Maximum Class Separation as Inductive Bias in One Matrix

Anonymous Author(s)
Affiliation
Address
email

Abstract

Maximizing the separation between classes constitutes a well-known inductive bias in machine learning and a pillar of many traditional algorithms. By default, deep networks are not equipped with this inductive bias and therefore many alternative solutions have been proposed through differential optimization. Current approaches tend to optimize classification and separation jointly: aligning inputs with class vectors and separating class vectors angularly. This paper proposes a simple alternative: encoding maximum separation as an inductive bias in the network by adding one fixed matrix multiplication before computing the softmax activations. The main observation behind our approach is that separation does not require optimization but can be solved in closed-form prior to training and plugged into a network. We outline a recursive approach to obtain the matrix consisting of maximally separable vectors for any number of classes, which can be added with negligible engineering effort and computational overhead. Despite its simple nature, this one matrix multiplication provides real impact. We show that our proposal directly boosts classification, long-tailed recognition, out-of-distribution detection, and open-set recognition, from CIFAR to ImageNet. We find empirically that maximum separation works best as a fixed bias; making the matrix learnable adds nothing to the performance. The closed-form implementation and code to reproduce the experiments are provided in the supplementary materials.

1 Introduction

An inductive bias of a learning algorithm describes a set of assumptions about the target function independent of training data [35]. Inductive biases play a vital role in the design of machine learning algorithms, consider for example inductive biases for image structures (*e.g.,* the convolution operator), symmetries (*e.g.,* rotational equivariance), or relational structures (*e.g.,* graph layers). Specifically for categorization, an important and long-standing inductive bias is optimal class separation. Perhaps the most well-known example of the use of this bias is Support Vector Machines, which maximize the margin of the hyperplane between samples of two classes [6]. Given many possible hyperplanes, the rationale is to select the one that represents the largest margin, *i.e.,* separation, known as a maximum-margin classifier. Similarly, boosting algorithms give weights to samples depending on their margin, such as AdaBoost which focuses on low margin samples [13]. In such algorithms and in many more, separation has been incorporated in their design to pull apart the classes, thereby improving generalization in classification settings to unseen data.

In deep learning-based classification, neural networks are by default not equipped with a maximum separation inductive bias. The cross-entropy loss with a softmax function is a common choice to train a classifier, aiming for discriminative power. Yet this framework does not explicitly drive maximum separation between the classes. Liu *et al.* [29] observed that networks by themselves try to decouple inter-class differences based on angles and intra-class variations based on norms. This decoupling is

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

however limited out-of-the-box and a wide range of works have investigated optimization strategies to explicitly force classes away from each other. A common approach is to introduce additional losses which incorporate a notion of class margins, see *e.g.,* [23, 27, 31, 32] or by including orthogonality constraints [23]. Specifically, hyperspherical uniformity has shown to be an effective way to separate classes, whether it be as additional objectives in a softmax cross-entropy optimization [24, 27, 28, 66] or through hyperspherical prototypes [34].

This paper provides an alternative view to embed separation in deep networks with a straightforward solution: encoding maximum separation as an inductive bias in the network architecture itself with one fixed matrix. Rather than treating separation as an optimization problem, we find that the problem can be optimally solved in closed-form. The closed-form solution is given by a recursive algorithm which embeds all class vectors uniformly, such that all are equiangular and with zero mean. The solution holds for any finite number of classes and results in a single matrix that can be added *a priori* as a matrix multiplication on top of any network. The outputs of the proposed embedding are the inner product scores that feed into the softmax function, allowing us to still leverage powerful objectives such as cross-entropy on softmax activations.

We demonstrate that remarkably, adding only one fixed matrix in the neural network provides compelling improvements across various classification tasks. We show that enriching ResNet with a maximum separation inductive bias improves standard classification and especially long-tailed recognition. We also show that our proposal improves methods specifically designed for long-tailed recognition, highlighting its general utility. The results generalize to ImageNet and improves both shallow and deep networks. The added inductive bias helps to balance classes by design, avoiding the under-representation of rare classes. Beyond classification, we illustrate that out-of-distribution and open-set recognition benefit from an embedded maximum separation. Our general-purpose solution comes with no measurable difference in runtime and can be added in one line of code, enabling it to be adopted generally. To highlight the broad applicability of our proposal, we also show how task-specific state-of-the-art approaches for open-set recognition benefit directly when adding the proposed maximum separation. All code is provided in the supplementary materials.

