POMELO: Black-Box Feature Attribution with Full-Input, In-Distribution Perturbations

Luan Ademi, Maximilian Noppel(✉), and Christian Wressnegger

KASTEL Security Research Labs, Karlsruhe Institute of Technology, Karlsruhe, Germany

noppel@kit.edu

Abstract. Model-agnostic explanation methods provide importance scores per feature by analyzing a model's responses to perturbed versions of the sample to be explained. The explanation's quality therefore hinges on the made perturbations and, most importantly, suffers if these lead to out-of-distribution samples. Unfortunately, this is the case for the popular LIME explanation method. In this paper, we thus introduce POMELO, an extension to LIME leveraging generative AI for full-input, in-distribution sampling. We define key properties of such samplers: distribution alignment, diversity, and locality. Based on these, we discuss different neural samplers based on normalizing flows and diffusion models. Our results demonstrate that neural samplers outperform traditional perturbation strategies and yield explanations that are better aligned with human intuition. Supplementary material to our paper is available at https://intellisec.de/research/pomelo.

Keywords: LIME · Normalizing Flows · Diffusion Models

1 Introduction

In recent years, the community has proposed a plethora of techniques for explaining machine learning models [4,12,13,20,28,35,40,41,46,53]. One of the most popular and most widely used methods in practice [19] is LIME [35]. As a post-hoc, black-box explanation method LIME operates without access to the model's weights, gradients, or its neurons' activation. Instead, it generates explanations using only the model's input-output behavior.

To do so, LIME queries the model's soft-label probabilities for perturbed variants of the sample to be explained. Based on these soft-labels and the binary masks representing the made perturbations, LIME trains an interpretable surrogate model. Each perturbed sample "misses" certain segments of the original sample, that is, segments are replaced with a baseline, e.g., a constant value like gray or black, a blurred patch [11], or the mean value of the segment (cf. Fig. 1).

L. Ademi and M. Noppel—Both authors contributed equally to this research.

© The Author(s) 2026

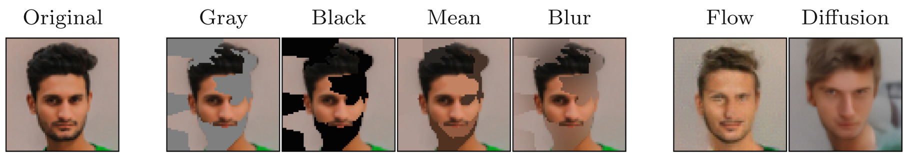


Fig. 1. Perturbed samples generated using different perturbation strategies.

Unfortunately, these strategies give rise to a fundamental limitation: the perturbations are out-of-distribution (o.o.d.), and thus, explanations may capture undefined and/or irrelevant model behavior [16,17,33].

To address this issue, we revise the perturbation strategies of LIME and propose our extension POMELO that can be thought of as a full-input variant of LIME. It leverages generative models to generate perturbed but in-distribution (i.d.) versions for an original sample. In contrast to related work, we allow changes on the full input, instead of random segments only. Our method thus grounds explanations on realistic changes, induced by the training distribution, and produces more interpretable insights into the model's decision-making process. As an example of our new, *full-input* perturbation strategies, Fig. 1 depicts perturbed samples from a normalizing flow and a diffusion model (cf. Sect. 2.4) on the right. The use of full-input perturbations has one crucial advantage over related work: the generative model "knows" the feature correlations and perturbs correlated feature together, independent of their spatial position in the image. With segment-wise replacement, as used in LIME, this is not possible. The probability that larger features or distributed correlations are replaced together is low if the segments are picked at random.

We are the first to show how generative AI with *full-input perturbations* can be used within the LIME-framework for the complex *image domain*. However, other extensions of LIME with similar motivations exist, though: For instance, Qiu et al. [33] use naive perturbation strategies and weighs the perturbed samples according to an inlier score when training the surrogate model. Others use in-painting strategies to fill in the "missing" segments in images [2,6], or for anomaly detection on tabular data [48]. Furthermore, one can use Variational Autoencoders (VAEs) to explain temperature time series forecasting in blast furnace [39] or Conditional Tabular GANs [52] to enhance LIME's robustness against attacks [38].

In summary, we make the following contributions:

- **Key Properties of Samplers.** We develop a framework that consists of three key properties of effective full-input samplers: *distribution alignment*, *diversity* and *locality*. We discuss the trade-offs and relations between the properties and propose metrics to assess different aspects of them.
- **Full-Input Perturbation Strategies for LIME.** We demonstrate how full-input perturbation strategies can be integrated into LIME, giving rise to our novel approach POMELO. Therefore, we examine two concrete strategies,

one based on normalizing flows and one based on denoising diffusion implicit models (DDIMs). Based on the key properties, we compare the resulting perturbed samples against the traditional LIME perturbation strategies.

– **Comprehensive Evaluation.** We compare the explanation performance of our approach against traditional LIME in terms of the explanation quality, diversity, locality, distribution alignment, and computational feasibility. We demonstrate that the descriptive accuracy metric is closely aligned with the used perturbation strategy in the explanation.

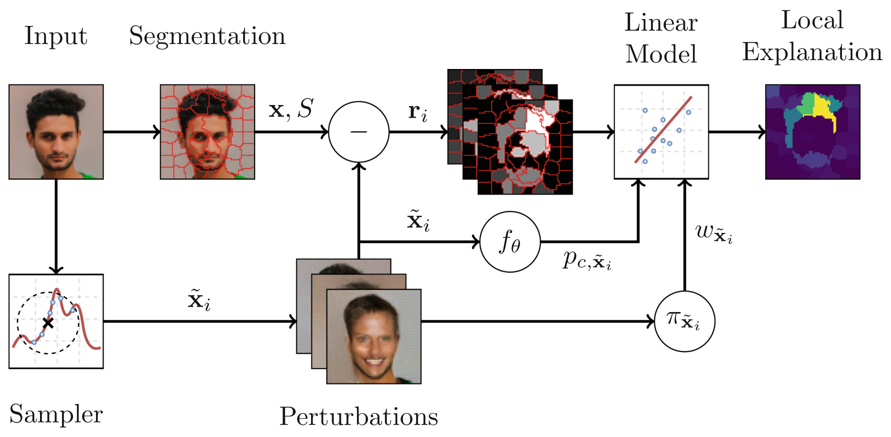


Fig. 2. Overview depiction of POMELO. We replace LIME's perturbation strategies with an in-distribution *full-input* sampler and employ a more concise interpretable representation based on the segment-wise ℓ_1 distance between the original sample and the perturbed samples.

2 POMELO

We extend the LIME explanation method as a remedy to the out-of-distribution (o.o.d.) problem. Our extension, POMELO, replaces the naive perturbation strategies with more powerful generative approaches, denoted as *neural samplers*. This change comes with peculiarities if the perturbations are not limited to specific segments, a problem we describe and solve in this section.

After introducing our basic notation, we first state the o.o.d. problem in Sect. 2.1, and its relation to meaningfulness in Sect. 2.2. In Sect. 2.3, we formalize three key properties a sampler should satisfy. Next, we describe three concrete perturbation strategies that adhere to these criteria in Sect. 2.4. We regard them as proof-of-concepts to demonstrate the benefits of our extension. In Sect. 2.5, we introduce our core contribution: how to generate interpretable representations from full-input perturbations. And lastly, Sect. 2.6 describes how we build explanations from all the above. In Fig. 2, we provide an overview on the methodology of POMELO.

Notation. Throughout the paper, we consider a classifier $\mathcal{F}_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, that maps samples $\mathbf{x} \in \mathcal{X}$ to classes/labels $y \in \mathcal{Y}$ based on a soft-label $\mathcal{F}_\theta(\mathbf{x}) = \arg\max_c f_\theta(\mathbf{x})_c$. The model's parameters θ are learned on a training dataset $\mathcal{D}_{train}$ and eventually are validated on a separate dataset $\mathcal{D}_{val}$. A sampler $\mathcal{S} : \mathcal{X} \rightarrow P(\mathcal{X})$, where P is the powerset, produces a set of perturbed samples, and a segmentation algorithm Seg provides a segmentation, given a sample.

