Forget Sharpness: Perturbed Forgetting of Model Biases Within SAM Dynamics

Ankit Vani [1 2 *] **Frederick Tung** [2] **Gabriel L. Oliveira** [2] **Hossein Sharifi-Noghabi** [2]

Abstract

Despite attaining high empirical generalization, the sharpness of models trained with sharpness-aware minimization (SAM) do not always correlate with generalization error. Instead of viewing SAM as minimizing sharpness to improve generalization, our paper considers a new perspective based on SAM's training dynamics. We propose that perturbations in SAM perform *perturbed forgetting*, where they discard undesirable model biases to exhibit learning signals that generalize better. We relate our notion of forgetting to the information bottleneck principle, use it to explain observations like the better generalization of smaller perturbation batches, and show that perturbed forgetting can exhibit a stronger correlation with generalization than flatness. While standard SAM targets model biases exposed by the steepest ascent directions, we propose a new perturbation that targets biases exposed through the model's outputs. Our output bias forgetting perturbations outperform standard SAM, GSAM, and ASAM on ImageNet, robustness benchmarks, and transfer to CIFAR-{10,100}, while sometimes converging to sharper regions. Our results suggest that the benefits of SAM can be explained by alternative mechanistic principles that do not require flatness of the loss surface.

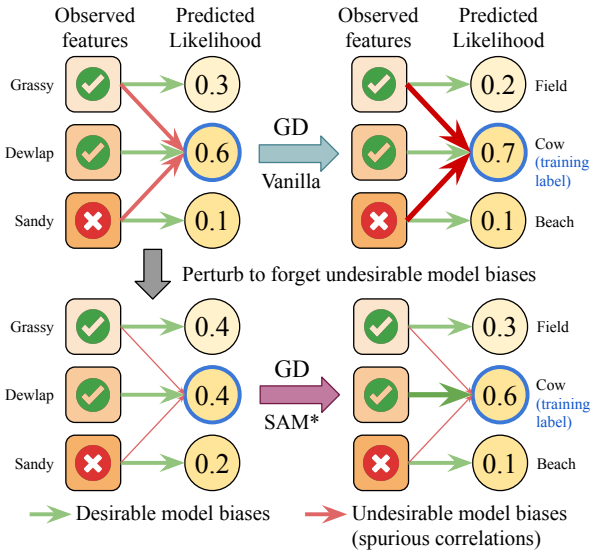


Figure 1. A simplified illustration of our mechanistic *perturbed forgetting* perspective of sharpness-aware minimization (SAM). We treat perturbations in each step of SAM as an opportunity to forget undesirable model biases. Here, the presence of 'grassy' or 'sandy' features spuriously contribute to the prediction of 'cow.' When gradient descent (GD) can strengthen these biases, leading to overfitting, the perturbation of SAM takes an ascent step to 'forget' them to allow computing a less biased gradient. *Not illustrated: this gradient is used to take a GD step at the *unperturbed* weights.

1. Introduction

The belief that flatter minima of the loss surface generalize better is commonplace in machine learning (Jiang et al., 2019). Sharpness-aware minimization (SAM) (Foret et al., 2020) and its variants (Kwon et al., 2021; Zhuang et al., 2022; Kim et al., 2022) are motivated and presented as methods to minimize sharpness to improve generalization. As many models trained with SAM exhibit better generalization, research continues to explore the principles behind it (Andriushchenko & Flammarion, 2022; Wen et al., 2022) to improve training algorithms. However, certain questions stand in the way of refining these techniques. First, the sharpness metric induced by SAM does not necessarily correlate with generalization in modern deep learning architectures (Andriushchenko et al., 2023b; Kaur et al., 2023). Furthermore, practically necessary concepts like m-sharpness are unsupported by the theory these methods are based on, casting doubt on the potential of progress upon assumptions that do not hold up empirically.

Instead of looking at these methods from the perspective of reducing sharpness, we offer a novel view by considering a mechanistic aspect of SAM's training dynamics, which

[*]Work done during an internship at Borealis AI [1]Mila, Université de Montréal [2]Borealis AI. Correspondence to: Ankit Vani <ankit.vani@umontreal.ca>.

Proceedings of the 41[st] International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

we term *perturbed forgetting*. Each update step in SAM comprises of perturbing the model parameters with a gradient ascent step, and using the gradients computed at these perturbed parameters to update the original weights. Under our perspective, we treat the perturbations as an opportunity to 'forget' undesirable model biases, as illustrated in Figure 1. While such model biases are forgotten only during perturbation, we can reinterpret concepts like minimizing the surrogate gap using GSAM (Zhuang et al., 2022) to provide explicit mechanisms to unlearn unnecessary biases. Our perspective also offers explanations for the generalization benefits of small perturbation batches (low m in m-sharpness) and the importance of worst-case perturbations over random ones with SAM (Andriushchenko & Flammarion, 2022), which we discuss in more detail in Section 3.

The biases a model learns can be exposed by probing its various aspects like gradients and outputs. Under the perturbed forgetting perspective, standard SAM perturbations target model biases exposed in the steepest gradient directions of small batches for forgetting. However, when we consider that a model's outputs can also expose model biases, we argue that the steepest ascent perturbation can have the opposite effect and amplify them. To address this limitation, we propose an output bias forgetting (OBF) perturbation in Section 4 that avoids amplifying these output-exposed biases, and optionally allows stronger forgetting by pushing predictions towards a uniform distribution. The success of our proposal suggests that non-standard probing mechanisms can be devised to target model biases in settings where the benefits of SAM are absent or minimal.

The notion of forgetting is related to the information bottleneck principle (Tishby et al., 2000), which suggests that optimal generalization may occur when a model retains only the information relevant to the task. Accordingly, the amount of task-irrelevant information discarded during perturbation enables us to quantify perturbed forgetting. We justify perturbed forgetting as an alternative to the narrative of sharpness minimization by showing that this quantity correlates with generalization more strongly than loss surface flatness in Section 6.1. Other forgetting techniques have been proposed in the literature (Zhou et al., 2022; Ash & Adams, 2020; Taha et al., 2021; Tiwari & Shenoy, 2023) which modify parameters in-place, but the dynamics of SAM offer the advantage of transient forgetting for computing updates without disrupting the learning state.

We summarize our contributions in this paper as follows[1]:

1. We present the perturbed forgetting perspective of SAM. We relate perturbed forgetting to generalization based on the information bottleneck principle, argue how standard SAM perturbations decrease an information-theoretic generalization bound, and empirically validate that forgetting can correlate with generalization better than loss surface flatness.
2. Embracing the perturbed forgetting perspective, we design the OBF perturbation that targets model biases exposed in the model's outputs. Despite not necessarily exhibiting the lowest sharpness, our perturbation leads to improved generalization with the SAM, GSAM, and ASAM frameworks on ImageNet (Deng et al., 2009) and robustness benchmarks using ViTs (Dosovitskiy et al., 2020) and ResNets (He et al., 2016).

Our results suggest that the training dynamics of SAM may be more important than minimizing loss surface sharpness. The pursuit of flat minima could be a red herring, and the benefits of SAM's training dynamics might be better explained by other mechanistic principles.

2. Background

We start by briefly detailing the preliminary concepts that we refer to in this paper.

2.1. Sharpness-Aware Minimization (SAM)

Sharpness-aware minimization (SAM) (Foret et al., 2020) is an optimization procedure that aims to minimize a PAC-Bayes upper-bound of the generalization error by considering perturbations of the model parameters. Let us represent a batch of n samples drawn from the data distribution $\mathcal{D}$ as $S \sim \mathcal{D}^n$. Then, for a loss function $L_{\boldsymbol{\theta}}(S)$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$, the sharpness-aware optimization problem is

$$\min_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{SAM}}(S) + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{1}$$

$$\text{where } L_{\boldsymbol{\theta}}^{\text{SAM}}(S) = \max_{\|\boldsymbol{\epsilon}\|_2 \leq \rho} L_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}(S). \tag{2}$$

Here, λ is an L2-regularization hyperparameter, and ρ is a hyperparameter that controls for the neighborhood size for the perturbation.

