

# When Do Learned Blocks Preserve Causal Meaning? Block Identifiability and Constructive Causal Abstractions

Karim Zaghw<sup>1,2,3</sup>   Wieland Brendel<sup>2,3</sup>   Patrik Reizinger<sup>2,3</sup>

<sup>1</sup>University of Tübingen   <sup>2</sup>ELLIS Institute Tübingen

<sup>3</sup>Max Planck Institute for Intelligent Systems, Tübingen, Germany

## Abstract

Standard identifiability theory assumes one-dimensional latent factors, but learned representations can encode latent information across multidimensional blocks whose internal coordinates are not uniquely determined. When such blocks are used to represent and intervene on higher-level concepts, it is unclear whether their causal meaning survives the transformations left unresolved by block-identifiability. We connect block-identifiability to constructive causal abstraction. We show that block-diagonal invertible transformations followed by compatible coordinate permutations form the maximal class that preserves every constructive causal abstraction built on those blocks. This guarantee assumes that the correspondence between blocks and concepts is updated with the representation. A small synthetic experiment with additive autoencoders shows that high-level interventions can be transferred between independently learned representations while approximately preserving their intended effects.

allow their causal relations to be recovered (Shimizu et al., 2006; Reizinger et al., 2023).

In practice, a concept may be encoded across a block of learned coordinates rather than a single coordinate. Several settings motivate this block-level view. Sparse autoencoders (SAEs) can split one concept across multiple learned features (Bricken et al., 2023; Mueller et al., 2025), while block-sparse featurizers explicitly learn groups of activation directions and recover visual concepts that are typically two- to four-dimensional (Fel et al., 2026). Object-centric models represent objects using vector-valued slots (Locatello et al., 2020). Formal block-identifiability results arise in independent subspace analysis (ISA), which recovers independent subspaces up to permutation and invertible within-subspace transformations (Hyvärinen and Hoyer, 2000; Theis, 2006), and in additive decoders, which identify latent blocks up to permutation and block-wise invertible transformations under their assumptions (Lachapelle et al., 2023). These results recover each block as a whole while leaving its internal coordinates undetermined. Different learned models may therefore represent the same block using different internal coordinates. While practitioners intervene on such blocks by steering feature groups, manipulating object slots, or clamping latent subspaces, block-identifiability does not tell us whether these interventions keep the same high-level causal meaning across different representations.

## 1 INTRODUCTION

Identifiability theory asks whether a learned representation recovers the latent factors that generated the observations (Hyvärinen et al., 2019; Khemakhem et al., 2020). Many standard results assume that the learned representation has the same dimensionality as the latent space, with one learned coordinate corresponding to each latent factor. Under the assumptions of these results, the factors can often be recovered up to a permutation and a separate invertible coordinate-wise transformation. Because these transformations do not mix the factors, additional assumptions can still

We use constructive causal abstractions (Beckers and Halpern, 2019; Geiger et al., 2025) to state this question precisely. We treat the coordinates of the learned representation as variables in a causal model, which we call the low-level model. The human-interpretable concepts form another causal model, which we call the high-level model. A constructive causal abstraction maps states and interventions in the low-level model to corresponding states and interventions in the high-level model, with each high-level concept computed from one block of learned coordinates. We ask which changes to the learned representation preserve this

mapping.

Our contributions are:

1. **We identify the largest class of transformations guaranteed to preserve causal abstractions built from learned blocks.** This class consists of invertible transformations within blocks followed by coordinate permutations that carry the block-to-concept correspondence with them. It includes the transformations allowed by block-identifiability guarantees for ISA and additive decoders. We prove that every transformation in this class preserves every abstraction of this form, while every transformation outside it fails to preserve at least one (Thm. 1).
2. **We illustrate high-level steering and its transfer between learned representations.** In a small synthetic experiment with independently trained additive autoencoders, we translate high-level interventions into low-level block edits. We transfer these edits between models using fitted blockwise maps and show that their intended high-level effects are approximately preserved (§ 4).

