CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 158 **July 2, 2025**

Multiplicative Regularization Generalizes Better Than Additive Regularization

Rafael Dubach[1,2,4], **Mohamed S. Abdallah**[3], **and Tomaso Poggio**[1,2]

1: Massachusetts Institute of Technology (MIT)
2: Center for Brains, Minds, and Machines
3: Harvard University
4: University of Zurich

Abstract

We investigate the effectiveness of multiplicative versus additive (L2) regularization in deep neural networks, focusing on convolutional neural networks for classification. While additive methods constrain the sum of squared weights, multiplicative regularization directly penalizes the product of layerwise Frobenius norms, a quantity theoretically linked to tighter Rademacher-based generalization bounds. Through experiments on binary classification tasks in a controlled setup, we observe that multiplicative regularization consistently yields wider margin distributions, stronger rank suppression in deeper layers, and improved robustness to label noise. Under 20% label corruption, multiplicative regularization preserves margins that are 5.2% higher and achieves 3.59% higher accuracy compared to additive regularization in our main network architecture. Furthermore, multiplicative regularization achieves a 3.53% boost in test performance for multiclass classification compared to additive regularization. Our analysis of training dynamics shows that directly constraining the global product of norms leads to flatter loss landscapes that correlate with greater resilience to overfitting. These findings highlight the practical benefits of multiplicative penalties for improving generalization and stability in deep models.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Multiplicative Regularization Generalizes Better Than Additive Regularization

Rafael Dubach
Massachusetts Institute of Technology
raduba@mit.edu

Mohamed S. Abdallah
Harvard University
mabdallah@g.harvard.edu

Tomaso Poggio
Massachusetts Institute of Technology
tp@mit.edu

Abstract

We investigate the effectiveness of multiplicative versus additive (L2) regularization in deep neural networks, focusing on convolutional neural networks for classification. While additive methods constrain the sum of squared weights, multiplicative regularization directly penalizes the product of layerwise Frobenius norms, a quantity theoretically linked to tighter Rademacher-based generalization bounds. Through experiments on binary classification tasks in a controlled setup, we observe that multiplicative regularization consistently yields wider margin distributions, stronger rank suppression in deeper layers, and improved robustness to label noise. Under 20% label corruption, multiplicative regularization preserves margins that are 5.2% higher and achieves 3.59% higher accuracy compared to additive regularization in our main network architecture. Furthermore, multiplicative regularization achieves a 3.53% boost in test performance for multiclass classification compared to additive regularization. Our analysis of training dynamics shows that directly constraining the global product of norms leads to flatter loss landscapes that correlate with greater resilience to overfitting. These findings highlight the practical benefits of multiplicative penalties for improving generalization and stability in deep models.

1 Introduction

Regularization is central to controlling overfitting and improving generalization in deep learning, as the capacity of neural networks are inherently tied to their generalization abilities [1, 2, 3]. This fact induced a lot of interest in defining generalization bounds that are reflective of the capacity of neural networks, and their ability to generalize beyond training data [1, 4, 5, 6, 7, 8].

Among standard techniques, additive approaches such as L2 regularization remain dominant, yet there is increasing interest in methods that constrain global weight configurations more directly. Recent theoretical work suggests that products of layer norms play a key role in capacity control, motivated by norm-based generalization bounds linking tighter Rademacher complexity guarantees to the product of weight norms across layers [9, 8, 10]. Moreover, Bottman et al. (2023) give geometric evidence that multiplicative penalties produce more isolated, well-conditioned minima in the loss landscape, which helps explain their superior robustness [11]. These bounds suggest that a direct penalty on the product of the layer norms should offer distinct advantages over conventional per-layer or summation-based penalties, which are the typical capacity control methods employed in practice to improve generalization in neural networks [12]. Our work compares these additive and multiplicative approaches in a systematically controlled setting to evaluate how each method affects training dynamics, margin behavior, and robustness to label noise.

We focus on the conceptual difference that L2 regularization penalizes the sum of squared parameters. In contrast, multiplicative penalties establish a coordinated constraint system across layers by penalizing the product of their Frobenius norms [13, 14]. We aim to determine whether explicitly targeting the product of norms elicits stronger generalization, better noise robustness, and more stable training. Through extensive experiments on binary classification tasks using convolutional neural networks, we find that multiplicative regularization promotes wider margins, sharper rank suppression, and increased resilience to label corruption. Furthermore, we carry out experiments to confirm that these observations might be generalizable to multiclass classification, observing results consistent with the binary classification case. Our results align with the theoretical motivation that bounding the product of layer norms provides a more principled approach to capacity control [15, 16].

