CSDS 452 Causality and Machine Learning

Lecture 16: Causal Explanation

Instructor: Jing Ma

Fall 2024, CDS@CWRU

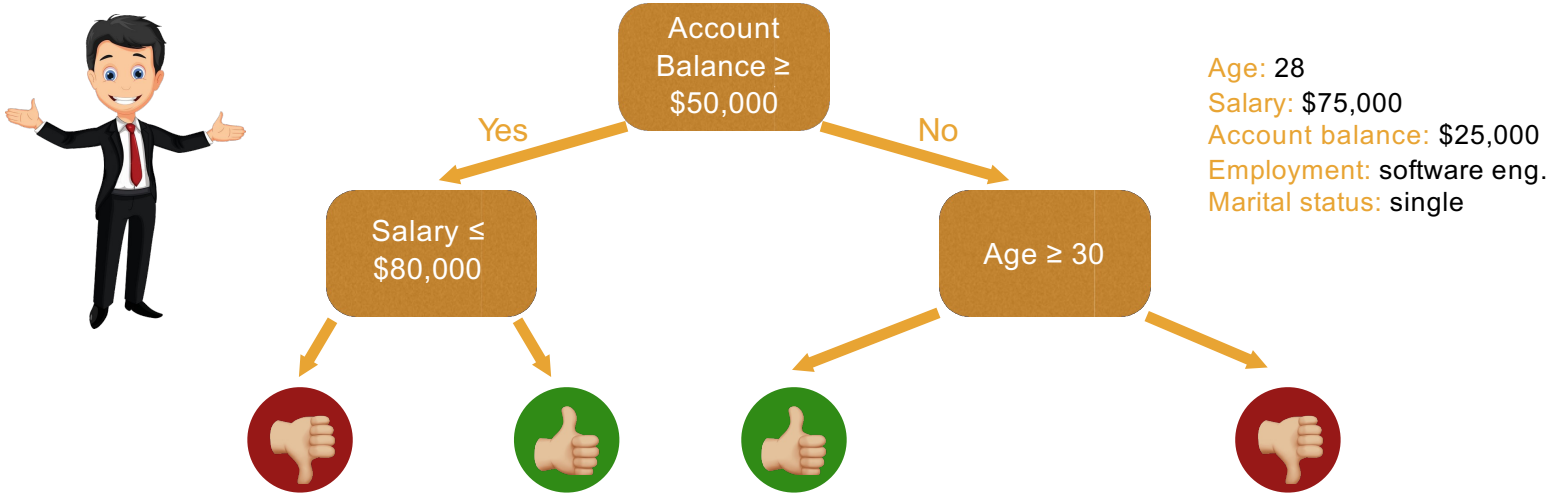
Outline

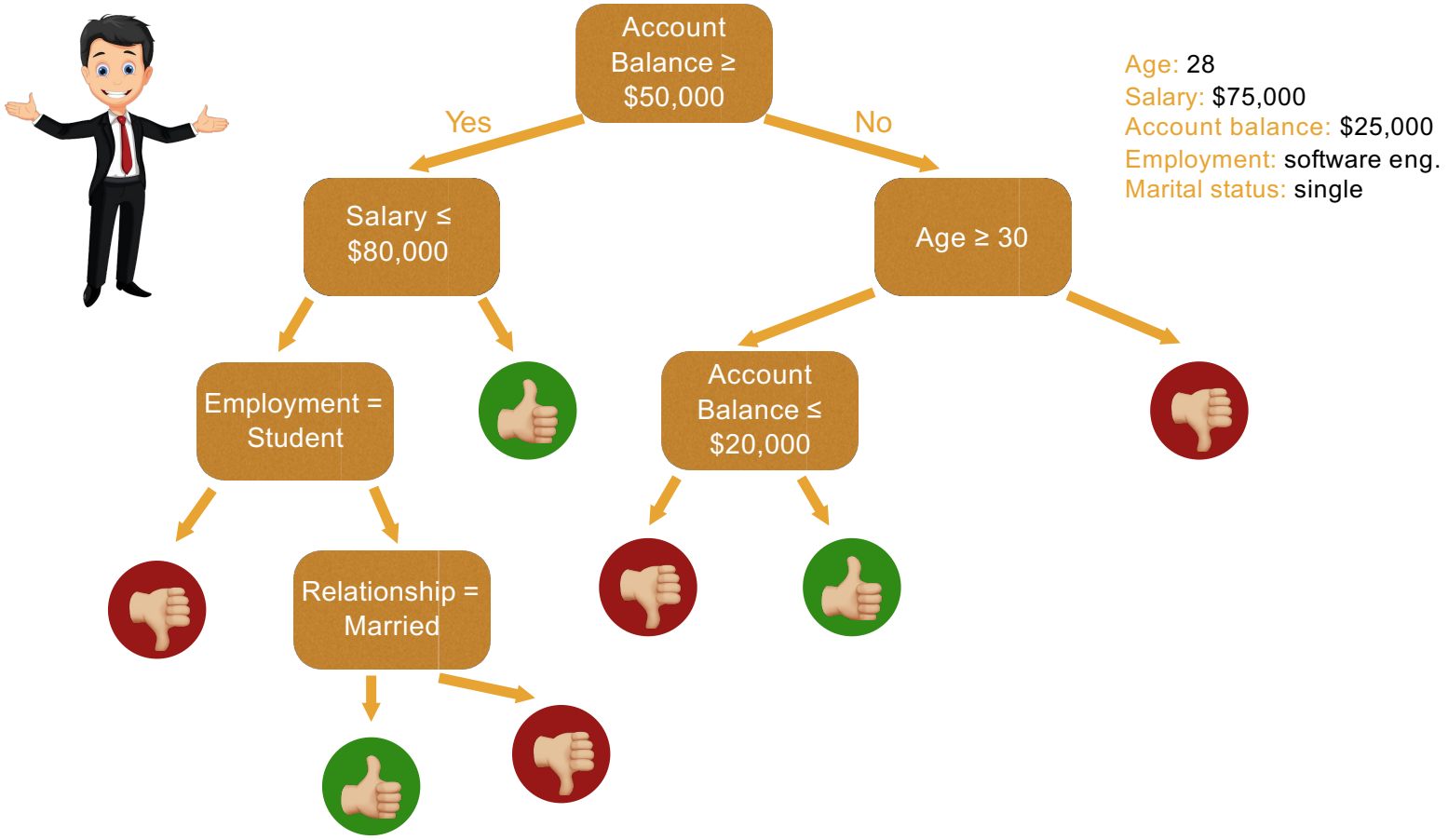
- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability
- Model-based interpretability (attribution)

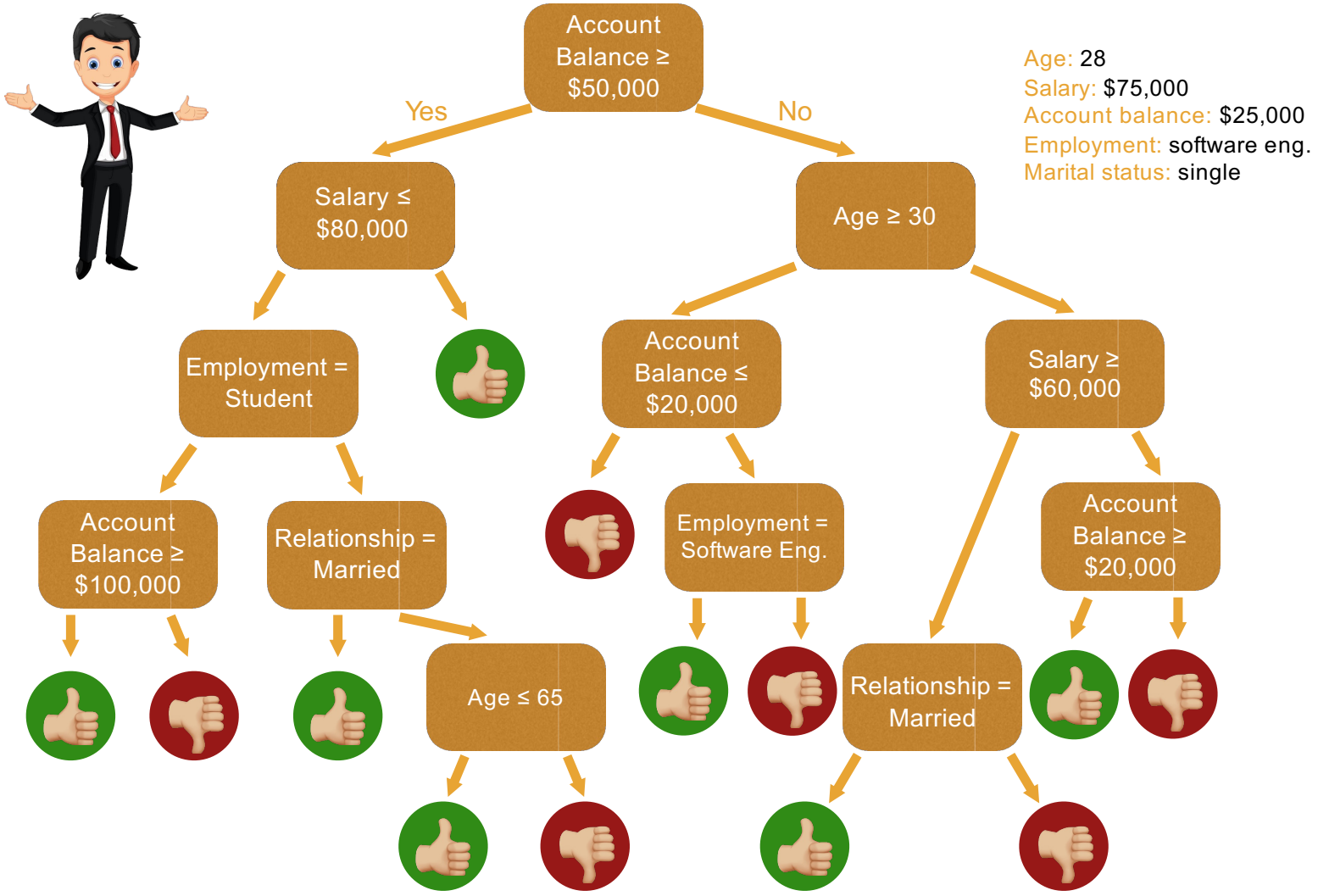
Case study

Edward

Age: 28
Salary: $75,000
Account balance: $25,000
Employment: software eng.
Marital status: single


Account Balance ≥ $50,000
Yes
No
Salary ≤ $80,000
Age ≥ 30
Age: 28
Salary: $75,000
Account balance: $25,000
Employment: software eng.
Marital status: single


Account Balance ≥ $50,000
Yes
No
Salary ≤ $80,000
Age ≥ 30
Employment = Student
Account Balance ≤ $20,000
Relationship = Married
Age: 28
Salary: $75,000
Account balance: $25,000
Employment: software eng.
Marital status: single


Account Balance ≥ $50,000
Yes
No
Salary ≤ $80,000
Age ≥ 30
Employment = Student
Account Balance ≤ $20,000
Salary ≥ $60,000
Account Balance ≥ $100,000
Relationship = Married
Employment = Software Eng.
Account Balance ≥ $20,000
Age ≤ 65
Relationship = Married
Age: 28
Salary: $75,000
Account balance: $25,000
Employment: software eng.
Marital status: single

Stakeholders

What are the most influential features towards the decision?

Is the system “fair” by relying on sensitive attributes such as *age* and *marital status*?

I didn’t get the loan; why not? And what should I do to get it next time I apply?

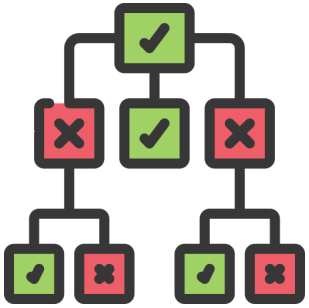
eXplainable AI (XAI)

Interpretable ML (IML a.k.a. XAI) — The *ability to explain or to present in understandable terms to a human* [Doshi-Velez & Kim, 2017]

Goals of XAI — Draw insights from data, models, and/or decisions

Causal Insights — Answer *what*, *how*, and *why (not)* questions

Data-based Interpretability



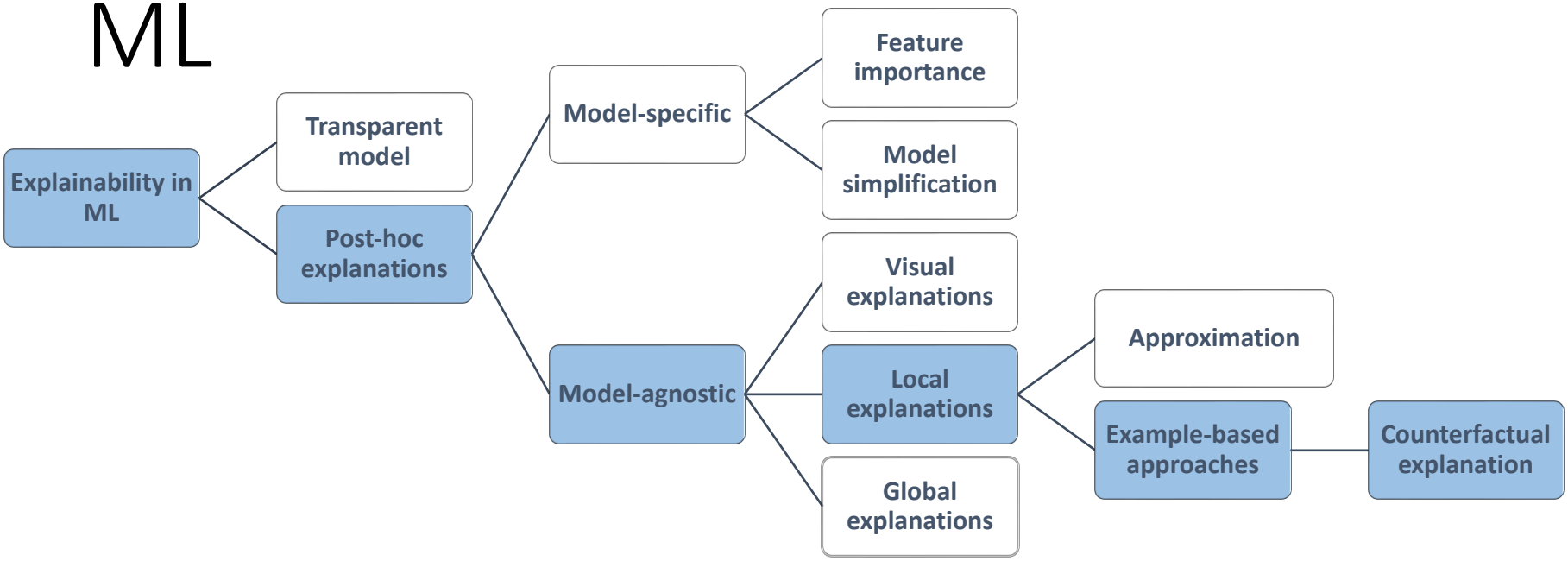
Model-based Interpretability

Decision-based Interpretability

Outline

- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability
- Model-based interpretability (attribution)

A Broad Picture of Explainability in ML



- This figure describes the role of the "major trend" of counterfactual explanation (CFE) in ML explainability
- Some CFE methods are in different categories (e.g., model specific)

[1] Verma S, Dickerson J, Hines K. Counterfactual explanations for machine learning: A review[J]. arXiv preprint arXiv:2010.10596, 2020.

Outline

- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability
- Model-based interpretability (attribution)

Counterfactual Explanation

- **Key question**: what small changes could be made to the input features of an instance to change its prediction?
- Motivation example:
  - In loan application, for an applicant who was rejected, what small change (e.g., improve education level) can the applicant make to achieve a desired outcome?

Basic Desiderata

- **Validity**: generated counterfactuals have desired labels

$$\arg\min_{x'} d(x, x') \text{ subject to } f(x') = y'$$

Desired label

Original goal

$$\arg\min_{x'} \max_{\lambda} \lambda(f(x') - y')^2 + d(x, x')$$

Differentiable, unconstrained objective

Distance (e.g., L1/L2)

- **Sparsity**: a counterfactual ideally should change smaller number of features in order to be most effective

[1] Wachter S, Mittelstadt B, Russell C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR[J]. Harv. JL & Tech., 2017, 31: 841.

Other Desiderata (e.g.)

