Causal Mosaic: Cause-Effect Inference via Nonlinear ICA and Ensemble Method

Pengzhou (Abel) Wu
The Graduate University for Advanced Studies

Kenji Fukumizu
The Institute of Statistical Mathematics

Abstract

We address the problem of distinguishing cause from effect in bivariate setting. Based on recent developments in nonlinear independent component analysis (ICA), we train general nonlinear causal models that are implemented by neural networks and allow non-additive noise. Further, we build an ensemble framework, namely Causal Mosaic, which models a causal pair by a mixture of nonlinear models. We compare this method with other recent methods on artificial and real world benchmark datasets, and our method shows state-of-the-art performance.

1 INTRODUCTION

Causal discovery (Spirtes and Zhang, 2016; Peters et al., 2017) is a fundamental problem which attracts increasing attention recently. The golden standard of causal discovery is randomized controlled experiments, but they often encounter ethical and practical issues. Thus, causal discovery from pure observational data provides an indispensable way to understand our nature. Traditionally, causal discovery algorithms learn the causal structure in the form of a directed acyclic graphical (DAG) model, by searching in the space of possible DAGs (Drton and Maathuis, 2017). Constraint-based search methods (e.g. FCI (Spirtes et al., 2000)) use conditional independence tests to determine the causal structure. Score-based search methods, such as GES (Chickering, 2002), typically search for a graph that optimizes a penalized likelihood score. However, the above methods are not applicable to bivariate case and unable to fully determine edge directions in a DAG.

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

In recent years, a line of research emerges that is particularly motivated to solve the problem of distinguishing cause from effect in bivariate case, i.e. cause-effect inference. All these methods exploit cause-effect asymmetry to identify causal direction (Mooij et al., 2016). One major approach is to restrict causal mechanism to a certain class of "functional causal models" (FCMs) (Hyvärinen and Zhang, 2016), and the causal direction between C and E is identifiable if $p(E|C)$ can be fitted by this class, while the opposite direction, $p(C|E)$, cannot. Typical FCMs are LiNGAM (Shimizu et al., 2006), ANM (Hoyer et al., 2009), PNL (Zhang and Hyvärinen, 2009) and ANM-MM (Hu et al., 2018). And all of them assume additive noise. Many other methods loosely exploit the idea that the process generating cause distribution $p(C)$ is in some way "independent" to the causal mechanism generating conditional distribution $p(E|C)$ (Janzing and Scholkopf, 2010). For example, IGCI (Janzing et al., 2012) uses orthogonality in information space to express independence between the two distributions. KCDC (Mitrovic et al., 2018) is based on the invariance of Kolmogorov complexity of conditional distribution. RECI (Blöbaum et al., 2018) extends IGCI to the setting with small noise, and proceeds by comparing the regression errors in both possible directions.

We can observe the following limitations in the existing methods. First, FCMs put too strong restrictions on the functional form of causal mechanism. Second, other works tend to propose simple "principles" that actually reflect the authors' own intuitions on causality. Thus, most methods fail to achieve high accuracy on real world data. Third, there are a few methods (e.g. KCDC, CGNN (Goudet et al., 2018)) that use more flexible models and achieve better performance, but without theoretical justifications. Fourth, they assume there exist no hidden confounders.

This work studies cause-effect inference and address the first three[1] limitations respectively as follows.

[1]To deal with confounders, we can combine our method with graphical search methods. See Supplementary Material.

First, we train nonlinear causal models on cause-effect pairs with (maybe partial) direction information, based on a recent nonlinear ICA method implemented by neural network, without strong restriction on the functional relationship among the variables or the noise structure. Second, the fact that each of the many approaches to causality works to some limited extent suggests us to take a "mosaic" view: causal systems are diverse and heterogeneous, so we should not fit all the different systems at once; instead, study at a time a small number of causal systems that share common aspects, and then build a whole picture. Specifically, we build an ensemble of nonlinear models, which amounts to a Causal Mosaic: a causal pair's mechanism is treated as a mixture of similar mechanisms. It is analogous to constructing a large piece of mosaic from tesserae, which are small blocks of material used in creating a mosaic. Finally, we provide theoretical results on the conditions under which our method will work.

The main contributions of this paper are : 1) two novel cause-effect inference rules with identifiability proofs, 2) an ensemble framework that works for real world datasets with only limited labeled pairs, 3) a neural network structure designed for causal-effect inference, and 4) state-of-the-art performance on a real-world benchmark dataset.

Related work RCC (Lopez-Paz et al., 2015) and its follow-up NCC (Lopez-Paz et al., 2017) also use training data, but they require large numbers of labeled pairs and thus rely on synthetic pairs for training. There is work which takes related viewpoints: KCDC uses majority voting, the simplest ensemble method; ANM-MM treats mechanism as a mixture. NonSENS (Monti et al., 2019) also employs the same nonlinear ICA method as ours, but needs samples of a casual system available over different environments, which requires interventions or even experiments. We should note that all the above methods neither take a mosaic view explicitly nor use ensemble method as a main building block.

2 PRELIMINARIES

2.1 Intuition

As mentioned, we encounter a large diversity of causal relationships in nature. And causality might only be studied and learned piecemeal. Our idea is to extract the common mechanism shared by a small number of causal systems. We should note that, systems that seem to have different mechanisms can actually share the same mechanism. When all we have at hand is observational data, the sample, it would be true that two systems sharing the same mechanism, but by looking at the samples, they seem very different, to the extent that we would be tempted to model them by different functional forms. As an example, we give some pairs we used in experiment in Figure 1.

