Exploring Image Regions Not Well Encoded by an INN

Zenan Ling
Key Laboratory of Machine Perception
School of Artificial Intelligence
Peking University

Fan Zhou
Department of Computer Science and Engineering
Shanghai Jiao Tong University

Meng Wei
Graphics and Interaction Group
University of Cambridge

Quanshi Zhang*
John Hopcroft Center
MoE Key Lab of Artificial Intelligence, AI Institute
Shanghai Jiao Tong University

Abstract

This paper proposes a method to clarify image regions that are not well encoded by an invertible neural network (INN), *i.e.* image regions that significantly decrease the likelihood of the input image. The proposed method can diagnose the limitation of the representation capacity of an INN. Given an input image, our method extracts image regions, which are not well encoded, by maximizing the likelihood of the image. We explicitly model the distribution of not-well-encoded regions. A metric is proposed to evaluate the extraction of the not-well-encoded regions. Finally, we use the proposed method to analyze several state-of-the-art INNs trained on various benchmark datasets.

1 INTRODUCTION

Deep generative models have achieved remarkable success in image generation [Miyato et al., 2018], natural language generation [Yu et al., 2016] and audio synthesis [Den Oord et al., 2016]. Among popular generative models, invertible neural networks (INNs) are distinct, because INNs guarantee the one-to-one correspondence between the input data and its latent vector. For each input image $\mathbf{x}$, an INN directly inverts the latent vector $\mathbf{z}$ back to $\mathbf{x}$ without an additional decoder, and INNs can explicitly compute the likelihood $p(\mathbf{x})$ through the change of variables theorem. Therefore, compared with the explanation for other generative networks, the explanation for INNs proposes distinct challenges and is of specific values (which will be explained later).

In this paper, we aim to clarify *image regions that are not well encoded by an INN*. Theoretically, INNs can represent all images, because each image $\mathbf{x}$ can be assigned to a unique latent vector $\mathbf{z}$. However, if an image region in $\mathbf{x}$ significantly decreases $p(\mathbf{x})$, we consider the image region not well encoded. In other words, the INN usually selectively models certain image regions and omits trivial or noisy information in the image. The not-well-encoded regions reflect regions with significant influence on the probability but have not been sophisticatedly modeled by the INN.

Given a trained INN and an input image $\mathbf{x}$, our goal is to disentangle $\mathbf{x}$ into two parts: image components $\hat{\mathbf{x}}$ that can be generated with a high likelihood, and $\Delta\mathbf{x}$ that are not well encoded, $\mathbf{x} = \hat{\mathbf{x}} + \Delta\mathbf{x}$. For example, in Figure 1, the "headwear" of the female and the "glasses" of the male are considered as regions that are not well encoded by INNs.

Through the quantitative disentanglement of not-well-encoded image regions, our work provides detailed analysis of the representation capacity of an INN. The proposed method can be used to visualize infrequent concepts that are not learned by the INN due to the dataset bias. Therefore, our work can guide the future collection of training samples.

In order to disentangle the not-well-encoded image region $\Delta\mathbf{x}$, we propose a method inspired by adversarial attacking. Given an input image $\mathbf{x}$, we estimate the image component $\hat{\mathbf{x}}$ which maximizes the likelihood $p(\hat{\mathbf{x}})$, and meanwhile extract a small not-well-encoded

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. * Corresponding author.

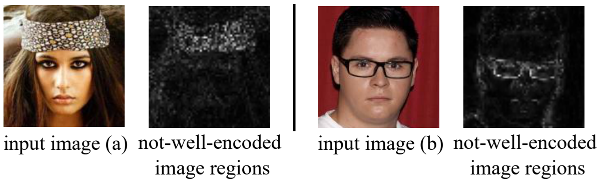

input image (a)
not-well-encoded image regions
input image (b)
not-well-encoded image regions

Figure 1: Given input images, the proposed method disentangles image regions that are not well encoded, *i.e.* those significantly decrease likelihoods of images.

perturbation $\Delta\mathbf{x}$. We assume that the concept is localized, *e.g.* an abnormal object. This assumption has been verified in Figure 1 as well as discussions in the experiment section. Moreover, these assumptions are discussed in the experiment section. Thus, we explicitly model the spatial distribution of $\Delta\mathbf{x}$.

Previous studies usually diagnose the representation capacity of neural networks at the sample level, *i.e.* clarifying the difference in distributions between generated images and true images. In comparison, this study aims to disentangle image regions that significantly decrease the likelihood at the pixel level, which sheds new light on the understanding of INNs.

The proposed method has been used to analyze on five state-of-the-art INNs, including the NICE network [Dinh et al., 2015], the Real NVP network [Dinh et al., 2016], the Glow network [Kingma and Dhariwal, 2018], the FFJORD network [Grathwohl et al., 2018], and the ResFlow [Chen et al., 2019] network. Experimental results demonstrate the effectiveness of our method in the disentanglement of not-well-encoded regions. Meanwhile, our method is applied to diagnose representation flaws of INNs.

Our contributions can be summarized as follows. (1) We propose a method to disentangle not-well-encoded image regions. (2) We apply the method to visualize concepts that are not well encoded by INNs. (3) We propose an evaluation metric to verify the effectiveness of our method. (4) We use the proposed method to analyze representation flaws of INNs.

2 RELATED WORK

Invertible neural networks: The INN was an emerging research direction in recent years. Many INNs were designed to learn promising feature representations, and generate more realistic samples. NICE [Dinh et al., 2015] proposed the additive coupling layer to construct invertible networks. The affine coupling was proposed in Real NVP [Dinh et al., 2016]. Glow [Kingma and Dhariwal, 2018] improved Real NVP by replacing the fixed shuffling permutation with 1 × 1 invertible convolution. Recently, many studies focused on how to design more flexible transformations to construct INNs, such as FFJORD [Grathwohl et al., 2018] and ResFlow [Behrmann et al., 2019, Chen et al., 2019]. Moreover, applications of INNs grew rapidly such as improving adversarial robustness [Jacobsen et al., 2019], semi-supervised learning [Nalisnick et al., 2019], solving inverse problems [Ardizzone et al., 2018], generating 3D point clouds [Yang et al., 2019], etc.

Semantic explanations for neural networks: It is an intuitive way to interpret neural networks through visualization of internal feature representations. Gradient-based methods [Simonyan et al., 2013, Zeiler and Fergus, 2014, Yosinski et al., 2015] measured contributions of intermediate-layer activation units or input units by exploiting gradients of outputs *w.r.t* the input image. [Dosovitskiy and Brox, 2016] inverted feature map convolutional layers back to the input. Other methods usually estimated the pixel-wise attribution on an input image [Lundberg and Lee, 2017, Fong and Vedaldi, 2017, Kindermans et al., 2018]. CAM [Zhou et al., 2016], Grad-CAM [Selvaraju et al., 2016], and Grad-CAM++ [Chattopadhay et al., 2018] estimated the saliency of the input image using intermediate-layer features. Unlike exploring the semantic explanation, we aimed to explain the representation capacity of INNs, which provided a new perspective on understanding neural networks.

Understanding and visualization of generative networks: Many previous studies focused on explanations of GANs. [Radford et al., 2016] visualized GANs by examining the discriminator. [Zhu et al., 2016] found that inversion could be used to explore the space of a GAN. [Creswell and Bharath., 2018] exploited the inversion of a GAN for glyphs to reveal specific strokes that could not be generated by the generator. Note that [Zhu et al., 2016, Creswell and Bharath., 2018] did not explicitly formulate the probability of the image, or just used the log-likelihood to regularize the output, instead of being taken as the direct evidence to explain a DNN. Later work [Bau et al., 2019] explained what a GAN had learned by examining concepts of intermediate features. Recently, [Bau et al., 2019] explored what a GAN cannot generate by proposing a two-stage method: layer-wise network inversion and layer-wise image optimization. However, the explanation of INNs has an essential difference from the explanation of GANs in both objective and formulation, as shown in Figure 2. Furthermore, we have compared the visualization of $\Delta\mathbf{x}$ extracted by our method and [Bau et al., 2019] in Figure 2. Therefore, our methods are not compatible with the explanation

	explaining GANs [Bau et al., 2019]	explaining INNs
objective	which pixels cannot be generated	which regions cannot be well encoded
formulation	$\min_{\mathbf{x}} \|\mathbf{x}-\mathbf{x}^*\|$	$\max_{\mathbf{x}} p(\mathbf{x})$

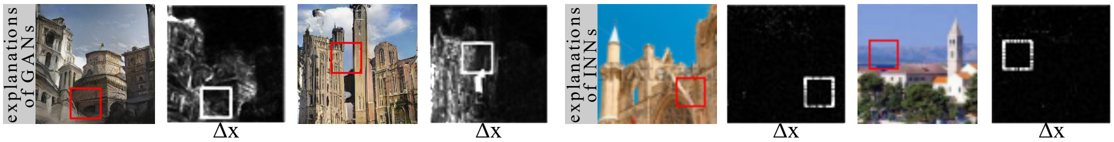


Figure 2: Distinct differences between explanations of GANs [Bau et al., 2019] and explanations of INNs. Please see supplementary materials for detailed experiment settings and more results. Theoretically, the explanation of INNs are not compatible with that of GANs.

of GANs. Moreover, a recent work [Esser et al., 2020] uses an INN to explain other neural networks with considerable impacts. In comparison, our work aims to explain the INN itself.

3 ALGORITHM

Preliminaries, invertible neural networks: Given an input image $\mathbf{x} \in \mathbb{R}^d$ and a trained INN f, the INN f uses $\mathbf{x}$ to compute a latent vector $\mathbf{z} = f(\mathbf{x})$ as the output. d denotes the dimension of the input image. As the main property that differentiates INNs from other generative networks, for the INN f, there exists an inverse function $g = f^{-1}$ which can invert the latent vector $\mathbf{z}$ back to the image $\mathbf{x}$, *i.e.* $\mathbf{x} = g(\mathbf{z})$. $\mathbf{z}$ is usually assumed to follow a Gaussian distribution $\mathbf{z} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, so that we can explicitly represent the probability density function of $p(\mathbf{z})$. Then, the probability density function of $\mathbf{x} = g(\mathbf{z})$ can be calculated through the change of variable theorem as $\log p(\mathbf{x}) = \log p(\mathbf{z}) + \log \left|\det\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$ where $\det(\cdot)$ is the determinant of a matrix. The change of variables theorem provides an efficient way to construct complex probability distributions by transforming a simple prior distribution $p(\mathbf{z})$. In general, INNs are trained using the maximum likelihood objective, *i.e.* $\max_f \sum_{\mathbf{x}} \log p(\mathbf{x})$. Note that the INN requires the dimension of the output to be exactly the same as the input dimension, and the INN is usually carefully designed to avoid $\det|\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}| = 0$.

Besides INNs, there are many generative networks, *e.g.* variational autoencoders (VAEs) and generative adversarial networks (GANs), which model the distribution of the input data. However, unlike the INN explicitly representing the probability density function $p(\mathbf{x})$, other generative models can only approximate or sample the data distribution implicitly.

Problem formulation: In this study, we aim to disentangle image regions that are not well encoded by the INN. To simplify the story, we only introduce and test our method based on the task of image generation. Given an input image $\mathbf{x}$, our goal is to disentangle image components $\hat{\mathbf{x}}$ that can be generated with a high likelihood and regions $\Delta\mathbf{x}$ that significantly decrease $p(\mathbf{x})$, as follows,

$$\mathbf{x} = \hat{\mathbf{x}} + \Delta\mathbf{x} \tag{1}$$

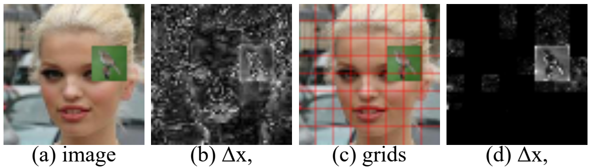
We assume that the signal of $\Delta\mathbf{x}$ of the not-well-encoded region should be much weaker. Therefore, we need to consider two terms. First, we maximize the likelihood $p(\hat{\mathbf{x}})$ of the obtained $\hat{\mathbf{x}}$, *i.e.* increasing the probability of well-encoded regions. Second, we minimize the image regions that are not well encoded, *i.e.* constraining the norm of $\Delta\mathbf{x}$. Thus, we formulate the problem as follows,

$$\max_{\hat{\mathbf{x}}} \quad \log p(\hat{\mathbf{x}}), \quad \text{s.t.} \quad \|\mathbf{x}-\hat{\mathbf{x}}\|_p < \epsilon, 0 \le \hat{\mathbf{x}}_i \le 1 \tag{2}$$

where $\|\cdot\|_p$ denotes L_p norm, and ϵ is a small positive scalar in order to constrain image regions that are not well encoded. Moreover, the obtained $\hat{\mathbf{x}}$ is ensured to be a valid image, *i.e.* $0 \le \hat{\mathbf{x}}_i \le 1$. Note that our main goal is to find $\Delta\mathbf{x}$, and $\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$. According to the change of variable theorem, we could maximize $\log p(\hat{\mathbf{x}})$ as

$$\begin{aligned} &\max_{\hat{\mathbf{x}}} \log p(\hat{\mathbf{x}}) \Leftrightarrow \\ &\max_{\Delta\mathbf{x}} \log p(f(\mathbf{x}-\Delta\mathbf{x})) + \log\left|\det\left(\frac{\partial f(\mathbf{x}-\Delta\mathbf{x})}{\partial(\mathbf{x}-\Delta\mathbf{x})}\right)\right|, \end{aligned} \tag{3}$$

such that $\|\Delta\mathbf{x}\|_p < \epsilon$, $\mathbf{x} - \Delta\mathbf{x} \in [0,1]^d$. Please see supplementary materials for the detailed deduction of Equation (3) based on Equation (2) and the change of variable theorem. Let $\hat{\mathbf{z}} = f(\hat{\mathbf{x}})$ denote the corresponding latent vector of $\hat{\mathbf{x}}$. The i-th pixel of the input image only affects $\hat{\mathbf{z}}$ slightly, if the norm of $\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{x}}_i}$ is small. Otherwise, if the norm of $\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{x}}_i}$ is large, the i-th pixel has a large impact on the latent vector $\hat{\mathbf{z}}$. For pixels whose norms of $\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{x}}_i}$ are small, we consider that information of these regions is ignored by the INN. For pixels whose norms of $\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{x}}_i}$ are large, we believe that the INN models these regions well. Meanwhile, pixels with the large



(a) image (b) Δx, without modeling spatial distribution (c) grids (d) Δx, with modeling spatial distribution

Figure 3: Comparisons between methods with and without modeling distributions of not-well-encoded regions.

norm of $\Delta\mathbf{x}_i$ have significant impact on the likelihood of the image. Thus, we can disentangle $\Delta\mathbf{x}$ that are not well encoded by the INN.

However, it is difficult to directly optimize Equation (3). Inspired by the study of adversarial attacking, we exploit the Lagrange multiplier method to relax the problem as

$$\begin{aligned} \min_{\Delta\mathbf{x}} \; & -\log p(\mathbf{x}-\Delta\mathbf{x}) + \alpha\|\Delta\mathbf{x}\|_p \\ \text{s.t.} \quad & 0 \leq \mathbf{x}_i - \Delta\mathbf{x}_i \leq 1 \end{aligned} \tag{4}$$

In the above equation, the first term aims to maximize the likelihood of image components $\hat{\mathbf{x}} = \mathbf{x} - \Delta\mathbf{x}$ and the second term forces the algorithm to extract a small perturbation $\Delta\mathbf{x}$ as the not-well-encoded region. Notice that adversarial attacks are only inspirations behind our methods. In fact, traditional adversarial attack methods cannot explicitly model the spatial information of not-well-encoded image regions. This will be shown in Figure 3.

