WHEN DO RELATIONS REVEAL FACTORS: LEARNING IDENTIFIABLE REPRESENTATIONS FROM TYPED RELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Disentangled representation learning seeks to recover the key factors of variation in data, providing representations that can be more interpretable, robust, and generalizable. Observational data alone generally do not uniquely determine the underlying factors, since different nonlinear mixtures can explain the same observations equally well, leading to an identifiability problem. To guarantee identifiability, additional information beyond observations themselves is therefore needed. In this work, we derive this information from *typed relations* between observations, with examples in healthcare including kinship ties, patient similarity, and, in social networks, various forms of social connections. We develop an identifiability theory that characterizes when and to what extent such relational information can help identify the underlying factors. Specifically, we define for each factor its *relation signature*, i.e., the set of typed relations affecting this factor, and our main results show that, under a common sparsity condition, the partial-order structure between relation signatures can determine to what extent each factor can be identified. For different partial-order structures, the theory produces a series of identifiability results, ranging from no guarantee, to directional ambiguity, further to identifiability of blocks of factors, and ultimately to factor-wise identifiability. Guided by this theory, we develop a variational method that jointly learns representations and relation-specific activity masks under a sparsity constraint. Experiments on controlled settings show that the learned factors can identify the true factors exactly in a way that our theory predicts. We also test our method on real image benchmarks, demonstrating that theoretical results hold even when conditions are not strictly satisfied and that our method consistently learns better-disentangled representations than baselines.

1 INTRODUCTION

Disentangled representation learning is a longstanding challenge in machine learning, motivated by the goal of recovering key factors of variation and learning representations that are robust (Suter et al., 2019), generalizable (Montero et al., 2021), and interpretable (Yang et al., 2021). A central difficulty is that fitting the observations does not by itself determine which latent learned factors should represent these factors. For a generic generative process $x = f(z)$, different nonlinear mixtures of z may explain the same observations equally well. Identifiability theory then asks whether additional information is sufficient to rule out such alternative representations and recover the underlying factors up to a specified class of transformations, since with observational data alone such recovery is impossible in general (Locatello et al., 2019). Existing approaches obtain identifiability from additional structures, such as temporal dependence (Hyvärinen & Morioka, 2016; 2017; Li et al., 2026), auxiliary variables (Hyvärinen et al., 2019; Khemakhem et al., 2020a), interventions (Buchholz et al., 2023), or sparse mechanisms (Lachapelle et al., 2022; 2026).

In this work, we consider as such an additional structure the *typed relations*, i.e., discrete labels describing how samples are connected. For example, healthcare records may be linked through kinship (Meng et al., 2024), lifestyle and patient similarity (Yang, 2026), while social data may contain several forms of family or acquaintance relations. Such observed relations are ubiquitous in real-world data, yet it remains unclear whether one could utilize such information to break non-identifiability and ultimately recover disentangled representations. This leads to the main question of this work: *when and how can we learn identifiable representations from typed relations?*

We observe high-dimensional samples $\{x^i\}_{i=1}^n$, each generated by latent factors z^i, i.e., $x^i = f(z^i)$, together with typed relations between them, as illustrated in Figure 1(a). Each relation type is assumed to correspond to a fixed but unobserved subset of factors z_j that varies across connected samples, while all remaining factors are shared. Figure 1(b) illustrates this assumption: the samples x^1 and x^3 connected by relation $\tau = 1$ share the second and fourth factors, i.e., $z_2^1 = z_2^3$ and $z_4^1 = z_4^3$, while the first and third factor vary, i.e., $z_1^1 \neq z_1^3$ and $z_3^1 \neq z_3^3$. Reading column-wise across relation types gives each factor its *relation signature*, i.e., the set of types under which this factor varies.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

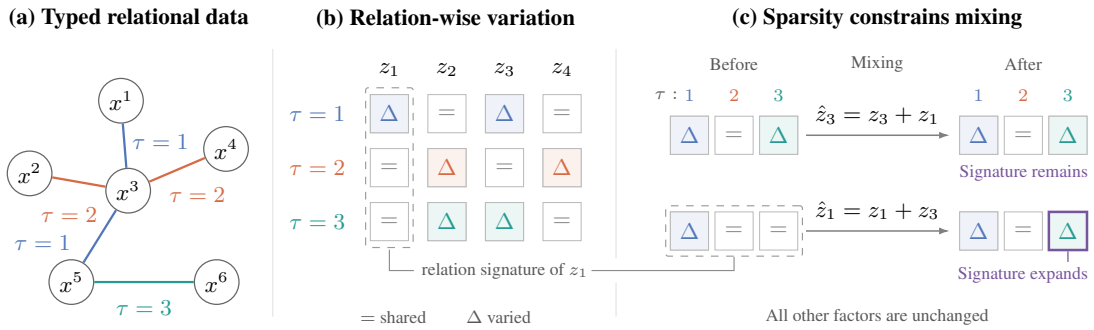

Figure 1: **Typed relations constrain factor mixing. (a)** We observe samples x^i and the types τ of their relations. **(b)** Each type specifies which factors would vary (Δ) or are shared ($=$). A factor's relation signature records the types under which it varies. **(c)** Mixing takes the union of factor signatures. Replacing z_3 by $z_3 + z_1$ leaves its signature unchanged, while replacing z_1 with $z_1 + z_3$ expands z_1's signature by adding type $\tau = 3$. The proposed sparsity condition excludes such an expansion as we forbid mixing that induces denser pattern of relation-wise variation. See explanations in the text.

The key observation is that *when a factor that is shared under a relation type is mixed with one that varies under that type, the resulting learned factor may also vary under that type, thereby enlarging its relation signature and making the overall relation-wise variation pattern denser*. Figure 1(c) illustrates this with an example. Consider mixing a true factor z_1 into z_3, i.e., letting the learned factor $\hat{z}_3$ depend on both z_1 and z_3, which results in $\hat{z}_3 = z_1 + z_3$. Since the relation signature of z_1 is contained in that of z_3, no new relation types are introduced into the signature of $\hat{z}_3$. Consequently, z_1 and z_3 cannot be distinguished solely from relation-wise variations: under relation $\tau = 1$, both factors are allowed to vary. In contrast, if z_3 is mixed into z_1, so that $\hat{z}_1 = z_1 + z_3$, then the mixing becomes detectable, because factor z_1 should remain invariant under relation $\tau = 3$, and any observed variation in $\hat{z}_1$ reveals the dependence on z_3. In this case, the signature of $\hat{z}_1$ expands and the overall relation-wise variation pattern becomes denser.

This observation suggests that, guided by the Occam's razor argument, imposing sparsity by excluding denser relation-wise variation patterns may resolve the ambiguities arising from factor mixing. Our main result (Theorem 3.2) formalizes this intuition by precisely characterizing how typed relations identify latent factors. We then derive a series of identifiability results (Corollaries 3.4, 3.5, and 3.6) showing how the remaining ambiguity depends on the inclusion structure of the ground-truth relation signatures. We informally summarize our theoretical results as follows.

Suppose: (i) the learned representations $\hat{z}$ explain the observations equally well as the ground-truth, i.e., $\forall\, x \in \mathcal{X},\ x = \hat{f}(\hat{z}) = f(z)$, and (ii) the learned signatures $\hat{S}_r$ are no denser than the ground-truth signatures, i.e., $\sum_r |\hat{S}_r| \leq \sum_r |S_r|$.

Then: any mixing that expands the signatures will be ruled out by the sparsity condition (Theorem 3.2), and what mixing would remain is determined by structure of the ground-truth relation-wise variation, giving rise to four different levels of identifiability:

Factor-wise Recovery (Corollary 3.4) if $\forall\, q \neq r,\ S_q \not\subseteq S_r$		**Block-wise Recovery** (Corollary 3.5) if $\forall\, q \neq r,\ S_q \not\supseteq S_r$	
Directional Ambiguity (Corollary 3.6) if $\exists\, q \neq r,\ S_q \subsetneq S_r$		**No Guarantee** (trivial case) if $\forall\, q, r,\ S_q = S_r$	

Our main contributions are as follows:

1. We theoretically characterize, in Section 3, when and to what extent typed relations can identify latent factors. Using the *relation signature* of each factor, we prove that their inclusion structure exactly characterizes the sparse factor mixing, which yields a series of identifiability results under different relation-wise variations. Appendix A compares our theoretical contributions with related work.

2. In Section 4, we show how these theoretical insights can be translated into a practical method to learn disentangled representations. Specifically, we introduce a parameterized mask on top of VAE to allow sharing latent factors such that the correct reconstruction would force learning the correct relation-wise variation pattern, i.e., the mask, from the observed samples and their relations.

3. In Section 5, we verify these theoretical insights and our learning method in a controlled setting, showing that, under different patterns of relation-wise variations, learning suppresses the mixing that violates what theory predicts. We also compare our method on image benchmarks, demonstrating that theoretical results hold even if conditions not holding strictly and that our method consistently improves over baselines.

Related Work. Identifiability of latent representations has been pursued using different sources of additional structure, including temporal dependence (Hyvärinen & Morioka, 2016; 2017), auxiliary variables (Hyvärinen et al., 2019), interventions (Buchholz et al., 2023), and sparsity in latent dependencies (Zheng et al., 2022; Zheng & Zhang, 2023; Lachapelle et al., 2022; 2026). Recent work further explores mechanistic independence (Matthes et al., 2026), structural priors (Patil et al., 2026), symmetry-based disentanglement (Dang-Nhu et al., 2026), and broader notions of statistical and structural identifiability (Nelson et al., 2026). Closest to our theory, Lachapelle et al. (2026) characterize partial disentanglement through entanglement graphs and graph-preserving transformations. Our setting instead provides typed pairwise modes of variation. Under this structure, factor mixing combines relation signatures by set union, making inclusion structure among signatures the exact obstruction to eliminating further mixing under sparsity.

A complementary line of work uses non-i.i.d. pairs as weak supervision. Locatello et al. (2020) show that pairs sharing latent factors can enable disentanglement without observing which factors are shared, while Fumero et al. (2021) exploit pairs sharing generative factors under product-manifold formulation. These works establish that paired observations can provide information for factor recovery. We ask how the structure across multiple relation types determines to what extent the factors can be recovered. The signature inclusion specifies the maximal mixing compatible with sparsity, yielding a series of results from unrestricted ambiguity to block-wise and factor-wise identifiability.

2 Preliminaries

Notation. For a set $\mathcal{S}$, let $\mathrm{Id}_{\mathcal{S}} : \mathcal{S} \to \mathcal{S}$ denote the identity map, and let I_m denote the $m \times m$ identity matrix. For a positive integer K, write $[K] := \{1, \dots, K\}$. For binary variables, let $\wedge$ and $\vee$ denote logical "and" and "or", respectively. For binary matrices $A \in \{0,1\}^{m\times \ell}$ and $C \in \{0,1\}^{\ell \times n}$, we write $A \odot C \in \{0,1\}^{m\times n}$ for their Boolean matrix product, defined entry-wise by $(A \odot C)_{ij} := \bigvee_{r=1}^{\ell} (A_{ir} \wedge C_{rj})$. For a binary matrix A, let $A_{i:} := \{j : A_{ij} = 1\}$ and $A_{:j} := \{i : A_{ij} = 1\}$ denote its row and column supports, respectively, and let $\|A\|_0$ denote its number of nonzero entries. For a differentiable map g, write J_g for its Jacobian. We denote by $\mathfrak{S}_K$ the set of all permutations of $[K]$, and by $\mathrm{sgn}(\pi) \in \{-1, 1\}$ the sign of $\pi \in \mathfrak{S}_K$. Throughout the paper, we use $r, q \in [K]$ to index true factors and $s \in [K]$ for learned factors. For $\mathcal{G} \subseteq [K]$, we write $z_{\mathcal{G}} := (z_r)_{r\in\mathcal{G}}$ with indices in increasing order.

Definition 2.1 (Latent generative model)**.** *Let $z = (z_1, \dots, z_K) \in \mathbb{R}^K$ be the true latent variable, whose coordinates represent K scalar factors. The latent generative model that gives observation x is defined as*

$$x = f(z), \tag{1}$$

where $f : \mathbb{R}^K \to \mathbb{R}^D$ is an injective C^1 function.

In practice, we aim to learn f^{-1}, e.g., through maximizing the likelihood of x. Identifiability then compares only those learned functions that generate the observations equally well as the true one, which we formalize below.

Definition 2.2 (Observational equivalence and entanglement map)**.** *Let $f : \mathbb{R}^K \to \mathbb{R}^D$ be the ground-truth generative function in Definition 2.1, and let $\hat{f} : \mathbb{R}^K \to \mathbb{R}^D$ be a learned injective function. Then, f and $\hat{f}$ are observationally equivalent if $f(\mathbb{R}^K) = \hat{f}(\mathbb{R}^K)$. The entanglement map h is a bijection and $h, h^{-1} \in C^1$:*

$$h := \hat{f}^{-1} \circ f : \mathbb{R}^K \to \mathbb{R}^K. \tag{2}$$

Observational equivalence means that the two functions $f, \hat{f}$ can generate the same x, although they may be represented by different latent factors $z, \hat{z}$, whose discrepancy is then entirely captured by the entanglement map h, given $\hat{z} = h(z)$, or factor-wise, $\hat{z}_s = h_s(z_1, \cdots, z_K)$. The exact form of h is not needed to characterize (dis-)entanglement, only which true factors each learned factor depends on. For a differentiable h, this dependency is captured by its Jacobian: h_s depends on z_r whenever $\partial h_s / \partial z_r$ is nonzero somewhere in the latent domain, defined as follows.

Definition 2.3 (Functional Jacobian support)**.** *The functional Jacobian support of h is*

$$H := \mathrm{supp}(J_h) \in \{0,1\}^{K\times K}, \qquad H_{sr} = 1 \iff \exists\, z \in \mathbb{R}^K : \frac{\partial h_s}{\partial z_r} \neq 0. \tag{3}$$

Each row $H_{s:} := \{r \in [K] : H_{sr} = 1\}$ therefore records the true factors mixed into the learned factor s. Different restrictions on this dependency pattern correspond to different levels of identifiability. In the strongest case, every learned factor depends on exactly one true factor, with a one-to-one correspondence between them. Up to permutation, H is then diagonal and h acts separately on each factor. This leads to the standard notion of factor-wise identifiability.

Definition 2.4 (Factor-wise identifiability)**.** *Let f and $\hat{f}$ be observationally equivalent with entanglement map $h = \hat{f}^{-1} \circ f$. The learned representation is factor-wise identifiable if there exist a permutation $\pi \in \mathfrak{S}_K$ and bijections $d_s : \mathbb{R} \to \mathbb{R}$, $s \in [K]$, with $d_s, d_s^{-1} \in C^1$, such that*

$$h_s(z) = d_s\big(z_{\pi(s)}\big), \qquad s \in [K], \quad z \in \mathbb{R}^K. \tag{4}$$

Equivalently, $h = d \circ P_\pi$, where $P_\pi z = (z_{\pi(1)}, \dots, z_{\pi(K)})$ and $d(u) = (d_1(u_1), \dots, d_K(u_K))$. Factor-wise identifiability thus excludes mixing between distinct factors. More generally, one may only require mixing to be confined within predefined groups of factors. Equivalently, after a suitable permutation, the dependency pattern of h is block diagonal. This gives the weaker notion of block-wise identifiability.

Definition 2.5 (Block-wise identifiability)**.** *Let f and $\hat{f}$ be observationally equivalent with entanglement map $h = \hat{f}^{-1} \circ f$, and let $\mathcal{P} = \{\mathcal{G}_1, \dots, \mathcal{G}_M\}$ be a partition of $[K]$. The learned representation is block-wise identifiable with respect to $\mathcal{P}$ if there exist a permutation $\pi \in \mathfrak{S}_K$ and bijections $d_a : \mathbb{R}^{|\mathcal{G}_a|} \to \mathbb{R}^{|\mathcal{G}_a|}$, $a \in [M]$, with $d_a, d_a^{-1} \in C^1$, such that*

$$\big(h_{\pi^{-1}(r)}(z)\big)_{r \in \mathcal{G}_a} = d_a(z_{\mathcal{G}_a}), \qquad a \in [M], \quad z \in \mathbb{R}^K. \tag{5}$$

Here π maps learned-factor indices to true-factor indices, and the entries of each block are ordered by increasing $r \in \mathcal{G}_a$. It indicates that factors may mix within each block but not across blocks.

3 Identifiability from Typed Relations

Section 2 reduced the identifiability problem to analyzing the entanglement map h and its functional Jacobian support $H = \mathrm{supp}(J_h)$. Building on these preliminaries, we first formalize the concept of *relation signatures* in Section 3.1. We then show in Section 3.2 that, under sparsity, the inclusion structure of these signatures completely determines the remaining mixing. Finally, leveraging these insights, we derive a series of identifiability results ranging from the strongest notion, factor-wise separation (Corollary 3.4), through block-wise separation of factors (Corollary 3.5), to directional identifiability (Corollary 3.6), and ultimately to the unrestricted case.

