Causal Counterfactual Forecasting

Sargun Nagpal
Center for Data Science, NYU
sargun.nagpal@nyu.edu

Abstract

To make informed decisions, decision-makers must forecast the potential outcomes that are likely to occur when they take a certain course of action. This involves predicting counterfactual outcomes under different treatment alternatives over time. For example, an e-commerce business may be interested in forecasting the sales of its products over time under different pricing interventions and performing what-if analyses to plan its business strategy. Training supervised machine learning models on historical observational data is incorrect since counterfactual data is absent, and empirical risk minimization does not account for interventions that were not undertaken. In this work, I adopted the Counterfactual Recurrent Network (CRN), originally developed to predict counterfactuals on a simulated tumor growth dataset with a single temporal and static covariate, and modified it to work on any time-series dataset. I evaluated the model on the American Causal Inference Conference (ACIC 2023) challenge dataset, studied the causal assumptions of the model, and discussed their plausibility in a real-world business setting.

Motivation

An extensive body of research has been conducted on causal inference in longitudinal studies, where confounders and treatments vary over time. Early methods, such as Marginal Structural Models (MSMs)[1] used Inverse Probability of Treatment Weighting (IPTW) to remove the effect of time-dependent confounding. In recent years, deep neural networks have been widely adopted to estimate counterfactual outcomes. The Recurrent Marginal Structural Network (R-MSN)[2] uses a Recurrent Neural Network to estimate the IPTW weights and a seq2seq architecture to forecast counterfactuals under a given sequence of treatments. The Counterfactual Recurrent Network (CRN)[3] uses the concept of domain adversarial training to learn a treatment-invariant representation of a unit's history to remove the bias from time-dependent confounding. This latent representation is used in an LSTM based seq2seq architecture like the R-MSN to forecast outcomes under different treatment plans. The Causal Transformer[4] claims to improve existing methods by adopting the Transformer network to learn these representations, which are better at capturing long-range temporal dependencies. Notably, most of the recent methods have been evaluated on a tumor growth dataset[5] or on data from healthcare settings.

In this study, I report the findings of training the Counterfactual Recurrent Network on data simulated by the organizers of the American Causal Inference Conference (ACIC 2023) data challenge[6]. It is suggested that the data reflects what a decision-maker might use in deciding item prices. Therefore, I focus on an example of an e-commerce business that wants to implement a

dynamic pricing strategy to maximize its sales. The main contributions of this work are as follows:

1. **Evaluation of a counterfactual forecasting model (CRN) outside the healthcare setting.** As mentioned above, most recent methods, including the CRN, have been evaluated on a simulated tumor growth dataset[5] or data in the healthcare domain. In this work, I evaluated the model on the ACIC challenge dataset, which mirrors data in an e-commerce business setting.
2. **General purpose implementation of CRN to work with any time-series dataset**. The open-source implementation[7] of CRN only works with the simulated tumor dataset. Upon close assessment, I found that it assumes the same data for both the encoder and the decoder (one static variable, and one temporal variable- same as the outcome). However, real-world datasets usually contain multiple temporal and static variables, and the decoder only requires the static covariates, since temporal covariates are not available at inference time. Therefore, I prepared a data-processing notebook and modified the CRN code so that it could be used with any time-series dataset. Finally, I prepared a script to infer future outcomes for the ACIC challenge. The full log of modifications made in the original code can be found here.
3. **Discussion of the plausibility of causal assumptions**. While all the studies listed above state the necessary assumptions for causal claims, the plausibility of these assumptions in real-world settings are skimmed over or omitted entirely. In this work, I discuss these assumptions in the context of the e-commerce example and argue that they are difficult to satisfy in practice.

To motivate interest in the problem, *Figure 1* shows some applications of counterfactual forecasting in the context of our example.

