Post-hoc counterfactual generation with supervised autoencoder

Victor Guyomard

Orange Lannion, France

Univ Rennes, Inria, Rennes, France

victor.guyomard@orange.com

Françoise Fessant

Orange Lannion, France

francoise.fessant@orange.com

Tassadit Bouadi

Univ Rennes, Inria, Rennes, France

tassadit.bouadi@irisa.fr

Thomas Guyet

Institut Agro/IRISA, Rennes, France

thomas.guyet@irisa.fr

AIMLAI, ECMLPKDD

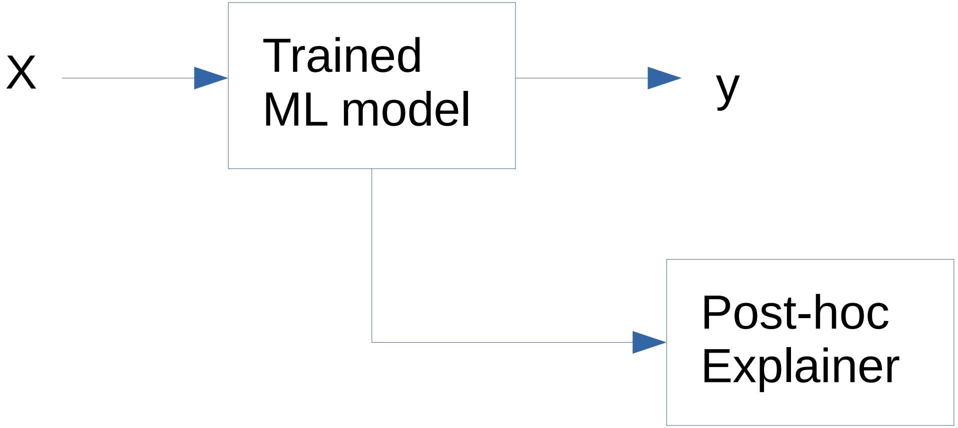
September 13, 2021

Context

- Supervised learning classifier $f_{pred} : \mathcal{X} \to \mathcal{Y}$ $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$

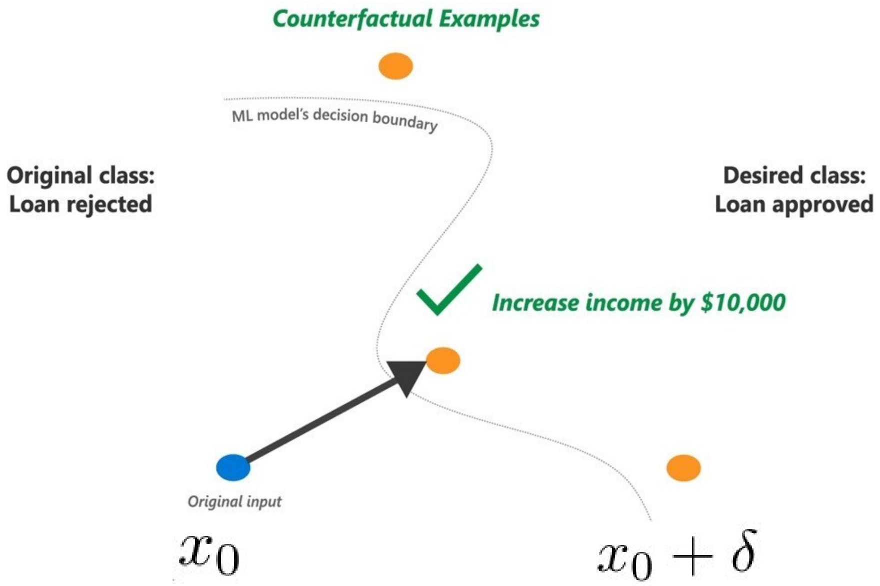
$$\mathcal{Y} = \{1, 2, \ldots, C\}$$

- Post-hoc explanations *Apply an explainable method on a trained machine learning model*



Explanation by counterfactuals

Counterfactual explanation for ML models: Smallest change of feature values that changes a prediction to a given output.