Modeling spatial distributions of not-well-encoded image regions: Equation (4) extracts not-well-encoded image regions in a pixel-wise manner. However, when the learning of an INN is converged, it usually well encodes most image regions. Not-well-encoded regions usually correspond to abnormal objects that are localized. We assume that 1. not-well-encoded image regions usually correspond to abnormal visual concepts, and 2. these concepts are localized and relatively small. Therefore, it is necessary to explicitly model spatial distributions of not-well-encoded regions.

Specifically, we introduce a mask matrix for each image, which divides the entire image into $m \times m$ grids as shown in Figure 3(c). Each grid in Figure 3(c) represents a small image region. Let $\mathbf{M} \in \{0,1\}^{m\times m}$ denote the mask matrix, and let Λ_i denote the set of pixels in the i-th grid. We estimate a set of grids $S \subseteq \{1, ..., m^2\}$ to represent spatial distributions of not-well-encoded regions. For each not-well-encoded region $i \in S$, $\mathbf{M}_i = 1$. For each well-encoded region $i' \notin S$, we have $\mathbf{M}_{i'} = 0$. Thus, we re-parameterize $\Delta\mathbf{x}$ in (4) as $\Delta\mathbf{x} = \Delta\mathbf{x}' \odot \mathbf{M}$ where $\Delta\mathbf{x}_j = \Delta\mathbf{x}'_j \cdot \mathbf{M}_i$, if $j \in \Lambda_i$. The overall area of not well-encoded regions can be computed as $A = |S|$. Hence, we consider the following three terms: (1) Minimizing A, *i.e.* minimizing the area of not-well-encoded regions. (2) Minimizing $\|\Delta\mathbf{x}\|_p$, *i.e.* $\|\mathbf{M} \odot \Delta\mathbf{x}'\|_p$. We need to estimate not-well-encoded regions at the pixel level. (3) Maximizing $\log p(\hat{\mathbf{x}})$. The maximization of the $\log p(\hat{\mathbf{x}})$ ensures $\hat{\mathbf{x}}$ to be well encoded by the INN.

For implementations, during the learning process, we approximate $\mathbf{M}$ as $\mathbf{M} = \mathbf{M}_{A,w} = A \cdot \text{softmax}(\mathbf{w})$. $\mathbf{w} \in \mathbb{R}^{m\times m}$ is the parameter to be learned. In this way, $\mathbf{M}$ is a function of A and $\mathbf{w}$. According to the property of softmax, the value of revised mask $\mathbf{M}_{A,w}$ is sparse. We can transform Equation (4) such that $\mathbf{x} - \mathbf{M}_{A,w} \odot \Delta\mathbf{x}' \in [0,1]^d$ as follows,

$$\begin{aligned} \min_{A,\mathbf{w},\Delta\mathbf{x}'} & -\log p\left(\mathbf{x} - \mathbf{M}(A,w) \odot \Delta\mathbf{x}'\right) \\ & + \alpha \left\|\mathbf{M}(A,w) \odot \Delta\mathbf{x}'\right\|_p + \beta A, \end{aligned} \tag{5}$$

where $\mathbf{M}$ measures the regional attention, while $\Delta\mathbf{x}'$ represents fine-grained analysis. Parameters of A, w and $\Delta x'$ are learned simultaneously. Figure 3 compares methods with and without modeling spatial distributions. In Figure 3(a), we manually add a bird into the face image. The INN trained on the CelebA dataset is supposed not to encode this bird. As shown in Figure 3(b), Equation (4) without encoding the spatial distribution of $\Delta\mathbf{x}$ leads to the global change of image pixels, while Equation (5) disentangles the added bird accurately by explicitly modeling spatial distributions of not-well-encoded regions (see Figure 3(d)). Moreover, the mask matrix in our method is learned individually for different images. More importantly, the knowledge of the mask matrix is learned only from the likelihood information encoded by a trained INN. Notice that our method does not use an additional supervised neural network to obtain the prior information about the mask matrix. Thus, the rigor of the proposed method is ensured.

Parameter settings: According to Equation (5), we need to ensure that the obtained $\hat{\mathbf{x}}$ is still a valid image, we need to control the value of $\hat{\mathbf{x}}_i$ in the range of $[0,1]$. This is termed the "box constraint" in the optimization literature. In order to use optimization algorithms that do not support box constraints, we adopt the method in [Carlini and Wagner, 2017] to re-parameterize $\Delta\mathbf{x}'$ in Equation (5) as $\Delta\mathbf{x}'_i = \frac{1}{A}[\mathbf{x}_i - \frac{1}{2}(\tanh(\Theta_i)+1)]$ by introducing a new parameter $\Theta \in \mathbb{R}^d$. In this way, we can ensure that $0 \leq \hat{\mathbf{x}}_i \leq 1$. Note that the value of $\Delta\mathbf{x}'$ keep changing along with both A and Θ during the training process. For implementations, we initialize $\Delta\mathbf{x}$ to $\mathbf{0}$, so that the objective function has the same value of $\log p(\mathbf{x})$ at the beginning of the learning process. Then, higher likelihood is obtained by maximizing $\log p(\hat{\mathbf{x}})$. After the re-parameterization, the zero initialization of $\Delta\mathbf{x}'$ is equivalent to set $\Theta_i = \tanh^{-1}(2\mathbf{x}_i - 1)$, where

$\tanh^{-1}(\cdot)$ denotes the inverse function of tanh.

4 EXPERIMENTS

Baselines & ablation methods: Although the research on INNs has emerged in recent years, to the best of our knowledge, this study has been the first to disentangle image regions that are not well encoded by an INN. Nevertheless, in order to conduct comprehensive comparisons, we have revisited previous methodologies of explaining generative networks, and extended them to be compatible with INNs as baselines. In general, previous methods of explaining generative models could be summarized as three types: likelihood-based revision of the input, likelihood-based revision of features, and learning of the encoder for inversion.

Likelihood-based Revision of the Input (LRI): Several previous studies [Zhu et al., 2016, Creswell and Bharath., 2018] of explaining GANs approximated the real boundary of image components that could be generated by GANs through solving $\mathbf{z}^* = \min_{\mathbf{z}} \|\mathbf{x} - g_{\text{GAN}}(\mathbf{z})\|_2$. The deviation $\Delta\mathbf{x} = \mathbf{x} - g_{\text{GAN}}(\mathbf{z}^*)$ revealed image components that the GAN could not generate. We extended the method of explaining GANs to explain INNs. We used a small perturbation $\Delta\mathbf{x}$, which maximized $\log p(\mathbf{x} - \Delta\mathbf{x})$, *i.e.* $\log p(\hat{\mathbf{x}})$, to disentangle image regions that were not well encoded by INNs. This baseline method, which directly maximized the likelihood of the input image, was termed *LRI*. Recall that the proposed method used a mask matrix $\mathbf{M}$ to make obtained image regions sparse. Thus, we designed another baseline method, which maximized the likelihood of the input image with the mask matrix. As a variant of *LRI*, this designed baseline method was termed *LRIM.*

Likelihood-based Revision of Features (LRF): Previous methods of explaining GANs faced the challenge of approximating the inversion of a GAN generator by solving $\min_{\mathbf{z}} \|\mathbf{x} - g_{\text{GAN}}(\mathbf{z})\|_2$. [Bau et al., 2019] relaxed the condition of directly inverting GAN by the layer-wise training approach, *i.e.*, optimizing feature representations of intermediate-layers in the generator. Therefore, we used the similar layer-wise strategy to design baselines for explaining INNs. For implementations, we first need to point out the difference between the method used in [Bau et al., 2019] and the baseline method extended to INNs. In [Bau et al., 2019], the authors first trained an additional neural network for reconstruction. Then, they optimized the feature representation of each intermediate-layer in the generator. Meanwhile, they added the regularization on the perturbation on each intermediate-layer. Due to the invertibility of the INN, we did not need an additional neural network for reconstruction. Thus, we revised their method and directly maximized $\log p(\hat{\mathbf{x}})$ by optimizing the representation of each intermediate-layer. For a given INN f with each layer denoted by f_l, the architecture of the INN could be represented as $\mathbf{x} \to \mathbf{z}^{(1)} \to \mathbf{z}^{(2)} \to \cdots \to \mathbf{z}^{(L)} = \mathbf{z}$ where $\mathbf{z}^{(l)} = f_l(\mathbf{z}^{(l-1)}) = g_l^{-1}(\mathbf{z}^{(l-1)})$, and g_l denoted the inverse function of f_l. For each l-th layer's output $\mathbf{z}^{(l)}$ of the INN, we trained the perturbation $\Delta\mathbf{z}^{(l)}$ to maximize the likelihood $\log p(\hat{\mathbf{x}})$ with the regularization on $\left\|\Delta\mathbf{z}^{(l)}\right\|_2$, *i.e.* we obtained $\hat{\mathbf{x}} = g_1(\Delta\mathbf{z}^{(0)} + g_2(\cdots(\Delta\mathbf{z}^{(L-2)} + g_L(\Delta\mathbf{z}^{(L-1)} + g_{L+1}(\mathbf{z}^{(L)}))))$ by $\min_{\Delta\mathbf{z}^1,\cdots,\Delta\mathbf{z}^L}(-\log p(\hat{\mathbf{x}}) + \sum_l \lambda_l \left\|\Delta\mathbf{z}^{(l)}\right\|^2)$. This baseline method was termed *LRFN*. The first variant of the baseline method aimed to maximize the likelihood $\log p(\hat{\mathbf{x}})$ by training the perturbation on a single intermediate-layer. Meanwhile, the magnitude of the perturbation was regularized. We termed this variant as likelihood-based revision of a single feature, *LRSFN.* The second variant of the baseline method aimed to maximize the likelihood $\log p(\hat{\mathbf{x}})$ by training the perturbation on a single intermediate-layer without the regularization of the perturbation. We termed this variant as *LRSF.*

Learning of the Encoder for Inversion (LEI): Another methodology of explaining generative networks was to use another network (which was termed an encoder) to model the inversion, *e.g.* for a GAN, $\mathbf{x} = g_{\text{GAN}}(\mathbf{z})$, the encoder approximated the inversion $\mathbf{z} = g_{\text{GAN}}^{-1}(\mathbf{x})$. Thus, we extended this methodology to be compatible with INNs.

Note that an INN could exactly compute both $\mathbf{x} = g(\mathbf{z})$ and $f(\mathbf{x}) = \mathbf{z}$. The motivation to learn an encoder could be understood as follows. The encoder usually learned concepts that frequently appeared in the dataset [Hinton et al., 2015]. The encoder usually extracted image regions that were well encoded by the INN. Thus, we conducted the baseline method as follows. Given an input image $\mathbf{x}$, we first approximated the inversion $\hat{\mathbf{z}} = \text{encoder}(\mathbf{x})$ of an INN. Then, we got $\hat{\mathbf{x}} = g(\hat{\mathbf{z}})$ by inverting the approximated $\hat{\mathbf{z}}$. Finally, we obtained $\Delta\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}}$ which could be regarded as not-well-encoded image regions. This baseline method was termed *LEI.* In experiments, we used ResNet-50 to learn an encoder.

Analysis of INNs based on real images and evaluation of the proposed method: We used our method to analyzed five state-of-the-art INNs, including the NICE network, the Real NVP network, the Glow network, the FFJORD network, and the ResFlow network. We conducted experiments on three benchmark datasets: the CelebA dataset, the CUB200-2011 dataset, and church images in the LSUN dataset. Please see detailed experimental settings in the supplementary materials.

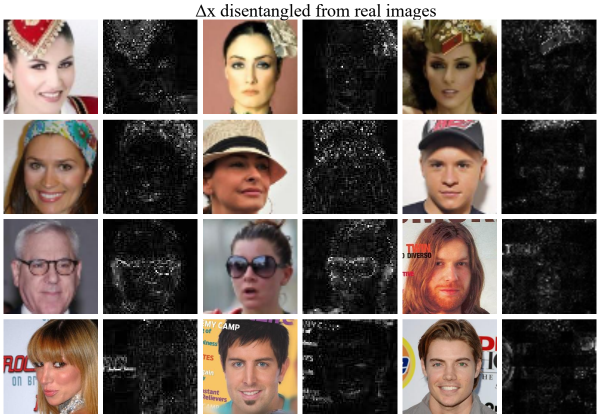


Figure 4: For real images, we visualized $\Delta\mathbf{x}$ disentangled from the Glow trained on the CelebA dataset. Visualization results indicated that the “glasses,” the “headwear,” and the “float text” could not be well encoded by the INN.

Note that different INNs were proposed and originally tested on images of different sizes. To enable fair comparisons, we trained all INNs using images that were resized to 64×64 pixels. We set $\mathbf{M}$ as an 8×8 matrix. We directly projected 8×8 elements in $\mathbf{M}$ back to the image resolution (64×64 pixels), *i.e.* image pixels in a local area shared a single $\mathbf{M}_i$ value. We also conducted experiments using $\mathbf{M}$ of different sizes as the ablation study. Please see the supplementary material for the ablation study about the size of $\mathbf{M}$. In this way, we could conduct the elementwise multiplication $\mathbf{M} \odot \Delta\mathbf{x}$ in Equation (5). Given a trained INN and an input image, we learned $\Delta\mathbf{x}$ for this image. For implementation details, we set $\beta = 0.01$ and $p = 1$ for all INNs in all experiments.

	Ours	LRI	LRIM	LRFN	LRSFN	LRSF	LEI
$S_{\Delta\mathbf{x}}$	**1.44**	1.33	1.30	1.32	1.31	1.30	1.02

Table 1: $S_{\Delta\mathbf{x}}$ of different methods on real images.

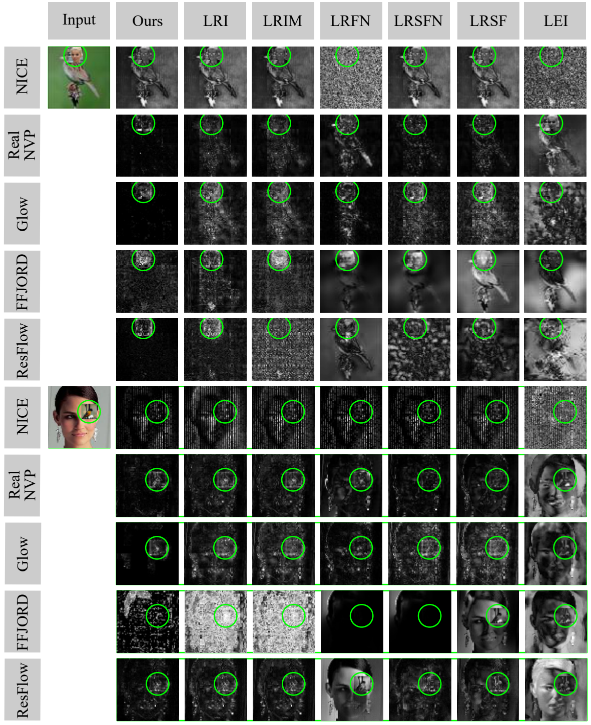


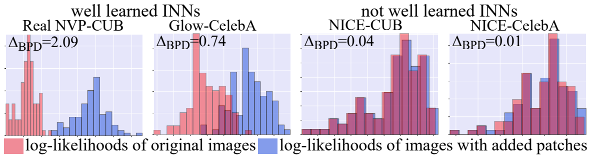
Figure 5: Visualization of outlier $\Delta\mathbf{x}$ extracted by different methods. The green circle in each sub-figure roughly indicated the square patch added to the image.