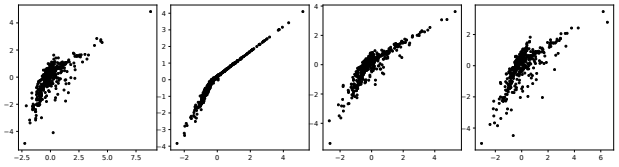

Figure 1: Artificial causal pairs sharing same mechanism. The pairs have significant diversity though still show some regularity. Please refer to Section 5 for details.

In the following subsections, we first formally introduce our problem setting, then show its connection to nonlinear ICA, and finally review the nonlinear ICA method which we exploit to learn shared mechanism.

2.2 Notation and Problem Setting

Generally, causal inference problems can be formalized by Structural Causal Models (SCMs) (Pearl, 2009), also known as Structural Equation Models (SEMs) (Bollen, 1989). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a causal DAG, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. Then, the SCM of observed variables $\mathbf{X} = (X_v)_{v\in\mathcal{V}}$ and *independent* hidden variables $\mathbf{E} = (E_v)_{v\in\mathcal{V}}$ is given by the set of equations [2]:

$$X_v = f_v(X_{pa_{\mathcal{G}}(v)}, E_v), v \in \mathcal{V} \tag{1}$$

f_v represents the *causal mechanism* between effect X_v and its direct causes (parents in the graph) $X_{pa_{\mathcal{G}}(v)}$. And E_v models exogenous (external) influences on X_v and is often treated as an unobserved noise.

Figure 2: Causal graphs of bivariate SCMs

In this work, we focus on bivariate cases, where there are only two possibilities: either X_1 or X_2 is the direct cause of the other, as shown in Figure 2. Their SCMs are the following (2) for $X_1 \to X_2$, and (3) for $X_2 \to X_1$. In cause-effect inference, our goal is to distinguish between these two possibilities, that is, tell cause from effect.

$$X_1 = f_1(E_1), \qquad X_2 = f_2(X_1, E_2) \tag{2}$$
$$X_1 = f_1(X_2, E_1), \qquad X_2 = f_2(E_2) \tag{3}$$

[2] As typical definition of SCM, we rule out *feedback* loops (two-way causal influences) and *confounders* (hidden common causes) here.

2.3 Nonlinear ICA and Causal Discovery

A straightforward definition of the generative model for nonlinear ICA is that independent hidden variables $\mathbf{Z} = (Z_1, ..., Z_n)$ are mixed by a differentiable and invertible nonlinear function $\mathbf{f}$, and produce observed variables $\mathbf{X} = (X_1, ..., X_n) = \mathbf{f}(\mathbf{Z})$. The goal is to recover the independent components Z_i and the unmixing function $\mathbf{g} = \mathbf{f}^{-1}$, only using observations of $\mathbf{X}$. The following definition formally states the connection between SCM and nonlinear ICA:

Definition 1. An SCM (1) is **analyzable** if there exists a differentiable and invertible[3] function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$, such that $\mathbf{X} = \mathbf{f}(\mathbf{E})$.

Obviously, an analyzable SCM is a special case of nonlinear ICA's generative model, with particular structure between the variables. For example, in bivariate SCM (2), let $f_3(E_1, E_2) = f_2(f_1(E_1), E_2)$ and $\mathbf{f} = (f_1, f_3)$, the SCM can be written as $(X_1, X_2) = \mathbf{f}(E_1, E_2)$. Now if $\mathbf{f}$ is differentiable and invertible on $\mathbf{R}^2$, the SCM is analyzable.

For analyzable SCM, if we can solve the corresponding nonlinear ICA problem, we obtain the hidden variables $\mathbf{E} = \mathbf{g}(\mathbf{X})$. In bivariate case, given E_1 and E_2, under causal Markov and faithfulness assumptions (Spirtes and Zhang, 2016), we can conclude:

$$\begin{aligned} &X_1 \to X_2 \text{ if } X_1 \perp\!\!\!\perp E_2, \\ &X_2 \to X_1 \text{ if } X_2 \perp\!\!\!\perp E_1 \end{aligned} \tag{4}$$

This criteria was exploited by many classical methods, e.g. LiNGAM and ANM, and can be easily understood as the independence of noise and cause.

2.4 Learning Shared Mechanism by TCL

Lately, Time-Contrastive Learning (TCL) (Hyvärinen and Morioka, 2016) provided the first general identifiability result for nonlinear ICA. The method depends on learning the different distributions of time series through time, and hence the name. After artificially dividing time series into segments, it trains a classification task to tell which segment each sample point belongs to. As indicated in Hyvärinen et al. (2019), the segment index could be treated as an *auxiliary variable* $\mathbf{u}$, which only needs to satisfy that hidden components $\mathbf{Z}$ are independent of each other given $\mathbf{u}$.

