

Explaining Deep Neural Networks Through Fooling

Shpresim Sadiku

(Technische Universität Berlin & Zuse Institute Berlin)



University of Prishtina (UP) Math Seminar · February 27, 2025

Deep Neural Networks (DNNs)

DNNs for (Image) Classification

- High success rate
- **Robustness?**
 - Highly unstable - minor input shifts result in major output shifts [Sze+13]
 - Utilize this vulnerability of NNs to alternate their decision
 - Provide suggestions to achieve the desired outcome

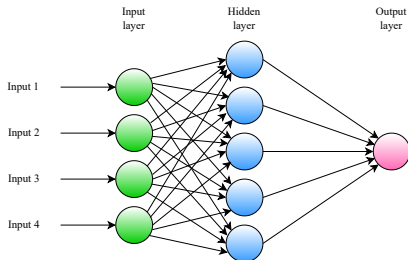


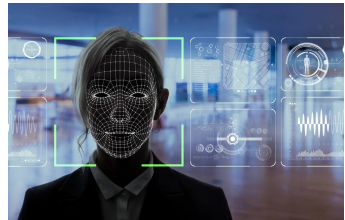
Figure 1: 1-hidden layer feed-forward NN.

Deep Learning Safety-critical Applications

Self-driving



Face recognition



■ Worst-case scenarios

- Life-threatening accidents in autonomous driving
- Information leakage in face recognition

Inverse Classification

- Input space $\mathcal{X} \subseteq \mathbb{R}^d$
- Output space \mathcal{Y} of class labels
- Classifier $f_l : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$
- Final decision

$$f(\mathbf{x}) = \arg \max_i [f_l(\mathbf{x})]_i$$

- *Adversarial examples* for images
 - $\mathcal{X} = [I_{\min}, I_{\max}]^{M \times N \times C}$

correctly classified image + small perturbation = incorrectly classified image

$$\mathbf{x} + \mathbf{w} = \mathbf{y}, \quad \|\mathbf{w}\|_p < \epsilon$$



visually indistinguishable

but

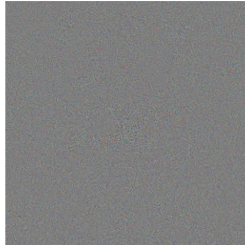
$$f(\mathbf{x}) \neq f(\mathbf{y})$$

Spot the Difference

Original
Label: 986 (daisy)



Perturbation scaled by 15
 $\epsilon = 0.03$



PGD adversarial example
Prediction: 524 (crutch)



Existence of Adversarial Attacks

- Phenomenon of *adversarial attacks* reveals critical vulnerabilities in DNNs
- Standard training methods produce non-robust models when trained on data lying in low-dimensional subspaces [MYV23]
 - ↪ Large gradients in directions orthogonal to the data subspace
- While humans perceive adversarial attacks as noise, machines perceive them as features [Ily+19]
 - ↪ Learning from adversarial attacks achieves similar accuracy to learning from normal training data [KKY24]

Adversarial Attack Generation

- White-box attack - f_l is known
- Benign image $\mathbf{x} \in \mathcal{X}$ of correct label $l \in \mathbb{N}$
- Target label $t \in \mathbb{N}, t \neq l$
- $\mathcal{L} : \mathcal{X} \times \mathbb{N} \rightarrow \mathbb{R}$ classification loss function (e.g. cross-entropy loss) tailored for f
- Goal of a traditional adversary - succeed under minimal distortion

$$\min_{\mathbf{w} \in \mathbb{R}^d} \mathcal{L}(\mathbf{x} + \mathbf{w}, t) + \lambda \|\mathbf{w}\|_p^p \quad (1)$$

for $\lambda > 0$ and $p \geq 0$

- $0 \leq p \leq 1$ changes very few pixels at high magnitudes
↪ Easily perceptible even for the human eye [Fan+20]
- $p > 1$ changes most of the pixels at low magnitudes
↪ Appear as noise to humans but as features to DNNs [Ily+19]
- Our goal - bridge the gap between human perception and machine interpretation by generating attacks that are
 - Imperceptible - low magnitude
 - Targeted at the most important regions of the image

