The Pitfalls of Memorization: When Memorization Hinders Generalization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural networks often learn simple explanations that fit the majority of the data while memorizing exceptions that deviate from these explanations. This leads to poor generalization when the learned explanations are spurious. In this work, we formalize *the interplay between memorization and generalization*, showing that spurious correlations, when combined with memorization, can reduce the training loss to zero, leaving no incentive to learn robust, generalizable patterns. To address this issue, we introduce *memorization-aware training* (MAT). MAT leverages the flip side of memorization by using held-out predictions to adjust a model's logits, guiding it towards learning robust patterns that remain invariant from training to test, thereby enhancing generalization under distribution shifts.

1 Introduction

Neural networks can learn simple explanations that work for the majority of their training data (Geirhos et al., 2020; Shah et al., 2020; Dherin et al., 2022). These models might then treat minority examples—those that do not conform to the learned explanation—as exceptions (Zhang et al., 2021). This becomes particularly problematic if the learned explanation is spurious, meaning it does not hold in general or is not representative of the true data distribution (Idrissi et al., 2022; Sagawa et al., 2020; Pezeshki et al., 2021; Puli et al., 2023).

Empirical Risk Minimization (ERM), the standard learning algorithm for neural networks, can exacerbate this issue. ERM enables neural networks to quickly capture spurious correlations and, with sufficient capacity, memorize the remaining examples rather than learning the true patterns that explain the entire dataset. This could be dangerously misleading, as a model that appears to excel in most cases may have actually captured a spurious correlation. Combined with memorization of the remaining minority examples, a neural network can **fully mask its failure** to grasp the true patterns in the data, giving a false sense of reliability and robustness.

Identifying whether a model with nearly perfect accuracy on the training data has learned generalizable patterns or merely relies on a mix of spurious correlations and memorization is critical. The answer lies in the model's performance on held-out data, particularly on minority examples. Metrics such as held-out average accuracy or more fine-grained group accuracies can help us identify a better model. A question that arises is: *How can one use held-out performance signals to proactively guide a model toward learning generalizable patterns?*

Traditionally, held-out performance signals are primarily used for hyperparameter tuning and model selection. However, in this work, we propose a novel approach that leverages these signals more strategically to guide the learning process. But we first need to precisely understand when memorization can hinder generalization. Towards this goal, our paper makes the following contributions:

Submitted to Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

- *Formalizing the interplay between memorization and spurious correlations*: We study how memorization affects generalization in an interpretable setup, revealing that spurious correlations *alone* do not cause poor generalization in neural networks. Instead, it is the combination of spurious correlations with memorization that leads to this problem. Our analysis shows that models trained with ERM tend to rely on spurious features for the majority of the data while memorizing exceptions, achieving zero training loss but failing to generalize on minority examples.
- *Introducing memorization-aware training (MAT)*: MAT is a novel learning algorithm that leverages the flip side of memorization by using held-out predictions to adjust a model's logits during training. This adjustment guides the model toward learning invariant features that generalize better under distribution shifts. Unlike ERM, which relies on the i.i.d. assumption, MAT is built upon an alternative assumption that takes into account the instability of spurious correlations across different data distributions.

The main body of the paper examines our first contribution: the link between memorization and generalization, showing how their interaction impacts a model's ability to learn robust patterns versus spurious correlations through controlled experiments. For more on this interaction in other tasks, such as regression, and how memorization-aware training (MAT), our second contribution, can improve generalization, see the appendix (Sections C and A). For related work, refer to Section D.

2 The Interplay between Memorization and Spurious Correlations in ERM

Problem Setup and Preliminaries. We consider a standard supervised learning setup for a K-class classification problem. The data consists of input-label pairs $(\boldsymbol{x}, y)$, where $\boldsymbol{x}$ is the input vector and $y \in \{1, \ldots, K\}$ is the class label. Let $p(y \mid \boldsymbol{x})$ denote the training data distribution and let a denote any attribute or combination of attributes within $\boldsymbol{x}$ that may or may not be relevant for predicting the target y.

The objective is to learn a model $\hat{p}(y \mid \boldsymbol{x})$ that accurately estimates $p(y \mid \boldsymbol{x})$. Given input $\boldsymbol{x}$, let $f(\boldsymbol{x}; \boldsymbol{w}) \in \mathbb{R}^K$ be the output logits of a model, where each $f_k(\boldsymbol{x}; \boldsymbol{w})$ represents the logit for class k. The estimated conditional probability $\hat{p}(y = k \mid \boldsymbol{x}; \boldsymbol{w})$ is computed using a softmax function with temperature $\tau > 0$:

$$\hat{p}^{\text{tr}}(y = k \mid \boldsymbol{x}; \boldsymbol{w}) = \frac{\exp(f_k(\boldsymbol{x}; \boldsymbol{w})/\tau)}{\sum_{j=1}^{K} \exp(f_j(\boldsymbol{x}; \boldsymbol{w})/\tau)}.$$

In order to generalize well under the i.i.d. assumption that $p(y, \boldsymbol{x})$ is invariant between training and test sets, empirical risk minimization (ERM) seeks to minimize the following regularized cross-entropy loss over a training dataset $D^{\text{tr}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$:

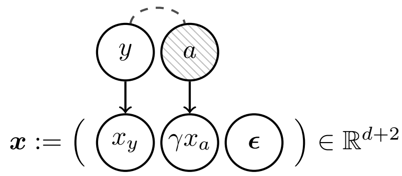
$$\mathcal{L}^{\text{ERM}} = \frac{1}{n} \sum_{i=1}^{n} l(y_i, \hat{p}^{\text{tr}}(y \mid \boldsymbol{x}_i; w)) + \frac{\lambda}{2} ||\boldsymbol{w}||^2,$$

where $l(y_i, \hat{p}^{\text{tr}}(y \mid \boldsymbol{x}_i; w)) = -\sum_{k=1}^{K} \mathbb{I}(y_i = k) \log \hat{p}^{\text{tr}}(y = k \mid \boldsymbol{x}_i; w)$ is the cross-entropy loss, and $\frac{\lambda}{2}||\boldsymbol{w}||^2$ is weight-decay regularization.

In cases where there is a distribution shift between training and test, in the presence of spurious correlations, the i.i.d. assumption breaks down, and when combined with memorization, ERM can result in poor generalization. We study such scenario in the following section.

Memorization Can Exacerbate Spurious Correlations Adapting the frameworks introduced in Sagawa et al. (2020) and Puli et al. (2023), we now study the interplay between memorization and spurious correlations in an interpretable setup.

Setup 2.1 (Spurious correlations and memorization)**.** *Consider a binary classification problem with labels $y \in \{-1, +1\}$ and an unknown spurious attribute $a \in \{-1, +1\}$. Each input $\boldsymbol{x} \in \mathbb{R}^{d+2}$ is given by $\boldsymbol{x} = (x_y, \gamma x_a, \boldsymbol{\epsilon})$, where $x_y \in \mathbb{R}$ is a core feature dependent only on y, $x_a \in \mathbb{R}$ is a spurious feature dependent only on a, and $\boldsymbol{\epsilon} \in \mathbb{R}^d$ are noise features uncorrelated with both y and a. The scalar $\gamma \in \mathbb{R}$ modulates the rate at which the model learns to rely on the spurious feature x_a, effectively acting as a scaling factor that increases the feature's learning rate relative to the core feature x_y. The attribute a is considered spurious because it is correlated with the labels y at training* *but has no correlation with y at test time, potentially leading to poor generalization if the model relies on x_a. Specifically, the data generation process is defined as:*



$$\begin{cases} x_y & \sim \mathcal{N}(y, {\sigma_y}^2) \\ x_a & \sim \mathcal{N}(a, {\sigma_a}^2) \\ \boldsymbol{\epsilon} & \sim \mathcal{N}(0, \sigma_{\boldsymbol{\epsilon}}^2 \boldsymbol{I}) \end{cases} \qquad a = \begin{cases} y & \textit{w.p. } \rho, \\ -y & \textit{w. p. } 1-\rho, \end{cases} \qquad \rho = \begin{cases} \rho^{tr}, & \textit{(train)}, \\ 0.5, & \textit{(test)}. \end{cases}$$

To better understand this setup, one can think of a classification task between cow and camel images. In this example, $\boldsymbol{x}$ represents the pixel data, $y \in \{\text{cow, camel}\}$ are the class labels, and $a \in \{\text{grass, sand}\}$ are the background labels. Here, x_y represents the pixels associated with the animal itself (either cow or camel), x_a represents the pixels associated with the background (grass or sand), and $\boldsymbol{\epsilon}$ represents irrelevant pixels that are specific to each individual example. The key assumption is that the joint distribution of class labels and attribute labels differs between training and test datasets, i.e., $p^{\text{tr}}(a, y) \neq p^{\text{te}}(a, y)$. For example, in the training set, most cows (camels) might appear on grass (sand), while in the test set, cows (camels) appear equally on each background.

