AIDE: Antithetical, Intent-based, and Diverse Example-Based Explanations

Ikhtiyor Nematov[1,2], **Dimitris Sacharidis**[1], **Tomer Sagi**[2], **Katja Hose**[3]

[1]Universite Libre de Bruxelles, Belgium
[2]Aalborg University, Denmark
[3]TU Wien, Austria
ikhtiyor.nematov@ulb.be, dimitris.sacharidis@ulb.be, tsagi@cs.aau.dk, katja.hose@tuwien.ac.at

Abstract

For many use-cases, it is often important to explain the prediction of a black-box model by identifying the most influential training data samples. Existing approaches lack customization for user intent and often provide a homogeneous set of explanation samples, failing to reveal the model's reasoning from different angles.

In this paper, we propose AIDE, an approach for providing antithetical (i.e., contrastive), intent-based, diverse explanations for opaque and complex models. AIDE distinguishes three types of explainability intents: interpreting a correct, investigating a wrong, and clarifying an ambiguous prediction. For each intent, AIDE selects an appropriate set of influential training samples that support or oppose the prediction either directly or by contrast. To provide a succinct summary, AIDE uses diversity-aware sampling to avoid redundancy and increase coverage of the training data.

We demonstrate the effectiveness of AIDE on image and text classification tasks, in three ways: quantitatively, assessing correctness and continuity; qualitatively, comparing anecdotal evidence from AIDE and other example-based approaches; and via a user study, evaluating multiple aspects of AIDE. The results show that AIDE addresses the limitations of existing methods and exhibits desirable traits for an explainability method.

Introduction

Failure of ML-based systems in numerous cases, e.g., due to data errors, biases, misalignment (Osoba and Welser IV 2017; Tashea 2017; Seymour et al. 2023; Burema et al. 2023), has prompted researchers to work on explainability techniques. Different taxonomies for such methods exist, e.g., (Guidotti et al. 2018), but one common classification is on the type of explanation generated (Molnar 2022). *Model-based* methods involve creating interpretable surrogate models, such as decision trees or linear models, which approximate the complex black box ML model (Ribeiro, Singh, and Guestrin 2018; Silva et al. 2020). *Feature-based* methods focus on pinpointing important features of the input, such as words in text or parts in an image, which contribute the most to the prediction (Ribeiro, Singh, and Guestrin 2016; Fong, Patrick, and Vedaldi 2019; Ancona 2017). *Example-based* methods provide explanations for a specific target outcome by deriving the importance of training samples (Ilyas et al. 2022; Garima et al. 2020; Koh and Liang 2017; Kwon and Zou 2022; Ghorbani and Zou 2019; Park et al. 2023), or provide a global overview of the model identifying representative examples (Yeh et al. 2018; Pruthi et al. 2020).

Example-based explainability offers several advantages. They are typically model-agnostic, and offer easy to understand explanations. More importantly, as they seek to discover a causal relationship between training examples and model behavior, they can assist in model debugging and data cleansing (Hara, Nitanda, and Maehara 2019). However, they have two key limitations.

First, they don't offer *contrastivity* (Nauta et al. 2023), which is key aspect in how humans understand decisions (Lipton 1990). While most methods can distinguish between *supporters* (aka proponents, helpful or excitatory examples), and *opposers* (aka opponents, harmful or inhibitory examples), they do not relate this information to ground truth labels (examples of class same as or different than predicted) or to the explanation intent (is the prediction correct/wrong, hard to tell). Contrastivity is the hallmark feature of counterfactual explanations (Wachter, Mittelstadt, and Russell 2017) and a major part of their appeal. Even though counterfactuals are instances, these are by design imaginary, not necessarily plausible (Pawelczyk, Broelemann, and Kasneci 2020) or robust (Slack et al. 2021). In essence, counterfactual explanations offer feature-based explainability, revealing the important feature values contributing to the outcome.

More importantly, existing example-based methods are highly susceptible to *class outliers*. An outlier is a training instance that is mislabeled, or an instance (training or target) that is ambiguous and does not clearly belong to a class. Mislabeled or ambiguous training instances tend to be explanations for any target instance, as they play a significant role in forming the decision boundary. Ambiguous target instances confuse the classifier (low confidence) and make it hard to pick good explanations.

In this paper, we propose a novel *Antithetical, Intent-based, and Diverse Example-based* (AIDE) explainability method, that offers contrastivity and is robust to outliers. At its core, AIDE is based on the concept of *influence functions* (Hampel 1974; Koh and Liang 2017). For a fixed target instance, the *influence* of a training sample is a score convey-

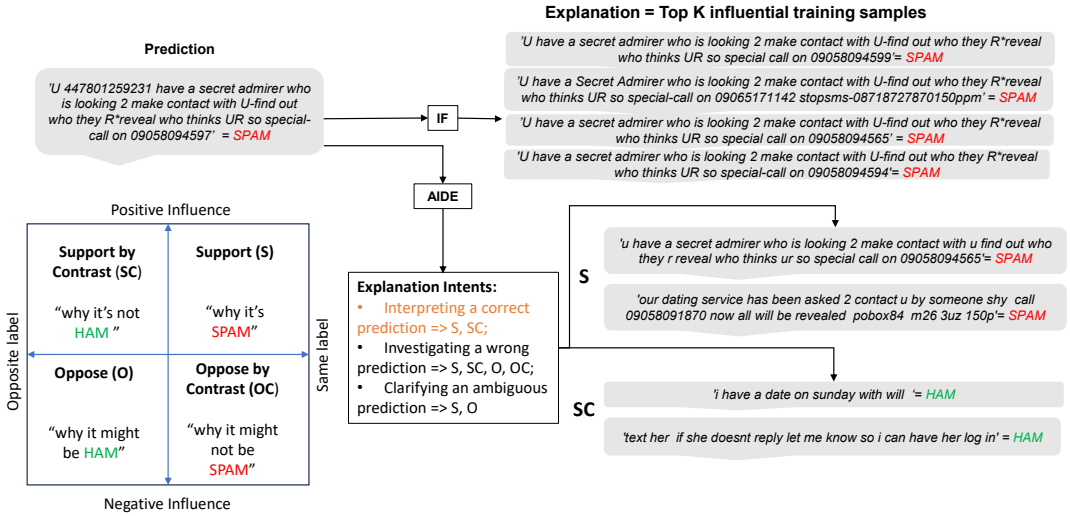


Figure 1: Explanations for a spam classification task, depicting a correctly classified spam message and its influence-based explanations generated by IF and AIDE.

ing its impact on the classifier's outcome. Ideally, the influence is the change observed in the loss value for the target if the training sample was excluded from the training data. While influence scores can be *estimated* by methods, such as TraceIn (Garima et al. 2020) and Datamodels (Ilyas et al. 2022), we use the framework of the influence function approach (Koh and Liang 2017), termed IF, to efficiently compute influence scores.

To better understand AIDE's contribution, we first showcase the issues that plague example-based explainability methods, taking IF as the representative—extensive qualitative and quantitative comparison with other methods is presented in the evaluation section. Consider a classifier that predicts whether short text messages are spam. Figure 1 shows that for the depicted target message, the prediction is spam. This is a correct prediction, and IF identifies the four most influential training samples at the top of Figure 1. We observe that explanations lack *diversity*, as they are highly similar to each other. More importantly, however, they lack *contrastivity*, as the user does not gain any insight about how the model decides what is spam and what not; all the user learns is that similar texts were labelled spam. The issue of susceptibility to outliers does not manifest in this example, mainly because the prediction is clearly correct. However, it manifests when for example what the correct prediction should be is not clear as in Figure 6.

