Reproducibility study of "Bilinear MLPs enable weight-based mechanistic interpretability"

Anonymous authors
Paper under double-blind review

Abstract

This paper presents a reproducibility study of "Bilinear MLPs enable weight-based mechanistic interpretability" by Pearce et al. (2024), which proposes that bilinear architectures possess intrinsic interpretability properties accessible via eigenvalue decomposition. We verify the core empirical image classification claims. Our results confirm the findings for image classification: bilinear layers consistently exhibit an interpretable low-rank structure where the leading eigenvectors capture the majority of task-relevant information, allowing for significant truncation without performance loss. Furthermore, we validate that these eigenstructures are stable across random initializations and varying model sizes. Additionally, we explore extensions to the original work, demonstrating that adversarial training (specifically PGD) enhances the interpretability of eigenvector features on MNIST. Finally, we explored generalization on more complex RGB datasets, such as CIFAR-10 and CIFAR-100, which have generated eigenvectors with uninterpretable structures. All our code is publicly available.[1]

1 Introduction

Mechanistic interpretability aims to understand how neural networks implement their computations by analyzing the structure and semantics of their learned parameters (Olah et al., 2017; 2020; Cammarata et al., 2020; Elhage et al., 2021), or relying on post-hoc or activation-based explanations (Montavon et al., 2018; Ribeiro et al., 2016; Petsiuk et al., 2018; Simonyan et al., 2013). Within this line of work, *Bilinear MLPs Enable Weight-Based Mechanistic Interpretability* (Pearce et al., 2024) uses a neural architecture that is intrinsically interpretable because their computations can be expressed in terms of linear operations with a third order tensor.

The paper introduces bilinear multi-layer perceptrons (MLPs) whose hidden layers compute structured interactions between learned projections (Chrysos et al., 2021; Li et al., 2017; Lin et al., 2015; Sharkey, 2023). This structure enables the model's weights to be analyzed via low-rank decompositions, allowing the authors to extract interpretable features directly from the learned parameters. The approach is evaluated on image classification tasks, where the authors argue that a small number of dominant eigenvectors capture most of the model's behavior and correspond to semantically meaningful features.

This approach aligns with the theme of transparency, as it seeks to understand model behavior through intrinsic properties of the learned parameters, rather than model-based or post-hoc explanations.

This paper presents a reproducibility study of the core empirical image classification claims made in the original work. Our focus is on verifying the reported low-rank structure, stability, and interpretability of bilinear MLPs across tasks. In addition, we explore extensions that test whether these interpretability properties persist under modifications to the training procedure.

In particular, we investigate the effect of adversarial training on weight-based interpretability. Adversarial training is known to substantially alter learned representations and has been widely studied as a means of improving model robustness (Goodfellow et al., 2014; Madry et al., 2017; Ilyas et al., 2019). While prior work

[1] `https://anonymous.4open.science/r/reproduced-mech-inter-image-class`

has examined the relationship between robustness and learned features, these analyses have largely relied on saliency methods (Simonyan et al., 2013), feature visualization (Olah et al., 2017), or other activation-based approaches (Montavon et al., 2018). Bilinear MLPs provide a unique opportunity to study this relationship directly in weight space through eigenvalue decomposition of the learned parameters. We therefore investigate whether adversarial training affects the low-rank structure and interpretability of bilinear representations.

We focus on two adversarial training methods, FGSM (Goodfellow et al., 2014) and PGD (Madry et al., 2017), which represent different robustness regimes. FGSM generates single-step perturbations, whereas PGD constructs stronger adversarial examples through iterative optimization. We hypothesize that if adversarial training encourages models to rely on more robust features, this effect should be reflected in the eigenstructure of bilinear MLPs, leading to more interpretable eigenvectors and a greater concentration of task-relevant information in the dominant spectral components.

2 Scope of reproducibility

A central contribution of the paper is a weight-based interpretability method that analyzes the bilinear layer parameters directly, rather than relying on post-hoc explanations or activation-based probes. By constructing a symmetric bilinear tensor from the learned weights and performing eigenvalue decomposition, the authors identify a small number of dominant eigenvectors that capture most of the model's behavior. These eigenvectors are shown to correspond to visually and semantically meaningful features, providing a mechanistic interpretation of the model's internal representations.

In this work, we aim to reproduce the key image classification claims made in the original paper concerning the structure, stability, and interpretability of bilinear MLPs. The original paper evaluates bilinear MLPs across both language and image domains. In this reproducibility study, we focus exclusively on the image classification setting. Rather than attempting a partial reproduction across multiple modalities, we chose to investigate a single experimental domain in greater depth. This allows us not only to verify the original claims regarding low-rank structure, eigenvector stability, and interpretability, but also to examine how these properties behave under additional training interventions and dataset shifts.

In addition, we extend the original experiments by evaluating the behavior of bilinear MLPs under adversarial perturbations and by testing generalization to additional image classification datasets. We consider two standard adversarial methods, FGSM and PGD, as they represent qualitatively different robustness regimes (Goodfellow et al., 2014; Madry et al., 2017). FGSM is a single-step method that applies a local linear perturbation (Goodfellow et al., 2014), whereas PGD performs multi-step optimization and produces stronger adversaries (Madry et al., 2017). This distinction allows us to examine whether changes in robustness are reflected in the learned eigenstructure and interpretability of bilinear representations.

This study will verify the following claims of the paper:

- Claim 1 (Image Classification): Eigenvector decomposition of bilinear MLP weights across image classification tasks reveals an interpretable low-rank structure.
- Claim 2 (Image Classification): Truncating lower-magnitude eigenvalue components has minimal impact on classification performance, indicating that most task-relevant information is captured by the leading eigenvectors.
- Claim 3 (Image Classification): The eigenvectors of bilinear MLPs are stable across random initializations and behave similarly across different model sizes.
- Claim 4 (Image Classification): Input noise regularization reduces overfitting and produces bilinear MLPs with eigenvector features that are more interpretable under eigenvalue analysis.

Beyond reproducing the original claims, we investigate the following extensions:

- Extension 1: We evaluate how adversarial training using FGSM (Goodfellow et al., 2014) and PGD (Madry et al., 2017) affects both adversarial robustness and the interpretability of eigenvector features learned by bilinear MLPs on MNIST.