## 2 BACKGROUND

**Notation.** Following Geiger et al. (2025), capital letters denote variables and lowercase letters their values, and boldface denotes a set of variables or a joint setting of one, so that  $x \in \text{Val}_X$  and  $\mathbf{x} \in \text{Val}_{\mathbf{V}_L}$ . Let  $\mathbf{S} = \{S_1, \dots, S_m\}$  be the ground-truth causal factors: each  $S_i$  is produced by a mechanism  $F_{S_i}$  from its parents  $\text{Pa}(S_i)$  in the ground-truth directed acyclic graph (DAG) and an exogenous noise variable  $N_i$ , written  $S_i := F_{S_i}(\text{Pa}(S_i), N_i)$ . The map  $f$  takes a factor setting  $\mathbf{s}$  to the activation vector  $\mathbf{o} = f(\mathbf{s})$  supplied to the featurizer  $g$ ; this includes generating the network input and passing it through the network. The featurizer returns the learned feature vector, or representation,  $\mathbf{x} = g(\mathbf{o})$ . We model the coordinates of  $\mathbf{x}$  as variables of the representation-level SCM  $L$ . They form the finite set  $\mathbf{V}_L$ , so a learned *feature* is a variable  $X \in \mathbf{V}_L$  with value set  $\text{Val}_X$ , and  $\mathbf{x} \in \text{Val}_{\mathbf{V}_L}$  is a total setting of  $L$ . The high-level SCM  $H$  has variables  $\mathbf{V}_H = \{Z^{(1)}, \dots, Z^{(B)}\}$ , where each  $Z^{(b)}$  is a human-interpretable *concept* chosen for intervention and may be computed from a block of learned features rather than from one coordinate. The ground-truth factors  $\mathbf{S}$  generate the activation vector  $\mathbf{o}$ , whereas  $\mathbf{V}_H$  contains the concepts abstracted from the learned representation; we do not assume that they coincide. For a subset  $\mathbf{U} \subseteq \mathbf{V}_L$ , write  $\text{Val}_{\mathbf{U}} := \prod_{U \in \mathbf{U}} \text{Val}_U$ , and the projection  $\text{Proj}_{\mathbf{U}} : \text{Val}_{\mathbf{V}_L} \rightarrow \text{Val}_{\mathbf{U}}$  restricts a setting to  $\mathbf{U}$ . Tab. 2 in § A summarizes this notation. Fig. 1

illustrates this conceptual pipeline.

To compare interventions across levels, we first need a common description of causal models, their solutions, and the interventions applied to them.

**Definition 1** (SCMs and interventions). A *deterministic SCM* is a pair  $M = (\Sigma, \{F_X\}_{X \in \mathbf{V}})$ , where  $\Sigma = (\mathbf{V}, \text{Val})$  is a signature and  $F_X : \text{Val}_{\mathbf{V}} \rightarrow \text{Val}_X$  is the mechanism for each  $X \in \mathbf{V}$ . A setting  $\mathbf{v} \in \text{Val}_{\mathbf{V}}$  is a solution of  $M$  if  $F_X(\mathbf{v}) = \text{Proj}_{\{X\}}(\mathbf{v})$  for every  $X \in \mathbf{V}$ ;  $\text{Solve}(M)$  denotes the set of all such solutions. A *hard intervention* on  $\mathbf{U} \subseteq \mathbf{V}$  replaces  $\{F_X\}_{X \in \mathbf{U}}$  by constants; we identify it with the idempotent map on  $\text{Val}_{\mathbf{V}}$  that overwrites the coordinates in  $\mathbf{U}$ , and the empty intervention with the identity map. A set  $\Psi$  of interventionals forms an *intervention algebra* when its composition algebra is isomorphic to that of hard interventions on some signature (Geiger et al., 2025). Hard interventions are partially ordered by extension:  $I \subseteq I'$  when  $I'$  sets every variable  $I$  sets, to the same value.