With the intuition that different causal pairs in real world can share the same mechanism, we can derive a method for learning the shared mechanism by TCL. We just need to feed TCL with pairs sharing mechanism, and replace segment index with pair index as auxiliary variable. Here, we restate the theory under our own setting:

Theorem 1 (Hyvärinen and Morioka (2016))**.** *Assume the following:*

A1. We observe causal pairs $\mathcal{X}(P) \coloneqq \{\mathbf{X}_p\}_{p=1}^P$ *which satisfy the same analyzable SCM* $\mathbf{X}_p = \mathbf{f}(\mathbf{E}_p)$*, and the hidden variables* $E_{i,p}, i = 1, 2$ *are of exponential family distribution* $p_{E_{i,p}}(e) = \exp[T_i(e)\eta_i(p) - A(\eta_i(p))]$ *where* $T_i(e)$ *is the sufficient statistic.*

A2. The matrix $\mathbf{L}$*, with elements* $[\mathbf{L}]_{p,i} = \eta_i(p) - \eta_i(1), p = 1, ..., P$*,* $i = 1, 2$*, has full column rank 2.*

A3. We train a feature extractor $\mathbf{h} : \mathbf{R}^2 \to \mathbf{R}^2$ *with universal approximation capability, followed by a final softmax layer to classify all sample points of the pairs, with pair index used as class label.*

Then, in the limit of infinite data, for each p*,* $\mathbf{T}(\mathbf{E}_p) \coloneqq (T_1(E_{1,p}), T_2(E_{2,p}))^T = \mathbf{A}\mathbf{h}(\mathbf{X}_p; \boldsymbol{\theta}) + \mathbf{b}$ *where* $\mathbf{A}, \mathbf{b}$ *are unknown constants, and* $\mathbf{A}$ *is invertible.*

In practice, a multilayer perceptron (MLP) is used as the feature extractor. The theorem implies that the identification (recovery) of $\mathbf{T}(\mathbf{E}_\mathrm{p})$ can be achieved by first performing TCL, and then linear ICA on $\mathbf{h}(\mathbf{X}_\mathrm{p})$. Denoting the composition of $\mathbf{h}$ and linear ICA as $\mathbf{hICA}$, we have $\mathbf{T}(\mathbf{E}_\mathrm{p}) = \mathbf{hICA}(\mathbf{X}_\mathrm{p})$. In this sense, we say that $\mathbf{h}$ is successfully learned and the nonlinear ICA of $\mathbf{X}_p$ is *realized* by $\mathbf{hICA}$. Here we learn the shared mechanism $\mathbf{f}$ (or precisely its inverse) as part of $\mathbf{h}$, along with $\mathbf{T}$.

While we can recover only the sufficient statistics $T_i(E_{i,p})$, not $E_{i,p}$, they are sufficient for building a method for cause-effect inference; $T_i(E_{i,p})$ generally has the same independence relationships with other variables as $E_{i,p}$. In practice, under the assumption that there exist direct causal effects, we can just compare values of an independence measure, as we will detail in Section 3.

Unlike the time contrast exploited in the original TCL, the contrast here is among the pairs. But, by convention, we will still use the word *"TCL"* when referring to the method trained on causal pairs, which *not* necessarily satisfy A1 and A2 of Theorem 1. By a slight abuse of terminology, the produced $\mathbf{h}$, which may *not* be successfully learned, is also called TCL in this paper.

3 THEORETICAL RESULTS

3.1 Separation of Training and Testing

It should be clear from Section 2 that we want to learn casual mechanism via TCL. However, readers might notice that, to successfully learn TCL, we at least need

[3]This does not imply such a strong restriction as it would seem. See Supplementary Material.

to know that the pairs indeed share causal mechanism! To address the above dilemma, our idea is to learn causal mechanism from some training pairs that we have good causal knowledge (e.g. we might know their SCMs and causal directions), and then predict the causal directions for unseen pairs. The following corollary of Theorem 1 makes this separation possible:

Corollary 1 (**Transferability** of TCL)**.** *Assume:*

A1. Pairs $\mathcal{X}^{tr}(P)$ *satisfy A1 and A2 of Theorem 1.*

A2. A pair $\mathbf{X}^{te}$ *satisfy A1 of Theorem 1, with the same* $\mathbf{f}$ *and* $\mathbf{T}$ *as* $\mathcal{X}^{tr}(P)$*, but different parameter* η_i*.*

A3. Let $\mathcal{R}_X$ *denote the support of a random variable* X*. We have* $\mathcal{R}_{E_i^{te}} \subseteq \cup_{p=1}^{P} \mathcal{R}_{E_{i,p}^{tr}}, i = 1, 2.$

A4. We learn a feature extractor $\mathbf{h}$ *on* $\mathcal{X}^{tr}(P)$ *as in A3 of Theorem 1 and have* $\mathbf{T}(\mathbf{E}_p^{tr}) = \mathbf{Ah}(\mathbf{X}_p^{tr}) + \mathbf{b}$*.*

Then, we have $\mathbf{T}(\mathbf{E}^{te}) = \mathbf{Ah}(\mathbf{X}^{te}) + \mathbf{b} = \mathbf{hICA}(\mathbf{X}^{te})$*.*

Intuitively, after we successfully learned TCL $\mathbf{h}$, we can re-use it to analyze other unseen pairs that have the same SCM and sufficient statistics as the training pairs. We should note that, as in transfer learning, training and testing pairs do *not* have the same distribution, and hence the name of this corollary. From now on, we will also refer to the learning of TCL and analysis of new pairs on it as training and testing, respectively.

3.2 Inference Methods and Identifiability

We first present a general procedure (Algorithm 1) as the common basis, before detailing the two inference rules (`inferule`) with their identifiability results (and also $Direction^{tr}$ and `align`). In the following, $\alpha_0 = (1, 2)$ and $\alpha_1 = (2, 1)$ denotes the two permutations on $\{1, 2\}$, and $\alpha_i(\mathbf{X}) \coloneqq (X_{\alpha_i(1)}, X_{\alpha_i(2)})$.

