MAXIMIZING ENTROPY ON ADVERSARIAL EXAMPLES CAN IMPROVE GENERALIZATION

Amrith Setlur[1] **Benjamin Eysenbach**[1] **Virginia Smith**[1] **Sergey Levine**[2]
[1] Carnegie Mellon University [2] UC Berkeley
asetlur@cs.cmu.edu

ABSTRACT

Supervised classification methods that directly optimize maximize the likelihood of the training data often overfit. This overfitting is typically mitigated through regularizing the loss function (e.g., label smoothing, weight decay) or by minimizing the same loss on new examples (e.g., data augmentation, adversarial training). In this work, we propose a complementary regularization strategy: training the model to be unconfident on examples that are generated so they have unclear labels. We call our approach Maximum Predictive Entropy (MPE). These automatically generated examples are cheap to compute, so our method is only 30% slower than standard data augmentation. Adding MPE to existing regularization techniques, such as label smoothing, increases test accuracy by $1-3\%$, with larger gains in the small data regime.[1].

1 INTRODUCTION

Prior work has proposed a number of highly-effective strategies for improving test performance like training the model on additional examples (e.g., augmented examples Shorten & Khoshgoftaar (2019), adversarial examples Volpi et al. (2018)). Replacing the standard maximum likelihood loss (i.e., cross entropy for classification) with alternative loss functions (e.g., label smoothing Müller et al. (2019), MixUp **?**, robust classification losses Madry et al. (2017)) can also improve generalization. In effect, these prior methods either make the model's predictions more certain on new training examples or make the distribution over potential models less certain.

In this paper we approach the problem from a different perspective: making the model's predictions less certain on new algorithmically derived training examples. The generation of new examples structurally resembles adversarial training, but these examples are used differently. Standard adversarial training (Madry et al., 2017; Miyato et al., 2018) assigns each adversarial example the same label as an unperturbed example, and includes these new training examples in the cross entropy loss. In effect, these methods train the model to be *more* confident on these adversarial examples. Typically, adversarial training provides some benefits, but *decreases* in-distribution test accuracy (Raghunathan et al., 2019; Zhang et al., 2019; Tsipras et al., 2019), the main focus of this paper. Might there be a way to use these adversarial examples with *increase* in-distribution test accuracy?

The main contribution of this work is a loss function for classification that decreases the generalization gap. We show that our approach, Maximum Predictive Entropy (MPE), improves generalization across a range of image classification tasks. Importantly, we show that the benefits of our method are complementary to prior methods, such as strong image augmentations, label smoothing, MixUp training and gradient clipping. Applying our technique on top of these existing techniques yields improved performance. Our method is easy to implement and computationally efficient (only 30% slower than standard Empirical Risk Minimization (ERM) training but 200% faster than multi-step adversarial training (Madry et al., 2017)). While the aim of our work is to improve in-distribution test accuracy, we also show that our method can increase robustness to out-of-distribution examples.

[1]Code for this work can be found at `https://github.com/ars22/MPE-regularizer`.

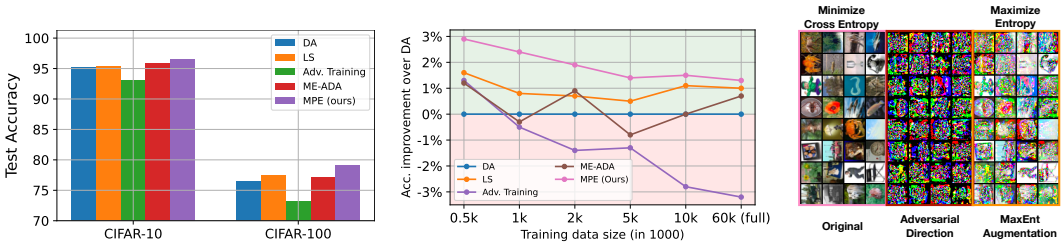


Figure 1: **Increasing test accuracy by maximizing predicting entropy**: We propose a simple and computationally efficient regularization technique: Maximum Predictive Entropy (MPE) that maximizes a model's predictive entropy on adversarial examples. In (a) we show that our approach achieves greater test accuracy than prior methods on CIFAR-10 and CIFAR-100. In (b) we compare the performance improvements over data augmentation (DA), furnished by MPE and baseline methods as we increase the training data size of CIFAR-100. Finally in (c), we show the training examples (left) on which our method minimizes cross entropy, the adversarial direction (center) which looks like a mixture of high frequency components and the adversarial examples (right) over which we propose to increase model uncertainty (predictive entropy).

