Weakly Supervised Representation Learning with Sparse Perturbations

Kartik Ahuja[*] **Jason Hartford**[*] **Yoshua Bengio**[†*]

Abstract

The theory of representation learning aims to build methods that provably invert the data generating process with minimal domain knowledge or any source of supervision. Most prior approaches require strong distributional assumptions on the latent variables and weak supervision (auxiliary information such as timestamps) to provide provable identification guarantees. In this work, we show that if one has weak supervision from observations generated by sparse perturbations of the latent variables–e.g. images in a reinforcement learning environment where actions move individual sprites–identification is achievable under unknown continuous latent distributions. We show that if the perturbations are applied only on mutually exclusive blocks of latents, we identify the latents up to those blocks. We also show that if these perturbation blocks overlap, we identify latents up to the smallest blocks shared across perturbations. Consequently, if there are blocks that intersect in one latent variable only, then such latents are identified up to permutation and scaling. We propose a natural estimation procedure based on this theory and illustrate it on low-dimensional synthetic and image-based experiments.

1 Introduction

If you are reading this paper on a computer, press one of the arrow keys... all the text you are reading jumps as the screen refreshes in response to your action. Now imagine you were playing a video game like Atari's Space Invaders—the same keystroke would cause a small sprite at the bottom of your screen to move in response. These actions induce changes in pixels that are very different, but in both cases, the visual feedback in response to our actions indicates the presence of some object on the screen—a virtual paper and a virtual spacecraft, respectively—with properties that we can manipulate. Our keystrokes induce sparse changes to a program's state, and these changes are reflected on the screen, albeit not necessarily in a correspondingly sparse way (e.g., most pixels change when scrolling). Similarly, many of our interactions with the real world induce sparse changes to the underlying causal factors of our environment: lift a coffee cup and the cup moves, but not the rest of the objects on your desk; turn your head laterally, and the coordinates of all the objects in the room shift, but only in the horizontal direction. These examples hint at the main question we aim to answer in this paper: if we know that actions have sparse effects on the latent factors of our system, can we use that knowledge as weak supervision to help disentangle these latent factors from pixel-level data?

Self– and weakly-supervised learning approaches have made phenomenal progress in the last few years, with large-scale systems like GPT-3 (Brown et al., 2020) offering large improvements on all natural language benchmarks, and CLIP (Radford et al., 2021) outperforming state-of-the-art supervised models from six years ago (Szegedy et al., 2016) on the ImageNet challenge (Deng et al., 2009) without using any of the labels.

[*]Mila - Quebec AI Institute, Université de Montréal. [†]CIFAR Fellow.
Correspondence to: kartik.ahuja@mila.quebec.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

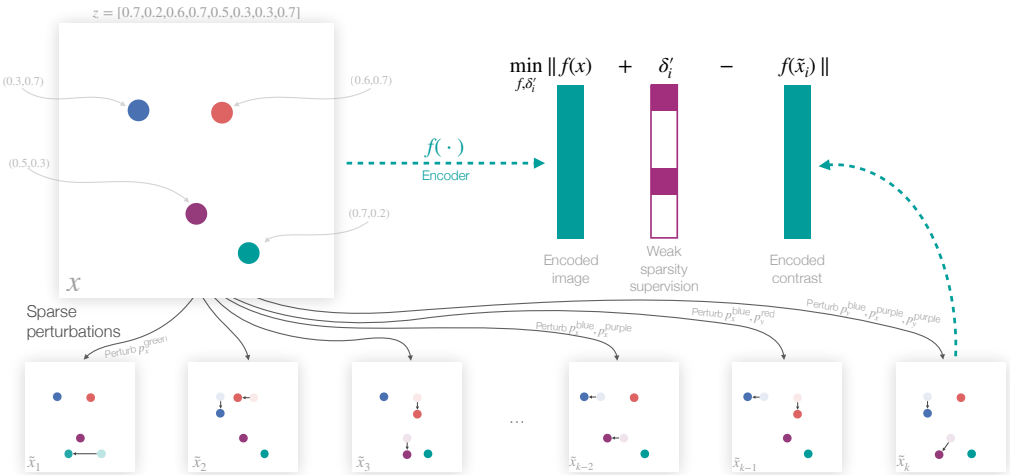

Figure 1: Ball agent interaction environment. Different frames show the effect of sparse perturbations.

Yet, despite these advances, these systems are still far from human reasoning abilities and often fail on out-of-distribution examples (Geirhos et al., 2020). To robustly generalize out of distribution, we need models that can infer the causal mechanisms that relate latent variables (Schölkopf et al., 2021; Schölkopf and von Kügelgen, 2022) because these mechanisms are invariant under distribution shift. The field of causal inference has developed theory and methods to infer causal mechanisms from data (Pearl, 2009; Peters et al., 2017), but these methods assume access to high-level abstract features, instead of low-level signal data such as video, text and images. We need representation learning methods that reliably recover these abstract features if we are to bridge the gap between causal inference and deep learning.

This is a challenging task because the problem of inferring latent variables is not identified with independent and identically distributed (IID) data (Hyvärinen and Pajunen, 1999; Locatello et al., 2019), even in the limit of an infinite number of such IID examples. However, there has been significant recent progress in developing representation learning approaches that provably recover latent factors Z (e.g., object positions, object colors, etc.) underlying complex data X (e.g. image), where $X \leftarrow g(Z)$, by going beyond the IID setting and using observations of X along with minimal domain knowledge and supervision (Hyvarinen and Morioka, 2016, 2017; Locatello et al., 2020; Khemakhem et al., 2020a). These works establish provable identification of latents by leveraging strong structural assumptions such as independence conditional on auxiliary information (e.g., timestamps). In this work, we aim to relax these distributional assumptions on the latent variables to achieve identification for arbitrary continuous latent distributions. Instead of distributional assumptions, we assume access to data generated under sparse perturbations that change only a few latent variables at a time as a source of weak supervision. Figure 1 illustrates our working example of this assumption: a simple environment where an agent's actions perturb the coordinates of a few balls at a time. Our main contributions are summarized as follows.

- We show that sparse perturbations that impact one latent at a time are sufficient to learn the latents (up to permutation and scaling) that follow any unknown continuous distribution.
- Next, we consider more general settings, where perturbations impact one block of latent variables at a time. In the setting where blocks do not overlap, we recover the latents up to an affine transformation of these blocks.
- Further, we show that when perturbation blocks overlap, we get stronger identification. In this setting, we prove identification up to affine transformation of the smallest intersecting block. Consequently, if there are blocks that intersect in one latent variable only, then such latents are identified up to permutation and scaling.
- We leverage these results to propose a natural estimation procedure and experimentally illustrate the theoretical claims on low-dimensional synthetic and high-dimensional image-based data.

2 Related works

Many of the works on provable identification of representations trace their roots to non-linear ICA (Hyvärinen and Pajunen, 1999). Hyvarinen and Morioka (2016, 2017) were the first to use auxiliary information in the form of timestamps and additional structure on the latent evolution to achieve provable identification. Since then, these works have been generalized in many exciting ways. Khemakhem et al. (2020a) assume independence of latents conditional on auxiliary information, and several of these assumptions were further relaxed by Khemakhem et al. (2020b).

Our work builds on the machinery developed by Ahuja et al. (2022). Ahuja et al. show that if we know the mechanisms that drive the evolution of latents, then the latents are identified up to equivariances of these mechanisms. However, the authors leave the question of achieving identification without such knowledge open. Here we consider a class of mechanisms where an agent's actions impact the latents through unknown perturbations. We show how to achieve identification by exploiting the sparsity in the perturbations. This class of perturbations was first leveraged to prove identification by Locatello et al. (2020). However, Locatello et al. assume that the latents are independent, whereas we make no assumptions on the distribution other than continuity. Lachapelle et al. (2022) also use sparse interventions on the latents to strengthen the identification guarantees in Khemakhem et al. (2020a) for conditional exponential distributions. However, the form of sparsity that is leveraged in their work is different from ours. In our work, we assume that the vector of changes in the latents is sparse, i.e., some components change and the rest of the components do not change. In Lachapelle et al. (2022), all the components of the latents change post interventions but the graphical model capturing the interaction of interventions (described as random variables) and the latents is sparse. Klindt et al. (2021) also use sparsity in time-series settings to attain identification. Klindt et al. (2021) enforce soft ℓ_1 norm driven sparsity in the vector of changes in latents by assuming that latents evolve independently under a Laplace distribution but do not require access to data under interventions.

Yao et al. (2021) and Lippe et al. (2022) model the latent evolution as a structural causal model unrolled in time. Yao et al. exploit non-stationarity and sufficient variability dictated by the auxiliary information to provide identification guarantees. Lippe et al. exploit causal interventions on the latents to provide identification guarantees but require the knowledge of intervention targets and assume an invariant causal model describing the relations between any adjacent time frames. In concurrent work, Brehmer et al. (2022) leverage data generated under causal interventions as a source of weak supervision and prove identification for structural causal models that are diffeomorphic transforms of exogenous noise. Our work also connects to an insightful line of work on multi-view ICA (Gresele et al., 2020), which proves identification under independent latents, in the following sense. We can interpret the data under different perturbations as different views of the same underlying latent. In addition, some recent papers explain the success of self-supervised contrastive learning Zimmermann et al. (2021); Von Kügelgen et al. (2021) through the lens of identification of representations.

Above, we focused on provable representation identification, which is central to this work. We now give a brief overview of empirical works on disentanglement, which is a weaker form of representation identification. Many variations of variational autoencoders (VAE) were developed over the years to achieve disentanglement. β-VAE (Higgins et al., 2016) uses a hyperparameter in front of the KL regularizer to make the learned latent independent. Factor VAE (Kim and Mnih, 2018) proposes an adversarial training-based approach, where the discriminator encourages the learned representation to have independent components. Annealed β-VAE (Burgess et al., 2018) proposes to progressively increase the capacity of bottleneck β to enforce independence one component at a time. Ideas based on disentanglement have been also used in reinforcement learning; (Higgins et al., 2017), (Dittadi et al., 2020), (Miladinović et al., 2019) are some of the representative works in the area. Locatello et al. (2019) showed that most of the above-said methods could often fail to disentangle in the absence of supervision or inductive biases. As a result, there has been a surge in the interest in building approaches that achieve provable representation identification. Lastly, there is a line of work, which does not focus on disentanglement or representation identification, but has shown the benefits of sparsity based inductive biases – sparse changes in latents over time (Goyal et al., 2019) or sparse interactions between latents (Goyal et al., 2021) – under distribution shifts.

3 Latent identification under sparse perturbations

Data Generation Process We start by describing the data generation process used for the rest of the work. There are two classes of variables we consider – a) unobserved latent variables $Z \in \mathcal{Z} \subseteq \mathbb{R}^d$

and b) observed variables $X \in \mathcal{X} \subseteq \mathbb{R}^n$. The latent variables Z are sampled from a distribution $\mathbb{P}_Z$ and then transformed by a map $g : \mathbb{R}^d \to \mathbb{R}^n$, where g is injective and analytic[2], to generate X. We write this as follows

$$z \sim \mathbb{P}_Z \qquad x \leftarrow g(z) \tag{1}$$

where z and x are realizations of the random variables Z and X respectively. It is impossible to invert g just from the realizations of X (Hyvärinen and Pajunen, 1999; Locatello et al., 2019). Most work has gone into understanding how structure of latents Z and auxiliary information in the form of timestamps or labels play a role in solving the above problem. In this work, we depart from these assumptions and instead investigate the role of data generated under perturbations of latents to achieve identification. Define the set of perturbations as $\mathcal{I} = \{1, \cdots, m\}$ and the corresponding perturbation vectors as $\Delta = \{\delta_1, \cdots, \delta_m\}$, where δ_i is the ith perturbation. Each latent z is sampled from an arbitrary and unknown distribution $\mathbb{P}_Z$. The *same set of unknown perturbations* in Δ are applied to each z to generate m perturbed latents $\{\tilde{z}_k\}_{k=1}^m$ per sampled z and the corresponding observed vectors $\{\tilde{x}_k\}_{k=1}^m$. Each of these latents are transformed by the map g and we observe $(x, \tilde{x}_1, \cdots, \tilde{x}_m)$. Our goal is to use these observations and estimate the underlying latents. We summarize this data generation process (DGP) in the following assumption.

Assumption 1. *The DGP follows*

$$z \sim \mathbb{P}_Z, x \leftarrow g(z) \qquad \tilde{z}_k \leftarrow z + \delta_k, \forall k \in \mathcal{I} \qquad \tilde{x}_k \leftarrow g(\tilde{z}_k), \forall k \in \mathcal{I} \tag{2}$$

where g is injective and analytic, and Z is a continuous random vector with full support over $\mathbb{R}^d$. [3]

The above DGP is very close to the DGP in Locatello et al. (2020) except we do not require latent dimensions to be mutually independent. To better understand the above DGP, let us turn to some examples. Consider a setting where an agent is interacting with an environment containing several balls (See Figure 1). The latent z captures the properties of the objects; for example, in Figure 1, z just captures the positions of each ball, but in general it could include more properties such as velocity, shape, color, etc.. The agent perturbs the objects in the scene by δ_k, which can modify a single property associated with one object or multiple properties from one or more objects depending on how the agent acts. Note that when the agent perturbs a latent, it can lead to downstream effects. For instance, if the agent moves a ball to the edge of the table, the ball falls in subsequent frames. For this work, we only consider the observations just before and after the perturbation and not the downstream effects. In the Appendix (Section A.2.5), we explain these downstream effects using structural causal models. We also explain the connection between the perturbations in equation (24) and causal interventions leveraged in Brehmer et al. (2022); Lachapelle et al. (2022). The above example is typical of a reinforcement learning environment, other examples include natural videos with sparse changes (e.g., MPI3D data (Gondal et al., 2019)).