Algorithm 1: Inferring causal direction

input : $\mathcal{X}^{tr}(P)$, $\mathbf{X}^{te}$, $Direction^{tr}$, `align`, `inferule`
output: $Cause^{te}$

1 Align training set, exploiting $Direction^{tr}$:
  $\mathcal{X}^{al}(P) = \texttt{align}(\mathcal{X}^{tr}(P), Direction^{tr})$
2 Learn TCL $\mathbf{h}$ on $\mathcal{X}^{al}(P)$
3 **foreach** $\alpha = \alpha_0, \alpha_1$ **do**
4 $(C_1, C_2)_\alpha^T = \mathbf{hICA}(\alpha_i(\mathbf{X}^{te}))$
5 Run inference rule:
  $Cause^{te} = \texttt{inferule}(\mathbf{C}_{\alpha_0}, \mathbf{C}_{\alpha_1}, \mathbf{X}^{\text{te}})$

With $\mathbf{T}(\mathbf{E}^{\text{te}})$ recovered, we can find ways to infer a causal direction for $\mathbf{X}^{te}$. To find the asymmetry between the two possible causal directions, we use the fact that, when testing, if we *flip* input direction to **hICA** and try nonlinear ICA for each (line 3,4 Algorithm 1), there will be one and only one trial that is realized by the **hICA**. This information will be exploited in `inferule` (line 5 Algorithm 1).

A remaining issue is that, to apply Theorem 1 and in turn Corollary 1, we need to at least partially know the directions of training pairs. More precisely, $\mathcal{X}^{tr}(P)$ must be *aligned*, as in the following definition. (This is implied by $\forall p\,(\mathbf{X}_{\text{p}} = \mathbf{f}(\mathbf{E}_{\text{p}}))$ in A1 of Theorem 1.)

Definition 2. Causal pairs $\mathcal{X}^{al}(P) \coloneqq \{\mathbf{X}_p\}_{p=1}^{P}$ are **aligned** if $\forall p\,(X_{1,p} \to X_{2,p})$ or $\forall p\,(X_{2,p} \to X_{1,p})$.

In the first inference rule, it is assumed that we know the causal direction for each of the training pairs so that they can be trivially aligned. For a test pair, a realized (successful) nonlinear ICA among the two trials should output independent components, and this in turn tells us the direction of the pair, because we know which input of $\mathbf{h}$ corresponds to the cause. This leads to the following theorem:

Theorem 2 (**Identifiability** by independence of hidden components)**.** *In Algorithm 1, let:*

$Direction^{tr} = \{c_p\}_{p=1}^{P}$ *where* $c_p \in \{1, 2\}$ *is the cause index:* $X_{c_p,p}^{tr} \to X_{3-c_p,p}^{tr}$*,*

$\texttt{align} = \{X_{c_p,p}^{tr}, X_{3-c_p,p}^{tr}\}_{p=1}^{P}$*,*

$\texttt{inferule} = \alpha^*(1), \alpha^* = \underset{\alpha \in \{\alpha_0, \alpha_1\}}{\text{argmax}}\ \texttt{dindep}(\mathbf{C}_\alpha)$ *where* `dindep` *measures degree of independence.*

And assume:

A1. Causal Markov assumption and causal faithfulness assumption hold for data generating SCMs and analysis procedure except[4] *for a realized nonlinear ICA.*

A2. $\mathcal{X}^{tr}(P)$ *and* $\mathbf{X}^{te}$ *satisfy A1–A3 of Corollary 1.*

Then, the `inferule` *defined above (*`inferule1` *afterwards) identifies the true cause variable.*

The second inference rule only assumes we know how to align the training pairs. In fact, under certain practical scenarios, we know the training pairs *are* aligned; for example, 1) pairs from multiple environments (per environment per pair), as in many domain adaptation problems and in Monti et al. (2019), and 2) pairs from stratified sampling (per sample per pair).

The `inferule` determines the realized trial and identifies causal directions, *without* the directions of training pairs. We examine the independence of the pair $\{T_j(E_j^{te}),\ X_i^{te}\}$, as in the relation (4). Note, however, that as described in Monti et al. (2019), the outputs of a realized nonlinear ICA are equivalent to hidden variables only up to a permutation, i.e. $\mathbf{T}(\mathbf{E}^{\text{te}}) = (C_{\alpha(1)}, C_{\alpha(2)})^T$, with α unknown. This requires us to evaluate the degree of independence for

[4] See Supplementary Materials on this.

four pairs at each trial, as in the following theorem:

Theorem 3 (**Identifiability** by independence of noise and cause)**.** *In Algorithm 1, let:*

$Direction^{tr} = \{i_p\}_{p=1}^{P}$ *where* $i_p \in \{1,2\}$ *such that* $\forall p\,(X_{i_p,p} \to X_{3-i_p,p})$ *or* $\forall p\,(X_{3-i_p,p} \to X_{i_p,p})$

$\texttt{align} = \{X^{tr}_{i_p,p}, X^{tr}_{3-i_p,p}\}_{p=1}^{P}$,

$\texttt{inferule} = i^*, (i^*, ., .) = \underset{i,j,\alpha}{\operatorname{argmax}}\, \texttt{dindep}(X_i^{te}, C_{j,\alpha})$.

And assume the same as Theorem 2.

Then, the `inferule` *defined above (*`inferule2` *afterwards) identifies true cause variable.*

Since we can use the causal directions to recover an aligned training set, so in Theorem 2, letting `inferule` = `inferule2`, the true causal index can also be identified. However, as we will see in the experiments, `inferule1` will outperform `inferule2` if the former is applicable in practice.

