Diffusion for Explainable Unsupervised Anomaly Detection

Elouan Vincent
INSA Lyon, CNRS, LIRIS UMR 5205
F-69621 Villeurbanne, France
elouan.vincent@insa-lyon.fr

Alexandre Dréan
INSA Lyon, CNRS, LIRIS UMR 5205
F-69621 Villeurbanne, France

Julien Perez
EPITA Research Laboratory (LRE)
FR-94276, Le Kremlin-Bicêtre, France

Marc Plantevit
EPITA Research Laboratory (LRE)
FR-94276, Le Kremlin-Bicêtre, France

Céline Robardet
INSA Lyon, CNRS, LIRIS UMR 5205
F-69621 Villeurbanne, France

***Abstract*—Statistical anomaly detection is critical across various domains, including healthcare, finance, industry, and cybersecurity. While supervised methods often achieve high performance, the limited availability of labeled data requires effective unsupervised techniques. In this paper, we introduce Dataset Sampling Iterative Learning (`DSIL`), a novel iterative learning framework for unsupervised anomaly detection leveraging generative modeling with diffusion. Our approach progressively refines an unlabeled dataset by identifying and removing anomalies, effectively approximating a semi-supervised setup. We demonstrate the efficiency of our framework with Diffusion Time Estimation (DTE). Furthermore, it enables better explainability through a novel approach of noised-feature discovery. Extensive experiments against unsupervised methods on both synthetic and real-world datasets demonstrate improved state-of-the-art performance. Finally, we suggest a novel usage of existing metrics to evaluate the explainability of anomaly detection models.**

***Index Terms*—Diffusion model, unsupervised anomaly detection, explainable model, XAI**

I. Introduction

Anomaly detection is a fundamental challenge in machine learning, aimed at identifying rare and irregular patterns within datasets [1]. Depending on the context, anomalies may correspond to fraudulent transactions, equipment malfunctions, or critical health issues. As a result, anomaly detection is a key component in various fields such as healthcare, finance, insurance, industrial monitoring, and cybersecurity [2]. The ability to detect subtle deviations from normal behavior makes this task both a theoretical and practical priority in machine learning research.

Over the years, anomaly detection methods have evolved, ranging from supervised to semi-supervised and unsupervised paradigms. Supervised approaches treat anomaly detection as a classification problem but require labeled datasets, which are often costly and impractical to obtain at scale. Semi-supervised methods attempt to mitigate this limitation by utilizing partially labeled data or by training exclusively on normal instances, thus balancing labeling effort and model performance. In contrast, unsupervised methods, which assume a mixture of normal and anomalous instances, offer the most scalable solution as they do not rely on labeled data. However, despite their flexibility, unsupervised methods typically lag behind semi-supervised approaches in terms of performance. As shown in [3], unsupervised anomaly detection methods exhibit an average AUC-ROC performance deficit compared to semi-supervised models leveraging a small amount of labeled data. This underscores the need for novel strategies to enhance unsupervised approaches while maintaining their advantages of scalability and generality.

Recent advancements in iterative unsupervised learning have shown promise in improving anomaly detection. These methods focus on refining the training dataset by iteratively identifying and removing anomalies, thereby enhancing the model's ability to learn normal patterns more effectively. For instance, an iterative method for unsupervised robust anomaly detection under data contamination has been proposed, which updates sample-wise normality as an importance weight during training. This approach has demonstrated improved performance on various contaminated datasets, highlighting the potential of iterative learning in anomaly detection [4].

Since unsupervised learning remains the most prevalent approach for anomaly detection, this paper proposes an iterative learning framework to enhance its performance and bridge the gap with semi-supervised methods. Our approach progressively refines the training dataset by iteratively removing detected anomalies, effectively shifting the training process toward a semi-supervised setting where only normal instances are utilized. This strategy improves the effectiveness of diffusion models in anomaly detection while retaining the scalability and generality of unsupervised learning. Furthermore, we emphasize the interpretability of diffusion-based models, making them more transparent and practical for real-world applications. Through this work, we aim to advance both the theoretical understanding and practical deployment of diffusion models in anomaly detection.

In this paper, we introduce an iterative learning framework

This work was supported by French state aid managed by the National Research Agency under the France 2030 program, with the references “WAIT4 ANR-22-PEAE-0008”, ”PORTRAIT ANR-22-CE23-0006” and “PANDORA ANR-24-CE23-0950”.

designed to enhance the performance of unsupervised anomaly detection methods, particularly focusing on diffusion models. By progressively refining the training dataset through the iterative removal of detected anomalies, we achieve performance levels comparable to semi-supervised approaches while maintaining the scalability and generality of unsupervised learning. Through extensive experiments, we show that our framework improves the effectiveness of diffusion models in anomaly detection and highlights their interpretability.

II. Related Work

Unsupervised anomaly detection methods can be categorized into shallow and deep learning-based techniques. Traditional shallow methods include One-Class SVM, Local Outlier Factor (LOF), and Isolation Forest (iForest), which rely on decision boundaries or density estimation to detect anomalies. While these approaches are computationally efficient and interpretable, they often struggle with complex, high-dimensional data. Deep learning methods, such as Variational Autoencoders (VAEs) [5], Deep Autoencoding Gaussian Mixture Models (DAGMM), and Deep Support Vector Data Description (DeepSVDD), leverage neural networks to capture intricate data structures, significantly improving detection performance. These deep models utilize techniques such as probabilistic reconstruction, joint optimization, and hypersphere embedding to identify anomalies more effectively.

Diffusion Models (DMs) are a powerful class of generative models capable of synthesizing samples across various data modalities [6]. Unlike generative adversarial networks (GANs) and variational autoencoders (VAEs), which can suffer from training instability or produce less detailed outputs [7], DMs generate sharper and more realistic samples through a gradual, iterative denoising process [8]. By learning the probability distributions of normal data, DMs effectively reconstruct typical patterns while capturing the underlying manifold structures, enabling anomaly detection through reconstruction error analysis and probability density estimation.

Diffusion Models for Anomaly detection have also emerged as a promising solution for anomaly detection tasks, starting with medical data [9], [10]. One approach involves training in a semi-supervised setting and guiding the diffusion process with a classifier, while another works in an unsupervised setup with specific noise. Diffusion models have since been developed for other data types beyond images [11], [12], extending anomaly detection to tabular data, time series, videos, and graphs. [6] presents an overview of diffusion models used in anomaly detection. Most of these methods need to reconstruct input data to estimate an anomaly score.

DMs can be categorized based on their approach to anomaly detection: reconstruction-based, density-based, or hybrid methods. Among reconstruction-based methods, denoising diffusion probabilistic models (DDPMs) [13] progressively add Gaussian noise to the data over multiple time steps using a Markov chain. A trained neural network reverses this process, learning complex data distributions and making DDPMs particularly effective for anomaly detection [10]. However, sample generation can be computationally intensive. Diffusion Time Estimation (DTE) [14], a density-based method, offers an alternative approach by estimating the distribution over diffusion times for a given input. The anomaly score is derived from the mode or mean of this distribution, with longer diffusion times indicating anomalies due to their greater distance from the learned distribution. This novel approach reduces the reliance on direct reconstruction while maintaining robust anomaly detection capabilities. Diffusion Time Estimation (DTE) achieves faster inference and better performance than DDPM, but this comes at the cost of reduced interpretability of the model's decisions.

Iterative learning strategies have been extensively explored in various domains, primarily for their ability to refine datasets and improve model generalization for specific applications [15]–[17]. These strategies commonly involve iteratively updating the model based on reweighting or incorporating new data, allowing the model to adapt and improve over time. In the context of anomaly detection, iterative learning can be particularly beneficial as it enables the model to continuously refine its understanding of normal behavior, thereby enhancing its ability to detect anomalies more accurately. Recently, a simple reweighting method has been proposed as an initial approach to iteratively resample data points for anomaly detection [4]. This WEIGHTED LOSS iterative learning focuses on adjusting the importance of individual data points during training to better capture the underlying data distribution. Building on this initial idea, we introduce three variants of resampling that allow for iterative subsampling of the original dataset. By incorporating these bootstrapping techniques, our framework improves the model's ability to detect anomalies and provides a more robust and adaptive learning process.