2.1 The Out-of-Distribution Problem

LIME employs relatively simple perturbation strategies, such as overwriting segments/super-pixels with constant values or blurring a segment. These perturbations are supposed to mimic the removal of the respective segment but also diverge heavily from the real-world distribution. Hence, LIME explanations are based on o.o.d. samples, not reflecting the model's real-world decision-making process in benign environments. A model's responses to black or gray patches simply express exactly this; how does the model react to black or gray patches. We argue, that explaining based on this results in misleading explanations [33]. In adversarial environments, in turn, the fact that LIME uses o.o.d. samples can even be exploited to bootstrap attacks on the explanation [15,42]. Training neural networks to make correct predictions on all samples remains a tricky problem [16,17,29,47], and thus, LIME explanations do not faithfully explain a model's decisions.

2.2 Relating In-Distribution Perturbations and Meaningfulness

Grounding explanations in the model's data distribution produces more reliable and interpretable insights. However, it is important to note, that i.d. perturbed samples are not always semantically meaningful variations of the original sample. Depending on the data distribution also meaningless but dominant features like copyright tags on horse images [3,23] or tags on images of skin cancer [36] are in-distribution, i.e., neural samplers might produce such features. Fortunately, the pure existence of such features does not influence the explanation much. Only, if they are (spuriously) correlated with the model's soft-labels, the explanation will pick them up, which is a desirable property. In other words, *POMELO explains the model* and *the underlying data distribution together.*

2.3 Key Properties of Perturbation Strategies

To achieve i.d. perturbed samples, we employ *samplers*. Given a sample $\mathbf{x}$, a sampler $\mathcal{S} : \mathcal{X} \rightarrow P(\mathcal{X})$ generates a set of perturbed samples. The perturbed samples should satisfy three key properties to ensure robust and meaningful explanations, which we discuss here. Later, in Sect. 4, we present metrics to measure to what extent the properties are achieved by our investigated samplers.

- **Distribution Alignment** (DA) **.** Given an i.d. sample $\mathbf{x}$, the perturbed samples $\tilde{\mathbf{x}}_i$ should also be i.d.. Formally, a sampler should satisfy $\mathbf{x} \sim \mathfrak{D} \Rightarrow \mathcal{S}(\mathbf{x}) =: \tilde{\mathbf{x}} \sim \mathfrak{D}$, where $\mathfrak{D}$ denotes the true distribution in deployment. For o.o.d. samples the samplers might generate o.o.d. perturbations. This scenario equals explaining o.o.d. samples and is left as future work, though.
- **Diversity** (D) **.** The set of perturbed samples should be diverse, exploring different classes and many facets of the decision surface. Explanations based on biased samples lead to a skewed view of the model's decision-making process, possibly missing influential directions. Therefore, the perturbed samples should spread in diverse directions and explore all nearby decision boundaries and the shape of the soft-label surface.
- **Locality** (L) **.** In addition to diversity, perturbed samples should be in the neighborhood of the original sample. Only so do they remain relevant to the explained prediction and preserve the semantic of a local explanation method.

Trade-Offs between the Key Properties. The three properties above restrict each other. Locality and distribution alignment can theoretically be satisfied simultaneously, e.g., via returning very close perturbed samples or even the orginal sample itself. In practice, distributions are seldom so spiky that insufficient variations can be found nearby. In other words, a very spiky distribution would be a distribution where one sample is absolutely in-distribution but moving it just an insignificant bit in any direction makes it out-of-distribution. Diversity, in turn, often requires larger changes and conflicts with locality. Depending on the shape of the distribution, the diversity could also conflict with the distribution alignment, e.g., if the distribution is so narrow (almost a line) that the sampler can only perturb in two directions, not allowing much diversity. Therefore, the sampler (or its parametrization) must balance the three properties.

While challenging to quantify, we propose to determine the parametrization by balancing locality with diversity. We provide details on this expensive process in Appendix B. The trade-offs with distribution alignment, on the other hand, are of theoretical nature, and depend more on the distribution than on the parametrization of the sampler.

2.4 Generating In-Distribution Samples

We present two full-input samplers based on normalizing flows [8,9,21,34] and diffusion models [18,43,45]. For comparison, we describe a third sampler utilizing diffusion-based in-painting similar to RePaint [27]. The training of each sampler requires access to the training dataset of the model being analyzed, or, at least, to a similar dataset with a comparable distribution. Note that this is an additional requirement compared to the naive perturbation strategies of LIME.

Normalizing Flows. Normalizing flows [8,9,21,34] are powerful density estimators with exact invertibility. Flows are composed out of a sequence of K *parameterized* bijective functions $flow = f_K \circ \cdots \circ f_1$. This sequence transforms a simple base distribution $p_{\mathcal{Z}}$ (often Gaussian) into a complex distribution $p_{\mathcal{X}}$, e.g., the distribution of realistic images of celebrities. The parameters of the flow are learned by minimizing the Kullback-Leibler (KL) divergence between the current distribution of images $p_{\mathcal{X}}$ and the desired distribution p_{train}, provided by the training data. Building on this foundation, architectures like Glow [21] achieve competitive results in image generation on datasets such as CelebA [24], but also in other applications like lattice field theory [31]. The core benefit of flows is their exact invertibility, which enables downstream tasks like interpolation and semantic manipulation, making flows a perfect candidate for POMELO.

By leveraging the latent space of a normalizing flow, we obtain the latent representation of $\mathbf{x} \in \mathcal{X}$ as $\mathbf{z} := flow^{-1}(\mathbf{x}) \in \mathcal{Z}$. This representation $\mathbf{z}$ is then perturbed to a new point $\tilde{\mathbf{z}}$ and mapped back to the input space of images $\tilde{\mathbf{x}} := flow(\tilde{\mathbf{z}})$. The perturbation in latent space equals an interpolation between $\mathbf{z}$ and a randomly sampled point $\mathbf{z}^* \sim p_{\mathcal{Z}}$ from the base distribution. Formally,

$$\tilde{\mathbf{z}} = (1 - \lambda) \odot \mathbf{z} + \lambda \odot \mathbf{z}^* \text{ with } \lambda \sim \mathcal{N}_{dim(\mathcal{Z})}(\mu, \sigma) \ ,$$

where μ and σ control the locality of the perturbed samples. In Appendix B, we provide further details on why we set μ to 0.5 and σ to 0.3. While picking the interpolation factors λ at random is a novel strategy, related work already suggests that interpolations in $\mathcal{Z}$ produce meaningful changes for CelebA [21].

Denoising Diffusion Implicit Models (DDIMs). Diffusion-based models show exceptional performance in image generation and data augmentation [7,18] and, therefore, are a second promising candidate for our aim. More specifically, we use denoising diffusion implicit models [44]. Compared to earlier denoising diffusion probabilistic models (DDPMs) [18], these implicit models provide fundamental benefits for our case.

But one step back. From a functional point of view, diffusion models generate realistic images from pure Gaussian noise. This capability is learned through reconstructing a sample $\mathbf{x}_0$, from a progressively corrupted version $\mathbf{x}_T$. By iteratively predicting and removing the noise over multiple time steps t, the model gradually refines its output, effectively reversing the corruption process. These two involved processes are the forward process ($\mathbf{x}_0 \rightarrow \mathbf{x}_T$) that introduces noise, and the reverse process ($\mathbf{x}_T \rightarrow \mathbf{x}_0$) that denoises the corrupted sample. In denoising diffusion implicit models (DDIMs), the reverse process can be *deterministic*, allowing for an accompanying deterministic forward process (cf. Appendix A). This configuration allows us to encode a sample as a noisy representation and reconstructing it with minimal error [7,44].