Figure 4 visualizes $\Delta\mathbf{x}$ disentangled from the Glow trained on the CelebA dataset. We analyzed images with concepts of the “hat,” the “glasses,” and the “float text” in the CelebA dataset. Notice that these concepts are typical attributes contained in the CelebA dataset. The disentangled $\Delta\mathbf{x}$ was relatively small and concentrated, which demonstrated the effectiveness of our method on real images. We found that regions of the “face” were well learned by the INN, but regions of the “hat,” the “glasses,” and the “float text” could not be well encoded by the INN. We noticed that concepts of the “hat,” the “glasses,” and the “float text” had significant influence on $p(\mathbf{x})$, although images containing some concepts only took a small portion of training images. However, these concepts were usually not well encoded, *i.e.* small perturbations may yield unstable probability. *Quantitative evaluation:* In this experiment, we tested whether our method could extract concepts that did not appear in the training set. INNs were not supposed to learn these concepts. First, we trained a Glow network using images without concepts of the “hat,” the “glasses,” and the “float text” in the CelabA dataset. The “hat” and the “glasses” were attributes provided in the CelebA dataset, and the “float text” was annotated by ourselves. Second, we used images with concepts of the “hat,” the “glasses,” and the “float text” for test. Moreover, we annotated bounding boxes of the “hat,” the “glasses,” and the “float text” as ground-truth regions.

We proposed an evaluation metric to evaluate whether these concepts could be disentangled by our method. Let I denote the region of the entire input image $\mathbf{x}$, and let Ω denote the ground-truth outlier region. We used the score $S_{\Delta\mathbf{x}} = \left(\frac{1}{|\Omega|}\sum_{i\in\Omega}\|\Delta\mathbf{x}_i\|\right) / \left(\frac{1}{|I|}\sum_{i\in I}\|\Delta\mathbf{x}_i\|\right)$ to measure the ratio of the average magnitude of the perturbation in outlier regions to the average magnitude over the entire image for evaluation. A large value of $S_{\Delta\mathbf{x}}$ indicated that outlier regions could be disentangled by our method. Note that, besides outlier regions, images still contained some regions that were not well encoded. However, we assumed that the not-well-encoded regions mainly existed in outlier regions.

Thus, $S_{\Delta \mathbf{x}}$ measured the relative concentration of the obtained image regions. $S_{\Delta \mathbf{x}}$ of different methods on real images are shown in Table 1. Quantitative results illustrated that the proposed method achieved better performance than baselines. Please see more results in supplementary materials.

Evaluation on constructed images: Although we had compared our method with baselines on real images in Table 1, in order to provide a new perspective to evaluate our method, we constructed new datasets based on existing datasets. For each image in the existing dataset, we added an image patch of the outlier concept into the image. The added patch was supposed the image region that was not well encoded. For the trustworthiness of our experiments, we would like first to choose off-the-shelf INNs provided by the authors if they were available online. Please see supplemental materials for sources of INNs that were trained using different datasets. Based on the evaluation of $\Delta \mathbf{x}$, we could differentiate well-learned and not well-learned INNs. We noticed that well-learned INNs could distinguish raw images from images with outlier patches; whereas not-well-learned INNs could not. Not every image yielded highly concentrated $\Delta \mathbf{x}$ because of the bad representation quality of the not well learned INN. On the one hand, some images naturally contained difficult concepts, but other images did not. On the other hand, some added image patches had no significant impact on the change of the likelihood. In order to compare the INN's representation quality between the raw image in the dataset and the image with added patches, we measured the difference of their bits per dimension (BPD) Δ_{BPD} as $\Delta_{\text{BPD}} = \frac{\log(p(\mathbf{x}_{\text{raw}})) - \log(p(\mathbf{x}))}{|I| \times \log 2}$. Note that sometimes Δ_{BPD} was even negative value, which indicated the bad representation quality of the INN, *i.e.* the image with outlier patches was even better modeled than the raw image by the INN. In further experiments, we computed likelihood distributions of raw images and images with outlier patches. Our method could identify INNs in which many concepts were not well-encoded. Meanwhile, likelihoods of raw images and images with outlier patches verified our conclusions.



INNs	adding red boxes			adding other datasets' patches		
	CUB	CelebA	Church	CUB	CelebA	Church
NICE	0.04	0.01	0.04	0.02	0.05	0.03
Real NVP	2.09	1.34	0.76	0.66	0.41	0.23
Glow	1.09	0.74	0.35	0.47	0.17	0.27
FFJORD	1.74	0.30	0.58	0.36	0.09	0.15
ResFlow	22.57	0.53	0.64	2.18	0.11	0.13

Figure 6: Histograms of likelihood distributions verified the representation capacity of different INNs which were evaluated by our method. Left: Histograms of likelihood distributions. Right: The table of average Δ_{BPD} of different INNs on different datasets. Note that BPD can only provide an overall score to evaluate whether the image is well encoded entirely. In comparison, our method is proposed to locate not-well-encoded regions within images in a fine-grained manner. Please see more results in supplementary materials.

We constructed two datasets by adding outlier image patches in two ways. (a): For each image in the dataset, we added an image patch of the red box. (b): For each image in the dataset, we added an image patch from another dataset. We compared the proposed method and baselines on constructed images. Visualization results are shown in Figure 5. Except for NICE and FFJORD networks, our method achieved better performance than other baselines in most cases. Moreover, neither our method nor baseline methods could disentangle added patches when we used NICE and FFJORD networks. Please see more visualization results in supplemental materials. Notice that Figure 5 showed results of $\Delta \mathbf{x}' \odot \mathbf{M}$, where $\mathbf{M}$ denotes the regional mask, and $\Delta \mathbf{x}'$ indicates the fine-grained analysis of not-well-encoded pixels. Thus, the pixel-wise visualization is more accurate than regional analysis.

Table 2 showed the efficiency of extracting not-well-encoded regions of different methods on different datasets. On the one hand, for Real NVP, Glow and ResFlow networks, our method had higher $S_{\Delta \mathbf{x}}$ than baselines. Quantitative results of $S_{\Delta \mathbf{x}}$ illustrated that image regions $\Delta \mathbf{x}$ extracted by our method were much more concentrated than those extracted by baselines in most cases. On the other hand, we noticed that NICE and FFJORD networks had relatively worse $S_{\Delta \mathbf{x}}$ than other INNs. It indicated that NICE and FFJORD networks could not distinguish raw images from images with outlier patches. Thus, we conjectured that an INN had low representation capacity if it had small $S_{\Delta \mathbf{x}}$ on the corresponding dataset. In order to verify the correctness of our conjecture, we calculated likelihood distributions and average Δ_{BPD} as shown in Figure 6. Results in Figure 6 were consistent with the conclusion conjectured from our method. Thus, our method can be used as an effective tool to diagnose the representation capacity of an INN. *Further discussion about results in Figure 6:* As shown in Figure 6, samples with added patches sometimes even yielded higher probabilities than original images. Such phenomenon was also shown in [Nalisnick et al., 2018]. If the obtained $\hat{\mathbf{x}}$ is far away from the input image $\mathbf{x}$, *i.e.* $\hat{\mathbf{x}}$ is out of the dataset distribution, the estimated

			LRI	LRIM	LRFN	LRSFN	LRSF	LEI
by adding red boxes								
CUB	NICE	1.40	1.36	1.39	**1.43**	1.32	1.26	1.16
	Real NVP	**6.49**	0.05	1.35	3.92	2.05	2.70	1.92
	Glow	**7.17**	3.01	3.63	4.62	5.28	4.42	2.08
	FFJORD	**2.50**	0.97	1.12	0.63	0.35	0.47	2.06
	ResFlow	**8.85**	4.09	5.41	0.13	0.69	0.67	2.05
CelebA	NICE	**1.23**	1.13	1.16	1.01	1.20	1.12	1.03
	Real NVP	**6.64**	0.03	1.44	4.69	1.60	2.67	1.97
	Glow	**9.55**	2.82	3.85	4.68	7.79	7.25	2.15
	FFJORD	**1.62**	0.61	0.59	0.67	0.71	0.34	1.61
	ResFlow	**6.29**	3.25	3.21	1.68	3.41	3.35	1.83
Church	NICE	1.36	1.34	1.40	**2.50**	1.56	1.49	1.17
	Real NVP	**8.47**	0.03	1.34	2.77	0.72	1.70	2.38
	Glow	**3.47**	1.61	1.66	2.67	1.20	1.57	1.92
	FFJORD	**1.27**	0.90	0.82	0.10	1.09	1.08	0.86
	ResFlow	**5.53**	2.92	3.52	0.93	2.89	1.22	1.96
by adding patches from other datasets								
CUB	NICE	**1.41**	1.29	1.37	1.05	1.36	1.29	1.25
	Real NVP	**4.35**	1.96	1.97	2.05	1.72	1.71	1.13
	Glow	**5.73**	1.61	1.69	2.14	1.75	1.58	1.18
	FFJORD	**4.21**	1.96	2.58	1.92	2.15	1.35	1.19
	ResFlow	**5.01**	2.82	1.18	1.16	1.17	1.37	1.14
CelebA	NICE	**1.47**	1.23	1.19	1.36	1.43	1.25	1.03
	Real NVP	**3.15**	1.67	1.62	2.39	1.60	1.50	0.94
	Glow	**4.98**	1.53	1.89	1.86	1.54	1.51	1.18
	FFJORD	0.84	1.05	1.08	**2.04**	1.72	1.51	1.20
	ResFlow	**1.95**	1.59	1.03	1.34	1.45	1.59	0.92
Church	NICE	1.36	1.36	1.34	1.43	**1.45**	1.37	1.23
	Real NVP	**1.84**	1.56	1.62	1.14	0.90	1.22	1.16
	Glow	**2.01**	1.34	1.33	1.20	0.69	0.73	1.20
	FFJORD	1.14	1.36	1.13	**2.39**	1.51	1.09	1.13
	ResFlow	**1.54**	1.32	1.46	1.04	1.00	1.01	0.99

Table 2: $S_{\Delta\mathbf{x}}$ of different methods on two constructed datasets. In most cases, our method had higher $S_{\Delta\mathbf{x}}$ than baselines. It indicated that image regions $\Delta\mathbf{x}$ extracted by our method were much more concentrated than those extracted by baselines.

likelihood $p(\hat{\mathbf{x}})$ is no longer a reliable index to reflect the "true probability" with which the INN can generate $\hat{\mathbf{x}}$. To avoid this phenomenon, the constraint on $\|\Delta\mathbf{x}\|_p$ is necessary here to keep the obtained $\hat{\mathbf{x}}$ close to the input image $\mathbf{x}$. In this way, $\hat{\mathbf{x}}$ is guaranteed to be in the dataset distribution and the comparison between $p(\hat{\mathbf{x}})$ and $p(\mathbf{x})$ is reliable.

Diagnosis of INNs: In this experiment, we applied the proposed method to diagnose representation flaws of existing state-of-the-art INNs. In order to diagnose representation flaws, we first constructed images with abnormal concepts, which did not exist in the dataset. Then, we used the proposed method to test whether abnormal concepts could be disentangled. We added three types of abnormal concepts to images in existing datasets. (a), we added an additional abnormal concept to the image. (b), we shuffled positions of concepts in the image. (c), given an image in the dataset, we first randomly selected an image region. Then, we maximized the saturation level in this region. We visualized $\Delta\mathbf{x}$ extracted from the Glow trained on the CelebA dataset in Figure 7. For the CelebA dataset specifically, we constructed abnormal concepts type (a) by adding an additional eye to the face; (b) by shuffling positions of eyes, mouths, and noses; and (c) by maximizing saturation level in a randomly selected region of 16 × 16 pixels on the face image.

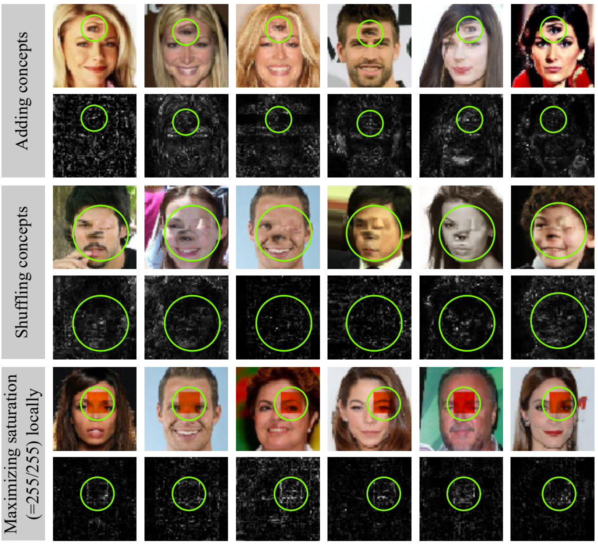


Figure 7: We visualized $\Delta\mathbf{x}$ extracted from the Glow trained on the CelebA dataset. The green circle in each sub-figure roughly indicated abnormal concepts. From top to bottom, we constructed abnormal concepts by adding an additional eye to the image, shuffling positions of eyes and the nose in the image, and maximizing the saturation level on the random selected region of 16 × 16 pixels in the image, respectively. For all of three cases, regions of abnormal concepts could not be disentangled. It illustrated that the INN did not well encode these concepts.

As shown in Figure 7, extracted image regions were not concentrated, *i.e.* abnormal concepts could not be disentangled. It indicated that abnormal concepts did not have significant influence on likelihoods. Specifically, the amount of concepts, the position relationship between concepts, or saturation level information could not be well learned by the INN for face images. Please see more results of various INNs on different datasets in supplementary materials.

5 CONCLUSION

In this paper, we focus on a new task, *i.e.* exploring image regions that are not well encoded by an INN.

Inspired by adversarial attacking, we propose a method to disentangle image regions that significantly decrease the likelihoods of the image. In particular, our method explicitly models the spatial information of not-well-encoded regions, so that we can disentangle visual concepts that are not well encoded by an INN. Considering that there is no ground truth of not-well-encoded image regions, we proposed a new evaluation metric to measure the concentration of extracted regions by our method. Experimental results have demonstrated the effectiveness of the proposed method. Moreover, we use the method to diagnose representation flaws of several state-of-the-art INNs both through quantitative analysis and visualization results.

Acknowledgements

This work is partially supported by National Science and Technology Innovation 2030 Major Project of the Ministry of Science and Technology of China under Grant (2021ZD0111602), the National Nature Science Foundation of China (No. 61906120, U19B2043), Shanghai Natural Science Fundation (21JC1403800,21ZR1434600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). This work is also partially supported by Huawei Technologies Inc.

References

[Ardizzone et al., 2018] Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. (2018). Analyzing inverse problems with invertible neural networks. *ICLR.*

[Bau et al., 2019] Bau, D., Zhu, J., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., and Torralba, A. (2019). Gan dissection: Visualizing and understanding generative adversarial networks. *ICLR.*

[Bau et al., 2019] Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., and Torralba, A. (2019). Seeing what a gan cannot generate. *ICCV.*

[Behrmann et al., 2019] Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J. H. (2019). Invertible residual networks. *ICML.*

[Carlini and Wagner, 2017] Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. pages 39–57.

[Chattopadhay et al., 2018] Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847.

[Chen et al., 2019] Chen, R. T. Q., Behrmann, J., Duvenaud, D., and Jacobsen, J.-H. (2019). Residual Flows for Invertible Generative Modeling. *International Conference on Neural Information Processing Systems.*

[Creswell and Bharath., 2018] Creswell, A. and Bharath., A. A. (2018). Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems.*

[Den Oord et al., 2016] Den Oord, A. V., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv: Sound.*

[Dinh et al., 2015] Dinh, L., Krueger, D., and Bengio, Y. (2015). Nice: Non-linear independent components estimation.

[Dinh et al., 2016] Dinh, L., Sohldickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv: Learning.*

[Dosovitskiy and Brox, 2016] Dosovitskiy, A. and Brox, T. (2016). Inverting visual representations with convolutional networks. *CVPR*, pages 4829–4837.

[Esser et al., 2020] Esser, P., Rombach, R., and Ommer, B. (2020). A disentangling invertible interpretation network for explaining latent representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[Fong and Vedaldi, 2017] Fong, R. and Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. *ICCV*, pages 3449–3457.