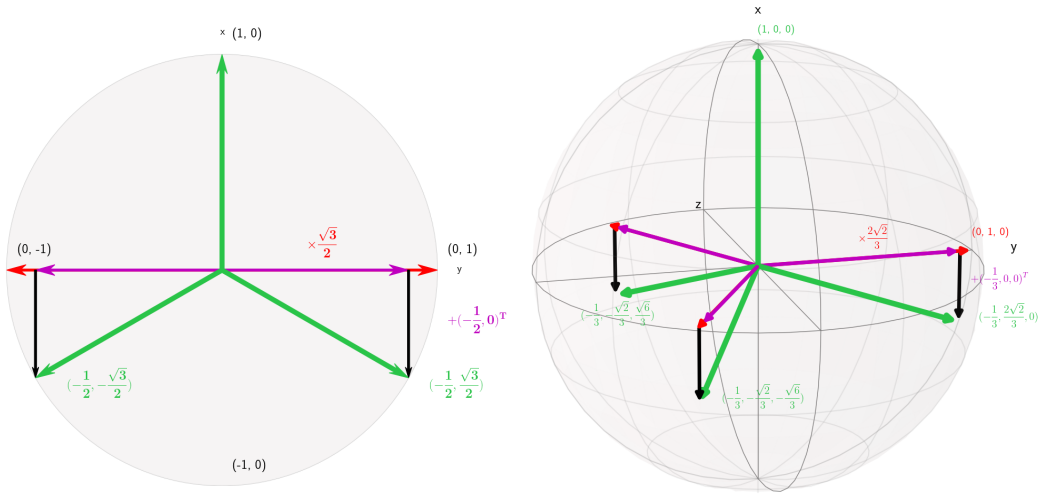
2 Closed-Form Maximum Separation between Classes

Classification constitutes a supervised setting, where we are given a training set of N examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i$ denotes input data at index i and $y_i \in \{1, \dots, C\}$ denotes the corresponding label for one of C classes. Conventionally, this problem can be addressed by feeding inputs through a network, *i.e.,* $\mathbf{o}_i = \Phi(\mathbf{x}_i) \in \mathbb{R}^C$. The outputs denote class logits, which serve as input for a softmax activation followed by a cross-entropy loss.

Our goal is to embed maximum separation as an inductive bias into any network, so that the network no longer needs to learn this behaviour. We do so by adding a single matrix multiplication on top of the output of network $\Phi'(\cdot)$. The matrix itself should contain all necessary information about maximum separation between classes. While the general problem of optimally separating arbitrary numbers of classes on arbitrary numbers of dimensions is an open problem [27, 34], we note that we only need maximum separation of C classes in a particular output space. To that end, let $\hat{\mathbf{x}}_i = \Phi'(\mathbf{x}_i) \in \mathbb{R}^{C-1}$ now denote the output of a network to a $(C-1)$-dimensional output space. We add the matrix multiplication to obtain C-dimensional class logits: $\mathbf{o}_i = P^T \hat{\mathbf{x}}_i$ with $P \in \mathbb{R}^{(C-1)\times C}$. The matrix P denotes a set of C vectors, one for each class, each of $(C-1)$ dimensions. This matrix is fixed such that the class vectors are separated both uniformly and maximally with respect to their pair-wise angles, which we in turn embed on top of a network. Below, we outline the definition and implementation of class vectors on a hypersphere with maximum separation. In what follows, the number of classes is denoted by $k+1 = C$ for notational convenience.

Definition 1. (Maximally separated matrix) *For $k+1$ classes, let $P_k = [p_0, \dots, p_k]$ denote a matrix of $k+1$ column vectors $\{p_i\}_{i=0}^k \subset \mathbb{S}^{k-1}$, such that $\forall_{i,j,k,i\neq j,j\neq k} \langle p_i, p_j \rangle = \langle p_i, p_k \rangle$ and $\sum_{i=0}^k p_i = 0$.*

Here, $\langle \cdot, \cdot \rangle$ denotes the cosine similarity. Definition 1 ensures that all class vectors are as far away from each other as possible based on two requirements. First, the angle between any two class vectors should be the same. Equal pair-wise similarity entails uniformity as any deviation from this setup leads to classes that move closer to each other [46]. Second, the mean over all class vectors should



(a) Recursive update from 2 to 3 classes.

(b) Recursive update from 3 to 4 classes.

Figure 1: **Closed-form separation through recursion.** In (a), we visualize how to obtain maximally separated vectors for three classes on $\mathbb{S}^1$ given two class vectors on the line. In (b), we do the same for the step from three to four classes on $\mathbb{S}^2$. For any new $(k+1)$-th class (red), we introduce a new k-th dimension and position the new prototype along its axis $(1, 0, \ldots, 0)$. All other prototypes are shrunken by a factor $\sqrt{1 - \frac{1}{k^2}}$ (purple) and repositioned such that their center is placed at $(-\frac{1}{k}, 0, \ldots, 0)$ in the hyperplane perpendicular to the new class (green). This recursive approach ensures that all classes remain of unit norm, with equal pair-wise similarity, and with zero mean.

be the origin. This constraint prevents trivial solutions such as an identical embedding of all classes as well as suboptimal embeddings in terms of separation such as one-hot encodings. In short, the definition prescribes that $k+1$ classes can be optimally separated specifically when using k output dimensions. To be able to construct such an embedding, we first need to know what the pair-wise angular similarity is for any fixed number of classes:

Lemma 1. *For maximally separated matrix P_k, $\forall_{i,j,i\neq j} \langle p_i, p_j \rangle = -\frac{1}{k}$.*

Proof. With P_k maximally separated, we have $\sum_{i=0}^{k} p_i = 0$ and hence:

$$0 = \langle p_0, \sum_{i=0}^{k} p_i \rangle = 1 + \sum_{i=1}^{k} \langle p_0, p_i \rangle = 1 + k \langle p_0, p_1 \rangle, \tag{1}$$

since all class vectors are at equal angle to each other. We have $\langle p_0, p_1 \rangle = -\frac{1}{k}$ and again by extension of equidistance: $\forall_{i,j,i\neq j} \langle p_i, p_j \rangle = -\frac{1}{k}$. □

The implication of the lemma above is that an optimal separation of $k+1$ classes on the manifold $\mathbb{S}^{k-1}$ results in classes being separated beyond orthogonality, a direct result of using the full space and different from classical one-hot encodings, which only span the positive sub-space and therefore provide only orthogonal separation. The outcome of the lemma allows us to construct P_k recursively [1].

