Like Oil and Water: Group Robustness and Poisoning Defenses Don't Mix

Michael-Andrei Panaitescu-Liess [*1] **Yiğitcan Kaya** [*1] **Tudor Dumitraș** [1]

Abstract

Group robustness has become a major concern in machine learning (ML) as conventional training paradigms were found to produce high error on minority groups. Without explicit group annotations, proposed solutions rely on heuristics that aim to identify and then amplify the minority samples during training. In our work, we first uncover a critical shortcoming of these heuristics: an inability to distinguish legitimate minority samples from poison samples in the training set. By amplifying poison samples as well, group robustness methods inadvertently boost the success rate of an adversary—e.g., from 0% without amplification to over 97% with it. Moreover, scrutinizing recent poisoning defenses both in centralized and federated learning, we observe that they rely on similar heuristics to identify which samples should be eliminated as poisons. In consequence, minority samples are eliminated along with poisons, which damages group robustness—e.g., from 55% without the removal of the minority samples to 41% with it. Finally, as they pursue opposing goals using similar heuristics, our attempts to conciliate group robustness and poisoning defenses come up short. We hope our work highlights how benchmark-driven ML scholarship can obscure the tensions between different metrics, potentially leading to harmful consequences.

1. Introduction

As ML finds adaptations in many fields with diverse priorities, new metrics of success, aside from prediction performance (e.g., accuracy), have come into play. For example, in security-critical applications, robustness to adversarial examples (Chen et al., 2021) or poisoning attacks (Steinhardt et al., 2017); or, in demographically-sensitive applications, fairness (Hashimoto et al., 2018) or group robustness (Liu et al., 2021) are popular metrics the ML community aims to improve. The sheer number of such metrics has led to a paradigm where researchers demonstrate progress on benchmarks often designed with a single metric in mind.

Recent work has exposed previously unknown trade-offs between some of these metrics, e.g., between privacy and fairness (Bagdasaryan et al., 2019) or between robustness and privacy (Song et al., 2019). As applications in practice require balancing various, often mission-critical, metrics, such unknown trade-offs might have catastrophic consequences. This makes the research into studying the intersection of multiple metrics to identify tensions and interactions particularly crucial. With this motivation, in a systematical quantitative study, our work uncovers an inherent tension between approaches designed for two critical metrics: group robustness methods and poisoning defenses.

Group robustness has become a concern as standard training via empirical risk minimization (ERM) has been shown to perform well on an average sample but poorly on samples belonging to under-represented, minority groups (Tatman, 2017). Effective solutions, such as minority upsampling (Byrd & Lipton, 2019), are not always feasible as the explicit group annotations they rely on are often not available due to privacy, e.g., demographic annotations, or financial concerns, e.g., large-scale data sets. To this end, research has proposed heuristics for identifying the minority training samples as a proxy for annotations (Liu et al., 2021). A common observation behind these heuristics is that minority samples are often *difficult to learn* and the model cannot achieve low training error on them. The candidates identified as minority samples are then amplified during training, e.g., through upsampling, which is shown to improve group robustness significantly.

First, we expose a vulnerability: when the training set contains *poison* samples, group robustness heuristics cannot distinguish legitimate minority samples from them. Poison samples are injected by an attacker to teach the model an undesirable behavior, e.g., a backdoor (Saha et al., 2020). As a result, two recent group robustness methods (Sohoni et al., 2020; Liu et al., 2021) end up assisting the attacker by encouraging low error on poison samples along with mi-

[*]Equal contribution [1]Department of Computer Science, University of Maryland, College Park. Correspondence to: Michael-Andrei Panaitescu-Liess <mpanaite@umd.edu>.

2^{nd} AdvML Frontiers workshop at 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

nority samples—attacker achieves $15 - 97\%$ higher success rate due to amplification. Aiming to understand this vulnerability, we observe that poison samples can be as difficult to learn as minority samples, especially in a realistic attack that can inject only a few samples. This suggests that any group robustness method that relies on difficulty-based heuristics might carry a similar vulnerability.

Second, on the other side of the coin, we show that poisoning defenses hurt group robustness when the training set contains legitimate minority samples. In particular, we focus on recent sanitization-based defenses in centralized (Yang et al., 2022) and federated learning (Panda et al., 2022) settings. As it is challenging to detect poisons reliably (Shan et al., 2022), these methods pursue a simpler goal by relying on heuristics to identify outliers. Such heuristics are empirically shown to eliminate poisons without hurting the overall accuracy. However, due to providing distinct learning signals during training, we observe that minority samples are often inadvertently identified (and eliminated) as outliers. This poses a trade-off for the defender: the more poisons are eliminated (lower attack success), the more minority samples will be eliminated as well (lower group robustness). In consequence, the accuracy on the minority group drops by up to 15% after applying an effective defense.

Finally, we make (unsuccessful) attempts to mitigate these tensions by applying poisoning defenses and group robustness methods in tandem. When aiming for low attack success, the defense removes enough minority samples to render group robustness methods ineffective. When aiming for high group robustness, the defense ignores enough poison samples that are still amplified by group robustness, which leads to high attack success. Ultimately, this implies an unintended alignment between defensive and group robustness heuristics. We hope to encourage future work to develop heuristics that conciliate these two critical success metrics.

In summary, we make the following contributions: *(i)* We find that group robustness methods fail to distinguish minority groups from poisons, leading to the risk of amplifying the attacks (Section 3); *(ii)* We show that poisoning defenses fail to distinguish poisons from under-represented groups and lead to the risk of lowering group robustness (Section 4); *(iii)* We demonstrate that combining group robustness with poisoning defenses is challenging and cannot mitigate these tensions (Appendix A.4).

2. Preliminaries

2.1. Related Work

Group Robustness Methods. Group robustness focuses on training models that obtain good performance on each pre-defined group in the data set. Approaches to group robustness fall into two broad categories. The approaches proposed by Sagawa et al. (2019); Byrd & Lipton (2019); Cao et al. (2019) rely on explicit group annotations during training. For example, group distributionally robust optimization (group DRO) (Sagawa et al., 2019) directly minimizes the worst-group error on the training set. The second set of approaches focuses on a more realistic scenario where annotations are not available during training (Liu et al., 2021; Nam et al., 2020; Sohoni et al., 2020; Namkoong & Duchi, 2017). Our work focuses on a promising type of approach within this set that essentially aims to obtain pseudo group labels by deploying various heuristics (Liu et al., 2021).

Poisoning Attacks. In a poisoning attack, the adversary injects a set of malicious samples into the victim's training set. The poisons are designed to induce certain vulnerabilities in the victim model. In dirty-label attacks, the adversary fully controls how the injected sample is crafted and labeled (Gu et al., 2017). In clean-label attacks, however, they can only make minor changes to existing training samples without changing their ground truth label (Suciu et al., 2018). In terms of the attacker's goal, indiscriminate attacks hurt the model's overall accuracy (Koh et al., 2022), targeted attacks cause misclassifications on specific samples (Shafahi et al., 2018), and backdoor attacks teach the model to misclassify any sample that contains a trigger pattern (Gu et al., 2017). In our work, we use a range of poisoning attacks to demonstrate a limitation of group robustness methods in distinguishing legitimate samples from poisons.

