when $i \neq i'$. It is easy to generate synthetic linearly

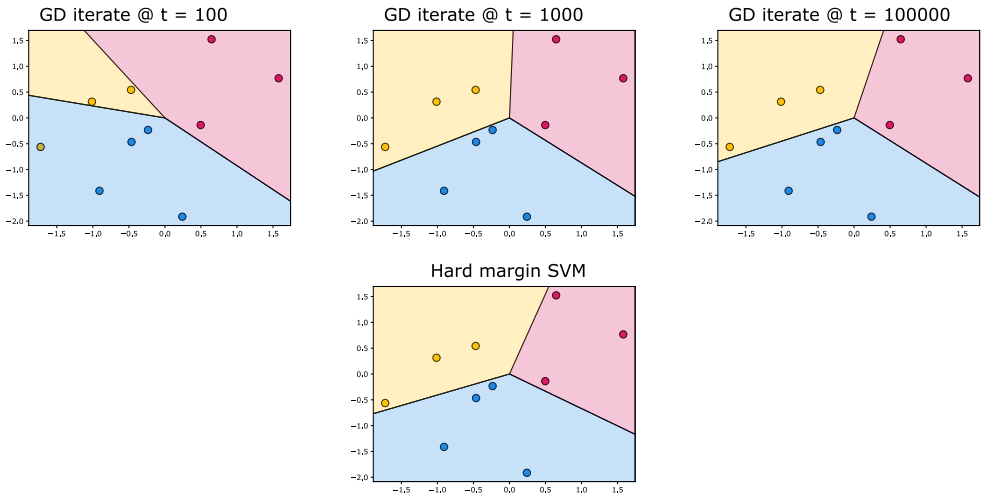

Figure 2: Small simulation with $N = 10$, $d = 2$ and $K = 3$. The loss used is the "PairLogLoss". *Top row.* Decision regions of classifiers along the gradient path $\mathbf{w}(t)$ at $t = 100$, 1000, and 100000, respectively from left to right. *Bottom row.* Decision regions of the hard-margin multiclass SVM. Note that most of the progress is made between iterations 100 and 1000.

separable multiclass datasets satisfying this condition. Nonetheless, we observe that even in such a case, the matrix $\mathbf{G}$ has rank $|\mathcal{S}|$, i.e., Condition 2 holds. We verify this experimentally in the Python notebook `checking_conjecture_in_Appendix_H.ipynb` available at

`https://github.com/YutongWangML/neurips2024-multiclass-IR-figures`

In the binary case, linear classifiers are parametrized simply as a single vector, rather than the more cumbersome one-vector-per-class parametrization. Under the one-vector parametrization, the $\overline{\mathbf{M}}$ matrix becomes a 1-by-$|\mathcal{S}|$ matrix consistings of only ± 1's, and $\mathbf{G}$ reduces to $\overline{\mathbf{X}}$. Moreover $\overline{\mathbf{X}}$ has no repeats. Thus, Condition 2 holds trivially. In both the multiclass and binary settings, given Condition 2, the proof for Condition 1 can proceed exactly as in Lemma 12 from Soudry et al. [2018] where their $\mathbf{X}_{\mathcal{S}}$ is replaced by our $\mathbf{G}$.

I Additional Experiments

We provide additional experimental support for our main theoretical result for the PairLogLoss [Wang et al., 2022]. Code for recreating the figures can be found at

`https://github.com/YutongWangML/neurips2024-multiclass-IR-figures`

The code can be ran on Google Colab with a CPU runtime in under one hour.

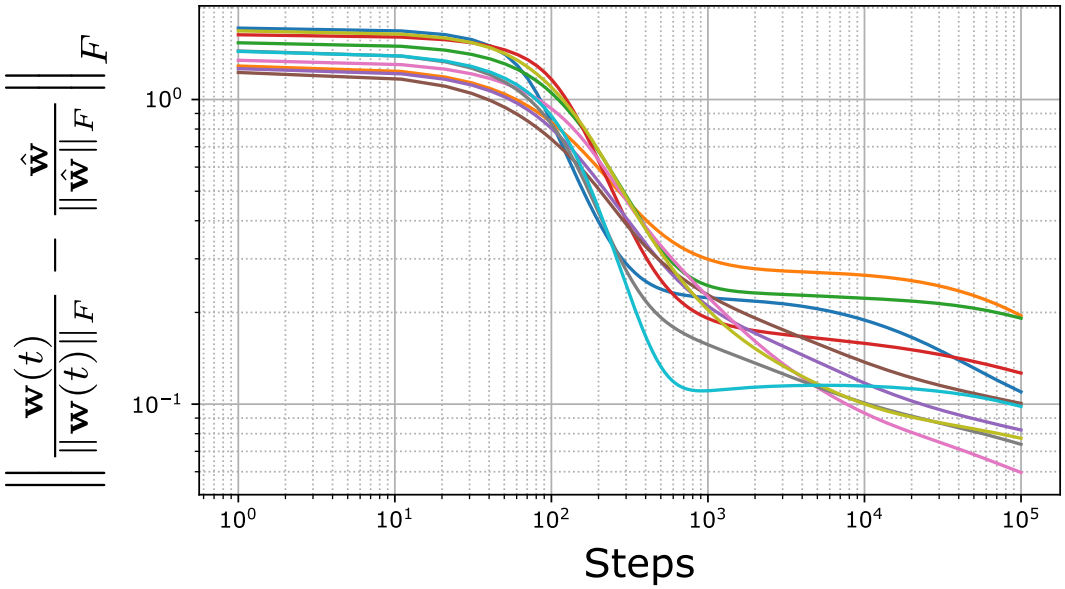

Figure 3: Large simulations with $N = 100$, $d = 10$ and $K = 3$. The loss used is the "PairLogLoss". The curves are 10 independent runs with randomly sampled data and random initialization for gradient descent over 100000 iterations. Note that the convergence in direction of the gradient descent iterates to the hard-margin SVM slows down in log-log space.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract clearly introduces the problem considered (implicit bias), identifies a gap in research (few multiclass results, which themselves are only for cross-entropy), and states our contributions (new ET property and implicit bias theorem for new losses). For the sake of brevity we do not state additional assumptions on the loss apart from ET (which we state later in the main text, i.e. smoothness, strictly decreasing, non-negative), because the ET property is a novel contribution and deserves to appear in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a separate section where we talk about our work's limitations. We highlight 2 natural questions one can ask: non-ET loss characterization, and non-asymptotic analysis (answering whether overfitting occurs after some *finite* number of (S)GD timesteps).

   Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All assumptions are stated clearly in bold at the beginning of section 3, and also re-iterated multiple times throughout the paper, The assumption on the learning rate being sufficiently small is mentioned in the theorem statement. Partial proofs are provided in the main text because they highlight salient features of our techniques (namely, simple generalization of binary proof techniques to multiclass). Complete proofs are provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We use a simple synthetic setup which can be reproduced easily with Google Colab.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In Appendices H and I, we include a link to our GitHub repo which contains the complete code to reproduce the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details can be found in the GitHub repository.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments are used to illustrate the main theoretical result, which is of mathematical nature. All experiments support the convergence behavior that we analyzed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the experiments can be run with a Google Colab CPU runtime as mentioned in the Appendix.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.