Continual Casual Representation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unsupervised identification of disentangled representations remains a challenging problem. Recent progress in nonlinear Independent Component Analysis (ICA) provides a promising causal representation learning framework by separating latent sources from observable nonlinear mixtures. However, its identifiability hinges on the incorporation of side information, such as time or domain indexes, which are challenging to obtain adequately offline in real-world scenarios. In this paper, we develop a novel approach for nonlinear ICA that effectively accommodates continually arriving domains. We first theoretically demonstrate that model identifiability escalates from subspace to component-wise identifiability as new domains are involved. It motivates us to maintain prior knowledge and progressively refine it using new arriving domains. Upon observing a new domain, our approach optimizes the model by satisfying two objectives: (1) reconstructing the observations within the current domain, and (2) preserving the reconstruction capabilities for prior domains through gradient constraints. Experiments demonstrate that our method achieves performance comparable to nonlinear ICA methods trained jointly on multiple offline domains, demonstrating its practical applicability in continual learning scenarios.

1 Introduction

Causal representation learning aims at recovering high-level semantic variables from low-level observations and their casual relations. Compared with current deep learning models which are trained as black-box functions, it is more explainable and generalizable by the identification of the underlying causal generation process. A widely recognized theoretical finding suggests that estimating those latent variables in a fully unsupervised way is inherently challenging without further assumptions [12]. Among multiple efforts towards this problem [35, 27], nonlinear ICA attracts a lot of attention by providing a promising framework and demonstrating an identification guarantee.

Nonlinear ICA focuses on recovering independent latent variables from their nonlinear mixtures. Denote an observed n-dimensional vector by $\mathbf{x}$, which is generated by a number of independent latent variables $\mathbf{z}$ through an arbitrary invertible mixing function g as $\mathbf{x} = g(\mathbf{z})$. The objective of nonlinear ICA is to reconstruct the latent variables $\mathbf{z}$ by discovering the inverse function g^{-1} based on the observation $\mathbf{x}$ only in an unsupervised manner. Apparently, without additional constraints, we can never find out a meaningful solution. More rigorously, the identifiability of nonlinear ICA cannot be guaranteed when only relying on independence assumption [12].

To address this problem, existing works focus on adding constraints on the mixing function [6, 38, 1], or most popularly, benefiting from the non-stationary of source data [10, 11, 13, 15] to advance the identifiability. By introducing auxiliary variable $\mathbf{u}$ and assuming the non-i.i.d sources are conditionally independent given $\mathbf{u}$, the latent variables can be estimated up to component-wise identifiability. Although current research on nonlinear ICA has made great progress, it still relies on observing sufficient domains simultaneously, which limits its application to scenarios where changing domains may arrive sequentially. Specifically, the model trained with sequential arrival of domains without making adjustments is equivalent to the scenario where only one domain is observed. Consequently, the model becomes unidentifiable.

In this paper, we present a novel approach to learning causal representation in continually arriving domains. Distinct from traditional continual classification tasks, continual causal representation learning (CCRL) requires that the model leverages the changes in distribution across varying domains. This implies that the problem cannot be segregated into discrete local learning tasks, such as learning causal representation within individual domains and subsequently fusing them. In this context, we conduct a theoretical examination of the relationships between model identification and the number of observed domains. Our findings suggest that model identifiability escalates as additional domains are incorporated. In particular, subspace identification can be achieved with $n + 1$ domains, while component-wise identification necessitates $2n + 1$ domains or more. This indicates that when the domain count is inadequate ($n + 1$), we can only identify the manifold spanned by a subset of latent variables. However, by utilizing the new side information in the distribution change of arriving domains, we can further disentangle this subset.

This discovery motivates us to develop a method that retains prior knowledge and refines it using information derived from incoming domains, a process reminiscent of human learning mechanisms. To realize causal representation learning, we employ two objectives: (1) the reconstruction of observations within the current domain, and (2) the preservation of reconstruction capabilities for preceding domains via gradient constraints. To accomplish these goals, we apply Gradient Episodic Memory (GEM) [19] to constrain the model's gradients. GEM aligns the gradients of the new domain with those of prior domains by eliminating factors within the current domain that are detrimental to previous domains. Through empirical evaluations, we demonstrate that our continual approach delivers performance on par with nonlinear ICA techniques trained jointly across multiple offline domains. Importantly, the guarantee of identifiability persists even when incoming domains do not introduce substantial changes for partial variables. Furthermore, we demonstrate that the sequential order of domains can enhance the identification process in causal representation learning.

2 Related Work

Causal representation learning. Beyond conventional representation learning, causal representation learning aims to identify the underlying causal generation process and recover the latent causal variables. There are pieces of work aiming towards this goal. For example, it has been demonstrated in previous studies that latent variables can be identified in linear-Gaussian models by utilizing the vanishing Tetrad conditions [28], as well as the more general concept of t-separation [27]. Additionally, the Generalized Independent Noise(GIN) condition tried to identify a linear non-Gaussian causal graph [35]. However, all of these methods are constrained to the linear case while nonlinear ICA provides a promising framework that learns identifiable latent causal representations based on their non-linear mixture. However, the identifiability of nonlinear ICA has proven to be a challenging task [12], which always requires further assumptions as auxiliary information, such as temporal structures [29], non-stationarities [10, 11], or a general form as auxiliary variable [13]. These methods indicate that sufficient domains (changes) are crucial for ensuring the identifiability of nonlinear ICA. In this paper, we consider the scenario that changing domains may arrive not simultaneously but sequentially or even not adequately.