GSE: Group-wise Sparse and Explainable Adversarial Attacks

joint with

Moritz Wagner (TU Berlin & ZIB)

Sebastian Pokutta (TU Berlin & ZIB)

*To Appear in the Proceedings of International Conference on Learning Representations
(2025)*

Proximal Operator

Definition ([PB+14])

The proximal operator with respect to a (possibly non-smooth) function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined for any $\mathbf{w} \in \mathbb{R}^d$

$$\text{prox}_{\lambda g}(\mathbf{w}) := \arg \min_{\mathbf{y} \in \mathbb{R}^d} \frac{1}{2\lambda} \|\mathbf{y} - \mathbf{w}\|_2^2 + g(\mathbf{y}),$$

where $\lambda > 0$ is a given parameter.

- Useful for analyzing non-smooth functions g
 - Can be computed analytically for many such functions

Sparse Adversarial Attack Generation

- Express problem in Eq. (1) as a sum of two functions
 - $h(\mathbf{w}) := \mathcal{L}(\mathbf{x} + \mathbf{w}, t)$ and $g(\mathbf{w}) := \lambda \|\mathbf{w}\|_p^p$
- Make a quadratic approximation $\tilde{h}_L(\mathbf{w})$ to $h(\mathbf{w})$ and replace $\nabla^2 h(\mathbf{w})$ by $\frac{L}{2} I$
 - Note $h(\cdot)$ is a smooth, possibly non-convex function, whose gradient has Lipschitz constant L

$$\begin{aligned}
 \mathbf{w}^{k+1} &:= \arg \min_{\mathbf{y} \in \mathbb{R}^d} \tilde{h}_L(\mathbf{w}^k) + g(\mathbf{y}) \\
 &= \arg \min_{\mathbf{y} \in \mathbb{R}^d} \nabla_{\mathbf{w}^k} h(\mathbf{w}^k)^\top (\mathbf{y} - \mathbf{w}^k) + \frac{L}{2} \|\mathbf{y} - \mathbf{w}^k\|_2^2 \\
 &\quad + g(\mathbf{y}) \\
 &= \arg \min_{\mathbf{y} \in \mathbb{R}^d} \frac{L}{2} \|\mathbf{y} - [\mathbf{w}^k - \frac{1}{L} \nabla_{\mathbf{w}^k} h(\mathbf{w}^k)]\|_2^2 \\
 &\quad + g(\mathbf{y}) \\
 &= \text{prox}_{\frac{1}{L}g} \left(\mathbf{w}^k - \frac{1}{L} \nabla_{\mathbf{w}^k} h(\mathbf{w}^k) \right)
 \end{aligned}$$

- The inverse Lipschitz constant is further replaced by a step size sequence $(\alpha_k)_{k \in \mathbb{N}}$

Sparse Adversarial Attack Generation (cont.)

- Solve Eq. (1) via Forward-backward Splitting

Forward-Backward Splitting Attack

Require: Image $\mathbf{x} \in \mathcal{X}$, target label t , loss function \mathcal{L} , sparsity parameter $\lambda > 0$, step sizes α_k , number of iterations K

- 1 Initialize $\mathbf{w}^{(0)} \leftarrow \mathbf{0}$
- 2 **for** $k \leftarrow 0, \dots, K - 1$ **do**
- 3 $\mathbf{w}^{(k+1)} \leftarrow \text{prox}_{\alpha_k \lambda \|\cdot\|_p^p} (\mathbf{w}^{(k)} - \alpha_k \nabla_{\mathbf{w}^{(k)}} \mathcal{L}(\mathbf{x} + \mathbf{w}^{(k)}, t))$
- 4 **end for**
- 5 **return** $\hat{\mathbf{w}} = \mathbf{w}^{(K)}$