To make this min-max problem tractable with stochastic gradient descent, SAM approximates the inner maximization problem by considering a first-order Taylor approximation of $L_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}(S)$ w.r.t. $\boldsymbol{\epsilon}$ around $\mathbf{0}$, giving us the gradient

$$\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{SAM}}(S) \approx \nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S) \Big|_{\boldsymbol{\theta}+\rho \frac{\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S)}{\|\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S)\|_2}}. \tag{3}$$

A variant of SAM called m-SAM considers multiple perturbations using m-sized subsets of a training batch, which generalizes better than SAM in practice when m is small (Foret et al., 2020; Andriushchenko & Flammarion, 2022; Wen et al., 2022). m-SAM exhibits the update gradient:

[1]Source code: https://github.com/BorealisAI/perturbed-forgetting.

$$\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{m\text{-SAM}}(S) = \mathop{\mathbb{E}}_{\tilde{S}\sim S^m}\left[\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S)\Big|_{\boldsymbol{\theta}+\rho\frac{\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(\tilde{S})}{\|\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(\tilde{S})\|_2}}\right], \tag{4}$$

and the associated sharpness metric is termed m-sharpness (Foret et al., 2020).

2.2. Surrogate Gap Minimization with SAM (GSAM)

GSAM (Zhuang et al., 2022) defines the surrogate gap $h(\boldsymbol{\theta})$ as the difference between the maximum loss within an ϵ-neighborhood of parameters $\boldsymbol{\theta}$ and the loss at $\boldsymbol{\theta}$:

$$h(\boldsymbol{\theta}) = \max_{\|\boldsymbol{\epsilon}\|_2\leq\rho} L_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}(S) - L_{\boldsymbol{\theta}}(S). \tag{5}$$

The authors show that the loss surface is flatter as h gets closer to zero. To minimize the surrogate gap, the gradient $\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S)$ is first decomposed into components parallel and orthogonal to $\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{SAM}}(S)$. Denoting the orthogonal component as $\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{gap}}(S)$, the GSAM update gradient is

$$\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{GSAM}}(S) = \nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{SAM}}(S) - \xi\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}^{\text{gap}}(S), \tag{6}$$

where ξ is a hyperparameter that controls the step size in the direction of closing the surrogate gap.

2.3. Information Bottleneck in Deep Learning

The information bottleneck principle (Tishby et al., 2000) describes the minimal sufficient statistics of an input random variable $\boldsymbol{X}\in\mathcal{X}$ w.r.t. a target random variable $Y\in\mathcal{Y}$. In a neural network with L layers, it suggests that an optimal representation $\boldsymbol{Z}_l\in\mathcal{Z}_l$ for any $l\in[L]$ minimizes $I(\boldsymbol{X};\boldsymbol{Z}_l) - \beta I(\boldsymbol{Z}_l;Y)$, where I denotes mutual information and β trades off the representation complexity $I(\boldsymbol{X};\boldsymbol{Z}_l)$ with the amount of relevant target information $I(\boldsymbol{Z}_l;Y)$.

In their work justifying the benefit of information bottleneck in deep learning, Kawaguchi et al. (2023) present information theoretic bounds for generalization errors in neural networks comprised of L layers. Consider the data distribution $\mathcal{D}$ and a model f trained on a dataset $\mathrm{D}\subseteq\mathcal{D}^n$ with n samples. Define the generalization gap as

$$\Delta(f) = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},Y)\sim\mathcal{D}\\ \boldsymbol{Z}_l=f_{1:l}(\boldsymbol{X})\\ \hat{\boldsymbol{Y}}=f_{l+1:L}(\boldsymbol{Z}_l)}}\left[\mathcal{L}(Y,\hat{\boldsymbol{Y}})\right] - \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},Y)\sim\mathrm{D}\\ \boldsymbol{Z}_l=f_{1:l}(\boldsymbol{X})\\ \hat{\boldsymbol{Y}}=f_{l+1:L}(\boldsymbol{Z}_l)}}\left[\mathcal{L}(Y,\hat{\boldsymbol{Y}})\right]. \tag{7}$$

Here, $f_{i:j}$ represents a sub-network that takes input at layer i and produces an output $\boldsymbol{Z}_j\in\mathcal{Z}_j$ at layer j. We denote $\hat{\boldsymbol{Y}}\in\hat{\mathcal{Y}}$ as the random variable of class likelihoods according to the model, and $\mathcal{L}(Y,\hat{\boldsymbol{Y}})$ as the loss between the target Y and predictions $\hat{\boldsymbol{Y}}$.

Then, with high probability, the following holds (Kawaguchi et al., 2023):

$$\Delta(f)\in\tilde{O}\left(\sqrt{\frac{I(\boldsymbol{X};\boldsymbol{Z}_l\mid Y)+I(f_{1:l};\mathrm{D})}{n}}\right). \tag{8}$$

Importantly, the authors show that even when $I(\boldsymbol{X};\boldsymbol{Z}_l\mid Y)$ is infinite, such as in the case of some deterministic networks with continuous domains, the generalization bound holds with finite mutual information computed by assuming binning (Saxe et al., 2019). When invoking mutual information in this paper, we assume that binning can be performed to make these quantities finite.

3. Perturbed Forgetting Perspective of SAM

In this section, we detail our perspective of *perturbed forgetting*, under which we assert that SAM dynamics exhibit forgetting of undesirable model biases through perturbations to benefit generalization.

To start, we consider that the perturbations in SAM seek to exhibit a smaller generalization gap by discarding undesirable biases like spurious relationships. However, the purpose of perturbing is to exhibit a better learning signal in the gradient update. Therefore, the perturbation must not increase the likelihood of the targets, as doing so would dampen the necessary learning signal for the weight update. Due to this constraint, the decrease in generalization gap comes at the expense of increased generalization error, and optimal perturbations schemes should allow attaining a low generalization gap with a minimal increase of error.

Relation to Information Bottleneck. To understand how SAM perturbations can reduce the generalization gap through forgetting, we utilize the information bottleneck principle and the results of Kawaguchi et al. (2023). Let us consider the class likelihoods $\hat{\boldsymbol{Y}}$ as the representation in Equation (8). Then, for a model parameterized by $\boldsymbol{\theta}$, with high probability the following holds:

$$\Delta(\boldsymbol{\theta})\in\tilde{O}\left(\sqrt{\frac{I(\boldsymbol{X};\hat{\boldsymbol{Y}}\mid Y)+I(\boldsymbol{\theta};\mathrm{D})}{n}}\right). \tag{9}$$

Consider the perturbed parameters from Equation (2) as $\boldsymbol{\theta}^p = \arg\max_{\|\boldsymbol{\epsilon}\|_2\leq\rho} L_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}(S)$. Denoting the class likelihoods as $\hat{\boldsymbol{Y}}^p$ at $\boldsymbol{\theta}^p$, we conjecture that the SAM perturbation reduces both $I(\boldsymbol{\theta}^p;\mathrm{D})$ and $I(\boldsymbol{X};\hat{\boldsymbol{Y}}^p\mid Y)$ w.r.t. $\boldsymbol{\theta}$.

$I(\boldsymbol{\theta};\mathrm{D})$ quantifies the ability to identify a specific sampling of the training dataset $\mathrm{D}\sim\mathcal{D}$ by observing parameters $\boldsymbol{\theta}$. A better fit on the training data allows easier identification of the training dataset from the parameters. In contrast, SAM maximizes the loss and reduces the likelihood of the targets under the model, implying $I(\boldsymbol{\theta}^p;\mathrm{D})\leq I(\boldsymbol{\theta};\mathrm{D})$.

The other term, $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$, quantifies the amount of superfluous information (irrelevant for classification) encoded in the output likelihoods $\hat{\boldsymbol{Y}}$ about the inputs $\boldsymbol{X}$. We consider that this superfluous information can provide a view of the biases the model exhibits, and we refer to these biases as *output-exposed biases*. We can write $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$ as

$$I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim \mathrm{D}} \Big[H\Big(p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}), p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid y)\Big) - H\Big(p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x})\Big)\Big]. \tag{10}$$

In accordance with the information bottleneck principle, $\hat{\boldsymbol{Y}} = \mathrm{one\,hot}(Y)$ is one solution to minimizing $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$. However, we need to consider the constraint for learnability of not increasing the likelihood of the target. We argue that SAM can decrease Equation (10) under this constraint when the perturbation batch size is small.