- Extension 2: We test whether the observed low-rank structure, eigenvector stability, and truncation robustness generalize to CIFAR-10 (Krizhevsky et al., 2009) and CIFAR-100 (Krizhevsky et al., 2009).

Our extensions are therefore confined to the image domain. This design choice enables a controlled investigation of how robustness and dataset complexity affect weight-based interpretability while maintaining direct comparability with the reproduced image-classification results. Furthermore, image classification provides a particularly suitable setting for this analysis because the learned eigenvectors can be visualized directly, allowing qualitative observations to be complemented by quantitative evaluation.

3 Methodology

To replicate the study, we utilized the public repository provided by the authors.[2] This included code for the models, helper functions for loading the datasets, and tools for visualizing the eigenvectors. All the experiments for replicating the image classification study were run on the CPU (with the exception of extension 2, which was run on a GPU).

3.1 Model descriptions

3.1.1 Image Models

The image classification models in this study consist of three parts: an embedding, the bilinear layer, and the classification head/unembedding, as shown in Figure 1.

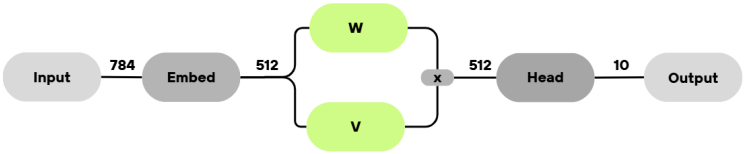


Figure 1: The architecture of the used image models.

The parameterization of the image models depend on the used dataset. Table 1 summarizes the parameter count per component for each dataset we have used.

Table 1: Architecture and parameter counts for bilinear MLP models across datasets. All models use one bilinear layer with hidden dimension $d_h = 512$.

Component	MNIST	CIFAR-10	CIFAR-100
Input dimension (d_{in})	784	3,072	3,072
Number of classes (d_{out})	10	10	100
Embedding	401,408	1,572,864	1,572,864
Bilinear Layer	524,288	524,288	524,288
Classification Head	5,120	5,120	51,200
Total Parameters	**930,816**	**2,102,272**	**2,148,352**

3.2 Datasets

MNIST and Fashion-MNIST The MNIST (LeCun, 1998) and Fashion-MNIST (Xiao et al., 2017) datasets consist of 70,000 grayscale images of handwritten digits (0-9) and fashion products from 10 categories, respectively. Each of size 28×28 pixels. The datasets are split into 60,000 training images and 10,000 test images, with a balanced distribution of approximately 6,000 training examples and 1,000 test examples

[2] `https://github.com/tdooms/bilinear-decomposition`

per class. We use the standard train/test split without additional validation set partitioning. Images are normalized to the range $[0, 1]$ by dividing pixel values by 255.

CIFAR-10 and CIFAR-100 The CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009) are classic benchmarks in image classification (Liu et al., 2018). They contain 60,000 color images across 10 and 100 classes, respectively, with each image having a size of $32 \times 32 \times 3$ pixels. Both datasets provide 50,000 training images and 10,000 test images. CIFAR-10 contains exactly 5,000 training examples and 1,000 test examples per class, while CIFAR-100 contains 500 training examples and 100 test examples per class. We use the standard train/test split. Preprocessing consists of converting the images to tensors, adding Gaussian noise, and normalizing each RGB channel to zero mean and unit variance using $\mu = 0.5$ and $\sigma = 0.5$ per channel.

3.3 Hyperparameters

Unless otherwise stated, all hyperparameters were taken directly from the implementation released by Pearce et al. (2024), and kept fixed across experiments in order to closely match the original study. No global hyperparameter optimization was performed beyond the variations explicitly described below.

For Claim 3, we evaluate the consistency of learned eigenvectors across training runs by varying the hidden size used to initialize the bilinear layer of the image models. We use the hidden sizes $\{30, 50, 100, 300, 500, 1000\}$, following the experimental setup of the original paper (Pearce et al., 2024).

For the CIFAR-10 and CIFAR-100 extension, a limited hyperparameter search was conducted over the number of bilinear hidden layers, considering models with 1, 2, and 3 layers. All other hyperparameters, including learning rate, batch size, optimizer, weight decay, and number of training epochs, were held constant.

3.4 Experimental setup and code

All experiments were implemented in PyTorch and conducted using mini-batch training with shuffled batches. Image classification models were trained for a fixed number of epochs using the AdamW optimizer, and performance was evaluated using classification accuracy on the standard test split of each dataset.

For adversarial training experiments, adversarial examples were generated on-the-fly during training using either FGSM or PGD, and training batches were augmented with these adversarial samples as will be described in Section 3.5. Adversarial robustness was evaluated by measuring test accuracy on adversarially perturbed inputs generated using a particular attack.

3.5 Extension

Extension 1 We evaluate how adversarial training affects both the adversarial robustness and the interpretability of learned eigenvector features in bilinear MLPs. Specifically, we train models on MNIST using two standard adversarial training methods: Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) and Projected Gradient Descent (PGD) (Madry et al., 2017).

During training, we augment each mini-batch with adversarially perturbed examples generated on-the-fly. For FGSM training, we generate adversarial examples using:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)) \tag{1}$$

where x is the original input, L is the cross-entropy loss, θ represents model parameters, y is the true label, and ϵ is the perturbation size. We use $\epsilon = 0.3$, following Goodfellow et al. (2014).

For PGD training, we use an iterative attack with 40 steps:

$$x_{\text{adv}}^{t+1} = \text{Proj}_\epsilon(x_{\text{adv}}^t + \alpha \cdot \text{sign}(\nabla_x L(\theta, x_{\text{adv}}^t, y))) \tag{2}$$

where $\alpha = \frac{2.5\epsilon}{40}$ is the step size and Proj_ϵ projects the perturbation back into the ℓ_∞ ball of radius ϵ centered at the original input x. We follow the configuration from Madry et al. (2017).

The training objective combines losses on clean and adversarial examples:

$$\mathcal{L}_{\text{total}} = \alpha L(\theta, x, y) + (1 - \alpha) L(\theta, x_{\text{adv}}, y) \tag{3}$$

where we use $\alpha = 0.5$, following Goodfellow et al. (2014).