- Closed-form solution for $g(\mathbf{w}) := \lambda \|\mathbf{w}\|_p^p$ and $p \in \{0, 1/2, 2/3, 1\}$
 - Generates sparse but perceptible adversarial attacks [Fan+20]
- Utilize Forward-backward Splitting with Nesterov momentum for more efficiency

AdjustLambda

- Consider a vector of tradeoff parameters $\lambda \in \mathbb{R}_{\geq 0}^{M \times N \times C}$
- Determine key group-wise sparse coordinates to perturb
 \hookrightarrow Heuristically select group-wise sparse coordinates [SWP23]
 - 1 Build a mask $\mathbf{m} = \text{sign} \left(\sum_{c=1}^C |\mathbf{w}^{(k)}|_{:, :, c} \right) \in \{0, 1\}^{M \times N}$
 - 2 Apply Gaussian Blur Kernel $\mathbf{M} = \mathbf{m} * * \mathbf{K} \in [0, 1]^{M \times N}$
 - 3 Build $\overline{\mathbf{M}} \in \mathbb{R}^{M \times N}$ via

$$\overline{M}_{ij} = \begin{cases} M_{ij} + 1, & \text{if } M_{ij} \neq 0 \\ q, & \text{else} \end{cases}$$

for $0 < q \leq 1$

- 4 Set

$$\lambda_{i,j,:}^{(k+1)} = \frac{\lambda_{i,j,:}^{(k)}}{\overline{M}_{i,j}}$$

- Denote the chosen pixel coordinates by V

Solve a Low Magnitude Adversarial Attack Only Over V

- Formulate a simplified optimization problem

$$\min_{\mathbf{w} \in V} \mathcal{L}(\mathbf{x} + \mathbf{w}, t) + \mu \|\mathbf{w}\|_2 \quad (2)$$

- $\mu > 0$ controls perturbation magnitude
- Use projected Nesterov's accelerated gradient descent (NAG) to solve Eq. (2)

Lemma ([SWP23])

The projected NAG solving Eq. (2) converges as NAG solving an unconstrained problem.

Evaluation Metrics

- $(\mathbf{x}^{(i)})_{0 < i \leq n}$ images of perturbation $(\mathbf{w}^{(i)})_{0 < i \leq n}$
- *Attack Success Rate* $ASR = \frac{m_s}{n}$ for m_s successful adversaries
- *Average Number of Changed Pixels*

$$ACP = \frac{1}{m_s MN} \sum_{i=1}^{m_s} \|\mathbf{m}^{(i)}\|_0,$$

- Perform depth-first search (DFS) on \mathbf{m} from each undiscovered 1-entry
- *Average Number of Clusters (ANC)* – average the DFS runs needed to discover all 1-entries
- Group-wise sparsity

$$d_{2,0}(\mathbf{w}) := |\{i : \|\mathbf{w}_{G_i}\|_2 \neq 0, i = 1, \dots, k\}|$$

- $\mathcal{G} = \{G_1, \dots, G_k\}$ contains index sets of all overlapping patches in \mathbf{w}

Results on (Un)targeted Attacks

Table 1: Untargeted attacks on ResNet20 classifier for CIFAR-10, and ResNet50 and ViT_B_16 classifiers for ImageNet. Tested on 10k images of each dataset.

	Attack	ASR	ACP	ANC	ℓ_2	$d_{2,0}$
CIFAR-10 ResNet20	GSE (Ours)	100%	41.7	1.66	0.80	177
	StrAttack	100%	118	7.50	1.02	428
	FWnucl	94.6%	460	1.99	2.01	594
ImageNet ResNet50	GSE (Ours)	100%	1629	8.42	1.50	3428
	StrAttack	100%	7265	15.3	2.31	11693
	FWnucl	47.4%	13760	3.79	1.81	16345
ImageNet ViT_B_16	GSE (Ours)	100%	941	5.11	1.95	1964
	StrAttack	100%	3589	10.8	2.03	8152
	FWnucl	57.9%	7515	5.67	3.04	9152

Table 2: Targeted attacks (average case) performed on ResNet20 classifier for CIFAR-10, and ResNet50 and ViT_B_16 classifiers for ImageNet. Tested on 1k images from each dataset, 9 target labels for CIFAR-10 and 10 target labels for ImageNet.

