GLANCE: Global to Local Architecture-Neutral Concept-based Explanations

Avinash Kori †
a.kori21@ic.ac.uk

Ben Glocker †
b.glocker@ic.ac.uk

Francesca Toni †
f.toni@ic.ac.uk

† **Department of Computing, Imperial College London**

Abstract

Most of the current explainability techniques focus on capturing the importance of features in input space. However, given the complexity of models and data-generating processes, the resulting explanations are far from being 'complete', in that they lack an indication of feature interactions and visualisation of their 'effect'. In this work, we propose a novel surrogate-model-based explainability framework to explain the decisions of any CNN-based image classifiers by extracting causal relations between the features. These causal relations serve as global explanations from which local explanations of different forms can be obtained. Specifically, we employ a generator to visualise the 'effect' of interactions among features in latent space and draw feature importance therefrom as local explanations. We demonstrate and evaluate explanations obtained with our framework on the Morpho-MNIST, the FFHQ, and the AFHQ datasets.

1 Introduction

Deep learning models have emerged as powerful tools for solving complex problems in diverse domains in the past decade; however, they still remain *black-boxes* due to their lack of interpretability. There is a consensus among researchers, ethicists, policymakers and the public on the need for explainability of these models, especially in high-stake applications like bio-medicine and autonomous driving [1, 2]. Explaining decisions made by deep learning classifiers cannot only help us understand the underpinning mechanism but also uncover model biases [3], which helps in identifying issues in the data-generating process [4]. There are many different forms of explainability techniques, including feature attribution methods [5], network dissection-based interpretability [6], mechanistic approaches for understanding neural networks [7, 8], and causal/counterfactual explanations [9, 10, 11]. In this paper, we contribute to this landscape by proposing a novel method for obtaining concept-based explanations.

Interpretability can be divided into two categories [12]: transparency and *post-hoc explanations*; most of the above mentioned techniques fall under the latter category, as does our proposed framework. Many existing frameworks for post-hoc explainability do not reflect concept-based thinking of the kind exhibited by humans [13], with a few recent exceptions. [14] shows the existence of these concepts, while [15] uses the idea of both existence and interaction between concepts to generate explanations. Our proposed framework generates concept-based explanations using unobserved latent and observed context features as concepts and identifying (causal) interactions between them.

Among the different forms of explanations, counterfactual explanations are recently gaining attention [9, 10, 16, 17]. At the same time, the language of causality is advocated as a precise and powerful way of extracting explanations [18]. Counterfactual explanations can be drawn by

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

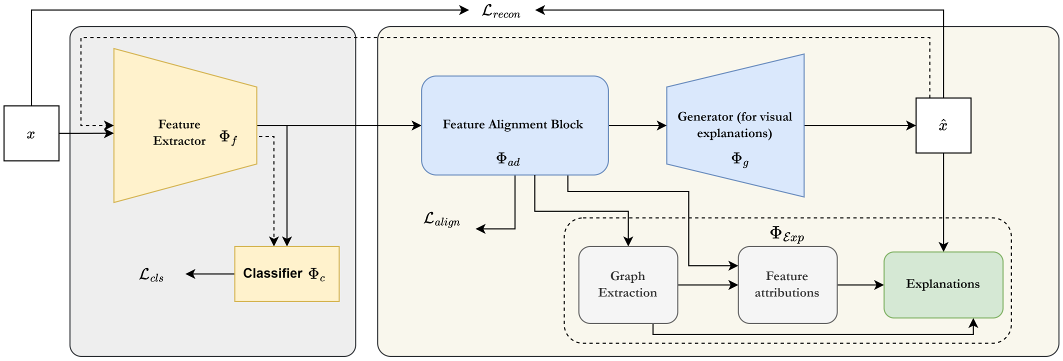


Figure 1: Overview of the proposed framework, in which the feature extractor (Φ_f) and the feature classifier (Φ_c) are blocks of a trained, given classifier model (Φ_C). The alignment (Φ_{ad}) and generator (Φ_g) blocks are part of our proposed surrogate model. The causal graph extraction, feature attribution blocks, and the generator provide our explanations.

intervening on the set of features in the data-generating process to construct hypothetical scenarios. The effectiveness of counterfactual explanations solely depends on an intuitive difference between original and intervened data. In this work, under mild assumptions, we show the existence of implicit causal knowledge by generating *causal graphs* using unobserved latent features which may or may not be human understandable and observed *context features* to model the data-generating process as perceived by the underlying model. The causal graph then serves as the basis for obtaining our explanations.

Our goal in this work is to explain what a given CNN-based classifier learns, rather than explaining the true label. We achieve this by learning a 'pseudo' data-generating process along with a feature interaction graph,[1] which serves as a *global* ground for extracting our *local* explanations (for given inputs). Note, we do not claim (nor require) any guarantees of this pseudo data-generating process to converge to the true causal process. These feature attribution based explanations respect the feature interactions in the extracted graph. The global and local explanations can be seen as two layers in a hierarchy, with the local explanations providing finer-grained information about the weight of features and their influence on the classifier's prediction while reflecting the interactions imposed by the global graph.

Our contribution in this work is threefold: (i) **Surrogate model 2.1:** we propose a novel surrogate model, aligning latent features for generating the classifier perceived data-generation process. (ii) ***Pseudo* data generating process 2.2:** we formalise a method to facilitate the causal discovery of feature interactions among unobserved latent and context features. (iii) **Explanations 2.3:** we propose a novel form of explanation that follows hierarchical steps of (i) global graph generation, capturing causal relationships as perceived by the model, and (ii) local feature attributions for a given input image along with a way to visualise and analyse feature interactions via counterfactuals.

2 GLANCE: Formalism

This section describes the methods underpinning our explainability framework, aiming to demystify a given classifier by distilling its knowledge into a surrogate model, first learning to align features, then causal discovery and generation of visual explanations. As illustrated in Fig. 1; the proposed method involves the following blocks each responsible for: (i) aligning disentangled features to observed context features, (ii) learning a generative decoder for visual explanations, (iii) constructing a causal graph, and, finally (iv) deriving explanations.

Notations. Let $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ be the dataset, such that elements of $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ and $\mathcal{Y} = \{1, \ldots, N\}$, where $H \times W \times C$ and N correspond to image resolution and total number of classes, respectively. Now, we define notations used to describe all the model components

[1] By *pseudo* we mean a data-generating process as perceived by the given classifier model.

shown in Fig. 1: (i) the pre-trained classifier is denoted by $\Phi_C : \mathcal{X} \to \mathcal{Y}$; that can be further decomposed into the feature extractor $\Phi_f : \mathcal{X} \to \mathcal{E}$ and the classifier $\Phi_c : \mathcal{E} \to \mathcal{Y}$, such that $\Phi_C(x) = (\Phi_c \circ \Phi_f)(\mathbf{X})$ for all $\mathbf{X} \sim \mathcal{X}$, (ii) the selective disentanglement and alignment block is denoted by $\Phi_{ad} : \mathcal{E} \to \mathcal{E}'$ - this block aims to learn the distribution $q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X}))$ [2], where $\mathbf{u} \in \mathcal{E}'_u$ and $\mathbf{z} \in \mathcal{E}'_z$ correspond to observed 'context' variables and unobserved latent variables; (iii) the generator block models $p(\mathbf{X} \mid \mathbf{u}, \mathbf{z})$ is used to construct visual explanations from latent space, denoted by $\Phi_g : \mathcal{E}' \to \mathcal{X}$; (iv) the graph extraction block is used to estimate causal interactions between the features over an entire dataset, described as $\Phi_{\mathcal{G}} : \{\mathcal{E}'\} \to \mathfrak{G}$, where $\mathfrak{G}$ is the space of all possible interactions and $\mathcal{G}$ is a specific instance of an estimated graph; and finally; (v) explanations block $\Phi_{exp} : \mathcal{E}' \times \mathfrak{G} \to \mathcal{A} \times \mathcal{X}$ generates feature attribution and counterfactual explanations, where $\mathcal{A} \subseteq \mathbb{R}^{|\mathcal{E}'|}$ corresponds to feature attribution space.

2.1 Surrogate Explainer Model

Our explainability framework involves learning a surrogate model with a definite set of properties: (i) alignment and (ii) an ability to generate visual explanations. Selective feature disentanglement and alignment help us construct a set of independent and dependent features responsible for the data-generating process, an alignment property transforms those obtained dependent features into semantically meaningful features (*thickness and intensity*), and the generator helps us to obtain visual explanations, explaining the *effect* of these features. Due to feature disentanglement and generator aspects, our implicit choice of model was reduced to variational auto-encoders [19] or adversarial models [20]. In our case, as the objective is to learn perceived data distribution, we condition our generator model with the classifier's features inspired by [21].