To properly evaluate this extension, we train four image models:

1. Baseline: The baseline image model as depicted in Figure 1, following Pearce et al. (2024).
2. Baseline + Regularization: The baseline image model, where additionally the input image is augmented with random gaussian noise using $\sigma = 0.15$, following Pearce et al. (2024).
3. Baseline + Regularization + FGSM Training: The baseline image model including gaussian noise regularization, where additionally we train on adversarial images of the batch of images created by FGSM (Goodfellow et al., 2014).
4. Baseline + Regularization + PGD Training: The baseline image model including gaussian noise regularization, where additionally we train on adversarial images of the batch of images created by PGD (Madry et al., 2017).

Extension 2 We evaluate how well do the observations of Pearce et al. (2024) on **MNIST** and **Fashion-MNIST** datasets transfer to color-rich and content-rich **CIFAR-10** and **CIFAR-100** datasets (Liu et al., 2018). Both datasets have 32×32 pixels in the spatial dimensions compared to 28×28 and have 3 color channels (*red, green, blue*) compared to grayscale. Both of these datasets are very common benchmarks when evaluating image classification and enable us to explore how well does weight-based mechanistic interpretability translate to more complex problems.

In the evaluation, we observe the performance of models using bilinear layers on these datasets and the structure of their eigenvalues when employing increasing levels of Gaussian noise. This is applied on the image using the following formula

$$x_{\text{noisy}} = x + \sigma \cdot \epsilon, \text{where } \epsilon \sim \mathcal{N}(0, I)$$

where σ is the standard deviation.

Furthermore, during preliminary research, the models trained on **CIFAR-100** dataset were performing poorly, barely reaching the accuracy of 0.25. For this reason, a hyperparameter sweep was conducted with the following parameters: coefficient of Gaussian noise $\sigma \in \{0, 0.4\}$, number of layers $l \in \{1, 2, 3\}$, and number of epochs $n \in \{20, 100\}$. Only after these experiments was the variation of CIFAR-100 trained with the same σs as CIFAR-10.

3.6 Computational Requirements

The majority of our replication experiments (Claims 1–4) and Extension 1 were executed entirely on consumer-grade CPUs, requiring a minimum of 4 cores and 8 GB of RAM. Running these baseline evaluations sequentially takes approximately 2.5 hours. Extension 2 introduces color-rich images and hyperparameter sweeps over multiple layers and epochs on CIFAR-10/100, which scales up the computational demands. To keep training times viable, Extension 2 requires an 8-core CPU or a dedicated GPU, taking approximately 4 hours to complete. Overall, the entire suite of experiments can be reproduced sequentially in less than 8 hours on standard personal hardware. A detailed breakdown of the exact hardware platforms, core counts, and individual runtimes for each experiment is provided in Appendix D.

4 Results

Overall, the work of Pearce et al. (2024) is mostly reproducible. Claims 1 and 2 are fully reproducible and yield results comparable to the original paper. However, Claims 3 and 4 are only reproducible if cosine

similarity is replaced by absolute cosine similarity and the L_2 norm is replaced by the standard deviation. Regarding the extensions, Extension 1 indicates that PGD improves interpretability while FGSM yields poor interpretability; Extension 2 indicates that CIFAR-10 and CIFAR-100 produce structures that are not significantly interpretable.

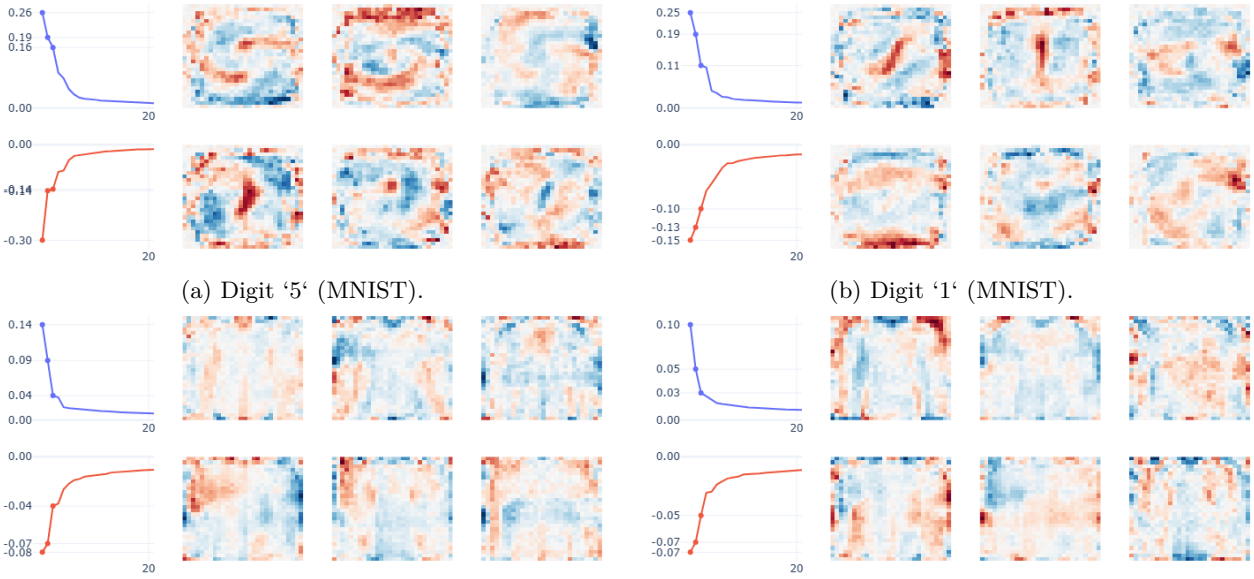
4.1 Results reproducing original paper

4.1.1 Results - Claim 1

Eigenvalue Spectra Reveal Low-Rank Structure The eigenvalue decomposition of the bilinear MLP weights for the image classification task reveals an interpretable, low-rank structure. Figure 2 plots the sorted eigenvalues by magnitude for two representative classes from MNIST (digit '1' and '5') and two from Fashion-MNIST ('Shirt' and 'Pullover').

All four spectra exhibit a sharp, exponential drop-off where a small number of leading eigenvalues contain the majority of the spectral mass, while the rest cluster near zero.

Therefore, our experimental results on the image classification model provide consistent support for the claim.



(a) Digit '5' (MNIST). (b) Digit '1' (MNIST).

(c) 'Shirt' (Fashion-MNIST). (d) 'Pullover' (Fashion-MNIST).

Figure 2: Eigenvalue spectra for bilinear MLP interaction matrices across tasks, ordered by eigenvalue magnitude.