In the above DGP in equation (24), we assumed that for each scene x there are multiple perturbations. It is possible to extend our results to settings where we perturb each scene only once, given a sufficiently diverse set of perturbations, i.e., for a small neighborhood of a scene around x, each scene in the neighbourhood receives a different perturbation. We compare these two approaches experimentally.

Learning objective The learner's objective is to use the observed samples $(x, \tilde{x}_1, \cdots, \tilde{x}_m)$ generated by the DGP in Assumption 1 and learn an encoder $f : \mathbb{R}^n \to \mathbb{R}^d$ that inverts the function g and recovers the true latents. For each observed sample $(x, \tilde{x}_1, \cdots, \tilde{x}_m)$, the learner compares all the pairs $(x, \tilde{x}_k)$ pre- and post-perturbation. For every unknown perturbation δ_k used in the DGP in equation (24), the learner guesses the perturbation $\delta_k^{'}$ and enforces that the latents predicted by the encoder for x and $\tilde{x}_k$ are consistent with the guess. We write this as $\forall\, (x, \tilde{x}_1, \cdots, \tilde{x}_m)$ generated by DGP in (24)

$$f(\tilde{x}_k) = f(x) + \delta_k^{'}. \tag{3}$$

We denote the set of guessed perturbations as $\Delta^{'} = \{\delta_1^{'}, \cdots, \delta_m^{'}\}$, where $\delta_i^{'}$ is the guess for perturbation δ_i. We can turn the above identity into a mean square error loss given as

[2] A *analytic* function, g, is an infinitely differentiable function such that for all z' in its domain, the Taylor series evaluated at z' converges pointwise to $g(z')$

[3] The assumption on the support of Z can be relaxed.

$$\min_{f,\Delta^{'}} \mathbb{E}\Big[\Big(f(\tilde{x}_k) - f(x) - \delta_k^{'}\Big)^2\Big] \tag{4}$$

where the expectation is taken over observed samples generated by the DGP in (24) and the minimization is over all the possible maps f and perturbation guesses in the set $\Delta^{'}$. Note that a trivial solution to the above problem is an encoder that maps everything to zero, and all guesses equal zero. In the next section, we get rid of these trivial solutions by imposing an additional requirement that the span of the set $\Delta^{'}$ is $\mathbb{R}^d$. It is worth pointing out that we do not restrict the set of f's to injective maps in theory and experiments. We denote the latent estimated by the encoder for a point x as $\hat{z} = f(x)$. It is related to the true latent as follows $\hat{z} = f \circ g(z) = a(z)$, where a is some function that relates true z to estimated $\hat{z}$. In the next section, we show that if perturbations are diverse, then a is an affine transform. Further, we show that if perturbations are sparse, then a takes an even simpler form.

3.1 Sparse perturbations

We first show that it is possible to identify the true latents up to an affine transformation without any sparsity assumptions. Later, we leverage sparsity to strengthen identification guarantees.

Assumption 2. *The dimension of the span of the perturbations in* (24) *is d, i.e.,* $\mathsf{dim}\Big(\mathsf{span}\big(\Delta\big)\Big) = d$.

The above assumption implies that the perturbations are diverse. We now state a regularity condition on the function a.

Assumption 3. *a is an analytic function. For each component $i \in \{1, \cdots, d\}$ of $a(z)$ and each component $j \in \{1, \cdots, d\}$ of z, define the set $\mathcal{S}^{ij} = \{\theta \mid \nabla_j a_i(z+b) = \nabla_j a_i(z) + \nabla_j^2 a_i(\theta) b, z \in \mathbb{R}^d\}$, where b is a fixed vector in $\mathbb{R}^d$. Each set $\mathcal{S}^{ij}$ has a non-zero Lebesgue measure in $\mathbb{R}^d$.*

If we restrict the encoder f to be analytic, then a is analytic since g is also analytic, thus satisfying the first part of the above assumption. The second part of the above assumption can be understood as follows: suppose we have a scalar valued function $h : \mathbb{R} \to \mathbb{R}$ that is differentiable. If we expand $h(u+v)$ around $h(u)$, by the mean value theorem we get $h(u+v) = h(u) + h'(\epsilon)v$, where $\epsilon \in [u, u+v]$. If we vary u to take all the values in $\mathbb{R}$, then ϵ also varies. The above assumption states that the set of $\epsilon's$ has a non-zero Lebesgue measure. Under the above assumptions, we show that an encoder that solves equation (3) identifies true latents up to an affine transform, i.e., $\hat{z} = Az + c$, where $A \in \mathbb{R}^{d \times d}$ is a matrix and $c \in \mathbb{R}^d$ is an offset.

Proposition 1. *If Assumptions 1, 2, and 3 hold, then the encoder that solves equation* (3) *(with $\Delta^{'}$ s.t.* $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$*) identifies true latents up to an invertible affine transform, i.e. $\hat{z} = Az + c$, where $A \in \mathbb{R}^{d \times d}$ is an invertible matrix and $c \in \mathbb{R}^d$ is an offset.*

The proof of above proposition follows the proof technique from Ahuja et al. (2022), for further details refer to the Appendix (Section A.1). We interpret the above result in the context of the agent interacting with balls (as shown in Figure 1), where the latent vector z captures the x and y coordinates of the n_{balls}. Under each perturbation, the balls move along the vector dictated by the perturbation. If there are at least $2n_{\mathsf{balls}}$ perturbations, then the latents estimated by the learned encoder are guaranteed to be an affine transformation of the actual positions of the balls.

3.1.1 Non-overlapping perturbations

In Proposition 1, we showed affine identification guarantees for the DGP from Assumption 1. We now explore identification when perturbations are one-sparse, i.e., one latent changes at a time.

Assumption 4. *The perturbations in Δ are one-sparse, i.e., each $\delta_i \in \Delta$ has one non-zero component.*

Next, we show that under one-sparse perturbations, the latents estimated identify true latents up to permutation and scaling.

Theorem 1. *If Assumptions 1-4 hold and the number of perturbations per example equals the latent dimension, $m = d$,* [4] *then the encoder that solves equation* (3) *(with $\Delta^{'}$ as one-sparse and*

[4] We can relax this condition to $m \geq d$, refer to the Appendix for details.

$\dim\Big(\text{span}(\Delta^{'})\Big) = d$) *identifies true latents up to permutation and scaling, i.e.* $\hat{z} = \Pi\Lambda z + c$, *where* $\Lambda \in \mathbb{R}^{d\times d}$ *is an invertible diagonal matrix,* $\Pi \in \mathbb{R}^{d\times d}$ *is a permutation matrix and* c *is an offset.*

For the proof of above theorem, refer to Section A.1 in the Appendix. The theorem does not require that learner knows either the identity or amount each component changed. However, the learner has to use one-sparse perturbations as guesses. Suppose the learner does not know that the actual perturbations are one-sparse and instead uses guesses that are p-sparse, i.e., p latents change at one time. In such a case, the $\hat{z}$ and true z are related to each other through a permutation and block diagonal matrix, i.e., we can replace Λ in the above result to be a block diagonal matrix instead of a diagonal matrix (see the Appendix for details). In the context of the ball agent interaction environment from Figure 1, the above result states that provided the agent interacts with one coordinate of each ball at a time, it is possible to learn the position of each ball up to scaling errors.

We now consider a natural extension of the setting above, where the perturbations simultaneously operate on blocks of latents. In the ball agent interaction environment, this can lead to multiple scenarios – i) the agent interacts with one ball at a time but perturbs both coordinates simultaneously, ii) the agent interacts with several balls simultaneously.

Consider a perturbation $\delta_i \in \Delta$ (from equation (24)). We define the block of latents that is impacted under perturbation $\delta_i \in \Delta$ as $\{j \mid \delta_i^j \neq 0, j \in \{1, \cdots, d\}\}$, where δ_i^j is the jth component of δ_i. We group the perturbations in $\mathcal{I}$ based on the block they act upon, i.e. perturbations in the same group act on the same block of latents. Define the set of the groups corresponding to perturbations in $\mathcal{I}$ as $\mathcal{G}_\mathcal{I}$. Define the set of corresponding blocks as $\mathcal{B}_\mathcal{I} = \{\mathcal{B}_1, \cdots, \mathcal{B}_g\}$, where $\mathcal{B}_k$ is the block impacted by perturbations in group k. If $\mathcal{B}_\mathcal{I}$ partitions the set of latent components indexed $\{1, \cdots, d\}$, then it implies all the distinct blocks are non-overlapping. We formally define this below.

Definition 1. ***Blockwise and non-overlapping perturbations.*** *If the the set of blocks* $\mathcal{B}_\mathcal{I}$ *corresponding to perturbations* $\mathcal{I}$ *form a partition of* $\{1, \cdots, d\}$, *then* $\mathcal{I}$ *is said to be blockwise and non-overlapping. Formally stated, any two distinct* $\mathcal{B}_i, \mathcal{B}_j \in \mathcal{B}_\mathcal{I}$ *do not intersect, i.e.,* $\mathcal{B}_i \cap \mathcal{B}_j = \emptyset$, *and* $\cup_i \mathcal{B}_i = \{1, \cdots, d\}$.

From the above definition it follows that two perturbations either act on the same block or completely different blocks with no overlapping variables.

Assumption 5. *The perturbations* $\mathcal{I}$ *(used in equation* (24)*) are blockwise and non-overlapping (see Definition 1). Each perturbation in* $\mathcal{I}$ *is* p*-sparse, i.e., it impacts blocks of length* p ($p \leq d$) *at a time.*

Assumption 6. *The learner knows the group label for each perturbation* $i \in \mathcal{I}$. *Therefore, any two perturbations in* $\Delta^{'}$ *associated with same group in* $\mathcal{G}_\mathcal{I}$ *impact the same block of latents.*

We illustrate the above Assumptions 5, 6 in the following example. Consider the ball agent interaction environment (Figure 1). $z = [z_{1x}, z_{1y}, \cdots, z_{n_{\text{balls}}x}, z_{n_{\text{balls}}y}]$ is the vector of positions of all balls, where $z_{ix/y}$ is the x/y coordinate of ball i. If the agent randomly perturbs ball i, then it changes the block (z_{ix}, z_{iy}). We would call such a system 2-sparse. All the perturbations on ball i are in one group. Since the agent knows the group of the perturbation, it does not know the ball index but it knows whenever we interact with the same ball.

Definition 2. *If the latent variables recovered* $\hat{z} = \Pi\tilde{\Lambda}z + c$, *where* Π *is a permutation matrix and* $\tilde{\Lambda}$ *is a block-diagonal matrix, then the latent variables are said to be recovered up to permutations and block-diagonal transforms.*

In the theorem that follows, we show that under the assumptions made in this section, we achieve identification up to permutations and block-diagonal transforms with invertible $p \times p$ blocks.

Theorem 2. *If Assumptions 1-3, 5, 6 hold, then the encoder that solves equation* (3) *(where* $\Delta^{'}$ *is* p*-sparse,* $\dim\Big(\text{span}(\Delta^{'})\Big) = d$) *identifies true latents up to permutation and block-diagonal transforms, i.e.* $f(x) = \hat{z} = \Pi\tilde{\Lambda}z + c$, *where* $\tilde{\Lambda} \in \mathbb{R}^{d\times d}$ *is an invertible block-diagonal matrix with blocks of size* $p \times p$, $\Pi \in \mathbb{R}^{d\times d}$ *is a permutation matrix and* $c \in \mathbb{R}^d$ *is an offset.*

For the proof of the above theorem, refer to Section A.1 in the Appendix. From the above theorem, we gather that the learner can separate the perturbed blocks. However, the latent dimensions within the block are linearly entangled. In the ball agent interaction with 2-sparse perturbations, the above

theorem implies that the agent can separate each ball out but not their respective x and y coordinates. In the above theorem, we require the learner to know the group of each intervention (Assumption 6). In the Appendix Section A.2.2, we relax Assumption 6 and show that we can continue to achieve identification up to permutation and block diagonal transforms. However, we need a computationally expensive procedure that searches over subsets of latent dimensions to identify the dimensions impacted under the current intervention.

We briefly compare with Von Kügelgen et al. (2021), where the authors also establish block identification guarantees. In Von Kügelgen et al. (2021), the latent vector is divided into two parts – the content block and the style block. Across augmentations, style is varied and content is fixed. Von Kügelgen et al. leverage this invariance of the content across augmentations to learn the content block and not the style block. To summarize, invariance of content across different views is the key signal that is used to achieve identification. In our case, the perturbations act on different blocks of latents. In contrast to Von Kügelgen et al., we leverage sparsity of changes, i.e., we exploit both the varying part and the invariant part to identify all the distinct blocks and not just the content block.

3.1.2 Overlapping perturbations

In the previous section, we assumed that the blocks across different perturbations are non-overlapping. This section relaxes this assumption and allows the perturbation blocks to overlap. We start with a motivating example to show how overlapping perturbations can lead to stronger identification.