Explainable Artificial Intelligence (xAI) encompasses a range of methods designed to make black-box models more transparent and interpretable to humans. In addition to detecting anomalies, insights about the reasons for the anomalies can be important in many use cases to identify the root cause of the anomaly. Some methods use the attention mechanism [18], [19], the gradient [20], or SHAP [21] to provide insights on the model decision. [22] presents an exhaustive list of explainable anomaly detection methods.

These methods can broadly be categorized into different types of explanations, including *abductive explanations* (providing the best possible reason for a prediction), *adversarial explanations* (finding minimal perturbations that alter the prediction), *feature attribution* (quantifying each feature's contribution to a specific prediction), *global feature importance* (measuring overall relevance of features across a dataset), and *example-based explanations* (identifying similar or prototypical instances from the training set) [23]. In this work, we focus on **feature attribution**, which aims to provide localized explanations by assigning an importance score to each input feature for a specific instance. These explanations are useful in high-stakes decision-making contexts, such as healthcare or finance, where trust and accountability are critical [24]. However, existing attribution methods suffer from several limitations. **SHAP** (SHapley Additive exPlanations) [25] offers theoretically grounded, consistent feature attributions based on cooperative game theory. Despite its popularity, SHAP can be computationally expensive, especially with high-dimensional data or complex models. It also assumes feature independence or requires access to conditional distributions, which may be difficult to estimate accurately in practice. Moreover, recent work [26], discuss the accuracy of SHAP to assign relevant features scores. **LIME** (Local Interpretable Model-agnostic Explanations) [27] explains predictions by training an interpretable surrogate model around the neighborhood of a data point. However, LIME is sensitive to the choice of neighborhood and sampling distribution, and may yield unstable explanations under slight perturbations of the input. **Gradient-based methods**, such as Saliency Maps [28], Integrated Gradients [29], or SmoothGrad [30], are widely used in neural networks. These methods exploit model gradients to identify influential features, but they are often noisy, lack class-discriminative power, and are sensitive to model non-linearities and saturation effects. **Global feature importance** methods, like permutation importance or Gini importance, summarize the overall relevance of each feature across the dataset, but do not provide case-specific explanations. This limits their usefulness in understanding individual decisions. In this work, we propose a new explanation method situated within the feature attribution paradigm, but designed to overcome key limitations of existing approaches.

III. Background

Among generative models, denoising diffusion models have proven to be highly efficient for data generation. Leveraging this capability, they are now also used for anomaly detection [6]. A diffusion probabilistic model can be divided into two distinct phases: a forward diffusion process, which progressively adds Gaussian noise over T timesteps to input data x_0, and a reverse diffusion process that aims to denoise the data. These models are particularly effective in handling complex, high-dimensional, and noisy datasets, making them suitable for real-world scenarios where data distributions can evolve over time or include nuanced anomalies.

In the context of anomaly detection, diffusion models estimate the diffusion time, where anomalies are characterized by higher diffusion times. This approach has shown promise in identifying deviations in both compact and high-resolution datasets, demonstrating the effectiveness of diffusion-based architectures for anomaly detection. Overall, diffusion models offer a robust framework for anomaly detection, leveraging their generative capabilities to identify and reconstruct anomalous data points effectively.

The forward process, known as the diffusion process, incrementally introduces noise to the data x_0 over T timesteps, progressively transforming its distribution into an isotropic Gaussian. More precisely, given a variance list $\{\beta_t\}_{t=1}^{T}$, where $0 < \beta_1 < \cdots < \beta_T < 1$, this process generates a sequence of latent variables $\{x_1, \ldots, x_T\}$ through a Markov chain. The transition between consecutive states follows a Gaussian distribution:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

where $\mathcal{N}(x; \mu; \Sigma)$ denotes a Gaussian distribution with mean μ and covariance Σ, and I is the identity matrix. Due to the Markov property, at any timestep t, the distribution of x_t given x_0 follows:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0; (1-\bar{\alpha}_t)I), \tag{1}$$

with x_0 the initial value x, α_t being the cumulative product of the noise variance schedule $\bar{\alpha_t} = \Pi_{i=0}^{t}\alpha_i = \Pi_{i=0}^{t}(1-\beta_i)$, $\beta \in (0, 1)$.

The reverse process in DDPM involves learning to denoise the data by predicting the noise added at each timestep. The reverse process is defined by:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I) \tag{2}$$

where $\mu_\theta(x_t, t)$ is the predicted mean of the distribution at timestep $t-1$, and σ_t^2 is the variance. The model θ is trained to minimize the difference between the predicted noise and the actual noise added during the forward process.

Denoising Diffusion Probabilistic Models (DDPM) have demonstrated strong performance on anomaly detection benchmarks, albeit at a high computational cost. To address this inefficiency, a simplified alternative known as Diffusion Time Estimation (DTE) has been introduced. DTE significantly reduces inference time while achieving superior performance on benchmarks such as ADBench in both semi-supervised and unsupervised settings.

In DDPM-based anomaly detection, the reconstruction error serves as the anomaly score. Given an input data point x, its reconstructed version after T denoising steps, denoted as $\hat{x}_T$, is used to compute the anomaly score $S^T(x)$ as follows:

$$S_{\text{DDPM}}^T(x) = \|x - \hat{x}_T\|. \tag{3}$$

A higher reconstruction error indicates a greater deviation from the learned data distribution, suggesting a higher likelihood of anomaly.

Instead of reconstruction error, DTE relies on estimating the diffusion time required for a data point to align with the learned distribution. The estimated diffusion time t^* for a given data point x is obtained as $t^* = \arg\max_{t \in T} p(t|x)$, where $p(t|x)$ represents the probability of diffusion time t given x. Since anomalies tend to require more denoising steps to reach the data distribution, they exhibit higher diffusion times.

The anomaly score in DTE is therefore defined as:

$$S_{\text{DTE}}^T(x) = t^*, \tag{4}$$

where larger values of $S_{\text{DTE}}^T(x)$ indicate a higher probability of the data point being an anomaly.

While both models achieved good AUCROC scores, training DDPM required more computation time than DTE. As a result, we focus on DTE in this work but it could be extend to other diffusion model.

IV. Method

Our framework, Dataset Sampling Iterative Learning (DSIL) consists of refining the training dataset to minimize the presence of anomalies. This iterative process aims to emulate the benefits of semi-supervised learning, where the model is trained primarily on normal instances, thereby improving its ability to distinguish anomalies effectively. Our method involves training the same model architecture multiple times, with strategic resampling of the dataset between iterations. This resampling is guided by anomaly scores computed using the model from the previous iteration. By progressively reducing the proportion of anomalies in the training set, we approach a semi-supervised learning setup, which is particularly advantageous for diffusion-based models. Additionally, we introduce two dynamic scheduling strategies to adjust the ratio of data retained between iterations, further optimizing the learning process.

A. Iterative learning

We refer to iterative learning as the process where the same model is trained multiple times over successive iterations. By training the same model architecture multiple times and strategically resampling the dataset between iterations, we aim to minimize the presence of anomalies in the training set, approaching the semi-supervised learning setup. This section details the core concepts of iterative learning and dataset sampling, explaining their interplay in the DSIL framework. We propose an iterative framework to enhance the performance of diffusion-based models for anomaly detection. Our approach involves resampling the dataset after each iteration, aiming to progressively refine the training set by reducing the proportion of anomalies.

As outlined in Algorithm 1, the proposed framework involves training the same model multiple times. After each training phase, anomaly scores are computed for every instance of the original training dataset using the previously trained model. These scores are then employed to select which instances to retain in the dataset for the next iteration. By systematically removing instances that are most likely to be anomalies, the ratio of anomalous data points in the training set is significantly reduced over successive iterations. This progressive "purification" of the dataset approximates a semi-supervised learning setup, which is particularly advantageous for diffusion-based models. Training on a cleaner dataset improves the model's ability to learn the distribution of normal data, leading to enhanced anomaly detection performance.