2 Background

Researchers have long studied how large models generalize effectively despite being overparameterized [3]. L2 regularization has proven valuable for constraining weight magnitudes and improving performance, but its effect on the product of norms remains unclear [14, 17]. Early work on learning theory and VC dimension laid the groundwork for tighter norm-based generalization bounds. In the case of multilayer networks, the Rademacher complexity bounds depend on the product ρ of layerwise spectral or Frobenius norms [8, 10] and can be therefore controlled by constraining ρ. Although earlier bounds [8] displayed a dependence on the product of the weight matrices' norms, as well as an exponential dependence on network depth, more recent bounds [5] reduced this to a polynomial dependence on depth, scaling polynomially with depth as

$$O\Big(\frac{B\sqrt{L}\prod_{i=1}^{L}\|W_i\|_F^2}{\sqrt{m}}\Big). \tag{1}$$

where L is the total number of layers, $\|W_i\|_F^2$ denotes the squared Frobenius norm of the weight matrix W_i for the i-th layer, B bounds the input norm, and m is the sample size.

From the perspective of these bounds on the Rademacher complexity, multiplicative regularization, which directly targets the product of norms, provides a more theoretically-grounded capacity-control mechanism than methods constraining only individual weights or the sum of squared weights [10].

Efforts to compare different penalties in practice have generally focused on variations in weight decay and related additive techniques [13, 14]. Ensemble- and dropout-based methods inject noise to reduce reliance on any single set of parameters but do not systematically target the product of norms [18]. Recent developments introduce multiplicative norm constraints or alternative reparameterizations that keep layer products bounded [19, 20]. A recent theoretical study focused on the effect of different regularizers on the loss landscape, showing that multiplicative regularization leads to non-Morse loss landscapes with degenerate minima, while conjecturing that additive regularization leads to a Morse landscape with isolated minima for general nonlinear networks [11]. However, few empirical studies have directly assessed whether a multiplicative penalty in norms outperforms standard additive L2 penalties under various training conditions. Our work aims to fill this gap by offering a systematic analysis in a controlled classification setting.

3 Theory

In this section, we examine the relationship between additive regularization and multiplicative regularization from a theoretical perspective. Later in the Results section, we link these theoretical findings to the observed results.

The goal here is to relate the two quantities $\prod_{i=1}^{L}\|W_i\|_F^2$ and $\sum_{i=1}^{L}\|W_i\|_F^2$. Since $\|W_i\|_F^2 \geq 0$, we can directly apply the inequality of arithmetic and geometric means (AM-GM inequality):

$$\frac{1}{L}\sum_{i=1}^{L}\|W_i\|_F^2 \;\geq\; \left(\prod_{i=1}^{L}\|W_i\|_F^2\right)^{\frac{1}{L}} \tag{2}$$

We note that the equality holds if and only if $\|W_1\|_F^2 = \|W_2\|_F^2 = \cdots = \|W_L\|_F^2$. Raising both sides to the power of L:

$$\left(\frac{1}{L}\sum_{i=1}^{L}\|W_i\|_F^2\right)^L \;\geq\; \prod_{i=1}^{L}\|W_i\|_F^2 \tag{3}$$

This result shows that in the general case, the multiplicative term is bounded by the additive term with a multiplicative constant and a positive power, implying that additive regularization could indirectly lead to a small $\prod_{i=1}^{L}\|W_i\|_F^2$. However, the result does not imply that multiplicative regularization should lead to a small $\frac{1}{L}\sum_{i=1}^{L}\|W_i\|_F^2$.

Another observation that follows from the preceding theoretical discussion is that the minima of additive regularization and multiplicative regularization generally do not correspond. This discussion means that, practically, if we want to train a network to find a small $\prod_{i=1}^{L}\|W_i\|_F^2$ solution to obtain tight bounds for the generalization gap, a working strategy would be to use additive regularization. Still, a much more effective strategy is to use multiplicative regularization instead. In passing, we note that a neat way of relating additive regularization and multiplicative regularization would be to observe that:

$$\log\Big(\prod_{i=1}^{L}\|W_i\|_F^2\Big) \;=\; \sum_{i=1}^{L}\log\big(\|W_i\|_F^2\big) \tag{4}$$

Since $\log(x)$ is a monotonic function, minimizing the left-hand side is equivalent to minimizing the right-hand side. Therefore, multiplicative regularization can be thought of as a form of additive regularization, but for the sum of the logs of the squared norms instead of a sum of the squared norms themselves.