Poisoning Defenses. Based on their core assumptions, poisoning defenses can be split into three. *(i)* First category corresponds to the assumption that poisons are difficult to be learned and this includes data sanitization defenses that detect anomalies and outliers in the training set (Steinhardt et al., 2017; Chen et al., 2019; Yang et al., 2022). Note that these are popular strategies against poisoning attacks, because of their low computational overhead and minimal impact on accuracy. In this work, we focus on defenses from this category and show that they lead to the problem we identify as they end up eliminating difficult-to-learn minority samples as poisons. *(ii)* Defenses from the second category assume that poisons are easy to be learned, making them suitable against strong adversaries (Li et al., 2021). *(iii)* Third category corresponds to the assumption that poisons follow a different distribution from clean samples. State-of-the-art defenses (Pan et al., 2023; Qi et al., 2023) use a small base set of clean samples to separate the clean training points from the poisons. We discuss the limitations and challenges of using defenses from the second and third categories in Appendix A.5.

2.2. Formal Setup

In Appendix A.1, we provide additional details about our setup and hyper-parameters.

Datasets. We consider two popular data sets: Waterbirds (Sagawa et al., 2019) and CelebA (Liu et al., 2015). Waterbirds contains $4,795$ training images of "land-bird" and "water-bird" classes, on either land or water backgrounds. CelebA contains of $162,770$ training images of faces, either male or female, and the task proposed in (Sagawa et al., 2019) is to classify them as "blond" or "not blond". In our experiments in Section 4, we randomly sample 10% of CelebA to save computation. We refer to the lowest-represented group (water-birds on land and blond males) as **LRG-1**; and the highest-represented groups (land-birds on land and water-birds on water or blond females, non-blond females and non-blond males) as **HRG**. Additionally, Waterbirds contains a second under-represented group (land-birds on water), which we refer to as **LRG-2**.

Models. Following prior work (Liu et al., 2021), we consider standard ResNet architectures (He et al., 2016), starting from ImageNet-pretrained weights. In our main manuscript, we show results on ResNet-18, then we include additional results on ResNet-50 as well as for the scenario when models are trained from scratch in Appendix A.2.

Group Robustness Methods. We consider two popular techniques from recent work: Just Train Twice (JTT) (Liu et al., 2021) and GEORGE (Sohoni et al., 2020). Both methods have two main phases: (i) *identification* uses heuristics to identify pseudo group annotations for training samples; and (ii) *amplification* uses these annotations to amplify under-represented groups. In (i), JTT trains a heavily regularized model via ERM and identifies the training samples this model misclassifies as belonging to an under-represented group. In (ii), it simply upsamples these samples to train a second model that achieves higher group robustness. On the other hand, in (i), GEORGE trains a standard model via ERM, clusters its latent representations on the training samples, and treats the cluster labels as pseudo group annotations. In (ii), it applies group DRO (Sagawa et al., 2019) on these groups, which ends up amplifying the samples in smaller clusters as under-represented groups.

Poisoning Attacks. We consider (i) a dirty-label backdoor (DLBD) attack that inserts samples containing a trigger with a wrong label; (ii) a sub-population attack (SA) that targets a specific group indiscriminately (Jagielski et al., 2021); and (iii) Gradient Matching (GM) (Geiping et al., 2020), a state-of-the-art clean-label targeted poisoning attack.

Similar to prior work, we consider that 1% of the training set is poisoned for DLBD and GM, and 2% for SA. We also consider other poison percentages in Appendix A.2. For GM, we select the base samples (i.e., the clean training samples the attack modifies into poisons) from LRG-1. For DLBD and SA, we select them from HRG and label them into the same class as LRG-1. For GM, we select 5 target samples from the class that does not contain LRG-1 and launch the attack to force the model to classify them as the class that contains LRG-1. We also consider a setting with 100 target samples and show the results in Appendix A.2.

Poisoning Defenses. In our centralized training experiments, we consider EPIc (Yang et al., 2022), a state-of-the-art technique that iteratively (i) *identifies* the training samples isolated in gradient space as outliers, and (ii) *eliminates* them as potential poisons during training. Additionally, we also consider STRIP (Gao et al., 2019), a run-time backdoor detection defense, and include the results in Appendix A.3. In our federated learning experiments, we consider several robust aggregation mechanisms and poisoning defenses, including coordinate median update (Yin et al., 2018), Trimmed Mean (Yin et al., 2018), and SparseFed (Panda et al., 2022). These techniques aim to sanitize the updates sent by clients and prevent the model from learning outliers.

2.3. Relevant Metrics

We perform each experiment 3 times and report the average and the standard deviation of its results.

Accuracy Metrics. We report two metrics (as percentages) for the prediction performance of a model. First, we report the standard test accuracy (denoted as **ACC**) measured over the entire test set. ACC is dominated by HRG as it does not consider the group labels. To report the group robustness of a model, we measure the Worst-Group Accuracy (denoted as **WGA**). For WGA, we use the ground truth annotations to measure the accuracy on each group (i.e., the percentage of the correctly classified samples from each group). We then report the lowest accuracy among all groups as WGA.

Identification Success Metrics. Group robustness methods ideally identify and amplify only the samples in legitimate minority groups, whereas the other groups (including poisons) remain untouched. To report how close we are to the ideal, we measure the Identification-Factor (denoted as **IDNF**) of each ground truth group individually. IDNF measures what percentage of samples in the group end up being amplified. A small gap between IDNF on poisons and IDNF on LRG indicates a failure scenario.

Attack Success Metrics. In all attacks, we report Attack Success Rate (denoted as **ASR**) as a percentage. The actual measurement of ASR depends on the attack. For DLBD, we measure the percentage of test samples that are correctly classified in the absence of the trigger, but misclassified in its presence; for GM, the percentage of the misclassified target samples, and for SA, the accuracy drop on the target group of the attack over a non-poisoned model.

Defense Success Metrics. An ideal defense only eliminates poisons and reduces the ASR, leaving ACC and WGA intact. For EPIc, we measure the Elimination-Factor (denoted as

ELMF) as the percentage of training samples removed from each individual ground truth group. A large gap between ELMF on LRG and ELMF on HRG indicates disparate impact. For the federated learning defenses (that operate on client updates, not on training samples), we report the drop in WGA and ACC over an undefended model.

3. Limitations of Group Robustness Methods

In this section, we show that heuristics deployed by group robustness methods identify the poisons as under-represented and amplify them. We also study the implications of this limitation regarding the ASR of poisoning attacks.

3.1. Group Robustness Heuristics Identify Poisons

We start by examining which samples are identified by group robustness methods. For JTT, these are the samples misclassified in the first phase, and, for GEORGE, the samples in the smallest cluster. In Table 1, we present the IDNFs on each group (LRG-1, LRG-2, and HRG) for each method. We do not consider LRG-2 for GEORGE because it creates clusters for each class individually and our poisons are only in the same class as LRG-1. In all experiments, the smallest cluster ends up in the same class as LRG-1, which indicates that GEORGE is working as intended.