Source: Microsoft Research Blog

Sandra Wachter et al, Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR, 2018, Harvard Journal of Law & Technology .

Explanation by counterfactuals

Counterfactual examples are most of the time found by minimizing a **cost function**.

- Generally **2 terms that are related to the definition** (closeness to the example + different predicted class)
- **Many possible cost functions** and implementations of the optimization method, depending of the expected properties of the counterfactual.
  Ex: Sparsity, actionability, closeness to training data, diversity…
- Many open challenges: **no consensus** on what a good counterfactual is and how it can be evaluated.

Verma, S., Dickerson, J., & Hines, K. (2020). Counterfactual Explanations for Machine Learning: A Review. arXiv preprint arXiv:2010.10596.

Interpretable Counterfactual Explanations guided by prototypes

<u>3 steps process:</u>

<u>1)</u>

$x_0 \longrightarrow$ ML model $\longrightarrow y_0$

Train a machine learning model to predict a given class (y_0 *)*

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

Interpretable Counterfactual Explanations guided by prototypes

2)

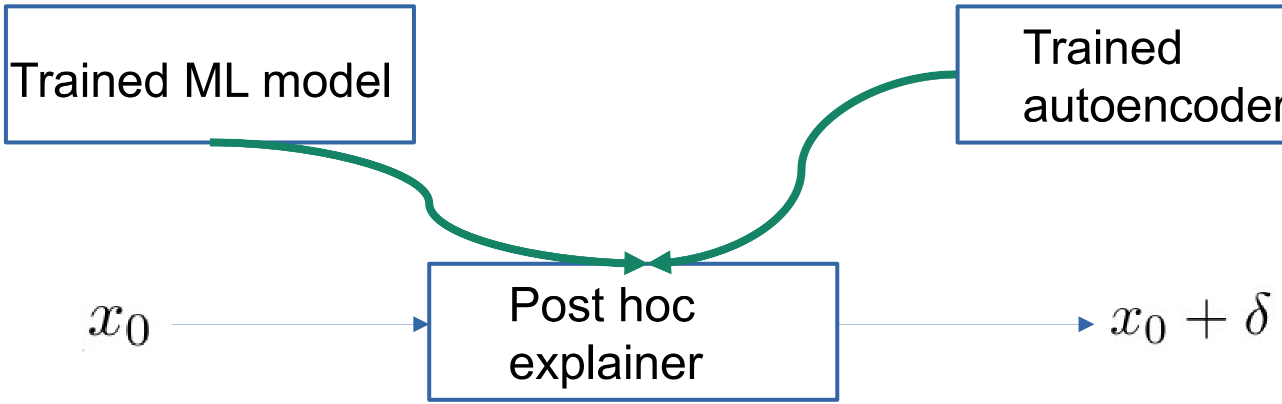
$$x_0 \longrightarrow \boxed{\text{Autoencoder}} \longrightarrow x_0'$$

Train an autoencoder to reconstruct a sample (x_0*)*

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

Interpretable Counterfactual Explanations guided by prototypes

3)



Find a counterfactual ($x_0 + \delta$) by optimizing a cost function that uses the trained autoencoder and the trained ML model

Limit: requires the training of 2 models

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

The cost function

$$\min_{\delta} \left(c \cdot f_{\kappa}(x_0, \delta) + f_{\text{dist}}(\delta) + L_{AE} + L_{\text{proto}} \right)$$

x_0 : Example to explain

$x_0 + \delta$: Counterfactual example

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

The cost function

$$\min_{\delta} \left(\underline{c \cdot f_{\kappa}(x_0, \delta)} + f_{\text{dist}}(\delta) + L_{AE} + L_{\text{proto}} \right)$$

$$f_{\kappa}(x_0, \delta) = \max\left([f_{\text{pred}}(x_0 + \delta)]_{y_0} - \max_{y_i \neq y_0} [f_{\text{pred}}(x_0 + \delta)]_{y_i},\ -\kappa \right)$$