4 Methodology

We examine two regularization schemes designed to enhance model generalization by mitigating overfitting. Additive regularization, perhaps the most widely recognized strategy, adds a term proportional to the sum of squared weights directly to the loss. Multiplicative regularization instead introduces an alternative approach by penalizing the product of Frobenius norms across layers. The additive penalty with coefficient λ is

$$\mathcal{L}_{\text{add}} \;=\; \mathcal{L}_{\text{original}} + \lambda\sum_{i=1}^{L}\|W_i\|_F^2 \tag{5}$$

while in this work, we introduce the multiplicative penalty as

$$\mathcal{L}_{\text{mult}} \;=\; \mathcal{L}_{\text{original}} + \lambda\prod_{i=1}^{L}\|W_i\|_F^2 \tag{6}$$

Motivated by theoretical bounds suggesting that the product of norms can more directly control model capacity, we systematically investigate how each penalty shapes training dynamics, convergence properties, and final model performance. While additive regularization applies independent penalties to each layer's weights, multiplicative regularization creates interdependence between layer norms, as reducing the product requires coordinated optimization across the entire network. This directly targets the product term appearing in Rademacher complexity bounds, potentially leading to better generalization in overparameterized networks.

Our experiments primarily focus on a specific convolutional neural network architecture. This architecture features four convolutional layers with channel counts increasing from 16 to 32 to 64 to 128, using ReLU activations after each layer. We use average pooling with a kernel size of two and a stride of two to downsample the feature maps. Two fully connected layers follow: the first reduces

features to 256 dimensions, and the final output layer provides the classification output. Bias terms are turned off in the convolutional layers and activated in the last two fully connected layers. The total number of trainable parameters is 228,785.

We train on a binary variant of the CIFAR-10 dataset [21] by mapping two chosen classes ("cats" with class index three and "deer" with class index 4) to -1 and 1, respectively. The training is framed as a regression problem optimizing a mean squared error (MSE) objective, and prediction is performed via $\mathrm{sign}(\mathrm{output})$. The CIFAR-10 dataset is normalized using channel-wise mean (0.4914, 0.4822, 0.4465) and standard deviation (0.2470, 0.2435, 0.2616). For computational efficiency while preserving analysis integrity, we limit the training set to 10,000 images and the test set to 2,000.

Weight initialization is handled with Kaiming (He) initialization [22], which is designed for deep networks using ReLU activation functions. This method initializes weights from a normal distribution with a mean of zero and a variance of $2/(\text{fan-in})$, which mitigates issues of vanishing or exploding gradients. Additionally, we apply scaling factors that are uniform across layers $s \in \{0.2, 0.4, 0.6, 0.8\}$ to the initialized weights, enforcing a lower-norm regime at the start of training and allowing finer control over early training dynamics.

Our training procedure employs Stochastic Gradient Descent (SGD) with a batch size of 16, running for up to 2000 epochs with feature normalization. We implement early stopping if the test accuracy fails to improve for 200 consecutive epochs. For both additive and multiplicative regularization, we test λ values spanning 5×10^{-1} to 5×10^{-15}. All experiments are repeated with five random seeds (1, 2, 3, 4, 5), discarding runs that remain at random guessing levels (i.e., models that have failed to train).

We track several evaluation metrics to comprehensively assess model performance. Beyond standard classification accuracy and loss on held-out data, we compute margins for correctly classified points and analyze their distribution. For a given sample $\mathbf{x}_i$ with label $y_i \in \{-1, 1\}$, the margin is

$$\mathrm{margin}_i = y_i \cdot f(\mathbf{x}_i) \tag{7}$$

Where $f(\mathbf{x}_i)$ is the network's real-valued output before taking its sign. Positive margins indicate correct classification, and larger values signify greater classification confidence. To ensure a fair comparison across regularization methods, we re-normalize the trained network weights to have a unit norm before computing margins.