3.1 From Typed Relations to Factor Signatures

We begin our formulation with the observed relations. Let $T \geq 1$ be the number of relation types, write $[T] := \{1, \dots, T\}$, and consider n observed data samples with typed relations $G_n = \big(\{x^i\}_{i=1}^n, \{E_\tau\}_{\tau=1}^T\big)$, where $E_\tau \subseteq \big\{(i,j) : i,j \in [n],\ i \neq j\big\}$. An edge $(i,j) \in E_\tau$ indicates samples x^i and x^j are connected by a relation of type τ. Each relation type τ is associated with an active factor set $R_\tau \subseteq [K]$ which we assume is not observed. The factors indexed by R_τ vary along pairs connected by the relation of type-τ, leaving the other factors invariant. Our analysis builds on two key objects: the *active factor matrix* and the *relation signature*.

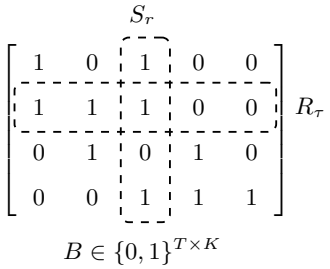

Figure 2: Example of active factor matrix B. Column $S_r = \{1, 2, 4\}$. Row $R_\tau = \{1, 2, 3\}$.

Definition 3.1 (Active factor matrix and relation signatures)**.** *The active factor matrix $B \in \{0,1\}^{T \times K}$ is defined by*

$$B_{\tau r} = 1 \quad \Longleftrightarrow \quad r \in R_\tau, \tag{6}$$

where $R_\tau \subseteq [K]$ is the set of factors that varies under relation type τ. Thus, the τ-th row of B records its active factor set:

$$R_\tau := \{\, r \in [K] : B_{\tau r} = 1 \,\}. \tag{7}$$

For each factor $r \in [K]$, its relation signature is

$$S_r := \{\, \tau \in [T] : B_{\tau r} = 1 \,\}, \tag{8}$$

i.e., the set of relation types under which factor r varies.

The rows and columns of B (see Figure 2) give complementary views: R_τ specifies which factors would vary under relation type τ, whereas S_r records how factor r varies across relation types. Thus far, we have defined relation signatures for the true factors. We next formalize how these signatures relate to those of the learned factors. Recall that in Definition 2.3 the rows of $H = \mathrm{supp}(J_h)$ describe which true factors each learned factor depends on. We therefore obtain analytically the *learned active matrix* $\hat{B}$ using H via a Boolean product

$$\hat{B} := B \odot H^\top, \quad \hat{B}_{\tau s} = \bigvee_{r=1}^{K} \big(B_{\tau r} \wedge H_{sr}\big). \tag{9}$$

The entry $\hat{B}_{\tau s} = 1$ exactly when relation type τ activates at least one true factor on which the learned factor s depends. If $\hat{B}_{\tau s} = 0$, then $h_s(z) = h_s(z')$ for every type-τ pair. For each learned factor s, we define its *learned*

relation signature as $\hat{S}_s := \{\tau \in [T] : \hat{B}_{\tau s} = 1\}$. By definition of the Boolean product, this signature is exactly the union of signatures of all true factors entering s

$$\hat{S}_s = \bigcup_{r \in H_{s:}} S_r, \qquad H_{s:} := \{r \in [K] : H_{sr} = 1\}. \tag{10}$$

That means that a learned factor acquires the signatures of all true factors contributing to it. Thus, the signature of a mixture is given by the Boolean union of the constituent signatures. We illustrate this mechanism in Figure 3. Arrows in B and $\hat{B}$ denote which factors vary under each relation type, while arrows in H denote mixing. In the true structure (B), factor z_1 has signature $S_1 = \{1, 2\}$, while factor z_2 has signature $S_2 = \{1, 3\}$. If $\hat{z}_1$ depends on both z_1 and z_2, as shown in H, then its learned signature becomes $\hat{S}_1 = S_1 \cup S_2 = \{1, 2, 3\}$. This introduces the red edge from τ_3 to $\hat{z}_1$, i.e., an expanded signature, in $\hat{B}$, so $\|\hat{B}\|_0 > \|B\|_0$. This example illustrates exactly why regularizing the learned $\hat{B}$ to be sparse can rule out mixing: if $\hat{B}$ is no denser than the true B which is itself sparse, then dependencies that expand a matched signature are excluded. It is exactly how we aim to achieve (partial) disentanglement. Yet, mixing a true factor into another would *not* enlarge the signature of a learned factor if its signature is already contained in that of the other. Indeed, if learned factor s already depends on the true factor z_r, and we additionally allow it to depend on factor z_q, then $\hat{S}_s = S_r \cup S_q$. This union is unchanged exactly when

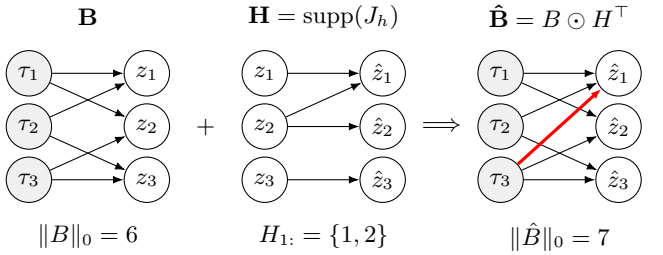

Figure 3: Mixing z_1 and z_2 in $\hat{z}_1 = h_1(z)$ replaces S_1 by $S_1 \cup S_2$ and adds the red edge. Sparsity punishes the denser pattern.

$$S_q \subseteq S_r \qquad \Longleftrightarrow \qquad S_r \cup S_q = S_r. \tag{11}$$

As we show in the next subsection, this phenomenon is not merely a sufficient example of hidden mixing. Under the global sparsity condition, signature inclusion provides an exact characterization of the factor mixing (i.e. dependencies) that could remain in learned representations that are observationally equivalent to the ground-truth.

3.2 Sparsity Allows Only Signature-Contained Mixing

Section 3.1 showed the principle that an extra dependence can evade detection when its signature is already contained in the signature of the learned factor. The remaining question is whether other forms of factor mixing can also survive under the global sparsity condition. The following theorem establishes that the answer is negative.

Theorem 3.2 (Mixing under Sparsity Condition)**.** *Let f and $\hat{f}$ be observationally equivalent, with entanglement map $h = \hat{f}^{-1} \circ f$, functional Jacobian support $H = \mathrm{supp}(J_h)$, and induced active matrix $\hat{B} = B \odot H^\top$. Then*

$$\textit{(Sparsity Condition)} \qquad \textit{Form 1:} \qquad \|\hat{B}\|_0 \le \|B\|_0 \tag{12}$$

$$\Longleftrightarrow \qquad \textit{Form 2:} \qquad \exists\, \pi \in \mathfrak{S}_K \quad \textit{s.t.} \quad \forall\, s, q \in [K], \quad q \in H_{s:} \implies S_q \subseteq S_{\pi(s)}. \tag{13}$$

When the sparsity condition of either form holds, then the corresponding permutation π satisfies

$$\hat{S}_s = S_{\pi(s)}, \qquad s \in [K]. \tag{14}$$

Remark 3.3 (Proof Sketch)**.** *By Definition 2.2, both h and h^{-1} are continuously differentiable. Differentiating $h^{-1} \circ h = \mathrm{Id}_{\mathbb{R}^K}$ gives $J_{h^{-1}}(h(z))J_h(z) = I_K$, $\forall\, z \in \mathbb{R}^K$. Hence $J_h(z)$ is nonsingular for every $z \in \mathbb{R}^K$. Fix $z^0 \in \mathbb{R}^K$. A nonzero term in the determinant expansion of $J_h(z^0)$ gives a permutation $\pi \in \mathfrak{S}_K$ with $\pi(s) \in H_{s:}$ for every $s \in [K]$. By Eq. (10), this implies $S_{\pi(s)} \subseteq \hat{S}_s$, and summing cardinalities yields the universal lower bound $\|B\|_0 \le \|\hat{B}\|_0$. Under Eq. (12), the bound is tight, so every inclusion is an equality: $\hat{S}_s = S_{\pi(s)}$. Hence any $q \in H_{s:}$ satisfies $S_q \subseteq S_{\pi(s)}$. Conversely, Eq. (13) gives $\hat{S}_s = \bigcup_{q \in H_{s:}} S_q \subseteq S_{\pi(s)}$, and therefore $\|\hat{B}\|_0 \le \|B\|_0$. Combining this with the universal lower bound gives Eq. (14). Full proofs in Appendix B.*

By showing equivalent forms, Theorem 3.2 turns the global sparsity constraint (Form 1) into a factor-wise restriction (Form 2). Nonsingularity of J_h yields a one-to-one matching between learned and true factors, with $\pi(s) \in H_{s:}$ for every learned factor s. After this assignment, Eq. (13) says that every other true factor entering the same learned factor must have a signature contained in that of the assigned true factor. Therefore, any extra dependence that introduces a new relation type is ruled out by sparsity. Eq. (14) further explains why such mixing remains undetected: although a learned factor may depend on several true factors, its induced signature is unchanged. Thus, sparsity does not eliminate

mixing altogether, but does exactly the mixing that would expand a learned signature. Theorem 3.2 hence shows that under sparsity the inclusion structure among the factor signatures of B determines which dependencies remain in h.

The matching π is bijective, but this alone does not imply factor-wise identifiability: a learned factor s may depend on true factors other than its matched factor $\pi(s)$. The remaining dependencies are governed by the inclusion structure of the true relation signatures. In the next subsection, we show that different structures of signature inclusion would lead to different levels of identifiability, ranging from unrestricted mixing to block-wise and factor-wise recovery.

3.3 Identifiability from signature structure

Theorem 3.2 reduces the remaining ambiguity to the inclusion structure among the true factor signatures $\{S_r\}_{r=1}^K$. We now present a series of identifiability results which are guaranteed by the inclusion structure, described in the previous subsection. Starting from the strongest separation between signatures, we progressively relax, under the same sparsity condition, the signature structure and track the mixing until it becomes indistinguishable (see Table 1).

Table 1: A series of identifiability results.

Signature regime	Signature structure	Identifiability
Antichain	$S_q \subseteq S_r \Rightarrow q = r$	Factor-wise
Incomparable	$S_q \subseteq S_r \Rightarrow S_q = S_r$	Block-wise
Containment	$\exists\, q \neq r :\ S_q \subsetneq S_r$	Directional
Identical	$S_q = S_r,\ \forall q, r$	Unrestricted

Corollary 3.4 (Factor-wise identifiability)**.** *Suppose the sparsity condition in Theorem 3.2 holds and*

$$\textit{(Antichain)} \qquad S_q \subseteq S_r \quad \Longrightarrow \quad q = r, \qquad \textit{for all } q, r \in [K]. \tag{15}$$

Then the learned representation is factor-wise identifiable in the sense of Definition 2.4.

The proof is given in Appendix C. Under Eq. (15), Theorem 3.2 reduces to $q \in H_{s:} \Rightarrow q = \pi(s)$, so each learned factor depends on a single true factor. We next allow equal signatures while requiring distinct signatures to remain incomparable. Define $q \sim r$ if and only if $S_q = S_r$, and let $\mathcal{P}_B = \{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$ be the partition of $[K]$ into the resulting equivalence classes.

Corollary 3.5 (Block-wise identifiability)**.** *Suppose the sparsity condition in Theorem 3.2 holds and*

$$\textit{(Incomparable Signatures)} \quad S_q \subseteq S_r \quad \Longrightarrow \quad S_q = S_r, \qquad \textit{for all } q, r \in [K]. \tag{16}$$

Then the learned representation is block-wise identifiable with respect to the equal-signature partition $\mathcal{P}_B$ in the sense of Definition 2.5.

The proof is given in Appendix E. Under Eq. (16), Theorem 3.2 gives $q \in H_{s:} \Rightarrow S_q = S_{\pi(s)}$, so each learned factor is confined to one equal-signature block and only intra-block mixing remains. We then ask if the structural condition of B can be further relaxed for a weaker identifiability. The following corollary constructs an entanglement map h that satisfies sparsity yet breaks the block-wise identifiability, which gives an instantiation of such remaining ambiguity.

Corollary 3.6 (Directional ambiguity)**.** *Let f be the ground-truth decoder in Definition 2.1, and suppose*

$$\textit{(Strict Containment)} \qquad S_q \subsetneq S_r, \qquad \textit{for some } q \neq r. \tag{17}$$

For any $\epsilon \neq 0$, define $c : \mathbb{R}^K \to \mathbb{R}^K$ by

$$c_j(z) = \begin{cases} z_r + \epsilon z_q, & j = r, \\ z_j, & j \neq r. \end{cases} \tag{18}$$

The map c is bijective, with $c, c^{-1} \in C^1$, and satisfies Form 2 of Theorem 3.2 with $\pi = \mathrm{Id}_{[K]}$. Hence $\hat{f} := f \circ c^{-1}$ is observationally equivalent to f, with entanglement map c. Its induced active matrix satisfies the sparsity condition, although c mixes factors from two distinct blocks of $\mathcal{P}_B$. Therefore, the sparsity condition cannot guarantee block-wise identifiability with respect to $\mathcal{P}_B$.

The proof is deferred to Appendix F. At the endpoint where all signatures coincide, Form 2 of Theorem 3.2 places no constraints on H. Consequently, the induced matrix $\hat{B}$ satisfies $\|\hat{B}\|_0 = \|B\|_0$ for every entanglement map h allowed by Definition 2.2. The sparsity condition therefore imposes no additional restriction on factor mixing.

At the end of this section, we give a sufficient condition to obtain factor-wise identifiability since signatures could form an antichain only if T is sufficiently large. See proof in Appendix D.

Proposition 3.7 (Relation capacity)**.** *There exists an active matrix $B \in \{0,1\}^{T \times K}$ satisfying the antichain condition in Eq. (15) if and only if $K \leq \binom{T}{\lfloor T/2 \rfloor}$.*

Together, these results show how signature equality and inclusion govern the ambiguities that can survive under the sparsity condition. Yet, they do not show how to design an algorithm which restricts the learned factors according to the results. To this end, Section 4 presents a variational implementation that learns representations and relation-specific activity masks from observed typed pairs under a sparsity constraint.

4 Learning Representations from Typed Relations

The theory motivates learning representations that preserve the observations while restricting which learned factors vary under each relation type and under a generic sparsity condition (Eq. (12)). We implement this theory by jointly learning a variational autoencoder and a set of relation-specific activity masks from typed pairs (x, x', τ). The masks estimate the relation-wise variation patterns of the learned factors and thereby reconstruct $\hat{B}$. To ensure that the variations encoded by B are reflected in the observations and therefore recoverable from the learned representations, we first introduce the following assumption. Further in Appendix A we shall discuss and compare it with similar conditions common in the literature.

Assumption 4.1 (Relation Richness). *If $\hat{B}_{\tau s} = 1$, then there exists (z, z') of type τ such that $h_s(z) \neq h_s(z')$.*

Learning which factors may vary. The encoder $q_\phi(z \mid x)$ is a diagonal Gaussian with mean $\mu(x)$ and factor-wise variances $v(x)$. For each relation type τ, a binary mask $M_{\tau:} \in \{0,1\}^K$ specifies which learned factors would vary. When $M_{\tau s} = 0$, we replace, for both observations x, x', the posterior parameters of learned factor s, i.e., $a_s(x) = (\mu_s(x), v_s(x))$ by their arithmetic averages (Locatello et al., 2020); otherwise, we retain their original parameters:

$$\widetilde{a}_s(x) = M_{\tau s} a_s(x) + (1 - M_{\tau s}) \frac{a_s(x) + a_s(x')}{2},$$

and similarly for x'. The resulting Gaussian posteriors $\widetilde{q}_x$ and $\widetilde{q}_{x'}$ provide latent samples to a decoder $p_\theta(x \mid z)$, which reconstructs the respective observations. Sharing a learned factor prevents its posterior parameters from distinguishing the pair, so reconstruction encourages necessary variation to be retained only in active learned factors but not in the others. Each mask is shared across all pairs of a relation type, enabling learning from repeated examples.