To say that  $L$  and  $H$  make consistent causal predictions, mapping an intervention from  $L$  to  $H$  should give the same high-level outcomes as mapping the resulting low-level solutions to  $H$ . Exact transformations formalize this requirement (Rubenstein et al., 2017; Geiger et al., 2025).

**Definition 2** (Exact transformation). An *exact transformation*  $(\tau, \omega) : L \rightarrow H$  consists of a surjective state map  $\tau : \text{Val}_{\mathbf{V}_L} \rightarrow \text{Val}_{\mathbf{V}_H}$  and a surjective order-preserving intervention map  $\omega$  such that for all  $I \in \text{dom}(\omega)$

$$\tau(\text{Solve}(L^I)) = \text{Solve}(H^{\omega(I)}) \quad (1)$$

To ask whether a causal abstraction survives a change of representation, we must express the low-level states, mechanisms, and interventions in the new coordinates. Bijective translations formalize this consistent reparameterization (Geiger et al., 2025).

**Definition 3** (Bijective translation). Let  $M$  be a deterministic SCM, and write  $F_{\mathbf{V}}(\mathbf{v}) := (F_X(\mathbf{v}))_{X \in \mathbf{V}}$  for the map collecting all of its mechanisms. For a bijection  $T : \text{Val}_{\mathbf{V}} \rightarrow \text{Val}_{\mathbf{V}}$ , the *bijective translation*  $M_T$  has mechanism map

$$F_{\mathbf{V}}^T := T \circ F_{\mathbf{V}} \circ T^{-1}. \quad (2)$$

An intervention  $I$  is transported to

$$I^T := T \circ I \circ T^{-1}. \quad (3)$$

For any intervention algebra  $\Psi$  on  $M$ , the transported interventions  $\Psi^T := \{I^T : I \in \Psi\}$  form an intervention algebra on  $M_T$ , ordered by  $I^T \subseteq J^T$  exactly when

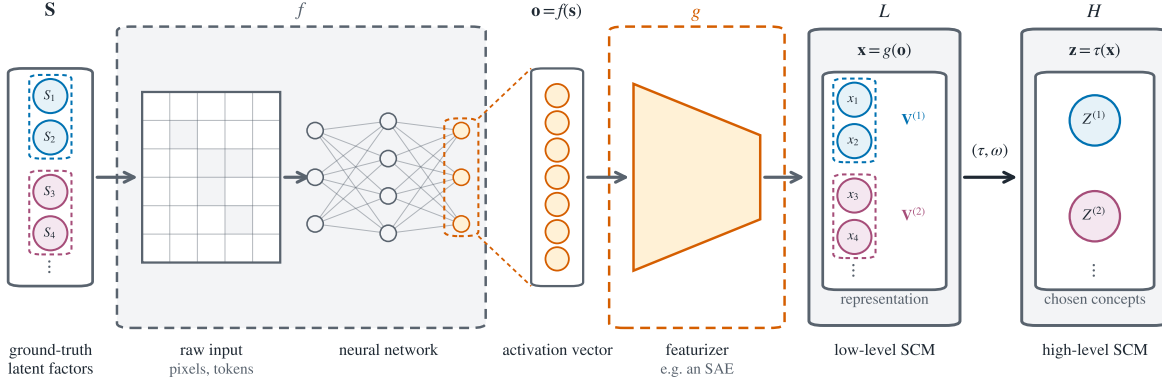


Figure 1: **From ground-truth factors to a constructive causal abstraction of a learned representation.** The ground-truth causal factors  $\mathbf{S}$  take values  $\mathbf{s}$ , which determine the network input and the resulting activation vector  $\mathbf{o} = f(\mathbf{s})$ . A featurizer  $g$ , such as a sparse autoencoder (SAE), maps these activations to the learned representation  $\mathbf{x} = g(\mathbf{o})$ . We treat the coordinates of  $\mathbf{x}$  as variables of the low-level structural causal model (SCM)  $L$  and partition them into blocks  $\mathbf{V}^{(b)}$ . The constructive causal abstraction  $(\tau, \omega)$  maps low-level states and complete-block interventions to states and interventions in the high-level SCM  $H$ , where each human-interpretable concept  $Z^{(b)}$  is computed from its corresponding block. Blue and pink indicate this block-to-concept correspondence. The ground-truth factors  $\mathbf{S}$  generate the activations but need not coincide with the concepts in  $H$ .