Consider the agent interacting with two balls, where $z = [z_{1x}, z_{1y}, z_{2x}, z_{2y}]$ describes the coordinates of the two balls. The agent perturbs the first ball and then perturbs the second ball. For the purpose of this example, assume that these perturbations satisfy the assumptions in Theorem 2. We obtain that the estimated position of each ball $\hat{z}_{ix/y}$ is linearly entangled w.r.t the true x and y coordinates. For the first ball we get $\hat{z}_{1x} = a_1 z_{1x} + a_2 z_{1y} + a_3$. We also have the agent perturb the x coordinates of the first and second ball together and then it does the same with the y coordinates. We apply Theorem 2 and obtain that the estimated x coordinates of each ball are linearly entangled. We write this as $\hat{z}_{1x} = b_1 z_{1x} + b_2 z_{2x} + b_3$. We take a difference of the two relations for $\hat{z}_{1x}$ to get

$$(a_1 - b_1) z_{1x} + a_2 z_{1y} - b_2 z_{2x} + a_3 - b_3 = 0 \tag{5}$$

Since the above has to hold for all z_{1x}, z_{1y}, z_{2x}, we get $a_1 = b_1$, $a_2 = 0$, $b_2 = 0$ and $a_3 = b_3$. Thus $\hat{z}_{1x} = a_1 z_{1x} + a_3$. Similarly, we can disentangle the rest of the balls.

We take the insights from the above example and generalize them below. Let us suppose that from the set of perturbations $\mathcal{I}$ we can construct at least two distinct subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ such that both subsets form a blockwise non-overlapping perturbation (see Definition 1). Perturbations in $\mathcal{I}_1$ ($\mathcal{I}_2$) partition $\{1, \cdots, d\}$ into blocks $\mathcal{B}_{\mathcal{I}_1}$ ($\mathcal{B}_{\mathcal{I}_2}$) respectively. It follows that there exists at least two blocks $\mathcal{B}^1 \in \mathcal{B}_{\mathcal{I}_1}$ and $\mathcal{B}^2 \in \mathcal{B}_{\mathcal{I}_2}$ such that $\mathcal{B}^1 \cap \mathcal{B}^2 \neq \emptyset$. From Theorem 2, we know that we can identify latents in block $\mathcal{B}^1$ and $\mathcal{B}^2$ up to affine transforms. In the next theorem, we show that we can identify latents in each of the blocks $\mathcal{B}^1 \cap \mathcal{B}^2$, $\mathcal{B}^1 \setminus \mathcal{B}^2$, $\mathcal{B}^2 \setminus \mathcal{B}^1$ up to affine transforms.

Assumption 7. *Each perturbation in $\mathcal{I}$ is p-sparse. The perturbations in each group span a p-dimensional space, i.e., $\forall q \in \mathcal{G}_{\mathcal{I}}$, $\mathsf{dim}\Big(\mathsf{span}\Big(\{\delta_i\}_{i \in q}\Big)\Big) = p$. There exist at least two distinct subsets of perturbations $\mathcal{I}_1 \subseteq \mathcal{I}$ and $\mathcal{I}_2 \subseteq \mathcal{I}$ that are both blockwise and non-overlapping.*

Theorem 3. *Suppose Assumptions 1, 3, 6 and 7 hold. Consider the subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ that satisfy Assumption 7. For every pair of blocks, $\mathcal{B}^1 \in \mathcal{B}_{\mathcal{I}_1}$ and $\mathcal{B}^2 \in \mathcal{B}_{\mathcal{I}_2}$, the encoder that solves equation (3) (where $\Delta^{'}$ is p-sparse, $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies latents in each of the blocks $\mathcal{B}^1 \cap \mathcal{B}^2$, $\mathcal{B}^1 \setminus \mathcal{B}^2$, $\mathcal{B}^2 \setminus \mathcal{B}^1$ up to invertible affine transforms.*

For the proof of the above theorem, refer to Section A.1 in the Appendix. From the above theorem, it follows that if blocks overlap at one latent only, then all such latents are identified up to permutation and scaling. We now construct an example to show the identification of all the latents under overlapping perturbations. Suppose we have a 4 dimensional latent. The set of all contiguous blocks of length 2 is given as follows $\{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}\}$. Different 2-sparse perturbations impact these blocks. Observe that every component between 1 to 4 gets to be the first element of a block exactly once and the last element of the block exactly once. As a result, each latent gets to be the only element at the intersection of two blocks. We apply Theorem 3 to this case and get that all the latents are identified up to permutation and scaling. We generalize this example below.

Assumption 8. *$\mathcal{B}_{\mathcal{I}}$ is a set of all the contiguous blocks of length p, where $p < d$. The perturbations in each block span a p dimensional space. Further, also assume that $d \bmod p = 0$.*

In the above assumption, we construct d contiguous blocks of length p. The construction ensures that each index in $\{1, \cdots, d\}$ forms the first element of exactly one block and last element of exactly one block. In Theorem 1 (Locatello et al., 2019) and in Theorem 5 (Lachapelle et al., 2022) a similar assumption is made that requires exactly one latent is at the intersection of multiple blocks. In the next theorem, we show that under the above assumption, we achieve identification up to permutation and scaling.

Theorem 4. *Suppose Assumptions 1, 3, 6 and 8 hold, then the encoder that solves the identity in equation (3) (where $\Delta^{'}$ is p-sparse, $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies true latents up to permutations and scaling, i.e., $\hat{z} = \Pi\Lambda z + c$, where $\Pi \in \mathbb{R}^{d\times d}$ matrix and $\Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix.*

For the proof of the above theorem, refer to Section A.1 in the Appendix. The total number of perturbations required in the above theorem is $p \times d$. If we plug $p = 1$, we recover Theorem 1 as a special case. The above result highlights that if the block lengths are larger, then we need to scale the number of perturbations accordingly by the same factor to achieve identification up to permutation and scaling. We assumed a special class of perturbations operating on contiguous blocks. In general, the total number of distinct blocks can be up to $\binom{d}{p}$. Suppose s distinct random blocks of length p are selected for perturbations. As s grows, we reach a point where each latent component is at the intersection of two blocks from different sets of blockwise non-overlapping perturbations. At that point, we identify all latents up to permutation and scaling.

Extensions In the discussion so far, we made some assumptions for ease of exposition. Next, we describe how to relax them. In the DGP in Assumption 1, the perturbations used are deterministic. In Section A.2.3, we extend the DGP in Assumption 1 to incorporate stochastic perturbations. Specifically, instead of $\tilde{z} \leftarrow z + \delta$ we consider a DGP where $\tilde{z} \leftarrow z + \delta + n$, where n is the noise vector added to the perturbation δ. We show that the key results presented in the paper extend provided the noise vector n follow the same sparsity pattern as δ. We also present experiments for the same model in Section A.3. In the DGP in Assumption 1, the perturbations used are independent of the value of z. Instead of $\tilde{z} \leftarrow z + \delta$ we consider a DGP given as $\tilde{z} \leftarrow z + m(z)$, where $m(\cdot)$ is a general non-linear perturbation map. In Section A.2.4, we show that the key results presented in the paper extend to this setting with non-linear perturbation mechanisms

4 Experiments

Data generation processes We conducted two sets of experiments – low-dimensional synthetic and image-based inputs – that follow the DGP in equation (24). In the low-dimensional synthetic experiments we experimented with two choices for $\mathbb{P}_Z$ a) uniform distribution with independent latents, b) normal distribution with latents that are blockwise independent (with block length $d/2$). We used an invertible multi-layer perceptron (MLP) (with 2 hidden layers) from Zimmermann et al. (2021) for g. We evaluated for latent dimensions $d \in \{6, 10, 20\}$. The training and test data size was 10000 and 5000 respectively. For the image-based experiments we used PyGame (Shinners, 2011)'s rendering engine for g and generated 64×64 pixel images that look like those shown in Figure 1. The coordinates of each ball, z_i, were drawn independently from a uniform distribution, $z_i \sim \mathcal{U}(0.1, 0.9)$. We varied the number of balls from 2 ($d = 4$) to 4 ($d = 8$). For these experiments, there was no fixed-size training set; instead the images are generated online and we trained to convergence. Because these problems are high dimensional, we only sampled a single perturbation for each image.

Loss function, architecture, evaluation metrics In all the experiments we optimized equation (4) with square error loss. The encoder f was an MLP with two hidden layers of size 100 for the low-dimensional synthetic experiments and a ResNet-18 (He et al., 2015) for the image-based experiments. Further training details such as the optimizers used, hyperparameters etc. are in the Appendix (Section A.3). We used the mean correlation coefficient (MCC) (Hyvarinen and Morioka, 2016) to verify the claims in Theorems 1 and 4. If MCC equals one, then the estimated latents identify true latents up to permutation and scaling. We extend MCC to blockwise MCC (BMCC) to verify the claims in Theorem 2. If BMCC equals one, then the estimated latents identify true latents up to permutation and block-diagonal transforms. Further details are in the Appendix (Section A.3).

Table 1: Comparing MCC and BMCC for non-overlapping perturbations. The number of perturbations applied for each example is given in parenthesis

d	p_Z	MCC C-wise (d)	MCC C-wise (1)	BMCC B-wise (d)	BMCC B-wise (1)
6	Normal	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.01
10	Normal	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	0.91 ± 0.02
20	Normal	0.99 ± 0.00	0.88 ± 0.03	0.99 ± 0.00	0.90 ± 0.01
6	Uniform	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.96 ± 0.04
10	Uniform	0.99 ± 0.00	0.99 ± 0.01	0.99 ± 0.00	0.81 ± 0.05
20	Uniform	0.99 ± 0.00	0.82 ± 0.02	0.85 ± 0.08	0.51 ± 0.04

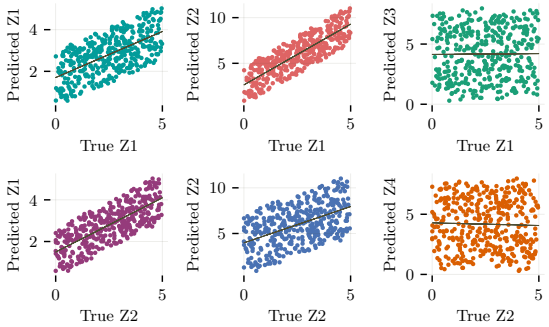

Figure 2: Illustrating blockwise dependence ($d = 10$).

Table 2: MCC for B-wise (overlap).

d	Distribution	MCC
6	Normal	0.95 ± 0.01
10	Normal	0.96 ± 0.01
20	Normal	0.99 ± 0.01
6	Uniform	0.86 ± 0.03
10	Uniform	0.88 ± 0.03
20	Uniform	0.81 ± 0.03

Non-overlapping perturbations We first conducted experiments with one-sparse perturbations, the set Δ consists of $m = d$ one-sparse perturbations that span a d dimensional space. In the context of the image experiments, these perturbations correspond to moving each ball individually along a single axis. The learner solves the identity in equation (3) using a set of random one-sparse perturbations $\Delta^{'}$ that span a d dimensional space. In Table 1, we used the low-dimensional synthetic data generating process to compare the effect of (i) applying all $m = d$ perturbations to each instance z (following the DGP in (24)), against a more practical setting (ii) where a perturbation is selected uniform at random from Δ and applied to each instance z. The results for (i) are shown in black and the results for (ii) are shown in gray font in the *C-wise* (componentwise) column in Table 1. We observed high MCCs in both settings. The results were similar in the more challenging image-based experiments (see Table 3, C-wise column) with MCC scores > 0.97 for all the settings that we tested, as expected given the results presented in Theorem 1.

In our next experiments, the set of perturbations Δ comprised of d 2-sparse non-overlapping perturbations that span a d dimensional space. We repeated the same synthetic experiments as above with one and d perturbations per instance. Under these assumptions we should expect to see that pairs of latents are separated blockwise but linearly entangled within the blocks (c.f. Theorem 2). We found this to be the case. The high BMCC numbers in Table 1 displayed under *B-wise* (blockwise) column (except for $d = 20$ and one perturbation per sample) show disentanglement between the blocks of latents. In Figure 2, the first two rows and columns show how the predicted latents corresponding to a block are correlated with their true counterpart (see Predicted Z_i vs True Z_i) and the other latent in the block (Predicted Z_1 vs True Z_2 and vice versa). The plots in the last column show that the predicted latents did not bear a correlation with a randomly selected latent from outside the block.

Overlapping perturbations In this section, we experimented with blocks of size two that overlap in order to conform with the setting described in Theorem 4. We used the same distributions as before and only changed the type of perturbations. The low-dimensional synthetic results are summarized in Table 2. The results were largely as expected, with a strong correspondence between the predicted and true latents reflected by high MCC values.

Table 3: Image experiments

d	MCC C-wise	MCC B-wise d	MCC B-wise $\binom{d}{p}$
4	0.994	0.710	0.864
6	0.981	0.817	0.912
8	0.975	0.866	0.934

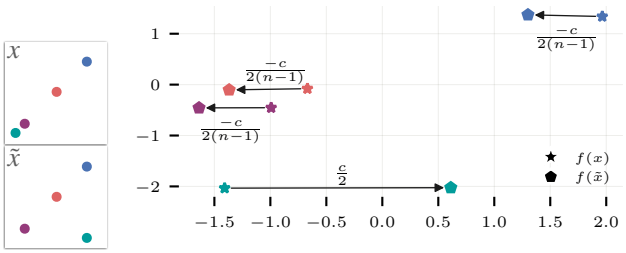

Figure 3: *(Left)* Results for the image-base experiments. *(Centre)* Example images in which the bottom left ball is shifted to the right. *(Right)* A trained encoder's predictions for the two images shown in *(centre)*. The green ball prediction shifts right by $\approx \frac{c}{2}$ and the other balls left by $\approx \frac{c}{2(n-1)}$. For further illustrations, refer to the animations in the supplement.

