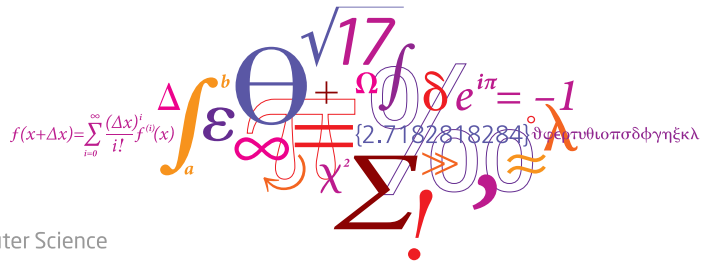
Explainability as statistical inference

Hugo Henri Joseph Senetaire[1], Damien Garreau[2], Jes Frellsen[1], Pierre-Alexandre Mattei[2]

[1]Technical University of Denmark (DTU)

[2] Université Côte d'Azur, Inria



DTU Compute
Department of Applied Mathematics and Computer Science

Model complexity is exploding

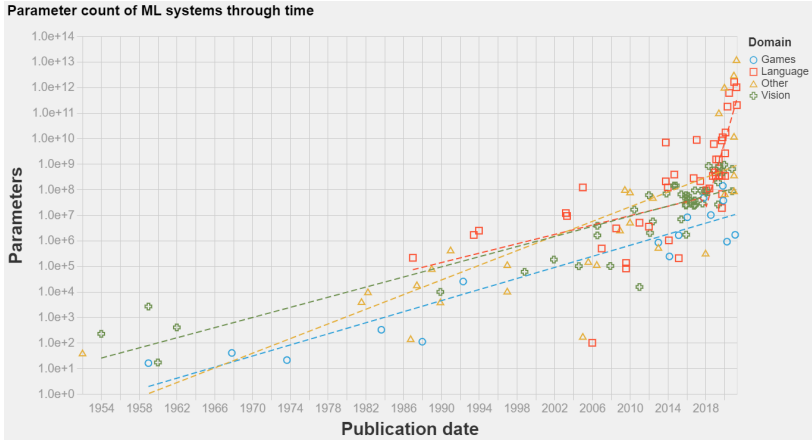


Figure: Sevilla (2021)

Need for inherently interpretable models

- Nowadays, explanation is separate from the prediction task. The explanation is made a posteriori on a black-box model.
- There is a growing need for self-interpretable models (Caruana, 2019; Rudin, 2019)

Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

Friends Don't Let Friends Deploy Black-Box Models: The Importance of Intelligibility in Machine Learning

Richard Caruana
Microsoft
rcaruana@microsoft.com

What properties are required for an inherently interpretable model ?

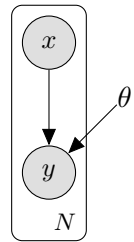
- We want to frame interpretability and prediction as a single statistical learning problem.
- We want to be able to turn any network architecture into an explainable one.
- The explanation must be easily understandable.

Traditional non interpretable supervised learning

We have access to a dataset $(\mathbf{x}_1,\dots,\mathbf{x}_N) \in \mathcal{X}$ and labels $y_1,\dots,y_N \in \mathcal{Y}$ where $\mathcal{X} = \prod_{d=1}^{D} \mathcal{X}_i$ a D-dimensional feature space and $\mathcal{Y}$ the target space following a distribution $p(x,y)$.

$p_\theta(y|x) = \Phi(y|f_\theta(x))$.

- $f_\theta : \mathcal{X} \to \{0,1\}$ is a neural network denoted the predictor.
- Φ parametric family of density (Bernoulli for binary classification, Gaussian for regression).



A very standard way to infer θ is by maximizing the log-likelihood :

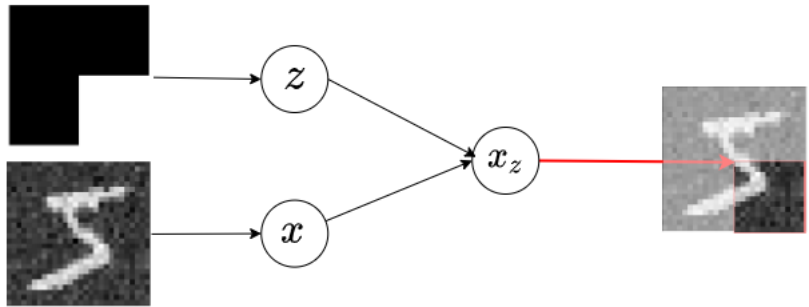
$$\max_\theta \mathcal{L}(\theta) = \sum_{n=1}^{N} \log p_\theta(y_n|x_n).$$

We will turn this model into an interpretable one.