Continual learning. In conventional machine learning tasks, the model is trained on a dedicated dataset for a specific task, then tested on a hold-out dataset drawn from the same distribution. However, this assumption may contradict some real-world scenarios, where the data distribution varies over time. It motivates researchers to explore continual learning to enable an artificial intelligence system to learn continuously over time from a stream of data, tasks, or experiences without losing its proficiency in the ones it has already learned. The most common setting is class incremental recognition [22, 8, 32], where new unseen classification categories with different domains arrive sequentially. To solve this problem, existing methods are commonly divided into three categories. Regulization-based methods [23, 37, 5, 25, 30, 33] add the constraints on the task-wise gradients to prevent the catastrophic forgetting when updating network weights for new arriving domains. Memory-based methods [24, 22, 19, 3, 4, 9, 14, 26, 21, 32] propose to store previous knowledge in a memory, such as a small set of examples, a part of weights, or episodic gradients to alleviate forgetting. Distillation-based methods [17, 22, 8, 2, 34, 36, 31, 18, 20] remember the knowledge trained on previous tasks by applying knowledge distillation between previous network and currently trained network. Please note that CCRL is distinct from conventional class incremental recognition. It is because CCRL needs to leverage the domain change (comparing two domains) to identify the latent variables. This implies that the problem cannot be divided into discrete local learning tasks, such as learning causal representation within individual domains and then merging them together, while training separate networks for different tasks will definitely reach state-of-the-art performance in a continual classification learning scenario. Thus, we introduce a memory model to store the information of previous domains and use it to adjust the model parameters.

3 Identifiable Nonlinear ICA with Sequentially Arriving Domains

In this section, we conduct a theoretical examination of the relationship between model identification and the number of domains. Initially, we introduce the causal generation process of our model (in Section 3.1), which considers the dynamics of changing domains. Subsequently, we demonstrate that model identifiability improves with the inclusion of additional domains. More specifically, we can achieve component-wise identification with $2n + 1$ domains (in Section 3.2.1), and subspace identification with $n + 1$ domains (in Section 3.2.2). Building on these theoretical insights, we introduce our method for learning causal representation in the context of continually emerging domains (in Section 3.3).

3.1 Problem Setting

As shown in Figure 1, we consider the data generation process as follows:

$$\mathbf{z}_c \sim p_{\mathbf{z}_c}, \quad \tilde{\mathbf{z}}_s \sim p_{\tilde{\mathbf{z}}_s}, \quad \mathbf{z}_s = f_{\mathbf{u}}(\tilde{\mathbf{z}}_s), \quad \mathbf{x} = g(\mathbf{z}_c, \mathbf{z}_s), \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ are the observations mixed by latent variables $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^n$ through an invertible and smooth nonlinear function $\mathbf{g} : \mathcal{Z} \rightarrow \mathcal{X}$. The latent varirables $\mathbf{z}$ can be partitioned into two groups: changing variables $\mathbf{z}_s \in \mathcal{Z}_s \subseteq \mathbb{R}^{n_s}$ whose distribution changes across domains $\mathbf{u}$, and invariant variables $\mathbf{z}_c \in \mathcal{Z}_c \subseteq \mathbb{R}^{n_c}$ which remains invariant. Given T domains in total, we have $p_{\mathbf{z}_s|\mathbf{u}_k} \neq p_{\mathbf{z}_s|\mathbf{u}_l}, p_{\mathbf{z}_c|\mathbf{u}_k} = p_{\mathbf{z}_s|\mathbf{u}_l}$ for all $k, l \in \{1, \dots, T\}, k \neq l$. We parameterize the influence of domains $\mathbf{u}$ for changing variables $\mathbf{z}_s$ as the function of $\mathbf{u}$ to its parent variables $\tilde{\mathbf{z}}_s$, i.e. $\mathbf{z}_s = f_{\mathbf{u}}(\tilde{\mathbf{z}}_s)$. One can understand this setting with the following example: suppose the higher level variables follow Gaussian distribution, i.e., $\tilde{\mathbf{z}}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{u}$ could be a vector denoting the variance of the distribution. The combination of $\mathbf{u}$ with $\tilde{\mathbf{z}}_s$ will produce a Gaussian variable with different variances at different domains. In this paper, we assume $\tilde{\mathbf{z}}_s$ follows the Gaussian distribution to make it tractable.

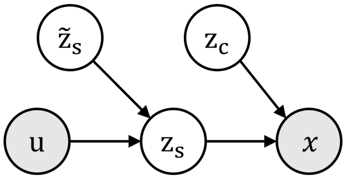

Figure 1: **Data generation process.** $\mathbf{x}$ is influenced by variables $\mathbf{z}_s$ (change with different domains $\mathbf{u}$) and invariant variables $\mathbf{z}_c$.

The objective of nonlinear ICA is to recover the latent variables $\mathbf{z}_s$ and $\mathbf{z}_c$ given the observation $\mathbf{x}$ and domain variables $\mathbf{u}$ by estimating the unmixing function $\mathbf{g}^{-1}$. In this paper, we consider the case where domains arrive sequentially, i.e., we aim to recover the latent variables by sequentially observing $\mathbf{x}|\mathbf{u}_1, \mathbf{x}|\mathbf{u}_2, \dots, \mathbf{x}|\mathbf{u}_T$.