4.1.2 Results - Claim 2

We tested the claim by evaluating the classification accuracy of models on the MNIST and Fashion-MNIST datasets when their bilinear layers were truncated to use only the top k eigenvectors per digit or clothing item.

Spectral Concentration For each output class, we calculated the proportion of the total spectral mass (sum of squared eigenvalues, $\sum \lambda_i^2$) contained within the top 5 eigenvalues. As shown in Table 2, the top 5 eigenvalues capture a considerable proportion of spectral mass, ranging from **16.41% to 60.25% for MNIST** (mean: 41.26%) and **21.78% to 66.09% for Fashion-MNIST** (mean: 40.71%). This distribution shows substantial variance across classes: some classes (e.g., MNIST digit 1 at 59.65%, Fashion-

MNIST sandal at 66.09%) exhibit stronger spectral concentration, while others (e.g., MNIST digit 8 at 16.41%, Fashion-MNIST coat at 21.78%) show more diffuse spectral structure. The results indicate that there are differences between classes in amount of information stored, but the top eigenvealues do manage to contain a significant percentage, suggesting a low-rank structure.

Table 2: Proportion of spectral mass (based on squared eigenvalues) contained within the top 5 eigenvectors for each class.

Class	MNIST	Fashion-MNIST
0	35.96%	46.97% (T-shirt/top)
1	59.65%	27.24% (Trouser)
2	54.76%	37.40% (Pullover)
3	41.04%	39.64% (Dress)
4	35.30%	21.78% (Coat)
5	46.80%	66.09% (Sandal)
6	34.48%	56.17% (Shirt)
7	60.25%	39.24% (Sneaker)
8	16.41%	31.57% (Bag)
9	27.94%	41.02% (Ankle boot)

Impact on Classification Accuracy The top eigenvectors manage to preserve most of the task-relevant information for classification. Table 3 shows the model's accuracy when using only the top k eigenvectors.

Using only the top 5 eigenvectors results in a small decrease in accuracy. This demonstrates that the vast majority of lower-magnitude components are not critical for the classification task. Interestingly, however, using the top 10 and 15 eigenvectors not only recovers the full model's accuracy, but exceeds it by **1-2 percentage points**. This suggests that the remaining eigenvectors may primarily encode noise or artifacts that slightly hinder generalization. Their removal has a regularizing effect, improving model performance.

Therefore, truncation experiments confirm that "truncating has minimal impact", while also holding the claim that "leading eigenvectors capture most task-relevant information."

Table 3: Test accuracy on MNIST and Fashion-MNIST when truncating bilinear layers to the top k eigenvectors.

Dataset	Top k	Accuracy	Drop from Full Model
MNIST	5	93.03%	-1.96%
	10	96.20%	**+1.21%**
	15	96.43%	**+1.44%**
Full Model (MNIST)	All (512)	94.99%	–
Fashion-MNIST	5	83.18%	-0.23%
	10	85.25%	**+1.84%**
	15	85.37%	**+1.96%**
Full Model (Fashion-MNIST)	All (512)	83.41%	–

4.1.3 Result - Claim 3

Related to Figure 3, Claim 3, i.e., the eigenvectors of bilinear MLPs are stable across random initializations and behave similarly across different model sizes, holds if not the cosine similarity but rather its absolute value is used. In this case, the associated main findings in Pearce et al. (2024) hold, i.e., "Both the ordering and contents of top eigenvectors are very consistent across runs", "The cosine similarities of the top eigenvector are between 0.8 and 0.9 depending on size", "increasing the model sizes results in more similar top eigenvectors". Figure 3 shows for the MNIST dataset and Figure 4 shows for the FMNIST dataset.

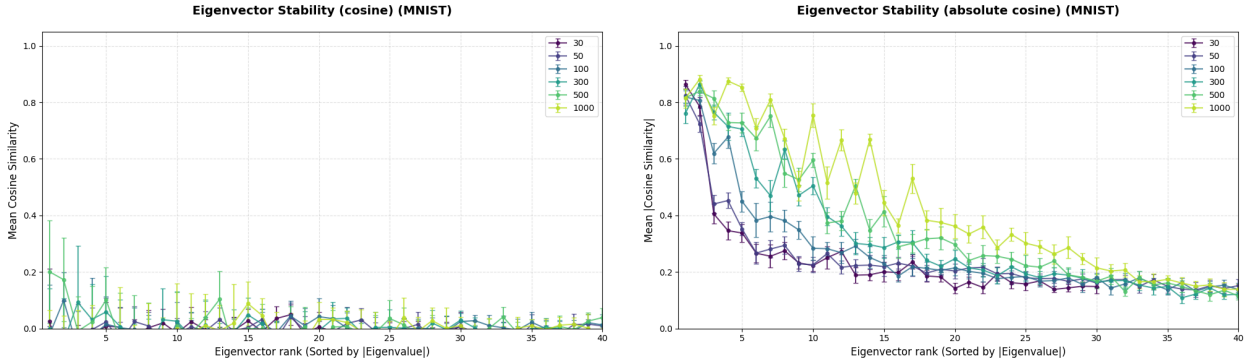


Figure 3: Eigenvector stability, for the MNIST dataset, for cosine similarity and absolute cosine similarity respectively.

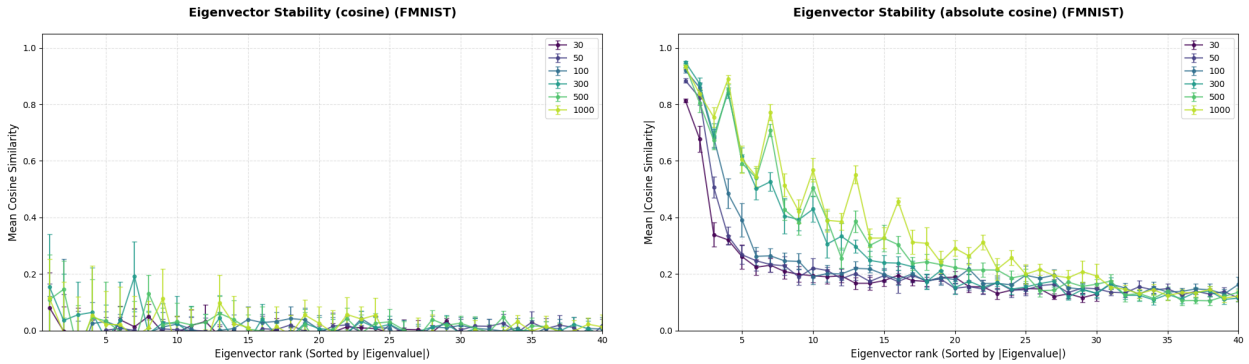


Figure 4: Eigenvector stability, for the FMNIST dataset, for cosine similarity and absolute cosine similarity respectively.