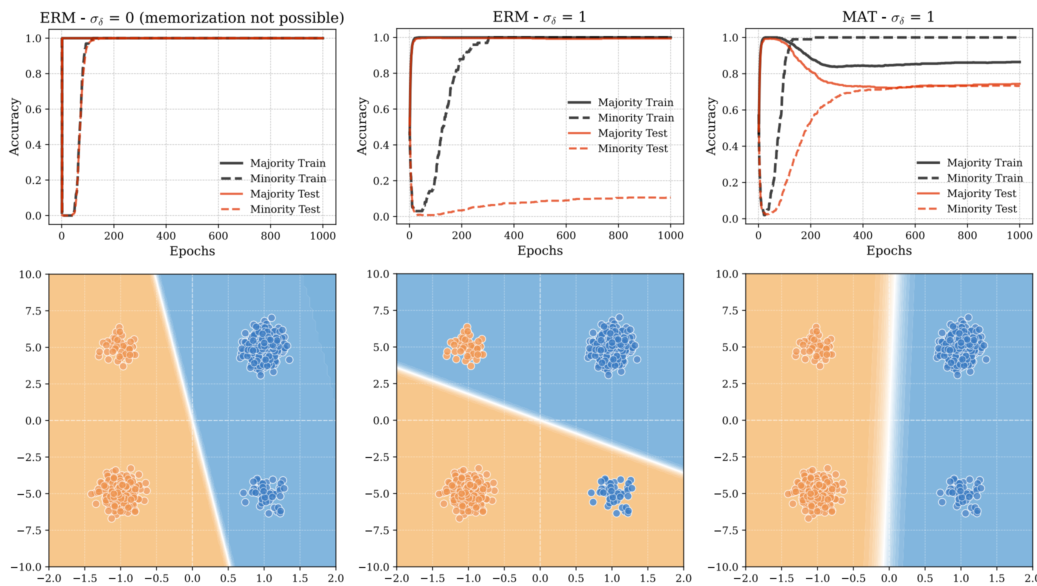


Figure 1: Illustration of two scenarios in the interpretable classification setup involving spurious correlations and memorization. The left panel represents a scenario without input noise ($\sigma_{\boldsymbol{\epsilon}} \to 0$), where memorization is not possible. In this case, the model trained with ERM initially learns the spurious feature x_a serving the majority, but eventually adjusts the decision boundary to the core feature x_y, resulting in good generalization on minority test examples. The middle and right panels depict a scenario with input noise ($\sigma_{\boldsymbol{\epsilon}} \gg 0$), where memorization is possible. Here, the model trained with ERM fails to generalize as it memorizes exceptions using the noise features $\boldsymbol{\epsilon}$ leaving no more incentive for the model to learn the core feature. In contrast, the model trained with MAT (Appendix A) learns the invariant features, and generalizes well even in the presence of noise.

Illustrative Scenarios. We first empirically study a configuration of the above setup where $\gamma = 5$ making the spurious feature easier and faster for the model to learn while being only 90% correlated with the class label, i.e., $\rho^{\text{tr}} = 0.9$. In contrast, the core feature x_y is 100% correlated with y, but due to a smaller norm, it is learned more slowly. Here we consider two cases:

1. ***Noiseless input $\Rightarrow$ Spurious Features but No Memorization $\Rightarrow$ ERM generalizes well.*** Figure 1-(left) presents a case where there are no input noise features ($\sigma_{\boldsymbol{\epsilon}} \to 0$). As training progresses, the model first learns x_a due to its larger norm, resulting in perfect accuracy on the majority examples. Once the model achieve nearly perfect accuracy on the majority examples, it starts to learn the minority examples. At this point, the model must adjust its decision boundary to place more emphasis on the core feature x_y, ultimately achieving perfect generalization on both majority and minority examples.

2. ***Noisy input ⇒ Spurious Features + Memorization ⇒ ERM fails to generalize.***
Figure 1-(middle) presents a similar setup to the former, but this time with input noise features ($\sigma_{\boldsymbol{\epsilon}} \gg 0$). Again, initially, the model learns the spurious feature x_a. However, unlike Case 1, the noise features $\boldsymbol{\epsilon}$ provides the model an opportunity to memorize minority examples directly. As a result, the model achieves zero training loss by memorizing minority examples using the noise dimensions instead of learning to rely on the core feature x_y. Consequently, the model fails to adjust its decision boundary to align with x_y, and does not generalize on held-out minority examples. We argue that most real-world scenarios resemble this case rather than the former case.

These results illustrate that the combination of spurious correlations and memorization creates a 'loophole' for the model. When memorization happens, there is **no more incentives** for the model to learn the true, underlying patterns necessary for robust generalization.

Theoretical Analysis. We now provide a formal analysis to formalize our empirical observations. Complete proofs are provided in Appendix E.

Theorem 2.2 (Memorization Exacerbates Spurious Correlations)**.** *Consider a binary classification problem under the setup described in Setup 2.1, where a linear model $f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{x}^\top \boldsymbol{w}$ is trained using Empirical Risk Minimization (ERM). Let $\widehat{\boldsymbol{w}}_{ERM} = (\widehat{w}_y, \widehat{w}_a, \widehat{\boldsymbol{w}}_{\boldsymbol{\epsilon}}) \in \mathbb{R}^{d+2}$ denote the learned parameters, where $\widehat{w}_y, \widehat{w}_a \in \mathbb{R}$ correspond to the core feature x_y and spurious feature x_a, respectively, and $\widehat{\boldsymbol{w}}_{\boldsymbol{\epsilon}} \in \mathbb{R}^d$ corresponds to the noise features $\boldsymbol{\epsilon}$.*

Assume the following asymptotic conditions hold:

$$\lambda \to 0^+,\ n \to \infty,\ \lambda\sqrt{n} \to \infty,$$

where $\lambda > 0$ is the weight decay regularization parameter, and n is the number of training samples. These conditions ensure that ERM converges to the maximum likelihood estimator. For a training dataset $\mathcal{D}^{tr}$ generated under $\rho^{tr} > 0.5$, where ρ^{tr} is the probability that $a = y$ at training time, the following results hold:

The ERM-trained classifier $\widehat{y}_{ERM}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{x}^\top \widehat{\boldsymbol{w}}_{ERM})$ achieves perfect accuracy on all training examples:

$$p\left(\widehat{y}_{ERM}(\boldsymbol{x}) = y \mid \boldsymbol{x} \in \mathcal{D}^{tr}\right) \to 1.$$

For held-out (test) examples, denote the classifier as $\widehat{y}^{ho}_{ERM}(\boldsymbol{x})$. Then:

1. ***Noiseless Input Case:*** *In the noiseless case where the noise variance $\sigma_\epsilon \to 0^+$, the ERM-trained classifier converges to a classifier that relies solely on the core feature x_y. For a random test point $\boldsymbol{x}$:*

$$p\left(\widehat{y}^{ho}_{ERM}(\boldsymbol{x}) = y\right) \to 1.$$

2. ***Noisy Input Case:*** *Suppose $d \gg \log n$ (where d is the dimension of the noise features) and $\gamma \gg \sigma_{\boldsymbol{\epsilon}}\sqrt{d/m}$, where $m := \rho^{tr} n$ is the number of majority samples in the training set. Then, at test time, the ERM-trained classifier $\widehat{y}_{ERM}(\boldsymbol{x})$ relies pathologically on the spurious feature x_a. For a random test point $\boldsymbol{x}$:*

$$p\left(\widehat{y}^{ho}_{ERM}(\boldsymbol{x}) = a\right) \to 1.$$

The condition $d \gg \log n$ ensures that noise features from different samples are approximately orthogonal, and $\gamma \gg \sigma_{\boldsymbol{\epsilon}}\sqrt{d/m}$ guarantees that the spurious feature x_a is learned faster by gradient descent than other features.