Ensembles of Perturbations (m-SAM). We interpret m-SAM as generating an ensemble of perturbed models per update step using a batch S, where the distribution of class likelihoods for an input $\boldsymbol{x} \in \mathcal{X}$ is

$$p^{m\text{-SAM}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}\sim p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}\mid S)} \Big[p_{\boldsymbol{\theta}+\boldsymbol{\epsilon}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}) \Big], \tag{11}$$

and $\boldsymbol{\epsilon}$ is a perturbation sampled as

$$\tilde{S} \sim S^m, \quad \boldsymbol{\epsilon} = \rho \frac{\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(\tilde{S})}{\|\nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(\tilde{S})\|_2}. \tag{12}$$

However, with a small learning rate, we can also consider an implicit ensemble of perturbations across update steps in full-batch SAM with $|S| = m$. By choosing an appropriate perturbation scheme, a diverse ensemble can minimize Equation (10) by increasing the entropy $H(p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}))$ and decreasing the cross-entropy $H(p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}), p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid y)) = H(p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}), \mathbb{E}_{\boldsymbol{x}'\sim \mathrm{D}(\boldsymbol{X}\mid y)}[p_{\boldsymbol{\theta}}(\hat{\boldsymbol{Y}} \mid \boldsymbol{x}')])$ by making the high-entropy distributions similar for each input per label.

Note that in addition to diversity, the change in the inductive biases with perturbation influences the quality of the update gradients and their validity at the unperturbed $\boldsymbol{\theta}$. Constraining perturbations to an ϵ-neighborhood, as done by SAM, is a simple approach to maintaining this gradient validity.

Small Perturbation Batches (Low m in m-Sharpness). Unlike the desirable outcome of improved generalization on unseen samples when training to fit larger datasets, the "generalization" of maximization with a large perturbation batch beyond its samples can hamper the diversity of perturbations. For instance, requiring maximization to affect a large number of examples simultaneously can limit possible perturbations to those that discard the most prominent globally useful features or simply reduce prediction confidence. On the other hand, perturbing using small batches can offer steepest ascent directions that "overfit" differently, introducing the desired noise in the estimation of $\hat{\boldsymbol{Y}}$.

Forgetting Undesirable Biases. Global maximization is not the goal of perturbed forgetting, as it can generate poor models exhibiting low-quality gradients. Instead, we view maximization on a small number of examples as a mechanism to expose and "forget" undesirable shortcuts, or model biases, learned by the model pertaining to those examples. Without this forgetting, the same gradient directions would otherwise contribute to the next update step, potentially causing overfitting. While SAM discards model biases when computing update gradients, it does not immediately unlearn the biases at the unperturbed parameters. We conjecture that SAM will implicitly unlearn them over training by not utilizing them, but we can interpret surrogate gap minimization of GSAM as an explicit mechanism to unlearn forgotten biases. The inconsistency between gradient directions at the original and the perturbed parameters comes from the forgetting of model biases, and minimizing the surrogate gap using Equation (6) can be seen as minimizing this inconsistency.

Relation to Other Empirical Observations. Our perspective of SAM discarding undesirable biases aligns with empirical observations like SAM learning low-rank features (Andriushchenko et al., 2023a) and reducing harmful overfitting (Chen et al., 2023). As we do not explicitly call for any notion of flatness in the loss surface, our perspective does not clash with the challenges in correlating flatness with generalization (Andriushchenko & Flammarion, 2022; Andriushchenko et al., 2023b; Wen et al., 2023). Finally, we note that worst-case perturbations, which we view as targeting model biases, have been claimed to be important in SAM (Andriushchenko & Flammarion, 2022). However, in Section 4, we design an alternative perturbation to target output-exposed biases, which significantly improves generalization over using steepest ascent perturbations.

4. Perturb to Forget Output-Exposed Biases

In this section, we design an alternative perturbation function to forget undesirable model biases in SAM that are exposed through the model's outputs.

4.1. Setup

Consider a more general class of extragradient methods (Korpelevich, 1976; Juditsky et al., 2011; Mishchenko et al., 2020) that SAM belongs to, generalizing Equation (3) as:

$$\nabla_{\boldsymbol{\theta}} L^{\mathrm{EG}}_{\boldsymbol{\theta}}(S) \approx \nabla_{\boldsymbol{\theta}} L_{\boldsymbol{\theta}}(S) \Big|_{\boldsymbol{\theta}+\rho \frac{\nabla_{\boldsymbol{\theta}} L^{p}_{\boldsymbol{\theta}}(S)}{\|\nabla_{\boldsymbol{\theta}} L^{p}_{\boldsymbol{\theta}}(S)\|_2}}. \tag{13}$$

Here, the perturbed parameters are computed by taking a gradient ascent step to maximize the perturbation objective

$L^p_{\boldsymbol{\theta}}(S)$, which equals the task loss $L_{\boldsymbol{\theta}}(S)$ for SAM.

SAM and its variants (Zhuang et al., 2022; Kwon et al., 2021; Liu et al., 2022; Bahri et al., 2021) have commonly been evaluated on tasks such as image classification and language modeling, where the models are trained to maximize the likelihood of discrete outputs such as class predictions or tokens of a sequence. Here, we consider the task of multi-class classification with C classes, where a model parameterized by $\boldsymbol{\theta}$ is trained by minimizing the cross-entropy or the sigmoid cross-entropy (Beyer et al., 2020) loss between the target label $y \in \{1, \dots, C\}$ and the model predictions. When using softmax on the model outputs $\boldsymbol{z} \in \mathbb{R}^C$ to represent the predicted distribution $\hat{\boldsymbol{y}}$, the gradient of the cross-entropy loss for a single example can be written as:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}^{\text{CE}}(y, \hat{\boldsymbol{y}}) = \mathop{\mathbb{E}}_{i \sim \hat{\boldsymbol{y}}} [\nabla_{\boldsymbol{\theta}} z_i] - \nabla_{\boldsymbol{\theta}} z_y. \tag{14}$$

The sigmoid cross-entropy loss, which has been shown to improve ImageNet accuracy, exhibits a similar gradient, but with $\mathbb{E}_{i \sim \hat{\boldsymbol{y}}} [\nabla_{\boldsymbol{\theta}} z_i]$ replaced by $\sum_{i=1}^C \hat{y}_i \nabla_{\boldsymbol{\theta}} z_i$ due to $\sum_{i=1}^C \hat{y}_i \neq 1$ in general. However, for ease of notation, we choose to abuse the expectation notation when referring to the gradients of both losses.

4.2. Output Bias Forgetting (OBF) Perturbation

Similar to the steepest ascent perturbation discussed in Section 3, we aim to reduce $I(\boldsymbol{\theta}^p; \mathrm{D})$ and $I(\boldsymbol{X}; \hat{\boldsymbol{Y}}^p \mid Y)$ at the perturbed parameters $\boldsymbol{\theta}^p$ w.r.t. $\boldsymbol{\theta}$. We choose to retain decreasing the target likelihood as a way to avoid increasing $I(\boldsymbol{\theta}^p; \mathrm{D})$. However, we approach minimizing the superfluous information $I(\boldsymbol{X}; \hat{\boldsymbol{Y}}^p \mid Y)$ with considerations to reduce output-exposed biases.

When minimizing the loss by taking a step in the negative direction of Equation (14), the non-target logits are chosen based on their current corresponding likelihoods and pushed down. While these semantics are desirable during minimization, maximizing sharpens the non-target predictions to arrive at parameters that potentially amplify the model biases if they are exposed in $\hat{\boldsymbol{Y}}$. Instead, we propose a perturbation function that avoids sharpening the model predictions on maximization, and optionally weakens the exposed model biases when they start being useful in training.