3.2 Identifiability Theory of Nonlinear ICA

The identifiability is the key to nonlinear ICA to guarantee meaningful recovery of the latent variables. Mathematically, the identifiability of a model is defined as

$$\forall(\boldsymbol{\theta}, \boldsymbol{\theta}') : \quad p_{\boldsymbol{\theta}}(\mathbf{x}) = p_{\boldsymbol{\theta}'}(\mathbf{x}) \Longrightarrow \boldsymbol{\theta} = \boldsymbol{\theta}', \tag{2}$$

where $\boldsymbol{\theta}$ represents the parameter generating the observation $\mathbf{x}$. That is, if any two different choices of model parameter $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ lead to the same distribution, then this implies that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are equal [15]. For our data generation defined in Eq (1), we have $\boldsymbol{\theta} = (g, \mathbf{z}_c, \mathbf{z}_s)$, and $\boldsymbol{\theta}' = (\hat{g}, \hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s)$ which denotes the estimated mixing function, estimated invariant variables, and estimated changing variables respectively. Thus, a fully identifiable nonlinear ICA needs to satisfy at least two requirements: the ability to reconstruct the observation and the complete consistency with the true generating process. Unfortunately, current research is far from achieving this level of identifiability. Therefore, existing works typically adopt a weaker notion of identifiability. In the following, we discuss two types of identifiability for the changing variable, and show that the identifiability progressively increases from subspace identifiability to component-wise one by incorporating more domains.

In this work, we follow [16] and assume our estimated latent process $(\hat{g}, \hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s)$ could generate observation $\hat{\mathbf{x}}$ with identical distribution with observation $\mathbf{x}$ generated by the true latent process $(g, \mathbf{z}_c, \mathbf{z}_s)$, i.e.,

$$p_{\mathbf{x}|\mathbf{u}}(\mathbf{x}'|\mathbf{u}') = p_{\hat{\mathbf{x}}|\mathbf{u}}(\mathbf{x}'|\mathbf{u}'), \quad \mathbf{x}' \in \mathcal{X}, \mathbf{u}' \in \mathcal{U}. \tag{3}$$

3.2.1 Component-wise Identifiability for Changing Variable

First, we show that the changing variable can be identified up to permutation and component-wise invertible transformation with sufficient changing domains. Specifically, for the true latent changing variable $\mathbf{z}_s$, there exists an invertible function $h = g^{-1} \circ \hat{g} : \mathbb{R}^{n_s} \to \mathbb{R}^{n_s}$ such that $\mathbf{z}_s = h(\hat{\mathbf{z}}_s)$, where h is composed of a permutation transformation π and a component-wise nonlinear invertible transformation A, i.e., $\hat{g} = g \circ \pi \circ A$. That is, the estimated variable $\hat{z}_j$ and the true variable z_i have a one-to-one correspondence with an invertible transformation for $\forall i, j \in \{1, \dots, n_s\}$. We have the following lemma from [16].

Lemma 1 *Suppose that the data generation process follows Eq.* (1) *and that the following assumptions hold:*

1. *The set* $\{\mathbf{z} \in \mathbb{Z} \mid p(\mathbf{z}) = 0\}$ *has measure zero.*
2. *The probability density given each domain should be sufficiently smooth. i.e.,* $p_{\mathbf{z}|\mathbf{u}}$ *is at least second-order differentiable.*
3. *Given domain* $\mathbf{u}$*, every element of latent variable* $\mathbf{z}$ *should be independent with each other. i.e.,* $z_i \perp\!\!\!\perp z_j|\mathbf{u}$ *for* $i, j \in \{1, \dots, n\}$ *and* $i \neq j$*.*
4. *For any* $\mathbf{z}_s \in \mathcal{Z}_s$*, there exists* $2n_s + 1$ *values of* $\mathbf{u}$*, such that for* $k = 1, \dots, 2n_s$*,* $i = 1, \dots, n_s$*, the following matrix is invertible:*

$$\begin{bmatrix} \phi''_1(\mathbf{1},\mathbf{0}) & \dots & \phi''_i(\mathbf{1},\mathbf{0}) & \dots & \phi''_{n_s}(\mathbf{1},\mathbf{0}) & \phi'_1(\mathbf{1},\mathbf{0}) & \dots & \phi'_i(\mathbf{1},\mathbf{0}) & \dots & \phi'_{n_s}(\mathbf{1},\mathbf{0}) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi''_1(\mathbf{k},\mathbf{0}) & \dots & \phi''_i(\mathbf{k},\mathbf{0}) & \dots & \phi''_{n_s}(\mathbf{k},\mathbf{0}) & \phi'_1(\mathbf{k},\mathbf{0}) & \dots & \phi'_i(\mathbf{k},\mathbf{0}) & \dots & \phi'_{n_s}(\mathbf{k},\mathbf{0}) \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi''_1(\mathbf{2n_s},\mathbf{0}) & \dots & \phi''_i(\mathbf{2n_s},\mathbf{0}) & \dots & \phi''_{n_s}(\mathbf{2n_s},\mathbf{0}) & \phi'_1(\mathbf{2n_s},\mathbf{0}) & \dots & \phi'_i(\mathbf{2n_s},\mathbf{0}) & \dots & \phi'_{n_s}(\mathbf{2n_s},\mathbf{0}) \end{bmatrix},$$

where

$$\phi''_i(\mathbf{k},\mathbf{0}) := \frac{\partial^2 \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_k}))}{\partial z_i^2} - \frac{\partial^2 \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_0}))}{\partial z_i^2}, \phi'_i(\mathbf{k},\mathbf{0}) := \frac{\partial \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_k}))}{\partial z_i} - \frac{\partial \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_0}))}{\partial z_i}$$

are defined as as the difference between second-order derivative and first-order derivative of log density of z_i *between domain* $\mathbf{u_k}$ *and domain* $\mathbf{u_0}$ *respectively,*

Then, by learning the estimation $\hat{g}, \hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s$ *to achieve Eq* (3)*,* $\mathbf{z}_s$ *is component-wise identifiable.* [1]

The proof can be found in Appendix A. Basically, the theorem states that if the distribution of latent variables are "complex" enough and each domain brings enough changes to those changing variables, those changing variables $\mathbf{z}_s$ are component-wise identifiable.