Finally, we will employ distance correlation (dCor) (Székely et al., 2007) as our main choice of `dindep` (See Supplementary Material for details).

3.3 Structural MLP

We discuss an MLP structure to improve TCL's performance on bivariate analyzable SCMs. We first study the form of the inverse SCM, since this is what the MLP should learn.

Proposition 1 (Inverse of bivariate analyzable SCM)**.** *For any analyzable SCM as shown in* (2)*, denote the whole system* $\mathbf{X} = \mathbf{f}(\mathbf{E})$*, if the Jacobian matrix of* $\mathbf{f}$ *is invertible, then* f_1 *is invertible.*

Denote $g_1 = f_1^{-1}$, then $E_1 = g_1(X_1)$. And we have $E_2 = g_2(X_1, X_2)$ in general. This implies the inverse SCM has the graph as shown in Figure 3 (left):

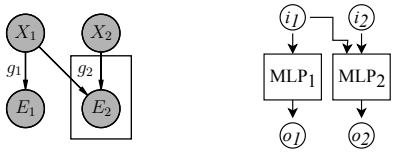

Figure 3: Inverse bivariate analyzable SCM (left) and the indicated MLP structure (right).

Building an MLP for TCL with this asymmetric structure will help TCL learn the inverse SCM. This can be easily implemented as shown in Figure 3 (right): we build an MLP with one output node for g_1 and g_2 respectively, and then concatenate the outputs together. Please see Supplementary Material for caveats on building and using this asymmetric structure.

4 ASSEMBLING CAUSAL MOSAIC

In the following, we will refer to training pairs that satisfy A1 (same SCM and exponential family) and A2 (enough variability among parameters) of Theorem 1 as *tessera pairs*, because they form the small portion of causal pairs that can be easily modeled together, and thus a small block of the whole mosaic. Also, we will refer to a TCL learned on tessera pairs as a *tessera*.

We have so far assumed that we have tessera pairs, under the ideal situation that we have well-studied systems. However, for many real world applications, it is unlikely that most training pairs amount to tessera pairs. Our idea for handling real world problems is to train many TCLs on random selections of pairs, and then choose from these TCLs the (imperfect) tesserae that are trained on *approximate* tessera pairs, in the sense that they have similar SCMs and are approximately in the same family. We further develop an ensemble method to effectively exploit imperfect tesserae.

In this section, Let S be the set of all labeled causal pairs we have at hand, and c_s be the true cause index for $s \in S$.

4.1 Preparing Materials

As in Algorithm 2, by training a large number (N) of TCLs on randomly chosen pairs, we hope some of these TCLs amount to tesserae. To ensure TCL is trained properly on each set of pairs, we train MLP M times with different hyperparameters (See experiment for details).

Algorithm 2: Random training of TCLs

input : S, M, N
output: $\{(\mathbf{h}_n, T_n)\}_{n=1}^{N}$
1 **foreach** *n in 1,...,N* **do**
2 Randomly choose training pairs $T_n \subset S$
3 Split the sample points of each training pair by half, and build training set Tr and testing set Te
4 **foreach** *m in 1,...,M* **do**
5 Randomly choose a set of hyperparameters and train TCL on Tr
6 Evaluate classification accuracy ($Cacc_m$) for pair index on Te.
7 Use the trained TCL with the highest $Cacc_m$ for this set of training pair, denote it $\mathbf{h}_n$

4.2 Choosing Tesserae

Because our goal is to infer causal directions, we choose TCLs that perform well on this task. First, we can use each TCL to infer the causal directions of its own training pairs (Algorithm 3, line 2,3), and choose TCLs that produce accuracy higher than a threshold $ThreT$.

Second, for each TCL, we also input unseen validation pairs and infer their directions, and we choose TCLs that produce accuracy higher than $ThreV$. The good training accuracy indicates the success of training and TCL indeed learned to infer causal directions. The good validation accuracy shows that the learning generalizes to unseen pairs.

To efficiently use S for training and validation, and still be able to test on all the pairs in S, we use the idea of leave-one-out cross validation (LOOCV). That is, each pair l not used in training a TCL is left out once when validating that TCL (line 5,6). As we can see, every pair in S is not used as a training pair or validating pair for its tessera (line 10,11). On the other hand, in training (T_n) and validation ($(S \setminus T_n) \setminus \{l\}$), every trained TCL exploits all the pairs except the left out one l.

Algorithm 3: Selecting TCLs

input : S, $ThreT$, $ThreV$, $\{(\mathbf{h}_n, T_n)\}_{n=1}^N$
output: $\{TSR_s : s \in S\}$
1 **foreach** *n in 1,...,N* **do**
  // Training accuracy $Tacc_n$ for $\mathbf{h}_n$ on T_n
2 **foreach** *t in* T_n **do**
3 Use $\mathbf{hICA}_n$, run line 3–5 of Algorithm 1 on t, get inferred direction $\hat{c}_t$
4 $Tacc_n = |\{t : \hat{c}_t = c_t\}|/|T_n|$
  // LOOCV
5 **foreach** *l in* $S \setminus T_n$ **do**
6 As line 2–4, get validation accuracy for $\mathbf{h}_n$ on $(S \setminus T_n) \setminus \{l\}$, denote it as $Vacc_n(l)$
  // Select TCLs by accuracy thresholds
7 **foreach** *s in S* **do**
8 Initialize tessera index set for s: $TSR_s = \varnothing$
9 **foreach** *n in 1,...,N* **do**
10 **if** $s \notin T_n$ **and** $Tacc_n > ThreT$ **and** $Vacc_n(s) > ThreV$ **then**
11 Add n to TSR_s

By the identifiability theorems, if TCL $\mathbf{h}_n$ has high training accuracy, it is likely that the training pairs T_n are approximate tessera pairs (required by A1 & A2 of Theorem 1). Similarly, if $\mathbf{h}_n$ gives high validation accuracy, the evidence for tessera pairs T_n is strengthened (required by A1 of Corollary 1), and further it is likely that pairs T_n are similar to many of pairs in $S \setminus T_n$ (required by A2 of Corollary 1).