Term to ensure that the predicted class for counterfactual is different

The cost function

$$\min_{\delta} \left(c \cdot f_{\kappa}(x_0, \delta) + f_{\text{dist}}(\delta) + L_{AE} + L_{\text{proto}} \right)$$

$$f_{\text{dist}}(\delta) = \beta \cdot ||\delta||_1 + ||\delta||_2^2.$$

Minimize distance between counterfactual and example / Sparse perturbation

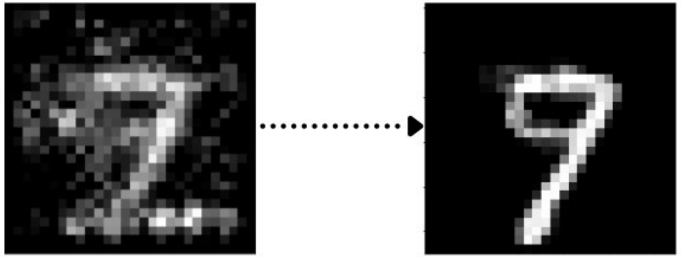
The cost function

$$\min_{\delta}\left(c \cdot f_{\kappa}(x_0, \delta) + f_{\text{dist}}(\delta) + L_{AE} + L_{\text{proto}}\right)$$

$$L_{\text{AE}} = \gamma \cdot \|x_0 + \delta - \text{AE}_D(x_0 + \delta)\|_2^2.$$

Reconstruction error of counterfactual evaluated by an autoencoder (AE) trained with a data distribution D.

Penalize out of distribution counterfactuals

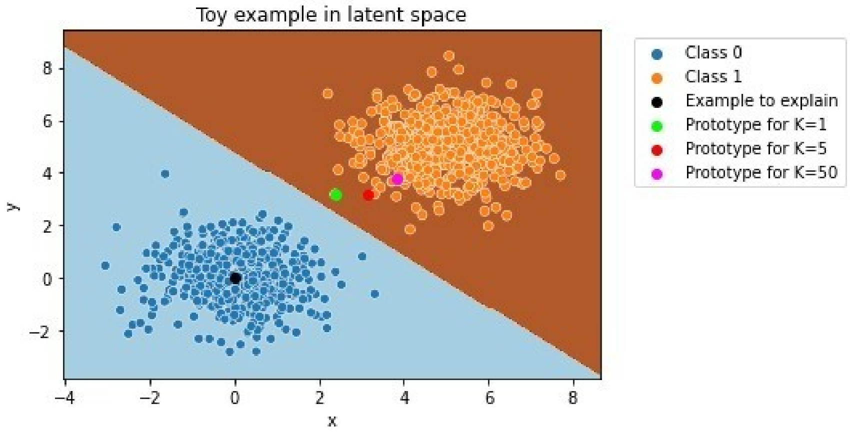


The cost function

$$\min_{\delta} \left(c \cdot f_{\kappa}(x_0, \delta) + f_{\text{dist}}(\delta) + L_{AE} + L_{\text{proto}} \right)$$

$$\text{proto}_{y_j} = \frac{1}{K} \sum_{k=1}^{K} \text{ENC}_D \left(x_k^i \right)$$

