A Causal AI Suite for Decision-Making

Abstract

Critical data science and decision-making questions across a wide variety of domains are fundamentally causal questions. We present a suite of open-source causal tools and libraries that aims to simultaneously provide core causal AI functionality to practitioners and create a platform for research advances to be rapidly deployed. In this paper, we describe our contributions towards such a comprehensive causal AI suite of tools and libraries, its design, and lessons we are learning from its growing adoption. We hope that our work accelerates use-inspired basic research for improvement of causal AI.

1 Introduction

Critical data science and decision-making questions across a wide variety of domains are fundamentally causal questions. Whether estimating the impact of a marketing campaign, understanding the reasons for customer churn, predicting the impact of climate change, or identifying which drug may work best for which patient, answering key questions relies on successfully modeling cause-and-effect relationships that underlie a system. As the field of data science and applications of data-driven decision making expand, many people are recognizing the importance of causal methods for providing robust insights. However, valid causal analysis requires a combination of human domain expertise (to make reasonable assumptions) and sophisticated analytical frameworks (to build those assumptions...
The causal AI research area is still early in its development, however, and as with any technology area, will require many more advances and iterative practical deployments to reach its full impact. As argued by Stokes [38], fundamental research is accelerated when it is motivated and informed by considerations of use. To enable and accelerate such “use-inspired basic research” in causality, we should create mechanisms that enable and ease the practical usage of state of the art technology, and collect results and challenges to inform the direction of research. This requires that we broaden the accessibility of causal methods beyond today’s causal experts, and make them available to the larger audiences of scientists, decision-makers, and other practitioners, so that we may discover the fundamental challenges of causality that stymie its full adoption. Learning from broad usage of these methods, and in particular the use cases in which current methods are insufficient or fail, will motivate and speed new research directions that our field might not otherwise prioritize or even be aware of.

We present a suite of causal tools and libraries that aims to simultaneously provide core causal AI functionality to practitioners and create a platform for research advances to be rapidly deployed. First, this suite aims to be easy for practitioners, including non-causal experts, to use. Importantly, the tools should provide scaffolding to those familiar with non-causal machine learning and data science processes as they learn causal workflows. Ideally, tools should make it easy for people to make the right assumptions and come to the correct causal conclusions, and make it difficult for people to make mistakes. Second, to enable new research advances to be rapidly deployed, the suite provides an interface that clearly separates the problem (e.g., a given causal task we wish to perform or causal question we wish to answer) from the algorithmic and methodological implementations that may solve the problem. Finally, to encourage cross-pollination of ideas across disparate causal communities of interest, the suite of tools and libraries bridges across multiple causal frameworks and tasks, including graphical approaches to causal reasoning, potential outcomes approaches to effect inference, causal discovery algorithms, and others. Parts of our causal AI suite are already widely used across industry and academia, with over 1.5 million downloads, and newer components are receiving significant interest as well.

In this paper, we describe our contributions towards such a comprehensive causal AI suite tools and libraries, our design decisions, lessons learned from their usage, and what is next.

2 A Causal AI Suite

The primary goals of our causal AI suite are (1) to broaden usage of causal methods by data scientists and decision-makers in practice; and (2) act as a flywheel to accelerate fundamental causal research and impact. The initial focus of our suite is the end-to-end causal inference task, as shown in Figure 1.

We first discuss our core design principles; and then present the major libraries and tools that make up our causal AI suite: DoWhy, EconML, Causica, and ShowWhy.

2.1 Design Principles and Decisions

Scaffolding the causal analysis process: The right API abstraction can guide users to follow best practices of causal inference and characterize the impact of causal assumptions on the estimated
effect. For instance, incorrect identification leads to bias in an estimate that cannot be corrected by simply optimizing across different estimation methods \cite{36}. In the absence of a global cross-validation evaluation procedure as in supervised machine learning, the steps for stating and validating assumptions are a critical part of causal inference \cite{33}.

Thus, a key design decision in our causal AI suite is organized around an analysis pipeline that elevates domain knowledge and its assumptions as a first-class entity. These assumptions dynamically guide the identification of algorithmic approaches, e.g., for causal estimation. And, these assumptions are all subject to validation, refutation, or sensitivity analysis in our pipeline.