Proposal 2.1. *We propose an alignment block that aligns latent features and encodes them into two different sets: (i) observed context features ($\mathcal{E}'_u$; i.e. observed, human-understandable features in the data-generating process), and (ii) unobserved latent features ($\mathcal{E}'_z$) such that $\mathcal{E}'_u \subset \mathcal{E}'$, $\mathcal{E}'_z \subset \mathcal{E}'$ and $\mathcal{E}' = \mathcal{E}'_u \cup \mathcal{E}'_z$.*

Remark 2.2. When $\mathcal{E}'_u = \emptyset$, we treat all the latent features as unobserved and apply disentanglement loss. Availability of $|\mathcal{E}'_u| > 0$, encourages identifiability of latent variables [22].

Now, we describe the properties and assumptions considered in constructing an alignment block. Typically $|\mathcal{E}'_u| < |\mathcal{E}'_z|$, namely, the number of context features is less than the number of unobserved latent features: this makes our alignment task a problem of subspace optimisation. Based on $\mathcal{E}'$s information, we propose an alignment regularisation term with the following properties:(i) **Regularisation should involve subspace optimisation**, which makes use of observed ground-truth context features; this forces the model to encode *relations* between context features and the morphology of an image, also encouraging identifiability of latent features. (ii) **Regularisation should impose orthogonality** on features in $\mathcal{E}'_z$ among each other and also with respect to features in $\mathcal{E}'_u$; this helps in optimising all the parameters in our alignment block, while just aligning a subset of features.

Definition 2.3. The alignment of latent subspace to observed context features can be achieved by maximising the log-likelihood of estimated context distribution $\log(p(\mathbf{u} \mid \Phi_f(\mathbf{X})))$, while maintaining the orthogonality of the style features ($\mathrm{z}_i \perp \mathrm{z}_j$, $\forall \mathrm{z}_i, \mathrm{z}_j \in \mathbf{z} \ni i \neq j$).

Remark 2.4. Instead of applying disentanglement on all the latent features, the proposed alignment regularisation helps in learning relations among some while separating the rest.

Orthogonality constraint. For this, we first compute and track the running mean of eigenvectors and condition the output of the alignment block to move close towards the mean eigenvectors. In practice, we use singular value decomposition (SVD) on matrix M, where $M = M_1 \Big|\Big| M_2$ is the batch output of aligned style features ($M_z \in \mathbb{R}^{b \times |\mathcal{E}'_z|}$), where the alignment block with $M_1 \in \mathbb{R}^{b \times |\mathcal{E}'_u|}$, $M_2 \in \mathbb{R}^{b \times |\mathcal{E}'_z|}$, $\Big|\Big|$ is a concatenation operation, and b is the batch size used in training[3]. The aim of the alignment block is to force $\mathcal{E}'_z u$ to align towards the mean eigenvectors of M_1. The SVD decomposition of M_2 can be described as $U\Sigma V^* = M_2$, where U, Σ, V^*

[2] The elements of $\mathcal{E}$ are in $\mathbb{R}^l$ and elements of $\mathcal{E}'$ are in $\mathbb{R}^m$, with $m < l$ and, typically, $l, m \geq N$ - here l, m corresponds to the dimension of the latent and encoded vector.

[3] We analyse the effect of batch size on alignment later in the appendix.

correspond to left singular vectors, singular value matrix, and right singular vectors, respectively. Eigenvectors of M_2 can be computed by simply multiplying left singular vectors with a singular value matrix. To control the maximum eigenvalue of unobserved latent features, we normalise the eigenvector matrix ($U\Sigma$) with the Forbinious-norm $||.||_f$ of the singular value matrix Σ. Equation 1, describes the proposed alignment loss mathematically, where $\hat{U\Sigma}$ corresponds to the running mean of an eigenvector of matrix M_2, λ_{max} corresponds to a hyper-parameter to control the maximum eigenvalue of M_2, and α corresponds to weighage term of orthogonal conditioning. [4].

$$U\Sigma V^* = SVD(M_2),\ M_2 \in \mathbb{R}^{b\times|\mathcal{E}'_u|}; \qquad \mathcal{L}_{align} = ||M_1 - \mathcal{C}||_2^2 + \alpha\Big\|M_2 - \frac{\lambda_{max}\hat{U\Sigma}}{||\Sigma||_f}\Big\|_2^2 \quad (1)$$

Generative Block. As previously discussed, we train our generator model to distil the knowledge from the feature extractor of the given classifier. This is done so that the generated images are faithful to the classifier, *it is important to note that our generator here is reconstructing images as perceived by the classifier, not the original data distribution. The reconstructed images only contain features that the classifier sees as important in making its decision*. The joint training objective corresponds to: $\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_1\mathcal{L}_{align} + \lambda_2\mathcal{L}_{recon} + \lambda_3\mathcal{L}_{cls}$, where $\mathcal{L}_{cls}$ is cross-entropy loss applied on the classifier's prediction between original images and the classifier's perceived images, the aim of this is to distil the knowledge of classifier's decision making features in the generator. Instead of randomly sampling a noise vector for generating images, we sample features from $q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X}))$ and in addition to an adversarial loss $\mathcal{L}_{adv}$, we also use $\mathcal{L}_{recon}$ to conditional the model to learn similar looking images. All λ_i are hyper-parameters to decide the weight for each loss component. We provide hyperparameter details in appendix 4.

2.2 Perceived data-generating Process

We analyse the aligned features (ie., the output of the alignment block) to extract the learned relations among features represented as directed edges in a DAG. In this work, we do not assume causal sufficiency on the observed context features (*i.e.:* the unobserved latent features can cause or confound the observed context features). After training, we can access the *pseudo* data-generating process (generator model) as an oracle and perform controlled interventional queries. These amount to questions of the form *"How would the generated image change if this particular feature is changed?"*. We determine the existence of directed edges between features by comparing original and intervened latent feature values. Now, we define some graph specific terms, which we use in our discovery step.

Definition 2.5. Node n_i and node n_j in a DAG are said to have a *Direct Causal Path (DCP)* if there exists an edge between n_i and n_j (either $n_i \rightarrow n_j$ or $n_j \rightarrow n_i$), and are said to have an *Indirect Causal Path (ICP)* if their exists a trail from n_i to n_j via a third node n_k (either $n_i \rightarrow n_k \rightarrow n_j$ or $n_j \rightarrow n_k \rightarrow n_i$). Finally, we define the *edge-weight* for an edge between n_i and n_j as the ratio of treatment effect on a particular node $\mathsf{l}_j \in \mathbf{u}\|\mathbf{z}$ with respect to intervened variable $\mathsf{l}_i \in \mathbf{u}\|\mathbf{z}$, is formally described as $\mathsf{EW}(\mathsf{l}_i, \mathsf{l}_j) \triangleq \left(\frac{\hat{\mathsf{l}}_j}{\hat{\mathsf{l}}_i} - \frac{\mathsf{l}_j}{\mathsf{l}_i}\right)$. where l_i, $\hat{\mathsf{l}}_i$ indicate the i^{th} element in vectors $\mathbf{l}, \hat{\mathbf{l}}$, respectively (with position i corresponding to node n_i in the graph, similarly for j) and $\hat{\mathbf{l}}$ is the *intervened latent vector*, formally defined as $\hat{\mathbf{l}} \sim q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X})), \mathbf{X} \sim p(\mathbf{X} \mid \mathbf{u}, \mathbf{z}; do(\mathsf{l}_i = I))$, for I an intervention on i.

Procedurally, for causal discovery via interventional queries, we propose the following steps (for simplicity, we equate nodes and positions in vectors): 1. We extract the aligned feature vector by passing a sampled feature vector through a composite function of encoder, disentanglement and alignment, formally described as $\mathbf{l} = q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X}))$, where $\mathbf{X} \sim \mathcal{X}$; 2. Without loss of generality we select, in turn, each feature l_i in $\mathbf{l}$ and perform a fixed intervention of $\pm\mathbf{p}$ [5] to obtain $\hat{\mathbf{l}} \sim q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X})), \mathbf{X} \sim p(\mathbf{X} \mid \mathbf{u}, \mathbf{z}; do(\mathsf{l}_i = \pm\mathbf{p}))$; we then find, in $\hat{\mathbf{l}}$, all other features affected

[4] The value of α is increased gradually from 0 to 1 with respect to training iterations (based on our experiments, we found the step-based incremental function to work best)

[5] The parameter $\mathbf{p}$ is a function of latent feature, which is estimated by measuring the feature values for all data points and setting its value based on individual feature statistics (in our case, $\mathbf{p}$ = standard deviation of the selected feature w.r.t to the dataset). This ensures that the resulting generated image is in-domain w.r.t the classifier.

by this intervention and note the change in their value with respect to their original value in $\mathbf{l}$; 3. Once we establish the change in value for feature $\hat{\mathsf{l}}_j$ with respect to l_j as a result of intervention on l_i, we perform a controlled intervention on feature l_j with the observed change $\hat{\mathsf{l}}_j$ resulting in $\hat{\mathbf{l}}' \sim q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X})), \mathbf{X} \sim p(\mathbf{X} \mid \mathbf{u}, \mathbf{z}; do(\mathsf{l}_j = \hat{\mathsf{l}}_j))$, and note changes in its descendent feature values with respect to $\hat{\mathbf{l}}$; 4. We repeat the previous two steps until all the features are covered; if the relative change before and after an intervention is greater than a given threshold in an expectational sense, we establish an edge between (nodes corresponding to) those two features $(\mathbf{l}_i \to \mathsf{l}_j)$. Equation 2 describes this process, where $\mathbf{1}$ is an indicator function determining the existence of a causal link between l_i and l_j.

