SPECTRA: Semantic Perturbation-Based Counterfactuals and Training for Robustness Against Adversarial Attacks

Aditya Sridhar, Ananya Varshney

Abstract

This paper introduces SPECTRA (Semantic Perturbation-based Counterfactuals for Robust Adversarial Training), a novel method for generating counterfactuals through minimal perturbations in the embedding space of convolutional neural networks (CNNs). By explicitly targeting semantic features, SPECTRA enhances the interpretability of classification decisions and provides a quantitative measure of adversarial attack vulnerability. Our method leverages fine-grained attribute-based datasets like CUB-200, ensuring perturbations align with known semantic structures. Through experiments, we demonstrate that SPECTRA not only supports precise and localized counterfactual reasoning but also improves model robustness by optimizing for stability against adversarial perturbations.

1 Introduction

Machine learning models, especially deep neural networks, have achieved remarkable success in various tasks, yet their susceptibility to adversarial attacks and opaque decision-making processes pose significant challenges. These issues underscore the importance of interpretability and robustness in modern AI systems. Counterfactual reasoning, which involves generating minimal alterations to features to change an outcome, has emerged as a powerful tool for interpreting model decisions. However, existing methods often fail to provide fine-grained, semantically meaningful perturbations or robust defenses against adversarial threats.

This paper presents SPECTRA, a method designed to address these gaps. Unlike traditional approaches that manipulate features in the input spaces, SPECTRA operates in the embedding space of convolutional neural networks, enabling minimal yet effective perturbations. Leveraging datasets with rich semantic annotations, such as CUB-200, our method achieves interpretable attribute-based perturbations that are both targeted and localized. These perturbations facilitate counterfactual analysis, offering valuable insights into model behavior.

In addition to interpretability, SPECTRA incorporates a robustness framework that optimizes for resistance to adversarial attacks. By evaluating the minimal perturbation required to alter classification outcomes, we introduce a novel stability metric to quantify model robustness. Experimental results demonstrate that SPECTRA not only enhances interpretability through clear, localized feature modifications but also improves resilience to adversarial attacks, setting a new standard for interpretable and robust AI systems.

1.1 Motivation

Adversarial attacks expose critical vulnerabilities in deep learning models, particularly in tasks like image classification, where small, imperceptible perturbations can drastically

alter predictions. Addressing these challenges requires methods that are both computationally efficient and effective in improving model robustness. Existing approaches often struggle with scalability and interpretability, necessitating retraining or complex processing pipelines. SPECTRA overcomes these limitations by operating in the embedding space to generate a scalable number of semantically meaningful counterfactuals while introducing the Attackibility Metric to measure model stability against adversarial perturbations. This aligns with recent advancements like CANARY, an adversarial robustness evaluation platform that benchmarks deep learning models across robustness metrics, as highlighted by [4] . By combining interpretability with computational efficiency, SPECTRA offers a promising avenue for enhancing adversarial defenses in deep learning.

1.2 Related Works

The need for interpretable deep learning models has led to numerous advancements. [2] introduced a framework for learning ante-hoc explainable models by associating predictions with human-interpretable concepts. While their work excels in semantic understanding, it does not address adversarial robustness, leaving models vulnerable to targeted attacks. Similarly, [1] proposed using auto-encoders for middle-level explanations, bridging low-level features and high-level semantics. However, their approach is computationally expensive and less scalable for generating multiple counterfactuals, limiting its practicality in real-world applications.

On the other hand, [3] focused on generating adversarial attacks in the latent space, revealing vulnerabilities in model embeddings. Although effective for evaluating weaknesses, their method lacks a defense-oriented framework or a mechanism for interpretability. SPECTRA builds on these ideas by integrating semantic perturbation-based counterfactuals with an adversarial robustness evaluation through the Attackibility Metric, ensuring a balance between interpretability and stability. Unlike prior works, SPECTRA achieves computational efficiency, scalability, and robustness, setting a new standard for explainable and secure deep learning models.