**Separation of causal questions from algorithmic and methodological implementations:** Given the rapid pace of advancement in causal methods and machine learning more broadly, we expect to see significant changes in algorithms and methodological approaches for causal inference. To enable and encourage such advancement, we build API abstractions that cleanly separate how we represent and ask causal questions from how those questions are answered. The right abstractions also allow easier reuse and flexible combinations of algorithms and core components across related causal tasks.

For example, given a graphical causal model, the identification step of causal effect inference is a causal reasoning problem whereas the estimation step is a statistical problem. Our libraries enforce this boundary by having two separate API calls for identification and estimation. This allows modularity, allowing us to rapidly expand our set of supported identification and estimation methods independently. Furthermore, we can “mix-and-match” identification and estimation methods: multiple estimation methods can be used for a single identified estimand and vice versa.

**Find common abstractions across causal frameworks.** To encourage cross-pollination of ideas, we build abstractions that bridge across multiple causal frameworks and tasks, including graphical approaches to causal reasoning, potential outcomes approaches to effect inference, causal discovery algorithms, and others. The exercise of finding common ways of expressing the causal assumptions used by graphical reasoning methods, panel data methods, and causal discovery encourages us to consider both the similarities and distinctive strengths of methods, algorithms, and scenarios.

All of the components below are available as open-source Python packages\footnote{DoWhy: \url{https://github.com/py-why/dowhy/} EconML: \url{https://github.com/microsoft/econml/}; Causica: \url{https://github.com/microsoft/causica/}; ShowWhy: \url{https://github.com/microsoft/showwhy/}}.

### 2.2 DoWhy

DoWhy is an end-to-end library for causal analysis that builds on the latest research in modeling assumptions and robustness checks \cite{5,23}, and provides an easy API interface. To scaffold and guide users through the best practices of causal inference, DoWhy’s API is organized around the four fundamental steps for causal analysis: Model, Identify, Estimate, and Refute. **Model** encodes prior knowledge as a formal causal graph, **identify** uses graph-based methods such as the ID algorithm to identify the causal effect, **estimate** uses statistical methods, such as EconML’s sophisticated methods for estimating the identified estimand, and finally **refute** tests the validity of the assumed graph and resultant estimate. DoWhy unifies two powerful frameworks: causal graphs \cite{23} and potential outcomes \cite{19}. Specifically, DoWhy uses graph-based criteria and do-calculus for modeling assumptions and identifying a non-parametric causal effect. For estimation, it switches to methods based on potential outcomes, including methods from external libraries like EconML \cite{26} and CausalML \cite{9}.

In 2022, DoWhy became the first library to join the independent community-governed PyWhy organization as part of an effort to encourage the development of an open source ecosystem for causal machine learning.

### 2.3 EconML

The EconML library complements DoWhy with a focus on sophisticated estimation methods based on the latest advances in causal machine learning. EconML applies advanced machine learning techniques to estimate individualized causal responses from observational or experimental data. The suite of estimation methods provided in EconML represents the latest advances in causal machine
learning. Supervised machine learning algorithms (such as random forests, boosting, lasso, and neural nets) have become enormously effective at predicting an outcome $Y$ from a set of features. They achieve this good performance, and improve on earlier ad hoc methods, with automated model selection, usually with cross-fitting. The central advance of causal machine learning is to recognize that many individual steps within causal analysis can benefit from machine learning. EconML encodes this causal machine learning approach (e.g., DoubleML, meta-learners, and many others) into a simple API that builds intuitively on other popular ML packages such as scikit-learn. Armed with an appropriate data sample, analysts can estimate reliable causal effects with a few lines of code. In the background, EconML estimates “nuisance” functions using flexible ML models—including the option to use AutoML to choose across multiple algorithms, automatically cross-fits to tune those models—and builds these estimates into carefully constructed final models that use the latest techniques to deliver interpretable causal effects. Each approach in the library includes confidence intervals, many constructed analytically for added efficiency, and enables flexible modeling of heterogeneous treatment effects to understand how treatment effects vary across a population and create targeted, personalized plans of action.

2.4 Causica

Causica develops deep learning methods for end-to-end causal inference (DECI), using existing data and domain constraints to both perform causal discovery and compute causal inference quantities such as (conditional) average treatment effects. Causica also provides core functionality for a variety of causal tasks, including missing value prediction, best next question, causal discovery, and causal inference. The DECI model is a generative approach that employs an additive noise structural equation model (ANM-SEM) to capture the functional relationships among variables and exogenous noise, while simultaneously learning a variational distribution over causal graphs [14]. DECI estimates causal quantities (ATE / CATE) by applying the relevant interventions to its learnt causal graph and then sampling from the generative model.