Easily readable intepretability with a binary mask

Let's incorporate a latent variable $Z \in \{0,1\}^D$ that correspond to a subset of selected feature or a mask.

$$z_j = \begin{cases} 1 & \text{if Feature } j \text{ used for prediction,} \\ 0 & \text{if Feature } j \text{ unused for prediction.} \end{cases}$$

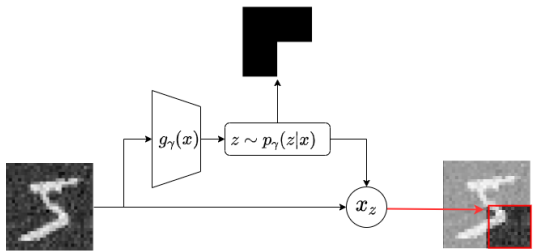


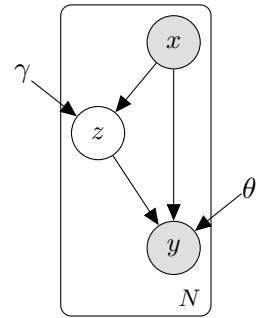
Latent variable as explanation

We endow this latent variable Z with a distribution $p_\gamma(z|x)$, parameterized by a neural network $g_\gamma : \mathcal{X} \to \{0,1\}^D$ called the selector.

This leads to the following factorisation of the complete model:

$$p_{\theta,\gamma}(y, x, z) = p_\theta(y|x, z)p_\gamma(z|x)p(x)$$



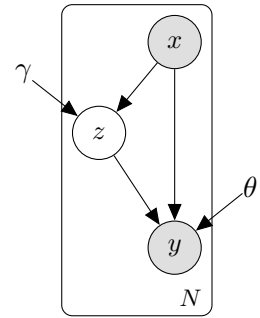


Turning off Features

$$p_{\theta,\gamma}(y, x, z) = p_\theta(y|x, z)p_\gamma(z|x)p(x)$$

We want to define $p_\theta(y|x, z)$ with some specifications:

- We want to be model agnostic $\rightarrow$ we can't change f_θ.
- We could want to make use of an already pre-trained classifier $\rightarrow$ we need to feed f_θ with something with the same dimension



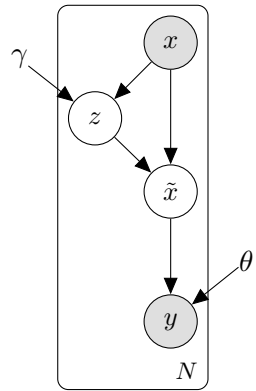
Turning off Features

We want to define $p_\theta(y|x,z)$ with some specifications:

- We want to be model agnostic $\rightarrow$ we can't change f_θ.
- We could want to make use of an already pre-trained classifier hence we need to be in the same domain and have the same input dimension.

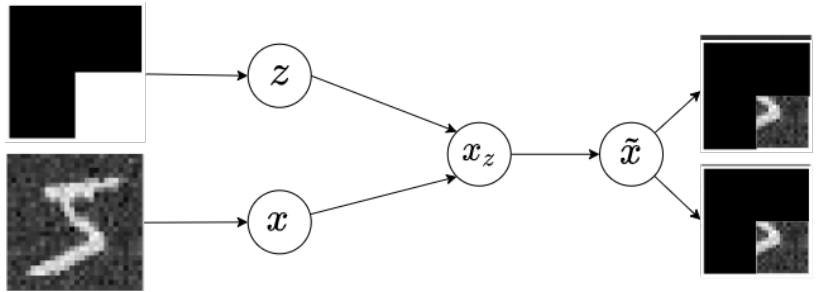
$\rightarrow$ replace the missing values by sampling from an imputation distribution p_ι.

$$p_\theta(y|x,z) = \int_{\tilde{x}} p_\theta(y|\hat{x}) p_\iota(\tilde{x}|x,z) d\tilde{x} = \int_{\tilde{x}} \Phi(y|f_\theta(\tilde{x})) p_\iota(\tilde{x}|x,z) d\tilde{x} \quad (1)$$



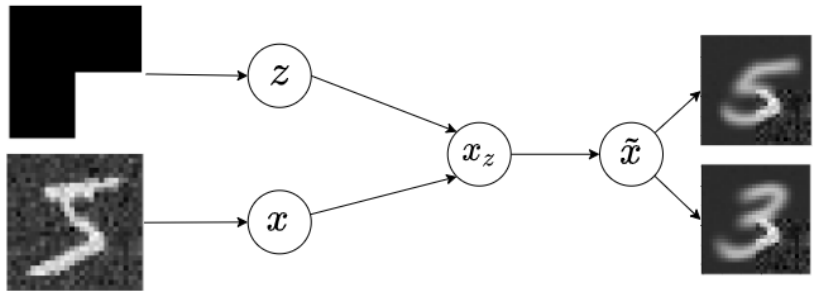
Single Imputation

$$p_\theta(y|x,z) = \int_{\tilde{x}} p_\theta(y|\hat{x}) p_\iota(\tilde{x}|x,z) d\tilde{x} = \Phi(y|f_\theta(x \odot z + (1-z) \odot c) \quad (2)$$