$I \subseteq J$ . Every intervention  $I$  on  $M$  is therefore paired with the transported intervention  $I^T$  on  $M_T$ . Under this pairing,

$$T(\text{Solve}(M^I)) = \text{Solve}(M_T^{I^T}),$$

so the state map  $T$  and intervention map  $I \mapsto I^T$  form an exact transformation from  $M$  to  $M_T$ . Bijective translations compose with other exact transformations (Geiger et al., 2025). This composition allows an abstraction from  $L$  to  $H$  to be carried over to a reparameterized model  $L_T$ .

Exactness tells us that the low- and high-level models agree under corresponding interventions, but it does not say where each high-level variable is represented in the low-level model. Our question requires this additional structure because we want each concept to be computed from a particular block of learned coordinates and interventions on that block to correspond to interventions on the concept. Constructive causal abstractions formalize this block-to-concept correspondence through an alignment (Beckers and Halpern, 2019).

**Definition 4** (Alignment and constructive causal abstraction). An *alignment* between the low-level variables  $\mathbf{V}_L$  and the high-level variables  $\mathbf{V}_H$  is a pair  $\langle \Pi, \pi \rangle$ , where  $\Pi = \{\Pi_Z : Z \in \mathbf{V}_H\} \cup \{\Pi_\perp\}$  is a partition of  $\mathbf{V}_L$ , and  $\pi = \{\pi_Z\}_{Z \in \mathbf{V}_H}$  is a family of surjective *cell maps*  $\pi_Z : \text{Val}_{\Pi_Z} \rightarrow \text{Val}_Z$ . The cell  $\Pi_Z$  contains the low-level variables associated with  $Z$ , while the optional cell  $\Pi_\perp$  contains variables that are not mapped

to any high-level variable.

The alignment  $\langle \Pi, \pi \rangle$  induces the state map

$$\tau_\pi(\mathbf{x}) := (\pi_Z(\text{Proj}_{\Pi_Z}(\mathbf{x})))_{Z \in \mathbf{V}_H}$$

and maps interventions on complete cells according to

$$\omega_\pi(\text{do}(\Pi_Z = \mathbf{i}_Z, Z \in A)) := \text{do}(Z = \pi_Z(\mathbf{i}_Z), Z \in A),$$

where  $A \subseteq \mathbf{V}_H$  and  $\mathbf{i}_Z \in \text{Val}_{\Pi_Z}$ . The high-level model  $H$  is a *constructive causal abstraction* of  $L$  under this alignment if  $(\tau_\pi, \omega_\pi) : L \rightarrow H$  is an exact transformation.

For a cell map  $\pi_Z$ , the *fiber* over  $z \in \text{Val}_Z$  is  $\pi_Z^{-1}(z)$ , the set of low-level cell values mapped to  $z$ . If  $\pi_Z$  is many-to-one, values in the same fiber correspond to the same high-level value and therefore cannot be distinguished in  $H$ .

### 3 PRESERVING CONSTRUCTIVE CAUSAL ABSTRACTIONS

We show that constructive causal abstractions are invariant under block-diagonal reparameterizations and that no larger class of reparameterizations preserves all of them, connecting block-identifiability to causal abstraction theory.

#### 3.1 Setup

Our aim is to determine whether the ambiguity left by block-identifiability changes the causal meaning as-

signed to learned blocks. We therefore assume that each learned block serves as the alignment cell for one high-level variable and consider interventions on complete blocks.

