Large Learning Rates Simultaneously Achieve Robustness to Spurious Correlations and Compressibility

Supplementary Material

In this document, we present in Sec. A an extended review of the preceding research to contextualize our paper's motivation and findings. Building on this, Sec. B we highlight how our findings extend and improve upon previous literature, and point toward fruitful future research directions. After providing additional details regarding our experiment settings in Sec. C, we provide additional results and statistics regarding the main paper's findings in Sec. D. Lastly, Sec. E reviews our attribution visualization methodology, and provides extensive additional visual evidence for our claims.

A. Extended Related Work

A large amount of recent work on spurious correlations (SCs) have focused on the "default" tendency of neural networks, trained under gradient-based empirical risk minimization (ERM), to exploit simple features in the training datasets at the expense of more complex yet robust/invariant ones. These include [4], who highlight the overreliance of vision models on background information; [67], who emphasize the "simplicity bias" of neural networks in preferring simple features over more complex and informative ones, and [22], who emphasize the tendency of neural networks in engaging "shortcut learning" in various modalities of application. Ensuing research proposed several explanations for this phenomenon. For example, [54, 59, 65] highlight the inductive bias of a maximum margin classifier as the primary reason for the exploitation of spurious features. Alternatively, [54, 57] emphasize the dynamics of gradient-based learning in creating this effect, where early adoption of simple (and spurious) features harms the later learning of more complex and more informative features.

Building on diagnoses, such as those mentioned above, for the cause of this unintended learning of spurious features, other research propose interventions to mitigate this problem. For example, [59] propose new losses that optimize for a *uniform* margin solution rather than a max-margin solution. On the other hand, [45, 56] propose two-stage methods that reweight the dataset by deemphasizing samples that are learned earlier, and [74] discourage neural network to produce representations predictive of the label early in the neural network. Other methods assume access to spurious feature labels at training time, and exploit these in various ways to improve robustness [28, 64]. While to our knowledge no previous research *systematically* investigates the effect of LR on generalization under SCs (and in relation to compressibility), some previous research hint at the outsized impact of LR on such behavior. [43] examine the effect of large LRs on feature learning and generalization, without explicitly addressing the implications in an OOD generalization context. While [58] speculate about the potential effects of LR tuning on OOD generalization, [28] empirically find that LR is most likely to affect robustness to SCs, and [59] speculate that large LRs might lead to improved performance due to inability of models trained thereunder to approximate a max-margin solution.

While previous research showed a positive relationship between compressibility and generalization through theoretical and empirical findings [2, 3, 5, 72, 73], it is much less clear how well this applies to OOD generalization. Indeed, existing research provides at best an ambivalent picture regarding the simultaneous achieveability of generalization, robustness, and compressibility [14, 16, 18, 77]. Various studies have highlighted the impact of large LRs on generalization [40, 43, 53], model compressibility [3], and representation sparsity [1]; making it a prime candidate for further investigation regarding its ability to facilitate compressibility and robustness in tandem. [29] point out how large LRs in early training prevent the iterates from being stuck in narrow valleys in the loss landscape, where the curvature in certain directions is high. [40, 53] point out the importance of large LRs in early training, where basin-jumping behavior leads to better generalizing and/or flatter minima [30]. While [1, 43, 85] focus on the effect of large LRs on feature learning, [63] demonstrate the crucial role of spurious / opposing signals in early training, and how progressive sharpening [8, 78] of the loss landscape in the directions that pertain to the representation of these features lead to the eventual down-weighting of such non-robust features. [63] further observe that this is due to discrete nature of practical steepest ascent methods (GD, SGD), as it is not observed in gradient flow regime, suggesting learning rate as a prime candidate for modulating this behavior.

B. Extended Discussion of Our Contributions

In this paper, we demonstrate through systematic experiments the unique role large learning rates (LRs) in simultaneously achieving robustness and resource-efficiency. More concretely, we demonstrate that:

- Large LRs simultaneously facilitate robustness to SCs and compressibility in a variety of datasets, model architectures, and training schemes.
- Increase in robustness and compressibility is accompanied by increased core (aka stable, invariant) feature uti-

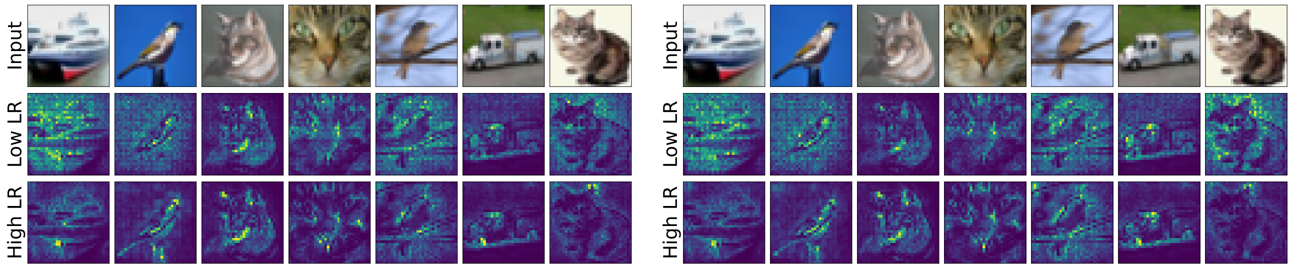


Figure 14. Visualizing attributions on a CIFAR-10 dataset with ResNet18 models using Integrated Gradients (left) and DeepLift (right).

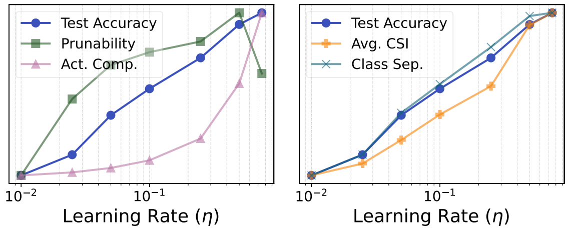


Figure 15. (Left) Effects of learning rate on OOD performance (unbiased test acc.), network prunability, and representation properties with the moon-star dataset.

lization and class separation in learned representations.

- Large LRs are unique in consistently achieving these properties across datasets compared to other interventions, and can be combined with explicit regularization for even better performance.
- Large LRs have a similar effect in naturalistic classification tasks by addressing hidden/rare spurious correlations in the dataset.
- Confident mispredictions of bias-conflicting samples play an important role in conferring robustness to models trained under large LRs.

We now discuss further implications of our results in light of recent findings in the literature.

Inductive biases of SGD. Our findings call into question the assumptions regarding the inductive biases of "default" SGD. We find that LR selection can change the unbiased test set accuracy by *up to* $\sim$*50%* (See Table S1). This has two major implications: (i) Any method that relies on assumptions regarding default behavior of neural networks (e.g. [45, 56]) should consider the fact that the said defaults can vary considerably based on training hyperparameters, including but not limited to LR. (ii) Any proposed intervention for improving robustness to SCs should consider utilizing large LRs as a strong baseline against the proposed method.

Overparametrization and robustness. We observe a strong interaction between overparametrization and LR in robustness to SCs. Our findings show that (Table S1) the range of test accuracies that can be obtained by tuning LRs increase as the models get more expressive. For example, in Colored MNIST dataset, while the difference between the highest vs. lowest performing models is $\sim$4% for a fully connected network (FCN), this increases to $\sim$23% for ResNet18 and $\sim$31% for ResNet50. This implies that while overparametrization indeed seems to play an important role in robustness to SCs as suggested by some previous work [65] (cf. [59]), this needs to be considered in the context of central training hyperparameters, as they can modulate this vulnerability to a great degree.

Mechanism of LR's Effects. As noted in the main paper, interventions developed to mitigate the effect of spurious correlations constitute two groups based on whether they assume access to spurious feature labels/annotations. Those that assume this, exploit this information to improve worst group or unbiased test set performance [28, 64]. In the absence of group annotations, other methods rely on assumptions about the nature of the spurious features, data distribution, and the inductive biases of the learning algorithms [45, 59, 74]. We highlight that the proposed mechanism in this paper, where large LRs cause BC sample losses to dominate acts as an *implicit* re-weighting, which makes it akin to two-stage methods like Learning from Failure [56]. This further highlights the importance of establishing the effects of core hyperparameters on robustness to spurious correlations - not only for a deeper understanding of the inductive biases of gradient-based learning under overparametrization, but also as strong baselines to compare against newly developed methods.

As noted above, [59] argue that max-margin classification inevitably leads to exploitation of spurious features, and LRs might protect against SCs by creating models closer to a uniform margin solution. To support their conjecture, they present evidence that shows average losses incurred from bias-aligned (BA) vs. bias-conflicting (BC) samples are closer in large LR models compared to small LR models. However, we find that their findings do not generalize across different data distributions. Computing avg. loss for BA samples / avg. loss for BC samples, we find that in both Colored MNIST and Double MNIST datasets, low LR models produce average losses that are closer in ratio (3.9×10^{-3} vs. 3.4×10^{-3} in Colored MNIST and

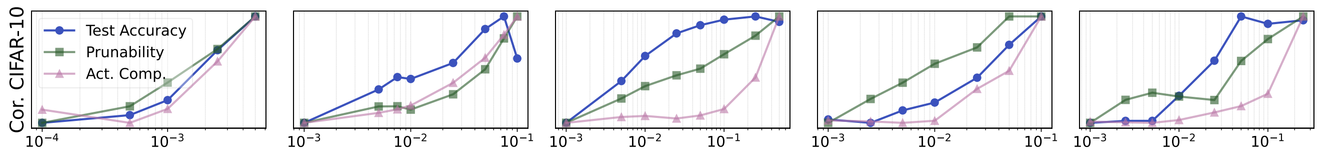


Figure 16. Effects of LR on OOD performance (unbiased test acc.), network prunability, and representation (activation) compressibility in Corrupted CIFAR-10 dataset. x-axes correspond to learning rate (η), y-axes are normalized within each figure for each variable.