	Attack	ASR	ACP	ANC	ℓ_2	$d_{2,0}$
CIFAR-10 ResNet20	GSE (Ours)	100%	86.3	1.76	1.13	262
	StrAttack	100%	231	10.1	1.86	534
	FWnucl	85.8%	373	2.52	2.54	564
ImageNet ResNet50	GSE (Ours)	100%	12014	14.6	2.93	16724
	StrAttack	100%	15071	18.0	3.97	20921
	FWnucl	7.34%	19356	7.58	3.17	26591
ImageNet ViT_B_16	GSE (Ours)	100%	2667	7.72	2.87	4571
	StrAttack	100%	8729	17.2	3.50	13349
	FWnucl	11.2%	6002	9.73	3.51	7427

Interpretability Metrics

- $Z(\mathbf{x})$ logits of vectorized image $\mathbf{x} \in [I_{\min}, I_{\max}]^d$
- *Adversarial Saliency Map (ASM)*, l – true label

$$[\text{ASM}(\mathbf{x}, l, t)]_i = \left(\frac{\partial Z(\mathbf{x})_t}{\partial \mathbf{x}_i} \right) \left| \frac{\partial Z(\mathbf{x})_l}{\partial \mathbf{x}_i} \right| \mathbb{1}_S(i)$$

$$S = \left\{ i \in \{1, \dots, d\} \mid \frac{\partial Z(\mathbf{x})_t}{\partial \mathbf{x}_i} \geq 0 \text{ or } \frac{\partial Z(\mathbf{x})_l}{\partial \mathbf{x}_i} \leq 0 \right\}$$

- Binary mask $\mathbf{B}(\mathbf{x}, l, t) \in \{0, 1\}^d$

$$[\mathbf{B}(\mathbf{x}, l, t)]_i = \begin{cases} 1, & \text{if } [\text{ASM}(\mathbf{x}, l, t)]_i > \nu \\ 0, & \text{otherwise} \end{cases}$$

- ν is some percentile of the entries of $\text{ASM}(\mathbf{x}, l, t)$
- *Interpretability score (IS)* given perturbation $\mathbf{w} \in \mathbb{R}^d$

$$\text{IS}(\mathbf{w}, \mathbf{x}, l, t) = \frac{\|\mathbf{B}(\mathbf{x}, l, t) \odot \mathbf{w}\|_2}{\|\mathbf{w}\|_2}$$

- *Class activation maps (CAMs)* identify class-specific discriminative image regions [Zho+16]

Quantitative Evaluation

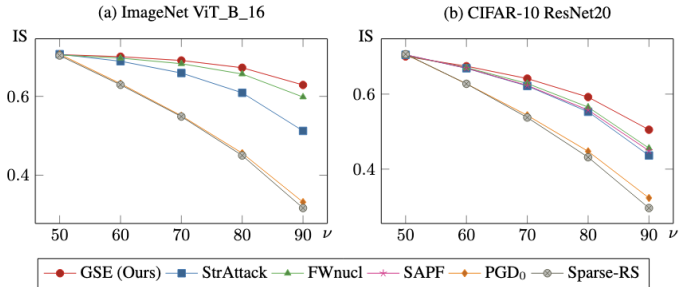


Figure 2: IS vs. percentile ν for targeted versions of GSE vs. five other attacks. Evaluated on an ImageNet ViT_B_16 classifier (a), and CIFAR-10 ResNet20 classifier (b). Tested on 1k images from each dataset, 9 target labels for CIFAR-10 and 10 target labels for ImageNet.