2 Regularization via Maximizing Predictive Entropy

Notation. We are given an *iid* sampled training dataset $\mathcal{S} \coloneqq \{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathcal{X} \subset \mathbb{R}^d$, $y_i \in \mathcal{Y} = [L]$. The training examples are assumed to be sampled from an underlying distribution $\mathcal{D}$ with density $p_{\mathcal{D}}$ over $\mathcal{X} \times \mathcal{Y}$. In parametric supervised learning, we train a parameterized model $\theta \in \Theta$ which defines the model's conditional predictive distribution over labels: $p_\theta(y|x)$. The standard loss function is the negative log-likelihood, which is equivalent to the cross entropy loss in the case of classification $\mathcal{L}(\theta) \triangleq -\sum_{i\in[N]} \frac{1}{N} \log p_\theta(y_i|x_i)$. The aim of maximum likelihood estimation is to find the model parameters $\hat{\theta}$ that optimize the following objective: $\hat{\theta} \triangleq \arg\min_\Theta \mathcal{L}(\theta)$. Solely optimizing the log-likelihood of the *training* data often leads to overfitting. We will measure overfitting using the *generalization gap*: the difference between the accuracy on the train set versus the true accuracy computed over $\mathcal{X} \times \mathcal{Y}$ as measured by $\mathcal{D}$: $\frac{1}{N}\sum_{(x_i,y_i)\in\mathcal{S}} \mathbb{1}(y = \arg\max_{y'} p_\theta(y' \mid x)) - \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}(y = \arg\max_{y'} p_\theta(y' \mid x))\right]$. As the training accuracy is typically higher than the accuracy on the true underlying data distribution, this generalization gap is positive, and our aim is to decrease it towards zero.

A large body of prior work has regularized training to prevent overfitting. Some methods regularize the training *process* (e.g., early stopping (Yao et al., 2007), novel optimizers (Neyshabur et al., 2015; Ji et al., 2021) and dropout (Srivastava et al., 2014)) while other works explicitly add an additional regularization term $\mathcal{C}(\boldsymbol{\theta})$ weighted by scalar λ to the log-likelihood objective:

$$\hat{\theta}_\lambda \triangleq \arg\min_\Theta \mathcal{L}(\theta) + \lambda \cdot \mathcal{C}(\theta) \tag{1}$$

Examples of such penalties include weight decay, gradient norms etc.. The main contribution of this paper is a new data-dependent regularization term, which will depend not just on the model parameters but also on the training dataset. Our proposed regularizer will depend on the model's *predictive entropy*, which is the conditional entropy of the distribution $p_\theta(y|x)$ at any given x: $\mathcal{H}_\theta(x) \triangleq -\int_{\mathcal{Y}} p_\theta(y \mid x) \log p_\theta(y \mid x) dy$. Prior work on *robustness* has augmented the training dataset $\mathcal{S}$ with additional, *adversarial examples*. These adversarial examples can be generated in many ways (Sabour et al., 2016; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Kurakin et al., 2016). Given a (real) training example $(x, y) \sim \mathcal{S}$, an adversarial example is generated by finding a nearby example x^* with higher loss on the same label:

$$x^* = \underset{x' :\, \|x'-x\|_1<\epsilon}{\arg\max} -\log p_\theta(y \mid x'), \tag{2}$$

where $\epsilon > 0$ is a hyperparameter controlling the strength of the adversarial attack. Adversarial examples have been used in many different contexts. Perhaps the most common use is *adversarial training* where the model is trained to be robust to the very attack used to generate the adversarial example at test time Madry et al. (2017). Such adversarial training tends to improve a model's robustness to out-of-distribution and adversarial examples, but typically decreases the model's test

