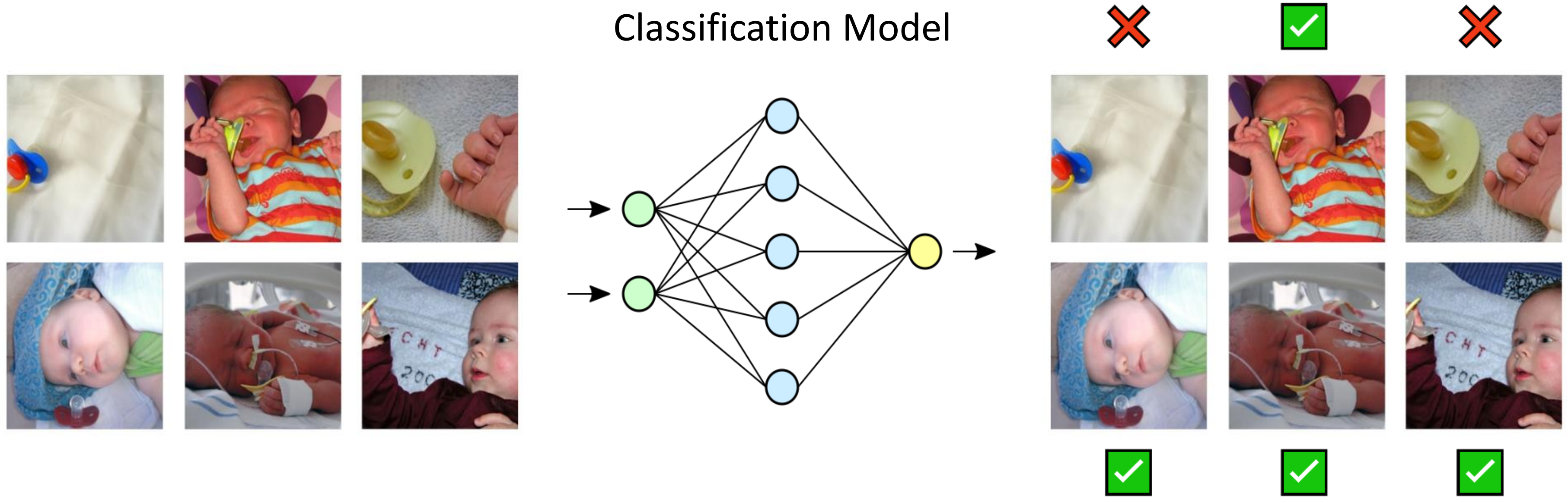
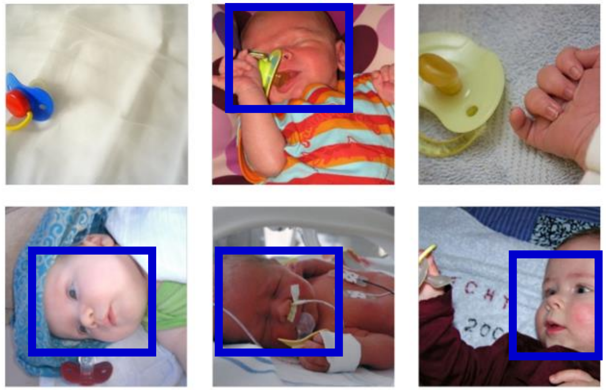
The Silent Majority: Demystifying Memorization Effect in the Presence of Spurious Correlations

Chenyu You*, Haocheng Dai*, Yifei Min*, Jasjeet Sekhon, Sarang Joshi, James Duncan. (*equal contribution)



The baby pacifier class in ImageNet

Classification Outcome



The baby pacifier class in ImageNet is <u>spuriously correlated</u> with the presence of babies.

When trying to identify hair color

	Non-blond Woman	Non-blond Man	Blond Woman	Blond Man
CelebA				
Training #	71629 (44%)	66874 (41%)	22880 (14%)	1387 (1%)
Validation #	8535	8276	2874	182
Accuracy	97.78%	99.86%	85.88%	36.99%

Liu et al. 2015

The blond hair class in CelebA is <u>spuriously correlated</u> with

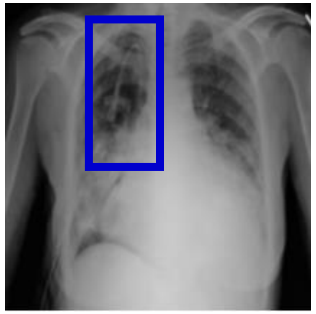
When trying to identify bird type

	Landbird on Land	Landbird on Water	Waterbird on Land	Waterbird on Water
Waterbird				
Training #	3498 (73%)	184 (4%)	56 (1%)	1057 (22%)
Validation #	467	466	133	133
Accuracy	99.79%	77.68%	38.35%	92.48%

Sagawa et al. 2019

The bird class in Waterbird is spuriously correlated with background.

When trying to identify pneumothorax

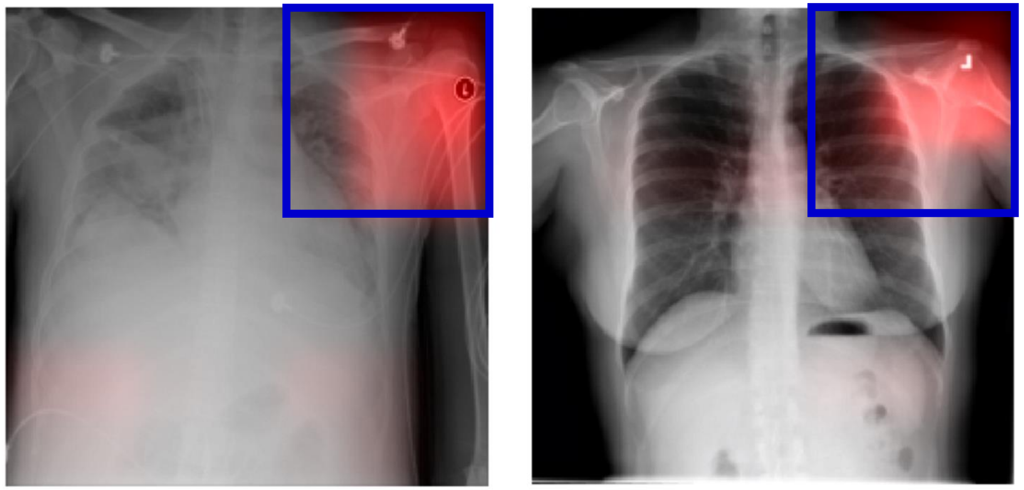
	Pneumothorax-free without chest drain	Pneumothorax without chest drain	Pneumothorax with chest drain	
CXR-14				Oakden-Rayner et al. 2019
Training #	?	?	?	
Validation #	10714 (96%)	204 (2%)	300 (2%)	

The pneumothorax-free class in CXR-14 is spuriously correlated with no chest

When trying to identify pneumonia

CNN has learned to identifying pneumonia by detecting a metal token that radiology technicians place on the patient.

Zech et al. 2018



Even the most advanced models trained with ERM* can develop
systematic biases from these spurious attributes in the data.

**Empirical Risk Minimization (ERM) represents conventional training often focus on minimizing average training error, without any procedures for improving worst-group accuracies.*

How previous work resolve this?

Without knowing group label

Just Train Twice (JTT)
Liu et al., 2021

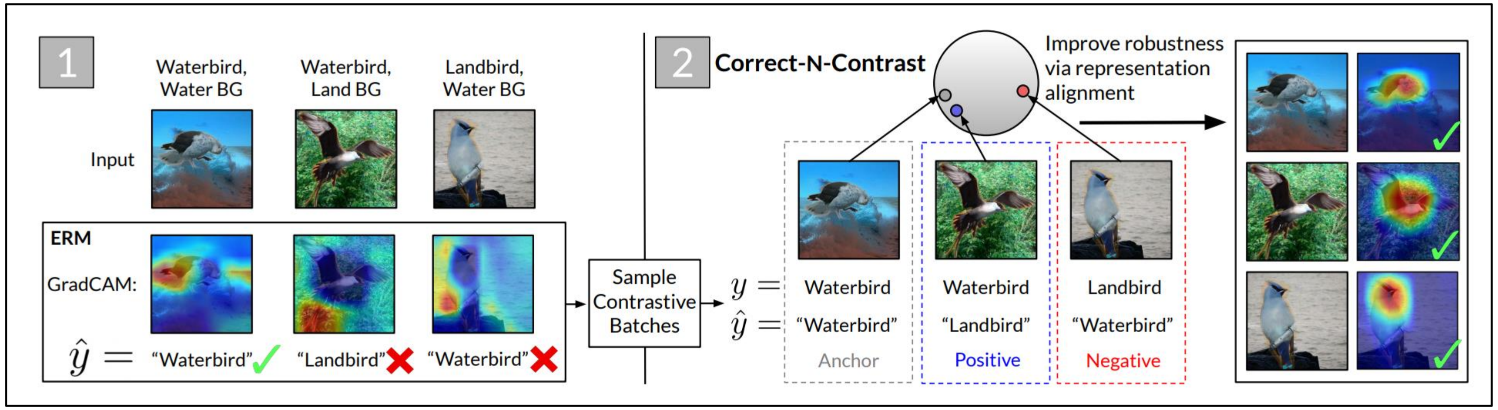
1. Identification

$$E = \{(x_i, y_i) \text{ s.t. } f_{\text{id}}(x_i) \neq y_i\}.$$

2. Upweighting

$$J_{\text{up-ERM}}(\theta, E) = \left(\lambda_{\text{up}} \sum_{(x,y)\in E} \ell(x,y;\theta) + \sum_{(x,y)\notin E} \ell(x,y;\theta)\right),$$