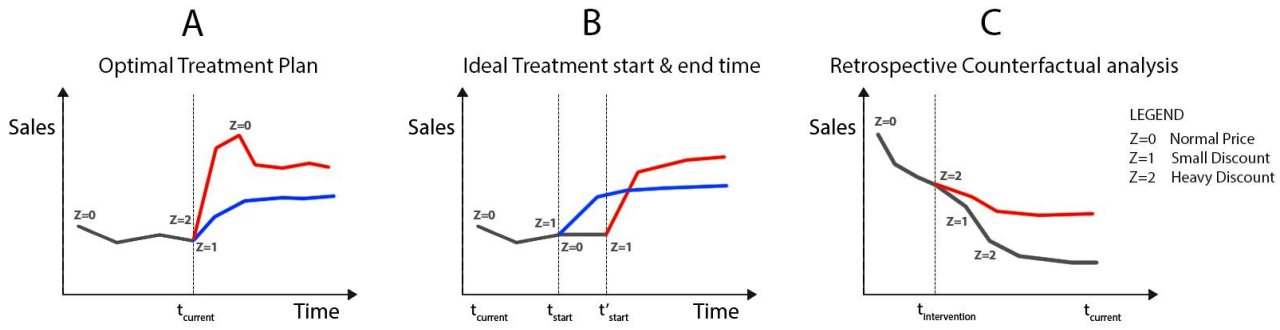

Figure 1: Applications of counterfactual estimation in an e-commerce setting. *The plots show the effect of pricing (treatment) on sales (outcome) of a product over time. (**A**) Predicting the optimal treatment plan. In plan 1 (red), a heavy discount is offered (Z=2) to create a demand for the product, and then the price is reduced back to normal (Z=0), while in plan 2 (blue), a small discount is offered (Z=1) for a prolonged period. Plan 1 is predicted to lead to higher*

*sales. (**B**) Optimal time of treatment. Starting a discount campaign (Z=1) at time t_{start} (blue) instead of time t'_{start} (red) is forecasted to lead to lower sales. (**C**) Retrospective Counterfactual Analysis. The sales of a product diminished heavily (gray). It is predicted that if a discount was offered at time $t_{intervention}$, the demand for the product could have been retained (red).*

ACIC data and competition set up

For the ACIC 2023 competition, we are given a simulated observational time-series dataset of outcomes under different treatment levels. The description of the columns is not given. However, drawing analogy to our e-commerce problem, the data consist of the sales (Outcome) of about 4000 products for 95 weeks each, under one or more of 6 different pricing strategies (Treatment). The sample size is about 371k, with six covariates - four static and two temporal. The goal is to predict the sales of each item for the next 5 weeks (weeks 96-100) under each of the six different pricing strategies. Therefore, we make 30 predictions per unit. *Table 1* describes the columns in the dataset with possible interpretations in the e-commerce world. A more detailed exploratory data analysis can be found here.

S no.	Attribute	Data type	Statistics	Variable type	Interpretation
1	unitID	Categorical	3908 units	-	Product IDs
2	weekID	Numeric	Range: 1-95	-	Time steps
3	Outcome	Numeric	Bimodal, right skewed. Min=0, Med=243, Max=3854	-	Sales
4	Treatment	Categorical	6 levels. Z=0, 5 make up 70% of the data	-	Pricing strategy
5	X1	Numeric	Right skewed. Min=21, Med=36, Max=400	Static	Average Historic Sales
6	X2	Numeric	Range of over 150k	Temporal	Inventory count
7	X3	Binary	Remains 0 about 90% time, Continuous time spans of 1	Temporal	High demand indicator
8	C1	Categorical	15 levels Equally distributed categories	Static	Product category
9	C2	Categorical	2495 levels. 60% levels only appear for a single product	Static	Fine-grained category
10	C3	Categorical	6 levels, Almost equally distributed	Static	Store ID

Table 1: Data characteristics. *There is one outcome variable, one treatment variable with six levels, and six covariates. Some covariates are unit-level static features, while others are temporal features. The interpretation of the features in an ecommerce setting is presented in the last column.*

For cross-validation, I partitioned the data into a 80:10:10 train-validation-test split (297k, 37k, 37k records, respectively). I log-transformed the outcome and continuous variables to reduce the skew and scale them, and one-hot encoded the categorical variables. I dropped variable C2 to avoid the curse of dimensionality since it is a categorical variable with 2495 levels. Note that the implications of this on ignorability are discussed in the Assumptions section. Finally, I processed the data for the CRN encoder and decoder. The encoder takes the observed data: the static covariates, current temporal covariates, and the previous treatment as input at each time step and performs a 1-step ahead prediction. The decoder takes the current intended treatment, previously predicted outcome, and static features as input and performs both 1-step ahead and 5-step ahead predictions.