Contribution. *AIDE features contrastivity*. Given a target instance to be explained, AIDE computes the influence of each training sample. But to present an explainability summary, AIDE distinguishes samples along two key explainability dimensions. The first is the *influence polarity*: a sample with positive influence *supports* the prediction, while one with negative influence *opposes* the decision. The second dimension is the label of the training sample, which is either the same or opposite as the target instance. These two dimensions define the four AIDE quadrants, denoted as support (S), support by contrast (SC), oppose (O), and oppose by contrast (OC). Assuming a binary classifier and that the prediction is $y \in \{-1, 1\}$, intuitively, S explains "*why it's* y", SC explains "*why it's not* $-y$", O explains "*why it might be* $-y$", and OC explains "*why it might not be* y". These quadrants offer contrastivity, providing to the user answers to distinct counterfactual questions. Figure 1 depicts the quadrants at the bottom left.

AIDE is intent-aware. A critical review (Keane et al. 2021) identifies misalignments between the design goals of XAI methods and the psychological and cognitive aspects of explainability. A key limitation they identify is that the majority of XAI methods neglect *user intent* during design and evaluation. Although mostly referring to counterfactual explanations, the authors highlight the importance of considering the user's needs and goals while generating explanations. Moreover, the authors identify another common limitation in that explanations may not be plausible, which they define as not being *relevant* to the prediction being explained. AIDE explicitly addresses both limitations by offering *intent-aware* and *relevant* example-based explanations.

AIDE acknowledges that users might have different intents. A user faced with a correct prediction, would more likely need additional evidence that the model has learned the correct patterns. A user recognizing a wrong prediction would want to narrow down the sources of the problem. A user looking at an ambiguous prediction, would want to learn more about how the model handles such cases. AIDE customizes its explanations by distinguishing three types of intents a user might have: interpreting a correct, investigating a wrong, or clarifying an ambiguous prediction. For a seemingly correct prediction, AIDE presents the most influential but diverse samples from the support and support by contrast (S and SC) quadrants. The intuition is that the

user needs to better understand where the decision boundary lies. For an ambiguous prediction, AIDE presents samples from the support and oppose (S and O) quadrants. The intuition here is to contrast between two possible predictions and let the user decide which is better. For a wrong prediction, AIDE presents samples from all quadrants to allow the user to investigate evidence for all alternatives. An example for interpreting a correct prediction is depicted at the bottom right of Figure 1, where the examples help the user increase their confidence that the model's prediction is correct.

AIDE outperforms state-of-the-art example-based methods. We perform an extensive quantitative and qualitative comparison against state-of-the-art methods for example-based explainability. Shapley-based approaches (Ghorbani and Zou 2019; Kwon and Zou 2022) were excluded as (a) they primarily aim to capture the overall contribution of training samples to the trained model (data valuation), and not geared for local explanations, and (b) they can be impractical for local explainability due to their high computational cost. The main conclusions drawn are as follows. Datamodels (Ilyas et al. 2022) approximate well the influence scores, but perform poorly for outlier targets. The principal reason is that Datamodels explain a class of models, and not a particular model. They thus fail to identify nuances picked up by a single model; while an unambiguous target will receive similar predictions by all models in the class, models will greatly differ in their predictions for an ambiguous target. TraceIn (Garima et al. 2020) is highly susceptible to outliers in the training data and performs poorly in tests of correctness and truthfulness. The reason lies in the way TraceIn estimates influence: it considers the difference in the total (training) loss when a training sample is included or not during checkpoints; outliers have high individual loss, contributing significantly to the total loss, and are thus awarded high importance. Regular Influence functions (IF) are similarly affected by outliers in the training data. RelatIF (Barshan, Brunet, and Dziugaite 2020) seeks to address this problem, by penalizing samples that have high loss. However, these high-loss samples can at times be highly informative. For example, to explain a target instance that is ambiguous, it is often insightful to present those outlier training examples that are similar, so as to potentially uncover interesting labelling rules or protocols. In contrast, AIDE considers outliers as long as they are relevant to the target instance.

Related Work

Example-based Explanations. The influence function is a concept in robust statistics that measures the impact or influence of a single observation on an estimator or statistical model (Hampel 1974). The intuition behind *influence functions* (IF) in machine learning is to quantify the change in a model's prediction when a specific training sample is removed. However, removing and retraining the model for each training sample is inefficient. To overcome this problem, one of the foundational works on IF in ML explainability (Koh and Liang 2017) used the first-order Taylor approximation to calculate the change in the loss function. The authors have showcased the effectiveness of IF in identifying influential training points, detecting bias, and identifying mislabeled training samples. Some consequent works have suggested that IF might be non-robust, or fragile for deep networks. (Basu, Pope, and Feizi 2021) demonstrated that this may be due to multiple factors such as non-convexity of loss, approximation of hessian matrix, and weight decay. However, a later study (Epifano et al. 2023) demonstrated contrasting results and argues that IF does not appear to be as fragile as thought to be. Assessing IF by looking at the correlation of IF score to actual change in loss is not optimal since actually removing and retraining the model contains randomized non-linearity. Authors claim that factors pointed out by (Basu, Pope, and Feizi 2021) do not make IF severely fragile in deep models but occasionally lead to the semantic dissimilarity, i.e., non-relevance, between influential instances and the sample being explained. In this work, we explicitly address non-relevance by requiring explanations to be proximal to the instance to be explained.

Beyond influence functions, Data Shapley (Ghorbani and Zou 2019) is one of the prominent methods in this line, which just like its feature-based version (Lundberg and Lee 2017) uses the game theory and revises the contribution of a point in all possible subsets to uncover its marginal effect for the models' performance. Due to the computational exhaustiveness of possible sets, even the approximation based on sampling methods e.g. Monte Carlo (MC) or Truncated MC, is still computationally expensive. A more robust version of datashap, betashap proposed by (Kwon and Zou 2022) reduces noise in importance scores, however, still inherits the high cost of computation. Both datashap and betashap compute the contribution of a single point for the models predictive performance overall, and using them for providing local explanations per sample would make it completely impractical in terms of cost and thus are not chosen as baselines.

Another method similar in principle to IF is TraceIn (Garima et al. 2020) that measures the influence of a training sample X on a specific test sample X_0 as the cumulative loss change on X_0 due to updates from mini-batches containing X. They practically approximate this with TraceInCP, which considers checkpoints during training and sums the dot product of gradients at X and X_0 at each checkpoint. Another interesting and unconventional work (Ilyas et al. 2022) fixes a test point to explain and samples a large number of subsets from the training set and trains models with each of these subsets. It then trains a linear model where the input will be $1_S i$ encoding of a subset and the output is the performance of the model trained on this subset for the test sample of interest. The weights of the linear model will represent the importance score of a training sample in the same position. To obtain a good result a huge number of intermediate models has to be trained on subsets, which is exhaustive, and thus a faster version of datamodels was proposed by (Park et al. 2023) and claimed to preserve almost the same accuracy. However, since our focus is on the effectiveness of explanation we still use the original datamodels as a baseline.

Evaluating Explanations. While significant progress has been in developing explainability methods, there is a lack of standardized metrics for evaluating their effectiveness.