B. Data point selection

This iterative framework fosters the effective identification and exclusion of anomalous instances from the training dataset. While the ratio of training data retained between iterations can be a fixed value, it can also be decreased over successive iterations. Selecting a single fixed ratio that remains constant throughout the process introduces challenges, particularly when there is no prior approximation of the anomaly content within the dataset. A poorly chosen fixed ratio can either risk retaining too many anomalous instances or discarding valuable normal instances, both of which can negatively impact model performance.

Algorithm 1 DSIL framework

```
Require: A dataset $X$, a diffusion model $model$ , the number of iterations max_iter, the ratio of data retained at each iteration $t$ $r_t \in [0, 1]$, and the number of anomalies $n$.
Ensure: The trained diffusion model $model$ and the set of anomalies $A$.
$X_{current} \leftarrow X$
for $t \leftarrow 0$ to max_iter do
    $model \leftarrow \text{train}(model, X_{current})$
    $scores \leftarrow S^t_{model}(X)$
    $X_{current} \leftarrow \arg\max_{v \in \mathbb{R}} \frac{|X[scores > v]|}{|X|} \geq r_t$
end for
$A \leftarrow \arg\max_{v \in \mathbb{R}} |X[scores > v]| \geq n$
return $model$, $A$
```

By adopting a progressively decreasing retention ratio, we aim to achieve two key benefits: 1) the initial iterations retain a larger portion of the data, any errors made by the model early on will have a limited influence on subsequent steps and will mitigated impact of early mistakes; 2) as the model becomes increasingly accurate over the iterations, we can discard more instances with greater confidence, thereby reinforcing and accelerating the learning process.

Concretely, we define r_t, the ratio of data retained from the original training set at iteration t (retention ratio), using scheduling strategies inspired by learning rate schedulers [31]. Specifically, we propose two types of schedulers to dynamically adjust the ratio of training data retained between iterations. The **exponential scheduler** adjusts the ratio r_t at iteration t using an exponential decay function. This scheduler is defined as:

$$r_t = \frac{r_T + 1}{2}(r_0 - r_T)\left(1 + \exp\left(\frac{-t}{T}\right)\right)$$

where T is the maximum number of iterations, r_T is the final ratio to reach, and r_0 is the initial ratio. The exponential scheduler starts with a higher ratio and decreases it more rapidly in the initial iterations, slowing down the decrease as the iterations progress. This approach ensures that the model initially retains a larger portion of the data, allowing it to learn from a broader set of examples, and gradually focuses on a more refined subset as training progresses.

Similarly, the **cosine scheduler** adjusts the ratio r_t at iteration t using a cosine function. This scheduler is defined as:

$$r_t = \frac{r_T + 1}{2}(r_0 - r_T)\left(1 + \cos\left(\frac{t\pi}{T}\right)\right)$$

where the parameters T, r_T, and r_0 are defined similarly to the exponential scheduler. The cosine scheduler provides a smoother transition compared to the exponential scheduler. It starts with a higher ratio and gradually decreases it in a more uniform manner throughout the iterations. This scheduler helps in maintaining a balanced learning rate, ensuring that the model neither retains too many anomalous instances nor discards valuable normal instances too quickly. In our experiments, we set $r_0 = 0.8$ and $r_T = 0.5$. These values ensure that the initial ratio of retained data is sufficiently high to capture a wide range of normal behaviors, while the final ratio is low enough to focus on the most relevant data points, thereby enhancing the model's ability to detect anomalies effectively. By using these schedulers, we aim to optimize the iterative learning process, making it more adaptive and robust to the presence of anomalies in the training data.

C. Explainable Anomaly Detection

Building a model with better performance often comes at the cost of interpretability. To address this, we employ iterative learning to enhance model performance without increasing its complexity. Additionally, we propose a method to improve the explainability of the model's predictions.

a) Feature importance score: We introduce a simple yet effective approach that leverages the forward process of diffusion models to develop a feature importance method based on feature perturbation. To assess the importance of each feature in the model's decision-making process for a given instance, we introduce noise at different timesteps of the forward process, perturbing only one feature at a time. By analyzing how the model's decision evolves in response to these perturbations, we gain insights into the relative importance of individual features:

$$s_i(x) = AGG_{t \in T}(S^t_{model}(x, i)) \tag{5}$$

with T the set of time-steps in the forward diffusion process q of a diffusion model applied exclusively to feature i. More precisely let β_t be a vector in $\mathbb{R}^d$ where d is the dimension of x, such that $\beta_t[i] \in [0, 1]$ and $\beta_t[j] = 0, j \neq i$. We differentiate the variance scheduling so that only feature i is modified. In this case, the transition distribution is: $q(x_t|x_{t-1}) = \mathcal{N}(x_t; 1 - \beta_t \cdot x_{t-1}, \beta_t))$ where $\cdot$ denotes element-wise multiplication.

The function AGG represents an aggregation operation, which can be either the mean or the maximum. This approach provides a clearer understanding of which features play a crucial role in the model's anomaly detection process, thereby enhancing interpretability.

b) Evaluating the Quality of the Explainability Method: Assessing the quality of an explainability method is challenging. We can leverage the feature importance score to rank the features. To evaluate the effectiveness of this ranking, we formulate the assessment as a ranking problem and propose using the normalized Discounted Cumulative Gain (nDCG), a widely used metric in recommender systems. However, feature importance scoring has not previously been treated as a ranking problem, and nDCG has never been applied to evaluate the performance of feature importance methods.

Computing nDCG requires a relevance score for each ranked feature. Let d be the number of feature and $rank$ their order with respect to the feature importance scores s. Specifically, $rank = [s_{\rho_1} \cdots s_{\rho_d}]$ where ρ is a permutation of $\{1, \cdots, d\}$ such that $\forall i < j,\ s_{\rho_i} \leq s_{\rho_j}$. We define the relevance score $rel[i] = 1$ if $\rho_i \leq m$, where m is the number of features that explain the anomalies according to the ground truth. Otherwise $rel[i] = 0$. The optimal relevance score $rel^\star$ is define by $rel^\star[i] = 1$, if i is a ground truth feature. Otherwise $rel^\star[i] = 0$.

The nDCG score is computed as the Discounted Cumulative Gain (DCG) normalized by the Ideal Discounted Cumulative Gain (iDCG):

$$nDCG_k = \frac{DCG_k}{iDCG_k} \quad \text{with}$$
$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel[i]} - 1}{\log_2(i+1)} \text{ and } iDCG_k = \sum_{i=1}^{k} \frac{2^{rel^*[i]} - 1}{\log_2(i+1)}$$

We also define another metric, which is similar but does not take into account any ranking. Building a mask using top$_k$ features in the feature importance ranking, we define the accuracy of explanation as:

$$Acc_m = \frac{||rel \cdot rel^\star||}{m} \tag{6}$$

Although these two metrics may appear similar, there is a fundamental distinction between the accuracy we have defined and the application of normalized Discounted Cumulative Gain (nDCG). While accuracy quantifies the number of top-m features that are correctly identified as perturbed according to the ground truth, nDCG also considers the ranking of errors. Specifically, if a method assigns the top-1 score to a feature that is not in the ground truth, it will incur a greater penalty than if a non-relevant feature is ranked at the bottom of the top-m list. This nuanced evaluation makes nDCG a more comprehensive metric for assessing the performance of feature importance methods, as it not only checks for the presence of correct features but also evaluates the quality of their ranking.

V. Experiments

In this section, we evaluate our proposed method and address the following key research questions: **RQ1** – Can iterative learning enhance model performance in an unsupervised learning setting? **RQ2** – Can we derive meaningful feature importance scores for anomalies based on the model's decisions?

A. Experimental setup

To address the research questions, we use both real-world and synthetic datasets. Real-world datasets enable us to perform an extensive performance study against state-of-the-art methods. However, the ground-truth information on these datasets is insufficient to thoroughly assess the explanations provided by our method. To overcome this limitation, we generate synthetic datasets and a novel version of ADBench, which includes the full ground truth.

