Machine Learning (2024) 113:1329–1349
https://doi.org/10.1007/s10994-023-06431-9

Structural causal models reveal confounder bias in linear program modelling

Matej Zečević[1] · **Devendra Singh Dhami**[2,4] · **Kristian Kersting**[1,2,3]

Received: 10 June 2023 / Revised: 24 August 2023 / Accepted: 7 October 2023 /
Published online: 9 January 2024

Abstract
The recent years have been marked by extended research on adversarial attacks, especially on deep neural networks. With this work we intend on posing and investigating the question of whether the phenomenon might be more general in nature, that is, adversarial-style attacks outside classical classification tasks. Specifically, we investigate optimization problems as they constitute a fundamental part of modern AI research. To this end, we consider the base class of optimizers namely Linear Programs (LPs). On our initial attempt of a naïve mapping between the formalism of adversarial examples and LPs, we quickly identify the key ingredients missing for making sense of a reasonable notion of adversarial examples for LPs. Intriguingly, the formalism of Pearl's notion to causality allows for the right description of adversarial like examples for LPs. Characteristically, we show the direct influence of the Structural Causal Model (SCM) onto the subsequent LP optimization, which ultimately exposes a notion of confounding in LPs (inherited by said SCM) that allows for adversarial-style attacks. We provide both the general proof formally alongside existential proofs of such intriguing LP-parameterizations based on SCM for three combinatorial problems, namely Linear Assignment, Shortest Path and a real world problem of energy systems.

Keywords Adversarial-style examples · Causality · Linear programming

Editors: Vu Nguyen, Dani Yogatama.

✉ Devendra Singh Dhami
d.s.dhami@tue.nl

Matej Zečević
matej.zecevic@tu-darmstadt.de

Kristian Kersting
kersting@cs.tu-darmstadt.de

[1] Technical University of Darmstadt, Darmstadt, Hesse, Germany

[2] hessian.AI, Darmstadt, Hesse, Germany

[3] DFKI, Darmstadt, Hesse, Germany

[4] Eindhoven University of Technology, Eindhoven, Netherlands

1 Introduction

Adversarial attacks have gained a lot of traction in recent years (Brendel et al., 2018; Guo et al., 2019) as there has been a lot of focus on safety and robustness of machine learning (ML) systems. An interesting observation, though, is that deep neural networks or rather over-parameterized models are the center of attention for most of such adversarial attacks (Zügner et al., 2018; Chen et al., 2018). We argue that this view is incomplete or even too narrow in the sense that the phenomenon around adversarials is more general *in nature* and actually depends on the problem setup. We conjecture that any differentiable perturbed optimizer (DPO) is prone to this new notion of attack similar to classical adversarials that we discuss in this paper. DPOs are a well studied, pragmatic approach to differentiability of general mathematical program (MP) solver by means of perturbation, consider (Papandreou & Yuille, 2011; Berthet et al., 2020; Gumbel, 1954; Bach, 2013) for reference. If our conjecture were to be true, then our new view on adversarial attacks would stand as a very general problem beyond learning to classify. While this might turn out to be more of a scientifically/mathematically valuable insight rather than practical implication, as we prove in this paper, there are examples that we can construct which are clearly of high relevance. As we will see, 'classical' classification adversarial examples might still pose a higher significance in terms of research in deep learning as they pose a threat to trust and explainability, however, attacks on LPs certainly hold major significance as well if we consider that many real world applications of high relevance, such as energy systems or online navigation services, depend on them. There has been previous works where MPs such as LPs but also Mixed Integer Programs (MIP) (Wu et al., 2020; Tjeng et al., 2019) have been used to compute adversarial attacks but not where such optimization modules (LP, MIP) themselves have been confronted with the attacks. In fact, and also due to the recent interest in tightly integrating MPs and deep learning (Amos & Kolter, 2017; Paulus et al., 2021), this extension of adversarial attacks beyond deep networks already significantly advances our understanding of adversarial attacks i.e., it is not just expressiveness that leads to uninterpretable solutions with counter-intuitive properties. These two key arguments serve as motivation to why studying adversarials in LPs (and more broadly MPs) is important beyond pure scientific inquiry.

Interestingly, it turns out that the new type of attack we formalize develops naturally from the Pearlian notion of Causality (Pearl, 2009) when starting from the formalism of classical adversarial attacks. Put differently, the mathematical theory of causality as given by Pearl provides the right formal tools to establish a reasonable interpretation of adversarial examples in LPs. Speaking of causality, the subject refers to a very general idea, in that understanding causal interactions is even central to human cognition and thereby of high value to science, engineering, business, and law (Penn & Povinelli, 2007). In the last decade, causality has been thoroughly formalized in various instances (Pearl, 2009; Peters et al., 2017; Hernán & Robins, 2020). At its core lies a Structural Causal Model (SCM) which is considered to be the model of reality responsible for data-generation. An SCM is a powerful model in that it is capable of many things. The SCM implies a graph structure over its modelled variables, and furthermore when specified in its entirety it can reason about (hidden) confounders, and of course handle both interventions and counterfactuals. The richness of the SCM has been crucial for its successful application for ML in marketing (Hair Jr & Sarstedt, 2021), healthcare (Bica et al., 2020) and education (Hoiles & Schaar, 2016). While we conjecture the applicability of our paper's results to the general class of DPO (which is an effective sub-class of MPs), the focus of this work will be to

motivate, illustrate and finally prove formally that we can exploit an SCM's hidden confounders to construct a new type of attack based on the classical adversarial attacks in order to attack LPs—which is the very first, basic sub-class of MPs. We coin this new attack Hidden Confounder Attacks, since exploiting knowledge of hidden confounders is both a necessary and sufficient condition for the construction of these adversarial-style examples.

Overall, we make a number of key contributions: (1) We derive for the first time a novel, theoretical connection between causality's SCMs and LPs, by which we then (2) use the hidden confounders of the SCM to devise an adversarial-style attack—which we call Hidden Confounder Attack (HCA)—onto the LP showing that non-classification problems can be prone to adversarial-style attacks; (3) We study and discuss two classical LP families and one real world applied optimization problem to further motivate research on HCA and their potentially worrisome consequences if being ignored. For reproduction, we make our code repository publicly available.[1]

2 Background and related work

In the following, we will briefly review the background on adversarial attacks as defined in their original setting of classification, then the formalism of LPs alongside two famous problem instances (linear assignment and shortest path, both of which we will use later on), and finally SCMs with their causal mechanism and hidden confounders. We use mathematical notation for (i) to *precisely* specify and capture important ideas and (ii) to eventually prove our theoretical insights, however, the reader is invited to skip formal details as they are not central for grasping the new ideas proposed in our paper, yet, a consideration will provide technical understanding about assumptions, limitations and reach of what is being proposed.

2.1 'Classical' adversarial attacks (classification)

We are in the setting of classification, specifically, image classification where the task of the model is to give the 'right' label to a given image fed as model input. By using a simple optimization procedure, Szegedy et al. (2014) were able to find what they called 'adversarial examples', which they defined to be imperceptibly perturbed input images such that these new images were no longer classified correctly by the predictive neural model. Note how we specifically talk about *neural* models here as in the regular deep learning context. Goodfellow et al. (2015) then proposed the Fast Gradient Sign Method (FGSM) that considers the gradient of the error of the classifier w.r.t to the input image. Mathematically, they investigated perturbations of the form

$$\boldsymbol{\eta} := \epsilon \operatorname{sign}\left(\nabla_{\mathbf{x}} J(\mathbf{x}, y; \boldsymbol{\theta})\right) \in \mathbb{R}^{w \times h \times c} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$ is the input image, $y \in \mathbb{N}$ a class label, $\boldsymbol{\theta}$ are the neural function approximator parameter, $J : \mathbb{R}^{w \times h \times c} \times \mathbb{N} \to \mathbb{R}$ a scalar-valued objective function, $\operatorname{sign} : \mathbb{R} \to [-1, 1]$ an element-wise sign function and $\epsilon \in \mathbb{R}$ a free-parameter. A perturbation $\boldsymbol{\eta}$ would then account for mis-classification of the given predictive model $f(\mathbf{x}; \boldsymbol{\theta})$ i.e.,

[1] https://anonymous.4open.science/r/Hidden-Conf-Attacks/.

$$f(\mathbf{x};\boldsymbol{\theta}) = y \neq f(\mathbf{x}+\boldsymbol{\eta};\boldsymbol{\theta}). \tag{2}$$

The inequality in Eq. 2 represents a possibly strongly significant divergence from the expected *semantic* meaning (i.e., from a human inspector's perspective) of the class to be predicted. For example, imagine a photograph of a dog that is being classified by f as a dog (that is f predicts the label 'dog'). However, the perturbed image $\mathbf{x}+\boldsymbol{\eta}$ is not classified by f as 'dog' but rather as, say, 'washing machine.' What came to a surprise for many is two-fold, (1) the new classification could be something drastically different e.g. not another animal like a cat but for instance a washing machine (2) from a human perspective the perturbed image would still lead to the same classification i.e., still a dog (put differently, the human inspector cannot tell a difference between the original and perturbed images). Naturally, said susceptibility (1-2) led to a significant increase in research interest regarding robustness (to adversarial examples) in neural function approximators evoking the narrative of "attacks" and subsequent "defences" on the inspected classification modules, as commonly found in alternate literature such as cyber-security (Handa et al., 2019).