An edge exists between nodes l_i and l_j only if there is a difference between l_j and $\hat{l}_j$ upon intervention on l_i, conditioned on $\mathbf{pa}_{l_i}$ (parent features of l_i). Let us consider an ICP (see Definition 2.5) example, where $n_1 \to n_2 \to n_3$ and the second step establishes causal relations $n_1 \to n_2$ and $n_1 \to n_3$, with respect to some threshold τ and v_2^1, v_3^1 corresponding to new values of n_2, n_3, respectively, due to an intervention on n_1. In the third step, when we perform an intervention on n_2 by setting its value to v_2^1, let the observed value of n_3 be v_3^2; then, if $|v_3^2 - v_3^1| < \epsilon$ we establish the correct edge $n_2 \to n_3$ by removing the spurious edge $n_1 \to n_3$. In the case of loops, we use the edge weight to determine the prominent causal direction. If the features are disentangled, intervening on l_i should not affect l_j: we observe a similar effect in our experiments, which we discuss later in section 3.

$$\mathsf{l}_i \to \mathsf{l}_j \triangleq \mathbf{1}[\mathbb{E}_{\mathbf{l}\sim\mathcal{E}'}(EW(\hat{\mathsf{l}}_i, \mathsf{l}_j)|\mathbf{pa}_{\hat{\mathsf{l}}_i}) > \tau], \hat{\mathbf{l}} \sim q(\mathbf{u}, \mathbf{z} \mid \Phi_f(\mathbf{X})), \mathbf{X} \sim p(\mathbf{X} \mid \mathbf{u}, \mathbf{z}; do(\mathsf{l}_i = \mathsf{l}_i)) \quad (2)$$

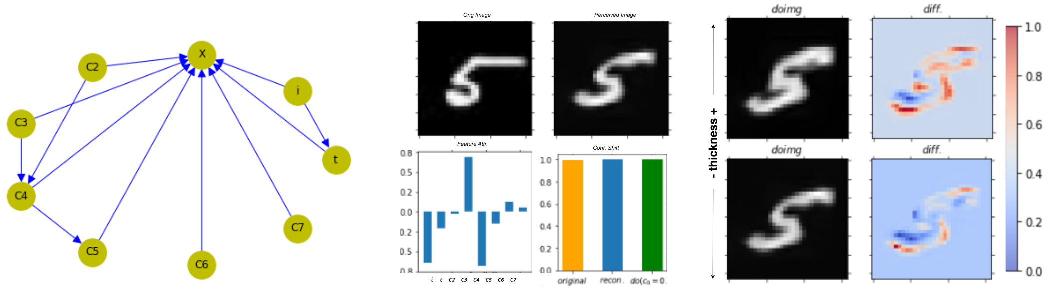
2.3 Explanations

The latent space feature vocabulary is much richer for extracting explanations beyond feature attributions and saliency maps. In contrast to importance scores and attention maps in input space, explanations based on latent features may help us analyse the model's perception of input features. Based on this, we generate globally-inspired local explanations using the feature interaction graph (extracted as described in the previous sub-section) as a global form of explanation. This feature interaction graph explains how the classifier perceives the relationships between various semantically meaningful concepts, which can reveal biases and be used to debug the classifier. For obtaining local explanations, we follow the LIME [23] feature attribution method on the aligned latent features while preserving the feature interactions (respecting the underlying global explanation), indicating the significance of all the latent features in classifying an image into a specific class. The generator model helps us visualise the effect of significant features and their interactions on a given image by constructing counterfactual samples. The feature importance, along with the interaction graph, helps users to identify features and scale of intervention to generate counterfactual images. *Note that the proposed method can adapt any feature attribution approaches for generating local explanations without loss of generality; we select LIME in this work.* Unlike global explanations, we do not possess any ground truths for the generated explanations, making the evaluation quite challenging. Here, we measure stability and faithfulness as a proxy for evaluating local explanations; we detail all the metrics in appendix **??**.

3 Results

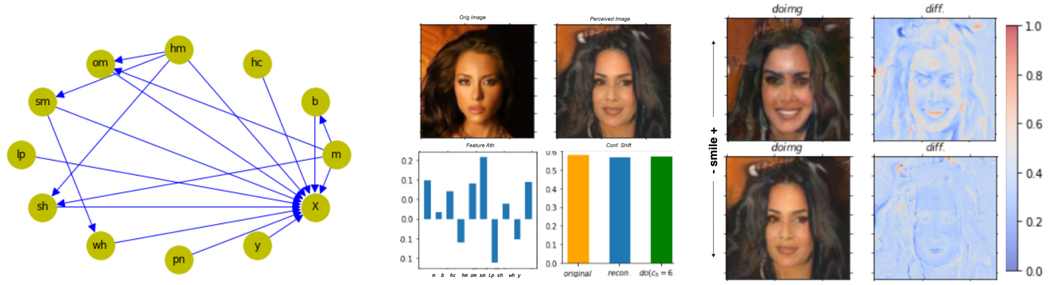
We evaluate the performance of our proposed framework for both causal discovery and explanations; we use classifiers trained on two different datasets with observed context features, namely Morpho-MNIST[24] and FFHQ[25]. As the focus in this work is to use the discoveries for explaining the reasoning process followed by a given classifier. We compare our graph generation technique against two standard methods for causal discovery, Linear Non-Gaussian Acyclic Model (LiNGAM) with latent confounders [26] and Greedy Equivalence Search (GES) [27]. We compare our explanations against saliency-based methods, LIME [5], DeepSHAP [28], deepLIFT [29], and gradCAM [30]. All the graph comparisons are described in the appendix.

In the case of Morpho-MNIST, we tested our framework on four different synthetically generated datasets by varying causal relationships among features; all four data-generating processes are described in the appendix. Fig. 5 and 2 indicate a qualitative difference between our method and the other standard methods mentioned above. Existing explanations cannot understand the



(a) (b) (c)

Figure 2: GLANCE explanations on the Morpho-MNIST-IT dataset. (a) Estimated global DAG with $< SHD >= 0.03, CI = 0.96$; (b) the first row indicates the original image and the image perceived by the classifier; the second row indicates aligned features' importance scores and effect on confidence scores with original, perceived, and intervened (w.r.t the thickness attribute) images. (c) The first column shows interventional images, and the second column shows the effect of an intervention (difference between perceived and intervened images).



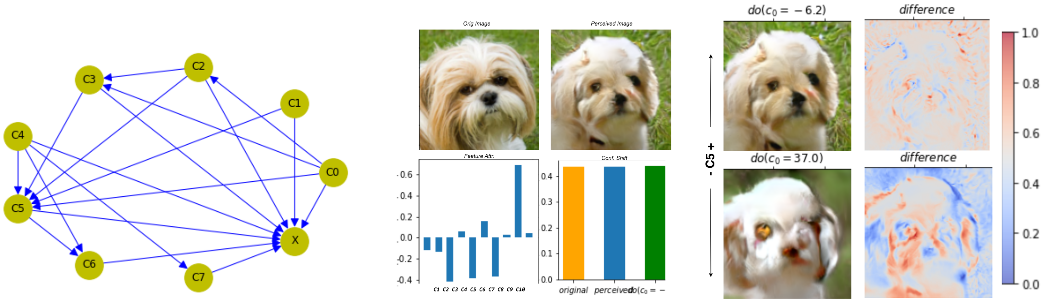
(a) (b) (c)

Figure 3: GLANCE explanation on FFHQ dataset: (a) Estimated global DAG withwith $< SHD >= 6.4, CI = 0.53$; (b) the first row indicates the original and the perceived images; the second row indicates feature importance scores and effect on confidence scores with original, perceived, and intervened (w.r.t the smile attribute) images. (c) The first column shows interventional images, and the second column shows the effect of an intervention.

effect of intermediate features or cannot differentiate the effect of multiple features involved in making specific predictions. For example, these methods cannot differentiate between the effect of thickness and intensity or geometric features. The global feature interaction graph generated by our method addresses this issue to an extent, as it captures complex feature interactions among aligned, semantically meaningful features. Qualitative results are shown in Fig. 2, 3, and 4. To quantitatively compare explanations, we make use of the *faithfulness index (FI)* and of the *stability index (SI)* described in Section 2.3. Table 1 shows the average faithfulness and stability obtained for 1000 generated explanations.