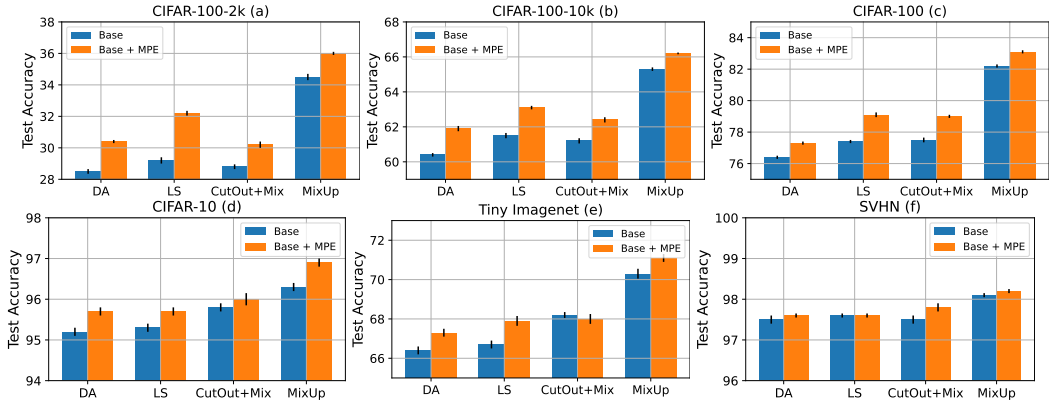


Figure 2: **Main results on various supervised image classification benchmarks:** We plot the test accuracies (averaged over 10 runs) of models trained with base methods: Data Augmentation (DA), Label Smoothing (LS), CutOut+CutMix data augmentation, MixUp training and compare them with the test accuracies of the models trained with the MPE objective. The regularizer of our objective is given by equation 3 and the full objective by equation 1. Error bars indicate 95% confidence intervals computed over 10 runs with different random seeds.

accuracy Rice et al. (2020). Our method will use these adversarial examples in a different way, which will *increase* the model's test accuracy.

2.1 Regularizing predictive uncertainty on new examples

Deep neural networks have a tendency to overfit to spurious features in the training examples (Zhang et al., 2021). Thus, a natural question to ask is: *If we wish to use adversarial examples as additional training examples, what label should we give them?*

Typical adversarial training assigns the adversarial examples the same label as the pre-corrupted example. Visualizing the adversarial examples in Figure 1(c), it is unclear whether this is the correct choice of label. Following the principle of maximum entropy (Jaynes, 1957), we propose to label adversarial examples with a *uniform* distribution over labels. Our proposed regularizer miminizes the Kullback-Leibler divergence between $p_\theta(y|x - \beta \cdot \nabla_x \log p_\theta(y \mid x))$ and $y_{\text{unif}} \triangleq \left(\frac{1}{L}, \frac{1}{L}, \cdots\right)$. Thus, we name our proposed objective in equation 3: Maximum Predictive Entropy (MPE), since it forces the model's predictions to be highly uncertain on adversarial examples.

$$\mathcal{C}(\theta) = -\sum_{x_i \in \mathcal{S}} \mathcal{H}_\theta(x_i - \beta \cdot \nabla_{x_i} \log p_\theta(y \mid x_i)). \tag{3}$$

Why should maximizing entropy on adversarial examples shrink generalization error? Since adversarial examples are constructed by computing the gradient of the loss, they would naturally lie closer to the decision boundary in comparison to the un-corrupted example. Typically, taking a step along these adversarial directions removes low-level features (see center block in Figure 1(c)) which are responsible for the activation of hidden units when the original image is passed through the network in the first place. Forcing the model to have a higher entropy over such examples would require the features removed to be actually predictive of the true label, and not some spurious noise that happens to be correlated with the true label in high dimensions.

Furthermore, if such a decision boundary existed because the model was overfitting on noise, then increasing entropy at the adversarial example would move the spuriously constructed decision boundary closer to the original example – preventing it from relying on spurious features anymore. If the gradient is in the direction of non-spurious feature, then the small value of α ensures that maximizing predictive entropy of the model at this new sample would have an effect similar to that of label smoothing on the un-corrupted example. On the other hand, if the adversarial example for a given image was indeed generated by removing spurious features from the image then the model would fail to increase entropy on the resulting image without increasing the cross-entropy loss over the un-corrupted image.

