A PROOFS

A.1 THEOREMS

A.1.1 Proof of Theorem 1

Theorem. With oracle estimates $PS(c, y)$ for all $c \in \mathcal{C}$, Alg. 1 is sound and complete.

Proof. Soundness and completeness follow directly from the specification of (P1) $\mathcal{C}$ and (P2) $\preceq$ in the algorithm's input $\mathcal{B}$, along with (P3) access to oracle estimates $PS(c, y)$ for all $c \in \mathcal{C}$. Recall that the partial ordering must be complete and transitive, as noted in Sect. 3.

Assume that Alg. 1 generates a false positive, i.e. outputs some c that is not τ-minimal. Then by Def. 4, either the algorithm failed to properly evaluate $PS(c, y)$, thereby violating (P3); or failed to identify some c' such that (i) $PS(c', y) \geq \tau$ and (ii) $c' \prec c$. (i) is impossible by (P3), and (ii) is impossible by (P2). Thus there can be no false positives.

Assume that Alg. 1 generates a false negative, i.e. fails to output some c that is in fact τ-minimal. By (P1), this c cannot exist outside the finite set $\mathcal{C}$. Therefore there must be some $c \in \mathcal{C}$ for which either the algorithm failed to properly evaluate $PS(c, y)$, thereby violating (P3); or wrongly identified some c' such that (i) $PS(c', y) \geq \tau$ and (ii) $c' \prec c$. Once again, (i) is impossible by (P3), and (ii) is impossible by (P2). Thus there can be no false negatives.

A.1.2 Proof of Theorem 2

Theorem. With sample estimates $\hat{PS}(c, y)$ for all $c \in \mathcal{C}$, Alg. 1 is uniformly most powerful.

Proof. A testing procedure is uniformly most powerful (UMP) if it attains the lowest type II error β of all tests with fixed type I error α. Let Θ_0, Θ_1 denote a partition of the parameter space into null and alternative regions, respectively. The goal in frequentist inference is to test the null hypothesis $H_0 : \theta \in \Theta_0$ against the alternative $H_1 : \theta \in \Theta_1$ for some parameter θ. Let $\psi(X)$ be a testing procedure of the form $\mathbb{1}[T(X) \geq c_\alpha]$, where X is a finite sample, $T(X)$ is a test statistic, and c_α is the critical value. This latter parameter defines a rejection region such that test statistics integrate to α under H_0. We say that $\psi(X)$ is UMP iff, for any other test $\psi'(X)$ such that

$$\sup_{\theta \in \Theta_0} \mathbb{E}_\theta[\psi'(X)] \leq \alpha,$$

we have

$$(\forall \theta \in \Theta_1)\ \mathbb{E}_\theta[\psi'(X)] \leq \mathbb{E}_\theta[\psi(X)],$$

where $\mathbb{E}_{\theta \in \Theta_1}[\psi(X)]$ denotes the power of the test to detect the true θ, $1 - \beta_\psi(\theta)$. The UMP-optimality of Alg. 1 follows from the UMP-optimality of the binomial test (see [Lehmann and Romano, 2005, Ch. 3]), which is used to decide between $H_0 : PS(c, y) < \tau$ and $H_1 : PS(c, y) \geq \tau$ on the basis of observed proportions $\hat{PS}(c, y)$, estimated from n samples for all $c \in \mathcal{C}$. The proof now takes the same structure as that of Thm. 1, with (P3) replaced by (P3′): access to UMP estimates of $PS(c, y)$. False positives are no longer impossible but bounded at level α; false negatives are no longer impossible but occur with frequency β. Because no procedure can find more τ-minimal factors for any fixed α, Alg. 1 is UMP.

A.2 PROPOSITIONS

A.2.1 Proof of Proposition 1

Proposition. Let $c_S(\boldsymbol{z}) = 1$ iff $\boldsymbol{x} \subseteq \boldsymbol{z}$ was constructed by holding $\boldsymbol{x}^S$ fixed and sampling $\boldsymbol{X}^R$ according to $\mathcal{D}(\cdot|S)$. Then $v(S) = PS(c_S, y)$.