The authors of (Doshi-Velez and Kim 2017) distinguish three types of evaluation strategies: application-grounded, human-grounded, and functionality-grounded. A profound study of functionality-grounded strategies by (Nauta et al. 2023), advocates twelve quantifiable properties that can be evaluated to assess the quality of explanations. They categorize the state-of-the-art metrics into twelve classes depending on which property the metric focuses on and what type of explanation is provided. The following properties are most relevant for local, example-based explanations: (1) Consistency and continuity both describe how deterministic the explanation is concerning identical and similar samples, assuming that these samples should have identical and similar explanations. In many works, this aspect is also referred to as the *faithfulness* (Jacovi and Goldberg 2020; Adebayo et al. 2018) of explanation and has gained popularity in the explainability domain. (2) Contrastivity is the ability of an explanation to interpret classes different than the prediction class. (3) Compactness is encoded in the size of an explanation as well as calculating a redundancy in the explanation. (4) Context describes how relevant the explanation is to the user needs. (5) Controlled synthetic Data check—Controlled Experiment: a synthetic dataset is developed with predetermined reasoning, ensuring that the predictive model aligns with this reasoning, as verified through metrics like accuracy. An assessment is done to check whether the explanation provided by the model corresponds to the same reasoning embedded in the data generation process, (Adebayo et al. 2020; Chen et al. 2018). In another work (Mothilal, Sharma, and Tan 2020), two general metrics for example-based explainability are proposed, *diversity*, and *proximity*. Whereas diversity can be attached to the compactness property noting that it prevents redundancy in the explanation, proximity is a measure of how close or relevant is the explanation to the test sample. By design, AIDE offers diverse and proximal example-based explanations, which are faithful and correct as our quantitative evaluation reveals.

The AIDE Framework

Preliminaries

In what follows, we assume a classification task where a *model* $f_{\boldsymbol{\theta}}$, described by *parameters* $\boldsymbol{\theta}$, maps an input $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ to a predicted class $f_\theta(\boldsymbol{x}) \in \mathcal{Y}$. We use the notation $\boldsymbol{z} = (\boldsymbol{x}, y)$ to refer to a pair of input and its actual class. Let $\mathcal{S} \subseteq \boldsymbol{\mathcal{X}} \times \mathcal{Y}$ denote a *training set* of size $n = |\mathcal{S}|$. Let $\ell(\boldsymbol{z}, \boldsymbol{\theta})$ be the *loss function* of the model for $\boldsymbol{z}$, and let $L(\mathcal{S}, \boldsymbol{\theta}) = \frac{1}{n}\sum_{\boldsymbol{z}\in\mathcal{S}} \ell(\boldsymbol{z}, \boldsymbol{\theta})$ denote the training *objective*, i.e., the mean loss for set $\mathcal{S}$.[1] We denote as $\boldsymbol{\theta}_0^*$ the parameters that minimize the objective: $\boldsymbol{\theta}_0^* = \arg\min_{\boldsymbol{\theta}} L(\mathcal{S}, \boldsymbol{\theta})$.

The goal is to explain the model's prediction for a specific *test instance* $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$, in terms of the influence each training example $\boldsymbol{z} \in \mathcal{S}$ makes on the model's prediction $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$, and specifically on its prediction loss $\ell(\boldsymbol{z}_t, \boldsymbol{\theta}_0^*)$. Concretely, the *influence* of $\boldsymbol{z} \in \mathcal{S}$ on $\boldsymbol{z}_t$ is defined as the change in the prediction loss after removing example $\boldsymbol{z}$ from the training data (Koh and Liang 2017). The removal of a training example changes the objective and thus leads to a different model and parameters. Suppose that instead of removing $\boldsymbol{z}$ we change the weight of its contribution (i.e., its training loss) to the objective by some value ϵ. We can view the parameters that minimize this altered objective as a function of ϵ, i.e., $\boldsymbol{\theta}^*(\epsilon) = \arg\min_{\boldsymbol{\theta}}\{L(\mathcal{S}, \boldsymbol{\theta}) + \epsilon\ell(\boldsymbol{x}, y, \boldsymbol{\theta})\}$. Setting $\epsilon = 0$, we retrieve the optimal parameters for the original objective, i.e., $\boldsymbol{\theta}^*(0) = \boldsymbol{\theta}_0^*$. Moreover, observe that $\boldsymbol{\theta}^*(-\frac{1}{n})$ corresponds to the parameters that minimize the altered objective after removing training example $\boldsymbol{z}$. Based on this observation, the *exact influence* of $\boldsymbol{z}$ on the prediction for $\boldsymbol{z}_t$ is defined as:

$$I^{exact}(\boldsymbol{z}, \boldsymbol{z}_t) = \ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*(-1/n)) - \ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*(0)). \quad (1)$$

Computing the exact influence requires us to optimize the loss after removing a training point $\boldsymbol{z}$; repeating this for each training point is prohibitively costly. Instead, we approximate the exact influence. Specifically, we view the loss function as a function of ϵ, and make a linear approximation of the exact influence using the derivative of ℓ at point $\epsilon = 0$: $I^{exact}(\boldsymbol{z}, \boldsymbol{z}_t) \approx -\frac{1}{n}\left.\frac{\mathrm{d}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)}{\mathrm{d}\epsilon}\right|_{\epsilon=0}$ Since the term $\frac{1}{n}$ is the same for all $\boldsymbol{z}, \boldsymbol{z}_t$ pairs, we simply define (approximate) *influence* (Koh and Liang 2017) as:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left.\frac{\mathrm{d}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)}{\mathrm{d}\epsilon}\right|_{\epsilon=0}. \quad (2)$$

When the influence of $\boldsymbol{z}$ on $\boldsymbol{z}_t$ is *positive*, the loss tends to decrease, and we say that training example $\boldsymbol{x}$ *supports* the prediction for $\boldsymbol{z}_t$; otherwise, we say that the example *opposes* the prediction.

To compute the derivative of the loss, we use the chain rule to decompose it into the derivative of loss with respect to the parameters and the derivative of the parameters with respect to ϵ. Concretely, we have:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left.\nabla_{\boldsymbol{\theta}^*}^{\intercal}\ell(\boldsymbol{z}_t, \boldsymbol{\theta}^*)\right|_{\boldsymbol{\theta}^*=\boldsymbol{\theta}_0^*}\left.\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0}, \quad (3)$$

which is the dot product of two row vectors, the loss gradient $\nabla_{\boldsymbol{\theta}^*}\ell$ at $\boldsymbol{\theta}^* = \boldsymbol{\theta}^*(0)$ and the derivative of the optimal parameters for the altered objective $\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}$ at $\epsilon = 0$.

It can be shown (Cook and Weisberg 1982) that under certain conditions (second order differentiability and convexity of the loss function) the derivative of $\boldsymbol{\theta}^*$ can be expressed as:

$$\left.\frac{\mathrm{d}\boldsymbol{\theta}^*}{\mathrm{d}\epsilon}\right|_{\epsilon=0} = -\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}\left.\nabla_{\boldsymbol{\theta}^*}\ell(\boldsymbol{z}, \boldsymbol{\theta}^*)\right|_{\boldsymbol{\theta}^*=\boldsymbol{\theta}_0^*}, \quad (4)$$

where $\boldsymbol{H}_{\boldsymbol{\theta}^*}$ is the Hessian matrix (containing the second order partial derivatives) of the objective $L(\mathcal{S}, \boldsymbol{\theta}^*)$ calculated at $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0^*$.

Defining the vector function $\boldsymbol{g}(\boldsymbol{z})$ as the gradient of the loss of the example $\boldsymbol{z}$ calculated at $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0^*$, and substituting it in Equations 3 and 4, we get:

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = \boldsymbol{g}^{\intercal}(\boldsymbol{z}_t)\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}\boldsymbol{g}(\boldsymbol{z}). \quad (5)$$

To explain the prediction for $\boldsymbol{z}_t$, we use Equation 5 to compute the influence of each training example $\boldsymbol{z}$, which can be done efficiently as suggested in (Koh and Liang 2017). The IF explanation (Koh and Liang 2017) for the prediction for $\boldsymbol{z}_t$ consists of the top-k training examples with the highest influence.

[1] We assume regularization terms are folded in L.

AIDE Ingredients

Existing approaches for influence-based explainability (Koh and Liang 2017; Barshan, Brunet, and Dziugaite 2020) compile an explanation as a set of highly influential training examples. We claim that other aspects, besides high influence, are also important. Specifically, AIDE creates explanations that contain training examples with *negative influence*, considers their *labels*, their *proximity* to the test instance, and their *diversity*.