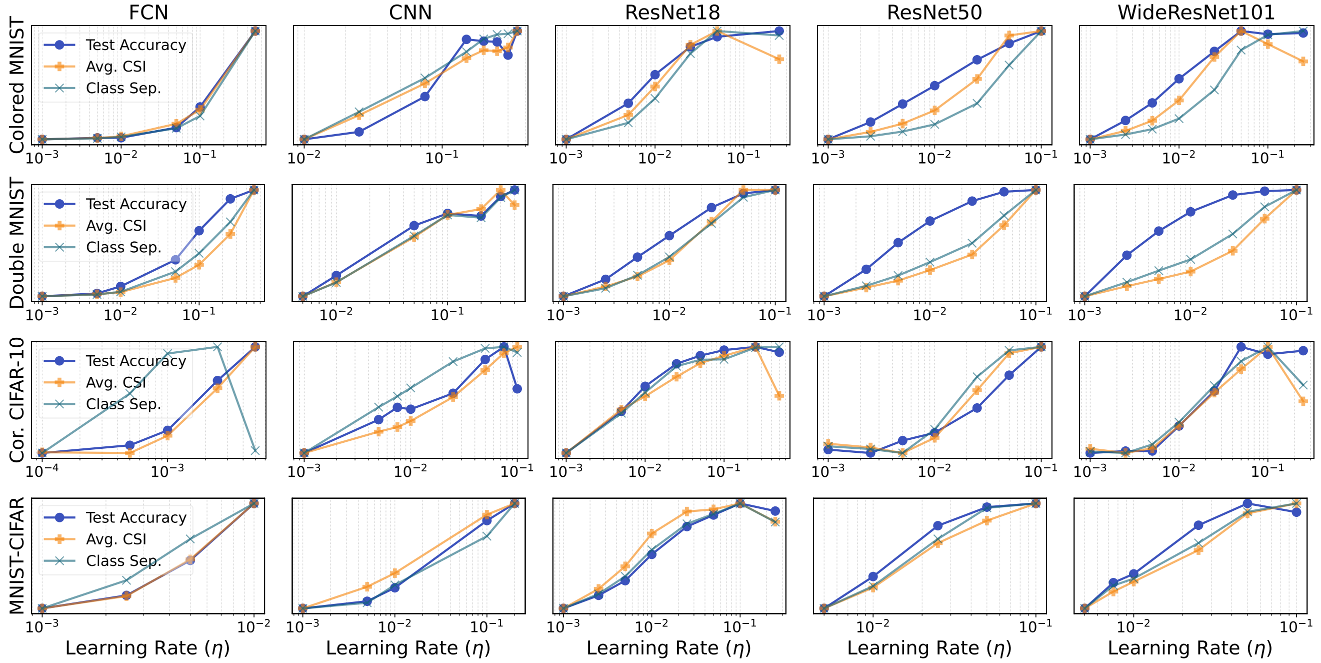


Figure 17. Effects of learning rate on representation (activation) statistics for semi-synthetic datasets. y-axes are normalized within each figure for each variable.

$9.5 \times 10^{-}3$ vs. $7.4 \times 10^{-}6$ in Double MNIST). This highlights the importance of testing such claims across diverse settings, and emphasizes the need for novel and systematic explanations for the effect of large LRs on robustness to SCs.

Compressibility and generalization. [1] argue that large LRs create models with sparse representations. Our findings support their claim across diverse settings. However, we also observe that LRs' effects on unbiased test accuracy and network compressibility (i.e. prunability) precede that of activation sparsity (i.e. there are LRs that are large enough to increase test accuracy and prunability but not large enough to increase representation sparsity). This strongly implies that the representation sparsity is a downstream effect of large LRs, rather than being a mediator of generalization and network compressibility. On the other hand, our findings include initial evidence for wide minima [30] found by large LRs to be associated with increased core feature utilization. Examining the interaction of parameter and representation space properties produced by LRs (see e.g. [63]) is a promising future direction for understanding the inductive bias of large LRs and SGD in general.

C. Further Details on Experiment Settings

We use Python programming language for all experiments included in this paper. For experiments with semi-synthetic, realistic SC, and naturalistic datasets we use the versions of ResNet18, ResNet50, Wide ResNet101-2, Swin Transformer (tiny) as included in the Python package `torchvision`, as well as a FCN with two hidden layers of width 1024, ReLU as the activation activation function, and with no bias. We also use a CNN with a similar architecture to VGG11 [69], with a single linear layer following the convolutional layers instead of three, and this version includes no bias terms. Due to the worse default performance of FCNs in the more difficult semi-synthetic datasets MNIST-CIFAR and Corrupted CIFAR-10, we increase bias-conflicting ratio ρ to 0.25, and increase network width to 2048 for these datasets. The synthetic dataset experiments have been conducted with an FCN of two hidden layers and 200 width.

For Colored MNIST and Corrupted CIFAR-10 we use the train/test splits from the original papers [45]. Double MNIST and MNIST-CIFAR are created using the canonical splits of these datasets. The training/test splits for these datasets are 60000/10000, 50000/10000, 60000/10000, and 50000/10000, respectively. While we use the original splits for CelebA and Waterbirds datasets, we use a 10000/10000 split for the synthetic parity dataset. The learning rate ranges for the experiments are provided in Tab. 1 and Tab. 2, while all experiments included a batch size of 100, except for experiments with Swin Transformer, where we utilize a batch size of 16. For computing activation statistics, we obtain the post-activation values for the penultimate layer, and compute the compressibility values for 1000 randomly sampled input from the test set, and present the average of these values.

The experiments in this paper were run on 4 NVIDIA A100-PCIE-40GB GPUs for 400 total hours of computation. We will make our source code public upon publication to allow for the replication of our results.

D. Additional Results and Statistics

D.1. Proof of Proposition 1

Proof 1 *Let y and y' for the correct and incorrect classes for a sample,* i.e. *$b = y'$ for bias-conflicting samples, and $b = y$ otherwise. For the mispredicted (bias-conflicting) examples, let $f_{\mathbf{w}}[y'] - f_{\mathbf{w}}[y] = \beta > 0$. This implies the following softmax (π) output for y*

$$\pi_y = \frac{e^{kf_{\mathbf{w}}[y]}}{e^{kf_{\mathbf{w}}[y']} + e^{kf_{\mathbf{w}}[y]}} = \frac{1}{1+e^{k\beta}}. \tag{2}$$

Notice that $\pi_y \approx e^{-k\beta}$, as $k \to \infty$. Then we can say $\ell(y, f_{\mathbf{w}}(\mathbf{x})) = -\log \pi_y \approx k\beta$.

For the correctly predicted samples, let $f_{\mathbf{w}}[y] - f_{\mathbf{w}}[y'] = \alpha > 0$. Similarly to above, note that

$$\begin{aligned} \pi_y = 1 - \pi_{y'} &= 1 - \frac{e^{kf_{\mathbf{w}}[y']}}{e^{kf_{\mathbf{w}}[y']} + e^{kf_{\mathbf{w}}[y]}} && (3)\\ &= 1 - \frac{1}{1+e^{k(f_{\mathbf{w}}[y]-f_{\mathbf{w}}[y'])}} && (4)\\ &= 1 - \frac{1}{1+e^{k\alpha}} && (5)\\ &\approx 1 - e^{-\alpha k} && (6)\end{aligned}$$

as $k \to \infty$. As $k \to \infty$, $-\log(\pi_y) \approx -\log(1-e^{-k\alpha}) \approx e^{-k\alpha}$. Assume (without loss of generality) that the margins β and α are shared by all bias-conflicting and bias-aligned samples in the minibatch. Then, as $k \to \infty$,

$$\frac{\sum_{\mathbf{x},y\in\Omega_{\mathrm{bc}}} \ell(y, kf_{\mathbf{w}}(\mathbf{x}))}{\sum_{\mathbf{x},y\in\Omega_{\mathrm{ba}}} \ell(y, kf_{\mathbf{w}}(\mathbf{x}))} \approx \frac{|\Omega_{\mathrm{bc}}| \cdot k\beta}{|\Omega_{\mathrm{ba}}| \cdot e^{-\alpha k}} = \mathcal{O}(ke^{\alpha k}), \tag{7}$$

proving Eq. (1).

D.2. Additional Theoretical Results

To investigate the effects of mispredicted bias-conflicting samples on the gradients of subnetworks that rely on core vs. spurious features, we first define *bias-decomposable network*s.