Across the board, we see that poisons and LRG-1 samples have the highest IDNF—between 93.4% and 100%) Most notably, in many cases, the IDNF on poisons is up to 5% higher than the IDNF on LRG-1. This suggests that group robustness methods are more likely to amplify poisons than legitimate under-represented samples. For Waterbirds, there is an expected gap between the IDNFs on LRG-1 and LRG-2 as LRG-2 contains over $3\times$ more samples than LRG-1, which makes it less difficult to learn. Finally, in all settings, the IDNF on HRG is much lower than the rest—at most 12.5% in the case of GEORGE on Waterbirds.

We also experiment with letting GEORGE automatically adjust the number of clusters for each class by maximizing the silhouette score, as done by Sohoni et al. (2020). This still ends up creating a small cluster that mostly contains the poisons and LRG-1, meaning that it has failed to distinguish between under-represented groups and poisons.

Takeaways. The heuristics used by group robustness methods achieve a high recall in identifying LRG, however, their precision is significantly reduced in the presence of poisons. These heuristics often identify most poisons as LRG—between 98.2% and 100%—which even exceeds the recall on the legitimate LRG. Overall, this supports our claim that current group robustness methods are limited in distinguishing between under-represented groups and poisons.

3.2. Group Robustness Methods Amplify Poisons

After finding that group robustness methods identify poisons as an under-represented group, in this section, we study how this impacts poisoning attacks and their success. To this end, we consider three evaluation settings. In the *standard* case, we apply the group robustness method as-is. In the *ideal* and *worst* cases, we intervene in the method to prevent it from amplifying any poison or to force it to amplify all poisons, respectively. These interventions aim to isolate the impact of amplifying poisons on the attack success rate (ASR). We implement these interventions by manually removing all poisons from (ideal) or placing all poisons into (worst) the set of samples identified by group robustness methods.

We present the results in Table 2, across different attacks, methods, and datasets. We first note that the ASR gap between the worst case and the standard case is often minimal. This highlights that the boost the attacker gains due to the shortcomings of group robustness heuristics is as significant as it can get. The large gap between ($6.7\% - 97.4\%$) the standard and ideal cases shows an opportunity for better heuristics. We believe the larger amplification in the case of CelebA stems from the fact that this is a more complex data set with more variability and samples, which makes it easier for poisoning. Note that most of the models maintain a relatively high WGA (at least 74.1%), which shows that group robustness methods are working as intended, but still worse than the case without any poisoning (86.7% as in Liu et al. (2021)). The only exception to this is the case of SA, where the WGA drops to 60.9%. However, this is expected as the goal of this attack is to hurt the accuracy on a specific group. Finally, the models tend to maintain a fairly high standard ACC, except against SA as it is an indiscriminate attack, which provides a sanity check to our results.

Additionally, in Appendix A.2, we study the impact of the hyper-parameters (early stopping for the identification model and upsampling factor) and consider more settings (different amount of poisoned samples, more targets for GM attack and using a larger model, as well as training the models from scratch). We observe that the results are consistent with our previous findings.

Takeaways. Due to identifying poisons as an under-represented group, group robustness methods end up directly or indirectly amplifying them. We show that this leads to a boost in the success rate of poisoning attacks, and, generally, this boost is almost as high as it could have been.

4. Poisoning Defenses Have Disparate Impact

After establishing how group robustness methods amplify poisons, in this section, we investigate whether poisoning defenses have any undesirable impact on under-represented samples and group robustness.

Table 1. The Identification-Factors (IDNFs) of group robustness methods on different groups of samples in the training set. We highlight the alarming cases where the poison samples are more amplified than the lowest-represented group.

METHOD	DATASET	ATTACK	POISONS	LRG-1	LRG-2	HRG
JTT	WATERBIRDS	DLBD	**98.5** $\pm$ 1.2%	94.6 $\pm$ 1.4%	36.9 $\pm$ 1.8%	5.0 $\pm$ 0.3%
JTT	WATERBIRDS	SA	**98.2** $\pm$ 0.6%	94.6 $\pm$ 1.7%	38.7 $\pm$ 3.1%	4.8 $\pm$ 0.3%
JTT	WATERBIRDS	GM	**100.0** $\pm$ 0.0%	96.2 $\pm$ 6.4%	35.3 $\pm$ 1.9%	5.4 $\pm$ 0.3%
JTT	CELEBA	DLBD	**99.9** $\pm$ 0.0%	96.7 $\pm$ 0.2%	N/A	9.8 $\pm$ 0.5%
GEORGE	WATERBIRDS	DLBD	**98.5** $\pm$ 1.2%	93.4 $\pm$ 1.0%	–	12.5 $\pm$ 1.2%

Table 2. Evaluating the impact of amplification in group robustness methods on worst group accuracy (WGA), attack success rate (ASR), and test accuracy (ACC). We highlight when there is a small gap in ASR between the worst and standard cases as this indicates that a method has given an advantage to the adversary.

METHOD	DATASET	ATTACK	CASE	WGA	ASR	ACC
JTT	WATERBIRDS	DLBD	WORST	76.9 $\pm$ 3.5%	**20.9** $\pm$ 7.9%	86.4 $\pm$ 1.2%
			STANDARD	78.0 $\pm$ 4.1%	**20.4** $\pm$ 5.9%	86.7 $\pm$ 1.2%
			IDEAL	81.4 $\pm$ 0.9%	**0.5** $\pm$ 0.3%	91.0 $\pm$ 0.4%
JTT	WATERBIRDS	SA	WORST	60.9 $\pm$ 4.5%	**24.0** $\pm$ 3.3%	70.9 $\pm$ 2.3%
			STANDARD	61.6 $\pm$ 6.8%	**31.4** $\pm$ 6.0%	66.0 $\pm$ 3.8%
			IDEAL	82.3 $\pm$ 1.5%	**0.8** $\pm$ 1.3%	90.8 $\pm$ 0.3%
JTT	WATERBIRDS	GM	WORST	76.1 $\pm$ 3.2%	**20.0** $\pm$ 0.0%	89.9 $\pm$ 1.0%
			STANDARD	76.1 $\pm$ 3.2%	**20.0** $\pm$ 0.0%	89.9 $\pm$ 1.0%
			IDEAL	75.9 $\pm$ 0.8%	**13.3** $\pm$ 11.5%	91.3 $\pm$ 0.1%
JTT	CELEBA	DLBD	WORST	79.7 $\pm$ 0.9%	**97.7** $\pm$ 0.1%	82.9 $\pm$ 0.8%
			STANDARD	79.3 $\pm$ 0.2%	**97.7** $\pm$ 0.1%	82.6 $\pm$ 0.2%
			IDEAL	79.2 $\pm$ 1.1%	**0.3** $\pm$ 0.0%	83.6 $\pm$ 1.4%
GEORGE	WATERBIRDS	DLBD	WORST	74.1 $\pm$ 1.7%	**16.5** $\pm$ 2.9%	76.4 $\pm$ 3.4%
			STANDARD	77.3 $\pm$ 2.5%	**15.8** $\pm$ 3.9%	79.4 $\pm$ 1.6%
			IDEAL	79.3 $\pm$ 2.1%	**0.4** $\pm$ 0.0%	93.1 $\pm$ 1.4%