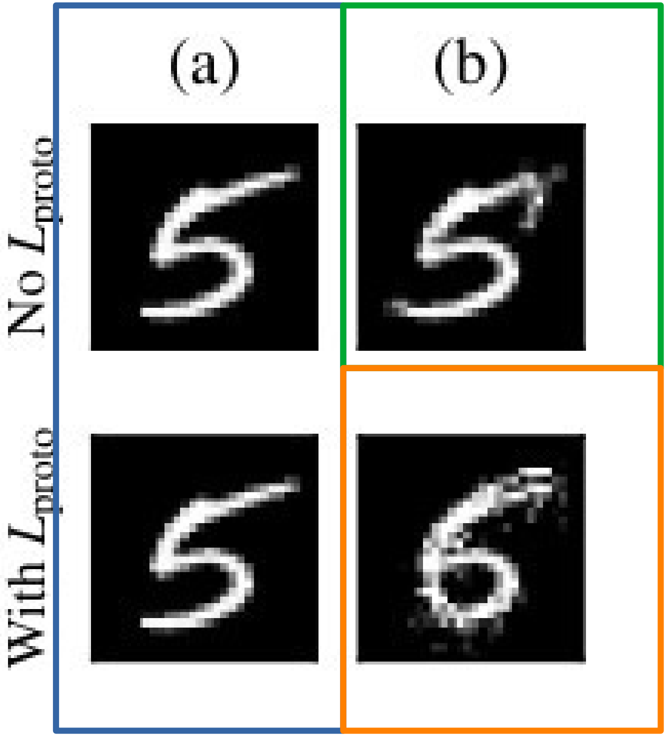
$$L_{\text{proto}} = \theta \cdot \|\text{ENC}_D(x_0 + \delta) - \text{proto}_{y_j}\|_2^2,$$



Counterfactual examples belong distribution of counterfactual class

The cost function

Why guiding by prototype in a latent space?



Example predicted as a 5

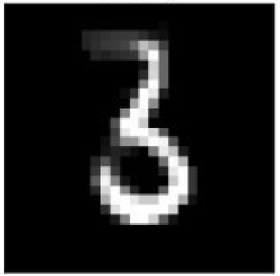
Counterfactual without guiding by prototype, predicted as a 6

Counterfactual with guiding by prototype, predicted as a 6

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

Illustration of limits

Example predicted as 3

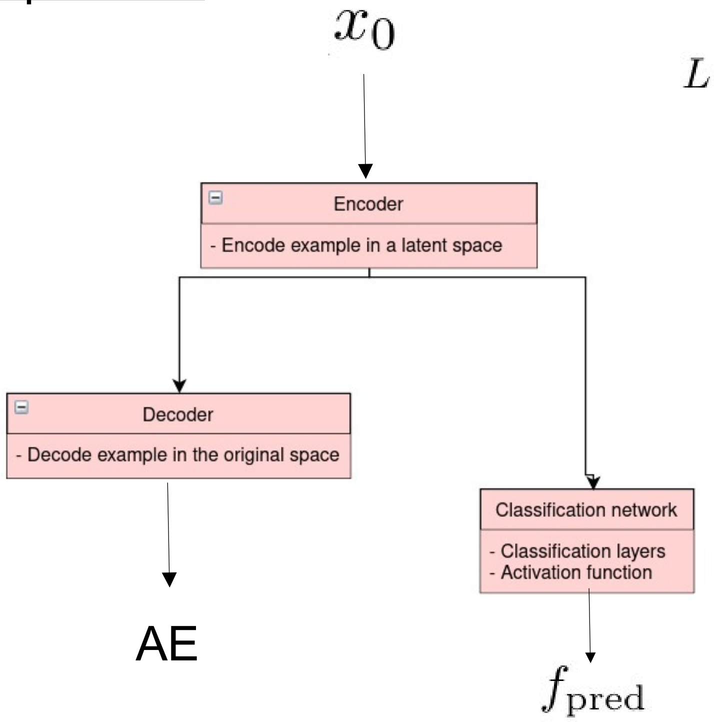


Counterfactual example predicted as a 6
→ **Not looks like a "6"**

Our contribution

2 steps process:

1)