Our proposed sampling strategy exploits this advantage of DDIMs. It uses the deterministic forward process, eventually reaching a noisy but revertible representation of the sample. Afterward, it follows a slightly probabilistic reverse process that introduces deviations, eventually resulting in different perturbed

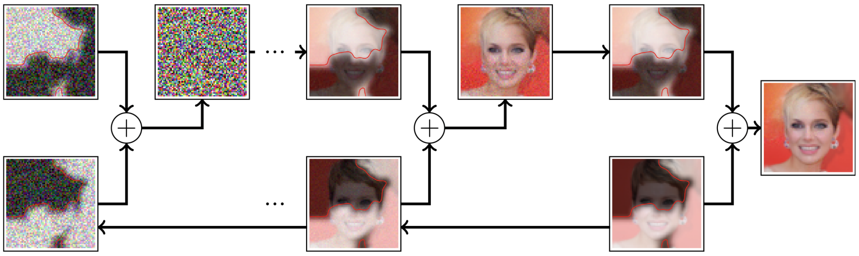


Fig. 3. Our in-painting methodology. In a multi-step process we combine the parts for the noisy original image for present patches(forward process from left to right) and pure noise for missing areas (reverse process, from left to right).

samples of the original sample. The involved randomness can be controlled with a noise scaling factor η, where $\eta = 1$ results in the DDIM reverse process equaling the probabilistic DDPM reverse process, and $\eta = 0$ yields in a fully deterministic reverse process.

Intuitively our approach generates perturbed samples by adding deterministic noise, resulting in a noisy version of the original sample that can be exactly reconstructed by the reverse process. However, by introducing noise in the reconstruction, we deviate from the original sample, creating a perturbed sample that is conditioned on the original sample.

In-Painting. An inherent "problem" of the aforementioned perturbation strategies is their tendency to change the entire image. *This "problem" is exactly what POMELO solves.* As a reference for our experiments we use diffusion-based in-painting as a localized but also neural alternative with changes in specific segments only. In-painting is performed by iteratively merging noisy versions of the original sample $\mathbf{x}_t^{\text{orig}}$ into a diffusion process of an initially random masked intermediate $\mathbf{x}_t^{\text{mask}}$. A schematic representation of this process is provided in Fig. 3. At each step, the merged sample $\mathbf{x}_t$ is calculated as follows:

$$\mathbf{x}_t = M \odot \mathbf{x}_t^{\text{orig}} + (1 - M) \odot \mathbf{x}_t^{\text{mask}},$$

where the in-painted sample $\mathbf{x}_t^{\text{mask}}$ is derived using a denoising step of the previous merged sample $\mathbf{x}_{t-1}$ and the noisy sample $\mathbf{x}_t^{\text{orig}}$ is generated by applying the forward process of the DDIM model on the original sample. For the mask M, we use the segmentation from LIME and create a random mask by selecting a subset of the segments, equivalent to LIME's methodology. To ensure smoother transitions, we apply a Gaussian blur to the mask, resulting in soft edges around the in-painted regions. Similarly to the DDIM-based strategy, the noise level can be controlled by the η parameter and the number of reverse steps.

This approach is closely resembles RePaint [27], but without the iterative resampling process. We omitted resampling because it is to computationally expensive when generating hundreds of images per explanation. While resampling could enhance the semantic consistency of perturbations, we are unable to perform an extensive evaluation with this method.

2.5 Interpretable Representations

In contrast to LIME, POMELO's full-input perturbation strategies produce global changes, unrestricted by the segmentation. Each segment is perturbed to some degree and the binary interpretable representation of LIME cannot be applied (removed/not removed). To maintain a low-dimensional interpretable feature space for the surrogate model, we quantify the full-input changes in a per-segment manner. More concretely, we use a *real-value interpretable space* based on a distance metric $D_{\mathcal{X}}$ in the input space $\mathcal{X}$. Given the segmentation of the original sample $S = \mathrm{Seg}(\mathbf{x})$, we calculate the interpretable representation $\mathbf{r} \in [0,1]^{|S|}$ of a given perturbed sample $\tilde{\mathbf{x}}$ as follows: For each segment $\mathbf{s}_i \in S$ we assign a score:

$$r_i = 1 - \frac{D_{\mathcal{X}}(\mathbf{x}_{[s_i]}, \tilde{\mathbf{x}}_{[s_i]})}{\max_{\mathbf{v}} D_{\mathcal{X}}(\mathbf{x}_{[s_i],\mathbf{v}})} , \tag{1}$$

where $\mathbf{x}_{[\mathbf{s}_i]}$ and $\tilde{\mathbf{x}}_{[\mathbf{s}_i]}$ are the pixel vectors of the segment $\mathbf{s}_i$ of the original and the perturbed sample, respectively. For our experiments, we employ the ℓ_1-norm as the distance metric $D_{\mathcal{X}}$. We denote this process of generating an interpretable representation from the original sample $\mathbf{x}$, a perturbed sample $\tilde{\mathbf{x}}$, and the segmentation S with a minus sign in Fig. 2.

2.6 Generation of Explanations

Based on the interpretable representations we craft an explanation via the LIME methodology. Given sample $\mathbf{x}$, its segmentation S, and a model θ, we create a set of perturbed samples and their corresponding representations, $\{(\tilde{\mathbf{x}}_1, \mathbf{r}_1), \ldots, (\tilde{\mathbf{x}}_n, \mathbf{r}_n)\}$. The perturbed samples are weighted using a kernel $\pi(\tilde{\mathbf{x}}_i, \mathbf{x})$, i.e., the cosine similarity between $\tilde{\mathbf{x}}_i$ and $\mathbf{x}$. Next, we collect the soft-labels $f_\theta(\tilde{\mathbf{x}}_i)_c$ of the winning class $c = \mathcal{F}_\theta(\mathbf{x})$ for each perturbed sample. This process results in a dataset consisting of $(\pi(\tilde{\mathbf{x}}_i, \mathbf{x}), \tilde{\mathbf{x}}_i, \mathbf{r}_i, f_\theta(\tilde{\mathbf{x}}_i)_c)$ tuples, which are then used to train a linear surrogate model as LIME does, but on the $[0,1]^{|S|}$ space instead of the binary $\{0,1\}^{|S|}$ space. The resulting surrogate model approximates the original model in a local neighborhood, and its learned coefficients represent the contribution of each feature of $\mathbf{r}$ (each segment) to the decision process.

3 Metrics

This section introduces the metrics to evaluate the perturbed samples' distribution alignment (Sect. 3.1), diversity (Sect. 3.2), and locality (Sect. 3.3). Thereafter follow the metrics for the explanation quality, which are reasonability (Sect. 3.4), fidelity (Sect. 3.5), and stability (Sect. 3.6).

3.1 Measuring Distribution Alignment DA

To evaluate distribution alignment, we embed perturbed samples in the penultimate layer of a pretrained model and apply two o.o.d. detection techniques:

- **Mahalanobis Score.** For each validation sample we generate n perturbed samples, which are embedded using the penultimate layer of a ResNet18 model trained on the CelebA dataset. We compute the Mahalanobis uncertainty scores $\mathcal{U}_{MD}$ over these embeddings, following Lee et al. [25]. The required class-conditional Gaussian distributions are estimated using 30,000 training samples from the CelebA dataset. As references for clearly o.o.d. samples (compared to CelebA) we use CIFAR-10 images [22] and Gaussian noise embedded in the same model. We omit to transform the uncertainty scores into binary decisions (i.d./o.o.d.), and report the raw scores instead.
- **UMAP.** To gain qualitative insights, we project the above embeddings into a 2D space using uniform manifold approximation and projection (UMAP) [30], allowing for visual assessment of their alignment with the original distribution. Following Rousseeuw [37], we use the silhouette score as a quantitative measure.

3.2 Measuring Diversity D

We measure the different aspects of diversity with the two entropy-based metrics.