4.1. Poisoning Defenses Eliminate Minority Samples

We start our investigation by studying the impact of EPIc on under-represented samples. In Figure 1, we show the Elimination-Factor (ELMF) of EPIc in four different settings. We observe that generally the poisons and LRG-1 samples are eliminated at a similar rate (the two curves are close to each other, across the board). On the other hand, the ELMF on HRG samples is always significantly less. Interestingly, ELMF on LRG-2 samples from Waterbirds is less than the poisons and LRG-1 samples and more than HRG samples. A large (reaching almost 100%) ELMF on poisons shows that EPIc is indeed an effective poisoning defense. However, by eliminating most poisons, EPIc also eliminates most LRG-1 samples as well. This suggests that legitimate under-represented samples are strong outliers from the perspective of EPIc, which demonstrates a disparate impact.

4.2. Poisoning Defenses Reduce Group Robustness

Here, we study the effect of EPIc on group robustness by considering three scenarios. In ideal and worst-case scenarios, we make interventions on EPIc to never eliminate any under-represented sample or to eliminate under-represented samples as early as possible, respectively. In standard EPIc, we make no intervention and apply the defense as-is. Through interventions, we hope to isolate the impact of EPIc on minority samples. For each scenario, we report WGA to measure group robustness.

We present the results in Table 3. First, we observe that in different datasets and attacks, EPIc reduces the WGA by $3.2\% - 14.3\%$. The most damage happens on CelebA data set as, we believe, it is more complex than Waterbirds. Overall, the ASR is low, indicating that EPIc works properly, with one exception for the GM attack, where the ASR is 13.3% for all three scenarios. In all cases, ACC is high, relatively close to ACC reported in prior work (Liu et al., 2021). We see a significant gap in WGA after applying EPIc and applying group robustness methods (in Table 2), almost up to 40%. This is expected as EPIc does not make any attempts to preserve group robustness. In Appendix A.4, we make an effort towards applying poisoning defenses while preserving group robustness.

Additionally, we study the impact that robust aggregation mechanisms and poisoning defenses have on the under-represented groups in Federated Learning, as well as more settings including different amounts of poisoned samples.

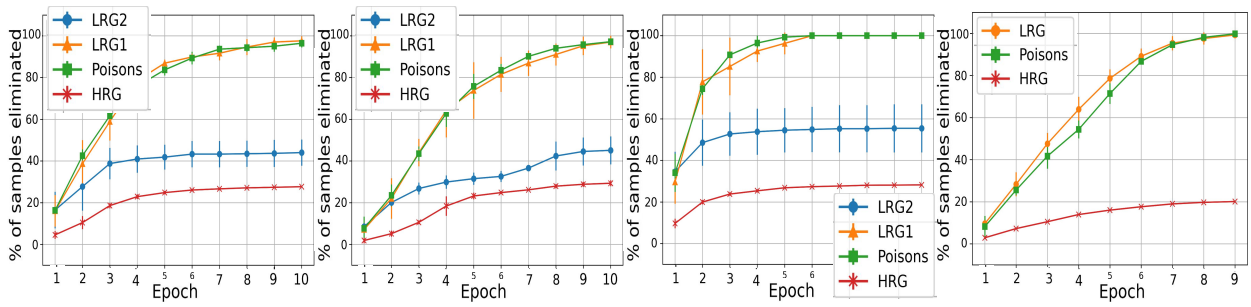

Figure 1. The elimination disparity between the under-represented (LRG) and over-represented (HRG) groups against EPIc. The x-axes show the iterations of EPIc, and y-axes show the Elimination-Factor (ELMF) for each group. From *left* to *right*, the first three plots are for DLBD, SA, GM attacks on Waterbirds and the last one is for DLBD on CelebA.

Table 3. The impact of EPIc on group robustness, measured by the Worst-Group Accuracy (WGA).

DATASET	ATTACK	CASE	WGA	ASR	ACC
WATERBIRDS	DLBD	IDEAL	**59.0** $\pm$ 2.5%	0.1 $\pm$ 0.1%	95.5 $\pm$ 0.1%
		STANDARD	**55.8** $\pm$ 6.0%	0.1 $\pm$ 0.0%	95.0 $\pm$ 0.1%
		WORST	**50.7** $\pm$ 6.9%	0.1 $\pm$ 0.0%	94.3 $\pm$ 0.8%
WATERBIRDS	SA	IDEAL	**59.1** $\pm$ 3.0%	$-$0.3 $\pm$ 1.8%	94.6 $\pm$ 0.4%
		STANDARD	**55.3** $\pm$ 3.2%	0.2 $\pm$ 1.7%	94.2 $\pm$ 0.4%
		WORST	**48.1** $\pm$ 7.3%	$-$0.1 $\pm$ 2.3%	93.9 $\pm$ 0.3%
WATERBIRDS	GM	IDEAL	**50.2** $\pm$ 2.0%	13.3 $\pm$ 11.5%	95.8 $\pm$ 0.3%
		STANDARD	**43.5** $\pm$ 7.8%	13.3 $\pm$ 11.5%	94.6 $\pm$ 0.5%
		WORST	**44.3** $\pm$ 8.3%	13.3 $\pm$ 11.5%	94.6 $\pm$ 0.4%
CELEBA*	DLBD	IDEAL	**54.8** $\pm$ 3.2%	0.3 $\pm$ 0.2%	93.9 $\pm$ 0.0%
		STANDARD	**40.5** $\pm$ 4.5%	0.1 $\pm$ 0.0%	94.0 $\pm$ 0.1%
		WORST	**34.8** $\pm$ 1.1%	0.0 $\pm$ 0.0%	93.7 $\pm$ 0.3%

We show the results in Appendix A.3 and observe that they are consistent with our previous findings.

Takeaways. Poisoning defenses either aim to identify and remove the outliers or make it more difficult for the model to learn poisons (e.g., in Federated Learning). They, however, also end up making minority groups more difficult to learn as well, which hurts the group robustness of the trained model. This shows that, despite the common practice, ACC can be over-optimistic in gauging the impact of a poisoning defense in the presence of under-represented groups.

5. Conclusions

In this work, we demonstrate a significant tension involving two critical metrics studied in the ML community: group robustness and poisoning resilience. The objective of group robustness methods is to amplify minority groups in the training set and create more equitable models. Our findings reveal that the samples injected by poisoning attacks consistently mislead these methods into amplifying them, resulting in an undesirable boost to the adversary. On the other hand, poisoning defenses aim to prevent attacks by removing problematic samples from the training set. However, these defenses remove legitimate under-represented samples as well, hence compromising the model's equity. After making unsuccessful attempts at mitigating these tensions, by combining different methods, we wish to emphasize the pressing need for the ML community to focus on the development of new methods that tackle the inherent challenges posed by poisoning attacks and group robustness.