**Assumption 1** (Block-aligned abstraction).

- (i)  $L = (\Sigma_L, \{F_X\}_{X \in \mathbf{V}_L})$  is a deterministic SCM with block partition

$$\mathbf{V}_L = \bigsqcup_{b=1}^B \mathbf{V}^{(b)}. \quad (4)$$

- (ii)  $(\tau, \omega) : L \rightarrow H$  is constructive, with each block  $\mathbf{V}^{(b)}$  serving as the alignment cell for  $Z^{(b)}$ . We write  $\pi_b : \text{Val}_{\mathbf{V}^{(b)}} \rightarrow \text{Val}_{Z^{(b)}}$  for the corresponding cell map, for  $b = 1, \dots, B$ .
- (iii)  $\Psi$  is the intervention algebra of all hard interventions on zero or more complete blocks,  $\text{dom}(\omega) = \Psi$ , and  $\omega$  is the intervention map induced by the alignment as in Defn. 4.

Block-identifiability recovers each block as a whole but leaves its individual coordinates undetermined. We therefore focus on interventions on complete blocks because the corresponding high-level concept can depend on every coordinate in the block. To test whether changes to the internal coordinates affect the block’s causal meaning, we collect all invertible within-block coordinate changes in the family  $\mathcal{D}$ .

**Definition 5** (Block reparameterizations). The group of block-diagonal reparameterizations is

$$\begin{aligned} \mathcal{D} := \{ & \text{Diag}(D^{(1)}, \dots, D^{(B)}) \mid \\ & D^{(b)} : \text{Val}_{\mathbf{V}^{(b)}} \rightarrow \text{Val}_{\mathbf{V}^{(b)}}, \\ & D^{(b)} \text{ is bijective for } b = 1, \dots, B \}. \end{aligned} \quad (5)$$

Block-identifiability may also permute whole blocks; we incorporate those permutations below. To characterize the largest class of transformations guaranteed to preserve every abstraction of this form, Thm. 1 compares these structured transformations with arbitrary bijections of the low-level state space.

To test whether a change of representation preserves the abstraction, we must compare the same high-level intervention before and after the change. Under a bijection  $T$ , a state  $\mathbf{y}$  in  $L_T$  corresponds to  $T^{-1}(\mathbf{y})$  in  $L$ , and a low-level intervention  $I$  becomes  $I^T$ . We therefore use  $\tau \circ T^{-1}$  for states and define  $\omega^T$  so that  $I^T$  has the same high-level meaning as  $I$ .

**Definition 6** (Transported high-level intervention map). For every  $I \in \Psi$ , define

$$\omega^T(I^T) := \omega(I). \quad (6)$$

Because  $I \mapsto I^T$  is an order-preserving bijection,  $\omega^T$  is well-defined, surjective, and order-preserving. By exactness of bijective translations and composition of exact transformations (Geiger et al., 2025),  $(\tau \circ T^{-1}, \omega^T) : L_T \rightarrow H$  is exact. Since every  $D \in \mathcal{D}$  is a bijection, this includes all within-block reparameterizations.

### 3.2 The Maximal Class Preserving Constructiveness

The family  $\mathcal{D}$  changes coordinates within blocks, but block-identifiability may also permute whole blocks. More generally, a coordinate permutation can move the coordinates of a block to new positions. To keep each concept associated with the same group of coordinates after this change, we track where each block moves under the permutation.

**Definition 7** (Compatible permutations and transported cells). A permutation  $\sigma$  of the low-level variables  $\mathbf{V}_L$  is *compatible* if  $\text{Val}_{\sigma(X)} = \text{Val}_X$  for every  $X \in \mathbf{V}_L$ . It acts on total settings according to  $(\sigma(\mathbf{x}))_{\sigma(X)} := x_X$ . For each block  $\mathbf{V}^{(b)}$ , its image  $\sigma(\mathbf{V}^{(b)})$  is the corresponding *transported cell*, and  $\sigma_b$  denotes the induced bijection from  $\text{Val}_{\mathbf{V}^{(b)}}$  to  $\text{Val}_{\sigma(\mathbf{V}^{(b)})}$ .