- **Data Manifold closeness**: a generated counterfactual is realistic in the sense that it is near the training data $l(x'; X)$ ← training set
  - include a penalty for adhering to the data manifold defined by the training set

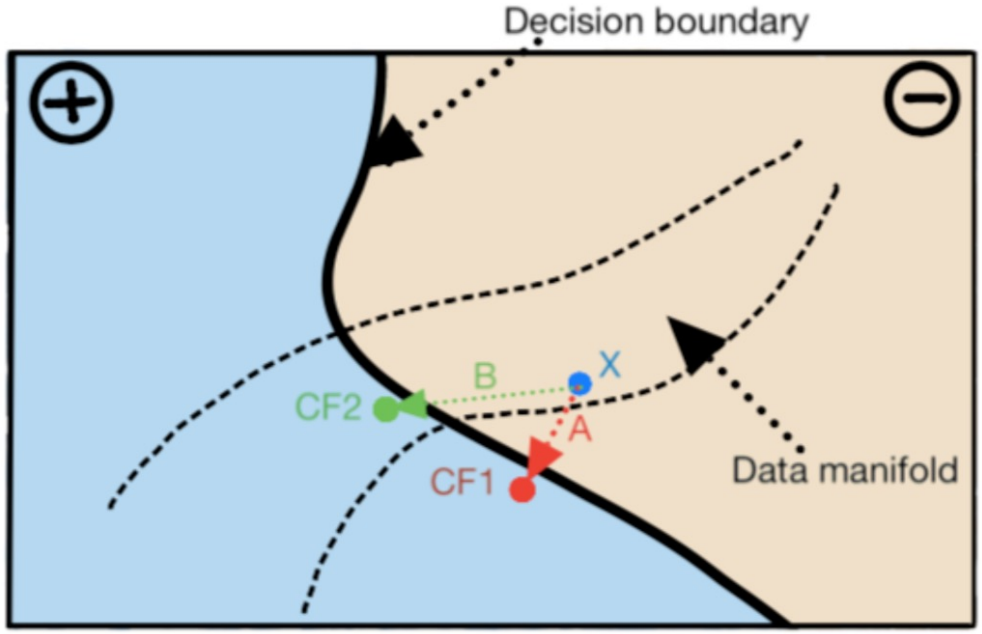


Figure: Two possible paths for a datapoint (shown in blue), originally classified in the negative class, to cross the decision boundary. The end points of both the paths (shown in red and green) are valid counterfactuals for the original point. Note that the red path is the shortest, whereas the green path adheres closely to the manifold of the training data, but is longer.

Properties of the Approaches

- **Model access**
  - access to complete model internals
  - access to gradients
  - access to only the prediction function (black-box)
- **Model agnostic**
  - some algorithms are model-specific, e.g., only work for those models like tree ensembles or with gradients
- **Optimization amortization**
  - **Amortized Inference**: whether the algorithm can generate counterfactuals for multiple input datapoints without optimizing separately
  - **Multiple counterfactual**: whether the algorithm can generate multiple counterfactual for a single input datapoint
- **Counterfactual (CF) attributes**
  - sparsity, data manifold adherence, and causality

Key Properties of Existing CFE Works

	Assumptions		Optimization amortization		CF attributes			CF opt. problem attributes	
Paper	Model access	Model domain	Amortized Inference	Multiple CF	Sparsity	Data manifold	Causal relation	Feature preference	Categorical dist. func
[72]	Black-box	Agnostic	No	No	Changes iteratively	No	No	Yes	-
[111]	Gradients	Differentiable	No	No	L1	No	No	No	-
[104]	Complete	Tree ensemble	No	No	No	No	No	No	-
[74]	Black-box	Agnostic	No	No	L0 and post-hoc	No	No	No	-
[57]	Black-box	Agnostic	No	Yes	Flips min. split nodes	No	No	No	Indicator
[29]	Gradients	Differentiable	No	No	L1	Yes	No	No	-
[56]	Black-box	Agnostic	No	No	No	No	No	No[2]	-
[95]	Complete	Linear	No	Yes	L1	No	No	No	N.A.[3]
[107]	Complete	Linear	No	No	Hard constraint	No	No	Yes	-
[98]	Black-box	Agnostic	No	Yes	No	No	No	Yes	Indicator
[30]	Black-box or gradient	Differentiable	No	No	L1	Yes	No	No	-
[91]	Black-box	Agnostic	No	No	No	No	No	No	-
[61]	Gradients	Differentiable	No	No	No	Yes	No	No	-
[90]	Gradients	Differentiable	No	No	No	No	No	No	-
[113]	Black-box	Agnostic	No	No	Changes one feature	No	No	No	-
[85]	Gradients	Differentiable	No	Yes	L1 and post-hoc	No	No	No	Indicator
[89]	Black-box	Agnostic	No	No	No	Yes[4]	No	No	-

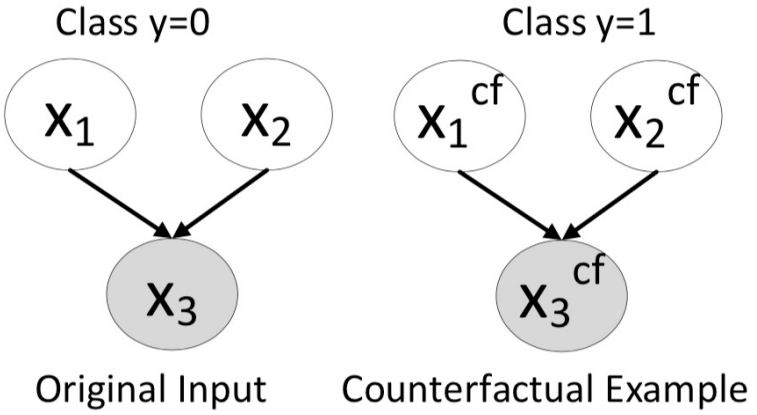
Verma S, Boonsanong V, Hoang M, et al. Counterfactual explanations and algorithmic recourses for machine learning: A review[J]. arXiv

Until now, no causality is involved

- In causal inference, counterfactuals refer to a different version that does not actually happen, e.g.,
  - What if I hadn't taken the medicine? (I took the medicine)
  - If I were a boy, I would … (I am a girl)
- The word "counterfactual" is often overused!
  - In many papers, counterfactual just refer to anything different than the current one without causality involved.
- Obviously, a counterfactual should take causal relations into consideration

Desiderata: Causality

- **Causality**: changing one feature in the real world affects other features
  - E.g., getting a new educational degree necessitates increases the age



[1] Mahajan D, Tan C, Sharma A. Preserving causal constraints in counterfactual explanations for machine learning classifiers[J]. NIPS Workshop, 2019.

Evaluation

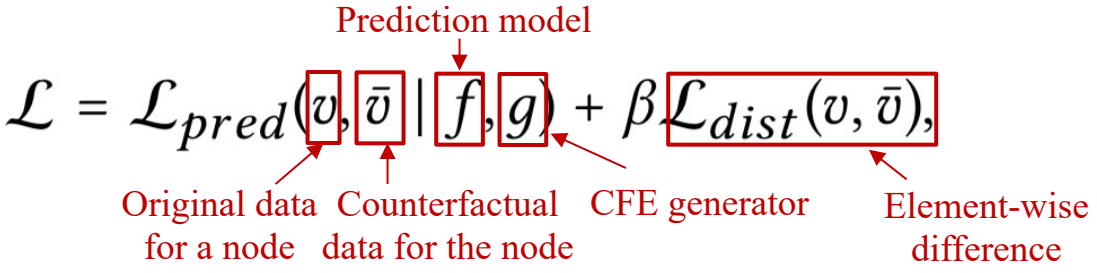
- Commonly used datasets
  - Image - MNIST
  - Tabular - Adult income, German credit, Compas recidivism, etc.
- Metrics
  - **Validity**: the ratio of the counterfactuals that have the desired class label to the total number of counterfactuals.
  - **Proximity**: the distance of a counterfactual from the input datapoint.
  - **Sparsity**: the number of modified features
  - **Diversity**: diversity is encouraged by maximizing the distance between the multiple counterfactuals
  - **Causal constraint**: whether the counterfactuals satisfy the causal relation between features.

Counterfactual Explanation on Graph

- Counterfactual explanations for graphs: the minimal perturbation to the input (graph) data such that the prediction changes.
- Motivation & applications:
  - Drug discovery: if the model predicts it does not have this desirable property, CFE can help identify the minimal change one should make to this molecule
- Challenges:
  - CFE contains different modalities
  - Perturbation on graph structure is non-differentiable
  - Some patterns in graph structure (e.g., circles) are hard to be learned and perturbed for CFE

CF-GNNExplainer: Method

- CF-GNNExplainer[1]
  - based on GNN
  - focus on node classification task
  - focus on perturbing graph structure
- Main idea of the method
  - iteratively remove edges (learn a perturbation matrix) from the original adjacency matrix based on matrix sparsification techniques
  - track of the perturbations that lead to a change in prediction
  - return the perturbation with the smallest change w.r.t. the number of edges



[1] Lucic A, ter Hoeve M, Tolomei G, et al. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks[J]. arXiv 2021.

CF-GNNExplainer: Evaluation

- Dataset
  - tree-cycles, tree-grids, ba-shapes
  - consists of (i) a base graph, (ii) motifs that are attached to random nodes of the base graph, and (iii) additional edges that are randomly added to the overall graph.
- Task: predict whether the nodes are part of the motif
- Metric:
  - **Fidelity**: the proportion of nodes where the original predictions match the prediction for the explanations
  - **Explanation Size**: the number of removed edges
  - **Sparsity:** the proportion of edges that are removed
  - **Accuracy**: the proportion of explanations that are “correct”.
    - only compute accuracy for nodes that are predicted as in the motifs
    - consider an explanation to be correct if it exclusively involves edges that are inside the motifs

CF-GNNExplainer: Evaluation

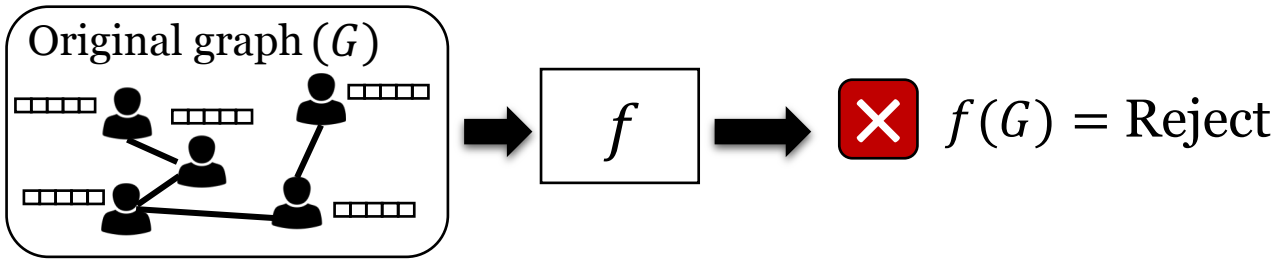
- Baselines:
  - **Random**: randomly remove edges in each subgraph, repeat K times and keep track of the most minimal perturbation resulting in a CF
  - **Only-1hop**: only keeps all edges in the 1-hop ego graph
  - **Rm-1hop**: removes all edges in the 1-hop ego graph
  - **GNNExplainer**: remove the subgraph generated by GNNExplainer (the most relevant subgraph for the prediction)

Table 1: Results comparing our method to the baselines. Below each metric, ▼ indicates a low value is desirable, while ▲ indicates a high value is desirable.

Metric	TREE-CYCLES				TREE-GRID				BA-SHAPES			
	Fid. ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲	*Fid.* ▼	*Size* ▼	*Spars.* ▲	*Acc.* ▲
RANDOM	**0.00**	4.70	0.79	0.63	**0.00**	9.06	0.75	0.77	**0.00**	503.31	0.58	0.17
ONLY-1HOP	0.32	15.64	0.13	0.45	0.32	29.30	0.09	0.72	0.60	504.18	0.05	0.18
RM-1HOP	0.46	2.11	0.89	—	0.61	2.27	0.92	—	0.21	10.56	0.97	**0.99**
GNNEXPLAINER	0.55	6.00	0.57	0.46	0.34	8.00	0.68	0.74	0.81	6.00	0.81	0.27
CF-GNNEXPLAINER	0.21	**2.09**	**0.90**	**0.94**	0.07	**1.47**	**0.94**	**0.96**	0.39	**2.39**	**0.99**	0.96

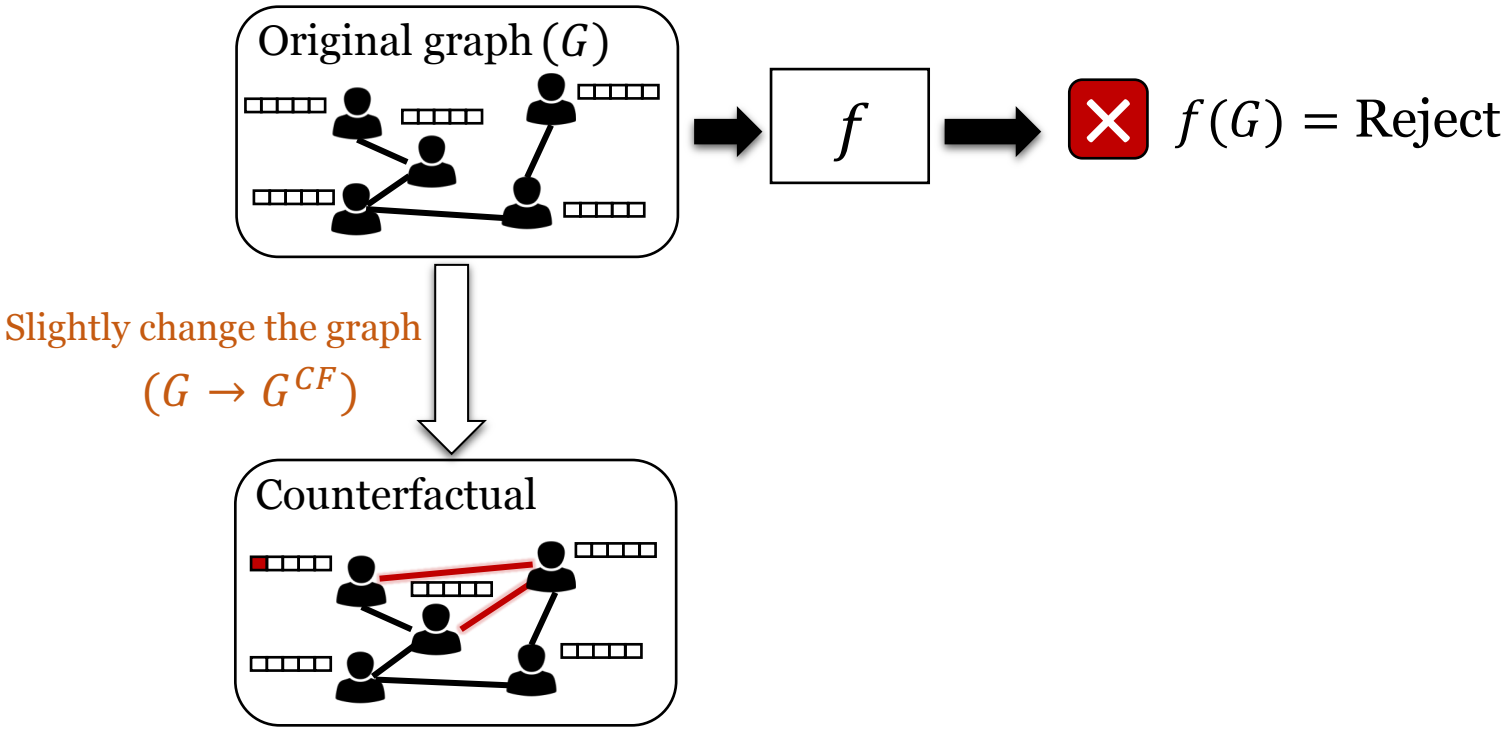
Counterfactual Explanation for Graph Classification

Example: a graph ML model f trained for grant application decision-making.