Definition 1 *$f_{\mathbf{w}}$ is called a bias-decomposable network if $f_{\mathbf{w}}(\mathbf{x}) = f_{\mathbf{c}}(\mathbf{x}) + f_{\mathbf{s}}(\mathbf{x}) + f_{\mathbf{r}}(\mathbf{x})$. Here, $f_{\mathbf{c}}$ is the core feature subnetwork, $f_{\mathbf{c}}(\mathbf{x})[y] - f_{\mathbf{c}}(\mathbf{x})[b] \geq 0$ with equality iff $y = b$. $f_{\mathbf{c}}(\mathbf{x})$ is assumed to have converged to a stable decision making rule, i.e. $\nabla_{\mathbf{c}}(f_{\mathbf{c}}(\mathbf{x})[y] - f_{\mathbf{c}}(\mathbf{x})[b]) = 0$. In contrast, $f_{\mathbf{s}}$ is the spurious feature subnetwork, $f_{\mathbf{s}}(\mathbf{x})[y] - f_{\mathbf{s}}(\mathbf{x})[b] \leq 0$ with equality iff $y = b$. $f_{\mathbf{r}}$ is the remainder subnetwork that does not conform to the behavior described for $f_{\mathbf{c}}, f_{\mathbf{s}}$.*

Before discussing the motivation for this idealization, in the following proposition, we investigate how the gradient norms for core and spurious subnetworks scale based on this definition and the results in the main paper.

Proposition 2 *Assume $f_{\mathbf{w}}$ is bias-decomposable network. If $f_{\mathbf{w}}$ predicts according to the spurious decision rule,* i.e. *$b = \arg\max_j f_{\mathbf{w}}(\mathbf{x})[j]$, then for some $\alpha > 0$, as the logit-scaling factor $k \to \infty$:*

$$\frac{\sum_{\mathbf{x},y\in\Omega} \|\nabla_{\mathbf{s}}\ell(y, kf_{\mathbf{w}}(\mathbf{x}))\|}{\sum_{\mathbf{x},y\in\Omega} \|\nabla_{\mathbf{c}}\ell(y, kf_{\mathbf{w}}(\mathbf{x}))\|} = \mathcal{O}(e^{\alpha k}), \tag{8}$$

for some $\alpha > 0$, where $\|\cdot\|$ stands for Frobenius norm.

Proof 2 *The softmax probability π_j for a class j is given by:*

$$\pi_j = \frac{e^{z_j}}{\sum_i e^{z_i}} \quad \text{where } z_i = kf_w(\mathbf{x})[i] \tag{9}$$

With some abuse of notation, the gradient of the loss ℓ can be expressed as:

$$\nabla_c \ell = k \sum_j (\pi_j - \delta_{jy}) \nabla_c f_c(\mathbf{x})[j] \tag{10}$$

$$\nabla_s \ell = k \sum_j (\pi_j - \delta_{jy}) \nabla_s f_s(\mathbf{x})[j] \tag{11}$$

where δ_{jy} is the Kronecker delta, defined as:

$$\delta_{jy} = \begin{cases} 1 & \text{if } j = y \\ 0 & \text{if } j \neq y \end{cases} \tag{12}$$

For bias-conflicting samples, i.e. $\mathbf{x} \in \Omega_{bc}$, as $k \to \infty$, the softmax probability $\pi_b \to 1$ and $\pi_j \to 0$ for $j \neq b$. The gradient for the core feature subnetwork converges to:

$$\begin{aligned} \nabla_c \ell &\approx k\left(\nabla_c f_c[b] - \nabla_c f_c[y]\right) && (13)\\ &\approx 0, && (14)\end{aligned}$$

with the latter due to Definition 1. Note that for the spurious feature subnetwork Eq. (13) *implies that the gradient norm scales linearly with k:*

$$\|\nabla_s \ell_{bc}\| = \mathcal{O}(k) \tag{15}$$

For the correctly classified bias-aligned samples with margin α, the gradient norms for both subnetworks are scaled by this vanishing factor:

$$\|\nabla_c \ell_{ba}\| = \mathcal{O}(ke^{-k\alpha}) \tag{16}$$

$$\|\nabla_s \ell_{ba}\| = \mathcal{O}(ke^{-k\alpha}) \tag{17}$$

As we sum the norms over the minibatch $\Omega = \Omega_{bc} \cup \Omega_{ba}$, the total spurious gradient norm (numerator) is:

$$\sum_{\Omega} \|\nabla_s \ell\| = \sum_{\Omega_{bc}} \mathcal{O}(k) + \sum_{\Omega_{ba}} \mathcal{O}(ke^{-k\alpha}) = \mathcal{O}(k) \tag{18}$$

The total core gradient norm (denominator) is:

$$\sum_{\Omega} \|\nabla_c \ell\| = \sum_{\Omega_{bc}} 0 + \sum_{\Omega_{ba}} \mathcal{O}(ke^{-k\alpha}) = \mathcal{O}(ke^{-k\alpha}) \tag{19}$$

The final ratio is the ratio of their asymptotic behaviors:

$$\frac{\sum_{\Omega} \|\nabla_s \ell\|}{\sum_{\Omega} \|\nabla_c \ell\|} = \frac{\mathcal{O}(k)}{\mathcal{O}(ke^{-k\alpha})} = \mathcal{O}(e^{\alpha k}) \tag{20}$$

Note that although the decomposition in question is an idealization, it follows in line of previous work that demonstrate such modularity [32], and similar decompositions have been utilized in related research previously [60]. However, note that our assumption regarding the stability of the core vs. spurious subnetworks differ from that of [60], who assume a Bayes-optimal, stable spurious subnetwork. Our assumption is motivated by our empirical observations. In Fig. 18 (top), we examine two ResNet18 models trained on Colored MNIST dataset, and investigate their learned representations, based on the pooled outputs of the last layer convolution filters. We compute the CSI - BSI values for each filter/neuron at different points in the training. Assume we categorize neurons for which CSI > BSI as "class-dominant", and "bias-dominant" otherwise. We then ask, "At iteration t, what percentage of the neurons that were class dominant *remained* class-dominant by the end of training, and what percent of the neurons that were bias-dominant *remained* so?" As the results show, for high learning rate ($\eta = 0.1$ vs. $\eta = 0.001$), after iteration ~ 10 the overwhelming majority (not infrequently 100%) of the neurons who were class-dominant remained so. This is not true for the bias-dominant neurons. Note that this is not explained solely by the final ratio of class-dominant neurons: 55.4% vs 85.9% for the two models respectively, which falls short of explaining the behavior observed. To provide a more microscopic examination of this, in Fig. 18 (bottom) we examine how neuron-specific gradients impact spurious feature utilization (computed through feature attribution) within that neuron, from a high LR ($\eta = 0.1$) experiment within our FCN + Colored MNIST setting. The updates clearly show that the increasing spurious feature utilization is "reset" by a large gradient update, driven presumably by mispredictions from bias-conflicting samples. We consider further examination of these phenomena as an important future research direction.

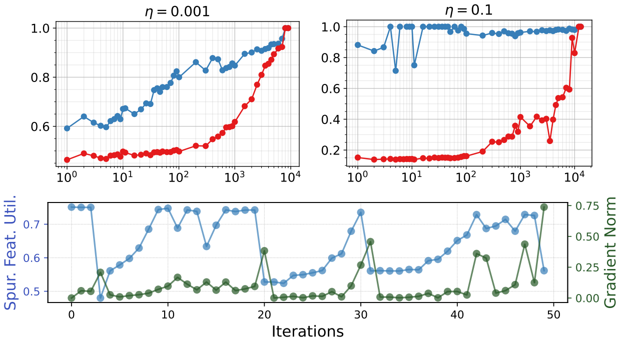


Figure 18. (Top) Ratio of class-dominant vs. bias-dominant neurons that survived as such by the end of training. (Bottom) Examining the impact of large gradient updates on feature attributions.

D.3. Value Ranges for Figures

Given that our figures depict multiple variables at the same time, and the results are normalized according to experiments to illuminate the patterns that LR and other interventions create, we present the min. and max. values the independent and dependent variables take in Tab. 1 and Tab. 2.

D.4. Additional Experiment Results

Here we present additional experimental results that were omitted in the main paper due to space concerns. Fig. 15 present our results with the synthetic moon-star dataset, Fig. 16 presents our results with the Corrupted CIFAR-10 dataset, Fig. 19 presents our results with the Waterbirds dataset, and Fig. 20 includes additional results for comparing the effects of various hyperparameters and regularization methods. Moreover, Fig. 21 and Fig. 22 provide a more in-depth look at the performance of models in terms of unbiased test accuracy under various pruning ratios, using column and magnitude pruning respectively.

To show that our results are not limited to training using SGD with a constant LR, we present qualitatively identical experiment results in Fig. 23 using the Colored MNIST dataset and ResNet18 model, where the initial learning rate (x-axis) is multiplied by 0.1 after 1000th iteration. Similar results using Adam optimization algorithm are presented in Fig. 24. Additionally, using the same setting, in Fig. 25 we

Table 1. Minimum and maximum values for each dataset-model combination included in our main experiments.