References

Bagdasaryan, E., Poursaeed, O., and Shmatikov, V. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Byrd, J. and Lipton, Z. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pp. 872–881. PMLR, 2019.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., and Srivastava, B. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety*. CEUR-WS, 2019.

Chen, Y., Wang, S., Qin, Y., Liao, X., Jana, S., and Wagner, D. Learning security classifiers with verified global robustness properties. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 477–494, 2021.

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pp. 1929–1938. PMLR, 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.

Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3104–3122, 2021.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *Machine Learning*, pp. 1–47, 2022.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., and Ma, X. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Lokhande, V. S., Sohn, K., Yoon, J., Udell, M., Lee, C.-Y., and Pfister, T. Towards group robustness in the presence of partial group labels. *arXiv preprint arXiv:2201.03668*, 2022.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33: 20673–20684, 2020.

Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.

Pan, M., Zeng, Y., Lyu, L., Lin, X., and Jia, R. Asset: Robust backdoor data detection across a multiplicity of deep learning paradigms. *arXiv preprint arXiv:2302.11408*, 2023.

Panda, A., Mahloujifar, S., Bhagoji, A. N., Chakraborty, S., and Mittal, P. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pp. 7587–7624. PMLR, 2022.

Qi, X., Xie, T., Wang, J. T., Wu, T., Mahloujifar, S., and Mittal, P. Towards a proactive ml approach for detecting backdoor poison samples, 2023.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Saha, A., Subramanya, A., and Pirsiavash, H. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11957–11965, 2020.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Shan, S., Bhagoji, A. N., Zheng, H., and Zhao, B. Y. Poison forensics: Traceback of data poisoning attacks in neural networks. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 3575–3592, 2022.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Song, L., Shokri, R., and Mittal, P. Membership inference attacks against adversarially robust deep learning models. In *2019 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56. IEEE, 2019.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Suciu, O., Marginean, R., Kaya, Y., Daume III, H., and Dumitras, T. When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1299–1316, 2018.

Tatman, R. Gender and dialect bias in youtube's automatic captions. In *Proceedings of the first ACL workshop on ethics in natural language processing*, pp. 53–59, 2017.

Yang, Y., Liu, T. Y., and Mirzasoleiman, B. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*, pp. 25154–25165. PMLR, 2022.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018.

Zeng, Y., Pan, M., Jahagirdar, H., Jin, M., Lyu, L., and Jia, R. How to sift out a clean data subset in the presence of data poisoning? *arXiv preprint arXiv:2210.06516*, 2022.

A. Supplementary Material

A.1. Additional details on the experimental setup

For the models trained with JTT on Waterbirds, unless differently specified, we use SGD with momentum with a factor of 0.9 as the optimizer, a batch size of 128, learning rate of $1e-5$ and weight decay of 1.0. We stop the identification model after 60 epochs, use an upsampling factor of 100, train the final model up to 100 epochs and choose the model that has the best WGA on the validation dataset, then report the results on the test dataset. In case of CelebA, we use the same hyper-parameters as for Waterbirds, with a few exceptions: weight decay of 0.1, early stopping for the identification model after 1 epoch and we use an upsampling factor of 50. Also, we train the final model up to 5 epochs and consider the same procedure to choose the best model as above. We build our experiments on top of the official code from [1].

For the models trained with GEORGE, we use a similar optimizer, batch size, learning rate, weight decay as in the Waterbirds case from above. For clustering we let the model find the optimal number of clusters (up to 10) for each class based on Silhouette criterion as in Sohoni et al. (2020). We train the final model up to 300 epochs and use the official code from [2]

For DLBD, the trigger is a 25x25 white square placed 1 pixel away from the bottom-right corner of the poisoned samples.

For SA, we consider FeatureMatch and Label Flipping to create poisoned samples as in prior work (Jagielski et al., 2021).

For GM, we use the multi-target version of the attack, an epsilon of 16 and 8 restarts. To build the poisoned samples, we rely on the official code from [3].

For EPIc, in case of Waterbirds, we consider SGD with momentum with a factor of 0.9 as the optimizer, a batch size of 128, learning rate of $1e-2$ and weight decay of $1e-4$. We train the models up to 40 epochs and consider the same selection criterion for the model based on WGA on the validation set as above. In the case of CelebA, we change the learning rate to $1e-3$ and we train the models up to 10 epochs. We build our experiments on top of the official code from [4].

For the federated learning experiments, we built the implementation on top of the code from here [5]. We consider a total of 100 users and 10% of them are chosen at each round. We use SGD with momentum with a factor of 0.9 as the optimizer, a local batch size of 128 and learning rate of $1e-2$. Also, we train each local model for 10 epochs at each round. In case of the non-IID setting, we consider that only 10% of the users have samples from the under-represented groups.

For the federated learning defenses, in the case of Trimmed Mean, we remove the lowest and highest two values for each coordinate in the update and in the case of SparseFed we use a value of $400,000$ for the number of parameters that we keep at each step which is equivalent to keeping less than 5% of the parameters. We consider a momentum factor of $\rho = 0.9$ as in prior work (Panda et al., 2022) or not using momentum ($\rho = 0$) in our experiments.

A.2. More results on Limitations of the Group Robustness Methods.

Impact of the Hyper-parameters. Here, we study how the hyper-parameters of JTT impact our findings from Section 2. We consider two hyper-parameters: *early stopping* for the model in the first phase used to identify LRG; and the *upsampling factor* for the samples identified samples. Our previous experiments use the original hyper-parameters, i.e., early stopping is set to 60 epochs, and upsampling factor is set to $100\times$. However, due to the introduction of poisoning, these hyper-parameters might not be optimal anymore. To this end, we search for new hyper-parameters in a grid (early stopping $\in \{40, 80, 100, 200\}$ and upsampling factors $\in \{20, 50, 150\}$ against DLDB attack on Waterbirds.

We present the results in Tables 4 and 5. We observe that the results are consistent with our previous findings: ASRs are very close between the standard and the worst cases (i.e., high amplification), while both WGA and ACC are still relatively high with the exceptions when early stopping is set to 40 and when the upsampling factor is set to 150. In these cases, the WGAs are only 65.3% and 64.6%, respectively. For the former, we believe that it could be due to how earlier stopping makes identification less selective (lower precision) and ends up upsampling HRG as well. Also, the ASR is higher by more than 15% compared to the other early stopping values. For the latter, we believe that it is because the amplified samples appear too often during training compared to the non-amplified ones. Also, surprisingly, using smaller upsampling factors of

[1] https://github.com/anniesch/jtt
[2] https://github.com/HazyResearch/hidden-stratification
[3] https://github.com/JonasGeiping/poisoning-gradient-matching
[4] https://github.com/YuYang0901/EPIC
[5] https://github.com/AshwinRJ/Federated-Learning-PyTorch

Table 4. Evaluating the impact of amplification in JTT on worst group accuracy (WGA), attack success rate (ASR) and test accuracy (ACC) when considering several early stopping epochs for the identification model.