Real-world Datasets. To assess the anomaly detection performance of iterative learning in a realistic setting, we conduct experiments on 46 real-world datasets from ADBench [3] whose main characteristics are given in supplementary material The versatility of these datasets in term of application domains and dimensions offers a wide range of anomaly types and complexities, making them suitable for benchmarking our method in an unsupervised learning setting. To extract anomalies using diffusion models that compute an anomaly score, we assume

TABLE I

ANOMALY DETECTION PERFORMANCE, EVALUATED USING THE AUCROC METRIC, IS REPORTED FOR THE DTE DIFFUSION MODEL ACROSS 46 DATASETS. BOLD VALUES INDICATE THE HIGHEST PERFORMANCE PER DATASET AMONG THE METHODS CONSIDERED: DTE-UNSUPERVISED, DTE-WEIGHTED LOSS AND THE PROPOSED DSIL WITH ITS VARIANTS. WHEN STANDARD DEVIATIONS SUGGEST OVERLAPPING PERFORMANCE, MULTIPLE METHODS MAY BE CONSIDERED STATISTICALLY EQUIVALENT. FOR COMPARISON, WE ALSO INCLUDE RESULTS FROM THE DTE-SEMI SUPERVISED MODEL AND THE BEST PERFORMANCE OBTAINED FROM A SET OF 23 UNSUPERVISED ALGORITHMS. UNDERLINED SCORES HIGHLIGHT DATASETS WHERE DTE INITIALLY UNDERPERFORMED RELATIVE TO THE ADBENCH BASELINE BUT ACHIEVED TOP PERFORMANCE WHEN TRAINED WITH THE DSIL FRAMEWORK.

Dataset name	DSIL Fixed	DSIL Cosine	DSIL Exponential	DTE-unsupervised	DTE-weighted loss	Best from [14]	DTE-semi supervised
http	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.90 ± 0.10	0.99 ± 0.01	1.00 ± 0.00
skin	**0.92 ± 0.00**	0.78 ± 0.02	0.77 ± 0.01	0.76 ± 0.03	0.86 ± 0.07	0.89 ± 0.00	0.92 ± 0.00
smtp	0.92 ± 0.03	0.94 ± 0.02	**0.95 ± 0.02**	**0.95 ± 0.02**	0.85 ± 0.10	0.96 ± 0.01	0.95 ± 0.02
thyroid	0.98 ± 0.00	**0.99 ± 0.00**	**0.99 ± 0.00**	**0.99 ± 0.00**	0.92 ± 0.07	0.99 ± 0.00	0.99 ± 0.00
vertebral	0.35 ± 0.02	0.39 ± 0.06	**0.45 ± 0.03**	0.41 ± 0.05	**0.64 ± 0.22**	0.56 ± 0.07	0.68 ± 0.03
Wilt	0.69 ± 0.01	0.77 ± 0.00	0.79 ± 0.01	**0.84 ± 0.02**	0.80 ± 0.04	0.86 ± 0.00	0.85 ± 0.01
annthyroid	0.85 ± 0.00	0.90 ± 0.00	0.94 ± 0.01	**0.96 ± 0.00**	0.92 ± 0.03	0.97 ± 0.01	0.98 ± 0.00
mammography	0.80 ± 0.06	**0.80 ± 0.01**	0.80 ± 0.02	0.80 ± 0.02	0.76 ± 0.04	0.91 ± 0.00	0.87 ± 0.01
glass	0.84 ± 0.02	**0.88 ± 0.01**	0.88 ± 0.04	0.87 ± 0.02	0.81 ± 0.04	0.87 ± 0.01	0.92 ± 0.02
yeast	0.40 ± 0.00	**0.43 ± 0.01**	0.43 ± 0.02	0.41 ± 0.01	**0.52 ± 0.10**	0.52 ± 0.04	0.47 ± 0.02
Pima	**0.66 ± 0.01**	0.62 ± 0.03	0.62 ± 0.01	0.63 ± 0.01	**0.70 ± 0.09**	0.72 ± 0.02	0.70 ± 0.02
shuttle	0.99 ± 0.00	0.99 ± 0.00	**1.00 ± 0.00**	0.98 ± 0.00	0.93 ± 0.07	1.00 ± 0.00	1.00 ± 0.00
Stamps	0.76 ± 0.01	**0.77 ± 0.05**	0.69 ± 0.09	0.71 ± 0.08	0.72 ± 0.04	0.93 ± 0.01	0.92 ± 0.02
breastw	**0.99 ± 0.00**	0.93 ± 0.02	0.92 ± 0.02	0.92 ± 0.01	0.82 ± 0.16	0.99 ± 0.00	0.94 ± 0.02
WBC	**0.99 ± 0.00**	0.94 ± 0.07	0.92 ± 0.06	0.87 ± 0.03	0.72 ± 0.27	1.00 ± 0.00	0.84 ± 0.06
donors	0.71 ± 0.09	**0.84 ± 0.08**	0.81 ± 0.02	0.78 ± 0.02	0.81 ± 0.07	0.90 ± 0.02	0.98 ± 0.00
cover	0.83 ± 0.02	0.89 ± 0.01	**0.92 ± 0.02**	0.71 ± 0.04	0.81 ± 0.07	0.95 ± 0.00	0.98 ± 0.00
PageBlocks	0.87 ± 0.01	0.90 ± 0.01	**0.92 ± 0.01**	0.91 ± 0.01	0.78 ± 0.14	0.92 ± 0.00	0.90 ± 0.01
vowels	0.96 ± 0.01	0.97 ± 0.00	**0.97 ± 0.00**	0.94 ± 0.03	0.86 ± 0.10	0.95 ± 0.00	0.87 ± 0.02
wine	**0.61 ± 0.37**	0.52 ± 0.19	0.47 ± 0.17	0.57 ± 0.28	**0.71 ± 0.14**	0.98 ± 0.02	1.00 ± 0.00
pendigits	**0.88 ± 0.06**	0.73 ± 0.03	0.72 ± 0.02	0.71 ± 0.04	0.80 ± 0.08	0.95 ± 0.01	0.98 ± 0.01
Lymphography	0.94 ± 0.10	**0.98 ± 0.01**	0.90 ± 0.15	0.86 ± 0.13	0.86 ± 0.15	1.00 ± 0.00	0.99 ± 0.00
Hepatitis	**0.76 ± 0.08**	0.67 ± 0.06	0.61 ± 0.11	0.69 ± 0.07	0.73 ± 0.08	0.81 ± 0.01	0.99 ± 0.01
Cardiotocography	0.48 ± 0.01	0.49 ± 0.03	**0.53 ± 0.04**	0.47 ± 0.02	**0.68 ± 0.18**	0.78 ± 0.00	0.62 ± 0.03
Waveform	0.62 ± 0.02	**0.62 ± 0.01**	0.61 ± 0.01	0.60 ± 0.02	**0.68 ± 0.11**	0.75 ± 0.01	0.65 ± 0.01
cardio	0.72 ± 0.01	0.74 ± 0.02	0.72 ± 0.02	0.71 ± 0.01	**0.75 ± 0.03**	0.95 ± 0.00	0.88 ± 0.01
ALOI	**0.54 ± 0.00**	0.53 ± 0.00	0.53 ± 0.00	0.53 ± 0.00	**0.63 ± 0.11**	0.79 ± 0.01	0.50 ± 0.00
fault	0.63 ± 0.03	**0.63 ± 0.02**	0.61 ± 0.03	0.59 ± 0.02	**0.68 ± 0.08**	0.72 ± 0.01	0.59 ± 0.01
fraud	0.93 ± 0.02	0.93 ± 0.02	**0.94 ± 0.01**	**0.94 ± 0.01**	0.90 ± 0.03	0.96 ± 0.01	0.94 ± 0.02
WDBC	**0.95 ± 0.01**	0.92 ± 0.08	0.88 ± 0.03	0.70 ± 0.21	0.86 ± 0.17	0.99 ± 0.00	0.99 ± 0.01
letter	0.85 ± 0.00	**0.87 ± 0.01**	0.85 ± 0.01	0.81 ± 0.00	0.86 ± 0.04	0.89 ± 0.01	0.37 ± 0.02
WPBC	0.49 ± 0.03	0.49 ± 0.09	0.48 ± 0.06	0.48 ± 0.03	**0.64 ± 0.15**	0.55 ± 0.03	0.70 ± 0.05
Ionosphere	**0.95 ± 0.01**	0.92 ± 0.02	0.92 ± 0.01	0.93 ± 0.01	0.82 ± 0.12	0.95 ± 0.01	0.95 ± 0.03
satimage-2	**1.00 ± 0.00**	**1.00 ± 0.00**	0.99 ± 0.00	0.94 ± 0.02	0.93 ± 0.02	1.00 ± 0.00	0.99 ± 0.00
satellite	**0.79 ± 0.00**	0.77 ± 0.01	0.77 ± 0.00	0.73 ± 0.01	**0.81 ± 0.09**	0.80 ± 0.00	0.79 ± 0.01
landsat	**0.58 ± 0.00**	0.56 ± 0.01	0.56 ± 0.01	0.53 ± 0.02	**0.69 ± 0.15**	0.67 ± 0.01	0.52 ± 0.02
celeba	**0.86 ± 0.01**	0.85 ± 0.01	0.84 ± 0.01	0.81 ± 0.00	0.83 ± 0.01	0.80 ± 0.04	0.82 ± 0.02
SpamBase	0.50 ± 0.00	0.50 ± 0.03	0.51 ± 0.02	**0.51 ± 0.01**	**0.60 ± 0.16**	0.69 ± 0.00	0.83 ± 0.01
campaign	0.77 ± 0.00	**0.78 ± 0.01**	0.78 ± 0.02	0.78 ± 0.01	**0.81 ± 0.05**	0.78 ± 0.00	0.79 ± 0.01
optdigits	0.40 ± 0.11	0.47 ± 0.00	0.48 ± 0.12	0.56 ± 0.18	**0.69 ± 0.19**	0.87 ± 0.00	0.85 ± 0.02
mnist	**0.88 ± 0.01**	0.80 ± 0.01	0.84 ± 0.01	0.84 ± 0.01	0.87 ± 0.02	0.87 ± 0.01	0.89 ± 0.01
musk	**1.00 ± 0.00**	**1.00 ± 0.00**	**1.00 ± 0.00**	0.98 ± 0.01	0.95 ± 0.05	1.00 ± 0.00	1.00 ± 0.00
backdoor	**0.90 ± 0.00**	0.90 ± 0.01	0.90 ± 0.01	0.87 ± 0.03	0.86 ± 0.04	0.94 ± 0.01	0.91 ± 0.01
speech	0.51 ± 0.04	**0.54 ± 0.05**	0.49 ± 0.03	0.51 ± 0.07	**0.59 ± 0.08**	0.52 ± 0.04	0.38 ± 0.02
census	**0.66 ± 0.00**	0.66 ± 0.01	0.65 ± 0.00	0.66 ± 0.01	**0.77 ± 0.13**	0.73 ± 0.02	0.69 ± 0.01
InternetAds	0.64 ± 0.02	0.65 ± 0.00	0.65 ± 0.02	**0.66 ± 0.02**	**0.68 ± 0.04**	0.70 ± 0.00	0.79 ± 0.02
# Best performance	**18**	14	10	7	**19**		
Mean rank	2.60	**2.41**	2.70	3.32	2.72		