Restricting relation support. Reconstruction alone permits every mask entry to remain active. To constrain the expected total number of active entries in mask to s_*, following Lachapelle et al. (2026), we learn independent Bernoulli mask entries with probabilities $\Pi_{\tau s}$. The differentiable objective is obtained by using the Gumbel–Softmax relaxation (Jang et al., 2017; Maddison et al., 2017) to obtain soft masks $\widetilde{M}$ which allows back-propagated gradients. The joint objective for the encoder, decoder, and mask probabilities is given as follows:

$$\min_{\theta,\phi,\Pi} \quad \mathbb{E}_{(x,x',\tau),\,\widetilde{M}} \left[\frac{\ell(x, \widetilde{q}_x) + \ell(x', \widetilde{q}_{x'})}{2} \right] \quad \text{s.t.} \quad \sum_{\tau,s} \Pi_{\tau s} \leq s_\star,$$

$$\text{where} \quad \ell(x, q) = -\mathbb{E}_{z \sim q} \log p_\theta(x \mid z) + \beta D_{\mathrm{KL}}(q \,\|\, \mathcal{N}(0, I_K)) \,.$$

5 Experiments

So far, in Section 3 we have answered when and to what extent typed relations can identify latent factors, and in Section 4 how to learn representations from these relations. We now examine whether the theoretical insights and the method translate into improvements in factor recovery. Section 5.1 uses controlled experiments with C^1 models to test whether sparsity suppresses the dependencies excluded by the theory and leaves the predicted mixing ambiguities unresolved. Section 5.2 then extends this to image benchmarks when theoretical conditions don't hold strictly, where we also examine our method and ask if it's helpful to learn disentangled representations, comparing it with baseline. Codes and data are anonymously available at: https://anonymous.4open.science/r/when-do-relations-reveal-factors/.

Evaluation metrics. We measure factor recovery using DCI-D (Eastwood & Williams, 2018) following Locatello et al. (2020), and also report MCC with optimal permutation (Khemakhem et al., 2020b). However, as these metrics only measure the performance of factor-wise identifiability, they do not allow evaluation of forbidden mixing since our Theorem 3.2 shows that a learned factor matched to true factor r cannot depend on factor q whenever $S_q \not\subseteq S_r$. Therefore, we calculate the $K \times K$ relative dependency (RD) matrix D with optimal permutation, where D_{rq} indicates how much the learned factor r depends upon the true factor q, then the disentanglement would require maximizing all diagonal entries when $r = q$. As such, the K^2 entries of D split into three disjoint sets: the diagonal $\mathcal{D} = \{(r, r)\}$, the off-diagonal dependence $\mathcal{P} = \{(r, q) : r \neq q,\ S_q \subseteq S_r\}$ that our theory permits, and the dependence $\mathcal{F} = \{(r, q) :$

$S_q \not\subseteq S_r\}$ it forbids, for which we introduce the *relative diagonal dependence* (RDD), *relative forbidden dependence* (RFD) and the *relative permitted dependence* (RPD) metrics, calculated as follows. See more details in Appendix G.1.

$$\text{RDD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{D}} D_{rq}, \qquad \text{RFD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{F}} D_{rq}, \qquad \text{RPD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{P}} D_{rq}. \tag{19}$$

5.1 Controlled Test of Signature Inclusion

Setup. Since Prop. 3.7 specifies the minimum number of relation types needed for the signatures to form an antichain, we set $K = 6$ independent factors and $T = 4$ relationship types to test this boundary case. Following Zheng et al. (2022); Zheng & Zhang (2023), we work in the undercomplete case with twice as many observed variables as true factors, generating $x \in \mathbb{R}^{12}$ from $z \in \mathbb{R}^6$ with a frozen random flow of 10 GLOW blocks (Sorrenson et al., 2020) using SiLU such that f is C^1. A relation pair of type τ changes exactly the factors with $B_{\tau q} = 1$ and preserves the others (see Figure 4 top row), so x and x' differ only through the factors activated by τ. For our method, we use a 3-layer MLP encoder of width 256 and implement the sparsity constraint using COOPER (Gallego-Posada et al., 2025) with budget $s^* = \|B\|_0$ as true support number. For each regime, we report the mean and standard deviation (MEAN $\pm$ STD) across 10 observation maps, each averaging 3 runs. For the complete experimental setup, see Appendix G.2.

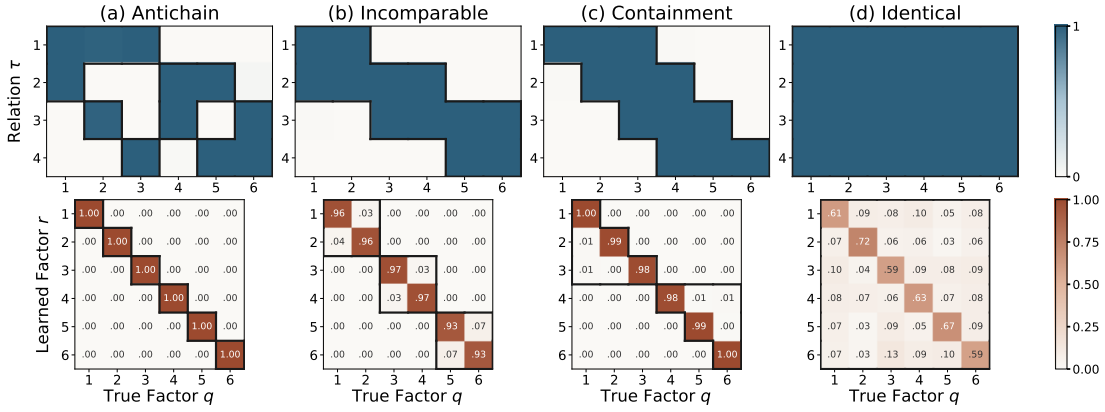

Figure 4: **Learned structure follows signature inclusion.** Learned relation mask (top) and relative dependency (bottom). Black contours mark the true support B (top) and the permitted dependency (bottom). Columns are true factors q, rows are true-factor aligned factors r, and each row sums to one. RDD, RPD and RFD are the average share of each row of D on the diagonal, inside the black contour off the diagonal, and outside the contour, respectively.

Results. For each regime, our theory predicts which dependencies are forbidden and which are permitted. We check *(i) whether the forbidden dependencies are suppressed, and whether the sparsity constraint is what suppresses them, and (ii) whether the permitted dependencies remain, and to what extent.* For (i), our method (*w/ sps.*) keeps RFD near zero in the three regimes that forbid any dependence (Table 2). The identical regime forbids nothing, so its RFD is not defined. However, a learner biased towards factor-wise solutions would also attain a low RFD, so we also evaluate, against the other regimes, the models trained under identical signatures, where the sparsity constraint is inactive (*w/o sps.*). See the last paragraph in Appendix G.2. For all the experiments, their RFD is more than two orders of magnitude higher than our method, the suppression therefore comes from the sparsity constraint. For (ii), the dependencies concentrate on the diagonal in the antichain regime, where Cor. 3.4 permits no cross-factor dependence. In the incomparable and containment regimes, they also concentrate on the diagonal, leaving only a small share of the permitted dependence predicted by Cor. 3.5, 3.6. In the identical regime, by contrast, RPD accounts for a substantial share, so much of the ambiguity the theory permits remains. See more complete setup and more results to Appendix G.2.

Table 2: **Dependence follows signature inclusion.** Diagonal, permitted and forbidden dependencies of our method; each row of D sums to one, so RDD + RPD + RFD = 1. RFD is in units of 10^{-3}.

	RDD ↑	RPD	RFD ↓	
			w/ sps.	w/o sps.
Antichain	0.999 ± 0.001	–	1.4 ± 0.8	369 ± 149
Incomparable	0.953 ± 0.048	0.045 ± 0.049	1.0 ± 0.5	278 ± 95
Containment	0.992 ± 0.003	0.007 ± 0.003	0.9 ± 0.5	302 ± 127
Identical	0.635 ± 0.143	0.365 ± 0.143	–	–

5.2 Image Benchmarks

Setup. Following Locatello et al. (2019; 2020); Roth et al. (2023), we adopt widely used datasets Shapes3D (Kim & Mnih, 2018), dSprites (Higgins et al., 2017), and MPI3D-Real (Gondal et al., 2019). For each dataset, we keep the

Table 3: **Dependence on Shapes3D** ($d = K = 6$)**.** MEAN $\pm$ STD over 20 seeds (true mask: 3 seeds). **Bold** indicates statistically significant improvement over both Ada-GVAE and Ours (w/o sps.) (Wilcoxon signed-rank test $p < 0.05$)

	Antichain		**Incomparable**			**Containment**		
	RDD ↑	RFD ↓	RDD ↑	RPD	RFD ↓	RDD ↑	RPD	RFD ↓
Ours (oracle)	0.85 ± 0.19	0.15 ± 0.19	0.89 ± 0.07	0.11 ± 0.07	0.00 ± 0.00	0.85 ± 0.05	0.12 ± 0.01	0.02 ± 0.04
Ours	$\mathbf{0.65 \pm 0.09}$	$\mathbf{0.35 \pm 0.09}$	$\mathbf{0.58 \pm 0.07}$	0.11 ± 0.06	$\mathbf{0.31 \pm 0.09}$	$\mathbf{0.61 \pm 0.12}$	0.03 ± 0.04	$\mathbf{0.36 \pm 0.14}$
Ours (w/o sps.)	0.52 ± 0.06	0.48 ± 0.06	0.42 ± 0.11	0.20 ± 0.12	0.38 ± 0.07	0.45 ± 0.11	0.03 ± 0.03	0.52 ± 0.13
Ada-GVAE	0.57 ± 0.07	0.43 ± 0.07	0.31 ± 0.11	0.19 ± 0.06	0.50 ± 0.11	0.53 ± 0.03	0.02 ± 0.04	0.45 ± 0.02

VAE latent dimension d to the number of nonconstant factors, e.g., $d = 6$ in Shapes3D, when verifying our theory that requires $d = K$. When $d > K$, we let $d = 10$ for fair comparison following the above mentioned papers. For the latter, we report DCI-D and defer MCC to the Appendix G.3.3 since it's ill-suited when $d > K$ (Joshi et al., 2026). The observed relations are generated following protocols in Sec. 5.1 (see details in Appendix G.3). We use the code base by Roth et al. (2023) to reproduce Ada-GVAE (Locatello et al., 2020), and unify the VAE architecture for all methods, including ours, to the Locatello et al. (2020) one. To measure the upper limit of our method's performance, we additionally replace the learned mask with the true one when $d = K$. The performance comparison is based on Wilcoxon signed-rank test (Wilcoxon, 1945). For each run, we train 600,000 steps with 128 images (64 pairs) per step and evaluate the best checkpoint on validation set. When $d = K$, the sparsity constraint grid collapses to $\|B\|_0$, so only β is selected, shared by the three regimes, and ours (incl. w/o sps. and oracle) use the selected β. When $d > K$, each method's β is chosen out of $\beta \in \{1, 6, 16\}$ by validation ELBO on the first 5 seeds and then run on the remaining 15 with the best β. For our method, we select jointly over that grid and a constraint budget grid $s_* = \|B\|_0 + \rho\,(d - K)$ with $\rho \in \{0, 1, 2\}$. Full details are given in the Appendix G.3.

Results. Table 3 shows that our method obtains consistently better RFD than Ada-GVAE and our ablation without sparsity, as predicted by our theory, which is consistent with our conclusion in controlled setting. We observe also that our method's RDD is not perfect RDD even it uses the true mask. We however argue that it is caused by the limited latent dimension, i.e., $d = K$, since, in Shapes3D, for instance, none of the hue factors can be represented using a single dimension under current inductive bias (see Appendix G.3.4). We now ask what would happen if $d > K$. As shown in Table 4, across all datasets our method achieves consistently better disentanglement. Ada-GVAE reaches lower absolute DCI-D than reported by Locatello et al. (2020) because they resample k factors, so a pair differs in *at most* k factors and k is 1 or random, whereas every pair here changes exactly the factors of row τ of B (three of six on Shapes3D). Their Figure 10 shows that Ada-GVAE degrades as more factors change, which is consistent with what we observe. Notably, Dai & Wipf (2019) predict that a VAE keeps only as many active dimensions as the data manifold requires and pushes the rest to the prior. Table 16 shows that our method does prune: out of 10 dimensions, 8.25 are active on Shapes3D, 7.10 on dSprites and 7.30 on MPI3D-Real, while Ada-GVAE keeps all ten active. We further investigate what factors would occupy more than 1 dimension, concluding that latent factor z_k with greater influence on pixel space (x) tends to be represented across multiple latent dimensions. See complete results in Appendix G.3.4.

Table 4: Performance measured by DCI-D ($d = 10 > K$). MEAN $\pm$ STD across 20 seeds. **Bold** indicates statistically better DCI-D.

Dataset	Method	DCI-D ↑
Shapes3D	Ada-GVAE	0.840 ± 0.064
	Ours	$\mathbf{0.910 \pm 0.070}$
dSprites	Ada-GVAE	0.317 ± 0.071
	Ours	$\mathbf{0.628 \pm 0.078}$
MPI3D-Real	Ada-GVAE	0.367 ± 0.019
	Ours	$\mathbf{0.466 \pm 0.056}$

6 Discussion and Limitations

Understanding when typed relations reveal the underlying factors is a fundamental question for overcoming non-identifiability. This work provides a precise characterization of this question. We show that mixing corresponds to taking unions of relation signatures, that sparsity restricts admissible mixing to signatures already containing the relevant factors, and that the partial order structure among signatures fully determines the attainable level of identifiability. These theoretical predictions are validated in controlled experiments, and translate into practical improvements over existing baselines on image benchmarks, even when the assumptions of the theory are only approximately satisfied.

Limitations Our identifiability theory builds on analysis over the ground-truth active factor matrix B. While our method obtains better-disentangled representations, the observed training dynamics, especially under constrained optimization, are unstable, which we evidence and discuss in G.3.5. Therefore, constructing more stable and consistent estimators, e.g., by better understanding and learning the inductive biases of the current method, or via statistical modeling over relation generation, remains a research gap, which we leave for the future work.

AI USE STATEMENT

We used large language models (LLMs) to assist with writing and editing, literature search, research brainstorming, experimental design, and mathematical reasoning, including suggesting proof strategies and identifying potential gaps in theoretical arguments. All AI-assisted content was critically reviewed and revised by the authors. We take full responsibility for the correctness and originality of the theoretical results, experimental findings, and final manuscript.

ETHICS STATEMENT

We focus on the foundational research on identifiability in representation learning which involves no human subjects, no personal or sensitive data, and no user study. The image experiments use three established public datasets, i.e., Shapes3D (Kim & Mnih, 2018), dSprites (Higgins et al., 2017) and MPI3D-Real (Gondal et al., 2019), under their original licences, and we redistribute none of them, and our code release documents how to obtain each one. We declare no conflicts of interest and no sponsorship that constrained the design, reporting or release of this work.

REPRODUCIBILITY STATEMENT

Each assumption is stated with the result it supports, and complete proofs of all theoretical claims are given in Appendices B–F, one appendix per claim. Appendix G specifies the experiments in full. An anonymised repository with the complete source for both experiments is available at `https://anonymous.4open.science/r/when-do-relations-reveal-factors/`. Its `REPRODUCE.md` maps each figure and table in this paper to the single command that regenerates it from those records, needing neither a GPU nor the datasets. We verified from a fresh clone that the figures come out pixel-identical and the tables byte-identical to the submitted ones.

REFERENCES

Lynton Ardizzone, Till Bungert, Felix Draxler, Ullrich Köthe, Jakob Kruse, Robert Schmier, and Peter Sorrenson. Framework for Easily Invertible Architectures (FrEIA). GitHub repository, 2018-2022. URL https://github.com/vislearn/FrEIA.

Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45419–45462. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8e5de4cb639ef718f44060dc257cb04f-Abstract-Conference.html.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.00519.

David F. Crouse. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016. URL https://doi.org/10.1109/TAES.2016.140952.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1e0X3C9tQ.

Barthélémy Dang-Nhu, Louis Annabi, and Sylvain Argentieri. Disentangled representation learning through unsupervised symmetry group discovery. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=I6xjMoLY3j.

Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By-7dz-AZ.

Marco Fumero, Luca Cosmo, Simone Melzi, and Emanuele Rodolà. Learning disentangled representations via product manifold projection. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3530–3540. PMLR, 2021. URL https://proceedings.mlr.press/v139/fumero21a.html.

Jose Gallego-Posada, Juan Ramirez, Meraj Hashemizadeh, and Simon Lacoste-Julien. Cooper: A library for constrained optimization in deep learning. arXiv:2504.01212, 2025. URL https://arxiv.org/abs/2504.01212.

Muhammad Waleed Gondal, Manuel Wüthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: A new disentanglement dataset. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/d97d404b6119214e4a7018391195240a-Abstract.html.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/d305281faf947ca7acade9ad5c8c818c-Abstract.html.

Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 460–469. PMLR, 2017. URL https://proceedings.mlr.press/v54/hyvarinen17a.html.

Aapo Hyvärinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 859–868. PMLR, 2019. URL https://proceedings.mlr.press/v89/hyvarinen19a.html.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Shruti Joshi, Théo Saulus, Wieland Brendel, Philippe Brouillard, Dhanya Sridhar, and Patrik Reizinger. Who guards the guardians? The challenges of evaluating identifiability of learned representations. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 2618–2660. PMLR, 2026. URL https://proceedings.mlr.press/v337/joshi26a.html.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2207–2217. PMLR, 2020a. URL https://proceedings.mlr.press/v108/khemakhem20a.html.

Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvärinen. ICE-BeeM: Identifiable conditional energy-based deep models based on nonlinear ICA. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12768–12778. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/962e56a8a0b0420d87272a682bfd1e53-Abstract.html.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658. PMLR, 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E. Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Proceedings of the 1st Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 428–484. PMLR, 2022. URL https://proceedings.mlr.press/v177/lachapelle22a.html.

Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Nonparametric partial disentanglement via mechanism sparsity: Sparse actions, interventions and sparse temporal dependencies. *Journal of Machine Learning Research*, 27(71):1–90, 2026. URL https://jmlr.org/papers/v27/24-0771.html.

Fan Li, Zixuan Liu, Yi Zhao, Qi Tan, Jinyang Li, Ke Xu, Shuihai Hu, Jingbin Zhou, and Tan Kun. Influence without confounding: Causal discovery from temporal data with long-term carry-over effects. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=s0nYSwlV3I.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 2019. URL https://proceedings.mlr.press/v97/locatello19a.html.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6348–6359. PMLR, 2020. URL https://proceedings.mlr.press/v119/locatello20a.html.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Stefan Matthes, Zhiwei Han, and Hao Shen. Mechanistic independence: A principle for identifiable disentangled representations. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0VVdai71xb.

Weimin Meng, Rohit Inampudi, Xiang Zhang, Jie Xu, Yu Huang, Mingyi Xie, Jiang Bian, and Rui Yin. An interpretable population graph network to identify rapid progression of Alzheimer's disease using UK Biobank. *AMIA Annual Symposium Proceedings*, 2024:808–817, 2024. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12099444/.

Milton Llera Montero, Casimir J. H. Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qbH974jKUVy.

Walter Nelson, Marco Fumero, Theofanis Karaletsos, and Francesco Locatello. Statistical and structural identifiability in representation learning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Wa3cfE3Iay.

Vaishnavi S. Patil, Siddhi Patil, Matthew Evanusa, Amit Kumar Kundu, Cornelia Fermuller, and Joseph JaJa. A Bayesian nonparametric framework for learning disentangled representations. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GVOLiaENgU.

Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. Disentanglement of correlated factors via Hausdorff factorized support. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OKcJhpQiGiX.

Peter Sorrenson, Carsten Rother, and Ullrich Köthe. Disentanglement by nonlinear ICA with general incompressible-flow networks (GIN). In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygeHgSFDH.

Emanuel Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928. URL https://doi.org/10.1007/BF01171114.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6056–6065. PMLR, 2019. URL https://proceedings.mlr.press/v97/suter19a.html.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. URL https://doi.org/10.1038/s41592-019-0686-2.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. URL https://doi.org/10.2307/3001968.

Hyung-Jeong Yang. Integrating knowledge graph and causal generative models for predicting risk of obstructive sleep apnea and cardiovascular disease. UK Biobank Research Project 1119620, 2026. URL https://www.ukbiobank.ac.uk/projects/integrating-knowledge-graph-and-causal-generative-models-for-predicting-risk-of-obstructive-sleep-apnea-and-cardiovascular-disease/. Accessed 6 August 2026.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9593–9602, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yang_CausalVAE_Disentangled_Representation_Learning_via_Neural_Structural_Causal_Models_CVPR_2021_paper.html.

Yujia Zheng and Kun Zhang. Generalizing nonlinear ICA beyond structural sparsity. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13326–13355. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2aebc17b683792a17dd4a24fcb038ba6-Abstract-Conference.html.

Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ICA: Sparsity and beyond. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16411–16422. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6801fa3fd290229efc490ee0cf1c5687-Abstract-Conference.html.

A Comparison with other sparsity principle works

Our result builds on the same broad sparsity principle as Lachapelle et al. (2022; 2026) and Zheng et al. (2022); Zheng & Zhang (2023): no entangled representation should admit a sparser explanation of the observations than the true representation. Realising this principle, however, requires more than a sparsity criterion alone. In Theorem 1(ii) of Zheng et al. (2022), the Jacobian row of each observed coordinate, evaluated at sufficiently many source points, is required to span the source-coordinate subspace specified by its true support. The assumption additionally constrains the transformed Jacobian support after multiplication by the change-of-coordinate matrix. Together, these conditions prevent degeneracies that would invalidate the required support-transfer argument. In the mechanism-sparsity setting, Lachapelle et al. (2022; 2026) impose a *sufficient influence* condition requiring that variation in an auxiliary variable induces sufficiently many linearly independent changes in the conditional score of every mechanism it affects.

Figure 5 illustrates the issue. In panel (a), repeatedly observing the same derivative vector cannot separate the contributions of z_1 and z_2. A learned factor change can therefore make one apparent dependence vanish. Panel (b) shows the analogous failure on the mechanism side. Although the auxiliary variable affects both latent learned factors, it does so only along the single direction $(1,1)^\top$. After rotating to the sum and difference learned factors, all auxiliary dependence is absorbed into the sum learned factor, producing a strictly sparser learned graph without recovering the original latent learned factors. These examples do not assert that influence richness is the only route to identifiability. Instead, they illustrate why it is required by the support-transfer arguments that underpin the corresponding identifiability proofs.

(a) Fixed decoder influence. Let

$$f_1(z_1, z_2) = z_1 + z_2, \qquad \nabla f_1 = (1, 1).$$

The gradient never supplies two independent directions. Under $u_1 = (z_1 + z_2)/\sqrt{2}$ and $u_2 = (z_1 - z_2)/\sqrt{2}$, the same output is $f_1 = \sqrt{2}\, u_1$; the dependence on u_2 disappears by cancellation.

(b) Insufficient auxiliary influence. Suppose

$$z \mid a \sim \mathcal{N}\Big(a(1,1)^\top, I_2\Big).$$

Then $\partial_a \nabla_z \log p(z \mid a) = (1,1)^\top$, so the response to a spans only one direction. With the same rotation,

$$u \mid a \sim \mathcal{N}\Big((\sqrt{2}a, 0)^\top, I_2\Big).$$

Thus the original two auxiliary-to-latent edges become one edge in the rotated, entangled learned factors. A sparsity objective alone would prefer the latter.

Figure 5: Two elementary failure modes motivating influence-richness assumptions. (a) illustrates the cancellation excluded by the Jacobian-spanning condition of Zheng et al. (2022). (b) gives a Gaussian failure of sufficient influence in Lachapelle et al. (2026).

Our theorem obtains the required support inclusion directly from the typed relation model. Each true factor r is assigned a binary relation signature $S_r = \operatorname{supp} B_{:r}$. If learned factor h_s depends on the factors in $H_{s:}$, its active relation set is, by the definition of the learned active factor matrix,

$$\hat{S}_s = \bigcup_{r \in H_{s:}} S_r.$$

Under the relation richness condition, i.e., Assumption 4.1, every active entry of this Boolean union is witnessed by the corresponding relation type, ruling out support loss due to degenerate co-variation of active factors. Consequently, once invertibility of h provides a distinct matched factor $\pi(s) \in H_{s:}$ for every learned factor, we immediately obtain $S_{\pi(s)} \subseteq \hat{B}_{:s}$. Under the sparsity condition of Theorem 3.2, the resulting global support lower bound is tight. Under the antichain condition of Corollary 3.4, adding any factor $q \neq \pi(s)$ contributes at least one relation type absent from $S_{\pi(s)}$, so the union becomes strictly larger. Sparsity therefore excludes every extra factor, yielding $H_{s:} = \{\pi(s)\}$ and hence factor-wise identifiability. The role of relation richness is thus analogous to the sufficient influence or spanning conditions above: it guarantees faithful support transfer. Conditional on this transfer, the remaining ambiguity is determined entirely by the inclusion structure of the relation signatures.

B Proof of Theorem 3.2

Theorem 3.2 (Mixing under Sparsity Condition). *Let f and $\hat{f}$ be observationally equivalent, with entanglement map $h = \hat{f}^{-1} \circ f$, functional Jacobian support $H = \operatorname{supp}(J_h)$, and induced active matrix $\hat{B} = B \odot H^\top$. Then*

$$\textit{(Sparsity Condition)} \quad \textit{Form 1:} \quad \|\hat{B}\|_0 \le \|B\|_0 \tag{12}$$

$$\iff \quad \textit{Form 2:} \quad \exists\, \pi \in \mathfrak{S}_K \ \ \textit{s.t.} \ \ \forall\, s, q \in [K], \quad q \in H_{s:} \implies S_q \subseteq S_{\pi(s)}. \tag{13}$$

When the sparsity condition of either form holds, then the corresponding permutation π satisfies

$$\hat{S}_s = S_{\pi(s)}, \qquad s \in [K]. \tag{14}$$

Proof. The proof follows the same conceptual progression as the proof sketch in the main text. We first leverage invertibility to match each learned factor to a unique true factor, and then translate this match into a universal lower bound on relation support. The sparsity condition makes this lower bound tight for each factor, producing the signature-containment rule. This proves Eq. (12) ⇒ Eq. (13) and Eq. (14). We conclude by proving the converse, i.e., Eq. (13) ⇒ Eq. (12).

Step 1: invertibility matches learned factors to distinct true factors. By Definition 2.2, $h : \mathbb{R}^K \to \mathbb{R}^K$ is bijective and $h, h^{-1} \in C^1$. Hence h^{-1} is C^1, and differentiating $h^{-1} \circ h = \mathrm{Id}_{\mathbb{R}^K}$ gives

$$J_{h^{-1}}(h(z))J_h(z) = I_K, \qquad z \in \mathbb{R}^K. \tag{20}$$

Therefore $J_h(z)$ is nonsingular, i.e., has a unique inverse, for every $z \in \mathbb{R}^K$. Suppose we fix any $z^0 \in \mathbb{R}^K$. Then the $\det J_h(z^0) \neq 0$ holds and the following Leibniz expansion contains at least one nonzero summand:

$$\det J_h(z^0) = \sum_{\pi \in \mathfrak{S}_K} \operatorname{sgn}(\pi) \prod_{s=1}^{K} \frac{\partial h_s}{\partial z_{\pi(s)}}(z^0) \tag{21}$$

Then there exists a permutation $\pi \in \mathfrak{S}_K$ such that

$$\frac{\partial h_s}{\partial z_{\pi(s)}}(z^0) \neq 0, \qquad s \in [K]. \tag{22}$$

Each derivative in Eq. (22) is therefore not identically zero on $\mathbb{R}^K$. By Definition 2.3,

$$\pi(s) \in H_{s:}, \qquad s \in [K]. \tag{23}$$

Because π is a permutation, the matched true factors are distinct across learned factors.

Step 2: the matching gives a universal lower bound on relation support. For every learned factor s, the signature-union identity Eq. (10) gives

$$\hat{S}_s = \bigcup_{r \in H_{s:}} S_r. \tag{24}$$

Since $\pi(s) \in H_{s:}$ by Eq. (23),

$$S_{\pi(s)} \subseteq \hat{S}_s, \qquad s \in [K]. \tag{25}$$

Summing the cardinality over s yields

$$\begin{aligned} \|B\|_0 = \sum_{r=1}^{K} |S_r| &= \sum_{s=1}^{K} |S_{\pi(s)}| \\ &\leq \sum_{s=1}^{K} |\hat{S}_s| = \|\hat{B}\|_0. \end{aligned} \tag{26}$$

The second equality follows from the fact that π is a permutation. Thus every observationally equivalent representation satisfies the universal lower bound $\|B\|_0 \leq \|\hat{B}\|_0$.

Step 3: Form 1 makes every matched-signature inclusion tight. Assume Form 1, namely Eq. (12). Combining it with Eq. (26) gives

$$\|\hat{B}\|_0 = \|B\|_0. \tag{27}$$

For each s, define

$$\Delta_s := |\hat{S}_s| - |S_{\pi(s)}| \geq 0,$$

where nonnegativity follows from Eq. (25). Using Eq. (27) and the permutation property,

$$\sum_{s=1}^{K} \Delta_s = \|\hat{B}\|_0 - \|B\|_0 = 0.$$

As every Δ_s is non-negative, $\Delta_s = 0$ for every s. Since Eq. (25) is an inclusion between finite sets of equal cardinality,

$$\hat{S}_s = S_{\pi(s)}, \qquad s \in [K]. \tag{28}$$

This proves Eq. (14).

Now take any $q \in H_{s:}$. By Eq. (24), $S_q \subseteq \hat{S}_s$; combining this with Eq. (28) yields

$$S_q \subseteq S_{\pi(s)}.$$

Thus the permutation constructed in Step 1 satisfies Form 2, proving Eq. (12)⇒Eq. (13).

Step 4: Form 2 implies Form 1. Conversely, suppose there exists $\pi \in \mathfrak{S}_K$ satisfying Form 2, i.e., Eq. (13). Then for every $s \in [K]$, the signature-union identity gives

$$\hat{S}_s = \bigcup_{q \in H_{s:}} S_q \subseteq S_{\pi(s)}. \tag{29}$$

Therefore

$$\|\hat{B}\|_0 = \sum_{s=1}^{K} |\hat{S}_s| \leq \sum_{s=1}^{K} |S_{\pi(s)}| = \|B\|_0,$$

which is exactly Form 1, Eq. (12). Combining this upper bound with the universal lower bound Eq. (26) gives equality of the two support counts:

$$\|\hat{B}\|_0 = \|B\|_0.$$

Since each inclusion in Eq. (29) is between finite sets and the sums of their cardinalities are equal, every inclusion is tight:

$$\hat{S}_s = S_{\pi(s)}, \qquad s \in [K].$$

Thus Eq. (14) also holds for the same witness π, completing the proof. □

C Proof of Corollary 3.4

Corollary 3.4 (Factor-wise identifiability)**.** *Suppose the sparsity condition in Theorem 3.2 holds and*

$$\textit{(Antichain)} \qquad S_q \subseteq S_r \implies q = r, \qquad \textit{for all } q, r \in [K]. \tag{15}$$

Then the learned representation is factor-wise identifiable in the sense of Definition 2.4.

Proof. By the assumed sparsity condition, Theorem 3.2 provides a permutation $\pi \in \mathfrak{S}_K$ such that

$$q \in H_{s:} \implies S_q \subseteq S_{\pi(s)}, \qquad s, q \in [K]. \tag{30}$$

Step 1: the antichain condition removes every cross-factor dependence. Combining Eq. (15) with Eq. (30) gives

$$q \in H_{s:} \implies q = \pi(s).$$

Hence $H_{s:} \subseteq \{\pi(s)\}$. Since $h, h^{-1} \in C^1$, the identity $J_{h^{-1}}(h(z))J_h(z) = I_K$ shows that $J_h(z)$ is nonsingular for every $z \in \mathbb{R}^K$. In particular, no row of J_h can vanish identically. Therefore $H_{s:} \neq \varnothing$ and

$$H_{s:} = \{\pi(s)\}, \qquad s \in [K]. \tag{31}$$

Thus each learned factor can depend on only its matched true factor.

Step 2: singleton functional support gives factor-wise factorization. For every $q \neq \pi(s)$, we have $\partial h_s / \partial z_q = 0$ everywhere on $\mathbb{R}^K$. Let e_r denote the r-th standard basis vector and define $d_s(t) := h_s(te_{\pi(s)})$. Changing the true factors z_q, $q \neq \pi(s)$, one at a time along line segments leaves h_s unchanged. Hence

$$h_s(z) = d_s(z_{\pi(s)}), \qquad s \in [K], \quad z \in \mathbb{R}^K, \tag{32}$$

where each $d_s : \mathbb{R} \to \mathbb{R}$ is C^1.

Step 3: each factor map has a continuously differentiable inverse. Let $P_\pi z = (z_{\pi(1)}, \ldots, z_{\pi(K)})$ and $d(u) = (d_1(u_1), \ldots, d_K(u_K))$. Then Eq. (32) gives $h = d \circ P_\pi$, so $d = h \circ P_\pi^{-1}$ is bijective since h is bijection by Definition 2.2 and P_π is bijection by construction. Because d acts separately on the factors, each d_s is bijective: failure of injectivity or surjectivity of any d_s would imply the corresponding failure for d. Moreover, as established in Step 1, $J_d(u) = \operatorname{diag}\big(d_1'(u_1), \ldots, d_K'(u_K)\big)$ is nonsingular everywhere, since $J_d(u) = J_h(P_\pi^{-1}u)P_\pi^{-1}$. Thus $d_s'(t) \neq 0$ for all s and t. The inverse function theorem gives a C^1 inverse for each d_s, proving factor-wise identifiability.