In the case of the high-resolution human faces dataset (FFHQ) [25], we explain the classifier trained to classify gender. For causal discovery, we only consider ten observed features out of forty given attributes in the dataset, selected based on the frequency of values of these features and their extent of being independent of each other (subjectively selected). A detailed list of selected features, along with additional examples, are described in appendix. Fig. 3(a) describes the generated causal structure on ten observed context features, Fig. 3(b) demonstrates a given image as perceived by the classifier along with importance scores for observed context features and the effect on confidence scores due to an intervention on the smile attribute, and Fig. 3(c) describes the effect of an intervention on the smile attribute.

In case of $\mathcal{E}'_o = \emptyset$, we consider AFHQ [31] dataset, where the objective is to segregate images into three different classes. We train our surrogate model with classifier's latent features to



(a) (b) (c)

Figure 4: (a) Demonstrates causal DAG generated by our proposed framework. (b) GLANCE explanations: the first row indicates the original image and the image perceived by the classifier; the second row indicates feature importance scores and effect on confidence scores with original, perceived, and intervened (w.r.t the $C5$ attribute) image. (c) The first column demonstrates an intervention on the perceived image and the second column corresponds to the effect of an intervention.

generate global and local explanations. Here, as there are no observed context variables, the estimated pseudo data-generating process may not be human understandable, we just denote them as concepts C_i, and select the concepts based on its node-degree and feature importance to generate counterfactual explanations. Fig. 4 demonstrates the qualitative results of the proposed framework. Additional results and ablations w.r.t different classifiers are detailed in appendix. We also layout few limitations and possible future directions in the appendix.

4 Conclusion

We present GLANCE, a novel explanation framework that uses latent space vocabulary to generate global explanations in terms of graphs and local explanations in terms of feature importance scores; then, a generator can be used to visualise the effect of feature importance and interactions. We validate both causal discovery and GLANCE explanations both qualitatively and quantitatively against existing standard explanations methods. The proposed method for extraction of global explanations (in the form of DAGs) follows carefully constructed steps using the ideas of intervention, indicating the causal interaction and influence among features in latent space. The quantification of faithfulness helps us consider explanations more carefully, and this helps us differentiate between explanations obtained from the underlying classifier model and explanations generated from data alone.

References

[1] Finale Doshi-Velez, Ryan Budish, and Mason Kortz. The role of explanation in algorithmic trust. Technical report, Technical report, Artificial Intelligence and Interpretability Working Group . . . , 2017.

[2] Joshua Alexander Kroll. *Accountable algorithms*. PhD thesis, Princeton University, 2015.

[3] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[4] Arunachalam Narayanaswamy, Subhashini Venugopalan, Dale R Webster, Lily Peng, Greg S Corrado, Paisan Ruamviboonsuk, Pinal Bavishi, Michael Brenner, Philip C Nelson, and Avinash V Varadarajan. Scientific discovery by generating counterfactuals using image translation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 273–283. Springer, 2020.

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

[6] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[7] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[8] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in.

[9] Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

[10] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

[11] Nick Pawlowski, Daniel C. Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. In *Advances in Neural Information Processing Systems*, 2020.

[12] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[13] Sharon Lee Armstrong, Lila R Gleitman, and Henry Gleitman. What some concepts might not be. *Cognition*, 13(3):263–308, 1983.

[14] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019.

[15] Avinash Kori, Parth Natekar, Ganapathy Krishnamurthi, and Balaji Srinivasan. Abstracting deep neural networks into concept graphs for concept level interpretability. *arXiv preprint arXiv:2008.06457*, 2020.

[16] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. Countergan: Generating realistic counterfactuals with residual generative adversarial nets. *arXiv preprint arXiv:2009.05199*, 2020.

[17] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T Freeman, Phillip Isola, Amir Globerson, Michal Irani, et al. Explaining in style: Training a gan to explain a classifier in stylespace. *arXiv preprint arXiv:2104.13369*, 2021.

[18] Matthew O'Shaughnessy, Gregory Canal, Marissa Connor, Mark Davenport, and Christopher Rozell. Generative causal explanations of black-box classifiers. *arXiv preprint arXiv:2006.13913*, 2020.

[19] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[21] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[22] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

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

[24] Daniel C Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-mnist: quantitative assessment and diagnostics for representation learning. *Journal of Machine Learning Research*, 20(178):1–29, 2019.

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[26] Takashi Nicholas Maeda and Shohei Shimizu. Rcd: Repetitive causal discovery of linear non-gaussian acyclic models with latent confounders. In *International Conference on Artificial Intelligence and Statistics*, pages 735–745. PMLR, 2020.

[27] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[28] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[29] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[31] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[32] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[33] Emily Denton, Ben Hutchinson, Margaret Mitchell, and Timnit Gebru. Detecting bias with generative counterfactual face attribute augmentation. 2019.

[34] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

[35] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

[36] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[37] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[38] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[39] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

[40] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[41] Dominique Mercier, Andreas Dengel, and Sheraz Ahmed. P2exnet: Patch-based prototype explanation network. In *International Conference on Neural Information Processing*, pages 318–330. Springer, 2020.

[42] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[43] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[44] Nihat Ay and Daniel Polani. Information flows in causal networks. *Advances in complex systems*, 11(01):17–41, 2008.

[45] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[46] Johann Brehmer, Pim De Haan, Phillip Lippe, and Taco Cohen. Weakly supervised causal representation learning. *arXiv preprint arXiv:2203.16437*, 2022.

Supplementary Material

A Related work

Our method falls within an active field of research on post-hoc explainability for deep learning models. Most post-hoc explainability methods can be categorized as (local or global) feature attribution-based or counterfactual explanations. Both feature attribution [23, 28] and counterfactual explanations [32, 33] have proved to be useful methods to interpret the reasons for models decisions. Feature attribution mainly focuses on estimating the importance weight for input features, indirectly indicating how features influence final decisions. In the case of images, features correspond to parts or patches of images responsible for the classifier's decisions [23]. On the other hand, counterfactual explanations are influenced mainly by hypothetical *"what-if"* scenarios. In case of generating counterfactual explanations for images, black-box models are usually explained via *surrogate* models to provide visual explanations with desired latent properties [34, 33, 32, 35, 36, 17]. Most of these methods train generators from scratch, leading to explanations that are more *faithful* to the given dataset than to the trained classifier. Some of these methods aim to generate samples that affect the classifier's decision [34, 35], while others work on changing the latent space and observing the classifier's decision change [33]. The main focus in the case of surrogate model based explanations is to extract disentangled representations [37]. As there exist infinitely many possibilities to disentangle features, it has been shown that disentanglement without supervision is a challenging problem [38]. Recent findings suggest that limited supervision can restrict the search space and can be used on a subset of latent features to optimize them to align towards desired properties [39]. Another parallel line of research focusing on concept based explanations involves [40, 41], these approaches focus on learning prototype concepts during training categorizing them into ante-hoc class of explanations.

Our work focuses on both feature attribution and the use of surrogate models to derive faithful explanations. Instead of training surrogate models from data, we distill knowledge from the pre-trained classifier, making use of a generator to depict the classifier's perceived data-generating process. We use the latent features from a the distilled model to determine the feature interactions, which is used to obtain counterfactual visual explanations and feature attributions indicating the contribution of each latent feature in towards the classifiers output.

B Assumptions

Assumption B.1. In the case of aligned features, we assume that the observed *context features* follow a Directed Acyclic Graph (DAG) structure.

Remark B.2. Here, in this work, we do not consider the data-generating processes with loops. In practice, we only consider the direction with higher edge weightage (refer 2.5).

Assumption B.3. The correctness of the entire generated graph is proportional to the correctness of the estimated subgraph with observed context features. We quantify the correctness of the subgraph by comparing it against the known ground-truth subgraph, along with stability and consistency properties.

Remark B.4. As $|\mathcal{E}'_u| < |\mathcal{E}'_z|$ generally, a direct way to validate a generated graph would be via visual inspection, which may prove to be challenging in case of large graphs; in that case, we can use the subgraph of observed features and compare against ground-truth.