Repeated distributions for partial changing variables. Previous works assume that when the domain changes, the changing variables will undergo a distribution shift. However, this assumption may be overly restrictive for practical scenarios, because there is no guarantee or clear justification that the data distribution of all changing variables will change when the domain changes. Specifically, there may exist domains with the same distribution for partial variables:

$$p_{\mathbf{z}|u}(z_i|\mathbf{u_k}) = p_{\mathbf{z}|u}(z_i|\mathbf{u_l}) \quad \exists k, l \in \{0, \dots, 2n_s\}, k \neq l, i \in \{1, \dots, n_s\}. \tag{4}$$

[1] We only focus on changing variables $\mathbf{z}_s$ in this paper. One may refer [16] for those who are interested in the identifiability of $\mathbf{z}_c$.

In practical human experience, we frequently encounter novel information that enhances or modifies our existing knowledge base. Often, these updates only alter specific aspects of our understanding, leaving the remainder intact. This raises an intriguing question about model identifiability: Does such partial knowledge alteration impact the invertibility of the matrix, as delineated in Assumption 4 of Lemma 1? To address this question, we present the following remark, with further details provided in Appendix A.

Remark 1 *Even when there are repeated distributions for partial variables, the component-wise identifiability for z_s still holds as long as the conditions of Lemma1 are satisfied. Furthermore, if there are more than two changing variables in the system, it is necessary for each changing variable to have at least three non-repetitive distributions across all domains.*

3.2.2 Subspace Identifiability for Changing Variable

Although component-wise identifiability is powerful and attractive, holding $2n_s + 1$ different domains with sufficient changes remains a rather strong condition and may be hard to meet in practice. In this regard, we investigate the problem of what will happen if we have fewer domains. We first introduce a notion of identifiability that is weaker compared to the component-wise identifiability discussed in the previous section.

Definition 1 (Subspace Identifiability of Changing Variable) *We say that the true changing variables $\mathbf{z}_s$ are subspace identifiable if, for the estimated changing variables $\hat{\mathbf{z}}_s$ and each changing variable $z_{s,i}$, there exists a function $h_i : \mathbb{R}^{n_s} \to \mathbb{R}$ such that $z_{s,i} = h_i(\hat{\mathbf{z}}_s)$.*

We now provide the following identifiability result that uses a considerably weaker condition (compared to Lemma 1) to achieve the subspace identifiability defined above, using only $n_s + 1$ domains.

Theorem 1 *Suppose that the data generation process follows Eq.* (1) *and that Assumptions 1, 2, and 3 of Lemma 1 hold. For any $\mathbf{z}_s \in \mathcal{Z}_s$, we further assume that there exists $n_s + 1$ values of $\mathbf{u}$ such that for $i = 1, \ldots, n_s$ and $k = 1, \ldots, n_s$, the following matrix*

$$\begin{bmatrix} \phi'_1(\mathbf{1},\mathbf{0}) & \ldots & \phi'_i(\mathbf{1},\mathbf{0}) & \ldots & \phi'_{n_s}(\mathbf{1},\mathbf{0}) \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \phi'(\mathbf{k},\mathbf{0}) & \ldots & \phi'_i(\mathbf{k},\mathbf{0}) & \ldots & \phi'_{n_s}(\mathbf{k},\mathbf{0}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi'_1(\mathbf{2n_s},\mathbf{0}) & \ldots & \phi'_i(\mathbf{2n_s},\mathbf{0}) & \ldots & \phi'_{n_s}(\mathbf{2n_s},\mathbf{0}) \end{bmatrix}$$

is invertible, where

$$\phi'_i(\mathbf{k},\mathbf{0}) := \frac{\partial \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_k}))}{\partial z_i} - \frac{\partial \log(p_{\mathbf{z}|\mathbf{u}}(z_i|\mathbf{u_0}))}{\partial z_i}$$

is the difference of first-order derivative of log density of z_i between domain $\mathbf{u_k}$ and domain $\mathbf{u_0}$ respectively. Then, by learning the estimation $\hat{g}, \hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s$ to achieve Eq (3)*, $\mathbf{z}_s$ is subspace identifiable.*

The proof can be found in Appendix A. Basically, Theorem 1 proposes a weaker form of identifiability with relaxed conditions. With $n_s + 1$ different domains, each true changing variable can be expressed as a function of all estimated changing variables. This indicates that the estimated changing variables capture all information for the true changing variables, and thus disentangle changing and invariant variables, enabling more robust and informed insights for potential applications in other fields such as domain adaptation. It is worth noting that if there is only one changing variable, such subspace identifiability can lead to component-wise identifiability.