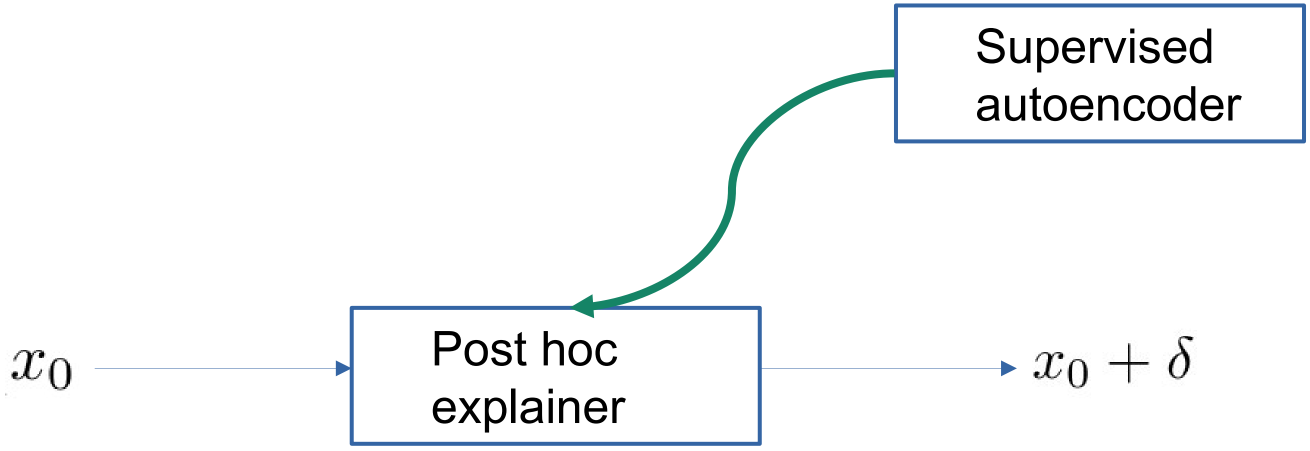
$$L((f_{\text{pred}}, \text{AE}), D) = \underbrace{E(f_{\text{pred}}, D)}_{\text{Classification loss}} + \lambda \underbrace{R(\text{AE}, D)}_{\text{Reconstruction loss}}$$

- Train a supervised autoencoder

- We then obtain:
    - A classifier
    - An autoencoder

Our contribution

2)



Find a counterfactual ($x_0 + \delta$) by optimizing a cost function

$$\min_{\delta} \left(c \cdot f_{\kappa}(x_0, \delta) + f_{\text{dist}}(\delta) + \boxed{L_{AE} + L_{\text{proto}}}\right)$$

Limit: No model agnostic (only neuronal networks models)