Correct-n-Contrast (CnC)
Zhang et al., 2022



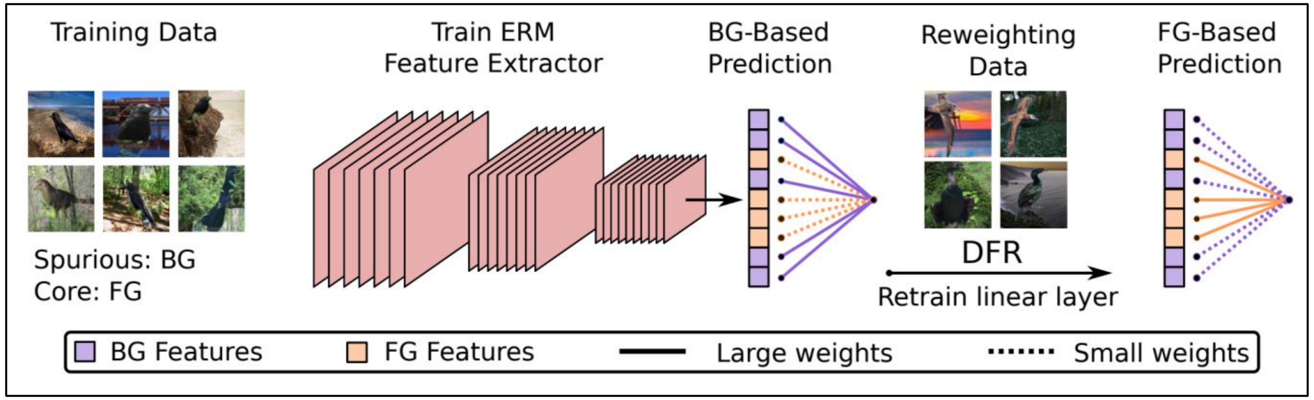
How previous work resolve this?

With knowing group label

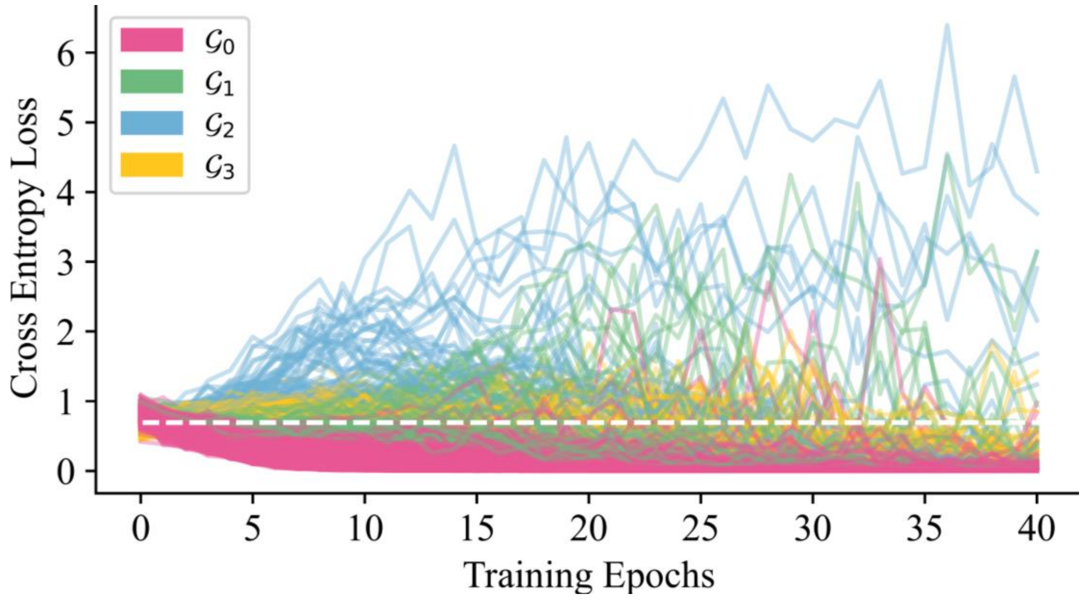
Group DRO
Sagawa et al.,
2019

$$\hat{\theta}_{\mathrm{DRO}} := \underset{\theta \in \Theta}{\arg\min} \left\{ \hat{\mathcal{R}}(\theta) := \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim \hat{P}_g} [\ell(\theta; (x,y))] \right\}$$

Deep Feature Reweighting (DFR)
Izmailov et al., 2022

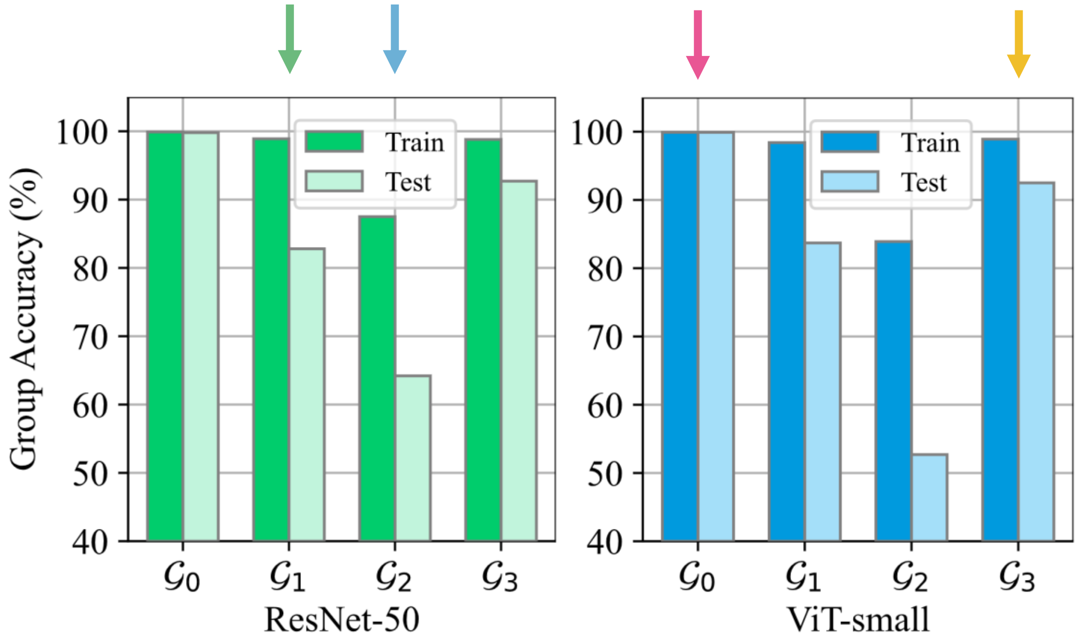


Minority groups manifest a significant gap in accuracy



Minority Group

Majority Group



Majority Group

$\mathcal{G}_0$: Landbird on Land
$\mathcal{G}_1$: Landbird on Water
$\mathcal{G}_2$: Waterbird on Land
$\mathcal{G}_3$: Waterbird on Water

Minority Group

Are minority group samples memorized by neural network?

Deep learning algorithms are well-known to have a propensity for fitting the training data very well and often fit even outliers and mislabeled data points.