4.3 From Tesserae to Causal Mosaic

We employ an ensemble method for making effective use of each imperfect tessera, and construct a whole piece of mosaic, in the same way as we will obtain a strong classifier from weaker ones by ensemble methods. Put simply, for each testing pair, ensemble method will take the causal direction predicted by tesserae, and produce a final, weighted average. We introduce two levels of weighting as follows.

First, as Algorithm 4, line 3, we weight a TCL $\mathbf{h}_n$ by the average $\texttt{dindep}(\mathbf{hICA}_n(.))$ for the training pairs T_n. This is to address the problem that, even if we have selected TCLs as in Algorithm 3, it is very possible that the chosen tesserae would not be perfect, e.g., the mechanisms of training pairs are not exactly the same. Thus, we use this weight to measure how well T_n fit together (by Theorem 1, if we get more independent components, A1 & A2 are more likely to hold), and in turn how likely the causal direction will be correctly inferred if we use this $\mathbf{h}_n$.

Second, we weight by the $\texttt{dindep}(\mathbf{hICA}_n(.))$ for a particular testing pair s. Again, even if w_n is large, it is possible that s and T_n do not satisfy A2 of Corollary 1, so we need to weight each tessera for *each* testing pair. Similarly to the reasoning for w_n, if we get independent components for s, A2 of Corollary 1 is likely to hold. Note that, as in Algorithm 1, in theory only realized nonlinear ICA outputs independent components, so we weight by the larger `dindep` of the two trials (line 4–7). We multiply the two weights as the final pair-specified weight.

Algorithm 4: Ensemble method

input : S, $\{TSR_s : s \in S\}$, $\{(\mathbf{h}_n, T_n)\}_{n=1}^N$
output: $\{Direction_s : s \in S\}$
1 **foreach** *s in S* **do**
2 **foreach** *n in* TSR_s **do**
3 $w_n = \sum_{t \in T_n} (\texttt{dindep}(\mathbf{hICA}_n(t)))/|T_n|$
4 **foreach** $i = 0, 1$ **do**
5 $\mathbf{C}_{\alpha_i} = \mathbf{hICA}_n(\alpha_i(s))$
6 $w_{ns,i+1} = \texttt{dindep}(\mathbf{C}_{\alpha_i})$
7 $w_{ns} = \max(w_{ns,1}, w_{ns,2})$
8 $\hat{c}_s = \texttt{inferule}(\mathbf{C}_{\alpha_0}, \mathbf{C}_{\alpha_1}, s)$
9 $Direction_{ns} = 1$ if $\hat{c}_s = 1$, -1 if $\hat{c}_s = 2$
10 Calculate weighted prediction $Score_s = \sum_{n \in TSR_s} w_n w_{ns} Direction_{ns}$
11 $Direction_s = \begin{cases} X_1 \to X_2 & Score_s > 0 \\ X_2 \to X_1 & Score_s < 0 \\ ? & Score_s = 0 \end{cases}$

5 EXPERIMENTS

5.1 Artificial Data

We compare NonSENS to variations of our method with different inference rules, independence measures, and MLP types on artificial data. To see the comparisons with other recent methods on similar artificial data, we refer readers to Monti et al. (2019).

Multi-environment setting This is the setting under which NonSENS works. Mathematically, our tessera pairs $\{\mathbf{X}_p^{tr}\}$ are equivalent to the samples $\mathcal{X}^{en} := \{\mathbf{X}_p^{en}\}$ of a *same* causal system under P different "environments" in their interpretation. That is, they define different environments by different param-

eter $\boldsymbol{\eta}$ of hidden variables, and $\forall p(\mathbf{X}_p^{en} = \mathbf{f}(\mathbf{E}_p^{en}))$ is by definition satisfied. Moreover, there is no separate testing pairs here. Our goal is to distinguish between two possibilities, $\forall p\,(X_{1,p}^{en} \to X_{2,p}^{en})$ or $\forall p\,(X_{2,p}^{en} \to X_{1,p}^{en})$, for $\mathcal{X}^{en}$ themselves (note the pairs (environments) are *aligned*), rather than 2^P possibilities for individual pairs $\mathcal{X}(P)$.

Our Algorithm 1 can reduce to this setting, as shown in Algorithm 5. Both training and testing pairs are $\mathcal{X}^{en}$ themselves. Note that $Direction^{tr}$, `align` and the input permutation (Algorithm 1, line 3,4) are not needed, since $\mathcal{X}^{en}$ is already aligned. We apply a simplified version of `inferule2` to infer direction for each environment without input permutation, but still need to deal with the output permutation.