We examine weight matrix ranks to estimate the effective dimensionality of each layer. This is calculated using a singular value decomposition (SVD) of each layer's weight matrix. We reshape the four-dimensional kernel into a two-dimensional matrix for convolutional layers, while fully connected layers are used directly. After computing singular values and normalizing by the largest one, we count how many exceed 1% of the maximum to determine the rank. This procedure yields a layer-specific measure of model capacity, capturing how many effective dimensions each layer contributes to the network's overall capacity.

We also implement label noise experiments, where a fraction of training labels is randomly flipped to assess model robustness under noisy conditions. This provides insights into how different regularization strategies affect a model's generalization ability despite corrupted training signals. We report aggregated averages or best individual results over multiple random seeds for each regularization type and hyperparameter configuration, ensuring reliable conclusions about the comparative effectiveness of additive versus multiplicative regularization.

For multi-class experiments, we train on the full CIFAR-10 dataset with all ten classes represented using one-hot encoded labels. Similar to the binary case, training is framed as a regression problem using an MSE objective, with classification predictions determined via $\arg\max$ over the output logits. Consistent with our binary classification experiments, we normalized the inputs using the CIFAR-10 channel-wise mean and standard deviation, and performed standard train-test splitting.

For multi-class classification, we use a convolutional neural network (CNN) with four convolutional blocks, each consisting of two convolutional layers with 3×3 kernels, ReLU activations, and 2×2 average pooling. The number of channels increases across blocks as follows: 64, 128, 256, and 512. After the final pooling layer, the $2 \times 2 \times 512$ feature map is flattened and passed through a fully connected layer with 512 hidden units and ReLU activation, followed by a final linear layer with 10 output units—corresponding to the 10 classes in CIFAR-10, with a total of $5,737,152$ parameters. The network does not include bias, and all weights are initialized using Kaiming (He) initialization.

We apply both additive and multiplicative norm-based regularization during training, varying the regularization strength λ across a scale from 5×10^{-1} to 5×10^{-20}. For the remaining hyperparameters and early-stopping methodology, we remain consistent with our binary classification experiments. All experiments were run on a GPU cluster equipped with multiple NVIDIA GPUs, though each training run used only a single GPU. This setup enabled efficient training across hyperparameter configurations and random seeds.

5 Results

Our experiments compare additive and multiplicative regularization across a sweep of coefficient λ values at the fixed initialization scale 0.2. For each regularization strategy and each misclassification rate, all runs that share the same λ are first averaged over five random seeds. The λ that maximizes this averaged test accuracy is then selected, and its averaged metrics are used as the representative performance for that group. Using this procedure, both strategies still reach near-perfect training accuracy, indicating minimal underfitting, but multiplicative regularization consistently yields lower test loss and higher test accuracy. The resulting gap in average test accuracy reaches up to 9.4 percentage points, with multiplicative regularization often achieving gains through broader margin distributions (see Fig. 1). The mean margin on the test set is systematically larger for models trained under multiplicative penalties, indicating that directly constraining the global norm product promotes more decisive classifications.

Our robustness analysis underscores the advantage of multiplicative regularization under label noise. At baseline (0 % noise), multiplicative regularization achieves 89.50 % test accuracy compared to 87.87 % for additive and 87.80 % for no regularization. When 10 % of labels are flipped, the gap widens: multiplicative regularization maintains 86.68 % accuracy, while additive drops to 81.13 % and unregularized models to 80.20 %. At 20 % label corruption, multiplicative regularization (83.10 %) surpasses additive regularization (73.69 %) by 9.41 percentage points and no regularization (73.08 %) by 10.02 points. These accuracy trends are mirrored in test loss values, with multiplicative regularization showing the smallest increase in loss as noise levels rise.