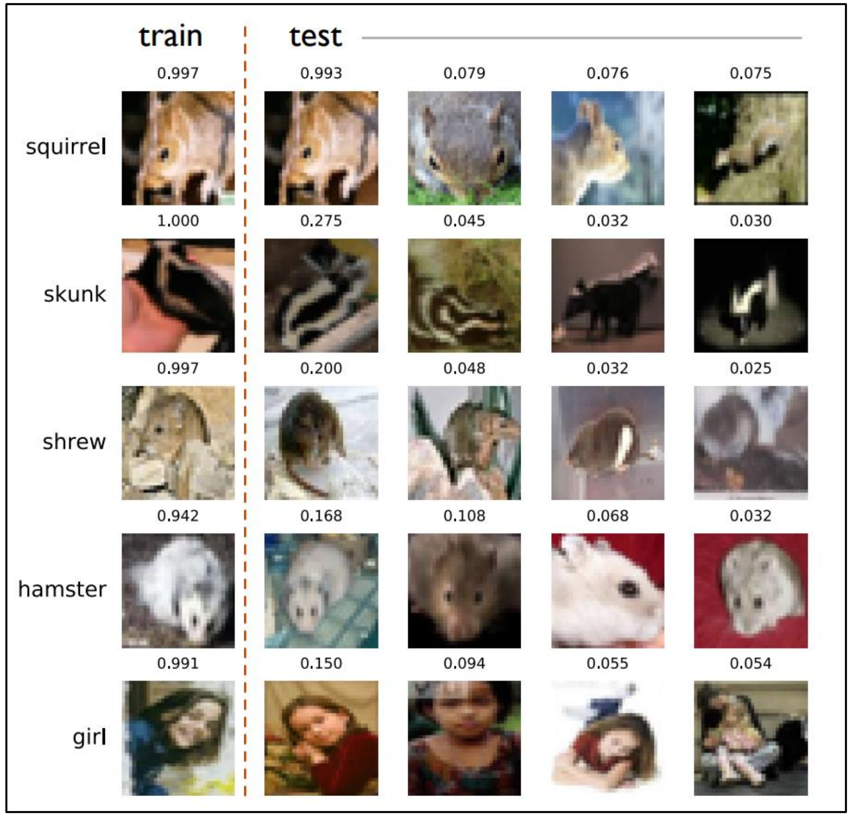
Such fitting requires memorization of training data labels.

Feldman & Zhang, 2020

Definition of memorization, Feldman 2021

Formally, for a dataset $S = (x_i, y_i)_{i \in [n]}$ and $i \in [n]$ define

$$\mathtt{mem}(\mathcal{A}, S, i) := \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S)}[h(x_i) = y_i] - \mathop{\mathbf{Pr}}_{h \sim \mathcal{A}(S^{\setminus i})}[h(x_i) = y_i],$$

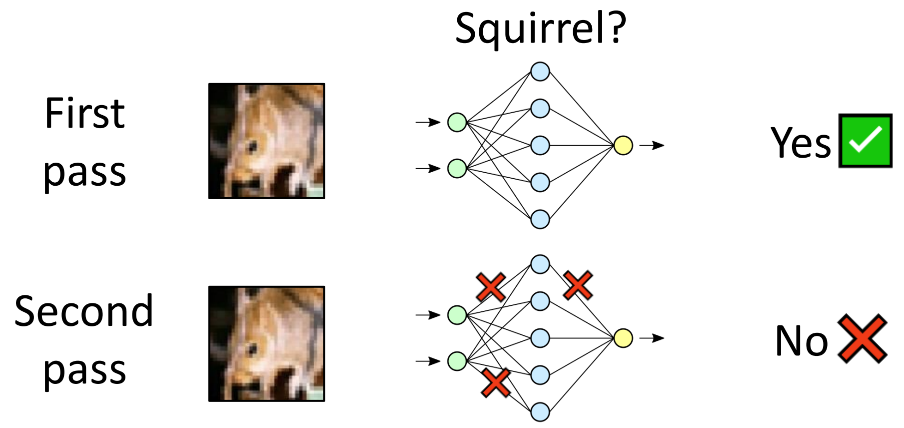


Feldman & Zhang, 2020

We formulate the spurious correlation problem as the memorization effect of the neural networks.

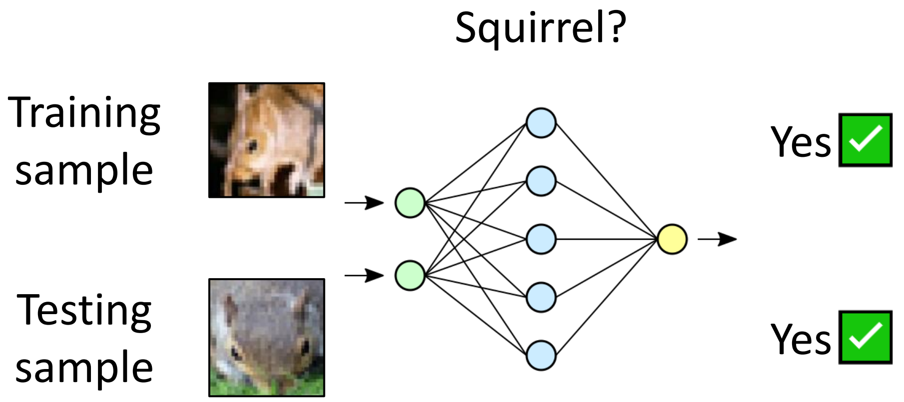
Question 1:

Can we find a set of neurons that play a critical role in the minority samples decision making?



Question 2:

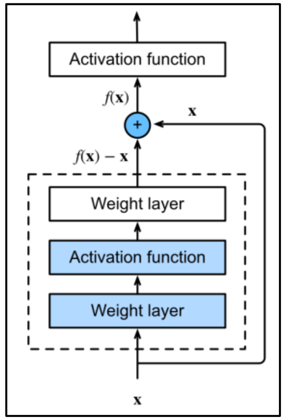
Can we find a way to cancel out the memorization effect caused by these neurons?



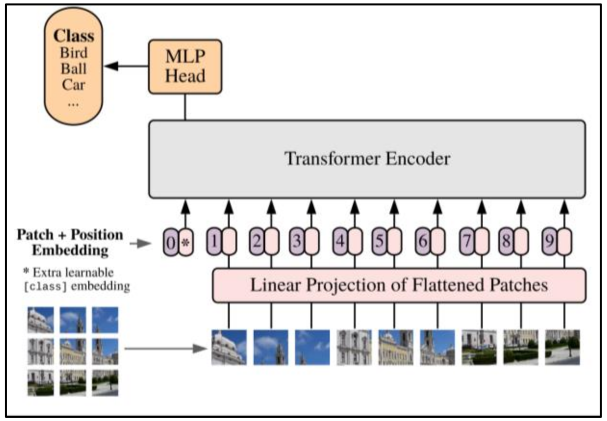
Preliminaries

Datasets and Models

ResNet50



ViT-Small



Identification Criterion of Critical Neurons

Magnitude-based: $||\mathbf{z}_i||_2$ Gradient-based: $||\mathbf{v}(i,j)||_2$

where $\boldsymbol{\theta} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$ $\quad \mathbf{v}(i,j) = \dfrac{\partial \mathcal{L}_{\text{CE}}(f(\boldsymbol{\theta}, \mathcal{D}_j))}{\partial \mathbf{z}_i}$

Definition of Neurons and Layers

Convolutional Layer: $\overbrace{256\times\underbrace{256\times3\times3}_{\text{Neuron}}}^{\text{Layer}}$

Linear Layer: $\overbrace{768\times2304\times\underbrace{1}_{\text{Neuron}}}^{\text{Layer}}$

Stage 1: Proving the Existence of Critical Neurons

Unstructured

- Zero-out (pruning)
- Random initialize
- Random noised

Top-1/2/3 largest (by magnitude/gradient) neuron within the whole model.

Structured Tracing:

Zero-out (pruning)

Top-1/2/3 largest (by magnitude/gradient) neuron within a specific layer.