- **Shannon Entropy.** First, we asses the hard-label diversity via the Shannon entropy of the distribution of winning classes induced by the perturbed samples. A good sampler would produce a uniform distribution, ensuring that the perturbed samples are diverse and not biased toward one class. Formally, we measure E_{hard} as $-\sum_{c \in C} p(c) \cdot \log p(c)$, where $p(c)$ is the probability that a perturbed sample is predicted as class c.
- **Differential Shannon Entropy.** In addition, we measure the differential Shannon entropy of the soft-labels. The reasoning behind this is that the interpretable model is trained on the soft-labels of the predicted class instead of the hard-labels. Theorically, the differential entropy of the soft-labels is defined as $E_{soft} := -\int_0^1 \mathrm{pdf}(s) \log \mathrm{pdf}(s)\ ds$, where $\mathrm{pdf}(s)$ is the probability density function of the soft-labels. We approximate this formula with 20 bins via the trapezoidal rule for numerical integration from `scikit-learn`.

3.3 Measuring Locality L

The locality of the perturbed samples around the original sample is measured with two metrics: the ℓ_2 distance and the structural similarity index (SSIM) [49]. *Note that the second measure is specific for the image domain. One might use any application-specific measures of locality for the domain at hand.*

3.4 Measuring Reasonability R

We assess the reasonability of the generated explanations by comparing the relevant regions identified by the explanations with ground truth annotation masks from the CelebAMask-HQ dataset [24]. Therefore, we first scale the explanations to the $[0, 1]$ interval. Then we transform each 0-to-1-scaled explanation $\mathbf{r}$ into a binary mask of relevant and irrelevant pixels based on a threshold τ via $\mathbf{r}^{\tau}_{bin} := \mathbf{r} > \tau$. Based on the binary mask $\mathbf{r}^{\tau}_{bin}$ and the binary annotation mask $\mathbf{a}$ we define the intersection size as $IS^{\tau} := ||\mathbf{r}^{\tau}_{bin} \wedge \mathbf{a}|\,|_1$ and construct three metrics:

– **Intersection over Union.** First, we measure the Intersection over Union as $IoU^{\tau} := \frac{IS^{\tau}}{||\mathbf{r}^{\tau}_{bin} \vee \mathbf{a}||_1}$, and the area under the curve as $\overline{IoU} := \int_0^1 IoU^{\tau} d\tau$.
– **Explanation Mask Recall.** Next, we compute the ratio of annotation pixels covered with relevant pixels and denote this metric as explanation mask recall and define it as $EMR^{\tau} := \frac{IS^{d}\tau}{||\mathbf{a}||_1}$ and $\overline{EMR} := \int_0^1 EMR^{\tau} d\tau$.
– **Explanation Mask Precision.** Lastly, we measure the explanation mask precision as $EMP^{\tau} := \frac{IS^{\tau}}{||\mathbf{r}^{\tau}_{bin}||_1}$, and $\overline{EMP}$ analog as the area under the curve.

3.5 Measuring Fidelity F

We assess the fidelity via the descriptive accuracy [50], originally termed as pixel flipping [4], and later as the deletion/insertion game [32] or very recently as most-influential first (MIF) [5]. In this work, we use the terms deletion and insertion game. In the deletion game we start with the original image and "remove" pixels in the order of descending importance scores in the explanation. In the insertion game, on the other hand, we start with a baseline image and add pixels from the original image in the same order. When multiple pixels share the same importance score, their deletion or insertion order is determined randomly. As LIME produces homogeneous importance scores within one segment this happens regularly. To compensate for this randomness we evaluate each explanation 10 times. At each insertion or deletion step the soft-label of the original prediction is recorded, leading to *deletion* and *insertion* graphs, respectively. Based on these graphs we calculate the bounded area under curve (AUC) at 50 % flipped pixels as $AUC^{50\%}_{del}$ and $AUC^{50\%}_{ins}$, respectively. For the calculation we apply the trapezoidal rule for numerical integration from `scikit-learn` and flip 1 pixel per step. Because of the computational effort involved in the in-painting baseline we flip 128 pixels at a time here.

3.6 Measuring Stability S

The stability of explanations assesses how consistent the explanations are in consecutive runs. Therefore, $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m\}$ represents a set of m explanations extracted for a given sample. We compute the mean explanation, denoted as $\bar{\mathbf{r}}$ and then measure the mean distance from $\bar{\mathbf{r}}$:

$$\text{Stability} = \frac{1}{n}\sum_{i=1}^{n} D_{\mathcal{E}}(\mathbf{r}_i, \bar{\mathbf{r}}), \tag{2}$$

where $D_{\mathcal{E}}(\mathbf{r}_i, \bar{\mathbf{r}})$ is a distance metric in the explanation space, e.g., in our experiments it is the ℓ_2-distance.

4 Evaluation

In this section, we first present our experimental setup in Sect. 4.1. The following subsections contain our results for the sampler property distribution alignment (Sect. 4.2) and the diversity-locality trade-off (Sect. 4.3). Thereafter follow the explanation quality metrics reasonablility (Sect. 4.4), fidelity (Sect. 4.5), and stability (Sect. 4.6).

4.1 Experimental Setup

We evaluated all samplers on the CelebA dataset [26], which contains portrait images of celebrities. For the reasonability metric, we used the CelebAMask-HQ dataset [24], which provides annotation masks for a subset of CelebA. We focus on hair color classification and reduce the classes to *blond*, *black*, *gray*, *brown*, and *bald*, and ignoring samples with ambiguous classification[1] [51].

Classification Models. We generate explanations for a ResNet18 [14] classifier trained to predict the hair color by means of the `torchvision` library. The images are scaled to 64×64 pixel before training with a batch size of 4,096 and a learning rate of 0.001 for 12 epochs. The model achieves 91 % validation accuracy which is sufficient for our purpose of evaluating an explanation method.

Parametrization of LIME. We configure the LIME algorithm according to its original implementation, using a cosine similarity kernel with a width of 0.25, and $n = 800$ perturbations per sample. The segmentation is generated with simple linear iterative clustering (SLIC) [1] set to 65 segments, which is suitable for 64×64 images.

Generative Models. We use the pretrained Generative FLOW (GLOW) model by Dombrowski et al. [10], the implementation from the `diffusers` library, and the author's public GitHub repository. In addition, we train a DDIM model using the code provided by the `diffusers` library on the CelebA training data.

[1] Some samples in the CelebA dataset have two hair colors assigned. These samples are ignored in our study.

Table 1. Quantitative results of neural samplers (top) and naive samplers (bottom), evaluated on their distribution alignment DA, diversity D, and locality L. Results are presented as the mean and standard deviations, where applicable.

Sampler	DA		D		L	
	D_M (↓)	Sil. (↑)	E_{hard} (↑)	E_{soft} (↑)	ℓ_2 (↓)	SSIM (↑)
Flow	**398.32**	0.013	$1.07_{\pm 0.43}$	$2.73_{\pm 0.64}$	$17.13_{\pm 4.38}$	$0.655_{\pm 0.064}$
Diffusion	415.33	**0.011**	$\mathbf{1.48}_{\pm 0.34}$	$2.56_{\pm 0.56}$	$17.88_{\pm 4.13}$	$0.552_{\pm 0.091}$
In-painting	418.80	0.031	$1.08_{\pm 0.48}$	$2.55_{\pm 0.74}$	$19.14_{\pm 5.39}$	$0.578_{\pm 0.081}$
Black-Out	717.17	0.578	$0.44_{\pm 0.32}$	$1.21_{\pm 0.92}$	$39.36_{\pm 9.58}$	$0.402_{\pm 0.084}$
Gray-Out	611.63	0.216	$1.20_{\pm 0.52}$	$\mathbf{2.95}_{\pm 0.87}$	$22.40_{\pm 4.29}$	$0.564_{\pm 0.071}$
Mean	423.31	0.019	$0.42_{\pm 0.48}$	$1.92_{\pm 1.49}$	$\mathbf{6.80}_{\pm 1.81}$	$\mathbf{0.792}_{\pm 0.048}$
Blur	496.39	0.055	$0.89_{\pm 0.53}$	$2.61_{\pm 1.06}$	$13.02_{\pm 3.00}$	$0.644_{\pm 0.065}$

Experiment Design. We analyze explanations for the winning class on 5,000 randomly selected samples from the CelebA dataset. Optimization techniques, such as reduced floating-point precision and graph compilation, minimize the computational overhead without compromising performance. The hyperparameters of all components of our work are displayed in Appendix B. If not mentioned differently, the number of perturbation n is 800 per sample and 5,000 randomly selected validation samples are evaluated due to resource restrictions. For some evaluations, the one or the other number is lowered because of computational restrictions. We always mention such limitations in the respective paragraph.