[Grathwohl et al., 2018] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv: Learning.*

[Hinton et al., 2015] Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv: Machine Learning.*

[Jacobsen et al., 2019] Jacobsen, J., Behrmann, J., Zemel, R. S., and Bethge, M. (2019). Excessive invariance causes adversarial vulnerability.

[Kindermans et al., 2018] Kindermans, P., Schutt, K. T., Alber, M., Muller, K., Erhan, D., Kim, B., and Dahne, S. (2018). Learning how to explain neural networks: Patternnet and patternattribution. *ICLR.*

[Kingma and Dhariwal, 2018] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *arXiv: Machine Learning*.

[Lundberg and Lee, 2017] Lundberg, S. and Lee, S. (2017). A unified approach to interpreting model predictions. *Neural information processing systems*, pages 4768–4777.

[Miyato et al., 2018] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv: Learning*.

[Nalisnick et al., 2018] Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2018). Do deep generative models know what they don't know?

[Nalisnick et al., 2019] Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2019). Hybrid models with deep and invertible features. *arXiv: Learning*.

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*.

[Selvaraju et al., 2016] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2016). Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV*.

[Simonyan et al., 2013] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps.

[Yang et al., 2019] Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S. J., and Hariharan, B. (2019). Pointflow: 3d point cloud generation with continuous normalizing flows. *ICCV*.

[Yosinski et al., 2015] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. J., and Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv: Computer Vision and Pattern Recognition*.

[Yu et al., 2016] Yu, L., Zhang, W., Wang, J., and Yu, Y. (2016). Seqgan: Sequence generative adversarial nets with policy gradient. pages 2852–2858.

[Zeiler and Fergus, 2014] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*, pages 818–833.

[Zhou et al., 2016] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. pages 2921–2929.

[Zhu et al., 2016] Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. (2016). Generative visual manipulation on the natural image manifold. *ECCV*.

Supplementary Material:
Exploring Image Regions Not Well Encoded by an INN

A Detailed deduction of Equation 3

$$
\begin{aligned}
&\max_{\hat{\mathbf{x}}} p(\hat{\mathbf{x}}),\ \mathrm{s.\,t.}\ \|\Delta\mathbf{x}\|_p < \epsilon \\
\Leftrightarrow &\max_{\hat{\mathbf{x}}} \log p(\hat{\mathbf{x}}),\ \mathrm{s.\,t.}\ \|\Delta\mathbf{x}\|_p < \epsilon \\
\Leftrightarrow &\max_{\hat{\mathbf{x}}} \log p(\hat{\mathbf{z}}) + \log\left|\det\left(\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{x}}}\right)\right|,\mathrm{s.\,t.}\ \|\Delta\mathbf{x}\|_p < \epsilon \\
\Leftrightarrow &\max_{\hat{\mathbf{x}}} \log p(f(\hat{\mathbf{x}})) + \log\left|\det\left(\frac{\partial f(\hat{\mathbf{x}})}{\partial\hat{\mathbf{x}}}\right)\right|,\mathrm{s.\,t.}\ \|\Delta\mathbf{x}\|_p < \epsilon \\
\Leftrightarrow &\max_{\Delta\mathbf{x}} \log p(f(\mathbf{x}-\Delta\mathbf{x})) + \log\left|\det\left(\frac{\partial f(\mathbf{x}-\Delta\mathbf{x})}{\partial(\mathbf{x}-\Delta\mathbf{x})}\right)\right| \\
&\mathrm{s.\,t.}\ \|\Delta\mathbf{x}\|_p < \epsilon
\end{aligned}
$$

where $\hat{\mathbf{z}} = f(\hat{\mathbf{x}}) = f(\mathbf{x} - \Delta\mathbf{x})$.

B Detailed experimental settings

	NICE	Real NVP	Glow	FFJORD	ResFlow
CUB	None	None	Code	None	None
CelebA	None	Code	Net Code	Code	Net Code
Church	None	None	Net Code	None	None

Table 3: Sources of INNs that were trained using different datasets. In order to obtain trustworthy conclusions, we preferred to use off-the-shelf INNs in the second experiment. "Net" indicates that pre-trained INNs were released. "Code" indicates that off-the-shelf codes were available. "None" indicates that neither the model nor the code was available online. In this case, we trained INNs by ourselves according to settings suggested by authors.

According to Equation (4), we need to ensure that the obtained $\hat{\mathbf{x}}$ is still a valid image, we need to control the value of $\hat{\mathbf{x}}_i$ in the range of $[0, 1]$. This is termed the "box constraint" in the optimization literature. In order to use optimization algorithms that do not support box constraints, we adopt the method in [Carlini and Wagner, 2017] to re-parameterize $\Delta\mathbf{x}'$ in Equation (4) as $\Delta\mathbf{x}'_i = \frac{1}{A}[\mathbf{x}_i - \frac{1}{2}(\tanh(\Theta_i) + 1)]$ by introducing a new parameter $\Theta \in \mathbb{R}^d$. In this way, we can ensure that $0 \leq \hat{\mathbf{x}}_i \leq 1$. Note that the value of $\Delta\mathbf{x}'$ keep changing along with both A and Θ during the training process. For implementations, we initialize $\Delta\mathbf{x}$ to $\mathbf{0}$, so that the objective function has the same value of $\log p(\mathbf{x})$ at the beginning of the learning process. Then, higher likelihood is obtained by maximizing $\log p(\hat{\mathbf{x}})$. After the re-parameterization, the zero initialization of $\Delta\mathbf{x}'$ is equivalent to set $\Theta_i = \tanh^{-1}(2\mathbf{x}_i - 1)$, where $\tanh^{-1}(\cdot)$ denotes the inverse function of tanh.

Note that different INNs were proposed and originally tested on images of different sizes . To enable fair comparisons, we trained all INNs using images that were resized to 64×64 pixels. We set $\mathbf{M}$ as an 8×8 matrix. We directly projected 8×8 elements in $\mathbf{M}$ back to the image resolution (64×64 pixels), *i.e.* image pixels in a local area shared a single $\mathbf{M}_i$ value. We also conducted experiments using $\mathbf{M}$ of different sizes as the ablation study. Please see the supplementary material for the ablation study about the size of $\mathbf{M}$. In this way, we could conduct the elementwise multiplication $\mathbf{M} \odot \Delta\mathbf{x}$ in Equation (4). Given a trained INN and an input image, we learned $\Delta\mathbf{x}$ for this image. For implementation details, we set $\beta = 0.01$ and $p = 1$ for all INNs in all experiments.

C More results on real images

For real images, we visualized $\Delta\mathbf{x}$ disentangled from the Glow trained on the CelebA dataset in Figure 8. As Figure 8 showed, the disentangled $\Delta\mathbf{x}$ was relatively small and concentrated, which demonstrated the effectiveness of our method on real images. From Figure 8, we observed that regions of "face" were well learned by the INN, but regions of the "headwear" , the "glasses", and the "foat text" could not be well encoded by the INN.

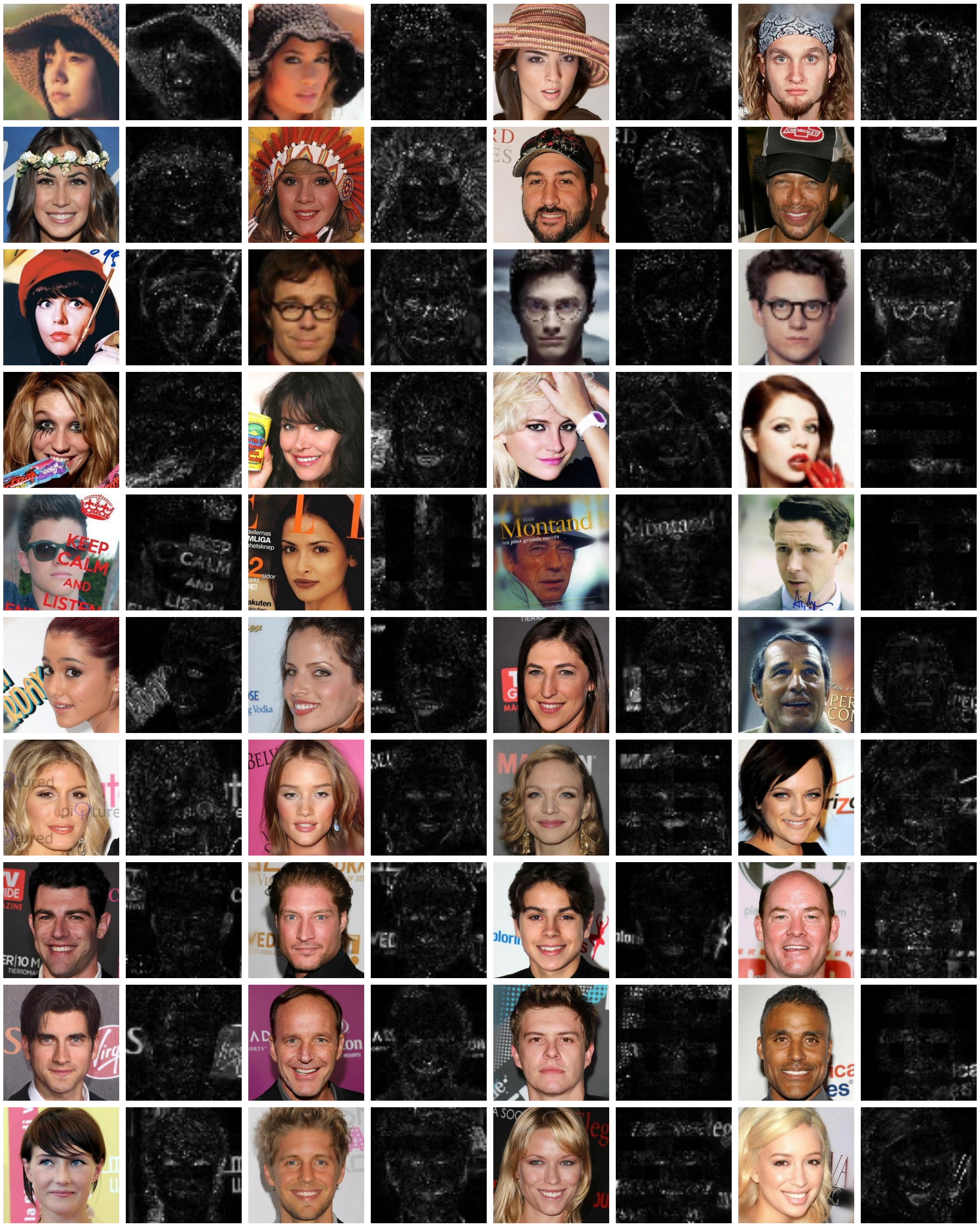


Figure 8: More results on real images.

D More results for constructed images by adding red boxes

We visualized $\Delta\mathbf{x}$ extracted by different methods. In this experiment, we added the red box to the image. For Real NVP, Glow and ResFlow networks, visualization results showed that our method could successfully disentangle added patches, while baseline methods could not. For NICE and FFJORD networks, neither our method nor baseline methods could disentangle red boxes. Moreover, image regions extracted by our method were more concentrated than those extracted by baselines.

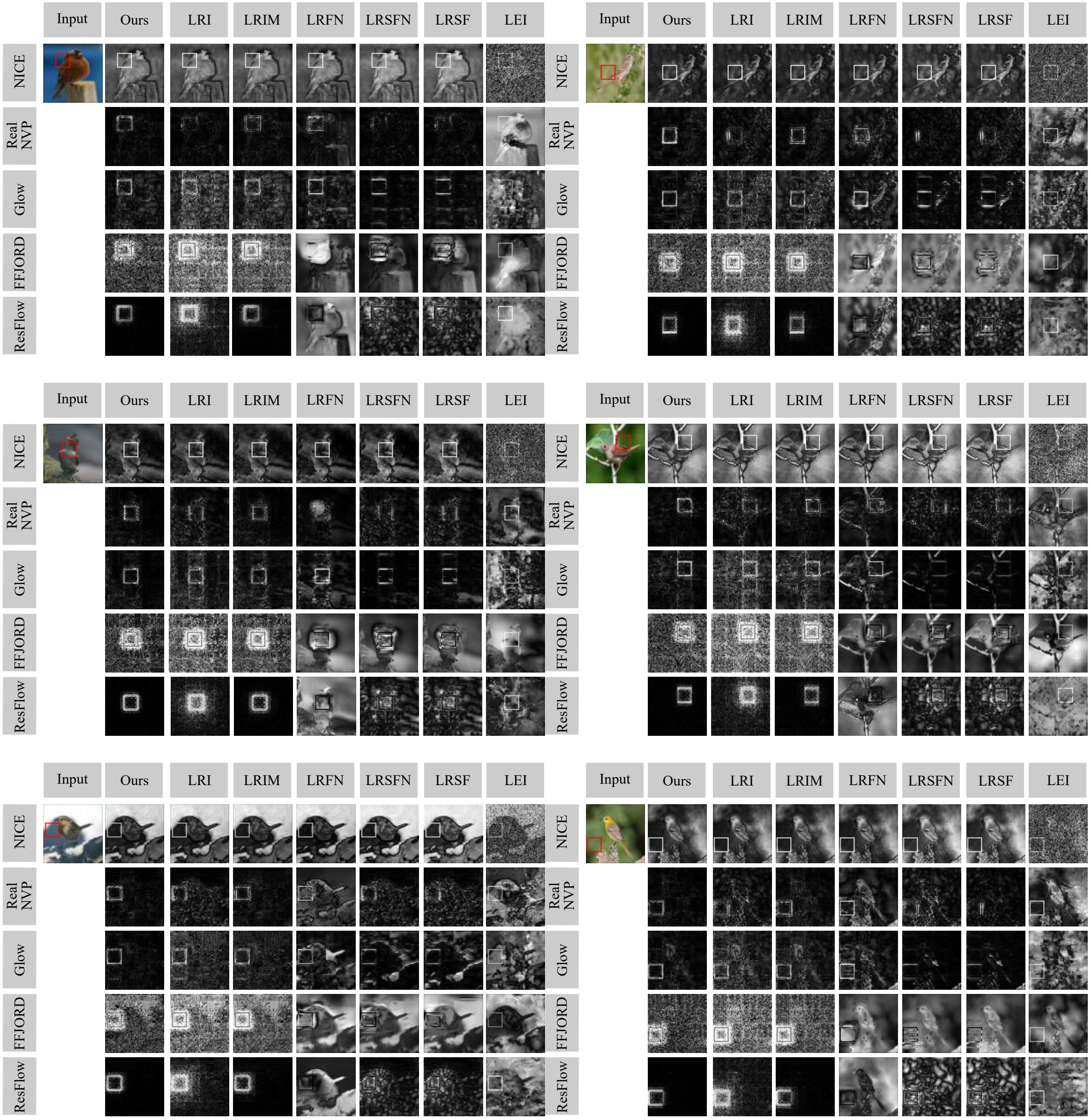


Figure 9: More results for revised images by adding red boxes to images in the CUB200-2011 dataset.