Multiple Imputation

$$p_\theta(y|x,z) = \int_{\tilde{x}} p_\theta(y|\hat{x}) p_\iota(\tilde{x}|x,z) d\tilde{x} = \int_{\tilde{x}} \Phi(y|f_\theta(\tilde{x})) p_{data}(\tilde{x}|x_z) d\tilde{x} \quad (3)$$



Using a surrogate f_θ

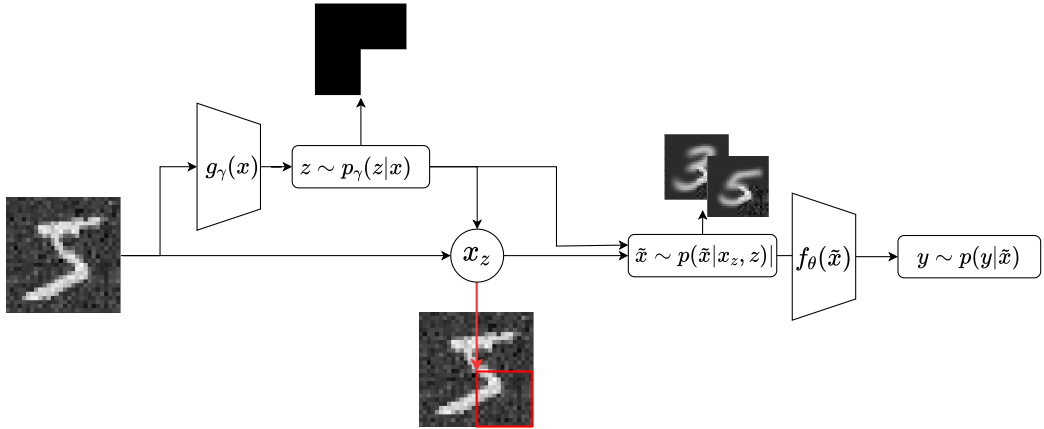
- Instead of explicitly using multiple imputation, pre-train f_θ to learn the marginalization for every mask pattern with 0-imputation.

$$\max_\theta \mathbb{E}_{x,y}\mathbb{E}_{z\sim\mathcal{B}(0.5)} \log p_\theta(y|x,z) \tag{4}$$

  where $p_\theta(y|x,z) = \Phi(y|f_\theta(x \odot z))$

- In theory (see Le Morvan et al 2021), even with rough imputation (e.g. 0-imputation), we get $\forall z$ in the limit of infinite data : $p_\theta(y|x,z) \approx \mathbb{E}_{\tilde{x}\sim p_{\text{data}}(\tilde{x}|x_z)} p_{\text{data}}(y|x_z)$
- In practice, this is a very complicated problem (Le Morvan et al. [2021], Ipsen et al. [2022]).

Complete model



Maximum likelihood inference

We can infer the parameters using a maximum likelihood estimation of the model.

$$\max_{\theta,\gamma} \mathcal{L}(\theta, \gamma)$$

$$\mathcal{L}(\theta, \gamma) = \sum_{n=1}^{N} \log p_{\theta,\gamma}(y^n|x^n) = \sum_{n=1}^{N} \log[\mathbb{E}_{Z\sim p_\gamma(.|x^n)} \mathbb{E}_{\tilde{X}\sim p_{\text{imp}}(.|(x^n)Z)} p_\theta(y^n|\tilde{X})].$$

Need for regularization

If $p_\gamma(z|x)$ can be any conditional distribution, then the model can just select all the input features.

- Explicit regularization controlled by a parameter $\lambda \in \mathbb{R}^+$ and an increasing function in the number of feature selected $\mathrm{R}(z) : \{0,1\}^d \to \mathbb{R}$

$$\max_{\theta,\gamma} \mathcal{L}(\theta, \gamma) - \sum_{n=1}^{N} \lambda \mathbb{E}_{z \sim p_\gamma(x^n)}[R(z)]$$

- L1-regularization $R(z) = \sum_{d=1}^{D} ||z_d||_1$

- Implicit regularization with the choice of p_γ parametrization.

$$\max_{\theta,\gamma} \mathcal{L}(\theta, \gamma)$$

- One could consider a parametrization with a fixed number of selected variables (Subset Sampling Xie and Ermon (2019))

A framework for explainability as statistical inference

- The distribution family and parametrization for the predictor.
- The distribution family and parametrization for the selector.
- Some form of regularization to ensure some selection happens. This regularization can be implicit in the distribution family of the selector.
- Some imputation that can be probabilistic or deterministic that will handle how we turn off features.

$$\max_{\theta,\gamma} \sum_{n=1}^{N} \left[\log \mathbb{E}_{Z \sim p_\gamma(.|x_n)} \mathbb{E}_{\tilde{X} \sim p_{\text{imp}}(.|x_n, Z)} p_\theta(y_n|\tilde{X}) - \lambda \mathsf{R}(\gamma, x_n) \right].$$

Different models fits this framework : Learning To Explain (Chen et al. (2018)), INVASE (Yoon et al. (2018)), REAL-X (Jethani et al. (2021))...