Dataset	Model	LR		Test Acc.		Prunability		Act. Comp.		Avg. CSI		Class Sep.	
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
MNIST-CIFAR	FCN	0.001	0.01	35.247	35.369	0.930	0.940	0.191	0.215	0.108	0.136	0.13	0.143
MNIST-CIFAR	CNN	0.001	0.2	24.507	41.717	0.326	0.902	0.173	0.513	0.079	0.225	0.049	0.149
MNIST-CIFAR	ResNet18	0.001	0.25	26.233	47.513	0.311	0.737	0.294	0.343	0.225	0.371	0.115	0.255
MNIST-CIFAR	ResNet50	0.005	0.1	23.287	34.358	0.378	0.515	0.25	0.34	0.143	0.25	0.065	0.143
MNIST-CIFAR	Wide ResNet101-2	0.005	0.1	24.855	33.537	0.418	0.522	0.252	0.34	0.166	0.247	0.082	0.149
CelebA	Swin Transformer	1e-07	0.0001	40.889	48.21	0.344	0.609	0.265	0.584	0.122	0.305	0.06	0.164
Colored MNIST	FCN	0.001	0.5	72.727	76.11	0.935	0.968	0.285	0.389	0.243	0.381	0.291	0.387
Colored MNIST	CNN	0.01	0.35	88.2	91.48	0.532	0.931	0.372	0.643	0.302	0.576	0.376	0.692
Colored MNIST	ResNet18	0.001	0.25	68.343	91.543	0.252	0.771	0.381	0.527	0.38	0.63	0.212	0.635
Colored MNIST	ResNet50	0.001	0.1	55.687	86.38	0.284	0.489	0.334	0.524	0.198	0.458	0.07	0.509
Colored MNIST	ResNet50 (PSGD)	1e-6	0.01	24.16	93.09	0.249	0.959	0.334	1.0	0.159	0.888	0.126	0.712
Colored MNIST	Wide ResNet101-2	0.001	0.25	61.643	85.25	0.258	0.559	0.337	0.787	0.215	0.473	0.084	0.569
Moon-Star	FCN	0.01	0.75	74.315	81.156	0.954	0.971	0.375	0.457	0.257	0.34	0.2	0.326
Cor. CIFAR-10	FCN	0.0001	0.005	46.92	51.93	0.574	0.847	0.202	0.252	0.131	0.183	0.169	0.182
Cor. CIFAR-10	CNN	0.001	0.1	43.555	48.503	0.39	0.861	0.189	0.341	0.144	0.387	0.14	0.329
Cor. CIFAR-10	ResNet18	0.001	0.5	35.153	47.795	0.267	0.803	0.382	0.634	0.266	0.393	0.115	0.205
Cor. CIFAR-10	ResNet50	0.001	0.1	36.3	45.52	0.391	0.556	0.334	0.475	0.203	0.304	0.081	0.134
Cor. CIFAR-10	Wide ResNet101-2	0.001	0.25	37.827	47.153	0.399	0.559	0.339	0.653	0.215	0.348	0.09	0.189
Double MNIST	FCN	0.001	0.5	69.287	72.47	0.986	0.999	0.236	0.406	0.22	0.485	0.428	0.481
Double MNIST	CNN	0.005	0.4	83.987	94.57	0.383	0.916	0.235	0.772	0.187	0.476	0.204	0.55
Double MNIST	ResNet18	0.001	0.1	85.44	95.577	0.285	0.706	0.306	0.365	0.467	0.602	0.485	0.776
Double MNIST	ResNet50	0.001	0.1	42.045	95.733	0.228	0.498	0.16	0.256	0.152	0.392	0.172	0.68
Double MNIST	Wide ResNet101-2	0.001	0.1	43.12	96.08	0.195	0.508	0.161	0.298	0.156	0.416	0.188	0.703
Parity	FCN	0.01	0.75	55.31	85.663	0.56	0.782	0.333	0.679	0.032	0.152	0.008	0.136

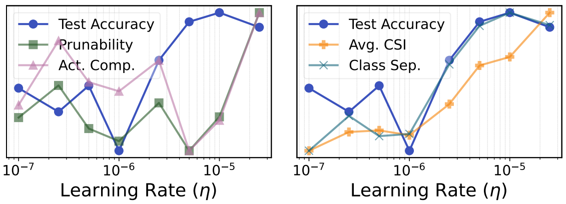


Figure 19. (Left) Effects of learning rate on OOD performance (unbiased test acc.), network prunability, and representation properties with the Waterbirds dataset.

show that training models for longer according to an additional criterion (CE loss $< 1e-5$) produces qualitatively identical results as test accuracy changes very little beyond convergence for both low and high LR models. Finally, Fig. 26 demonstrates that alternative choices to characterize parameter and representation compressibility, such as (q, κ)-Compressibility, sparsity, and the recently proposed PQ-Index [13] produce qualitatively identical results.

Optimizers & LR schedules. We confirm our findings on robustness to SCs and comp. extend to modern and standard training setups. Fig. 29 (left) shows that core benefits persist with ResNet50 (Colored MNIST) using the PSGD (Kron) optimizer and a WSD LR schedule. Fig. 29 (right) shows that our key findings also hold under a standard CIFAR-100 setup with AdamW, cosine annealing, weight decay, and validation set based model selection, addressing concerns about reliance on constant LR SGD.

Fig. 30 (left) compares ResNet18 models trained on CIFAR-10 with a starting LR of 0.1, which is decreased to 0.001 by step S. We plot the eventual unbiased test performance of a model as function of S, with the performance of a model trained by a constant LR of 0.1 vs. 0.001 depicted as horizontal lines for reference. The results show that the effects of LR are almost completely integrated by $S = 1000$. Fig. 30 (right) shows that creating a "model soup" [81] is another way of obtaining robustness vs. compressibility disentanglement.

E. Additional Details and Results Regarding Neural Network Attribution

One of the most commonly used methods include Integrated Gradients (IG) [70]. Given a predictor f, an input $\mathbf{x}$, and a baseline input $\mathbf{x}'$, IG for the i'th component of $\mathbf{x}$ is computed as follows:

$$\mathrm{IG}_i(\mathbf{x}) := (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial f(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial x_i} d\alpha.$$

Intuitively, this corresponds to integrating the sensitivity of the output to changes in x_i throughout linear interpolation from x' to x. See [70] for a justification of IG's methodology, and see [50] for strengths and weaknesses of various attribution methods. To investigate whether our results are an artifact of using IG as our attribution method, we visually compare the attributions computed by Integrated Gradients (IG) and another prominent attribution method, DeepLift

Table 2. Minimum and maximum values for each dataset-model combination included in our regularization experiments.

Dataset	**Model**	**HP**		**Test Acc.**		**Prunability**		**Avg. CSI**	
		Min.	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**
Colored MNIST	Learning Rate	0.001	0.15	72.297	92.203	0.262	0.8	0.441	0.644
Colored MNIST	High LR + L2 Reg.	1e-05	0.001	92.183	94.25	0.79	0.87	0.651	0.848
Colored MNIST	Batch Size	0.01	0.3333	68.363	90.303	0.253	0.661	0.383	0.587
Colored MNIST	Momentum	0.1	0.99	68.38	91.197	0.256	0.748	0.387	0.633
Colored MNIST	L1 Regularization	1e-05	0.001	72.887	90.347	0.275	0.864	0.458	0.859
Colored MNIST	L2 Regularization	0.0001	0.05	71.727	89.687	0.263	0.76	0.432	0.873
Colored MNIST	γ (Focal Loss)	0.001	25.0	68.203	91.8	0.254	0.645	0.377	0.626
Colored MNIST	ρ (ASAM)	0.01	10.0	68.723	90.977	0.256	0.522	0.38	0.508
MNIST-CIFAR	Learning Rate	0.001	0.1	24.327	47.497	0.293	0.69	0.24	0.376
MNIST-CIFAR	High LR + L2 Reg.	1e-06	0.0001	46.947	48.327	0.677	0.727	0.384	0.497
MNIST-CIFAR	Batch Size	0.01	0.3333	25.95	46.24	0.29	0.529	0.201	0.262
MNIST-CIFAR	Momentum	0.1	0.99	25.887	46.397	0.29	0.666	0.226	0.354
MNIST-CIFAR	L1 Regularization	1e-05	0.001	24.65	46.097	0.28	0.845	0.24	0.486
MNIST-CIFAR	L2 Regularization	0.0001	0.075	24.18	47.13	0.287	0.619	0.189	0.533
MNIST-CIFAR	γ (Focal Loss)	0.01	25.0	25.87	40.393	0.283	0.597	0.223	0.314
MNIST-CIFAR	ρ (ASAM)	0.01	10.0	26.3	34.053	0.286	0.769	0.225	0.261
Double MNIST	Learning Rate	0.001	0.25	88.13	96.39	0.282	0.801	0.529	0.631
Double MNIST	High LR + L2 Reg.	1e-05	0.001	96.68	97.75	0.664	0.89	0.641	0.836
Double MNIST	Batch Size	0.01	0.3333	85.39	96.62	0.279	0.514	0.375	0.471
Double MNIST	Momentum	0.1	0.99	85.58	95.71	0.278	0.755	0.471	0.57
Double MNIST	L1 Regularization	1e-06	0.001	87.64	95.87	0.28	0.932	0.517	0.914
Double MNIST	L2 Regularization	0.0001	0.05	86.83	97.08	0.285	0.727	0.504	0.896
Double MNIST	γ (Focal Loss)	0.001	25.0	85.01	95.89	0.275	0.661	0.456	0.58
Double MNIST	ρ (ASAM)	0.01	10.0	85.71	95.82	0.277	0.626	0.467	0.488