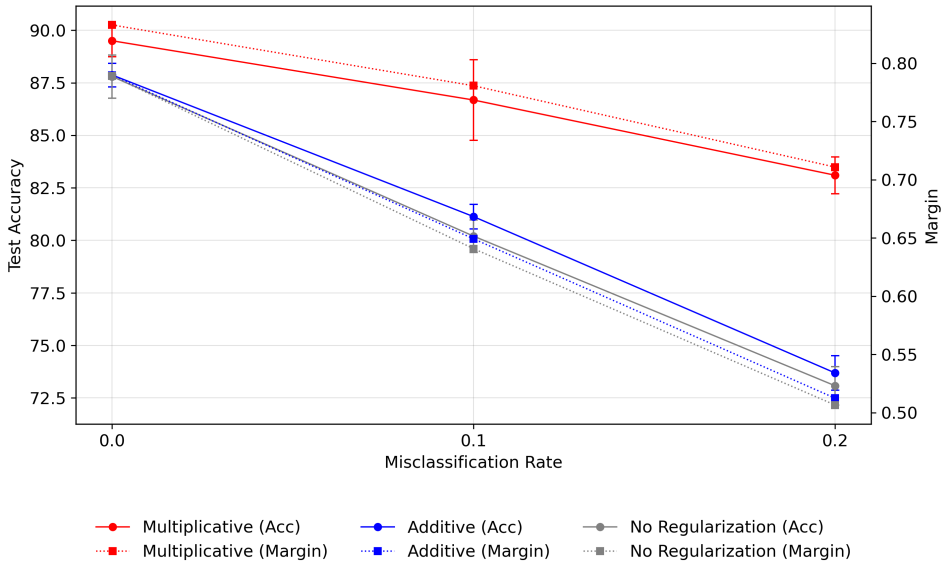

Figure 1: Test accuracy and margin under multiplicative, additive, and no regularization across misclassification rates (0.0, 0.1, 0.2). Multiplicative regularization exhibits the smallest degradation in both metrics and maintains higher margins across all noise levels, correlating with superior accuracy. Error bars represent standard deviation for accuracy measurements only.

This robustness advantage correlates strongly with margin preservation. Under clean conditions, multiplicative regularization yields the highest mean margin (0.8329) compared to additive (0.7896) and no regularization (0.7890). With 20 % label corruption, multiplicative regularization maintains

a margin of 0.7109 (a 14.6 % reduction from baseline) compared to 0.5124 for additive (35.1 % reduction) and 0.5066 for no regularization (35.8 % reduction). This pattern suggests that directly constraining the product of norms leads to decision boundaries that degrade less severely as label noise increases.

In terms of training dynamics, multiplicative regularization and additive regularization show distinct behaviors and lead to different minima in the unregularized loss landscape. Fig. 2 shows typical behavior of the training dynamics of the unregularized, multiplicatively-regularized, and additively-regularized models in our binary classification experiments. From Fig. 2, we can observe that all three models have trained for a few hundred epochs to (mostly) stable loss minima for the loss landscapes defined by the corresponding loss equations. The result seen in Fig. 2(a)is consistent with our earlier discussion of the better performance of multiplicative regularization compared to additive regularization and the unregularized cases. Furthermore, we see from Fig. 2(b) that multiplicative regularization has found a minimum of more than an order of magnitude smaller $\prod_{i=1}^{L} \|W_i\|_F$ compared to additive regularization, whose $\prod_{i=1}^{L} \|W_i\|_F$ is in turn one order of magnitude smaller than the unregularized models. Interestingly, for both the unregularized and the additively-regularized models, we can see from Fig. 2 that the curves for $\prod_{i=1}^{L} \|W_i\|_F$ and $\sum_{i=1}^{L} \|W_i\|_F^2$ follow very similar trajectories throughout training. However, the multiplicative regularization trajectories for $\prod_{i=1}^{L} \|W_i\|_F$ and $\sum_{i=1}^{L} \|W_i\|_F^2$ do not correspond, with the model reaching a stable low value of $\prod_{i=1}^{L} \|W_i\|_F$ while $\sum_{i=1}^{L} \|W_i\|_F^2$ remains in a transient increasing regime.