3 Discussion

In our first contribution, we showed that spurious features *alone* do not solely cause poor generalization. Instead, memorization features remove the incentive for the model to learn the true underlying patterns from minority cases. However, to achieve our main goal of learning generalizable patterns, it is crucial to provide the model with feedback on its failures. Held-out performance, which is free of memorization, offers a way to achieve this. To address this, we propose MAT (Memorization-Aware Training), a method that adjusts model logits during training to encourage the learning of generalizable features. Details of MAT are provided in Appendix A. In addition, we introduce and analyze three types of memorization—bad, good, and ugly—highlighting their effects and relevance in different scenarios of benign and malign overfitting. This discussion is elaborated in Appendix C.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a.

Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. Does data interpolation contradict statistical optimality? In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1611–1619. PMLR, 2019b.

Simone Bombari and Marco Mondelli. How spurious features are memorized: Precise analysis for random and ntk features. In *Forty-first International Conference on Machine Learning*, 2024.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*, 2022.

Guillem Collell, Drazen Prelec, and Kaustubh Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *arXiv preprint arXiv:1606.08698*, 2016.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

Nikolay Dagaev, Brett D Roads, Xiaoliang Luo, Daniel N Barry, Kaustubh R Patil, and Bradley C Love. A too-good-to-be-true prior to reduce shortcut reliance. *Pattern recognition letters*, 166: 164–171, 2023.

B. Dherin, M. Munn, M. Rosca, and D. Barrett. Why neural networks find simple solutions: the many regularizers of geometric complexity. *Neural Information Processing Systems*, 2022. doi: 10.48550/arXiv.2209.13083.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Zongbo Han, Zhipeng Liang, Fan Yang, Liu Liu, Lanqing Li, Yatao Bian, Peilin Zhao, Bingzhe Wu, Changqing Zhang, and Jianhua Yao. Umix: Improving importance weighting for subpopulation shift via uncertainty-aware mixup. *Advances in Neural Information Processing Systems*, 35: 37704–37718, 2022.

Reyhane Askari Hemmat, Mohammad Pezeshki, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Feedback-guided data synthesis for imbalanced classification. *arXiv preprint arXiv:2310.00158*, 2023.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *CLeaR*, 2022. URL https://arxiv.org/abs/2110.14503.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020.

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9012–9020, 2019.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. 2017.

Tyler LaBonte, Vidya Muthukumar, and Abhishek Kumar. Towards last-layer retraining for group robustness with fewer annotations. *Advances in Neural Information Processing Systems*, 36, 2024.

Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9572–9581, 2019.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Sheng Liu, Xu Zhang, Nitesh Sekhar, Yue Wu, Prateek Singhal, and Carlos Fernandez-Granda. Avoiding spurious correlations via logit correction. *arXiv preprint arXiv:2212.01433*, 2022.

Pratyush Maini, Saurabh Garg, Zachary Lipton, and J Zico Kolter. Characterizing datapoints via second-split forgetting. *Advances in Neural Information Processing Systems*, 35:30044–30057, 2022.

Pratyush Maini, Michael C Mozer, Hanie Sedghi, Zachary C Lipton, J Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? *arXiv preprint arXiv:2307.09542*, 2023.

Neil Mallinar, James Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. *Advances in Neural Information Processing Systems*, 35:1182–1195, 2022.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. Harmless interpolation of noisy data in regression. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 67–83, 2020.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and A. Madry. Trak: Attributing model behavior at scale. *International Conference on Machine Learning*, 2023. doi: 10.48550/arXiv.2303.14186.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

Mohammad Pezeshki, Diane Bouchacourt, Mark Ibrahim, Nicolas Ballas, Pascal Vincent, and David Lopez-Paz. Discovering environments with xrm. *arXiv preprint arXiv:2309.16748*, 2023.

Aahlad Manas Puli, Lily Zhang, Yoav Wald, and Rajesh Ranganath. Don't blame dataset shift! shortcut learning due to gradients and cross entropy. *Advances in Neural Information Processing Systems*, 36:71874–71910, 2023.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv: 1911.08731*, 2019.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C Lipton, and J Zico Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. On the geometry of generalization and memorization in deep neural networks. *arXiv preprint arXiv:2105.14602*, 2021.

Christos Tsirigotis, Joao Monteiro, Pau Rodriguez, David Vazquez, and Aaron C Courville. Group robust classification without any group information. *Advances in Neural Information Processing Systems*, 36, 2024.

Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Yao-Yuan Yang, Chi-Ning Chou, and Kamalika Chaudhuri. Understanding rare spurious correlations in neural networks. *arXiv preprint arXiv:2202.05189*, 2022.

Zitong Yang, Michal Lukasik, Vaishnavh Nagarajan, Zonglin Li, Ankit Rawat, Manzil Zaheer, Aditya K Menon, and Sanjiv Kumar. Resmem: Learn what you can and memorize the rest. *Advances in Neural Information Processing Systems*, 36, 2024.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, pp. 25407–25437. PMLR, 2022.

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

Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*, 2022.

A Memorization-Aware Training (MAT)

As exemplified in Section 2, the i.i.d. assumption underlying ERM is violated in the presense of *spurious correlations* between the label y and certain attributes a. If a classifier $\widehat{y}(\boldsymbol{x})$ relies on these unstable correlations, it may fail to generalize to test data where $p^{\text{tr}}(y, a) \neq p^{\text{te}}(y, a)$. To address this distribution shift, we propose an alternative assumption.

A.1 An Alternative to the i.i.d. Assumption

We assume that any predictive path involving a is unreliable because $p(y, a)$ changes between training and test. To focus on the stable relationship independent of a, we introduce an *invariant quantity* that remains consistent across both distributions. Specifically, we assume that:

$$\frac{p^{\text{tr}}(y \mid \boldsymbol{x})}{\sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x})} \propto \frac{p^{\text{te}}(y \mid \boldsymbol{x})}{\sum_a p^{\text{te}}(y \mid a) p^{\text{te}}(a \mid \boldsymbol{x})}, \tag{1}$$

where the term $\sum_a p(y \mid a) p(a \mid \boldsymbol{x})$ represents the conditional probability of y given $\boldsymbol{x}$ when passing through a as an intermediate attribute.

Deriving a New Learning Algorithm. To derive a new learning algorithm based on this assumption, we aim to express $p^{\text{tr}}(y \mid \boldsymbol{x})$ in terms of $p^{\text{te}}(y \mid \boldsymbol{x})$. Starting from the assumption in Equation 1, we have:

$$\frac{p^{\text{tr}}(y \mid \boldsymbol{x})}{\sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x})} \propto \frac{p^{\text{te}}(y \mid \boldsymbol{x})}{\sum_a p^{\text{te}}(y \mid a) p^{\text{te}}(a \mid \boldsymbol{x})}, \qquad \text{(by assumption)} \tag{2}$$

$$\Rightarrow \frac{p^{\text{tr}}(y \mid \boldsymbol{x})}{\sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x})} \propto \frac{p^{\text{te}}(y \mid \boldsymbol{x})}{p^{\text{te}}(y)}, \qquad \text{(assuming } y \perp a \text{ in test set)} \tag{3}$$

$$\Rightarrow p^{\text{tr}}(y \mid \boldsymbol{x}) \propto p^{\text{te}}(y \mid \boldsymbol{x}) \sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x}), \qquad \text{(assuming } p^{\text{te}}(y) \sim \mathcal{U}) \tag{4}$$

$$\Rightarrow p^{\text{tr}}(y \mid \boldsymbol{x}) \propto p^{\text{te}}(y \mid \boldsymbol{x})\, p_a^{\text{tr}}(y \mid \boldsymbol{x}), \qquad \text{(change of variable)} \tag{5}$$

$$\Rightarrow p^{\text{tr}}(y \mid \boldsymbol{x}) = \frac{p^{\text{te}}(y \mid \boldsymbol{x})\, p_a^{\text{tr}}(y \mid \boldsymbol{x})}{\sum_{y'} p^{\text{te}}(y' \mid \boldsymbol{x})\, p_a^{\text{tr}}(y' \mid \boldsymbol{x})}, \qquad \text{(normalization so it sums to 1)} \tag{6}$$

where $p_a^{\text{tr}}(y \mid \boldsymbol{x})$ is the correction term accounting for the prediction of label y given $\boldsymbol{x}$ that goes through a:

$$p_a^{\text{tr}}(y \mid \boldsymbol{x}) := \sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x}).$$