Estimand

We aim to estimate the counterfactual outcomes for each unit at future time steps. Thus we are interested in the expected value of the potential outcome at time $t+\tau$ under a sequence of treatments from time t to $t+\tau-1$, given the observed data till the current time step t. Mathematically, the estimand can be written as follows.

$$\text{Estimand} = E[\, Y_{t+\tau}[Z(t,\, t+\tau-1)] \mid H_t \,]$$

where, $Z(t,\, t+\tau-1) = [z_t,\, z_{t+1},\, \ldots, z_{t+\tau-1}]$ is the intended sequence of treatments from time t to $t+\tau-1$, and

$H_t = (\overline{X}_t, \overline{Z}_{t-1}, V)$ represents the observed history of the covariates $\overline{X}_t = [X_1, \ldots, X_t]$, the treatment assignments $\overline{Z}_t = [Z_1, \ldots, Z_t]$, and static features V.

The chosen estimand minimizes the squared loss of the prediction. This aligns with the objective of the ACIC competition to minimize the Root Mean Squared Error (RMSE) of prediction over all time steps and counterfactual states for each unit.

$$\text{RMSE} = \sqrt{\sum_{i,t,z} \left(y_{i,t}(z) - \hat{y}_{i,t}(z)\right)^2 / (N \times N_t \times N_z)}$$

where i = Unit number, t = Time step, z = Treatment level, y = True outcome, $\hat{y}$ = Predicted outcome, N = Total units (3908), Nt = Number of time steps (5), Nz = Number of treatments (6).

RMSE is commonly used to evaluate continuous predictions and measures how far the predictions are from the true values on average. However, a disadvantage of using RMSE is that it heavily penalizes outliers.

Methods

Background on why supervised learning models fail and time-dependent confounding

Cross-sectional observational studies suffer from the problem of selection bias, where the treatment group can differ from the control group in terms of covariates that are also predictive of the outcome. In such cases, any difference in observed outcomes cannot be attributed to the treatment alone. Accordingly, these covariates that predict both the treatment and the outcome are called confounders. Furthermore, the two groups could have a lack of balance and overlap. Therefore standard machine learning models trained on these data make strong assumptions in parts of input space where no data is observed, and may not be generalizable[8].

In a temporal study, we have the additional problem of time-dependent confounding. The treatment assignment at time step t affects the values of covariates in subsequent time steps, which in turn affect the choice of future treatments. Therefore, the covariates at time t+1 are post-treatment variables since their value is determined by prior treatments. These confounders are called time-varying confounders. For instance, in our e-commerce example, offering a heavy discount (Z=2) on a product at time step t with low demand (X3=0) may significantly increase its demand (X3=1) at time t+1, which would in turn affect future pricing. Furthermore, the effect of the confounder on the outcome could change with time (time-modified confounding)[9]. In such settings, supervised learning methods can be unreliable because the training data is affected by the treatment policy, and since these models only fit observed data, they do not generalize to predict the outcome when the policy changes[10]. It is worth noting, however, that some studies show that machine learning models achieve similar performance to causal counterfactual models when the degree of time-dependent confounding is low[3].

Domain Adversarial Training

The CRN model is based on the idea of domain adversarial training of neural networks[11]. If the train and test distributions are different in a machine learning problem (domain shift), then models trained on the train set do not generalize well on the test set. A naive way to deal with this problem is to only train on features that have similar distributions in the two datasets. Ganin et al. extended this idea to learn input representations that cannot discriminate between whether input examples belong to the train or test set, but at the same time, are predictive of the output. Therefore, the learned input representations for the train and test set are similar, despite having different covariate distributions. The adversarial loss function aims to minimize the outcome prediction loss and maximize the domain prediction loss. This helps achieve domain adaptation.

Counterfactual Recurrent Network (CRN)

The CRN model uses this idea to learn a latent representation of a unit's current state, that is predictive of the next step outcome, but not the next treatment. Therefore, even though the value of the confounders is influenced by prior treatments, the representation at the current time step cannot discriminate between what treatment will be assigned next. Therefore, this

'treatment-invariant' representation removes the effect of time-dependent confounding. *Figure 2* shows the architecture of the CRN network, which consists of an encoder that learns the treatment invariant balancing representation $\phi(H_t)$, and a decoder that is used to make autoregressive predictions for a given sequence of treatments.