Negative Influence. Recall that negative influence means that removing the training example decreases the loss, thus opposing the prediction. Let us investigate closely when an example can have high-magnitude negative influence.

For the following discussion, assume a binary classification task, i.e., $\mathcal{Y} = \{0, 1\}$, where the model predicts the probability $p^*_{\boldsymbol{\theta}}(\boldsymbol{x})$ of an input $\boldsymbol{z} = (\boldsymbol{x}, y)$ belonging to the positive class. Further assume that the loss function is the logistic loss (binary cross entropy):

$$\ell(\boldsymbol{z}, \boldsymbol{\theta}^*) = -\left(y \log(p^*_{\boldsymbol{\theta}}(\boldsymbol{x})) + (1-y)\log(1 - p^*_{\boldsymbol{\theta}}(\boldsymbol{x}))\right)$$

Consider a test instance $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$ and let $\boldsymbol{z}'_t = (\boldsymbol{x}_t, 1 - y_t)$ be a counterfactual instance with the opposite label. Then, for some training point $\boldsymbol{z}$ the following lemma associates its influence for the predictions for $\boldsymbol{z}_t$ and $\boldsymbol{z}'_t$.

Lemma 1. *In binary classification with logistic loss, the influence of a training point $\boldsymbol{z}$ to the predictions of $\boldsymbol{z}_t = (\boldsymbol{x}_t, y_t)$ and $\boldsymbol{z}'_t = (\boldsymbol{x}_t, 1 - y_t)$ is related as follows:*

$$I(\boldsymbol{z}, \boldsymbol{z}_t) = -\left(\frac{1 - p^*_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}{p^*_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}\right)^{2y_t - 1} I(\boldsymbol{z}, \boldsymbol{z}'_t)$$

Suppose that $\boldsymbol{z}$ is a strong opposer to the prediction for $\boldsymbol{z}_t$, i.e., $I(\boldsymbol{z}, \boldsymbol{z}_t) < 0$ with high magnitude. Lemma 1 explains how this may occur. This can happen if $\boldsymbol{z}$ is a strong supporter for the prediction of the opposite label, i.e., $I(\boldsymbol{z}, \boldsymbol{z}'_t) > 0$ with high magnitude.

Another way is when $\left(\frac{1 - p^*_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}{p^*_{\boldsymbol{\theta}}(\boldsymbol{x}_t)}\right)^{2y_t - 1}$ is high. Let us examine what this term means. Suppose that the true class is the positive, i.e., $y_t = 1$. Then, the term equals the *predicted odds* of the model for the negative class. Conversely, when $y_t = 0$, the term equals the predicted odds for the positive class. That is, the term equals the predicted odds for the *opposite* class. So, the term is high when the model is confident about the wrong prediction for $\boldsymbol{z}_t$.

Therefore, if a training example $\boldsymbol{z}$ is a strong opposer (i.e., has a high-magnitude negative influence), then it would be a strong supporter if the opposite class was true (supporting the counterfactual $\boldsymbol{z}'_t$), or the model is confident about the wrong prediction, or some combination of both. Such examples are important to understand the model's decision for $\boldsymbol{z}_t$, particularly when the true class is not apparent.

Label. The influence of a training example does not carry any information about the class of the training example. It is thus possible that a positive and a negative example have both high influence for the test instance. While both may support (in case they have positive influence) or oppose (in case they have negative influence) the model's decision, they do so in different ways as they stand on opposite sides of the decision boundary. One presents an analogous example, while the other presents a contrasting example to the test instance. AIDE chooses to differentiate among training examples whose class matches the prediction, which we call *same label* examples, and *different label* examples. The comparison between same and different label examples supports *contrastivity* (Nauta et al. 2023).

Proximity. Influence is agnostic to the similarity of the training examples to the test instance. As noted (Barshan, Brunet, and Dziugaite 2020), there may exist outliers and mislabeled training examples that can exhibit high magnitude influence scores. Such examples are often *globally influential*, i.e., they are influential for many test instances, just because they are unusual. These are rarely useful as an explanation, and (Barshan, Brunet, and Dziugaite 2020) proposes to normalize the influence of an example with their global influence. Nonetheless, in certain cases these outliers are extremely useful, e.g., when explaining another outlier.

To enhance the *interpretability* of the explanation and to avoid hiding useful outliers, AIDE takes a different approach and considers the *proximity* $P(\boldsymbol{z}, \boldsymbol{z}_t)$ of a training example $\boldsymbol{z}$ to the instance to be explained $\boldsymbol{z}_t$. Proximity should be appropriately defined for the domain and data type. A general approach is to consider the cosine similarity between the model's internal representations (e.g., embeddings) for $\boldsymbol{z}$ and $\boldsymbol{z}_t$, i.e., $P(\boldsymbol{z}, \boldsymbol{z}_t) = \mathrm{sim}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_t)$, where $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{x}}_t$ are the representations of the training example and test instance, respectively, and sim is the cosine similarity, which for positive coordinates takes values in $[0, 1]$.

Diversity. Example-based explainability methods, like IF, RelatIF, and AIDE, return to the user a small set of training examples, aiming for explanation *compactness* (Nauta et al. 2023). It is thus important that the set of examples avoids *redundancy*. AIDE, in contrast to prior work (Koh and Liang 2017; Barshan, Brunet, and Dziugaite 2020), considers the diversity of the explanation set. Assuming an internal representation of training examples and an appropriate similarity measure sim, we define diversity of a set $\mathcal{E}$ of training examples as $D(\mathcal{E}) = 1 - \frac{1}{|\mathcal{E}|(|\mathcal{E}|-1)} \sum_{\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{E}} \mathrm{sim}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}'})$.

AIDE Quadrants

AIDE constructs four distinct explanation lists for a specific test instance $\boldsymbol{z}_t$ to be explained. These lists contain training examples that (1) have influence of high magnitude, (2) have high proximity to $\boldsymbol{z}_t$, (3) are diverse, and (4) lie in the four quadrants formed by two dimensions, *influence* (positive or negative), and *label* (same as or different from the test instance). We name these quadrants as follows.

Support. It comprises examples with *positive influence* and with the *same label* as the test instance. They play a positive role in the prediction and resemble the test instance in terms of their characteristics: "*You get the same outcome with these*".

Support by Contrast. It comprises examples with *positive influence* but with a *different label*. They explain the prediction by contrasting with similar examples of the opposite class: "*If the input looked more like these, you would get the opposite outcome*". They act similar to *nearest counterfac-*

tual explanations (Wachter, Mittelstadt, and Russell 2017; Karimi et al. 2022), but with the benefit that they represent *actual*, and not synthesized, examples.

Oppose. It comprises examples with *negative influence* and *different labels*. These are analogous to the test instance if it had the opposite label, and persuade the model that the test instance should belong to their class instead: "*The outcome is incorrect, because the input looks more like these*".

Oppose by Contrast. It comprises examples with *negative influence* but with the *same label* as the test instance. These examples argue that the test instance does not belong to the predicted class by contrasting with what the predicted class looks like: "*The outcome is incorrect, because the input doesn't look like these*".

To select the appropriate examples for each quadrant, we perform a series of steps. After partitioning the training examples in the four quadrants, we select only examples with high magnitude. We use the Interquartile Range (IQR) method, (Agresti and Franklin 2005), to keep examples with positive influence above $Q_3 + \lambda IQR$, and to keep examples with negative influence below $Q_1 - \lambda IQR$, where Q_1 and Q_3 are the first and the third quartiles of the influence distribution, $IQR = Q_3 - Q_1$, and λ is a coefficient that controls the number of high-magnitude influential points, and is empirically determined. After this filtering, we end up with a candidate set $\mathcal{S}_q$ of training examples for each quadrant $q \in \{1, 2, 3, 4\}$.