□

D Proof of Proposition 3.7

Proposition 3.7 (Relation capacity). *There exists an active matrix $B \in \{0,1\}^{T\times K}$ satisfying the antichain condition in Eq. (15) if and only if $K \le \binom{T}{\lfloor T/2\rfloor}$.*

Proof. Each factor signature S_r is a subset of $[T]$. Eq. (15) therefore says exactly that the K columns of B, viewed as subsets $S_1, \dots, S_K \subseteq [T]$, form an antichain in the Boolean lattice $2^{[T]}$.

Necessity. By Sperner's theorem (Sperner, 1928), every antichain in $2^{[T]}$ has cardinality at most

$$\binom{T}{\lfloor T/2\rfloor}.$$

Thus the existence of an active matrix satisfying Eq. (15) requires

$$K \le \binom{T}{\lfloor T/2\rfloor}.$$

Sufficiency. Conversely, suppose $K \le \binom{T}{\lfloor T/2\rfloor}$. The family

$$\mathcal{A} := \{S \subseteq [T] : |S| = \lfloor T/2\rfloor\}$$

has cardinality $\binom{T}{\lfloor T/2\rfloor}$. Since all its members have the same cardinality, no two distinct members can contain one another. Then, given that $K \le \binom{T}{\lfloor T/2\rfloor}$, we can choose K distinct sets $S_1, \dots, S_K \in \mathcal{A}$ and define $B \in \{0,1\}^{T\times K}$ as

$$B_{\tau r} = 1 \quad \Longleftrightarrow \quad \tau \in S_r.$$

Then the columns of B have signatures $S_1, \dots, S_K$, which satisfy Eq. (15). Hence such an active matrix exists.

The two directions establish the equivalence in Proposition 3.7. □

E Proof of Corollary 3.5

Corollary 3.5 (Block-wise identifiability). *Suppose the sparsity condition in Theorem 3.2 holds and*

$$\textit{(Incomparable Signatures)} \quad S_q \subseteq S_r \quad \Longrightarrow \quad S_q = S_r, \qquad \textit{for all } q, r \in [K]. \tag{16}$$

Then the learned representation is block-wise identifiable with respect to the equal-signature partition $\mathcal{P}_B$ in the sense of Definition 2.5.

Proof. We first show that, after reordering the learned factors, each output block depends only on the corresponding input block. We then show that each block map is bijective with a C^1 inverse.

By Theorem 3.2, there exists $\pi \in \mathfrak{S}_K$ such that

$$q \in H_{s:} \quad \Longrightarrow \quad S_q \subseteq S_{\pi(s)}, \qquad s, q \in [K]. \tag{33}$$

Step 1: factors from different signature blocks cannot mix. The assumed signature condition turns the inclusion in Eq. (33) into $S_q = S_{\pi(s)}$. Recall that $\mathcal{P}_B = \{\mathcal{G}_1, \dots, \mathcal{G}_M\}$, defined in Section 3.3, groups together true factors with identical signatures. Thus every true factor on which h_s depends belongs to the same block as its assigned true factor $\pi(s)$. To place each learned factor in the position of its assigned true factor, reorder the outputs of h by setting

$$\bar{h}_r := h_{\pi^{-1}(r)}, \qquad r \in [K].$$

This is only an output permutation, so $\bar{h}$ remains bijective, with $\bar{h}, \bar{h}^{-1} \in C^1$. In this ordering, if $r \in \mathcal{G}_a$, every true factor entering $\bar{h}_r$ has signature S_r and therefore lies in $\mathcal{G}_a$. By the definition of functional Jacobian support,

$$\frac{\partial \bar{h}_r}{\partial z_q}(z) = 0, \qquad r \in \mathcal{G}_a, \quad q \notin \mathcal{G}_a, \quad z \in \mathbb{R}^K.$$

Step 2: construct a C^1 map for each block. Fix a block $\mathcal{G}_a$. Any two inputs $z, z' \in \mathbb{R}^K$ with $z_{\mathcal{G}_a} = z'_{\mathcal{G}_a}$ can be connected by changing the factors outside $\mathcal{G}_a$ one at a time. Along each such line segment, the derivatives above vanish, so $\bar{h}_{\mathcal{G}_a}(z) = \bar{h}_{\mathcal{G}_a}(z')$. Hence the output block is determined entirely by $z_{\mathcal{G}_a}$.

We can therefore define $d_a(u)$ by evaluating $\bar{h}_{\mathcal{G}_a}$ with $z_{\mathcal{G}_a} = u$ and all other input factors set to zero. This defines a C^1 map $d_a : \mathbb{R}^{|\mathcal{G}_a|} \to \mathbb{R}^{|\mathcal{G}_a|}$, because $\bar{h}$ is C^1. The preceding argument ensures that, for arbitrary inputs,

$$\bar{h}_{\mathcal{G}_a}(z) = \big(h_{\pi^{-1}(r)}(z)\big)_{r \in \mathcal{G}_a} = d_a(z_{\mathcal{G}_a}), \qquad a \in [M], \quad z \in \mathbb{R}^K.$$

Step 3: each block map is bijective and has a C^1 inverse. The factorization above shows that $\bar{h}$ transforms each block separately. If d_a mapped two distinct inputs to the same output, keeping all other input blocks fixed would give two distinct inputs with the same output under $\bar{h}$, contradicting injectivity. To see surjectivity, extend any desired output block to a vector in $\mathbb{R}^K$. Since $\bar{h}$ is surjective, it attains this vector, so d_a attains the desired block. Thus d_a is bijective.

It remains to show that d_a^{-1} is C^1. After grouping its rows and columns by $\mathcal{P}_B$, $J_{\bar{h}}(z)$ is block diagonal: Step 1 shows that derivatives across different blocks vanish, while Step 2 identifies the diagonal blocks as $J_{d_a}(z_{\mathcal{G}_a})$. Since $\bar{h}$ has a C^1 inverse, $J_{\bar{h}}(z)$ is nonsingular everywhere. A block diagonal matrix is nonsingular only if each diagonal block is nonsingular. Hence $J_{d_a}(u)$ is nonsingular for every $u \in \mathbb{R}^{|\mathcal{G}_a|}$.

The inverse function theorem therefore gives a C^1 local inverse around each output point of d_a. Since d_a is bijective, these local inverses agree with its global inverse, so $d_a^{-1} \in C^1$. The permutation π and block maps $d_1, \dots, d_M$ therefore satisfy Definition 2.5, proving block-wise identifiability with respect to $\mathcal{P}_B$. □

F Proof of Corollary 3.6

Corollary 3.6 (Directional ambiguity)**.** *Let f be the ground-truth decoder in Definition 2.1, and suppose*

$$\textit{(Strict Containment)} \qquad S_q \subsetneq S_r, \qquad \textit{for some } q \neq r. \tag{17}$$

For any $\epsilon \neq 0$, define $c : \mathbb{R}^K \to \mathbb{R}^K$ by

$$c_j(z) = \begin{cases} z_r + \epsilon z_q, & j = r, \\ z_j, & j \neq r. \end{cases} \tag{18}$$

The map c is bijective, with $c, c^{-1} \in C^1$, and satisfies Form 2 of Theorem 3.2 with $\pi = \mathrm{Id}_{[K]}$. Hence $\hat{f} := f \circ c^{-1}$ is observationally equivalent to f, with entanglement map c. Its induced active matrix satisfies the sparsity condition, although c mixes factors from two distinct blocks of $\mathcal{P}_B$. Therefore, the sparsity condition cannot guarantee block-wise identifiability with respect to $\mathcal{P}_B$.

Proof. Fix $q \neq r$ with $S_q \subsetneq S_r$ and $\epsilon \neq 0$, and let c be the map in Eq. (18).

Step 1: the construction is an invertible cross-block mixing. To find the inverse, write $u = c(z)$. Since $q \neq r$, the construction leaves factor q unchanged, so $u_q = z_q$. We can therefore recover the modified factor from $u_r = z_r + \epsilon z_q$ as $z_r = u_r - \epsilon u_q$. All other factors satisfy $z_j = u_j$ for $j \neq r$. These formulas give a unique input for every $u \in \mathbb{R}^K$, so c is bijective, with inverse

$$(c^{-1})_j(u) = \begin{cases} u_r - \epsilon u_q, & j = r, \\ u_j, & j \neq r. \end{cases} \tag{34}$$

Both c and c^{-1} are linear, and hence belong to C^1. Let $H^c := \mathrm{supp}(J_c)$. Since $\epsilon \neq 0$, the output c_r depends on both z_r and z_q, while each other output c_j depends only on z_j. Thus, by Definition 2.3,

$$H^c_{j:} = \begin{cases} \{r, q\}, & j = r, \\ \{j\}, & j \neq r. \end{cases} \tag{35}$$

Because $S_q \subsetneq S_r$, factors q and r lie in distinct blocks of the equal-signature partition $\mathcal{P}_B$; the r-th output of c therefore mixes two distinct blocks.

Step 2: strict containment makes this mixing compatible with sparsity. The added dependence on z_q introduces no new relation type into the signature of the r-th output, since $S_r \cup S_q = S_r$. To verify the sparsity condition, take $\pi = \mathrm{Id}_{[K]}$ in Form 2 of Theorem 3.2. For every $j \neq r$, Eq. (35) gives $H^c_{j:} = \{j\}$, and hence $S_j \subseteq S_j$. For $j = r$, the support is $\{r, q\}$, and

$$S_r \subseteq S_r, \qquad S_q \subsetneq S_r.$$

Thus every $k \in H^c_{j:}$ satisfies $S_k \subseteq S_j$, so c satisfies Form 2 with $\pi = \mathrm{Id}_{[K]}$. By Theorem 3.2, it therefore satisfies the sparsity condition from Eq. (12).

Step 3: constructing an observationally equivalent counterexample. To decode the transformed representation $u = c(z)$, we first recover $z = c^{-1}(u)$ and then apply the original decoder f. Accordingly, define $\hat{f} := f \circ c^{-1}$, so that $\hat{f}(c(z)) = f(z)$ for every $z \in \mathbb{R}^K$. Since f is injective and C^1, and $c^{-1} : \mathbb{R}^K \to \mathbb{R}^K$ is a C^1 bijection, $\hat{f}$ is injective and C^1. Moreover,

$$\hat{f}(\mathbb{R}^K) = f(c^{-1}(\mathbb{R}^K)) = f(\mathbb{R}^K).$$

Thus f and $\hat{f}$ are observationally equivalent. On their common image, $\hat{f}^{-1} = c \circ f^{-1}$, so

$$\hat{f}^{-1} \circ f = c \circ f^{-1} \circ f = c.$$

The entanglement map is therefore $h = c$, and Step 1 verifies $h, h^{-1} \in C^1$ as required by Definition 2.2. The component $c_r(z) = z_r + \epsilon z_q$ depends on true factors from two distinct blocks of $\mathcal{P}_B$. No permutation of the learned factors can place this component in a block whose inputs contain both q and r. Hence this representation is not block-wise identifiable with respect to $\mathcal{P}_B$.

Finally, consider the reverse mixing: replace the q-th output by $z_q + \epsilon z_r$ while leaving all other outputs unchanged. Its signature then expands from S_q to $S_q \cup S_r = S_r \supsetneq S_q$, so the total induced support strictly increases. In contrast, the construction above leaves the r-th output signature unchanged because $S_r \cup S_q = S_r$.

□

G Experimental Details

G.1 Evaluation Metrics

The metrics in Section 5 serve two different purposes. DCI-D and MCC measure factor-wise recovery (Eastwood & Williams, 2018; Locatello et al., 2020; Khemakhem et al., 2020b), whereas the learned-to-true factor dependency matrix D tests the more general prediction of Theorem 3.2: a learned factor matched to true factor r may depend on true factor q if and only if $S_q \subseteq S_r$. We define D and its three summaries first, and only then the two matching steps they rest on, since those steps exist solely to make the summaries invariant to the arbitrary ordering of the learned factors. Every quantity below is computed separately for each trained model from deterministic encoder means, and true factor values and B are used only for evaluation.

Estimating factor dependence by intervention. Following the general use of interventions to evaluate learned factor dependence (Suter et al., 2019), we resample one true factor at a time. Let $z^{(q)}$ be obtained by independently resampling true factor q while holding all other true factors fixed, and let $x^{(q)} = f(z^{(q)})$. For each learned factor s we measure

$$A_{sq} = \mathbb{E}\left[(\mu_s(x) - \mu_s(x^{(q)}))^2\right], \qquad R_s = \sum_{j=1}^{K} A_{sj}, \qquad \overline{A}_{sq} = \frac{A_{sq}}{R_s}. \tag{36}$$

We call R_s the *total intervention response* of learned factor s: it collects the squared change of μ_s over all K single-factor interventions, so $\overline{A}_{sq}$ is the share of that change attributable to true factor q. The normalization is across true factors, separately for each learned factor.

Summarizing the predicted dependency regions. The theorem constrains which true factors a given learned factor may depend on, so the matrix has to be written in true-factor order before it can be compared with B. Writing $\pi^\star$ for the matching of learned to true factors constructed in the two paragraphs that follow, we define

$$D_{rq} = \overline{A}_{(\pi^\star)^{-1}(r),q} = \frac{A_{(\pi^\star)^{-1}(r),q}}{R_{(\pi^\star)^{-1}(r)}}, \qquad \sum_{q=1}^{K} D_{rq} = 1, \tag{37}$$

so that D_{rq} is the share of the total intervention response of the learned factor labelled r that true factor q accounts for. Its K^2 entries split into the diagonal, the off-diagonal dependencies that signature inclusion permits, and the dependencies it forbids,

$$\mathcal{D} = \{(r, r) : r \in [K]\}, \quad \mathcal{P} = \{(r, q) : r \neq q,\ S_q \subseteq S_r\}, \quad \mathcal{F} = \{(r, q) : S_q \nsubseteq S_r\}, \tag{38}$$

and we report the share of the total dependency mass that falls in each region:

$$\mathrm{RDD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{D}} D_{rq}, \qquad \mathrm{RPD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{P}} D_{rq}, \qquad \mathrm{RFD} = \frac{1}{K} \sum_{(r,q)\in\mathcal{F}} D_{rq}. \tag{39}$$

All three denominators are K, rather than the number of entries $|\mathcal{D}|$, $|\mathcal{P}|$ or $|\mathcal{F}|$ in the corresponding region, because the three regions partition one fixed quantity: every row of D sums to one, so $\mathrm{RDD} + \mathrm{RPD} + \mathrm{RFD} = 1$ once an empty region is assigned mass zero. Dividing each region by its own size would instead give three per-entry averages that no longer add up, and those sizes range from 0 to 30 entries across the regimes we study (Table 5). We display a region as not applicable when it is empty, which is RPD in the antichain regime and RFD under identical signatures.

RFD is the quantity the theorem constrains: it is exactly the dependency mass that Theorem 3.2 predicts to be absent, so a low value is evidence for the prediction and a high value is evidence against it. RPD is not a target. The theory states which dependencies are allowed to remain, not which of the admissible solutions an optimizer must choose, so the size of RPD neither establishes nor rules out the ambiguity the theory permits.

Two properties of the estimate are worth stating. First, D measures the change caused by resampling a true factor over its whole range, whereas the theorem concerns exact functional dependence; RFD is therefore evidence about the predicted exclusion pattern at the scale of the sampled changes. Second, no entry of D is thresholded: each one enters Eq. (39) at its measured value and none is rounded to zero, so a learner cannot obtain a low RFD by leaving small residual dependencies just below a cutoff. The one degenerate case is a learned factor with $R_s = 0$, for which Eq. (36) cannot be normalized; such a run is recorded as a failed diagnostic and excluded from the results rather than being credited with zero RFD. It did not occur in any run of Section 5.1 (Table 7).

Matching learned and true factors. Learned factors carry no intrinsic order, so $\pi^\star$ has to be recovered from the model itself. We recover it from the pattern of change under each relation type, which identifies a learned factor with a signature without using any true factor value. For relation type τ and learned factor s, define

$$E_{\tau s} = \mathbb{E}_{(x,x')|\tau}\left[(\mu_s(x) - \mu_s(x'))^2\right], \qquad \widetilde{E}_{\tau s} = \frac{E_{\tau s}}{\max_{\tau'} E_{\tau' s}}, \tag{40}$$

where the implementation lower-bounds the denominator by 10^{-12} for numerical stability. The column $\widetilde{E}_{:s}$ approximates the indicator of the relation types that move learned factor s, and the signature $B_{:q}$ is the indicator of the relation types that move true factor q, so we match the two:

$$\widetilde{\pi} \in \arg\min_{\pi\in\mathfrak{S}_K} \sum_{s=1}^{K} \left\| \widetilde{E}_{:s} - B_{:\pi(s)} \right\|_2^2. \tag{41}$$

We solve this linear sum assignment problem with SciPy's `linear_sum_assignment` implementation (Virtanen et al., 2020), which uses the modified Jonker–Volgenant algorithm described by Crouse (2016).

