AR-Pro: Counterfactual Explanations for Anomaly Repair with Formal Properties

Xiayan Ji* **Anton Xue*** **Eric Wong** **Oleg Sokolsky** **Insup Lee**
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
{xjiae,antonxue,exwong,sokolsky,lee}@seas.upenn.edu

Abstract

Anomaly detection is widely used for identifying critical errors and suspicious behaviors, but current methods lack interpretability. We leverage common properties of existing methods and recent advances in generative models to introduce counterfactual explanations for anomaly detection. Given an input, we generate its counterfactual as a diffusion-based repair that shows what a non-anomalous version *should have looked like*. A key advantage of this approach is that it enables a domain-independent formal specification of explainability desiderata, offering a unified framework for generating and evaluating explanations. We demonstrate the effectiveness of our anomaly explainability framework, AR-Pro, on vision (MVTec, VisA) and time-series (SWaT, WADI, HAI) anomaly datasets. The code used for the experiments is accessible at: `https://github.com/xjiae/arpro`.

1 Introduction

Anomaly detectors measure how much their inputs deviate from an established norm, where too much deviation implies an instance that warrants closer inspection [15, 52]. For example, unusual network traffic may indicate potential malicious attacks that necessitate a review of security logs [16, 30], irregular sensor readings in civil engineering may suggest structural weaknesses [40, 49, 53], and atypical financial transactions point to potential fraud [3, 47]. As a benign occurrence, unexpected anomalies in scientific data can also lead to new insights and discoveries [25, 45].

Although state-of-the-art anomaly detectors are good at catching anomalies, they often rely on black-box models. This opacity undermines reliability: inexperienced users might over-rely on the model without understanding its rationale, while experts may not trust model predictions that are not backed by well-founded explanations. This limitation of common machine learning techniques has led to growing interest in model explainability [12], particularly in domains such as medicine [50] and law [14]. We refer to [36, 44] and the references therein for recent surveys on explainability.

While many anomaly detection methods can localize which parts of the input are anomalous, this may not be a satisfactory explanation, especially when the data is complex. In medicine [20], heart sound recordings [17] and EEG brain data [5] can differ greatly between individuals; in industrial manufacturing, it can be hard to understand subtle defects of PCB for inexperienced workers [73]. Thus, even when the location of the anomaly is known, it may be hard to articulate *why* it is anomalous. In such cases, it can be helpful to ask the **counterfactual** question: "What *should* a non-anomalous version look like?" For example, a doctor might ask what changes in a patient's chest X-ray might improve diagnostic outcomes [58], while a quality assurance engineer may ask what changes would fix the defect [6].

[*] Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

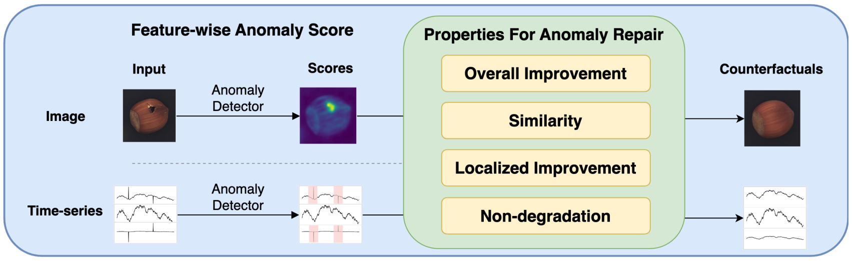


Figure 1: Overview of the AR-Pro framework. We first identify an input's anomalous region and then use property-guided diffusion to repair it. This repair is the counterfactual anomaly explanation, where the following properties are defined with respect to a *linearly decomposable* anomaly detector (AD). (*Overall Improvement*) The repair has a lower anomaly score. (*Similarity*) The repair should resemble the original. (*Localized Improvement*) The score over the repaired region should improve. (*Non-degradation*) The score over the non-anomalous region should not significantly worsen.

To answer such counterfactual questions, we begin with two observations. First, anomalies are commonly localized to small regions of the input [13, 38, 73]. Second, recent generative models, e.g., diffusion [24] models, can be trained to produce high-quality, non-anomalous examples. These observations motivate us to *repair anomalies* as a *counterfactual explanation*. However, simply generating a repair is not sufficient; we must also ensure its quality. For example, it may be undesirable for a repair to significantly improve the anomaly score but barely resemble the original input. Therefore, it is important to measure the quality of repairs using *formal properties*.

However, formalizing broadly applicable properties is challenging, as different domains (e.g., vision, time series) and tasks (e.g., manufacturing, security) have unique considerations. To overcome this, we further observe that many anomaly detectors in practice [4] satisfy a condition that we call *linear decomposability*: the overall anomaly score is an aggregation of feature-wise anomaly scores. Importantly, this is a strong but common condition with which we may formalize the desiderata of counterfactual explanations. Conveniently, many of these desiderata are, in fact, domain-independent. We leverage these conditions to formalize a general, domain-independent framework for counterfactual anomaly explanation: we use a generative model to produce a repair and then evaluate this repair with respect to the detector model.

We present an overview of our anomaly explainability framework, AR-Pro, in Figure 1. While anomaly repair [21, 31, 70] and counterfactual explanations [56] have been explored in the literature, we are the first to study this in a unified context. We summarize our contributions as follows:

- We observe that many anomaly detectors satisfy *linear decomposability* and use this condition to define general, domain-independent properties for counterfactual explanations. This approach lets us measure explanation quality with respect to the anomaly detector, as shown in Figure 1.
- We use these properties to guide diffusion models towards a high-quality repair of the anomalous input. Such a repair serves as the counterfactual explanation of the anomaly.
- Our framework, AR-Pro, can produce semantically meaningful repairs that outperform off-the-shelf diffusion models with respect to our explainability criteria. Our vision anomaly benchmarks include MVTec [13] and VisA [73]. Our time-series anomaly benchmarks include SWaT [38], HAI [54], and WADI [2].

2 Overview and Formal Properties

In Section 2.1, we first observe that many anomaly detector paradigms are *linearly decomposable*. Intuitively, this condition means that the overall anomaly score is an aggregation of feature-wise anomaly scores. Next, we use linear decomposability in Section 2.2 to formalize general, domain-independent properties of counterfactual explanations. We will use the terms *counterfactual*, *explanation*, and *repair* interchangeably throughout this paper.

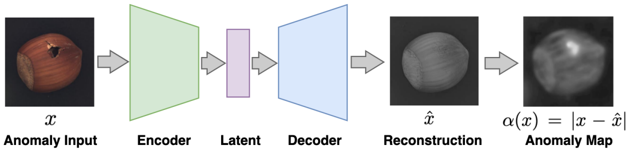


Figure 2: Reconstruction-based anomaly detection exemplifies linear decomposition. The anomalous input $x \in \mathbb{R}^n$ is first reconstructed into $\hat{x} \in \mathbb{R}^n$, and the feature-wise anomaly scores are given by $\alpha_i(x) = |\hat{x}_i - x_i|^2 \in \mathbb{R}^n$ for $i = 1, \dots, n$. Then, the standard ℓ^2 reconstruction-based anomaly score is a linear combination of the feature scores: $s(x) = \alpha_1(x) + \cdots + \alpha_n(x)$.

2.1 Common Anomaly Detectors are Linearly Decomposable

Many anomaly detection techniques use a scoring function $s : \mathbb{R}^n \to \mathbb{R}$ to measure the anomalousness an input $x \in \mathbb{R}^n$, where $s(x)$ is called the *anomaly score* of x. This is commonly done in a two-stage process: the *feature-wise anomaly scores* $\alpha_1(x), \dots, \alpha_n(x) \in \mathbb{R}$ are first computed and then aggregated [15]. We observe that this aggregation often satisfies the following form:

Definition 2.1 (Linear Decomposition)**.** The feature-wise anomaly scores $\alpha : \mathbb{R}^n \to \mathbb{R}^n$ and regularizer $\beta : \mathbb{R}^n \to \mathbb{R}$ *linearly decomposes* the anomaly score $s : \mathbb{R}^n \to \mathbb{R}$ if for all $x \in \mathbb{R}^n$:

$$s(x) = \alpha_1(x) + \cdots + \alpha_n(x) + \beta(x).$$

While linear decomposability appears to be a strong assumption, it is common in practice. We show a number of examples below, where we also flexibly refer to $\alpha : \mathbb{R}^n \to \mathbb{R}^n$ as the *feature scores*.

Example 2.2. In reconstruction-based anomaly detection [8, 29], the input x is passed through an encoder-decoder architecture to generate a *reconstruction* $\hat{x}$. The motivation is that it should be harder to reconstruct out-of-distribution (anomalous) inputs. Empirically, it is observed that if x is anomalous, then $\hat{x}$ will have a large reconstruction error when feature i is in the anomalous region, e.g., a pixel in the defect area of a manufacturing artifact image. A typical example of such an anomaly score is:

$$s(x) = |\hat{x}_1 - x_1|^2 + \cdots + |\hat{x}_n - x_n|^2,$$

which is also known as the ℓ^2 reconstruction error. This lets us define the feature-wise anomaly scores by $\alpha_i(x) = |\hat{x}_i - x_i|^2$, and we illustrate this example in Figure 2.

Example 2.3. In maximum likelihood-based anomaly detection [15], one measures the likelihood of a test input x with respect to a set of non-anomalous training examples. Intuitively, an out-of-distribution (anomalous) x should be unlikely with respect to the non-anomalous training examples and will thus have a lower likelihood. In variants such as normalizing flow-based anomaly detection for images [66], it is common to define a joint probability distribution over all the features:

$$s(x) = -\big[\log p_1(x) + \cdots + \log p_n(x) + \log|\det J(x)|\big],$$

where $p_1(x), \dots, p_n(x)$ are the probabilities of each feature that lets us define $\alpha_i(x) = -\log p_i(x)$, while the change-of-variable Jacobian $\log|\det J(x)|$ may be viewed as a regularization term.