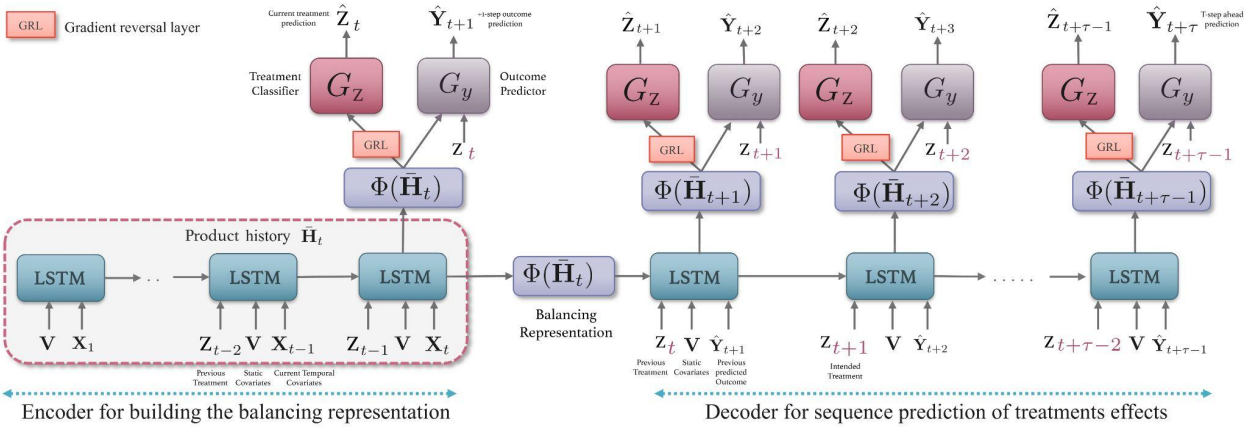

Figure 2: Architecture of the Counterfactual Recurrent Network. *Figure sourced from Bica et al [3]. The left half shows the encoder network, which learns the balancing representation from the unit history (past confounders and treatment values). The right half shows the decoder network, which uses the balancing representation, and an intended sequence of treatments to produce output predictions.*

The encoder is an LSTM-based recurrent network, which at each time step *t* takes the temporal and static confounders (X_t and V) and the prior treatment Z_{t-1} as input. A treatment classifier (cross-entropy loss) and an outcome predictor network (L2 loss) are fitted to the output state. These are responsible for the domain adversarial learning, which helps to learn treatment invariant representations. Mathematically, $P(\phi(H_t \mid Z = 1)) = \ldots = P(\phi(H_t \mid Z = k))$ for k treatment levels.

The decoder is also a LSTM-based recurrent network that is initialized by the output of the encoder. It takes the static variables (V), outcomes produced at prior stages $\hat{Y}_{t+1}$, and the intended treatment assignment z_t as inputs and produces intermediate balancing representations as well as next-step output predictions $Y_{t+\tau}$.

Implementation details

The encoder and decoder are trained in sequence. The decoder is initialized with the balancing representations from the trained encoder. For both the components, I used the default hyperparameters for the number of hidden units of the LSTM (24), the treatment classifier (36), balancing representation dimension (12), Dropout (0.1), and trained the models for 150 and 100 epochs respectively till convergence on an NVIDIA GeForce GTX 970M GPU. I performed hyperparameter tuning on the learning rate. The results can be found in the results section.

Causal Assumptions

Sequential Ignorability

It is necessary to make several assumptions to claim that the model can make causal counterfactual predictions. First, we need to assume that conditional ignorability is satisfied, which means that there are no unmeasured confounders. The ACIC organizers declared that this is true; however, in practice, it is hard to satisfy this for an observational study. For example, product popularity could be a missing confounder in our example. A highly marketed product could have high sales and may not need aggressive pricing treatments, compared to one that is not. Even if we measure a large number of possible confounders, we should do a sensitivity analysis to understand if there is potential unmeasured confounding. Moreover, for the temporal setting, we need to make a stronger assumption called the sequential ignorability. Under this assumption, we assume that at any time, the potential outcomes are independent of the treatment given all prior observed data - the past covariates and treatment histories.

$$Y_{t+1}(z_t) \perp Z_t \mid \overline{Z}_{t-1}, \overline{X}_t \quad \forall z_t, \forall t$$

Note that I dropped a categorical variable from the analysis since it had very high cardinality (2495 levels, with ~60% levels appearing for only 1 unit). This was done to avoid making the data high dimensional, which could cause potential issues with model training. However, this leads to a violation of the above assumption.