Resolving equal-signature ties. When several true factors share a signature, Eq. (41) fixes which signature class each learned factor belongs to but not which member of that class it is labelled with, because $B_{:q} = B_{:r}$ makes the two objectives identical. We break the tie with the intervention responses, choosing among the labellings that leave the signature classes of $\widetilde{\pi}$ unchanged:

$$\pi^\star \in \operatorname*{arg\,max}_{\substack{\pi\in\mathfrak{S}_K \\ S_{\pi(s)} = S_{\widetilde{\pi}(s)}\ \forall s\in[K]}} \sum_{s=1}^{K} \overline{A}_{s,\pi(s)}, \tag{42}$$

again with `linear_sum_assignment`. Among the admissible labellings this is the one that places the most mass on the diagonal, that is, the one that maximizes RDD and therefore minimizes RPD. The relabelling step operates only within a single equal-signature class. As a result, it can redistribute mass between $\mathcal{D}$ and $\mathcal{P}$, but can neither move mass into nor out of $\mathcal{F}$, since true factors with identical signatures induce the same permitted and forbidden regions. Consequently, it cannot improve RFD, which is precisely the metric that tests the theorem. What it eliminates is a relabelling artefact. Without it, a model that otherwise recovers the factors perfectly could be penalized simply for reversing the labels of two factors within the same signature class, causing a pure relabelling to be recorded as cross-factor mixing. In contrast, genuine dependence of a learned factor on multiple true factors remains unchanged.

G.2 Further experimental details for Section 5.1

Theorem 3.2 rules out dependencies that are not allowed by signature inclusion. Section 5.1 tests this prediction in a fully controlled setting, where the true-factor distribution, the observation maps, the amount of data, the learner and its optimization procedure are fixed, so that the only difference between the experimental conditions is the partial order on relation signatures. This section specifies the data, the learner and the validity checks, and reports the complete results behind Table 2.

Signature regimes. We draw $K = 6$ independent true factors from $z \sim \mathcal{N}(0, I_6)$ and use $T = 4$ relation types[1]. The four regimes in Table 5 differ only in the active factor matrix B, which specifies which true factors vary under each relation type and therefore determines what the signatures can distinguish. The first three regimes keep $\|B\|_0 = 12$, ensuring that any differences between them are not driven by the number of active entries. Identical signatures is the limiting case in which the relations cannot tell any two true factors apart, and it necessarily has all $TK = 24$ entries active. The table also lists the sizes of the permitted and forbidden regions of Eq. (38), which range from 0 to 30 of the 36 entries of the relative dependency (RD) matrix D.

Table 5: The four signature regimes. $\mathcal{P}$ and $\mathcal{F}$ are the permitted and forbidden regions of Eq. (38); the remaining 6 entries of D are its diagonal.

Regime	Signatures $(S_1, \dots, S_6)$	$\|B\|_0$	$\|\mathcal{P}\|$	$\|\mathcal{F}\|$	Predicted ambiguity
Antichain	$\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}$	12	0	30	factor-wise
Incomparable	$\{1,2\}, \{1,2\}, \{2,3\}, \{2,3\}, \{3,4\}, \{3,4\}$	12	6	24	within three blocks
Containment	$\{1\}, \{1,2\}, \{1,2,3\}, \{2,3,4\}, \{3,4\}, \{4\}$	12	6	24	directional
Identical	$\{1,2,3,4\}$ for every q	24	30	0	unrestricted

Table 6: Settings shared by every run of Section 5.1.

Factors and relations	
true factors	$K = 6$, $z \sim \mathcal{N}(0, I_6)$
relation types	$T = 4$
observation map	10 GLOW coupling blocks, subnet width 6, SiLU, clamp 2
observed variables	12, from z padded with six nuisance inputs $\epsilon \sim \mathcal{N}(0, 10^{-4} I_6)$
observation maps	10, frozen and shared by every regime
Learner	
encoder, decoder	MLP, 3 hidden layers of width 256, SiLU
learned factors	$d = 6$, diagonal Gaussian, $\log \sigma^2 \in [-10, 10]$
decoder noise	fixed $\sigma = 0.1$
KL weight	$\beta = 1$ in every regime
mask	$\Pi = \sigma(\Gamma)$, Γ initialised at 5, Binary Concrete, temperature 1
Optimisation	
updates	100,000, budget annealed $TK = 24 \to s_*$ over the first 50,000
pairs per update	256 per type, 1,024 in total, generated online
learning rates	model 10^{-3}, mask logits 5×10^{-4} (Adam); multiplier 5×10^{-4} (SGD)
Adam, gradients	$(\beta_1, \beta_2) = (0.9, 0.999)$, no weight decay, gradient-norm clip 5
multiplier reset	to zero whenever the mask is already below budget
Selection and evaluation	
checkpoints	every 2,000 updates, 4 noise draws on fixed validation pairs
selected	lowest validation VAE loss among the final-budget checkpoints with $\sum_{\tau,s} \Pi_{\tau s} \le s_* + 0.1$
validation, test	2,500 and 10,000 pairs per type
single-factor pairs	10,000 per true factor
observations	15,000; DCI fitted on 10,000, scored on 5,000
runs	3 per observation map per regime, 120 in total

Observation maps. The observation map is chosen to be a smooth nonlinear mixing function and is held fixed across all regimes. For each $m \in \{1, \dots, 10\}$ we therefore sample and freeze an independent map $F_m : \mathbb{R}^{12} \to \mathbb{R}^{12}$, built

[1] The antichain regime requires four relation types. By Proposition 3.7, the maximum number of pairwise incomparable signatures is $\binom{T}{\lfloor T/2 \rfloor}$, which equals 6 when $T = 4$. Accordingly, we use six independent true factors.

from affine coupling blocks with a fixed random permutation after each block, using the `GLOWCouplingBlock` of the FREIA library for invertible architectures (Ardizzone et al., 2018-2022); Table 6 summarizes the setting. This architecture mirrors the synthetic mixing used by Sorrenson et al. (2020); Zheng et al. (2022); Zheng & Zhang (2023), except that we replace ReLU with SiLU so that F_m is C^1, as the theory requires. As in those studies, an observation has twice as many variables as there are true factors: we pad the six true factors with six independent nuisance variables $\epsilon \sim \mathcal{N}(0, 10^{-4} I_6)$ and generate

$$x = \mathcal{S}_m(F_m([z; \epsilon])) \,, \tag{43}$$

where $\mathcal{S}_m$ standardizes each observed variable using $50{,}000$ independently generated observations. The full map is invertible in twelve dimensions and the true factors occupy six of its inputs, so the observations are not a function of z alone. We reuse the same ten maps and the same standardization statistics in every regime, so the regimes are compared on identical observation maps.

Relation pairs and supervision. A relation type specifies which true factors may change between two observations. For a pair of type τ we draw $z, \widetilde{z} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_6)$ and set

$$z'_q = \begin{cases} \widetilde{z}_q, & B_{\tau q} = 1, \\ z_q, & B_{\tau q} = 0, \end{cases} \tag{44}$$

so that an active true factor is redrawn independently and an inactive one is copied. Both observations of a pair use the same nuisance value ϵ in Eq. (43), so every difference within a pair is caused by the true factors that τ selects and none of it by the padding. The learner receives the pair (x, x'), its type τ and the total budget $\|B\|_0$; it never receives an individual entry of B or a true factor value.

Learner, optimization and model selection. We use the masked Ada-GVAE of Section 4, with the settings in Table 6. Its relation-specific mask decides which learned factors may differ within a pair: when the sampled mask entry $\widetilde{M}_{\tau s}$ is inactive, the posterior mean and variance of learned factor s are averaged between the two observations, and when it is active the two posteriors stay separate, as in Ada-GVAE (Locatello et al., 2020). Reconstruction therefore provides the learning signal: the mask model can suppress an entry only when the corresponding learned factor is unnecessary for representing differences within that relation type. The sparsity constraint is $\sum_{\tau,s} \Pi_{\tau s} \leq s(t)$, where the schedule $s(t)$ begins with all $TK = 24$ entries available and decreases linearly to the final budget $s^\star = \|B\|_0$ over the first half of training, after which $s(t) = s^\star$; this lets the model learn to reconstruct the observations before it has to decide which entries can be removed. The COOPER library (Gallego-Posada et al., 2025) enforces the sparsity constraint through a nonnegative Lagrange multiplier. Once the target budget is met to within 0.1, we retain all saved checkpoints and select the one with the lowest validation ELBO, i.e., RECON + KL. The selection process has no access to the true factor values, the entries of B, the test pairs, or any metric reported below (e.g. DCI-D, Spearman MCC).

Validity checks. Two potential confounders could affect the comparison, and we test for both before including a model in the analysis (Table 7). First, a randomly initialized observation map may make some true-factor nearly invisible from the observations while disproportionately amplifying others. Such effects would hinder factor recovery independently of the relation signatures, introducing variance unrelated to the regime structure. We therefore require each map to be numerically invertible, to induce non-negligible variation in every observed variable, and to be well conditioned. At 128 sampled inputs, the smallest singular value of $\partial x / \partial z$ must exceed 0.05, ensuring that no combination of true factors is locally hidden, and the ratio of the largest to the smallest singular values must remain below 100 for at least 95% of samples. All ten maps passed these checks on the first draw. Second, certain training pathologies can make the RD matrix D ill-defined. These include collapsed learned factors, a representation of rank below K, a mask that remains unresolved at the selected checkpoint, or a learned factor with zero total intervention response, the sole situation in which Eq. (36) cannot be normalized. None of the 120 runs failed any of these checks.

Complete results. For each observation map and regime, we train the model from three independent random initializations and average the resulting metrics. Reported variation is therefore computed across the ten observation maps rather than across the thirty individual runs. Table 8 extends Table 2 with DCI-D and Spearman MCC. Two findings merit particular attention. First, reducing RFD does not require discarding information. In the three regimes containing forbidden entries, the learned factors achieve a held-out DCI R^2 of at least 0.998 for predicting the true factors and retain at least 5.97 of 6 effective dimensions (Table 9). Thus, the representations remain informative while placing negligible mass in $\mathcal{F}$. Second, factor-wise recovery deteriorates precisely as the signatures become less informative. DCI-D and MCC are both close to one in the antichain regime, decrease in the incomparable and containment regimes, and attain their lowest values under identical signatures. In the latter case, the learned representation still predicts the

Table 7: Validity checks applied before a map or a run is used, with the requirement and the range observed. No map and no run were rejected.

Quantity	Required	Observed
Observation map, before any training (10 maps)		
forward–inverse error	$\leq 10^{-4}$	$(2.15 \text{ to } 4.17) \times 10^{-6}$
smallest observed-variable s.d.	$> 10^{-3}$	0.069 to 0.332
smallest singular value of $\partial x/\partial z$	> 0.05	0.119 to 0.413
95th-percentile singular-value ratio	< 100	6.9 to 28.2
draws needed	–	1 for all 10
Trained run, before evaluation (120 runs)		
validation loss	finite	120/120
smallest posterior-mean variance	$> 10^{-4}$	≥ 0.93
effective rank of the representation	> 5.5	≥ 5.95 of 6
mask probabilities in $[0.4, 0.6]$	≤ 2	≤ 1
total intervention response $\sum_j A_{sj}$	> 0 for every s	120/120
expected mask entries	$\leq s_* + 0.1$	$\leq s_* + 0.04$

true factors accurately, with a held-out DCI $R^2 = 0.984$, but encodes each true factor across multiple learned factors rather than recovering the factors individually.

Table 8: **Complete results for the controlled test.** Mean $\pm$ sample standard deviation across 10 observation maps after averaging 3 runs per map. RFD is displayed in units of 10^{-3}. RPD and RFD are not applicable when the corresponding region is empty. DCI-D and Spearman MCC measure factor-wise recovery, which need not remain high when the signatures permit mixing between factors.

Regime	**RDD** $\uparrow$	**RPD**	**RFD** $\downarrow$	**DCI-D** $\uparrow$	**Spearman MCC** $\uparrow$
Antichain	0.9986 ± 0.0008	–	1.426 ± 0.801	0.9975 ± 0.0017	0.9991 ± 0.0005
Incomparable	0.9535 ± 0.0483	0.0455 ± 0.0485	1.034 ± 0.529	0.9229 ± 0.0510	0.9735 ± 0.0291
Strict containment	0.9918 ± 0.0026	0.0073 ± 0.0028	0.935 ± 0.472	0.9801 ± 0.0066	0.9959 ± 0.0016
Identical signatures	0.6350 ± 0.1427	0.3650 ± 0.1427	–	0.4829 ± 0.1558	0.7772 ± 0.1005
Without sparsity (models trained under identical signatures)					
Antichain	0.6314 ± 0.1488	–	368.566 ± 148.755	0.4826 ± 0.1554	0.7776 ± 0.0996
Incomparable	0.6294 ± 0.1475	0.0928 ± 0.0587	277.790 ± 94.838	0.4826 ± 0.1554	0.7776 ± 0.0996
Strict containment	0.6216 ± 0.1534	0.0759 ± 0.0418	302.452 ± 126.842	0.4826 ± 0.1554	0.7776 ± 0.0996

It is the sparsity constraint that suppresses forbidden dependence. A low RFD alone does not establish that the constraint is responsible, since a learner with an intrinsic preference for factor-wise solutions would also achieve a low RFD. Therefore, the models trained under identical signatures are needed as a control for the effect of sparsity. Here the budget matches the total number of mask entries, $s^\star = TK = 24$, so the sparsity constraint never becomes active, and after training the mask is still fully open (Table 9). With the mask fully open, the two observations in a pair are encoded independently, and the loss decomposes into terms involving only individual observations. Its expectation therefore depends solely on the marginal distribution of a single observation. That distribution is identical across regimes: in Eq. (44), both z and z' are drawn from $\mathcal{N}(0, I_6)$ and are mapped through the same observation map in Eq. (43). The active factor matrix B affects only which observations are paired and does not alter the marginal distribution itself. Because training pairs are generated online, there is no fixed training set whose composition could differ across regimes. The identical-signature models therefore provide a clean control: they use the same learner, the same observation map, and the same data-generating process, but with the sparsity constraint effectively inactive. To isolate the effect of sparsity, we evaluate these models against the supports associated with the antichain, incomparable, and containment regimes without retraining, using exactly the same test pairs, single-factor pairs, evaluation observations, alignment procedure, and aggregation scheme as those used by our method on the corresponding observation map. Two checks confirm that these models provide a valid control rather than a weaker learner (Table 9): their validation loss is at least as good as ours on every support, and they preserve the information about the true factors, at a held-out DCI R^2 of 0.983 and an effective rank of 5.99 out of 6. Any difference observed relative to our method therefore cannot be attributed to poorer optimization or information loss in the control models.

Figure 6 summarizes the experimental outcomes. Removing the constraint increases RFD by orders of magnitude on every one of the ten observation maps in each of the three regimes: by a median factor of 287, 335 and 359, and by at least 41 even on the map where it helps least (Table 9). These results therefore confirm, for each observation map individually and not merely on average as presented in Table 2, that it is the sparsity constraint that suppresses forbidden dependence.

Table 9: Our runs and the no-sparsity control, on the quantities omitted in Table 2. Each row is one support: our model trained on it, against the identical-signature model scored against it. Those control models are the same ones Table 2 reports under identical signatures; there the budget equals the number of mask entries, $s^\star = TK = 24$, so the constraint never binds, and after training the expected number of active entries lies between 23.991 and 23.996 with every mask probability at least 0.9996. The control reconstructs slightly better, because an open mask leaves the two observations of a pair encoded independently; it is not a weaker model. The last two columns give the factor by which RFD rises when the constraint is removed, computed on each observation map separately. Every cell averages the ten maps, each averaging its three runs.

	Reconstruction ↓		KL		DCI R^2 ↑		Effective rank ↑		RFD ratio	
Support	ours	no sps.	ours	no sps.	ours	no sps.	ours	no sps.	median	worst
Antichain	1.418	1.226	15.65	15.35	0.998	0.983	6.00	5.99	287×	45×
Incomparable	1.307	1.232	15.60	15.33	0.998	0.983	5.99	5.99	335×	41×
Containment	1.300	1.223	15.62	15.33	0.998	0.983	5.97	5.99	359×	46×

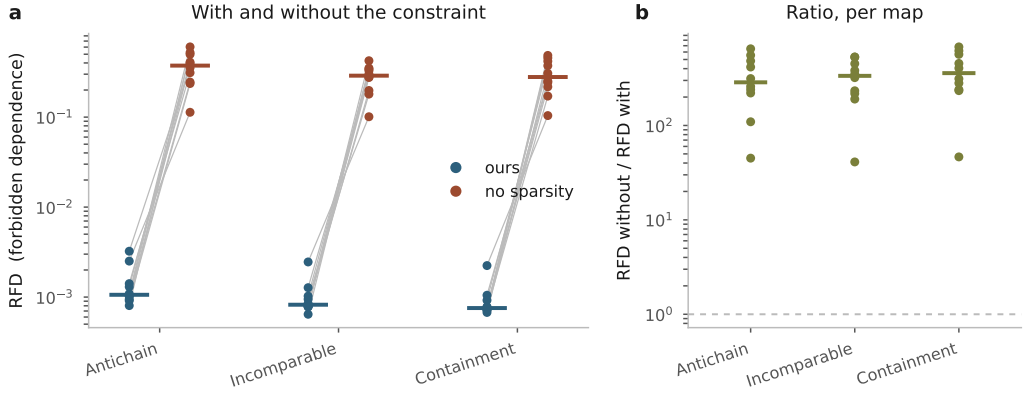

Figure 6: It is the sparsity constraint, not the learner, that suppresses forbidden dependence. Each point is one observation map, averaging its three runs, as in Table 2; bars are medians and both vertical axes are logarithmic. **a**, RFD for our method and for the same learner with the constraint inactive, joined by a line within each map. **b**, the ratio of the two on that map; the dashed line marks a ratio of one, where removing the constraint would change nothing. Every map lies far above it.