2.5 ShowWhy

ShowWhy builds atop this ecosystem of Causal AI libraries to develop a new class of no-code user interfaces (GUIs) that empower domain experts to become “decision scientists” who can independently ask a causal question, develop causal estimates, and present and defend causal evidence to an audience of decision makers. We expect that making causal methods available to this broader audience of stakeholders will motivate and accelerate new use-inspired basic research. As new algorithms and methods are implemented underneath the shared APIs of the DoWhy, EconML, and Causica libraries, ShowWhy’s no-code user interface can easily leverage them.

Since the broader audience of ShowWhy users may not be familiar with specific machine learning estimators or causal inference more generally, a key design principle for our no-code interfaces is to use a range of estimators, problem specifications, and visualization techniques to communicate the overall balance of evidence regarding the existence and strength of potential causal relationships. ShowWhy’s no-code interfaces also recognize the fact that decision making is an ongoing process, often involving rapid iterative feedback loops, that extends beyond any individual decision. We can even view it as a new kind of decision science: first, discovering candidate interventions; second, deciding which interventions to make; and third, evaluating the impact of chosen interventions over time. Figure 2 presents the three major workflows supported by ShowWhy. The first workflow helps users answer exploratory questions of the form “do variables in some dataset have causal links? The next workflow helps users answer questions of the form “does exposure to a specific treatment cause a specific outcome?” The third and final workflow helps users answer questions of the form “did a specific event cause a specific outcome over time?”

3 Example usage

Parts of the causal suite are widely used across industry and academia: DoWhy and EconML, for example, have been downloaded over 1.5 million times. Through adoption and usage of these tools, we have found several templates of causal analysis scenarios to be commonly used: (1) identifying causal drivers of key metrics and outcomes; (2) return-on-investment analyses, such as identifying the
impact of customer loyalty programs; (3) robust prediction and forecasting, such as modeling future impacts of expected climate change; and (4) segmentation and personalized treatment scenarios.

Here, we present an example usage of these causal analysis libraries for a multi-investment attribution task, identifying the business actions that have a causal effect on revenue from enterprise customers.

3.1 Example: Multi-investment Attribution

A software company would like to know whether its multiple outreach efforts to their business customers are successful in boosting sales. They would also like to learn how to better target different incentives to different customers. In other words, they would like to learn the treatment effect of each investment on customers’ total expenditure on the company’s products—particularly the heterogeneous treatment effect.

Here, we show how tools from the Causica, EconML and DoWhy libraries can use historical investment data to learn the effects of multiple investments. We then use Causica to discover the causal graph: the relationship between each variable in the simulated data. With this generated graph, we use DoWhy to identify an appropriate strategy to estimate the causal effect. We pass this recommendation to EconML to estimate the personalized treatment effects for each customer. We also show an alternative effect estimation using Causica. Finally, we use DoWhy to test the assumptions underlying the causal estimation. The full walkthrough is available as a Jupyter notebook at https://aka.ms/causal_suite_notebook.

3.1.1 Data

We create a simulated dataset of 10,000 customers. Each customer is associated with annual revenue from the customer and whether they received technical support, a discount, or were targeted by new outreach strategies. Each customer is annotated with 10 additional characteristics that might affect revenue, such as the company’s employee count, whether the customer is a commercial or public sector, a global company, etc. Data is simulated to mimic realistic correlations between features.

3.1.2 Discovering the Causal Graph

The first step in our analysis, shown in Listing 1, is to extract the causal graph using a causal discovery algorithm, here using the DECI algorithm from Causica [14]. To inform the causal discovery procedure, we provide several simple constraints in the form of a constraint_matrix. In this case study, we declare that the revenue outcome variable cannot be the cause of other nodes; that certain attributes of companies cannot be changed by others; and that treatments do not cause each other. We omit this code for brevity.

Listing 1: Causal Discovery Code Example

```python
## Causal Discovery step
model = DECI.create("mymodel", modeldir, dataset.variables, cfg, device="gpu")
model.set_graph_constraint(constraint_matrix) # provide domain knowledge
model.run_train(dataset, training_params)
```
3.1.3 Treatment effect identification and estimation

The causal graph identifies the likely paths of connection between features. This next step will quantify the strength of those relationships between treatment and outcomes. Our tools estimate both the average treatment effect across all customers (ATE) and how these treatment effects vary across customer features.