Visual Analysis

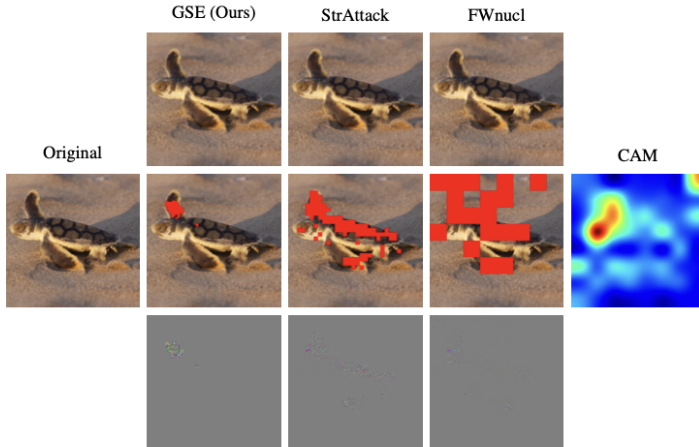


Figure 3: Visual comparison of successful, untargeted adversarial examples for our attack, StrAttack, and FWnucl. (Top row) adversarial examples, (middle row) perturbed pixels highlighted in red, (bottom row) perturbations scaled by 5. The target model is a ResNet50.

Further Results

- GSE exhibits significantly faster performance compared to benchmark methods
- ASR when attacking adversarially robust models?
 - GSE generates perturbations that adversarially robust models struggle to defend against effectively
- Transferability when targeting a different model?
 - GSE demonstrates transferability (maintains a high ASR) on par with benchmark methods

From Adversarial Attacks to Counterfactual Explanations

Input space \mathcal{X}

- More general tabular data
- Applications in credit lending, parole, medical treatment etc

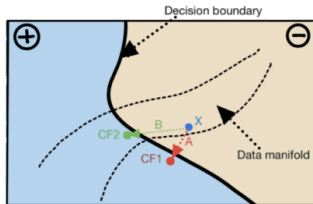


Figure 4: Two possible paths to misclassify a datapoint \mathbf{x} (shortest path (red) vs. path adhering closest to the manifold (green) of training data).

Credit lending example

- Alice seeks a home mortgage loan
- ML classifier considers Alice's feature vector $\{Income, CreditScore, Education, Age\}$
- Alice is denied the loan
 - Why the loan was denied? - Explainable AI (XAI)
 - *CreditScore* was too low
 - What can she do differently so that the loan will be approved in the future? - Counterfactual Explanations (CFEs)
 - Increase *Income* by \$10K
 - Get a master's degree
 - A combination of both

Core Difference Between Adversarial Attacks and CFEs

- Both want the network to misclassify (*Validity*) under minimal distortion (*Proximity*)
- Adversarial attacks push the data point out of its original class distribution
- CFEs aim to nudge the data point toward the target class's distribution (*Plausibility*)
 - Changes should apply only to valid feature ranges (*Actionability*)
 - E.g. Alice cannot decrease her age by ten years

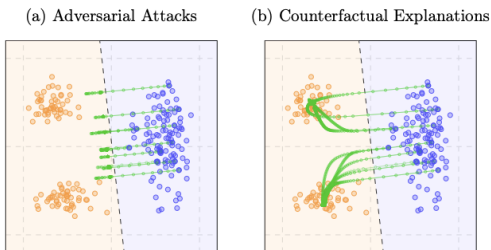


Figure 5: (a) Methods without a plausibility term generate points near the factual blue data points, but they remain distant from the distribution of correctly classified orange data points. (b) Methods combined with a plausibility term produce points within high-density regions. The dashed black line represents the decision boundary of a linear classifier.