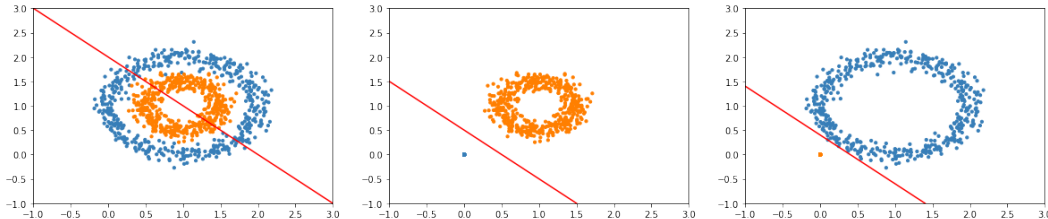
Existing models in this framework

Table: Existing models and their parametrization in the LEX framework.

Model	Sampling Mask (p_γ)	Regularization (R)	Imputation (p_ι)
L2X	Subset sampling	Implicit in p_γ	0 imputation
Invase	Bernoulli	L1-regularization	0 imputation
REAL-X	Bernoulli	L1-regularization	Surrogate 0 imputation

Cheating with a constant imputation c

If we use constant imputation, we allow the selector model to encode the prediction.



Using a surrogate f_θ

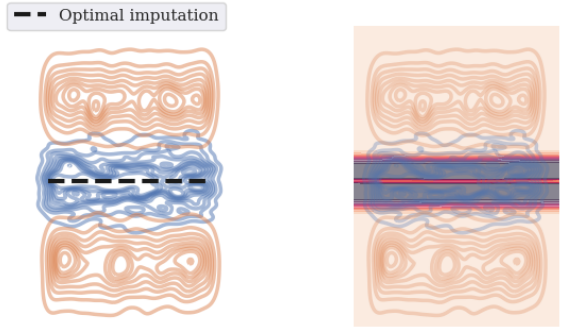


Figure: Figure from Ipsen et al. [2022].

Using a surrogate f_θ

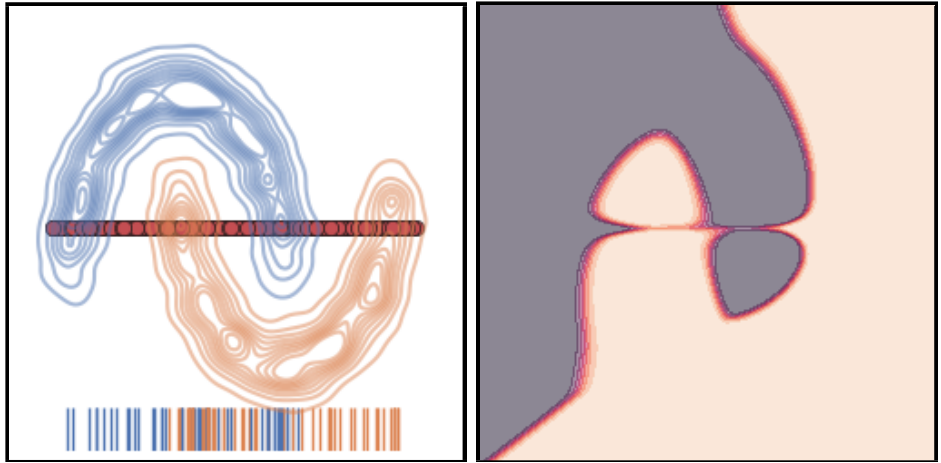


Figure: Figure from Ipsen et al. [2022].

Test on a artificial dataset

$$\{x_i\}_{i=1}^{11} \sim \mathcal{N}(0,1) \quad y \sim \mathcal{B}\left(\frac{1}{1+f(x)}\right). \tag{5}$$

- $f_A(x) = \exp(x_1 x_2)$,
- $f_B(x) = \exp(\sum_{i=3}^{6} x_i^2 - 4)$,
- $f_C(x) = \exp(-10\sin(0.2x_7) + |x_8| + x_9 + e^{x_{-10}} - 2.4)$.