4.1.4 Result - Claim 4

Related to Figure 5, Claim 4, i.e., input noise regularization reduces overfitting and produces bilinear MLPs with eigenvector features that are more interpretable under eigenvalue analysis, holds if standard deviation is used and not Pearce et al. (2024)'s (L2) norm, (Pearce et al. (2024) did not mention what kind of norm was used). Also, the associated main findings in Pearce et al. (2024) hold, i.e., "the eigenvectors of unregularized models focus on certain outlying pixels," and "Increasing the scale of the added noise results produce more digit-like eigenvectors." Figure 5 shows for the MNIST data set and Figure 6 shows for the FMNIST data set.

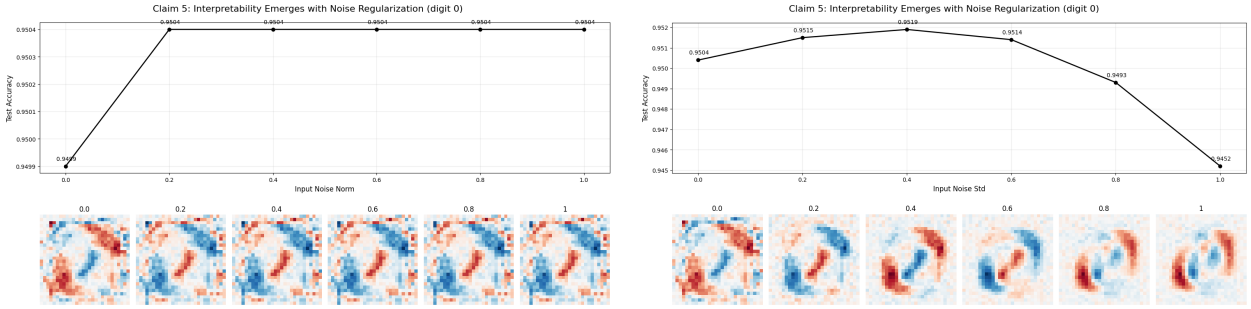


Figure 5: For the MNIST dataset, noise regularization, i.e., noise scaled to a specific L2 norm and noise scaled with a specific standard deviation respectively.

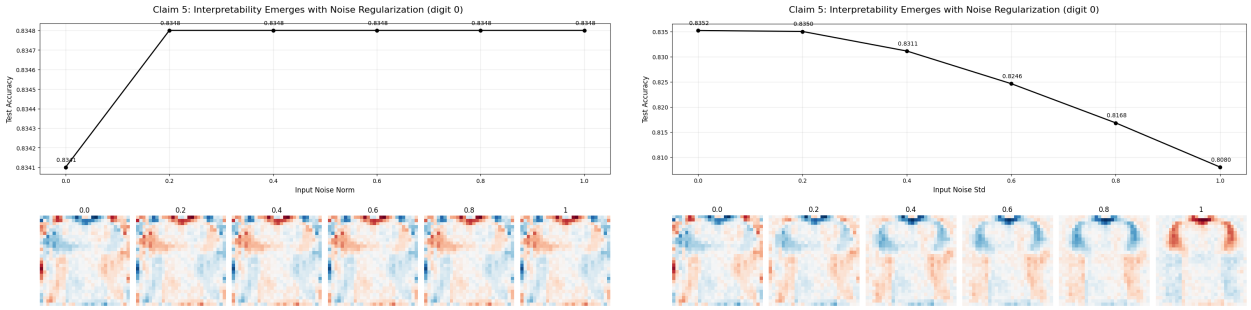


Figure 6: For the FMNIST dataset, noise regularization, i.e., noise scaled to a specific L2 norm and noise scaled with a specific standard deviation respectively.

4.2 Extensions

4.2.1 Extension using adversarial training - FGSM and PGD

For our first extension we investigated how adversarial training using FGSM (Goodfellow et al., 2014) and PGD (Madry et al., 2017) influenced adversarial robustness and accuracy, and how the interpretability of eigenvectors is influenced. In Table 4, we show the test accuracy, FGSM accuracy, and PGD accuracy across the different image models we have trained. We observe that the baseline model and the regularized model achieve high test accuracy, but almost 0% accuracy on an FGSM and PGD attack. Moreover, we can see that the regularized model with FGSM training still achieves a high test accuracy, while also achieving a much higher accuracy during an FGSM attack, albeit the PGD accuracy is still 0%. For the regularized model with PGD training, we achieve a robust model that is to some degree robust to adversarial attacks.

Table 4: Test accuracy and adversarial robustness across different models.

Model	**Clean Accuracy (%)**	**FGSM Accuracy (%)**	**PGD Accuracy (%)**
Baseline	96.5	0.02	0.13
Baseline + Regularization	96.6	0.20	0.01
Baseline + Regularization + FGSM Training	95.8	69.60	0.00
Baseline + Regularization + PGD Training	95.5	67.77	29.33

Next, we investigate the top positive eigenvector constructed from the weights of the learned models, across five different digits for all of our trained image models. Figure 8 (Appendix A), shows the top positive

eigenvector across five digits for all of our trained models. We observe that the eigenvector of the FGSM model is not as interpretable as it was before (for the regularized model), it seems to exhibit a checkerboard pattern across the pixels. However, it is interesting to note that there is some structure in the eigenvector. Generally, the pixels belonging to a particular digit seem to be white (no activation).

The eigenvector of the PGD model on the other hand, seems to be the most interpretable across all models we have trained. We can see that the eigenvector has a value of zero (white color) across pixels that do not belong to a specific part of a digit. The blue (positive) patches of pixels capture digit specific strokes, and the red (negative) patches capture pixels that would negatively affect the classification of the digit. This suggests that adversarial training using PGD helps the model to focus more on the important digit specific features, making the model more interpretable, while also retaining a high accuracy.