Layer Rewinding:

Rewind

Every layer 5/10/20/30/40 epochs back in turn and keep all the other parameters unchanged

Zero-out Top-k Global Largest Neurons

1. Train the model by ERM for 40 epochs

2. Find the top-k global largest neurons by calculating

- Gradient norm (group variant) $\|\mathbf{v}(i,j)\|_2$ where $\mathbf{v}(i,j) = \frac{\partial \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{\theta}, \mathcal{D}_j))}{\partial \mathbf{z}_i}, \underbrace{i}_{\text{Neuron index}} \in \{1, \cdots, M\}; \underbrace{j}_{\text{Group index}} \in \{0, \cdots, 3\}$

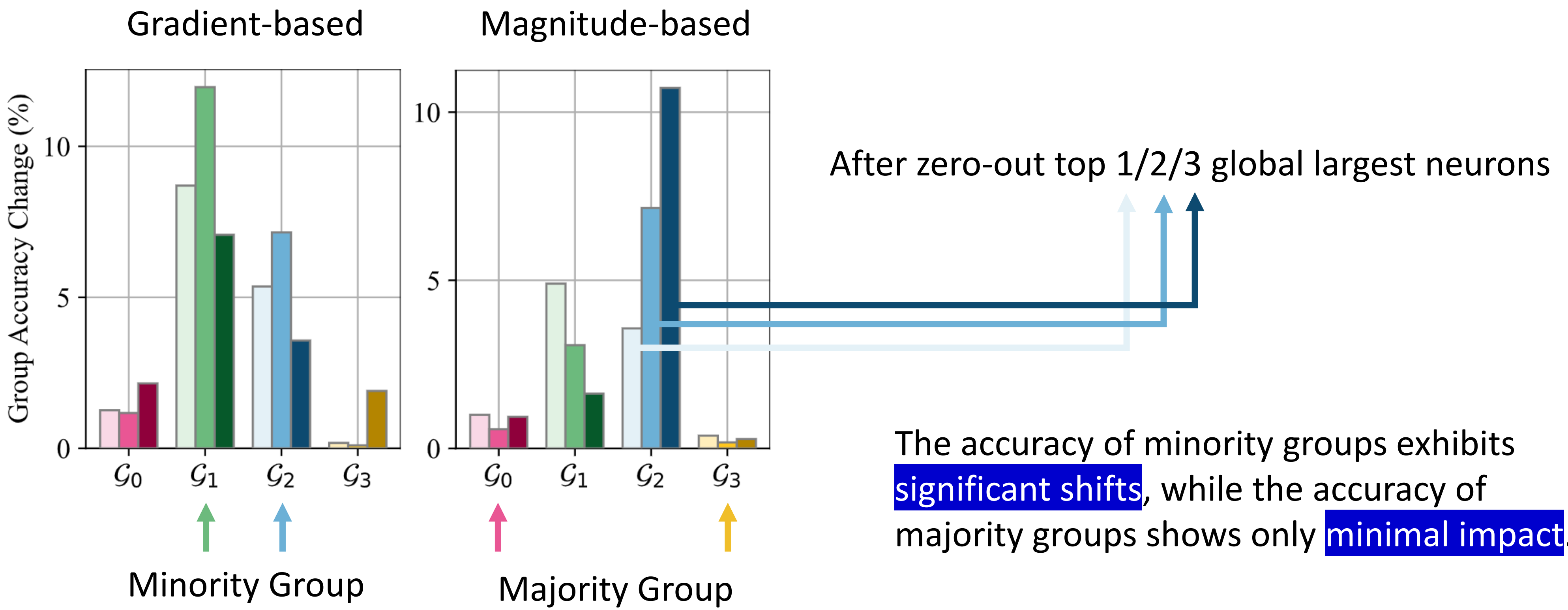
  where $\mathcal{D}_j$ comprises examples only from group $\mathcal{G}_j$

- Magnitude norm (group invariant) $\|\mathbf{z}_i\|_2$

3. Zero-out the identified neurons

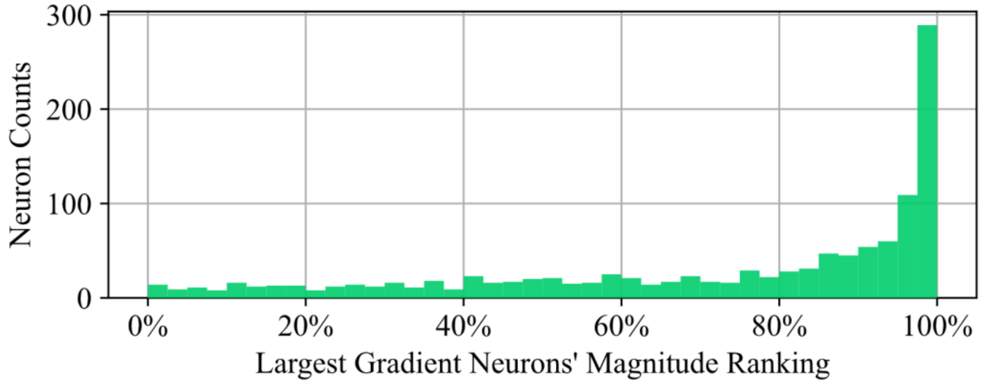
4. Calculate the group accuracy change by $\Delta_{\mathrm{acc}}(j) = |\,\mathrm{acc}(\mathcal{D}_j, f(\theta, \cdot)) - \mathrm{acc}(\mathcal{D}_j, f(\underbrace{\mathbf{m}_j}_{\text{Binary Mask}} \odot \theta, \cdot))|$

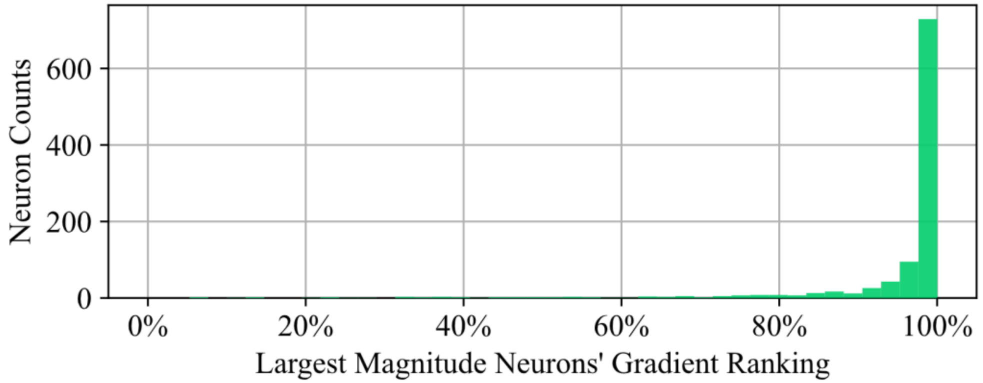
Result after Zero-out Top-k Global Largest Neurons



The accuracy of minority groups exhibits significant shifts, while the accuracy of majority groups shows only minimal impact.

Neurons Distribution by Gradient and Magnitude





In the top, we show the global magnitude ranking for the neurons with top 0.01% global largest gradient.

In the bottom, we show the global gradient ranking for the neurons with top 0.01% global largest magnitude.

In both histograms, there is a noticeable clustering in the rightmost two bins (ranging from 95% to 100%).

This suggests that the neurons with the highest magnitudes tend to exhibit large gradients, and the neuron with the largest gradient often coincides with a high weight magnitude.

Random-initialize Top-k Global Largest Neurons

1. Train the model by ERM for 40 epochs

2. Find the top-k global largest neurons by calculating

Neuron index

Group index

- Gradient norm (group variant): $\|\mathbf{v}(i,j)\|_2$ where $\mathbf{v}(i,j) = \dfrac{\partial \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{\theta}, \mathcal{D}_j))}{\partial \mathbf{z}_i}, i \in \{1, \cdots, M\}; j \in \{0, \cdots, 3\}$