2.2 Mathematical programming/optimization

Selecting the best candidate from some given set with regard to some criterion (or objective) is a general description of MPs (or just 'optimization'), which arguably lies at the core of machine learning and many applications in science and engineering since we are in search of models and solution that are somehow the 'best.' Classification, e.g, can be considered as a special instance of mathematical programming, where the optimum is reached when the model is able to provide the correct label each time it is being queried. An important (if not, the most important and fundamental) optimization family are LPs that are concerned with the optimization of an objective function and constraints that are *linear* in the respective optimization variables. LPs are being applied widely in the real world, e.g., energy systems (Schaber et al., 2012). More formally, the optimal solution of an LP is given by

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \mathrm{LP}(\mathbf{x};\mathbf{w},\mathbf{A},\mathbf{b}) \tag{3}$$

$$= \arg\max_{\mathbf{x}\in\mathcal{P}_{\mathbf{A},\mathbf{b}}} \langle \mathbf{w},\mathbf{x}\rangle, \tag{4}$$

where $\langle \mathbf{a},\mathbf{b}\rangle := \mathbf{a}^\top \mathbf{b} = \sum_i a_i b_i \in \mathbb{R}$ is the inner product (dot product), $\mathbf{w} \in \mathbb{R}^n$ is called the weight/cost vector, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ are the constraint matrix and vector respectively, and finally $\mathcal{P}_{\mathbf{A},\mathbf{b}}$ is the solution polytope (or feasible region) i.e., the convex subset of state space $\mathbf{X}$ such that each $\mathbf{x}$ satisfies the constraints. Formally, $\mathcal{P}_{\mathbf{A},\mathbf{b}} := \{\mathbf{x} | \mathbf{A}\mathbf{x} \leq \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}\} \subset \mathbb{R}^n$ (if clear from context, we abbreviate to $\mathcal{P}$). Table 1 presents two classical problems that can be expressed as linear programs: the *Linear Assignment* Problem (LA) and the *Shortest Path* Problem (SP). Both problems formulate the optimization variable $\mathbf{x} \in \mathbb{R}^n$ with either $n = |A \times B|$ or $n = |E|$ to be a selector/indicator variable. That is, for LA we have the respective dimensions to mean matches between different worker and tasks, whereas for SP they denote the edges that are part of the fianl, selected path that should end up being the shortest path in the network. Although the original formulation of the LA and SP problems are actually *integer* LP formulations, which are generally known to be NP-complete opposed to the less restrictive regular LPs since they require the solutions to be integers and not some arbitrary real number, both problems can still be solved in *polynomial* time.

Table 1 Classical problems formulable as LPs

$\forall i \in A : \sum_{j \in B} x_{ij} = 1$	$\sum_{(i,j) \in E} x_{ij} - \sum_{(i,j) \in E} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{else} \end{cases}$
$\forall j \in B : \sum_{i \in A} x_{ij} = 1$	
$x_{ij} \in [0, 1]$	$x_{ij} \in [0, 1]$

Linear assignment (left; abbrev. LA) and Shortest Path (right; SP). In LA one matches "workers" to "tasks" according to their "skills". In SP one finds the "quickest", valid path (collection of edges) from some node i to node j within the given graph/network. The constraints on the left specify rules such as each worker can only have one task and any task can only have one worker, whereas the constraints on the right define a 'valid' path, that is, if one enters a certain node, then one needs to also exit out of said node to continue with a valid path.

However, extensions of regular SP like the Travelling Salesman or the Canadian Traveller problems are known to be NP- and PSPACE-complete respectively without any such benefits as LA/SP beg to offer.

2.3 Causality

The question of causality is a highly philosophical, timeless question and subject of study by the likes of Plato and his fellows, but only recently has the AI/ML community started investing more into causality as a means for the next generation of intelligent systems using new formalizations that capture certain key ideas rigorously. Following the Pearlian notion of Causality (Pearl, 2009), an SCM is defined as a 4-tuple $\mathcal{M} := \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ where the so-called structural equations

$$v_i = f_i(\mathrm{pa}_i, u_i) \in \mathcal{F} \tag{5}$$

assign values (denoted by lowercase letters) to the respective endogenous/system variables $V_i \in \mathbf{V}$ based on the values of their parents $\mathrm{Pa}_i \subseteq \mathbf{V} \setminus V_i$ and the values of their respective exogenous/noise/nature variables $\mathbf{U}_i \subseteq \mathbf{U}$, and $P(\mathbf{U})$ denotes the probability function defined over $\mathbf{U}$. In other words, f is the causal (possibly physical) mechanism that converts values of the parent variables to the values of the variable interest, or how Pearl says "V_i listens to pa_i." We usually say that pa_i are the *direct causes* of V_i. Note that, opposed to the Markovian SCM discussed in for instance (Peters et al., 2017), the definition of $\mathcal{M}$ is semi-Markovian thus allowing for shared U between the different V_i. Such a U is also called hidden confounder since it is a common cause of at least two $V_i, V_j (i \neq j)$. Opposed to a hidden confounder, a "common" confounder would be a common cause from within $\mathbf{V}$ (that is, we would have a specific name for that given confounder, it would not be in $\mathbf{U}$). An important concept that the formalization later on will require is the concept of causal sufficiency. Following Spirtes (2010): "The set of endogenous variables on which SCM $\mathcal{M}$ enacts is called *causally sufficient* if there exist no hidden confounders." Put differently, if our modelled system has no unobserved confounders (or we simply assume it to be that way), then we can call our system causally sufficient. While SCM provide a formalization (a language) for reasoning over (possibly hidden) confounders, the practical consideration of confounders is difficult and requires positing of (often times overly strong) assumptions. Both adjustment for confounders and the identification of confounding structures in the graph is a challenging task, which increases in difficulty when the confounder are unmeasured. For

completion's sake, we mention more interesting properties of any SCM. The SCM induces (i) a causal graph G, (ii) an observational/associational distribution denoted $p^{\mathcal{M}}$, and (ii) they can generate infinitely many interventional and counterfactual distributions using the *do*-operator.[2]

3 Step-by-step derivation of adversarial examples for linear programs

This section covers our key contribution and develops it step-by-step from ground up. This main section is structured in the following manner: we first discuss MPs/DPOs and how our results are expected to transfer to those to motivate the overall research direction and justify the investigation of LPs as initial case. Then, secondly, we present how a naïve mapping of the classical adversarial attack framework fails for LPs in the sense that we could not claim it to be an adversarial example (or even something similar). Then, thirdly, we present how the tools from causality can provide additional semantics to formulate our new adversarial-style attack, which we refer to as Hidden Confounder Attacks (HCA). In the fourth subsection, we conclude with what we consider to be important mathematical insights for HCA.

3.1 Overarching hypothesis and the importance of differentiability

In the past, different classes of MPs (LPs, MIPs) have been used defensively for verifying the robustness of neural learners to adversarial examples (Tjeng et al., 2019) and offensively for generating actual adversarial examples (Zhou et al., 2020). Here, we are concerned with a fundamentally different research question: *"How do adversarial attacks affect MPs themselves?"*. That is, we turn the table and instead of considering MPs as a service to the system to be attacked, we consider the programs themselves to be the system under attack. Our overarching hypothesis for this and possible future work is the following:

> **Hypothesis:** Adversarial examples or attacks refer to a concept more general than that of classification in that it also affects MPs. Thereby, adversarial examples are a property of the problem specification and not per se a property of the expressiveness of deep models or of the classification task.