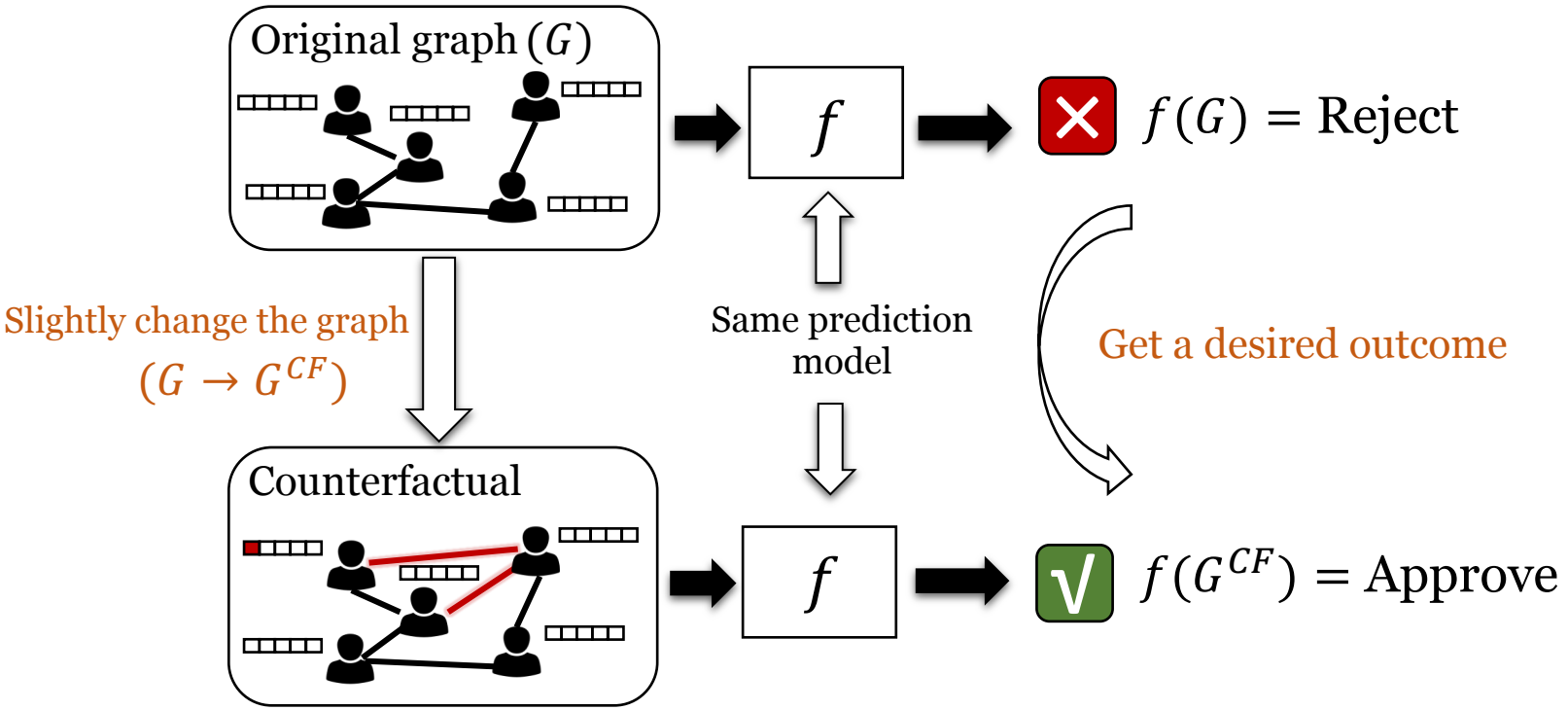
Counterfactual Explanation for Graph Classification

Example: a graph ML model f trained for grant application decision-making.



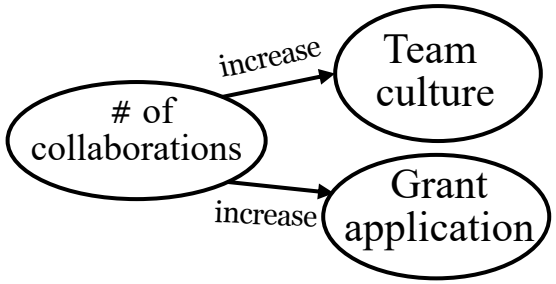
Counterfactual Explanation for Graph Classification

Example: a graph ML model f trained for grant application decision-making.

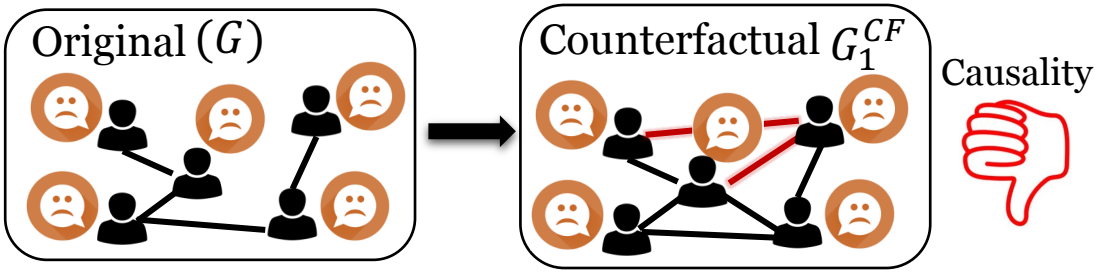


Challenges of CFE on Graphs

- Optimization
  - The space of perturbation operations on graphs (e.g., add/remove nodes/edges) is discrete, disorganized, and vast
- Generalization
  - Most existing CFE methods on graphs generate counterfactuals for each graph separately, and cannot generalize to unseen graphs
- Causality
  - It is challenging to generate counterfactuals that are consistent with the underlying causality

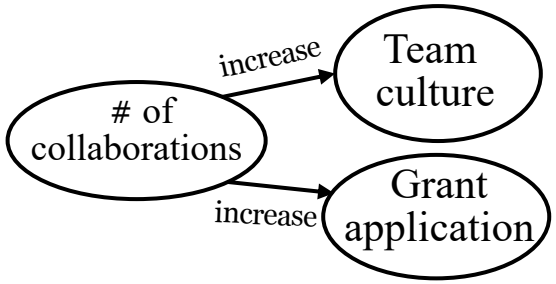


Target: Reject → Approve

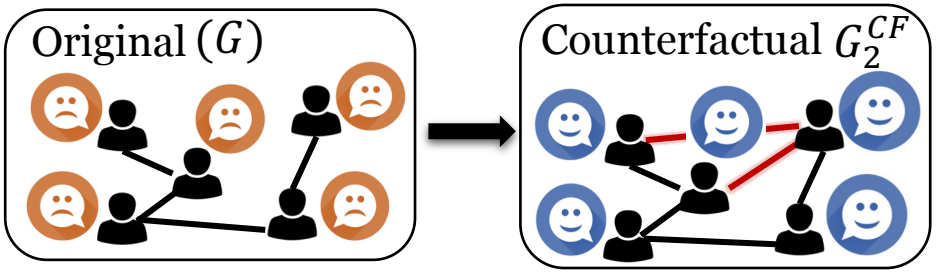


Challenges of CFE on Graphs

- Optimization
  - The space of perturbation operations on graphs (e.g., add/remove nodes/edges) is discrete, disorganized, and vast
- Generalization
  - Most existing CFE methods on graphs generate counterfactuals for each graph separately, and cannot generalize to unseen graphs
- Causality
  - It is challenging to generate counterfactuals that are consistent with the underlying causality

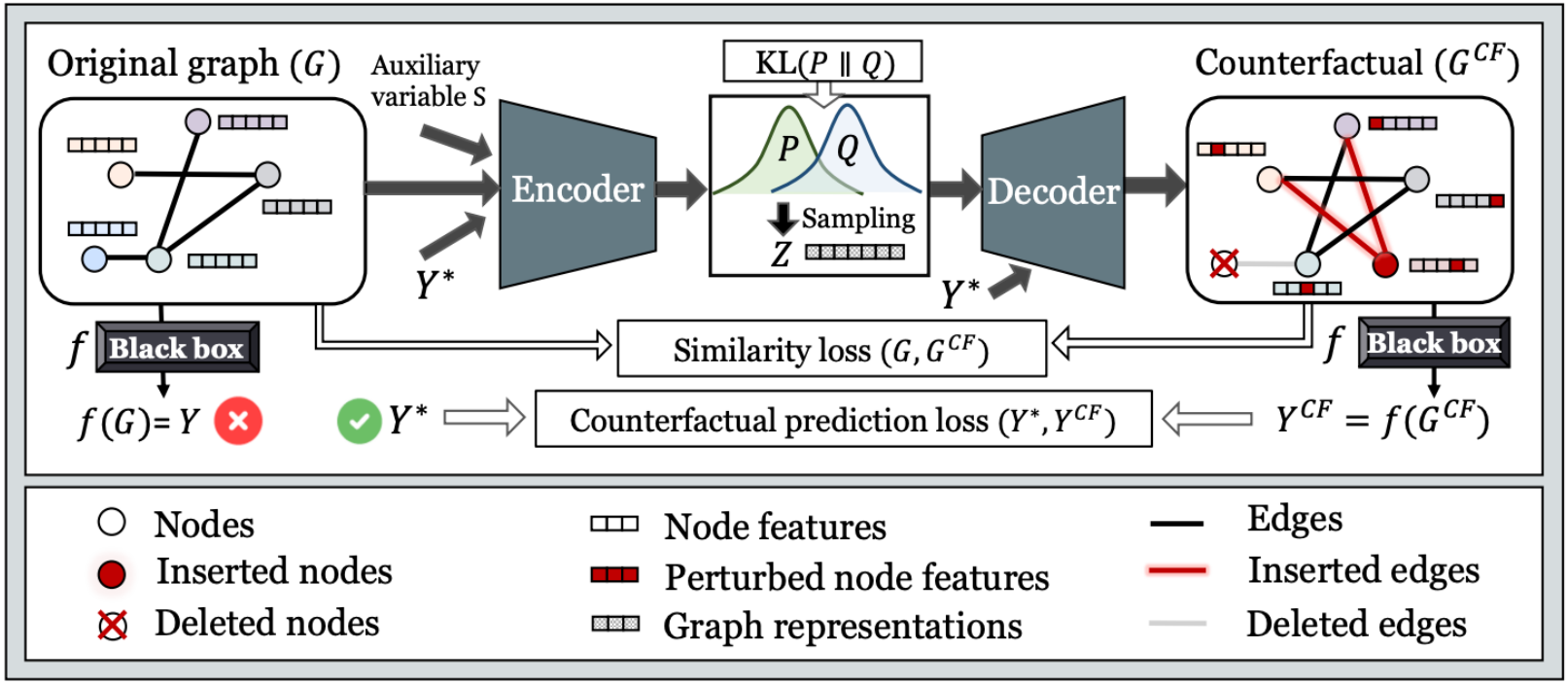


Target: Reject → Approve



CLEAR – CFE Generator for Graphs

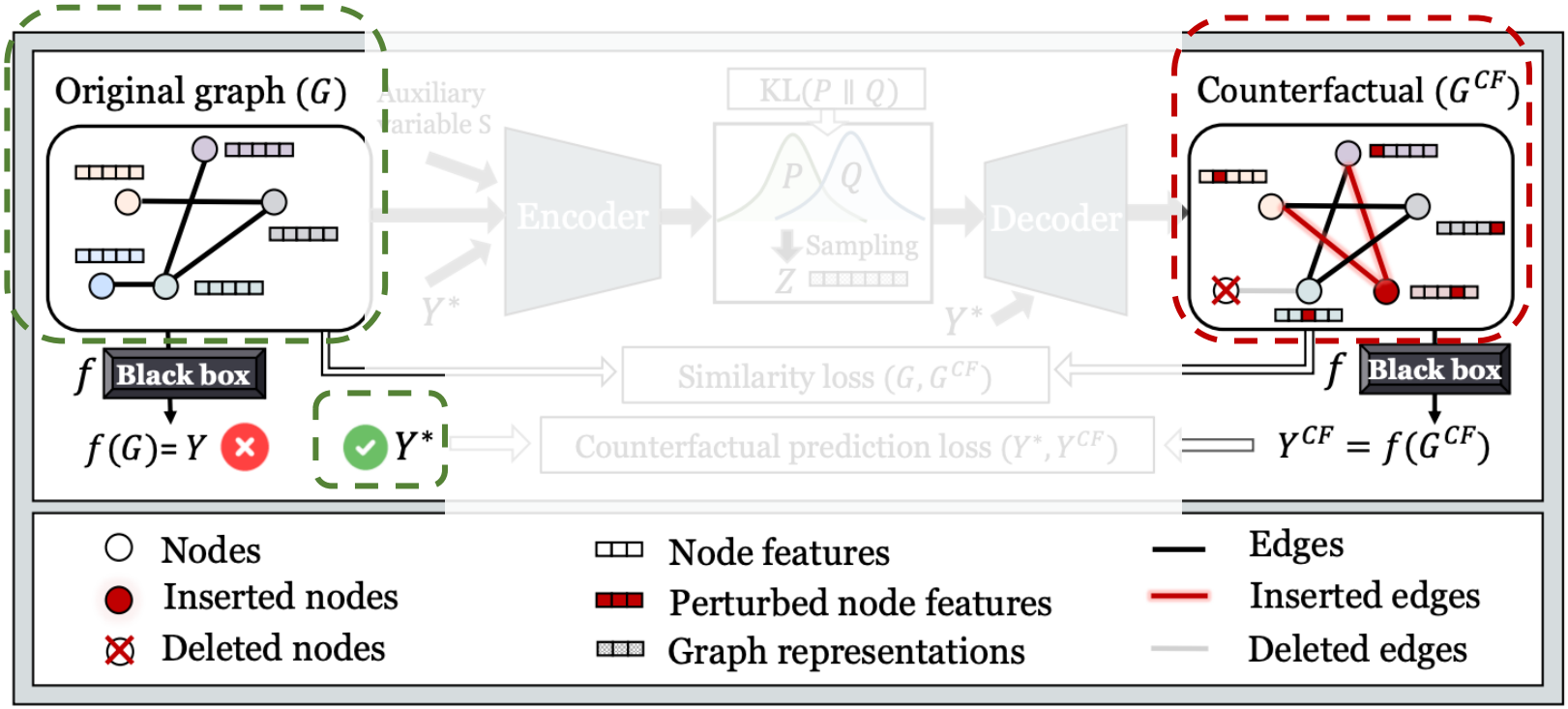
- Backbone CLEAR-VAE enables optimization and generalization on graph data
- On top of CLEAR-VAE, CLEAR promotes the causality of CFEs with an auxiliary variable S



Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

CLEAR – CFE Generator for Graphs

- CLEAR-VAE:
  - Input: original graph (G), the desired label (Y^*)
  - Output: counterfactual (G^{CF})

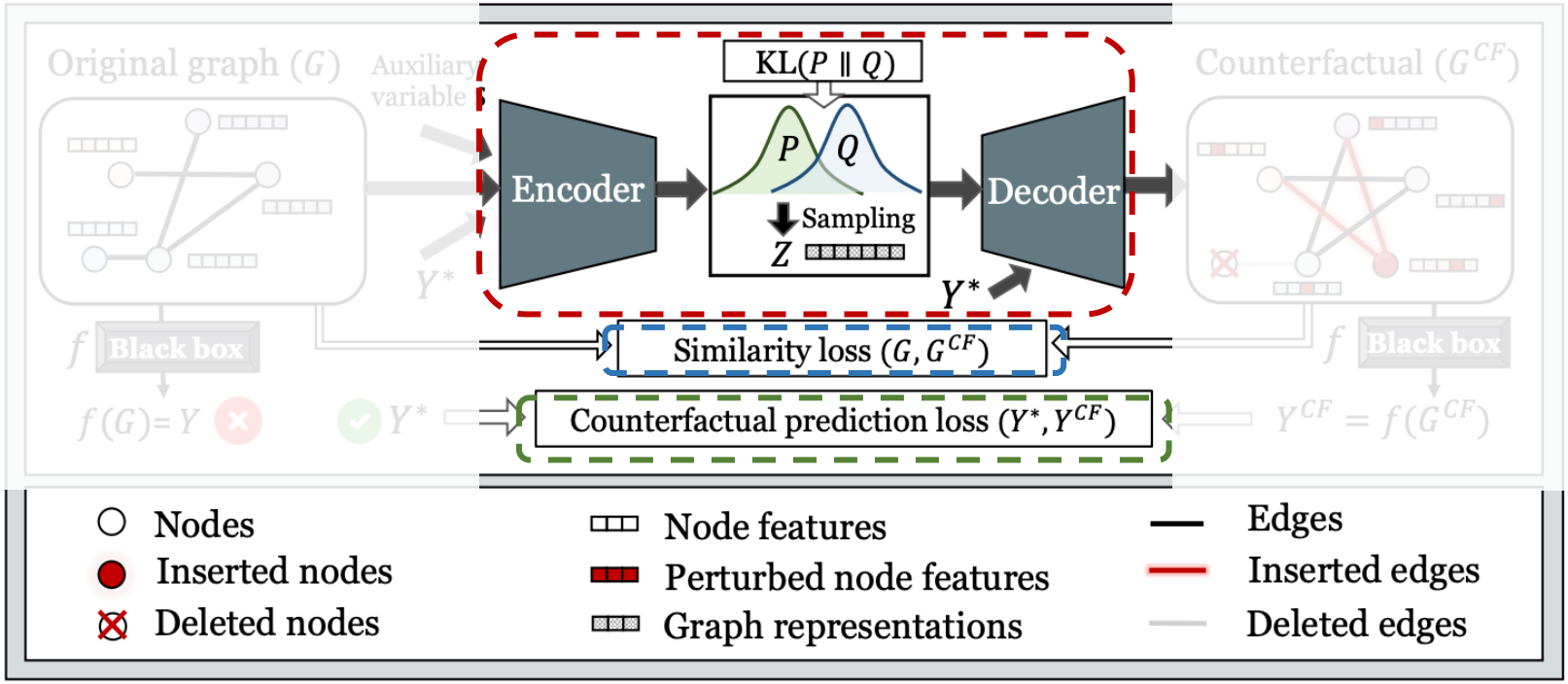


Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

CLEAR – CFE Generator for Graphs

- CLEAR-VAE: graph variational autoencoder (VAE) based, with representation Z learned in the bottleneck layer.
- Loss:

$$\mathcal{L} = \mathbb{E}_Q[d(G, G^{CF})] + \alpha \cdot l(f(G^{CF}), Y^*)] + \mathrm{KL}(Q(Z|G, Y^*) \| P(Z|G, Y^*))$$