  where $\mathcal{D}_j$ comprises examples only from group $\mathcal{G}_j$

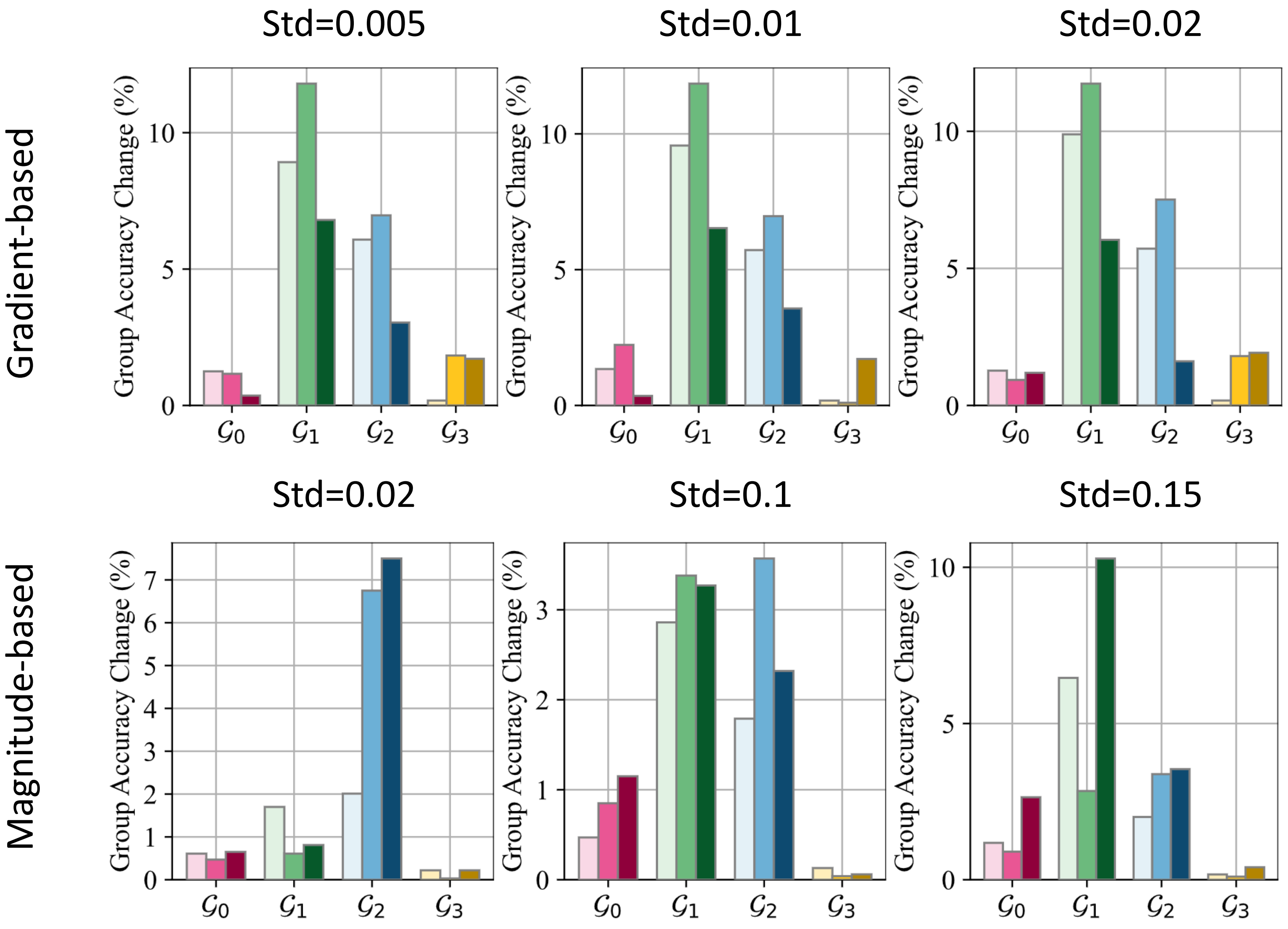
- Magnitude norm (group invariant): $\|\mathbf{z}_i\|_2$

3. Random-initialize the identified neurons b' replace the neuron weight $\mathbf{z}_i$ with ϵ_i where $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2)$

4. Calculate the group accuracy change by $\Delta_{\mathrm{acc}}(j) = |\mathrm{acc}(\mathcal{D}_j, f(\boldsymbol{\theta}, \cdot)) - \mathrm{acc}(\mathcal{D}_j, f(\widetilde{\boldsymbol{\theta}}, \cdot))|,$

where $\widetilde{\boldsymbol{\theta}} = \{\mathbf{z}_i\}_{i \notin \mathcal{I}_j} \cup \{\widetilde{\mathbf{z}}_i\}_{i \in \mathcal{I}_j}.$

Result after Random-initialize Top-k Global Largest Neurons



1) The results from random initialization closely resemble those from the pruning method.

2) The accuracy changes in minority groups still surpass those in majority groups.

3) All the results visualized here are the average of 10 independent runs.

Random-noise Top-k Global Largest Neurons

1. Train the model by ERM for 40 epochs

2. Find the top-k global largest neurons by calculating

- Gradient norm (group variant): $\|\mathbf{v}(i,j)\|_2$ where $\mathbf{v}(i,j) = \frac{\partial \mathcal{L}_{\text{CE}}(f(\boldsymbol{\theta}, \mathcal{D}_j))}{\partial \mathbf{z}_i}, \underbrace{i}_{\text{Neuron index}} \in \{1, \cdots, M\}; \underbrace{j}_{\text{Group index}} \in \{0, \cdots, 3\}$

  where $\mathcal{D}_j$ comprises examples only from group $\mathcal{G}_j$

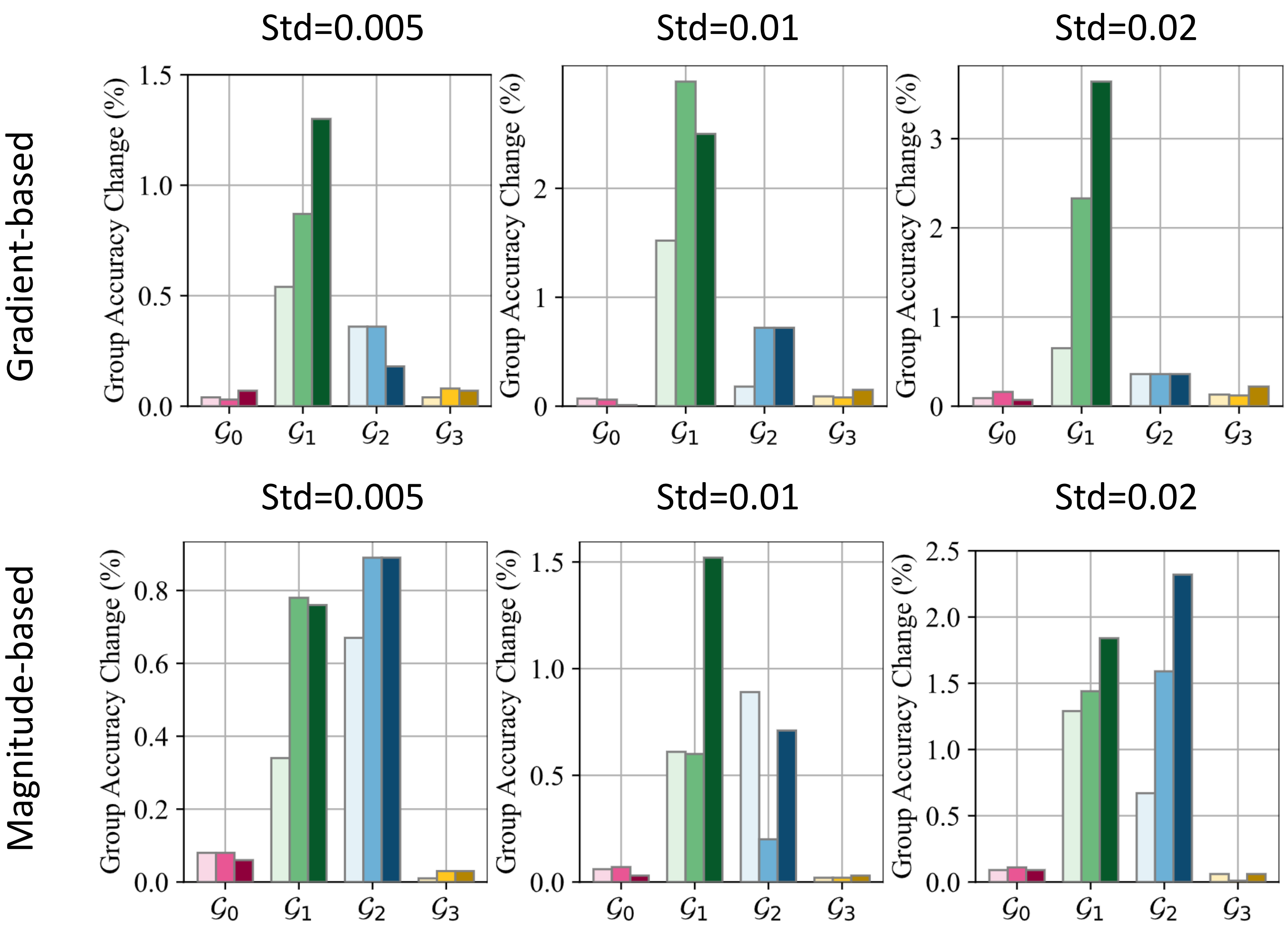
- Magnitude norm (group invariant): $\|\mathbf{z}_i\|_2$

3. Random-noise the identified neurons by add the neuron weight $\mathbf{z}_i$ with ϵ_i where $\epsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma^2)$

4. Calculate the group accuracy change by $\Delta_{\text{acc}}(j) = |\text{acc}(\mathcal{D}_j, f(\boldsymbol{\theta}, \cdot)) - \text{acc}(\mathcal{D}_j, f(\widetilde{\boldsymbol{\theta}}, \cdot))|$,

where $\widetilde{\boldsymbol{\theta}} = \{\mathbf{z}_i\}_{i \notin \mathcal{I}_j} \cup \{\widetilde{\mathbf{z}}_i\}_{i \in \mathcal{I}_j}$.