Instead of directly estimating $p^{\text{tr}}(y \mid \boldsymbol{x})$, we estimate $p^{\text{te}}(y \mid \boldsymbol{x})$ using a softmax on the logits of a model. Thus, the expression for $\hat{p}^{\text{tr}}(y = k \mid \boldsymbol{x})$ is:

$$\hat{p}^{\text{tr}}(y = k \mid \boldsymbol{x}) = \frac{\exp\left(f_k(\boldsymbol{x}; \boldsymbol{w}) + \log p_a^{\text{tr}}(y = k \mid \boldsymbol{x})\right)}{\sum_{y'} \exp\left(f_{y'}(\boldsymbol{x}; \boldsymbol{w}) + \log p_a^{\text{tr}}(y' \mid \boldsymbol{x})\right)}, \quad \text{(see Lemma E.1)}. \tag{7}$$

Equation 7 proposes adjusting the logits of a model to account for the fact that $p_a(y \mid \boldsymbol{x})$ is unreliable and varies from training to test. This formulation is related to prior work on logit adjustment (Kang et al., 2019; Menon et al., 2020; Ren et al., 2020; Liu et al., 2022; Tsirigotis et al., 2024), but differs in how the adjustment is computed.

A.2 Estimating $p_a(y \mid \mathrm{x})$ Using Held-Out Predictions

In Section 2, we showed that a model trained with ERM under Setup 2.1 tends to rely heavily on spurious attributes when evaluated on held-out data. Specifically, for a given input $\boldsymbol{x}$, the predicted label $\widehat{y}^{\text{ho}}_{\text{ERM}}$ aligns almost exclusively with the spurious attribute a, implying $p(\widehat{y}^{\text{ho}}_{\text{ERM}} = a \mid \boldsymbol{x}) \to 1$. This implies that for a specific $a = a^*$, $p(a^* \mid \boldsymbol{x}) \approx 1$ and $p(a \mid \boldsymbol{x}) \approx 0$ for all other $a \neq a^*$.

Table 1: Average/worst accuracies comparing methods for environment discovery. We specify access to annotations in training data (e^{tr}) and validation data (e^{va}). Symbol $\sim$ denotes inferred group annotations by the method, and symbol † denotes original numbers.

			Waterbirds		CelebA		MNLI		CivilComms	
e^{tr}	e^{va}		Avg	Worst	Avg	Worst	Avg	Worst	Avg	Worst
✗	✓	ERM	83.8	66.4	95.5	55.1	81.6	72.0	84.3	74.0
✓	✓	GroupDRO	90.2	86.5	93.1	88.3	80.6	73.4	84.2	73.8
✗	✓	LC†	-	90.5	-	88.1	-	-	-	70.3
✗	✓	MAT	89.4	88.2	88.0	85.6	TBD	TBD	TBD	TBD
✗	✗	ERM	83.6	66.4	95.3	58.6	81.8	69.1	81.5	64.7
✗	$\sim$	uLA†	91.5	86.1	93.9	86.5	-	-	-	-
$\sim$	$\sim$	XRM+GroupDRO†	89.3	88.1	91.4	89.1	75.8	72.1	84.0	72.2
✗	$\sim$	MAT	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD

This observation simplifies the estimation of $p_a^{\text{tr}}(y \mid \boldsymbol{x})$ as:

$$p_a^{\text{tr}}(y \mid \boldsymbol{x}) = \sum_a p^{\text{tr}}(y \mid a) p^{\text{tr}}(a \mid \boldsymbol{x}) \approx p^{\text{tr}}(y \mid a^*),$$

where $a^* = \arg\max_a p(\hat{y}_{\text{ERM}}^{\text{ho}} = a \mid \boldsymbol{x})$.

To compute $p^{\text{tr}}(y \mid a^*)$, following (Liu et al., 2022), we use the empirical counts from the training data:

$$p^{\text{tr}}(y \mid a^*) = \frac{\text{count}(y, a^*)}{\text{count}(a^*)}.$$

Thus, the held-out predictions of the ERM model provide a straightforward way to estimate $p_a^{\text{tr}}(y \mid \boldsymbol{x})$, allowing us to adjust the model logits accordingly for improved generalization.

Specifically, MAT employs a shared backbone network with three classification heads:

- **Heads A and B**: These heads are trained on two random non-overlapping splits of the training data using ERM. Each head provides held-out predictions for the other head's split, from which we estimate $p^{\text{tr}}(y \mid a^*)$.
- **Head C**: This is the main classifier whose logits are adjusted using Equation 7, based on the held-out predictions from Heads A and B.

During training, all three heads—A, B, and C—are updated simultaneously. Heads A and B optimize only their head parameters. Head C updates its own parameters as well as those of the shared backbone. To further reinforce reliance on spurious correlations, we employ *Label Flipping* strategy (Pezeshki et al., 2023) on Heads A and B. Flipping is done according to held-out probabilities and hence amplifies the biases of the auxiliary classifiers.

B Experiments

We first, conduct experiments to demonstrate the effectiveness of Memorization-Aware Training (MAT) in improving generalization under subpopulation shift. We then provide a detailed analysis of the memorization behaviors of models trained with ERM and MAT.

B.1 Experiments on Subpopulation Shift

We evaluate our approach on four datasets under subpopulation shift. In all experiments, we assume that spurious correlation or environment annotations are not available during training. We consider two settings: (1) group annotations are available in the validation set for model selection, and (2) no annotations are available even in the validation set.

For evaluation, we report two key metrics on the test set: (1) average test accuracy and (2) worst-group test accuracy, the latter being computed using ground-truth annotations.

Table 1 compares the performance of MAT with several baseline methods, including ERM, GroupDRO (Sagawa et al., 2019), and other environment discovery methods like LC (Liu et al., 2022), uLA (Tsirigotis et al., 2024) and XRM+GroupDRO (Pezeshki et al., 2023). These methods vary in their assumptions about access to annotations, both in training and validation for model selection. For instance, ERM does not assume any training group annotations, while GroupDRO has full access to group annotations for both training and validation data.

In the Waterbirds dataset, MAT demonstrates strong performance with 88.2% worst-group accuracy when the groung truth group annotations of the validation set are used for model selection, improving substantially over ERM. Similarly, on the CelebA dataset, MAT achieves competitive results, with a worst-group accuracy of 85.6%. These results suggest that MAT's memorization-aware approach effectively mitigates overfitting to spurious correlations, particularly in challenging worst-group scenarios.

B.2 Analysis of Memorization Scores

To understand the extent of memorization in models trained with ERM, we analyze the distribution of memorization scores across subpopulations. We focus on the Waterbird dataset, which includes two main classes—Waterbird and Landbird—each divided into majority and minority subpopulations based on their background (e.g., Waterbird on water vs. Waterbird on land). This setup allows us to investigate how memorization varies with group size and context.

The memorization score is derived from the influence function, which measures the effect of each training sample on a model's prediction. Formally, the influence of a training sample i on a target sample j under a training algorithm $\mathcal{A}$ is defined as:

$$\text{infl}(\mathcal{A}, \mathcal{D}, i, j) := \hat{p}_{\mathcal{D}}^{(\mathcal{A})}(y_j \mid \boldsymbol{x}_j) - \hat{p}_{\mathcal{D}_{\neg(\boldsymbol{x}_i, y_i)}}^{(\mathcal{A})}(y_j \mid \boldsymbol{x}_j) \tag{8}$$

where $\mathcal{D}$ is the training dataset, $\mathcal{D}_{\neg(\boldsymbol{x}_i, y_i)}$ denotes the dataset with the sample $(\boldsymbol{x}_i, y_i)$ removed. The memorization score is a specific case of this function where the target sample $(\boldsymbol{x}_j, y_j)$ is the same as the training sample. It measures the difference between a model's performance on a training sample when that sample is included in the training set (held-in) versus when it is excluded (held-out).