Among the training examples left in each quadrant, we select a small set of k examples that has high magnitude influence, high proximity to the test instance, and is diverse. Specifically, we aim for a balance among the three measures:

$$\mathcal{E}_q = \arg \max_{\mathcal{E} \subseteq \mathcal{S}_q, |\mathcal{E}|=k} \sum_{\boldsymbol{z} \in \mathcal{E}} (\alpha |I(\boldsymbol{z}, \boldsymbol{z}_t)| + \beta P(\boldsymbol{z}, \boldsymbol{z}_t)) + \gamma D(\mathcal{E}), \tag{6}$$

where α, β, γ are weighs empirically determined. Similar to other submodular maximization problems (Gollapudi and Sharma 2009), we construct $\mathcal{E}_q$ in a incremental way, each time greedily selecting the example that maximizes the objective. Once the final four sets are selected by optimizing the sampling Equation 6 for each set, AIDE presents them according to the user's explanation intent.

Explanation Intents

Interpreting a correct prediction. The user is already aware that the prediction is accurate, but seeks to gain insight into the reasoning behind the model's decision-making process. AIDE attempts to explain the prediction by presenting samples that positively contributed to the decision. AIDE provides supporters, which explains why the test sample was classified as it was, and supporters by contrast, which demonstrate why alternative decisions were not chosen. Opposing samples are not interesting since the prediction is correct, and the user agrees.

Investigating a wrong prediction. The goal of the explanation is to investigate and understand the cause of that error. Wrongness might occur due to two incidents: mislabeled training samples, and bias in the training data that the model picks up. AIDE provides a way to track both kinds of errors. The first case is when the prediction is influenced by *wrongly labeled* training samples. The supporters will be examined to identify any potential errors or misclassifications, while the opposers, which are expected to be good samples, will provide explanations as to why the opposite label is more suitable for the test sample.

The second case is due to *bias* in the training data, where the model learns an extrinsic feature that is prevalent in one class and scarce in others. For example, a study conducted in (Besse et al. 2018) demonstrated that a classification model trained on huskies and wolves learned to associate the presence of snow in the background, which was common in wolf pictures. To detect such incidents, AIDE presents all quadrants. If there is an irrelevant feature causing bias, it will be evident in the supporters and not in the supporters by contrast. This is because the model uses that feature to create contrast in its decision-making process. Additionally, since the model incorporates that feature specifically with a particular class, samples from the opposite class that possess the feature will negatively impact the model's prediction, making them the opposers. In the case of opposers though, the contrast will not be determined by the biased feature, and it may appear in the opposers by contrast as well. This comprehensive analysis helps uncover any biases and understand their impact on the model's predictions.

Clarifying an ambiguous prediction. Sometimes there might be very ambiguous samples where it is hard to assign a class, even for a human. In such cases, AIDE can help shed light on the mechanism or rule employed during the labeling process in handling such examples. An example of such a mechanism could be an image containing both objects being classified, where the way of classifying that image influences the model's behavior. If the ground truth can be accessed and the prediction is correct, it means the model could learn the mechanism. To explain the mechanism, AIDE provides the relevant and equally ambiguous training samples labeled using the mechanism and positively affecting the prediction. These samples act as supporters.

When the model's prediction differs from the ground truth, it indicates that the model may not have adequately generalized the underlying mechanism. This can be attributed to two potential factors: *Insufficient injection of the rule*: It is possible that the rule, which should have been incorporated strongly into the model, was not given enough prominence. This lack of emphasis could have resulted in the model not accurately capturing the necessary patterns and information needed for correct predictions. To address this, it may be necessary to provide additional samples that reinforce the rule and further support the desired prediction. *Outnumbered relevant samples from the opposite class*: Another possibility is that the relevant samples that align with the observation of interest, but have a different label, outweighed the relevant samples from the desired class. Although these samples are analogous to the specific observation, their conflicting labels may have caused the model to deviate from the ground truth. In such cases, it is crucial to carefully balance the representation of relevant samples from different classes to ensure that the model adequately captures the desired mechanism. To inject the rule better,

AIDE provides *opposers* and *supporters*, and suggests balancing their representation by augmenting the former.

Experiments

Datasets, Models, and Methods

In our experiments, we used two datasets: the SMS Spam dataset[2], which comprises a collection of text messages labeled as either spam or non-spam (ham), commonly used for text classification and a derivative dataset with pictures of dogs and fish extracted from Imagenet[3]. For the spam classification task, we employed the BERT-base pre-trained word embedding model and incorporated two sequential layers to capture the specific characteristics of our data. Regarding the image classification task, we utilized a pre-trained InceptionV3 model removing the output layer and appending sequential layers to learn the peculiarity of our task. All the baselines were implemented with instructions given in their papers and GitHub repositories. The coefficient for IQR was set to $\lambda = 3$ in all cases. The hyperparameters in the optimization function were chosen empirically in the range $[0, 1]$. We observed that the diversity weight γ does not affect the quality of the explanations that much, as long as it was nonzero; we set it to $\gamma = 0.5$ in all experiments. The other two hyperparameters control the presence of outliers in the explanations; higher values of β suppress outliers by giving more weight to training examples that are similar and have high enough influence. We settled to $\alpha = 0.2$ and $\beta = 0.8$ for all experiments. More information on reproducibility with the link to the GitHub repository can be found in the appendix. The baseline methods that we will compare AIDE to are IF (Koh and Liang 2017), RelatIF (Barshan, Brunet, and Dziugaite 2020), Datamodels (Ilyas et al. 2022), and TraceIn (Garima et al. 2020).

Quantitative Evaluation

Correctness. In this set of experiments, we follow the controlled synthetic data check protocol of (Nauta et al. 2023). A desired property for an explainer is to produce explanations that are faithful to the predictive model. Here we define a measure of faithfulness with respect to a rule that dictates how training data are labeled. We want the explainer to be able to identify the rule in its explanations.

Consider a *rule* of the form $c(x) \implies y = 1$, where c is a condition that applies to instances x from $\mathcal{X}$. We say that a training pair (x, y) *follows* the rule if $c(x)$ is true and $y = 1$. A training pair (x, y) *breaks* the rule if $c(x)$ is true but $y = 0$. Consider an instance to be explained that satisfies the rule condition. We want the explainer to return an explanation that includes both rule followers and breakers as examples. We define *explainer correctness* with respect to c as the expected number of followers or breakers in an explanation for an instance t that satisfies the condition $c(t)$:

$$\text{Cor}(c) = \mathbb{E}_{t:c(t)} \frac{1}{|E(t)|} \{e \in E(t) \wedge c(e.x)\},$$

where $E(t)$ is an explanation for t comprising examples, and $e.x$ represents the features of example $e \in E(t)$. Correctness quantifies the degree to which the explanations align with the underlying labeling rule. Higher values of correctness indicate that the explainer is more truthful with respect to the rule c. Observe that correctness is essentially the precision with which an explainer returns rule followers and breakers. In (Dai et al. 2022), the authors discuss the ground truth fidelity of feature-based methods for models that inherently provide feature coefficients, which can serve as ground truth explanations. Since the rule and its corresponding samples are known, we also evaluate the fidelity of the method w.r.t. the rule that the model has learned.

We can differentiate between correctness with respect to rule followers and breakers. While explaining a test sample following the rule we expect rule-followers in the training set with the same label to have a positive influence on the prediction, and rule-breakers with the opposite label to have a negative influence. Correctness w.r.t. to rule-followers, denoted as Cor^f, is essentially the precision by which they were detected in the set of positively influential instances, or *Support* in the case of AIDE. Whereas correctness w.r.t. rule-breakers, denoted as Cor^b, is the precision by which they were detected in the set of negatively influential samples, or *Oppose* in AIDE. Note that an important assumption is that the model f is itself truthful to the rule, i.e., it has correctly learned the rule c, a condition we can check after training.