EARLY STOPPING	CASE	WGA	ASR	ACC
40	WORST	$65.4 \pm 9.4\%$	**46.3** $\pm 30.6\%$	$83.3 \pm 4.4\%$
	STANDARD	$65.3 \pm 9.3\%$	**45.7** $\pm 31.5\%$	$83.2 \pm 4.3\%$
	IDEAL	$76.8 \pm 3.2\%$	**0.4** $\pm 0.1\%$	$91.5 \pm 0.5\%$
60	WORST	$76.9 \pm 3.5\%$	**20.9** $\pm 7.9\%$	$86.4 \pm 1.2\%$
	STANDARD	$78.0 \pm 4.1\%$	**20.4** $\pm 5.9\%$	$86.7 \pm 1.2\%$
	IDEAL	$81.4 \pm 0.9\%$	**0.5** $\pm 0.3\%$	$91.0 \pm 0.4\%$
80	WORST	$80.3 \pm 2.5\%$	**18.7** $\pm 5.9\%$	$87.3 \pm 0.8\%$
	STANDARD	$80.9 \pm 2.8\%$	**17.7** $\pm 5.5\%$	$87.6 \pm 1.0\%$
	IDEAL	$82.2 \pm 2.4\%$	**0.5** $\pm 0.1\%$	$91.3 \pm 0.5\%$
100	WORST	$81.8 \pm 2.0\%$	**19.7** $\pm 8.9\%$	$87.6 \pm 1.2\%$
	STANDARD	$82.0 \pm 2.1\%$	**23.4** $\pm 8.7\%$	$87.2 \pm 0.9\%$
	IDEAL	$83.3 \pm 0.9\%$	**0.9** $\pm 0.1\%$	$91.0 \pm 0.4\%$
200	WORST	$83.5 \pm 2.4\%$	**27.8** $\pm 9.5\%$	$87.9 \pm 1.7\%$
	STANDARD	$83.1 \pm 2.1\%$	**29.5** $\pm 8.5\%$	$87.5 \pm 1.3\%$
	IDEAL	$84.5 \pm 1.5\%$	**0.7** $\pm 0.1\%$	$90.0 \pm 1.7\%$

Table 5. Evaluating the impact of amplification in JTT on worst group accuracy (WGA), attack success rate (ASR) and test accuracy (ACC) when considering several upsampling factors.

UPSAMPLING FACTOR	CASE	WGA	ASR	ACC
20	WORST	$71.9 \pm 4.0\%$	**69.5** $\pm 9.1\%$	$91.4 \pm 0.6\%$
	STANDARD	$72.3 \pm 4.5\%$	**66.6** $\pm 10.0\%$	$91.5 \pm 0.6\%$
	IDEAL	$74.9 \pm 2.3\%$	**1.3** $\pm 0.3\%$	$93.0 \pm 0.3\%$
50	WORST	$79.7 \pm 1.8\%$	**33.1** $\pm 3.5\%$	$91.4 \pm 0.5\%$
	STANDARD	$79.6 \pm 1.9\%$	**28.8** $\pm 6.6\%$	$91.9 \pm 0.6\%$
	IDEAL	$79.5 \pm 1.0\%$	**0.9** $\pm 0.4\%$	$92.6 \pm 0.4\%$
100	WORST	$76.9 \pm 3.5\%$	**20.9** $\pm 7.9\%$	$86.4 \pm 1.2\%$
	STANDARD	$78.0 \pm 4.1\%$	**20.4** $\pm 5.9\%$	$86.7 \pm 1.2\%$
	IDEAL	$81.4 \pm 0.9\%$	**0.5** $\pm 0.3\%$	$91.0 \pm 0.4\%$
150	WORST	$65.7 \pm 10.9\%$	**29.2** $\pm 29.5\%$	$74.1 \pm 3.7\%$
	STANDARD	$64.6 \pm 9.9\%$	**29.4** $\pm 34.4\%$	$73.2 \pm 3.8\%$
	IDEAL	$73.0 \pm 13.2\%$	**0.7** $\pm 0.1\%$	$84.3 \pm 7.1\%$

$50\times$ or $20\times$ instead of $100\times$ makes the attack even more powerful by 8.4% and 46.2% respectively, in terms of ASR.

Additional Settings. For completeness, in this section, we analyze several other scenarios on Waterbirds data set, including using different poisons percentages, different numbers of targets for the GM attack, and different model architectures. In Table 6, we consider JTT and GEORGE with the DLBD attack using several poison percentages in the range $0.4\% - 2\%$. The results are consistent with our previous findings. Additionally, we observe that an attacker who could introduce 2% poisoned samples, generally obtains a higher ASR (of $52.5\% - 59.8\%$). In Table 7, we compare the results in two settings for the GM attack: (i) when the attack has 5 targeted samples (same as our previous experiments) and (ii) when the attack has 100 targeted samples. The results show that, in the case of 100 targets, the amplification is as high as it could be. The only distinction from the case with 5 targets is an overall lower ASR, but this is expected as the attack becomes more difficult as it targets more samples. Finally, we also consider a larger architecture, ResNet-50, instead of ResNet-18, for the DLBD attack and several scenarios, including training the models from scratch. Our results in Table 8 are generally consistent, however, the ASR is significantly higher against ResNet-50. We believe the additional learning capacity of ResNet-50 over ResNet-18 facilitates the attack. Also, we observe that training the models from scratch and tuning them for high group robustness (WGA) damages the overall accuracy (ACC) significantly. This challenge explains the prior practice of using pre-trained models as they can alleviate this trade-off to some extent. To conclude, our extensive experiments show that the limitation of group robustness methods is consistent across different settings, suggesting that this might be an inherent vulnerability.

Table 6. Evaluating the impact of amplification in group robustness methods on worst group accuracy (WGA), attack success rate (ASR) and test accuracy (ACC) when considering several poison percentages for the DLBD attack.