C Metrics

Graph correctness: To measure the correctness of the estimated graph ($\mathcal{G}_{est}$), we measure the structural hamming distance (SHD) [42] between the estimated graph with respect to the truth graph. Along with the SHD scores we measure the properties like consistence and stability with respect to the true graph($\mathcal{G}_{gt}$). In this case the notion of stability captures the variation of a generated graph when the method is applied to different subsets of a dataset, while consistency captures the variation in the generated graph when the method is applied to the same data multiple times.

As the classifiers are trained to select *minimal feature set* required for estimating decision boundary, due to this the interactions between features represent perceived data generating process rather than actual data generating process. To capture the graph properties of perceived data generating process we define the different variant of SHD score, not penalising for the missing edges, formally described in C.1.

In practice, to capture both stability and consistency properties, We sample P random subsets with repetition from the test dataset and run Q iterations of graph generation on each set. These properties are analogous to aleatoric and epistemic uncertainties [43]. The behaviour of the estimated graphs along Q iterations measures the consistency of our method also capturing the notion of epistemic uncertainty, while graph generation behaviour across all P subsets of data measures the stability or captures the notion of aleatoric uncertainty. We measure average SHD over all PQ iterations, denoted by $< SHD >$.

Definition C.1. Graph Correctness Index: To quantify the correctness of the perceived data generating process, we compare the edges in the generated subgraph with the known ground-truth graph and consider the average over all PQ graphs. *correctnessIndex*(CI) is formally defined in 3:

$$CI(\mathcal{G}_{est}, \mathcal{G}_{gt}) \triangleq \frac{1}{PQ} \sum_{P} \sum_{Q} \frac{\#CE - \#AE}{\#TE} \tag{3}$$

where $\#CE, \#AE$, and $\#TE$ correspond to the total number of correct edges predicted in a subgraph, wrong edges predicted in a subgraph and total number of ground-truth edges, respectively.

Remark C.2. An edge with a wrong direction is considered as an additional edge, so the defined metric accounts for both wrong directions and additional edges, while not penalising for missing edges. Both CI and $< SHD >$ are inversely related, *i.e.*, higher the value of CI better the estimated discoveries, while its otherway around for $< SHD >$. Both $< SHD >$ and CI are unnormalised, $< SHD > \in [0, 2^K]$ and $CI \in [-\frac{2^K}{\#TE}, 1.0]$, where K corresponds to the total number of nodes in a graph.

Definition C.3. Stability: We consider explanations to be stable if they are consistent across multiple iterations for the same image. To quantify stability, we perturb an image sample with Gaussian noise generating P samples to obtain Q local explanations, one for each sample. We then consider a negative average of the variance in all the local explanations as stability. Formally,

$$SI(exps) \triangleq -\frac{1}{P} \sum \mathbb{E}_{x \sim exps}((x - \mathbb{E}(x))^2)$$

Where $exps \in \mathcal{A}$ or $exps \in \mathcal{X}$ is a set of Q explanations for one of the P samples. The negative sign makes the metric directly proportional to the stability of explanations.

As our method follows the surrogate model, explanations are a function of both data and classifier. We characterise explanations to be faithful if the contribution of the classifier is higher than the contribution of data in generating a particular explanation. We follow an information-theoretical approach to measure the flow of information [18] to quantify faithfulness. Proposition C.5 provides a quantitative metric.

Definition C.4. Information Flow: The *information flow* between two independent sets of nodes A and B ($\mathcal{I}(A \to B)$) [44] is:

$$\int \int \mathcal{P}(a)\mathcal{P}(b \mid do(a)) \log \frac{\mathcal{P}(b \mid do(a))}{\int_{a'} \mathcal{P}(a')\mathcal{P}(b \mid do(a'))da'} dbda \tag{4}$$

where $do(v)$ represents an intervention that fixes the value of a variable to v irrespective to its parents and $\mathcal{P}$ is a probability distribution.

Proposition C.5. *Based on the above Definition C.4, and with the reference to framework DAG in Figure* **??***, we show that the bounded mutual information between* $\mathcal{E}'$ *and* $\mathcal{E}xps$ *is the same as the information flow from the classifier to the generated explanations. Due to this, we consider the normalised mutual information as the 'faithfulness' metric, given by* $\left(FI(\mathcal{E}xps, \mathcal{E}') = \frac{\mathcal{I}(\mathcal{E}'; exps)}{\sqrt{\mathcal{H}(exps)\mathcal{H}(\mathcal{E}')}}\right)$*, where* $\mathcal{H}(.)$ *corresponds to entropy.*

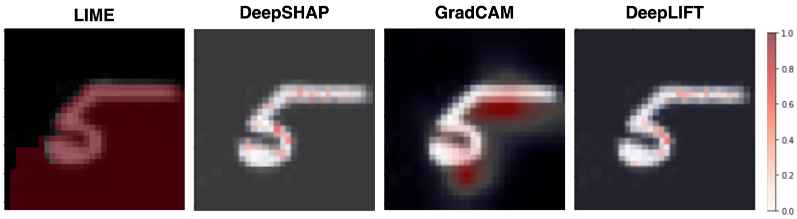


Figure 5: Demonstrates the obtained explanations from existing standard approaches, LIME, DeepSHAP, GradCAM, and DeepLIFT explanations from left to right respectively.

Proof. Here, we show the bounded mutual information (normalized mutual information) can be considered as a 'faithfulness' metric to quantify the classifier's contribution to generating explanations.

Let us consider the feature attribution based method, probability of generating explanation $\mathcal{E}xp \in \mathcal{E}xps$ can be formally described by a conditional $\mathcal{P}(\mathcal{E}xp \mid l), l \in \mathcal{E}'$.

$$\mathcal{I}(\mathcal{E}' \rightarrow \mathcal{E}xps) =$$

$$\int\int \mathcal{P}(l)\mathcal{P}(\mathcal{E}xp|do(l)) \log \frac{\mathcal{P}(\mathcal{E}xp|do(l))}{\int \mathcal{P}(l')\mathcal{P}(\mathcal{E}xp|do(l'))dl'} d\mathcal{E}xpdl$$

As integrals are applied over entire space, intervention can be replaced by conditionals which simplifies the above equation as:

$$\mathcal{I}(\mathcal{E}' \rightarrow \mathcal{E}xps) =$$

$$\int_l\int_{\mathcal{E}xp} \mathcal{P}(l)\mathcal{P}(\mathcal{E}xp \mid l) \log \frac{\mathcal{P}(\mathcal{E}xp \mid l)}{\int_{l'} \mathcal{P}(l')\mathcal{P}(\mathcal{E}xp \mid l')dl'} d\mathcal{E}xpdl$$

$$\Rightarrow \int_l\int_{\mathcal{E}xp} \mathcal{P}(l, \mathcal{E}xp) \log \frac{\mathcal{P}(\mathcal{E}xp \mid l)}{\mathcal{P}(\mathcal{E}xp)} d\mathcal{E}xpdl$$

$$\Rightarrow \int_l\int_{\mathcal{E}xp} \mathcal{P}(l, \mathcal{E}xp) \log \frac{\mathcal{P}(\mathcal{E}xp, l)}{\mathcal{P}(\mathcal{E}xp)\mathcal{P}(l)} d\mathcal{E}xpdl$$

$$\therefore \mathbb{I}(\mathcal{E}' \rightarrow \mathcal{E}xps) = \mathcal{I}(\mathcal{E}'; \mathcal{E}xps) \propto \frac{\mathcal{I}(\mathcal{E}'; \mathcal{E}xps)}{\sqrt{\mathcal{H}(\mathcal{E}xps)\mathcal{H}(\mathcal{E}')}}$$

In the case of a counterfactual method, explanations $\mathcal{E}xps$ are a part of input data ($\mathcal{E}xps \in \mathcal{X}$). Without loss of generality, we can apply the same metric to quantify the explanation's faithfulness to a classifier.

□

D Network Architectures

In this work, we mainly use three different types of architectures as a pre-trained classifier/feature-extractor for GLANCE framework. We use standard DenseNet-121 [], ResNet-18 [], and VanillaCNN. For VanillaCNN we use 7 convolutional blocks, where each block is a sequence of 3×3 convoltuional layer followed by batch normalisation layer and ReLU non-linearity. To reduce dimensionality we apply max pooling layer after the first, third, and fifth layers. The final convolutional layer activations are further global average pooled and projected on to appropriate vector space for generating class probabilities using a single densely connected linear layer followed by softmax non-linearity.

Table 1: Quantitative comparison between multiple explanation methods with respect to faithfulness and stability properties.