First benefit: Only one model to train

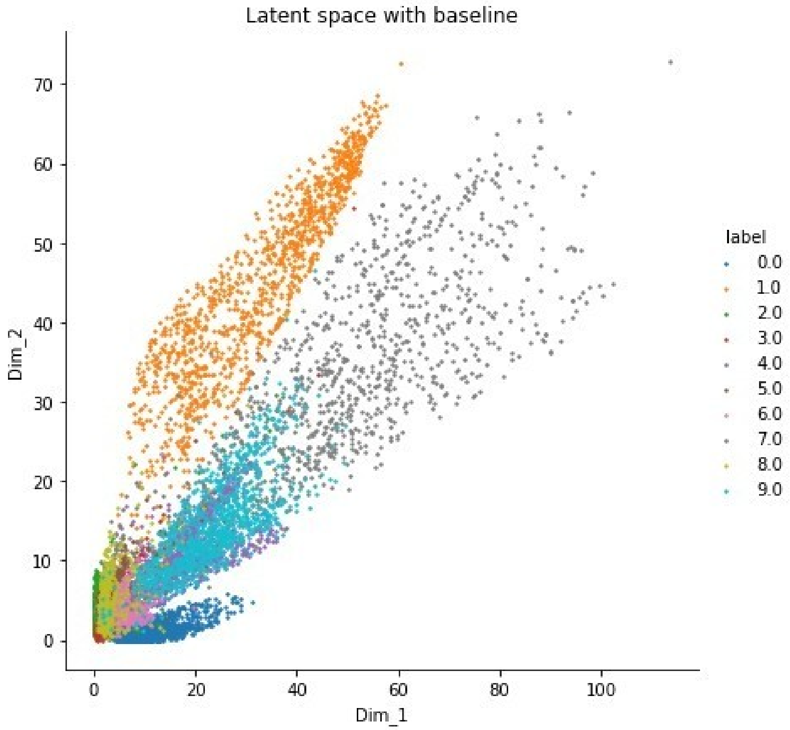
Our contribution

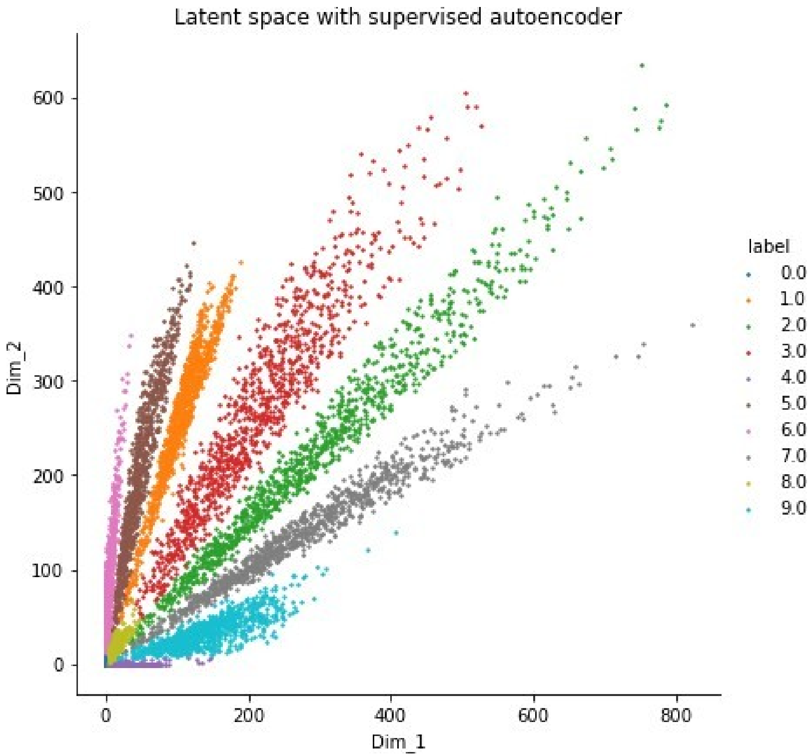
Intuition behind the use of a supervised autoencoder:

Design an **organized latent** space according to **classes**.

Prototypes will be **more representative of a given class** / Hence more representative counterfactuals

Visualization of a 2 dimensional latent space on MNIST





Examples of the same predicted class are «mixed» in the latent space

Examples of the same predicted class are clustered in the latent space

Experimental setting

- MNIST Dataset

- Random sample of 5000 examples.

- Same hyperparameters as *Van Looveren et al.* for counterfactual generation.

Arnaud Van Looveren and Janis Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, 2021,European Conference on Machine Learning and Knowledge Discovery in Databases (ECMLPKDD'21)

Evaluation Metrics

Predicted probability for counterfactual (according to counterfactual class)

$$\text{Gain} = [f_{\text{pred}}(x_{cf})]_{y_i} - [f_{\text{pred}}(x_0)]_{y_i}$$

Predicted probability for example (according to counterfactual class)

$$\text{Realism} = \|\text{AE}_{\text{evaluate}}(x_{cf}) - x_{cf}\|_2^2$$

$$\text{Actionability} = \|x_{cf} - x_0\|_1 = \|\delta\|_1$$

y_i : Counterfactual predicted class

δ : Perturbation

x_{cf} : Counterfactual example

Daniel Nemirovsky et al, CounteRGAN: Generating Realistic Counterfactuals with Residual Generative Adversarial Nets, 2020, arXiv.