New domains may impair original identifiability. Consider a toy case where there are three variables with four domains in total as shown in the top case of Figure 2. The first variable z_1 changes in domain $\mathbf{u}_1$ and both z_1 and z_2 change in domain $\mathbf{u}_2$. When considering only domains $\mathbf{u}_0$, $\mathbf{u}_1$, we can achieve component-wise identifiability for z_1. Given our subspace identifiability theory, z_1 can achieve subspace identifiability. As there is no change in the other variables in those two domains, this subspace identifiability is equal to competent-wise identifiability. However, when considering domains $\mathbf{u_0}, \mathbf{u_1}, \mathbf{u_2}$, the component-wise identifiability for z_1 disappears, and instead, we can only achieve subspace identifiability for both z_1 and z_2. In this case, information from domain $\mathbf{u}_2$ can be viewed as "noise" for z_1.

Contrasted with the traditional joint learning setting, where the data of all domains are overwhelmed, the continual learning setting offers a unique advantage. It allows for achieving and maintaining original identifiability, effectively insulating it from the potential "noise" introduced by newly arriving domains. In Section 4, we empirically demonstrate that the causal representation of z_1 obtained through continual learning exhibits better identifiability compared to that obtained through joint training.

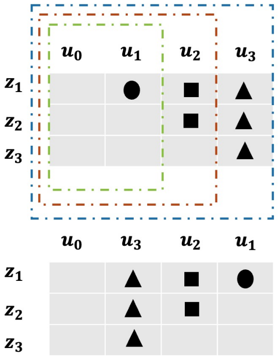

Figure 2: **A toy example with three variables and four domains.** z_1 changes in all three domains, z_2 changes in $\mathbf{u}_2, \mathbf{u}_3$, and z_3 changes in $\mathbf{u}_3$.

Learning order matters. Comparing both cases in Figure 2, they show the same identifiability considering all domains. However, we observe that in the top case, each new domain introduces a new changing variable, while in the bottom case, the domain order is reversed. Apparently, we can achieve subspace identifiability after learning each new domain in the top case, indicating that we can progressively improve our understanding and representation of the underlying causal factors with the arrival of each new domain. However, we can only achieve subspace identifiability until learning all domains in the bottom case. This is in line with the current learning system, where we first learn subjects with fewer changes before moving on to subjects with more complexity.

3.3 Method

In this section, we leverage the insight of the identifiability theory from previous section to develop our estimation method.

Generative model. As shown in Lemma 1 and Theorem 1, we are aiming at estimating causal process $\hat{g}, \hat{\mathbf{z}}_c, \hat{\mathbf{z}}_s$ to reconstruct the distribution of observation. As shown in Figure 3, we construct a Variational Autoencoder (VAE) with its encoder $q_{\hat{g}_\mu^{-1}, \hat{g}_\Sigma^{-1}}(\hat{\mathbf{z}}|\mathbf{x})$ to simulate the mixing process and the decoder $\hat{g}$ to reconstruct a matched distribution $\hat{\mathbf{x}} = \hat{g}(\hat{\mathbf{z}})$. Besides, as introduced in data generation in Equation 1, the changing latent variable is generated as the function of high-level invariance $\hat{\tilde{\mathbf{z}}}_s$ with a specific domain influence $\mathbf{u}$. Assuming the function is invertible, we employ a flow model to obtain the high-level variable $\hat{\tilde{\mathbf{z}}}_s$ by inverting the function, i.e., $\hat{\tilde{\mathbf{z}}}_s = \hat{f}_{\mathbf{u}}^{-1}(\hat{\mathbf{z}}_s)$. To train this model, we apply an ELBO loss as:

$$\begin{aligned}\mathcal{L}(\hat{g}_\mu^{-1}, \hat{g}_\Sigma^{-1}, \hat{f}_{\mathbf{u}}, \hat{g}) = \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\hat{\mathbf{z}} \sim q_{\hat{g}_\mu^{-1}, \hat{g}_\Sigma^{-1}}} \frac{1}{2} \|x - \hat{x}\|^2 + \alpha KL(q_{\hat{g}_\mu^{-1}, \hat{g}_\Sigma^{-1}}(\hat{\mathbf{z}}_c|\mathbf{x}) \| p(\mathbf{z}_c)) \\ + \beta KL(q_{\hat{g}_\mu^{-1}, \hat{g}_\Sigma^{-1}, \hat{f}_{\mathbf{u}}}(\hat{\tilde{\mathbf{z}}}_s|\mathbf{x} \| p(\tilde{\mathbf{z}}_s)),\end{aligned} \tag{5}$$

where α and β are hyperparameters controlling the factor as introduced in [7]. To make the Eq (5) tractable, we choose the prior distributions $p(\tilde{\mathbf{z}}_s)$ and $p(\tilde{\mathbf{z}}_c)$ as standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Continual casual representation learning. The subspace identifiability theory in Section 3.2.2 implies that the ground-truth solution lies on a manifold that can be further constrained with more side information, up to the solution with component-wise identifiability. Consequently, it is intuitive to expect that when we observe domains sequentially, the solution space should progressively narrow down in a reasonable manner.

It motivates us to first learn a local solution with existing domains and further improve it to align with the new arriving domain without destroying the original capacity. Specifically, to realize causal representation learning, we employ two objectives: (1) the reconstruction of observations within the current domain, and (2) the preservation of reconstruction capabilities for preceding domains. In terms of implementation, this implies that the movement of network parameters learning a new domain should not result in an increased loss for the previous domains.