Overlap

Next, we need to assume that we have non-zero overlap. While diagnostic plots can help assess overlap issues for single covariates, it is hard to check lack of overlap in higher dimensions. The ACIC instructions tell us that there is a lack of complete overlap, which means that the probability of some treatments given a history of prior treatments and covariates is zero. In our example, it is plausible that some treatments are never assigned for certain products. For example, it may be against the company's policy to offer heavy discounts on expensive items such as jewelry or watches. Mathematically, this assumption can be expressed as: $0 < P(Z_t = z_t \mid \overline{Z}_{t-1}, \overline{X}_t) < 1, \quad \forall z_t$

SUTVA

Finally, we need to assume SUTVA (Stable Unit Treatment Value Assumption), which implies that there is no spillover across units. For our example, this means that the pricing strategy of one product should not affect the sales of other products. But this can easily be violated since offering a discount on one product could lead to a decline in sales of other related products. The ACIC instructions claim to satisfy this assumption, however.

In summary, it can be observed that we need to make very strong causal assumptions for

counterfactual forecasting, which are unlikely to be satisfied in an observational study. Hence, it is important to state the plausibility of these assumptions when we use these models. It is worth noting, however, that these techniques can remove the bias from time-dependent confounding and therefore, may be less biased than standard supervised models.

Results & Diagnostics

The encoder was trained for 150 epochs and achieved an RMSE of 1.28 on the validation set. *Figure 3A,* shows the learning curves for the total, outcome, and treatment loss. It can be observed that both the total loss and the treatment loss decreased with time and converged in about 100 epochs. We expect the treatment loss to increase, but it remains roughly constant. *Figure 3B* shows the results from hyperparameter tuning of the learning rate. The best value was found to be 0.005. The decoder training (*Figures 3C, 3D*) exhibited a similar behavior. The decoder achieved a RMSE of 1.32 on the validation set for 1-step ahead prediction, and 2.24 for 5-step ahead prediction.

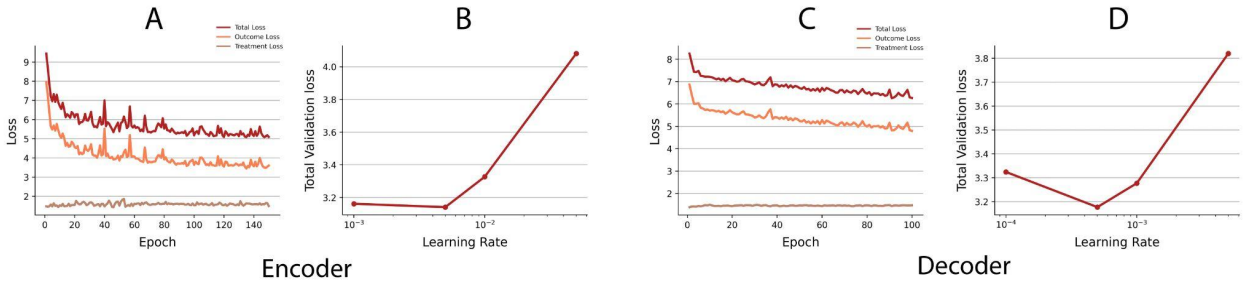

Figure 3: Model Training diagnostics. (***A)*** *Learning curve of Total, Outcome and Treatment loss with number of epochs for the Encoder.* ***(B)*** *Hyperparameter tuning of learning rate for the Encoder. Total validation loss wrt learning rate.* ***(C)*** *Learning curves for the Decoder.* ***(D)*** *Hyperparameter tuning of learning rate for the Decoder.*

Table 1 compares the results of the model with a baseline persistence forecast model, which predicts the last observed value of the outcome for all future time steps.

Model	RMSE on log-transformed output	RMSE on raw output
CRN	3.08	116.24
Persistence (Baseline)	**0.73**	**2.10**

Table 1: Model results on the test set. *The outcome variable was log-transformed before fitting the model. The RMSE is reported on both the log-transformed and unscaled output. The baseline persistence model performs better than the CRN model.*

The CRN model performs worse than the baseline model. The bias is further amplified when the log-transformed predictions are exponentiated back to the original scale. Note that scaling the output is important to avoid the outcome and treatment losses being at very different scales. In

this case, the adversarial loss would simply prefer to minimize the outcome loss, since that would lead to a large dip in the total loss. Trying different scaling techniques on the outcome variable can be explored as a next step. The density plot of the residuals of the model shows a long left tail. This means that the model systematically overestimates the true values. Further investigation is necessary in this regard as well.