S-CFE: Simple Counterfactual Explanations

joint with

Moritz Wagner (TU Berlin & ZIB)

Sai Ganesh Nagarajan (ZIB)

Sebastian Pokutta (TU Berlin & ZIB)

To Appear in the Proceedings of International Conference on Artificial Intelligence and Statistics (2025)

CFE Formulation

- Assume data points are generated from the joint density $\psi : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}_+$
 - $q(\mathbf{x}, t) := \psi(\mathbf{x}|t)$ - density of inputs conditioned on target label t
- Denoting $\mathbf{y} := \mathbf{x} + \mathbf{w}$, basic adversarial attack problem (1) transforms into

$$\min_{\mathbf{y} \in \mathbb{R}^d} \mathcal{L}(\mathbf{y}, t) + \lambda \|\mathbf{y} - \mathbf{x}\|_2^2$$

- Accounts for *Validity* and *Proximity*
- Utilize indicator function for *Actionability* constraint $\mathbf{y} \in \mathcal{A}$ where $\mathcal{A} := \times_{i=1}^d [-\mathcal{A}_i, \mathcal{A}_i]$, for $\mathcal{A}_i \in \mathbb{R}$

$$I_{\mathcal{A}}(\mathbf{y}) := \begin{cases} 0, & \text{if } \mathbf{y} \in \mathcal{A} \\ +\infty, & \text{otherwise} \end{cases}$$

- Add additional regularizers for *Plausibility*, *Actionability*, and *Sparsity*

$$\begin{aligned} \mathbf{y}_{cf} := \arg \min_{\mathbf{y} \in \mathbb{R}^d} & \mathcal{L}(\mathbf{y}, t) + \lambda \|\mathbf{y} - \mathbf{x}\|_2^2 + I_{\mathcal{A}}(\mathbf{y}) \\ & - \tau \hat{q}(\mathbf{y}, t) + \beta \|\mathbf{y} - \mathbf{x}\|_0 \end{aligned} \quad (3)$$

- $\hat{q}(\mathbf{y}, t)$ is a density estimate for the target class t in \mathcal{X}

CFE Formulation (cont.)

- Similarly
 - $h(\mathbf{y}, t) := \mathcal{L}(\mathbf{y}, t) + \lambda \|\mathbf{y} - \mathbf{x}\|_2^2 - \tau \hat{q}(\mathbf{y}, t)$
 - $g(\mathbf{y}) := I_{\mathcal{A}}(\mathbf{y}) + \beta \|\mathbf{y} - \mathbf{x}\|_0$
- Differentiable density estimators
 - Gaussian mixture models (GMMs)
 - Kernel density estimates (KDE)
- Solve Eq. (3) via accelerated proximal gradient (APG) method [BT09]
 - Backpropagation to compute $\nabla_{\mathbf{y}} h(\mathbf{y}, t)$
 - Proximal operator for $g(\mathbf{y})$ is given by the clipped iterative hard-thresholding algorithm [ZCW21]

Constraining the Sparsity

- Regularize using the indicator function of the sparsity constraint
↪ Improved control over sparsity

$$I_{\|\mathbf{y}-\mathbf{x}\|_0 \leq m}(\mathbf{y}) := \begin{cases} 0, & \text{if } \|\mathbf{y}-\mathbf{x}\|_0 \leq m \\ +\infty, & \text{otherwise} \end{cases}$$

- Reformulate Eq. (3)

$$\begin{aligned} \mathbf{y}_{cf} := \arg \min_{\mathbf{y} \in \mathbb{R}^d} & \mathcal{L}(\mathbf{y}, t) + \lambda \|\mathbf{y}-\mathbf{x}\|_2^2 + I_{\mathcal{A}}(\mathbf{y}) \\ & - \tau \hat{q}(\mathbf{y}, t) + \beta I_{\|\mathbf{y}-\mathbf{x}\|_0 \leq m}(\mathbf{y}) \end{aligned} \quad (4)$$

- $g(\mathbf{y}) := I_{\mathcal{A}}(\mathbf{y}) + \beta I_{\|\mathbf{y}-\mathbf{x}\|_0 \leq m}(\mathbf{y})$ is an indicator function
↪ Proximal operator coincides with the projection onto the intersection

$$\{\|\mathbf{y}-\mathbf{x}\|_0 \leq m\} \cap \mathcal{A}$$

Constraining the Sparsity (cont.)