As noted in the text, $\mathcal{D}(\boldsymbol{x}|S)$ may be defined in a variety of ways (e.g., via marginal, conditional, or interventional distributions). For any given choice, let $c_S(\boldsymbol{z}) = 1$ iff $\boldsymbol{x}$ is constructed by holding $\boldsymbol{x}_i^S$ fixed and sampling $\boldsymbol{X}^R$ according to $\mathcal{D}(\boldsymbol{x}|S)$. Since we assume binary Y (or binarized, as discussed in Sect. 3), we can rewrite Eq. 2 as a probability:

$$v(S) = P_{\mathcal{D}(\boldsymbol{x}|S)}(f(\boldsymbol{x}_i) = f(\boldsymbol{x})),$$

where $\boldsymbol{x}_i$ denotes the input point. Since conditional sampling is equivalent to conditioning after sampling, this value function is equivalent to $PS(c_S, y)$ by Def. 2.

A.2.2 Proof of Proposition 2

Proposition. Let $c_A(\boldsymbol{z}) = 1$ iff $A(\boldsymbol{x}) = 1$. Then $\text{prec}(A) = PS(c_A, y)$.

The proof for this proposition is essentially identical, except in this case our conditioning event is $A(\boldsymbol{x}) = 1$. Let $c_A = 1$ iff $A(\boldsymbol{x}) = 1$. Precision prec(A), given by the lhs of Eq. 3, is defined over a conditional distribution $\mathcal{D}(\boldsymbol{x}|A)$. Since conditional sampling is equivalent to conditioning after sampling, this probability reduces to $PS(c_A, y)$.

A.2.3 Proof of Proposition 3

Proposition. Let $cost$ be a function representing $\preceq$, and let c be some factor spanning reference values. Then the counterfactual recourse objective is:

$$c^* = \underset{c \in \mathcal{C}}{\text{argmin}}\ cost(c)\ \text{ s.t. } PS(c, 1 - y) \geq \tau, \quad (1)$$

where τ denotes a decision threshold. Counterfactual outputs will then be any $\boldsymbol{z} \sim \mathcal{D}$ such that $c^*(\boldsymbol{z}) = 1$.

There are two closely related ways of expressing the counterfactual objective: as a search for optimal *points*, or optimal *actions*. We start with the latter interpretation, reframing actions as factors. We are only interested in solutions that flip the original outcome, and so we constrain the search to factors that meet an I2R sufficiency threshold, $PS(c, 1-y) \geq \tau$. Then the optimal action is attained by whatever factor (i) meets the sufficiency criterion and (ii) minimizes cost. Call this factor c^*. The optimal point is then any $\boldsymbol{z}$ such that $c^*(\boldsymbol{z}) = 1$.

A.2.4 Proof of Proposition 4

Proposition. Consider the bivariate Boolean setting, as in Sect. 2. We have two counterfactual distributions: an input space $\mathcal{I}$, in which we observe x, y but intervene to set $X = x'$; and a reference space $\mathcal{R}$, in which we observe x', y' but intervene to set $X = x$. Let $\mathcal{D}$ denote a uniform mixture over both spaces, and let auxiliary variable W tag each sample with a label indicating whether it comes from the original ($W = 1$) or contrastive ($W = 0$) counterfactual space. Define $c(\boldsymbol{z}) = w$. Then we have $\texttt{suf}(x, y) = PS(c, y)$ and $\texttt{nec}(x, y) = PS(1 - c, y')$.

Recall from Sect. 2 that Pearl [2000, Ch. 9] defines $\texttt{suf}(x, y) := P(y_x | x', y')$ and $\texttt{nec}(x, y) := P(y'_{x'} | x, y)$. We may rewrite the former as $P_{\mathcal{R}}(y)$, where the reference space $\mathcal{R}$ denotes a counterfactual distribution conditioned on $x', y', do(x)$. Similarly, we may rewrite the latter as $P_{\mathcal{I}}(y')$, where the input space $\mathcal{I}$ denotes a counterfactual distribution conditioned on $x, y, do(x')$. Our context $\mathcal{D}$ is a uniform mixture over both spaces.