For any finite number of classes $k+1$, the closed-form solution for P_k is given as:

$$P_1 = \begin{pmatrix} 1 & -1 \end{pmatrix} \in \mathbb{R}^{1\times 2} \tag{2}$$

$$P_k = \begin{pmatrix} 1 & -\frac{1}{k}\mathbf{1}^T \\ \mathbf{0} & \sqrt{1 - \frac{1}{k^2}}\, P_{k-1} \end{pmatrix} \in \mathbb{R}^{k\times(k+1)} \tag{3}$$

The closed-form solution for two classes (P_1) is trivially given as +1 and -1 on the line. For more than 2 classes, $k \geq 2$, we start from the solution given for $k-1$ on $\mathbb{S}^{k-2}$. We add the vector for

the new class as $p_1 = (1, 0, \ldots, 0)$. We place the solution P_{k-1} at $(-\frac{1}{k}, 0, \ldots, 0)$, in the hyperplane perpendicular to p_k, where P_{k-1} is shrunk so that its radius is $\sqrt{1 - \frac{1}{k^2}}$. In this manner, all points in P_k are of radius 1 and uniformly distributed, as proven below. Figure 2 shows the construction of our embeddings for 3 and 4 classes.

Theorem 1. *For any $k \geq 1$, P_k is a maximally separated matrix.*

Proof. We proceed by induction. Note that, for $k = 1$, the angular similarity of the two vectors is clearly given by $-\frac{1}{k} = -1$. Now, assume the theorem holds for $k - 1 \in \mathbb{N}$ and let $(P_n)_{*i}$ denote the i-th column vector of P_n for arbitrary $n \in \mathbb{N}$. Then, for k and $i \neq j$, we obtain the case where either i or j is 1 and the case where neither is 1. For the first case, without loss of generality assume that the i-th element is the first, so that:

$$\left\langle (P_k)_{*i}, (P_k)_{*j} \right\rangle = -\frac{1}{k} + \sqrt{1 - \frac{1}{k^2}} \left\langle \mathbf{0}, (P_{k-1})_{*j} \right\rangle = -\frac{1}{k}.$$

For the second case, we derive:

$$\begin{aligned} \left\langle (P_k)_{*i}, (P_k)_{*j} \right\rangle &= \frac{1}{k^2} + \left(1 - \frac{1}{k^2}\right) \left\langle (P_{k-1})_{*i}, (P_{k-1})_{*j} \right\rangle \\ &= \frac{1}{k^2} - \left(1 - \frac{1}{k^2}\right) \left(\frac{1}{k-1}\right) \\ &= -\frac{1}{k}, \end{aligned}$$

since $\left\langle (P_{k-1})_{*i}, (P_{k-1})_{*j} \right\rangle = -\frac{1}{k-1}$ by assumption. Therefore, the vectors of P_k have a pair-wise angular similarity of $-\frac{1}{k}$. To prove that P_k has zero mean, we again proceed by induction. Note that for P_1, the statement obviously holds. Now, assume the statement holds for $k - 1 \in \mathbb{N}$, then

$$\sum_{i=0}^{k} (P_k)_{*i} = \begin{pmatrix} 1 - k * \frac{1}{k} \\ \sqrt{1 - \frac{1}{k^2}} \sum_{i=0}^{k-1} (P_{k-1})_{*i} \end{pmatrix} = \mathbf{0},$$

since $\sum_{i=0}^{k-1} (P_{k-1})_{*i} = 0$ by assumption. □

For $C = k + 1$ classes, the solution P_k consists of C vectors of unit length. The main idea of this work is to utilize P_k to obtain class logits under a fixed maximum separation, see Figure 2. For sample $\mathbf{x}_i$, we want its network output $\hat{\mathbf{x}}_i = \Phi'(\mathbf{x}_i)$ to align with the vector of its corresponding class. Given P_k, the alignment is maximized if the network output points in the same direction as the class vector. This can be obtained by computing the dot product between both. Since we need to compute the logits for all class vectors in the final softmax activation, we simply augment the network output with a single matrix multiplication: $\mathbf{o}_i = P^T \hat{\mathbf{x}}_i$. By enforcing maximum separation from the start, a network no longer needs to learn this behaviour, essentially simplifying the learning objective to maximizing the alignment between samples and their fixed class vectors. We note that the norm of the class vectors is not taken into account during their construction as we operate on the hypersphere. We can optionally include a radius hyperparameter which controls the norm of the class vectors, *i.e.,* the network output is given as: $\mathbf{o}_i = \rho P^T \hat{\mathbf{x}}_i$ with ρ as hyperparameter. In practice however, the radius of the class vectors is of little influence to the task performance and we set it to 1 unless specified otherwise.