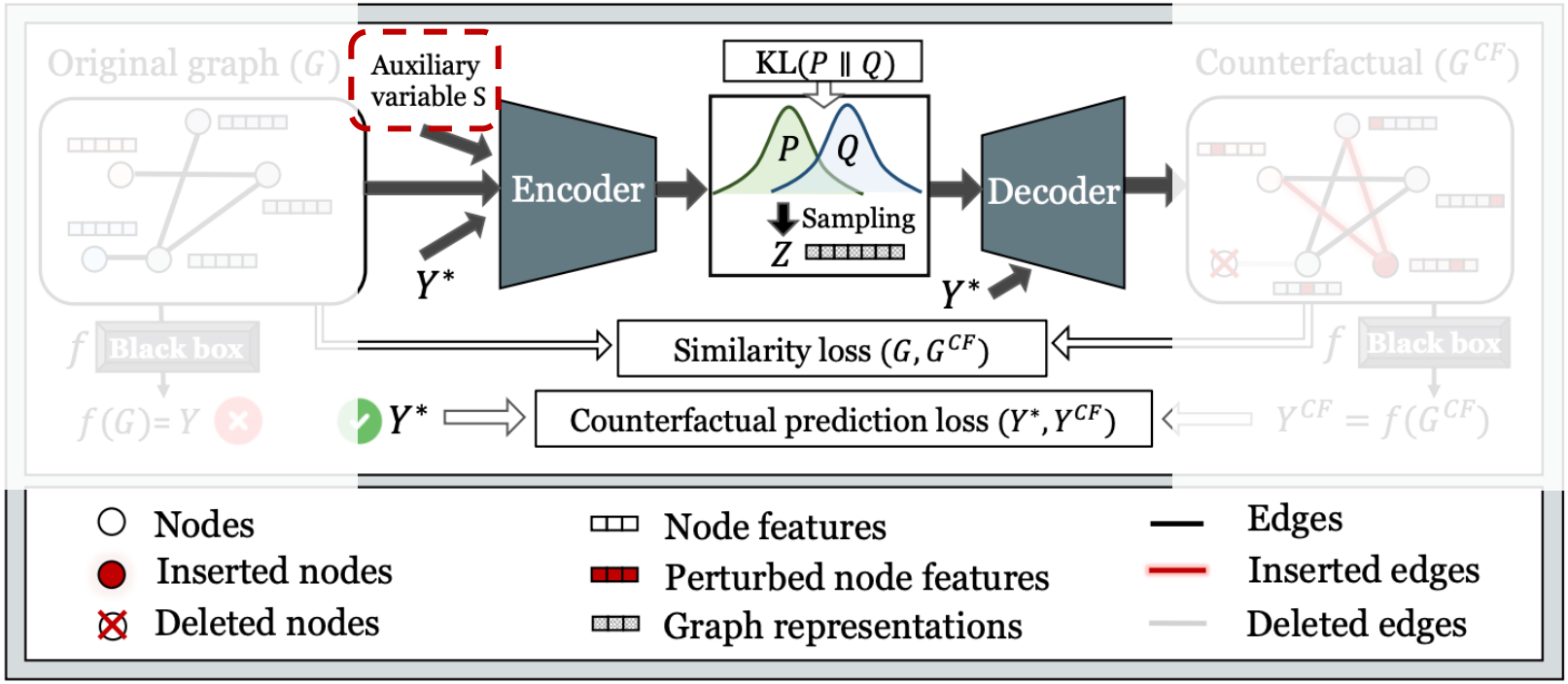


Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

CLEAR – CFE Generator for Graphs

- CLEAR: To further promote causality, we leverage an auxiliary S to better identify the underlying causal model
- Final loss:

$$\mathcal{L} = \mathbb{E}_Q[d(G, G^{CF}) + \alpha \cdot l(f(G^{CF}), Y^*)] + \mathrm{KL}(Q(Z|G, S, Y^*) \| P(Z|G, S, Y^*))$$



Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

Experiments

Ratio of counterfactuals which achieve the desired outcome

Similarity between original and counterfactual graphs

- Observations:
  - CLEAR achieves good performance in validity and proximity. (Effective CFE)
  - CLEAR can time-efficiently generate counterfactuals for a new graph.
  - CLEAR significantly outperforms all the baselines in causality.

Ratio of counterfactuals which satisfy the causal constraints of interest

Table: The performance of different methods of CFEs on graphs.

Datasets	Methods	Validity (↑)	Proximity$_X$ (↑)	Proximity$_A$ (↑)	Causality (↑)	Time (↓)
Community	Random	0.53 ± 0.05	N/A	0.77 ± 0.02	0.52 ± 0.06	0.20 ± 0.01
	EG-IST	0.53 ± 0.05	N/A	0.66 ± 0.03	0.13 ± 0.06	0.27 ± 0.03
	EG-RMV	0.55 ± 0.04	N/A	**0.85 ± 0.01**	0.03 ± 0.02	0.15 ± 0.01
	GNNExplainer	0.52 ± 0.06	N/A	0.71 ± 0.01	0.05 ± 0.00	2.87 ± 0.08
	CF-GNNExplainer	0.90 ± 0.04	N/A	0.72 ± 0.00	0.14 ± 0.02	25.14 ± 1.22
	MEG	0.88 ± 0.04	N/A	0.71 ± 0.01	0.10 ± 0.03	27.29 ± 1.32
	CLEAR(ours)	**0.94 ± 0.02**	**0.91 ± 0.01**	0.77 ± 0.00	**0.65 ± 0.03**	**0.01 ± 0.01**
Ogbg-molhiv	Random	0.48 ± 0.09	N/A	0.87 ± 0.02	0.46 ± 0.1	0.17 ± 0.02
	EG-IST	0.48 ± 0.09	N/A	0.83 ± 0.03	0.46 ± 0.09	0.19 ± 0.04
	EG-RM	0.483 ± 0.09	N/A	**0.96 ± 0.01**	0.47 ± 0.09	0.17 ± 0.04
	GNNExplainer	0.50 ± 0.01	N/A	0.92 ± 0.00	0.48 ± 0.10	2.78 ± 0.10
	CF-GNNExplainer	0.54 ± 0.02	N/A	0.92 ± 0.01	0.49 ± 0.02	27.93 ± 1.20
	MEG	0.49 ± 0.03	N/A	0.93 ± 0.01	0.50 ± 0.10	22.39 ± 2.20
	CLEAR(ours)	**0.98 ± 0.01**	**0.92 ± 0.02**	0.95 ± 0.01	**0.64 ± 0.02**	**0.01 ± 0.00**
IMDB-M	Random	0.50 ± 0.04	N/A	0.67 ± 0.01	0.43 ± 0.08	0.19 ± 0.01
	EG-IST	0.56 ± 0.12	N/A	0.67 ± 0.06	0.45 ± 0.07	0.16 ± 0.03
	EG-RM	0.45 ± 0.11	N/A	**0.75 ± 0.03**	0.53 ± 0.08	0.18 ± 0.02
	GNNExplainer	0.43 ± 0.10	N/A	0.62 ± 0.02	0.50 ± 0.02	2.46 ± 0.50
	CF-GNNExplainer	0.95 ± 0.02	N/A	0.74 ± 0.02	0.51 ± 0.02	22.21 ± 1.42
	MEG	0.90 ± 0.02	N/A	0.72 ± 0.02	0.51 ± 0.02	24.12 ± 1.08
	CLEAR(ours)	**0.96 ± 0.01**	**0.99 ± 0.00**	0.75 ± 0.01	**0.73 ± 0.01**	**0.01 ± 0.00**

Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

Experiments

- Observations:
  - Both CLEAR-VAE and CLEAR can generate valid counterfactual

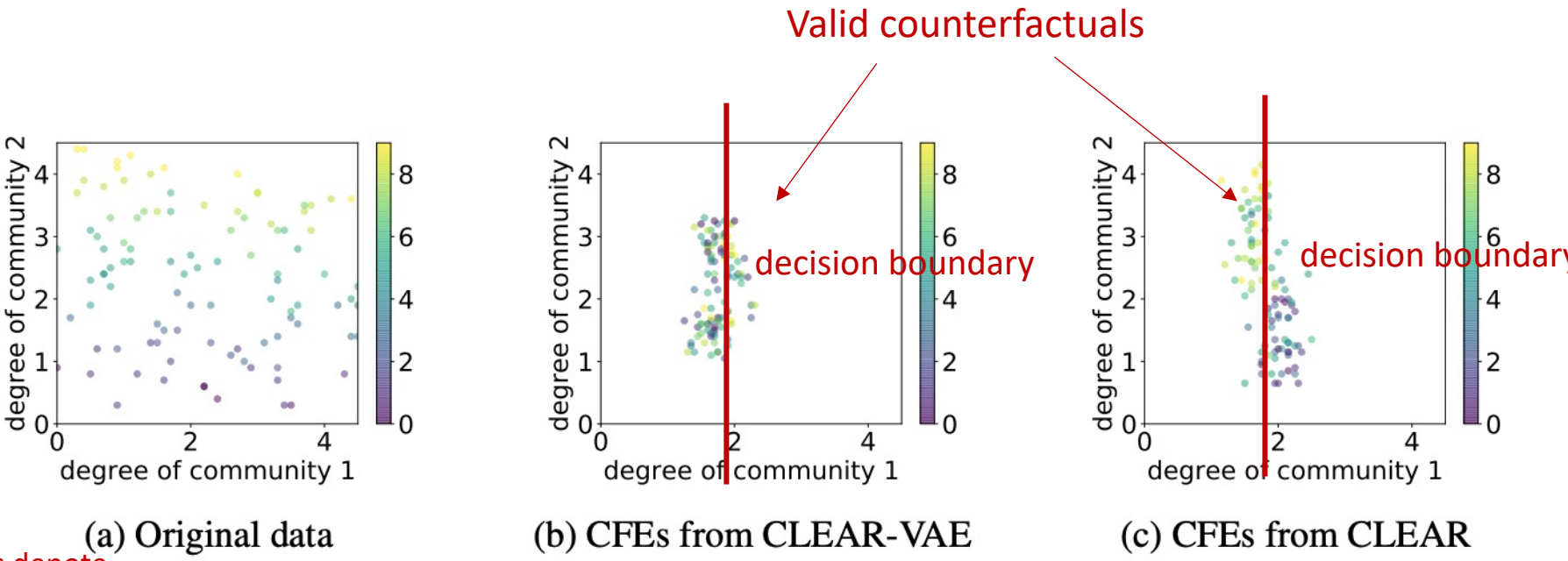


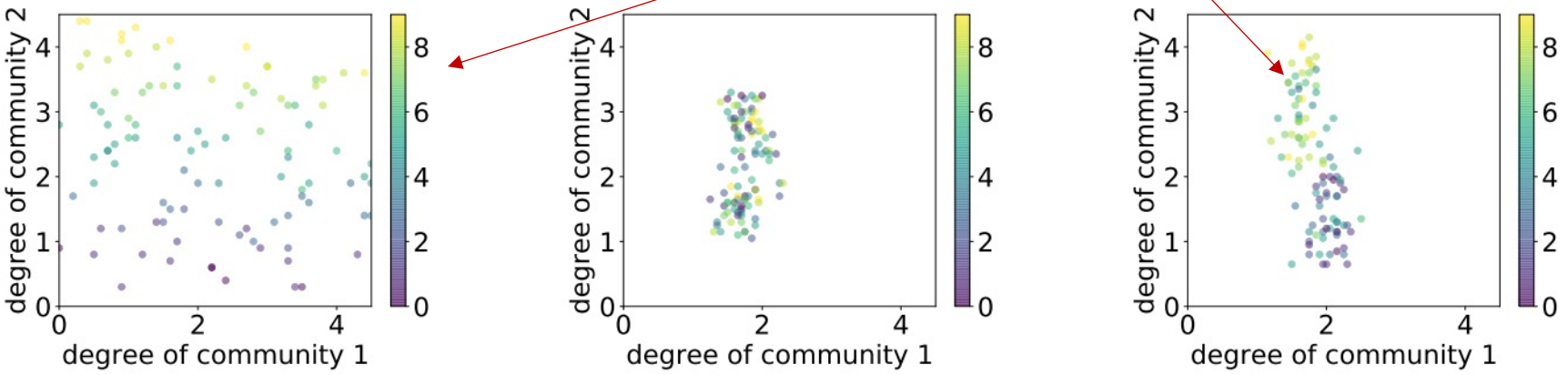
Figure: Explainability through CFEs on Community.

Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

Experiments

- Observations:
  - Both CLEAR-VAE and CLEAR can generate valid counterfactual
  - Compared with CLEAR-VAE, the counterfactuals generated by CLEAR are more consistent with the underlying causal model

S is not changed in counterfactuals



Colors denote different values of *S*

Figure: Explainability through CFEs on Community.

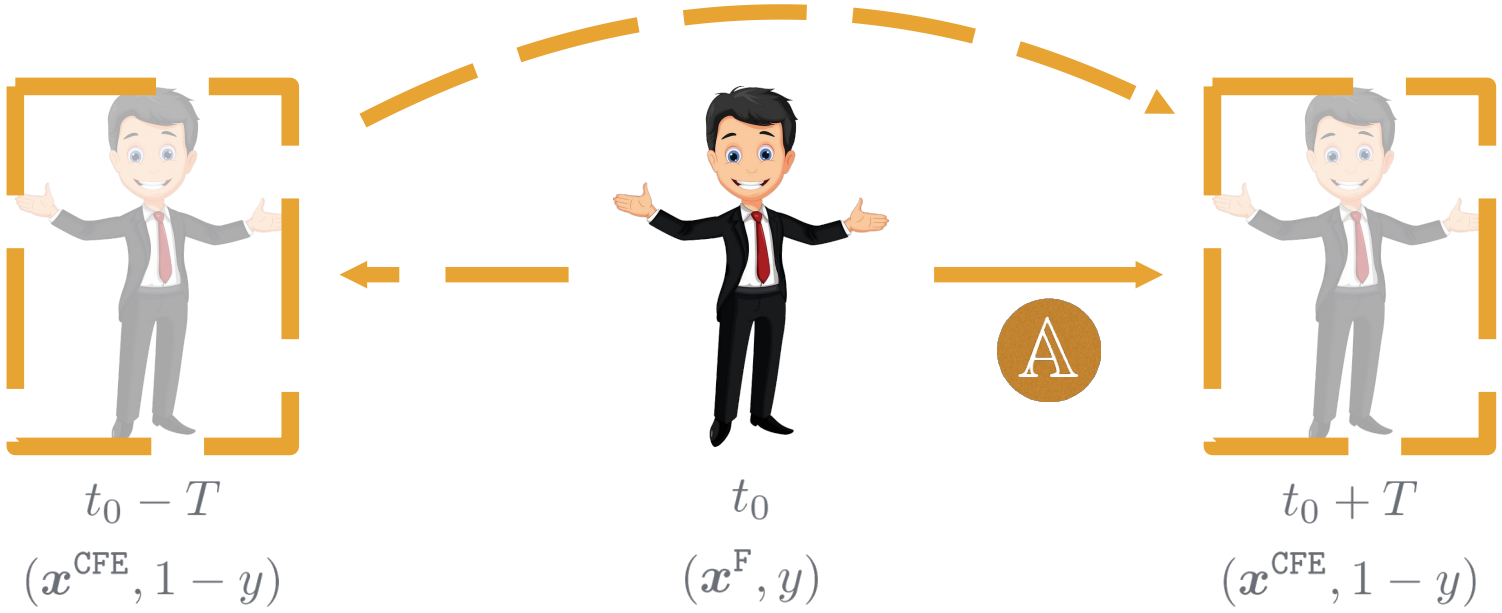
Ma, Jing, et al. "CLEAR: Generative Counterfactual Explanations on Graphs." *NeurIPS* (2022).