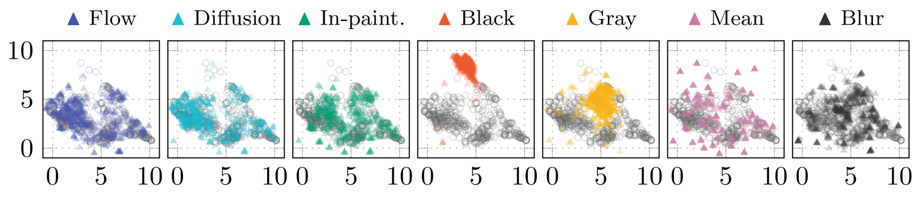


Fig. 4. *Distribution Alignment* DA. The 2D UMAP projections of 500 randomly selected perturbed samples and 500 training samples (gray circles) based on their feature embeddings.

4.2 Distribution Alignment DA

At the core of this work we aim to generate i.d. perturbed samples to overcome the o.o.d. problem in LIME. Here, we evaluate the distributional alignment of the perturbed samples via the Mahalanobis uncertainty scores [25] of 20 perturbed samples for each selected validation samples. We compare the scores to other unperturbed validation samples and two sets of clearly o.o.d. samples: First CIFAR-10 images [22] and, second, Gaussian noise. In addition, we visualize the feature space using UMAP [30] to gain qualitative insights.

According to our results in Table 1 the neural samplers achieve uncertainty scores on par with the validation samples, which reach 413. The flow sampler performs best (398), followed closely by the diffusion sampler (415) and the in-painting sampler (418). The perturbations of all three can be considered i.d. according to these scores. Expectedly, the mean sampler and blur sampler exhibit slighly worse but comparable performance with 423 and 496, respectively. The black-out sampler and the gray-out sampler, in turn, generate o.o.d. samples and achieve scores of 717 and 611, respectively. Even the clearly o.o.d. CIFAR-10 images achieve an score of 640. The Gaussian noise sets the referenz with 1,388.

In Fig. 4, we provide a 2D UMAP projection of the embedded perturbed samples. The figure, supports our above finding visually. The flow sampler, the diffusion sampler, and the in-painting sampler exhibit a close overlap with the validation samples while the mean sampler and blur sampler show a slight deviation only. The black-out sampler shows a clear separation, indicating a strong o.o.d. behavior. Similarly, the gray-out sampler exhibits a vastly different spread, being more clustered. These results confirm the ability of neural samplers to produce in-distribution perturbed samples, while traditional samplers induce a clear out-of-distribution behavior.

4.3 Diversity D and Locality L

Robust, meaningful, and local explanations require a careful balance between diversity and locality of the perturbed samples. Our evaluation reveals that neural samplers generally achieve this balance better than traditional samplers. The diffusion sampler achieves the highest diversity score of 1.48. This superior performance can be attributed to the ability to induce *different* class changes. Traditional samplers may also induce class changes, but in shattered regions of the decision surface or toward specific classes (cf. next paragraph). The in-painting sampler exhibits weaker diversity due to its focus on localized perturbations, hindering consistent full-input changes. Perturbations on the basis of segments may not expose interactions between segments, e.g., the probability of changing all hair segments at once is rather low. Among the traditional samplers, the gray-out sampler performs best. The black-out sampler performs poorly in both, the locality and the diversity property.

In Fig. 5, we depict the diversity-locality trade-off. The arrows represent the desired direction. While the mean samplers consistently excels in one property, it also fails in the other one. Neural samplers make a good trade-off together with the in-painting, blur, and gray-out sampler. The black-out sampler is worst.

Analysis of the Label Distributions. We analyze the label distribution induced by the perturbed samples and compare them to the label distribution of

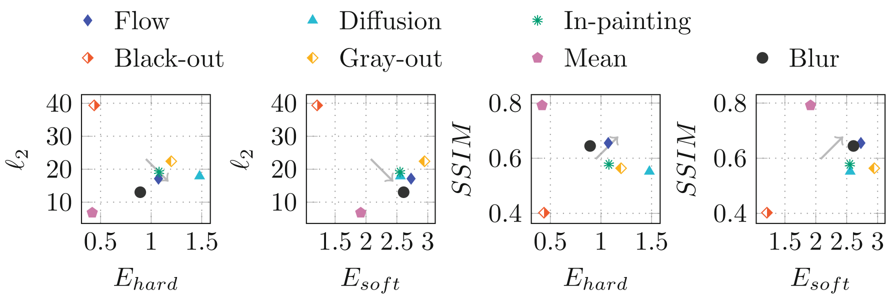


Fig. 5. The trade-offs between the diversity D and the locality L. The gray arrows indicate the desired direction.

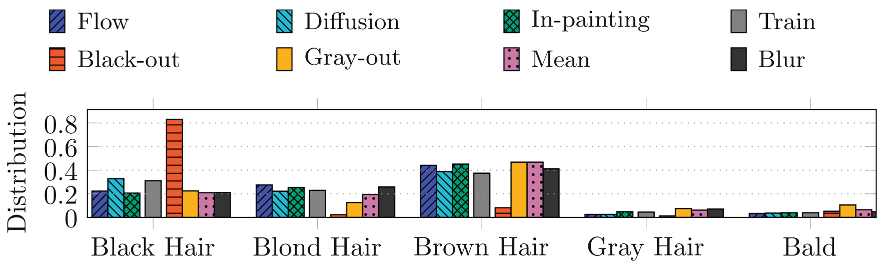


Fig. 6. The hard-label distributions of the perturbed samples as a proxy for diversity D. The black-out sampler shows a strong bias toward black hair. Neural samplers are depicted in left chunk, the training data in the middle, and naive samplers on the right.

the training data (cf. Fig. 6). In contrast to diversity, which examines variability in the perturbed samples at the individual instance level, we take a global perspective here by evaluating a random subset of all perturbed samples, uncovering broader patterns. Interestingly, the black-out sampler, in fact, has a bias toward the *black hair* class. This bias limits the representational effectiveness of the perturbations, as it skews them toward a specific label. Such biases can hinder a sampler's ability to provide meaningful explanations for certain classes, because it cannot generate sufficiently diverse or representative perturbed samples for all classes. This observation aligns with and explains the diversity and locality results of the black-out sampler.