This leads to the following datasets:

- **S1**:

$$f(x) = \begin{cases} f_A(x) & \text{if } x_{11} < 0 \\ f_B(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{6}$$

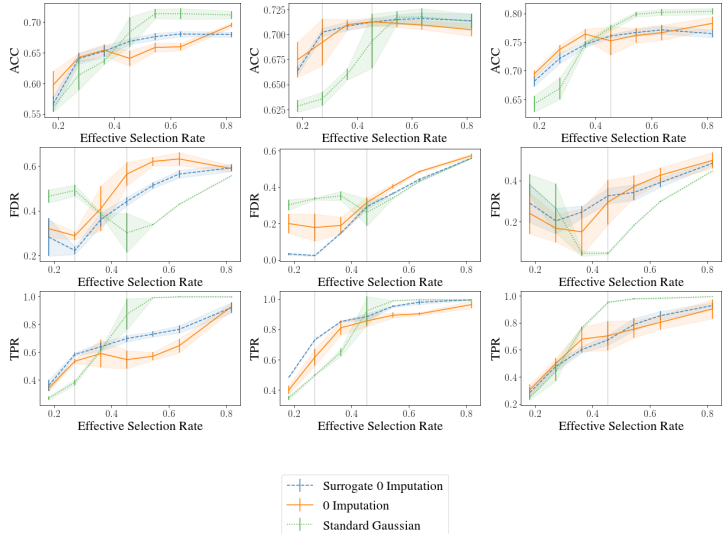
- **S2**:

$$f(x) = \begin{cases} f_A(x) & \text{if } x_{11} < 0 \\ f_C(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{7}$$

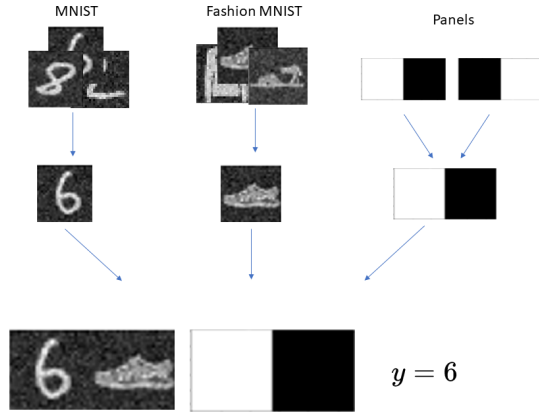
- **S3**:

$$f(x) = \begin{cases} f_B(x) & \text{if } x_{11} < 0 \\ f_C(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{8}$$

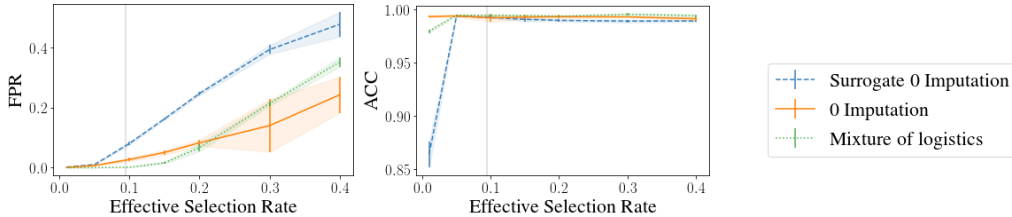
Evaluation on the artificial dataset



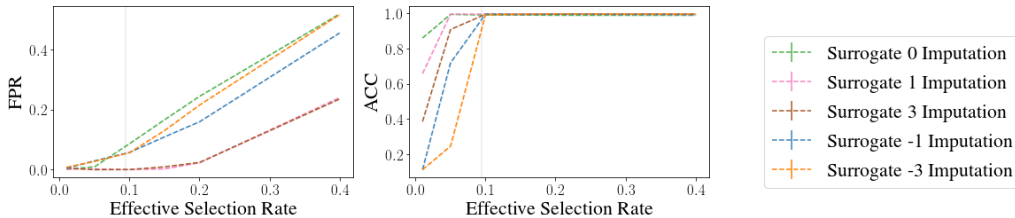
A new dataset to evaluate interpretability



Switching panel dataset



Results depends on the constant



Is surrogate imputation not cheating ?

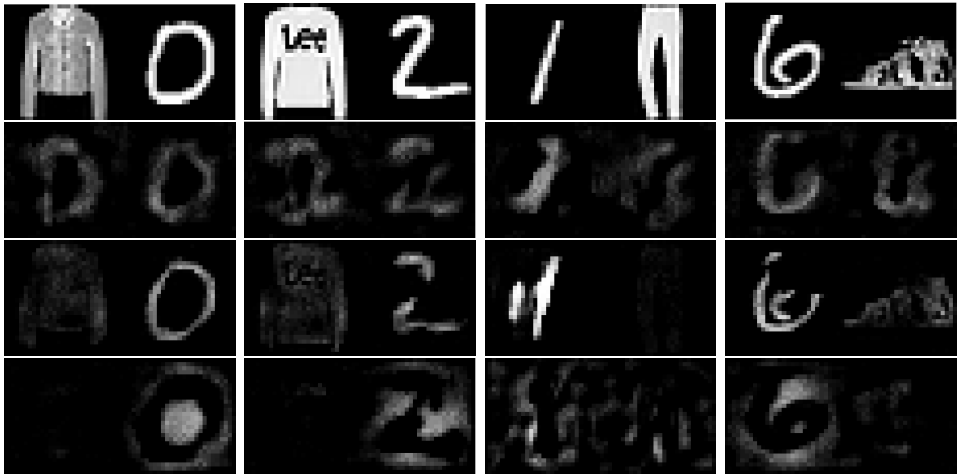
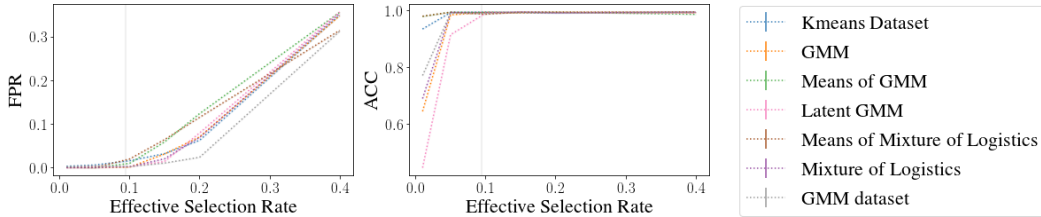