POISONS	METHOD	CASE	WGA	ASR	ACC
0.4%	JTT	WORST	$78.7 \pm 2.0\%$	**4.8** $\pm 0.7\%$	$89.3 \pm 1.2\%$
		STANDARD	$78.6 \pm 2.0\%$	**3.5** $\pm 1.7\%$	$89.1 \pm 1.0\%$
		IDEAL	$81.1 \pm 1.9\%$	**0.3** $\pm 0.1\%$	$91.4 \pm 0.3\%$
0.6%	JTT	WORST	$80.6 \pm 2.7\%$	**8.7** $\pm 1.4\%$	$89.4 \pm 1.2\%$
		STANDARD	$79.3 \pm 1.7\%$	**9.6** $\pm 1.8\%$	$89.0 \pm 0.8\%$
		IDEAL	$81.2 \pm 2.0\%$	**0.5** $\pm 0.1\%$	$91.6 \pm 0.5\%$
0.8%	JTT	WORST	$78.5 \pm 3.4\%$	**15.8** $\pm 4.4\%$	$87.8 \pm 1.2\%$
		STANDARD	$78.0 \pm 3.0\%$	**16.8** $\pm 5.4\%$	$87.7 \pm 1.2\%$
		IDEAL	$81.2 \pm 1.5\%$	**0.5** $\pm 0.1\%$	$91.0 \pm 0.6\%$
1%	JTT	WORST	$76.9 \pm 3.5\%$	**20.9** $\pm 7.9\%$	$86.4 \pm 1.2\%$
		STANDARD	$78.0 \pm 4.1\%$	**20.4** $\pm 5.9\%$	$86.7 \pm 1.2\%$
		IDEAL	$81.4 \pm 0.9\%$	**0.5** $\pm 0.3\%$	$91.0 \pm 0.4\%$
2%	JTT	WORST	$64.7 \pm 7.4\%$	**56.6** $\pm 41.9\%$	$69.8 \pm 5.6\%$
		STANDARD	$62.0 \pm 9.8\%$	**59.8** $\pm 45.9\%$	$66.0 \pm 9.6\%$
		IDEAL	$82.5 \pm 1.6\%$	**0.6** $\pm 0.3\%$	$91.1 \pm 0.2\%$
1%	GEORGE	WORST	$74.1 \pm 1.7\%$	**16.5** $\pm 2.9\%$	$76.4 \pm 3.4\%$
		STANDARD	$77.3 \pm 2.5\%$	**15.8** $\pm 3.9\%$	$79.4 \pm 1.6\%$
		IDEAL	$79.3 \pm 2.1\%$	**0.4** $\pm 0.0\%$	$93.1 \pm 1.4\%$
2%	GEORGE	WORST	$74.3 \pm 5.4\%$	**56.1** $\pm 13.6\%$	$78.0 \pm 9.0\%$
		STANDARD	$74.5 \pm 5.2\%$	**52.5** $\pm 7.1\%$	$77.6 \pm 8.5\%$
		IDEAL	$80.2 \pm 0.4\%$	**0.7** $\pm 0.1\%$	$92.4 \pm 0.9\%$

Table 7. Evaluating the impact of amplification in JTT on worst group accuracy (WGA), attack success rate (ASR) and test accuracy (ACC) when considering different targets for the GM attack.

TARGETS	CASE	WGA	ASR	ACC
5	WORST	$76.1 \pm 3.2\%$	**20.0** $\pm 0.0\%$	$89.9 \pm 1.0\%$
5	STANDARD	$76.1 \pm 3.2\%$	**20.0** $\pm 0.0\%$	$89.9 \pm 1.0\%$
5	IDEAL	$75.9 \pm 0.8\%$	**13.3** $\pm 11.5\%$	$91.3 \pm 0.1\%$
100	WORST	$75.4 \pm 2.3\%$	**8.6** $\pm 0.5\%$	$88.6 \pm 0.9\%$
100	STANDARD	$75.8 \pm 2.2\%$	**8.6** $\pm 0.5\%$	$88.8 \pm 1.0\%$
100	IDEAL	$75.6 \pm 0.5\%$	**5.6** $\pm 0.5\%$	$91.1 \pm 0.6\%$

Table 8. Evaluating the impact of amplification in JTT on worst group accuracy (WGA), attack success rate (ASR) and test accuracy (ACC) when considering a different architecture (ResNet-50). Note that ES denotes early stopping for the identification model.

SETTING	ES	CASE	WGA	ASR	ACC
PRE-TRAINED	60	WORST	$72.4 \pm 2.1\%$	**70.6** $\pm 13.8\%$	$76.2 \pm 1.3\%$
		STANDARD	$72.4 \pm 2.1\%$	**70.6** $\pm 13.8\%$	$76.2 \pm 1.3\%$
		IDEAL	$85.1 \pm 1.4\%$	**0.9** $\pm 0.3\%$	$90.2 \pm 1.0\%$
PRE-TRAINED	200	WORST	$83.4 \pm 2.1\%$	**58.0** $\pm 23.7\%$	$86.5 \pm 2.5\%$
		STANDARD	$83.5 \pm 1.9\%$	**56.5** $\pm 21.2\%$	$86.7 \pm 2.1\%$
		IDEAL	$84.7 \pm 1.5\%$	**0.5** $\pm 0.5\%$	$90.8 \pm 2.6\%$
FROM SCRATCH	200	WORST	$39.5 \pm 7.8\%$	**53.7** $\pm 9.7\%$	$59.0 \pm 9.8\%$
		STANDARD	$41.0 \pm 8.5\%$	**34.5** $\pm 26.4\%$	$64.0 \pm 8.3\%$
		IDEAL	$37.8 \pm 6.9\%$	**5.8** $\pm 4.6\%$	$61.0 \pm 12.9\%$

Table 9. The impact of poisoning defenses in federated learning on group robustness.

METHOD	IID?	WGA DROP	ACC DROP
MEDIAN	YES	$23.7 \pm 7.3\%$	$-0.2 \pm 0.3\%$
TRIMMED MEAN	YES	$45.7 \pm 8.8\%$	$10.6 \pm 3.5\%$
SPARSEFED ($\rho = 0$)	YES	$45.4 \pm 27.7\%$	$11.0 \pm 17.7\%$
SPARSEFED ($\rho = 0.9$)	YES	$42.7 \pm 24.3\%$	$19.0 \pm 14.6\%$
SPARSEFED ($\rho = 0.9$)	NO	$55.1 \pm 21.6\%$	$12.3 \pm 16.3\%$

Table 10. The impact of EPIc on group robustness, when considering the DLBD attack and several poison percentages.

POISONS	CASE	WGA	ASR	ACC
0.5%	IDEAL	**61.2** $\pm 2.1\%$	$0.1 \pm 0.0\%$	$95.4 \pm 0.5\%$
	STANDARD	**57.6** $\pm 4.9\%$	$0.1 \pm 0.0\%$	$95.0 \pm 0.6\%$
	WORST	**52.0** $\pm 8.1\%$	$0.1 \pm 0.0\%$	$94.4 \pm 0.5\%$
1%	IDEAL	**59.0** $\pm 2.5\%$	$0.1 \pm 0.1\%$	$95.5 \pm 0.1\%$
	STANDARD	**55.8** $\pm 6.0\%$	$0.1 \pm 0.0\%$	$95.0 \pm 0.1\%$
	WORST	**50.7** $\pm 6.9\%$	$0.1 \pm 0.0\%$	$94.3 \pm 0.8\%$
2%	IDEAL	**57.3** $\pm 2.5\%$	$0.3 \pm 0.3\%$	$94.5 \pm 0.6\%$
	STANDARD	**57.0** $\pm 0.4\%$	$0.5 \pm 0.4\%$	$94.5 \pm 0.4\%$
	WORST	**48.6** $\pm 7.6\%$	$0.2 \pm 0.0\%$	$93.8 \pm 0.1\%$

A.3. More results on the Limitations of the Poisoning Defenses.