Metrics ↓ Methods →	LIME	Deep SHAP	Deep LIFT	Grad-CAM	Ours
$FI(\mathcal{E}xps, \mathcal{E}')$	0.22	0.67	0.92	0.22	**0.97**
$SI(\mathcal{E}xps)$	-1.40	-0.07	-0.04	-2.53	-**0.02**

E Case Study 1: Morpho-MNIST

Here, we consider explaining a model trained on synthetic data based on MNIST digits [24]. We define multiple data-generating process with four different variables thickness, width, slant, and intensity, and observe how our proposed method retrieves this causal structure using latent information via controlled interventions. In this setup thickness corresponds to the stroke thickness of a digit, width corresponds to the total width of a written digit, slant corresponds to the shear factor along a horizontal direction, and intensity corresponds to the average intensity of pixels in a digit. Functions $SetIntensity(\mathring{u}; i)$, $SetSlant(\mathring{u}; s)$, $SetWidth(\mathring{u}; w)$, and $SetThickness(\mathring{u}; t)$ refer to the operations applied to original MNIST digit to generate new image x with desired properties by controlling image morphology. Below we formally define 4 different data-generating senarios, Fig. 6 pictorially demonstrates causal structure used in data-generating performance and our model performance.

Morpho-MNIST-TI: In this setting we consider two causal variables thickness and intensity, where thickness causes intensity. Mathematically the functional relationship between variables are defined as described in equation 5.

$$\begin{aligned}
&t := f_t \triangleq 0.5 + \epsilon_t \quad \epsilon_t \sim \Gamma(10, 5) \\
&i := f_i \triangleq 64 + 191 * \sigma(2 * w + 5) + \epsilon_i \quad \epsilon_i \sim \mathbb{N}(0, 1) \\
&x := f_x = SetIntensity(SetThickness(X; t); i)
\end{aligned} \tag{5}$$

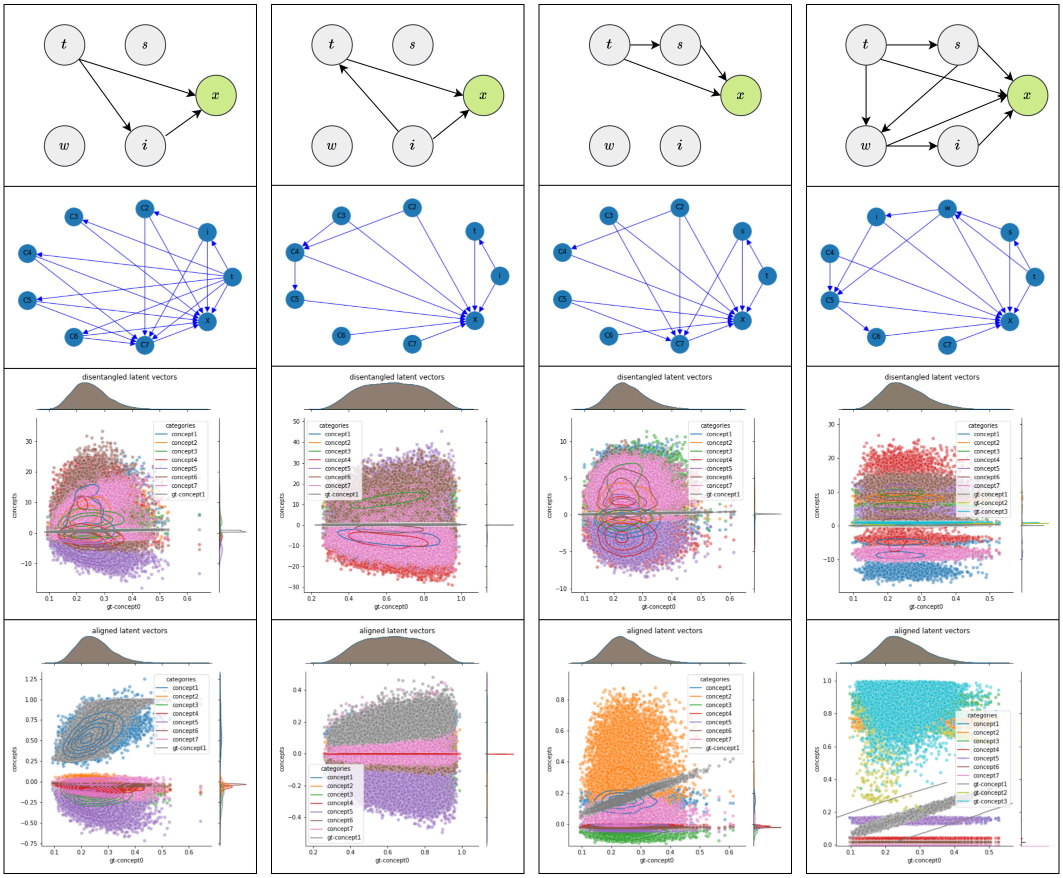
Morpho-MNIST-IT: In this experiment we inverted a directionality from previous setting resulting in intensity to cause thickness, which is mathematically described in equation 6

$$\begin{aligned}
&i := f_i \triangleq \epsilon_i \quad \epsilon_i \sim \mathbb{U}(60, 255) \\
&t := f_t \triangleq 3 + \sigma(i/255) + \epsilon_s \quad \epsilon_s \sim \mathbb{N}(0, 0.5) \\
&x := f_x = SetThickness(SetIntensity(X; i); t)
\end{aligned} \tag{6}$$

Morpho-MNIST-TS: In this setup we use thickness and slant as causal attributes, where thickness causes digit slantness, which is formally described in equation 7

$$\begin{aligned}
&t := f_t \triangleq \epsilon_t \quad \epsilon_t \sim \Gamma(0, 5) \\
&s := f_s \triangleq 10 + 5 * \sigma(2 * t - 5) + \epsilon_s \quad \epsilon_s \sim \mathbb{N}(0, 0.5) \\
&x := f_x = SetSlant(SetThickness(X; t); s)
\end{aligned} \tag{7}$$

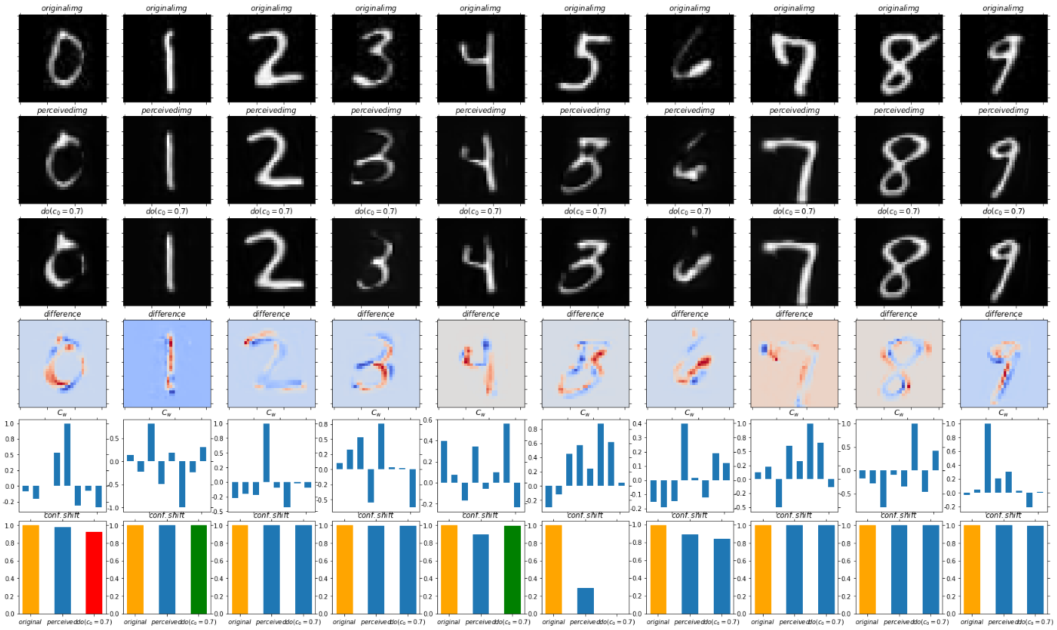
Morpho-MNIST-TSWI: In this setup we increased a complexity by using intensity, thickness, slant, and digit width as a causal attributes, where thickness causes slant, thickness and slant causes width, and width causes intensity. This data-generating process is formally described in equation 8



(a) (b) (c) (d)

Figure 6: Causal discoveries on various different data-generating processes. Top row describes the causal relationships followed in data-generating process, second row shows the discoveries made by our proposed method, third row shows the cluster formed by a feature disentanglement block to describe an alignment effect, fourth row describes the effectiveness of our alignment block in aligning model latent features to observed context features. In (a) thickness causes intensity and both thickness and intensity causally affects image, the same behaviour can be observed in generated causal graph with graph *correctnessIndex=1.0*. In (b) intensity causes thickness, we selected this example to examine algorithms behaviour in the case of a reversed causal link(w.r.t (a)), and the correct behaviour is observed in the generated graph with *correctnessIndex=1.0*. In (c) thickness causes slant, again the generated graph shows similar behaviour with *correctnessIndex=0.98*. In (d) we tried adding multiple causal variables with higher relationship depth, and our proposed method was able to reconstruct these relationships with *correctnessIndex=0.94*.