Outline

- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability
- Model-based interpretability (attribution)

Algorithmic Recourse

One of the primary objectives of "explanations as a means to help a data-subject act rather than merely understand." *[Wachter et al., 2017]*



Algorithmic Recourse

the actions required for, or "the systematic process of reversing unfavorable decisions by algorithms & bureaucracies across a range of counterfactual scenarios" *[Venkata-subramanian & Alfano, 2020]*

CFE-based Recourse

Counterfactual Explanation

$$\boldsymbol{x}_*^{\texttt{CFE}} \in \operatorname{argmin}_{\boldsymbol{x}} \quad \operatorname{dist}(\boldsymbol{x}, \boldsymbol{x}^{\texttt{F}})$$
$$\text{s.t.} \quad h_\theta(\boldsymbol{x}) \neq h_\theta(\boldsymbol{x}^{\texttt{F}})$$
$$\boldsymbol{x} \in \mathcal{P}\text{lausible} \ ,$$

[Wachter et al., 2017]

CFE-based Recourse

$$\boldsymbol{\delta}^* \in \operatorname{argmin}_{\boldsymbol{\delta}} \quad \operatorname{cost}(\boldsymbol{\delta}; \boldsymbol{x}^{\texttt{F}})$$
$$\text{s.t.} \quad h_\theta(\boldsymbol{x}^{\texttt{CFE}}) \neq h_\theta(\boldsymbol{x}^{\texttt{F}})$$
$$\boldsymbol{x}^{\texttt{CFE}} = \boldsymbol{x}^{\texttt{F}} + \boldsymbol{\delta}$$
$$\boldsymbol{x}^{\texttt{CFE}} \in \mathcal{P}\text{lausible}$$
$$\boldsymbol{\delta} \in \mathcal{F}\text{easible} \ ,$$

[Ustun et al., 2019]

Question: Do CFE-based recourse actions translate to optimal and feasible real-world actions for recourse?

Sub-optimal Recourse

- Issue: Optimal changes may not be feasible!
  - e.g., “decrease your age”

Question: Do CFE-based recourse actions translate to ~~optimal~~ and feasible real-world actions for recourse?

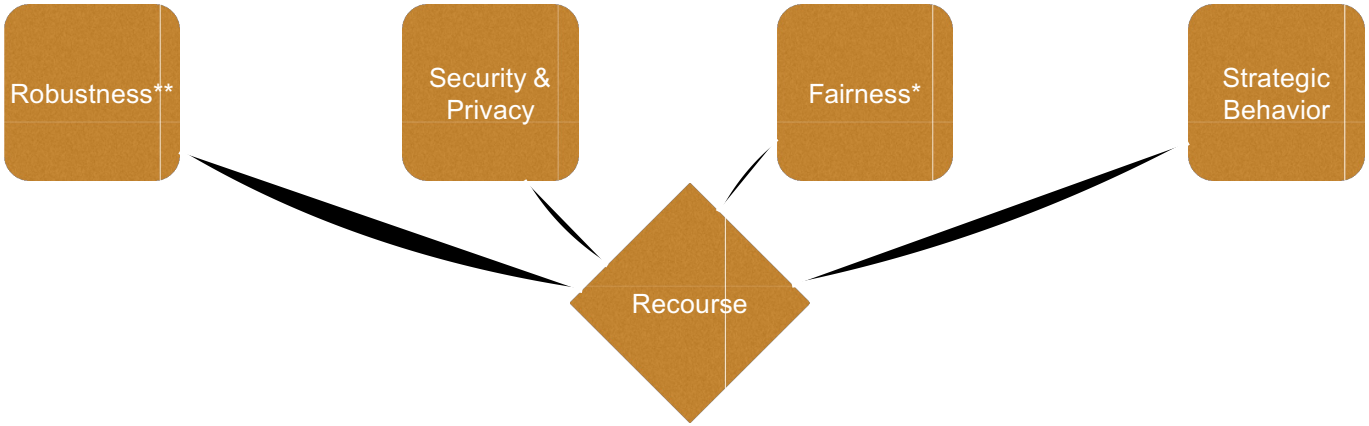
Recourse & Other Ethical Desiderata

Researcher & Developer

Owner & Deployer

Examiner & Regulator

Data-subjects & Decision-subjects

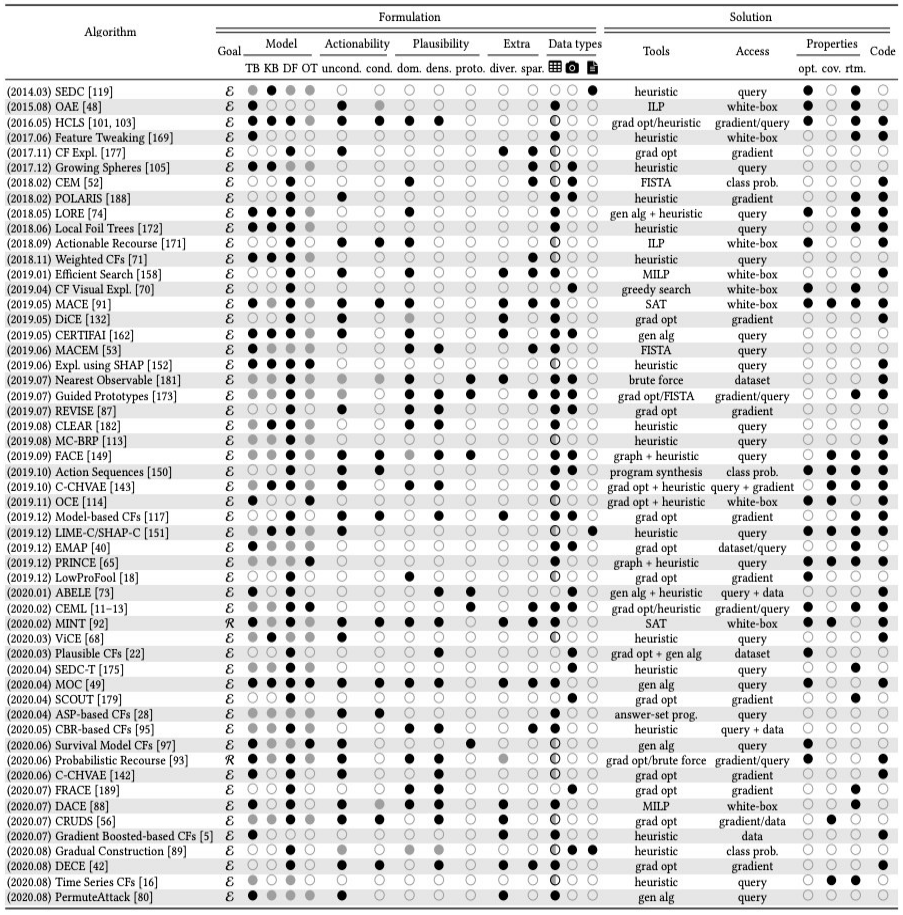


*[JvK, Karimi, et al., AAAI (oral), 2022]

**[RDO, Karimi, et al., ICML (spotlight), 2022]

Recourse Summary

- In general, counterfactual explanations **do not, imply** feasible actions.
- In general, **recourse can only be guaranteed** under perfect causal knowledge.
- **Diversity of stakeholder needs** illustrates a tension between many desirable system properties.
- Trade-offs between different needs may require new techniques, perhaps on the **cross-disciplinary expertise.**



Algorithm	Formulation														Solution					
	Goal	Model				Actionability		Plausibility			Extra		Data types				Properties			Code
		TB	KB	DF	OT	uncond.	cond.	dom.	dens.	proto.	diver.	spar.	▦	📷	📄			opt.	cov.	rtm.	
(2014.03) SEDC [119]	$\mathcal{E}$	○	●	◍	◍	○	○	○	○	○	○	○	○	○	●	heuristic	query	●	○	●	○
(2015.08) OAE [48]	$\mathcal{E}$	●	○	○	○	●	◍	○	○	○	○	○	●	○	○	ILP	white-box	●	○	●	○
(2016.05) HCLS [101, 103]	$\mathcal{E}$	●	●	●	◍	●	●	●	●	○	○	○	◐	○	○	grad opt/heuristic	gradient/query	●	○	●	●
(2017.06) Feature Tweaking [169]	$\mathcal{E}$	●	○	○	○	○	○	○	○	○	○	○	●	○	○	heuristic	white-box	○	○	●	●
(2017.11) CF Expl. [177]	$\mathcal{E}$	○	○	●	○	●	○	○	○	○	●	●	◐	○	○	grad opt	gradient	○	○	○	○
(2017.12) Growing Spheres [105]	$\mathcal{E}$	●	●	◍	◍	○	○	○	○	○	○	●	◐	●	○	heuristic	query	○	○	○	○
(2018.02) CEM [52]	$\mathcal{E}$	○	○	●	○	○	○	●	○	○	○	●	◐	●	○	FISTA	class prob.	○	○	○	●
(2018.02) POLARIS [188]	$\mathcal{E}$	○	○	◍	○	●	○	○	○	○	○	○	●	●	○	heuristic	gradient	○	○	●	●
(2018.05) LORE [74]	$\mathcal{E}$	●	●	●	◍	○	○	●	○	○	○	○	●	○	○	gen alg + heuristic	query	●	○	●	●
(2018.06) Local Foil Trees [172]	$\mathcal{E}$	●	●	●	◍	○	○	○	○	○	○	○	●	○	○	heuristic	query	○	○	●	●
(2018.09) Actionable Recourse [171]	$\mathcal{E}$	○	○	●	○	●	●	●	○	○	○	○	◐	○	○	ILP	white-box	●	○	○	●
(2018.11) Weighted CFs [71]	$\mathcal{E}$	●	●	●	◍	○	○	○	○	○	○	●	◐	○	○	heuristic	query	○	○	○	○
(2019.01) Efficient Search [158]	$\mathcal{E}$	○	○	●	○	●	○	●	○	○	●	●	●	○	○	MILP	white-box	○	○	○	●
(2019.04) CF Visual Expl. [70]	$\mathcal{E}$	○	○	●	○	○	○	○	○	○	○	○	○	●	○	greedy search	white-box	●	○	●	○
(2019.05) MACE [91]	$\mathcal{E}$	●	◍	●	◍	●	●	●	○	○	●	●	●	○	○	SAT	white-box	●	●	●	●
(2019.05) DiCE [132]	$\mathcal{E}$	○	○	●	○	●	○	◍	○	○	●	○	●	○	○	grad opt	gradient	○	○	○	●
(2019.05) CERTIFAI [162]	$\mathcal{E}$	●	●	●	◍	●	○	●	○	○	●	○	●	●	○	gen alg	query	○	○	○	○
(2019.06) MACEM [53]	$\mathcal{E}$	●	◍	◍	◍	○	○	●	●	○	○	●	●	○	○	FISTA	query	○	○	○	○
(2019.06) Expl. using SHAP [152]	$\mathcal{E}$	●	●	●	●	○	○	○	○	○	○	○	◐	○	○	heuristic	query	○	○	○	●
(2019.07) Nearest Observable [181]	$\mathcal{E}$	◍	◍	●	◍	◍	◍	●	○	●	●	○	●	●	○	brute force	dataset	○	○	○	●
(2019.07) Guided Prototypes [173]	$\mathcal{E}$	◍	◍	●	◍	◍	○	●	●	●	○	●	●	●	○	grad opt/FISTA	gradient/query	○	○	●	●
(2019.07) REVISE [87]	$\mathcal{E}$	○	○	●	○	●	○	●	●	○	○	○	●	●	○	grad opt	gradient	○	○	○	○
(2019.08) CLEAR [182]	$\mathcal{E}$	◍	●	●	◍	○	○	●	●	○	○	○	●	○	○	heuristic	query	○	○	○	●
(2019.08) MC-BRP [113]	$\mathcal{E}$	○	◍	●	◍	○	○	○	○	○	○	○	◐	○	○	heuristic	query	○	○	○	●
(2019.09) FACE [149]	$\mathcal{E}$	◍	◍	●	◍	●	●	◍	●	●	○	○	●	●	○	graph + heuristic	query	○	●	●	●
(2019.10) Action Sequences [150]	$\mathcal{E}$	○	○	●	○	●	●	○	○	○	○	○	●	●	○	program synthesis	class prob.	●	●	●	●
(2019.10) C-CHVAE [143]	$\mathcal{E}$	◍	●	●	◍	●	○	●	●	○	○	○	●	○	○	grad opt + heuristic	query + gradient	○	●	●	●
(2019.11) OCE [114]	$\mathcal{E}$	●	○	○	●	○	○	○	○	○	○	○	◐	○	○	grad opt + heuristic	white-box	●	●	○	●
(2019.12) Model-based CFs [117]	$\mathcal{E}$	○	○	●	○	●	●	○	●	○	●	○	●	●	○	grad opt	gradient	○	○	●	●
(2019.12) LIME-C/SHAP-C [151]	$\mathcal{E}$	◍	●	●	◍	●	○	○	○	○	○	○	◐	○	●	heuristic	query	●	●	●	●
(2019.12) EMAP [40]	$\mathcal{E}$	●	◍	◍	◍	○	○	○	○	○	○	○	●	●	○	grad opt	dataset/query	○	○	●	○
(2019.12) PRINCE [65]	$\mathcal{E}$	◍	◍	◍	●	○	○	○	○	○	○	○	●	○	○	graph + heuristic	query	●	●	●	●
(2019.12) LowProFool [18]	$\mathcal{E}$	○	○	●	○	○	○	●	○	○	○	○	◐	○	○	grad opt	gradient	●	○	○	○
(2020.01) ABELE [73]	$\mathcal{E}$	●	○	◍	○	○	○	○	●	●	○	○	○	●	○	gen alg + heuristic	query + data	○	○	○	●
(2020.02) CEML [11–13]	$\mathcal{E}$	◍	◍	●	●	○	○	○	○	●	○	●	●	●	○	grad opt/heuristic	gradient/query	●	○	●	●
(2020.02) MINT [92]	$\mathcal{R}$	●	◍	●	◍	●	●	●	●	○	●	●	●	○	○	SAT	white-box	●	●	○	●
(2020.03) ViCE [68]	$\mathcal{E}$	◍	●	◍	◍	●	○	○	○	○	○	○	◐	○	○	heuristic	query	○	○	○	●
(2020.03) Plausible CFs [22]	$\mathcal{E}$	○	○	●	○	○	○	○	●	○	○	○	○	●	○	grad opt + gen alg	dataset	●	○	○	○
(2020.04) SEDC-T [175]	$\mathcal{E}$	◍	◍	●	◍	○	○	○	○	○	○	○	○	●	○	heuristic	query	○	○	●	○
(2020.04) MOC [49]	$\mathcal{E}$	●	●	●	●	●	●	●	●	○	●	●	●	○	○	gen alg	query	●	○	○	●
(2020.04) SCOUT [179]	$\mathcal{E}$	○	○	●	○	○	○	○	○	○	○	○	○	●	○	grad opt	gradient	○	○	●	○
(2020.04) ASP-based CFs [28]	$\mathcal{E}$	◍	◍	◍	◍	●	●	○	○	○	○	○	●	○	○	answer-set prog.	query	○	○	○	○
(2020.05) CBR-based CFs [95]	$\mathcal{E}$	◍	◍	●	◍	○	○	●	●	○	○	●	●	○	○	heuristic	query + data	○	○	○	○
(2020.06) Survival Model CFs [97]	$\mathcal{E}$	●	◍	◍	●	●	○	○	○	●	○	○	◐	○	○	gen alg	query	●	○	○	○
(2020.06) Probabilistic Recourse [93]	$\mathcal{R}$	●	◍	●	◍	●	○	●	●	○	◍	○	◐	○	○	grad opt/brute force	gradient/query	●	○	○	●
(2020.06) C-CHVAE [142]	$\mathcal{E}$	●	○	●	○	●	○	○	●	○	○	○	◐	○	○	grad opt	gradient	○	○	○	●
(2020.07) FRACE [189]	$\mathcal{E}$	○	○	●	○	○	○	●	●	○	○	○	○	●	○	grad opt	gradient	○	○	●	○
(2020.07) DACE [88]	$\mathcal{E}$	●	○	●	○	●	◍	●	●	○	●	○	●	○	○	MILP	white-box	○	○	●	○
(2020.07) CRUDS [56]	$\mathcal{E}$	◍	◍	●	◍	●	●	○	●	○	●	○	◐	○	○	grad opt	gradient/data	○	●	○	○
(2020.07) Gradient Boosted-based CFs [5]	$\mathcal{E}$	●	○	○	○	○	○	○	○	○	●	○	●	○	○	heuristic	data	○	○	○	●
(2020.08) Gradual Construction [89]	$\mathcal{E}$	○	○	●	◍	○	○	◍	◍	○	○	○	◐	●	●	heuristic	class prob.	○	○	○	○
(2020.08) DECE [42]	$\mathcal{E}$	○	○	●	○	●	●	○	●	○	●	●	●	○	○	grad opt	gradient	○	○	○	●
(2020.08) Time Series CFs [16]	$\mathcal{E}$	◍	○	◍	○	○	○	○	○	○	○	○	◐	○	○	heuristic	query	○	●	●	○
(2020.08) PermuteAttack [80]	$\mathcal{E}$	●	◍	◍	◍	●	○	○	○	○	●	○	●	○	○	gen alg	query	○	○	○	○

[Karimi et al., ACM Computing Surveys, 2022]

What remains to be done?

Owner & Deployer

Researcher & Developer

Examiner & Regulator

Underlying model:
classification / regression —> matching / scheduling?