Figure: Surrogate imputation on Switching-Panels MNIST with constant varying in $\{-1, 0, 1\}$

Multiple imputation is more robust



Multiple imputation with SP-MNIST

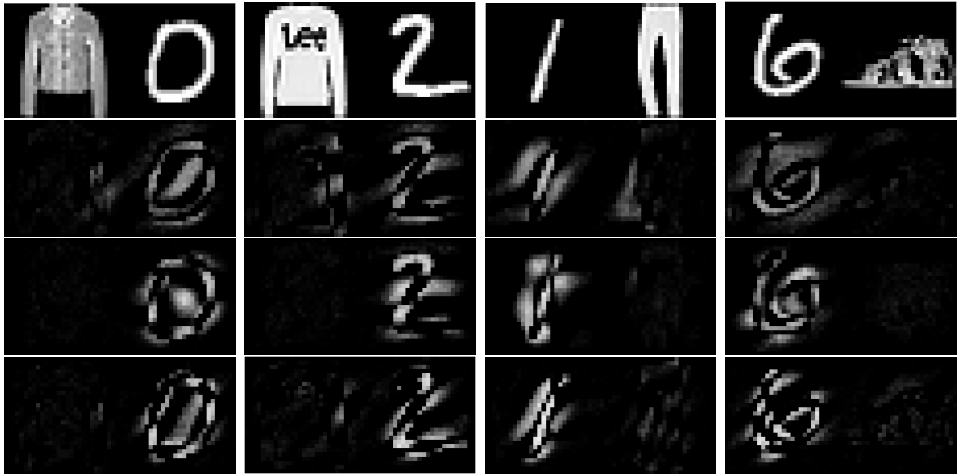
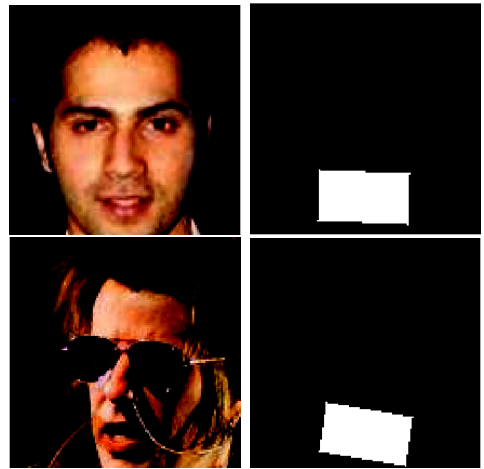
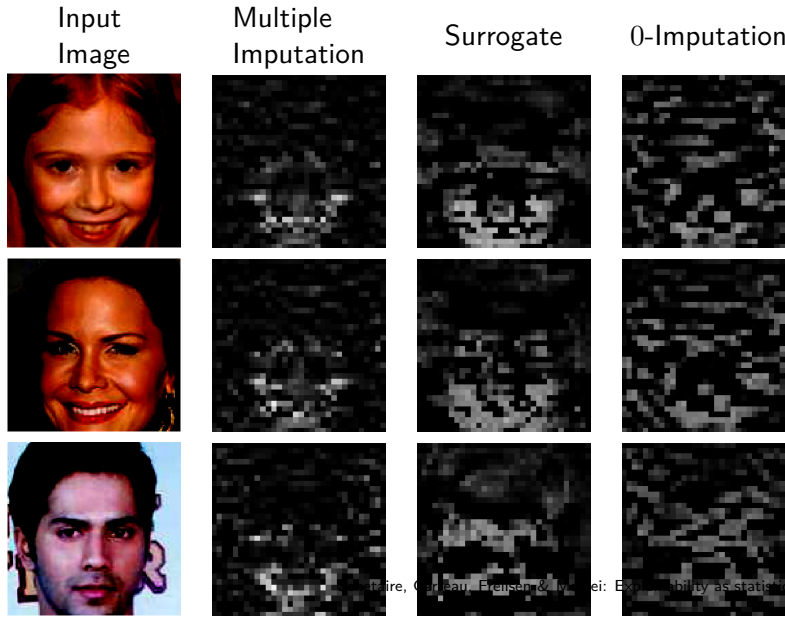


Figure: Surrogate imputation on Switching-Panels MNIST with different multiple imputation procedures (dataset imputation, mixture of logistics and means of a gaussian mixture model).