Because  $\sigma$  is a permutation and the original blocks partition  $\mathbf{V}_L$ , the transported cells also partition  $\mathbf{V}_L$ , although they may be interleaved with respect to the original partition. We now combine the two changes of representation:  $D$  changes coordinates within each block, and  $\sigma$  moves those coordinates to their transported cells. The family  $\mathcal{T}$  contains all transformations of this form.

**Definition 8** (Transformation family  $\mathcal{T}$ ). The family of block-diagonal reparameterizations followed by compatible coordinate permutations is

$$\mathcal{T} := \{ \sigma \circ D \mid \sigma \text{ is compatible and } D \in \mathcal{D} \}.$$

Any bijection preserves exactness when the model and its interventions are transported as in Defn. 3. Our goal is stronger: after changing the representation, each high-level variable should still be computed from one cell, and setting that cell should still be a hard intervention. Both properties hold when  $T = \sigma \circ D$ . If  $I = \text{do}(\mathbf{V}^{(b)} = \mathbf{i}_b)$  for some block value  $\mathbf{i}_b \in \text{Val}_{\mathbf{V}^{(b)}}$ , then  $I^T$  sets the transported cell  $\sigma(\mathbf{V}^{(b)})$  to the fixed value  $\sigma_b(D^{(b)}(\mathbf{i}_b))$ . For a general bijection, either property can fail: the transported intervention may depend on the current state, or a high-level variable may depend on coordinates from several cells.

A two-variable XOR transformation illustrates both failures. Let  $\mathbf{V} = \{X_1, X_2\}$  with binary singleton

blocks  $\mathbf{V}^{(1)} = \{X_1\}$  and  $\mathbf{V}^{(2)} = \{X_2\}$ , and let  $\tau(x_1, x_2) = (x_1, x_2)$  map them to  $Z^{(1)}$  and  $Z^{(2)}$ . Consider

$$\begin{aligned} T(x_1, x_2) &= (x_1 \oplus x_2, x_2), \\ T^{-1}(y_1, y_2) &= (y_1 \oplus y_2, y_2). \end{aligned}$$

For the hard intervention  $I = \text{do}(X_1=0)$ , the transported operation is  $I^T(y_1, y_2) = (y_2, y_2)$  which copies the current value of  $Y_2$  into  $Y_1$  rather than setting  $Y_1$  to a constant. It is therefore not a hard intervention in the new coordinates. The transported state map also satisfies

$$(\tau \circ T^{-1})(y_1, y_2) = (y_1 \oplus y_2, y_2).$$

Thus  $Z^{(1)}$  requires both  $Y_1$  and  $Y_2$ , while  $Z^{(2)}$  also requires  $Y_2$ , so the two high-level variables cannot be computed from disjoint cells. Exactness survives relative to the transported intervention algebra, but constructiveness does not.

The XOR example shows that preserving exactness does not necessarily preserve the block-to-concept structure. We now ask which transformations preserve this structure for every constructive abstraction built on the fixed blocks. For each target SCM  $H$ , let  $\mathcal{A}(L, H)$  be the set of constructive abstractions  $(\tau, \omega) : L \rightarrow H$  whose cell for  $Z^{(b)}$  is  $\mathbf{V}^{(b)}$  for every  $b$ , with  $\text{dom}(\omega) = \Psi$  and  $\omega$  the intervention map induced by that alignment (Assum. 1).

**Theorem 1** (Universal maximality of block-diagonal invariance). *For the low-level SCM  $L$  and block partition in Assum. 1, fix any bijection  $T : \text{Val}_{\mathbf{V}_L} \rightarrow \text{Val}_{\mathbf{V}_L}$  and any compatible permutation  $\sigma$ . The following are equivalent:*

- (a) *For every target SCM  $H$  and every  $(\tau, \omega) \in \mathcal{A}(L, H)$ , the transported pair  $(\tau \circ T^{-1}, \omega^T) : L_T \rightarrow H$  is constructive with cells  $\sigma(\mathbf{V}^{(b)})$  for every  $b$ .*
- (b)  *$T = \sigma \circ D$  for some  $D \in \mathcal{D}$ .*

The proof is given in § B.1. The theorem identifies the largest family of transformations for which constructiveness is guaranteed, connecting the result directly to block-identifiability. In block-identifiability results, the permitted permutations are whole-block permutations, meaning that each original block is mapped onto another original block. Together with invertible transformations within blocks, these transformations are a special case of  $\mathcal{T}$ . The family  $\mathcal{T}$  is broader because it also permits compatible coordinate permutations that move a block to an interleaved transported cell.