4.4 Reasonability R

We assess the reasonability of our method POMELO on the basis of annotation masks from the CelebAMask-HQ dataset. The overlap between these human-provided annotations and the explanations indicates how well the explanations

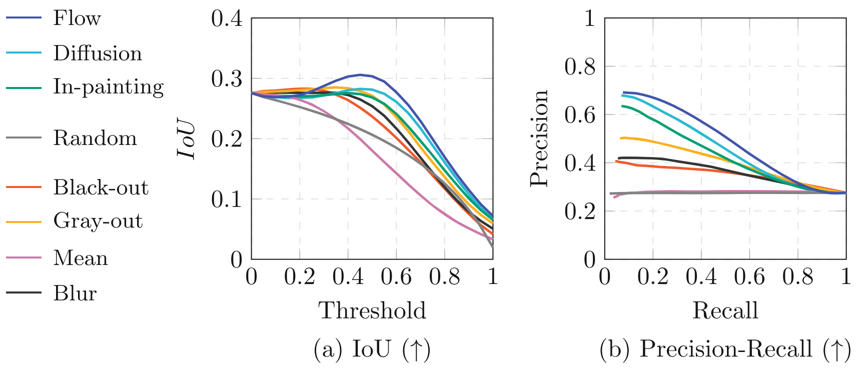
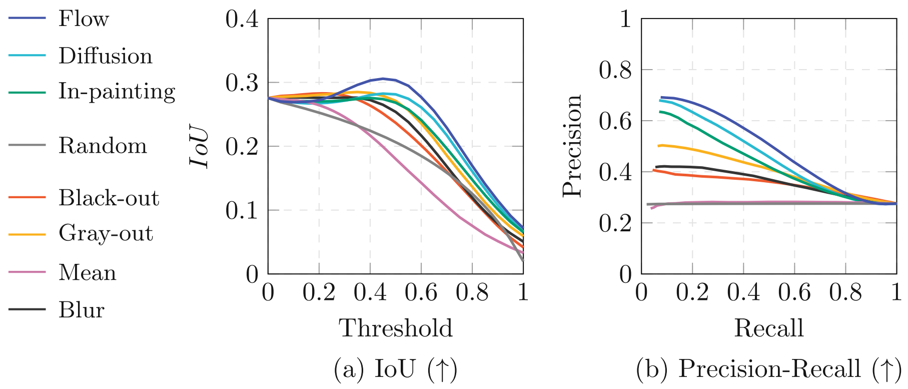


Fig. 7. Reasonability R results for all samplers and a baseline of random explanations. Find the intersection-over-union curve on the left (a), and the precision-recall curve on the right (b).

Table 2. Quantitative results of the explanation methods for properties reasonability R, stability S, and fidelity F. The fidelity is measured as the AUC with the in-painting replacement strategy for 50 % replaced pixels. Results are presented as the mean with standard deviations, where applicable.



align with the human understanding of the problem (Table 2). We measure this overlap with three metrics and display the results in Fig. 7. Explanations based on neural samplers, in particular based on the flow sampler, outperform traditional LIME explanations in all three metrics, indicating a stronger alignment with the human-provided masks. In particular, for the intersection over union and the precision POMELO excels.

This performance might stem from the fact that neural samplers modify dominant features, such as hair color, eye color, or facial structure, and that reasonable correlations dominate the training data. Hence, the perturbations are often consistent with the human-ontological understanding of the data.

Reasonability does not assess, however, how well the explanations capture a model's actual decision process (cf. Fidelity). For instance, a model might base its decisions on a-typical patterns and artifacts that are not captured by human-defined ontology but still are present in the training data, e.g., a copyright tag on many images of horses [3]. By training the neural sampler on the same training data, such patterns would also be captured and be highlighted at their spatial position. Based on the high alignment with the annotation masks and through the manual investigation of various explanation, we assume that no such dominate spurious correlations are present in CelebA.

To make this more clear; If every blond celebrity would have blue eyes, the neural sampler would only perturb both features together. But, the difference between the perturbed samples and the original sample would then also be at the spatial position of the eyes and the hair. Thus, both areas would be hightlighted as relevant. Our method therefore serves as a helpful debugging tool to find such spurios correlations with spatial constraints.

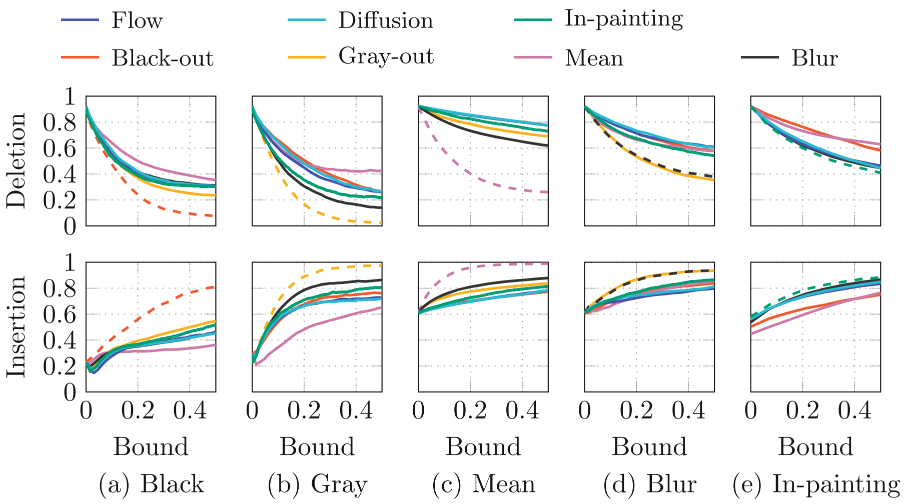


Fig. 8. Results for the deletion game (↓) in the top row and the insertion game (↑) in the bottom row. The fidelity F depends on the replacement strategy in the sense that samplers perform best when paired with their corresponding replacement strategy, highlighted as dashed lines. In the insertion game with an in-painting replacement strategy the starting point is uncertain as almost the whole image is generated by the in-painter.

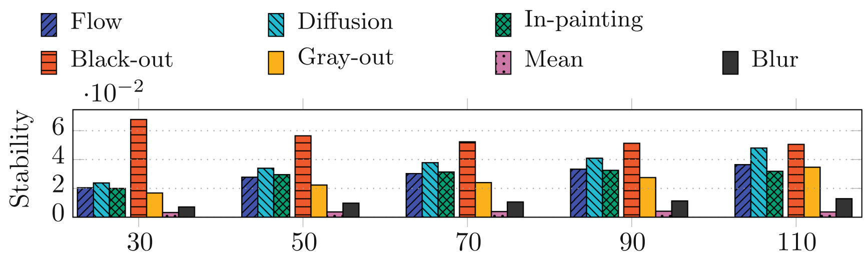


Fig. 9. The stability S (↓) evaluated for a varying number of segments between 30 and 110, with neural samplers on the left and naive samplers on the right.

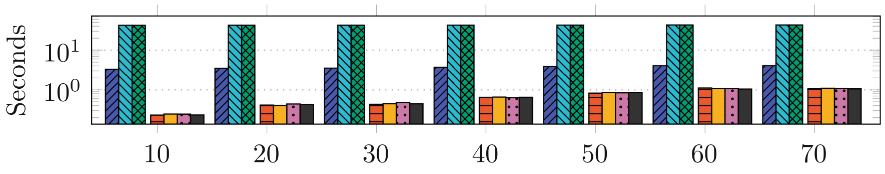


Fig. 10. The computational costs C (↓) for generating 1,000 samples in seconds of wall time across a varying amount of segments between 10 and 70. Neural samplers are in the left and naive samplers in the right chunk.

4.5 Fidelity F

With fidelity we measure how well the explanation method captures the model's decision process, independently from the human-understanding of the problem. Therefore, we evaluate the descriptive accuracy in the deletion and insertion game for different base images. Concretely, our base images are black, gray, the segment-wise mean of the original image, and the blurred original image (Fig. 10). In addition, we use the in-painting sampler to fill in the deleted parts for the image based on the remaining information, as already suggested by related work [5].

Figure 8 depicts the deletion graph (top) and the insertion graph (bottom) for the respective base images from left to right. In the deletion game a lower curve represents a better explanation quality, while in the insertion game a higher curve represents a better explanation quality. Not surprisingly, the perturbation strategies perform best when paired with their corresponding replacement strategies. This outcome is likely because fidelity aligns directly with the samplers' objectives: Capturing the model's behavior when pixels are replaced with the respective base. The fidelity evaluation effectively tests the explanations with the same perturbations used in their creation, and consequently inherently favors the corresponding perturbation strategy.

Due to these limitations of the metric we are hesitate to draw definitive conclusions from it (Fig. 11). Instead, we consider our findings as a strong indication of the need for different fidelity metrics. These novel metric should more appropriately assess the performance of various explanation strategies without being biased toward certain replacement strategies. We argue that the in-painting replacement strategy is in line with our methodology and captures interactions between feature, while the other strategies do not. Interactions are the reasons why we use complex black-box models in the first place and should therefore definetely be considered in the explanation method as well.