Calculating self-influence scores with a naive leave-one-out approach is computationally expensive, but recent methods like TRAK (Park et al., 2023) provide an efficient alternative. TRAK approximates the data attribution matrix. The diagonal of this matrix directly gives the self-influence scores.

Figure 2 depicts the distribution of self-influence scores across subpopulations in the Waterbird dataset. We note that minority subpopulations (e.g., Waterbirds on land) show higher self-influence scores compared to their majority counterparts (e.g., Waterbirds on water) for a model trained with ERM. A model trained with MAT, however, shows a similar distribution of self-influence for both the majority and minority examples.

C Memorization: The Good, the Bad, and the Ugly

We showed that the combination of memorization and spurious correlations, rather than spurious correlations alone, could be key reason for poor generalization. Neural networks can exploit spurious features and memorize exceptions to achieve zero training loss, thereby avoiding learning more generalizable patterns. However, an interesting and somewhat controversial question arises: *Is memorization always bad?*

To explore this, we look into a simple regression task to understand different types of memorization and their effects on generalization. We argue that the impact of memorization on generalization can vary depending on the nature of the data and the model's learning dynamics, and we categorize these types of memorization into three distinct forms.

Setup C.1 (Regression with Memorization)**.** *Let $x_y \in \mathbb{R}$ be a scalar feature that determines the true target, $y^* = f(x_y)$. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset consisting of input-target pairs $(\boldsymbol{x}, y)$. Define the input vector as $\boldsymbol{x} = concat(x_y, \boldsymbol{\epsilon}) \in \mathbb{R}^{m+1}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 I) \in \mathbb{R}^m$ represents input noise concatenated with the true feature x_y. The target is defined as $y = y^* + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$ represents additive target noise.*

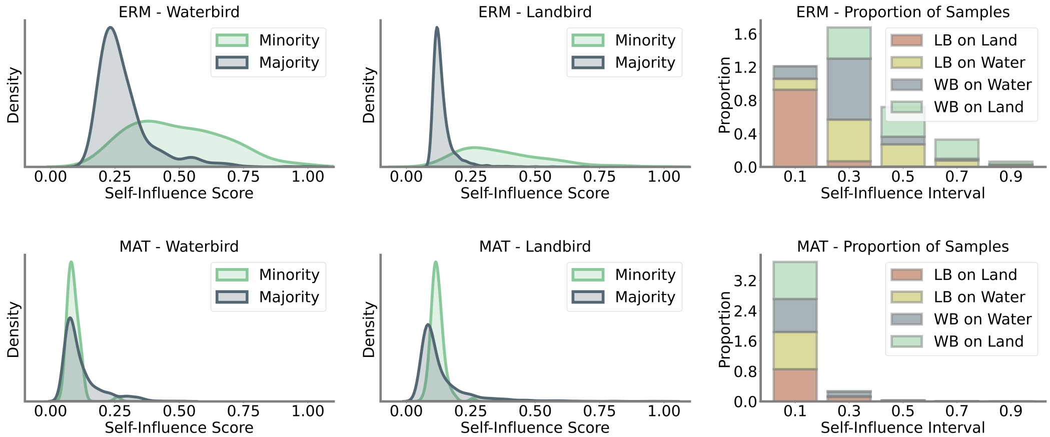


Figure 2: Self-Influence estimation of the Waterbird groups by ERM and MAT. The distribution of self-influence scores is shown for both the majority and minority subpopulations (e.g., Waterbirds on water vs. Waterbirds on land). Models trained with ERM exhibit higher self-influence scores for minority subpopulations, suggesting increased memorization in these groups. In contrast, models trained with MAT show more uniform self-influence distributions across both majority and minority subpopulations. The rightmost plots display the proportion of samples in different self-influence intervals, with MAT producing a more balanced distribution compared to ERM.

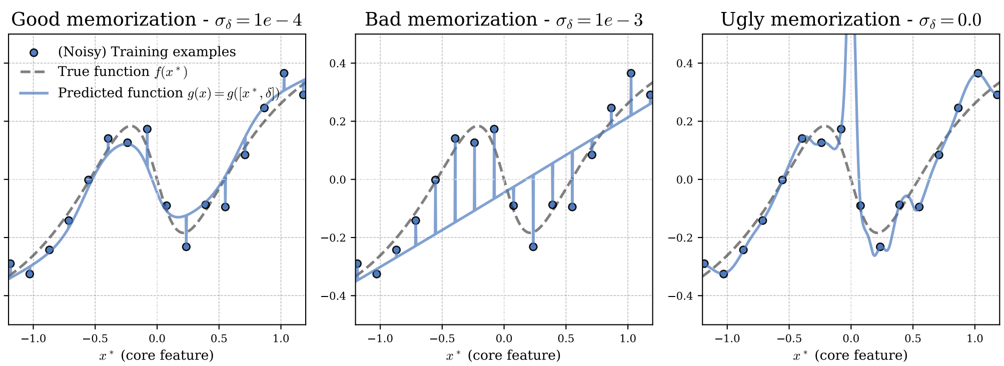


Figure 3: Three types of memorization in regression models trained with different levels of input noise (σ_ϵ). The plots show the ERM-trained model $g(x) = g(x_y, \epsilon)$ (solid blue line) versus the true underlying function $f(x_y)$ (dashed gray line) and the noisy training examples. In all the three, the models are trained until the training loss goes below $1e-6$. **Good memorization (Left, $\sigma_\epsilon = 1e-4$)**: Model learns the true function $f(x_y)$ well but slightly memorizes residual noise in the training data using the input noise ϵ. This type of memorization is benign, as it does not compromise generalization. **Bad memorization (Middle, $\sigma_\epsilon = 1e-3$)**: The model relies more on noise features than learning the true function $f(x_y)$, leading to partial learning of $f(x_y)$ and fitting of noise-dominated input features. This type of memorization impedes learning of generalizable patterns and is considered malign. **Ugly memorization (Right, $\sigma_\epsilon = 0.0$)**: Without input noise, the model overfits the training data, including label noise, resulting in a highly non-linear and complex model that fails to generalize to new data. This type is referred to as catastrophic overfitting.

In this context, x_y can be interpreted as the core feature (e.g., the object in an object classification task), $\boldsymbol{\epsilon}$ as irrelevant random noise, and ϵ as labeling noise or error. Now, consider training linear regression models $\hat{y} = g(\boldsymbol{x})$ on this dataset. Fixing σ_ϵ, we train three models under three different input noise levels: $\sigma_\epsilon \in \{0, 0.01, 0.1\}$. The results, summarized in Figure 3, showcases three types of memorization:

The Good. At an intermediate level of input noise, $\sigma_\epsilon = 1e-4$, the model effectively captures the true underlying function, $f(x_y)$. However, due to the label noise, the model cannot achieve a zero training loss solely by learning $f(x_y)$. As a result, it begins to memorize the residual noise in the training data by using the input noise ϵ. This is evidenced by sharp spikes at each training point, where the model, $g(x)$, precisely predicts the noisy label if given the exact same input as during training. Nevertheless, for a neighboring test example with no input noise, the model's predictions align well with $f(x_y)$, demonstrating good generalization.

This phenomenon is often referred to as "benign overfitting" where a model can perfectly fit (overfit in fact) the training data while relying on noise and unreliable features, yet still generalize well to unseen data (Belkin et al., 2019a; Muthukumar et al., 2020; Bartlett et al., 2020). The key insight is that the overfitting in this case is "'benign" because the model's memorization by relying on noise features does not compromise the underlying structure of the true signal. Instead, the model retains a close approximation to the true function on test data, even though it memorizes specific noise in the training data. This has been shown to occur particularly in over-parameterized neural networks (Belkin et al., 2019b; Nakkiran et al., 2021).