Finally, we use majority voting to combine the results of all environments and give the final decision, and this is an important difference between our method and NonSENS under this setting. NonSENS treats the samples of environments as coming from a mixture, runs `dindep` on pooled sample and output, and gives $c^{en} = i^*, (i^*, j^*) = \text{argmax}_{i,j}\, \texttt{dindep}(\{X_{i,p}^{en}\}, \{C_{j,p}\})$[5]. In practice, as we will see, majority voting often outperforms NonSENS since it uses information from each environment and thus is more robust.

Algorithm 5: Algorithm 1 on multi-env. setting

input : $\mathcal{X}^{en}$
output: c^{en}
1 Learn TCL $\mathbf{h}$ on $\mathcal{X}^{en}$
2 $\mathcal{C} = \mathbf{hICA}(\mathcal{X}^{en})$
3 **foreach** $\mathbf{X}_p^{en}$ *in* $\mathcal{X}^{en}$, $\mathbf{C}_p$ *in* $\mathcal{C}$ **do**
4 $\quad c_p = i^*, (i^*, j^*) = \underset{i,j}{\text{argmax}}\, \texttt{dindep}(X_{i,p}^{en}, C_{j,p})$
// Majority voting
5 $c^{en} = \underset{i}{\text{argmax}}\, |\{c_p : c_p = i\}|$

Multi-pair setting If we know the *directions* of training pairs, we separate training and testing, and both Theorem 2 (`inferule1`) and Theorem 3 (`inferule2`) can apply. Here, we infer the direction for each individual testing pair. NonSENS cannot apply here, so we compare different variations of our method. We name this multi-pair setting, to contrast the multi-environment setting, although the main difference is the direction information of training pairs (our method *can* also infer for each environment as in Algorithm 5, line 3,4).

Data generation As in Hyvärinen and Morioka (2016) and Monti et al. (2019), we use 5-layer randomly initialized MLPs as mixing functions, with leaky ReLU activation and 2 units in each layer to ensure invertibility. To simulate the independent relationships of a direct causal graph, we use a lower-triangle weight matrix for each layer of the MLP. We use Laplace distribution for both hidden components, and their variance parameters are i.i.d. generated across different pairs. Multi-environment setting can be easily simulated by aligning all the pairs and then perform nonlinear ICA.

We generate 100 mixing functions and same number of training/testing pairs for each mixing function. To observe how the pair number affect results, we try 5 different number ranging from 10 to 50. Please see Supplementary Material for more details.

Hyperparameters To make fair comparisons, for both our method and NonSENS, we keep all the hyperparameters the same, including the parameters for training and independent tests. Please see Supplementary Material for details.

Assuming direct causal effect Our method and NonSENS[6] formally requires direct causal effects exist between pairs, and this is our main experiment setting. Please see Supplementary Material for the experiment without this assumption.

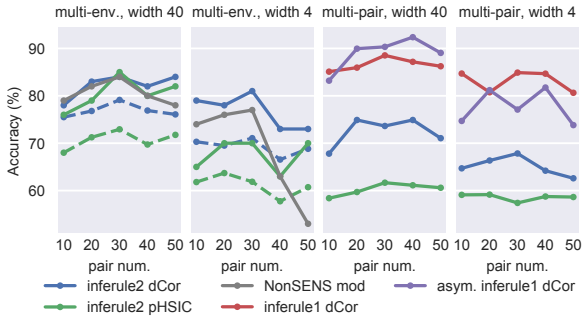

Figure 4: Performance assuming direct causal effect. "width" means MLP width. In the legend, "dCor/pHSIC" indicates the independence measure, and "asym." means asymmetric MLP in TCL. Dashed lines are intended to show transferability of TCL, see Supplementary Material.

As shown in Figure 4, in multi-environment setting, our method outperforms NonSENS, particularly when the pair number is large. The decreasing performance of NonSENS is consistent with the results when not assuming pure causal effects and is due to the unwanted dependence between estimated noise and the cause, as explained in detail in Supplementary Material.

In multi-pair setting, `inferule1` is applicable and performs much better than `inferule2`. The main reason is that the independence between two output components is much easier to realize than the independence

[5]Originally, NonSENS uses independence tests with a threshold. We write it here using `dindep` for easy comparison, because we will use this modified rule for NonSENS in experiment.

[6]We cannot reproduce the likelihood ratio based NonSENS proposed for this setting. Instead, we use a slightly modified version of NonSENS originally proposed for may-not-direct-causal setting, see the previous footnote.

between estimated noise and observed cause. And this is in turn because of the direct dependence between observed variables and outputs (see Figure 1 in Supplementary Material). Note that Theorem 2 required known causal directions of training pairs, and thus cannot be used in multi-environment setting.

Moreover, when the MLP width is 40, `inferule1` achieves near-optimal results when applied with asymmetry MLP. This is also the best result we have obtained with artificial data. While the asymmetry MLP with width 4 performs worse than the fully-connected one, this is due to the limited fitting capacity (see Supplementary Material for details).

When inferring by Theorem 3, we try both dCor and the p-value of HSIC (Gretton et al., 2005) as `dindep`. dCor constantly outperforms HSIC (See Supplementary Material for details).

5.2 Real World Dataset

Tuebingen cause-effect pairs (TCEP) dataset (Mooij et al. (2016), dataset version December 2017) is a commonly used benchmark for cause-effect inference tasks. Causal Mosaic can be suitably applied here because of the very diverse scenarios of the pairs. Each pair is assigned a weight in order to account for the possible correlation with other pairs that are selected from the same multivariate scenario. Currently, the dataset contains 108 real-world cause-effect pairs with true causal directions labeled by human experts. We exclude 6 multivariate pairs in our evaluation.