4.6 Stability S

A stable explanation method facilitates trust in the generated explanations. Neural samplers can produce a wide variety of perturbed samples, potentially reducing the stability of explanations. To investigate this, we assess the stability across 80 test sample, each with 5 explanations generated using $n = 800$ perturbations. We conduct this analysis across three segmentation counts (30, 70, and 110 segments) and add randomly generated explanations as a baseline.

Figure 9 presents the results of our stability study. All samplers outperform the random baseline by $\sim$1.5 orders of magnitude. The mean sampler and the blur sampler dominate the field and generate $\sim$1, respectively $\sim$0.5 orders of magnitude more stable explanation than the other 5 samplers. The commonly used black-out sampler generates the most instable explanations. Also, while subtle, we observe a decrease in stability with an increasing number of segments. In summary, neural sampling achieves competitive stability compared to traditional approaches within the evaluated context.

4.7 Computational Cost C

Here, we compare the feasibility of generative perturbation strategies based on the mean wall time for generating 1,000 perturbed samples. The evaluation is conducted on a compute node with 24 CPU cores (AMD Ryzen 9 5900X) and a single NVIDIA RTX 3090 Ti GPU. LIME's default perturbations leverage multi-threading on all CPU cores, and neural samplers utilize GPU acceleration.

Not surprisingly, neural sampler require more time. However, the flow sampler demonstrated acceptable performance by taking around 2.5$\times$ longer (5 seconds versus 2 seconds). The DDIM-based strategy, in turn, takes around 35 seconds. Fortunately, the runtime of neural samplers remains unaffected by the number of segments used during the perturbation process, while the runtime of naive sampler increases with the number of segments. This makes POMELO a feasible solution for large and very segmented input domains. In addition, this property offers a practical benefit: It allows us to generate explanations with different segmentations using the same set of generatively-perturbed samples.

5 Conclusion

We address a major limitation of LIME: The used samplers are restricted to per-segment perturbations, resulting in out-of-distribution (o.o.d.) inputs, and unavoidably, in misleading explanations. Our extension POMELO uses *full-input* perturbations instead, allowing to capture large, complex, or even distributed correlations. We show that our samplers based on normalizing flows and denoising diffusion implicit models (DDIMs) exhibit a larger overlap with human-provided annotation masks. Our method's ability to utilize full-input, and in-distribution (i.d.) perturbations benefits explanation quality decisively.

Acknowledgment. We thank Christopher Anders for his feedback on our work and Klaus-Robert Müller for his support during the project. The authors gratefully acknowledge funding from the Helmholtz Association (HGF) within the topic "46.23 Engineering Secure Systems", and by the state of Baden-Württemberg through bwHPC.

A Details on the DDIM-Based Sampler

The reverse process of a DDIM is given by:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_t-\sigma_t^2}\epsilon_\theta(x_t, t) + \sigma_t\epsilon_t \qquad (3)$$

where σ_t controls the noise added in the reverse process, α_t represents the cumulative product of noise schedule coefficients, and $\epsilon_\theta(x_t, t)$ is the predicted noise at timestep t [44]. The noise schedule $\sigma_t(\eta)$ is defined as:

$$\sigma_t(\eta) = \eta\sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}} \qquad (4)$$

where $\eta = 1$ results in the DDIM reverse process equaling the DDPM reverse process, and $\eta = 0$ results in a deterministic reverse process.

Leveraging DDIM for Perturbation Generation. Assuming a deterministic reverse process ($\eta = 0$), the DDIM reverse process equation can be rearranged to derive the iterative definition of the forward process [7]:

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}} \right) + \sqrt{1-\alpha_{t+1}}\epsilon_\theta(\mathbf{x}_t, t) \qquad (5)$$

With many forward and reverse steps, this deterministic diffusion process can faithfully reconstruct $\mathbf{x}_0$ with minimal error [44]. To generate an in-distribution perturbation $\mathbf{x}'$ we first perform n forward-steps and collect the noisy intermediates $\tilde{\mathbf{x}_i}$ at each timestep. Next perform $m = n - i$ probabilistic ($\eta > 0$) reverse steps starting from the i-th intermediate. This probabilistic nature introduces

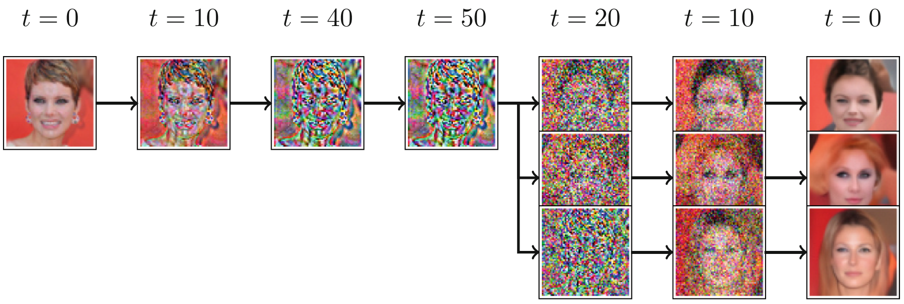


Fig. 11. Schematic of the proposed perturbation pipeline, which utilizes a deterministic forward process to generate noisy intermediates, followed by a probabilistic reverse process to introduce controlled randomness into the reconstruction.

randomness into the denoising process, which alters the trajectory at each reverse step, producing different perturbations even when starting from the same intermediate. In our experiments we always use $i = 0$, meaning we start the reverse process from the initial noisy intermediate.

B Hyperparameters

In Table 3, we present the hyperparameters we use in our experiment.

While an extensive grid search was computationally infeasible, we performed a manual search to find the best hyperparameters for each sampler. We generated perturbed samples for different samples with different values for the listed hyperparameters and selected the best in terms of locality and diversity by inspecting the resulting perturbed samples.

Table 3. Hyperparameter settings for our experiments.

Method	Hyperparameters
LIME	Kernel width = 0.25, Number of perturbations = 800
SLIC	Number of segments = 35, Compactness = 10
Traditional Samplers	Segment turn off rate = 0.5
Flow Sampler	$\mu = 0.5$, $\sigma = 0.3$
Diffusion Sampler	$\eta = 0.4$, $n = 50$, $i = 0$, $m = 50$
Inpainting	$\eta = 0.8$, $n = 50$, $i = 0$, $m = 50$, Segment turn off rate = 0.75

C Examples

In Fig. 12, we present examples of perturbed samples for the three samplers.

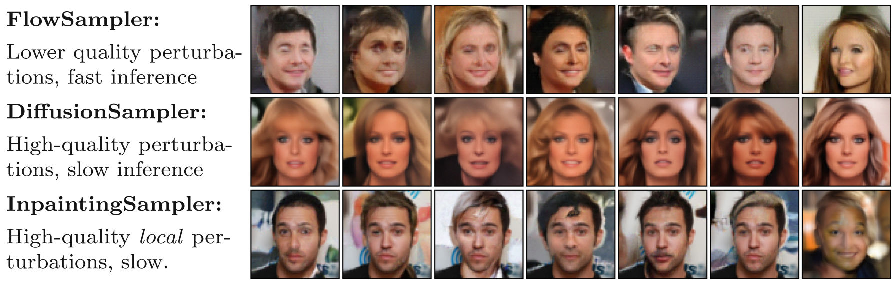

FlowSampler:
Lower quality perturbations, fast inference
DiffusionSampler:
High-quality perturbations, slow inference
InpaintingSampler:
High-quality *local* perturbations, slow.

Fig. 12. In-distribution perturbations generated using the three proposed in-distribution sampling strategies.

D Additional Reasonability Results

In addition to the IoU and the precision-recall plots in Sect. 4.4, we present the plots for the explanation mask recall and the explanation mask precision in Fig. 13. Generative samplers show au par results with the other baselines in the precision measure. In the recall measure they outperform traditional samplers.