$$
\begin{aligned}
t &:= f_t \triangleq \epsilon_t \quad \epsilon_t \sim \Gamma(0,5) \\
s &:= f_s \triangleq 10 + 20 * t + \epsilon_s \quad \epsilon_s \sim \mathbb{N}(0,5) \\
w &:= f_w \triangleq 10 + 15 * \sigma(0.5 * t) - 0.25 * s + \epsilon_w \\
&\qquad\qquad \epsilon_w \sim \mathcal{N}(0,1) \\
i &:= f_i \triangleq 64 + 191 * \sigma(w/25) + \epsilon_i \quad \epsilon_i \sim \mathbb{N}(0,1) \\
x &:= f_x = SetIntensity(SetWidth(\\
&\qquad SetSlant(SetThickness(X;t);s);w);i)
\end{aligned}
\tag{8}
$$



(a) First row shows an original image, second row describes reconstructed images, while third shows the effect of intervention (intervention on *thickness* t node), followed by difference between intervention and perceived images, and feature importance as perceived by classifier. Last row corresponds to feature attribution explanations on extracted features in latent space.

Figure 7: Following figure demonstrates the effectiveness of our proposed explanations and other standard existing explanation frameworks

Fig. 6 row 2 describes the explicit graph generated as a result of our framework. In all the cases subgraph with nodes $\in \{t, i, w, s, x\}$ matches precisely with the causal structure followed in our data-generating process, with a graph *correctnessIndex* close to 1.0. This indicates the existence of implicit mechanisms and causal structures, providing global explanations for a given data-generating process.

To explain the importance of each feature for a classifier, we perform a LIME feature attribution while preserving the causal structure. We perform a fixed interventional study on all the features by constructing multiple counterfactual images and observing the shift in confidence scores predicted by the classifier with respect to the original image, indicating the effect of features on the given classifier. If confidence increases, we claim that a specific feature has a positive effect on a classifier; otherwise, it negatively affects a classifier.

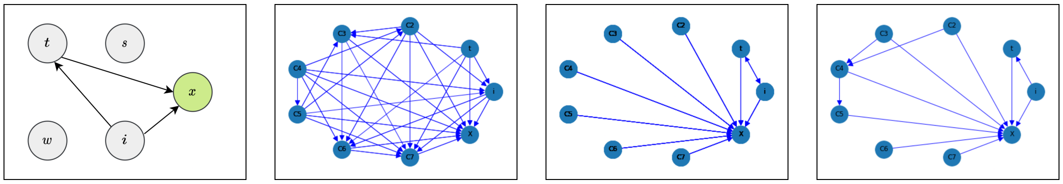
As interventions may not have a monotonic effect, we conduct two specific queries, one positively increasing feature value while the other reducing the feature value. Fig. 7 shows the generated counterfactual and the classifier probability describing the importance of a specific positively intervened feature. Based on the extracted graph, an interventional behaviour of a feature on an image, and each feature's contribution to the final classifier's decision, we get a comprehensive idea of the classifier's reasoning.

E.1 Pre-trained Classifiers

For MNIST case study we use custom architecture consisting 7 convolutional layers with 3×3 kernels followed by batch normalisation layer and ReLU non-linearity. The final layer logits are further global average pooled and projected on to appropriate vector space for generating class probabilities using a single densely connected linear layer followed by softmax non-linearity. To reduce dimensionality we apply max pooling layer after the first, third and fifth layers. We train this classifier for 50 epochs with a batch size of 64. We use Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.001. We achieve 99.3% accuracy with this pre-trained classifier.

Datasets ↓ \ Methods →	LinGAM Based [26]	GES Based [27]	Ours
Morpho-MNIST (TI)	0.84	0.66	**1.0**
Morpho-MNIST (IT)	0.66	0.66	**1.0**
Morpho-MNIST (TS)	0.82	0.66	**0.98**
Morpho-MNIST (TSWI)	0.58	0.42	**0.94**

Table 2: Table describes quantitative comparison between causal multiple causal discovery methods on four different versions of Morpho-MNIST dataset as described in 6 using graph *correctnessIndex*.



(a) Ground-truth sub graph

(b) GES

(c) LiNGAM

(d) Proposed

Figure 8: (Quantitative and qualitative) Comparison of causal discovery methods. The scores for GES, LiNGAM, and Proposed method are ($CI = 0.66, < SHD >= 3.24$), ($CI = 0.66, < SHD >= 1.17$), and ($CI = 0.96, < SHD >= 0.03$) respectively.

E.2 Comparative study

E.2.1 Graph Generation

Most of the existing causal discoveries method try to extract relationships between nodes by assuming a particular structure of models. However, in our case, since we have access to an implicit causal model, we consider trained models as an oracle to perform specific interventions. As we observe feature behaviour against an actual cause rather than a model hypothesis, we have the flexibility to extract feature relations without any explicit assumptions. Fig. 8, describes the discoveries made by these two methods using latent features obtained from a model trained using Morpho-MNIST data, where intensity causes thickness. We use the graph correctness index defined in 2.2 to quantify the performance difference between all three methods; table 2 describes the results.

E.2.2 Explanation

As previously mentioned, In this study we compare our method against standard saliency based explanation methods, we consider LIME [5], DeepSHAP [28], deepLIFT [29], and gradCAM [30] explanation and compare them against our explanations. These methods generate an attention map for an input image given the model's prediction confidence on that image. These explanations can provide a simple understanding of what the network is looking at in making certain decisions, but they fail to understand complex feature interactions or even fail to capture relations between pixels in input space. These explanation methods do not yield a way to quantify the faithfulness of their generated explanations, which raises the question of trust in the explanations themselves. Explanations generated using our method can overcome this kind of issue. Fig. 7, demonstrates our framework performance on multiple images, while Fig. 6 shows the behavior of intermediate latent features.

F Case Study 2: FFHQ

For the second case study, we consider the high resolution human faces dataset (FFHQ) [25], this dataset consists of approximately 200k images of 128x128 resolution with 40 different binary

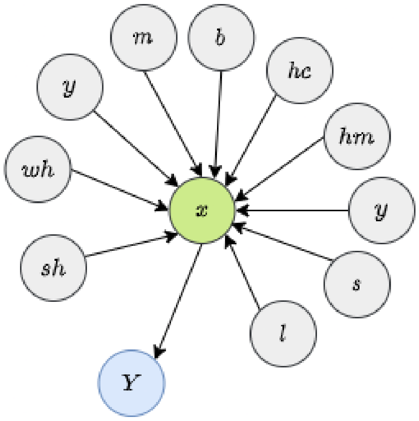


Figure 9: Assumed true causal structure between attributes and image.

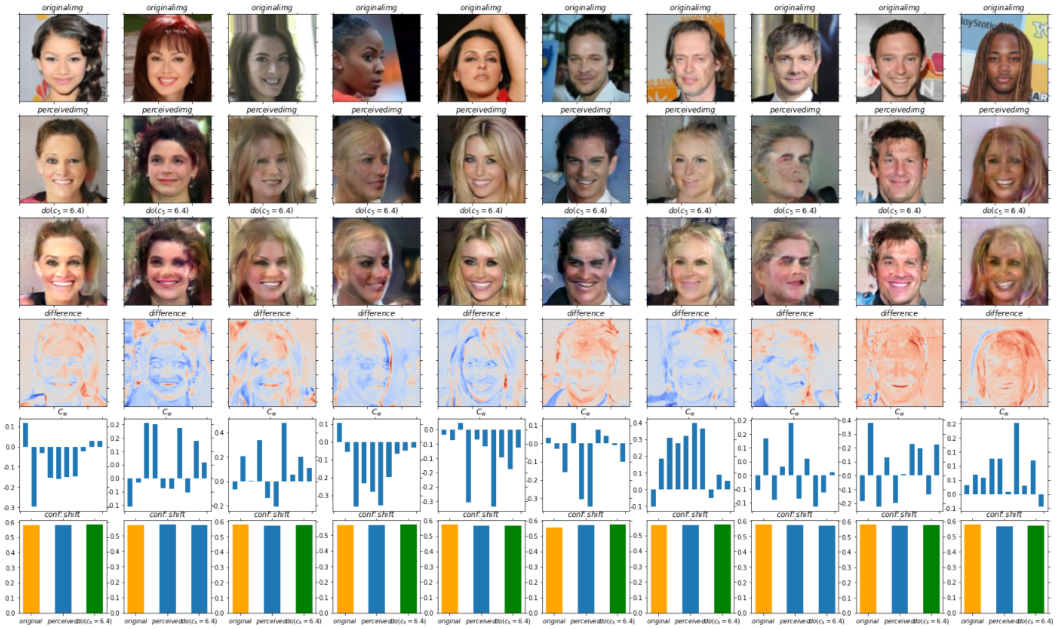


Figure 10: First row shows an original image, second row describes reconstructed images, while third shows the effect of intervention (intervention on *open mouth* o node), followed by difference between interventional and perceived images, and feature importance as perceived by classifier. Last row corresponds to feature attribution explanations on extracted features in latent space

attributes, and the task is to categorize images based on gender (0 = male; 1=female). As the causal structure is unknown, we consider only ten significantly present attributes in the dataset. In our experiment, these attributes are subjectively based on their interactions with respect to other attributes. We pick features that are seemingly orthogonal to one another because that helps us to assume ground-truth causal structure to follow naive Bayes structure with all the selected features. The ten attributes which we used in our experiment include (*sh: straight-hair, wh: wavy-hair, y: young, m: mustache, b: beard, hc: high-cheekbones, hm: heavy-makeup, s: smiling, l: lipstick, o: open-mouth*), and the structure of assumed ground truth DAG is described in Fig. 9.