The key point here is that the auxiliary variable W indicates whether samples are drawn from $\mathcal{I}$ or $\mathcal{R}$. Thus conditioning on different values of W allows us to toggle between probabilities over the two spaces. Therefore, for $c(\boldsymbol{z}) = w$, we have $\texttt{suf}(x, y) = PS(c, y)$ and $\texttt{nec}(x, y) = PS(1 - c, y')$.

B ADDITIONAL DISCUSSIONS OF METHOD

B.1 τ-MINIMALITY AND NECESSITY

As a follow up to Remark 2 in Sect. 3.2, we expand here upon the relationship between τ and cumulative probabilities of necessity, which is similar to a precision-recall curve quantifying and qualifying errors in classification tasks. In this case, as we lower τ, we allow more factors to be taken into account, thus covering more pathways towards a desired outcome in a cumulative sense. We provide an example of such a precision-recall curve in Fig. 1, using an R2I view of the `German` credit dataset. Different levels of cumulative necessity may be warranted for different tasks, depending on how important it is to survey multiple paths towards an outcome. Users can therefore adjust τ to accommodate desired levels of cumulative PN over successive calls to LENS.

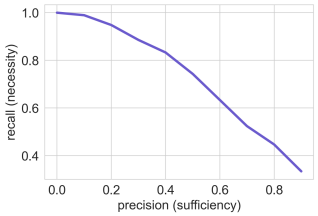

Figure 1: An example curve exemplifying the relationship between τ and cumulative probability necessity attained by selected τ-minimal factors.

C ADDITIONAL DISCUSSIONS OF EXPERIMENTAL RESULTS

C.1 DATA PRE-PROCESSING AND MODEL TRAINING

German Credit Risk. We first download the dataset from Kaggle,[1] which is a slight modification of the UCI version [Dua and Graff, 2017]. We follow the pre-processing steps from a Kaggle tutorial.[2] In particular, we map the categorical string variables in the dataset (`Savings`, `Checking`, `Sex`, `Housing`, `Purpose` and the outcome `Risk`) to numeric encodings, and mean-impute values missing values for `Savings` and `Checking`. We then train an Extra-Tree classifier [Geurts et al., 2006] using scikit-learn, with random state 0 and max depth 15. All other hyperparameters are left to their default values. The model achieves a 71% accuracy.

German Credit Risk - Causal. We assume a partial ordering over the features in the dataset, as described in Fig. 5. We use this DAG to fit a structural causal model (SCM) based on the original data. In particular, we fit linear regressions for every continuous variable and a random forest classifier for every categorical variable. When sampling from $\mathcal{D}$, we let variables remain at their original values unless either (a) they are directly intervened on, or (b) one of their ancestors was intervened on. In the latter case, changes are propagated via the structural equations. We add stochasticity via Gaussian noise for continuous outcomes, with variance given by each model's residual mean squared error. For categorical variables, we perform multinomial sampling over predicted class probabilities. We use the same f model as for the non-causal `German` credit risk description above.

[1] See `https://www.kaggle.com/kabure/german-credit-data-with-risk?select=german_credit_data.csv`.

[2] See `https://www.kaggle.com/vigneshj6/german-credit-data-analysis-python`.

SpamAssassins. The original spam assassins dataset comes in the form of raw, multi-sentence emails captured on the Apache SpamAssassins project, 2003-2015.[3] We segmented the emails to the following "features": `From` is the sender; `To` is the recipient; `Subject` is the email's subject line; `Urls` records any URLs found in the body; `Emails` denotes any email addresses found in the body; `First Sentence`, `Second Sentence`, `Penult Sentence`, and `Last Sentence` refer to the first, second, penultimate, and final sentences of the email, respectively. We use the original outcome label from the dataset (indicated by which folder the different emails were saved to). Once we obtain a dataset in the form above, we continue to pre-process by lower-casing all characters, only keeping words or digits, clearing most punctuation (except for '-' and '_'), and removing stopwords based on nltk's provided list [Bird et al., 2009]. Finally, we convert all clean strings to their mean 50-dim GloVe vector representation [Pennington et al., 2014]. We train a standard MLP classifier using scikit-learn, with random state 1, max iteration 300, and all other hyperparameters set to their default values.[4] This model attains an accuracy of 98.3%.