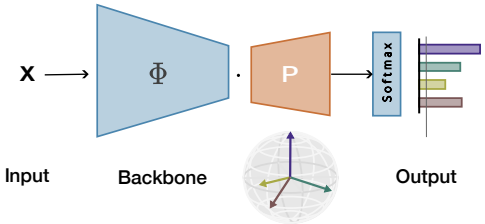


Figure 2: **Maximum separation pipeline.**

3 Experiments

Implementation details. Across our experiments, we train a range of network architectures including AlexNet [20], multiple ResNet architectures [15, 63], and VGG32 [43], as provided by PyTorch [36]. All networks are optimized with SGD with a cosine annealing learning rate scheduler, initial learning rate of 0.1, momentum 0.9, and weight decay 5e-4. For AlexNet we use a step learning rate decay scheduler. All hyperparameters and experimental details are also available in the provided code.

	CIFAR-100					CIFAR-10				
	-	0.2	0.1	0.02	0.01	-	0.2	0.1	0.02	0.01
ConvNet	56.70	45.97	40.34	27.35	16.59	86.68	79.47	73.90	51.40	43.67
+ This paper	**57.05**	**46.59**	**40.44**	**28.27**	**18.40**	**86.76**	**79.63**	**75.88**	**55.25**	**48.05**
	+0.35	+0.62	+0.10	+0.92	+1.81	+0.08	+0.16	+1.98	+3.85	+4.38
ResNet-32	75.77	65.74	58.98	42.71	35.02	94.63	88.17	83.10	68.64	56.98
+ This paper	**76.54**	**66.01**	**60.54**	**45.12**	**38.85**	**95.09**	**91.42**	**88.16**	**77.02**	**69.70**
	+0.77	+0.27	+1.56	+2.41	+3.83	+0.46	+3.25	+5.06	+8.38	+12.72

Table 1: **Adding maximum separation as inductive bias on CIFAR-10 and CIFAR-100** for standard and multiple imbalanced settings. Both an AlexNet-style ConvNet and a ResNet architecture benefit from an embedded maximum separation, especially when imbalance increases and networks are more expressive.

3.1 Classification and long-tailed recognition

Classification with maximum separation. In the first set of experiments, we evaluate the potential of maximum separation as inductive bias in classification and long-tailed recognition settings using the CIFAR-100 and CIFAR-10 datasets. Our maximum separation is expected to improve learning especially when dealing with under-represented classes which will be separated by design with our proposal. We evaluate on a standard ConvNet and a ResNet-32 architecture with four imbalance factors: 0.2, 0.1, 0.02, and 0.01. We set $\rho = 0.1$ for ResNet-32 as this provides a minor improvement over $\rho = 1$. The results are shown in Table 1. For an AlexNet-style ConvNet, classification accuracy improves in all settings, especially when imbalances is highest. With a more expressive network such as a ResNet, we find that embedding maximum separation provides a bigger boost in performance. The higher the imbalance, the bigger the improvements; on CIFAR-100 the accuracy improves from 35.02 to 38.85, on CIFAR-10 from 56.98 to 69.70, an improvement of 12.72 percent point. For further analysis, we have investigated the relation between accuracy improvement and train sample frequency. In Figure 3 we show the results over the imbalance factors, highlighting that our proposal improves all classes as imbalance increases, especially classes with lower sample frequencies.

Should maximum separation be a fixed inductive bias? Separation in this work takes the form of a fixed matrix multiplication that can be plugged on top of any network. But is the information in this matrix as inductive bias optimal for classification? To test this hypothesis, we compare the previous results on CIFAR-100 and CIFAR-10 and their long-tailed variants to two baselines. The first baseline makes the matrix P learnable and our proposal acts as an initialization of the learnable matrix. Intuitively, if the results improve by making the matrix learnable instead of fixed, there is more information that is not captured in the closed-form solution. The second baseline adds a learnable fully-connected layer with standard random initialization instead of a fixed matrix multiplication, to investigate whether our obtained improvements are not simply a result of extra computational efforts.

The results are shown in Figure 4. On both datasets across all imbalance factors, embedding maximum separation as a fixed constraint in a network works best. Making the matrix learnable actually decreases the performance, indicating that further optimization discards important information and highlighting that separation is an inductive bias we should not steer away from. Simply adding an extra learnable layer with standard initialization is not effective, which indicates that improvements

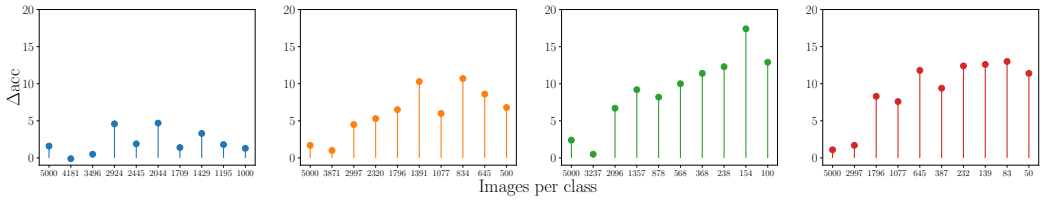
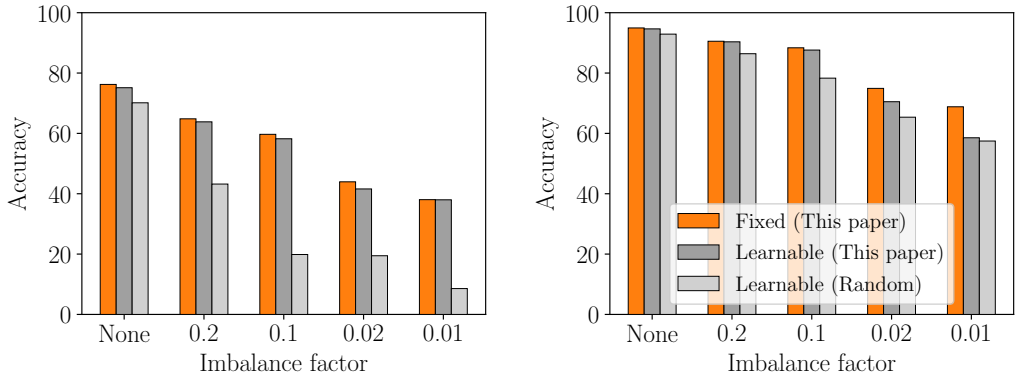