Scope of the controlled test. This experiment does not impose the assumptions of the theorem on the learner. Instead, it tests whether the predicted exclusion pattern survives when they are relaxed along the two directions in which real applications are expected to depart from the theoretical setting. The theorem is concerned exactly about reconstructing representations connected by C^1 diffeomorphisms. Here, however, this assumption holds only for the fixed observation map F_m. The learner is a non-invertible VAE whose reconstructions are necessarily approximate, as is typical of reconstruction-based learning methods. The theorem assumes that observations are generated solely by the K true factors. In contrast, here the observations also depend on six nuisance variables, which the learner must disregard while representing the true factors using only six learned dimensions. In practice, one rarely knows in advance that no additional sources of variation are present. Neither relaxation provides the learner with any information about which dependencies are permitted, so neither can generate the predicted exclusion pattern. If anything, they could obscure or eliminate it, yet we describe below that they do not. Across the three regimes with forbidden entries, the learned factors still predict the true factors with a held-out DCI R^2 of at least 0.998 and use at least 5.97 of the 6 dimensions (Table 9), while RFD stays near 10^{-3} (Table 2). Theorem 3.2 remains a population-level result about exact representations. What this experiment demonstrates is that its prediction remains valid under the two approximations that separate the theorem from this experimental setting.

G.3 Further experimental details for Section 5.2

This subsection expands upon Section 5.2. Appendix G.3.1 describes the three benchmark datasets, the active factor matrix B used for each benchmark, and the procedure for sampling typed pairs, also highlighting deviations from Locatello et al. (2020). Appendix G.3.2 describes the hyperparameter selection protocol together with the hyperparameter grid search during model selection. Appendix G.3.3 reports complete versions of both main-text tables, including the metrics omitted there: DCI completeness (Eastwood & Williams, 2018), MCC (Khemakhem et al., 2020b), and the number of active coordinates (Burda et al., 2016). Appendix G.3.4 investigates the structure of the learned representation, analyzing how many factors are used, which factors are encoded across multiple coordinates, and why such allocations arise. Appendix G.3.5 presents the ablation studies together with results on instable training dynamics, while Appendix G.3.6 reports the computational resources used.

G.3.1 Data

Datasets. We use Shapes3D (Kim & Mnih, 2018), dSprites (Higgins et al., 2017) and MPI3D-Real (Gondal et al., 2019). Each dataset forms a complete Cartesian grid over its K nonconstant factors (Table 10), so every factor combination corresponds to exactly one image and any factor can be varied while holding all others fixed. For dSprites, the color channel is constant and is therefore not counted as a factor. Its images are stored as binary $0/1$ pixels and are used without further preprocessing.

Table 10: Datasets and antichain relation matrices.

Dataset	K	factor values	images	T	$\|B\|_0$	changed per type
Shapes3D	6	$10 \times 10 \times 10 \times 8 \times 4 \times 15$	480,000	4	12	3, 3, 3, 3
dSprites	5	$3 \times 6 \times 40 \times 32 \times 32$	737,280	4	10	3, 3, 2, 2
MPI3D-Real	7	$6 \times 6 \times 2 \times 3 \times 3 \times 40 \times 40$	1,036,800	5	14	3, 3, 3, 3, 2

Active Factor Matrix. The theory experiment ($d = K$) reuses the four regimes of Section 5.1 without modification. For the comparison experiment ($d = 10$), each dataset is assigned a single antichain active factor matrix B, constructed according to three principles. First, every factor receives a distinct signature of weight two, ensuring that no signature contains another and that Corollary 3.4 applies. Second, the number of relation types T is chosen to be the smallest value satisfying $\binom{T}{2} \geq K$, giving $T = 4, 4, 5$ for the three datasets. Third, the rows of B are made as balanced as possible, so that the relation types have similar numbers of active factors. Hence $\|B\|_0 = 2K$, and each type varies the two or three factors listed in Table 10.

Pairs. A type-τ pair is generated by first sampling from a source image uniformly from the factor grid. Every factor q satisfying $B_{\tau q} = 1$ is then resampled uniformly from its set of admissible values *excluding* its current one, while all remaining factors are left unchanged. The learner is provided only with (x, x', τ); neither the factor values nor active factor matrix B are observed. Relation types are balanced within each update, and any remainder is assigned cyclically across updates. With 64 pairs and $T = 5$, the per-type counts are $13, 13, 13, 13, 12$, with the underrepresented type rotating between updates. The sampler is seeded solely by the seed index, so all methods are evaluated on identical pairs for a given seed. This guarantees the matched comparisons required by the seed-paired tests in the main text. Validation uses a fixed set of 2,000 pairs per type, shared by all seeds and methods; the test metrics use a separate sample of images.

Since the current value is excluded, the factors that actually change are precisely those in the support of row τ. A plain resampling would leave a discrete factor unchanged with probability $1/n$, namely $1/3$ for the three dSprites shapes and $1/2$ for the two MPI3D-Real object sizes (Table 10). Theorem 3.2, however, is stated in terms of the support of the change, which is not related to whether all such factors would change. In the continuous setting of Section 5.1 an independent redraw already changes a factor almost surely, so both settings implement the same condition.

Relation to the Ada-GVAE protocol. As shown in Table 11, our pairs are denser and deterministic than those in the setting of Locatello et al. (2020). Locatello et al. (2020) report in their Figure 10 that Ada-GVAE performs best when exactly one factor varies ($k = 1$) and weakens as more factors change, approaching the unsupervised β-VAE as more factors vary (larger k). Our scores for Ada-GVAE are accordingly lower than Locatello et al. (2020)'s medians at random k: DCI-D 0.840 against 0.946 on Shapes3D and 0.367 against 0.401 on MPI3D-Real. The comparison in Table 4 is unaffected by this difference, as both methods receive identical training pairs and share the same architecture, optimization budget, and selection protocol; only our method has access to τ.

Table 11: Pair generation compared with Locatello et al. (2020).

	Locatello et al. (2020)	This work
Criterion	how many factors changed (k)	which may change (type τ)
Changed set	a random subset of size k	the support of row τ of B
Realised change	at most k factors	exactly that support
Density	$k = 1$ or random	2–3 of K

Table 12: Settings shared by both methods.

	Ours	Ada-GVAE
Architecture	VAE of Locatello et al. (2020), Bernoulli decoder	
Updates	600,000, 64 pairs per update	
Learning rate	0.0005 (model and mask)	0.0001
β	selected from $\{1, 6, 16\}$	
Budget s_*	selected from $\|B\|_0 + \rho\,(d - K)$, $\rho \in \{0, 1, 2\}$	–
Constraint	Cooper, dual SGD 0.0005	–
Schedule	$Td \to s_*$ over 150,000 updates, then equality	–
Checkpoint	lowest validation objective after update 150,000 (ours: also $\mathbb{E}\|M\|_0 \le s_* + 0.1$)	

G.3.2 Training and hyperparameter selection

All methods use the same architecture, the same number of updates and the same pair stream (Table 12). Validation uses the training objective without the Lagrangian term on the held-out pairs of data described in Appendix G.3.1; test metrics use 15,000 images, with DCI fitted on 10,000 and scored on 5,000 of them. Only our method has access to τ, and only our method operates under a sparsity budget.

Hyperparameter selection. Every configuration of a grid is trained on seeds 0–4. The configuration with the lowest mean validation ELBO, defined as reconstruction plus unweighted KL at the selected checkpoint, is then retrained on seeds 5–19, and results are reported across 20 seeds. Test metrics are never used in the model selection. The hyperparameter grid is $\beta \in \{1, 6, 16\}$ for both methods. For our method it additionally includes the budget $s_* = \|B\|_0 + \rho\,(d - K)$ with $\rho \in \{0, 1, 2\}$, so that $\rho = 2$ gives $s_* = 2d$, corresponding to the number of mask entries obtained if every factor carries a signature of size 2. At $d = K$ the budget grid collapses to $\|B\|_0$ and only β needs to be selected, shared by all three regimes.

Table 13: Selection grid at $d = 10$ (seeds 0–4, mean over seeds; the selected configuration **in bold**). ELBO is reconstruction plus unweighted KL on held-out pairs and is the only quantity used in hyperparameter selection. DCI-D is reported for reference only and plays no role in the selection procedure; cases where the two disagree are discussed in Appendix G.3.5. *Bold marks the row the rule selected, not the best value in a column.*

		Shapes3D		dSprites		MPI3D-Real	
β	s_*	ELBO ↓	DCI-D	ELBO ↓	DCI-D	ELBO ↓	DCI-D
Ours							
1	$\|B\|_0$	3525.3	0.897	53.9	0.653	4588.8	0.473
1	$+(d - K)$	3494.2	0.956	**33.8**	**0.593**	4584.8	0.485
1	$2d$	**3491.0**	**0.929**	34.4	0.471	**4584.1**	**0.409**
6	$\|B\|_0$	3533.9	0.906	72.1	0.469	4604.8	0.272
6	$+(d - K)$	3501.3	0.961	53.6	0.486	4601.7	0.252
6	$2d$	3495.0	0.874	49.6	0.448	4600.9	0.237
16	$\|B\|_0$	3622.7	0.792	120.5	0.503	4639.6	0.110
16	$+(d - K)$	3532.8	0.838	113.4	0.395	4638.7	0.114
16	$2d$	3517.2	0.844	106.8	0.530	4639.1	0.112
Ada-GVAE							
1	–	3835.9	0.782	**116.8**	**0.304**	**4607.8**	**0.352**
6	–	**3831.8**	**0.833**	143.5	0.457	4629.3	0.237
16	–	3864.2	0.800	175.1	0.340	4651.4	0.127

Table 14: Selection grid at $d = K$: mean validation ELBO per regime, the selected β shown **in bold**. The budget grid collapses to $\|B\|_0$ here, so only β is selected, shared by all three regimes.

Method	β	Antichain	Incomparable	Containment	mean
Ours	1	**3534.2**	**3553.5**	**3587.8**	**3558.5**
	6	3569.3	3579.5	3610.3	3586.4
	16	3592.8	3569.7	3602.4	3588.3
Ada-GVAE	1	**3936.4**	**4051.5**	**4025.7**	**4004.5**
	6	3920.5	4118.7	4060.8	4033.3
	16	3954.7	4165.6	4100.9	4073.8

The hyperparameter search selects $\beta = 1$ in every case except for Ada-GVAE on Shapes3D, where it selects $\beta = 6$ (Tables 13 and 14). The selected budget s_* differs across datasets: $\|B\|_0 + (d - K)$ on dSprites and $2d$ on Shapes3D and MPI3D-Real. For each dataset, the gap between the two best configurations is smaller than the seed-to-seed spread.. DCI-D is reported for reference only and is not used for model selection; cases where it disagrees with the validation ELBO are discussed in Appendix G.3.5.

G.3.3 Full results

Table 15: Extension of Table 3 with RPD, DCI-D and MCC (20 seeds; oracle: 3). **Bold** indicates that our method is significantly better than both Ada-GVAE and Ours (w/o sps.) according to a seed-paired Wilcoxon signed-rank test at $p < 0.05$; the oracle is shown for reference and is not part of the comparison.

Regime	Model	RDD ↑	RPD	RFD ↓	DCI-D ↑	MCC ↑
Antichain	Ours (oracle)	0.85 ± 0.19	–	0.15 ± 0.19	0.95 ± 0.09	0.55 ± 0.16
	Ours	$\mathbf{0.65 \pm 0.09}$	–	$\mathbf{0.35 \pm 0.09}$	$\mathbf{0.68 \pm 0.11}$	0.41 ± 0.08
	Ours (w/o sps.)	0.52 ± 0.06	–	0.48 ± 0.06	0.51 ± 0.13	0.38 ± 0.07
	Ada-GVAE	0.57 ± 0.07	–	0.43 ± 0.07	0.55 ± 0.04	0.43 ± 0.09
Incomparable	Ours (oracle)	0.89 ± 0.07	0.11 ± 0.07	0.00 ± 0.00	0.86 ± 0.03	0.54 ± 0.17
	Ours	$\mathbf{0.58 \pm 0.07}$	0.11 ± 0.06	$\mathbf{0.31 \pm 0.09}$	$\mathbf{0.66 \pm 0.06}$	0.37 ± 0.07
	Ours (w/o sps.)	0.42 ± 0.11	0.20 ± 0.12	0.38 ± 0.07	0.51 ± 0.13	0.38 ± 0.07
	Ada-GVAE	0.31 ± 0.11	0.19 ± 0.06	0.50 ± 0.11	0.40 ± 0.06	0.36 ± 0.07
Containment	Ours (oracle)	0.85 ± 0.05	0.12 ± 0.01	0.02 ± 0.04	0.92 ± 0.04	0.63 ± 0.06
	Ours	$\mathbf{0.61 \pm 0.12}$	0.03 ± 0.04	$\mathbf{0.36 \pm 0.14}$	$\mathbf{0.67 \pm 0.09}$	0.39 ± 0.08
	Ours (w/o sps.)	0.45 ± 0.11	0.03 ± 0.03	0.52 ± 0.13	0.51 ± 0.13	0.38 ± 0.07
	Ada-GVAE	0.53 ± 0.03	0.02 ± 0.04	0.45 ± 0.02	0.56 ± 0.03	0.35 ± 0.04

Table 16: Extension of Table 4 with DCI completeness, MCC and the number of active coordinates (20 seeds). **Bold** marks metrics on which our method significantly outperforms than Ada-GVAE according to a seed-paired Wilcoxon signed-rank test at $p < 0.05$.

Dataset	Method	DCI-D ↑	DCI-C ↑	MCC ↑	active coordinates
Shapes3D	Ada-GVAE	0.840 ± 0.064	0.707 ± 0.064	0.726 ± 0.063	10.00 ± 0.00
	Ours	$\mathbf{0.910 \pm 0.070}$	$\mathbf{0.824 \pm 0.051}$	0.694 ± 0.132	8.25 ± 0.55
dSprites	Ada-GVAE	0.317 ± 0.071	0.282 ± 0.071	0.439 ± 0.052	10.00 ± 0.00
	Ours	$\mathbf{0.628 \pm 0.078}$	$\mathbf{0.601 \pm 0.072}$	$\mathbf{0.586 \pm 0.044}$	7.10 ± 1.41
MPI3D-Real	Ada-GVAE	0.367 ± 0.019	0.393 ± 0.033	0.299 ± 0.015	10.00 ± 0.00
	Ours	$\mathbf{0.466 \pm 0.056}$	$\mathbf{0.486 \pm 0.041}$	$\mathbf{0.351 \pm 0.070}$	7.30 ± 0.57

Tables 15 and 16 extend the results of Section 5.2 with the metrics omitted from the main text. RDD, RPD, and RFD are defined in Eq. (19) and Appendix G.1, DCI follows Eastwood & Williams (2018), and MCC follows Khemakhem et al. (2020a). Two observations stand out. First, DCI completeness (DCI-C) improves alongside DCI disentanglement (DCI-D) (0.824 against 0.707, 0.601 against 0.282, 0.486 against 0.393). The gains therefore cannot be attributed to spreading a factor across more coordinates, precisely the behaviour that completeness penalizes. Second, MCC is the only metric for which the advantage is not consistent: it is significant on dSprites and MPI3D-Real, but not on

Shapes3D (0.694 against 0.726); at $d = K$ it is significant in none of the three regimes. The same runs activate only 7.1 to 8.3 of the 10 dimensions, while Ada-GVAE uses all of them. Appendix G.3.4 describes a single explanation underlying both observations.

G.3.4 What the learned features encode

All five quantities below are evaluated on the checkpoint selected by the training protocol of Section G.3.2; factor values are only used for evaluation.