Result after Random-noise Top-k Global Largest Neurons



1) The extent of accuracy change with random noise is much smaller than that observed with random initialization and pruning.

2) With random noise added, the accuracy changes in minority groups still surpass those in majority groups.

3) All the results visualized here are the average of 10 independent runs.

Zero-out Top-k Largest Neurons within a Layer

1. Train the model by ERM for 40 epochs

2. Find the top-k largest neurons within a layer by calculating

Gradient norm (group variant) $\|\mathbf{v}(i,j)\|_2$ where $\mathbf{v}(i,j) = \frac{\partial \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{\theta}, \mathcal{D}_j))}{\partial \mathbf{z}_i}, i \in \{1, \cdots, M\}; j \in \{0, \cdots, 3\}$

where $\mathcal{D}_j$ comprises examples only from group $\mathcal{G}_j$

Magnitude norm (group invariant) $\|\mathbf{z}_i\|_2$

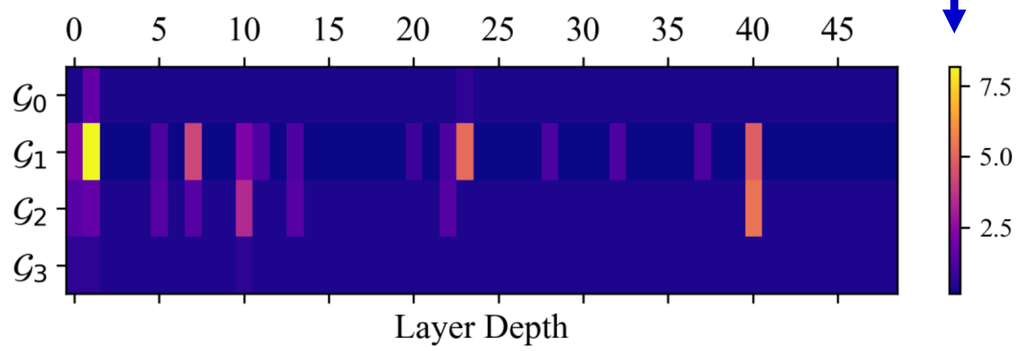
3. Zero-out the identified neurons

4. Calculate the group accuracy change by $\Delta_{\mathrm{acc}}(j) = |\,\mathrm{acc}(\mathcal{D}_j, f(\theta,\cdot)) - \mathrm{acc}(\mathcal{D}_j, f(\mathbf{m}_j \odot \theta, \cdot))|$

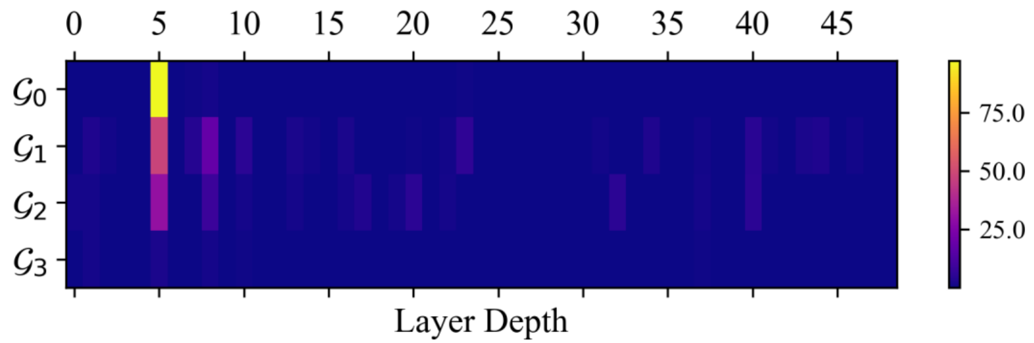
Binary Mask

Result of Zero-out Top-k Largest Neurons within a Layer

Group Accuracy
Percentage Change

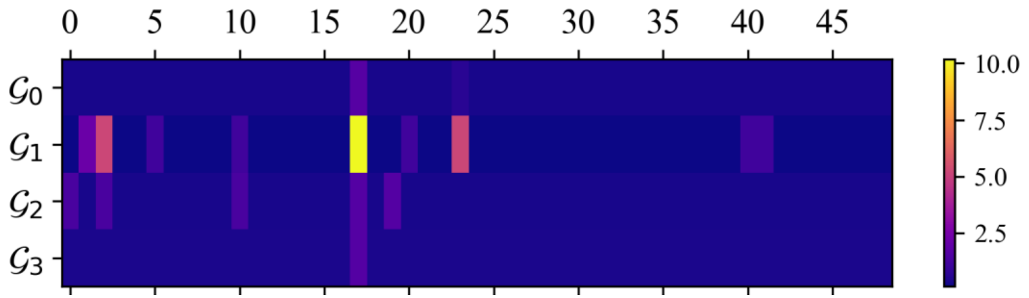


Zero-out Top-1 Neuron with Largest Magnitude

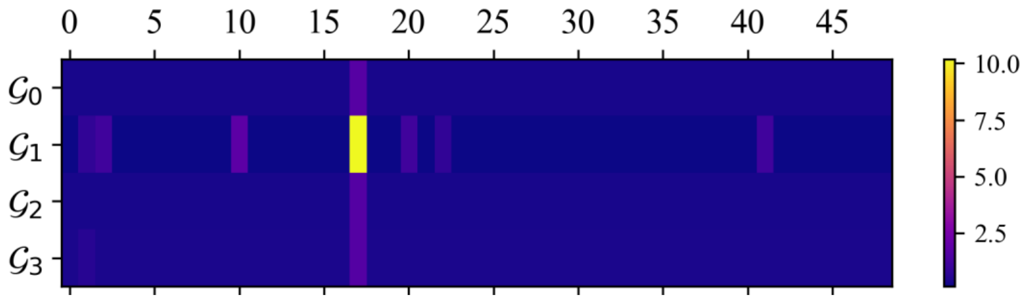


Zero-out Top-3 Neurons with Largest Magnitude

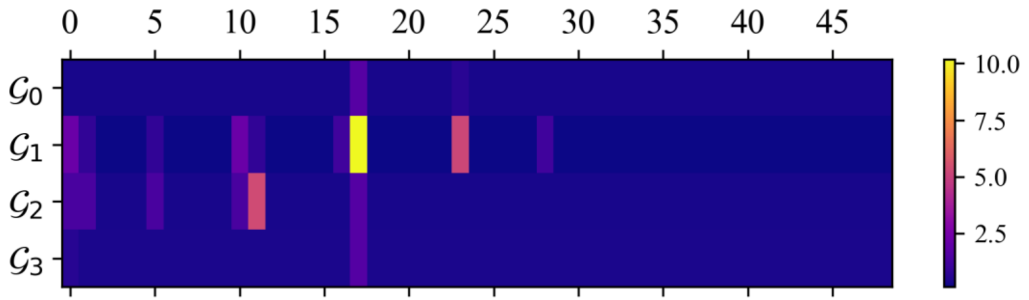
Result of Zero-out Top-1 Largest Neurons within a Layer



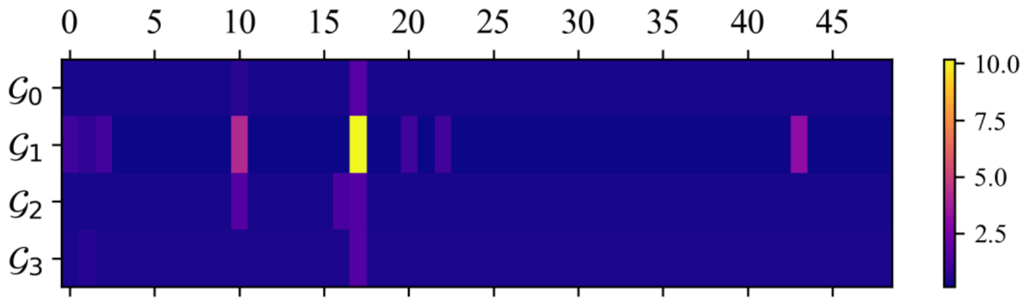
Zero-out Top-1 Neuron with Largest Gradient by Group 0 samples