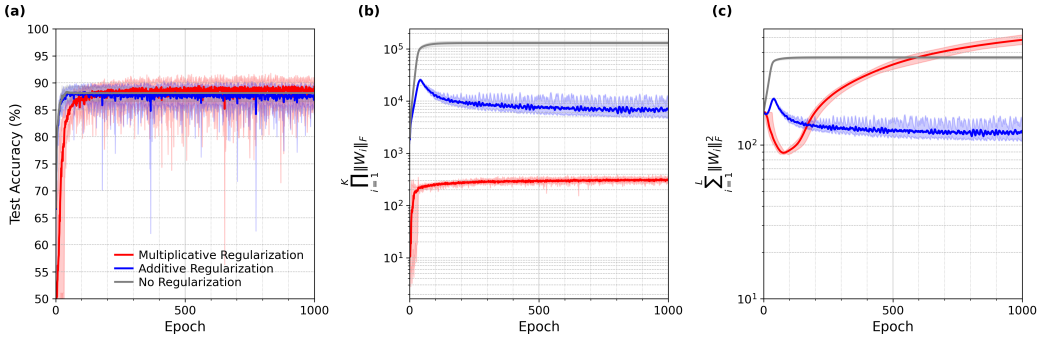

Figure 2: Training dynamics for our binary classification experiments, corresponding to the best performing values of λ: (a) test set accuracy, (b) the product of the layers' Frobenius norm $\prod_{i=1}^{L} \|W_i\|_F$, and (c) the sum of the squares of the layer's Frobenius norm $\sum_{i=1}^{L} \|W_i\|_F^2$. The solid lines show the average over the seeds, while the shaded areas show the range across the seeds.

Further evidence emerges from analyzing parameter structure at the point of early-stopped convergence. While early convolutional layers maintain similar ranks across both regularization methods, later layers show notable differences.

As shown in Table 1, the contrast is most striking in the first fully connected layer, with ranks of 2 and 256, respectively. This substantial rank reduction under multiplicative regularization suggests the formation of more coordinated, low-dimensional representations that maintain high classification accuracy. These observations support the theoretical link between multiplicative norm constraints and tighter generalization bounds. Our data also shows a stronger correlation between final test loss and the product of layer norms than between final test loss and the sum of squared weights, further underscoring the theoretical advantages of multiplicative regularization.

Table 1: By-layer comparison of rank values and Frobenius norms under multiplicative, additive, and unregularized settings for the best-performing model configuration.

Layer	**Rank**			**Frobenius Norm** $\|W_i\|_F$		
	Multiplicative	Additive	Unreg.	Multiplicative	Additive	Unreg.
conv1	16	16	16	7.39	4.55	5.71
conv2	32	32	32	16.84	5.14	6.78
conv3	64	64	64	10.74	6.39	8.47
conv4	47	128	128	1.29	6.18	9.04
fc1	2	256	256	0.46	5.19	10.25
fc2	1	1	1	0.38	3.57	3.10
$\sum_{i=1}^{L} \|W_i\|_F^2$				456	166	347
$\prod_{i=1}^{L} \|W_i\|_F$				3.01×10^2	1.71×10^4	9.42×10^4

Other observations can be drawn from Table 1 based on the changes induced in the Frobenius norm of the weight matrices by the regularization methods. The results for $\sum_{i=1}^{L} \|W_i\|_F^2$ and $\prod_{i=1}^{L} \|W_i\|_F$ are consistent with those shown in Fig. 2, with multiplicative regularization achieving almost two orders of magnitude smaller $\prod_{i=1}^{L} \|W_i\|_F$ compared to additive regularization, which in turn achieves one order of magnitude smaller compared to the unregularized model. This comes at at the expense of $\sum_{i=1}^{L} \|W_i\|_F^2$, for which the model trained with multiplicative regularization achieves a value that is even larger than the unregularized case, while additive regularization achieves a value that is about half of the unregularized case. This echoes the results in Fig. 2, with each type of regularization performing better than the other in reducing the specific quantity it was designed to penalize. As was explained in Section 3, additive regularization indirectly leads to a smaller $\prod_{i=1}^{L} \|W_i\|_F$ than the unregularized case, while not matching the performance of multiplicative regularization in that regard. On the other hand, multiplicative regularization does not yield a smaller $\sum_{i=1}^{L} \|W_i\|_F^2$ than the unregularized case.

A related, but distinct observation can also be drawn from the individual values of $\|W_i\|_F$. While additive regularization leads to values of $\|W_i\|_F$ on the same order as each other, multiplicative regularization can induce extreme sparsity in specific layers that would in the end lead to a much smaller $\prod_{i=1}^{L} \|W_i\|_F$. Again, this can be understood by the discussion in Section 3. Whereas additive regularization has to minimize the sum term by minimizing each individual contribution, multiplicative regularization has the flexibility of reducing the product term by minimizing specific $\|W_i\|_F$ significantly. An equivalent way of seeing this is that multiplicative regularization can benefit more than additive regularization (in terms of reducing the objective function) by inducing specific $\|W_i\|_F$ to be of a smaller order of magnitude compared to the other $\|W_i\|_F$.