There is one important qualification: the guarantee is universal. Every transformation in  $\mathcal{T}$  preserves every constructive abstraction built on the fixed blocks.

A transformation outside  $\mathcal{T}$  must break at least one such abstraction, but it may still preserve a particular one. For example, take four binary variables with blocks  $\{X_1, X_2\}$  and  $\{X_3, X_4\}$ , and let  $\tau(\mathbf{x}) = (x_1 \oplus x_2, x_3 \oplus x_4)$ . Consider the transformation

$$T_+(\mathbf{x}) = (x_1 \oplus x_3, x_2 \oplus x_3, x_3, x_4),$$

which mixes the two blocks. Nevertheless,

$$(\tau \circ T_+^{-1})(\mathbf{y}) = (y_1 \oplus y_2, y_3 \oplus y_4).$$

The first high-level variable is unchanged because the added value  $x_3$  appears twice and therefore disappears under XOR:  $(x_1 \oplus x_3) \oplus (x_2 \oplus x_3) = x_1 \oplus x_2$ . Thus  $T_+ \notin \mathcal{T}$  still preserves this particular abstraction.

This does not contradict Thm. 1, which concerns preservation of every abstraction built on the fixed blocks. Its proof shows that a transformation outside  $\mathcal{T}$  fails for the identity block abstraction, whose cell maps retain the complete value of each block. Preserving one particular abstraction therefore does not imply that  $T \in \mathcal{T}$ .

**Implication for block-identifiable representations.** Under Assum. 1, the transformations left unresolved by block-identifiability do not change the predicted high-level effect of an intervention. After transporting the alignment and interventions, whole-block permutations and invertible within-block reparameterizations yield the same solution set in  $H$  for every corresponding intervention. Thm. 1 identifies the boundary of this universal guarantee: every  $T \in \mathcal{T}$  preserves every block-aligned constructive abstraction, whereas each  $T \notin \mathcal{T}$  breaks at least one, although a particular abstraction may survive. Thus complete blocks, rather than their non-identifiable internal coordinates, are the stable units of intervention. When  $\pi_b$  is many-to-one, the high-level intervention  $\text{do}(Z^{(b)}=z)$  corresponds to every complete-block intervention  $\text{do}(\mathbf{V}^{(b)}=\mathbf{i}_b)$  with  $\mathbf{i}_b \in \pi_b^{-1}(z)$ . Two abstractions whose cell maps have the same fibers in every cell differ only by a relabeling of their high-level values (Corollary 1). The abstraction does not distinguish these low-level interventions, so implementing a high-level intervention requires an additional selection rule. This ambiguity becomes relevant in § 4, where a nearest-point rule selects one representative from the fiber.

## 4 EXPERIMENTS

**Goal.** We test whether a high-level intervention in the human-interpretable model  $H$  can be translated into a block intervention in a reference low-level model  $L_0$  and transported to independently trained low-level