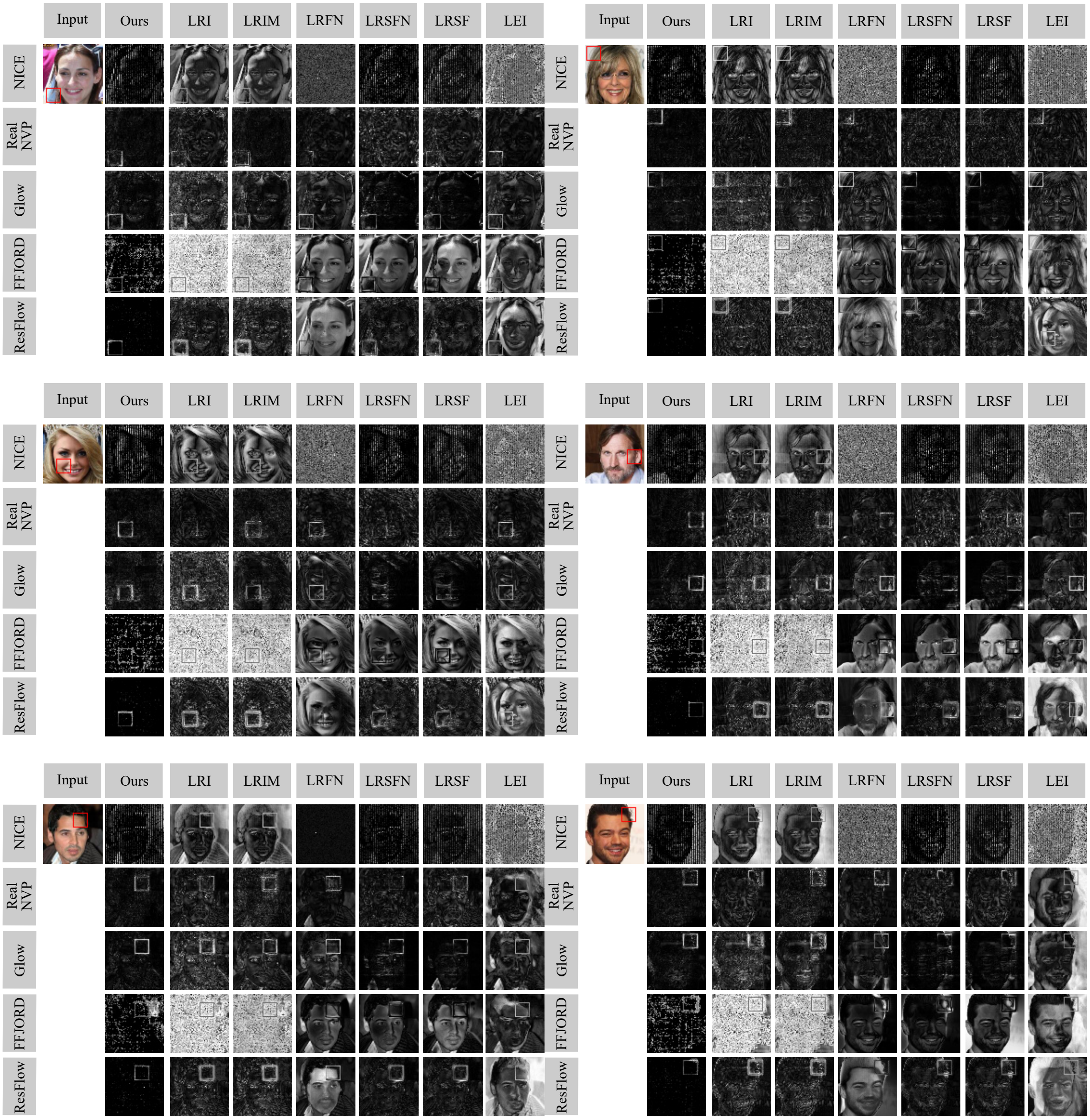


Figure 10: More results for revised images by adding red boxes to images in the CelebA dataset.

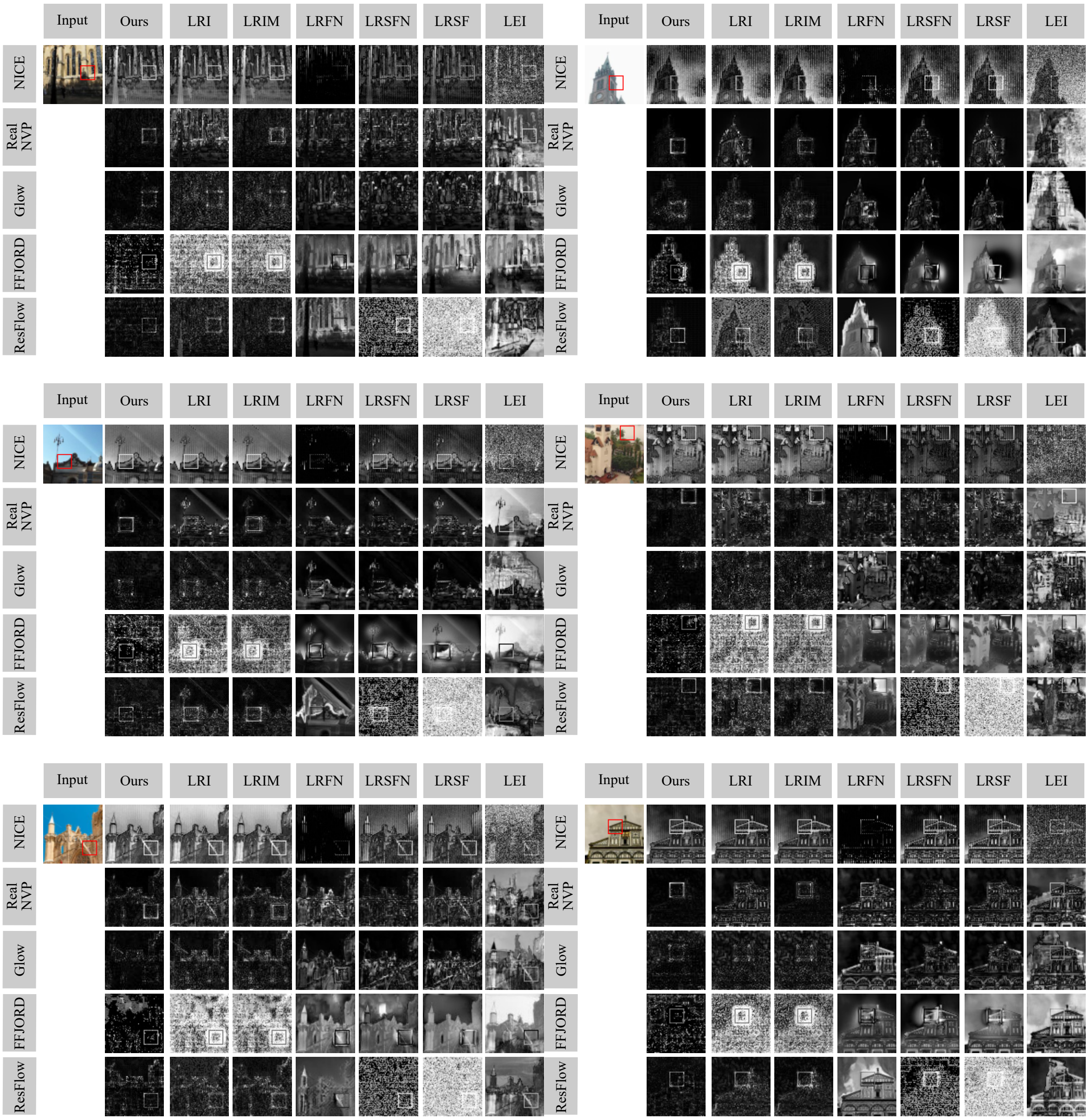


Figure 11: More results for revised images by adding red boxes to church images in the LSUN dataset.

E More results for constructed images by adding image patches from other datasets

We visualized $\Delta\mathbf{x}$ extracted by different methods. The green circle in each sub-figure roughly indicated the square patch added added to the image. For Real NVP, Glow and ResFlow networks, results showed that our method could successfully disentangle added patches, while baseline methods could not. For NICE and FFJORD networks, neither our method nor baseline methods could disentangle outlier patches. Moreover, image regions extracted by our method were more concentrated than those extracted by baselines. Moreover, in Figure 15, we used green boxes to mark INNs that had lower $S_{\Delta\mathbf{x}}$. Figure 15 showed that in green boxes, likelihood distributions of raw images and images with outlier patches had no significant differences, *i.e.* image distributions were not well learned by corresponding INNs. Results in Figure 15 were consistent with the conclusion conjectured from our method. Thus, our method can be used as an effective tool to diagnose the representation capacity of an INN.

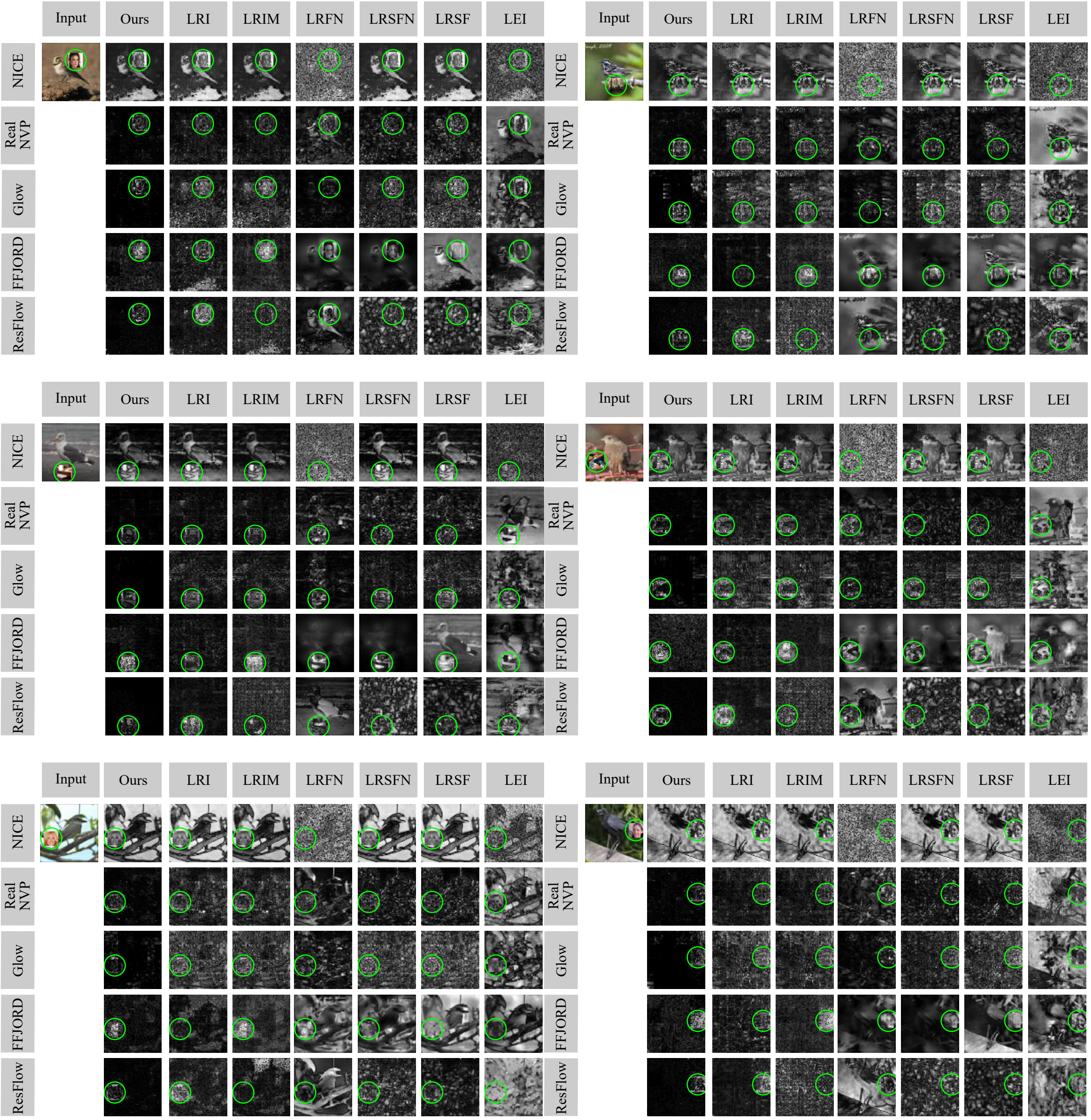


Figure 12: More results for revised images by adding image patches from other datasets to images in the CUB200-2011 dataset.

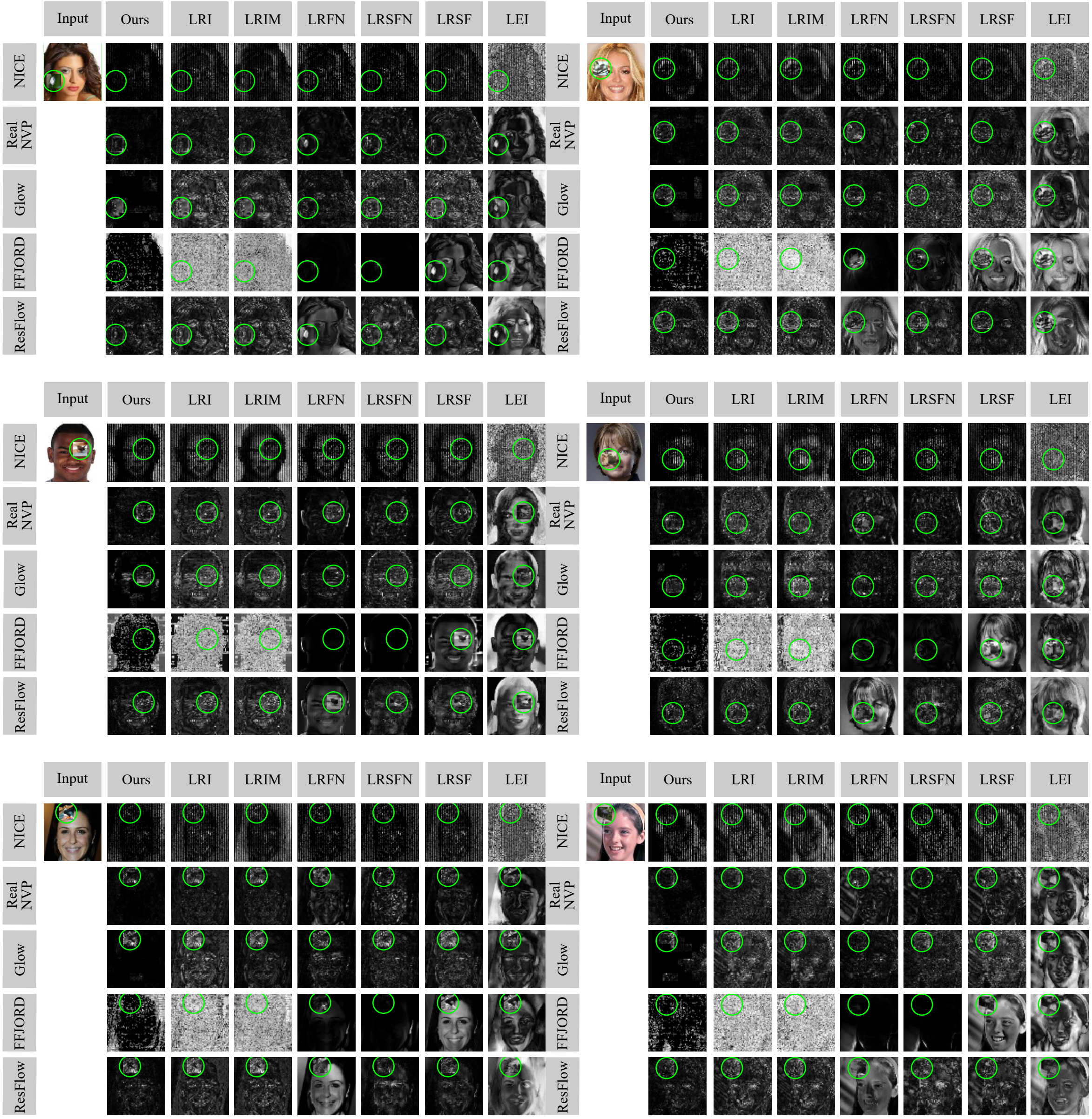


Figure 13: More results for revised images by adding image patches from other datasets to images in the CelebA dataset.