To achieve this goal, we found the classical technique GEM [19] enables constraining the gradient update of network training to memorize knowledge from previous domains. The basic intuition of the algorithm can be illustrated with the following toy example: suppose data from those two domains are denoted as $\{\mathbf{x}|\mathbf{u_1}, \mathbf{x}|\mathbf{u_2}\}$ and the parameter of the network $\boldsymbol{\theta}$ and the loss calculated on data from kth domain is denoted as $l(\boldsymbol{\theta}, \mathbf{x}|\mathbf{u_k})$. At the moment of finishing the learning of the first domain, if

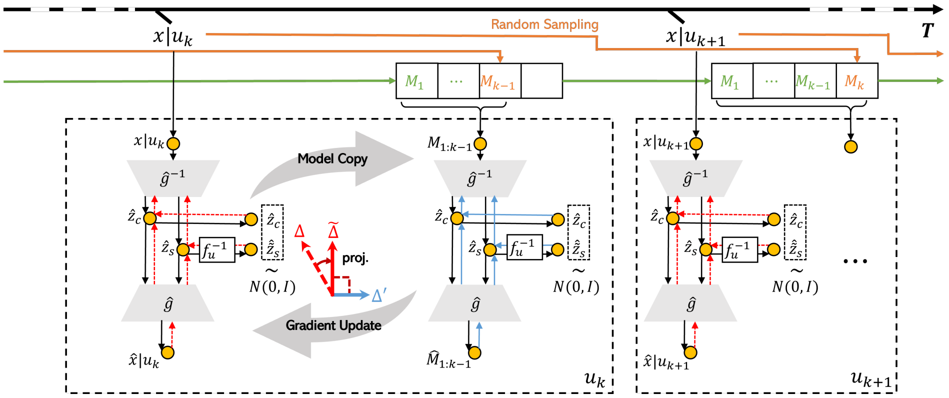

Figure 3: **Overall framework**. For the data from new domain $\mathbf{x}|\mathbf{u}_i$, we calculate the gradients Δ and Δ' of our model with both current data and previous memory. Then, we project the gradient Δ to $\tilde{\Delta}$ using Equation 7 when the angle between Δ and Δ' is larger than 90 degrees. Finally, we randomly sample a part of the data in the current domain and add them to the memory bank.

we don't make any constraints, the model should start the training using data from the second domain with the direction $\frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}$.

At this moment, if the direction $\frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}$ happens to have the property that $\langle \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}, \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_1})}{\partial \boldsymbol{\theta}} \rangle > 0$, the current direction will contribute to both domains and we remain the direction. Once the $\langle \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}, \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_1})}{\partial \boldsymbol{\theta}} \rangle < 0$ happens, we project the $\frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}$ to the direction where $\langle \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}, \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_1})}{\partial \boldsymbol{\theta}} \rangle = 0$, the orthogonal direction to $\frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_1})}{\partial \boldsymbol{\theta}}$ where no loss increment for previous domains. However, there are infinite possible directions satisfying the orthogonal direction requirement. e.g., we can always use the vector containing all zeros. To make the projected gradient as close as the original gradient, we solve for the projected gradient $\frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}'$ that minimizes the objective function

$$\left\| \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}} - \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}' \right\|^2 \quad \text{s.t.} \quad \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_2})}{\partial \boldsymbol{\theta}}^T \frac{\partial l(\boldsymbol{\theta},\mathbf{x}|\mathbf{u_1})}{\partial \boldsymbol{\theta}}' \geq 0. \tag{6}$$

Extend into the general case for multiple domains, we consider the following quadratic programming problem w.r.t. vector $\mathbf{v}'$:

$$\min_{\mathbf{v}'} \|\mathbf{v}-\mathbf{v}'\|^2 \quad \text{s.t.} \quad \mathbf{B}\mathbf{v}' \geq 0, \tag{7}$$

where $\mathbf{v}$ denotes the original gradient, $\mathbf{v}'$ denotes the projected gradient, $\mathbf{B}$ is the matrix storing all gradients of past domains. Note that practically, we only store a small portion of data for each domain, thus $\mathbf{B}_i$ is the row of $\mathbf{B}$ storing the memory gradient $\frac{\partial l(\boldsymbol{\theta},\mathbf{M}|\mathbf{u_i})}{\partial \boldsymbol{\theta}}$, where $\mathbf{M}|\mathbf{u_i} \in \mathbf{x}|\mathbf{u_i}$. We provide the complete procedure in Algorithm 1.

Algorithm 1 Continual Nonlinear ICA

```
Require: Training data sequentially arriving $\{\mathbf{x}|\mathbf{u_1},\ldots,\mathbf{x}|\mathbf{u_T}\}$
  Kaiming_init($\boldsymbol{\theta}$), $\mathcal{M}_t \leftarrow \{\}$ for all $t = 1,\ldots,T$
  for $\mathbf{u} = \mathbf{u_1},\ldots,\mathbf{u_T}$ do:
      for $\{\mathbf{x_1},\ldots,\mathbf{x_d}\}|\mathbf{u}$ do
          $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup$ random select $\mathbf{x}$
          Calculate loss $\mathcal{L}(\boldsymbol{\theta})$ as Eq (5)
          $\mathbf{v} \leftarrow \nabla_\theta \mathcal{L}(\boldsymbol{\theta},\mathbf{x})$
          $\mathbf{v}_k \leftarrow \nabla_\theta \mathcal{L}(\boldsymbol{\theta},\mathcal{M}_k)$ for all $k < t$
          $\mathbf{v}' \leftarrow$ Solve quadratic programming as Eq (7)
          $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha\mathbf{v}'$
  Return $\boldsymbol{\theta}$
```

4 Experiments

In this section, we present the implementing details of our method, the experimental results, and the corresponding analysis.