The Bad. At a higher level of input noise, $\sigma_\epsilon = 1e - 3$, the model increasingly rely on the input noise features ϵ rather than fully learning the true underlying function $f(x_y)$. In this case, memorization is more tempting for the model because the noise dominates the input, making it difficult to recover the true signal. As a result, the model $g(x)$ might achieve zero training loss by only partially learning $f(x_y)$ and instead relying heavily on the noise in the inputs to fit the remaining variance in the training data.

This is an instance of bad memorization as it hinders the learning of generalizable patterns. It becomes particularly problematic when the data contains spurious correlations. A model can achieve zero training loss by relying on a combination of spurious correlations and memorization of any errors that are not already satisfied by the spurious correlation. This phenomenon is referred to as "malign overfitting" in Wald et al. (2022), where a model perfectly fits the training data but in a way that compromises its ability to generalize, especially in situations where robustness, fairness, or invariance are critical.

It is important to note that both good and bad memorization stem from the same learning dynamics. ERM, and the SGD that drives it, do not differentiate between the types of correlations or features they are learning. Whether a features contributes to generalization or memorization is only revealed when the model is evaluated on held-out data. If the features learned are generalizable, the model will perform well on new data; if they are not, the model will struggle, revealing its reliance on memorized, non-generalizable patterns.

The Ugly. Finally, consider the case where there is no input noise, $\sigma_\epsilon = 0.0$. In this case, the model may initially capture the true function $f(x_y)$, but due to the presence of label noise, it cannot achieve zero training loss by learning only $f(x_y)$. Unlike the previous cases, the absence of input noise means the model has no additional features to leverage in explaining the residual error. As a result, the model is forced to learn a highly non-linear and complex function of the input $x = x_y$ to fit the noisy labels.

In this situation, memorization is ugly: The model may achieve perfect predictions on the training data, but this comes at the cost of catastrophic overfitting— where the model overfits so severely that it not only memorizes every detail of the training data, including noise, but also loses its ability to generalize to new data (Mallinar et al., 2022).

These examples show that memorization is not always bad; its impact varies with the nature of the data. While MAT mitigates the negative effects of memorization in the presence of spurious correlations, there are cases where memorization can benefit generalization or even be essential (Feldman & Zhang, 2020). Future work could focus on distinguishing these scenarios and exploring the nuanced role of memorization in large language models (LLMs). Recent work (Carlini et al., 2022; Schwarzschild et al., 2024) have highlighted the importance of defining and understanding memorization in LLMs, as it can inform how these models balance between storing training data and learning generalizable patterns.

D Related Work

Detecting Spurious Correlations. Early methods for detecting spurious correlations rely on human annotations (Kim et al., 2019; Sagawa et al., 2019; Li & Vasconcelos, 2019), which are costly and susceptible to bias. Without explicit annotations, detecting spurious correlations requires assumptions. A common assumption is that spurious correlations are learned more quickly or are simpler to learn than core features (Geirhos et al., 2020; Arjovsky et al., 2019; Sagawa et al., 2020). Based on this, methods like Just Train Twice (JTT) (Liu et al., 2021), Environment Inference for Invariant Learning (EIIL) (Creager et al., 2021), Too-Good-To-Be-True Prior (Dagaev et al., 2023), and Correct-n-Contrast (CnC) (Zhang et al., 2022) train models with limited capacity to identify "hard" (minority) examples. Other methods such as Learning from Failure (LfF) (Nam et al., 2020) and Logit Correction (LC) (Liu et al., 2022) use generalized cross-entropy to bias classifiers toward spurious features. Closely related to this work is Cross-Risk Minimization (XRM) (Pezeshki et al., 2023), where uses the held-out mistakes as a signal for the spurious correlations.

Mitigating Spurious Correlations. Reweighting, resampling, and retraining techniques are widely used to enhance minority group performance by adjusting weights or sampling rates (Idrissi et al., 2022; Nagarajan et al., 2020; Ren et al., 2018). Methods like Deep Feature Reweighting (DFR) (Kirichenko et al., 2022) and Selective Last-Layer Finetuning (SELF) (LaBonte et al., 2024) retrain the last layer on balanced or selectively sampled data. GroupDRO (Sagawa et al., 2019) minimizes worst-case group loss, while approaches like LfF and JTT increase loss weights for likely minority examples. Data balancing can also be achieved through data synthesis, feature augmentation, or domain mixing (Hemmat et al., 2023; Yao et al., 2022; Han et al., 2022).

Logit adjustment methods are crucial for robust classification under biased training conditions. Menon et al. (2020) propose a method that corrects model predictions based on class frequencies, building on prior work in post-hoc adjustments (Collell et al., 2016; Kim & Kim, 2020; Kang et al., 2019). Other methods, such as Label-Distribution-Aware Margin (LDAM) loss (Cao et al., 2019), Balanced Softmax (Ren et al., 2020), Logit Correction (LC) (Liu et al., 2022), and Unsupervised Logit Adjustment (uLA) (Tsirigotis et al., 2024), adjust classifier margins to handle class or group imbalance effectively.

Memorization and Spurious Correlations. Research has shown that memorization in neural networks can significantly affect model robustness and generalization. Arpit et al. (2017); Maini et al. (2022); Stephenson et al. (2021); Maini et al. (2023); Krueger et al. (2017) explore memorization's impact on neural networks, examining aspects like loss sensitivity, curvature, and the layer where memorization occurs. Yang et al. (2022) investigate "rare spurious correlations," which are akin to example-specific noise features that models memorize. Bombari & Mondelli (2024) provide a theoretical framework quantifying the memorization of spurious features, differentiating between model stability with respect to individual samples and alignment with spurious patterns. Finally, Yang et al. (2024) propose Residual-Memorization (ResMem), which combines neural networks with k-nearest neighbor-based regression to fit residuals, enhancing test performance across benchmarks.

E Proofs

Lemma E.1. *Let* $p_1(y = j \mid x) = \frac{e^{\phi_j(x)}}{\sum_{i=1}^k e^{\phi_i(x)}}$ *be a softmax over the logits* $\phi_i(x)$ *and define* $p_2(y = j \mid x)$ *such that* $p_2(y = j \mid x) \propto w(j,x)p_1(y = j \mid x)$ *for some weighting function* $w(j,x)$*. Then,*

$$p_2(y = j \mid x) = \frac{e^{\phi_j(x)+\log w(j,x)}}{\sum_{i=1}^k e^{\phi_i(x)+\log w(i,x)}}.$$

Proof. Given $p_2(y = j \mid x) \propto w(j,x)p_1(y = j \mid x)$, we substitute the expression for $p_1(y = j \mid x)$:

$$p_2(y = j \mid x) \propto w(j,x) \cdot \frac{e^{\phi_j(x)}}{\sum_{i=1}^k e^{\phi_i(x)}}.$$

Since $w(j,x) \cdot e^{\phi_j(x)} = e^{\phi_j(x)+\log w(j,x)}$, we have

$$p_2(y = j \mid x) \propto \frac{e^{\phi_j(x)+\log w(j,x)}}{\sum_{i=1}^k e^{\phi_i(x)}}.$$

To ensure that $p_2(y = j \mid x)$ is a valid probability distribution that sums to 1, we need a normalization factor. Define the normalization constant Z as follows:

$$Z = \sum_{j=1}^k e^{\phi_j(x)+\log w(j,x)}.$$

Thus, the properly normalized form of $p_2(y = j \mid x)$ is:

$$p_2(y = j \mid x) = \frac{e^{\phi_j(x)+\log w(j,x)}}{Z}.$$

Substituting back the expression for Z, we get

$$p_2(y = j \mid x) = \frac{e^{\phi_j(x)+\log w(j,x)}}{\sum_{i=1}^k e^{\phi_i(x)+\log w(i,x)}}.$$

This completes the proof. □

F Proof of Theorem 2.2

Setting derivatives of the objective equation **??** zero gives the normal equation

$$\frac{1}{n}\sum_{j=1}^n (s(x_i^\top w) - y_i)x_i + \lambda w = 0.$$

Solving for w then gives

$$\widehat{w} = \sum_{i=1}^n \alpha_i x_i, \text{ with } \alpha_i := \frac{\pi_i - \widehat{\pi}_i}{\eta}, \text{ with } \pi_i := 1_{\{y_i>0\}},\ \widehat{\pi}_i := s(v_i),\ v_i := x_i^\top \widehat{w},\ \eta := n\lambda. \tag{9}$$

Note that the v_i's correspond to logits, while the $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$ should be thought of as the dual representation of the weights vector $\widehat{w}$. Indeed, by construction, one has

$$\widehat{w} = X^\top \alpha, \tag{10}$$

where $X \in \mathbb{R}^{n\times d}$ is the design matrix.