We introduce our output bias forgetting (OBF) perturbation $\mathcal{L}^{\text{BF}}$, defined for a single example as:

$$\mathcal{L}^{\text{BF}}(y, \hat{\boldsymbol{y}}) = (1-\alpha)\mathcal{L}^{\text{CE}}(y, \hat{\boldsymbol{y}}) - \mathcal{L}^{\text{CE}}(\mathcal{U}, \hat{\boldsymbol{y}}), \tag{15}$$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}^{\text{BF}}(y, \hat{\boldsymbol{y}}) = \mathop{\mathbb{E}}_{i \sim \mathcal{U}} [\nabla_{\boldsymbol{\theta}} z_i] - \left(\alpha \mathop{\mathbb{E}}_{i \sim \hat{\boldsymbol{y}}} [\nabla_{\boldsymbol{\theta}} z_i] + (1-\alpha) \nabla_{\boldsymbol{\theta}} z_y \right). \tag{16}$$

Here, $\mathcal{U}$ denotes a uniform distribution and $\alpha \in [0, 1]$ controls how much to weaken the model biases. When $\alpha = 0$, we avoid explicitly changing the magnitude of the model biases. Such perturbations can be beneficial at the beginning of training, when the model's biases are not useful for the training task but need to be considered to efficiently traverse away from the initialization. When $\alpha = 1$, maximizing $\mathcal{L}^{\text{BF}}$ becomes equivalent to minimizing the cross-entropy loss for a uniform target. Once the model has learned biases that are useful for the training task, but undesirable for generalization, a perturbation towards uniformity can help the model discard these biases in computing the update gradient.

As useful but undesirable model biases could emerge later in training, we propose determining the value of α for each sample during training based on the likelihood $\hat{y}_y$ the model assigns to the ground-truth target y:

$$\alpha = \gamma \max \left(\frac{1 - \lambda / \hat{y}_y}{1 - \lambda}, 0 \right). \tag{17}$$

We treat $\gamma \in [0, 1]$ and $\lambda \in [0, 1)$ as hyperparameters such that α becomes non-zero if $\hat{y}_y > \lambda$, and increases linearly from 0 to γ as the perplexity $1/\hat{y}_y$ goes from $1/\lambda$ to 1. A reasonable choice for λ is $1/C$ and optimal values of γ are either 1 or close to 0 depending on the model architecture.

Finally, we note that the complexity of replacing the steepest ascent perturbation with OBF remains the same as standard SAM. The two forward and backward passes dominate the computation time for each iteration. We present the full algorithm utilizing OBF within SAM dynamics in Appendix A.

5. Related Work

We situate our contributions amongst other approaches of explaining the workings of SAM (Foret et al., 2020) and other methods of improving generalization by forgetting undesirable model biases.

Understanding SAM. The original explanation for SAM is based on minimizing the PAC-Bayes upper bound from Foret et al. (2020). Methods like GSAM (Zhuang et al., 2022) and ASAM (Kwon et al., 2021) adapt this bound to propose variants of the SAM algorithm. Often, the importance of minimizing sharpness is assumed, and explanations for the success of SAM comprise of showing how it attains flatter minima (Bartlett et al., 2023; Wen et al., 2022; Ujváry et al., 2022; Möllenhoff & Khan, 2023; Kwon et al., 2021). However, the importance of sharpness is debated, as it does not necessarily correlate with generalization error (Andriushchenko et al., 2023b; Mueller et al., 2023; Kim et al., 2023) in modern deep neural networks or shallow architectures (Wen et al., 2023). Other factors such as data distribution (Wen et al., 2023), architecture, and hyperparameters play critical roles in success of SAM and its variants (Andriushchenko et al., 2023b; Wen et al., 2023). Andriushchenko & Flammarion (2022) point out that the

original PAC-Bayes bound does not explain all the aspects of SAM's success. For example, using the worst-case perturbations instead of average-case as is practically done, only makes this bound less tight. They also suggest that some quantity other than sharpness is implicitly minimized when using small perturbation batches in SAM. Our paper offers a response by highlighting the advantage of smaller perturbation batches from a different perspective. Complementary to our notion of SAM perturbations discarding undesirable model biases to improve generalization in realistic training settings, Chen et al. (2023) formally prove that SAM avoids harmful overfitting in two-layer ReLU convolutional networks. Like us, Baek et al. (2024) identify a different set of principles than sharpness minimization to explain SAM's benefits. They do so in the setting of label noise, attributing SAM's label noise robustness to a dynamic mechanism that learns clean examples before fitting noisy ones.

Forgetting. "Forget-and-relearn" (Zhou et al., 2022) is a general framework that proposes that a mechanism of iteratively forgetting undesirable information and relearning it can improve generalization. This framework encompasses other methods such as iterative magnitude pruning (Frankle & Carbin, 2018), knowledge evolution (Taha et al., 2021), and neural iterated learning (Ren et al., 2020). Existing forget-and-relearn approaches modify the model parameters in-place, necessitating infrequent forgetting operations and the inefficiency of retraining parts of the network. In contrast, under the perturbed forgetting perspective, the dynamics of SAM allow constructing transient information bottlenecks for computing update gradients without damaging the current learning state at every update step. Like the OBF perturbation, Tiwari & Shenoy (2023) use the gradient of the cross-entropy loss towards a uniform distribution to target model biases to forget. However, they utilize an auxiliary layer for predictions for computing these gradients to avoid affecting the actual model likelihoods. In contrast, we utilize the model likelihoods themselves as the affected likelihoods persist only temporarily per perturbation.

6. Experiments

6.1. Perturbed Forgetting and Generalization

In Sections 3 and 4, we posited that the superfluous information quantified by $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$ enables access to the model's biases and perturbing to minimize this quantity with SAM dynamics improves generalization. In this section, we support our claims by measuring the correlation of forgetting output-exposed biases $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y) - I(\boldsymbol{X}; \hat{\boldsymbol{Y}}^p \mid Y)$ with the model's generalization. When $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$ is estimated by thresholding at different values, we find the existence of thresholds that exhibit stronger correlation with generalization than loss surface flatness.

Training. We train a pool of ViT-S/32 models on CIFAR-10 with three different SAM perturbation strategies: Steepest Ascent (standard SAM), OBF with $\lambda = 1/3$, and OBF with $\gamma = 0$. For each strategy, we train models with perturbation batch sizes $m \in \{2^k \mid k \in \{0, \ldots, 9\}\}$. All models are trained with the same learning rate without decay, and with the same weight decay and perturbation radius ρ hyperparameters to avoid their confounding effects on the training dynamics. We tune these hyperparameters by sweeping across a representative subset of the perturbation settings to ensure that they are comparable to the best-performing hyperparameters for the individual settings. We provide the training hyperparameters in Appendix C.

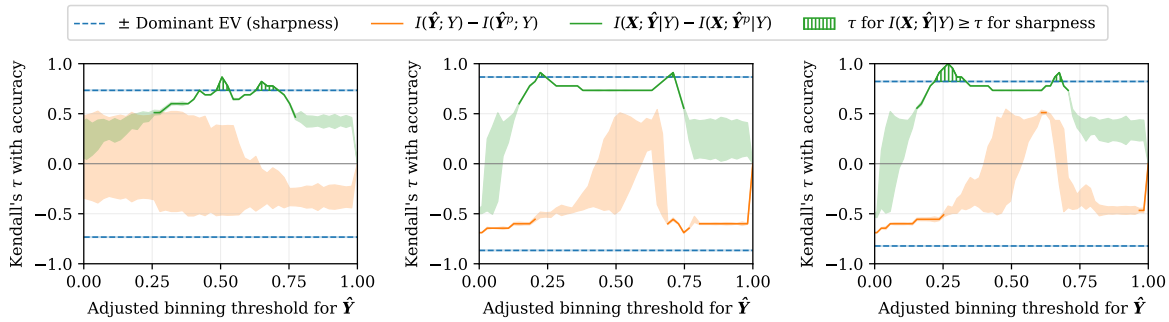
Data Collection. Unlike sharpness, which can be evaluated on the converged parameters of the model, perturbations are inherently dynamic and need to be captured at various points during training. To this end, we collect the softmax model outputs $\hat{\boldsymbol{Y}}$ on the CIFAR-10 validation set every 25th epoch during training for unperturbed and perturbed parameters for our pool of models.

Mutual Information Estimation. As the bounds introduced by Kawaguchi et al. (2023) also hold with the assumption of binning, we utilize a simple binning strategy of discretizing the model's softmax outputs to binary based on a threshold. With 10 output dimensions for the 10 classes, the maximum possible number of bins is 2^{10}. We estimate the mutual information for thresholds $t = 10^r$ at 100 values of r linearly spaced between -12 to 0.