Example 2.4. In language modeling [61], it is common to measure the anomaly of a token sequence based on the likelihood of each token [7]. Although similar to the vision case, the standard formulation for language models is different: given a token sequence $x_1, \dots, x_n \in \{1, \dots, \texttt{vocab\_size}\}$, its measure of unlikeliness (anomalousness) may be defined as:

$$s(x) = -\frac{1}{n}\big[\log p(x_1) + \log p(x_2|x_1) + \log p(x_3|x_1, x_2) + \cdots + \log p(x_n|x_1, \dots, x_{n-1})\big],$$

where p is a probabilistic generative model, and so $\alpha_i(x) = -(1/n)\log p(x_i|x_1, \dots, x_{i-1})$. This is also known as a *preplexity measure*, which has been used to detect jailbreaks against LLMs [64].

Beyond the above examples, anomaly detectors for time-series data [29, 60, 63] and text [35] also commonly use this convention. We further remark that the feature-wise anomaly scores α are related to *feature attribution scores* in explainability literature [33, 51, 55].

2.2 Formal Properties for Counterfactual Explanations

We now present the formal properties of counterfactual explanations. We will assume a given anomaly score function s that is linearly decomposed by the feature-wise score function α and regularizer β. Given some input x, it is common to convert $\alpha(x) \in \mathbb{R}^n$ into a binary-valued vector $\omega(x) \in \{0,1\}^n$ to classify which input feature is anomalous, and this is commonly done by a threshold:

$$\omega(x) = (\alpha_1(x) \geq \tau_1, \ldots, \alpha_n(x) \geq \tau_n), \quad \text{for some feature-wise threshold } \tau \in \mathbb{R}^n.$$

A binarization of the feature-wise scores suggests the need for region-specific anomaly scores, which we implement with the following indexing scheme on $s(x)$:

$$s_z(x) = \beta(x) + \sum_{i: z_i = 1} \alpha_i(x), \quad \text{for all } z \in \{0,1\}^n.$$

Then, the anomalous and non-anomalous regions have scores $s_{\omega(x)}(x)$ and $s_{\overline{\omega}(x)}(x)$, respectively, where $\overline{\omega}(x) = 1 - \omega(x)$ denotes non-anomalous region. We next enumerate some common desiderata of counterfactual explanations, where we will refer to the anomalous input as x_{bad} and the repaired version as x_{fix}.

Property 1 (Overall Improvement): The anomaly score should improve. Because a "repaired" version should fix the anomaly by definition, one would reasonably expect that:

$$s(x_{\mathsf{fix}}) < s(x_{\mathsf{bad}}). \tag{P1}$$

Property 2 (Similarity): The repair should resemble the original. When s is generated by a complex machine-learning model, it may be the case that it has a value x_{fix} where $s(x_{\mathsf{fix}}) \ll s(x_{\mathsf{bad}})$, but x_{fix} and x_{bad} bear little resemblance. In the case of vision models, x_{fix} may even resemble static noise. Such extreme dissimilarities between x_{fix} and x_{bad} are not desirable because a user cannot be expected to feasibly interpret this information. Thus, we desire a similarity condition as:

$$\overline{\omega}(x_{\mathsf{bad}}) \odot x_{\mathsf{bad}} \approx \overline{\omega}(x_{\mathsf{bad}}) \odot x_{\mathsf{fix}}, \tag{P2}$$

where $\odot$ denotes element-wise vector multiplication. This similarity condition states that the non-anomalous regions of the original and the repair, as given by $\omega(x_{\mathsf{bad}})$, should remain similar.

Property 3 (Localized Improvement): The anomalous region should improve. However, P1 and P2 are not sufficient. For example, one might have $s(x_{\mathsf{fix}}) < s(x_{\mathsf{bad}})$, but have a higher score on the anomalous region $\omega(x_{\mathsf{bad}})$. This is not desirable because it means that x_{fix} has not actually fixed the anomalous region of x_{bad}. To ensure progress, we would like:

$$s_{\omega(x_{\mathsf{bad}})}(x_{\mathsf{fix}}) < s_{\omega(x_{\mathsf{bad}})}(x_{\mathsf{bad}}). \tag{P3}$$

Property 4 (Non-degradation): The non-anomalous region should not significantly worsen. Even when the above properties are satisfied, it is possible that the proposed repair could inadvertently increase the anomaly score on the non-anomalous region $\overline{\omega}(x_{\mathsf{bad}})$. This would mean repairing the anomalous region at the cost of corrupting the non-anomalous parts. We thus state the property against this as follows, where $\delta_4 > 0$ is a given tolerance threshold:

$$s_{\overline{\omega}(x_{\mathsf{bad}})}(x_{\mathsf{fix}}) \leq s_{\overline{\omega}(x_{\mathsf{bad}})}(x_{\mathsf{bad}}) + \delta_4. \tag{P4}$$

The benefit of our above formulation is that it encapsulates general, domain-independent desiderata of anomaly repairs. Importantly, this is achieved under the mild assumption of a linearly decomposable anomaly detector. We comment that some overlap among our proposed formal properties may arise in certain scenarios, and alternative sets could be more tailored for specific applications. Our goal, however, is to provide a foundational set of properties that ensures broad applicability, allowing further customization to suit individual application needs.

3 Property-guided Generation of Counterfactual Explanations

We now outline the process of conducting a property-guided repair for an anomalous input. In Section 3.1, we first introduce the generalized setup, where we define the four previously outlined properties as objective functions and frame the problem using risk-constrained optimization. Although this formulation clarifies the objectives, it is generally intractable. Therefore, in Section 3.2, we propose a diffusion-based algorithm to approximate the solution and achieve high-quality repairs.

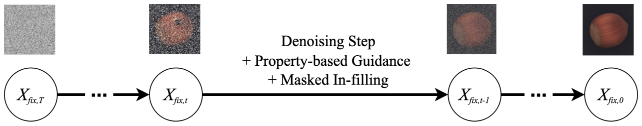


Figure 3: We run property-guided diffusion with masked in-filling.

3.1 A Generalized Formulation with Properties as Loss Functions

We first present a generalized setup for generating repairs. We conceptualize this in terms of a repair model $\mathcal{R}_{s,\omega}$ parametrized by an anomaly score function $s : \mathbb{R}^n \to \mathbb{R}$ with known linear decompositions α, β and a feature-wise binarization ω. To generate a repair, we sample the model:

$$x_{\mathsf{fix}} \sim \mathcal{R}_{s,\omega}(x_{\mathsf{bad}}),$$

where a probabilistic formulation is relevant in the context of variational auto-encoders [27] or diffusion models [24]. However, it is important that x_{fix} obeys the formal properties as outlined in Section 2.2. To do this, we cast these properties as loss functions listed below:

$$L_1 = s(x_{\mathsf{fix}}) \qquad \text{(P1 loss)}$$

$$L_2 = \|\overline{\omega}(x_{\mathsf{bad}}) \odot (x_{\mathsf{fix}} - x_{\mathsf{bad}})\|_2 \qquad \text{(P2 loss)}$$

$$L_3 = \max\big\{0,\ s_{\omega(x_{\mathsf{bad}})}(x_{\mathsf{fix}}) - s_{\omega(x_{\mathsf{bad}})}(x_{\mathsf{bad}})\big\} \qquad \text{(P3 loss)}$$

$$L_4 = \max\big\{0,\ s_{\overline{\omega}(x_{\mathsf{bad}})}(x_{\mathsf{fix}}) - s_{\overline{\omega}(x_{\mathsf{bad}})}(x_{\mathsf{bad}}) - \delta_4\big\} \qquad \text{(P4 loss)}$$

Our rationale is as follows. First, because the primary objective of anomaly repair is to reduce the anomaly score, we set L_1 as simply the score of x_{fix}. Second, we would like to ensure that x_{fix} and x_{bad} are similar in the non-anomalous region, and so formulate L_2 as the ℓ^2 distance between x_{fix} and x_{bad} over $\overline{\omega}(x_{\mathsf{bad}})$. Third, we formulate L_3 to apply a penalty when the anomalous region degrades in performance. Fourth, we allow for a degradation of score in the non-anomalous region $\overline{\omega}(x_{\mathsf{bad}})$, up to some tolerance threshold δ_4. We cast these as a risk-constrained optimization problem as follows:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathop{\mathbb{E}}_{x_{\mathsf{fix}} \sim \mathcal{R}_{s,\omega}(x_{\mathsf{bad}};\theta)} s(x_{\mathsf{fix}}) \\ \text{subject to} \quad & \mathop{\mathbb{P}}_{x_{\mathsf{fix}} \sim \mathcal{R}_{s,\omega}(x_{\mathsf{bad}};\theta)} \big[L_2 \leq \delta_2,\ \ L_3 \leq 0,\ \ L_4 \leq 0\big] \geq 1 - \delta \end{aligned} \tag{1}$$

where θ is a parameter of the repair model, $\delta_2 > 0$ is a given threshold for L_2 and $\delta > 0$ is a given failure probability for the violation of at least one of (P2), (P3), or (P4). We acknowledge that multiple formulations for property-based losses are valid; however, the chosen approach is optimally suited to our context.

3.2 Formal Property-guided Diffusion

We adopt techniques from guided diffusion [19] to generate repairs x_{fix}. We first give a brief overview of a standard diffusion process and then adapt it to perform property-guided generation. We refer to [62] for a comprehensive guide on diffusion but attempt to make the exposition self-contained.