Implementation We use Theorem 2 with asymmetric MLP since it already shows much better results on artificial data. Unlike on artificial data with Laplace hidden variables, we use maxout activation for the output layer. Since the sample sizes of TCEP pairs range wildly from a hundred to several thousands, we fix this imbalance in classification by under-sampling using imbalanced-learn package (Lemaître et al., 2017). When implementing Algorithm 4 line 10, we use a simplified version $Score_s = \sum_{n \in TSR_s}(w_{ns,1} - w_{ns,2})$, since this works the best. See Supplementary Material for details.

Hyperparameters We train TCL on 300 (N) sets of randomly picked pairs, which are of size ranging from 4 to 32. For selecting TCLs, we randomly search 100 pairs of accuracy thresholds $(ThreT, ThreV)$ in $[65\%, 75\%]^2$ and rule out too large thresholds that give 0 or only 1 tessera for more than 10 TCEP pairs. We train 10 (M) TCLs on each pair set and choose the best, and the following hyperparameters are randomly searched from uniform distributions: depth and width of MLP, learning rate, decay factor, max step (decay step is 10% of max step), momentum, and batch size. Among them, the depth of MLP larger than 10 might lead to divergence in training, but the ranges of other parameters seem to have few impacts if we do not use some extreme values. To save training time, we change the ranges of MLP width and max. step according to training pair number (small width and step for small pair number).

Table 1: Accuracy (%) on TCEP. "A/B" means with/without applying pair weight.

ANM	IGCI	RECI	NCC	OURS
52.5/52.0	60.4/60.8	70.5/62.8	51.8/56.9	$\mathbf{81.5}_{\pm 4.1}/\mathbf{83.3}_{\pm 5.2}$

We compare our method to ANM, IGCI, RECI and NCC, using implementations from CDT package (Kalainathan and Goudet, 2019). The results are shown in Table 1. We report the *median* and std-error of accuracies of our method calculated on all the 83 pairs of thresholds. And this already shows state-of-the-art performance. The best result on all thresholds is 86.3% without pair weight and might overfit TCEP dataset. For NCC, we infer each pair by training the method on rest of the pairs. The accuracy is much worse than the reported 79% in Lopez-Paz et al. (2017), the most possible reason is that NCC requires much more training data (320,000 artificial pairs in the original paper). The performance of ANM is worse than reported in Mooij et al. (2016), possibly because of the different implementation of independence test.

6 CONCLUSION

In this work, we proposed a highly flexible cause-effect inference method that learns a mixture of general nonlinear causal models, with proof of identifiability. We exploited TCL to extract the common mechanism shared by different causal pairs, and transferred the causal knowledge to unseen pairs. More specifically, our method learns how to distinguish cause from effect, from some training pairs, and predicts the causal direction on testing pairs. We gave two inference rules with identifiability proofs and an ensemble framework that works on real world cause-effect pairs with limited labeled causal directions. We compared our method to recent methods on artificial and real world benchmark datasets, and it showed state-of-the-art results.

Hence, we justified the "mosaic" perspective of causal discovery, which proposes to learn causality piecemeal, and then build a whole picture by the pieces. Here, shared mechanism learned by TCL forms a tessera of the whole causal mosaic, and many tesserae are learned and further combined into a whole picture by ensemble method. We believe this new perspective would promote other novel methods for bivariate and also more general causal discovery problems.

Acknowledgments

This work has been supported in part by JSPS KAKENHI 18K19793. We thank Ricardo Pio Monti for providing core code for NonSENS and discussions on experimental results. We thank Diviyan Kalainathan for discussions on the CDT package on GitHub.

References

Patrick Blöbaum, Dominik Janzing, Takashi Washio, Shohei Shimizu, and Bernhard Schölkopf. Cause-effect inference by comparing regression errors. In *International Conference on Artificial Intelligence and Statistics*, pages 900–909, 2018.

Kenneth A Bollen. *Structural equations with latent variables*. New York John Wiley and Sons, 1989.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–552, 2002.

Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 39–80. Springer, 2018.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696, 2009.

Shoubo Hu, Zhitang Chen, Vahid Partovi Nia, Laiwan Chan, and Yanhui Geng. Causal inference and mechanism clustering of a mixture of additive noise models. In *Advances in Neural Information Processing Systems*, pages 5206–5216, 2018.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

Aapo Hyvärinen and Kun Zhang. Nonlinear functional causal models for distinguishing cause from effect. In *Statistics and Causality: Methods for Applied Empirical Research*, pages 185–201. John Wiley, 2016.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868, 2019.

Dominik Janzing and Bernhard Scholkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10): 5168–5194, 2010.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv:1903.02278*, 2019.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365`.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. In *International Conference on Machine Learning*, pages 1452–1461, 2015.

David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6979–6987, 2017.

Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. Causal inference via kernel deviance measures. In *Advances in Neural Information Processing Systems*, pages 6986–6994, 2018.

Ricardo Pio Monti, Kun Zhang, and Aapo Hyvärinen. Causal discovery with general non-linear relationships using non-linear ICA. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019*, page 45, 2019.

Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

Judea Pearl. *Causality: models, reasoning and inference*. Cambridge University Press, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT Press, 2017.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, page 3. SpringerOpen, 2016.

Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. 2000.

Gábor J Székely, Maria L Rizzo, Nail K Bakirov, et al. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 647–655. AUAI Press, 2009.