Figure 3: **How should you use adversarial examples to improve test accuracy?** Comparison of our method with adversarial data augmentation (ADA), Max-Entropy Adversarial Data Augmentation (ME-ADA) and Adversarial Training on *(Left)* hard and *(Right)* easy classification benchmarks.

3 EXPERIMENTS

The main aim of our experiments is to show that maximizing predictive entropy is an effective regularization strategy, with effects *complementary* to existing regularization techniques. That is, we aim to show that adding our method *on top* of prior methods boosts performance. We will also directly compare to other prior methods that use adversarial examples in different ways. Apart from the reductions in generalization gaps, we find that models trained with our MPE objective yield additional benefits of robustness to certain adversarial samples and test time distribution shifts. Finally, we compare the training accuracy convergence rates for our method with label smoothing and standard ERM training, identifying a trend that may be an interesting direction of future research to further analyze the benefits of our method.

Baselines. The primary aim of our experiments is to study whether entropy maximization shrinks the generalization gap. Hence, we compare our regularizer to existing methods that *(i)* directly constrain the model's predictive distribution (label smoothing (Müller et al., 2019)) or *(ii)* implicitly regularize the model by training on additional images. These additional training images may be generated through data augmentation strategies (e.g., CutMix (Yun et al., 2019), CutOut (DeVries & Taylor, 2017)) through MixUp training Zhang et al. (2018), or using adversarial augmentations (Volpi et al., 2018). Since our objective structurally resembles adversarial training Madry et al. (2017), we add it as a baseline in addition to Maximum Entropy Adversarial Data Augmentation (ME-ADA) Zhao et al. (2020).

3.1 HOW EFFECTIVELY DOES MPE IMPROVE TEST ACCURACY?

Figure 2 presents the main empirical findings. Each bar corresponds to the mean of ten random seeds, and error bars depict the standard deviation. Across all four datasets, we observe that our proposed regularizer improves the baseline method in almost all cases. For example, on CIFAR-100 with 2k training examples, adding our method on top of label smoothing boosts the test accuracy by $\approx 3\%$. The gains from our method are more pronounced on benchmarks where the number of training samples are low: CIFAR-100-2k and CIFAR-100-10k. To test the statistical significance, we computed 1-sided p-values to test the hypothesis that our method achieves higher test accuracy than the baseline: in all cases, the p-values are ≤ 0.03, indicating that our findings are statistically significant. In summary, these results show that our proposed regularizer is complementary with prior methods. While prior methods such as MixUp tend to outperform label smoothing, a combination of MixUp and our MPE objective outperforms both. Similarly, a combination of our method and label smoothing outperforms vanilla label smoothing.

Our next set of experiments compares different ways of using adversarial examples. While our method maximizes the model's predictive entropy on these examples, other methods we compare against directly minimize the cross entropy loss on these adversarial examples (see Appendix C). We compare with adversarial training (Madry et al., 2017), adversarial data augmentation (ADA) (Volpi et al., 2018), and Maximum Entropy Adversarial Data Augmentation (ME-ADA) (Zhao et al., 2020). We evaluate all methods using the standard test accuracy, noting that some of these methods were proposed to optimize robustness, a different metric. We show the results on six benchmarks in Figure 3. On most of these benchmarks, our method outperforms the baselines. The difference from baselines is most pronounced in settings where all methods achieve low test accuracy. For example, on CIFAR-100 with 2,000 training examples, our method achieves a test accuracy that is $+3\%$ better than standard adversarial training, and $+1\%$ better than ME-ADA.

4 CONCLUSION

In this paper, we proposed a regularization technique based on the idea of maximizing a model's predictive entropy on adversarial samples. Through extensive experiments, we showed that this technique can increase test accuracy in a wide range of settings, and can readily be combined with prior regularization techniques. We also demonstrated that our method comes with some small-but-noticeable robustness benefits, benefits lacking from typical ERM methods (see Appendix B). Given that our method is computationally efficient and easy to implement, we believe that it may serve as a useful tool for practitioners, and a simple baseline for researchers.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *stat*, 1050:27, 2020.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

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

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in Neural Information Processing Systems*, 32:3496–3506, 2019.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-sgd: path-normalized optimization in deep neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2422–2430, 2015.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. In *ICLR (Poster)*, 2016.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv: Machine Learning*, 2019.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *arXiv preprint arXiv:1805.12018*, 2018.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

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.