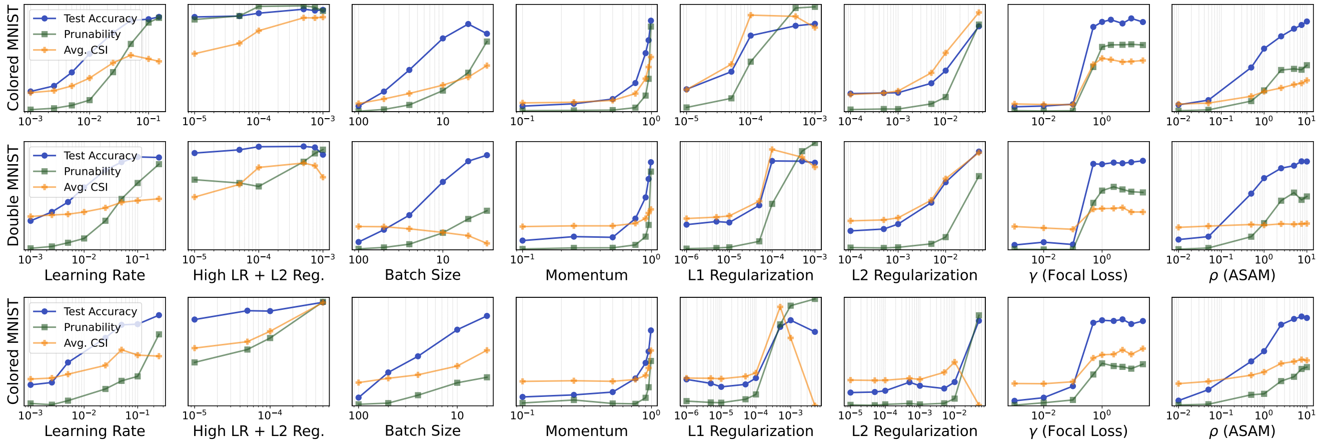


Figure 20. Comparing various hyperparameters, regularization methods, and losses in terms of OOD robustness, compressibility, and core feature utilization in Double MNIST dataset with a ResNet18 model (top), and Colored MNIST dataset with a ResNet50 model (bottom). y-axes are normalized within each figure for each variable.

[68], for CIFAR-10 samples under ResNet18 models in Fig. 14. The two methods produce identical results for the purposes of this paper. Both methods are implemented using the `captum` package for `PyTorch` framework [33].

We can utilize attribution methods for convergent validation of class-selectivity index as a measure of spurious feature utilization. Although in datasets such as Colored MNIST pixels for spurious and core features overlap, they

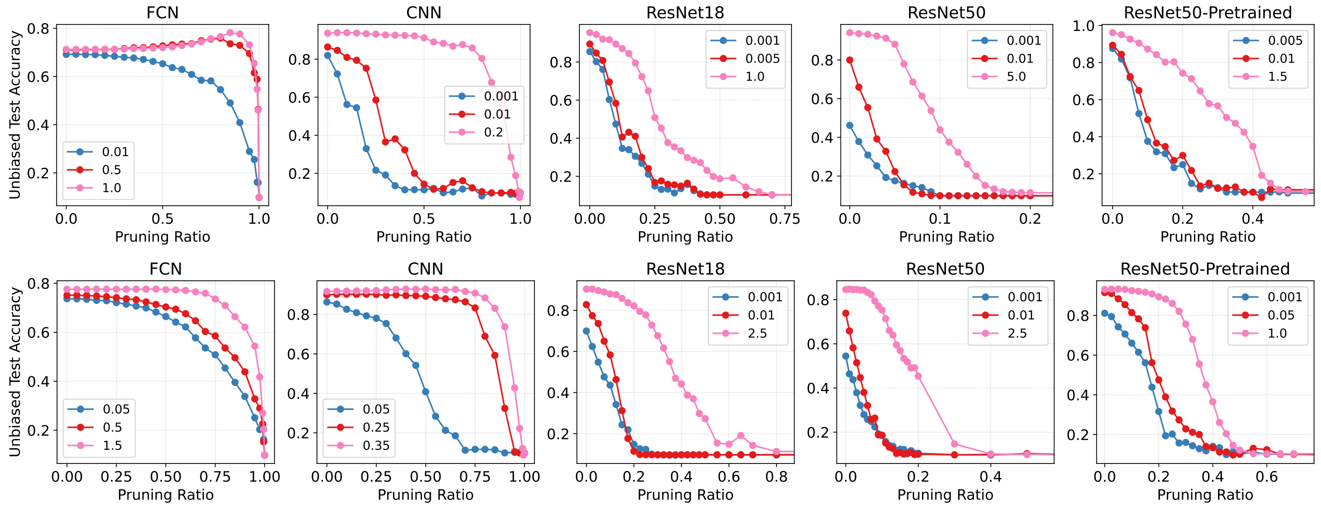


Figure 21. Effects of column pruning on models trained on Double MNIST (top) and Colored MNIST (bottom) datasets, under various learning rates. x-axes are modified for visualization.

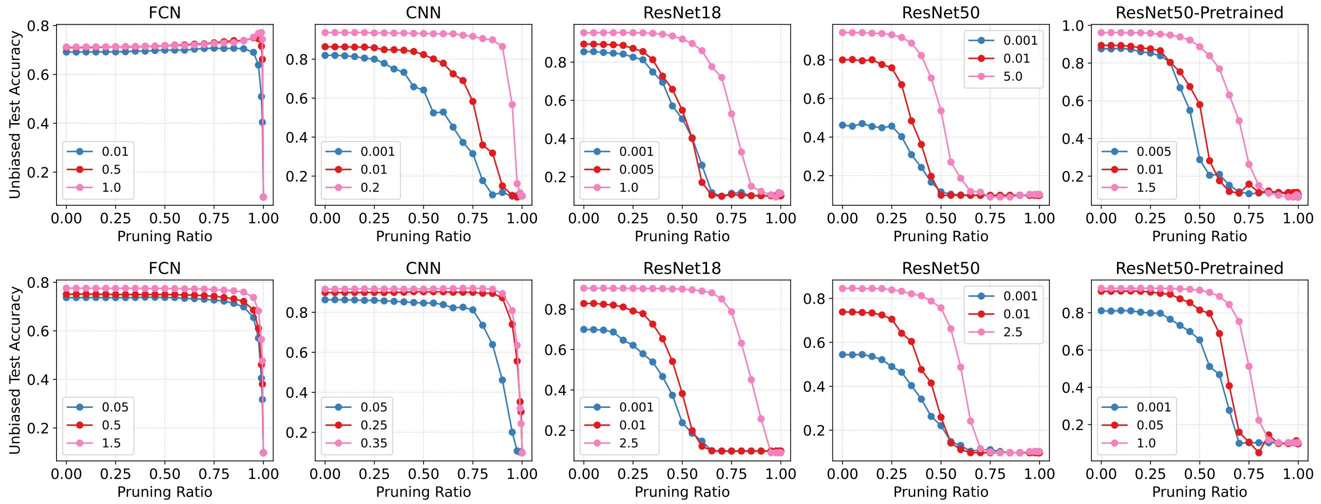
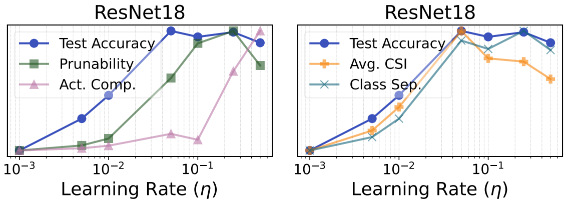


Figure 22. Effects of magnitude pruning on models trained on Double MNIST (top) and Colored MNIST (bottom), under various LRs.



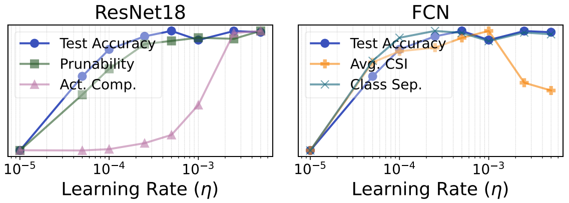


Figure 23. Effects of learning rate on OOD performance (unbiased test acc.), network prunability, and representation properties with a learning rate annealing setting, where the LR is multiplied by 0.1 after 1000th iteration.

Figure 24. Effects of learning rate on OOD performance (unbiased test acc.), network prunability, and representation properties with an Adam optimizer, with $\beta_1 = 0.9, \beta_2 = 0.999$.

are distinct in others such as Double MNIST. Thus, we can compute input attribution on Double MNIST and through normalization we can determine how much (*i.e.* what percentage) of models' attribution is on the spurious vs. core feature. We can then see whether the patterns demonstrated by CSI parallel that computed through input attribution. Fig. 27 shows a comparison of the two metrics across five datasets and LRs for Double MNIST dataset. Remarkably, the two demonstrate qualitatively identical patterns, confirming CSI as a useful metric of core feature utilization.