For any model checkpoint, the estimated mutual information monotonically increases and then decreases as the binning threshold is increased from 0 to 1. Note that as a model is trained, its predictions get sharper and the binning threshold at which the maximum is attained quickly becomes much smaller than chance. We focus on the higher-magnitude variations in model outputs to understand the impact of forgetting, which are captured at thresholds greater than the one exhibiting the highest mutual information.

Different checkpoints, including those for different epochs of the same training run, attain the maximum at different thresholds. To allow comparison across epochs and different values of m, we first normalize the maximum to 1. Then, we adjust the binning thresholds by resampling such that the unperturbed mutual information decreases linearly from 1 to 0 between thresholds 0 to 1. Consequently, any specific adjusted binning threshold exhibits the same unperturbed $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y)$ across all collected checkpoints.

Correlating Forgetting with Generalization. We average the difference $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y) - I(\boldsymbol{X}; \hat{\boldsymbol{Y}}^p \mid Y)$ across all adjusted thresholds for every model and epoch per perturbation type. We evaluate the Kendall rank correlation



(a) SAM (b) OBF ($\gamma = 1$) (c) OBF ($\gamma = 1/3$)

Figure 2. Kendall's τ correlation of accuracy with sharpness and mutual information metrics averaged over epochs for models trained with different SAM perturbations on CIFAR-10. We train models with perturbation batch size $m \in \{2^k \mid k \in \{0, \ldots, 9\}\}$ for each perturbation. Shaded regions indicate the p-value estimated with a permutation test, and we show solid lines only when the p-value ≤ 0.05.

between this difference and the final CIFAR-10 test accuracy the model attains. We follow the same procedure to also evaluate $I(\hat{\boldsymbol{Y}}; Y) - I(\hat{\boldsymbol{Y}}^p; Y)$, which quantifies the change in task-relevant information when perturbing. Finally, we calculate the correlation between sharpness and the same accuracy for comparison, where sharpness is the Hessian's dominant eigenvalue estimated using power iteration.

Results. We present the estimated correlations for each adjusted threshold value and perturbation type in Figure 2. Discretizing at the thresholds with the highest correlation of accuracy with $I(\boldsymbol{X}; \hat{\boldsymbol{Y}} \mid Y) - I(\boldsymbol{X}; \hat{\boldsymbol{Y}}^p \mid Y)$ (green curves) reveals undesirable information encoded in the model outputs, that if targeted for forgetting, leads to improved generalization. At these thresholds, we find the correlation of accuracy with forgetting information about the classification target (orange curves) to remain negative, further indicating that the generalization benefits come from discarding superfluous information. Finally, we highlight the regions where forgetting correlates with generalization (green curves) more strongly than flatness (blue lines) with a green hatch pattern.

Our results demonstrate the existence of output-exposed biases and the generalization benefit of forgetting them.

6.2. Standard Benchmarks

We now study the generalization benefits of OBF by comparing models trained with varying architectures and perturbation schemes on standard benchmarks. We also present additional baselines and settings, including ASAM (Kwon et al., 2021), in Appendix B.

Datasets. We train our models on ImageNet-1K, also known as ImageNet-V1 (Deng et al., 2009), and also perform finetuning experiments with CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). When training from scratch, we evaluate on the ImageNet validation set, and the additional test sets ImageNet-Real (Beyer et al., 2020) and ImageNet-V2 (Recht et al., 2019). ImageNet-Real corrects idiosyncrasies and errors in the labeling of the original validation set and ImageNet-V2 contains newly-collected data following the original ImageNet data creation process. Additionally, we evaluate our models on the out-of-distribution robustness benchmarks ImageNet-R (Hendrycks et al., 2021), which contains renditions of 200 ImageNet classes in various forms, and ImageNet-Sketch (Wang et al., 2019), which has black-and-white sketch images for every ImageNet class. For our transfer learning experiments, we evaluate the models on the test splits of CIFAR-$\{10,100\}$.

Models. We run our experiments with two model families: vision transformers (ViT) (Dosovitskiy et al., 2020) and residual networks (ResNet) (He et al., 2016). For ViT, we experiment with ViT-S/32 and ViT-S/16, and choose ResNet-50 and ResNet-101 for the ResNet experiments.

Training. We follow the setting of GSAM (Zhuang et al., 2022) and Chen et al. (2021), and train our models with Inception-style pre-processing (Szegedy et al., 2015) without strong data augmentations for both ViT and ResNet models. All models are trained with a global batch size of 4096, perturbation batch size $m = 64$, and linear learning rate decay schedule with warmup. We apply the the same scheduling of the perturbation radius ρ that GSAM uses for both GSAM and SAM, which provide stronger baseline results, but keep ρ constant when using the OBF perturbation. We provide all hyperparameter values in Appendix C.

Finetuning. When finetuning on CIFAR-$\{10,100\}$, we use the same pre-processing scheme as we do for training. We finetune ViT-S/32 and ResNet-50 with SGD with momentum 0.9 for 100 epochs, without weight decay, and gradients clipped to global norm 1. We use a smaller batch

Table 1. Top-1 accuracies on ImageNet and robustness datasets. For SAM and GSAM, models are trained with standard steepest ascent (STEEP) and output bias forgetting (OBF) perturbations. Sharpness (dominant eigenvalue) is estimated for each model using power iteration. Standard deviations are reported with three trials.

MODEL	METHOD	PERTURB	IMAGENET-					SHARPNESS
			V1	REAL	V2	R	SKETCH	
VIT-S/32	ADAMW	NONE	69.29±0.26	75.31±0.28	55.48±0.58	19.02±0.47	16.38±0.34	165.6±15.2
	SAM	STEEP	72.77±0.06	78.89±0.05	58.81±0.33	21.63±0.23	19.68±0.50	14.9±1.1
		OBF	**74.49±0.04**	81.31±0.05	**61.13±0.18**	**25.31±0.41**	**22.58±0.13**	3.9±1.4
	GSAM	STEEP	73.41±0.05	79.48±0.08	59.94±0.15	22.18±0.15	20.28±0.15	11.6±1.2
		OBF	74.41±0.12	**81.41±0.11**	**61.08±0.18**	25.15±0.23	22.24±0.07	**3.1±0.7**
VIT-S/16	ADAMW	NONE	74.30±0.10	80.04±0.04	61.28±0.09	20.25±0.27	18.15±0.11	59.0±9.4
	SAM	STEEP	78.73±0.08	85.47±0.05	66.98±0.14	25.69±0.09	24.10±0.33	**2.4±0.4**
		OBF	**80.30±0.13**	86.14±0.13	68.62±0.05	27.19±0.27	**26.45±0.22**	17.1±2.8
	GSAM	STEEP	78.95±0.13	84.31±0.06	66.80±0.43	24.92±0.43	24.41±0.52	6.0±0.3
		OBF	**80.32±0.06**	**86.26±0.07**	**68.83±0.12**	**27.48±0.10**	25.92±0.13	4.3±1.4
RESNET-50	SGD	NONE	76.86±0.07	83.28±0.11	65.00±0.14	20.29±0.36	20.53±0.46	230.4±42.7
	SAM	STEEP	77.49±0.06	83.78±0.05	65.26±0.21	21.08±0.16	21.18±0.32	170.1±18.9
		OBF	**77.67±0.07**	84.01±0.03	65.70±0.45	21.63±0.18	**22.17±0.26**	**164.4±25.0**
	GSAM	STEEP	77.43±0.12	83.79±0.19	65.37±0.26	21.37±0.21	21.52±0.56	171.0±16.8
		OBF	**77.66±0.08**	**84.09±0.07**	**66.01±0.09**	**21.76±0.23**	**22.26±0.47**	**161.4±10.9**
RESNET-101	SGD	NONE	78.44±0.08	84.39±0.02	66.61±0.19	22.91±0.83	23.45±1.31	228.1±29.8
	SAM	STEEP	79.09±0.08	85.05±0.09	67.24±0.20	23.64±0.38	24.80±0.20	**155.1±12.0**
		OBF	79.27±0.06	85.17±0.10	67.85±0.17	24.21±0.26	**25.56±0.47**	170.4±2.1
	GSAM	STEEP	79.11±0.04	85.00±0.08	67.52±0.21	23.65±0.39	24.79±0.13	166.1±2.8
		OBF	**79.40±0.07**	**85.37±0.16**	**68.05±0.35**	**24.52±0.10**	**25.44±0.33**	165.7±28.4

size of 512, but keep the perturbation batch size $m = 64$. All other hyperparameters are provided in Appendix C.