Zero-out Top-1 Neuron with Largest Gradient by Group 1 samples

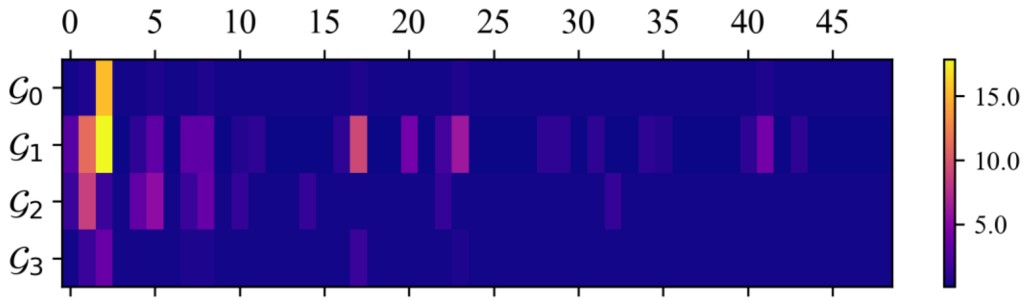


Zero-out Top-1 Neuron with Largest Gradient by Group 2 samples

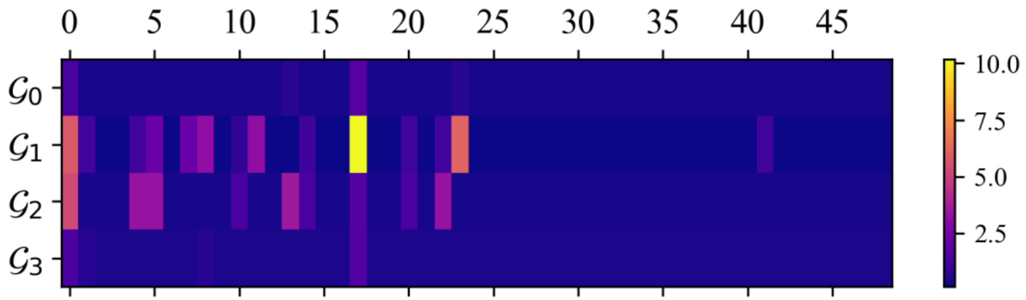


Zero-out Top-1 Neuron with Largest Gradient by Group 3 samples

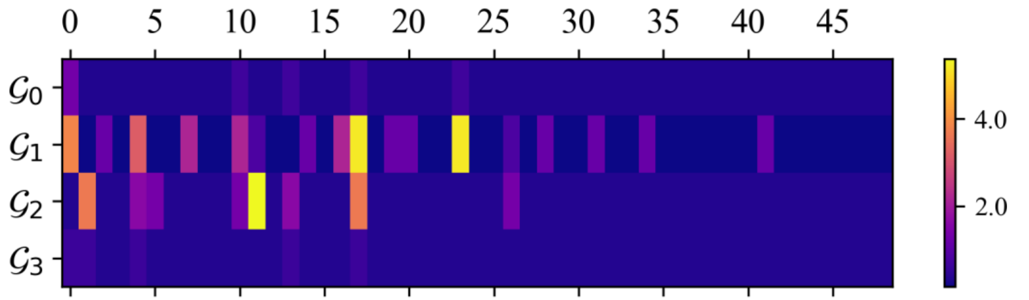
Result of Zero-out Top-3 Largest Neurons within a Layer



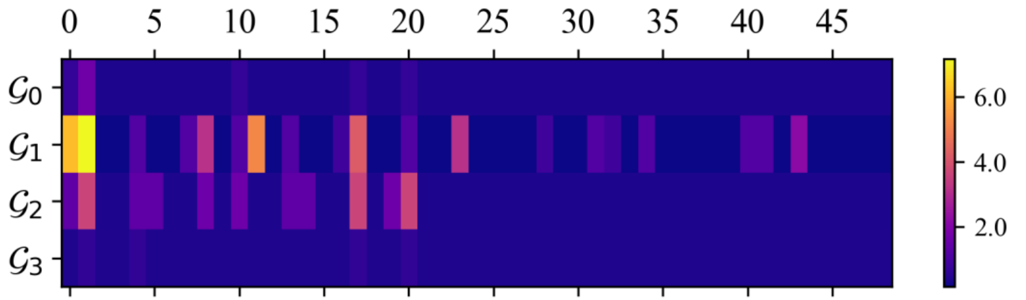
Zero-out Top-3 Neuron with Largest Gradient by Group 0 samples



Zero-out Top-3 Neuron with Largest Gradient by Group 1 samples



Zero-out Top-3 Neuron with Largest Gradient by Group 2 samples



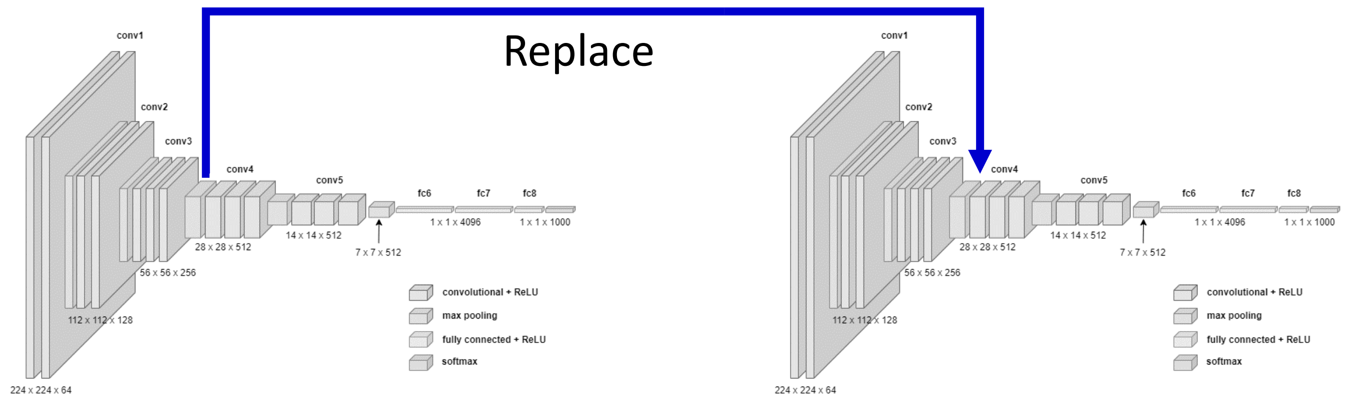
Zero-out Top-3 Neuron with Largest Gradient by Group 3 samples

Rewind Layer

1. Train the model by ERM for 40 epochs, save every checkpoint during training

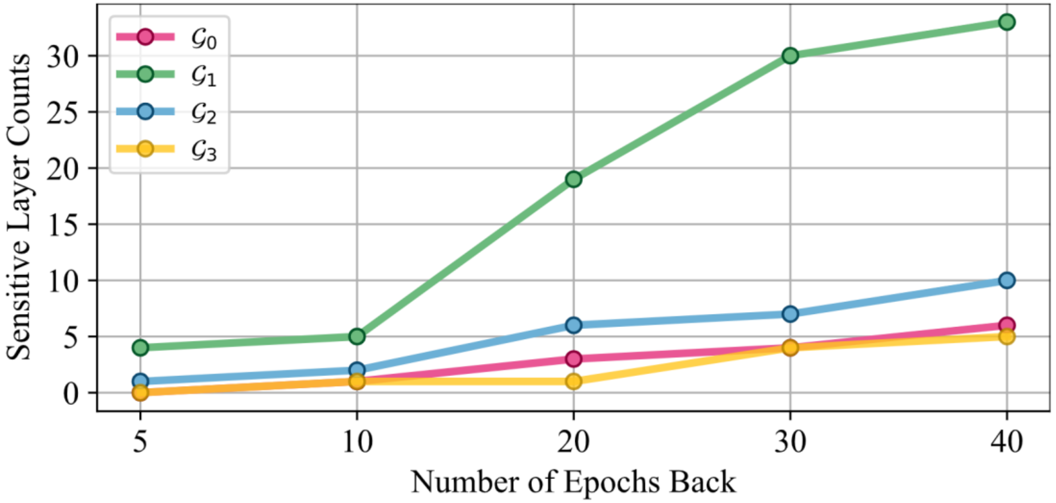
2. Replace the layer with the corresponding parameters 5/10/20/30/40 epochs earlier, keep all the other parameters unchanged

3. Calculate the group accuracy change by $$\Delta_{\mathrm{acc}}(j) = |\mathrm{acc}(\mathcal{D}_j, f(\boldsymbol{\theta}, \cdot)) - \mathrm{acc}(\mathcal{D}_j, f(\widetilde{\boldsymbol{\theta}}, \cdot))|,$$



Earlier Checkpoint Current Checkpoint

Result of Rewind Layer



$\mathcal{G}_0$: Landbird on Land
$\mathcal{G}_1$: Landbird on Water
$\mathcal{G}_2$: Waterbird on Land
$\mathcal{G}_3$: Waterbird on Water

A sensitive layer is defined as the layer rewinding on which can bring +1% change in corresponding group accuracy.