Our mission is then to derive necessary and sufficient conditions for $e > 0$, where

$$e := \gamma \widehat{w}_{spure} - \widehat{w}_{core} = \sum_{i=1}^n (\gamma x_i^{(1)} - x_i^{(2)})\alpha_i = \sum_{i=1}^n (\gamma^2 a_i - s_i)\alpha_i, \tag{11}$$

where $s_j := 2y_j - 1$.

F.1 Fixed-Point Equations

Define subsets $I_\pm, S, L \subseteq [n]$ and integers $m, k \in [n]$ by

$$I_\pm := \{i \in [n] \mid y_i = \pm 1\}, \quad S := \{i \in [n] \mid a_i = \gamma y_i\}, \quad L := \{i \in [n] \mid a_i = -\gamma y_i\}, \tag{12}$$

$$m := \mathbb{E}\,|S| = pn, \quad k := \mathbb{E}\,|L| = (1-p)n. \tag{13}$$

Thus, S (resp. L) corresponds to the sample indices in the majority (resp. the minority) class.

One computes the logits as follows

$$\begin{aligned} v_i = x_i^\top \widehat{w} = \sum_{j=1}^n \alpha_j x_j^\top x_i = \gamma^2 \sum_{j=1}^n \alpha_j z_j a_i + \sum_j \alpha_j s_j s_i + \sum_{j=1}^n \alpha_j \epsilon_j^\top \epsilon_i \\ = \begin{cases} a + \sum_{j=1}^n \alpha_j \epsilon_j^\top \epsilon_i, & \text{if } i \in S \cap I_+, \\ b + \sum_{j=1}^n \alpha_j \epsilon_j^\top \epsilon_i, & \text{if } i \in S \cap I_-, \\ c + \sum_{j=1}^n \alpha_j \epsilon_j^\top \epsilon_i, & \text{if } i \in L \cap I_+, \\ e + \sum_{j=1}^n \alpha_j \epsilon_j^\top \epsilon_i, & \text{if } i \in L \cap I_-, \end{cases} \end{aligned} \tag{14}$$

where $a, b, c, e \in \mathbb{R}$ are defined by

$$\begin{aligned} a &:= \gamma \widehat{w}_{spu} + \widehat{w}_{core} = \sum_{j=1}^n (\gamma^2 z_j + s_j)\alpha_j, \\ b &:= -\gamma \widehat{w}_{spu} - \widehat{w}_{core} = -\sum_{j=1}^n (\gamma^2 z_j + s_j)\alpha_j, \\ e &:= \gamma \widehat{w}_{spu} - \widehat{w}_{core} = \sum_{j=1}^n (\gamma^2 z_j - s_j)\alpha_j, \\ c &:= \widehat{w}_{core} - \gamma \widehat{w}_{spu} = \sum_{j=1}^n (-\gamma^2 z_j + s_j)\alpha_j. \end{aligned} \tag{15}$$

Observe that

$$b = -a, \quad c = -e. \tag{16}$$

The following lemma will be crucial to our proof.

Lemma F.1. *If $a \geq 0$ and $e \geq 0$, then part (B) of Theorem **??** holds. On the other hand, if $a \geq 0$ and $e \leq 0$, then part (C) of Theorem **??** holds.*

Proof. Indeed, for a random test point (x, a, y), we have

$$\begin{aligned} \mathbb{P}(C_{ERM}(x) = C_{spu}(x)) = \mathbb{P}(x_{spu} \times x^\top \widehat{w} \geq 0) = \mathbb{P}(\gamma^2 \widehat{w}_{spu} + y x_{spu} \widehat{w}_{core} + x_{spu} x_\epsilon^\top \widehat{w}_\epsilon \geq 0) \\ = \mathbb{P}(-x_{spu} x_\epsilon^\top \widehat{w}_\epsilon \leq \gamma^2 \widehat{w}_{spu} + y x_{spu} \widehat{w}_{core}) \end{aligned}$$

Now, independent of y, the random variable $-x_{spu} x_\epsilon^\top \widehat{w}$ has distribution $N(0, \sigma_\epsilon^2 \|\widehat{w}_\epsilon\|^2)$. Now, because $\widehat{w} = X^\top \alpha$ by construction, the variance can be written as $\sigma_\epsilon^2 \|\widehat{w}_\epsilon\|^2 = \sigma_\epsilon^2 \|X_\epsilon^\top \alpha\|^2$, which is itself chi-squared random variable which concentrates around its mean $\sigma_\epsilon^4 \|\alpha\|^2$. Furthermore, thanks to equation 19, $\|\alpha\|^2 \leq 1/(n\lambda^2)$, which vanishes in the limit **??**. We deduce that

$$\begin{aligned} \mathbb{P}(C_{ERM}(x) = C_{spu}(x)) &\to \mathbb{P}(\gamma^2 \widehat{w}_{spu} + y x_{spu} \widehat{w}_{core} \geq 0) \\ &= p 1_{\{\gamma \widehat{w}_{spu} + \widehat{w}_{core} \geq 0\}} + (1-p) 1_{\{\gamma \widehat{w}_{spu} - \widehat{w}_{core} \geq 0\}} \\ &= p 1_{\{a \geq 0\}} + (1-p) 1_{\{e \geq 0\}}. \end{aligned}$$

Thus, if $a \geq 0$ and $e \geq 0$, we must have $\mathbb{P}(C_{ERM}(x) = C_{spu}(x)) = p + 1 - p = 1$, that is, part (B) of Theorem 2.2 holds.

On the other hand, one has

$$\begin{aligned} \mathbb{P}(C_{ERM}(x) = C_{core}(x)) = \mathbb{P}(x_{core} \times x^\top \widehat{w} \geq 0) &= \mathbb{P}(\widehat{w}_{core} + y x_{spu} \widehat{w}_{spu} \geq 0) \\ &= p 1_{\{\widehat{w}_{core} + \gamma \widehat{w}_{spu} \geq 0\}} + (1-p) 1_{\{\widehat{w}_{core} - \gamma \widehat{w}_{spu} \geq 0\}} \\ &= q 1_{\{a \geq 0\}} + (1-q) 1_{\{e \leq 0\}}, \end{aligned}$$

where $q := \mathbb{P}(a = y)$. We deduce that if $a \geq 0$ and $e \leq 0$, then $\mathbb{P}(C_{ERM}(x) = C_{core}(x)) = q + 1 - q = 1$, i.e part (C) of Theorem **??** holds. $\square$

F.2 Structure of the Dual Weights

The following result shows that the dual weights $\alpha_1, \ldots, \alpha_n$ cluster into 4 lumps corresponding to the following 4 sets of indices $S \cap I_+$, $S \cap I_-$, $L \cap I_+$, and $L \cap I_-$.