2 Proposed Method

2.1 Dataset and Motivation

For our experiments, we utilized the CUB-200 (Caltech UCSD Birds) dataset, a fine-grained dataset annotated with 312 semantic attributes across 200 bird species. This dataset allows for explicit attribute-wise perturbation analysis. Using datasets with well-defined attributes mitigates reliance on assumptions regarding the semantics of latent representations learned by classifiers on unannotated datasets, ensuring that the interpretability of perturbations aligns with known semantic structures.

2.2 Model Architecture Setup

We implemented a series of classifiers with varying levels of complexity: 1) a linear encoder-based model, 2) a simple convolutional neural network (CNN), 3) a more complex CNN architecture, and 4) a ResNet-based model.

Each model is designed to output an *attribute vector* $\mathbf{e} \in \mathbb{R}^n$ at the penultimate layer, where n represents the number of attributes (set to $n = 312$ in our case). The final classification layer maps the attribute vector $\mathbf{e}$ to the label space using a dense linear layer $\mathbf{W} \in \mathbb{R}^{c \times n}$, where c is the number of classes. The classification logits are

computed as:

$$\mathbf{l} = \mathbf{W} \cdot \mathbf{e}, \quad \mathbf{l} \in \mathbb{R}^c.$$

2.3 Training Procedure

The model was trained using a multi-objective loss function comprising:

1. **Attribute Loss:** A binary cross-entropy loss between the predicted normalized attribute vector $\hat{\mathbf{e}}$ and the ground-truth attribute vector $\mathbf{e}$.
2. **Classification Loss:** A categorical cross-entropy loss between the predicted class logits $\hat{\mathbf{l}}$ and the true class label y.

The combined loss function is:

$$\mathcal{L} = \lambda_a \cdot \text{BCE}(\hat{\mathbf{e}}, \mathbf{e}) + \lambda_c \cdot \text{CE}(\hat{\mathbf{l}}, y),$$

where λ_a and λ_c are hyperparameters controlling the relative weights of the attribute and classification losses. Through empirical evaluation, we found $\lambda_a = 0.7$ and $\lambda_c = 0.3$ to yield optimal performance.

2.4 Feature Visualization

To analyze the model's semantic understanding of attributes, we visualized activation maps for specific attributes. For a given attribute f, we computed the gradient of the attribute vector component e_f with respect to the input image $\mathbf{x}$:

$$\mathbf{g}_f = \frac{\partial e_f}{\partial \mathbf{x}}.$$

The top 500 pixels with the highest gradient values were visualized as an activation map, highlighting the most influential regions of the image for the attribute.

2.5 Minimal Perturbation Computation

Given the weight matrix $W \in \mathbb{R}^{c \times n}$, where c is the number of classes and n is the embedding vector size, we define the classification operation as $W \cdot e = l$, where $l \in \mathbb{R}^c$ is the label vector.

2.5.1 Targeted Minimal Perturbations

Assume the current predicted label is i, and the desired label is j. The feature index to be perturbed is f.

Let the perturbed embedding be $e' = e + \delta$, where $\delta \in \mathbb{R}^n$ is the perturbation vector. Specifically, $\delta = [0, 0, \ldots, \delta_f, \ldots, 0]^\top$.

To modify the classification output from an initial label i to a target label j, we compute a perturbation $\delta \in \mathbb{R}^n$ in the attribute space. The new logits are given by:

$$\mathbf{l}' = \mathbf{W} \cdot \mathbf{e}' = \mathbf{W} \cdot (\mathbf{e} + \delta).$$

The perturbation is given by:

$$W(e + \delta) = l' \quad \Rightarrow \quad W\delta = l' - l \tag{1}$$

$$(W_{kf})(\delta_f) = l'_k - l_k \tag{2}$$

$$(W_{jf})(\delta_f) = l'_j - l_j \tag{3}$$

$$(W_{jf} - W_{kf})(\delta_f) = (l'_j - l'_k) + (l_k - l_j) \quad > l_k - l_j, \forall k/j \tag{4}$$

This represents a linear constraint for δ_f, the component of the perturbation corresponding to attribute f. Solving this for δ_f across all f yields the minimal perturbation required to transition to the target label j.