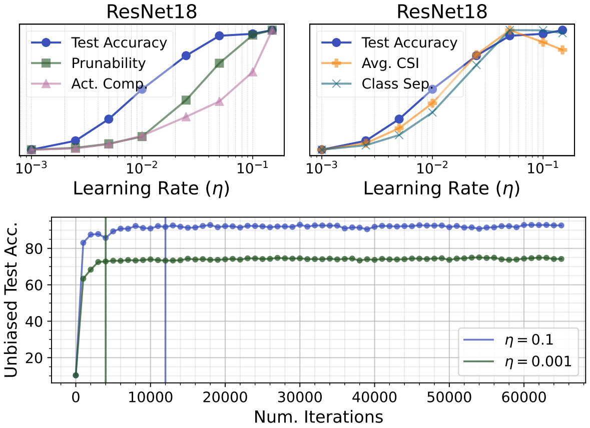


Figure 25. (Top) Effects of learning rate on OOD performance (unbiased test acc.), network prunability, and representation properties when trained for 100% training accuracy and < 0.00001 training loss. (Bottom) Test accuracy does not meaningfully change beyond convergence (vertical lines correspond to the point where 100% was reached).

Creation of attribution maps for CIFAR datasets. We train a ResNet18 model using a low vs. high LR with *10 different seeds* on CIFAR-10 and CIFAR-100 datasets. Then, we extract those samples in the test set which have been correctly predicted by $> .75$ of the high LR models and $< .25$ of low LR ones. Then, we investigate the attribution maps of low vs. high LR models in Fig. 8.

E.1. Additional Attribution Visualizations

We provide additional visualization of attributions for our experiments in the main paper; for Colored MNIST (Fig. 31), MNIST-CIFAR (Fig. 32), Double MNIST (Fig. 33), CelebA (Fig. 34), CIFAR-10 (Fig. 35), and CIFAR-100 (Fig. 36) datasets. Notice that as in the main paper, low LR models are more likely to focus on spurious features compared to high LR models.

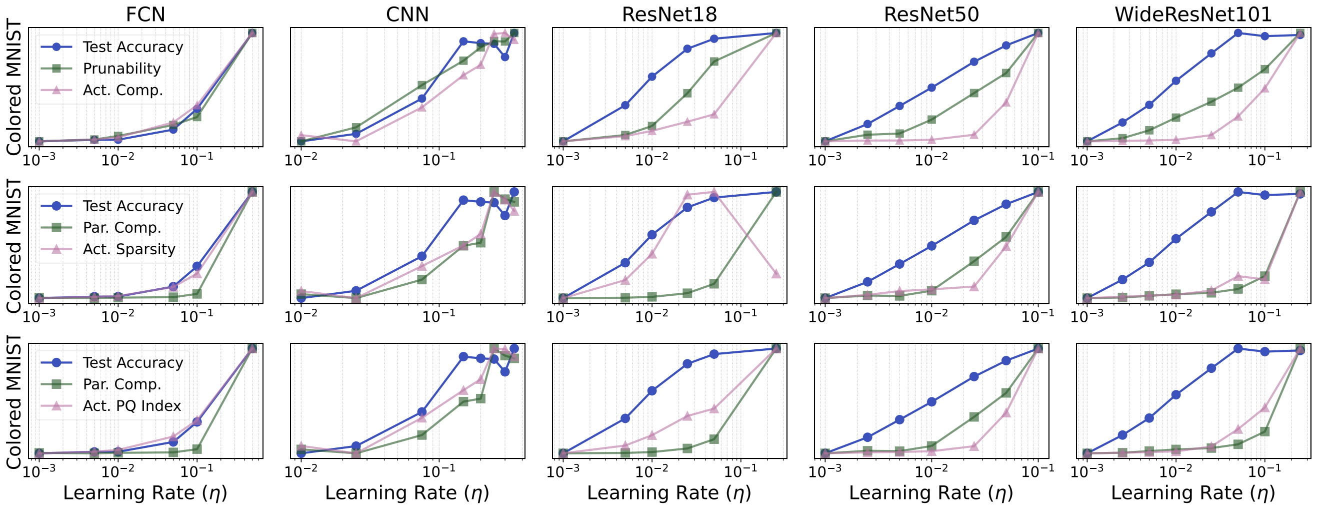


Figure 26. Utilizing alternative notions of parameter and representation compressibility such as prunability, (q, κ)-Compressibility (with $q = 2, \kappa = 0.1$), sparsity, and the recently proposed PQ-Index (with $p = 2, q = 1$).

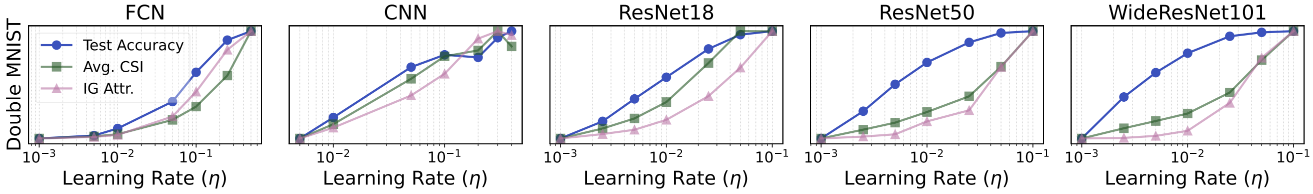


Figure 27. Comparing CSI vs. input attribution to core features (%), using Integrated Gradients.

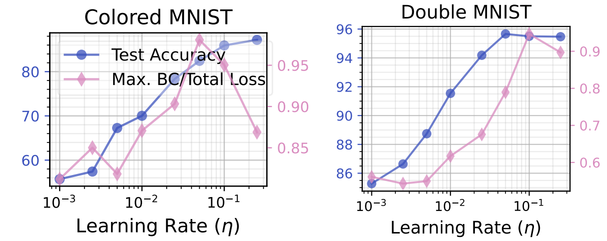


Figure 28. Examining the effect of LR on unbiased test accuracy and BC loss ratio in ResNet18 and Double MNIST dataset, as well as ResNet50 and Colored MNIST dataset.

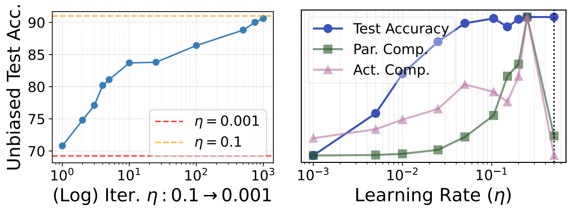


Figure 30. (Left) Unbiased test set performance as a function of LR reduction. (Right) Disentangling compressibility and robustness through a "model soup" (the rightmost model).

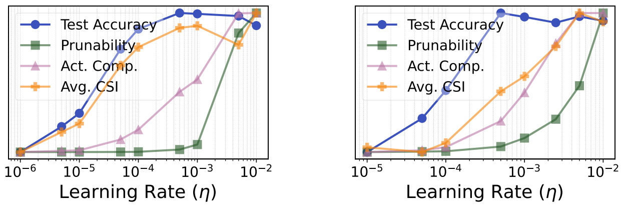


Figure 29. Experiments with alternative optimizers, schedulers, and convergence criteria.