Hongyang R. Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, 33:14435–14447, 2020.

A APPENDIX

A.1 MOTIVATING OUR METHOD VIA A TOY EXAMPLE

To provide some intuition about how our objective prevents the model from overfitting on spurious features, we present results on a toy classification problem in high-dimensions. Typically, in classification problems the true generalizable features span a lower dimensional space (compared to the ambient dimension) (Arjovsky et al., 2020). Ideally, we would learn a classifier that is only sensitive to these few generalizable features, and independent of all the other, spurious features. However, training neural networks with the cross-entropy loss and SGD often leads to overfitting: the model picks up on spurious and noisy features that are randomly correlated with the label Peters et al. (2016); Heinze-Deml & Meinshausen (2017). To simulate this phenomenon, we use a $d-$dimensional toy classification problem where the true features are given by the first two-dimensions only, while the rest of the $d-2$ dimensions are pure noise.

A dataset $\mathcal{S}$ of 20,000 training examples is generated by first sampling a two-dimensional sample $\tilde{x}$ with equal probability from one of two classes supported over two different well separated moon shaped regions (see Figure 4). Here, $\tilde{x}$ is a two dimensional vector with label $y \in \{0, 1\}$. In order to simulate a $d-$dimensional ($d = 625$) classification problem using a two dimensional sample, we first append to each sample a vector of 623 ($d-2$) zeros, and then add add Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$ where $\sigma = 0.5$:

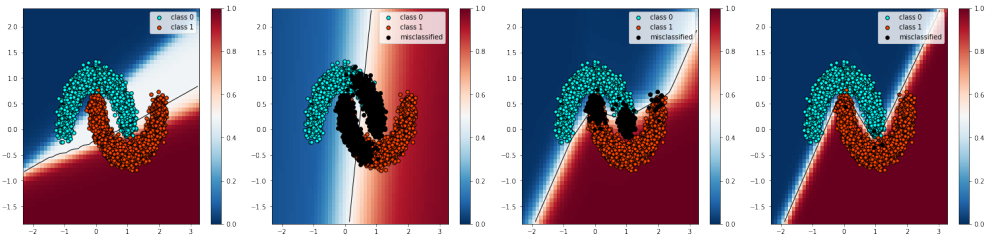
$$\tilde{x} \leftarrow (\tilde{x}, \underbrace{0, \cdots, 0}_{\text{d-2=623 zeros}}) + \epsilon.$$

We will measure the in-distribution performance of a classifier, training and evaluating on examples from this same distribution.

In this example, the first two dimensional of the data perfectly explain the class label; we call these dimensions the generalizable features. We are interested in classification models that successfully identify these generalizable features, while ignoring the remaining, spurious features

Models trained with the standard cross entropy loss on a fixed dataset $\mathcal{S}$ are liable to overfit (Zhang et al., 2021). A model can best reduce the (empirical) cross entropy loss by learning features that span all dimensions, including the spurious feature dimensions. Precisely, the model can reduce the cross entropy loss on example x_i by aligning some of the weights of its hidden units along the direction of the corresponding noise ϵ_i.

We study this phenomenon by training neural networks on this dataset. Each is trained for 300 epochs of SGD. To start, we train a model using the standard empirical cross entropy loss. This model achieves perfect training accuracy (100%), but performs poorly on the validation set (80.4%). To visualize the learned model, we project the decision boundary on the first two coordinates, the only ones that are truly correlated with the labels (on population data). The decision boundary for this model, shown in Figure 4a, is quite different from the true decision boundary. Rather than



(a) ERM (b) MPE (100 epochs) (c) MPE (200 epochs) (d) MPE (300 epochs)

Figure 4: **MPE objective learns decision boundary using only generalizable features:** We simulate high-dimensional classification by projecting a simple 2-d dataset into a 625-dimensional space. *(a)* Standard ERM training overfits to this dataset, achieving perfect training accuracy by picking up on spurious features. Plotting a 2D projection of the decision boundary, we see that it poorly separates the data. *(b, c, d)* Visualizing our method (MPE) at different snapshots throughout training, we see that it converges to the true decision boundary.