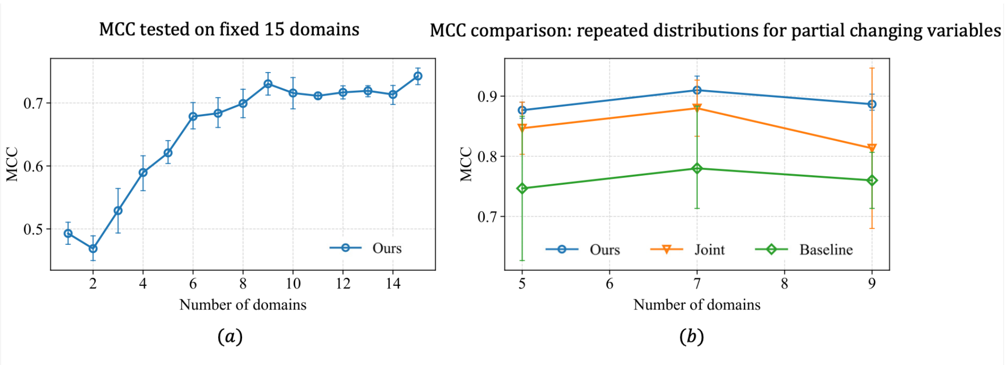

Figure 5: (a) MCC for increasing domains with models tested on all 15 domains after training of each domain. (b) MCC for models trained with repeated distributions for partial changing variables.

4.1 Experiment Setup

Data. We follow the standard practice employed in previous work [13, 16] and compare our method to the baselines on synthetic data. We generate the latent variables $\mathbf{z}_s$ for both non-stationary Gaussian and mixed Gaussian with domain-influenced variance and mean, while $\mathbf{z}_c$ for standard Gaussian and mixed Gaussian with constant mean and variance. The mixing function is estimated by a 2-layer Multi-Layer Perception(MLP) with Leaky-Relu activation. More details can be found in Appendix C.

Evaluation metrics. We use Mean Correlation Coefficient (MCC) to measure the identifiability of the changing variable $\mathbf{z}_s$. However, as the identifiability result can only guarantee component-wise identifiability, it may not be fair to directly use MCC between $\hat{\mathbf{z}}_s$ and $\mathbf{z}_s$ (e.g. if $\hat{\mathbf{z}}_s = \mathbf{z}_s^3$, we will get a distorted MCC value). We thus separate the test data into the training part and test part, and further train separate MLP to learn a simple regression for each $\hat{\mathbf{z}}_s$ to $\mathbf{z}_s$ to remove its nonlinearity on the training part and compute the final MCC on these part. We repeat our experiments over 5 or 3 random seeds for different settings.

4.2 Experimental Results

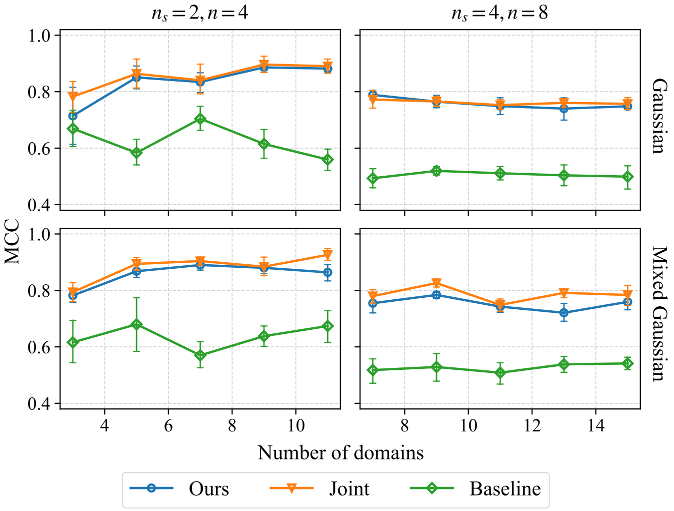

Figure 4: Comparison of MCC for all four datasets with the number of domains from $2n_s - 1$ to $2n_s + 7$. Note in this case, the number of domains for training is consistent with the number of domains for testing (different from cases we investigate for increasing domains).

Comparison to baseline and joint training. We evaluate the efficacy of our proposed approach by comparing it against the nonlinear ICA methods trained on sequentially arriving domains and multiple domains simultaneously, referred to as the baseline and theoretical upper bound by the continual learning community. We employ identical network architectures for all three models and examine four distinct datasets, with respective parameters of $\mathbf{z}_s$ being Gaussian and mixed Gaussian with $n_s = 4$, $n = 8$, as well as $n_s = 2$, $n = 4$. Increasing numbers of domains are assessed for each dataset. Figure 4 show our method reaches comparable performance with joint training. Further visualization can be found in Appendix B.

Increasing domains. For dataset $n_s = 4, n = 8$ of Gaussian, we save every trained model after each domain and evaluate their MCC. Specifically, we tested the dataset original test dataset containing data from 15 domains. As shown in part (a) of Figure 5, remarkably, increasing domains lead to greater identifiability results, which align with our expectations that sequential learning uncovers the true underlying causal relationships as more information is revealed. Specifically, our approach progressively transitions from subspace identifiable to component-wise identifiable as new domains are involved. Given $2n_s + 1 = 9$ domains, we observe that our model achieves the component-wise identifiability and the extra domains (from 9 to 15) do not provide further improvement.