Figure 3: **Effect of maximum separation on class accuracy as a function of sample frequency.** We show the per class accuracy improvements when adding separation on CIFAR-10 with a ResNet-32 for imbalance factors (from left to right) 0.2, 0.1, 0.02, 0.01. The accuracy increases over all classes as imbalance increases, especially classes with lower sample frequencies.



(a) CIFAR-100.

(b) CIFAR-10.

Figure 4: **Should maximum separation be embedded as a fixed inductive bias?** On both CIFAR-100 and CIFAR-10, we find that embedding a fixed separation (green) works best, outperforming baselines that seek to further optimize a matrix initialized with maximum separation and that simply add another learnable matrix initialized like any other layer. Maximum separation should be computed *a priori* and kept fixed in the network architecture.

are not guaranteed when adding more fully-connected layers. We conclude that maximum separation is best added in a rigid manner as strict constraint for networks to adapt to.

Enriching long-tailed recognition approaches. So far, we have shown that conventional network architectures are better off with maximum separation as inductive bias for classification and long-tailed recognition. In the third experiment, we investigate the potential of adding our proposal on top of methods that are specifically designed for long-tailed recognition. We perform experiments on two methods, LDAM [8] and MiSLAS [65]. LDAM addresses learning on imbalanced data through a label-distribution-aware margin loss which replaces the conventional cross-entropy loss. We can easily augment LDAM by placing the matrix on top of the used backbone. Their hyperparameter μ in LDAM is set to 0.5 when using their method as is and set to 0.4 when augmented with our proposal as this maximizes the scores for both settings. MiSLAS is a recent state-of-the-art approach for long-tailed recognition that improves two-stage methods by means of label-aware smoothing and shifted batch normalization. We again plug in our approach in the provided code simply as a fixed matrix multiplication on top of the used backbone.

In Table 2, we report the results on CIFAR-100 using a ResNet-32 backbone across three imbalance factors. For LDAM, we find that both for the SGD and the DRW variants, adding maximum separation on top improves the accuracy. For imbalance factor 0.01 for LDAM, we improve the accuracy from 39.87 to 42.02 with the SGD variant and from 42.37 to 43.19 for the DRW variant. The recent MiSLAS approach to long-tailed recognition also benefits from maximum separation after both stages. The most competitive results after their stage 2 are improved by 1.59, 0.83, and 0.42 percent point for respectively imbalance factors 0.1, 0.02, and 0.01. The long-tailed baselines are specifically designed to address class imbalance but additionally benefit from maximum separation across the board. We conclude that maximum separation as a fixed inductive bias complements task-specific approaches, highlighting its general applicability and usefulness.

ImageNet experiments. To highlight that our proposal is also beneficial with many classes and deeper networks, we provide results on ImageNet for two ResNet backbones in Table 3. With a ResNet-50 backbone, we obtain improvements of 1.6 percent point on ImageNet and 3.5 percent point on its long-tailed variant in top 1 accuracy. With a deeper ResNet-152, the improvements are 0.6 and 1.4 percent points for the standard and long-tailed benchmarks. We also find that the top 5 accuracy improves from 92.4 to 94.9 with ResNet-50 and from 94.3 to 95.1 with ResNet-152. We conclude that the proposed inductive bias in matrix form is also useful for larger-scale classification.

	Imbalance factor				Imbalance factor		
	0.1	0.02	0.01		0.1	0.02	0.01
LDAM-SGD	55.05	43.85	39.87	MiSLAS (stage 1)	58.36	44.69	40.29
+ This paper	**57.72**	**45.14**	**42.02**	+ This paper	**59.63**	**45.65**	**40.56**
	+2.67	+1.29	+2.20		+1.27	+0.96	+0.27
LDAM-DRW	57.45	47.56	42.37	MiSLAS (stage 2)	61.93	52.53	48.00
+ This paper	**58.37**	**48.02**	**43.19**	+ This paper	**63.52**	**53.36**	**48.42**
	+0.92	+0.46	+0.82		+1.59	+0.83	+0.42

Table 2: **Enriching imbalanced learning algorithms with our maximum separation** on CIFAR-100 for imbalance factors 0.1, 0.02, and 0.01. We perform experiments with two variants of LDAM of [8] and with the recent MiSLAS of [65] after both stages of their approach. All results are obtained by running the author-provided code. Across all methods and imbalances factors, we find that a maximum separation inductive bias provides a consistent boost in accuracy.