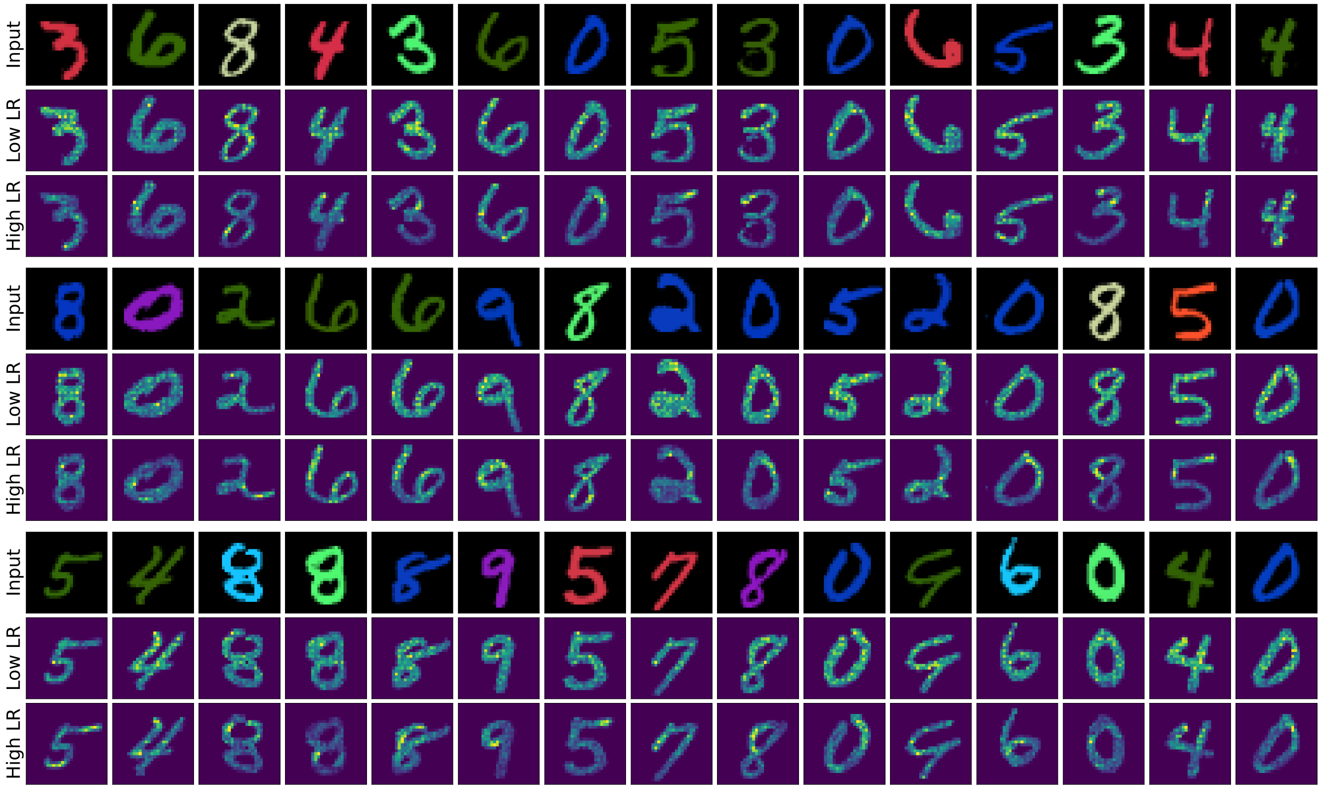


Figure 31. Attributions of trained ResNet18 models on Colored MNIST dataset.

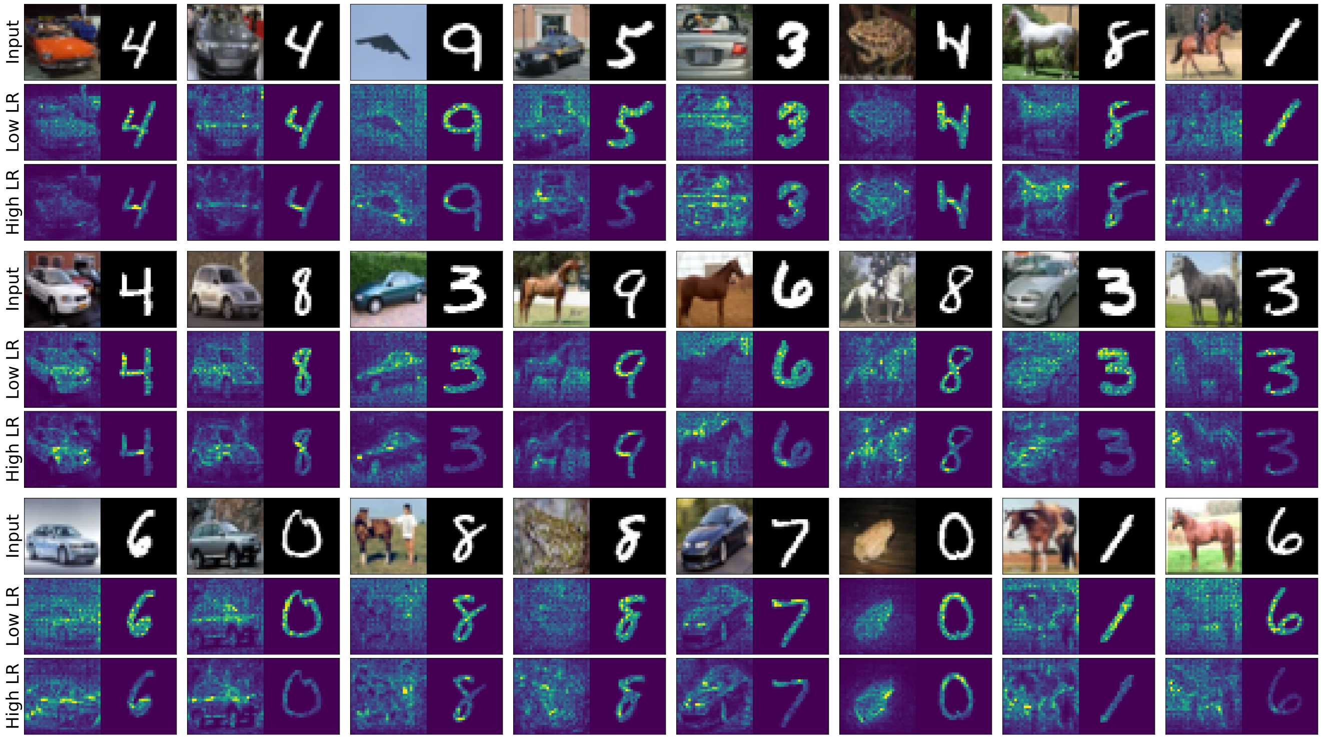


Figure 32. Attributions of trained ResNet18 models on MNIST-CIFAR dataset.

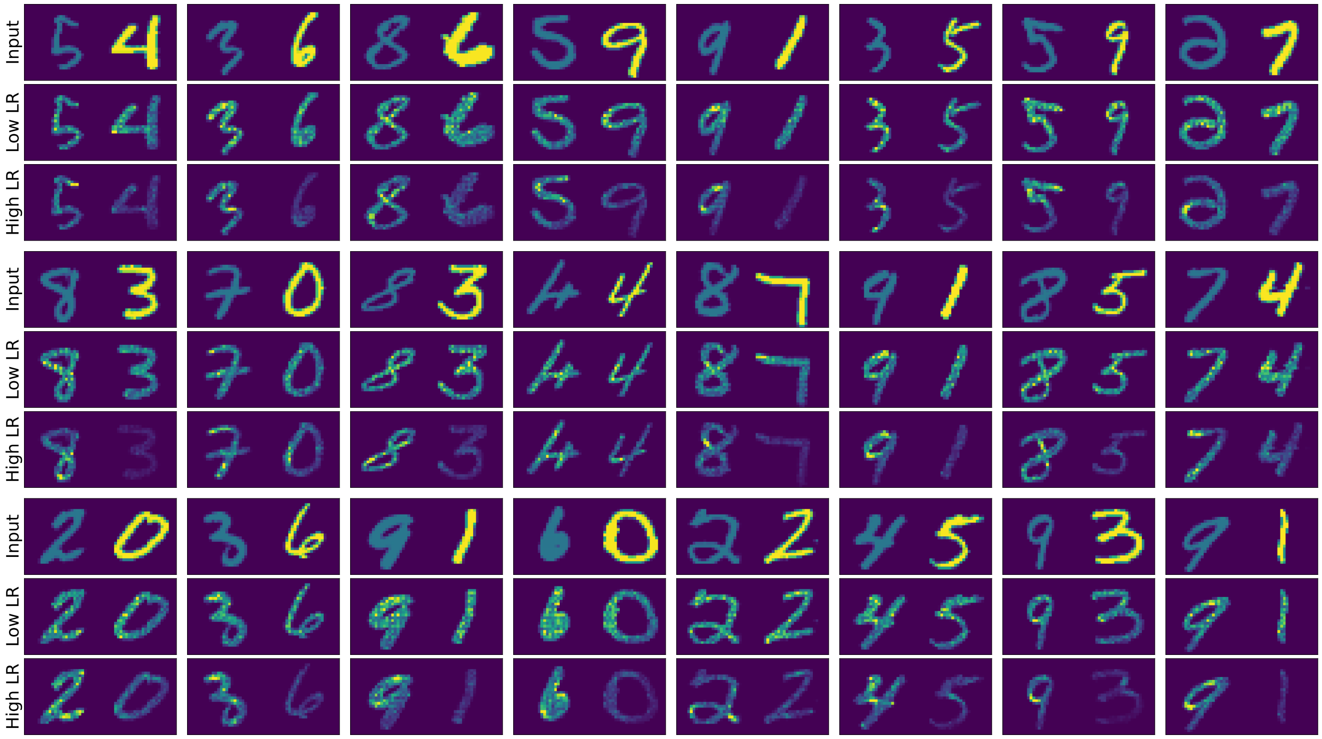


Figure 33. Attributions of trained ResNet18 models on MNIST-CIFAR dataset.