Background. A basic variant of diffusion models takes the form:

$$p(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), b_t^2 I), \quad \text{for } t = T, \dots, 1, \tag{2}$$

where μ_θ is the *denoising model* with parameters θ, and $b_1 < \cdots < b_T$ is the variance schedule. When μ_θ is trained on non-anomalous data (see [62] for training details), one can generate non-anomalous samples of the training distribution by running the following iterative process:

$$x_T \sim \mathcal{N}(0, I), \quad x_{t-1} = \mu_\theta(x_t, t) + b_t z_t, \quad z_t \sim \mathcal{N}(0, I), \quad \text{for } t = T, \dots, 1, \tag{3}$$

where x_T is the initial noise and x_0 is the output sample. The idea is to repeatedly remove noise from a Gaussian x_T using μ_θ until the final x_0 resembles a high-resolution image without defects, for instance. The iterations (3) is also known as *backward process*.

Property-guided Diffusion. We next show how to adapt the denoising iterations (3) to produce repairs. We use two main ideas: first, we use guidance [19] to slightly nudge the iterates x_{t-1} of (3) at every step using a property-based loss, to encourage that the final iterate x_0, which we take to be x_{fix}, is more amenable to our properties. Second, we used masked-infilling [39] to ensure that the non-anomalous region $\overline{\omega}(x_{\mathsf{bad}})$ is generally preserved by the iterates. We implement this modified iteration as follows: beginning from the initial noise $x_{\mathsf{fix},T} \sim \mathcal{N}(0, I)$, let:

$$\hat{x}_{\mathsf{fix},t-1} = \underbrace{\mu_\theta(x_{\mathsf{fix},t}, t) + b_t z_t}_{\text{Denoising step}} - \underbrace{\eta_t \nabla L(x_{\mathsf{fix},t})}_{\text{Guidance term}}, \quad z_t \sim \mathcal{N}(0, I), \tag{4}$$

$$x_{\mathsf{bad},t} = \sqrt{a_t} x_{\mathsf{bad}} + \sqrt{1 - a_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I), \tag{5}$$

$$x_{\mathsf{fix},t-1} = \overline{\omega}(x_{\mathsf{bad}}) \odot x_{\mathsf{bad},t} + \omega(x_{\mathsf{bad}}) \odot \hat{x}_{\mathsf{fix},t-1}, \tag{6}$$

for $t = T, \dots, 1$, where $a_t = \prod_{i=1}^{t}(1 - b_i)$ and $\eta_1 < \cdots < \eta_T$ is the guidance schedule. The property-based loss is given by:

$$L(x_{\mathsf{fix},t}) = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 L_4, \tag{7}$$

with L_1, L_2, L_3, L_4 as in Section 3.1 and weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$. In the above, (4) first generates $\hat{x}_{\mathsf{fix},t-1}$ from $x_{\mathsf{fix},t}$ by combining a standard denoising step with a guidance term. Then, (6) combines $\hat{x}_{\mathsf{fix},t-1}$ with $\hat{x}_{\mathsf{bad},t}$, from (5), in a masked-infilling operation [39] to yield $x_{\mathsf{fix},t-1}$. This masked in-filling ensures similarity between x_{bad} and x_{fix} (i.e., $x_{\mathsf{fix},0}$) over the non-anomalous region $\overline{\omega}(x_{\mathsf{bad}})$.

We emphasize that the diffusion iterations given by (4), (5), (6), and do not guarantee the satisfaction of our formal properties. Rather, these iterations tend toward an output that better respects these properties — as we later shown in our experiments. In particular, our property-based losses define a way to evaluate the quality to which each property is satisfied or violated.

4 Experiments

Our experiments evaluate the performance of AR-Pro across vision and time-series datasets. In particular, we aim to address the following research questions:

- **(RQ1) Empirical Validation**: How well does AR-Pro repair anomalies for different domains? In particular, we investigate how well the four properties are satisfied when the diffusion process is guided or unguided, as in the baseline.
- **(RQ2) Ablation Study**: How do the different hyper-parameters affect the repair quality? We focus on the weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ for the four property-based losses.

Vision Anomaly Models and Datasets. For anomaly detectors, we used the anomalib [4] implementation of Fastflow [66] (with ResNet-50-2 backbone [68]) and Efficient-AD [11]. For datasets, we used the VisA [73] and MVTec-AD [13] datasets. VisA and MVTec-AD involve anomaly detection in the context of industrial manufacturing, where VisA consists of 12 image classes, and MVTec-AD consists of 15 image classes. Both FastFlow and Efficient-AD were trained with AdamW and a learning rate of 10^{-4} until convergence.

Time-series Anomaly Datasets and Models. For anomaly detectors, we used the GPT-2 [48] and Llama2 [59] architectures for time-series anomaly detection. In particular, we use only the first 6 layers of GPT-2 and the first 4 layers of Llama-2 (with an embedding dimension of 1024) to accelerate training. For datasets, we used the SWaT (51 features) [38], HAI (86 features) [54], and WADI (127 features) [2] datasets, split into sliding windows of size 100. Both our versions of GPT-2 and Llama-2 were trained with AdamW and a learning rate of 10^{-5} until convergence.

Diffusion-based Repair Models. We used the HuggingFace implementation of DDPM [24] for vision data and Diffusion-TS [67] for time-series data. Both models were trained on the non-anomalous instances of their respective datasets using AdamW and a learning rate of 10^{-4} until convergence.

Evaluation Metrics. We use the four property-based loss functions defined in Section 3.1 as our evaluation metrics. In particular, we measure the improvement of property-guided diffusion over un-guided diffusion. We adapt these metrics below, where we write ω to mean $\omega(x_{\mathsf{bad}})$ for brevity:

- **Property 1 (Overall Improvement):** $M_s \equiv s(x_{\mathsf{fix}})$.
- **Property 2 (Similarity):** $M_d \equiv \|\overline{\omega} \odot (x_{\mathsf{fix}} - x_{\mathsf{bad}})\|_2$
- **Property 3 (Localized Improvement):** $M_\omega \equiv s_\omega(x_{\mathsf{fix}}) - s_\omega(x_{\mathsf{bad}})$
- **Property 4 (Non-degradation):** $M_{\overline{\omega}} \equiv s_{\overline{\omega}}(x_{\mathsf{fix}}) - s_{\overline{\omega}}(x_{\mathsf{bad}})$