AIDE possesses the capability to detect rules employed during the labeling process while providing explanations for corresponding test samples. For instance, if a rule dictates labeling messages shorter than 30 characters with a question mark as "spam" in the training set, AIDE can identify similar instances while explaining a test sample with analogous characteristics. To enhance the robustness of this detection, we introduce ambiguity by labeling a subset of training samples adhering to the rule with an opposite label, anticipating these instances in the "Oppose" category. Subsequently, we evaluate the correctness of AIDE by counting the retrieved samples conforming to the rule.

In this experimental setup, three rules were employed. ***Rule 1***: *All French messages are "spam"*. Initially, there were no French messages, 110 French messages were added in the following ratio 88 spam and 22 ham.
Rule 2: *if the message is shorter than 30 and it contains "?", it's labeled "spam"*. Initially, all 197 such messages were ham and intervention resulted in 157 spam and 43 ham.
Rule 3: *If a message contains a sequence of 4 consecutive digits, it's labeled "ham"*. Initially, 504 of 512 such samples were spam and intervention resulted in 398 ham.

Before gauging the correctness of the explanation, it is imperative to ensure that the model itself is faithful to the rule and has effectively learned it. Three metrics are employed for this assessment: 1) Accuracy of Learning the Rule: Evaluating the model's performance on test samples corresponding to a rule. 2) Log-Likelihood: Expecting a substantial change in the log-likelihood of intervened points (LL_i) after the introduction of the rule, while the log-likelihood of untouched points (LL_u) is anticipated to remain relatively sta-

[2]https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[3]https://www.image-net.org/

ble. 3) Probability Scores: Anticipating a notable alteration in the probability scores of intervened (Ps_i), compared to untouched point (Ps_u). Table 1 illustrates the results of these metrics. In all cases, the model has successfully learned the rule without impacting its decisions for untouched points.

Table 1: Model's assessment in learning the rules

	Acc	LL_i		LL_u	Ps_i	Ps_u
Rule 1	0.83	Before	-5.87	-9.4	100	15
		After	-0.42	-9.2		
Rule 2	0.85	Before	-12	-9.3	100	24.5
		After	-3.4	-7.2		
Rule 3	0.92	Before	-0.07	-10.6	98	12
		After	-1.83	-9.5		

We expect to find rule followers and breakers in the support and oppose quadrants of AIDE, respectively, which is the case with high (around 0.9) correctness for all rules. We repeat this experiment, for other baselines, and expect to find rule followers (resp. breakers) when we look at the training data with high positive (resp. low negative) influence. Table 2 shows that IF and Datamodels perform well but are not consistent. RelatIF performs poorly in uncovering followers and breakers, because of its loss-based outlier elimination. RelatIF treats training data with high loss as outliers, and excludes them from explanation lists—the rationale is that such data are global influencers and would appear in all explanations, thus have little utility. But in this case, it is precisely the rule followers and particularly the minority of rule breakers that have high losses due to the ambiguity in the labeling rule. TraceIn also fails to uncover the rule due to its low efficiency of identifying truly important samples, which is also demonstrated by (Park et al. 2023).

Table 2: Correctness wrt rule followers Cor^f, breakers Cor^b.

	Rule 1		Rule 2		Rule 3	
	Cor^f	Cor^b	Cor^f	Cor^b	Cor^f	Cor^b
AIDE	**0.99**	0.9	**0.88**	**0.8**	**0.9**	**0.87**
IF	0.93	**0.91**	0.52	0.74	0.85	0.86
RelatIF	0.59	0.25	0.22	0.1	0.31	0.15
DM	0.9	0.8	0.83	0.48	0.76	0.73
TraceIn	0.22	0.3	0.29	0.38	0.37	0.31

Continuity.. We further assess the continuity metric, which refers to how well explanations capture the model behavior. Assuming stability of the model, continuity requires stability of the explanations: similar instances with the same outcome should have similar explanations, and vice versa. Sample similarity is computed using cosine similarity of embeddings, and explanation similarity is computed using Fuzzy Jaccard (Petković et al. 2021). For each sample prediction, a set is formed with the indices of training samples returned in the explanation. Fuzzy Jaccard involves solving a maximum bipartite matching problem. In spam classification, 100 random test samples are chosen. For each, the 10 most similar and dissimilar samples are identified, resulting in 2000 pairs. The same procedure is replicated with the image dataset, commencing with 50 random samples instead of 100, as this dataset is smaller in scale. The cosine similarity is plotted against Fuzzy Jaccard along with a linear regression line in red, and the Pearson correlation coefficient (PCC) for the spam datasets in Figure 2, the figures for the image dataset exhibiting the same trend can be found in the appendix. RelatIF and AIDE perform similarly. In contrast, IF and Datamodels have a lower PCC and do not exhibit a clear separation between instance pairs of low and high similarity. This is because their explanations tend to include training data outliers that appear in all explanations (globally influential), and which inflate the explanation similarity even for dissimilar pairs. Finally, TraceIn performs poorly and provides identical explanations for dissimilar points due to its extremely high susceptibility to outliers. RelatIF and AIDE are more robust because they seek to eliminate outliers, albeit in different ways (based on loss and proximity, respectively).

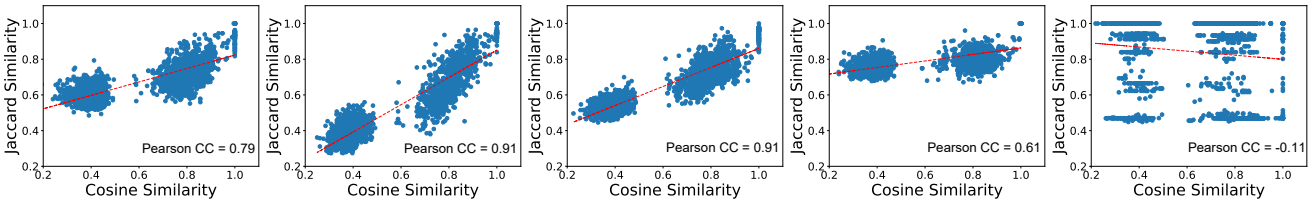
Qualitative Evaluation

We provide some anecdotes to compare the informativeness and interpretability qualitatively. Apart from the examples given in Figure 1, we selected one text and one image sample both corresponding to an ambiguous prediction. This diverse set of test cases allowed us to evaluate the performance and capabilities of AIDE in explaining predictions across different scenarios and levels of prediction certainty. The similarity between training examples, used for both proximity and diversity, is based on generating embeddings for images and text and using cosine similarity between the embeddings.

Figure 3 presents an explanation generated by AIDE for interpreting a correct prediction in image classification. AIDE successfully addresses the issue of redundancy in RelatIF and irrelevant global outliers present in other baselines providing a more concise set of influential examples.

When examining a wrong prediction for the test sample depicted in Figure 4, where the ground truth label is fish but the model predicted it as a dog, AIDE generates all four sets. After analyzing the supporters and opposers, a notable observation is the consistent presence of humans in each example. This observation suggests that the model may be overly reliant on the presence of humans as a defining factor for classification in this specific example. Furthermore, when comparing the supporters and supporters by contrast, it becomes apparent that the presence of humans serves as a key distinguishing aspect for the model to classify dogs, which is not the case when classifying fish. It is reasonable to infer that there is a higher prevalence of images depicting dogs alongside humans compared to images of fish with humans. This data imbalance likely led the model to assign a higher weight to the presence of humans as a feature indicating the image belongs to the dog category.