- Closed-form solution [CH19]

$$\begin{aligned}
 [P_{\{\|\mathbf{y}-\mathbf{x}\|_0 \leq m\}} \cap \mathcal{A}(S_\alpha(\mathbf{y}, t))]_i &= \begin{cases} z_i, & \text{if } i \in Q, \\ 0, & \text{otherwise,} \end{cases} \\
 \mathbf{z} &= \Pi_{\mathcal{A}}(S_\alpha(\mathbf{y}, t)), \\
 Q &= \operatorname{argtopk}(\mathbf{v}, m),
 \end{aligned}$$

- $\mathbf{v} = \mathbf{w} \odot \mathbf{w} - (\mathbf{w} - \mathbf{z}) \odot (\mathbf{w} - \mathbf{z})$ with $\mathbf{w} = \mathbf{y} - \mathbf{x}$
 - \odot element-wise product
- $\operatorname{argtopk}(\mathbf{v}, m)$ indices corresponding to the m largest absolute values of the entries of \mathbf{v}
- $S_\alpha(\mathbf{y}, t) = \mathbf{y} - \alpha \nabla_{\mathbf{y}} h(\mathbf{y}, t)$
- $\Pi_{\mathcal{A}}(\mathbf{y}) = \operatorname{arg min}_{\mathbf{y}'} \{\|\mathbf{y}' - \mathbf{y}\|_2^2 \mid \mathbf{y}' \in \mathcal{A}\}$

Lemma

Since $g(\mathbf{y}) := I_{\mathcal{A}}(\mathbf{y}) + \beta I_{\|\mathbf{y}-\mathbf{x}\|_0 \leq m}(\mathbf{y})$ is a proper and lower semicontinuous function, the convergence of APG to a critical point of the minimization problem (4) can be assured (even for non-convex and non-smooth $g(\cdot)$), under some mild conditions [LL15].

Evaluation Metrics

- Ratio of CFEs with the desired class label for *Validity*
- 2-norm for *Proximity*
- 0-norm for *Sparsity*
- LOF metric for *Plausibility* [Bre+00]
- Average runtime per method

Quantitative Evaluation

Table 3: CFEs for DNN classifiers on the Boston Housing and Wine datasets, and for a CNN classifier on the MNIST dataset. Evaluated on 1000 test points for MNIST and 100 test points for the other two datasets.

Dataset	Method	Validity (std)	2-norm (std)	0-norm (std)	LOF (std)	Time
Housing 12 features	S-CFE _{KDE}	100 (0.00)	2.59 (1.21)	2.00 (0.00)	1.23 (0.29)	12.7
	S-CFE _{GMM}	100 (0.00)	2.91 (1.38)	2.00 (0.00)	1.12 (0.26)	13.3
	S-CFE _{kNN}	100 (0.00)	3.64 (1.73)	2.00 (0.00)	1.17 (0.31)	5.85
	DCFE	100 (0.00)	3.50 (1.68)	6.86 (1.42)	1.27 (0.38)	5.33
	CEM	94.0 (0.23)	2.93 (2.23)	2.99 (1.17)	1.36 (0.60)	7.51
Wine 13 features	S-CFE _{KDE}	100 (0.00)	3.31 (1.16)	2.00 (0.00)	0.99 (0.01)	12.4
	S-CFE _{GMM}	100 (0.00)	3.44 (1.09)	2.00 (0.00)	0.98 (0.02)	13.1
	S-CFE _{k-NN}	100 (0.00)	4.04 (1.59)	2.00 (0.00)	1.01 (0.07)	5.80
	DCFE	100 (0.00)	3.21 (2.70)	7.13 (1.31)	1.03 (0.18)	4.95
	CEM	92.0 (0.29)	5.40 (3.25)	5.14 (2.68)	1.07 (0.14)	5.71
MNIST 784 features	S-CFE _{GMM}	99.1 (0.09)	6.74 (2.92)	25.0 (0.00)	1.21 (0.18)	55.3
	S-CFE _{k-NN}	99.8 (0.04)	7.04 (2.99)	25.0 (0.00)	1.30 (0.22)	13.1
	DCFE	99.3 (0.08)	8.06 (3.48)	118 (6.30)	1.32 (2.24)	11.8