IMDB. We follow the pre-processing and modeling steps taken in a standard tutorial on LSTM training for sentiment prediction with the IMDB dataset.[5] The CSV is included in the repository named above, and can be additionally downloaded from Kaggle or ai.standford.[6] In particular, these include removal of HTML-tags, non-alphabetical characters, and stopwords based on the the list provided in the ntlk package, as well as changing all alphabetical characters to lower-case. We then train a standard LSTM model, with 32 as the embedding dimension and 64 as the dimensionality of the output space of the LSTM layer, and an additional dense layer with output size 1. We use the sigmoid activation function, binary cross-entropy loss, and optimize with Adam [Kingma and Ba, 2015]. All other hyperparameters are set to their default values as specified by Keras.[7] The model achieves an accuracy of 87.03%.

Adult Income. We obtain the adult income dataset via DiCE's implementation[8] and followed Haojun Zhu's pre-processing steps.[9] For our recourse comparison, we use a pretrained MLP model provided by the authors of DiCE, which is a single layer, non-linear model trained with TensorFlow and stored in their repository as 'adult.h5'.

C.2 TASKS

Comparison with attributions. For completeness, we also include here comparison of cumulative attribution scores per cardinality with probabilities of sufficiency for the I2R view (see Fig. 2).

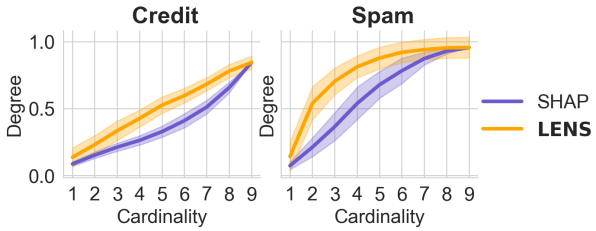

Figure 2: Comparison of degrees of sufficiency in I2R setting, for top k features based on SHAP scores, against the best performing subset of cardinality k identified by our method. Results for `German` are averaged over 50 inputs; results for `SpamAssassins` are averaged over 25 inputs.

Sentiment sensitivity analysis. We identify sentences in the original `IMDB` dataset that are up to 10 words long. Out of those, for the first example we only look at wrongly predicted sentences to identify a suitable example. For the other example, we simply consider a random example from the 10-word maximum length examples. We noted that Anchors uses stochastic word-level perturbations for this setting. This leads them to identify explanations of higher cardinality for some sentences, which include elements that are not strictly necessary. In other words, their outputs are not minimal, as required for descriptions of "actual causes" [Halpern and Pearl, 2005, Halpern, 2016].

Comparison with Anchors. To complete the picture of our comparison with Anchors on the `German` Credit Risk dataset, we provide here additional results. In the main text, we included a comparison of Anchors's single output precision against the mean degree of sufficiency attained by our multiple suggestions per input. We sample 100 different inputs from the `German` Credit dataset and repeat this same comparison. Here we additionally consider the minimum and maximum $PS(c, y)$ attained by LENS against Anchors. Note that even when considering minimum PS suggestions by LENS, i.e. our worst output, the method shows more consistent performance. We qualify this discussion by noting that Anchors may generate results comparable to our own by setting the δ hyperparameter to a lower value. However, Ribeiro et al. [2018] do not discuss this parameter in

[3] See `https://spamassassin.apache.org/old/credits.html`.

[4] See `https://scikit-learn.org/stable/modules/generated/sklearn.\neural_network.MLPClassifier.html`.

[5] See `https://github.com/hansmichaels/sentiment-analysis-IMDB-Review-using-LSTM/blob/master/sentiment_analysis.py.ipynb`.

[6] See `https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews` or `http://ai.stanford.edu/~amaas/data/sentiment/`.

[7] See `https://keras.io`.