Repeated distributions for partial changing variables. For dataset $n_s = 2, n = 4$ of Gaussian, we test the case that $z_{s,1}$ have changing distributions over all domains while $z_{s,2}$ only holds three different distributions across domains. As shown in part (b) of Figure 5, our method outperforms both joint train and baseline. It may be because our method has the ability to maintain the performance learned from previous domains and prevents potential impairment from new arriving domains with repeated distributions. For this instance, our method exhibits more robust performance than joint training against negative effects from 8, 9th domains.

Discussion: is joint training always better than learning sequentially? Not necessarily. As discussed in Section3.2.2, the new domain may impair the identifiability of partial variables. While joint training always shuffles the data and doesn't care about the order information, learning sequentially to some extent mitigates the impairment of identifiability. Specifically, we conducted an experiment in which both z_1 and z_2 are Gaussian variables. The variance and mean of z_1 change in the second domain, while the other variable change in the third domain. We then compare our method with joint training only for latent variable z_1. We repeat our experiments with 3 random seeds and the experiment shows that the MCC of our method for z_1 reaches up to 0.785 while joint training retains at 0.68 as shown in Figure 6. In terms of visual contrast, the scatter plot obtained using our method on the left of Figure 6 exhibits a significantly stronger linear correlation compared to joint training.

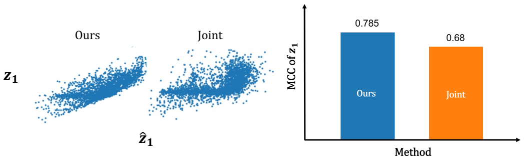

Figure 6: Comparison of identifiability for z_1 using Joint training and our method qualitatively and quantitatively.

5 Conclusion

Conclusion. In this paper, we present a novel approach for learning causal representation in continually arriving domains. Through theoretical analysis, we have examined the relationship between model identification and the number of observed domains. Our findings indicate that as additional domains are incorporated, the identifiability of changing variables escalates, with subspace identification achievable with $n_s + 1$ domains and component-wise identification requiring $2n_s + 1$ domains or more. Besides, we briefly show that a carefully chosen order of learning leads to meaningful disentanglement after each domain is learned, and the introduction of new domains does not necessarily contribute to all variables. To realize CCRL, we employed GEM to preserve prior knowledge and refine it using information derived from incoming domains, resembling human learning mechanisms. Empirical evaluations have demonstrated that our approach achieves performance on par with nonlinear ICA techniques trained jointly across multiple offline domains, exhibiting greater identifiability with increasing domains observed.

Limitation. To improve our methodology, we should address two key areas. Firstly, develop a continual learning method that can automatically determine the number of changing variables without prior knowledge. This would enhance adaptability to real-world scenarios. Secondly, consider incorporating anchor points or high-level semantic representations into the memory storage process, aligning it with human memory processes for more effective storage and recall of information.

Broader Impact. This work proposes a theoretical analysis to learn the causal representation, which aids in the construction of more transparent and interpretable models. This could be beneficial in a variety of sectors, including healthcare, finance, and technology. For instance, in healthcare, it could be used to continually monitor and analyze patient data, helping to predict health risks and inform treatment decisions in a timely manner. In contrast, misinterpretations of causal relationships could have significant implications in these fields, which must be deliberated.

References

[1] Simon Buchholz, Michel Besserve, and Bernhard Schölkopf. Function classes for identifiable nonlinear independent component analysis. In *Advances in Neural Information Processing Systems*, 2022.

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[3] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[4] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019.

[5] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020.

[6] Luigi Gresele, Julius Von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? In *Advances in Neural Information Processing Systems*, 2021.

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[8] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[9] Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao Tao, Dongyan Zhao, Jinwen Ma, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International conference on learning representations*, 2019.

[10] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica, 2016.

[11] Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR, 2017.

[12] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[13] Aapo Hyvarinen, Hiroaki Sasaki, and Richard E. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning, 2018.

[14] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[15] Ilyes Khemakhem, Diederik P. Kingma, Ricardo Pio Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ica: A unifying framework, 2019.

[16] Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial disentanglement for domain adaptation. In *International Conference on Machine Learning*, pages 11455–11472. PMLR, 2022.

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[18] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020.

[19] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[20] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3513–3522, 2021.

[21] Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent replay for real-time continual learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10203–10209. IEEE, 2020.

[22] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[23] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[24] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[25] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

[26] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[27] Ricardo Silva, Richard Scheine, Clark Glymour, and Peter Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(8):191–246, 2006.

[28] Charles Spearman. Pearson's contribution to the theory of two factors. *British Journal of Psychology*, 19(1):95, 1928.

[29] Henning Sprekeler, Tiziano Zito, and Laurenz Wiskott. An extension of slow feature analysis for nonlinear blind source separation. *The Journal of Machine Learning Research*, 15(1):921–947, 2014.

[30] Shixiang Tang, Dapeng Chen, Jinguo Zhu, Shijie Yu, and Wanli Ouyang. Layerwise optimization by gradient decomposition for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9634–9643, 2021.

[31] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-preserving class-incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 254–270. Springer, 2020.

[32] Gido M Van De Ven, Zhe Li, and Andreas S Tolias. Class-incremental learning with generative classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3611–3620, 2021.

[33] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021.

[34] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[35] Feng Xie, Ruichu Cai, Biwei Huang, Clark Glymour, Zhifeng Hao, and Kun Zhang. Generalized independent noise condition for estimating latent variable causal graphs, 2020.

[36] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6982–6991, 2020.

[37] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

[38] Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *arXiv preprint arXiv:2206.07751*, 2022.