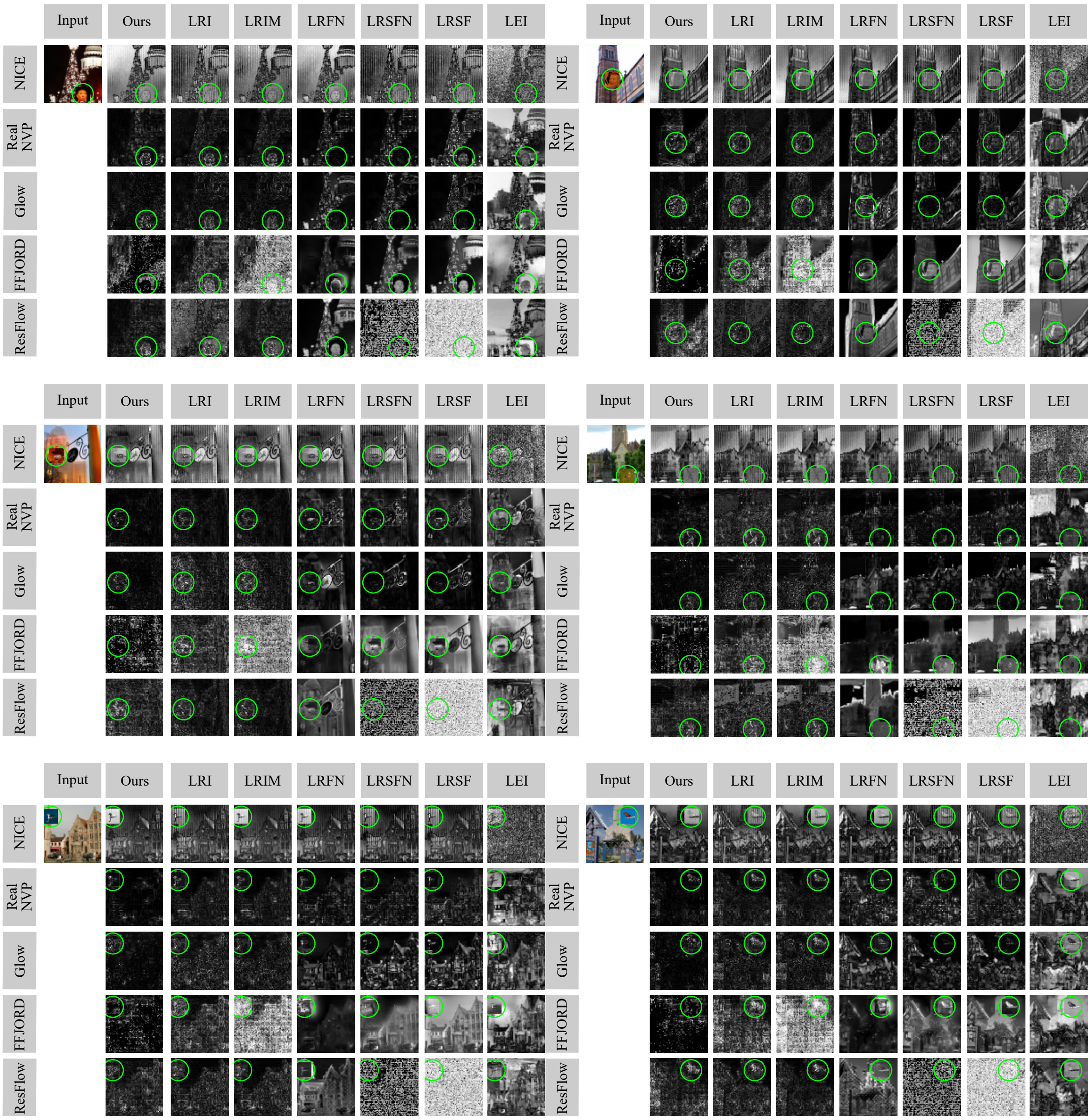


Figure 14: More results for revised images by adding image patches from other datasets to church images in the LSUN dataset.

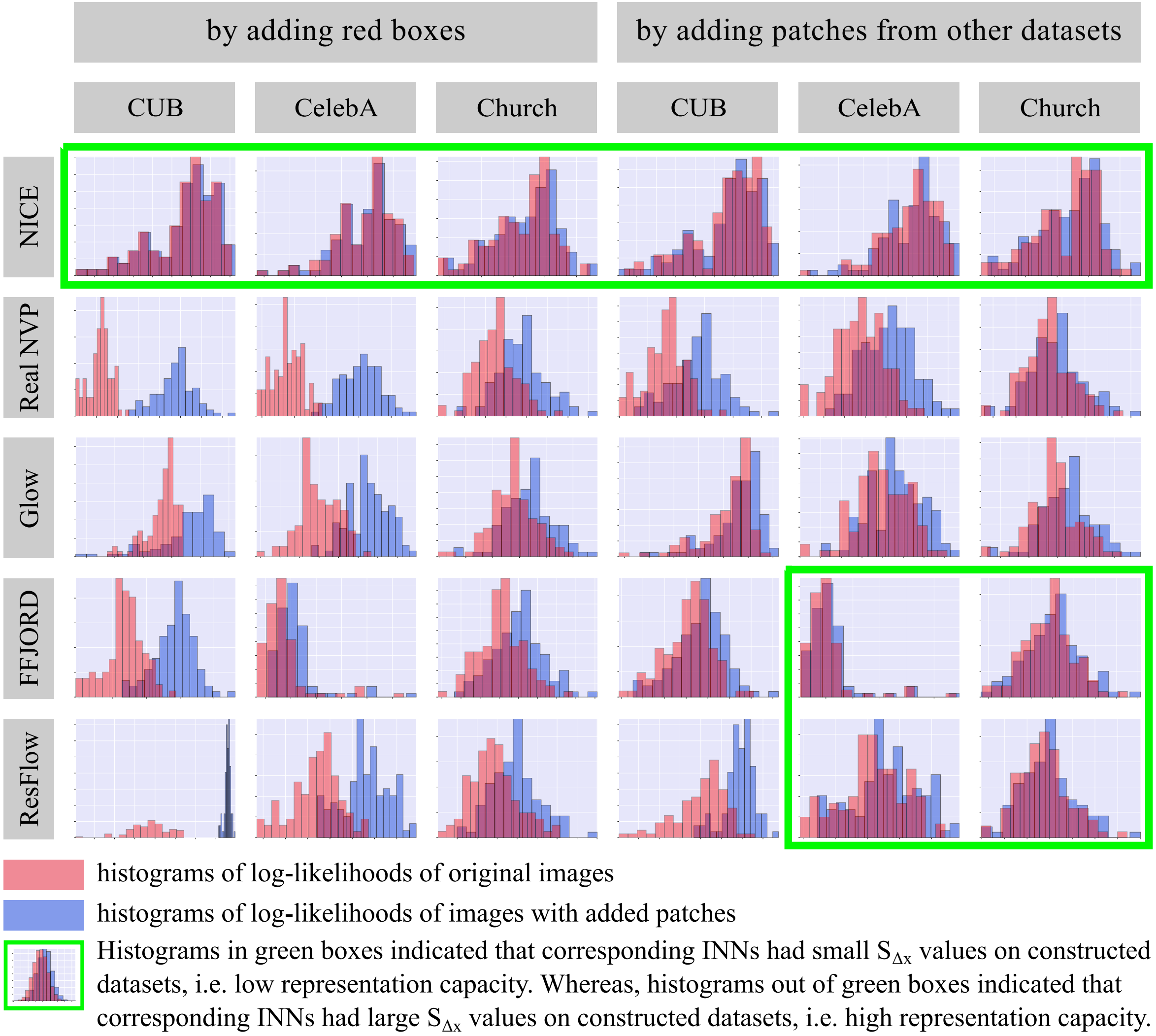


Figure 15: We used histograms of likelihood distributions images to verify the representation capacity of different INNs that were evaluated by our method. First, according to our quantitative analysis of the representation capacity based on $S_{\Delta \mathbf{x}}$, we differentiated all INNs into well-learned ones (with relatively high $S_{\Delta \mathbf{x}}$ values) and not well-learned ones (with relatively low $S_{\Delta \mathbf{x}}$ values), indicated by green boxes. Thus, in this figure, we could see well-learned INNs usually yielded significantly distinct histograms of log-probabilities between raw images and modified ones. Whereas, not well-learned INNs could not differentiate raw images from modified ones. Thus, this figure verified the effectiveness of our method.

F More results for diagnosis of INNs

F.1 The CelebA dataset

In Figure 20, Figure 16, Figure 17, Figure 18, and Figure 19, we visualized $\Delta\mathbf{x}$ extracted from the NICE, the Real NVP, the Glow, the FFJORD, and the ResFlow trained on the CelebA dataset, respectively. The green circle in each sub-figure roughly indicated abnormal concepts. In each figure, from top to bottom, we constructed abnormal concepts by adding an additional eye to the image, shuffling positions of eyes and the nose in the image, and maximizing the saturation level on the random selected region of 16 × 16 pixels, respectively. For all of three cases, regions of abnormal concepts could not be disentangled. It illustrated that the INN did not well encode these concepts.

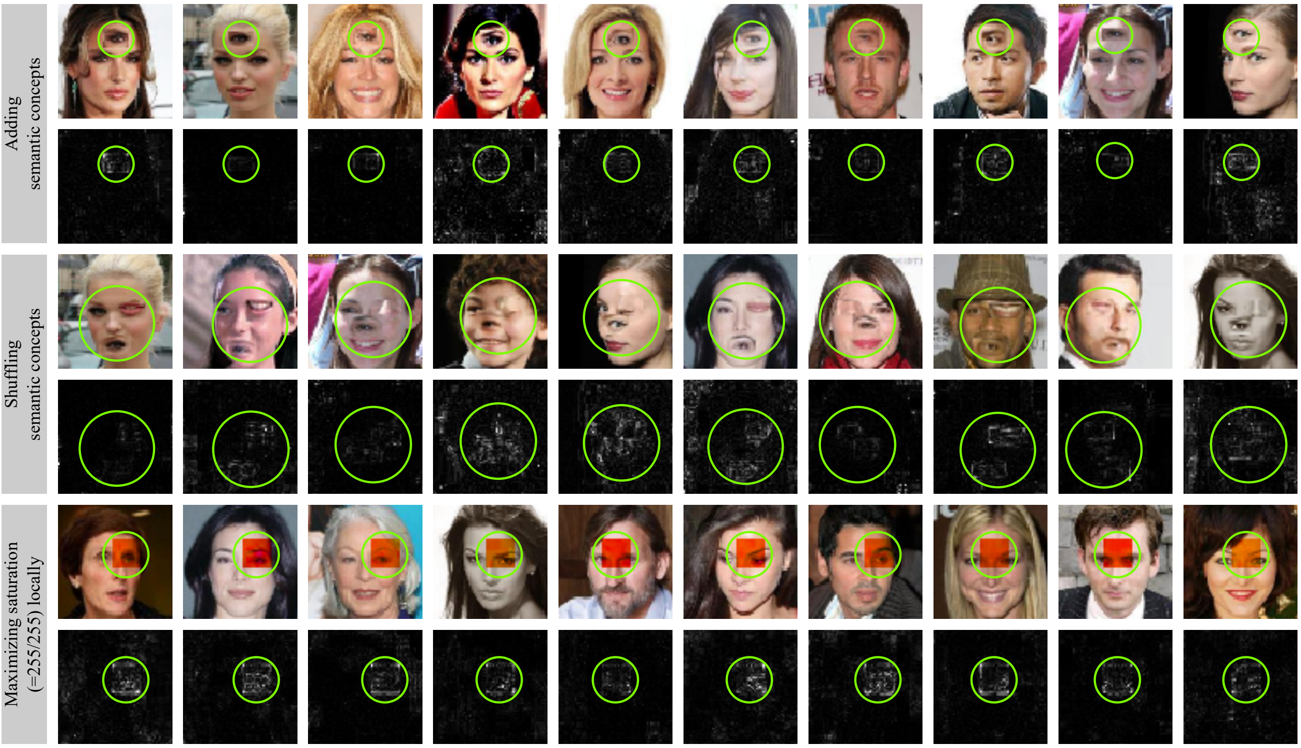


Figure 16: More results for diagnosis of the Real NVP network.

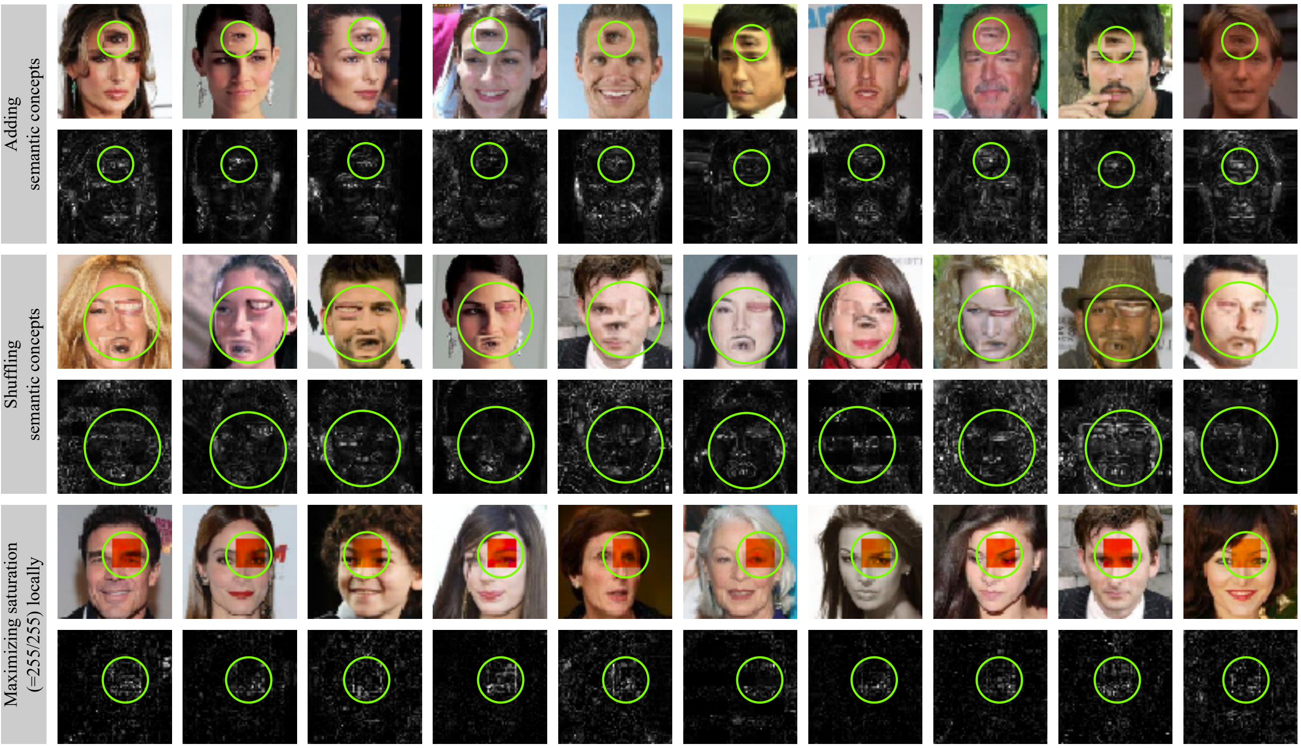


Figure 17: More results for diagnosis of the GLOW network.