Table 1: **Robustness to adversarial shifts in distribution compared against in-distribution performance:** Test accuracies of adversarial methods (adversarial training, ADA and ME-ADA), ERM training and MPE objective to Fast Gradient Sign (FGSM) attacks with $\|\delta\|_1 = 0.5$ (see equation 2).

Method	**CIFAR-100-2k**		**CIFAR-100**		**CIFAR-10**	
	Clean	FGSM Attack	Clean	FGSM Attack	Clean	FGSM Attack
ERM Training	28.5 ± 0.03	24.7 ± 0.06	76.4 ± 0.03	67.4 ± 0.06	95.4 ± 0.04	88.3 ± 0.04
Adv. Training	27.4 ± 0.05	27.2 ± 0.03	73.2 ± 0.03	73.1 ± 0.04	93.1 ± 0.04	92.9 ± 0.04
ADA	28.7 ± 0.05	27.0 ± 0.03	76.5 ± 0.06	72.7 ± 0.04	95.2 ± 0.05	88.1 ± 0.05
ME-ADA	29.4 ± 0.04	27.6 ± 0.03	77.1 ± 0.05	$\mathbf{74.8 \pm 0.03}$	$\mathbf{96.1 \pm 0.03}$	$\mathbf{93.1 \pm 0.04}$
Ours	$\mathbf{30.4 \pm 0.03}$	$\mathbf{28.1 \pm 0.04}$	$\mathbf{77.3 \pm 0.04}$	74.5 ± 0.05	96.1 ± 0.04	93.0 ± 0.04

identifying the true generalizable features, this model has overfit to the noisy dimensions, which are perpendicular to the span of the true features. Training this model for more epochs leads to additional overfitting, further decreasing the test accuracy.

We next apply our method (MPE) to this same dataset. In addition to the standard empirical cross entropy loss, our method also maximizes the predictive entropy of the model on adversarial examples. Intuitively, we expect that these adversarial examples will be along the directions of the spurious features. Thus, in training the model to be less confident on adversarial features, we aim to have the model learn to ignore these spurious features. Applying MPE to this dataset, we achieve a much larger test accuracy of 94.9%. When we visualize the decision boundary in Figure 4d, we observe that it correctly separates the data. While SGD is implicitly biased towards learning simple (e.g., linear) decision boundaries (Kalimeris et al., 2019), our results show that MPE partially counters this bias, forcing the model to learn a non-linear decision boundary along the true features and ignoring the noisy dimensions.

B Robustness to distribution shift

Typically, methods for handling distribution shift are different from methods for improving test accuracy. Prior work has found that increasing robustness to distribution shifts tends to be at odds with increasing test accuracy: methods that are more robust often achieve lower test accuracy, and methods that achieve higher test accuracy tend to be less robust (Raghunathan et al., 2019; Zhang et al., 2019; Tsipras et al., 2019). While the main aim of our experiments is to show that MPE improves test accuracy, our next set of experiments investigate whether MPE is any more robust to distribution shift than baseline methods.

For these experiments on distribution shift, we use the exact same hyperparameters as in the previous experiments. Better results are likely achievable by tuning the method for performance on these robustness benchmarks. By reporting results using the exact same hyperparameters, we demonstrate that the same method might both achieve high in-distribution performance and out-of-distribution performance.

Robustness to adversarial attacks. We first look at robustness to adversarial attacks, using FGSM (Goodfellow et al., 2014) as the attack method. The conventional approach to fending off adversarial attacks is adversarial training, wherein the training objective exactly matches the testing objective. Thus, adversarial training represents the "gold standard" for performance in this task. We compare the adversarial robustness of MPE, adversarial training, ADA, and ME-ADA in Table 1. Not only does our method achieve higher (clean) test accuracy than adversarial training on all datasets, but surprisingly it also achieves higher robust test accuracy on the harder CIFAR-100-2k benchmark where the clean test accuracy of MPE is $+3\%$ greater than adversarial training, and robust test accuracy is $+0.5\%$ better than ME-ADA.