To the best of our knowledge, we are the first researchers to ask and investigate this question thoroughly. Therefore, in order to establish an initial connection between adversarial attacks and MPs we will start off with the most basic class of MPs: Linear Programs. Since an adversarial attack typically depends on gradient/first-order information to determine where the perturbation (or attack) is most effective, we also require such first-order information from our LPs. One way to achieve this is to consider the class of ML models which inject some noise, which is distributed w.r.t. some differentiable probability distribution, into the LP optimizer. These so-called perturbed models have differentiability properties because of that. Therefore, these perturbed models have also been considered for inference tasks within energy models (Papandreou & Yuille, 2011) and regularization in online settings (Abernethy et al., 2014) as immediate consequence of said differentiability. Initial works in this research direction date back to the Gumbel-max (Gumbel, 1954) and were recently generalized to *Differentiable*

[2] Loosely speaking, the *do*-operation "overwrites" structural equations.

Perturbed Optimizers (DPO) featuring end-to-end learnability (Berthet et al., 2020). As stated in the initial section of this paper, we conjecture (that is believe to be true) that DPO are susceptible to the same (or similar) style of adversarial examples that we are developing in this paper. To formulate an LP optimizer, $\mathbf{x}^*(\mathbf{w}) = \arg\max_{\mathbf{x}\in\mathcal{P}_{\mathbf{A},\mathbf{b}}} \langle \mathbf{w}, \mathbf{x}\rangle$, as a DPO one requires only the existence of a random noise vector $\mathbf{z} \in \mathbb{R}^n$ with positive and differentiable density $p(\mathbf{z})$ such that for $\epsilon \in \mathbb{R}_{>0}$,

$$\mathbf{x}^*(\hat{\boldsymbol{w}}) = \mathbb{E}_{p(\mathbf{z})}[\arg\max_{\mathbf{x}\in\mathcal{P}_{\mathbf{A},\mathbf{b}}} \langle \mathbf{w} + \epsilon\mathbf{z}, \mathbf{x}\rangle], \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable living in the solution polytope $\mathcal{P}_{\mathbf{A},\mathbf{b}}$ described by LP constraints $\mathbf{A}, \mathbf{b}$, where $\hat{\boldsymbol{w}} := \mathbf{w} + \epsilon\mathbf{z}$ is the peturbed LP cost parameterization, $\langle \cdot, \cdot \rangle \in \mathbb{R} \in \mathbb{R}^n$ the inner product and $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{X})]$ the expected value of random variable $\mathbf{X}$ under the predictive model f. Related work on differentiability of more general MPs like quadratic/cone programs (Agrawal et al., 2019) or linear optimization within predict-and-optimize settings (Mandi & Guns, 2020) generally rely on the Karush-Kuhn-Tucker (KKT) conditions. The general advantage of a perturbation method (as in Eq. 6) over the analytical approaches is its "black-box" nature i.e., we don't require to know what kind of MP we are dealing with, since we simply add stochasticity into the problem. In other words, we don't need to be experts on any specific MP (or any MP at all for that matter) to be able to reap the benefits of differentiability. This "invariance" to the underlying MP and the fact that differentiability is a necessary key concept behind adversarial attacks, leads to following (informally stated) conjecture.

Conjecture 1 (HCAs on MPs, informal) *Differentiability is a necessary condition for constructing Hidden Confounder Attacks on MPs* (*to be defined in Sect*. 3.4).

3.2 Naïve mapping or "trying to make sense of what adversarial examples could mean in LPs"

Let's start our derivation of HCA by first providing a naïve mapping/perspective between/for the classical adversarial attack and the famous class of LPs known as Linear Assignment (LA), both of which have been previously introduced in Sect. 2. Mathematically, the following correspondence can be found,

$$\begin{aligned} \mathbf{x} + \boldsymbol{\eta} &:= \hat{\boldsymbol{w}}, & f_\theta &:= \mathbf{x}^*(\cdot), \\ y &:= \mathbf{x}^*(\mathbf{w}), & J &:= F(\hat{\boldsymbol{w}}, \mathbf{w}), \end{aligned} \tag{7}$$

where $\mathbf{x}$ is the feature vector (e.g. an image), y the class label, f_θ the neural predictive model, and J the cost function (e.g. mean-squared error)—all of these symbols follow the notation from Goodfellow et al. (2015). From the LP perspective, we interpret $\mathbf{w} \in \mathbb{R}^{|A|\times|B|}$ as describing the suitability of worker $a \in A$ for job $b \in B$ and the optimal solution $\mathbf{x}^*(\mathbf{w}) \in [0, 1]^{|A|\times|B|}$ as indicators highlighting the matched pairs (a_i, b_j). The only addition we have to make to achieve the naïve mapping to adversarials is some distance measure F acting on the original $\mathbf{x}^*(\mathbf{w})$ and the expected perturbed solution $\mathbf{x}^*(\hat{\boldsymbol{w}})$. This is necessary since we need to allow for a "class" change between solutions that should occur (or are caused) through an adversarial attack i.e., in the case of LA, F could be for instance the Structural Hamming Distance (Hamming, 1950). Also note, in slight abuse of notation, our program solver $\mathbf{x}^*(\cdot)$ denotes $\arg\max_{\mathbf{x}} \text{LP}(\mathbf{x}; \mathbf{w}, \mathbf{A}, \mathbf{b})$.

Having completed the naïve mapping in LA specifically, we could then naïvely view each optimal matching "code" $\mathbf{x}^*(\mathbf{w})$ as a certain class (or label) and then the gradient $\nabla_{\mathbf{w}} F$ could be used for performing an 'adversarial attack' such that the 'class' changes (significantly) while the input remains approximately the same. As one can easily make out, the major problem being faced here is that there exists no "semantic impact" to be observed for the human inspector akin to a neural network wrongly classifying a dog (small animal) as a plane (big travel machine). In other words, there is change but said change is not significant (or surprising, and thus not interesting). To make this point more clear, we summarize our key insight about adversarial examples obtained by looking at our naïve mapping onto LPs:

Missing Semantic Component. The human inspector's invariance to the adversarial example is characteristic of an adversarial attack (e.g. still dog-looking image being classified as depicting a washing machine), while a naïve mapping to LPs as in (7) leaves out said human component. In other words, there is no general human expectation towards different optimal solutions to an LP since humans measure LPs only on their objective and therefore different optimal solutions are not 'different.'

3.3 Concepts from causality provide the missing semantic component

In the previous section we concluded that there is a missing semantic component when naïvely mapping between adversarials and LPs. Yet, the optimal solutions when considered in terms of codes as in the LA example will actually significantly *deviate* from each other. This deviation (or difference) seems to suggest that there exists some more 'fundamental' difference in solution albeit not for the specific optimization objective at hand since the cost values will remain similar (or even the same). But as suggested by the missing semantic component, a human inspector will have no general expectation towards either of the LP solutions. To put it differently, "they look different but that is that." Nonetheless, to complete the picture it is worth taking a step back and observing the LP from a 'meta' level perspective. *On this meta level, we can ask the question of how the LP cost vector* $\mathbf{w}$ *was provided in the first place.* Here causality and its SCM come into play. The SCM $\mathcal{M}$ defines a mechanistic data-generating process which will generate the observational probability distribution $p^{\mathcal{M}}$ that the human modeller usually observes an empirical fraction of, denoted as data $\mathbf{D}$. So, to loop back to the meta-question of how the LP parameterization came to be, we observe the following relation for some function ϕ:

$$\phi(\mathbf{D}) = \mathbf{w} \quad \text{where } \mathbf{D} \sim p^{\mathcal{M}}. \tag{8}$$

According to Eq. 8, the human modeller that takes the observed data as a basis for determining the cost vector of the LP and then uses some function-mapping between the SCM and the LP denoted as ϕ to produce said cost vector. To give a concrete example of such a modelling, consider Fig. 1 in which the human modeller observes a data set $\mathbf{D} := \{(h_i, p_i)\}_i^n \sim p(H, P) = p^{\mathcal{M}}$ but no other information. The human modeller *does neither* observe the SCM induced distribution $p^{\mathcal{M}}$ *nor* any more information about the partial SCM $\mathcal{M}$ which would include knowledge on the structural equations f_H,f_P and the hidden confounder U_C (i.e., U_C is shared by both equations). Therefore, also no information on the true SCM $\mathcal{M}^*$ either, where U_C would be part of the endogenous variables (i.e., there would also be f_W and all U_C being replaced by W standing for Wealth). The only knowledge available is the data set $\mathbf{D} \sim p^{\mathcal{M}}$ which numerically describes the Health (H) and a general notion of Vaccine Priority (P) of certain individuals. Note that a simple linear