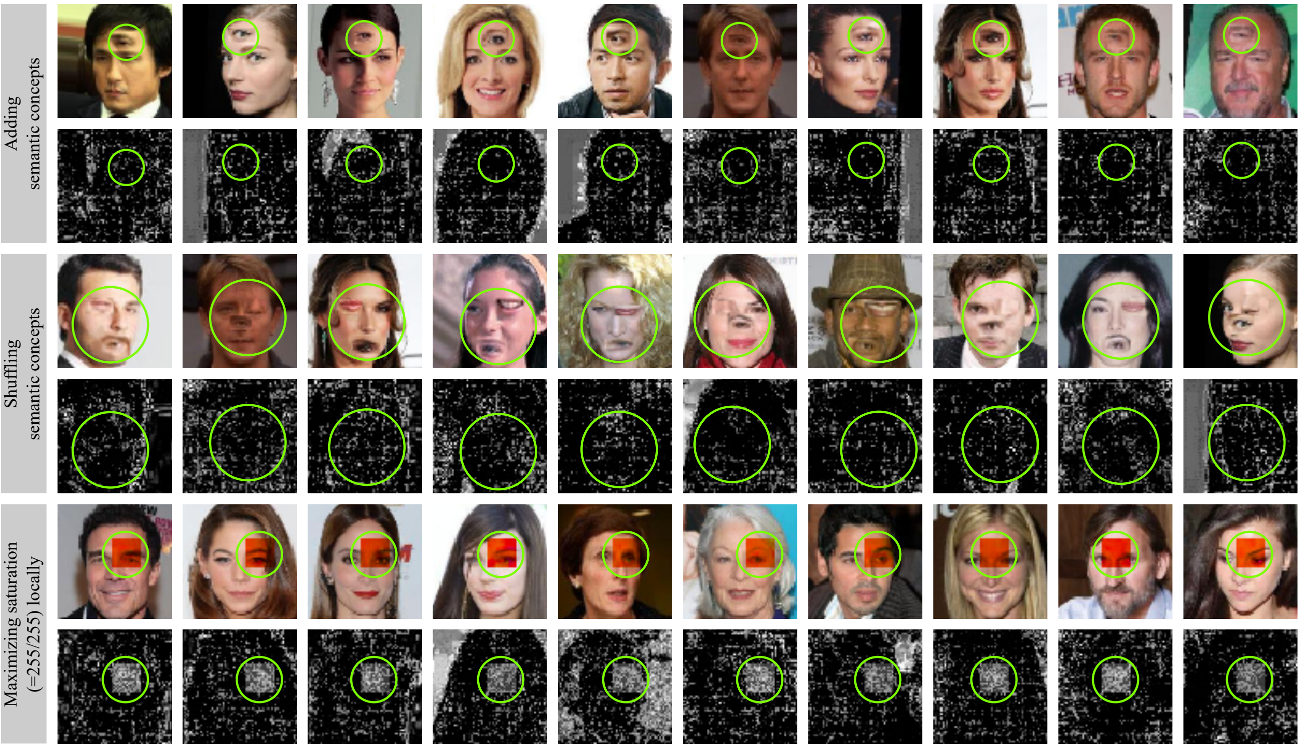


Figure 18: More results for diagnosis of the FFJORD network.

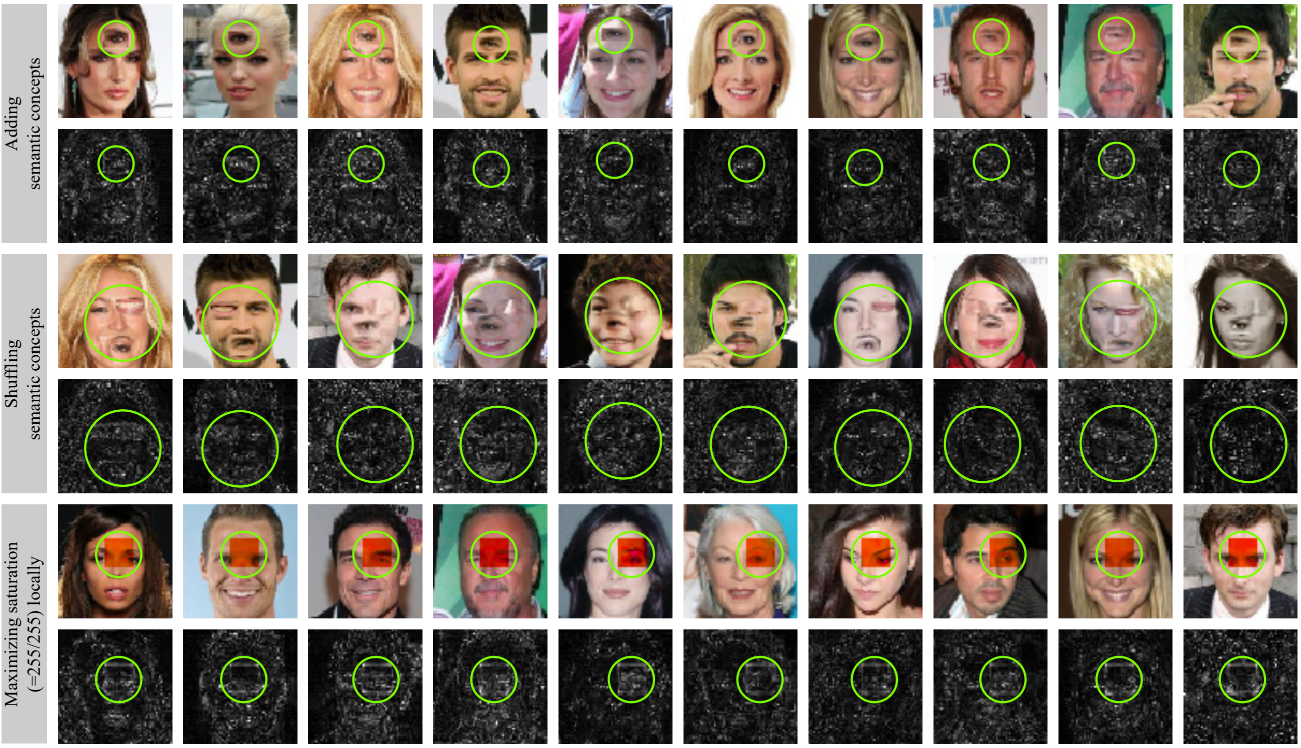


Figure 19: More results for diagnosis of the ResFlow network.

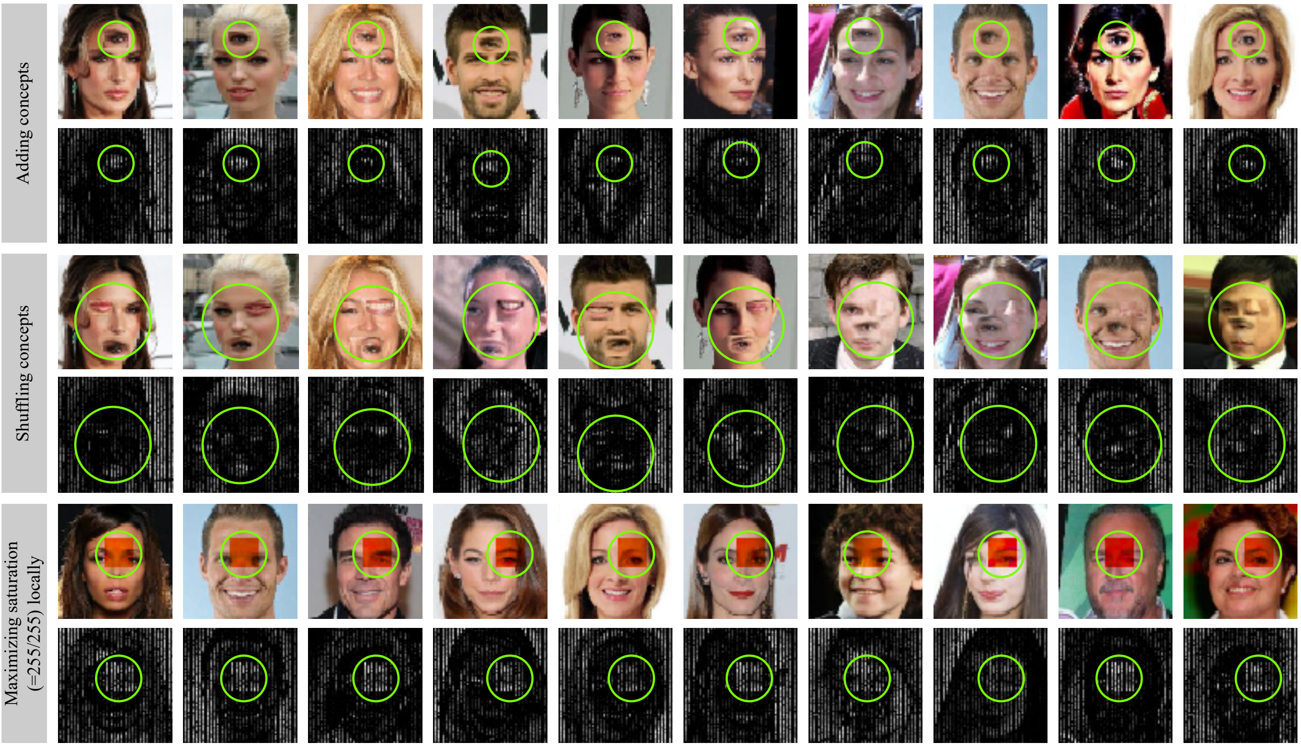


Figure 20: More results for diagnosis of the NICE network.

F.2 Church images in the LSUN dataset

In Figure 21, Figure 22, Figure 23, Figure 24, and Figure 25, we visualized $\Delta\mathbf{x}$ extracted from the NICE, the Real NVP, the Glow, the FFJORD and the ResFlow trained on the LSUN dataset, respectively. The green circle in each sub-figure roughly indicated abnormal concepts. In each figure, from top to bottom, we constructed abnormal concepts by adding an additional church to the image, shuffling positions of churches, and maximizing the saturation level on the random selected region of 16 × 16 pixels, respectively. For all of three cases, regions of abnormal concepts could not be disentangled. It illustrated that the INN did not well encode these concepts.

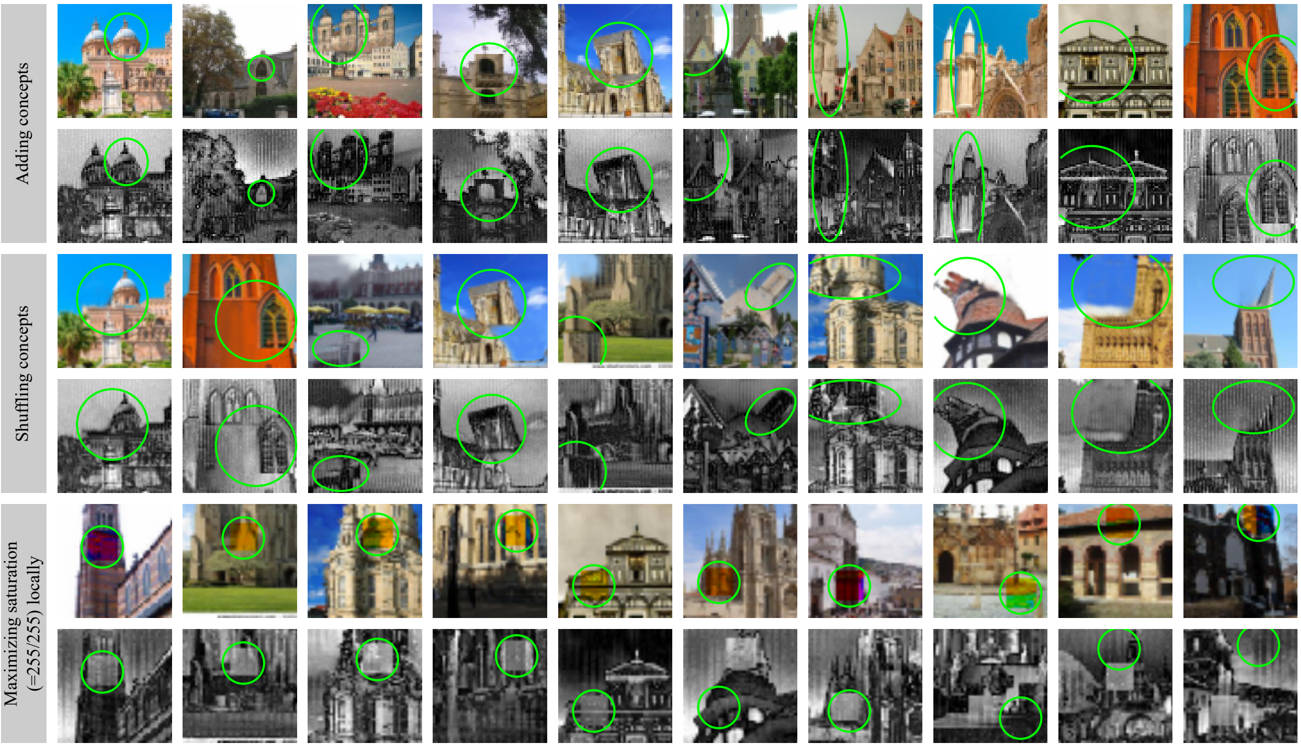


Figure 21: More results for diagnosis of the NICE network.

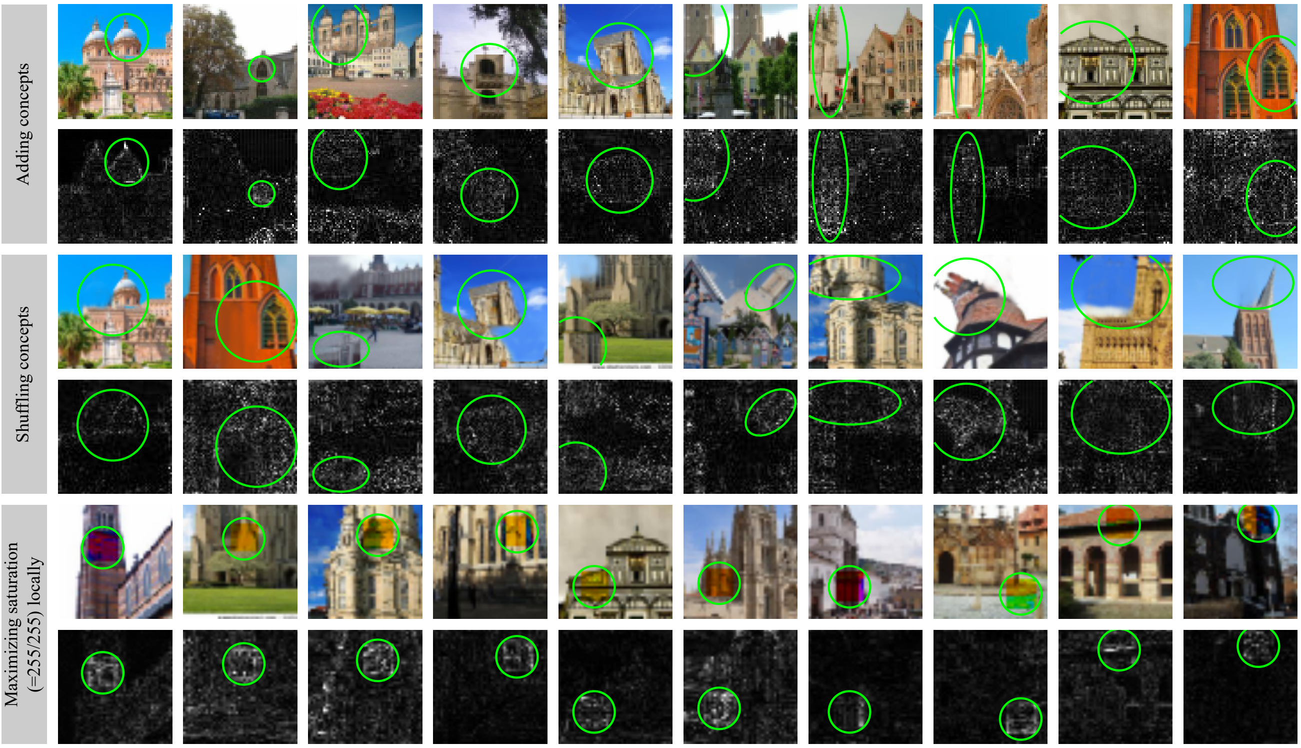


Figure 22: More results for diagnosis of the Real NVP network.