that the total number of anomalies is known. Based on this assumption, we select the top n instances with the highest anomaly scores S^t_{model} assigned by the model.

Synthetic dataset generation. While real-world datasets are valuable for evaluating detection performance, they often lack ground-truth explanations for anomalies, making it challenging to assess the quality of explanations. To address this limitation, we generate synthetic datasets with varying numbers of dimensions, introducing controlled anomalies with known ground truth. Specifically, we construct synthetic datasets based on arbitrarily generated probabilistic graphical models. We introduce anomalies into each initial model by altering the parameters of the initial variable defining the model of a dataset. For each dataset, each feature corresponds to a variable in the corresponding graphical model. Each variable can follow one of the given distributions: Uniform, which is defined by its minimum and maximum values; Normal, characterized by its mean and standard deviation; Exponential, determined by its rate parameter; or Gamma, specified by its shape and scale parameters. This approach ensures that we have precise ground truth for the perturbed features corresponding to each anomaly, enabling a thorough evaluation of our method's explainability. We define five types of anomalies:

- **Cluster**: all the parameters are multiplied by a value α, defining each initial distribution by $x_i \sim \mathcal{X}_i(\alpha\Theta)$, with $\mathcal{X}_i$ the distribution governing the ith feature and defined by its Θ parameter. For a given point, all its features are affected by this anomaly.
- **Global**: Features that are affected by this anomaly are perturbed by redrawing the value of the feature following a uniform distribution with the minimum and maximum of the dataset both multiplied by α as parameters of the distribution: $x_i \sim \text{Unif}(\alpha \min(X_i), \alpha \max(X_i))$
- **Local**: Features that are affected by a local anomaly are perturbed by drawing a new value using the covariance of the variables scaled by a parameter α:

$x_M = \mathcal{N}(\text{mean}(X_M), \alpha \times \text{cov}(X_M))$

- **Additive and Multiplicative Noise**: Features that are affected by this anomaly are perturbed by either adding to or multiplying by a random value: $x_i = x_i + \epsilon$ or $x_i = x_i \times \epsilon$, where $\epsilon \sim \mathcal{N}(\mu, \sigma)$.

TABLE II
CHARACTERISTICS OF THE SYNTHETIC DATASET.

d	# Causality	Anomalies type
4	2	Global, Cluster
10	4	Global, Cluster, Local
50	12	All
100	30	All
1000	30	All

Creating synthetic anomalies allows us to identify the specific features responsible for an instance's abnormality, providing a well-defined ground truth for explanations. Consequently, we expect the explanation method to assign higher feature importance scores to the perturbed features that contributed to the anomaly. With this ground truth available, we can compute metrics to systematically evaluate the effectiveness of our method. Table II presents the size of each dataset used in our experiments, the number and the types of anomalies introduced.

Each dataset consists of 5,000 samples and is available in two versions, with anomaly ratios of either 5% or 10%. For each dataset configuration, we generate five distinct datasets using different random seeds. Since the ground truth in ADBench is incomplete (providing labels for anomalous instances but not the specific features responsible) we introduce a novel version of ADBench that includes ground truth for both anomalous instances and the features affected by the anomalies by starting from the normal samples of an ADBench dataset and injecting the synthetic anomalies described earlier.

Models. We evaluate several variants of the DSIL framework, which is controlled by two main parameters: the diffusion model used and the data retention ratio. Regarding the data retention strategy, we tested three configurations: a fixed ratio of 50% (Fixed), as well as two scheduling strategies: Cosine and Exponential, described in Section IV-B (Study of the fixed ratio parameter in supplementary material). We compare these DSIL variants with DTE, a fully unsupervised baseline where the DTE model is trained on the entire dataset, and $\text{DTE}_{\text{weighted loss}}$, where the DTE model is trained using the loss reweighting strategy proposed by [4] which incorporates instance-specific weights in the loss function that are updated during model retraining.

Additionally, we benchmark our method against results from [14] which includes 23 unsupervised algorithms. We also report the performance of a semi-supervised DTE variant, DTE-semi supervised, where the model is trained exclusively on normal samples, and anomalies are detected as instances that deviate from the learned normal representations.

Hyperparameters. To determine the optimal hyperparameters for DTE, we conducted a grid search focusing on the number of diffusion timesteps, the number of bins, and the size of the hidden layers in the model. We selected the hyperparameter set that achieved the highest mean AUC-ROC across all datasets under study. Consequently, for both unsupervised learning and our iterative learning methods, the DTE model consistently uses the following hyperparameters, regardless of the dataset: 7 bins, 400 diffusion timesteps, and hidden layers of sizes [256, 512, 256]. We chose to maintain these hyperparameters uniformly across datasets after a preliminary grid search, given the unsupervised nature of our setup. Models are trained using a learning rate of 10^{-4} with Adam optimizer over 400 epochs. An analysis of the impact of hyperparameters of DSIL is provided in supplementary material. This section compares different retention ratio values and examines how performance evolves over iterations, both with a fixed retention ratio and when using a scheduling strategy.