2.5.2 Global Minimal Perturbations

For globally minimal perturbations, we consider the full attribute vector $\delta = [\delta_1, \delta_2, \ldots, \delta_n]^\top$. The perturbation must satisfy:

$$\mathbf{W}_{\text{diff}} \cdot \delta > l_{\text{diff}},$$

where $\mathbf{W}_{\text{diff}} = \mathbf{W}_j - \mathbf{W}_i$ and $l_{\text{diff}} = l_i - l_j$. Minimizing $\|\delta\|_2$ under these constraints can be solved as follows:

2.5.3 Computing the Minimum-Norm Perturbation δ

Given classifier weight matrix $W \in \mathbb{R}^{c\times n}$, where c is the number of classes and n is the feature dimension, we aim to compute a perturbation vector $\delta \in \mathbb{R}^n$ that satisfies:

$$W_{\text{global}}\delta = \mathbf{l}_{\text{global}},$$

where:

$$W_{\text{global}} = \begin{bmatrix} W_j - W_0 \\ W_j - W_1 \\ \vdots \\ W_j - W_{c-1} \end{bmatrix}, \quad \mathbf{l}_{\text{global}} = \begin{bmatrix} l_0 - l_j \\ l_1 - l_j \\ \vdots \\ l_{c-1} - l_j \end{bmatrix},$$

with $W_j - W_k \in \mathbb{R}^n$ (for $k \neq j$), $\mathbf{l}_{\text{global}} \in \mathbb{R}^{(c-1)\times 1}$, and $W_{\text{global}} \in \mathbb{R}^{(c-1)\times n}$.

The goal is to find the minimum-norm perturbation δ that satisfies this equation. The equation $W_{\text{global}}\delta = \mathbf{l}_{\text{global}}$ is a linear system that may be underdetermined, overdetermined, or exactly determined depending on the dimensions of W_{global}. To find the minimum-norm solution for δ, we use the **Moore-Penrose pseudoinverse** $W^\dagger_{\textbf{global}}$, which gives:

$$\delta_{\min} = W^\dagger_{\text{global}}\mathbf{l}_{\text{global}},$$

where:

$$W^\dagger_{\text{global}} = V\Sigma^\dagger U^\top,$$

is computed from the singular value decomposition (SVD) of W_{global}:

$$W_{\text{global}} = U\Sigma V^\top.$$

Here, $U \in \mathbb{R}^{(c-1)\times(c-1)}$, $\Sigma \in \mathbb{R}^{(c-1)\times n}$, and $V \in \mathbb{R}^{n\times n}$ are the SVD components, and $\Sigma^\dagger$ is the pseudoinverse of the diagonal matrix Σ, defined by reciprocals of its nonzero singular values.

The minimum-norm solution $\delta_{\min}$ satisfies:

$$\delta_{\min} = \arg\min_{\delta} \|\delta\|_2 \quad \text{subject to } W_{\text{global}}\delta = \mathbf{l}_{\text{global}}.$$

2.6 Perturbed Image Construction

To construct the required adversarially perturbed image, we iteratively adjust the input image $\mathbf{x}$ such that its attribute representation $\mathbf{e}'$ approaches the target attribute vector $\mathbf{e} + \delta$. At each step, the perturbed image $\mathbf{x}'$ is updated via gradient descent:

$$\mathbf{x}' \leftarrow \mathbf{x}' - \eta\nabla_{\mathbf{x}'}\mathcal{L}_{\text{perturb}},$$

where the perturbation loss is:

$$\mathcal{L}_{\text{perturb}} = \text{MSE}(\mathbf{e}', \mathbf{e} + \delta) + \lambda_{\text{SSIM}} \cdot (1 - \text{SSIM}(\mathbf{x}', \mathbf{x})).$$

Here, MSE ensures the attribute vector matches the target perturbation, and SSIM preserves structural similarity between the perturbed and original images.

The process terminates when the classifier outputs the desired target class for $\mathbf{x}'$.

2.7 Metric for Stability of Classification Model against Adversarial Attacks

To evaluate the stability of a classifier against adversarial attacks, we aim to measure how resistant the model is to label changes when subjected to minimal perturbations. Specifically, the goal is to ensure that the minimum possible perturbation required to change a label is as large as possible, thereby ensuring the model's robustness.