Variables:
meaningful concepts —> learn / abstract concepts?

Stakeholders:
is recourse enough? —> fair, robust, private recourse?

Interaction:
single-step recommendation —> multi-step interventions?

World knowledge:
implausible or costly actions —> collaborative discussion?

Outline

- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability (Discovery)
- Model-based interpretability (attribution)

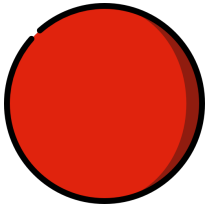
Explaining a car to a 4-year-old

What's a car?

A car is a machine with **wheels** that can take you from one place to another. It usually has seats for people to sit in and a steering wheel for the driver to control where the car goes. […] People use cars to go to work, go on vacations, or visit their friends and family.

What's a "wheel"?

Wheels are **circular**, flat objects that are attached to the bottom of a car. They help the car move by rolling along the ground. Think of a wheel like a **circular** plate that is thick in the middle and thinner on the edges. The edge of the wheel is called the "rim," and it's usually made of […]



What's a "circle"?

A circle is a **shape** that is round like a ball or a tire. It has no corners or edges, and all points on its edge are the same distance away from the center. […] Circles are all around us, from the **sun** and **moon** in the sky to the wheels on a car or **bicycle**.

Concepts

Explanations are effective when presented in a *language* consistent with that of the explainee

Language is built atop a set of *mutually understandable concepts*

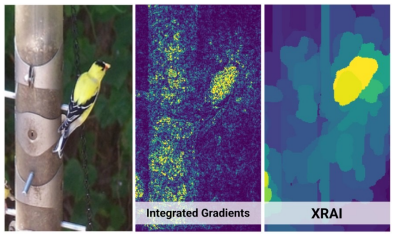
Concepts subsume

- tone
- colors
- shapes
- Patterns
- processes & demonstrations

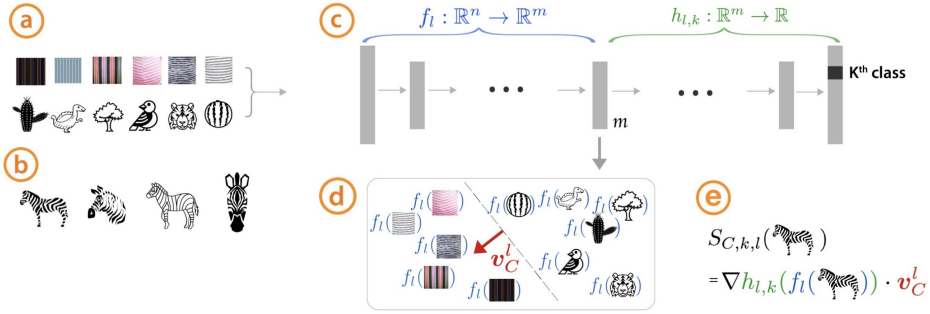
good explanations are built atop good concepts

The *explanation-concept alignment* spectrum

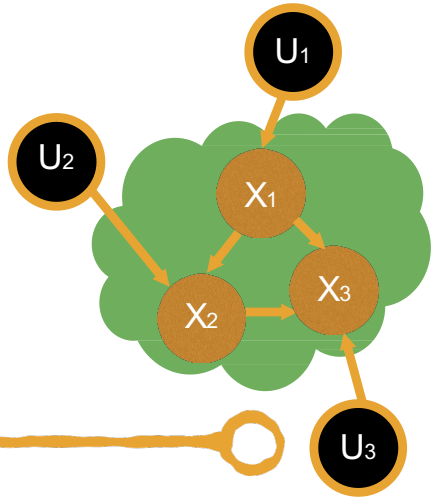
Concepts *unclear*



Concepts *"extracted"*



Concepts *clear*



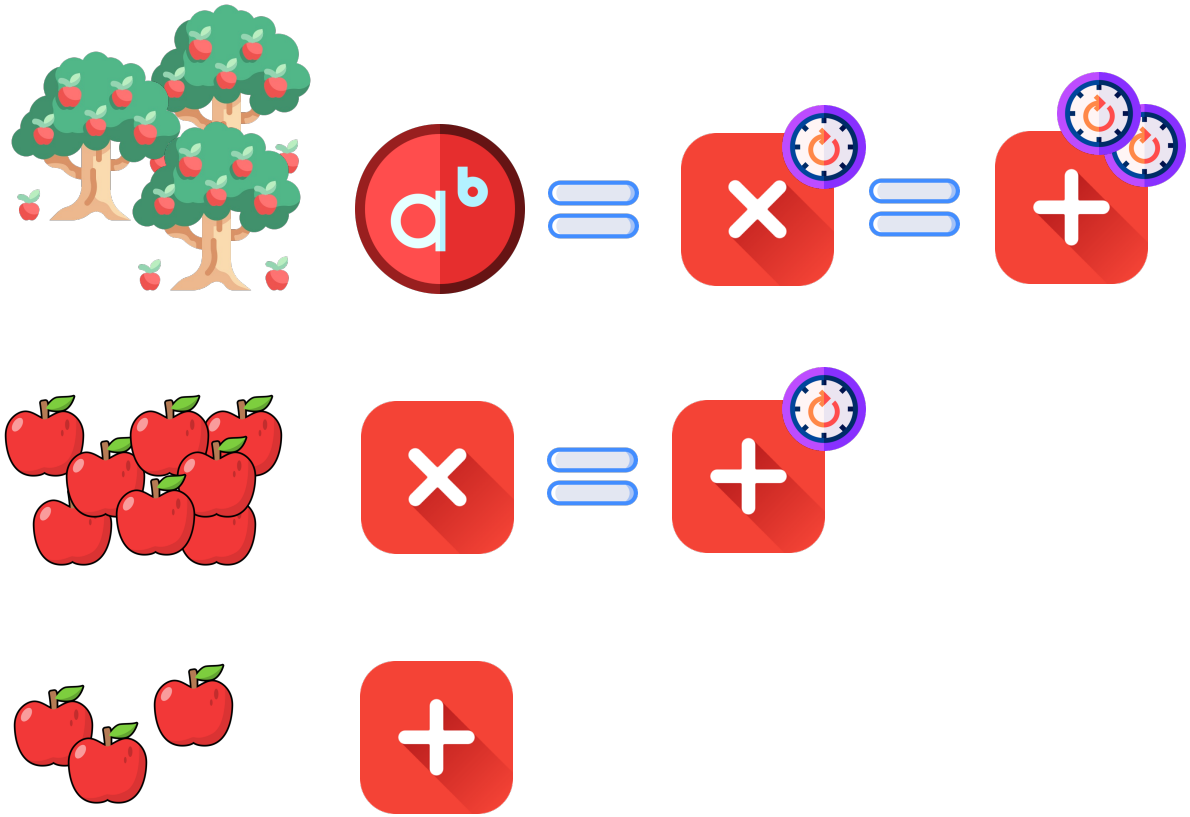
Attribution-based Explanations

Testing with Concept Activation Vectors
[Kim et al., ICML, 2018]

Recourse-based Explanations

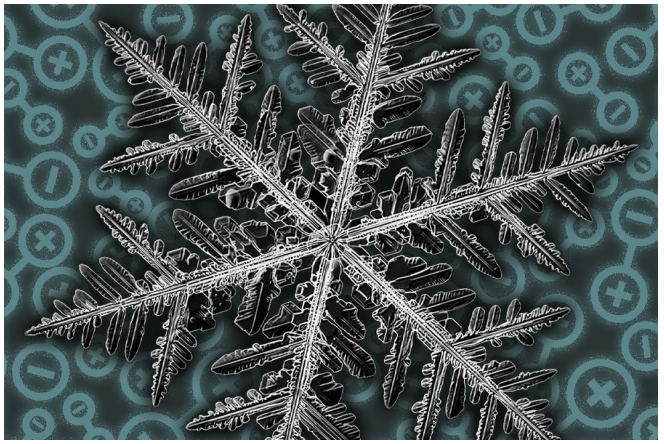
Ideally, an AI will expand our concept bank

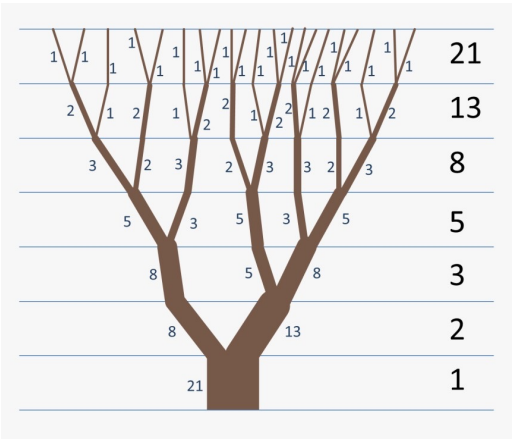
Learning concepts from first principles

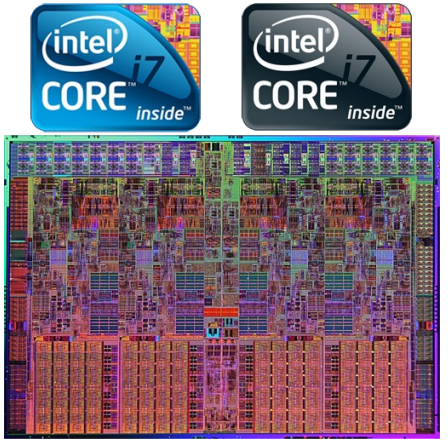


```
def pow(x: int, y: int) -> int:
    out = 1
    for _ in range(y)
        out = mult(out, x)
    return out


def mult(x: int, y: int) -> int:
    out = 0
    for _ in range(y)
        out = add(out, x)
    return out


def add(x: int, y: int) -> int:
    return x + y
```

Abstractions







Causal Discovery for XAI

High-dimensional set of "low-level" features (e.g., pixels)

$X = (X_1, X_2, \ldots, X_q)$

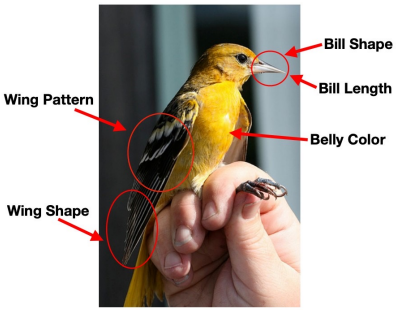


Fig. 1: An image of a Baltimore Oriole annotated with interpretable features.

Low-dim set of "high-level" features (human interpretable)

$Z = (Z_1, Z_2, \ldots, Z_p)$

Assume dataset with X covariates, Y targets, and Z annotations

- Train a classifier f : X -> Y
- Estimate a causal PAG G over $V = (Z, Y = f(X))$ $(q \gg p)$
- Determine which high-level features Z are
  - causes
  - potential causes
  - non-causes of Y

PAG Edge	Meaning
$Z_i \rightarrow \widehat{Y}$	Z_i is a cause of $\widehat{Y}$
$Z_i \leftrightarrow \widehat{Y}$	Z_i and $\widehat{Y}$ share an unmeasured common cause $Z_i \leftarrow U \rightarrow \widehat{Y}$
$Z_i \circ\!\!\rightarrow \widehat{Y}$	Either Z_i is a cause of $\widehat{Y}$ or there is unmeasured confounding, or both

[Sani et al., 2020]

Sample Experiment

Each d x d image may contain

- Horizontal bar (H)
- Vertical bar (V)
- Circle (C)
- Rectangle (R)
- Random pixel noise

$$Y = V \vee C$$

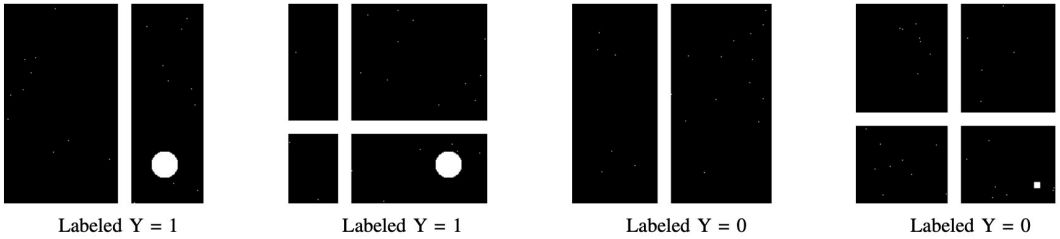
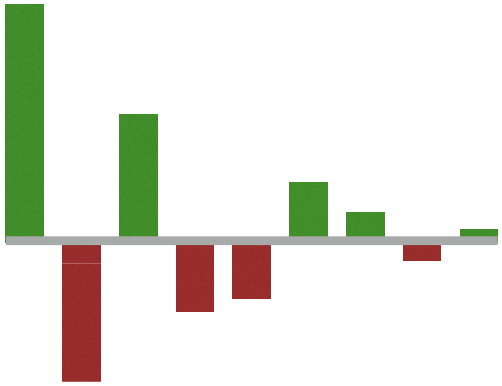


Fig. 3: Simulated image examples with horizontal bars, vertical bars, circles, and rectangles.

Fig. 4: (a) A causal diagram representing the true data generating process. (b) The PAG learned using FCI with output $\widehat{Y}$ from a convolutional neural network.

Outline

- Causal explanation: general introduction
- Decision based interpretability
  - Counterfactual explanation
  - Recourse
- Data-based interpretability (Discovery)
- Model-based interpretability (attribution)



Attribution, causal or not

Non-causal Attribution

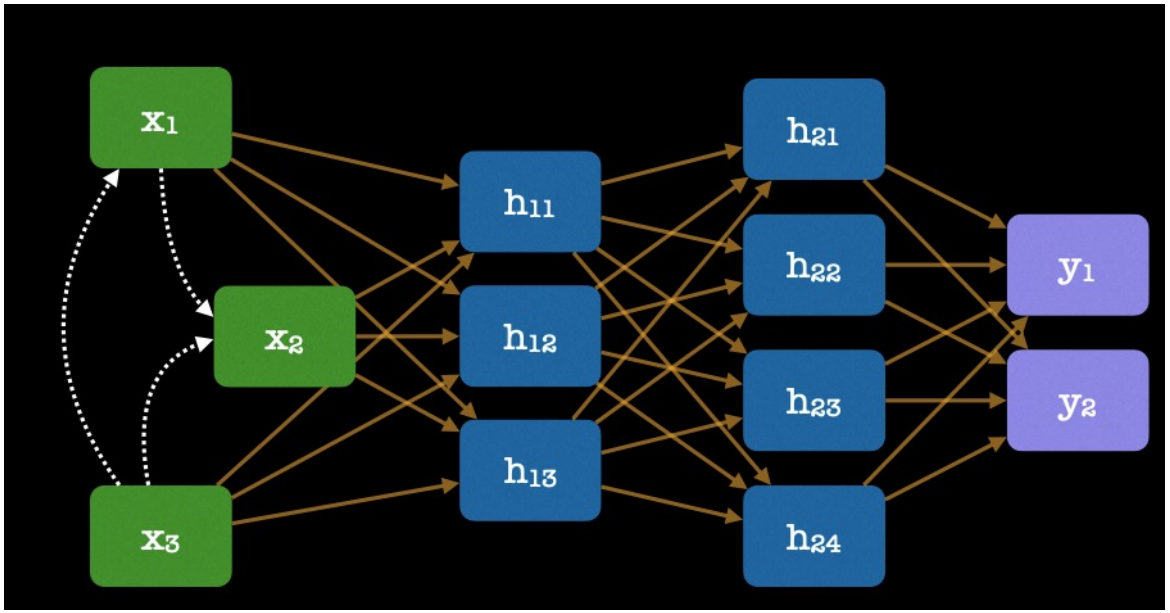
How much would perturbing a particular input neuron affect a particular output neuron?