	Resnet-50				**Resnet-152**			
	top 1		top 5		top 1		top 5	
	Base	+ Ours	Base	+ Ours	Base	+ Ours	Base	+ Ours
Imagenet	73.2	**74.8**	92.4	**94.9**	77.9	**78.5**	94.3	**95.1**
Imagenet-LT	43.8	**47.3**	70.4	**73.6**	48.3	**49.7**	73.9	**74.8**

Table 3: **Evaluations on ImageNet and ImageNet-LT** for two ResNet architectures. Both the top 1 and top 5 accuracies improve on ImageNet when embedding maximum separation on top of the architectures, with a further boost for the long-tailed ImageNet variant.

3.2 Out-of-distribution detection and open-set recognition

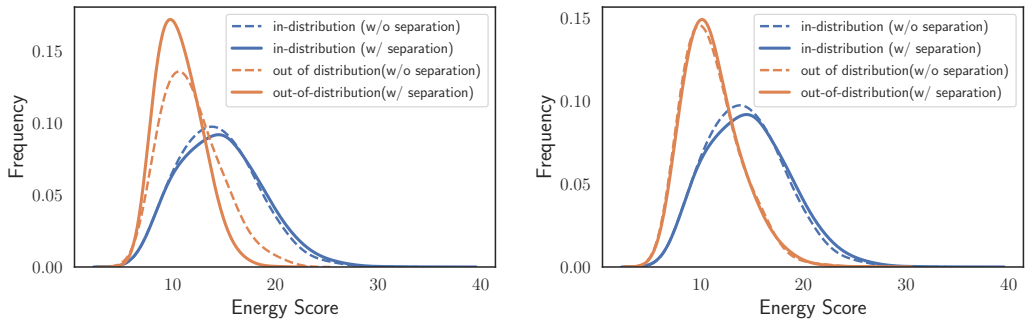
Beyond classification, we also investigate the potential of our proposal outside the closed set of known classes. To that end, we perform additional experiments on out-of-distribution detection and open-set recognition.

Out-of-distribution with maximum separation. Intuitively, maximum separation and uniformity between classes allows for more opportunities for samples outside the distribution of known classes to fall in between the spaces of class vectors, especially since all classes are positioned beyond orthogonal to each other as per Lemma 1. We evaluate this intuition first on out-of-distribution detection. Following Liu et.al. [26], we use CIFAR-100 as in-distribution set and experiment with SVHN and Placed 365 as out-of-distribution sets. Using the same experiment settings reported in their paper, we train on a WideResNet and compare the out-of-distribution performance on the common metrics FPR95, AUROC, and AUPR. We determine whether a sample is in- or out-of-distribution using three scoring functions. The first directly uses the maximum softmax probability to distinguish in- and out-of-distribution samples [17]. The second is the energy score as introduced in Liu *et al.* [26], which classifies an input as out-of-distribution if the negative energy score is smaller than a threshold value, The third is the confidence score based on Mahalanobis distance from Lee *et al.* [21].

The results are shown in Table 4. With SVHN as out-of-distribution set, embedding a maximum separation inductive bias improves all metrics across all out-of-distribution detection methods. Especially the energy score benefits from maximum separation with an improvement of 6.90 in FPR95, 11.48 in AUROC, and 3.01 in AUPR. On Places 365, the results are overall closer. Maximum separation improves on the AUROC and AUPR metrics but not the FPR95 metric. We observe that adding maximum separation performs best for near out-of-distribution data. In Figure 5 we provide further analyses on the effect of maximum separation for out-of-distribution detection with energy scores, highlighting that separation increases the gap in energy scores between both sets, making it easier to discriminate both. We conclude that a maximum separation inductive bias is valuable for out-of-distribution detection, especially for near out-of-distribution data.

	Metric	w/o maximum separation			w/ maximum separation		
		FPR95 ↓	AUROC ↑	AUPR ↑	FPR95 ↓	AUROC ↑	AUPR ↑
SVHN	Softmax Score	84.00	71.40	92.87	**83.04**	**75.58**	**94.59**
	Energy Score	85.76	73.94	93.91	**78.86**	**85.42**	**96.92**
	Mahalanobis	44.02	90.48	97.83	**35.88**	**91.45**	**97.93**
Places 365	Softmax Score	**82.86**	73.46	93.14	83.08	**73.53**	**93.54**
	Energy Score	**80.87**	75.17	93.40	81.36	**76.01**	**93.64**
	Mahalanobis	**88.83**	67.87	90.71	89.16	**69.33**	**91.49**

Table 4: **Out-of-distribution detection on CIFAR-100** with and without maximum separation embedded in a Wide ResNet, using SVHN and Placed 365 as out-of-distribution datasets. On SVHN we observe improvements across all matrics and measures. On Places 365, out-of-distribution detection is better for the AUROC and AUPR metrics, but not the FPR95.



(a) SVHN as OOD.

(b) Places365 as OOD.

Figure 5: **Energy scores for out-of-distribution detection.** We plot the inverse of the energy scores for ease of visualization. With SVHN as out-of-distribution dataset for CIFAR-100 as in-distribution dataset, we find that embedding maximum separation increases the gap in energy scores between both datasets, making it easier to differentiate between both. With Places 365 as out-of-distribution dataset, the energy scores are affected less, resulting in smaller improvements with maximum separation.