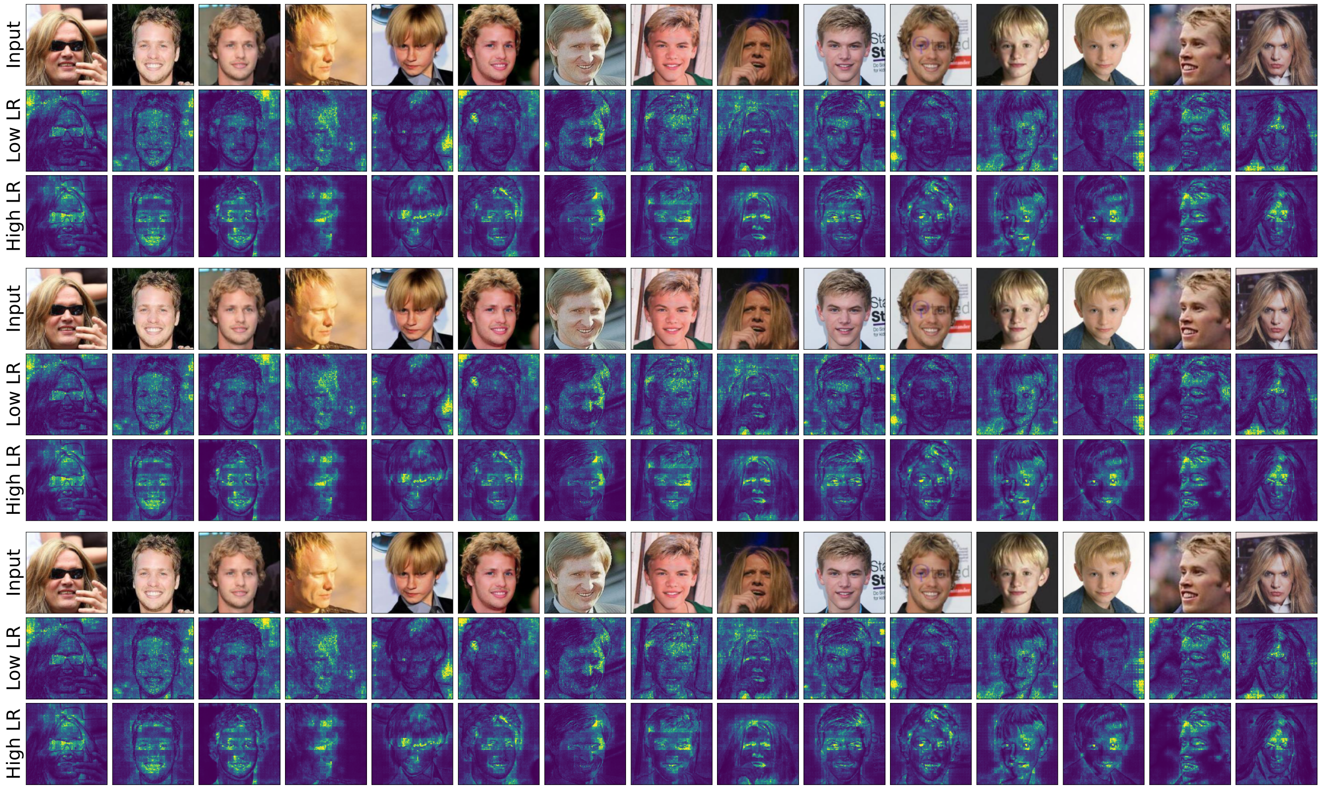


Figure 34. Attributions of trained Swin Transformer models on CelebA dataset.

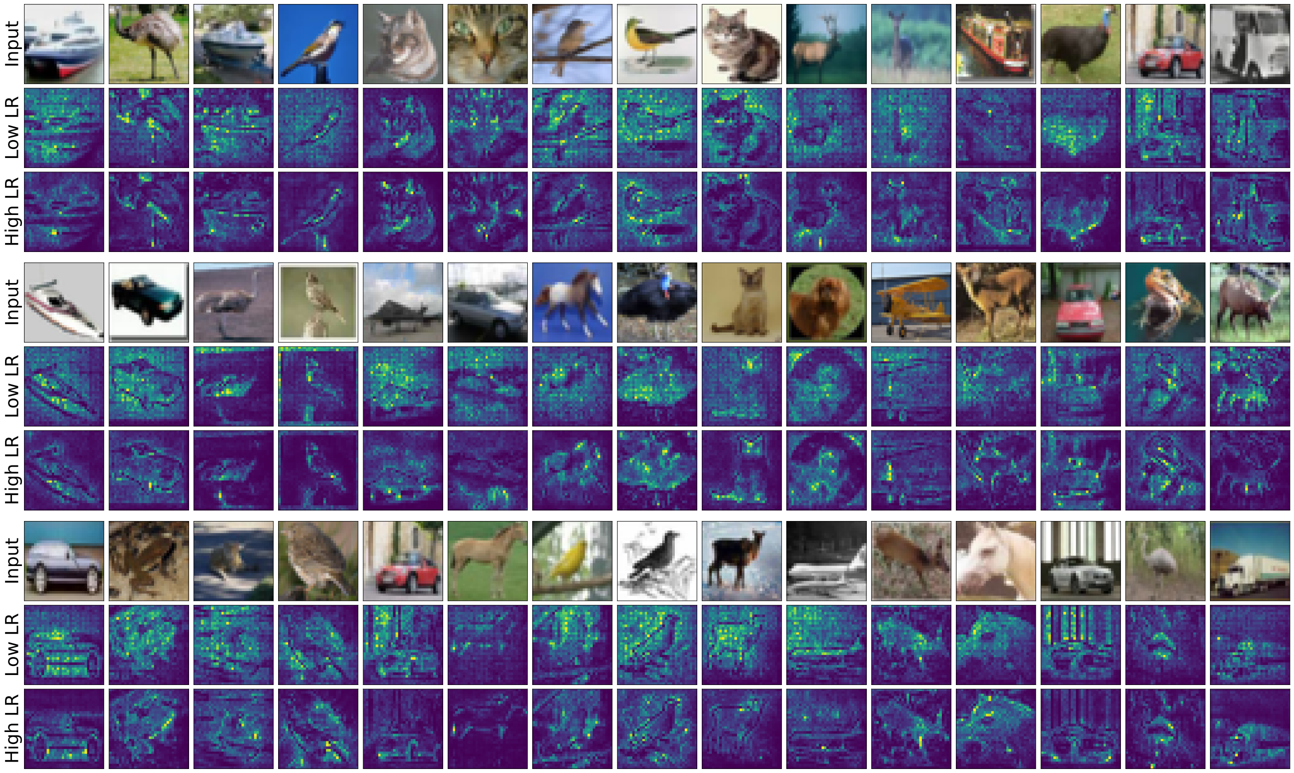


Figure 35. Attributions of trained ResNet18 models on CIFAR-10 dataset.

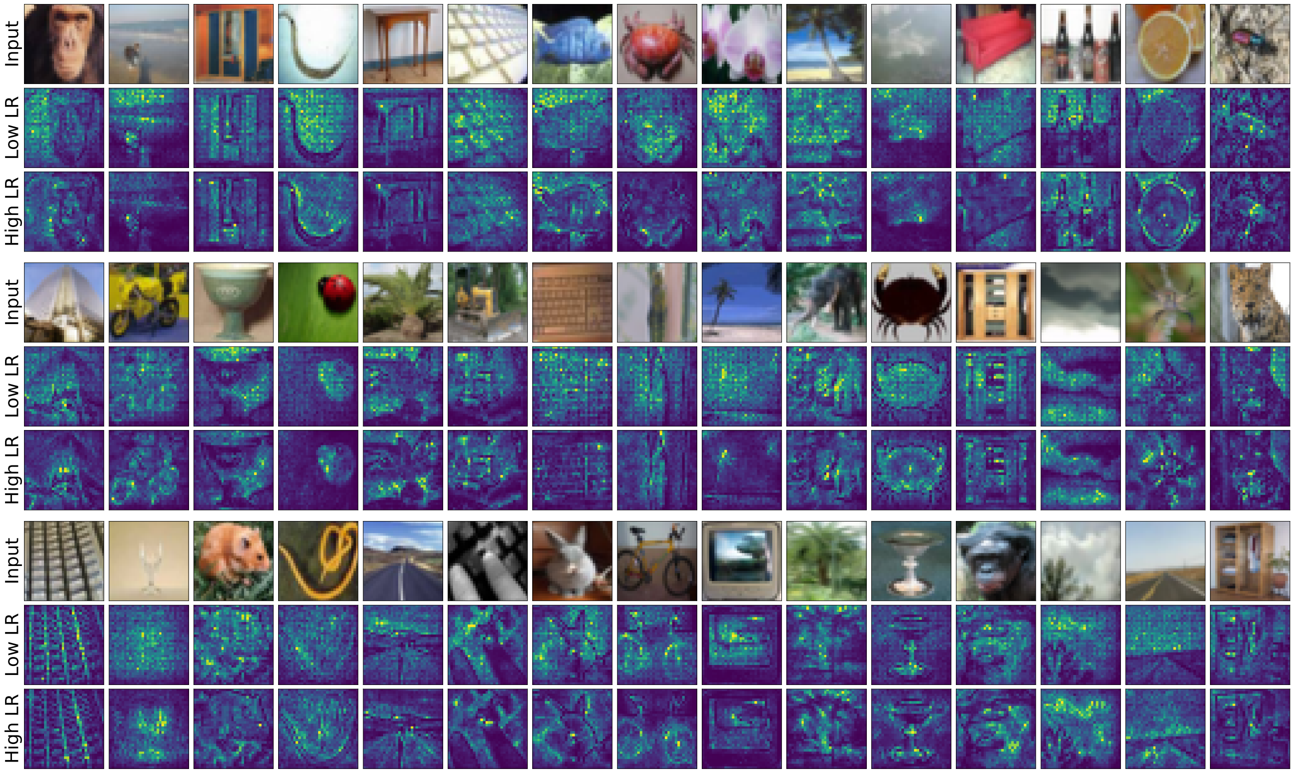


Figure 36. Attributions of trained ResNet18 models on CIFAR-100 dataset.