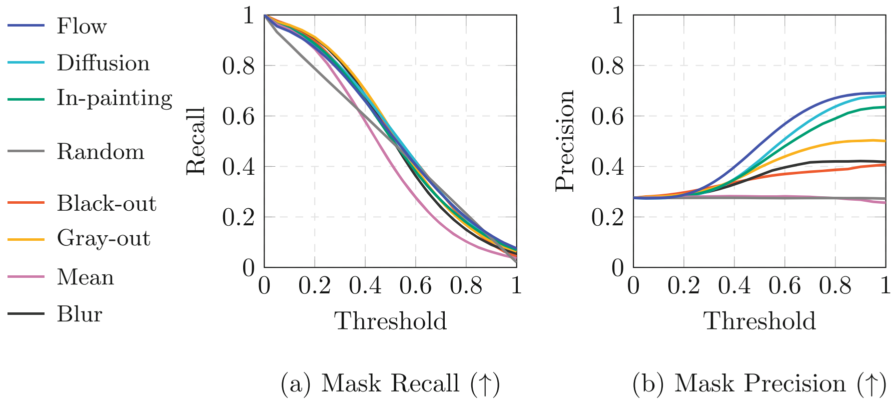


Fig. 13. Additional results of the explanation mask recall and precision.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Agarwal, C., Nguyen, A.: Explaining image classifiers by removing input features using generative models. In: Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020,

Revised Selected Papers, Part VI, Lecture Notes in Computer Science, vol. 12627, pp. 101–118 (2020)

3. Anders, C.J., Weber, L., Neumann, D., Samek, W., Müller, K.R., Lapuschkin, S.: Finding and removing clever Hans: using explanation methods to debug and improve deep models. Info. Fusion **77**, 261–295 (2022)
4. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by Layer-Wise Relevance Propagation. PLOS ONE **77**, 46 (2015)
5. Bluecher, S., Vielhaben, J., Strodthoff, N.: Decoupling pixel flipping and occlusion strategy for consistent XAI benchmarks. Trans. Mach .Learn. Res. **2024** (2024)
6. Chang, C.H., Creager, E., Goldenberg, A., Duvenaud, D.: Explaining image classifiers by counterfactual generation. In: Proc. of the International Conference on Learning Representations (ICLR) (2019)
7. Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (2021)
8. Dinh, L., Krueger, D., Bengio, Y.: NICE: Non-linear independent components estimation. In: Proc. of the International Conference on Learning Representations (ICLR) (2015)
9. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using Real NVP. In: Proc. of the International Conference on Learning Representations (ICLR) (2017)
10. Dombrowski, A.K., Gerken, J.E., Müller, K.R., Kessel, P.: Diffeomorphic counterfactuals with generative models. IEEE Trans. Pattern Anal. Mach. Intell. **46**(5), 3257–3274 (2024)
11. Fong, R., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3449–3457 (2017)
12. Goyal, Y., Feder, A., Shalit, U., Kim, B.: Explaining Classifiers with Causal Concept Effect (CaCE). CoRR **abs/1907.07165** (2020)
13. Guo, W., Mu, D., Xu, J., Su, P., Wang, G., Xing, X.: LEMNA: Explaining deep learning based security applications. In: Proc. of the ACM Conference on Computer and Communications Security (CCS), pp. 364–379 (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
15. Hegde, A., Noppel, M., Wressnegger, C.: Model-manipulation attacks against black-box explanations. In: Proc. of the Annual Computer Security Applications Conference (ACSAC) (2024)
16. Hendrycks, D., Dietterich, T.G.: Benchmarking neural network robustness to common corruptions and perturbations. In: Proc. of the International Conference on Learning Representations (ICLR) (2019)
17. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: AugMix: a simple data processing method to improve robustness and uncertainty. In: Proc. of the International Conference on Learning Representations (ICLR) (2020)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS) (2020)
19. Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W.: Explainable AI methods - a brief overview. In: Proc. of the International Workshop, beyond Explainable AI (xxAI), Lecture Notes in Computer Science, vol. 13200, pp. 13–38 (2020)

20. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Proc. of the International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, vol. 80, pp. 2673–2682 (2018)
21. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1×1 convolutions. In: Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS), pp. 10236–10245 (2018)
22. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images. Citeseer (2009)
23. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking clever Hans predictors and assessing what machines really learn. Nat. Commun. **10**(1), 1096 (2019)
24. Lee, C.H., Liu, Z., Wu, L., Luo, P.: MaskGAN: towards diverse and interactive facial image manipulation. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5548–5557 (2020)
25. Lee, K., Lee, H., Lee, J., Shin: A simple unified framework for detecting out-of-distribution samples and adversarial attacks (2018)
26. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV) (2015)
27. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Gool, L.V.: RePaint: inpainting using denoising diffusion probabilistic models. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11451–11461 (2022)
28. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proc. of the Annual Conference on Neural Information Processing Systems (NIPS), p. 10 (2017)
29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: Proc. of the International Conference on Learning Representations (ICLR) (2018)
30. McInnes, L., Healy, J., Saul, N., Großberger, L.: UMAP: uniform manifold approximation and projection. J. Open Source Softw. **3**(29), 861 (2018)
31. Nicoli, K.A., Nakajima, S., Strodthoff, N., Samek, W., Müller, K.R., Kessel, P.: Asymptotically unbiased estimation of physical observables with neural samplers. Phys. Rev. E: Stat. Phys., Plasmas, Fluids **101**(2), 023304 (2020)
32. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized input sampling for explanation of black-box models. CoRR **abs/1806.07421** (2018)
33. Qiu, L., et al.: Generating perturbation-based explanations with robustness to out-of-distribution data. In: Proc. of the International World Wide Web Conference (WWW), pp. 3594–3605 (2022)
34. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. In: Proc. of the International Conference on Machine Learning (ICML), JMLR Workshop and Conference Proceedings, vol. 37, pp. 1530–1538 (2015)
35. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": explaining the predictions of any classifier. In: Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD) (2016)
36. Rieger, L., Singh, C., Murdoch, W.J., Yu, B.: Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In: Proc. of the International Conference on Machine Learning (ICML), vol. 119 (2020)
37. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)

38. Saito, S., Chua, E., Capel, N., Hu, R.: Improving LIME robustness with smarter locality sampling. CoRR **abs/2006.12302** (2020)
39. Schockaert, C., Macher, V., Schmitz, A.: VAE-LIME: deep generative model based approach for local data-driven model interpretability applied to the ironmaking industry. CoRR **abs/2007.10256** (2020)
40. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vision **128**(2), 336–359 (2020)
41. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: Proc. of the International Conference on Learning Representations (ICLR) Workshop Track Proceedings (2014)
42. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods. In: Proc. of the AAAI/ACM Conference AI, Ethics, and Society (AIES), pp. 180–186 (2020)
43. Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proc. of the International Conference on Machine Learning (ICML), pp. 2256–2265 (2015)
44. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: Proc. of the International Conference on Learning Representations (ICLR) (2021)
45. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS) (2019)
46. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. CoRR **abs/1703.01365** (2017)
47. Szegedy, C., et al.: Intriguing properties of neural networks. In: Proc. of the International Conference on Learning Representations (ICLR) (2014)
48. Tritscher, J., Lissmann, P., Wolf, M., Krause, A., Hotho, A., Schlör, D.: Generative inpainting for shapley-value-based anomaly explanation. In: Proc. of the World Conference on eXplainable Artificial Intelligence (XAI), Communications in Computer and Information Science, vol. 2153, pp. 230–243 (2024)
49. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. Proc. of the IEEE Trans. Image Process. **13**(4), 600–612 (2004)
50. Warnecke, A., Arp, D., Wressnegger, C., Rieck, K.: Evaluating explanation methods for deep learning in security. In: Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P) (2020)
51. Wu, H., Bezold, G., Günther, M., Boult, T.E., King, M.C., Bowyer, K.W.: Consistency and accuracy of CelebA attribute values. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3258–3266 (2023)
52. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 7333–7343 (2019)
53. Ying, R., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks. In: Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS), pp. 9240–9251 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.