Lemma F.2. *There exist positive constants $A, B, C, E > 0$ such that the following holds with large probability uniformly over all indices $i \in [n]$*

$$\alpha_i \simeq \begin{cases} A, & \textit{if } i \in S \cap I_+, \\ -B, & \textit{if } i \in S \cap I_-, \\ C, & \textit{if } i \in L \cap I_+, \\ -E, & \textit{if } i \in L \cap I_-. \end{cases} \tag{17}$$

Furthermore, the empirical probabilities predicted by ERM are given by

$$\widehat{\pi}_i = y_i - \eta\alpha_i = \begin{cases} 1 - \eta A, & \textit{if } i \in S \cap I_+, \\ \eta B, & \textit{if } i \in S \cap I_-, \\ 1 - \eta C, & \textit{if } i \in L \cap I_+, \\ \eta E, & \textit{if } i \in L \cap I_-. \end{cases} \tag{18}$$

Proof. First observe that

$$\|\alpha\| \leq \frac{1}{\lambda\sqrt{n}}. \tag{19}$$

Indeed, one computes

$$\|\alpha\|^2 = \frac{1}{\eta^2}\sum_{i=1}^n (\pi_i - \widehat{\pi}_i)^2 \leq \frac{1}{\eta^2}\sum_{i=1}^n 1 \leq \frac{n}{\eta^2} = \frac{1}{\lambda^2 n}$$

Next, observe that $\sum_j \alpha_j \epsilon_j^\top \epsilon_i = \alpha_i\|\epsilon_i\|^2 + \sum_{j\neq i}\alpha_j\epsilon_j^\top\epsilon_i \simeq \sigma_\epsilon^2\alpha_i d$. This is because $\alpha_i\|\epsilon_i\|^2$ concentrates around it mean which equals $\sigma_\epsilon^2\alpha_i d$, while w.h.p,

$$\frac{1}{\sigma_\epsilon^2 d}\sup_{i\in[n]}\left|\sum_{j\neq i}\alpha_j\epsilon_j^\top\epsilon_i\right| \lesssim \|\alpha\|\sqrt{\frac{n\log n}{d}} = \sigma_\epsilon\|\alpha\|\sqrt{n}\cdot\sqrt{\frac{\log n}{d}} \leq \sigma_\epsilon\lambda\sqrt{\frac{\log n}{d}} = o(1).$$

The above is because $\lambda \to 0$ and $(\log n)/d \to 0$ by assumption. Henceforth we simply ignore the contributions of the terms $\sum_{j\neq i}\alpha_j\epsilon_j^\top\epsilon_i$. We get the following equations in the limit equation **??**

$$\begin{aligned} v_i &= \begin{cases} \sigma_\epsilon^2\alpha_i d + a, & \text{if } i \in S \cap I_+, \\ \sigma_\epsilon^2\alpha_i d + b, & \text{if } i \in S \cap I_-, \\ \sigma_\epsilon^2\alpha_i d + c, & \text{if } i \in L \cap I_+, \\ \sigma_\epsilon^2\alpha_i d + e, & \text{if } i \in L \cap I_-, \end{cases} \\ \eta\alpha_i = y_i - s(v_i) &= \begin{cases} 1 - s(\sigma_\epsilon^2\alpha_i d + a), & \text{if } i \in S \cap I_+, \\ -s(\sigma_\epsilon^2\alpha_i d + b), & \text{if } i \in S \cap I_-, \\ 1 - s(\sigma_\epsilon^2\alpha_i d + c), & \text{if } i \in L \cap I_+, \\ -s(\sigma_\epsilon^2\alpha_i d + e), & \text{if } i \in L \cap I_-. \end{cases} \end{aligned} \tag{20}$$

Now, because of monotonicity of σ, we can find $A, B, C, E > 0$ such that

$$\alpha_i = \begin{cases} A, & \text{if } i \in S \cap I_+, \\ -B, & \text{if } i \in S \cap I_-, \\ C, & \text{if } i \in L \cap I_+, \\ -E, & \text{if } i \in L \cap I_-, \end{cases}$$

as claimed. □

We will make use of the following lemma.

Lemma F.3. *In the unregularized limit $\lambda \to 0^+$, it holds that $\eta A, \eta B, \eta C, \eta E \in [0, 1/2]$.*

Proof. Indeed, in that unregularized limit, ERM attains zero classification error on the training dataset (first part of Theorem 2.2). This means mean that $\widehat{\pi}_i \geq 1/2$ iff $y_i = 1$, and the result follows. □

F.3 Final Touch (Proof of Theorem ??

We resume the proof of Theorem 2.2. The scalars A, B, C, E must verify

$$\begin{aligned}
\eta A &= 1 - s(\sigma_\epsilon^2 Ad + a) = s(-\sigma_\epsilon^2 Ad - a), \\
\eta B &= s(-\sigma_\epsilon^2 Bd + b) = s(-\sigma_\epsilon^2 Bd - a) = 1 - s(\sigma_\epsilon^2 Bd + a), \\
\eta E &= s(-\sigma_\epsilon^2 Ed + e), \\
\eta C &= 1 - s(\sigma_\epsilon^2 Cd + c) = 1 - s(\sigma_\epsilon^2 Cd - e) = s(-\sigma_\epsilon^2 Cd + e).
\end{aligned} \tag{21}$$

We deduce that

$$A = B, \quad C = E, \tag{22}$$

$$\eta A = s(-\sigma_\epsilon^2 Ad - a), \quad \eta E = s(-\sigma_\epsilon^2 Ed + e). \tag{23}$$

Proof of Part (C). In particular, for the noiseless case where $\sigma_\epsilon \to 0^+$, we have $\eta A \simeq s(-a)$ and $\eta E \simeq s(e)$. We know from Lemma F.3 that $\eta A, \eta E \leq 1/2$. This implies $a \geq 0$ and $e \leq 0$, and thanks to Lemma F.1, we deduce part (C) of Theorem 2.2.

Proof of Part (B). In remains to show that $a \geq 0$ and $e \geq 0$ in the noisy regime $\sigma_\epsilon > 0$, and then conclude via Lemma F.1.

Define $N_1 := |S \cap I_+|$, $N_2 := |S \cap I_-|$, $N_3 := |L \cap I_+|$, $N_4 := |L \cap I_-|$. Note that from the definition of a, b, c, e in equation 15, one has

$$\begin{aligned}
a &= (\gamma^2 + 1)(N_1 + N_2)A - (\gamma^2 - 1)(N_3 + N_4)E, \\
e &= (\gamma^2 - 1)(N_1 + N_2)A - (\gamma^2 + 1)(N_3 + N_4)E, \\
b &= -a, \quad c = -e, \\
\eta A &= s(-\sigma_\epsilon^2 Ad - a), \quad \eta E = s(-\sigma_\epsilon^2 Ed + e), \\
B &= A, \quad C = E.
\end{aligned} \tag{24}$$

We now show that $a \geq 0$ and $e \geq 0$ under the conditions $d \gg \log n$ and $\gamma \gg \sigma_\epsilon \sqrt{d/m}$.

Indeed, under the second condition, the following holds w.h.p

$$\begin{aligned}
\sigma_\epsilon^2 d + (\gamma^2 + 1)(N_1 + N_2) &= ((\gamma^2 + 1)(N_1 + N_2) + \sigma_\epsilon^2 d) \simeq ((\gamma^2 + 1)m + \sigma_\epsilon^2 d) \\
&\simeq (\gamma^2 + 1)m \simeq (\gamma^2 + 1)(N_1 + N_2),
\end{aligned}$$

where we have used the fact that $N_1 + N_2$ concentrates around its mean $m = pn$. We deduce that

$$\begin{aligned}
\sigma_\epsilon^2 Ad + a &= (\sigma_\epsilon^2 d + (\gamma^2 + 1)(N_1 + N_2))A - (\gamma^2 - 1)(N_3 + N_4)E \\
&\simeq (\gamma^2 + 1)(N_1 + N_2)A - (\gamma^2 - 1)(N_3 + N_4)E \\
&\simeq a,
\end{aligned}$$

from which we get.

$$1/2 \geq \eta A \geq s(-\sigma_\epsilon^2 Ad - a) = s(-(1 + o(1))a) = s(-a) + o(1),$$

i.e $s(-a) \geq 1/2 - o(1)$. But this can only happen if $a \geq 0$.

Finally, the conditions $d \gg \log n$ and $\gamma \gg \sigma_\epsilon \sqrt{d/m}$ imply $\gamma \gg K\sigma_\epsilon \sqrt{d/k}$ and $g \geq K \log(3n)$ for any constant $K > 0$. Theorem 1 of Puli et al. (2023) https://arxiv.org/abs/2308.12553 then gives $e = \gamma \widehat{w}_{spu} - \widehat{w}_{core} > 0$, and we are done. □