Dataset	**Class**	$M_s(\downarrow)$		$M_d(\downarrow)$		$M_\omega(\downarrow)$		$M_{\overline{\omega}}(\downarrow)$	
		Baseline	Guided	Baseline	Guided	Baseline	Guided	Baseline	Guided
VisA	candle	-23.39 ± 1.23	-26.43 ± 0.14	336.06 ± 44.41	8.34 ± 0.06	-0.003 ± 0.002	-0.007 ± 0.001	0.05 ± 0.01	-0.02 ± 0.004
	capsules	12.45 ± 30.02	-17.86 ± 0.20	426.69 ± 45.30	8.41 ± 0.01	0.009 ± 0.003	-0.005 ± 0.001	0.15 ± 0.02	-0.01 ± 0.002
	cashew	-16.08 ± 0.85	-17.13 ± 0.33	307.39 ± 63.72	8.40 ± 0.01	-0.002 ± 0.004	-0.005 ± 0.001	0.06 ± 0.03	-0.00 ± 0.002
	chewinggum	-14.56 ± 0.89	-15.70 ± 0.14	258.32 ± 53.66	8.41 ± 0.01	-0.007 ± 0.003	-0.009 ± 0.003	0.02 ± 0.02	-0.01 ± 0.002
	fryum	-13.54 ± 3.02	-18.93 ± 0.42	349.18 ± 32.39	8.36 ± 0.01	0.003 ± 0.005	-0.007 ± 0.001	0.12 ± 0.04	-0.01 ± 0.003
	macaroni1	199.95 ± 145.07	-25.63 ± 0.25	658.18 ± 81.65	8.41 ± 0.01	0.012 ± 0.002	-0.005 ± 0.001	0.18 ± 0.02	-0.01 ± 0.004
	macaroni2	-15.61 ± 11.75	-25.18 ± 0.17	550.91 ± 55.19	8.41 ± 0.01	0.004 ± 0.003	-0.004 ± 0.001	0.11 ± 0.02	-0.01 ± 0.002
	pcb1	-23.49 ± 0.49	-25.04 ± 1.08	325.54 ± 50.82	8.38 ± 0.01	-0.005 ± 0.006	-0.006 ± 0.003	0.05 ± 0.02	-0.01 ± 0.004
	pcb2	-18.20 ± 0.67	-19.15 ± 0.12	289.54 ± 48.32	8.41 ± 0.01	-0.005 ± 0.003	-0.005 ± 0.001	0.03 ± 0.02	-0.01 ± 0.002
	pcb3	-19.87 ± 4.29	-24.19 ± 0.15	291.16 ± 50.72	8.41 ± 0.01	-0.003 ± 0.004	-0.005 ± 0.002	0.05 ± 0.04	-0.01 ± 0.002
	pcb4	2.51 ± 23.68	-20.24 ± 0.09	337.12 ± 75.48	8.41 ± 0.01	0.000 ± 0.004	-0.005 ± 0.001	0.12 ± 0.05	-0.01 ± 0.003
	pipefryum	-15.08 ± 3.83	-19.80 ± 0.22	252.68 ± 78.99	8.41 ± 0.01	-0.002 ± 0.006	-0.008 ± 0.001	0.09 ± 0.05	-0.02 ± 0.005
	$\Delta(\uparrow)$	**+26.54%**		**+97.46%**		**+146.42%**		**+114.16%**	
MVTec-AD	bottle	-13.67 ± 0.03	-13.16 ± 1.42	252.11 ± 41.14	8.70 ± 1.76	-0.016 ± 0.001	-0.012 ± 0.001	-0.05 ± 0.01	-0.05 ± 0.003
	cable	-12.56 ± 0.13	-13.18 ± 0.17	206.08 ± 80.64	5.03 ± 1.94	-0.006 ± 0.003	-0.007 ± 0.003	0.02 ± 0.01	-0.01 ± 0.005
	capsule	-13.73 ± 0.35	-15.21 ± 0.31	105.41 ± 48.80	5.07 ± 2.23	-0.004 ± 0.003	-0.006 ± 0.002	0.04 ± 0.01	-0.01 ± 0.006
	carpet	-18.93 ± 0.35	-19.47 ± 0.70	201.02 ± 82.18	4.98 ± 2.04	-0.014 ± 0.005	-0.014 ± 0.005	0.00 ± 0.02	-0.01 ± 0.010
	grid	40.78 ± 115.98	-13.73 ± 0.11	556.52 ± 65.32	10.01 ± 0.85	0.010 ± 0.002	-0.010 ± 0.002	0.17 ± 0.02	-0.02 ± 0.004
	hazelnut	-10.34 ± 0.23	-12.94 ± 0.21	127.33 ± 52.05	5.07 ± 1.98	-0.003 ± 0.004	-0.011 ± 0.004	0.09 ± 0.01	-0.02 ± 0.005
	leather	-12.81 ± 0.14	-13.85 ± 0.08	484.22 ± 75.46	10.01 ± 1.30	-0.013 ± 0.002	-0.014 ± 0.002	0.02 ± 0.01	-0.02 ± 0.005
	metal nut	-8.94 ± 0.34	-11.05 ± 0.33	214.22 ± 104.32	5.71 ± 2.74	-0.001 ± 0.004	-0.007 ± 0.002	0.07 ± 0.02	-0.02 ± 0.003
	pill	-10.90 ± 0.08	-11.81 ± 0.17	242.05 ± 20.04	10.88 ± 0.79	-0.006 ± 0.001	-0.007 ± 0.001	0.01 ± 0.01	-0.02 ± 0.001
	screw	-11.28 ± 0.28	-13.10 ± 0.07	432.16 ± 66.10	10.16 ± 1.49	-0.001 ± 0.001	-0.006 ± 0.001	0.05 ± 0.01	-0.02 ± 0.002
	tile	-11.25 ± 0.44	-12.18 ± 0.57	425.23 ± 44.84	9.91 ± 1.03	-0.012 ± 0.002	-0.015 ± 0.004	0.04 ± 0.01	-0.03 ± 0.006
	toothbrush	-6.43 ± 0.10	-8.33 ± 0.05	151.76 ± 25.94	9.27 ± 1.77	-0.002 ± 0.002	-0.009 ± 0.002	0.05 ± 0.00	-0.03 ± 0.003
	transistor	-12.15 ± 0.18	-12.64 ± 5.51	278.70 ± 69.53	7.27 ± 1.83	-0.007 ± 0.002	-0.007 ± 0.001	0.02 ± 0.01	-0.01 ± 0.002
	wood	-9.65 ± 2.29	-13.54 ± 0.26	344.91 ± 67.58	10.17 ± 1.75	-0.013 ± 0.003	-0.019 ± 0.001	0.03 ± 0.02	-0.02 ± 0.005
	zipper	-12.62 ± 0.19	-13.24 ± 0.05	314.25 ± 34.11	10.10 ± 1.27	-0.009 ± 0.001	-0.009 ± 0.001	-0.00 ± 0.00	-0.03 ± 0.002
	$\Delta(\uparrow)$	**+8.35%**		**+97.34%**		**+29.15%**		**+154.76%**	

Table 1: AR-Pro outperforms a non-guided diffusion baseline across our four metrics. We show the results for all VisA and MVTec-AD categories, where Δ is the median percentage improvement.

Each experiment employs a representative anomaly detector and dataset with predefined train-test splits. The performance of the anomaly detectors is detailed in Appendix B. For ω, each feature-wise threshold τ_i is taken to be the 90% quantile of the training set's feature-wise anomaly scores.

4.1 (RQ1) Empirical Validation

We now evaluate the performance of AR-Pro for generating anomaly repairs with respect to the four evaluation metrics. We report the mean and standard deviation values. Because there may be a large range of values across classes, we report the median improvement (Δ) of the guided generation over the baseline. We present results for the FastFlow anomaly detector on the VisA and MVTec dataset in Table 1, and refer to Appendix C for additional results with Efficient-AD.

Quantitative Results. For all the categories, the guided results show significant improvement over the baseline on the formal criteria, with an average improvement of 84.27%. There is a wide range of M_s values for baseline models in certain classes, possibly due to deviations in the color scheme of the generated images from their training distribution. Hence, we report the median improvement across the classes in the last row to mitigate the impact of outliers.

We show results for time-series data in Table 2. AR-Pro achieves lower error on M_d, M_ω, and $M_{\overline{\omega}}$, with an overall improvement of 60.03% over the baseline. Specifically, Llama-2 achieves an average improvement of 67.17% in formal metrics, while GPT-2 increases by 53.90% on average. Compared to image data, the performance on M_d is not as competitive and exhibits considerable variability, likely due to the broader range of adjustments required for repairs to ensure smooth signals.

In addition, we evaluate whether the guided repairs generate non-anomalous samples by comparing them against a conformity threshold with 95% confidence derived from the training set [9]. Treating the non-anomalous class as the negative, we report the True Negative Rate (TNR) in Table 3. The TNR of most VisA and MVTec-AD categories reaches 100%, surpassing the majority of baseline values. As a result, the guided repair achieves a median TNR of 100% for both VisA and MVTec, representing an average improvement of 2.50% over the baseline. Across the three time-series datasets (SWaT, HAI, and WADI), the guided repair obtains an average TNR of 95.33%, which is 92.66% higher than the baseline average of 2.67%. Overall, 99.26% of guided repairs across both domains are classified as non-anomalous with 95% confidence, statistically confirming the effectiveness of AR-Pro.

Qualitative Results. We present qualitative examples here to illustrate that our method can generate semantically meaningful repairs, as shown in Figure 4 for VisA and Figure 5 for MVTec-AD. For

Model	Dataset	$M_s(\downarrow)$		$M_d(\downarrow)$		$M_\omega(\downarrow)$		$M_{\overline{\omega}}(\downarrow)$	
		Baseline	Guided	Baseline	Guided	Baseline	Guided	Baseline	Guided
Llama2	SWaT	0.83 ± 0.05	0.59 ± 0.04	3.05 ± 0.61	7.25 ± 2.50	0.084 ± 0.026	-0.026 ± 0.008	0.19 ± 0.02	0.05 ± 0.012
	WADI	0.97 ± 0.00	0.26 ± 0.01	0.98 ± 0.01	0.00 ± 0.00	0.087 ± 0.005	0.000 ± 0.000	0.61 ± 0.01	0.00 ± 0.000
	HAI	0.92 ± 0.00	0.58 ± 0.01	0.75 ± 0.00	0.00 ± 0.00	0.166 ± 0.008	0.000 ± 0.000	0.18 ± 0.01	0.00 ± 0.000
	Δ(↑)	**+46.36%**		**+20.77%**		**+110.32%**		**+91.23%**	
GPT-2	SWaT	0.68 ± 0.06	0.57 ± 0.05	12.81 ± 15.82	16.28 ± 16.30	-0.029 ± 0.041	-0.094 ± 0.038	0.13 ± 0.04	0.09 ± 0.031
	WADI	0.71 ± 0.04	0.32 ± 0.00	34.45 ± 0.60	42.15 ± 0.94	0.018 ± 0.003	-0.019 ± 0.002	0.43 ± 0.04	0.07 ± 0.002
	HAI	0.92 ± 0.02	0.61 ± 0.11	3.69 ± 9.72	5.67 ± 18.80	0.136 ± 0.050	-0.005 ± 0.018	0.21 ± 0.03	0.04 ± 0.127
	Δ(↑)	**+34.93%**		**-34.37%**		**+177.79%**		**+37.24%**	

Table 2: Comparison of baseline and guided performance across four metrics for SWaT, WADI, and HAI dataset categories with Llama2 and GPT2 model. Δ is the median improvement percentage of the guided result from baseline.

Dataset	Category	Baseline TNR (↑)	Guided TNR (↑)	Category	Baseline TNR (↑)	Guided TNR (↑)
VisA	Candle	**1.00**	**1.00**	Fryum	0.66	**1.00**
	Capsules	0.00	**1.00**	Pipe Fryum	**1.00**	**1.00**
	Cashew	**1.00**	**1.00**	PCB 1	**1.00**	0.96
	Chewinggum	**1.00**	**1.00**	PCB 2	1.00	**1.00**
	Macaroni 1	0.11	**1.00**	PCB 3	0.90	**1.00**
	Macaroni 2	0.52	**1.00**	PCB 4	0.18	**1.00**
MVTec-AD	Bottle	**1.00**	**1.00**	Grid	0.00	**1.00**
	Cable	**1.00**	**1.00**	Hazelnut	**1.00**	**1.00**
	Capsule	**1.00**	**1.00**	Leather	**1.00**	**1.00**
	Carpet	**1.00**	**1.00**	Metal Nut	**1.00**	**1.00**
	Pill	**1.00**	**1.00**	Screw	**1.00**	**1.00**
	Tile	**1.00**	**1.00**	Toothbrush	**1.00**	0.96
	Transistor	0.56	**1.00**	Wood	**1.00**	**1.00**
	Zipper	**1.00**	**1.00**			
Overvall Median	VisA	0.95	**1.00**	MVTec-AD	**1.00**	**1.00**
	SWaT	0.08	**0.86**	HAI	0.00	**1.00**
	WADI	0.00	**1.00**			

Table 3: True Negative Rate (TNR) for categories in the vision (VisA, MVTec) and time-series (SWaT, HAI, and WADI) anomaly detection datasets.

example in Figure 4, we observe that for categories PCB 1, PCB 2, PCB 3, and PCB 4, the baseline fails to rigorously repair the anomaly: the orientation of the board is reversed for PCB 1; unintended white marks appear on the lower left of PCB 2; the dark lines in the potentiometer change for PCB 3; and the “FC-75” label is missing for PCB 4. However, by integrating formal property guidance, our approach accurately reconstructs all these details while effectively removing the anomalies. For the MVTec-AD examples, AR-Pro produces repairs that more closely resemble the original inputs compared to those generated by the baseline. This demonstrates that our method’s generated repairs adhere more rigorously to the formal properties.