To further demonstrate the utility of adversarial training in weight-based interpretability, we analyzed for a misclassified example how an adversarial perturbation affects eigenvector activations, for details see Figure 9 (Appendix A). We can see that the original image activates eigenvectors associated with the true class (digit 4), in particular EV 252 (the top eigenvector), which we show in the last column of the first row. This specific eigenvector activates on vertical strokes resembling the digit 4. For the adversarial image on the other hand, we observe that EV 252 completely disappears from the top 10 eigenvectors, and instead EV 247 results in the highest activation, which we show in the last column of the second row. We can now see that this eigenvector captures strokes resembling the digit 0, hence resulting in a misclassification. This is confirmed by the middle plot of the last row, where we can see that the PGD perturbation causes the activation for the true class (digit 4) to vastly decrease, while the activations for the negative classes increase in general.

4.2.2 Extension on RGB datasets - CIFAR-10 and CIFAR-100

CIFAR-10 The models have been trained for 20 epochs with gradually increasing Gaussian noise because Pearce et al. (2024) shows that adding noise improves the discernibility of MNIST digits. We have observed a similar effect. The most positive eigenvalues without any noise activate mostly in the center and around it which can be seen in Figure 10a (Appendix B). Since CIFAR-10 is an RGB dataset we can observe individual RGB channels which have shown to often be complementary to each other as can be seen in Figure 10a with the class *frog*. Furthermore, very similar classes *automobile* and *truck* produce almost identical eigenvalues. Showing that even without noise these eigenvalues can reveal similarity between classes.

Increased noise has resulted in much more interesting eigenvalues. As can be seen in Figure 10b, many classes have now developed oscillating activations suggesting that they act like edge detectors on their most positive eigenvalue. And classes that do not show such activations have often developed them on their most negative eigenvalue instead, as can be seen in Figure 7. Introducing noise has caused the model to create very different eigenvalues for previously almost identical *automobile* and *truck*. This shows the level of impact that noise has on the model.

With increasing noise the eigenvalues were getting more and more influenced by the noise, and when using $\sigma = 5$ the eigenvalues resembled pure Gaussian noise as can be seen in Figure 10c. However with these levels of noise there is almost nothing left from the original image which can be seen in Table 5 which shows that the model's performance is only 0.151, however it also shows that the model performance does not decrease when reasonable amount of Gaussian noise is applied.

Table 5: CIFAR-10 and CIFAR-100 models test accuracy based on standard deviation of noise (trained for 20 and 60 epochs, respectively)

Standard deviation	CIFAR-10: Test Accuracy	CIFAR-100: Test Accuracy
0.0	0.499	0.213
0.2	0.505	0.240
0.4	0.504	0.256
0.6	0.485	0.255
1	0.440	0.224
2	0.348	0.160
5	0.151	0.079

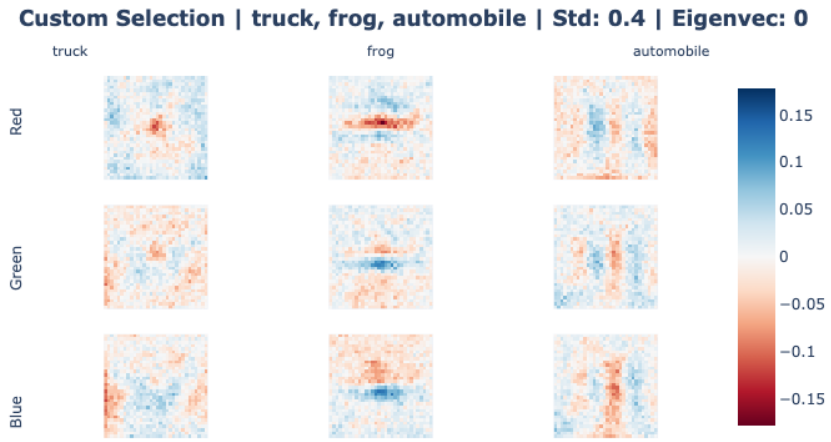


Figure 7: CIFAR-10 - most negative eigenvector (0) by channel and class with noise $\sigma = 0.4$

CIFAR-100 As a response to poor performance of CIFAR-100, an experiment has been conducted where models with 1, 2 and 3 layers were created and trained for 20 and 100 epochs before inspecting eigenvalues. These experiments have shown that increasing noise and epochs has increased the performance as can be seen in Table 6. However, validation accuracy has shown that the models have not improved further after 60 epochs. Therefore, since it has achieved the highest test accuracy of 0.258, a single-layered model has been trained for 60 epochs.

Table 6: CIFAR-100 performance with various number of layers and epochs

Standard deviation	Test Accuracy	Epochs	Layers
0.0	0.240	20	1
0.0	0.215	20	2
0.0	0.172	20	3
0.4	0.240	20	1
0.4	0.223	20	2
0.4	0.149	20	3
0.0	0.206	100	1
0.0	0.201	100	2
0.0	0.183	100	3
0.4	0.258	100	1
0.4	0.234	100	2
0.4	0.209	100	3

This has produced very similar results, where no noise produces centered activations but more dispersed than what was observed in **CIFAR-10** as can be seen in Figure 11a (Appendix B). Since **CIFAR-100** has to detect 100 classes, this is expected. When more noise is introduced again oscillating structures begin to appear as can be seen in Figure 11b. Further increasing the noise amplifies the structures as can be seen in Figure 11c.

5 Discussion

Overall, our experimental results largely support the central empirical claims made in the paper. Moreover, our extensions demonstrate that the proposed interpretability framework remains robust under adversarial training and generalizes, albeit with limitations, to more complex RGB datasets (such as the CIFAR-10 and CIFAR-100) datasets (e.g., the learned structures from CIFAR-10 and CIFAR-100 are not that interpretable).

Our adversarial training experiments provide insight into the generalizability of the original claims. We find that interpretability improves under PGD training, but degrades under FGSM, indicating that interpretability is sensitive to the choice of training objective. This could suggest that interpretability of bilinear MLPs is not solely a property of the architecture, but depends critically on the structure of the training signal. In particular, training procedures that enforce global robustness, such as PGD, appear to support more interpretable representations, whereas weaker or noisier constraints, such as FGSM, may hinder them.

5.1 What was easy

Reproducing the core experimental pipeline for the image classification tasks was relatively straightforward. The authors' publicly available codebase was modular, which allowed us to more quickly understand the architectures of the models and the visualization methods. In particular, the provided .decompose() method of the Model class in the image folder came in handy for inspecting and making judgements about eigenvectors and eigenvalues.