Metrics. We report the generalization performance as the classification top-1 accuracy on the selected evaluation datasets. Additionally, we also report sharpness of our reported models, which is the Hessian's dominant eigenvalue estimated using the power iteration method. We report standard deviations for our metrics where available with three trials when training from scratch and six trials when finetuning. The six finetuning trials comprise of three groups of two finetuning trials, each group finetuning a model from one of the three training trials.

6.2.1. IMAGENET GENERALIZATION

We present our results on ImageNet evaluation and robustness benchmarks in Table 1. First, we note that training SAM with the OBF perturbation generalizes better than using standard steepest ascent perturbations with either SAM or GSAM in a majority of the studied benchmarks and methods. Additionally, utilizing GSAM with OBF further improves results in most settings for ViTs. Under the perturbed forgetting perspective, both steepest ascent perturbations and OBF target model biases for forgetting, but the kinds of biases exposed through the probing mechanisms they utilize can be different. Our results suggest that the outputs of ViTs provide significantly better access to its undesirable biases compared to its steepest ascent directions, whereas the improvements of using one perturbation over another is small for ResNets.

Furthermore, our results also indicate that the models that converge to the flattest regions of the loss surface seldom perform the best. Moreover, despite OBF outperforming steepest ascent perturbations in most settings, it only exhibits lowest sharpness in the case of ViT-S/32. Our results support the notion that the training dynamics of SAM are critical for generalization. We also remark that while SAM and GSAM work best with tricks like scheduling the perturbation radius, the OBF perturbation outperforms them without resorting to doing so.

6.2.2. TRANSFER LEARNING TO CIFAR DATASETS

We study the transfer learning capability of models trained on ImageNet with SAM with our studied perturbations, as well as the ability to finetune ImageNet-pretrained models with these methods. We present our transfer learning results

Table 2. Transfer learning top-1 accuracies of models finetuned on CIFAR-{10,100} after pretraining on ImageNet, where either pretraining (PRE) or finetuning (FT) uses SAM. Standard deviations are reported with six trials.

MODEL	SAM PERTURB		CIFAR-	
	PRE	FT	10	100
VIT-S /32	NONE	STEEP OBF	97.74±0.08 **97.79±0.04**	86.94±0.09 **87.04±0.13**
	STEEP OBF	NONE	97.69±0.04 **97.92±0.05**	86.21±0.17 **86.99±0.10**
RESNET -50	NONE	STEEP OBF	96.84±0.12 **96.91±0.10**	83.29±0.25 **83.41±0.23**
	STEEP OBF	NONE	96.16±0.16 **96.40±0.24**	81.81±0.21 **81.91±0.23**

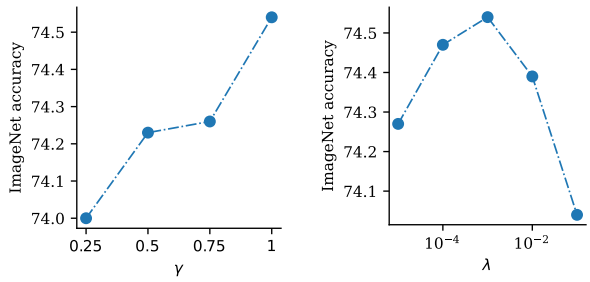


Figure 3. Effect of the hyperparameters γ (with fixed $\lambda = 1/C$, where C is the number of classes) and λ (with fixed $\gamma = 1$) on ImageNet top-1 accuracy for ViT-S/32 trained using the output bias forgetting (OBF) perturbation in SAM.

for CIFAR-10 and CIFAR-100 in Table 2.

We observe that models pretrained with OBF allow improved transfer to CIFAR-10 and CIFAR-100. The advantages of OBF can also be seen when finetuning models pretrained without SAM in all settings but one.

6.2.3. EFFECT OF γ AND λ

The hyperparameters γ and λ control the conditions for activating a push towards a uniform model prediction during OBF. We find that their optimal values differ substantially by model architecture. For ResNets, we find the best value of γ to be close to zero, making it insensitive to the choice of λ. This setting weighs the gradients for each non-target prediction equally, unlike steepest ascent perturbations which weigh the gradients proportionally to the predicted likelihoods. For ViTs, we find it useful to allow an explicit pressure towards uniform predictions by setting $\gamma = 1$. Both settings avoid sharpening the predictions when perturbing, as sharpening can exacerbate output-exposed biases.

Figure 3 shows how ImageNet performance changes with varying γ and λ with ViT-S/32. When γ is low, the forgetting strength is lower, and the performance gain from perturbed forgetting is lower. With $\gamma = 1$, λ determines the target likelihood threshold for each sample beyond which the perturbation pushes predictions towards uniformity. When λ is too high, the push towards uniformity does not start until much later in training, missing opportunities to improve generalization. On the other hand, when λ is too small, the push starts before the model learns useful mappings, hurting training efficiency and, consequently, generalization. The optimal value $1/C$, where C is the number of classes, suggests that such explicit forgetting is beneficial when it targets model biases that become available once the model starts performing better than chance for a given sample.

7. Discussion

Generalization to the true data distribution can be understood as generalization across plausible structural variations of the data. Accordingly, the flatness we desire is in the space of these variations, not necessarily in the parameter space. There is no inherent causal link between these two notions of flatness without additional assumptions. Relating our contributions to sharpness minimization, we argue that perturbed forgetting can provide a framework for pursuing flatness in the space of data variations by reducing sensitivity to sampling of the training dataset. Small perturbation batches can reveal shortcuts the model has learned for the included examples, and gathering gradients after forgetting encourages the model to instead rely on and strengthen a global structure that benefits a larger set of examples.

Our paper uses the perturbed forgetting perspective to devise a perturbation that can outperform standard SAM, GSAM, and ASAM (in Appendix B). However, we do not claim OBF to be an optimal perturbation for all settings. Both steepest ascent and OBF perturbations allow perturbed forgetting of model biases, albeit potentially of different biases. Furthermore, like many prior works on SAM (Foret et al., 2020; Kwon et al., 2021; Liu et al., 2022), our paper is limited to image classification experiments. Undesirable model biases can be easier to target and forget with different perturbations under different architectures and training domains. Characterizing the exact nature of the biases targeted by different perturbations and a more formal theoretical treatment of perturbed forgetting can provide insights for generalizing in a wide range of settings, including those where standard SAM is ineffective or difficult to integrate.

Acknowledgements

We are grateful to the anonymous reviewers for their valuable feedback during the review period.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Andriushchenko, M. and Flammarion, N. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 639–668. PMLR, 2022.

Andriushchenko, M., Bahri, D., Mobahi, H., and Flammarion, N. Sharpness-aware minimization leads to low-rank features. *arXiv preprint arXiv:2305.16292*, 2023a.

Andriushchenko, M., Croce, F., Müller, M., Hein, M., and Flammarion, N. A modern look at the relationship between sharpness and generalization. *arXiv preprint arXiv:2302.07011*, 2023b.

Ash, J. and Adams, R. P. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Baek, C., Kolter, J. Z., and Raghunathan, A. Why is SAM robust to label noise? In *The Twelfth International Conference on Learning Representations*, 2024.

Bahri, D., Mobahi, H., and Tay, Y. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

Bartlett, P. L., Long, P. M., and Bousquet, O. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24(316):1–36, 2023.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Chen, X., Hsieh, C.-J., and Gong, B. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

Chen, Z., Zhang, J., Kou, Y., Chen, X., Hsieh, C.-J., and Gu, Q. Why does sharpness-aware minimization generalize better than sgd? *arXiv preprint arXiv:2310.07269*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Dohare, S., Hernandez-Garcia, J. F., Rahman, P., Sutton, R. S., and Mahmood, A. R. Maintaining plasticity in deep continual learning. *arXiv preprint arXiv:2306.13812*, 2023.