In Figure 6, we present examples from our time-series repair, where the anomalous time segment is shaded in red. Our results demonstrate that AR-Pro generates a signal that resembles the original signal better, as shown in the first two plots of Figure 6. In addition, we recover the sensor time series to non-anomalous values when the baseline fails to repair the anomaly, as shown in the last two plots of Figure 6.

More repair examples are available in Appendix D. However, although the quality of the generated repairs has improved, we notice that this enhancement comes with a trade-off of increased inference time. Further details can be found in Appendix E.

4.2 (RQ2) Ablation Study

We randomly sampled 100 instances to compute the mean of each metric in order to evaluate the effect of hyper-parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ associated with each property-based loss. Each line in the plots represents results obtained while keeping the other hyper-parameters at 1.0. The ablation results for the time series are presented in Figure 7, with additional plots in Appendix F.

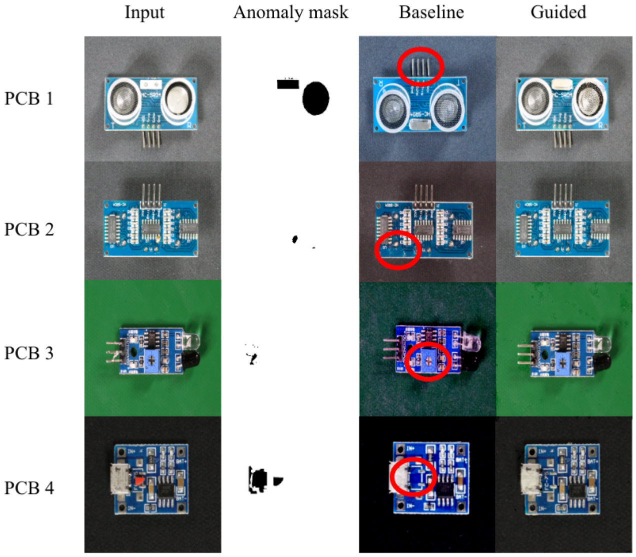


Figure 4: The original input and ground truth anomaly mask are displayed in the first two columns. The baseline method fails to preserve close similarity to the input PCB boards, as highlighted in the third column. Guided vision repair examples in the fourth column address these deficiencies.

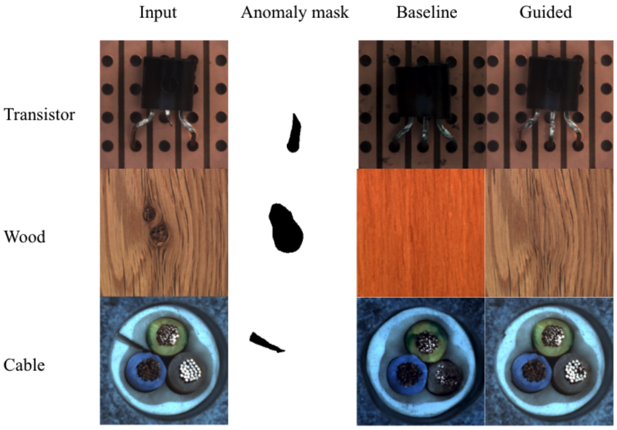


Figure 5: MVTec repairs with AR-Pro; better resemble the original compared to the baseline.

Our observations indicate that variations in the scale of property hyperparameters do not significantly impact the formal metrics, as the range of change remains relatively small. In addition, no consistent trends were observed when varying the $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ hyper-parameters. This suggests that our framework demonstrates robust performance, and extensive tuning may be unnecessary.

5 Related Work

Numerous techniques have been developed for anomaly detection across various domains [3, 49]. Traditional approaches to anomaly detection include clustering [43, 57] and statistical methods [34] such as ARIMA [41] and Gaussian models [52]. However, these methods often struggle with high-dimensional data. More recently, deep learning-based anomaly detection [20, 28], including autoencoders [27] and GANs [18], can detect high-dimensional anomalies via reconstruction error. For time-series data, LSTMs [32] and transformer-based models [60, 63, 69, 72] have shown exceptional performance. Additionally, diffusion models are emerging as promising tools for visual anomaly detection [42, 71]. While these methods vary in strengths and are continually improving, explaining

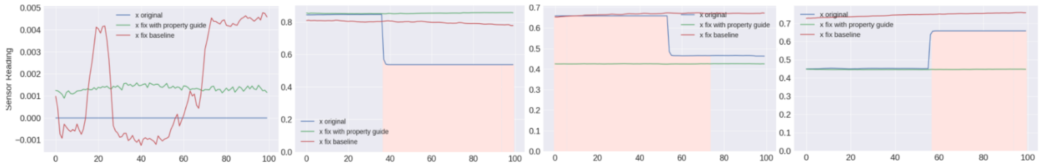
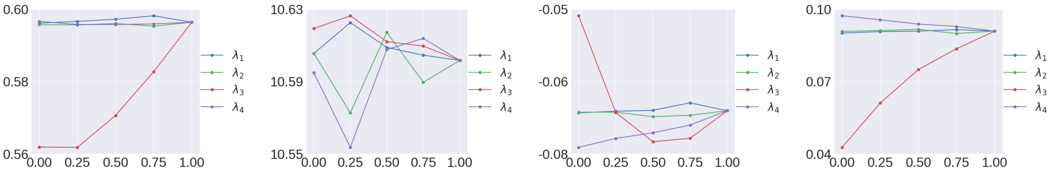


Figure 6: Original input is the blue line, property guided fix is the green line, and the baseline is the red line. The first image shows that the baseline generates a spurious signal when there is no anomaly. The second image shows that the baseline repairs the anomaly, but not as effectively as with guidance. The last two images show instances where the baseline fails to repair the anomaly.



(a) M_s (b) M_d (c) M_ω (d) $M_{\overline{\varpi}}$

Figure 7: Varying the hyper-parameters does significantly change M_s, M_d, M_ω, and $M_{\overline{\varpi}}$.

anomalies remains a challenge [10]. Most current methods rely on feature importance scores or visualizations [44], such as gradients or reconstructions [46], which often fail to provide actionable insights. The lack of formal frameworks [65] and consistent evaluation metrics [1] complicates this issue. For example, the absence of formal metrics leads to inconsistencies in evaluation [26], underscoring the need for more rigorous approaches and reliable criteria [37]. Most similar to our work is [56], which also performs time series-specific generation of counterfactual explanations in the form of anomaly repairs but considers a different set of properties and does not use diffusion. Our work also focuses on generative modeling to produce counterfactual explanations in the form of anomaly repairs. For vision, the current leading paradigms are diffusion models [24] and generative adversarial networks [23]. Diffusion models are also applicable to time-series data [67], and we refer to [22] for a survey on other techniques.

6 Discussion

The main theoretical contribution of our work is the identification of common counterfactual explanation desiderata for linearly decomposable anomaly detectors. While we have identified four formal properties, we acknowledge that other valid ones may also exist. Moreover, we recommend that practitioners evaluate and choose the properties necessary for the particular problem, and this is made possible by the form of our diffusion guidance function in (7). The quality of anomaly repairs depends on the performance of the anomaly detector and the generative model. While there may have been limitations in our efforts, we found it challenging to use variational auto-encoders [27] for generating high-quality repairs. Furthermore, our implementation is focused on diffusion models, but the ideas presented can also be extended to other generative techniques.

7 Conclusion

We present AR-Pro, a framework for generating and evaluating counterfactual explanations in anomaly detection. We use the fact that common anomaly detectors are linearly decomposable, which lets us define formal, general, domain-independent properties for explainability. Using these properties, we show how to generate high-quality counterfactuals using a property-guided diffusion setup. We demonstrate the effectiveness of AR-Pro on vision and time-series datasets and showcase our improvement over off-the-shelf diffusion models.

Acknowledgement This work was supported in part by ARO MURI W911NF2010080, NSF-2125561, and NSF-2143274.

References

[1] C. Agarwal, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, and H. Lakkaraju. Openxai: Towards a transparent evaluation of model explanations. *Advances in Neural Information Processing Systems*, 35:15784–15799, 2022.

[2] C. M. Ahmed, V. R. Palleti, and A. P. Mathur. Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks*, pages 25–28, 2017.

[3] M. Ahmed, A. N. Mahmood, and M. R. Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[4] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc. Anomalib: A deep learning library for anomaly detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710. IEEE, 2022.

[5] N. Alahmadi, S. A. Evdokimov, Y. Kropotov, A. M. Müller, and L. Jäncke. Different resting state eeg features in children from switzerland and saudi arabia. *Frontiers in human neuroscience*, 10:559, 2016.

[6] A. L. Alfeo and M. G. Cimino. Counterfactual-based feature importance for explainable regression of manufacturing production quality measure. In *ICPRAM*, pages 48–56, 2024.