On the image datasets (see Table 3), we found that the MCC scores depended on both the number of balls and how the blocks were selected. We compared two strategies for selecting blocks of latents to perturb: either select uniformly from all adjacent pairs $\mathcal{I} = \{(i \bmod d, i+1 \bmod d)\}$ (d blocks), or uniformly from all combinations of latent indices, $\mathcal{I} = \{(i,j) : i \in \{1,\dots,d\}, j > i\}$ ($\binom{d}{2}$ blocks). The latter lead to higher MCC scores (ranging from 0.86 to 0.93) as it placed more constraints on the solution space. The dependence on the number of balls is more surprising. To investigate the implied entanglement from the lower MCC scores, we evaluated trained encoders on images where we kept $n_{\text{balls}} - 1$ balls in a fixed location and moved one of the balls (see Section A.3 in the Appendix for example images). If the coordinates were perfectly disentangled, the encoder should predict no movement for static balls. When the moving ball shifted by c units, the predicted location of the static balls shifted by $\approx \frac{-c}{2(n_{\text{balls}}-1)}$ and the moving ball shifted $\approx \frac{c}{2}$ units. We further verified this claim and ran blockwise experiments with $n_{\text{balls}} = 10$ balls ($d = 20$) and got MCC scores of 0.930 and 0.969 for d and $\binom{d}{2}$ blocks respectively. In the Appendix (Section A.3), we show that this solution is a stationary point, and we achieve a perfect MCC of one when $n_{\text{balls}} = \infty$. Finally, the codes to reproduce the experiments presented above can be found at `https://github.com/ahujak/WSRL`.

5 Discussion and limitations

Our work presents the first systematic analysis of the role of sparsity in achieving latent identification under unknown arbitrary latent distributions. We assume that every sample (or at least every neighborhood of a sample) experiences the same set of perturbations. A natural question is how to extend our results to settings where this assumption may not hold. Data augmentation provides a rich source of perturbations; our results cover translations, but extending them to other forms of augmentation is an important future direction. We followed the literature on non-linear ICA (Hyvarinen et al., 2019) and made two assumptions – i) the map g that mixes latents is injective, and ii) the dimension of the latent d is known. We believe future works should aim to relax these assumptions. In reinforcement learning (RL) environments, the effects of actions can often be sparse. Therefore, we believe illustrating the efficacy of the proposed approach in RL environments (Ahmed et al., 2020) is an important direction to further the case of the proposed theory and methods in real-world applications.

6 Acknowledgements

We thank Sébastien Lachapelle and Anirudh Goyal for insightful discussions. Kartik Ahuja acknowledges the support from the IVADO postdoctoral fellowship funding program. Jason Hartford acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and Recursion Pharmaceuticals. Yoshua Bengio acknowledges the support from CIFAR, Samsung and IBM.

References

Ahmed, O., Träuble, F., Goyal, A., Neitz, A., Bengio, Y., Schölkopf, B., Wüthrich, M., and Bauer, S. (2020). Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *arXiv preprint arXiv:2010.04296*.

Ahuja, K., Hartford, J., and Bengio, Y. (2022). Properties from mechanisms: an equivariance perspective on identifiable representation learning. In *International Conference on Learning Representations*.

Brehmer, J., De Haan, P., Lippe, P., and Cohen, T. (2022). Weakly supervised causal representation learning. *arXiv preprint arXiv:2203.16437*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dittadi, A., Träuble, F., Locatello, F., Wüthrich, M., Agrawal, V., Winther, O., Bauer, S., and Schölkopf, B. (2020). On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407*.

Geirhos, R., Jacobsen, J., Michaelis, C., Zemel, R. S., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780.

Gondal, M. W., Wuthrich, M., Miladinovic, D., Locatello, F., Breidt, M., Volchkov, V., Akpo, J., Bachem, O., Schölkopf, B., and Bauer, S. (2019). On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems*, 32.

Goyal, A., Didolkar, A., Ke, N. R., Blundell, C., Beaudoin, P., Heess, N., Mozer, M. C., and Bengio, Y. (2021). Neural production systems. *Advances in Neural Information Processing Systems*, 34:25673–25687.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. (2019). Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*.

Gresele, L., Rubenstein, P. K., Mehrjou, A., Locatello, F., and Schölkopf, B. (2020). The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. In *Uncertainty in Artificial Intelligence*, pages 217–227. PMLR.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. (2017). Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 1480–1490. PMLR.

Hyvarinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in Neural Information Processing Systems*, 29.

Hyvarinen, A. and Morioka, H. (2017). Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR.

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

Hyvarinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020a). Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR.

Khemakhem, I., Monti, R., Kingma, D., and Hyvarinen, A. (2020b). Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778.

Kim, H. and Mnih, A. (2018). Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Klindt, D. A., Schott, L., Sharma, Y., Ustyuzhaninov, I., Brendel, W., Bethge, M., and Paiton, D. (2021). Towards nonlinear disentanglement in natural data with temporal sparse coding. In *International Conference on Learning Representations*.

Lachapelle, S., Rodriguez, P., Sharma, Y., Everett, K. E., PRIOL, R. L., Lacoste, A., and Lacoste-Julien, S. (2022). Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *First Conference on Causal Learning and Reasoning*.

Lippe, P., Magliacane, S., Löwe, S., Asano, Y. M., Cohen, T., and Gavves, E. (2022). Citris: Causal identifiability from temporal intervened sequences. *arXiv preprint arXiv:2202.03169*.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. (2020). Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR.

Miladinović, Đ., Gondal, M. W., Schölkopf, B., Buhmann, J. M., and Bauer, S. (2019). Disentangled state space representations. *arXiv preprint arXiv:1906.03255*.

Mityagin, B. (2015). The zero set of a real analytic function. *arXiv preprint arXiv:1512.07276*.

Pearl, J. (2009). *Causality*. Cambridge university press.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634.

Schölkopf, B. and von Kügelgen, J. (2022). From statistical to causal learning. *arXiv preprint arXiv:2204.00607*.

Shinners, P. (2011). Pygame. `http://pygame.org/`.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

Von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). Self-supervised learning with data augmentations provably isolates content from style. *Advances in Neural Information Processing Systems*, 34.

Yao, W., Sun, Y., Ho, A., Sun, C., and Zhang, K. (2021). Learning temporally causal latent processes from general temporal data. *arXiv preprint arXiv:2110.05428*.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. (2021). Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Each of the main contributions listed in the introduction correspond to a theorem in Section 3 and the final claim corresponds to the experiments described in Section 4.
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not foresee any potential negative societal impacts specific to this work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the Appendix (Section A.1) in the Supplementary Material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We have provided the codes in the Supplementary Material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the Appendix (Section A.3).
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] For the low-dimensional synthetic experiments, which were computationally less demanding we ran five seeds. For image-based experiments, the run time for each case was on average 12 hours. Since we had several such cases, running several seeds was not feasible given the compute we had at our disposal.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the Appendix (Section A.3).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See the Appendix (Section A.3).
   (b) Did you mention the license of the assets? [Yes] See the Appendix (Section A.3).
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The codes can be found at `https://github.com/ahujak/WSRL`.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

We organize the Appendix into three sections. In Section A.1, we provide the proofs to all the propositions and the theorems. In Section A.2, we discuss how some of the proposed results can be extended. In Section A.3, we provide supplementary materials for the experiments.

A.1 Proofs

We restate all the propositions and the theorems below for convenience. In the proofs that follow, we use Δ ($\Delta^{'}$) to denote the set of perturbations and the matrix of perturbations interchangeably (their usage is clear from the context). We start with the proof of Proposition 1, which follows the proof technique from Ahuja et al. (2022).

Proposition 2. *(Restatement of Proposition 1) If Assumptions 1, 2, and 3 hold, then the encoder that solves equation* (3) *(with $\Delta^{'}$ s.t. $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies true latents up to an invertible affine transform, i.e. $\hat{z} = Az + c$, where $A \in \mathbb{R}^{d\times d}$ is an invertible matrix and $c \in \mathbb{R}^d$ is an offset.*

Proof. We simplify the identity in equation (3) as follows.

$$\begin{aligned} f(x) + \delta_i^{'} &= f(\tilde{x}_k) \\ f \circ g(z) + \delta_i^{'} &= f \circ g(\tilde{z}_k) \\ a(z) + \delta_i^{'} &= a(\tilde{z}) \\ a(z) + \delta_{i'}^{'} &= a(z + \delta_i) \end{aligned} \tag{6}$$

In the above simplification, we use the following observation. Since x and $\tilde{x}_k$ are generated from g and g is injective, we can substitute $x = g(z)$ and $\tilde{x}_k = g(\tilde{z}_k)$, where $\tilde{z}_k = z + \delta_k$.

For simplicity denote the last line in above equation (6) as

$$a(z) + b^{'} = a(z + b). \tag{7}$$

We take gradient of the function in the LHS and RHS of the above equation (7) separately w.r.t z. Consider the jth component of $a(z + b)$ denoted as $a_j(z + b)$. We first take the gradient of $a_j(z + b)$ w.r.t z

$$\nabla_z a_j(z + b) = \Big(\frac{dy}{dz}\Big)^{\mathsf{T}} \nabla_y a_j(y), \tag{8}$$

where $y = z + b$, $\nabla_y a_j(y)$ is the gradient of a_j w.r.t y and $\frac{dy}{dz}$ denotes the Jacobian of y w.r.t z. We simplify the above further to get

$$\nabla_z a_j(z + b) = \nabla_y a_j(y) = \nabla_y a_j(z + b). \tag{9}$$

We can write the above for each component of a as follows.

$$\begin{aligned} &\big[\nabla_z a_1(z + b), \cdots, \nabla_z a_d(z + b)\big] = \big[\nabla_y a_1(z + b), \cdots, \nabla_y a_d(z + b)\big] \\ &= [\nabla_y a_1(z + b), \cdots, \nabla_y a_d(z + b)] = J^{\mathsf{T}}(z + b), \end{aligned} \tag{10}$$

where $J(z + b)$ is the Jacobian of a computed at $z + b$. We equate the gradient of LHS and RHS in (7) to obtain

$$a(z + b) = a(z) + b^{'} \implies J^{\mathsf{T}}(z + b) - J^{\mathsf{T}}(z) = 0. \tag{11}$$

Consider row j of this identity. For each $z \in \mathbb{R}^d$

$$\nabla a_j(z+b) - \nabla a_j(z) = 0 \implies \begin{bmatrix} \nabla_1^2 a_j(\theta_1) \\ \nabla_2^2 a_j(\theta_2) \\ \vdots \\ \nabla_d^2 a_j(\theta_d) \end{bmatrix} (b) = 0 \tag{12}$$

where $\nabla^2 a_j$ is the Hessian of a_j and $\nabla_k^2 a_j(\theta_k)$ corresponds to the kth row of the Hessian matrix. Note that in the above expansion there is a different θ_k for each row (mean value theorem applied to each component of ∇a_j yields a different point θ_k on the line joinining $\tilde{z}$ and $\tilde{z}+b$). From Assumption 3 it follows that $\nabla_k^2 a_j(\theta_k)(b) = 0$ over a set with non-zero measure. Since a_j is analytic $\nabla_k^2 a_j(z)(b)$ is also analytic (each component of the vector is a weighted sum of analytic functions). Therefore, we can conclude that $\nabla_k^2 a_j(z)(b) = 0$ for all z (follows from Mityagin (2015)). We can make the same argument for each component k and conclude that $\nabla^2 a_j(z)(b) = 0$. From the identity in equation (3), it follows that $\nabla^2 a_j(z)(\delta_j) = 0$ for all $j \in \{1, \cdots, d\}$ and since the set $\Delta = \{\delta_1, \cdots, \delta_d\}$ is linearly independent $\nabla^2 a_j(z) = 0$ for all z. This implies $a(z) = Az + c$.

We substitute this in equation (6) to get $A\Delta = \Delta^{'}$, where Δ is the matrix of true perturbations and $\Delta^{'}$ is the matrix of guessed perturbations (recall we stated above that we use $\Delta, \Delta^{'}$ as sets and matrices interchangeably). We now need to show that A is invertible. Suppose A was not invertible, which implies the rank of $A \leq n-1$. Following Assumption 2, rank of Δ is n. Note that rank of $\Delta^{'}$ is also n. Note that if $E = FG$, where E, F, G are three matrices, then $\mathsf{rank}(E) \leq \min\{\mathsf{rank}(F), \mathsf{rank}(G)\}$. Following this identity, $\mathsf{rank}(\Delta^{'}) \leq n-1$, which is a contradiction. Therefore, A has to be invertible. This completes the proof.