D'Oro, P., Schwarzer, M., Nikishin, E., Bacon, P.-L., Bellemare, M. G., and Courville, A. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Elsayed, M. and Mahmood, A. R. Utility-based perturbed gradient descent: An optimizer for continual learning. In *OPT 2023: Optimization for Machine Learning*, 2023.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Juditsky, A., Nemirovski, A., and Tauvel, C. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

Kaur, S., Cohen, J., and Lipton, Z. C. On the maximum hessian eigenvalue and generalization. In *Proceedings on*, pp. 51–65. PMLR, 2023.

Kawaguchi, K., Deng, Z., Ji, X., and Huang, J. How does information bottleneck help deep learning? *arXiv preprint arXiv:2305.18887*, 2023.

Kim, H., Park, J., Choi, Y., and Lee, J. Fantastic robustness measures: The secrets of robust generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kim, M., Li, D., Hu, S. X., and Hospedales, T. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pp. 11148–11161. PMLR, 2022.

Korpelevich, G. M. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Kumar, S., Marklund, H., and Van Roy, B. Maintaining plasticity via regenerative regularization. *arXiv preprint arXiv:2308.11958*, 2023.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

Liu, Y., Mai, S., Cheng, M., Chen, X., Hsieh, C.-J., and You, Y. Random sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 35:24543–24556, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Mishchenko, K., Kovalev, D., Shulgin, E., Richtárik, P., and Malitsky, Y. Revisiting stochastic extragradient. In *International Conference on Artificial Intelligence and Statistics*, pp. 4573–4582. PMLR, 2020.

Möllenhoff, T. and Khan, M. E. SAM as an optimal relaxation of bayes. In *The Eleventh International Conference on Learning Representations*, 2023.

Mueller, M., Vlaar, T., Rolnick, D., and Hein, M. Normalization layers are all that sharpness-aware minimization needs. *arXiv preprint arXiv:2306.04226*, 2023.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Ren, Y., Guo, S., Labeau, M., Cohen, S. B., and Kirby, S. Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*, 2020.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Taha, A., Shrivastava, A., and Davis, L. S. Knowledge evolution in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12843–12852, 2021.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Tiwari, R. and Shenoy, P. Overcoming simplicity bias in deep networks using a feature sieve. In *International Conference on Machine Learning*, pp. 34330–34343. PMLR, 2023.

Ujváry, S., Telek, Z., Kerekes, A., Mészáros, A., and Huszár, F. Rethinking sharpness-aware minimization as variational inference. *arXiv preprint arXiv:2210.10452*, 2022.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Wen, K., Ma, T., and Li, Z. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2022.

Wen, K., Ma, T., and Li, Z. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *arXiv preprint arXiv:2307.11007*, 2023.

Zhou, H., Vani, A., Larochelle, H., and Courville, A. Fortuitous forgetting in connectionist networks. In *International Conference on Learning Representations*, 2022.

Zhuang, J., Gong, B., Yuan, L., Cui, Y., Adam, H., Dvornek, N., Tatikonda, S., Duncan, J., and Liu, T. Surrogate gap minimization improves sharpness-aware training. *arXiv preprint arXiv:2203.08065*, 2022.

A. Optimization Algorithm

Algorithm A.1 Iterated Output Bias Forgetting

Input: Training dataset $\mathrm{D} = \cup_{i=1}^{|\mathrm{D}|}\{\boldsymbol{x}^{(i)}, y^{(i)}\} \subseteq \mathcal{D} \subseteq \mathcal{X} \times \{1, \ldots, C\}$, likelihood function $f_{\boldsymbol{\theta}}$, perturbation step size ρ, update step size η, bias forgetting strength hyperparameters γ and λ.
Initialize weights $\boldsymbol{\theta}_0$, step $t = 0$;
repeat
  Sample $(\boldsymbol{x}, y)$ from D;
  $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}_t}(\boldsymbol{x})$; {Current likelihood}
  $\alpha = \gamma \max\left(\frac{1-\lambda/\hat{y}_y}{1-\lambda}, 0\right)$; {Forgetting strength}
  $\Delta_t = \nabla_{\boldsymbol{\theta}_t}\left[(1-\alpha)\mathcal{L}^{\mathrm{CE}}(y, \hat{\boldsymbol{y}}) - \mathcal{L}^{\mathrm{CE}}(\mathcal{U}, \hat{\boldsymbol{y}})\right]$;
  $\boldsymbol{\theta}_t^p = \boldsymbol{\theta}_t + \rho\frac{\Delta_t}{\|\Delta_t\|}$; {Perturbed weights}
  $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla_{\boldsymbol{\theta}_t^p}\mathcal{L}^{\mathrm{CE}}\left(y, f_{\boldsymbol{\theta}_t^p}(\boldsymbol{x})\right)$; {Update}
until convergence

We present the optimization algorithm incorporating OBF within SAM dynamics, without perturbation ensembling for simplicity, in Algorithm A.1.

B. Additional Experiments

B.1. Standard Benchmarks

We provide additional results on ViT-S/32 and ResNet-50 in Table B.1, with the following new settings compared to Table 1.

ASAM. We compare OBF with the standard steepest ascent perturbations in the ASAM (Kwon et al., 2021) framework. ASAM with OBF outperforms standard ASAM on all benchmarks, and yet exhibits higher sharpness. Our results further support the importance of the forgetting mechanism over loss surface flatness of the parameters.

Shrink and Perturb. Ideas related to forgetting have also been explored in settings of non-stationarity, with a goal of restoring plasticity under continually changing data distributions (Ash & Adams, 2020; Elsayed & Mahmood, 2023; Dohare et al., 2023). Although non-stationarity is not a problem in our evaluation settings and SAM does not perform in-place forgetting of the parameters, we consider a shrink-and-perturb (Ash & Adams, 2020) baseline for empirical comparison. To ease hyperparameter search and maintain reasonable parameter value scales, we adopt modifications from D'Oro et al. (2023); Kumar et al. (2023) to perform shrinking and perturbing simultaneously through the linear interpolation of parameters towards their initialization. We perturb the parameters for every update using the tuned interpolation factor of 10^{-5}. We find shrink-and-perturb to remain comparable with the vanilla baselines, suggesting a generalization advantage of perturbed forgetting over in-place forgetting.

Random SAM Perturbations. We add a SAM baseline that samples perturbations uniformly on a unit hypersphere of radius ρ. Without an explicit mechanism to target undesirable model biases, we find that SAM with random perturbations exhibits performance closer to vanilla training than SAM with steepest ascent or OBF perturbations.

Label Smoothing. Finally, we consider a vanilla baseline with label smoothing of 0.1, which provides an alternative approach for encouraging uniform predictions to improve generalization. We find label smoothing to improve performance over the vanilla settings, but it does not outperform our SAM settings. We note that label smoothing deviates from our other settings by changing the training objective, adding constraints not otherwise imposed in fitting the training data.

B.2. Ablating Dynamic vs. Fixed OBF α

The OBF hyperparameters λ and γ are used to dynamically produce a value of α per sample based on Equation (17). In this section, we perform an ablation experiment with fixed values of α with ViT-S/32 models trained on ImageNet. Our results

Table B.1. Top-1 accuracies on ImageNet and robustness datasets. Without SAM, models are trained in either a vanilla setting, with per-step shrink-and-perturb (SHRINKPERTURB), or with label smoothing of 0.1 (LABELSMOOTH). SAM and ASAM models are trained with standard steepest ascent (STEEP) and output bias forgetting (OBF) perturbations. SAM is also trained with random (RANDOM) perturbations. Sharpness (dominant eigenvalue) is estimated for each model using power iteration. Standard deviations are reported with three trials. †*Note:* Label smoothing produces a different training objective and loss surface geometry compared to the other settings.