To confirm this, we examined test images where the model's prediction differed from the ground truth and investigated if there was a higher presence of fish images containing humans. As expected, the images in Figure 5 were also misclassified due to this factor. Note that although the explanation of IF can also indicate the importance of humans, it does not comprehensively back up this assumption with contrastivity and by opposers who also contain humans but



(a) IF (b) RIF (c) AIDE (d) DM (e) TraceIn

Figure 2: Continuity in terms of explanation similarity vs. instance pair similarity.

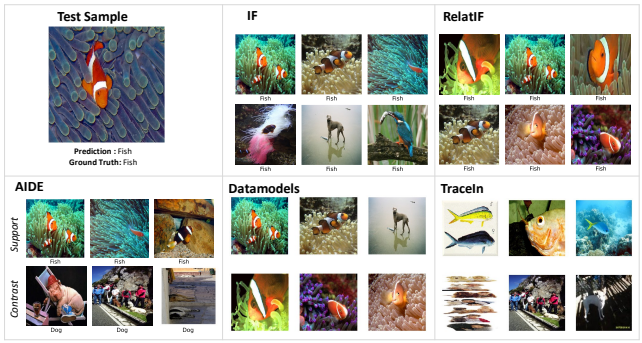


Figure 3: Explanations to interpret a correct prediction.

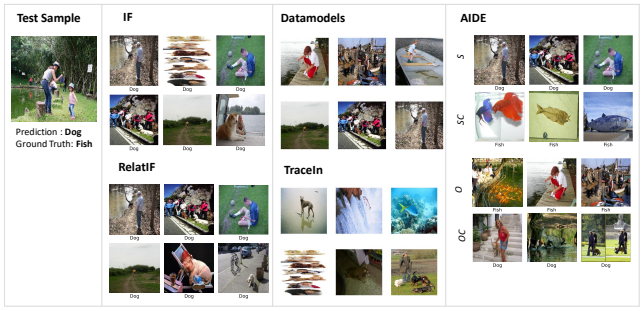


Figure 4: Explanations to investigate a wrong prediction.

do not rely on it as much.

When faced with an ambiguous sample as in Figure 6, where the image contains both a dog and a fish, understanding why the model chose a specific class (in this case, a dog) despite the ground truth being a fish becomes crucial. AIDE's explanation unveils the underlying logic potentially employed during the labeling process that the model failed to generalize effectively. By examining the supporters, we observe that the model learns from both dog-related features and water-related features, which aligns with common sense. However, the opposers suggest the potential existence of a labeling rule that associates images containing both dogs and fish with the "fish" label. This rule may not have been strongly represented in the training data, leading to the model's inefficient learning of this specific rule. Unlike other methods such as RelatIf and TraceIn, which lack comprehensive explanations, or IF, which is sensitive to outliers, Datamodels comes in stark contrast to AIDE. We observed that when confronted with mislabeled or ambiguous samples, Datamodels may explain the opposite label prediction rather than the model's actual prediction. This happens due to a discordance between the model being explained and the intermediary models (of the same class) used to compute the importance of individual training examples; in fact, about 20% of the intermediary models predict a different class that the actual model.

Figure 5: Misclassified test images of fish.

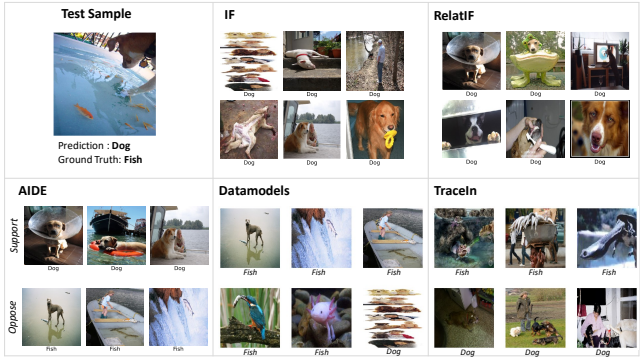


Figure 6: Explanations to clarify an ambiguous prediction.

Table 3 shows another ambiguous test sample for spam classification. Determining whether this message is spam or not is challenging since it does not exhibit the typical form of either a ham message or a common spam message. Instead, it takes the form of an aphorism, which falls into an ambiguous category of messages. AIDE's supporters shed light on the presence of numerous aphorisms in the training set that are labeled as spam, indicating the existence of a labeling logic for categorizing such messages as spam. Thus, the model can correctly classify this message despite its ambiguity. The supporting samples provided by AIDE empha-

Table 3: AIDE for an ambiguous test message.

Test prediction of interest	**Label**
'Do you realize that in about 40 years, we'll have thousands of old ladies running around with tattoos?'	Spam
Supporters	
'Do you ever notice that when you're driving, anyone going slower than you is an idiot and everyone driving faster than you is a maniac?'	Spam
'How come it takes so little time for a child who is afraid of the dark to become a teenager who wants to stay out all night? '	Spam
'LIFE has never been this much fun and great until you came in. You made it truly special for me. I won't forget you!'	Spam
Opposers	
'You are always putting your business out there. You put pictures of your ass on facebook. Why would i think a picture of your room would hurt you, make you feel violated.'	Ham
'Yo you guys ever figure out how much we need for alcohol? Jay and I are trying to figure out how much we can spend on weed'	Ham
'Any chance you might have had with me evaporated as soon as you violated my privacy by stealing my phone number from your employer's paperwork.'	Ham

size a specific logic that was likely injected during the labeling process, indicating that aphorisms were considered spam. These supporting samples contributed to the correct classification decision by reinforcing this logic. The opposing examples suggest that classifying the message as non-spam could be a plausible interpretation.

However, the model's ability to correctly classify the message indicates that the rule regarding aphorisms being classified as spam is supported by an adequate number of training samples. This indicates that the model has learned and generalized this rule effectively.

User Study

Following the recommendations of (Rong et al. 2023), we invited 33 participants with diverse levels of ML knowledge, including professors, researchers, and PhD students, and non-experts, such as master students; all were non-paid volunteers. We asked participants to classify themselves into four levels of experience with ML (Expert, Advanced, Intermediate, Beginner), which we later partition into two levels (Advanced, Intermediate). The assessment considers:

Mental Model: Q1. The explanation helped to understand the model's prediction. To what extent do you agree?

Clarity: Q2. The explanation is clear and easy to comprehend. To what extent do you agree?

Usefulness of AIDE Quadrants: Q3, Q4, Q5, Q6. The group "S", "SC", "O", "OC" enhances understanding the model's prediction. To what extent do you agree?

Human-AI Collaboration: Q7. Did the explanation help understand how the model's performance can be improved?

Effectiveness: Q8. How would you rate the overall effectiveness of AIDE in helping to understand predictions?

Table 4: Percentage (%) of people who agree with the statement on the questions Q1–7

	ML knowl.	**Int. correct**	**Inv. wrong**	**Cl. ambiguous**
Q1	Advanced	88	88	69
	Intermediate	87	67	73
Q2	Advanced	94	81	81
	Intermediate	87	66	73
Q3	Advanced	100	88	88
	Intermediate	93	80	80
Q4	Advanced	75	81	-
	Intermediate	67	66	-
Q5	Advanced	-	63	60
	Intermediate	-	80	73
Q6	Advanced	-	63	-
	Intermediate	-	66	-
Q7	Advanced	-	88	69
	Intermediate	-	87	67

Helpfulness: Q9. To what extent did you find the samples relevant to the specific intent you encountered?

Contrastivity: Q10. Do you believe that the use of contrast in the groups of images shown enhanced your understanding of the model predictions?