□

Theorem 5. *(Restatement of Theorem 1) If Assumptions 1-4 hold and the number of perturbations per example equals the latent dimension, $m = d$, then the encoder that solves equation (3) (with $\Delta^{'}$ as one-sparse and $\mathsf{dim}\left(\mathsf{span}\left(\Delta^{'}\right)\right) = d$) identifies true latents up to permutation and scaling, i.e. $\hat{z} = \Pi\Lambda z + c$, where $\Lambda \in \mathbb{R}^{d\times d}$ is an invertible diagonal matrix, $\Pi \in \mathbb{R}^{d\times d}$ is a permutation matrix and c is an offset.*

Proof. Since Assumptions 1, 2, and 3 hold, we can use Proposition 1 to obtain that any solution to equation (3) achieves affine identification guarantees w.r.t the true latents, i.e. $\hat{z} = Az + c$, where $\hat{z} = f(x)$, z is the inverse image of x ($x = g(z)$), $A \in \mathbb{R}^{d\times d}$ is an inverible matrix and $c \in \mathbb{R}^d$ is the offset vector.

Define $e_i = [0, \cdots, 1_i, \cdots 0]$ as the vector, which takes a value 1 at i^{th} component and 0 everywhere else. Without loss of generality set of true perturbations is $\Delta = \{b_1 e_1, \cdots, b_d e_d\}$. Note that all b_i's are non-zero as the span of Δ has a dimension d.

Denote the corresponding set of guesses from the agent are $\Delta^{'} = \{c_1 e_{\pi(1)}, \cdots, c_d e_{\pi(d)}\}$, where $\pi : \{1, \cdots, d\} \to \{1, \cdots, d\}$ is a map used by the agent to guess the coordinate impacted by the perturbation. Note that since $\Delta^{'}$ spans d dimensions π has to be a bijection c_j's are non-zero as the span of $\Delta^{'}$.

Take $b_j e_j \in \Delta$ and the corresponding guess $c_k e_k$ and substitute it in the relation $\hat{z} = Az + c$ to get

$$\begin{aligned} &\hat{z} = Az + c, \\ &\hat{z} + c_k e_k = A(z + b_j e_j) + c, \\ &c_k e_k = b_j A e_j, \\ &\frac{c_k}{b_j} e_k = A e_j. \end{aligned} \tag{13}$$

Since π is a bijection, for every j there is a unique k in the RHS above. From the above equation, we gather that the jth column of A is $\frac{c_k}{b_j} e_k$. We apply this to all the columns and conculde that $\hat{z} = \Pi\Lambda z + c$, where Λ is a diagonal matrix and Π is a permutation matrix decided based on the bijection π (($\Pi_k = e_{\pi(k)}$), where Π_k is the kth colum of the matrix). □

Theorem 6. *(Restatement of Theorem 2) If Assumptions 1-3, 5, 6 hold, then the encoder that solves equation (3) (where $\Delta^{'}$ is p-sparse, $\dim\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies true latents up to permutation and block-diagonal transforms, i.e. $f(x) = \hat{z} = \Pi\tilde{\Lambda}z + c$, where $\tilde{\Lambda} \in \mathbb{R}^{d\times d}$ is an invertible block-diagonal matrix with blocks of size $p \times p$, $\Pi \in \mathbb{R}^{d\times d}$ is a permutation matrix and $c \in \mathbb{R}^d$ is an offset.*

Proof. Since Assumptions 1, 2, and 3 hold, we can use Proposition 1 to obtain that any solution to equation (3) achieves affine identification guarantees w.r.t the true latents, i.e. $\hat{z} = Az + c$, where $\hat{z} = f(x)$, z is the inverse image of x ($x = g(z)$), $A \in \mathbb{R}^{d\times d}$ is an inverible matrix and $c \in \mathbb{R}^d$ is the offset vector.

We start the proof by assuming that the agent knows the blocks that are impacted under each perturbation, i.e., for each $i \in \mathcal{I}$, the agent knows the block of the latents that are impacted denoted as $\mathcal{A}_i$. We relax this assumption later.

Following Assumption 5, we know that perturbations are p-sparse, blockwise and non-overlapping. Without loss of generality, we can assume that the different groups on which perturbations in Δ act are given as $\{1, \cdots, p\}$, $\{p+1, \cdots, 2p\}$ and so on. Consider a perturbation δ_i, which belongs to Group 1 and impacts the latents $\{1, \cdots, p\}$. For this perturbation, the agent selects $\delta_i^{'}$, which shares the same sparsity pattern. Therefore, the first p elements of $\delta_i^{'}$ and δ_i are both non-zero and the rest of the elements are zero. Under these assumptions, we can write the relationship between true and guessed perturbations as follows.

$$\begin{aligned} \hat{z} + \delta_i^{'} &= A(z + \delta_i) + c \\ \delta_i^{'} &= A\delta_i \end{aligned} \tag{14}$$

Denote the first p elements of row k of matrix A as $a_k[1:p]$ and the first p elements of the vector δ_i as $\delta_i[1:p]$. For $k > p$, we use the equation (14) to get $a_k[1:p]^{\mathsf{T}}\delta_i[1:p] = 0$.

For all perturbations in Group 1, we can write the same condition, i.e., $a_k[1:p]^{\mathsf{T}}\delta_i[1:p] = 0$. Since the perturbations in Group 1 span a p dimensional space (following Assumption 2, 5), we get that $a_k[1:p] = 0$. Therefore, $a_k[1:p] = 0$ for all $k > p$.

Let q denote the number of perturbations in Group 1, where $q \geq p$. For all $k \leq p$ we can solve for the first $p \times p$ block using the perturbations guessed by the agent and the true perturbations in Group 1. Denote the first $p \times p$ block of A as $A[1:p, 1:p]$ and the first p components of the q perturbations in Group 1 as $\Delta[1:p, 1:q]$. Similarly, the first p components of the q perturbations guessed by the learner is denoted as $\Delta^{'}[1:p, 1:q]$. We now need to show that the block $A[1:p, 1:p]$ is invertible. From the above equation in (14), we get

$$A[1:p, 1:p]\Delta^{'}[1:p, 1:q] = \Delta[1:p, 1:q]$$

.

where q is the number of perturbations in Group 1.

Since rank of $\Delta[1:p, 1:q]$ and $\Delta^{'}[1:p, 1:q]$ is p, the rank of $A[1:p, 1:p]$ cannot be less than p or else it would lead to a contradiction. This shows that $A[1:p, 1:p]$ is invertible. We derived the properties of the first p columns of matrix A. For Group 2, we similarly obtain that $A[p+1:2p, p+1:2p]$ is an invertible matrix and rest of the elements in columns $\{p+1, \cdots, 2p\}$ are zero. Due to symmetry of the setting, we can apply the same argument to all the other blocks as well. Therefore, we conclude that A is block-diagonal and invertible. This leads to the conclusion that $\hat{z} = \tilde{\Lambda}z + c$, where $\tilde{\Lambda} \in \mathbb{R}^{d\times d}$ and $c \in \mathbb{R}^d$.

So far we assumed that the agent knows how the interventions in $\{1, \cdots, m\}$ impact the blocks $\{\mathcal{A}_1, \cdots, \mathcal{A}_m\}$. Under Assumption 6, the agent knows the groups of the perturbations only. For perturbations $\{\delta_1, \cdots, \delta_p\}$ in Group 1 that impact $\{1, \cdots, p\}$, the agent guesses $\{\delta_1^{'}, \cdots, \delta_p^{'}\}$. Note that perturbations in $\{\delta_1^{'}, \cdots, \delta_p^{'}\}$ impact the same block of length p with indices given as $\{\alpha_1, \cdots, \alpha_p\}$. Recall the first p elements of row k of matrix A and vector δ_i are denoted as $a_k[1:p]$ and $\delta_i[1:p]$ respectively. There exist $d-p$ rows in A for which we get $a_k[1:p]^{\mathsf{T}}\delta_i[1:p] = 0$. Thus $a_k[1:p] = 0$ for

all these rows. The first p elements of remaining p form a square matrix denoted as $A[\alpha_1 : \alpha_p, 1 : p]$, where $\{\alpha_1, \cdots, \alpha_p\}$ are the indices guessed by the agent for the block corresponding to Group 1. $A[\alpha_1 : \alpha_p, 1 : p]$ satisfies

$$A[\alpha_1 : \alpha_p, 1 : p]\Delta[1 : p, 1 : q] = \Delta^{'}[\alpha_1 : \alpha_p, 1 : q]$$

where $\Delta^{'}[\alpha_1 : \alpha_p, 1 : q]$ is the matrix of non-zero components of the q perturbation vectors that the agent guesses. Using the same argument as above, we can argue that $A[\alpha_1 : \alpha_p, 1 : p]$ is invertible. We have derived the properties of first p columns of A. We apply the same argument to other groups as well. Since the agent selects a set of unique p indices for each group, we obtain that the matrix A can be factorized as a permutation matrix times a block diagonal matrix. The first p rows of the permutation matrix with index $\{1, \ldots, p\}$ have ones at locations $\{\alpha_1, \cdots, \alpha_p\}$ and so on. As a result, we get that $\hat{z} = \Pi\tilde{\Lambda}z + c$

This completes the proof. □

Theorem 7. *(Restatement of Theorem 3) Suppose Assumptions 1, 3, 6 and 7 hold. Consider the subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ that satisfy Assumption 7. For every pair of blocks, $\mathcal{B}^1 \in \mathcal{B}_{\mathcal{I}_1}$ and $\mathcal{B}^2 \in \mathcal{B}_{\mathcal{I}_2}$, the encoder that solves equation (3) (where $\Delta^{'}$ is p-sparse, $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies latents in each of the blocks $\mathcal{B}^1 \cap \mathcal{B}^2$, $\mathcal{B}^1 \setminus \mathcal{B}^2$, $\mathcal{B}^2 \setminus \mathcal{B}^1$ up to invertible affine transforms.*

Proof. Following Assumption 7, we know that there exists at least two subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ that satisfy blockwise non-overlapping perturbations. Like in the previous proof, we start this proof also with the case where the agent knows the exact sparsity pattern in the perturbations. We relax this assumption in a bit. Consider a block $\mathcal{B}^1 = \{\beta_1, \cdots, \beta_p\}$ impacted by the perturbations in $\mathcal{I}_1$. Since $\mathcal{I}_1$ is blockwise and non-overlapping, we can follow the analysis in the first part of the previous theorem to get $[\hat{z}_{\beta_1}, \cdots, \hat{z}_{\beta_p}]$ is an invertible affine transform of $[z_{\beta_1}, \cdots, z_{\beta_p}]$. Hence, the latents in each of the blocks $\mathcal{B}^1 \in \mathcal{G}_{\mathcal{I}_1}$ are identified up to an afffine transform. Similarly, each block $\mathcal{B}^2 \in \mathcal{G}_{\mathcal{I}_2}$ is identified up to an affine transform. Consider an element $i \in \mathcal{B}^1 \cap \mathcal{B}^2$. $\hat{z}_i$ can be expressed as an affine transform of two different blocks of latents z^1 and z^2. z^1 and z^2 share some components, we denote them as z^{12}. The components exclusive to z^1 (z^2) is denoted as z^{11} (z^{22}).

We write this condition as follows.

$$\begin{aligned}
&\hat{z}_i = a_1^{\mathsf{T}} z^{11} + a_2 z^{12} + a_3 \\
&\hat{z}_i = b_1^{\mathsf{T}} z^{22} + b_2 z^{12} + b_3 \\
&a_1^{\mathsf{T}} z^{11} + (a_2 - b_2)^{\mathsf{T}} z^{12} - b_1^{\mathsf{T}} z^{22} = b_3 - a_3
\end{aligned} \tag{15}$$

If $[a_1, a_2 - b_2, b_1]$ is non-zero, i.e., at least one element is non-zero, then the range of LHS is $\mathbb{R}$. But the range of the RHS is a constant. Therefore, for the above to be true $[a_1, a_2 - b_2, b_1] = 0$ and that implies $a_3 = b_3$. As a result, the linear entanglement is now confined to only the intersecting variables z^{12}. We can repeat this argument for all elements in $\mathcal{B}^1 \cap \mathcal{B}^2$.

In the proof so far, we relied on the assumption that the components impacted by each intervention $i \in \mathcal{I}$ are known. We now relax this assumption and work with assumption that was used in the previous theorem (Assumption 6), which states that the agent knows the group label of each perturbation.

Consider the latents in the block $\mathcal{B}^1 \in \mathcal{G}_{\mathcal{I}_1}$, which we denote as z^1. We apply Theorem 2 to this block. Let the set of estimated latents that affine identify $\mathcal{B}^1$ be $\hat{z}^1 = [\hat{z}_{\alpha_1}, \cdots, \hat{z}_{\alpha_p}]$, where $\{\alpha_1, \cdots, \alpha_p\}$ is the set of indices in $\hat{z}$. We write this as $[\hat{z}_{\alpha_1}, \cdots, \hat{z}_{\alpha_p}] = A^1 z^1 + c^1$. $\tilde{\mathcal{B}}^1$ denotes the set of remaining latents not in the block $\mathcal{B}^1$. We denote the latents in the block $\tilde{\mathcal{B}}^1$ as z_c^1. Following Theorem 2, we get that the remaining elements of $\hat{z}$ other than $\hat{z}^1$, which we denote as $\hat{z}_c^1$, affine identify the latents z_c^1 in the block $\tilde{\mathcal{B}}^1$.