We define the SPECTRA Attackability as:

$$\text{SPECTRA Attackability (a)} = \max_{k \in 0,1,\ldots,c-1 \setminus i} \frac{|e|}{|\delta_{i,k}|}, \tag{5}$$

where δ_k is the perturbation required to change the label from i to any other label k. This metric seeks the smallest perturbation magnitude across all possible labels, excluding the current label i.

To obtain a comprehensive measure of the model's robustness, we take the average SPECTRA Attackability across all samples in the dataset. The final metric is given by:

$$\text{Model SPECTRA Attackability} = \frac{1}{N} \sum_{n=1}^{N} a_n, \tag{6}$$

where N is the total number of samples.

The lower the SPECTRA Attackability, the more stable and robust the model is against adversarial attacks.

3 Computing and Maximizing δ for Stability

To enhance the model's stability against adversarial attacks, we propose modifying the current training methodology. Currently, the classifier is trained on a joint loss function comprising the classification loss and attribute loss.

3.1 Maximizing Stability via δ

To enhance stability, we aim to maximize the magnitude of $\|\delta_{\min}\|_2$, which corresponds to maximizing the norm of the pseudoinverse-scaled vector $W_{\text{global}}^{\dagger}\mathbf{l}_{\text{global}}$. This leads to the stability loss function:

$$\mathcal{L}_{\text{stability}} = -\|W_{\text{global}}^{\dagger}\mathbf{l}_{\text{global}}\|_2^2.$$

3.2 Gradient-Based Optimization of W_{global}

To optimize the stability loss function $\mathcal{L}_{\text{stability}}$, we compute the gradient of $\mathcal{L}_{\text{stability}}$ with respect to W_{global}. Using the SVD decomposition:

$$W_{\text{global}} = U\Sigma V^{\top},$$

we backpropagate through the SVD to update W_{global} to maximize $\|W_{\text{global}}^{\dagger}\mathbf{l}_{\text{global}}\|_2^2$. Additionally, $\mathbf{l}_{\text{global}}$ can be treated as a learnable parameter for further optimization, depending on the problem constraints.

The total optimization framework is:

$$\min_{W_{\text{global}},\mathbf{l}_{\text{global}}} \mathcal{L}_{\text{stability}}.$$

This ensures that the learned perturbations maximize the stability metric $\|\delta_{\min}\|_2$.

3.3 Training Models from Scratch

For models trained from scratch, we introduce a sparsity constraint on the embeddings e to improve feature interpretability and visualization. This sparsity constraint penalizes non-informative activations in the embedding space, allowing for clearer feature-based reasoning.

The new loss function becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{classification}} + \mathcal{L}_{\text{reconstruction}} + \mathcal{L}_{\text{stability}} + \mathcal{L}_{\text{sparsity}}, \tag{7}$$

where $\mathcal{L}_{\text{sparsity}}$ enforces a desired level of sparsity in the embeddings.

4 Evaluation Metrics

The effectiveness of our method was assessed using:

- **Attack Success Rate (ASR):** The proportion of successful adversarial attacks.
- **Peak Signal-to-Noise Ratio (PSNR):** Measured in decibels (dB), computed as:
$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{\text{MSE}(\mathbf{x}, \mathbf{x}')}\right).$$
- **Perturbation Noise Ratio:** The ratio of the norms of input-space perturbations $\|\delta_{\text{input}}\|_2$ to latent-space perturbations $\|\delta_{\text{latent}}\|_2$, assessing trade-offs between interpretability and perturbation minimality.

5 Qualitative Results and Analysis

5.1 Experiment Description

The proposed architecture was applied to a simple CNN to retrieve qualitative results and analyze the perturbed images generated for specific attribute perturbations. The goal of the experiment was to infer the model's understanding of the semantic attributes and their localized impact on image regions during class transitions.

5.2 Results and Observations

5.2.1 Interpretability of Perturbed Attributes

In Figure 1, the original class, *Brewer Blackbird* (Class Index 8), was transformed into the required class, *Laysan Albatross* (Class Index 1). To achieve this, the attribute `has_nape_color::purple` (Feature 185) required a perturbation of 132.02 in the unnormalized attribute vector.