[7] G. Alon and M. Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

[8] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

[9] A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

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

[11] K. Batzner, L. Heckler, and R. König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 128–138, 2024.

[12] V. Belle and I. Papantonis. Principles and practice of explainable machine learning. *Frontiers in big Data*, 4:688969, 2021.

[13] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[14] A. Bibal, M. Lognoul, A. De Streel, and B. Frénay. Legal requirements on explainability in machine learning. *Artificial Intelligence and Law*, 29:149–169, 2021.

[15] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[16] Z. Chen, G. Xu, V. Mahalingam, L. Ge, J. Nguyen, W. Yu, and C. Lu. A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3:10–23, 2016.

[17] G. D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, and R. G. Mark. Classification of normal/abnormal heart sound recordings: The physionet/computing in cardiology challenge 2016. In *2016 Computing in cardiology conference (CinC)*, pages 609–612. IEEE, 2016.

[18] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[19] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[20] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 54(7):1–37, 2021.

[21] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

[22] F. Gatta, F. Giampaolo, E. Prezioso, G. Mei, S. Cuomo, and F. Piccialli. Neural networks generative models for time series. *Journal of King Saud University-Computer and Information Sciences*, 34(10):7920–7939, 2022.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[24] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[25] A. Høst-Madsen, E. Sabeti, and C. Walton. Data discovery and anomaly detection using atypicality: Theory. *IEEE Transactions on Information Theory*, 65(9):5302–5322, 2019.

[26] J. Jiang, F. Leofante, A. Rago, and F. Toni. Formalising the robustness of counterfactual explanations for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[27] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22:949–961, 2019.

[29] C.-Y. Lai, F.-K. Sun, Z. Gao, J. H. Lang, and D. S. Boning. Nominality score conditioned time series anomaly detection by point/sequential reconstruction. *arXiv preprint arXiv:2310.15416*, 2023.

[30] X. A. Larriva-Novo, M. Vega-Barbas, V. A. Villagrá, and M. S. Rodrigo. Evaluation of cybersecurity data set characteristics for their applicability to neural networks algorithms detecting cybersecurity anomalies. *IEEE Access*, 8:9005–9014, 2020.

[31] V. Lin, K. J. Jang, S. Dutta, M. Caprio, O. Sokolsky, and I. Lee. Dc4l: Distribution shift recovery via data-driven control for deep learning models. In *6th Annual Learning for Dynamics & Control Conference*, pages 1526–1538. PMLR, 2024.

[32] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich. A survey on anomaly detection for technical systems using lstm networks. *Computers in Industry*, 131:103498, 2021.

[33] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[34] C. Manikopoulos and S. Papavassiliou. Network intrusion and fault detection: a statistical anomaly approach. *IEEE Communications Magazine*, 40(10):76–82, 2002.

[35] A. Manolache, F. Brad, and E. Burceanu. Date: Detecting anomalies in text via self-supervision of transformers. *arXiv preprint arXiv:2104.05591*, 2021.

[36] R. Marcinkevičs and J. E. Vogt. Interpretable and explainable machine learning: a methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(3):e1493, 2023.

[37] J. Marques-Silva and A. Ignatiev. Delivering trustworthy ai through formal xai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[38] A. P. Mathur and N. O. Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE, 2016.

[39] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[40] A. Moallemi, A. Burrello, D. Brunelli, and L. Benini. Exploring scalable, distributed real-time anomaly detection for bridge health monitoring. *IEEE Internet of Things Journal*, 9(18):17660–17674, 2022.

[41] H. Z. Moayedi and M. Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *2008 international symposium on information technology*, volume 4, pages 1–6. IEEE, 2008.

[42] A. Mousakhan, T. Brox, and J. Tayyub. Anomaly detection with conditioned denoising diffusion models. *arXiv preprint arXiv:2305.15956*, 2023.

[43] G. Münz, S. Li, and G. Carle. Traffic anomaly detection using k-means clustering. In *Gi/itg workshop mmbnet*, volume 7, 2007.

[44] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys*, 55(13s):1–42, 2023.

[45] G. Pallotta, M. Vespe, and K. Bryan. Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. *Entropy*, 15(6):2218–2245, 2013.

[46] G. Pang and C. Aggarwal. Toward explainable deep anomaly detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4056–4057, 2021.

[47] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133:113303, 2020.

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[49] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke. A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study. *Sensors*, 18(8):2491, 2018.

[50] M. Reyes, R. Meier, S. Pereira, C. A. Silva, F.-M. Dahlweid, H. v. Tengg-Kobligk, R. M. Summers, and R. Wiest. On the interpretability of artificial intelligence in radiology: challenges and opportunities. *Radiology: artificial intelligence*, 2(3):e190043, 2020.

[51] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[52] P. J. Rousseeuw and M. Hubert. Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2):e1236, 2018.

[53] K. K. Santhosh, D. P. Dogra, and P. P. Roy. Anomaly detection in road traffic using visual surveillance: A survey. *ACM Computing Surveys (CSUR)*, 53(6):1–26, 2020.

[54] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim. {HAI} 1.0:{HIL-based} augmented {ICS} security dataset. In *13Th USENIX workshop on cyber security experimentation and test (CSET 20)*, 2020.

[55] K. Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[56] D. Sulem, M. Donini, M. B. Zafar, F.-X. Aubet, J. Gasthaus, T. Januschowski, S. Das, K. Kenthapadi, and C. Archambeau. Diverse counterfactual explanations for anomaly detection in time series. *arXiv preprint arXiv:2203.11103*, 2022.

[57] I. Syarif, A. Prugel-Bennett, and G. Wills. Unsupervised clustering approach for network anomaly detection. In *Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. Proceedings, Part I 4*, pages 135–145. Springer, 2012.

[58] J. J. Thiagarajan, K. Thopalli, D. Rajan, and P. Turaga. Training calibration-based counterfactual explainers for deep learning models in medical image analysis. *Scientific reports*, 12(1):597, 2022.

[59] H. Touvron, L. Martin, K. Stone, P.-E. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, A. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[60] S. Tuli, G. Casale, and N. R. Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[62] L. Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

[63] J. Xu, H. Wu, J. Wang, and M. Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

[64] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek. A comprehensive study of jailbreak attack versus defense for large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7432–7449, 2024.

[65] J. Yu, A. Ignatiev, P. J. Stuckey, N. Narodytska, and J. Marques-Silva. Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[66] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.

[67] X. Yuan and Y. Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*, 2024.

[68] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[69] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[70] A. Zhang, S. Song, J. Wang, and P. S. Yu. Time series data cleaning: From anomaly detection to anomaly repairing. *Proceedings of the VLDB Endowment*, 10(10):1046–1057, 2017.

[71] H. Zhang, Z. Wang, Z. Wu, and Y.-G. Jiang. Diffusionad: Denoising diffusion for anomaly detection. *arXiv preprint arXiv:2303.08730*, 2023.

[72] T. Zhou, P. Niu, L. Sun, R. Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36, 2024.

[73] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022.

A Computational Resources

All experiments were done on a server with three NVIDIA GeForce RTX 4090 GPUs.

B Anomaly Detector Performance

In this section, we report the performance of anomaly detectors. The results for FastFlow [66] are presented in Table 4, and the results for GPT-2 [48] are shown in Table 5. We did not perform extensive parameter tuning, as the performance of anomaly detectors is not the primary focus of our work. In addition, it is recommended that users adhere to usage guidelines of using GPT-2 or implementing safety filters.

C Additional Experiments

We include additional model EfficientAD [11] and dataset MVTec [13]. The results are shown in Table 6. EfficientAD achieved an 88.24% average improvement across the four metrics on MVTec-AD and 74.85% improvement on VisA.

D More Qualitative Examples

Additional VisA examples can be found in Figure 8, while additional MVTec-AD examples can be found in Figure 9 and Figure 10. Additional time-series examples can be found in Figure 11.

E Inference Time

We randomly sampled 50 instances from the testing set to compare inference times, as shown in Table 7. We observed that although property guidance generates higher-quality repairs, it results in slightly longer inference times for time series data and significantly longer times for image data, possibly due to the higher dimensionality of the inputs. In total, it took about 40 hours to finish RQ1 for VisA and 25 hours to finish SWaT. For RQ2, it took 5 hours for SWaT. Improving computation time will be a focus of our future work.

F More Ablation Plots

In addition to λ_1 to λ_4, we also perform ablation on the overall guidance scale, which we denote using λ_ϕ. The The ablation study for λ_ϕ on time series can be found in Figure 12. The ablation for image data, using cashew class as an example can be found in Figure 13 and Figure 14.