Minority groups tend to have a higher count of sensitive layers compared to majority groups. This suggests that majority groups exhibit greater resilience when it comes to rewinding layers.

For any given group, a larger number of layers influence group accuracy when rewound to earlier checkpoints.

Group accuracy change by pruning non-critical neuron (control group)

	Group 0	Group 1	Group 2	Group 3
Zero-out 0.01%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 0.02%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 0.03%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 0.1%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 0.2%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 0.3%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 1%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 2%	0.001338%	0.003913%	0.002857%	0.004816%
Zero-out 3%	0.001338%	0.003913%	0.002857%	0.004816%

We found that all pruning actions had minimal impact on the accuracy of all groups.

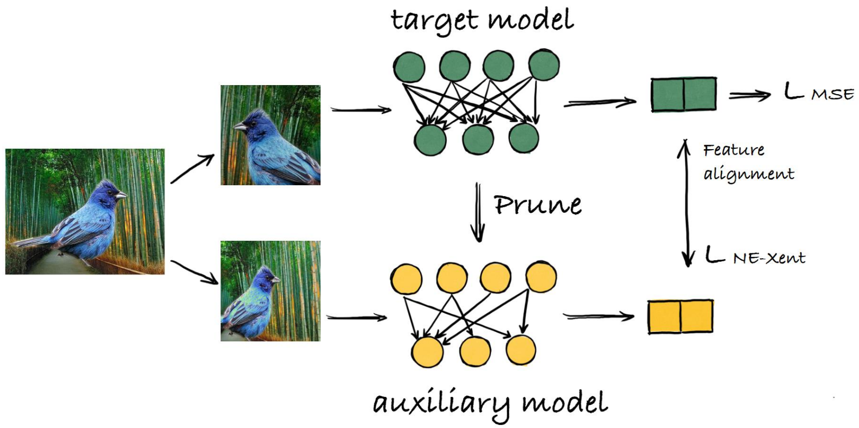
Stage 2: Mitigating the spurious correlation via pruning

1. Train the model by ERM for 40 epochs

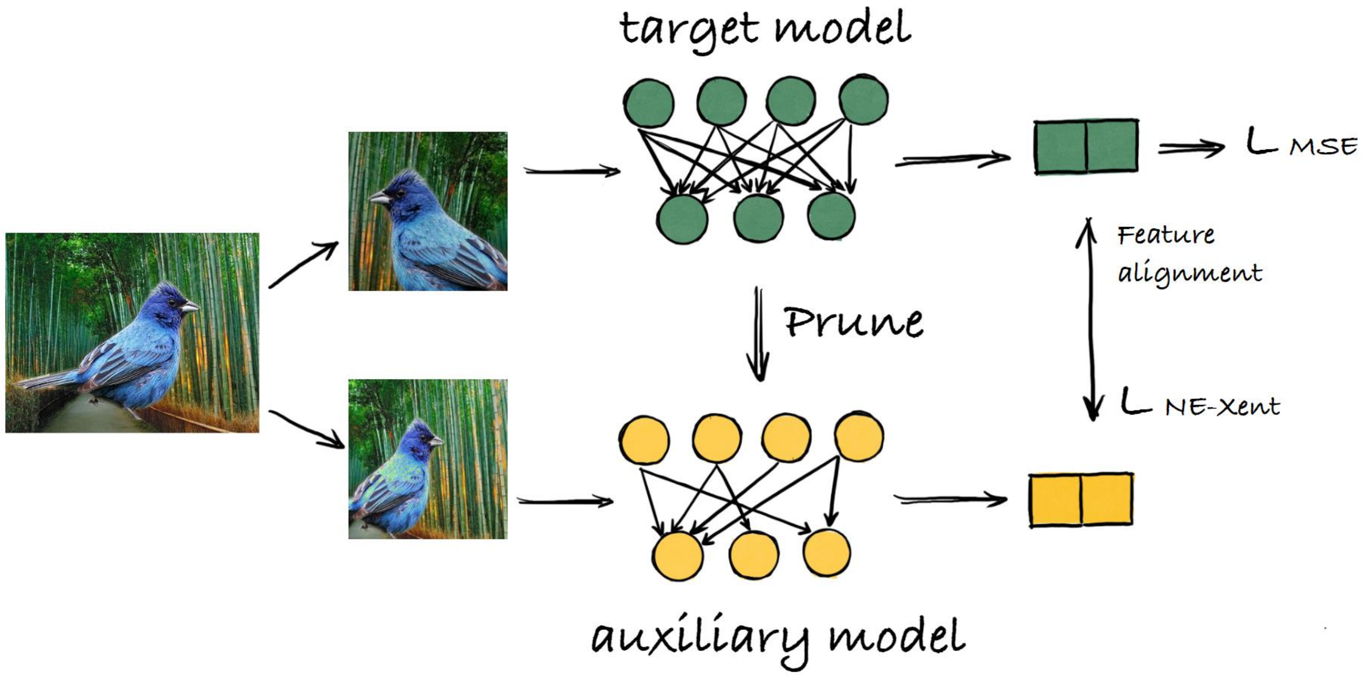
2. Find the critical neurons

3. Prune the critical neurons in auxiliary model

4. Finetune the model by this framework for 20 more epochs by contrastive learning



Stage 2: Mitigating the spurious correlation via pruning



Positive (negative) pairs are output features that originate from the same (different) input image.

We wish this term be as big as possible

$$\mathcal{L}_{\text{NT-Xent}}(\boldsymbol{\theta}, \mathbf{x}) = -\log \frac{\exp(\text{sim}(\mathbf{r}, \mathbf{r}_p)/\tau)}{\sum_k \exp(\text{sim}(\mathbf{r}, \mathbf{r}_k)/\tau)}$$

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \mathbf{v} / (\|\mathbf{u}\| \cdot \|\mathbf{b}\|)$$

Training Objective

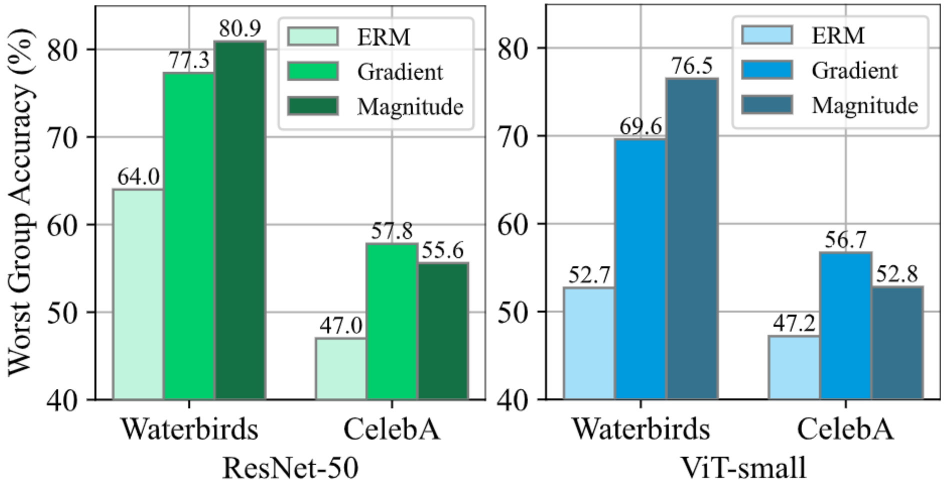
$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) = \mathcal{L}_{\text{NT}}(\boldsymbol{\theta}, \mathbf{x}) + \lambda \mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y})$$

$$\mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}) = \|\widehat{\mathbf{y}} - \mathbf{y}\|_2^2$$

How do we decide which neurons to prune?

Gradient-based:
prune 0.01% neurons with largest gradient

Magnitude-based :
prune 0.01% neurons with largest magnitude



How do we calculate the gradient for gradient-based pruning?

1. Calculate the cross-entropy loss for each sample

2. Select the top 256 samples with the highest

3. Randomly sample 128 out to form the batch for gradient computation

Our finetuning strategy does not rely on group labels!

Conclusions

1. Our comprehensive study verifies the presence of spurious memorization, a mechanism involving critical neurons significantly influencing the accuracy of minority examples while having minimal impact on majority examples.

2. Building upon these key findings, we demonstrate that by intervening with these critical neurons, we can effectively mitigate the influence of spurious memorization and enhance the performance on the worst group.