For causal discovery and explanations, we consider 512 latent features to capture all the information in the data distribution. Obtained explanations from our method for the classifiers trained on this dataset are described in Fig. 10. Here, we consider 'smile' as an interventional attribute; the higher attention around the mouth region can be easily seen in different images, indicating the effect of smile intervention. In this current work, we faced challenges with generating high quality counterfactuals. In the future, we are planning to extend this work with auxiliary modules to learn and associate causal attributes with generating high quality and meaningful counterfactuals.

F.1 Pre-trained Classifier

For evaluating our framework on high resolution images, we consider the standard DenseNet-121 [45] architecture and train the model on FFHQ dataset, with the aim of gender classification. The training utilizes Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.005. The model is trained for 64 epochs achieving 98% accuracy.

To validate the framework behavior on different feature extractors we consider, three different pre-trained classifiers (i) vanilla feature extractor, (ii) DenseNet-121 architecture, and (iii) ResNet-18 architecture. All the models were trained with previously defined hyperparameters. Table 3 describes the classifier performance and the resulting evaluation metrics.

Table 3: Here, we describe the accuracy of the pre-trained classifier, FID measuring the generator quality (we report mean and variance across 5 random seeds), faithfulness index, and stability index for evaluating generated explanations for FFHQ dataset.

Feature Extractor(↓), Metrics(→)	Accuracy (↑)	FID(↓)	Faithfulness (FI ↑)	Stability (SI ↑)
Vanilla-CNN	91.37 ± 0.87	34.82 ± 2.69	0.61	-0.012
ResNet-18	94.80 ± 0.22	24.67 ± 1.43	0.71	-0.008
DenseNet-121	96.87 ± 0.38	22.51 ± 1.41	0.70	-0.005

G Case Study 3: AFHQ

G.1 Pre-trained Classifier

In case of AFHQ all the images were resized to the dimension of 128x128, we made use of DenseNet-121 [45] architecture and train the model on AFHQ dataset, with the aim of classifying images into 'cat', 'dog', and 'wild' category. The training utilizes Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.005. The model is trained for 64 epochs achieving 97% accuracy.

To validate the framework behavior on different feature extractors we consider, three different pre-trained classifiers (i) vanilla feature extractor, (ii) DenseNet-121 architecture, and (iii) ResNet-18 architecture. All the models were trained with previously defined hyperparameters. Table 4 describes the classifier performance and the resulting evaluation metrics.

H Training

We trained all our models on a system with GPU: Nvidia Telsa T4 16GB, CPU: Intel(R) Xeon(R) Gold 6230, and RAM of 384GB. In case of Morpho-MNIST, images were resized to 32×32 and

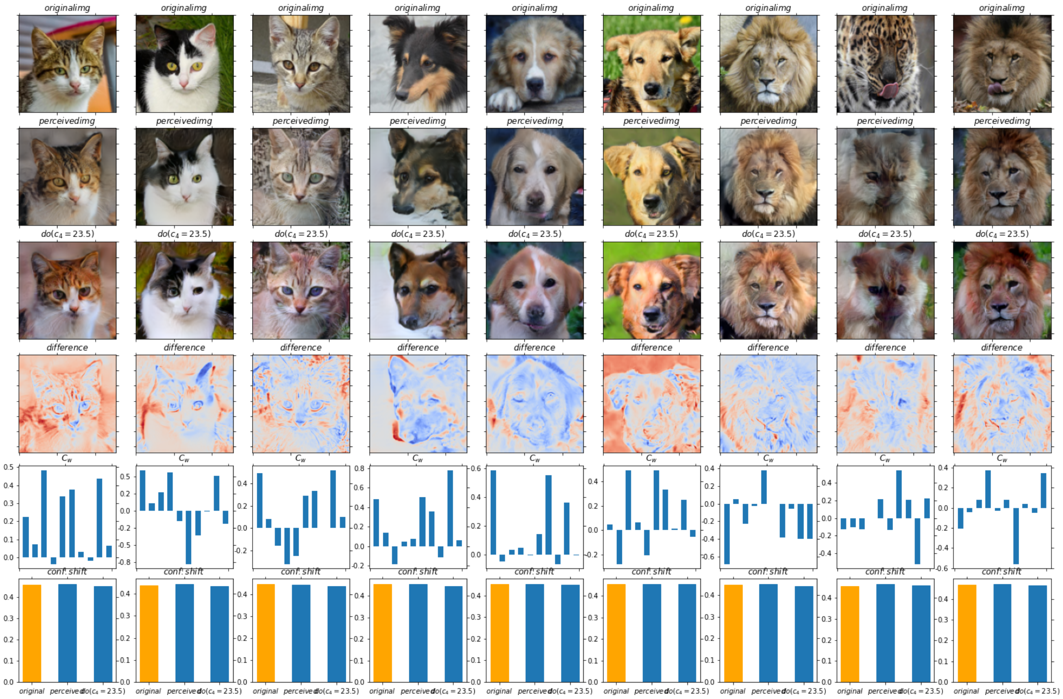


Figure 11: First row shows an original image, second row describes reconstructed images, while third shows the effect of intervention (intervention on $C3$ node, seems to be like a contrast change), followed by difference between interventional and perceived images, and feature importance as perceived by classifier. Last row corresponds to feature attribution explanations on extracted features in latent space

Table 4: Here, we describe the accuracy of the pre-trained classifier, FID measuring the generator quality (we report mean and variance across 5 random seeds), faithfulness index, and stability index for evaluating generated explanations for AFHQ dataset.

Feature Extractor(↓), Metrics(→)	Accuracy (↑)	FID(↓)	Faithfulness (FI ↑)	Stability (SI ↑)
VanillaCNN	88.86 ± 0.66	46.50 ± 2.05	0.66	-0.03
ResNet-18	98.78 ± 0.27	34.33 ± 1.67	0.71	-0.04
DenseNet-121	99.11 ± 0.13	28.38 ± 1.39	0.72	-0.01

models were trained with batchsize of 32 with learning rate = 1e-3, $\lambda_1 = 10., \lambda_2 = 30., \lambda_3 = 30.0, \&\lambda_4 = 1.0$. In case of AFHQ, images were resized to 128×128 and models were trained with batchsize of 16 with learning rate = 2e-4, $\lambda_1 = 10.0, \lambda_2 = 40.0, \lambda_3 = 80.0$, and $\lambda_4 = 1.0$.

I Future Work

This paper opens several avenues for future work. Differently from [14], we do not consider assigning semantic meaning to unobserved aligned features: careful investigation can bring out interesting evidences on the learned latent representation. Also, our framework depends on context features for extracting human understandable data-generating process: it would be interesting to perform unsupervised or weekly supervised causal representation learning [46] to estimate the causal dependencies among latent variables. The extension of GLANCE to accommodate unsupervised causal representations and assigned semantic meaning to latent features would effectively allow communicating the decision making parameters in classifiers to humans, thereby increasing the quality of explanations. Finally, extending the framework to explain other models

Table 5: Ablation results by varying loss coefficients on MorphoMNIST-IT dataset.

λ_1 (alignment loss)	0	1.0	1.0	10.0	10.0
λ_2 (recon. loss)	1.0	0	1.0	50.0	40.0
λ_3 (cls loss)	1.0	1.0	0	100.0	80.0
λ_4 (pl in adv loss)	1.0	1.0	1.0	1.0	1.0
FID	10.21	13.63	7.30	5.46	**3.33**

beyond classifiers for visual data, such as time series, text, or tabular data, can broaden the impact of our framework.