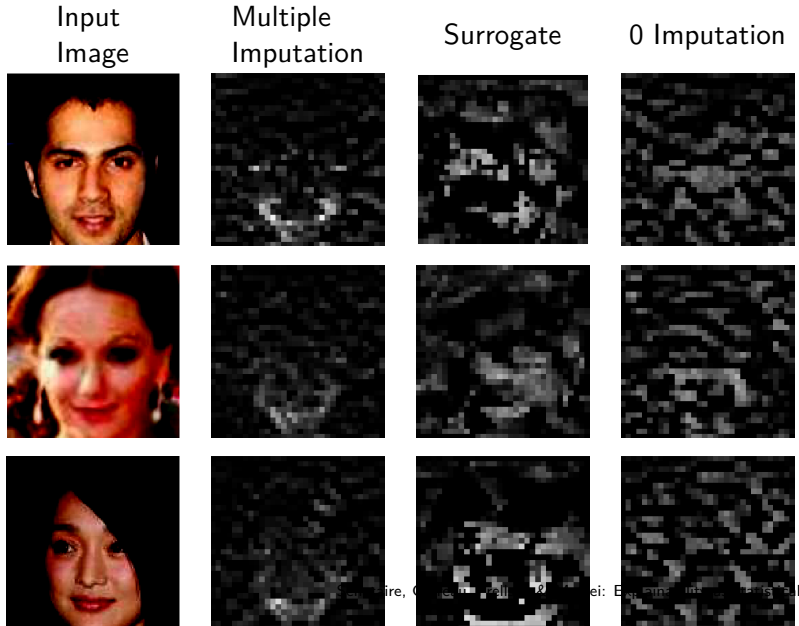
CelebA Smile Dataset



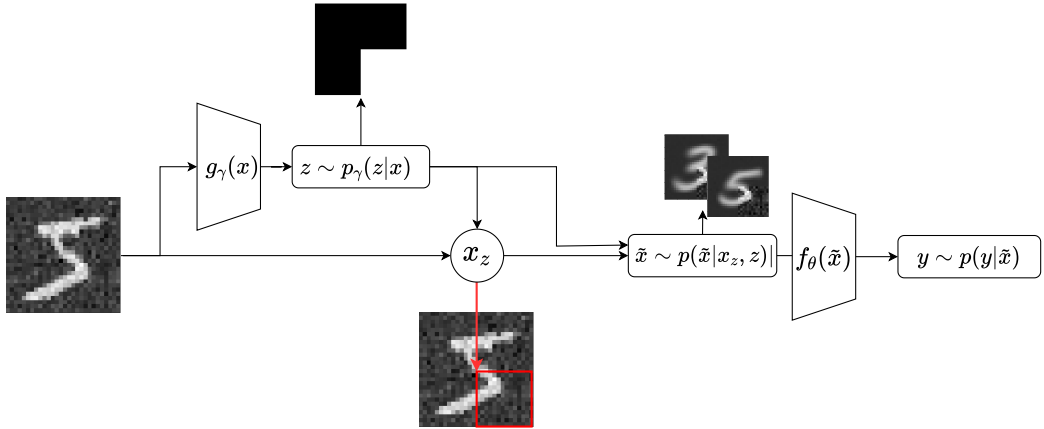
Multiple imputation with CelebA Smile



CelebA Smile



Thank you for your attention !



Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Richard Caruana. Friends don't let friends deploy black-box models: The importance of intelligibility in machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3174–3174, 2019.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pages 1459–1467. PMLR, 2021.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Jaime Sevilla. Parameter counts in machine learning, 2021. URL `https://towardsdatascience.com/parameter-counts-in-machine-learning-a312dc4753d0`.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 3919–3925. AAAI Press, 2019. ISBN 9780999241141.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

Practicalities

$$\max_{\theta,\gamma} \sum_{n=1}^{N} \left[\log \mathbb{E}_{Z \sim p_\gamma(.|x_n)} \mathbb{E}_{\tilde{X} \sim p_{\text{imp}}(.|x_n, Z)} p_\theta(y_n|\tilde{X}) - \lambda \mathsf{R}(\gamma, x_n) \right].$$

- We have the log of an expectation, we use an IWAE trick to get a tight lower bound of this loss (Burda et al., 2015)
- Optimizing in γ because we optimize the gradient of an expectation of a discrete distribution with its parameters, we can use some continuous approximation tricks (Xie and Ermon, 2019; Tucker et al., 2017; Bengio et al., 2013).
- Another difficulty is that p_{imp} is not available. We approximate it with a mixture of Gaussians.