$$E[Y_k \mid X_i = x_i] - E[Y_k \mid X_i = x_j]$$

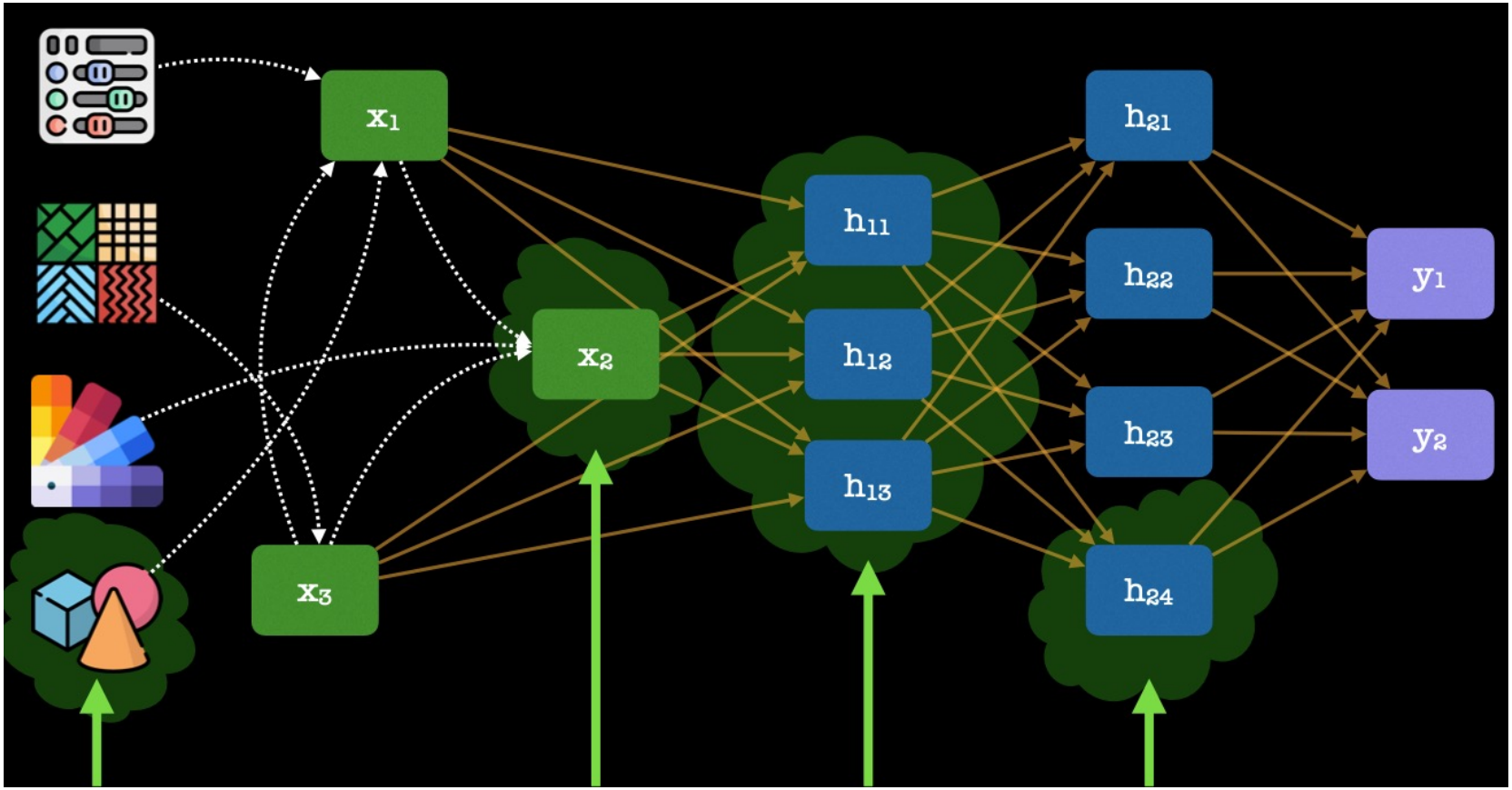
Causal Attribution

What is the causal effect of a particular input neuron on a particular output neuron?

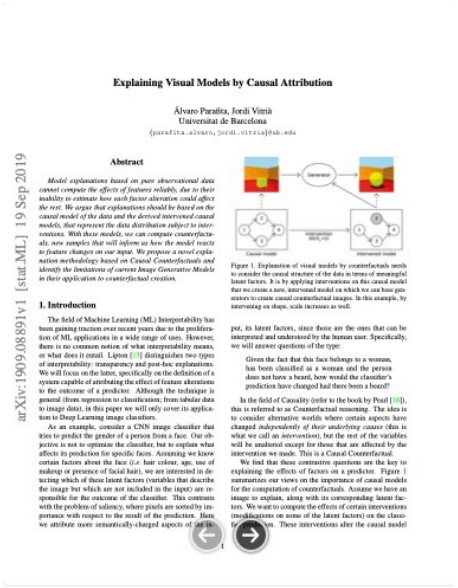
$$E[Y_k \mid do(X_i = x_i)] - E[Y_k \mid do(X_i = x_j)]$$



$$E[Y_k|\, do(X_i = x_i)\,] - E[Y_k|\, do(X_i = x_j)\,]$$



Attribution Literature



Explaining Visual Models by Causal Attribution

Álvaro Parafita, Jordi Vitrià
Universitat de Barcelona
{parafita.alvaro,jordi.vitria}@ub.edu

Abstract

Model explanations based on pure observational data cannot compute the effects of features reliably, due to their inability to estimate how each factor alteration could affect the rest. We argue that explanations should be based on the causal model of the data and the derived intervened causal models, that represent the data distribution subject to interventions. With these models, we can compute counterfactuals, new samples that will inform us how the model reacts to feature changes on our input. We propose a novel explanation methodology based on Causal Counterfactuals and identify the limitations of current Image Generative Models in their application to counterfactual creation.

1. Introduction

The field of Machine Learning (ML) Interpretability has been gaining traction over recent years due to the proliferation of ML applications in a wide range of uses. However, there is no common notion of what interpretability means, or what does it entail. Lipton [13] distinguishes two types of interpretability: transparency and post-hoc explanations. We will focus on the latter, specifically on the definition of a system capable of attributing the effect of feature alterations to the outcome of a predictor. Although the technique is general (from regression to classification; from tabular data to image data), in this paper we will only cover its application to Deep Learning image classifiers.

As an example, consider a CNN image classifier that tries to predict the gender of a person from a face. Our objective is not to optimize the classifier, but to explain what affects its prediction for specific faces. Assuming we know certain factors about the face (*i.e.* hair colour, age, use of makeup or presence of facial hair), we are interested in detecting which of these latent factors (variables that describe the image but which are not included in the input) are responsible for the outcome of the classifier. This contrasts with the problem of saliency, where pixels are sorted by importance with respect to the result of the prediction. Here we attribute more semantically-charged aspects of the

Figure 1. Explanation of visual models by counterfactuals needs to consider the causal structure of the data in terms of meaningful latent factors. It is by applying interventions on this causal model that we create a new, intervened model on which we can base guarantees to create causal counterfactual images. In this example, by intervening on shape, scale increases as well.

put, its latent factors, since those are the ones that can be interpreted and understood by the human user. Specifically, we will answer questions of the type:

> Given the fact that this face belongs to a woman, has been classified as a woman and the person does not have a beard, how would the classifier's prediction have changed had there been a beard?

In the field of Causality (refer to the book by Pearl [16]), this is referred to as Counterfactual reasoning. The idea is to consider alternative worlds where certain aspects have changed *independently of their underlying causes* (this is what we call an *intervention*), but the rest of the variables will be unaltered except for those that are affected by the intervention we made. This is a Causal Counterfactual.

We find that these contrastive questions are the key to explaining the effects of factors on a predictor. Figure 1 summarizes our views on the importance of causal models for the computation of counterfactuals. Assume we have an image to explain, along with its corresponding latent factors. We want to compute the effects of certain interventions (modifications on some of the latent factors) on the classifier's prediction. These interventions alter the causal model

[Parafita et al., 2019]

Explaining Classifiers with Causal Concept Effect (CaCE)

Yash Goyal[1]* Amir Feder[2]* Uri Shalit[2] Been Kim[3]

Abstract

How can we understand classification decisions made by deep neural networks? Many existing explainability methods rely solely on correlations and fail to account for confounding, which may result in potentially misleading explanations. To overcome this problem, we define the Causal Concept Effect (CaCE) as the causal effect of (the presence or absence of) a human-interpretable concept on a deep neural net's predictions. We show that the CaCE measure can avoid errors stemming from confounding. Estimating CaCE is difficult in situations where we cannot easily simulate the do-operator. To mitigate this problem, we use a generative model, specifically a Variational Autoencoder (VAE), to measure VAE-CaCE. In an extensive experimental analysis, we show that the VAE-CaCE is able to estimate the true concept causal effect, compared to baselines for a number of datasets including high dimensional images.

1. Introduction

The rise of machine learning use in many practical applications has brought up a new challenge: how to interpret and understand the reasons behind a model's prediction. Particularly in high-risk domains such as medicine, security, etc., it has been widely recognized that understanding the model's reasoning for a prediction is one of the crucial components for wide and safe adoption of the technology.

The machine learning community has been responding to this demand. Many approaches have been proposed to tackle the challenge: for example by developing a model with interpretable components built-in (Kim et al., 2014; Lei[1] & Rudin, 2014) or by building post-training interpretability methods (Ribeiro et al., 2016; Smilkov et al., 2017; Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Chang et al., 2018; Kim et al., 2018). While these methods may be useful in showing features or concepts that are correlated with a model's prediction, their explanations might be confounded by correlations present in the data which are not causally relevant to the model, as we describe now.

Say we wish to explain what drives classification decisions for an entire class by a deep neural network, e.g. "what drives the decision to classify an image as BICYCLE?". Now consider the following case: within the training dataset, there is 0.8 correlation between the presence of cars and the presence of bicycles. However, the dataset is diverse enough and the classifier is powerful enough such that it does not rely on the presence of cars in order to classify bicycles: If we were to take images containing bicycles and edit out the cars, we will find that the classifier's output for the label BICYCLE is virtually unchanged. Even so, as we show below, the strong correlation between cars and bicycles can lead many interpretability methods to wrongfully give the concept CAR as an explanation for classifying bicycles. In this work, we attempt to tease out the causal aspect: does the presence of a concept like CAR actually change the classifier's output. The case of "editing out the cars" is an example of what is known as the *do*-operator (Pearl, 2009): it formalizes the act of intervening in the world, an act which lies in the heart of defining and understanding causal effects.

In this paper we propose explaining classifiers with the Causal Concept Effect (CaCE) for high-level concepts whose presence or absence (everything else being equal) affect the model's prediction, as opposed to merely being correlated with the model's prediction. CaCE is a global explanation method, where the goal is to explain a model's prediction for an entire class, rather than individual data points (i.e., local explanation methods). As global methods aim to summarize all data points, they are much more vulnerable to confounding of concepts. By concept, we mean a higher level unit than low level, individual input features such as pixels. The example of the cars and bicycles above illustrates the issue. Concepts are often highly correlated with each other in datasets, and we want our explanations to zero-in on the concepts whose presence or absence in isolation causally affects the model's output.

One of the challenges in estimating CaCE for high dimensional data such as images is that there is often no easy way to directly perform the intervention of adding or removing a concept, as in the example above regarding CARS and bi-

[1]Georgia Tech [2]Technion [3]Google Brain. *Work partially done while the first author was an intern at Google Brain. Correspondence to: Yash Goyal <ygoyal@gatech.edu>.

Preprint. Under review.

[Goyal et al., 2020]

Neural Network Attributions: A Causal Perspective

Aditya Chattopadhyay[1] Piyushi Manupriya[2] Anirban Sarkar[2] Vineeth N Balasubramanian[2]

Abstract

We propose a new attribution method for neural networks developed using first principles of causality (to the best of our knowledge, the first such). The neural network architecture is viewed as a Structural Causal Model, and a methodology to compute the causal effect of each feature on the output is presented. With reasonable assumptions on the causal structure of the input data, we propose algorithms to efficiently compute the causal effects, as well as scale the approach to data with large dimensionality. We also show how this method can be used for recurrent neural networks. We report experimental results on both simulated and real datasets showcasing the promise and usefulness of the proposed algorithm.

1. Introduction

Over the last decade, deep learning models have been highly successful in solving complex problems in various fields ranging from vision, speech to more core fields such as chemistry and physics (Deng et al., 2014; Sadowski et al., 2014; Gilmer et al., 2017). However, a key bottleneck in accepting such models in real-life applications, especially risk-sensitive ones, is the "interpretability problem". Usually, these models are treated as black boxes without any knowledge of their internal workings. This makes troubleshooting difficult in case of erroneous behaviour. Moreover, these algorithms are trained on a limited amount of data which most often is different from real-world data. Artifacts that creep into the training dataset due to human error or unwarranted correlations in data creation have an adverse effect on the hypothesis learned by these models. If treated as black boxes, there is no way of knowing whether the model actually learned a concept or a high accuracy was just fortuitous. This limitation of black-box deep learned models has paved way for a new paradigm, "explainable machine learning".

While the field is nascent, several broad approaches have emerged (Simonyan et al., 2013; Yosinski et al., 2015; Frosst & Hinton, 2017; Letham et al., 2015), each having its own perspective to explainable machine learning. In this work, we focus on a class of interpretability algorithms called "attribution-based methods". Formally, attributions are defined as the effect of an input feature on the prediction function's output (Sundararajan et al., 2017). This is an inherently causal question, which motivates this work. Current approaches involve backpropogating the signals to input to decipher input-output relations (Sundararajan et al., 2017; Selvaraju et al., 2016; Buch et al., 2015; Ribeiro et al., 2016) or approximating the local decision boundary (around the input data point in question) via "interpretable" regressors like linear classifiers (Ribeiro et al., 2016; Selvaraju et al., 2016; Zhou & Troyanskaya, 2015; Alvarez-Melis & Jaakkola, 2017) or decision trees.

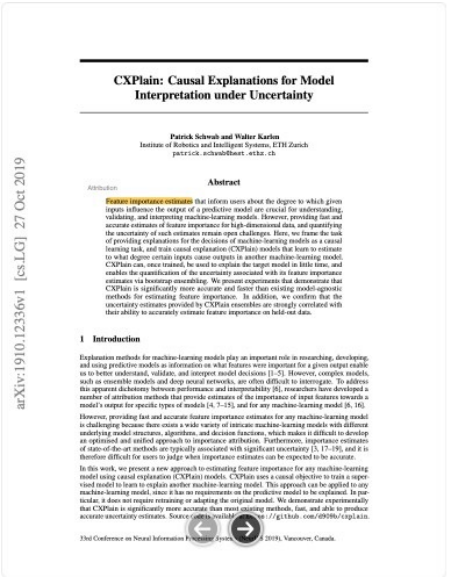
In the former category of methods, while gradients answer the question "How much would perturbing a particular input affect the output?", they do not capture the causal influence of an input on a particular output neuron. The latter category of methods that rely on "interpretable" regression is also prone to artifacts as regression primarily maps correlations rather than causation. In this work, we propose a neural network attribution methodology built from first principles of causality. To the best of our knowledge, while neural networks have been modeled as causal graphs (Kocaoglu et al., 2017), this is the first effort on a *causal approach to attribution in neural networks*.

Our approach views the neural network as a Structural Causal Model (SCM), and proposes a new method to compute the Average Causal Effect of an input neuron on an output neuron. Using standard principles of causality to make the problem tractable, this approach induces a setting where input neurons are not causally related to each other, but can be jointly caused by a latent confounder (say, data-generating mechanisms). This setting is valid in many application domains that use neural networks, including images where neighboring pixels are often affected jointly by a latent confounder, rather than direct causal influence (a "dog" can take a paintbrush and oddly color a certain part of an image and the neighboring pixels need not change). We

[1]Center for Imaging Science, Johns Hopkins University, Baltimore, USA. [2]Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, Telangana, India. Correspondence to: Aditya Chattopadhyay <achatto10@jhu.edu>, Vineeth N Balasubramanian <vineethnb@iith.ac.in>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[Chattopadhyay et al., 2019]



CXPlain: Causal Explanations for Model Interpretation under Uncertainty

Patrick Schwab and Walter Karlen
Institute of Robotics and Intelligent Systems, ETH Zurich
patrick.schwab@hest.ethz.ch

Abstract

Feature importance estimates that inform users about the degree to which given inputs influence the output of a predictive model are crucial for understanding, validating, and interpreting machine-learning models. However, providing fast and accurate estimates of feature importance for high-dimensional data, and quantifying the uncertainty of such estimates remain open challenges. Here, we frame the task of providing explanations for the decisions of machine-learning models as a causal learning task, and train causal explanation (CXPlain) models that learn to estimate to what degree certain inputs cause outputs in another machine-learning model. CXPlain can, once trained, be used to explain the target model in little time, and enables the quantification of the uncertainty associated with its feature importance estimates via bootstrap ensembling. We present experiments that demonstrate that CXPlain is significantly more accurate and faster than existing model-agnostic methods for estimating feature importance. In addition, we confirm that the uncertainty estimates provided by CXPlain ensembles are strongly correlated with their ability to accurately estimate feature importance on held-out data.

1 Introduction

Explanation methods for machine-learning models play an important role in researching, developing, and using predictive models as information on what features were important for a given output enable us to better understand, validate, and interpret model decisions [1–5]. However, complex models, such as ensemble models and deep neural networks, are often difficult to interrogate. To address this apparent dichotomy between performance and interpretability [6], researchers have developed a number of attribution methods that provide estimates of the importance of input features towards a model's output for specific types of models [4, 7–15], and for any machine-learning model [6, 16].

However, providing fast and accurate feature importance estimates for any machine-learning model is challenging because there exists a wide variety of intricate machine-learning models with different underlying model structures, algorithms, and decision functions, which makes it difficult to develop an optimized and unified approach to importance attribution. Furthermore, importance estimates of state-of-the-art methods are typically associated with significant uncertainty [3, 17–19], and it is therefore difficult for users to judge when importance estimates can be expected to be accurate.