Open-set recognition with maximum separation. We also investigate the potential of maximum separation for open-set recognition, which can be viewed as a generalization of out-of-distribution detection [33]. To take our proposal to the ultimate test, we start from the recent state-of-the-art work of Vaze *et al.* [51]. We experiment with adding our proposal to two approaches outlined in their work. The first applies various training tricks and hyperparameter tuning to improve the closed-set classifier, followed by open-set recognition using the maximum softmax probability (MSP+) score. The second performs the same training with the Maximum Logits Score (MLS) before the softmax activations to differentiate closed- from open-set examples. All experiments are run on author-provided code on four benchmarks: SVHN, CIFAR10, CIFAR + 10, and CIFAR + 50, where we use the 5 splits and report the average scores as outlined by Vaze *et al.* [51].

The results of the open-set experiments are shown in Table 5 with the AUROC metric. For both the MSP+ and the state-of-the-art MLS scores, maximum separation is a positive addition. This result follows the intuition of the original approach that improving the closed-set classifier – here by means of the extra inductive bias – is also beneficial for open-set recognition.

4 Related work

In deep learning literature, the principle of separation has been investigated from several perspectives. Separation can be achieved by enforcing or regularizing orthogonality. For example, Li et al. [23] propose to regularise class vectors orthogonal to each other. Orthogonality is also often used as regularisation for intermediate network layers, see *e.g.,* [3, 38, 39, 55, 60, 61]. Orthogonal constraints are fixed and easy to obtain by design. For classification specifically however, orthogonality does not

	SVHN	CIFAR10	CIFAR + 10	CIFAR + 50
MSP+	95.94	90.10	94.48	93.58
+ This paper	**96.22**	**91.05**	**95.57**	**94.31**
	+0.28	+0.95	+1.09	+0.73
MLS	97.10	93.82	97.94	96.48
+ This paper	**97.58**	**95.30**	**98.33**	**96.74**
	+0.48	+1.48	+0.39	+0.26

Table 5: **Open-set recognition on four benchmarks** with and without maximum separation building on the experimental settings of Vaze et.al. [51]. Whether using the maximum probability or the recently proposed maximum logit to perform open-set recognition, a fixed maximum class separation helps to differentiate closed-set from open-set samples.

result in a maximal separation between classes, as only the positive sub-space is utilized. We strive for a solution that makes use of the entire output space.

Beyond orthogonality, multiple works improve class separation by promoting angular diversity between classes. A common approach to do so is by incorporating class margin losses [11, 30, 32, 54, 64]. For example, the generalized large-margin softmax loss enforces both intra-class compactness and inter-class separation [31]. These works are in line with the observations of Liu *et al.* [29] that deep networks have a natural tendency to break classification down into maximizing class separation and minimizing intra-class variance. In particular, separation has in recent years been successfully tackled by adopting the perspective of hyperspherical uniformity. Such uniformity can for example be optimized in deep networks by minimizing the hyperspherical energy between class vectors [24, 27, 28]. Recently, Zhou *et al.* [66] extend this line of work by learning towards the largest margin with a zero-centroid regularization. Hyperspherical uniformity can also be approximated by minimizing the maximum cosine similarity between pairs of class vectors [14, 34, 57]. In this work, we also start from hyperspherical uniformity and deviate from current literature on one important axis: hyperspherical uniformity does not require optimization. We show and prove that maximum separation has a closed-form solution and can be added as a fixed matrix to any network. Having maximum separation as a fixed inductive bias does not require additional hyperparameters and comes with minimal engineering effort, lowering the barrier towards broad integration in the field.

Our work fits into a broader tendency of incorporating inductive biases into deep learning, for example by incorporating rotational symmetries [10, 12, 22, 42, 45, 47, 50, 53, 59], scale equivariance [44, 58], gauge equivariance [9, 16] graph structure [4, 7, 18, 52], physical inductive biases [2, 19, 37, 40], symmetries between reinforcement learning agents [5, 25, 49, 56], and visual inductive biases [41, 48, 62]. Embedding maximum separation as inductive bias is complementary to the listed works.

5 Conclusions

This paper strives to embed a well-known inductive bias into deep networks: separate classes maximally. Separation is not an optimization problem and can be solved optimally in closed-form. We outline a recursive algorithm to compute maximum class separation and add it as a fixed matrix multiplication on top of any network. In this manner, maximum separation can be embedded with negligible effort and computational complexity. To showcase that our proposal is an effective building block for deep networks, we perform various experiments on classification, long-tailed recognition, out-of-distribution detection, and open-set recognition. We find that our solution to maximum separation as inductive bias (i) improves classification, especially when classes are imbalanced, (ii) is best treated as a fixed matrix, (iii) improves standard networks and task-specific state-of-the-art algorithms, and (iv) helps to detect samples from outside the training distribution. With separation decoupled and solved prior to training, a network only needs to optimize the alignment between samples and their fixed class vectors. We conclude that our one fixed matrix is an effective, broadly applicable, and easy-to-use addition to classification in networks. **Limitations.** Maximum separation as inductive bias applies to supervised settings and does not generalize to settings that require relational information between class vectors, such as zero-shot learning.