The experiments on MNIST and Fashion-MNIST were easier to reproduce. The default hyperparameters transferred well, requiring minimal tuning to achieve significant performance (approximately 97% accuracy) and to achieve eigenvector structures similar to the eigenvector structures of the paper.

5.2 What was difficult

One of the more challenging aspects of the reproduction of the experiments of the paper was that these experiments were not included in the paper's code. Thus, in order for us to reproduce these experiments, we inspected the descriptions of these experiments in the paper, but these descriptions were not completely straightforward, as some terms of phrases were not defined more concretely (e.g., in subsection 4.2, it was not completely straightforward what was meant by "norm"; it could have been L2 norm, L1 norm or possibly another norm).

One of the more challenging aspects related to the extensions involved scaling the experiments of the paper to CIFAR-10 and CIFAR-100. These datasets introduce higher input dimensionality, color channels, and increased semantic complexity, which significantly complicates optimization and eigenvector interpretation. Achieving stable training and interpretable eigenstructures required extensive hyperparameter tuning, particularly with respect to noise regularization strength, number of training epochs, and model depth.

5.3 Environmental Impact

The computational footprint of our experiments was moderate, as they were primarily conducted on the CPU. Table 7 (Appendix C) shows the emissions of each part of the project, measured in kg CO_2e, followed by the amount of power (measured in W) used by the CPU, GPU and RAM. Training times ranged from minutes for MNIST and Fashion-MNIST to several hours for CIFAR-based experiments and adversarial training.

Nevertheless, adversarial training and extensive hyperparameter sweeps substantially increase computational cost, highlighting an environmental trade-off between robustness, interpretability, and sustainability. Future work could investigate more computationally efficient training procedures or approximate adversarial methods to reduce energy usage while maintaining interpretability benefits.

5.4 Communication with original authors

For this study, we did not contact the original authors. Although the paper's ambiguity made it difficult to determine the correct cosine similarity and the correct norm, these issues were resolved empirically, allowing all claims to be successfully reproduced.

References

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: circuits. *Distill*, 5(3):e24, 2020.

Grigorios G Chrysos, Stylianos Moschoglou, Giorgos Bouritsas, Jiankang Deng, Yannis Panagakis, and Stefanos Zafeiriou. Deep polynomial neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 44(8):4021–4034, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 2079–2087, 2017.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1457, 2015.

Ling Liu, Yanzhao Wu, Wenqi Wei, Wenqi Cao, Semih Sahin, and Qi Zhang. Benchmarking deep learning frameworks: Design considerations, metrics and beyond. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1258–1269. IEEE, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Michael T Pearce, Thomas Dooms, Alice Rigg, Jose M Oramas, and Lee Sharkey. Bilinear mlps enable weight-based mechanistic interpretability, 2024. *URL https://arxiv. org/abs/2410.08417*, 2024.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Lee Sharkey. A technical note on bilinear layers for interpretability. *arXiv preprint arXiv:2305.03452*, 2023.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A FGSM and PGD Extension

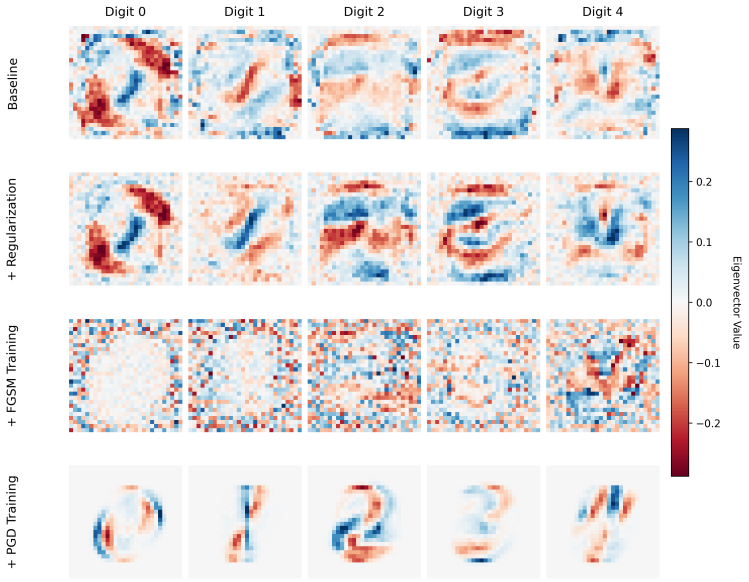


Figure 8: The top positive eigenvector across five different digits of the MNIST dataset, for all of our trained image models.

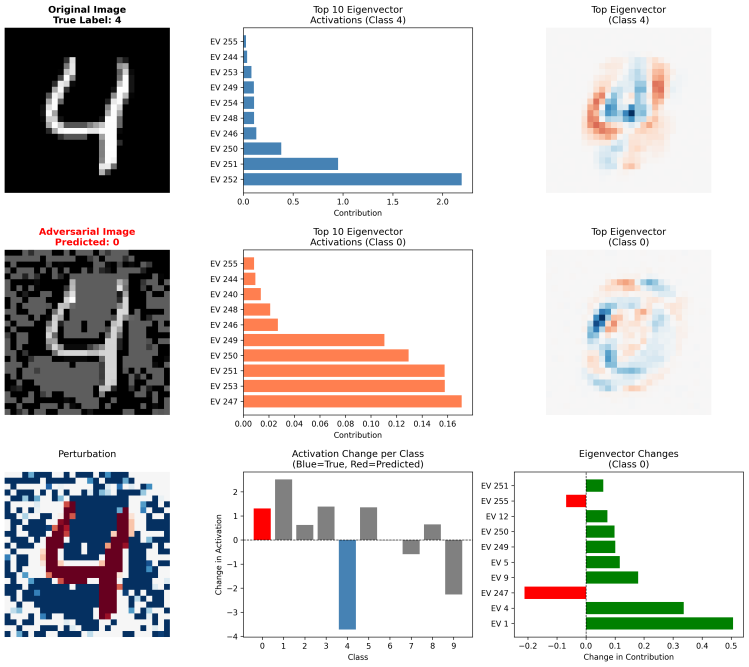
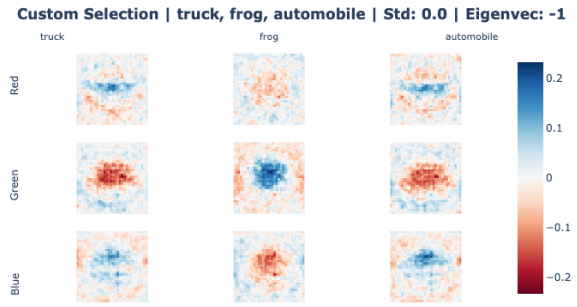
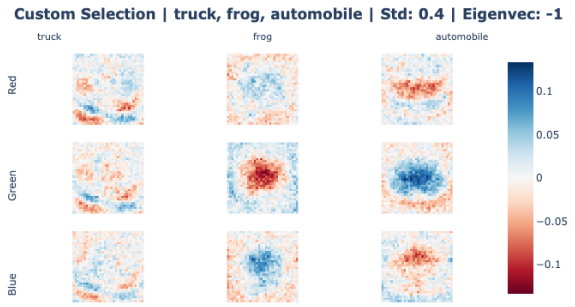