Category	Image AUROC	Pixel AUROC
Candle	0.6501	0.5617
Capsules	0.5352	0.8188
Cashew	0.3816	0.7558
Chewing Gum	0.3754	0.9362
Fryum	0.761	0.4244
Macaroni1	0.2607	0.9127
Macaroni2	0.504	0.7729
PCB1	0.7173	0.9163
PCB2	0.7398	0.7945
PCB3	0.5361	0.8073
PCB4	0.6331	0.6183
Pipe Fryum	0.4696	0.2799
Average	0.5470	0.7166

Table 4: AUROC Scores of Fastflow for Various Categories in VisA

Metric	GPT-2 on SWaT
Accuracy	0.9784
Precision	0.8831
Recall	0.9472
F1-score	0.9140

Table 5: Performance of GPT-2 anomaly detectors on SWaT Datasets

Model	**Class**	$M_s(\downarrow)$		$M_d(\downarrow)$		$M_\omega(\downarrow)$		$M_{\overline{\omega}}(\downarrow)$	
		Baseline	Guided	Baseline	Guided	Baseline	Guided	Baseline	Guided
EfficientAD (MVTec-AD)	bottle	24.91 ± 1.73	15.34 ± 2.49	252.67 ± 25.10	9.00 ± 1.11	0.014 ± 0.462	-0.388 ± 0.411	1.53 ± 0.21	-0.02 ± 0.030
	cable	6.35 ± 1.00	6.35 ± 0.63	377.28 ± 55.71	9.14 ± 1.20	-0.120 ± 0.052	-0.138 ± 0.031	0.32 ± 0.07	-0.03 ± 0.009
	capsule	141.59 ± 3.43	142.81 ± 2.38	223.74 ± 26.83	10.58 ± 1.23	-0.366 ± 0.395	-0.120 ± 0.202	-0.29 ± 0.31	-0.07 ± 0.081
	carpet	6.71 ± 3.11	0.70 ± 0.04	412.91 ± 50.80	10.17 ± 1.28	0.340 ± 0.220	-0.011 ± 0.004	2.23 ± 1.34	-0.00 ± 0.001
	grid	3.24 ± 1.06	0.32 ± 0.02	555.47 ± 74.34	10.03 ± 0.59	0.220 ± 0.086	-0.009 ± 0.003	1.56 ± 0.58	-0.00 ± 0.001
	hazelnut	94.40 ± 7.79	48.76 ± 18.15	238.92 ± 26.92	9.54 ± 0.94	2.380 ± 1.002	-0.856 ± 0.775	20.99 ± 1.98	0.05 ± 0.061
	leather	37.15 ± 2.45	16.42 ± 0.63	481.82 ± 68.25	9.99 ± 1.43	1.347 ± 0.373	-0.249 ± 0.186	12.03 ± 1.86	0.01 ± 0.014
	metal nut	3.51 ± 0.09	2.56 ± 0.11	421.94 ± 56.50	10.40 ± 1.32	0.024 ± 0.033	-0.025 ± 0.035	0.63 ± 0.05	-0.00 ± 0.002
	pill	4.56 ± 0.09	2.38 ± 0.11	261.21 ± 15.65	10.88 ± 0.70	0.178 ± 0.026	-0.014 ± 0.020	0.55 ± 0.04	-0.01 ± 0.005
	screw	1.05 ± 0.20	0.98 ± 0.34	376.92 ± 40.00	10.22 ± 1.00	0.020 ± 0.004	0.006 ± 0.004	0.10 ± 0.00	0.01 ± 0.001
	tile	6.67 ± 1.98	0.82 ± 0.27	432.18 ± 62.29	9.85 ± 1.47	0.094 ± 0.308	-0.318 ± 0.236	2.75 ± 1.08	-0.02 ± 0.012
	toothbrush	50.02 ± 5.44	35.26 ± 9.41	142.70 ± 25.12	9.27 ± 1.69	-0.175 ± 0.265	-0.013 ± 0.056	1.13 ± 0.63	-0.02 ± 0.011
	transistor	37.40 ± 3.66	46.66 ± 10.74	270.86 ± 72.84	26.60 ± 100.98	-1.355 ± 0.672	-0.311 ± 0.485	-1.78 ± 1.96	-0.35 ± 1.506
	wood	91.43 ± 54.07	22.96 ± 2.72	353.12 ± 47.96	10.24 ± 1.24	1.829 ± 0.560	-0.489 ± 0.487	14.30 ± 3.73	-0.15 ± 0.066
	zipper	1.37 ± 0.20	0.52 ± 0.12	305.86 ± 32.46	9.91 ± 1.01	0.030 ± 0.009	-0.015 ± 0.003	0.25 ± 0.02	-0.00 ± 0.002
	$\Delta(\uparrow)$	**+47.85%**		**+97.10%**		**+107.84%**		**+100.16%**	
EfficientAD (VisA)	candle	280.37 ± 48.14	387.79 ± 60.55	166.18 ± 27.27	8.71 ± 1.27	-7.219 ± 3.677	-0.212 ± 0.379	-6.98 ± 7.23	-0.22 ± 0.285
	capsules	28.98 ± 10.27	31.01 ± 5.50	371.05 ± 52.72	8.41 ± 0.01	-0.570 ± 0.227	-0.036 ± 0.042	2.06 ± 0.74	-0.01 ± 0.012
	cashew	364.66 ± 35.11	363.13 ± 31.62	255.69 ± 54.96	8.76 ± 1.85	-2.968 ± 4.098	-2.778 ± 3.437	-0.32 ± 2.53	0.06 ± 1.063
	chewinggum	81.47 ± 4.01	75.87 ± 11.40	179.61 ± 15.37	8.40 ± 0.01	-0.216 ± 0.498	-0.176 ± 0.486	0.76 ± 0.53	-0.02 ± 0.020
	fryum	267.90 ± 55.84	130.03 ± 18.19	222.72 ± 32.44	8.38 ± 0.01	5.766 ± 2.300	-0.539 ± 1.032	9.62 ± 2.04	-0.08 ± 0.348
	macaroni1	881.24 ± 32.80	657.44 ± 16.24	227.79 ± 13.91	8.41 ± 0.01	16.696 ± 3.118	-1.117 ± 0.735	25.61 ± 4.36	-0.18 ± 0.195
	macaroni2	350.04 ± 63.90	198.60 ± 29.18	244.22 ± 25.98	8.41 ± 0.01	3.070 ± 2.147	-0.523 ± 0.799	3.65 ± 1.63	0.17 ± 0.375
	pcb1	24.09 ± 2.89	15.87 ± 0.56	208.50 ± 31.13	8.37 ± 0.02	0.293 ± 0.068	-0.001 ± 0.027	0.21 ± 0.09	-0.00 ± 0.003
	pcb2	10.61 ± 0.67	9.07 ± 1.22	496.76 ± 52.56	8.41 ± 0.01	0.125 ± 0.157	-0.040 ± 0.029	3.66 ± 0.38	-0.01 ± 0.005
	pcb3	15.06 ± 8.65	27.82 ± 3.88	363.07 ± 38.92	8.41 ± 0.01	-0.708 ± 0.368	-0.007 ± 0.011	-1.34 ± 0.95	-0.01 ± 0.007
	pcb4	61.88 ± 14.91	79.72 ± 20.28	298.36 ± 27.41	8.41 ± 0.01	0.020 ± 0.204	-0.110 ± 0.179	1.53 ± 0.19	0.02 ± 0.014
	pipe fryum	112.89 ± 16.76	111.60 ± 8.36	117.01 ± 33.67	8.41 ± 0.01	0.619 ± 1.512	0.018 ± 1.140	0.40 ± 0.97	-0.20 ± 0.289
	$\Delta(\uparrow)$	**+4.01%**		**+96.43%**		**+98.65%**		**+100.30%**	

Table 6: Comparison of baseline and guided performance across four metrics for MVTec dataset categories with EfficientAD and FastFlow. Δ is the median improvement percentage of the guided result from baseline.

Vision		Time-series	
Baseline	Guided	Baseline	Guided
126.50	236.37	10.86	12.49

Table 7: Comparison of baseline and guided generation median runtimes (seconds).

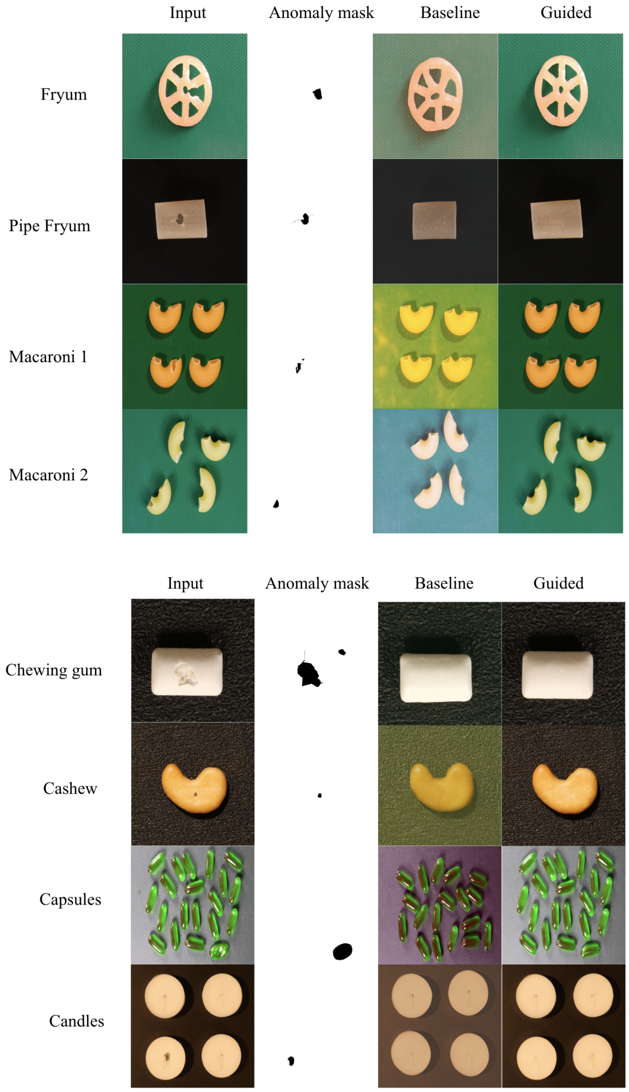


Figure 8: More VisA examples. With AR-Pro, we have anomaly repairs resemble inputs better, compared with baseline.