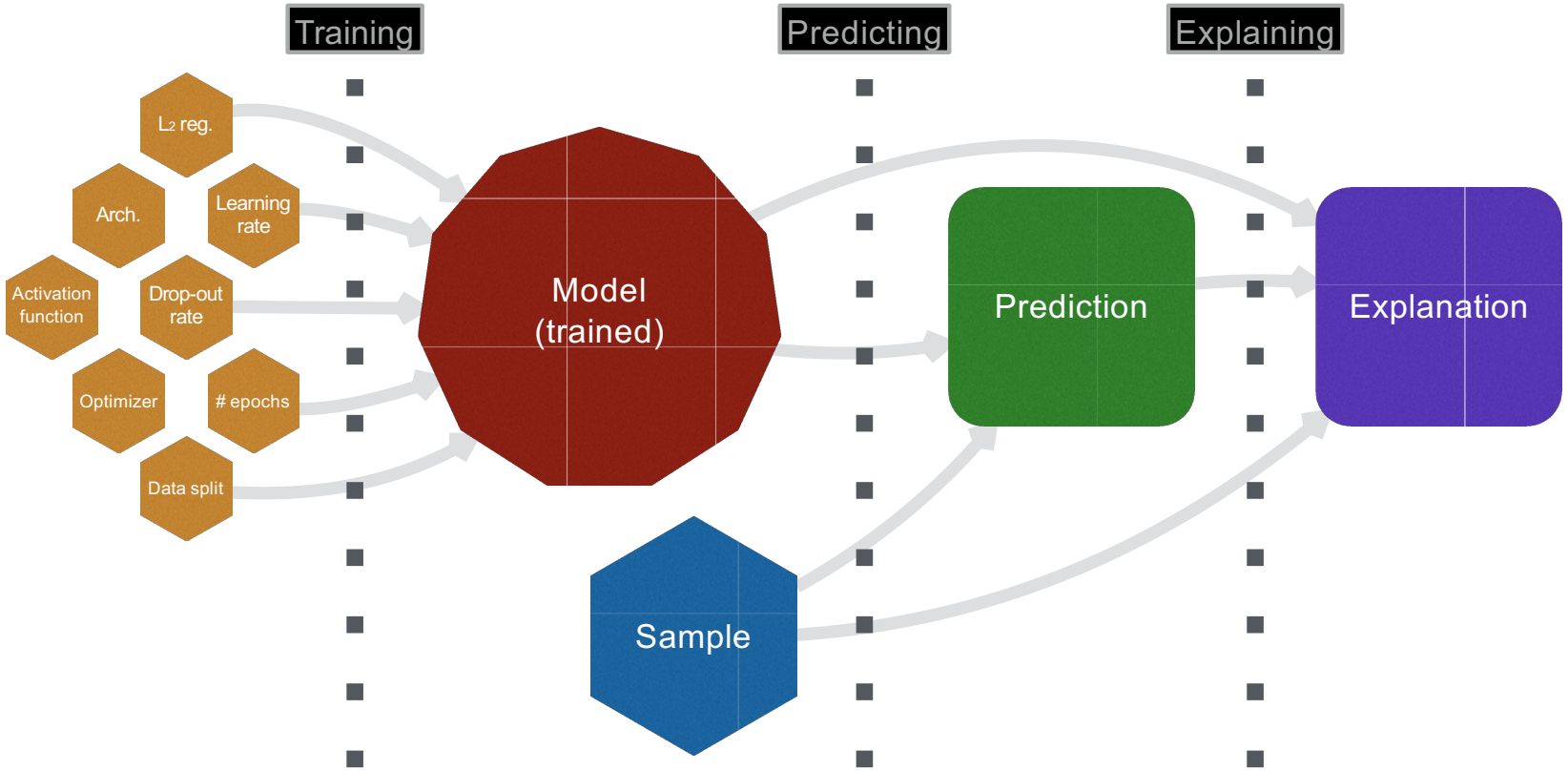
In this work, we present a new approach to estimating feature importance for any machine-learning model using causal explanation (CXPlain) models. CXPlain uses a causal objective to train a supervised model to learn to explain another machine-learning model. This approach can be applied to any machine-learning model, since it has no requirements on the predictive model to be explained. In particular, it does not require retraining or adapting the original model. We demonstrate experimentally that CXPlain is significantly more accurate than most existing methods, fast, and able to produce accurate uncertainty estimates. Source code is available at https://github.com/d909b/cxplain.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[Schwab et al., 2019]

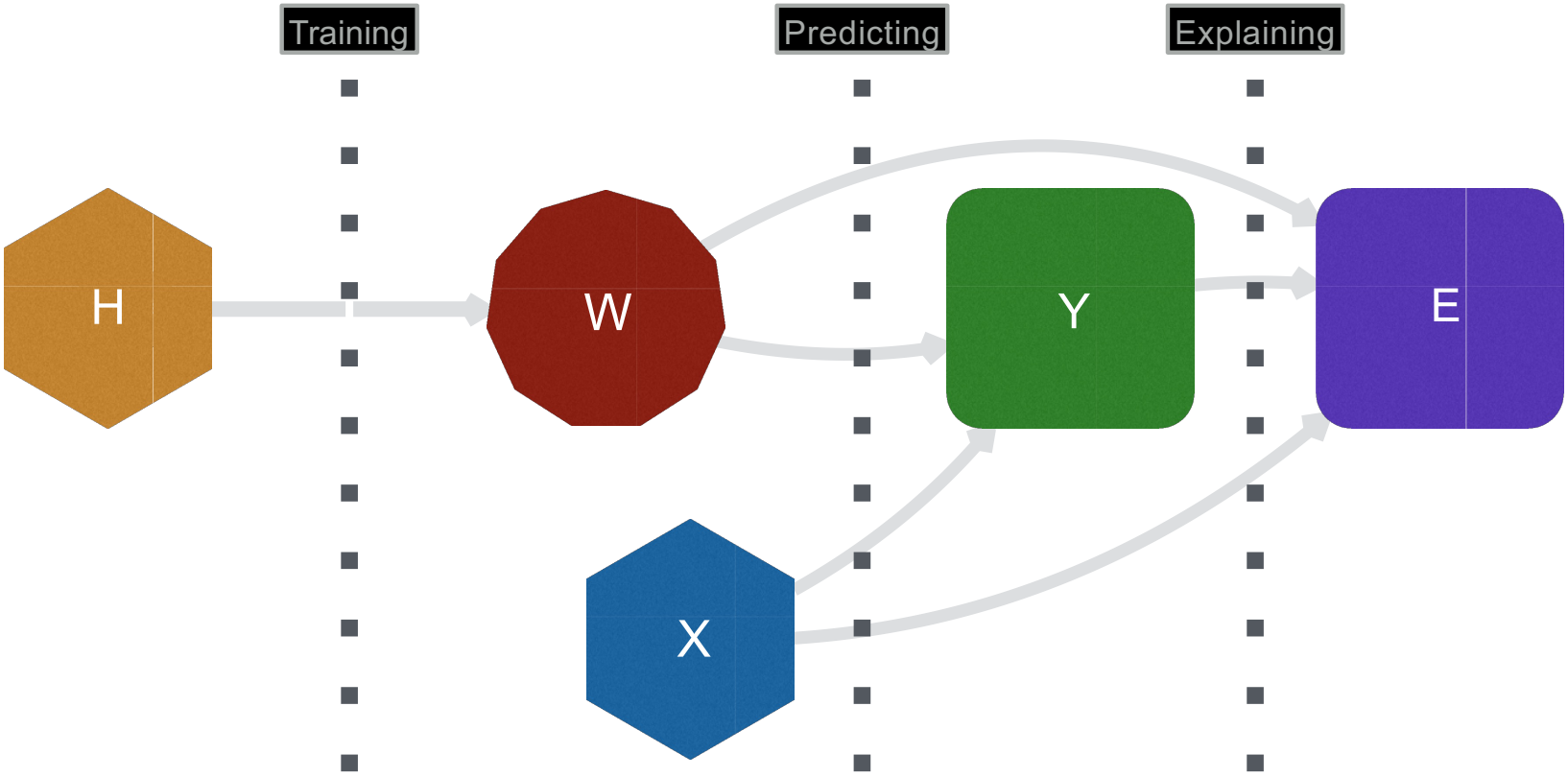
Evaluating explanations

Motivation: Explanation Generation Process

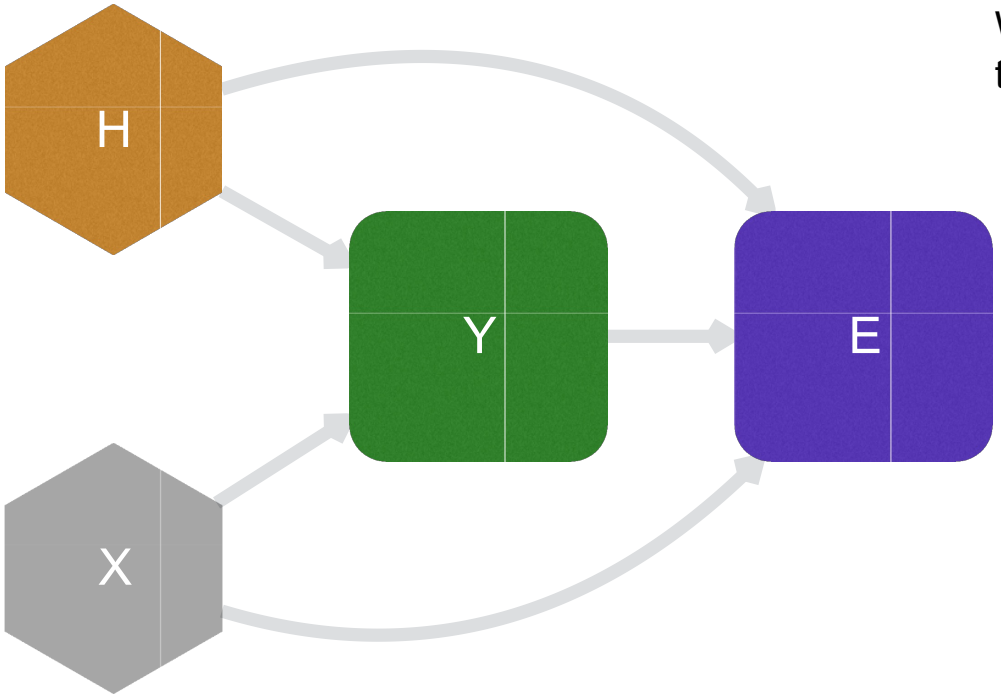


[Karimi et al., ICML 2023]

Explanation Generation Process

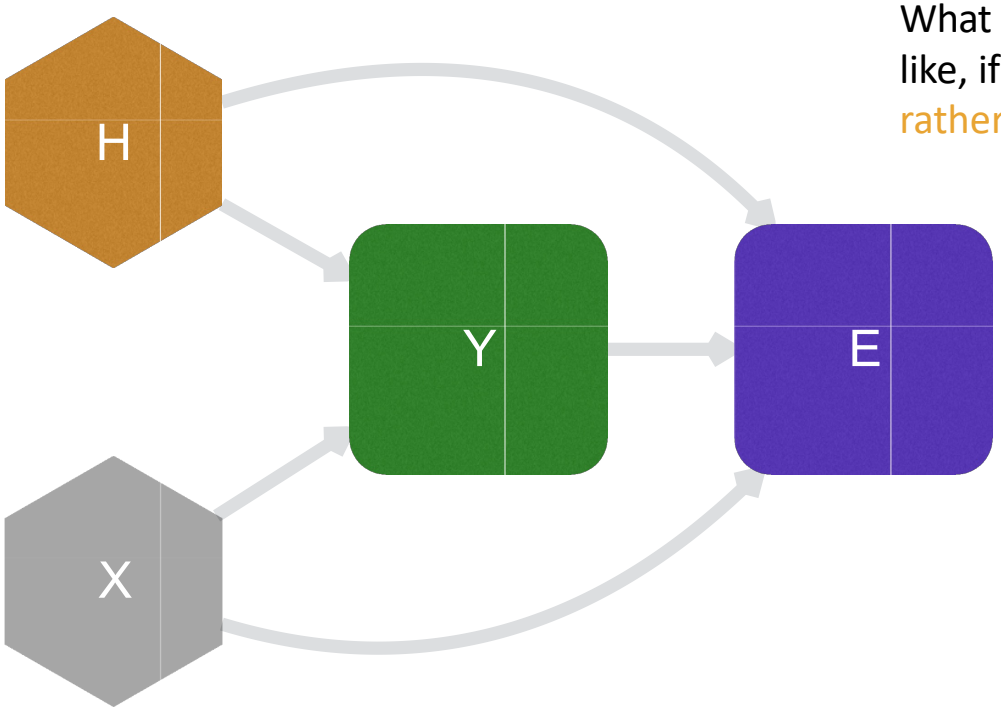


Hyperparameters as Treatments



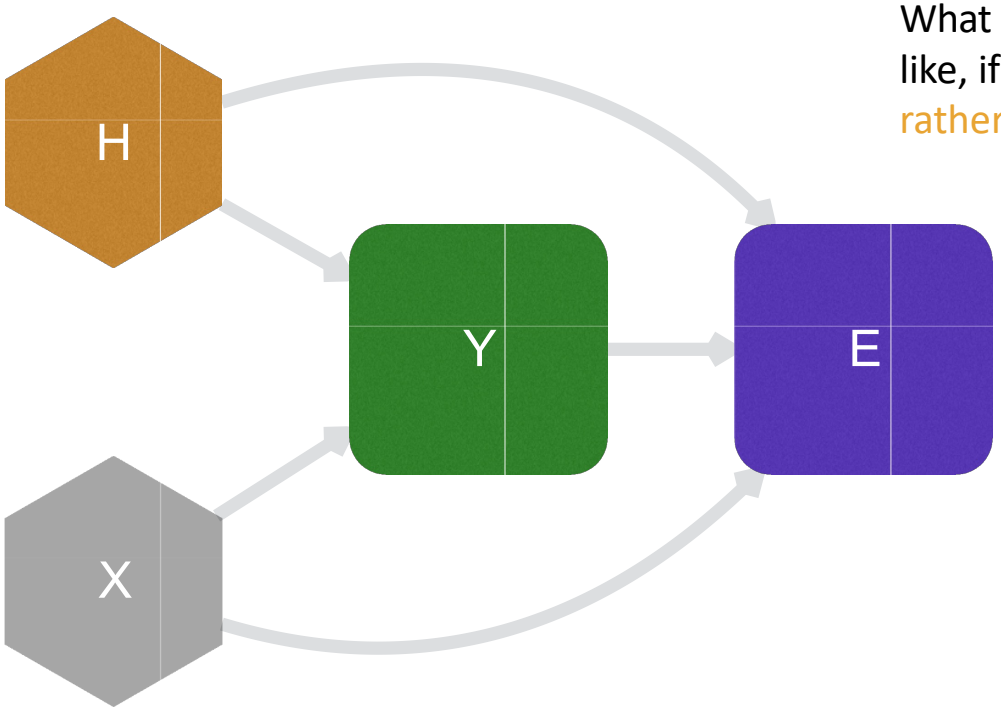
What is the effect of the hyperparameters on the resulting prediction/explanation?

Hyperparameters as Treatments



What does the prediction/explanation for X = x look like, if the hyperparameters take on value H = h rather than H = h', *all else being equal?*

Hyperparameters as Treatments



What does the prediction/explanation for X = x look like, if the hyperparameters take on value H = h rather than H = h', *all else being equal?*

$$Y_{h=1} - Y_{h=0}$$
single binary treatment

$$E_{m \neq n} [\, Y_{h=n} - Y_{h=m} \,]$$
single non-binary treatment

$$E_{h \setminus i} [\, E_{m \neq n} [\, Y_{hi=n,\, h \setminus i} - Y_{hi=m,\, h \setminus i} \,] \,]$$
multiple non-binary treatment

What remains to be done?

- We need better explainability methods.
- Studied hyperparameters independently; how to study joint effects?
- Analysis limited to the model zoo; how to analyze when zoo not available?
- Assumed linear differences in total and direct effects; what about non-linear effects?
- Extensions beyond saliency map, e.g., SHAP, LIME, CFE, etc.

References & Reading

- Verma S, Boonsanong V, Hoang M, et al. Counterfactual explanations and algorithmic recourses for machine learning: A review[J]. arXiv
- Lucic A, ter Hoeve M, Tolomei G, et al. CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks[J]. arXiv 2021.
- Ma, J, et al. “CLEAR: Generative Counterfactual Explanations on Graphs.” *NeurIPS* (2022).
- Karimi A H, Von Kügelgen J, Schölkopf B, et al. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach[J]. Advances in neural information processing systems, 2020, 33: 265-277.
- Socially Responsible Machine Learning: A Causal Perspective. KDD 2023 Tutorial.

Thank you!
Questions?