Both ADA and ME-ADA perform some form of adversarial training, so it is not surprising that they outperform MPE on this task. We suspect that these methods outperform adversarial training because they are trained using the multi-step projected gradient descent (PGD) (Madry et al., 2017), rather than the one-step FGSM Goodfellow et al. (2014) and also have a higher clean test accuracy. While our aim is *not* to propose a state-of-the-art method for withstanding adversarial attacks, these

preliminary results suggest that MPE may be somewhat robust to adversarial attacks, but does so without inheriting the poor (clean) test accuracy of standard adversarial training.

Improved performance on shifted test distributions. Our final set of experiments probe robustness to more systematic distribution shifts using the corrupted CIFAR-10 dataset (Hendrycks & Dietterich, 2019). These shifts go beyond the small perturbations introduced by adversarial examples, and are a more faithful reflection of the sorts of perturbations a machine learning model might face "in the wild."

We compare MPE to standard ERM and ME-ADA on this benchmark; all methods are trained on the un-corrupted CIFAR-10 dataset, but evaluated on different types of corruptions. We report results in Figure 5. Both MPE and ME-ADA consistently outperform ERM. On certain corruptions (e.g., Gaussian noise, glass blur), MPE and ME-ADA achieve test accuracies that are around $+25\%$ larger than the ERM baseline. We do not notice any systematic difference in the results of MPE versus ME-ADA, but note that ME-ADA requires $2\times$ more compute than MPE because its adversarial examples require multiple gradient steps to compute.

While the main aim of our experiments has been to show that MPE achieves higher test accuracy, its good performance on robustness benchmarks suggest that it may be a simpler yet appealing choice for practitioners.

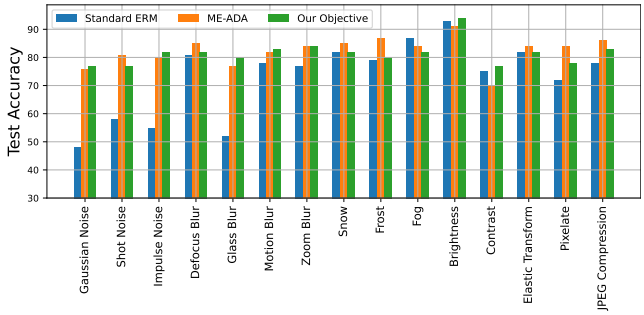


Figure 5: **Robustness to natural shifts in distribution:** Plots comparing the performance of standard ERM training and ME-ADA against our method trained on CIFAR-10 benchmark and tested on various distribution shifts in the corrupted CIFAR-10 dataset.

C SUMMARY OF RELATED REGULARIZATION OBJECTIVES

In the table below, we compare the objectives of various regularization methods proposed in prior works, that involve the adversarial examples and or the model's predictive distribution.

name	objective
cross entropy	$\min_\theta -\sum_{x,y} \log p_\theta(y \mid x)$
label smoothing (Müller et al., 2019) $[\alpha]$	$\min_\theta -\sum_{x,y}((1-\alpha)\log p_\theta(y \mid x) + \sum_{y' \neq y} \frac{\alpha}{L-1} \log p_\theta(y' \mid x))$
Adv. training (Madry et al., 2017) $[\alpha]$	$\min_\theta -\sum_{x,y} \max_{\delta:\|\delta\|_2 \leq \alpha} \log p_\theta(y \mid x+\delta)$
ME-ADA (Zhao et al., 2020) $[\alpha, \beta]$	$\min_\theta - \sum_{(x,y)\in\mathcal{S}\cup\mathcal{S}'} \log p_\theta(y \mid x)$ where, for a distance metric C_θ: $\mathcal{S}' \triangleq \left\{(\tilde{x}, y) : \tilde{x} := \sup_{x_0 \in \mathcal{X}} \arg\max - \log p_\theta(x_0 \mid y) + \alpha \mathcal{H}_\theta(x_0) - \beta C_\theta((x_0, y), (x, y)) \right\}$ $_{(x,y)\in\mathcal{S}}$
MPE (ours) $[\alpha, \beta]$	$\min_\theta -\sum_{x,y} \log p_\theta(y \mid x) - \alpha \mathcal{H}_\theta(x - \beta \nabla_x \log p_\theta(y \mid x))$

Table 2: **Regularization objectives**: We summarize regularization objectives from prior work that employ adversarial examples or directly regularize model's predictions $p_\theta(y \mid x)$.