The observed perturbed image shows that the nape color of the bird became distinctly more purple. This indicates that the classification model interprets the *Laysan Albatross* as having a comparatively "more purple" nape region than the *Brewer Blackbird*, even though the attribute does not strictly describe the true appearance of a *Laysan Albatross*. This result underscores how the model uses learned attribute correlations to bridge class transitions.

Figure 1: Perturbed image illustrating a transition from *Brewer Blackbird* to *Laysan Albatross* via nape color alteration.

5.2.2 Proof of Feature Localization

Figure 2 demonstrates the localization of attributes during class transitions. In the transformation from *Black-Footed Albatross* to *Parakeet Auklet*, the attribute `breast_color::green` required a positive perturbation. The resulting perturbation was observed as green noise concentrated near the breast region, with minimal or no noise in other parts of the bird. This strongly indicates that the model has learned to associate the breast region with the specified attribute, showcasing its spatial understanding of semantic features.

Figure 2: Localized perturbation near the breast region during a transition from *Black-Footed Albatross* to *Parakeet Auklet*.

5.2.3 Complementary Color Observation

An intriguing observation was made in Figure 3, where the transformation from *Black-Footed Albatross* to *Least Auklet* required the attribute `crown_color::buff` to be perturbed negatively by -243.10. The perturbation resulted in a reduction in the "buffness" of the crown by introducing a light-blue color. Interestingly, light-blue (or cornflower blue) is complementary to buff, implying that the model compensates for reductions in one color by increasing its complement. This suggests an implicit understanding of color relationships within the attribute space.

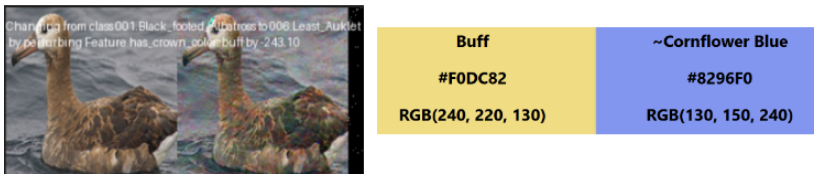

Figure 3: Complementary color shift during a transition from *Black-Footed Albatross* to *Least Auklet*, demonstrating reduced "buffness" in the crown region.

5.2.4 Minimal Pixel Alterations for Class Transitions

Figure 4 illustrates how minor pixel modifications can lead to class transitions. In the first example, altering a few yellow pixels in the *Yellow Blackbird* to green was sufficient to induce a class change. In the second example, modifying just one or two pixels in the throat region to green successfully transitioned the class. This demonstrates the efficiency of the proposed perturbation method, where minimal pixel-level changes yield significant attribute-level shifts.

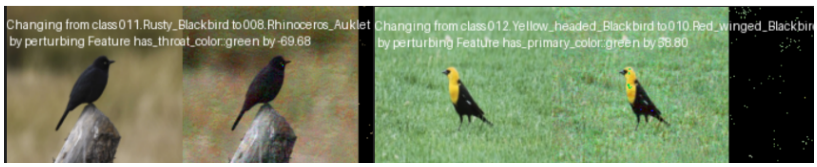

Figure 4: Minimal pixel alterations resulting in significant attribute-level shifts.

5.2.5 Localized Random Noise for Negative Perturbations

Figure 5 highlights the model's response to negatively perturbing the attribute `has_underparts_color::black`. The model added random noise to the underparts region, effectively reducing the "blackness" of the underparts. Notably, the noise was confined to the specified region, further demonstrating the model's semantic and spatial understanding of the attributes.

5.3 Limitations and Challenges

5.3.1 Un-perturbable Attributes

A limitation of the proposed method is evident in Figure 6, where the attribute `has_size::very_small` needed to be perturbed. Since modifying the perceived size of a bird in an image is not semantically feasible through attribute-based perturbations, the model introduced random noise, resulting in nonsensical outputs. This limitation

Figure 5: Localized random noise applied to the underparts to negatively perturb the attribute `has_underparts_color::black`.

highlights the model's reliance on interpretable, perturbable attributes and its inability to semantically adjust certain global or abstract features.