References

[1] https://math.stackexchange.com/questions/714711/how-to-find-n1-equidistant-vectors-on-an-n-sphere/714781714781. 3

[2] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *NeurIPS*, 2019. 9

[3] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *NeurIPS*, 2018. 8

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv*, 2018. 9

[5] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *ICLR*, 2020. 9

[6] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992. 1

[7] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J. Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. *ICLR*, 2022. 9

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 6, 7

[9] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *ICML*, 2019. 9

[10] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016. 9

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 9

[12] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International conference on machine learning*, pages 1889–1898. PMLR, 2016. 9

[13] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS*, 1997. 1

[14] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *NeurIPS*, 2021. 9

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[16] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *NeurIPS*, 2021. 9

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017. 7

[18] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *ICLR*, 2020. 9

[19] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *ICLR*, 2020. 9

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 4

[21] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 7

[22] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *NeurIPS*, 2018. 9

[23] Xiaoxu Li, Dongliang Chang, Zhanyu Ma, Zheng-Hua Tan, Jing-Hao Xue, Jie Cao, Jingyi Yu, and Jun Guo. Oslnet: Deep small-sample classification with an orthogonal softmax layer. *TIP*, 2020. 2, 8

[24] Rongmei Lin, Weiyang Liu, Zhen Liu, Chen Feng, Zhiding Yu, James M Rehg, Li Xiong, and Le Song. Regularizing neural networks via minimizing hyperspherical energy. In *CVPR*, 2020. 2, 9

[25] Iou-Jen Liu, Raymond A. Yeh, and Alexander G. Schwing. PIC: Permutation invariant critic for multi-agent deep reinforcement learning. In *CoRL*, 2019. 9

[26] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 2020. 7

[27] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. *NeurIPS*, 2018. 2, 9

[28] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *AISTATS*, 2021. 2, 9

[29] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M Rehg, and Le Song. Decoupled networks. In *CVPR*, 2018. 1, 9

[30] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 9

[31] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 2, 9

[32] Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. Deep hyperspherical learning. *NeurIPS*, 2017. 2, 9

[33] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 8

[34] Pascal Mettes, Elise van der Pol, and Cees G M Snoek. Hyperspherical prototype networks. *NeurIPS*, 32, 2019. 2, 9

[35] Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . . , 1980. 1

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 4

[37] David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2020. 9

[38] Kanchana Ranasinghe, Muzammal Naseer, Munawar Hayat, Salman Khan, and Fahad Shahbaz Khan. Orthogonal projection loss. In *ICCV*, 2021. 8

[39] Pau Rodríguez, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. *ICLR*, 2017. 8

[40] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 2018. 9

[41] Zenglin Shi, Pascal Mettes, Subhransu Maji, and Cees G M Snoek. On measuring and controlling the spectral bias of the deep image prior. *IJCV*, 2022. 9

[42] Gregor NC Simm, Robert Pinsler, Gábor Csányi, and José Miguel Hernández-Lobato. Symmetry-aware actor-critic for 3d molecular design. *arXiv*, 2020. 9

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 4

[44] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. *arXiv preprint arXiv:1910.11093*, 2019. 9

[45] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant transformer networks. In *International Conference on Machine Learning*, pages 6086–6095. PMLR, 2019. 9

[46] Pieter Merkus Lambertus Tammes. On the origin of number and arrangement of the places of exit on the surface of pollen-grains. *Recueil des travaux botaniques néerlandais*, 1930. 2

[47] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv*, 2018. 9

[48] Nergis Tomen, Silvia-Laura Pintea, and Jan Van Gemert. Deep continuous networks. In *ICML*, 2021. 9

[49] Elise van der Pol, Herke van Hoof, Frans A Oliehoek, and Max Welling. Multi-agent mdp homomorphic networks. *ICLR*, 2022. 9

[50] Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. In *NeurIPS*, 2020. 9

[51] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: a good closed-set classifier is all you need? In *ICLR*, 2022. 8, 9

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 9

[53] Dian Wang, Robin Walters, Xupeng Zhu, and Robert Platt. Equivariant q learning in spatial action spaces. In *CoRL*, 2022. 9

[54] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018. 9

[55] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *CVPR*, 2020. 8

[56] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *ICLR*, 2018. 9

[57] Zhennan Wang, Canqun Xiang, Wenbin Zou, and Chen Xu. Mma regularization: Decorrelating weights of neural networks by maximizing the minimal angles. *NeurIPS*, 2020. 9

[58] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *NeurIPS*, 2019. 9

[59] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *CVPR*, 2017. 9

[60] D. Xie, J. Xiong, and S. Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *CVPR*, 2017. 8

[61] Pengtao Xie, Yuntian Deng, Yi Zhou, Abhimanu Kumar, Yaoliang Yu, James Zou, and Eric P. Xing. Learning latent space models with angular constraints. In *ICML*, 2017. 8

[62] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *NeurIPS*, 2021. 9

[63] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *BMVC*, 2016. 4

[64] Yutong Zheng, Dipan K Pal, and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *CVPR*, 2018. 9

[65] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, 2021. 6, 7

[66] Xiong Zhou, Xianming Liu, Deming Zhai, Junjun Jiang, Xin Gao, and Xiangyang Ji. Learning towards the largest margins. In *ICLR*, 2022. 2, 9

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]