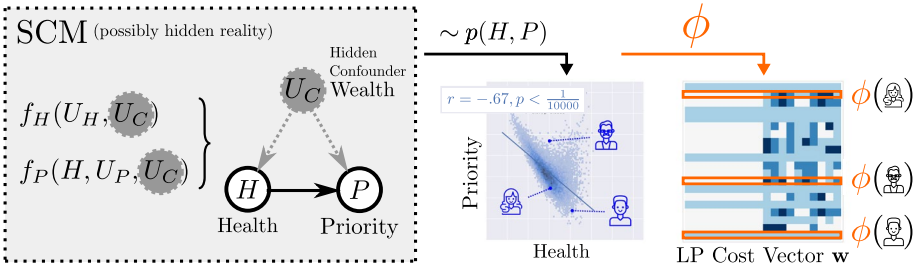

Fig. 1 Intriguing LP-Parameterizations Based on SCM. The observed observational distribution $p(H, P)$ was generated by some unobserved SCM. Even if we had *some* SCM, it might not be the actual underlying, complete SCM i.e., there could still be hidden confounders in our estimate. The cost vector $\mathbf{w}$ of an LP can be viewed as a function ϕ applied to population individuals $(h, p) \sim p(H, P)$. *Intriguingly, ϕ may very well be unaware of confounders in the true SCM* (Color figure online)

regression on the data shown in Fig. 1 reveals the existence of a causal relation between H and P (but not the direction). The true causal direction is read as "lower health values cause higher priority values" and written $H \rightarrow P$. However, P describes other factors as well (for instance the age of an individual) since P also depends on U_P (and not only on its cause H and the hidden confounder U_C).

As in the previous section, let's consider an LA problem as our LP instance. We will make the example explicit. The setting is that of a pandemic and recently it has been announced that there is a new vaccine that can help in stopping the pandemic. For this, people need to get vaccinated and so it is now up to a human modeller scheduled for making a plan on assigning the empty vaccine spots. Unfortunately, the amount of spots is limited and so there must be some sort of rule to decide how to actually assign the available free spots. The human modeller is trying to find the *optimal matching* of individuals (based on the data that covers relevant information/features about the individuals) to respective, available vaccination spots. The human modeller might choose to find the cost parameterization of the LP that will determine the optimal matching by following some sort of *policy* (or rule) like *"individuals of low health and high priority should be matched to vaccine spots, while others can wait."* More importantly, in this case, we would now argue that the human modeller *implicitly* performed a mapping ϕ as in Eq. 8 based on the observed data, and that this ϕ essentially captures the policy description from before. Both interestingly and intriguingly, by construction, ϕ *does not* consider the hidden confounder U_C—that might be viewed as something like wealth of an individual—since U_C is not even defined within the data distribution accessible to the human modeller (since $\mathcal{M} \neq \mathcal{M}^*$). This ignorance is the key to defining a meaningful adversarial-style attack on LPs since we can use it to explicate what a change in optimal solutions could mean. In informal terms, we are now ready to formulate our main idea of the paper:

Theorem 1 (HCAs on LPs, informal) *Let $\phi_{\mathcal{M}}$ denote an LP parameterization based on SCM $\mathcal{M}$ while $\mathcal{M}^*$ denotes the 'true' or optimal SCM for the phenomenon of interest. Any $\phi_{\mathcal{M}}$ that identifies individuals in $\mathcal{M}$ is prone to Hidden Confounder Attacks (to be defined in Sect. 3.4) unless $\mathcal{M} = \mathcal{M}^*$.*

Put loosely, choosing the 'wrong' $\phi_{\mathcal{M}}$, one that does not represent the true, underlying SCM $\mathcal{M}^*$, allows for an adversarial-style attacks on LPs. In summary we state:

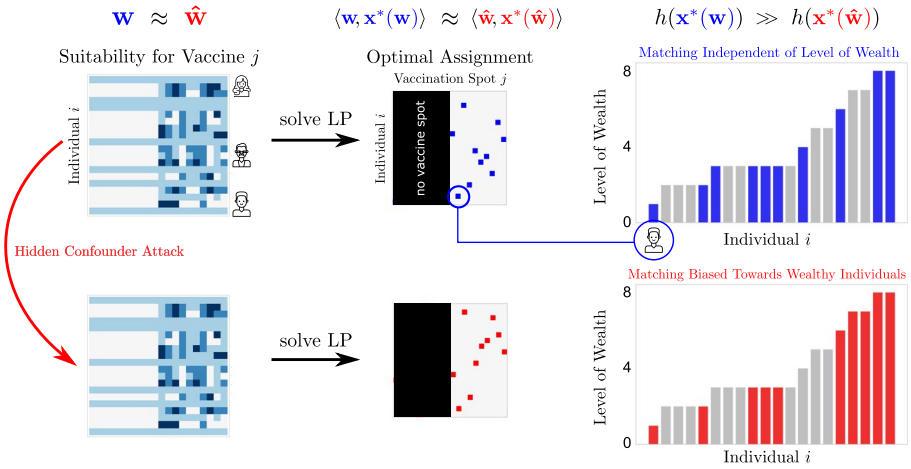

Fig. 2 Lead Example: Hidden Confounder Attack on LP. A real world inspired matching problem under attack. The new matching shows significant bias for individuals with high wealth value. The adversarial LP cost vector is close to the original both value-wise (left) and cost-wise w.r.t. their optimal solution (middle) which means in words that health-wise people in higher need of vaccination are still guaranteed a vaccine spot. However, w.r.t. hidden confounder (Wealth) the adversarial solution drastically deviates (right) i.e., the distribution of vaccines is being skewed towards people of higher wealth, which can circumvent the originally intended policy (Color figure online)

Bias in Vaccine Spot Assignments, Example from Fig. 2: The LP-parameterization $\phi_{\mathcal{M}}$ based on the SCM $\mathcal{M}$, with $\mathbf{w} = \phi(\mathbf{D})$ from Eq. 8 where $\mathbf{w}$ is the LP cost vector, follows the previously, informally described policy, so $\phi_{\mathcal{M}}$ models only H, P since U_C is not even defined in $\mathbf{D}$. Therefore, Thm. 1 predicts the existence of a HCA for LP($\mathbf{x}$;$\mathbf{w}, \mathbf{A}, \mathbf{b}$). Figure 2 reveals such an example HCA: the attack is unnoticable in visual terms, just as for a classical adversarial example, and so is the difference in cost w.r.t. the optimal matchings—however, w.r.t. to the wealth value of each individual, the new matching shows a significant skew towards individuals (or data points) of higher wealth value.

Rationale for Hidden Confounders within Mathematical Programming Considerably orthogonal at first glance as this bridge might seem, both Pearlian causality (and associated concepts such as hidden confounders) and Mathematical Programming are concerned with *modelling assumptions* that lie outside the data. While the MP is arguably data-independent per definition of the paradigm, MPs such as LPs can in fact be modelled through data. To give an example, a warehouse that is concerned with profits will record its sales and eventually settle on optimizing the profits based on the recorded data (using both the data features or variables and insights on customer preferences), which in turn is formulated as an MP. Since data that we are given (or that is being recorded in the example of the warehouse) is assumed to be governed by some underlying dynamics, it is safe to assume that we can denote said underlying data-generating process with causality's center-piece formalism being that of SCM. In conclusion, we naturally find situations in which our approximation to the SCM that might be underlying our data (what is typically called our *induction hypothesis*) is insufficient in that there are hidden confounders. However, since we used the SCM's data to model an MP previously, we now have an eventuality of hidden confounders within our MP instance.

This rationale is the very basis of the present manuscript that proves the existence of hidden confounders within MPs that are modelled in accordance to an insufficient SCM (even in the cases where the MP generation is unaware of the assumptions placed on the SCM part of the modelling of the cost and constraint vectors).

Next, to complete the discussion, we finally formalize the informal notions presented in this section. However, it is worth noting that technical parts of this paper can be safely skipped as understanding them is not central to understanding the overall idea (as just now presented informally). Naturally, reading the technical part allows for a precise understanding of the assumptions and key aspects to our definition of HCA and subsequent insights.