Similarly, consider the latents in the group $\mathcal{B}^2 \in \mathcal{G}_{\mathcal{I}_2}$ denoted as z^2. $\hat{z}^2 = [\hat{z}_{\beta_1}, \cdots, \hat{z}_{\beta_p}]$ denotes the latents that affine identify z^2. $\tilde{\mathcal{B}}^2$ is the set of remaining latents. The remaining elements of $\hat{z}$ other than $\hat{z}^2$ are denoted as $\hat{z}_c^2$. $\hat{z}_c^2$ affine identifies the latents in the block $\tilde{\mathcal{B}}^2$, which are denoted as z_c^2.

The latents $z^{11} \in \mathcal{B}^1 \setminus \mathcal{B}^2$, $z^{12} \in \mathcal{B}^1 \cap \mathcal{B}^2$, and $z^{22} \in \mathcal{B}^2 \setminus \mathcal{B}^1$. Consider a latent that is shared between $\hat{z}^1$ and $\hat{z}^2$. Using the same analysis from equation (15), we show that such an element puts a non-zero weight only on z^{12}. Therefore, all the latents shared between $\hat{z}^1$ and $\hat{z}^2$ have a non-zero weight on z^{12}. Now consider a component of $\hat{z}^1$ denoted as $\hat{z}_{\alpha_k}$, which is not present in $\hat{z}^2$. We can write the affine identification condition as

$$\hat{z}_{\alpha_k} = c_1^{\mathsf{T}} z^{11} + c_2^{\mathsf{T}} z^{12} + c_3 \tag{16}$$

We selected $\hat{z}_{\alpha_k}$, which is not present in $\hat{z}^2$. Since $\hat{z}_{\alpha_k}$ is in $\hat{z}^2_c$, we have

$$\hat{z}_{\alpha_k} = d_1^{\mathsf{T}} z_c^2 + d_3 \tag{17}$$

If we take a difference of the above two equations (16) and (17), we get that c_2 is equal to zero (see the justification below).

$$d_1^{\mathsf{T}} z_c^2 + d_3 - c_1^{\mathsf{T}} z^{11} - c_2^{\mathsf{T}} z^{12} - c_3 = 0 \tag{18}$$

Note that there is no term associated with z^{12} in equation (17) as z_c^2 is the set of elements not in z^2. Now since the above equation (17) holds for all z, we get $c_2 = 0$.

From the above analysis we conclude that the latents in $\hat{z}^1$ can be divided into two parts i) the latents that are shared with $\hat{z}^2$; these latents are an affine transform of z^{12}, ii) the latents that are not shared with $\hat{z}^2$; these latents are an affine transform of z^{11}. We write this condition as

$$\hat{z}^1 = \begin{bmatrix} e_1 & 0 \\ 0 & e_2 \end{bmatrix} \begin{bmatrix} z^{11} \\ z^{12} \end{bmatrix} + e_3 \tag{19}$$

Similarly, we get

$$\hat{z}^2 = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix} \begin{bmatrix} z^{22} \\ z^{12} \end{bmatrix} + f_3 \tag{20}$$

We have already discussed above that $f_2 = e_2$ and the latter half of f_3 corresponding to z^{12} is equals corresponding half of e_3.

From the previous theorem, we know that the matrices in the above equations (19) and (20) are invertible. Thus if z^{12} has q components, then e_2 is an invertible $q \times q$ matrix and e_1 is an invertible $p - q \times p - q$ matrix. This establishes the affine identification of the smaller blocks obtained by intersection of the blocks across two sets of non-overlapping blockwise perturbations. This completes the proof. □

Theorem 8. *(Restatement of Theorem 4) Suppose Assumptions 1, 3, 6 and 8 hold, then the encoder that solves the identity in equation* (3) *(where $\Delta^{'}$ is p-sparse, $\mathsf{dim}\Big(\mathsf{span}\big(\Delta^{'}\big)\Big) = d$) identifies true latents up to permutations and scaling, i.e., $\hat{z} = \Pi\Lambda z + c$, where $\Pi \in \mathbb{R}^{d\times d}$ matrix and $\Lambda \in \mathbb{R}^{d\times d}$ is a diagonal matrix.*

Proof. In the above theorem, we use a set of perturbations $\mathcal{I}$ that are p-sparse and satisfy the following property. The first $d - (p-1)$ blocks are $\{i, \cdots, i+p-1\}$ from $i = 1$ to $i = d - p + 1$. The remaining $p - 1$ blocks are $\{i, \cdots, (i+p-1) \mod (d+1) + 1\}$ from $i = d - p + 2$ to d. In the d blocks each latent component i is the first element of the block exactly once and also the last component exactly once.

Construct a partition of perturbations $\mathcal{I}_1$ with continguous blocks $\{k, \cdots, k+p-1\}$ and so on. Similarly, construct a partition of perturbations $\mathcal{I}_2$ $\{k-(p-1), \cdots, k\}$ and so on. Note that k is the first element of its block in $\mathcal{I}_1$ and it is the last element of its block in $\mathcal{I}_2$. We can apply the Theorem 3 to conclude that k^{th} component is identified up to scaling and permutation error. We can state the same for all the components. This completes the proof. □

A.2 Extensions

A.2.1 Extending Theorem 1

In Theorem 1, we assumed that the number of perturbations m is equal to the number of latent dimensions d. Suppose the number of perturbations is larger than d. We subsample d distinct perturbation indices from $\{1, \cdots, m\}$. We solve the identity with the data generated under sub-sampled perturbations in equation (3) with one-sparse guesses. If a solution exists, then we can continue to use the analysis in Theorem 1. If a solution does not exist, we sub-sample again and solve the identity in equation (3) until we find a solution.

In Theorem 1, we assumed that the learner knows that the perturbations in equation (24) are one sparse. Suppose the learner instead guesses that the perturbations are p-sparse, where $1 < p < d$ and $d \bmod p = 0$. In this case, we can use analysis similar to Theorem 2 and guarantee blockwise identification, where the blocks are of size $p \times p$.

A.2.2 Extending Theorem 2

In this section, we discuss how we can relax Assumption 6. We first show how to extend Theorem 2 to this setting. Later we describe how the same ideas can be extended to setting presented in Theorem 4. In this section, we propose a sparsity test, which would be used to test if the encoder learned is p-sparse or not. For the rest of the section, we assume that $d \bmod p = 0$. The number of blocks is $r = \frac{d}{p}$.

We take each sample point $(x, \tilde{x}_1, \cdots, \tilde{x}_m)$ and divide it into two parts. We keep the first d perturbations in one set $(\tilde{x}_1, \cdots, \tilde{x}_d)$ to train the encoder and we use the remaining $(\tilde{x}_{d+1}, \cdots, \tilde{x}_m)$ for checking sparsity. We refer to the first d perturbations as training perturbations and the remaining perturbations as validation perturbations.

Assumption 9. *$\{\delta_i\}_{i=1}^{d}$ is the set of training perturbations, which are p-sparse, blockwise and non-overlapping. $\{\delta_i\}_{i=d+1}^{m}$ is the set of validation perturbations, which are p-sparse, blockwise and non-overlapping. The training perturbations span $\mathbb{R}^d$.*

Consider d perturbations and represent them as follows Δ_d

$$\Delta_d = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \cdots, \Delta_{1r} \\ \Delta_{12} & \Delta_{22} & \cdots, \Delta_{2r} \\ & \vdots & \\ \Delta_{r1} & \Delta_{22} & \cdots, \Delta_{rr} \end{bmatrix} \tag{21}$$

where Δ_{ij} is $p \times p$ matrix. Without loss of generality under Assumption 9, we can write Δ_d as a blockdiagonal matrix such that all matrices $\Delta_{ij} = 0$ for all $i \neq j$.

We write the inverse of Δ_d as

$$\Delta_d^{-1} = \begin{bmatrix} \tilde{\Delta}_{11} & \tilde{\Delta}_{12} & \cdots, \tilde{\Delta}_{1r} \\ \tilde{\Delta}_{12} & \tilde{\Delta}_{22} & \cdots, \tilde{\Delta}_{2r} \\ & \vdots & \\ \tilde{\Delta}_{r1} & \tilde{\Delta}_{22} & \cdots, \tilde{\Delta}_{rr} \end{bmatrix} \tag{22}$$

Assumption 10. *Each element in the matrix along the diagonal of Δ_d^{-1} is non-zero, i.e., $\forall k \in \{1, \cdots, \frac{d}{p}\}, \forall i, j \in \{1, \cdots, p\}, \tilde{\Delta}_{kk}[i, j] \neq 0$*

Assumption 11. *$\Delta^{'}$ consists of d perturbations, which are p-sparse, blockwise and non-overlapping.*

We write the d corresponding perturbations that the agent guesses in the form of a matrix as

$$\Delta_d^{'} = \begin{bmatrix} \Delta_{11}^{'} & \Delta_{12}^{'} & \cdots, \Delta_{1r}^{'} \\ \Delta_{12}^{'} & \Delta_{22}^{'} & \cdots, \Delta_{2r}^{'} \\ & \vdots & \\ \Delta_{r1}^{'} & \Delta_{22}^{'} & \cdots, \Delta_{rr}^{'} \end{bmatrix} \tag{23}$$

where $\Delta^{'}_{ij}$ is $p \times p$ matrix.

Define an indicator mask of the underlying matrix $\Delta^{'}$; it takes a value one wherever there is a non-zero entry and zero otherwise. Define the set of all the masks for $\Delta^{'}$ that satisfy the above assumption (Assumption 11) as $\mathcal{M} = \{1, \cdots, n_{\mathsf{masks}}\}$. Now under the Assumption 9, we get that the validation perturbations are blockwise and non-overlapping as well (though they are not required to span the blocks). We now formalize a simple iterative procedure in which the learner searches over masks that are compliant with the assumption above (Assumption 9)

In the sparsity test, we take a trained encoder and check if for each of the perturbations in the validation set, it ensures only p components change. If for any perturbation more than p estimated components change, then the encoder fails the test.

Joint mask search and encoder learning

- Select candidate mask i from $\mathcal{M}$. Fill the non-zero entries with random values from some distribution $\mathbb{P}_M$ (we assume that $\mathbb{P}_M$ has no mass on zero) to generate a candidate $\Delta^{'}$
- Solve the identity in equation (3) using samples from the perturbations selected in the step above $\Delta^{'}$. Check for p-sparsity on the set of validation perturbations. If the solution is at most p-sparse on all the validation perturbations, then select the encoder. If the solution fails, then $i = i + 1$ and go to step one.

The mask search procedure described above requires brute force search over many masks. Even though the procedure is computationally intractable it helps demonstrate that knowledge of sparsity can suffice (See Theorem 9 below).

Theorem 9. *Suppose Assumptions 1, 3, 9, 10, and 11 hold, then an encoder that is output learned following the joint mask search and encoder learning procedure above identifies latents up to permutation and block-diagonal transforms with probability one.*

Proof. We take the encoder $f(x)$ learned from joint mask search and encoder learning procedure described above. Following Assumptions 1, 3, 9 and 11, we obtain that $f(x) = \hat{z} = Az + c$, where $x = g(z)$, A is an invertible matrix and c is an offset. Following the analysis in Proposition 1, we obtain A matrix is given as $A = \Delta^{'}_d \Delta_d^{-1}$ (substitue $\hat{z} = Az + c$, $\hat{z} + \Delta_d = Az + c + A\Delta^{'}_d$). We index the matrix in terms of the blocks.

The matrix at location (i, j) is $A_{ij} = \Delta^{'}_{ij}\tilde{\Delta}_{jj}$ (since Δ is a blockdiagonal matrix, i.e., $\Delta_{ij} = 0$ for $i \neq j$ but $\Delta_{ii} \neq 0$). Each column of $\Delta^{'}$ consists of p non-zero entries. Using this and $A_{ij} = \Delta^{'}_{ij}\tilde{\Delta}_{jj}$ we obtain that the number of non-zero entries in each column of A are at least p. We write $A_{ij}[k, q] = \sum_l \Delta^{'}_{ij}[k, l]\tilde{\Delta}_{jj}[l, q]$. Since $\Delta^{'}_{ij}[k, l]$ and $\tilde{\Delta}_{jj}[l, q]$ both take non-zero value, the first term in the above summation is non-zero. Since the other terms depend on random variables drawn independently, the probability that the sum equals zero is zero. Therefore, for each of the p indices k where the mask is non-zero, the $A_{ij}[k, q]$ is non-zero.

Suppose at least one column block of A, say $jp + 1 : (j + 1)p$, contains two columns which exhibit a different sparsity pattern. Since there are at least two columns which share a different sparsity pattern, there is at least one row where only one of them is zero and other is non-zero. Therefore, in this column block we have at least $p + 1$ rows which have at least one non-zero element. The encoder passed the sparsity test, i.e., for all the perturbations on blocks of the form $jp + 1 : (j + 1)p$ we have at most p-sparse output. Therefore, at least one of the $p + 1$ rows has to multiply with the block and output a zero, which is a zero probability event (since the non-zero elements of A matrix are all continuous random variables). Thus if any contiguous block has different sparsity pattern across columns, then the encoder is selected with probablity zero. Thus from this we can conclude that for a selected encoder, each column block exhibits a sparsity pattern that is same across all the columns in the block. To ensure that A is an invertible, all blocks exhibit a non-overlapping sparsity pattern. Therefore, A is permutation times a diagonal matrix. We now illustrate what choices of $\Delta^{'}$ lead to an A that passes the sparsity test. If for every i there exists a unique j for which $\Delta^{'}_{ij}$ is invertible and every other value of j, $\Delta^{'}_{ij} = 0$, then A is permutation times a diagonal matrix. This completes the proof.