Results

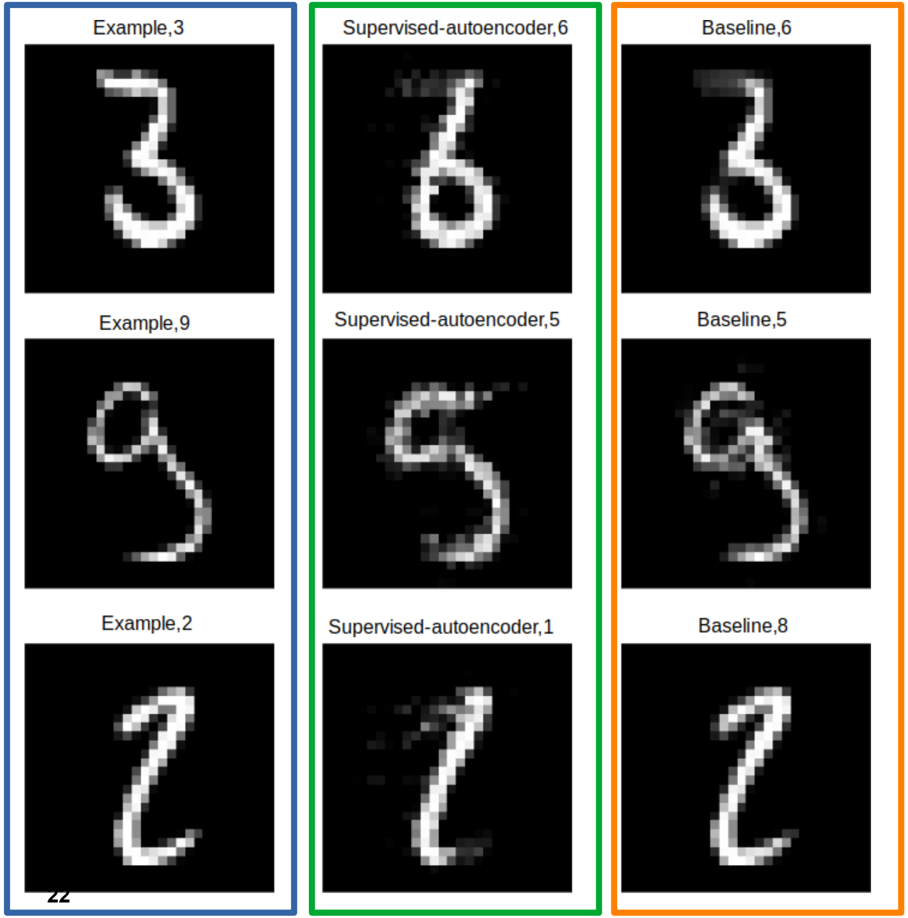
Table 1. Counterfactual metrics comparison. The arrows indicate whether larger ↑ or lower ↓ values are better, and the best results are in bold.

Metrics	Baseline	Supervised autoencoder
↑ Prediction gain	0.552±0.106	**0.839±0.160**
↓ Realism	0.253±0.010	**0.249±0.012**
↓ Actionability	**26.174±13.762**	38.360 ±18.465

Higher gain = **more confidence** in the class change of the counterfactual example.

Higher actionability / Equivalent realism

Results Illustration



Examples

Counterfactuals with supervised autoencoder

Counterfactuals with baseline

Conclusion and future work

Conclusion:

- 2 steps process (train only one model by using a supervised autoencoder) instead of 3 steps process

- Organize the latent space according to classes (more meaningful prototypes hence counterfactuals)

- Evaluation on MNIST dataset

- Higher prediction gain with less actionability and equivalent realism

Future work:

- Adapt this method to tabular data

  Scientific issues: *Using a latent space still relevant? / How to treat categorical variables? / Take into account loss of visual interpretability?*

Thank you !