5.3.2 Non-Interpretability of Perturbed Images

While the proposed method demonstrates semantic understanding of attributes in most cases, it is observed that not all perturbed images are interpretable. In some instances, the perturbations manifest as random Gaussian noise or non-local changes that do not correspond to the attribute intended to be perturbed. These artifacts may arise due to limitations in the disentanglement of learned attribute representations, leading to the model's inability to localize changes accurately for specific attributes. This behavior reduces the interpretability of some outputs, which may hinder their applicability in tasks requiring robust semantic understanding.

5.3.3 Inexact Perturbation of Encodings

Achieving the exact required perturbation in the attribute encoding space is inherently challenging due to the iterative nature of gradient-based optimization. Gradient descent does not guarantee convergence to the precise target encoding within a reasonable number of iterations, particularly when the optimization landscape is complex or poorly conditioned. Furthermore, the learned latent representations of the model are not perfectly disentangled, meaning that perturbations to the intended attribute often induce changes in other attributes as well. While the method ensures that the perturbation of the targeted attribute significantly outweighs the perturbation of other attributes on average, this trade-off may limit its precision and robustness in fine-grained applications.

5.4 Analysis

The qualitative results demonstrate the following:

- The model effectively understands and utilizes semantic attributes to transition between classes, as evidenced by localized and interpretable perturbations.
- The spatial localization of attribute changes, such as those in the breast or crown regions, indicates a learned association between attributes and specific image regions.

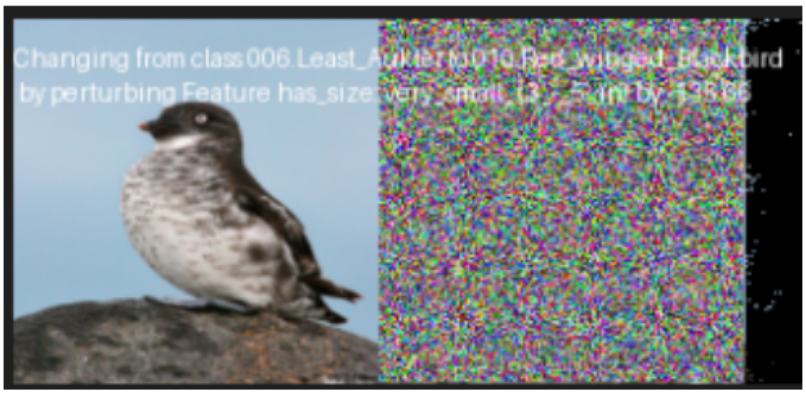

Figure 6: Nonsensical perturbations resulting from an un-perturbable attribute `has_size::very_small`.

- The observation of complementary color relationships reflects the model's implicit understanding of color dynamics, adding to its interpretability.
- Minimal pixel alterations leading to significant class transitions highlight the efficiency and precision of the attribute perturbation approach.
- The model's inability to handle un-perturbable attributes (e.g., size) suggests a need for improved mechanisms to handle non-spatial or abstract features.

These observations validate the interpretability and robustness of the proposed method while revealing areas for improvement in handling abstract, global attributes.

6 Conclusion

This paper introduces SPECTRA, a method for generating counterfactuals by applying minimal perturbations to features in the embedding space. This approach not only enhances interpretability but also provides a robust metric for assessing the vulnerability of models to adversarial attacks.

7 References

References

[1] Andrea Apicella, Salvatore Giugliano, Francesco Isgro, and Roberto Prevete. Exploiting auto-encoders for middle-level explanations of image classification systems. *arXiv preprint arXiv:2106.05037*, 2021.

[2] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N Balasubramanian. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10286–10295, 2022.

[3] Nitish Shukla and Sudipta Banerjee. Generating adversarial attacks in the latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 730–739, 2023.

[4] Jiazheng Sun, Li Chen, Chenxiao Xia, Da Zhang, Rong Huang, Zhi Qiu, Wenqi Xiong, Jun Zheng, and Yu-An Tan. Canary: An adversarial robustness evaluation platform for deep learning models on image classification. *Electronics*, 12(17):3665, 2023.