3.4 Formalizing the newly proposed LP attacks "hidden confounder attacks"

Notation We use standard notation from deep learning for various mathematical base concepts such as scalars (a), vectors ($\mathbf{a}$), matrices ($\mathbf{A}$), sets (A), parameterized functions (f_θ). In the cases where a set is actually denoted with (what is considered typically to be) matrix notation, we consider said set to be of special meaning in that it is literature-specific notation where the different considered literatures are adversarial examples, linear programming and causality. In the following, we go over each. From literature specific to adversarial examples we use: perturbation ($\boldsymbol{\eta}$). We also make use of linear programming specific notation: optimal solution ($\mathbf{x}^*$), cost vector or matrix ($\mathbf{w}$), inner product ($\langle\cdot,\cdot\rangle$), linear constraints ($\mathbf{A}, \mathbf{b}$). Finally, since HCA are based on causality's conception of confounders, we use causality notation as well: SCM ($\mathcal{M}$), exo- and endogenous variables ($\mathbf{U}, \mathbf{V}$). With the matching of the two paradigms as in Eq. 7, we can consider the notational extensions for HCA introduced in this work as a 'generalization' of the previous notations deployed for discussing adversarial-style attacks. In other words, either side of the definitions (in the mapping of Eq. 7) can be used to denote HCA. The authors opt for notational convenience in the sense that notation which is useful for a given context should be deployed e.g. when discussing optimal solutions then $\mathbf{x}^*(\mathbf{w}) \neq \mathbf{x}^*(\mathbf{w}')$ is easier to parse than $y \neq y'$ since the former notation explicates that optimal solutions under different cost vectors are being considered, whereas the latter might only be interpreted as a difference in scalars. We now introduce extended fosrmalism to capture all the previously established ideas precisely, to define HCA and discuss its properties. We first start with Eq. 8.

Definition 1 We call a function $\phi_{\mathcal{M}}$ LP-parameterization based on SCM $\mathcal{M}$ if for an observational data set $\mathbf{D} \sim p^{\mathcal{M}}$ we can define an LP($\mathbf{x}$;$\phi(\mathbf{D})$, $\mathbf{A}$, $\mathbf{b}$).

By default (unless clear from context), we might refer to such $\phi_{\mathcal{M}}$ simply as LP-parameterizations. Some LP-parameterizations fulfill a property that "identifies individuals in $\mathcal{M}$" which we define next.

Definition 2 Let $\phi_{\mathcal{M}}$ be an LP-parameterization based on SCM $\mathcal{M}$ with $\phi_{\mathcal{M}}(\mathbf{D}) = \mathbf{w} = (w_1, \ldots, w_k)$ and $\mathbf{D} = \{\mathbf{d}_i\}_i^n$. We call $\phi_{\mathcal{M}}$ integral if it satisfies:

$$\forall i \in \{1, \ldots, n\}.\exists I \subset \{1, \ldots, k\}.(\mathbf{d}_i = \phi_{\mathcal{M}}^{-1}((w_j)_{j \in I}).$$

In words, the parameterization decomposes on the the data point (or unit) level.

These first two definitions give us the ability to talk about 'special' types of LPs, namely those that underly some SCM and further some that even allow for talking about the different units U_j of these SCM (exogenous variables). A simple yet important insight that immediately follows is:

Corollary 1 *The LP problems Linear Assignment* (*LA*) *and Shortest Path* (*SP*) *have integral LP-parameterizations*.

Proof For LA, simply map each data point indexed by i to the cost vector slice indexed by indices in the set I s.t. each i refers to the same unique a from the "workers" set A for all the different "jobs" $b \in B$. I.e., one data point sampled from the SCM's observational distribution will correspond to one worker in the LP cost vector. For SP, there is a more direct one-to-one correspondence where each data point is mapped to a unique edge in the graph. □

This insight is important since LA and SP constitute arguably the two most important LPs in existence as accounted by their widespread use in applications in ML and beyond. Before we can define HCA, we need to state our two main assumptions that are necessary to HCA:

Assumption 1 For some fixed constraints $\mathbf{A}, \mathbf{b}$ let $\mathbf{X}^*(\mathbf{w}) := \{\mathbf{x} \mid \mathbf{x} = \arg\max_{\mathbf{x}} \text{LP}(\mathbf{x}; \mathbf{w}, \mathbf{A}, \mathbf{b})\}$ denote the set of optimal LP solutions under $\mathbf{w}$. Further let $B_\epsilon^{\mathbf{w}}$ denote an ϵ-Ball with $\epsilon > 0$ around some LP cost $\mathbf{w}$. Then there exists a $\hat{\boldsymbol{w}} \in B_\epsilon^{\mathbf{w}}$ such that $|\mathbf{X}^*(\hat{\boldsymbol{w}})| > 1$. In words, we can find an ϵ-close LP instance with multiple optimal solutions.

Assumption 2 Like before, let $\mathbf{X}^*(\mathbf{w})$ be the set of optimal LP solutions under cost vector $\mathbf{w}$. Further, let $\mathbf{x}^*(\mathbf{w}) \in \mathbf{X}^*(\mathbf{w})$ denote the solution returned by our solver and let $\hat{\boldsymbol{w}} = \mathbf{w} + \epsilon \nabla_{\mathbf{w}} F$ be the perturbed cost vector for some function F and $\epsilon > 0$. We assume $\mathbf{x}^*(\mathbf{w}) \neq \mathbf{x}^*(\hat{\boldsymbol{w}})$. In words, the perturbed LP instance returns a different optimal solution.

Arguably, both assumptions are fairly weak and compare to what can be found in standard adversarial learning literature, yet, it is crucial to make them explicit both for transparency on the given setting and for proving our theorem of HCAs on LPs. Now, we are set to give the technical description of what a Hidden Confounder Attack really is:

Definition 3 Let $\phi_{\mathcal{M}}$ be an LP-parameterization based on SCM $\mathcal{M}$. We call $\phi_{\mathcal{M}}$ prone to Hidden Confounder Attacks if there exists an injective function $h : \mathbf{X}^* \to \mathbb{R}$ with properties

1. $h(\mathbf{x}^*) = f(\bigoplus_{i \in \mathbf{x}^*} \mathcal{M}'_C(i))$ and
2. $\exists \hat{\boldsymbol{w}}.(h(\mathbf{x}^*(\mathbf{w})) \neq h(\mathbf{x}^*(\hat{\boldsymbol{w}})))$

for some function $f : \mathbb{R} \to \mathbb{R}$, aggregation function $\bigoplus$ over units i active in $\mathbf{x}^*$ (e.g. the sum for all matched "workers"), and LP cost vector $\mathbf{w}$ where

$$\mathcal{M}'_C : \mathbb{N} \to \text{Val}(C)$$

is the projection of a unit i to its respective confounder value $\mathcal{M}'_C(i) \in \text{Val}(C)$ where $\mathcal{M}'$ is an alternate SCM containing C. That is, C is a hidden confounder of $\mathcal{M}$.

In simple terms, property 1 in Def. 3 refers to the uncountable number of functions that can leverage information on the hidden confounder by using the alternate SCM $\mathcal{M}'$ to distinguish between different optimal LP solutions which is required by property 2 (since otherwise, there would be no observed difference, alas no attack). Finally, we can provide our key result, that we have encountered previously in informal terms, in its complete formal version.

Theorem 2 (HCAs on LPs, formal) *Let $\phi_{\mathcal{M}}$ be an integral LP-parameterization based on SCM $\mathcal{M}$, then we have*:

$$\mathcal{M} \text{ is causally insufficient } \iff \phi_{\mathcal{M}} \text{ is prone to HCA.}$$

Proof "⇒": As discussed in Sect. 2, for an SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ to be causally insufficient there must exist at least one hidden confounder, denoted C, that is not an endogenous variable, $C \notin \mathbf{V}$. Therefore, for any LP-parameterization $\phi_{\mathcal{M}}$ and any LP cost vector $\mathbf{w}$, the latter also doesn't depend on C. Then, take an alternate $\mathcal{M}' = \langle \mathbf{U}', \mathbf{V}', \mathcal{F}', P'(\mathbf{U}) \rangle$ for which $C \in \mathbf{V}'$ and construct h as described by property 1 from Def. 3, which is guaranteed to exist since $\phi_{\mathcal{M}}$ is integral. Through Assumption 1 we have guaranteed multiple optimal LP solutions for LP($\mathbf{x}$;$\mathbf{w}, \mathbf{A}, \mathbf{b}$) to choose from. On the other hand, through Assumption 2 we can perturb said LP cost vector $\mathbf{w}$ such that $\mathbf{x}^*(\mathbf{w}) \neq \mathbf{x}^*(\hat{\boldsymbol{w}})$. Since h is injective, we have that $h(\mathbf{x}^*(\mathbf{w})) \neq h(\mathbf{x}^*(\hat{\boldsymbol{w}}))$ which is property 2 of Def. 3 completing the implication.