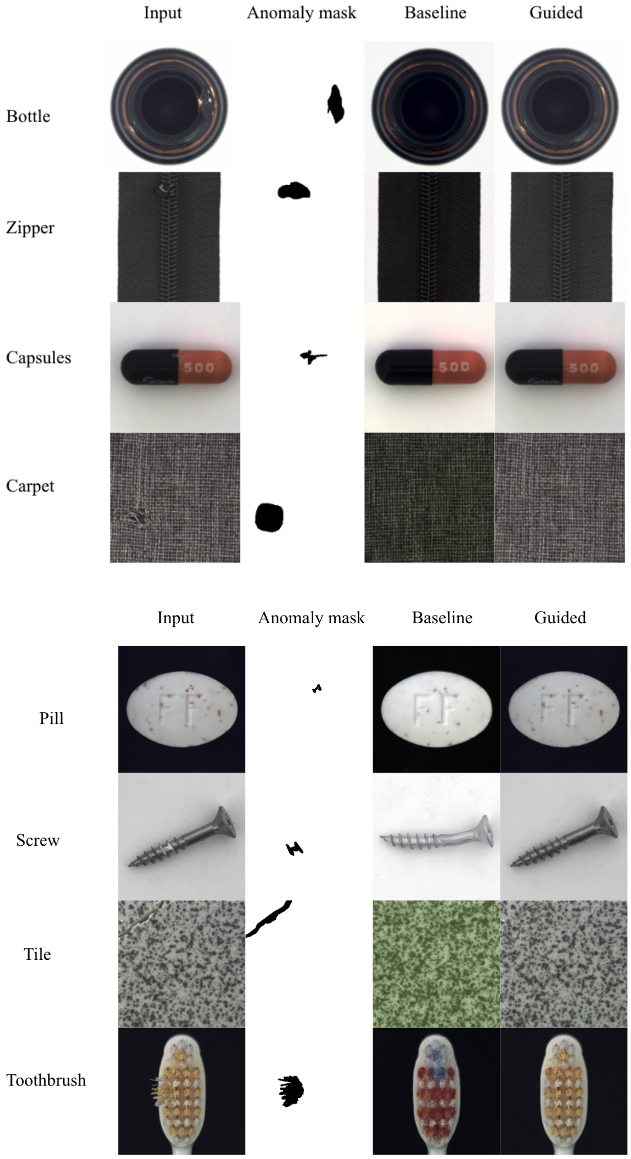


Figure 9: More MVTec examples (Part 1). With AR-Pro, anomaly repairs resemble inputs better compared with the baseline.

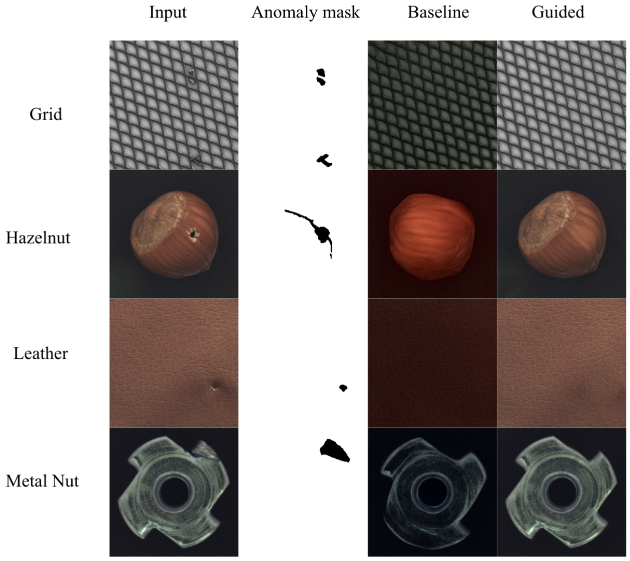
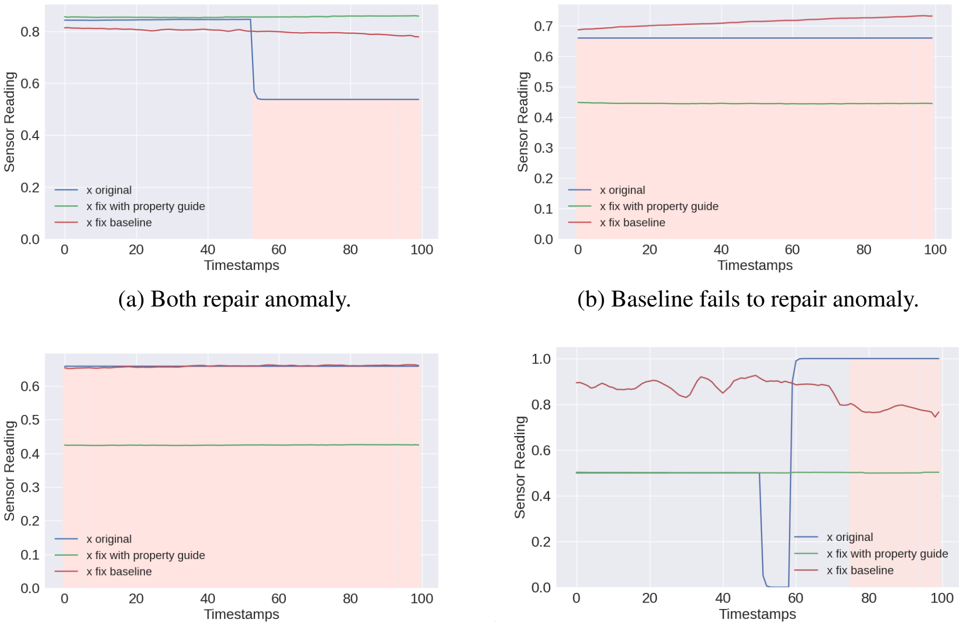


Figure 10: More MVTec examples (Part 2). With AR-Pro, anomaly repairs resemble inputs better compared with the baseline.



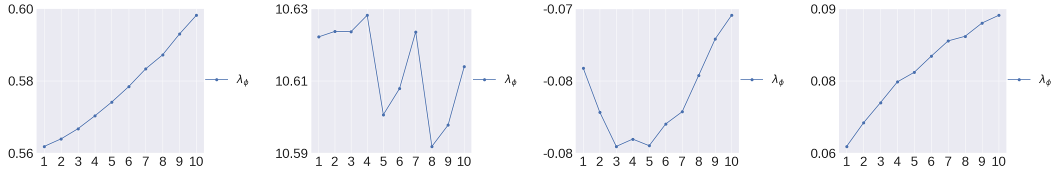
(a) Both repair anomaly.

(b) Baseline fails to repair anomaly.

(c) Baseline fails to repair anomaly.

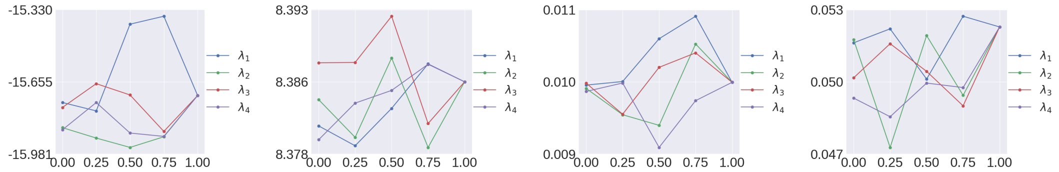
(d) Baseline fails to repair anomaly.

Figure 11: The first image shows an instance where the baseline repairs the anomaly, but not as effectively as with property guidance. The last three images show instances where the baseline fails to repair the anomaly.



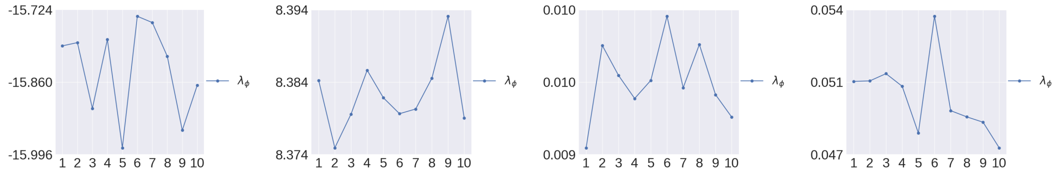
(a) M_s (b) M_d (c) M_ω (d) $M_{\overline{\varpi}}$

Figure 12: Effects of hyperparameter λ_ϕ on M_s, M_d, M_ω and $M_{\overline{\varpi}}$ on SWaT; the effect of λ_ϕ varies across metrics, but the range remain relatively small.



(a) M_s (b) M_d (c) M_ω (d) $M_{\overline{\varpi}}$

Figure 13: Effects of hyperparameter on M_s, M_d, M_ω and $M_{\overline{\varpi}}$ on VisA cashew class; the effect of λ_ϕ varies across metrics, but the range remain relatively small.



(a) M_s (b) M_d (c) M_ω (d) $M_{\overline{\varpi}}$

Figure 14: Effects of hyperparameter λ_ϕ on M_s, M_d, M_ω and $M_{\overline{\varpi}}$ on VisA cashew class; the effect of λ_ϕ varies across metrics, but the range remain relatively small.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope, as they clearly outline the development of a domain-independent framework for anomaly repair based on formal properties and the demonstration of its effectiveness through empirical results on VisA and SWaT datasets. This alignment ensures that the reader's expectations are met and that the paper delivers on its proposed objectives.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitation in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Although we did not have proofs, we listed the assumptions in Section 2 and Section 3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We attach the code for reproducing the result to the zip file.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Please find the appendix for the implementation details.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include the experiment details in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We reported the mean and standard deviation for our main results in Table 1. We also report TNR with 95% statistical significance in Table 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We indicate GPU details in Appendix A and the amount of compute required for each of the individual experimental runs as well as estimate the total compute in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no noticeable negative societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We call for users adhere to usage guidelines of GPT2 in Appendix B or implementing safety filters.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the original paper that produced the code packages and datasets. We also state which version of the model is used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We communicate the details of the code as part of our submissions in the zip file. This includes details about training, license, limitations, etc.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.