□

In this section, we showed that we we do not need to make Assumption 6 and the knowledge of sparsity suffices to do blockwise identification. Following similar analysis as above, we can extend Theorem 4 as well.

A.2.3 Extension to stochastic perturbations

In the DGP considered in Assumption 1, we assumed that the perturbations are determinstic. We now consider stochastic perturbations.

Assumption 12. *The DGP follows*

$$z_i \sim \mathbb{P}_Z, n_{ik} \sim \mathbb{P}_{N_k}\ \forall k \in \mathcal{I},\ \ x_i \leftarrow g(z_i) \qquad \tilde{z}_{ik} \leftarrow z_i + \delta_k + n_{ik}, \forall k \in \mathcal{I} \qquad \tilde{x}_{ik} \leftarrow g(\tilde{z}_{ik}), \forall k \in \mathcal{I} \tag{24}$$

where g is injective and analytic, and Z is a continuous random vector with full support over $\mathbb{R}^d$, $\mathbb{P}_{N_k}$ is the noise distribution for the kth perturbation.

Assumption 13. *The perturbations in $\Delta = \{\delta_1, \cdots, \delta_m\}$ are one-sparse. Further, the noise vectors are also one-sparse and follow the same sparsity pattern, i.e. n_{ik} follows the same sparsity pattern as the perturbation vector δ_k to which they are added.*

The above two assumptions can be understood as follows. Each perturbation is one sparse, i.e., under each perturbation one component of the latent z changes by a fixed amount plus some noise. In the data generation process described above $\tilde{x}_{ik}$ corresponds to the kth perturbation of instance x_i. We write X for the random vector corresponding to unperturbed observation and $\tilde{X}_k$ as the random vector associated with the kth perturbation. Denote the distribution of kth perturbation conditional on X as $\mathbb{P}(\tilde{X}_k|X)$. The learner guesses the perturbation $\hat{x}_{ik}$ for instance x_i as follows

$$\hat{x}_{ik} = f^{-1}\Big(f(x_i) + \delta_k^{'} + n_{ik}^{'}\Big) \tag{25}$$

where f is the encoder (assumed to be bijective here) used by the learner, $\delta_k^{'}$ is the perturbation guessed by the learner and $n_{ik}^{'}$ is the noise sampled by the learner. We write the random variable based version of the above relationship as follows.

$$\hat{X}_k = f^{-1}\Big(f(X) + \delta_k^{'} + N_k^{'}\Big) \tag{26}$$

The learner's goal is to satisfy the following identity

$$\begin{aligned} \mathbb{P}(\hat{X}_k|X) &= \mathbb{P}(X_k|X),\ \forall k \in \mathcal{I} \\ f^{-1}\Big(f(X) + \delta_k^{'} + N_k^{'}\Big) &\overset{d}{=} g(Z + \delta_k + N_k),\ \forall k \in \mathcal{I} \\ f(X) + \delta_k^{'} + N_k^{'} &\overset{d}{=} f(X_k) = f \circ g(Z + \delta_k + N_k),\ \forall k \in \mathcal{I} \\ a(Z) + \delta_k^{'} + N_k^{'} &\overset{d}{=} a(Z + \delta_k + N_k),\ \forall k \in \mathcal{I} \end{aligned} \tag{27}$$

where $\overset{d}{=}$ denotes equality in distribution. The above identity is the same as the equivariance in distribution condition arrived at in Ahuja et al. (2022). We now see how sparsity in the changes can be exploited to guarantee strong identification similar to our result in Theorem 1.

$$a(Z + \delta_k + N_k) = a(Z) + \begin{bmatrix} J_1(Z_1^{'}) \\ J_2(Z_2^{'}) \\ \vdots \\ J_d(Z_d^{'}) \end{bmatrix} (\delta_k + N_k) \tag{28}$$

where J corresponds to the Jacobian of a. In the above equation (28), we carried out the first order Taylor expansion. We further simplify the equivariance in distribution to get the following.

$$a(Z) + \begin{bmatrix} J_1(Z_1^{'}) \\ J_2(Z_2^{'}) \\ \vdots \\ J_d(Z_d^{'}) \end{bmatrix} (\delta_k + N_k) \stackrel{d}{=} a(Z) + \delta_k^{'} + N_k^{'}$$
$$\begin{bmatrix} J_1(Z_1^{'}) \\ J_2(Z_2^{'}) \\ \vdots \\ J_d(Z_d^{'}) \end{bmatrix} (\delta_k + N_k) \stackrel{d}{=} \delta_k^{'} + N_k^{'} \tag{29}$$

Suppose that the first component of $\delta_k + N_k$ is non-zero. Due to one-sparsity we know that all the remaining components are zero. Suppose $\delta_k^{'} + N_k^{'}$ also have the same sparsity pattern as $\delta_k + N_k$ (we can arrive at qualitatively the same result that follows even in the absence of this assumption). As a result, $[J_{21}(Z_2^{'}), \cdots, J_{d1}(Z_d^{'})] = 0$. Suppose a is analytic. As a result, the Jacobian J is analytic as well. Further, suppose that the support of $Z_j^{'}$ for all $j \geq 2$ has a non-zero measure (this condition is the extension of Assumption 3 from deterministic case to the stochastic case). From (Mityagin, 2015) it follows that $[J_{21}(z), \cdots, J_{d1}(z)] = 0$ is identically zero. Since the identity in equation (28) holds for all k we can conclude that the Jacobian of a is a diagonal matrix. Since $\hat{z} = a(z)$, we can conclude that changes to one component of z impact exactly one component of $\hat{z}$ and not the rest. Thus we can conclude we have perfect disentanglement. So far we analyzed the case where perturbations are one-sparse. We can generalize the above argument to non-overlapping blockwise perturbations following similar arguments to the deterministic case as well.

A.2.4 Extension to non-linear mechanisms for perturbations

In the main body of the paper, we assumed that the data is generated under sparse and fixed perturbations. We now extend our analysis to the case where different perturbation can be applied to different points z. A mechanism $m : \mathbb{R}^d \to \mathbb{R}^d$ takes as input the latent and outputs the perturbation vector. We call a mechanism p-sparse, if it only changes p components out of the d latents, i.e., $\forall z \in \mathbb{R}^d$, $\exists\, d-p$ components of z which remain unchanged on application of m. We write this data generation process as follows.

Assumption 14. *The DGP follows*

$$z \sim \mathbb{P}_Z,\ \tilde{z}_k \leftarrow z + m_k(z)\ \forall k \in \mathcal{I}, \tilde{x}_k \leftarrow g(\tilde{z}_k)\ \forall k \in \mathcal{I}, \tag{30}$$

where g is injective and analytic, and Z is a continuous random vector with full support over $\mathbb{R}^d$, m_k is the kth perturbation mechanism.

Assumption 15. *Each m_k is one-sparse. For each latent dimension $i \in \{1, \cdots, d\}$, $\exists$ a mechanism $m_k \in \{m_1, \cdots, m_m\}$ that changes that latent dimension.*

Recall that $a = f \circ g$, where f is the encoder that the learner uses. We state the assumption on a below.

Assumption 16. *a is an analytic function. For each component $i \in \{1, \cdots, d\}$ of $a(z)$ and each component $j \in \{1, \cdots, d\}$ of z, define the set $\mathcal{S}^{ij} = \{\theta \mid a_i(z+b) = a_i(z) + \nabla_j a_i(\theta) b, z \in \mathbb{R}^d\}$, where b is a fixed vector in $\mathbb{R}^d$. Each set $\mathcal{S}^{ij}$ has a non-zero Lebesgue measure in $\mathbb{R}^d$.*

For each perturbation, the learner uses a $m_k^{'} : \mathbb{R}^n \to \mathbb{R}^d$ to guess the changes caused by the true mechanism m_k. We write the identity that the learner solves as follows. $\forall k \in \{1, \cdots, m\}$ and $\forall (x, \tilde{x}_k)$ generated by the DGP in Assumption 14

$$f(\tilde{x}_k) = f(x) + m_k^{'}(x). \tag{31}$$

Theorem 10. *If Assumption 14, 15, and 16, hold, then the solution to equation (31) (with one-sparse $m_k^{'}$), satisfies $\hat{z} = \Pi\Lambda(z) + c$, where Π is a permutation matrix, $\Lambda(z) = \mathsf{diag}[\lambda(z_1), \cdots, \lambda(z_d)]$ is a function whose each component exactly depends on one latent dimension.*

Proof.

$$
\begin{aligned}
f(\tilde{x}_k) &= f(x) + m_k^{'}(x), \\
a(\tilde{z}_k) &= a(z) + m_k^{\dagger}(z), \\
a(z + m_k(z)) &= a(z) + m_k^{\dagger}(z),
\end{aligned}
\tag{32}
$$

where $m_k^{\dagger} = m_k^{'} \circ g$. We drop k from the above equation for ease of presentation and get

$$
a(z + m(z)) = a(z) + m^{\dagger}(z) \tag{33}
$$

We do a Taylor expansion of a around z to get

$$
\begin{aligned}
&a(z) + [J_1(z_1^{'}), J_2(z_2^{'}), \cdots, J_d(z_d^{'})]m(z) = a(z) + m^{'}(z) \\
&\implies [J_1(z_1^{'}), J_2(z_2^{'}), \cdots, J_d(z_d^{'})]m(z) = m^{'}(z)
\end{aligned}
\tag{34}
$$

Suppose $m(\cdot)$ and $m^{'}(\cdot)$ are both one-sparse and impact the first component of z. From the above it follows that all elements of $[J_{21}(z_1'), \cdots, J_{d1}(z_d')]$ are zero except $J_{11}(z_j^{'})$. Since the above holds true for all z, $[J_{21}(z_1'), \cdots, J_{d1}(z_d')]$ would be zero on a set of measure non-zero. Thus $[J_{21}, \cdots, J_{d1}]$ is identically zero. We can repeat the same argument for all the columns and conclude that in each column all rows except one are zero. From this we can conclude that $a(z) = \Pi\Lambda(z) + c$. □

We can follow the same line of reasoning and argue for blockwise identification as well.

A.2.5 Connection with causal interventions

In the DGP in equation (24), we assumed that Z is sampled from any distribution $\mathbb{P}_Z$. We now consider a special case, where $Z = [Z_1, \cdots, Z_d]$ follows a certain structural causal model $\mathcal{S}$ given as

$$Z_i \leftarrow f_i(\mathsf{Pa}(Z_i), U_i), \forall i \in \{1, \cdots, d\} \tag{35}$$

where Z_i is generated from its parent variables denoted by $\mathsf{Pa}(Z_i)$ using the mechanism $f_i : \Pi_{\mathsf{Pa}(Z_i)}\mathcal{Z}_i \times \mathcal{U}_i \rightarrow \mathbb{R}$, which also takes the noise variable U_i as input. The support of Z_i is denoted by $\mathcal{Z}_i$ and that of U_i is denoted by $\mathcal{U}_i$. Suppose we perturb Z_k. Under this perturbation all the latent variables for which Z_k is an ancestor are going to be also affected, while keeping the rest of the variables unchanged.

Post the perturbation, the immediate children of Z_k are affected and then their children and so on. Therefore, it is reasonable to assume that we first observe the impact of perturbation on Z_k itself and eventually observe the impact on child variables. Consider a sample point $[(z_1, \cdots z_d), (x_1, \cdots, x_n)]$ generated by equation (1). The different observations under perturbations are

- **Pre perturbation:** $[(z_1, \cdots z_k, \cdots, z_d), (x_1, \cdots, x_n)]$
- **At the time of perturbation:** $[(z_1, \cdots z_k + \delta, \cdots, z_d), (x_1^{'}, \cdots, x_n^{'})]$
- **Sufficiently long after the perturbation:** $[(z_1, \cdots z_k + \delta, \cdots, z_d^{''}), (x_1^{''}, \cdots, x_n^{''})]$

In the above, the latent of the sample pre perturbation and at the time of perturbation only differ in the perturbed components. However, when sufficient period has passed, other latent variables that are on the downstream path from Z_k also change. In this work, we only deal with original samples and the samples at the time of perturbation. In works that rely on causal interventions such as Brehmer et al. (2022), one assumes access to samples before perturbation and those generated sufficiently long after the perturbation.

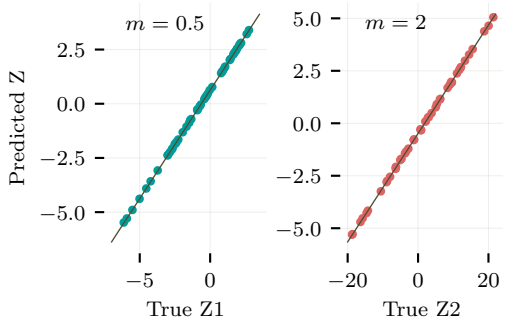

Figure 4: Regression of predicted latent values against true latent values for componentwise perturbations ($d = 10$).

A.3 Supplementary materials for experiments

Loss function, architecture, and other hyperparameters In all the experiments, we optimized equation (4) with square error loss. The encoder f was an MLP with two hidden layers of size 100 for the low-dimensional synthetic experiments and a ResNet-18 (He et al., 2015) for the image-based experiments. For the low-dimensional synthetic experiments, we used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.005 with batches of 10000 examples for 2000 epochs. For the image-based experiments, we trained online with a learning rate of $1e-4$ and a batch size of 100.