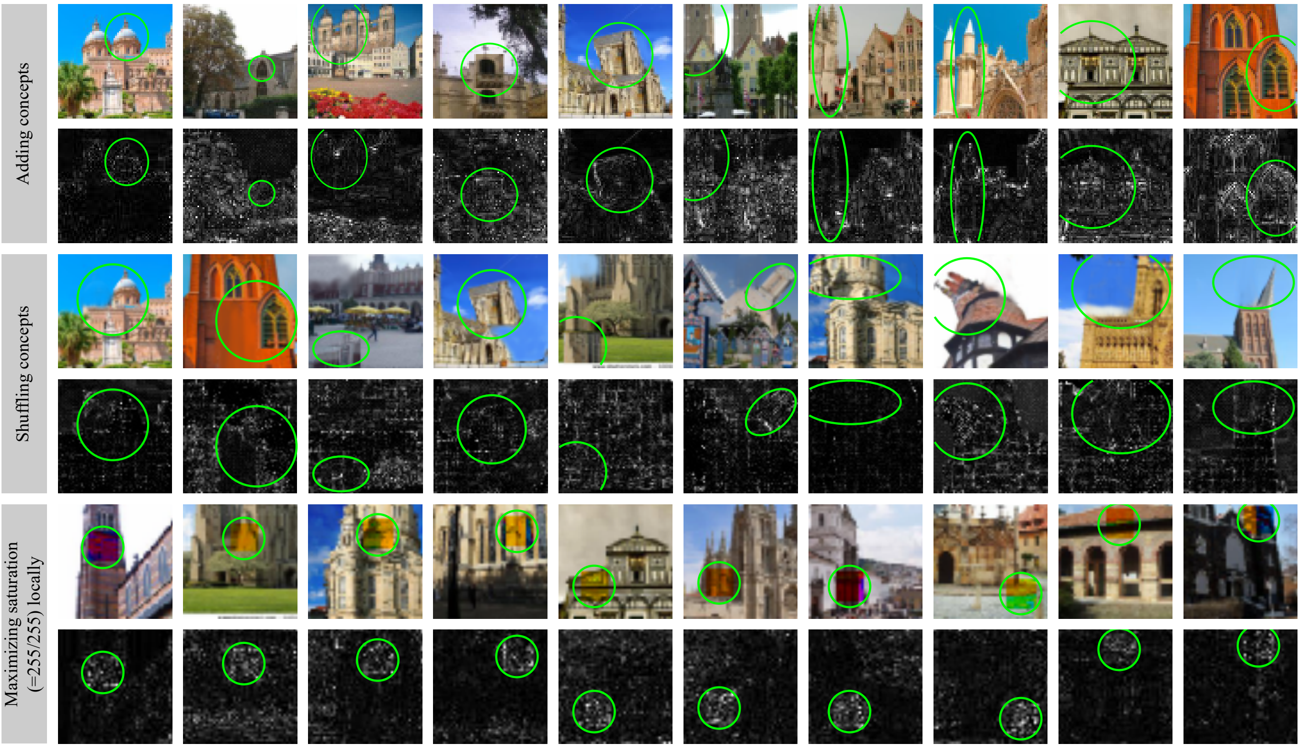


Figure 23: More results for diagnosis of the GLOW network.

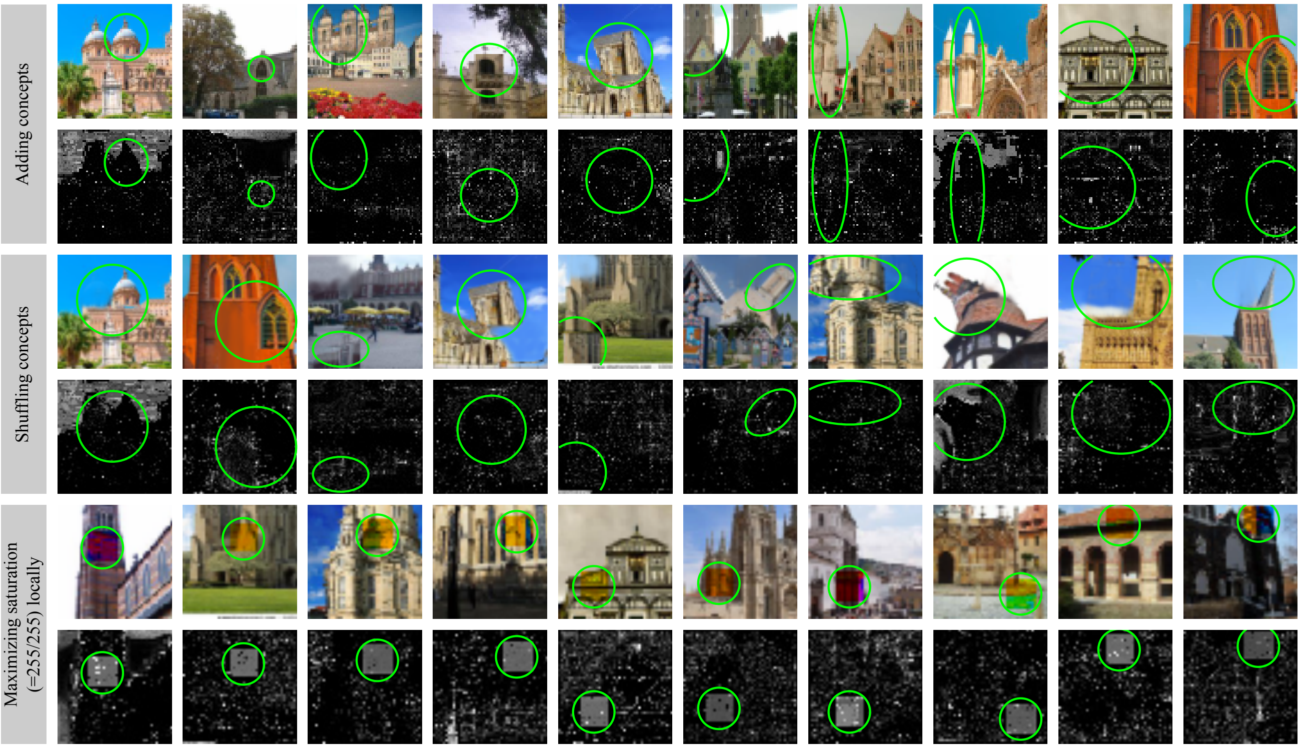


Figure 24: More results for diagnosis of the FFJORD network.

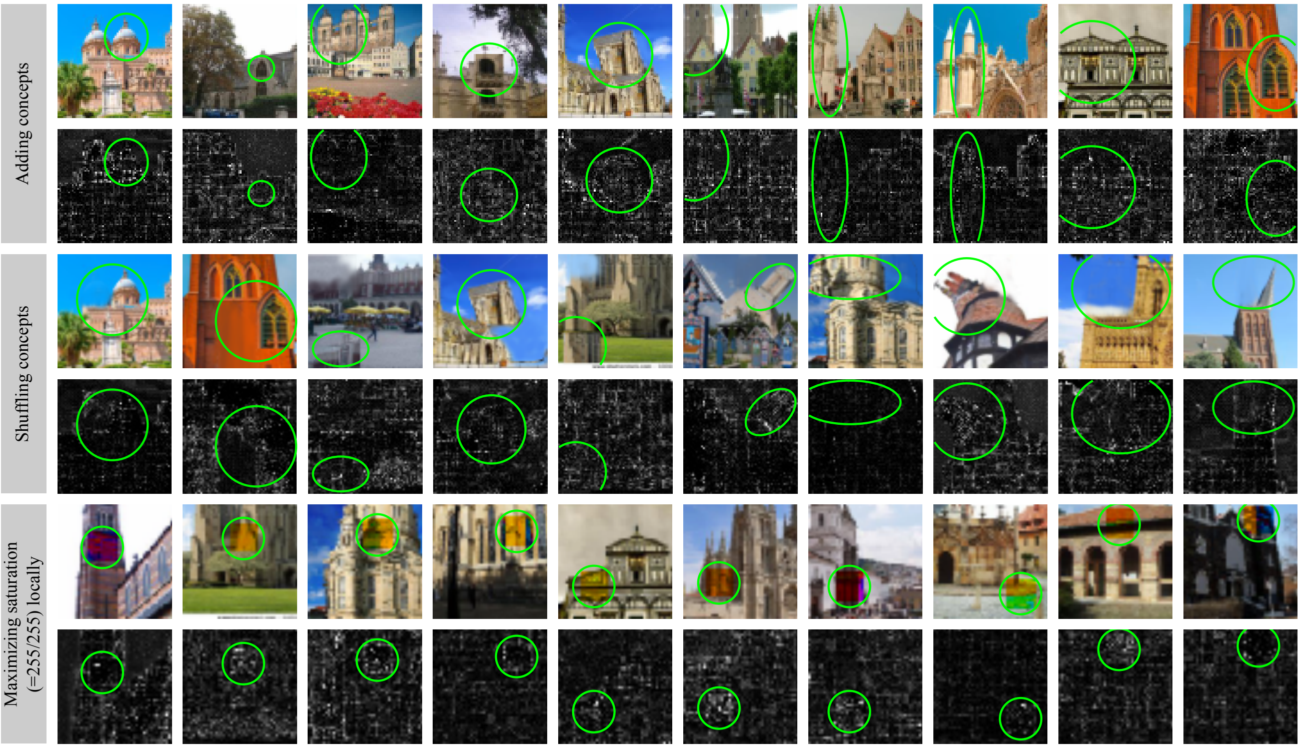


Figure 25: More results for diagnosis of the ResFlow network.

G More results for essential differences between explaining GANs and explaining INNs

Figure 26 presents differences between explanations of GANs [Bau et al., 2019] (top) and explanations of INNs (bottom). For explanations of GANs, Bau et al. [Bau et al., 2019] use a ProGAN trained on church images in the LSUN dataset. We use this method to disentangle the added red box from the image. For explanations of INNs, we use a ResFlow network (an INN, not a GAN) trained on church images in the LSUN dataset. We use the proposed method to disentangle $\Delta\mathbf{x}$ from each testing image. Results show that our method can disentangle not-well-encoded regions accurately, while the method of explaining GANs can not.

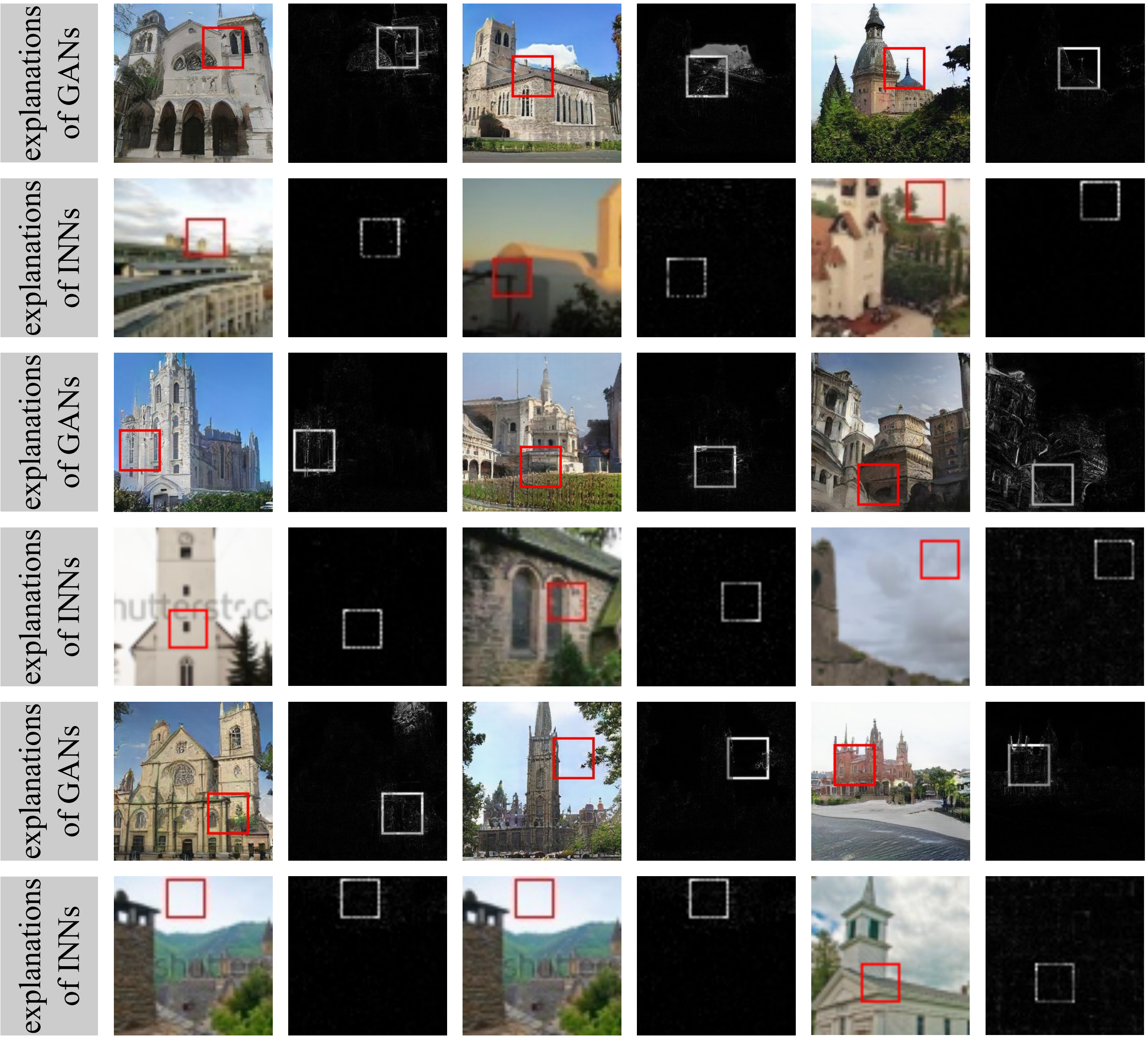


Figure 26: More results for essential differences between explaining GANs and explaining INNs.

H Ablation Study on the Size of the Mask Matrix

In this section, we conduct th ablation study of the size of the mask matrix $\mathbf{M}$. We use the proposed method with different sizes, *i.e.* 8×8 *and* 16×16, of the mask matrix $\mathbf{M}$ to disentangle $\Delta\mathbf{x}$. In Figure 27, we visualized

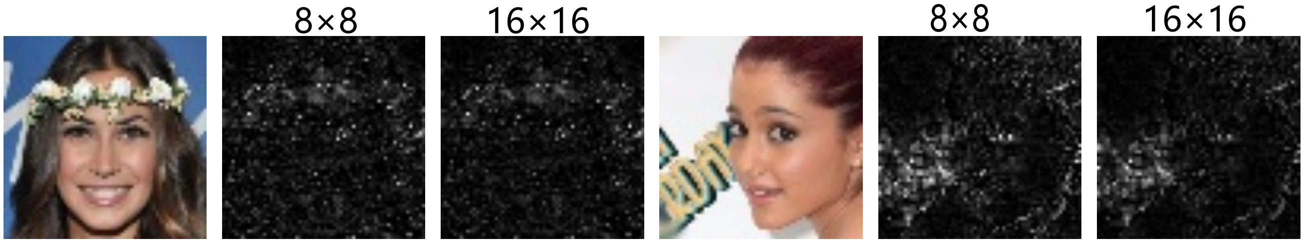


Figure 27: Ablation study on the size of the mask matrix.

$\Delta\mathbf{x}$ from the Glow trained on the CelebA dataset. The results demonstrate that the size of the matrix does not have a significant effect of the proposed method.