Code and reproductibility. Our code and supplementary results are available in a public repository[1]. For the implementation of DTE [14] and ADBench [3], we rely on the authors' code[2].

B. Performance study

We evaluate the performance of iterative learning on real-world datasets from ADBench. Since DTE outputs anomaly scores, we evaluate the results under the hypothesis that the desired proportion of anomalies to extract from the dataset is known. Accordingly, we treat instances with the highest anomaly scores as anomalies and compute the AUC-ROC and F1 score for assessment.

Figure 1 presents the overall performance of 23 anomaly detection methods across different algorithmic families in an unsupervised setting, using datasets from ADBench. The results highlighted in blue correspond to the DTE model trained with the DSIL framework and a weighted loss, both of which involve iterative learning. The figure demonstrates a consistent performance improvement when incorporating iterative learning, with the DSIL framework also exhibiting reduced variability. Among the retention strategies, a fixed retention ratio yields slightly higher average ROCAUC compared to the schedulers, albeit with increased variance across datasets. Interestingly, while the fixed retention approach achieves the highest mean performance, the cosine scheduler leads in mean rank, suggesting greater robustness across datasets.

Table I summarizes the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) scores for the ADBench datasets.

Analysis of the raw scores provide several observations. DSIL Fixed achieves the best performance on 18 datasets, demonstrating its robustness across diverse data types, with particularly strong results on 'http', 'skin', and 'breastw'. DSIL Cosine performs best on 14 datasets, delivering strong but slightly less consistent results compared to the fixed variant; representative cases include 'thyroid' and 'glass'. DSIL

[1] https://github.com/ElouanV/iterative_learning_for_anomaly_detection

[2] https://github.com/vicliv/DTE and https://github.com/Minqi824/ADBench

TABLE III

$nDCG_m$, Acc_m, AND TOTAL COMPUTATION TIME REQUIRED TO EXPLAIN ALL ANOMALIES DETECTED USING DSIL WITH DTE AND FIXED STRATEGY ON SYNTHETIC DATASETS.

Dataset		SHAP			Gradient			DSIL (Mean diffusion)			DSIL (Max diffusion)		
d	% ano.	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time
4	5 %	0.99 ± 0.01	0.99 ± 0.01	2.29 ± 0.18	0.59 ± 0.04	0.99 ± 0.01	0.09 ± 0.12	0.99 ± 0.01	0.99 ± 0.01	**0.63 ± 0.02**	0.99 ± 0.01	0.99 ± 0.01	1.67 ± 0.06
4	10 %	0.94 ± 0.01	0.94 ± 0.01	4.11 ± 0.20	0.50 ± 0.00	0.95 ± 0.02	0.01 ± 0.00	0.94 ± 0.01	0.94 ± 0.01	**3.71 ± 0.15**	0.94 ± 0.013	0.94 ± 0.013	2.18 ± 0.03
10	4 %	0.84 ± 0.01	0.81 ± 0.01	10.45 ± 0.91	0.44 ± 0.03	0.77 ± 0.02	0.01 ± 0.01	**0.86 ± 0.01**	**0.82 ± 0.01**	7.57 ± 0.35	0.84 ± 0.01	0.81 ± 0.01	4.02 ± 0.05
10	9 %	**0.89 ± 0.01**	0.85 ± 0.01	34.15 ± 1.01	0.56 ± 0.01	0.79 ± 0.01	0.01 ± 0.00	**0.89 ± 0.01**	**0.86 ± 0.01**	11.21 ± 0.30	0.85 ± 0.01	0.82 ± 0.001	6.25 ± 0.09
50	5 %	0.71 ± 0.01	0.66 ± 0.02	14.61 ± 0.94	0.36 ± 0.02	0.57 ± 0.03	0.15 ± 0.22	**0.73 ± 0.02**	**0.67 ± 0.02**	12.32 ± 0.20	0.67 ± 0.02	0.63 ± 0.02	19.946 ± 0.45
50	10 %	0.64 ± 0.05	0.59 ± 0.04	72.33 ± 9.42	0.61 ± 0.01	0.63 ± 0.01	0.01 ± 0.00	**0.67 ± 0.01**	**0.61 ± 0.01**	60.99 ± 4.80	0.62 ± 0.01	0.57 ± 0.01	39.43 ± 0.21
100	5 %	0.67 ± 0.00	0.59 ± 0.00	29.58 ± 1.21	0.47 ± 0.02	0.54 ± 0.03	0.01 ± 0.00	**0.69 ± 0.01**	**0.62 ± 0.01**	27.77 ± 2.50	0.63 ± 0.01	0.57 ± 0.01	43.85 ± 4.50
100	10 %	0.59 ± 0.04	0.52 ± 0.03	55.38 ± 1.53	0.49 ± 0.01	0.56 ± 0.01	0.01 ± 0.00	**0.62 ± 0.01**	**0.56 ± 0.01**	47.38 ± 2.73	0.57 ± 0.01	0.52 ± 0.01	73.40 ± 3.01
1000	5 %	**0.51 ± 0.00**	**0.48 ± 0.01**	358.91 ± 10.22	0.35 ± 0.02	0.45 ± 0.03	0.07 ± 0.14	0.48 ± 0.01	0.47 ± 0.01	350.66 ± 5.53	0.47 ± 0.01	0.46 ± 0.01	624.95 ± 64.44
1000	10 %	0.51 ± 0.00	0.48 ± 0.00	7948.22 ± 5082.92	0.42 ± 0.01	0.48 ± 0.00	0.01 ± 0.01	**0.54 ± 0.00**	**0.52 ± 0.00**	227.26 ± 69.77	**0.55 ± 0.00**	**0.52 ± 0.00**	1082.04 ± 29.49

TABLE IV

$nDCG_m$ AND Acc_m ON 5 DATASETS FROM ADBENCH IN WHICH WE ADDED SYNTHETIC ANOMALIES WITH GROUND TRUTH.

	SHAP			Gradient			DSIL (Mean diffusion)			DSIL (Max diffusion)		
Dataset	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time	$nDCG_m$	Acc_m	Time
PageBlocks	0.52 ± 0.02	0.49 ± 0.02	12.27 ± 6.91	0.6 ± 0.01	0.62 ± 0.01	0.1 ± 0.01	**0.65 ± 0.01**	0.63 ± 0.01	4.98 ± 2.39	0.62 ± 0.03	**0.65 ± 0.02**	5.52 ± 2.55
Wilt	0.88 ± 0.03	0.88 ± 0.03	4.29 ± 1.39	0.93 ± 0.01	0.92 ± 0.01	0.09 ± 0.01	**0.93 ± 0.00**	**0.93 ± 0.00**	3.01 ± 0.82	**0.93 ± 0.00**	**0.93 ± 0.00**	2.69 ± 0.64
Campaign	0.47 ± 0.01	0.40 ± 0.00	1262.73 ± 1329.13	0.43 ± 0.01	**0.50 ± 0.01**	0.25 ± 0.03	**0.52 ± 0.02**	0.44 ± 0.02	664.20 ± 500.28	0.38 ± 0.01	0.46 ± 0.02	707.28 ± 472.96
Landsat	0.56 ± 0.00	0.48 ± 0.00	234.44 ± 118.73	0.21 ± 0.04	0.61 ± 0.03	0.51 ± 0.03	**0.69 ± 0.00**	0.61 ± 0.00	83.58 ± 43.86	0.57 ± 0.01	**0.64 ± 0.00**	84.61 ± 44.86
Vertebral	0.86 ± 0.01	0.84 ± 0.01	1.48 ± 0.46	**0.95 ± 0.01**	**0.96 ± 0.01**	0.01 ± 0.01	0.83 ± 0.06	0.79 ± 0.08	2.85 ± 1.08	0.78 ± 0.06	0.83 ± 0.04	3.16 ± 1.10

Exponential leads on 10 datasets, indicating effectiveness, though with reduced consistency. It performs notably well on 'smtp' and 'vertebral'. The DTE-unsupervised method outperforms the others on 7 datasets, suggesting that DSIL does not universally enhance performance, particularly on datasets such as 'Wilt' and 'annthyroid'. Meanwhile, the DTE model with weighted loss achieves the highest AUC-ROC on 19 datasets but exhibits instability, with variances reaching up to 0.27 on 'WDBC'. Among the 23 unsupervised methods benchmarked in [14], top performance is recorded on 44 datasets. However, achieving this level of performance required combining many different algorithms, reflecting a substantial variability across methods. Notably, on 8 datasets, underlined in the "Best from [14]" column, the application of the DSIL framework allows the DTE model to outperform the best-performing methods reported in the benchmark. These cases, where the standalone DTE model was initially outperformed by existing unsupervised approaches, underscore the effectiveness of iterative learning in improving anomaly detection performance.