"⇐": Trivial, since HCA (Def. 3) are defined as attacks that exploit hidden confounders. □

This fundamental Theorem of our formalism on HCA guarantees us that the existence of hidden confounders implies the susceptibility of LPs to HCAs. In fact, we can even *construct an uncountable number* of HCAs based on said confounders. We further argue that the HCAs that follow from Thm. 2 are highly non-trivial in the sense that they exploit information "outside of the data" i.e., assuming that the human modeller only uses a causally insufficient $\phi_{\mathcal{M}}$, then an adversary is guaranteed a chance to exploit his/her better knowledge on the true, underlying SCM to perform an attack. Also, a simple corollary of Thm. 2 is that LA and SP are *always* prone to HCAs since they have integral $\phi_{\mathcal{M}}$ and the odds are stacked against the model under consideration $\mathcal{M}$ actually being the true, underlying SCM $\mathcal{M}^*$ (that is, in most practical cases we will encounter the situation where $\mathcal{M} \neq \mathcal{M}^*$). Another way of looking at HCA is possibly even more intriguing: since we need to take care of modelling assumptions to prevent HCA, the modelling of LPs ultimately becomes a *causal problem* since causality is mainly concerned with the discussion of modelling assumptions (usually for identifiability of causal quantities, whereas in this case for the robustness of an LP towards HCAs).

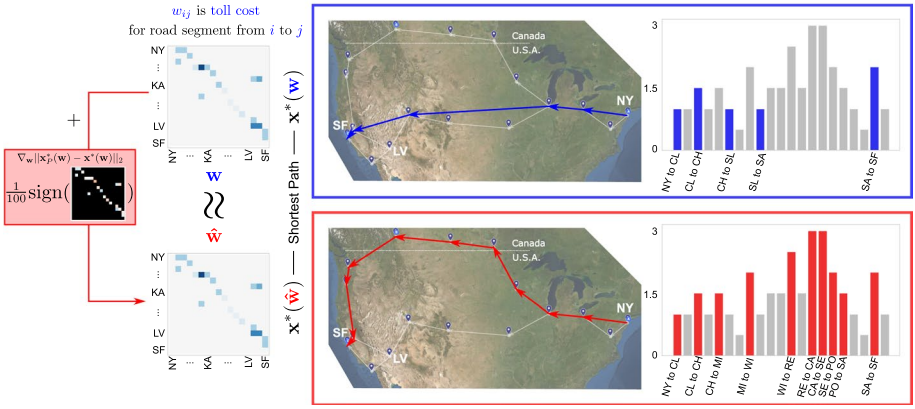

Fig. 3 Another Example: Increased CO_2 Emissions. The edges in the graph represent tolls to be paid for travelling a given road segment. The hidden confounder are CO_2 emissions. We visualize the results of performing a HCA using said hidden confounder on our SP-LP instance. Our HCA reveals that travelling via Canada instead of mid-US will amount to the same total travel toll to be paid but the CO_2 emissions drastically deviate between the solutions (Color figure online)

4 Empirical evidence

This section is purely dedicated to discussing the existence of relevant HCA apart from the example that we have already discussed with the LA problem of assigning people to free vaccination spots. We first cover a SP for travelling between cities and then also a general LP for a real world inspired model of an energy system.

4.1 Shortest path LP

The caption of this example might be the following captivating headline for a newspaper article: "Travelling from New York City to San Francisco...*via Canada*?". Surely, when travelling between NYC and SF, one would not take the detour over Canada, but this is exactly what occurs in our example in which we perform a HCA on a corresponding LP. Like before with our HCA on the LA-LP with the vaccine example, we depict the application of the HCA visually within a figure. This new example is being showcased in Fig. 3 which is concerned with a classical Shortest Path (SP) problem and we discuss the details in the following. In a corresponding real world setting for the example, we might consider the development of an autonomous car (since arguably any conscious driver would surely notice passing border patrol when actually travelling such a route) to argue for realism. We let the developmental autonomous car travel within North America from New York City (NY) to San Francisco (SF). Our SP has the intention of reducing overall toll costs for the optimal route. Therefore, the SP-LP is not concerned with the shortest route in terms of actual travelling distance but in terms of actual toll costs accumulated (noting this distinction is important). From experience, it is known that these toll costs can be hefty. Our LP cost $w_{ij} \in \mathbb{R}_{>0}$ represents the toll cost when travelling on any road segment from i to j. In this example, we know the toll costs for a relevant set of road segments within North America where the Canadian road toll policy is comparably modest when compared

Table 2 Real-world optimization modelling example: 1-year energy systems LP for an average household

$$\min_{Cap,p} \quad c_{PV} \times Cap_{PV} + c_{Bat} \times Cap^{S}_{Bat} + \sum_t c_{Ele} \times p_{Ele}(t) + \sum_t c_{Gas} \times p_{Gas}(t)$$
$$s.t. \quad p_{Ele}(t) + p_{PV}(t) + p^{out}_{Bat}(t) - p^{in}_{Bat}(t) + p_{Gas}(t) = D(t), \forall t \quad 0 \leq p_{Ele}$$
$$p^{S}_{Bat}(t) = p^{S}_{Bat}(t-1) + p^{out}_{Bat}(t) - p^{in}_{Bat}(t), t \in 2, \ldots, T$$
$$0 \leq p_{PV}(t) \leq Cap_{PV} \times avail_{PV}(t) \times \delta t, \forall t$$
$$0 \leq p^{in}_{Bat}(t), p^{out}_{Bat}(t) \leq Cap_{Bat}, \forall t$$
$$0 \leq p_{Gas}(t) \leq U_{Gas}, \forall t$$
$$p^{S}_{Bat}(0) = 0$$

A large LP that unrolls for 8760 time steps (8760 h = 1 year). Model based on Schaber et al. (2012), the quantities represent: Cost for Photovoltaics c_{PV} (€/kW), Battery c_{Bat} (€/kWh), Market Electricity c_{Ele} (€/kWh), Gas c_{Gas} (€/kWh), and the total Demand D (kWh/Year).

to the one in the US. Our LP model subsequently solves any given SP problem instance, fully parameterized by the directed acyclic graph (DAG) $\mathbf{w} \in \mathbb{R}^{n \times n}$ with n being the total number of different cities we have specified, returning $\mathbf{x}_{US} := \mathbf{x}^*(\mathbf{w}) \in [0,1]^{n \times n}$ suggesting a route through the mid-US. We change this result by constructing a HCA. The HCA leads to a minimal perturbation of the original DAG (representing the toll costs), that is $\hat{\boldsymbol{w}} \approx \mathbf{w}$, but our solver now chooses an alternate solution $\mathbf{x}_{CA} := \mathbf{x}^*(\hat{\boldsymbol{w}})$ suggesting a route across the border via Canada.[3] While evidently the alternate route deviates strongly in terms of selected road segments, mathematically $SHD(\mathbf{x}_{US}, \mathbf{x}_{CA}) \gg 0$ where $SHD(\cdot, \cdot) \in \mathbb{N}$ is the Structural Hamming Distance, our model is in fact trustfully returning the optimal solution. In other words, cost-wise the statement $\mathbf{w}^\top \mathbf{x}_{US} \approx \hat{\boldsymbol{w}}^\top \mathbf{x}_{CA}$ holds. Nonetheless, the aforementioned deviation in terms of the resulting binary codes lends itself to a severe consequence in terms of the hidden confounder i.e., with respect to CO_2 emissions. Like in the main text, we construct an HCA with function h accordingly. The hidden confounder is being exploited by the adversary, the alternate optimal solution performs significantly worse: $h(\mathbf{x}_{CA}) \gg h(\mathbf{x}_{US})$. In words, both solutions require approximately the same toll costs and are therefore deemed equivalent in that regard but in terms of CO_2 emmissions, the (distance-wise) longer route via Canada is far worse for the environment. By this, we have again provided existential proof that a hidden confounder can more generally define adversarial attacks for mathematical programs beyond the original formulation in the classical setting for classification (and deep networks), making the attack a consequence of not the specific methodology being applied to the problem but problem specification itself.