To gain geometric insight into how the two regularizers guide optimization, we visualize the local loss surface around each minimum following the methodology proposed in Ref. [23]. Using the same CNN (initialization scale $g = 0.2$, fixed seed), we took the best additive regularization run ($\lambda = 5 \times 10^{-9}$) and the best multiplicative regularization run ($\lambda = 5 \times 10^{-7}$) and plotted their loss landscapes in log scale (Fig. 3).

This geometric distinction aligns with our quantitative results: the flatter basin induced by multiplicative regularization is consistent with its superior test accuracy. Generally, flatter minima characterized by lower curvature are associated with broader, more stable regions in the loss landscape and tend to yield better generalization performance [24, 25].

Going beyond the binary classification experiments, we have tested the generality of our observations by running multi-class classification experiments on the full CIFAR-10 dataset with uncorrupted labels. The average results across 10 seeds for the best performing regularization λ are shown in Table 2. Similar to the binary classification results, we observe that multiplicative regularization achieves better test accuracy, with a smaller generalization gap, as compared to the additive regularization and the unregularized cases. In fact, in our search for the optimal λ, we could not find a value that would lead to additive regularization outperforming the unregularized case for test accuracy, except marginally. The optimal additive regularization λ we have found seems to be so small that it only reduces $\sum_{i=1}^{L} \|W_i\|_F$ in a minor way. Customarily, such an observation could lead one to

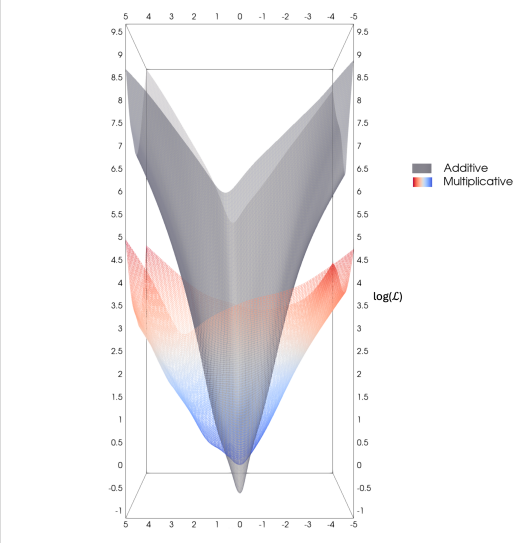

Figure 3: Loss landscape near the found minima for additive (white) and multiplicative (colormap) regularization along two filter-wise normalized random directions. Multiplicative regularization produces a broader, flatter basin, whereas additive regularization yields a sharper, steeper well.

believe that the model's size is in the underfitting regime. However, the results of the multiplicative regularization lead to a boost of more than 3% in test accuracy compared to additive regularization and the unregularized cases. This observation suggests that there are cases where a deep learning model could benefit from using multiplicative regularization, even when additive regularization offers little benefit, although this might be highly dependent on model initialization and other model hyperparameters and priors. Also consistent with our binary classification results, multiplicative regularization leads to much smaller norm product, signaling effective capacity control, and smaller generalization gap.

Table 2: Comparison of weight norms and classification accuracies for multi-class classification under different regularization schemes. The results correspond to the average over 10 seeds, and the best performing λ for the additive and multiplicative regularization cases.

Regularization	$\prod_{i=1}^{L} \|W_i\|_F$	$\sum_{i=1}^{L} \|W_i\|_F^2$	Train Acc.	Test Acc.
Multiplicative	3.73×10^7	679	0.891	0.687
Additive	1.10×10^9	875	0.925	0.651
Unregularized	1.67×10^9	960	0.937	0.650

6 Discussion

Our findings confirm that multiplicative regularization can offer practical benefits over additive penalties in controlling capacity and improving resilience to noise. Prior theoretical work argues that bounding the product of norms is intimately tied to generalization bounds derived from Rademacher's complexity. Our analysis of multiplicative penalties suggest that they would lead to distinct training dynamics and converge to minima that are in general different from those obtained by additive regularization. Our empirical tests show that multiplicative constraints can be realized in convolutional networks trained on image data, guiding optimization toward flatter minima with simplified high-level representations. The margin and rank patterns we observe reinforce this view: models with multiplicative penalties exhibit consistently higher margins and more pronounced rank suppression, particularly in later-layer transformations that prove influential for final decision formation. The training dynamics show distinct behaviors for multiplicative vs. additive regularizers, with multiplicative penalties leading to models with much smaller norm products while allowing the sum term to be flexible. Moreover, multiplicative regularization leads to norm distributions across layers reflecting

global coupling and does not show any preference towards a uniform norm distribution across layers – unlike additive regularization.