[8] See `https://github.com/interpretml/DiCE`.

[9] See `https://rpubs.com/H_Zhu/235617`.

Table 1: Recourse options for a single input given by DiCE and our method. We report targets of interventions as suggested options, but they could correspond to different values of interventions. Our method tends to propose more minimal and diverse intervention targets. Note that all of DiCE's outputs are already subsets of LENS's two top suggestions, and due to τ-minimality LENS is forced to pick the next factors to be non-supersets of the two top rows. This explains the higher cost of LENS's bottom three rows.

input								DiCE output		**LENS output**	
Age	Wrkcls	Edu.	Marital	Occp.	Race	Sex	Hrs/week	Targets of intervention	Cost	Targets of intervention	Cost
42	Govt.	HS-grad	Single	Service	White	Male	40	Age, Edu., Marital, Hrs/week	8.13	Edu.	1
								Age, Edu., Marital, Occp., Sex, Hrs/week	5.866	Martial	1
								Age, Wrkcls, Educ., Marital, Hrs/week	5.36	Occp., Hrs/week	19.3
								Age, Edu., Occp., Hrs/week	3.2	Wrkcls, Occp., Hrs/week	12.6
								Edu., Hrs/week	11.6	Age, Wrkcls, Occp., Hrs/week	12.2

detail in either their original article or subsequent notebook guides. They use default settings in their own experiments, and we expect most practitioners will do the same.

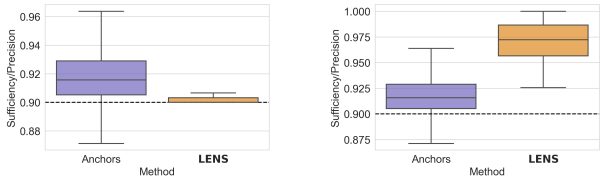

Figure 3: We compare degree of sufficiency against precision scores attained by the output of LENS and Anchors for examples from `German`. We repeat the experiment for 100 sampled inputs, and each time consider the single output by Anchors against the min (left) and max (right) $PS(c, y)$ among LENS's multiple candidates. Dotted line indicates $\tau = 0.9$, the threshold we chose for this experiment.

Recourse: DiCE comparison First, we provide a single illustrative example of the lack of diversity in intervention targets we identify in DiCE's output. Let us consider one example, shown in Table 1. While DiCE outputs are diverse in terms of values and target combinations, they tend to have great overlap in intervention targets. For instance, `Age` and `Education` appear in almost all of them. Our method would focus on minimal paths to recourse that would involve different combinations of features.

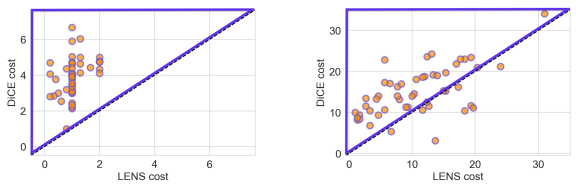

Figure 4: We show results over 50 input points sampled from the original dataset, and all possible references of the opposite class, across two metrics: the min cost (left) of counterfactuals suggested by our method vs. DiCE, and the max cost (right) of counterfactuals.

Next, we also provide additional results from our cost comparison with DiCE's output in Fig. 3. While in the main text we include a comparison of our mean cost output against DiCE's, here we additionally include a comparison of min and max cost of the methods' respective outputs. We see that even when considering minimum and maximum cost, our method tends to suggest lower cost recourse options. In particular, note that all of DiCE's outputs are already subsets of LENS's two top suggestions. The higher costs incurred by LENS for the next two lines are a reflection of this fact: due to τ-minimality, LENS is forced to find other interventions that are no longer supersets of options already listed above.

References

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly, 2009.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, 2006.

Joseph Y Halpern. *Actual Causality*. The MIT Press, Cambridge, MA, 2016.

Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. Part I: Causes. *Br. J. Philos. Sci.*, 56(4):843–887, 2005.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The 3rd International Conference for Learning Representations*, 2015.

E.L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, New York, Third edition, 2005.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, 2000.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In

Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, pages 1527–1535, 2018.