Mean rank analysis provides an overview of the relative performance of each method. DSIL Cosine achieves the lowest mean rank of 2.41, indicating the best average performance across datasets. It is followed by DSIL Fixed with a mean rank of 2.60, while DSIL Exponential ranks slightly lower at 2.72. Interestingly, the Weighted Loss method matches the mean rank of Exponential (2.72), despite achieving the highest number of individual best performances. In contrast, the DTE model trained in an unsupervised fashion has the highest mean rank of 3.32, reflecting weaker average performance across the benchmark.

Notably, DSIL significantly enhances DTE on datasets like 'pendigits', where the AUC-ROC improves from 0.71 (unsupervised) to 0.88 (iterative learning). Overall, DSIL, especially the Fixed and Cosine variants, delivers robust results across datasets, frequently outperforming unsupervised methods and nearing benchmark scores. However, the Exponential scheduler has limited utility, with exceptions like the 'cover' dataset. F1 score results are provided in supplementary material

C. Explainability

On synthetic anomalies. The nDCG and Accuracy metrics used to evaluate feature importance methods require ground truth information about perturbed features. Therefore, we employed synthetic anomalies to ensure access to this ground truth. We define a binary vector where 0 means that the feature is not involved in the perturbation apply to the data at the generation, and 1 for a feature which was involved and is therefore expected to get an important feature score in the decision process. We use this vector as the relevance for the nDCG score and the ground truth of the explanation accuracy. For both methods we set m to be the size of the ground truth explanation. We compare our method against SHAP as a primary competitor and use the gradient of the model's output as a lower bound for execution time. For our method, we employ both mean and max aggregation functions (see Eq. 5) to evaluate its performance comprehensively. To make a fair comparison between SHAP and our methods, hyperparameters were chosen so that execution time is similar between methods. However, for some datasets, reducing the SHAP execution time too much resulted in a number of coalitions considered too small, and caused the linear regression to not converge according to the criteria given by the author of the library. Thus, in some cases we chose to let SHAP more computation time by increasing the number of coalition it can use for its approximation of the feature importance score.

We reported in Table III the nDCG and accuracy obtained under these conditions. For our methods, we set $|T| = 200$, except for the dataset with 1000 features, where we reduced it to $|T| = 40$ to limit computation time. For SHAP, we set the number of coalitions to consider as $n_{\text{samples}} = \alpha \times d$, where d is the data dimensionality and $\alpha \in [1.5, 2]$. Our method, using $AGG = \text{mean}()$, consistently outperforms SHAP across

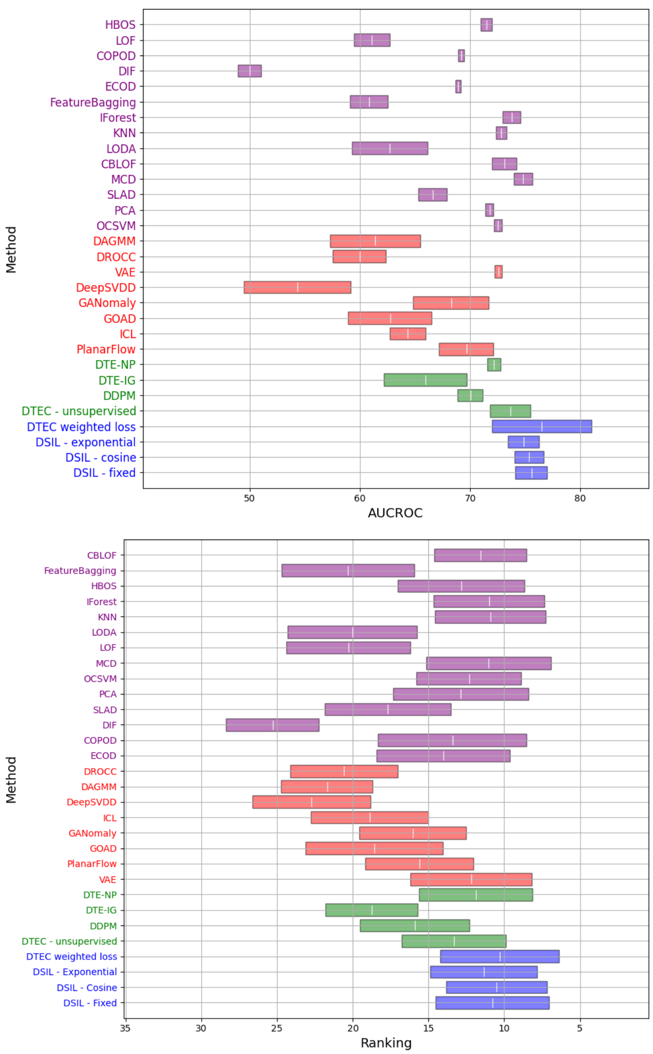


Fig. 1. Mean AUROC (top) and average rank (bottom) across ADBench datasets, comparing different anomaly detection approaches: classical methods (purple), deep learning (red), diffusion-based (green), and our proposed model (blue). The blue bars represent our DTE model trained using the iterative learning framework (DSIL) and DTE-weighted loss implementation.

all scenarios in terms of both nDCG and Accuracy, even when SHAP is allocated more computation time for feature scoring. While SHAP's execution times remain substantially higher than those of the gradient-based method, the latter yields suboptimal results, which highlights the need for more sophisticated approaches that strike a better balance between accuracy and efficiency.

Experiments on real-world datasets with injected synthetic anomalies (Table IV), which provide ground-truth on perturbed features, support these findings. A notable exception is the 'Vertebral' dataset, where SHAP performs competitively due to the dataset's small size, which allows for exhaustive computations. In all other cases, DSIL achieves superior performance with significantly lower computation times. Moreover, using $AGG = \text{mean}()$ generally leads to higher nDCG scores, while $AGG = \max()$ tends to produce better Accuracy.

On real world data. Next, we evaluate our method on real-world datasets containing naturally occurring anomalies. Since ground-truth perturbed features are not available in this setting, we compare different methods using infidelity as defined in [32] and faithfulness presented in [33], both of which can be computed without requiring ground-truth annotations. A faithfulness score close to 1 indicates that the model's explanations are highly aligned with its actual behavior. Figure 2 (top) shows the Faithfulness@k across different values of k on ADBench datasets with synthetic anomalies, as used in Table IV. In terms of faithfulness on synthetic data, SHAP demonstrates marginally superior performance overall, with a particularly notable advantage in the 'landsat' dataset where it consistently outperforms across all values of k. Figure 2 (bottom) illustrates the Faithfulness@k for various values of k across four real-world datasets of different sizes. When compared to SHAP, our method excels in achieving better faithfulness at lower values of k, whereas SHAP performs better at higher values of k. This indicates that DSIL more effectively identifies the top-k critical features, though it assigns less discriminative scores to lower-importance features compared to SHAP. In real-world applications, anomalies typically manifest in only a subset of features rather than across all features simultaneously. Therefore, explanation methods that achieve high faithfulness with a small number of selected features, low k, provide more practical value than those requiring a large k approaching the total feature count in the dataset.