Figure 9: Comparison of the top 10 eigenvector activations and top eigenvector between an original image and adversarial image, and change in eigenvector activations per class for a PGD perturbation. This figure shows how a PGD perturbation changes the eigenvector activations, resulting in a misclassification.

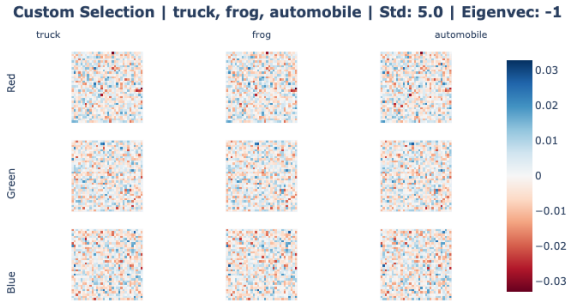
B CIFAR-10 and CIFAR-100 Extension



(a) Without noise ($\sigma = 0$)

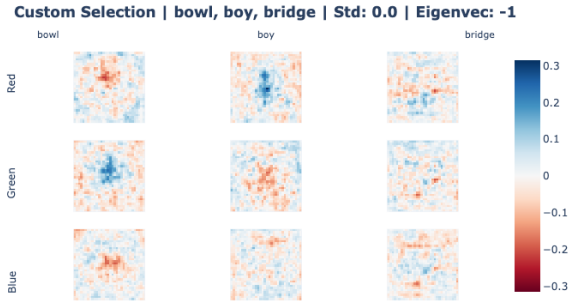


(b) With noise ($\sigma = 0.4$)

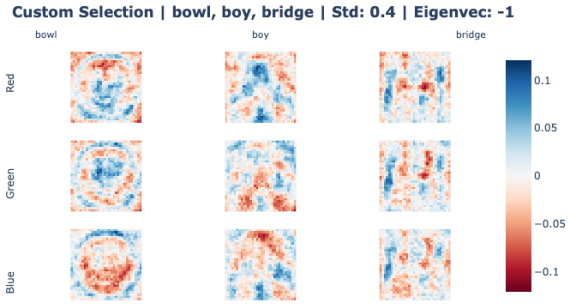


(c) With too much noise ($\sigma = 5$)

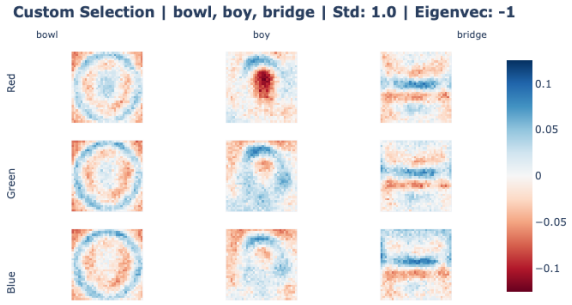
Figure 10: **Effect of input noise regularization on CIFAR-10 most positive eigenvector (-1).** Top positive eigenvector for classes *truck, frog and automobile* across RGB channels. (a) Baseline model without regularization shows noisy, less interpretable features. (b) Regularization with Gaussian noise ($\sigma = 0.4$) produces cleaner eigenvectors with more structured patterns. Where Red/Blue indicate negative/positive eigenvalue/feature contribution, respectively. (c) Adding too much noise produces noisy eigenvectors.



(a) Without noise ($\sigma = 0$)



(b) With noise ($\sigma = 0.4$)



(c) With strong noise ($\sigma = 1$)

Figure 11: **Effect of input noise regularization on CIFAR-100 most positive eigenvector (-1).** Top positive eigenvector for classes *bowl, boy, bridge* across RGB channels. (a) Baseline model without regularization also shows noisy, less interpretable features. (b) Regularization with Gaussian noise ($\sigma = 0.4$) again produces cleaner eigenvectors with more structured patterns. Where Red/Blue indicate negative/positive eigenvalue/feature contribution, respectively. (c) Adding strong noise creates even cleaner eigenvectors.

C Environmental Impact

Table 7: Environmental impact of individual segments of the project

Project section	kg CO_2e	CPU (W)	GPU (W)	RAM (W)
Claims 1-2	1.16×10^{-3}	720.0	24.24	40.0
Claims 3-4	20.27×10^{-3}	10.30	0	20
Extension 1	7.52×10^{-3}	39.70	0	80
Extension 2	8.80×10^{-3}	4.78	0.025	3.0
TOTAL	37.75×10^{-3}	774.78	24.265	143

D Detailed Hardware Specifications and Runtimes

To provide full transparency, Table 8 outlines the specific hardware environments, architectural allocations, and execution runtimes for each claim and extension evaluated in this study.

Table 8: Detailed hardware platforms and execution runtimes per experiment.

Experiment	**Hardware Platform**	**Specifications**	**Approx. Runtime**
Claims 1 & 2	Intel Core i7-11800H	8 Cores, 16 Threads @ 2.30 GHz	$<$ 5 minutes
Claim 3	Intel Core i7-6820HQ	4 Cores, 8 Threads @ 2.70 GHz	44 minutes
Claim 4	Intel Core i7-6820HQ	4 Cores, 8 Threads @ 2.70 GHz	30 minutes
Extension 1	Intel Core Ultra 7 155H	16 Cores, 22 Threads @ 1.40 GHz	1.5 hours
Extension 2	Apple M1	8 Cores (CPU), 7 Cores (GPU), 8 GB Unified Memory	4.0 hours