Federated Learning Scenarios. To demonstrate that the limitation of EPIc we identified (in Section 4) is consistent in other defenses, we also run experiments on federated learning, where robust aggregation mechanisms are widely studied. In this scenario, the defense does not sanitize the training set but sanitizes the updates sent by each client to prevent poisoning. We consider FedAvg (McMahan et al., 2017) as an un-defended baseline and study the drop in WGA and ACC relative to it. We included more details about the experimental setup in Appendix A.1. In Table 9, we observe that the defenses we consider cause significantly more drop in WGA than in ACC, over the baseline. This exposes that all these methods, while attempting to fight against poisoning, end up having a disparate impact on the model's accuracy on under-represented groups.

More Settings. In Table 10, we study the impact of EPIc when there are 0.5% or 2% poisons for the DLBD attack, instead of 1%. We observe that EPIc drops the WGA by $0.3\% - 3.6\%$, while maintaining a low ASR and high ACC. Also, aligning with our previous results, we observe that the overall WGA is low compared to the values obtained when considering a group robustness method. Overall, these results are consistent with our main claims.

Run-time defenses. Additionally, we ran experiments using STRIP (Gao et al., 2019), a run-time backdoor detection mechanism. The main assumption of this method is that a backdoored model's outputs will have lower entropy on perturbed backdoored samples compared to perturbed clean samples. In Table 11, we show the means and standard deviations for the output entropy values on different types of samples when they are perturbed (we use the first model from Table 2 in the standard case). The results suggest that STRIP cannot accurately separate clean samples from the poisons by thresholding the entropy values. We believe the strong regularization needed for the models to achieve group robustness (Liu et al., 2021; Sagawa et al., 2019) contributes to the limitation of such defenses as the model will not produce very confident outputs, leading to generally high entropy values on all types of inputs.

Table 11. Entropy of a model's output on perturbed samples from Waterbirds under DLBD attack.

	HRG	LRG-1	LRG-2	POISONS
OUTPUT ENTROPY	0.93 ± 0.03	0.92 ± 0.04	0.91 ± 0.03	0.93 ± 0.03

Table 12. Applying EPIc and JTT together to combine poison resilience with group robustness.

EPIc	WGA	ASR	ACC
No	$78.0 \pm 4.1\%$	$20.4 \pm 5.9\%$	$86.7 \pm 1.2\%$
stop = 3	$76.5 \pm 3.8\%$	$14.8 \pm 4.0\%$	$91.9 \pm 0.3\%$
stop = 5	$73.4 \pm 7.4\%$	$6.7 \pm 1.7\%$	$82.8 \pm 7.3\%$
stop = 7	$42.3 \pm 6.6\%$	$0.9 \pm 0.6\%$	$93.4 \pm 0.2\%$
Ideal	$81.4 \pm 0.9\%$	$0.5 \pm 0.3\%$	$91.0 \pm 0.4\%$

A.4. Combining Group Robustness Methods and Poisoning Defenses

In this section, we study the feasibility of achieving both high group robustness and poisoning resilience by combining the current state-of-the-art methods.

We first apply EPIc to identify potential poisons in the training set. Then, when we apply JTT, we intervene so that it does not amplify the potential poisons found in its first phase (i.e., we remove them from JTT's upsampling set). We have considered the following two baselines in our pipeline: an ideal EPIc that identifies only the poisons and not using EPIc that we would want to improve upon. Because EPIc removes samples iteratively, we considered three stopping epochs for the removal process, so that we have control over how many samples EPIc identifies as poisons. As shown in Figure 1, stopping EPIc sooner leads to a lower percentage of samples from LRG that are removed, but also a lower amount of poisons. Whereas, stopping EPIc later increases both of these rates.

For the rest of this section, we consider Waterbirds dataset and DLBD attack. We present further details on the experimental setup, including the hyper-parameters used in Appendix A.1.

First of all, as shown in Table 12, not using EPIc at all results in a WGA relatively close to the WGA that could be obtained if none of the poisons were amplified (Ideal EPIc). However, the ASR is still high if we do not use EPIc to identify potential poisons. We evaluate three possible stopping epochs for EPIc and measure the effects on WGA and ASR as a function of EPIC's stopping epoch. With a higher stopping epoch (i.e., more samples are identified as poisons), the ASR decreases, however, the WGA also decreases. For example, to mitigate the attack and obtain below 1% ASR, we need to sacrifice over 35% WGA—significant damage to group robustness. Also, in ideal EPIc (only poisons are eliminated), we could obtain both high WGA (over 80%) and low ASR (lower than 1%). Moreover, note that for all the models, the ACC stays relatively high (over 80%), though, lower than the settings without any poisoning (e.g., 93.3% in (Liu et al., 2021)). This further shows how ACC can be misleading to judge the side effects of a poisoning defense.

In conclusion, we have attempted to combine EPIc and JTT, in hopes of achieving both high poisoning resilience and group robustness, but this task is not trivial. Both legitimate under-represented samples and poison samples in realistic attacks can be difficult-to-learn and without making specific assumptions, (e.g., poisons contain detectable artifacts), it might be difficult to distinguish them. Using EPIc (which makes no such assumptions) to identify potential poisons and use that information as an intervention into JTT is not enough to mitigate the trade-off between WGA and ASR.

A.5. Discussion and Future Work

In this work, we have focused only on defenses from the first category in Section 2.1 (that consider poisons as difficult to learn). We have exposed their vulnerability and the potentially harmful consequences of these defenses. We believe that defenses from the second category (that consider poisons as easy to learn), would not lead to these problems. However, it is important to note that, poisons will not be easy to learn in realistic attacks where adversaries can only inject a limited number of poisons, violating the assumption. As a result, defenses from this category would be ineffective against such attacks and, therefore, group robustness methods would still inadvertently offer a needed boost to the weaker adversary. Finally, for the third category of defenses (poisons are different from clean samples), the state-of-the-art defenses (Pan et al., 2023; Qi et al., 2023) use a small base set (10-1000 samples) to model the distribution of clean samples and identify the training points most distinct from this distribution as poisons. These base sets are often assumed to follow the same distribution as the clean training set, which makes them unlikely to contain sufficiently many minority samples. We hypothesize that this will cause such defenses to still eliminate clean minority samples as poisons and hurt the WGA. However, it might be possible to avoid this problem by providing such defenses with carefully curated base sets that are balanced (in terms

of groups) and free from poisons. In this case, instead of mistakenly penalizing difficult clean samples as poisons, they can isolate the real poisons from a more inclusive clean distribution captured by the base set. However, research suggests that making a base set poison-free (Zeng et al., 2022) or collecting enough labeled minority samples that capture their distribution properly (Lokhande et al., 2022) might be challenging in practice. We believe addressing these challenges is a promising avenue for future work to conciliate between group robustness and poisoning resilience.