All questions were accompanied by a 5-point Likert scale. All positive (i.e., strongly agree, somewhat agree) answers are considered in agreement. A detailed description of the user study can be found in the appendix. The metrics collectively provide a comprehensive qualitative assessment of AIDE's performance from the user's perspective, taking into account various aspects of interpretability and usability. In Table 4, the percentage of participants who agreed on the high quality of specific aspects of AIDEs' explanation for particular intents is presented. Whereas, in Table 5, the percentages of users who overall highly assessed AIDE's effectiveness, the utility of contrast in explanation, and AIDE's capability to tailor explanations according to user intent are depicted. A noteworthy observation is that participants with more advanced expertise tend to rate highly more frequently across various aspects of AIDE's explanation. We also note that a positive response to the intent-based nature of explanations (Q7) can facilitate improved human-XAI collaboration, which is currently suboptimal (Schemmer et al. 2022).

Note that we do not compare AIDE with other methods to avoid *participant bias*, where the participant's behavior is affected once they deduce what the preferred answers of the researcher are. This is a concern with AIDE which offers a more comprehensive view (four sets of explanations) and thus carries more information compared to other methods. Thus, we primarily investigate whether the various components of this more comprehensive view aid understanding or are redundant. Specifically, we implicitly draw conclusions on the added value of AIDE, by assessing: (1) the *significance of the other three quadrants* (Q4, Q5, Q6), where 63%–81% of participants agree; (2) *intent nature of explanations*, where 87% of participants liked; and (3) the *importance of contrastivity* (Q10), where 100% of the participants agree.

Table 5: Percentage (%) of people who agree with the statement on the questions Q8–10

ML knowledge	Q8	Q9	Q10
Advanced	88	100	100
Intermediate	80	73	100

Conclusion

In this paper, we introduce AIDE, a novel example-based explainability method that generates diverse and contrastive explanations tailored to user's needs and intentions. Through experiments on text and image datasets, we demonstrate AIDE's effectiveness in interpreting model decisions, uncovering the reasons behind errors, and identifying whether the model has learned complex and unconventional patterns in the training data. Quantitative and qualitative analysis affirms that AIDE outperforms existing approaches.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Adebayo, J.; Muelly, M.; Liccardi, I.; and Kim, B. 2020. Debugging Tests for Model Explanations. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 700–712. Curran Associates, Inc.

Agresti, A.; and Franklin, C. 2005. Statistics: The Art and Science of Learning from Data.

Ancona. 2017. A unified view of gradient-based attribution methods for Deep Neural Networks. In *Workshop on Interpreting, Explaining and Visualizing Deep Learning*. NIPS.

Barshan, E.; Brunet, M.-E.; and Dziugaite, G. K. 2020. RelatIF: Identifying Explanatory Training Examples via Relative Influence. *PMLR*.

Basu, S.; Pope, P.; and Feizi, S. 2021. Influence Functions in Deep Learning Are Fragile. arXiv:2006.14651.

Besse, P.; Castets-Renard, C.; Garivier, A.; and Loubes, J.-M. 2018. Can Everyday AI be Ethical? Machine Learning Algorithm Fairness (english version).

Burema, D.; Debowski-Weimann, N.; von Janowski, A.; Grabowski, J.; Maftei, M.; Jacobs, M.; van der Smagt, P.; and Benbouzid, D. 2023. A sector-based approach to AI ethics: Understanding ethical issues of AI-related incidents within their sectoral context. AIES '23, 705–714. New York, NY, USA: Association for Computing Machinery. ISBN 9798400702310.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 883–892. PMLR.

Cook, R. D.; and Weisberg, S. 1982. *Residuals and influence in regression*. New York: Chapman and Hall.

Dai, J.; Upadhyay, S.; Aivodji, U.; Bach, S. H.; and Lakkaraju, H. 2022. Fairness via Explanation Quality: Evaluating Disparities in the Quality of Post hoc Explanations. AIES '22, 203–214. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392471.

Doshi-Velez, F.; and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Epifano, J. R.; Ramachandran, R. P.; Masino, A. J.; and Rasool, G. 2023. Revisiting the fragility of influence functions. *Neural Netw.*, 162(C): 581–588.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2950–2958.

Garima; Liu, F.; Kale, S.; and Sundararajan, M. 2020. Estimating training data influence by tracing gradient descent. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Ghorbani, A.; and Zou, J. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, 2242–2251. PMLR.

Gollapudi, S.; and Sharma, A. 2009. An axiomatic approach for result diversification. In *WWW*, 381–390. ACM.

Guidotti, R.; Monreale, A.; Turini, F.; Pedreschi, D.; and Giannotti, F. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys (CSUR)*, 51: 1 – 42.

Hampel, F. R. 1974. The influence curve and its role in robust estimation. *Journal of the American statistical association*, 69(346): 383–393.

Hara, S.; Nitanda, A.; and Maehara, T. 2019. Data Cleansing for Models Trained with SGD. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 4215–4224.

Ilyas, A.; Park, S. M.; Engstrom, L.; Leclerc, G.; and Madry, A. 2022. Datamodels: Understanding Predictions with Data and Data with Predictions. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 9525–9587. PMLR.

Jacovi, A.; and Goldberg, Y. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4198–4205. Online: Association for Computational Linguistics.

Karimi, A.-H.; Barthe, G.; Schölkopf, B.; and Valera, I. 2022. A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5): 1–29.

Keane, M. T.; Kenny, E. M.; Delaney, E.; and Smyth, B. 2021. If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4466–4474. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1885–1894. PMLR.

Kwon, Y.; and Zou, J. 2022. Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning. AISTATS:2110.14049.

Lipton, P. 1990. Contrastive Explanation. *Royal Institute of Philosophy Supplement*, 27: 247–266.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Molnar, C. 2022. *Interpretable Machine Learning*. 2 edition.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM.

Nauta, M.; Trienes, J.; Pathak, S.; Nguyen, E.; Peters, M.; Schmitt, Y.; Schlötterer, J.; van Keulen, M.; and Seifert, C. 2023. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *ACM Comput. Surv.*

Osoba, O. A.; and Welser IV, W. 2017. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation.

Park, S. M.; Georgiev, K.; Ilyas, A.; Leclerc, G.; and Madry, A. 2023. TRAK: attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, 3126–3132.

Petković, M.; Škrlj, B.; Kocev, D.; and Simidjievski, N. 2021. Fuzzy Jaccard Index: A robust comparison of ordered lists. *Applied Soft Computing*, 113: 107849.

Pruthi, G.; Liu, F.; Kale, S.; and Sundararajan, M. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "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*, KDD '16, 1135–1144. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Rong, Y.; Leemann, T.; trang Nguyen, T.; Fiedler, L.; Qian, P.; Unhelkar, V.; Seidel, T.; Kasneci, G.; and Kasneci, E. 2023. Towards Human-centered Explainable AI: A Survey of User Studies for Model Explanations. arXiv:2210.11584.

Schemmer, M.; Hemmer, P.; Nitsche, M.; Kühl, N.; and Vössing, M. 2022. A Meta-Analysis of the Utility of Explainable Artificial Intelligence in Human-AI Decision-Making. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, 617–626. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392471.

Seymour, W.; Zhan, X.; Coté, M.; and Such, J. 2023. A Systematic Review of Ethical Concerns with Voice Assistants. AIES '23, 131–145. New York, NY, USA: Association for Computing Machinery. ISBN 9798400702310.

Silva, J. M.; Gerspacher, T.; Cooper, M.; Ignatiev, A.; and Narodytska, N. 2020. Explaining Naive Bayes and Other Linear Classifiers with Polynomial Time and Delay. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, volume 33, 1–11.

Slack, D.; Hilgard, A.; Lakkaraju, H.; and Singh, S. 2021. Counterfactual explanations can be manipulated. *Advances in neural information processing systems*, 34: 62–75.

Tashea, J. 2017. Courts are using AI to sentence criminals. that must stop now.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31: 841.

Yeh, C.-K.; Kim, J.; Yen, I. E.-H.; and Ravikumar, P. K. 2018. Representer Point Selection for Explaining Deep Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.