4.2 Energy-systems LP

In this final empirical simulation, we consider a *large scale* LP. Furthermore, this LP is a general LP, so neither LA nor SP and thereby it does not satisfy *integrality*. Our real world based example considers an energy model for modelling the energy portfolio of a single-family house based on actual real world data for demand and commonly used equations from energy systems research (Schaber et al., 2012) to model the evolution of respective quantities. The examined model considers concepts such as photovoltaics (PV), market electricty and heating gas over a year time frame (in hours) and resembles a simplified version of the TIMES

[3] Remember, it is essential for the technical Assumption 1 and 2, discussed in the previous section, to hold.

Table 3 Dominating technologies

Dem. (h)	Cap_{PV}	Cap_{Bat}	Self-Gen	TOTEX	CAPEX	Con_{Gas}	Con_{Ele}	w_{PV}
3000	1.76	2.45	0.42	597.41	161.64	1.70	1743.06	.005
3000	7.15	4.78	0.66	468.24	214.87	1.95	1013.49	.001

We perform an approximate HCA to reveal a new solution that highlights the fact that PV end up as a new "dominating technology." Price perturbations in (w_{PV}) have boosted PV production Cap_{PV} (green), which then again can be argued leads to a significant increase in risk of fire or injuries for the workers installing the panels.

model Loulou et al. (2005). The optimal solution balances the usage of the different technologies for matching the required demand of the family household such that overall cost is being minimized. The specific LP template is given in Table 2. Note the summation and functional dependencies on $t \in \{0, \ldots, 8760\}$ with 1 year = 8760 h rendering the template a *very large single LP modelling each hour of the year* with well over 40,000 constraints and an objective function with over 17,000 terms. Still, there are some technologies like for instance PV, in their capacity (Cap_{PV}), that do not depend on t which would correspond to the real world intuition that one does not decide and subsequently build new PV for any given hour as it poses a single investment fixed in time. Since in this example we cannot specify the underlying SCM to then also identify confounders, as previously done for the LA- and SP-LPs, we need to treat this LP instance differently. We use heuristics for the SCM-part to produce an approximation to HCA to overcome the fact that we are not provided with a reasonable SCM a-priori. We then perform said approximation to produce an HCA that creates the results presented in Table 3. We observe the effect which energy-systems researchers call "dominating technologies," where PV is being preferred over market-bought electricity. While we do not have a function h this time to evaluate the difference in solutions for the adversary, we can still make the argument that building this many PV modules comes at an increased risk of fire (which could be considered the hidden confounder in this case). Another possible interpretation would be risk of working injury for the panel installing workers, since installing the panels usually happens at the upper level height of the house. To conclude this example with an important discussion, we want to mention that the limitations on PV-production and Market-buy of electricity act as discrepancy counter-measures that require the system to balance out different technologies i.e., while there will still be dominating technologies under price advantages the maximum skew of the portfolio is naturally being protected from being too drastic as both PV and bought electricity are limited in their 'availability' (e.g. solar exposure, roof capacity, law regulations etc.) and thus cannot be naïvely maximized. In other words, we observe that this LP behaves qualitatively a little different when compared to the LA- and SP-LP examples in the sense that this energy system LP is more 'balanced' and thus somehow less prone to HCA. The aforementioned lack of integrality might be one of the reasons, but there is reason to believe based on the previous argument of the *dynamics of the competing technologies* that this might be the main cause. A precise formalization of these aspects is a remaining open problem.

4.3 Synthetic simulations, scalability and key assumptions

We've conducted several experiments of synthetic nature on LP problems. These LP problems included well-known integer problems such as matching or shortest path, and further

general LP formulations as in the case of energy-systems. While the LP problems themselves are considerably data-independent, the presented HCA formalism relies on SCM as data-generating processes, therefore, sample or data set sizes in the LP parameterizations $\phi_{\mathcal{M}}$ can vary. Unsurprisingly, since Def. 1 puts a constraint on the data set under consideration there will be no downstream influence by the data set size onto the HCA. That is, since hidden confounders are a property of an SCM (on the model level) they do not have an influence on the data set we consider (on the sample level). Similarly, our simulations corroborate on the invariance (or rather orthogonality) of HCA to variable scales as this is a general property of the LPs under consideration e.g., the scale of a variable in the SP setting that denotes "road segment will be taken, yes/no" can not be changed. In an analogue argument for the general LP, the variables having well-specified scales is a defining (thus necessary) property e.g. the quantities in the energy-system example need to satisfy given physical conservation laws.

Regarding the scalability of HCA, it can be added that the knowledge on a confounder and subsequent h is always $\mathcal{O}(1)$, that is, independent of the size of any given SCM $\mathcal{M}$, if that $\mathcal{M}$ is causally insufficient, then a single hidden confounder is sufficient for constructing a h that acts as HCA. Other than the causal aspect, regular scaling properties of DPOs apply to the construction of HCA. Concerning the usage of different LP solvers for obtaining optimal solutions $\mathbf{x}^*$: the presented approach crucially builds upon DPOs that are characterized by their differentiability. Differentiability being a crucial property for adversarial examples, makes it a necessary condition for HCA as well, as they can be viewed as an extension to LP adversarial-style examples. In conclusion, only differentiable LP solvers can be employed, and to the best of our knowledge no other solvers, apart from the ones employed here, have been studied thus far. Regarding the assumptions, there are more nuanced views to be considered for justification purposes. The existence of an underlying SCM is the foundational assumption in Pearlian causality and can be taken as granted. Similarly, the fact that the underlying SCM is only being approximated by any given model of the data. Therefore, knowledge on hidden confounders is a strong assumption. In our case, this assumption is relaxed since only a *single* hidden confounder need be known. Regarding the two technical assumptions 1 and 2 in this work, the latter is concerned with a 'tie-break' resolution, that is, if there exist multiple solution to a given LP parameterization, then a certain permutation will always favour a certain solution. This is a straightforward assumption since the parameterization can be altered arbitrarily by an ϵ change. However, the prior assumption which is concerned with the *existence* of multiple solutions within an ϵ-ball is more demanding than in the standard adversarial case since LPs are polytopes of potentially complex shape. Nonetheless, on a conceptual level, the presented showcases for LA/SP demonstrate that various situations of multiple-solution sets for LPs do occur in practice.

5 Conclusive discussion

Ultimately, we believe HCAs to be a fundamental problem of mathematical optimization—to the same extent that hidden confounding is a fundamental problem of the Pearlian notion of causality (or science in general). It is intriguing that the formal tools from causality allowed for bridging the gap between classical adversarial attacks from deep learning and the first basic class of mathematical optimization namely LPs. While it is arguably of great scientific value to purely study the existence and properties of

HCA, naturally, the question arises on the severity of HCA for the real world. We believe that our examples have shown potentially worrisome real-world implications. Especially our lead example in Fig. 2 captures the *Zeitgeist* of the pandemic times with the rise of Covid (that hopeful has found an end finally). To thus ask the inverse question on how to defend against HCA is equally important, yet, we believe this question to be ill-posed to begin with. *Essentially, Thm.*2 *suggests an equivalence on the existence of hidden confounders and such attacks.* Put differently, as long as our model assumptions are imperfect, we are exploitable—again, giving us reason to believe HCA to be of fundamental nature. However, that does not mean that we are doomed to accept that HCA will always be something that can happen anywhere but rather take the opportunity to further study HCA beyond LPs (as initially conjectured, see Conj.1) but also alternate notions of attacks, in order to really understand what our assumptions cover and what not. With HCA we have presented one new way of thinking of adversarial style attacks and its serves as a representative example of what we mean by studying these phenomena. While our perspective put causality to good, there might exist other notions of attacks similar to HCA that might in fact not be based on causal knowledge (confounders) to begin with. Since we were able to develop HCA from first principles, by starting from classical adversarial attacks and naïve mappings to LPs, we have good reason to believe in the actual existence of related families of attacks. From a theoretical standpoint the question of whether the integral property (Def. 2) applied to Thm. 2 could be dropped might be interesting for broadening the applicability of HCA, as we saw with our energy systems example where we still achieved some sort of reasonable HCA approximation although integrality did not hold.