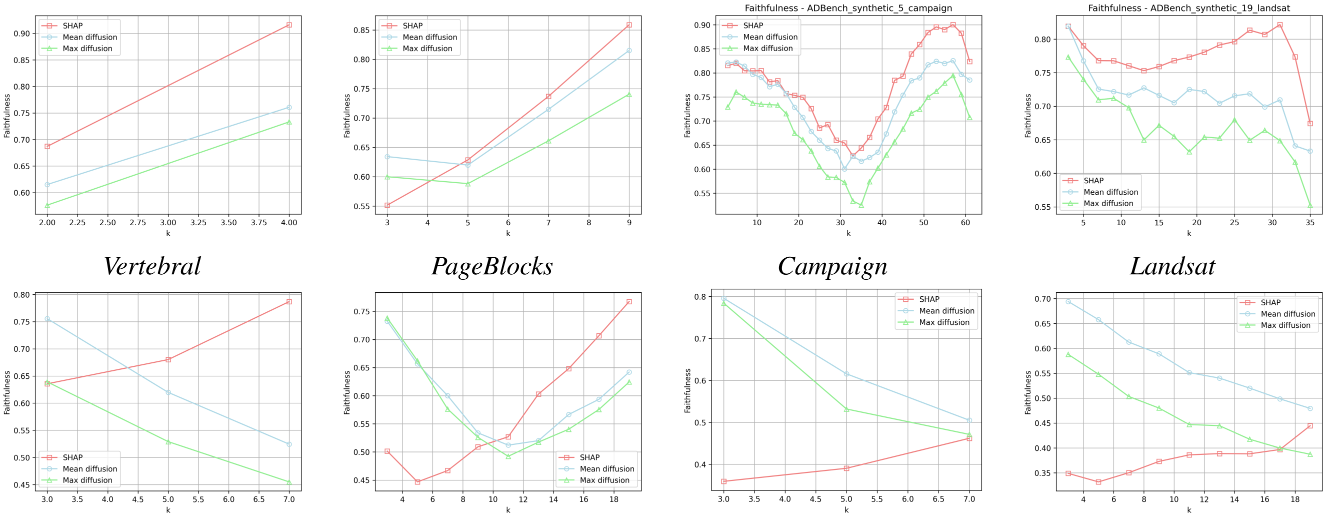
TABLE V
INFIDELITY SCORES ON FOUR REAL-WORLD DATASETS FOR SHAP AND DSIL USING MEAN AND MAX DIFFUSION (LOWER IS BETTER).

Dataset	SHAP	DSIL (Mean diffusion)	DSIL (Max diffusion)
Ionosphere	1.42×10^{-4}	1.37×10^{-4}	1.37×10^{-4}
WBC	2×10^{-6}	3×10^{-6}	3×10^{-6}
breastw	1.1×10^{-5}	1.1×10^{-5}	1.1×10^{-5}
cardio	6×10^{-6}	5×10^{-6}	5×10^{-6}

Table V summarizes the mean infidelity metrics across all four benchmark datasets. No statistically significant difference in infidelity is observed between SHAP and DSIL. Importantly, our method maintains consistent infidelity values regardless of the chosen aggregation function, demonstrating the robustness of our approach to this implementation choice.

VI. Conclusion

In this work, we introduced DSIL, an novel framework that improves unsupervised anomaly detection that leverage generative diffusion models. Concretely, through iterative refinement of unlabeled datasets, DSIL effectively narrows the gap between unsupervised and semi-supervised learning and achieve superior performance on real-world datasets. In addition, our experimental results demonstrate DSIL's capabilities for interpretability. We illustrate it with a novel application of metrics commonly used in recommender systems. Specifically, we use *nDCG* for evaluating synthetic anomaly detection, in addition to employing established metrics such



Vertebral *PageBlocks* *Campaign* *Landsat*

Breastw *Cardio* *WBC* *Ionoshpere*

Fig. 2. Mean Faithfulness@k for different value of k on ADBench datasets with synthetic anomalies (top) and for four different datasets of various size (bottom).

as *faithfulness* and *infidelity* for assessing performance on real-world anomalies. Future work will extend DSIL to dynamic datasets and integrate domain knowledge for improved anomaly detection in specialized contexts.

References

[1] D. M. Hawkins, *Identification of Outliers*. Monographs on Applied Probability and Statistics, Springer, 1980.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[3] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," *NEURIPS*, vol. 35, pp. 32142–32159, 2022.

[4] M. Kim, J. Yu, J. Kim, T. Oh, and J. K. Choi, "An iterative method for unsupervised robust anomaly detection under data contamination," *IEEE Trans. Neur. Net. Learn. Syst.*, vol. 35, no. 10, pp. 13327–13339, 2024.

[5] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[6] J. Liu, Z. Ma, Z. Wang, Y. Liu, Z. Wang, P. Sun, L. Song, B. Hu, A. Boukerche, and V. Leung, "A survey on diffusion models for anomaly detection," *arXiv preprint arXiv:2501.11430*, 2025.

[7] Q. Dao, B. Ta, T. Pham, and A. Tran, "A high-quality robust diffusion framework for corrupted dataset," in *ECCV*, pp. 107–123, 2025.

[8] Y. Chen, X. Li, P. Hu, D. Peng, and X. Wang, "Diffilter: Defending against adversarial perturbations with diffusion filter," *Trans. Info. For. Sec.*, vol. 19, p. 6779–6794, Jan. 2024.

[9] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, "Diffusion models for medical anomaly detection," in *MICCAI*, pp. 35–45, Springer, 2022.

[10] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *CVPR*, pp. 650–656, 2022.

[11] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "Tabddpm: Modelling tabular data with diffusion models," in *ICML*, pp. 17564–17579, PMLR, 2023.

[12] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard, "Digress: Discrete denoising diffusion for graph generation," *arXiv preprint arXiv:2209.14734*, 2022.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NEURIPS*, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[14] V. Livernoche, V. Jain, Y. Hezaveh, and S. Ravanbakhsh, "On diffusion modeling for anomaly detection," in *ICLR*, OpenReview.net, 2024.

[15] J. Yoon, K. Sohn, C.-L. Li, S. O. Arik, C.-Y. Lee, and T. Pfister, "Self-supervise, refine, repeat: Improving unsupervised anomaly detection," *ICLR*, 2022.

[16] J. Liu, M. He, X. Shang, J. Shi, B. Cui, and H. Yin, "Bourne: Bootstrapped self-supervised learning framework for unified graph anomaly detection," *ICDE*, pp. 2820–2833, 2023.

[17] C. Lacoquelle, X. Pucel, L. Trave-Massuyes, A. Reymonet, and B. Enaux, "Warped time series anomaly detection," *ArXiv*, vol. 2404.12134, 2024.

[18] A. Brown, A. Tuor, B. Hutchinson, and N. Nichols, "Recurrent neural network attention mechanisms for interpretable system log anomaly detection," in *workshop on ML4CS*, 2018.

[19] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *ECCV*, pp. 485–503, Springer, 2020.

[20] G. Pang, C. Ding, C. Shen, and A. v. d. Hengel, "Explainable deep few-shot anomaly detection with deviation networks," *arXiv preprint arXiv:2108.00462*, 2021.

[21] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert systems with applications*, vol. 186, p. 115736, 2021.

[22] Z. Li, Y. Zhu, and M. Van Leeuwen, "A survey on explainable anomaly detection," *ACM TKDD*, vol. 18, no. 1, pp. 1–54, 2023.

[23] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

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

[25] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *NEURIPS*, vol. 30, 2017.

[26] X. Huang and J. Marques-Silva, "On the failings of shapley values for explainability," *Approx. Reason.*, vol. 171, p. 109112, 2024.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?," in *SIGKDD*, pp. 1135–1144, 2016.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[29] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *ICML*, pp. 3319–3328, PMLR, 2017.

[30] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *Preprint arXiv:1706.03825*, 2017.

[31] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[32] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. Inouye, and P. Ravikumar, "On the (in)fidelity and sensitivity of explanations," *NEURIPS*, vol. 32, 2019.

[33] U. Bhatt, A. Weller, and J. M. F. Moura, "Evaluating and aggregating feature-based model explanations," in *IJCAI*, 2021.