Evaluation metrics Blockwise MCC (BMCC) is a natural extension of MCC. We compute the R^2 score (using linear regression) between every pair of blocks impacted under true perturbation and the guessed perturbation. We find the optimal matching between pairs of blocks to maximize the average R^2 score between the matched blocks. We report the R^2 score under the optimal matching in Table 1.

Supplementary figures In Figure 4, we plot the predicted latents against the true latent value for two of the ten latent dimensions (the two dimensions that we plot are randomly selected) when we perturb one component at a time (setting corresponds to the paragraph on non-overlapping perturbations in Section 4). The plot shows a linear relationship between the true and the predicted latent; note that there are different slope and intercept for the different latents. The slope depends on the ratio between the change in the true latents and the predicted latent. In Figure 5, we plot the predicted latents against the true latent value for two of the ten latent dimensions (the two dimensions that we plot are randomly selected) when we perturb a block of two components at a time and the blocks overlap (setting corresponds to the paragraph on overlapping perturbations in Section 4). In Figure 6, we show a full set of images for the experiment shown in Figure 1.

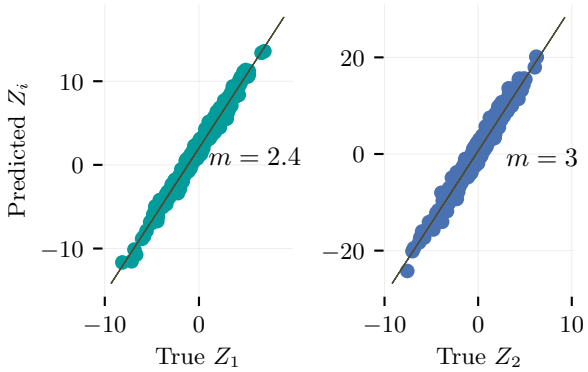

Figure 5: Regression of predicted latent values against true latent values for overlapping perturbations ($d = 10$).

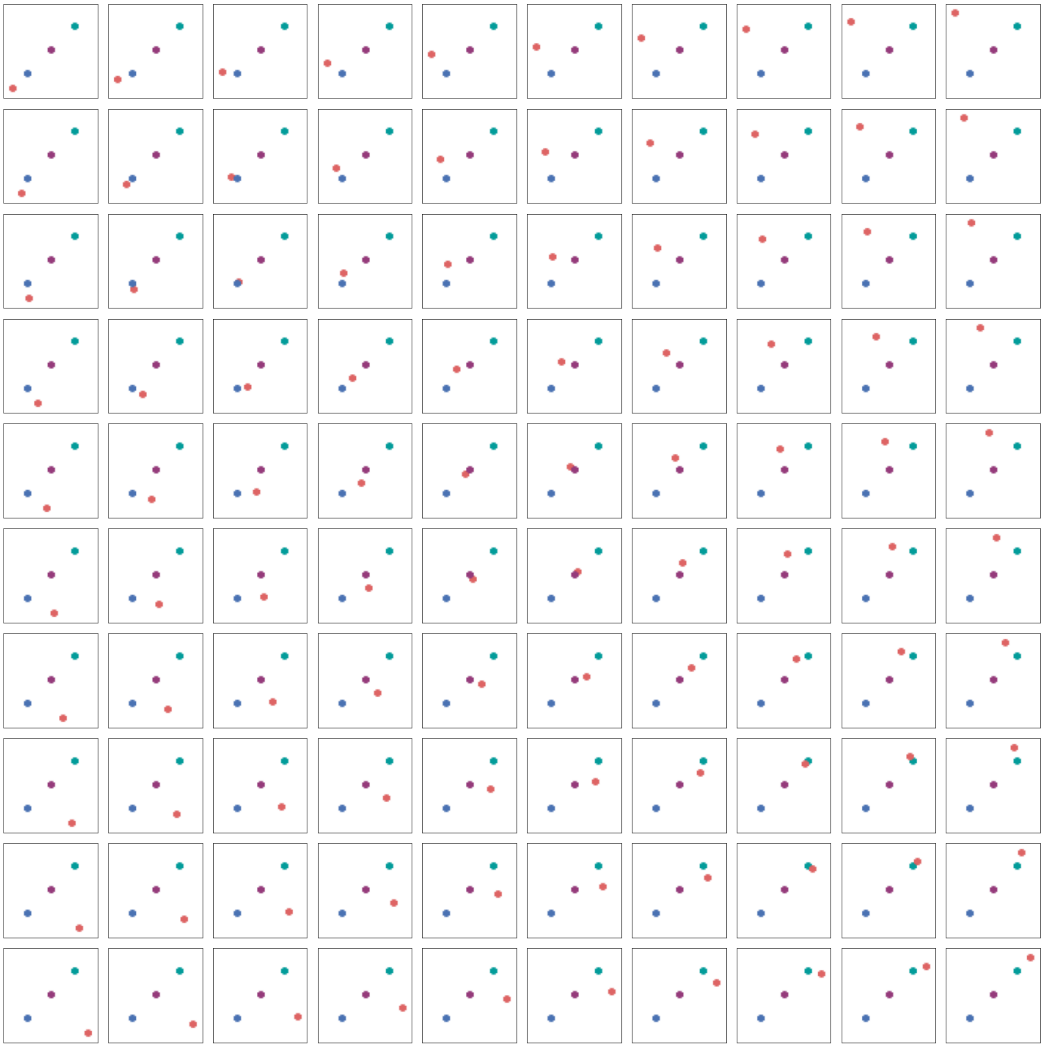

Figure 6: Full set of images for the experiment shown in Figure 1 used to render the supplementary animation. The three balls on the diagonal are stationary throughout and the fourth ball is moved across a 10×10 grid; we get the associated network predictions and animate them to show the predicted movement of the stationary balls in the attached animation.

Table 4: Comparing MCC and BMCC for stochastic perturbations

d	MCC C-wise	BMCC B-wise	MCC (overlap) C-wise
6	0.99 ± 0.00	0.99 ± 0.00	0.95 ± 0.00
10	0.99 ± 0.00	0.99 ± 0.00	0.96 ± 0.00
20	0.99 ± 0.00	0.99 ± 0.00	0.98 ± 0.00

Table 5: Comparing MCC for perturbations (Normal latent)

d	MCC $\beta = 1$	MCC $\beta = 10$	MCC $\beta = 100$
6	0.68 ± 0.04	0.69 ± 0.02	0.72 ± 0.02
10	0.69 ± 0.03	0.69 ± 0.02	0.72 ± 0.02
20	0.70 ± 0.02	0.74 ± 0.03	0.72 ± 0.04

A.3.1 Experiments for the stochastic model

In Section 4, we provided experiments for deterministic perturbation model. In Section A.2.3, we discussed the extension of the theory to stochastic perturbation model. In this section, we present the results for the experiments on stochastic perturbation model. We consider the same data generation process that is used in Table 1 and Table 2. We draw the latents from normal distribution used in Table 1 and Table 2. To each deterministic perturbation, we add standard normal noise (consistent with the data generation process described in Assumption 12). Instead of exactly equating the distribution in the LHS and RHS of the identity in equation (27), we take the expectation of the random variables on the LHS and RHS of equation (27). Note that when we take expectation we get the same identity that we use in equation (3). Therefore, we continue to use the loss defined in equation (4). We show the results of the experiments averaged over five trials in Table 4. These results show that the insights from the deterministic case carry over to the stochastic case as well.

A.3.2 Supplementary experiments for comparisons with beta-VAE

In this section, we use β-VAE from (Higgins et al., 2016). We consider three different values of β – 1, 10 and 100. We use the similar encoder architecture as in our earlier experiments for synthetic datasets, except now we have two linear heads one for the mean embedding and other for variance embedding. We use the same decoder architecture as the encoder architecture used for our earlier experiments for synthetic datasets. We use the Adam optimizer with a learning rate of 0.001. Recall that in Table 1 we used two choices for the latent distributions. For the normal distribution (which satisfies blockwise independence), we show the results in Table 5. For the uniform distribution, we show the results in Table 6.

Table 6: Comparing MCC for perturbations (Uniform latent)

d	MCC $\beta = 1$	MCC $\beta = 10$	MCC $\beta = 100$
6	0.63 ± 0.03	0.60 ± 0.02	0.50 ± 0.08
10	0.53 ± 0.01	0.51 ± 0.01	0.43 ± 0.03
20	0.42 ± 0.01	0.40 ± 0.01	0.36 ± 0.01

A.3.3 Stationary point

Recall our learning objective is to minimize the objective given in Equation 4. We use a deep network, $\tilde{f}(\cdot;\theta)$ parameterized by θ as our encoder and we can rewrite Equation 4 as a loss function that depends on our choice of θ and Δ' (the learner's guess for the offsets),

$$\mathcal{L}(\theta,\Delta^{'}) = \mathbb{E}\Big[\Big\|f(\tilde{x}_k;\theta) - f(x;\theta) - \delta_k^{'}\Big\|^2\Big] = \mathbb{E}\Big[\sum_j \Big(f_j(\tilde{x}_k;\theta) - f_j(x;\theta) - \delta_{jk}^{'}\Big)^2\Big] \tag{36}$$

We take the partial derivative of the loss with respect to one of the parameters θ_i and obtain

$$\frac{\partial\mathcal{L}(\theta,\Delta^{'})}{\partial\theta_i} = \mathbb{E}_{x,\tilde{x}_k}\left[\sum_j \underbrace{(f_j(\tilde{x}_k;\theta) - f_j(x;\theta) - \delta_{jk}^{'})}_{=:e_j(x,\tilde{x}_k,\theta)} \underbrace{(\frac{\partial f_j(\tilde{x};\theta)}{\partial\theta_i} - \frac{\partial f_j(x;\theta)}{\partial\theta_i})}_{=:\phi_j(x,\tilde{x}_k,\theta)}\right]$$

Suppose we learn a function $\tilde{f}$ for which $e_j(x,\tilde{x}_k,\theta)$ is independent of x and $\tilde{x}$ and we denote it as $e_j(\theta)$ for all $j \in \{1,\cdots,d\}$. Under this assumption, we simplify the above expression as follows.

$$\frac{\partial\mathcal{L}(\theta,\Delta^{'})}{\partial\theta_i} = \sum_j e_j(\theta)\mathbb{E}_{x,\tilde{x}_k}\Big[\phi_j(x,\tilde{x}_k,\theta)\Big] = \sum_j e_j(\theta)\mu_j(\theta)$$

where $\mu_j(\theta) = \mathbb{E}_{x,\tilde{x}_k}\Big[\phi_j(x,\tilde{x}_k,\theta)\Big]$. $\mu_j(\theta)$ measures the expected difference in the guessed perturbation for the component j when parameter θ_i of the neural network is changed. If the impact of change in the parameter is similar on average across all the components, then $\mu_j(\theta) = \mu_k(\theta) = \mu(\theta)$ for all $j \neq k$, which leads to

$$\frac{\partial\mathcal{L}(\theta,\Delta^{'})}{\partial\theta_i} = \sum_j e_j(\theta)\mathbb{E}_{x,\tilde{x}_k}\Big[\phi_j(x,\tilde{x}_k,\theta)\Big] = \mu(\theta)\sum_j e_j(\theta) \tag{37}$$

Under these conditions, this is a stationary point if $\sum_j e_j(\theta) = 0$. Empirically we observe that if j is perturbed by c, then $e_j(\theta) = \frac{c}{2}$ and other components $k \neq j$, $e_j(\theta) = \frac{-c}{2(n_{\mathsf{balls}}-1)}$. If we substitute this in the equation above, we find that the partial derivative is zero. Since this holds for all the components θ_i, we can conclude that the point observed empirically is a stationary point. Under the assumption that $e_j(x,\tilde{x}_k,\theta)$ is independent of $x,\tilde{x}_k$, we can follow the analysis presented in proof of Theorem 1, we get $\hat{z} = Az + c$. If z changes by $[c,0,\cdots,0]$, then $\hat{z} = [\frac{c}{2}, -\frac{c}{2(n_{\mathsf{balls}}-1)},\cdots,-\frac{c}{2(n_{\mathsf{balls}}-1)}]$. We use this to obtain $A[i,j] = \frac{-1}{2(n_{\mathsf{balls}}-1)}$, where $i \neq j$ and $A[i,i] = \frac{1}{2}$. If $n_{\mathsf{balls}} = \infty$, then A is a diagonal matrix, which implies that the MCC is one. In the discussion above, we assumed that the learner knows the component that changes. If the learner does not know the component that changes, then that introduces permutation errors as well.

A.3.4 Compute used

The synthetic experiments were conducted on a 2.2 GHz Quad-core Intel Core i7. The image-based experiments were each conducted on a single GPU on a 6 core node with 16GB of allocated memory. The nodes were requested from am internal shared compute cluster with approximately 500 GPUs shared across a large number of users. Most of the GPUs are Nvidia RTX-8000 and a small number are Nvidia V100s; both types of GPUs were used to conduct the experiments depending on availability.

A.3.5 Assets used and the license details

We used the code from `https://github.com/brendel-group/cl-ica`, which uses the MIT license. We also used code from `https://github.com/pygame/`, which is distributed under GNU LGPL version 2.1.