On another note, we observe this work to form a triangular relationship to the works of Ilse et al. (2021) and Eghbal-zadeh et al. (2020). To elaborate: the first publication is concerned, again, with Pearlian causality and sees how it relates to data augmentation, while the second paper bridges augmentation and adversarials—our work can be seen as the missing link that then loops back adversarials to causality. From that perspective, we can clearly see that there seems to be an overarching research theme yet to be uncovered. Separating our discussion from HCA for the moment, we nonetheless are tempted to believe that HCA (although being the focus and motivation behind this work) are *not* the most important discovery in this paper. The concept of LP-parameterization based on SCM (Def. 1) is an intriguing and original concept that for the first time connects the two seemingly independent notions of causality and mathematical optimization in a non-trivial manner. So, it might turn out to be more fruitful long-term to actually study the properties of mathematical programs which stand in direct relation to the data-generating capabilities of SCM since that might lead to interesting concepts such as the estimation of *causal effects* in, say, constraints of mathematical programs. To put it bluntly, our current thinking imagines future research around LP-SCM relations as even more fundamental than HCA, but ideally both HCA and related attacks should be further studied alongside LP-SCM relations since both have real world implications. Arguably, for causality the latter is more important, whereas for ML the former is more important.

Takeaway and Societal Implications Our work seems to suggest (1) that we can have a similar, adversarial phenomenon outside classification and (2) that the current state of ML can be viewed 'causal' to the extent that the assumptions are 'causal' i.e., ϕ summarizes essentially these causal assumptions. Of concern are mostly (a) raising awareness on the issue that adversaries could in fact use HCA as shown in the examples to produce harm and (b) provide incentives for studying the integration of ML with causality since modelling assumptions seem to lie at the core of it.

Table 4 Parameterization energy-system

c_{PV}	c_{Bat}	c_{Ele}	D	c_{Gas}
0.005	300	0.25	3000	0.25
0.001	300	0.25	3000	0.25

Cost for photovoltaics c_{PV} (€/kW), battery c_{Bat} (€/kWh), market electricity c_{Ele} (€/kWh), gas c_{Gas} (€/kWh), and the total demand D (kWh/Year).

6 Reproduction details

LA-LP: For the LA example, with the vaccination matching bias towards the wealthy, we use $N = 15$ sampling iterations with temperature parameter $\sigma = 0.5$ for the perturbation in the DPO as defined by Berthet et al. (2020) and an attack step $\epsilon = 0.01$ for the final HCA. SP-LP: For the SP example, travelling from NY to SF via Canada, we use more sampling iterations ($N = 20$) using a lower temperature ($\sigma = 0.25$). Real World LP: The energy systems LP has following parameter specifications (Table 4):

All experiments are being performed on a MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports) laptop running a 2,3 GHz Quad-Core Intel Core i7 CPU with a 16 GB 3733 MHz LPDDR4X RAM on time scales ranging from a few minutes (e.g. evaluating LA/SP examples) up to a few hours (e.g. energy systems real world example). Code link in Sect. 1.

Author Contributions Matej Zečević (MZ), Devendra Singh Dhami (DSD), Kristian Kersting (KK) contributed to conception of the idea. MZ performed the data preparation and experimental analysis. MZ wrote the first draft of the manuscript. DSD and KK wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the ICT-48 Network of AI Research Excellence Center "TAILOR" (EU Horizon 2020, GA No 952215), the Nexplore Collaboration Lab "AI in Construction" (AICO) and by the Federal Ministry of Education and Research (BMBF; project "PlexPlain", FKZ 01IS19081). It benefited from the Hessian research priority programme LOEWE within the project WhiteBox and the HMWK cluster project "The Third Wave of AI" (3AI). The authors thank Jonas Hülsmann and Florian Steinke for providing the LP model for the energy system example.

Data availability We make our code publicly available at https://github.com/zecevic-matej/Hidden-Confounder-Attacks

Declarations

Conflict of interest The authors have no competing interests to declare.

Ethics approval Ethics approval was not required for this research.

References

Abernethy, J., Lee, C., Sinha, A., & Tewari, A. (2014). Online linear optimization via smoothing. In *Conference on learning theory* (pp. 807–823). PMLR.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., & Kolter, Z. (2019). Differentiable convex optimization layers. *Neural Information Processing Systems*, *32*, 9562–9574.

Amos, B., & Kolter, J. Z. (2017). OptNet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning* (pp. 136–145). PMLR.

Bach, F. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning, 6*(2–3), 145–373.

Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., & Bach, F. (2020). Learning with differentiable perturbed optimizers. *Neural Information Processing Systems, 33*, 9508–9519.

Bica, I., Alaa, A., & Van Der Schaar, M. (2020). Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International conference on machine learning* (pp. 884–895). PMLR.

Brendel, W., Rauber, J., & Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models, In *International conference on learning representations*. https://openreview.net/forum?id=SyZI0GWCZ

Chen, S.-T., Cornelius, C., Martin, J., & Chau, D. H. P. (2018). Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector. In *Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18* (pp. 52–68). Springer International Publishing.

Eghbal-zadeh, H., Koutini, K., Primus, P., Haunschmid, V., Lewandowski, M., Zellinger, W., Moser, B. A., & Widmer, G. (2020). On data augmentation and adversarial risk: An empirical analysis. arxiv preprint arxiv:2007.02650

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International conference on learning representations*. arxiv.org/abs/1412.6572

Gumbel, E. J. (1954). *Statistical theory of extreme values and some practical applications: A series of lectures* (Vol. 33). US Government Printing Office.

Guo, C., Gardner, J., You, Y., Wilson, A. G., & Weinberger, K. (2019). Simple black-box adversarial attacks. In *International conference on machine learning* (pp. 2484–2493). PMLR.

Hair, J. F., Jr., & Sarstedt, M. (2021). Data, measurement, and causal inferences in machine learning: Opportunities and challenges for marketing. *Journal of Marketing Theory and Practice, 29*(1), 65–77.

Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Booktitle, 29*(2), 147–160.

Handa, A., Sharma, A., & Shukla, S. K. (2019). *Machine learning in cybersecurity: A review*. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.

Hernán, M. A., & Robins, J. M. (2020). *Causal inference: What if*. Chapman & Hall/CRC.

Hoiles, W., & Schaar, M. (2016). Bounded off-policy evaluation with missing data for course recommendation and curriculum design. In *International conference on machine learning* (pp. 1596–1604). PMLR.

Ilse, M., Tomczak, J. M., & Forré, P. (2021). Selecting data augmentation for simulating interventions. In *International conference on machine learning* (pp. 4555–4562). PMLR.

Loulou, R., Remme, U., Kanudia, A., Lehtila, A., & Goldstein, G. (2005). *Documentation for the times model part II. Energy technology systems analysis programme*. International Energy Agency.

Mandi, J., & Guns, T. (2020). Interior point solving for LP-based prediction+ optimisation. *Neural Information Processing Systems, 33*, 7272–7282.

Papandreou, G., & Yuille, A. L. (2011). Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *2011 international conference on computer vision* (pp. 193–200). IEEE.

Paulus, A., Rolínek, M., Musil, V., Amos, B., & Martius, G. (2021). *CombOptNet: Fit the right NP-hard problem by learning integer programming constraints*. arxiv preprint arxiv:2105.02343

Pearl, J. (2009). *Causality*. Cambridge University Press.

Penn, D. C., & Povinelli, D. J. (2007). Causal cognition in human and nonhuman animals: A comparative, critical review. *Annual Review of Psychology, 58*, 97–118.

Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of causal inference*. MIT Press.

Schaber, K., Steinke, F., & Hamacher, T. (2012). Transmission grid extensions for the integration of variable renewable energies in Europe: Who benefits where? *Energy Policy, 43*, 123–135.

Spirtes, P. (2010). Introduction to causal inference. *Journal of Machine Learning Research, 11*, 1643–1662.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *International conference on learning representations*. arxiv.org/abs/1312.6199

Tjeng, V., Xiao, K. Y., & Tedrake, R. (2019). Evaluating robustness of neural networks with mixed integer programming. In *International conference on learning representations*. https://openreview.net/forum?id=HyGIdiRqtm

Wu, K., Wang, A., & Yu, Y. (2020). Stronger and faster wasserstein adversarial attacks. In *International conference on machine learning* (pp. 10377–10387). PMLR.

Zhou, N., Luo, W., Lin, X., Xu, P., & Zhang, Z. (2020). Generating multi-label adversarial examples by linear programming. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Zügner, D., Akbarnejad, A., & Günnemann, S. (2018). Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2847–2856).