MODEL	METHOD	PERTURB	IMAGENET-					SHARPNESS
			V1	REAL	V2	R	SKETCH	
VIT-S /32	ADAMW	NONE	69.29±0.26	75.31±0.28	55.48±0.58	19.02±0.47	16.38±0.34	165.6±15.2
	SHRINKPERTURB	NONE	69.05±0.07	75.29±0.08	55.45±0.32	18.99±0.11	16.14±0.23	325.8±290.9
	LABELSMOOTH†	NONE	69.75±0.10	75.94±0.10	55.95±0.30	19.66±0.16	16.82±0.20	1959.9±1319.8
	SAM	STEEP	72.77±0.06	78.89±0.05	58.81±0.33	21.63±0.23	19.68±0.50	14.9±1.1
		OBF	74.49±0.04	**81.31±0.05**	**61.13±0.18**	**25.31±0.41**	**22.58±0.13**	**3.9±1.4**
		RANDOM	69.23±0.28	75.43±0.31	55.27±0.26	19.03±0.27	16.45±0.37	147.4±30.4
	ASAM	STEEP	74.45±0.11	81.23±0.11	60.78±0.25	24.07±0.12	21.68±0.23	6.5±0.4
		OBF	**74.73±0.19**	81.24±0.25	60.95±0.28	24.65±0.26	22.40±0.10	30.3±11.6
RESNET -50	SGD	NONE	76.86±0.07	83.28±0.11	65.00±0.14	20.29±0.36	20.53±0.46	230.4±42.7
	SHRINKPERTURB	NONE	76.83±0.03	83.28±0.10	64.62±0.27	20.25±0.31	20.97±0.37	256.6±31.7
	LABELSMOOTH†	NONE	77.18±0.31	83.93±0.21	65.53±0.19	21.25±0.30	21.11±0.01	277.8±6.3
	SAM	STEEP	77.49±0.06	83.78±0.05	65.26±0.21	21.08±0.16	21.18±0.32	170.1±18.9
		OBF	77.67±0.07	84.01±0.03	65.70±0.45	21.63±0.18	22.17±0.26	164.4±25.0
		RANDOM	77.00±0.10	83.27±0.11	64.76±0.15	20.56±0.26	20.94±0.24	220.4±10.9
	ASAM	STEEP	77.30±0.02	84.07±0.03	65.55±0.16	21.71±0.02	21.75±0.15	**33.6±2.99**
		OBF	**78.17±0.07**	**84.66±0.05**	**66.55±0.15**	**23.84±0.12**	**24.21±0.42**	39.1±1.28

in Table B.2 confirm that determining α dynamically leads to the best performance in our setting. Additionally, we also see fixed values of α that exhibit better generalization than steepest ascent perturbations with SAM.

C. Training Details and Hyperparameters

We provide the hyperparameters used in our experiments in Table C.1, with descriptions and details of its columns below:

- MODEL. The model architecture.
- TASK. The task the hyperparameters are for. CIFAR FORGET VS. ACC provides settings for training the pool of models for Section 6.1, and IMAGENET TRAIN and CIFAR FINETUNE provide the training and finetuning hyperparameters for Section 6.2.
- BATCH SIZE. The global batch size used for computing the update step. This is separate from the perturbation batch size m, which is set to 64 in all cases except in Section 6.1.
- EPOCHS. The total training epochs are provided under TRAIN, and WARM provides the number of warmup epochs for a linear learning rate decay schedule with linear warmup.
- OPTIM. The base optimizer used. Here, SGD uses momentum 0.9. Both ADAMW and SGD use decoupled weight decay (Loshchilov & Hutter, 2017).
- LOSS. The loss function used for computing the update gradients. CE is the cross-entropy loss and BCE is the sigmoid cross-entropy (Beyer et al., 2020) loss.
- WEIGHT DECAY. The weight decay strength.
- LEARNING RATE. The maximum (MAX) and minimum (MIN) learning rates for the linear learning rate schedule.
- CLIP GRAD. Gradients are clipped to norm of this value before taking the update step. Gradient clipping is disabled if this value is N/A.

Table B.2. Ablating output bias forgetting (OBF) hyperparameters for SAM with ViT-S/32.

SAM PERTURB	OBF HYPERPARAMS	IMAGENET-V1
STEEP	N/A	72.81
OBF	$\alpha = 0$	73.33
	$\alpha = 10^{-5}$	72.73
	$\alpha = 10^{-4}$	72.49
	$\alpha = 10^{-3}$	73.11
	$\alpha = 10^{-2}$	73.65
	$\alpha = 10^{-1}$	73.77
	$\alpha = 1$	74.15
	$\gamma = 1, \lambda = 10^{-3}$	**74.53**

- HEAD BIAS INIT. The initial value for the bias parameters of the classification head.
- ALGO. The SAM-like algorithm for which these hyperparameters are for. A value of NONE indicates vanilla training.
- PERTURB. The perturbation type, which can be steepest ascent (STEEP), output bias forgetting (OBF), or NONE in case of vanilla training.
- ρ. The perturbation neighborhood size for SAM-like algorithms. When MAX and MIN have different values, ρ is decayed linearly with a linear warmup using the same scheme as the learning rate.
- OBF. The hyperparameters λ and γ for the OBF perturbation. Here, C in the λ values indicates the number of classes, which are 1000, 100, and 10 for ImageNet, CIFAR-100, and CIFAR-10 respectively.
- GSAM. The hyperparameters for GSAM (Zhuang et al., 2022). We were unable to reproduce the officially reported numbers using the authors' hyperparameters. Although we report lower performance for GSAM with ViT, we outperform or match the authors in all other settings including the vanilla and SAM baselines. Additionally, we achieve improved performance with GSAM in some settings by normalizing the perturbing gradient before decomposing it. This normalization is performed if NORM BACKUP is YES.
- ASAM FIXED NORM. When using ASAM (Kwon et al., 2021) with ResNet-50, we outperform the authors' reported numbers by ensuring that the perturbation always has a fixed norm by applying the inverse normalization operator on the gradients before normalizing, and not after. For ViT, we achieve the best baseline performance by applying it after normalization, and not before.

Table C.1. Hyperparameters used for training models for each task, optimization algorithm, and perturbation type.

Model	Task	Batch Size	Epochs		Optim	Loss	Weight Decay	Learning Rate		Clip Grad	Head Bias Init	Algo	Perturb	ρ		OBF		GSAM			ASAM Fixed Norm
			Train	Warm				Max	Min					Max	Min	λ	γ	α	Norm	Backup	
ViT-S /32	CIFAR Forget vs. Acc	512	300	32	AdamW	BCE	1.2	3×10^{-4}		1.0	-10	SAM	Steep	0.2		N/A		N/A			N/A
													OBF			$1/3$	0 1				
	ImageNet Train	4096	300	32	AdamW	BCE	0.3	0.003	3×10^{-5}	1.0	-10	None		N/A							N/A
												SAM	Steep	0.6	0	N/A		N/A			
													OBF	0.6		$1/C$	1				
												GSAM	Steep	0.6	0	N/A		0.4	No		
													OBF	0.6		$1/C$	1		Yes		
												ASAM	Steep	6.0		N/A		N/A			No
													OBF			$1/C$	10^{-12}				
	CIFAR Fine-tune	512	100	5	SGD	BCE	0	0.003	0	1.0	0	None		N/A							N/A
												SAM	Steep	0.05		N/A		N/A			
													OBF			$1/C$	1				
ViT-S /16	ImageNet Train	4096	300	32	AdamW	BCE	0.3	0.003	3×10^{-5}	1.0	-10	None		N/A							N/A
												SAM	Steep	0.6		N/A		N/A			
													OBF			$1/C$	1				
												GSAM	Steep	0.6	0	N/A		1	No		
													OBF	0.6		$1/C$	1	0.4	Yes		
ResNet -50	ImageNet Train	4096	90	16	SGD	CE	0.001	1.6	0.016	N/A	0	None		N/A							N/A
												SAM	Steep	0.04	0.02	N/A		N/A			
													OBF	0.04		$1/C$	10^{-12}				
												GSAM	Steep	0.04	0.02	N/A		0.01	Yes		
													OBF	0.04		$1/C$	10^{-12}		No		
												ASAM	Steep	0.8	0.4	N/A		N/A			Yes
													OBF	0.8		$1/C$	10^{-12}				
	CIFAR Fine-tune	512	100	5	SGD	CE	0	0.01	0	1.0	0	None		N/A							N/A
												SAM	Steep	0.1		N/A		N/A			
													OBF			$1/C$	10^{-12}				
ResNet -101	ImageNet Train	4096	90	16	SGD	CE	0.001	1.6	0.016	N/A	0	None		N/A							N/A
												SAM	Steep	0.04	0.02	N/A		N/A			
													OBF	0.04		$1/C$	10^{-12}				
												GSAM	Steep	0.04	0.02	N/A		0.01	Yes		
													OBF	0.04		$1/C$	10^{-12}		No		