DoWhy tools can help users identify an appropriate causal model for their question and understand which confounders, or conditioning variables, they should include in their estimation. To identify an estimation approach, DoWhy needs to work with a causal graph that describes the problem space. DoWhy can work with a causal graph generated through discovery, or input directly by users based on their domain knowledge, including perhaps identifying features that may affect the outcome and treatment but are unobservable.

Listing 2 runs DoWhy’s identification algorithm. In this scenario, DoWhy recommends a backdoor estimation approach, which relies on the unconfoundedness assumption, and selects all of the potential confounding variables except for one which was not identified as a confounder in the graph. We then use DoWhy to call the Linear Double Machine Learning model from the EconML library for the estimation step.

Listing 2: Identification and Estimation Code Example

```python
## Identification
dw = dowhy.CausalModel(data=dataset, graph=discovered_graph,
treatment="Tech Support", outcome="Revenue",
effect_modifiers=['Global Flag', 'Size'])
estimand = dw.identify_effect(method_name="maximal-adjustment",
estimand_type='nonparametric-cde')

# Estimate
estimate = dw.estimate_effect(estimand,
    confidence_intervals=True,
    method_name="backdoor.econml.dml.LinearDML")
```

3.1.4 Validation and Reporting

If causal assumptions are incorrect or otherwise violated, then identified causal estimands will be biased in ways that cannot be corrected through improved estimation methods ([35]). As validating assumptions is a critical part of causal inference ([33]), we briefly describe the refute step here. We leave code examples of refutation and validation out of the paper due to space considerations.

To aid data scientists’ inspection of the validity of causal assumptions, DoWhy provides two kinds of tests: 1) Refutations are necessary—but not sufficient—tests of the validity of a causal analysis must pass; and 2) Sensitivity analyses that measure the robustness of a causal estimate to violations of assumptions [32]. Tests supported by DoWhy include addition of random common causes, placebo treatments, data subset validation, bootstrap validation, and testing of independence constraints entailed by the causal graph.

4 Learning from practical deployment and next steps

These tools have been used in a wide variety of domains, including health and medicine [40, 12, 4, 3, 11, 10, 1, 37, 27, 6], logistics and travel [41, 8, 29, 39], marketing and sales [2, 17, 34], environment and agriculture [20, 13, 15], manufacturing [24], law [18], education [21, 7], and others. From the questions and feedback we receive, as well as from our use of these packages in our own application scenarios, we see several areas of both fundamental interest and with potential for practical impact.

**Improved elicitation of domain knowledge.** First, we see that data scientists and domain experts often have difficulty encoding domain knowledge correctly—whether because domain experts are
unavailable or for lack of causal expertise. As a result, we are seeking new ways to ease the elicitation of domain knowledge, including tighter integration of causal discovery and interactive tools for exploration and bootstrapping. Other promising directions include seeking to extract structural domain knowledge from other sources, such as the academic literature [30] and trusted simulators [25].

**Improved validation, refutation, and sensitivity analyses.** Continual improvement of validation and refutation methods, including integration with A/B experimentation methodologies when available is an important need for many of the scenarios we observe.

**Improved support for unstructured and high-dimensional text and image data.** In many application areas today, the richest source of data about a domain is text or image data, such as clinical notes in electronic health records, or satellite imagery capturing climate impacts. While methods for causal inference in the context of high-dimensional data are being explored [31][16], further work is needed to create approaches for analysis of unstructured data, e.g., building on causal representation methods [22].

Our next steps include continuing to grow the open-source community and ecosystem for Causal AI tools and libraries; expanding capabilities to new tasks and APIs, such as causal representation learning, and robust prediction. We encourage others to join us in creating a vibrant ecosystem that serves both researchers and practitioners.

## 5 Acknowledgments

We would like to thank Keith Battocchi, Mónica Carvajal, Denise Chen, Nathan Evans, Andrés Morales Esquivel, Jonathan Larson, Friederike Niedner, Dayenne de Souza, Christopher Trevino, Ha Trinh, Fabio Vera, Robert King, Ahmed Mustafa, and Thomas Kapler, our collaborators in the PyWhy organization, Peter Götz, Patrick Blöbaum, Kailash Budhathoki and all our open-source contributors for discussions and contributions building the Causal AI Suite.

## References