- *Intervention response*: for each true factor q, we draw 10,000 pairs that differ only in q and record the mean squared response of every learned factor. After normalization, each entry gives the fraction of the response carried by a learned factor; a learned factor *carries* q when this fraction is at least 0.2.
- *Relation profile*: we compute $P[\tau, s] = \mathbb{E}[(z_s - z'_s)^2 \mid \tau]/(2 \operatorname{Var} z_s)$ over 10,000 pairs per type. A value $P[\tau, s] \approx 1$ indicates that learned factor s is fully resampled by type τ. Two learned factors that encode the same true factor have similar profiles, and hence cosine similarity close to one.
- *Cyclic closure*: for a true factor carried by two learned factors, let c_v denote the mean of those learned factors over all images with factor value v. We define the cyclic closure as $\frac{\|c_{\text{last}} - c_{\text{first}}\|}{\text{median}_v \|c_{v+1} - c_v\|}$, so a closed loop has a score close to 1 and an arc gives a substantially larger value. This is useful to distinguish the representations of cyclic factors, where the largest value is semantically similar to the smallest one.
- *Pixel impact*: for each true factor, we draw 2,000 source images, resample that factor alone as in a pair, and average the resulting squared pixel difference over all pixels and pairs, with images represented in $[0, 1]$.
- *Recovery criterion*: A true factor is considered recovered if every learned factor that carries it carries no other true factor and has a binary-mask column equal to the factor's signature in B. This criterion is intentionally strict: a true factor split across two learned factors can still satisfy it, provided both learned factors meet these conditions.

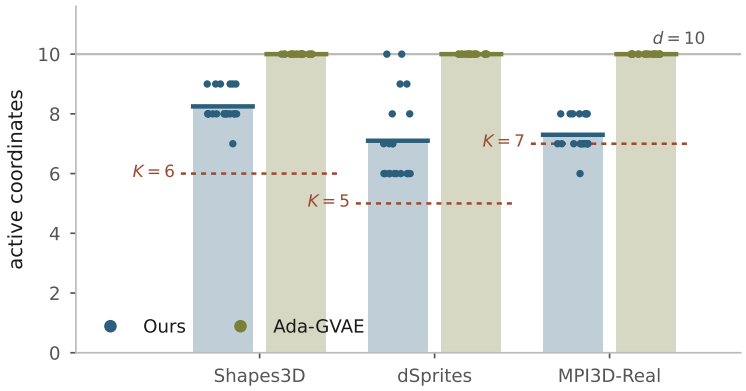

Figure 7: Active learned factors, for which posterior means have variance above 0.05 on the evaluation set (Burda et al., 2016). Dots denote individual seeds and bars their mean. The count is largely insensitive to the threshold: it moves by at most 0.4 between 0.01 and 0.2, and a criterion based on a per-factor KL agrees within 0.1.

Ada-GVAE keeps all ten learned factors active on every dataset, while our method settles at 8.25, 7.10 and 7.30 (Figure 7). The constraint therefore does prune, as Dai & Wipf (2019) predict for a latent space larger than the data requires, but it does not prune to $K = 6$, 5 and 7. On Shapes3D, all active learned factors carry a true factor. The surplus is therefore used, not noise.

Table 17 details what this surplus is precisely used for: a few factors take two learned factors each, and the rest take one. On Shapes3D, splitting occurs in every run for the two dominant hue factors and in 17 of 20 runs for object hue. Shape, scale, and orientation are far less affected, together accounting for only 10 of the other 60 cases. Splitting is not, in itself, a failure: floor hue splits in all 20 runs yet is still recovered in 12 of them. Because the recovery criterion, described in the beginning of this section, is strict, recovery rates are low on the two harder datasets even though they attain the best DCI-D values we observe.

Table 17: Per-factor splitting and recovery (20 seeds). *Values* is the factor's cardinality and *pixel impact* the mean squared change of the image when that factor alone is resampled; both are defined in Appendix G.3.4. *Split* counts the runs in which the true factor is carried by two learned factors, and *recovered* those in which it is recovered.

Dataset	Factor	values	pixel impact	split	recovered
Shapes3D	floor hue	10	0.1088	20/20	12/20
	wall hue	10	0.1133	20/20	11/20
	object hue	10	0.0422	17/20	9/20
	scale	8	0.0186	6/20	10/20
	shape	4	0.0206	3/20	12/20
	orientation	15	0.0117	1/20	10/20
dSprites	shape	3	0.0218	12/20	0/20
	scale	6	0.0251	0/20	10/20
	orientation	40	0.0165	16/20	0/20
	pos x	32	0.0578	18/20	8/20
	pos y	32	0.0574	18/20	4/20
MPI3D-Real	object color	6	0.0030	2/20	17/20
	object shape	6	0.0014	11/20	0/20
	object size	2	0.0017	13/20	0/20
	camera height	3	0.0156	15/20	5/20
	background color	3	0.0010	3/20	9/20
	horizontal axis	40	0.0044	20/20	0/20
	vertical axis	40	0.0038	18/20	0/20

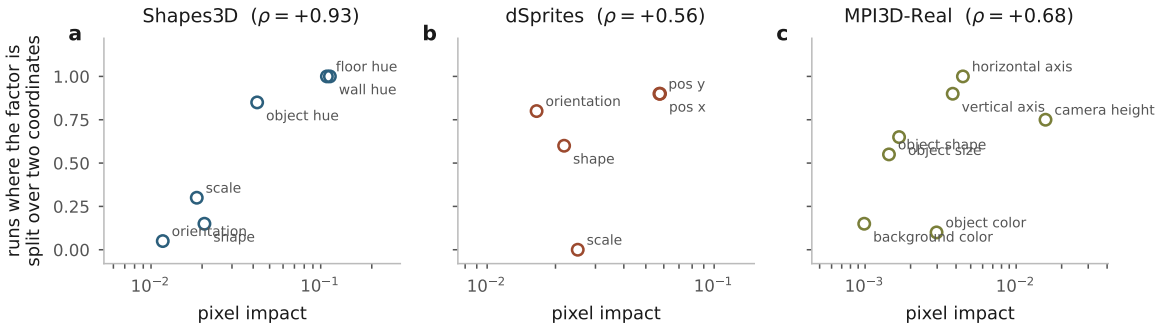

Figure 8: Pixel impact against how often a true factor is carried by two learned factors. One point corresponds to one true factor, the horizontal axis is logarithmic and the Spearman correlation values are shown in the titles.

Which factors split across two learned factors is largely predictable from their pixel impact (Figure 8): the Spearman correlation is $+0.93$, $+0.56$ and $+0.68$. The number of values a factor can take is not predictive of splitting ($+0.03$ on Shapes3D). For example, orientation has 15 values yet splits in only 1 of 20 runs, whereas the three dSprites shape factors split in 12. A similar pattern appears among the Shapes3D hue factors. Because the object covers only a small part of the image, object hue has about one third of the pixel impact of wall and floor hue, and is the only hue factor that does not split in every run.

A second learned factor is taken in because the model represents cyclic factors as loops (Figure 9). For split hue factors, which are cyclic, the two learned factors have nearly identical relation profiles (median cosine similarity 1.00), and their means over the ten hue values trace a closed loop with median closure 1.06 across the 57 split hue instances. By contrast, the ten split cases of shape, scale, and orientation, none of which are cyclic, trace open arcs with median closure 6.4. This behavior is not imposed by the model: ten discrete values could equally well be embedded on a line. Rather, reconstruction appears to favor a loop representation, whereas encoding a cyclic factor with a single learned factor would require introducing a discontinuity.

Two consequences follow. First, a budget of $\|B\|_0$ implicitly assumes one learned factor per true factor. When a true factor splits, the additional mask entries it occupies must come at the expense of another factor. The factors that lose entries are typically those with the smallest pixel impact, and relaxing the budget restores them (Appendix G.3.5). Second, MCC matches each true factor to a single learned factor. A split factor is therefore penalized even when it is

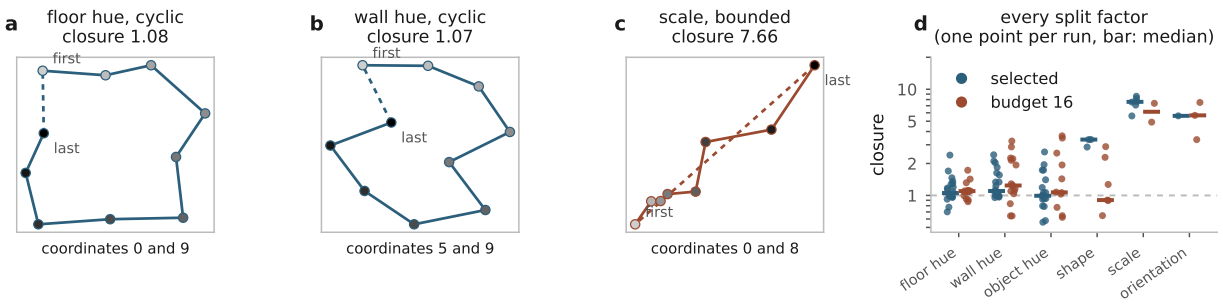

Figure 9: The curve traced by the two carriers of a split factor, over the values of that factor. **a**–**c**, the median run for that factor; grey to black marks the first to the last value and the dashed segment joins the last back to the first. **d**, closure of every split factor of every run, for the selected budget and for $s_* = 16$; the dashed line is closure 1.

otherwise correctly recovered. This explains why MCC is the only metric on which we do not improve on Shapes3D, the dataset where splitting is most common.

G.3.5 Ablations

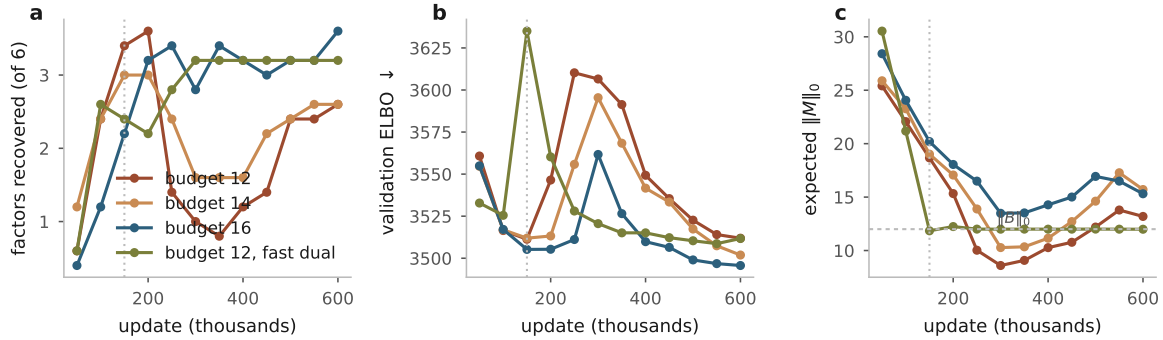

Figure 10: Intermediate checkpoints on Shapes3D (seeds 0–4, mean). **a**, factors recovered; **b**, validation ELBO; **c**, expected mask entries. The dotted vertical line is the end of the budget schedule.

The diagnostics used for B are identical to those in Appendix G.3.4, and are applied to checkpoints saved every 50,000 training updates. At $s_* = \|B\|_0$, the constraint becomes overly restrictive. Because the multiplier builds up slowly, the expected number of active entries falls to roughly 8, below $\|B\|_0$. Between 250k and 350k updates, the number of recovered factors drops from 3.4 to 0.8, while the validation ELBO increases by approximately 100 nats [2] (Figure 10). A budget of 14 exhibits the same behavior. By contrast, with a budget of 16, the expected number of entries never falls below 13.5, and the collapse disappears. Increasing the dual learning rate by a factor of 100 keeps the expected number of entries near 12 and eliminates the collapse without increasing the number of active entries (H1 in Table 18). It recovers 18 of the 31 nats separating $s_* = \|B\|_0$ from $s_* = 16$, but no more. The evidence therefore points to two distinct effects: optimization overshoot and an inherently inadequate budget of $\|B\|_0$ entries.

Table 18 presents the remaining ablations. Two effects are apparent. First, observing the same number of pairs in only a quarter of the updates is clearly worse. Second, the constraint-handling procedure of Section 5.1, which employs dual restarts under an inequality constraint, performs worse than maintaining the budget as an equality throughout. Estimating the required number of entries from the relation energy, rather than selecting s_* by grid search, converges to $s_* \approx 16.8$ and matches the performance of the fixed budget $s_* = 16$ (3493.0 versus 3494.2). However, the safeguards are essential. Without them, the estimate ranges from 19 to 36 entries early in training, the mask remains dense for 80k–125k updates, and one seed collapses to a DCI-D score of 0.30.

[2] $1 \text{ nat} = 1/\ln 2 \approx 1.44 \text{ bits}$

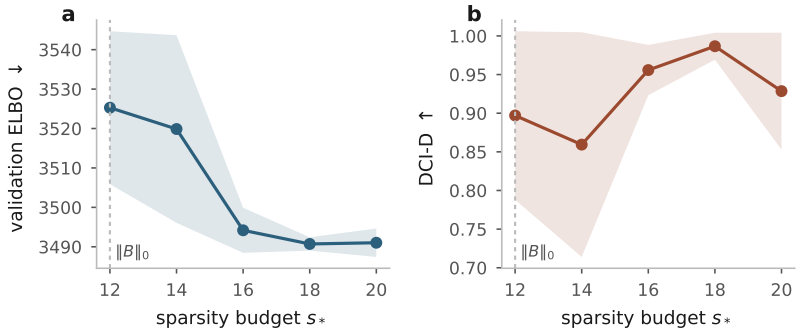

Figure 11: Budget sensitivity on Shapes3D (seeds 0–4; shaded: one standard deviation). **a**, validation ELBO; **b**, DCI-D. The dotted vertical line marks $s_* = \|B\|_0$, the budget the theory prescribes.

Table 18: One change at a time from the base configuration of the tuning stage ($\beta = 1$, $s_* = \|B\|_0$; Shapes3D, seeds 0–4). The budget itself is varied in Figure 11.

Change from the base configuration	ELBO ↓	DCI-D	MCC	final s_*
$\beta = 16$	3622.7	0.79 ± 0.16	0.60 ± 0.09	12.0
$\beta = 6$	3533.9	0.91 ± 0.08	0.66 ± 0.06	12.0
$\beta = 1$	3525.3	0.90 ± 0.11	0.70 ± 0.12	12.0
256 pairs per update, 150k updates	3673.5	0.66 ± 0.07	0.55 ± 0.02	12.0
dual learning rate $\times 10$	3575.7	0.86 ± 0.09	0.65 ± 0.08	12.0
dual learning rate $\times 100$	3574.3	0.83 ± 0.08	0.60 ± 0.06	12.0
dual restarts, inequality throughout	3621.5	0.77 ± 0.15	0.59 ± 0.08	12.0
dual learning rate $\times 100$ (H1)	3507.0	0.96 ± 0.06	0.66 ± 0.10	12.0
budget from relation energy	3498.1	0.81 ± 0.28	0.68 ± 0.15	15.2
budget from relation energy, guarded	3493.0	0.95 ± 0.06	0.69 ± 0.12	16.8

Validation ELBO is not consistent with DCI-D. Relaxing the budget improves performance on Shapes3D (Figure 11), but the effect is not universal. On dSprites, the smallest budget, $s_* = \|B\|_0$, achieves the highest DCI-D in the grid (0.653 versus 0.593 for the selected configuration), despite having a worse and substantially less stable validation ELBO. Validation ELBO and the structural metrics also disagree on the configuration choice in three cases. On Shapes3D, the selection rule prefers $s_* = 2d$ to $s_* = \|B\|_0 + (d - K)$ on the basis of a validation-ELBO advantage of only 3.2 nats. Yet, over the full set of 20 seeds, the selected budget achieves the lower DCI-D of the two 0.910 versus 0.937), although the difference is not significant ($p = 0.10$). On MPI3D-Real, the selected budget $s_* = 2d$ produces a lower DCI-D score than the runner-up configuration (0.409 versus 0.485), despite a validation-ELBO advantage of only 0.7 nats. Likewise, at $d = K$, the selected value $\beta = 1$ performs worse than $\beta = 16$ on RFD (0.31 versus 0.23 in the incomparable regime), with the difference being significant ($p = 0.027$). We report the configuration selected by the rule in every case; determining how to make this choice without access to ground-truth labels remains an open problem.

Table 19: Computational cost for Section 5.2. Shapes3D and dSprites runs used L40S and A100 GPUs; each MPI3D-Real run used one A100, or shared an L40S with two others.

Runs	count	hours each	GPU hours
Tables 3 and 15 ($d = K$)	149	2.7	404
Table 4, Shapes3D	40	3.3	133
Table 4, dSprites	40	2.4	97
Table 4, MPI3D-Real	40	1.5	61
Selection grids and ablations	449	2.9	1316
Total	718		2010

G.3.6 COMPUTE

Roughly two thirds of the total compute was spent on the selection grids rather than the final reported runs (Table 19), reflecting the cost of choosing β and s_* without access to labels.