Robustness of Plausible CFEs to Input Manipulations

- CFEs without plausibility diverge significantly
 - Minor input perturbations result in major output shifts
 - Two similar individuals may receive drastically different explanations

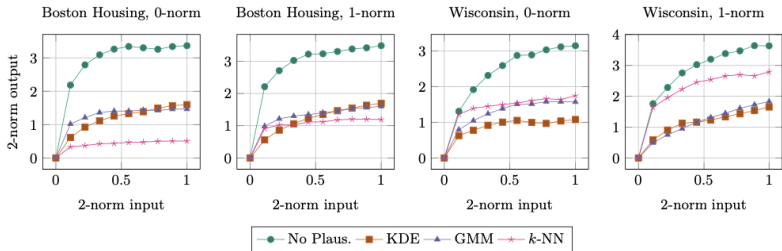


Figure 6: Robustness of the different methods. The distance of the input data points to the original data points on the x -axis and the distance of the generated CFEs to the CFE generated from the original data points on the y -axis. Tested on 100 data points from each data set.

Discussion

- Plausible CFEs, in general, cannot be interpreted as action recommendations
- CFEs provide hints about which alternative feature values would yield acceptance by the predictor
 - Do not guide the user on which interventions yield the desired change in the real world
 - To guide action, causal knowledge is required
- *Improvement* of the underlying target is more desirable than *acceptance* by a specific predictor
 - E.g., Covid infection prediction - intervening on the symptoms may change the diagnosis (prediction), but will not affect whether someone is infected (real-world state) [KFG23]

THANK YOU!

Slides available at:

www.shpresimsadiku.com

References I

- [Bre+00] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.
- [BT09] Amir Beck and Marc Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.
- [Sze+13] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *arXiv preprint arXiv:1312.6199* (2013).
- [PB+14] Neal Parikh, Stephen Boyd, et al. “Proximal algorithms”. In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.
- [LL15] Huan Li and Zhouchen Lin. “Accelerated proximal gradient methods for nonconvex programming”. In: *Advances in neural information processing systems* 28 (2015).
- [Zho+16] Bolei Zhou et al. “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.
- [CH19] Francesco Croce and Matthias Hein. “Sparse and imperceivable adversarial attacks”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 4724–4732.

References II

- [Ily+19] Andrew Ilyas et al. “Adversarial examples are not bugs, they are features”. In: *Advances in neural information processing systems 32* (2019).
- [Fan+20] Yanbo Fan et al. “Sparse adversarial attack via perturbation factorization”. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer. 2020, pp. 35–50.
- [ZCW21] Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. “Sparse and imperceptible adversarial attack via a homotopy algorithm”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12868–12877.
- [KFG23] Gunnar König, Timo Freiesleben, and Moritz Grosse-Wentrup. “Improvement-focused causal recourse (ICR)”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 10. 2023, pp. 11847–11855.
- [MYV23] Odelia Melamed, Gilad Yehudai, and Gal Vardi. “Adversarial examples exist in two-layer relu networks for low dimensional linear subspaces”. In: *Advances in Neural Information Processing Systems 36* (2023), pp. 5028–5049.
- [SWP23] Shpresim Sadiku, Moritz Wagner, and Sebastian Pokutta. “Group-wise Sparse and Explainable Adversarial Attacks”. In: *arXiv preprint arXiv:2311.17434* (2023).

References III

- [KKY24] Soichiro Kumano, Hiroshi Kera, and Toshihiko Yamasaki. “Theoretical Understanding of Learning from Adversarial Perturbations”. In: *International Conference on Learning Representations (2024)*.