Discussion

In this study, I tested the CRN model for counterfactual prediction on the ACIC 2023 dataset, programmed it to make it more general-purpose, and discussed the assumptions of the model. The problem of counterfactual forecasting has interesting applications in various domains. For example, a quant trader could benefit from forecasting potential outcomes under different trading strategies. An oncologist could benefit from knowing if they should use chemotherapy or radiotherapy on a patient, how the patient's tumor will likely shrink with time in response to different treatments, and deciding when to stop the treatment.

Analyzing causal assumptions has been an important focus of our class, and thinking through the plausibility of these assumptions (like we did with various in-class examples), led me to the revelation that although the technicalities of these models are fascinating, the assumptions under which a causal prediction can be made with these models are very strong, and are likely to be violated in an observational study.

Limitations

This study has several limitations. The final CRN model underperformed the baseline model and requires further investigation. Nevertheless, I posted a submission to the ACIC challenge in the spirit of participation. The reliance on strong causal assumptions discussed above is another major limitation. A sensitivity analysis must be carried out to check the plausibility of the assumptions. Finally, LSTM models are slow to train as they are not parallelizable, and do not capture long-range temporal context as well as other state-of-the-art methods for time series data.

Challenges

I faced numerous challenges in completing this work. A major hurdle was understanding and modifying the CRN source code. I also struggled with understanding the related methods and the sequential ignorability assumption. However, the process of learning about the topic and assessing the veracity of the causal assumptions was immensely satisfying.

Future work

As part of future work, I would like to investigate ways to improve the model predictions by experimenting with different transformations on the outcome variable, doing an error analysis, and hyperparameter tuning of additional hyperparameters. I would like to benchmark the model against standard supervised learning approaches such as ARIMA, and LSTMs, as well as other causal models such as R-MSNs, and Causal Transformers. Furthermore, I would like to do a

qualitative analysis of the balancing representations by performing dimensionality reduction and plotting them to inspect if the representations are indeed treatment-invariant. Finally, the CRN paper claims that it is possible to obtain uncertainty estimates of the outcome predictions, which is a useful next extension to explore.

References

1. Robins, J. M., Hernán, M. Á. & Brumback, B. Marginal Structural Models and Causal Inference in Epidemiology. *Epidemiology* **11**, 550 (2000).
2. Lim, B. Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks. *Adv. Neural Inf. Process. Syst.* **31**, (2018).
3. Bica, I., Alaa, A. M., Jordon, J. & van der Schaar, M. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. in *International Conference on Learning Representations* (2019).
4. Melnychuk, V., Frauen, D. & Feuerriegel, S. Causal Transformer for Estimating Counterfactual Outcomes. (2022).
5. Geng, C., Paganetti, H. & Grassberger, C. Prediction of Treatment Response for Combined Chemo- and Radiation Therapy for Non-Small Cell Lung Cancer Patients Using a Bio-Mathematical Model. *Sci. Rep.* **7**, 1–12 (2017).
6. Data Competition – SOCIETY FOR CAUSAL INFERENCE. https://sci-info.org/data-competition/ (2023).
7. GitHub - ioanabica/Counterfactual-Recurrent-Network: Code for ICLR 2020 paper: ‘Estimating counterfactual treatment outcomes over time through adversarially balanced representations’ by I. Bica, A. M. Alaa, J. Jordon, M. van der Schaar. *GitHub* https://github.com/ioanabica/Counterfactual-Recurrent-Network.
8. Alaa, A. & Schaar, M. Limits of Estimating Heterogeneous Treatment Effects: Guidelines for Practical Algorithm Design. in *International Conference on Machine Learning* 129–138 (PMLR, 2018).
9. Platt, R. W., Schisterman, E. F. & Cole, S. R. Time-modified Confounding. *Am. J. Epidemiol.* **170**, 687–694 (2009).
10. Schulam, P. & Saria, S. Reliable Decision Support using Counterfactual Models. *Adv. Neural Inf. Process. Syst.* **30**, (2017).
11. Ganin, Y. *et al.* Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **17**, 1–35 (2016).