Although the average difference in clean-set accuracy between methods can be moderate, our label noise experiments highlight the practical utility of multiplicative regularization for tasks subject to noise. When training data are imperfect or partially mislabeled, constraining norms via their product can yield noticeably better performance than standard L2. Interestingly, the initialization scale has a significant effect since smaller initial weights allow multiplicative constraints to take effect early, preventing excessive norm growth. With larger initial scales, the distinction between methods narrows.

Extending these observations, we note that multiplicative regularization not only stabilizes training dynamics but also imposes a mathematically coupled constraint system across layers. By penalizing the product of norms rather than their sum, multiplicative regularization creates interdependence among all layers during optimization, as a reduction in any single layer's norm directly affects the overall penalty term in a way that allows greater flexibility for the individual norms.

From a representational-geometry viewpoint, the low effective rank in deeper layers implies that the network is mapping inputs onto lower-dimensional subspaces. This reduces overall model capacity and thus helps prevent overfitting in overparameterized networks.

Furthermore, many of our observations apply to both binary and multi-class classification cases, indicating generalization across different paradigms. These findings collectively demonstrate that controlling the product of norms provides a theoretically grounded alternative to conventional additive penalties.

Our noise-robustness experiments further suggest an avenue for future research in domains prone to label uncertainties, such as semi-supervised learning, active learning, or medical image analysis. Given the strong correlation between margin preservation and improved accuracy under label noise, the observed advantages of multiplicative regularization suggest that product-based penalties may be especially beneficial in practice when dealing with noisy or less reliable annotations. Beyond classification tasks, examining how these constraints interact with other architectures (e.g., Transformer models) or generative frameworks (e.g., autoencoders) could illuminate whether they generalize to other deep learning paradigms.

7 Conclusion

We present a systematic comparison of multiplicative and additive regularization in neural networks, showing that directly penalizing the product of layerwise norms tends to yield higher test accuracy, improved margin behavior, better noise resilience, and more effective rank suppression. These findings align with theoretical arguments that multiplicative constraints provide a more principled mechanism for reducing model capacity in overparameterized networks. The gap between multiplicative and additive approaches is especially evident under label corruption, suggesting that multiplicative regularization becomes even more advantageous for noisy datasets.

As modern architectures grow in depth and width, employing global measures such as the product of layer norms can be a powerful countermeasure to overfitting, complementing strategies like dropout or layer-specific penalties. Future directions could extend these insights to deeper tasks, explore hybrid strategies combining multiplicative penalties with other structural priors, or investigate whether advanced optimizers and adaptive schedules amplify the benefits of multiplicative regularization. The empirical evidence presented here indicates that multiplicative regularization is a promising strategy for controlling capacity and enhancing robustness across diverse deep learning challenges.

References

[1] Tomer Galanti, Liane Galanti, and Ido Ben-Shaul. Comparative generalization bounds for deep neural networks. *Transactions on Machine Learning Research*, 2023.

[2] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[3] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

[4] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[5] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks, 2019.

[6] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

[7] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[8] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks, 2015.

[9] Tomer Galanti, Mengjia Xu, Liane Galanti, and Tomaso Poggio. Norm-based generalization bounds for sparse neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pages 42482–42501. Curran Associates, Inc., 2023.

[10] Mengjia Xu, Akshay Rangamani, Qianli Liao, Tomer Galanti, and Tomaso Poggio. Dynamics in deep classifiers trained with the square loss: Normalization, low rank, neural collapse, and generalization bounds. *Research*, 6:0024, 2023.

[11] Nathaniel Bottman, Y. Cooper, and Antonio Lerario. How regularization affects the geometry of loss functions. *arXiv preprint arXiv:2307.15744*, 2023.

[12] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[13] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[14] Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, NIPS'91, pages 950–957. Morgan Kaufmann Publishers Inc., 1991.

[15] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers, 2020.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, 2014.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[20] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *CoRR*, abs/1602.07868, 2016.

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Canadian Institute For Advanced Research, 2009.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[23] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2018.

[24] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations*, 2017.

[25] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning – generalization gap and sharp minima. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1742–1751, 2017.