Test on a artificial dataset

$$\{x_i\}_{i=1}^{11} \sim \mathcal{N}(0,1) \quad y \sim \mathcal{B}\left(\frac{1}{1+f(x)}\right). \tag{9}$$

- $f_A(x) = \exp(x_1 x_2)$,
- $f_B(x) = \exp(\sum_{i=3}^{6} x_i^2 - 4)$,
- $f_C(x) = \exp(-10\sin(0.2x_7) + |x_8| + x_9 + e^{x_{-10}} - 2.4)$.

This leads to the following datasets:

- **S1**:

$$f(x) = \begin{cases} f_A(x) & \text{if } x_{11} < 0 \\ f_B(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{10}$$

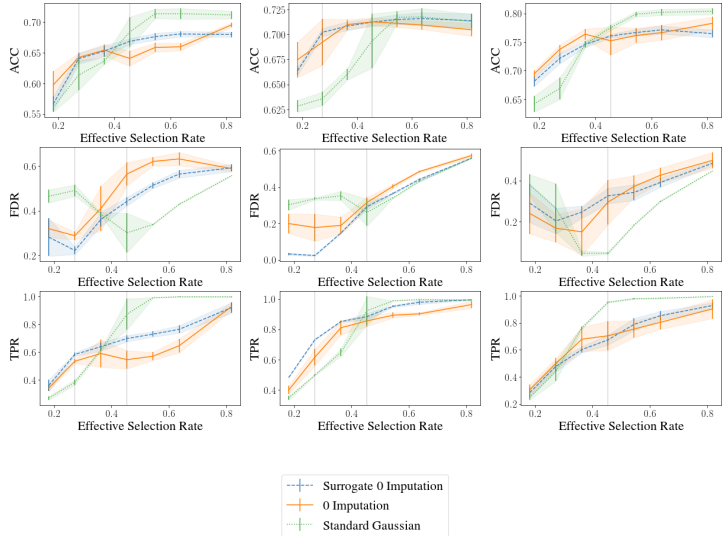
- **S2**:

$$f(x) = \begin{cases} f_A(x) & \text{if } x_{11} < 0 \\ f_C(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{11}$$

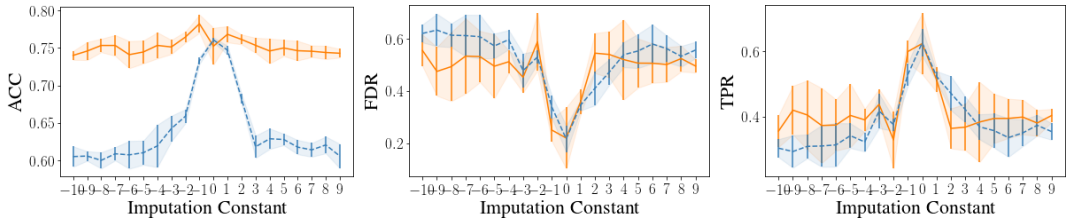
- **S3**:

$$f(x) = \begin{cases} f_B(x) & \text{if } x_{11} < 0 \\ f_C(x) & \text{if } x_{11} \geq 0. \end{cases} \tag{12}$$

Evaluation on the artificial dataset

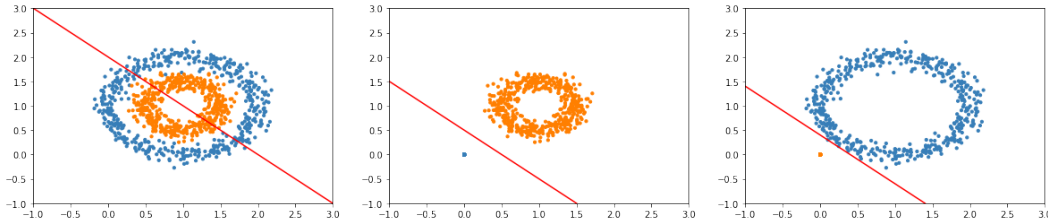


Evaluation on the artificial dataset



Cheating with a constant imputation c

If we use constant imputation, we allow the selector model to encode the prediction.



0-Imputation encodes the target in the mask

Figure: Results on MNIST using 0-imputation. Figure from Jethani et al. [2021]

Is REAL-X enough to prevent the encoding ?

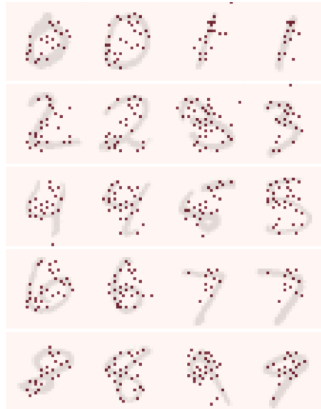


Figure: Results on MNIST using surrogate 0-Imputation. Figure from Jethani et al. [2021]