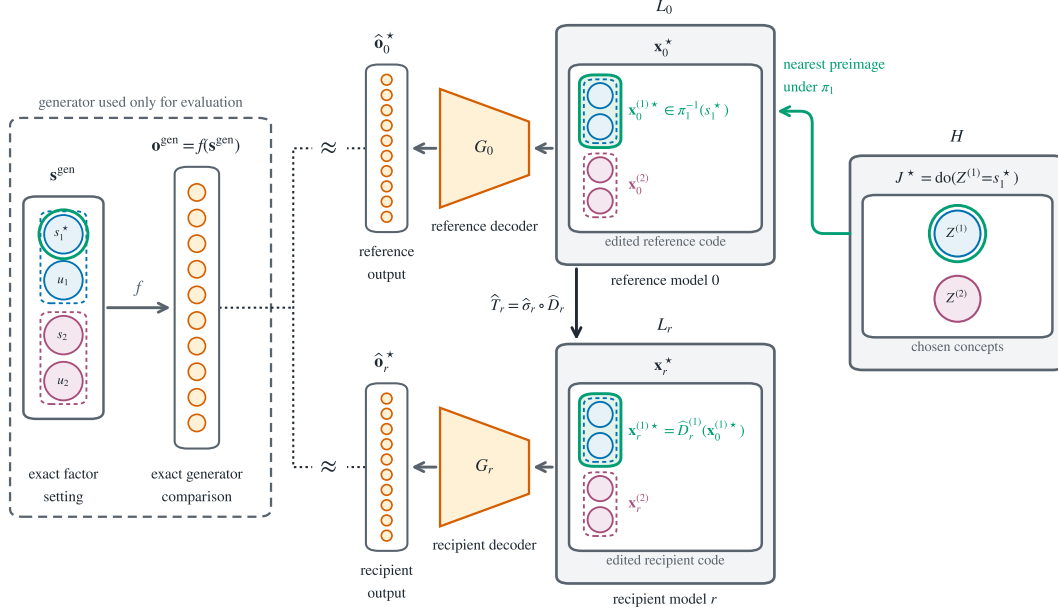


Figure 2: **Concept-directed steering and transport between independently trained additive autoencoders.** Starting from a base observation, the high-level intervention  $J^* = \text{do}(Z^{(1)} = s_1^*)$  in  $H$  is implemented in the reference model  $L_0$  by editing block 1. Here,  $\pi_1$  maps that block to  $Z^{(1)}$ , and  $\mathbf{x}_0^{(1)*}$  is the closest block value satisfying  $\pi_1(\mathbf{x}_0^{(1)*}) = s_1^*$ ; block 2 remains unchanged. The fitted transformation  $\hat{T}_r = \hat{o}_r \circ \hat{D}_r$  transfers the edit to recipient model  $L_r$ , where  $\hat{D}_r$  maps corresponding blocks and  $\hat{o}_r$  accounts for their ordering. The decoders  $G_0$  and  $G_r$  produce the edited outputs  $\hat{o}_0^*$  and  $\hat{o}_r^*$ . For evaluation, the exact factor setting  $\mathbf{s}^{\text{gen}} = (s_1^*, u_1, s_2, u_2)$  changes only  $s_1$ , setting it to the intervention value  $s_1^*$ , and the data-generating map  $f$  produces the comparison observation  $\mathbf{o}^{\text{gen}} = f(\mathbf{s}^{\text{gen}})$ . Both decoded outputs are compared with this same observation, as indicated by the dotted lines. Green marks the intervention, while blue and pink identify the two blocks throughout.

models  $L_r$  while preserving its high-level effect. Fig. 2 summarizes this steering, transport, and evaluation procedure.

**Setting.** Synthetic factors  $(s_1, u_1, s_2, u_2)$  generate observations through the nonlinear additive map  $\mathbf{o} = A_1\phi(s_1, u_1) + A_2\phi(s_2, u_2)$ , where  $\phi(s, u) = (s, u, s^2, su, u^2)$ ; the distribution and mixing matrix are specified in § C. Following additive-decoder models (Lachapelle et al., 2023), model  $r$  has an encoder  $E_r(\mathbf{o}) = \mathbf{x}_r$  and a decoder  $G_r(\mathbf{x}_r) = \hat{\mathbf{o}}_r$ . The four-dimensional representation  $\mathbf{x}_r$  has two two-dimensional blocks  $\mathbf{x}_r^{(1)}, \mathbf{x}_r^{(2)}$ . Each block is passed through its own nonlinear decoder branch, and the two outputs are added:

$$\hat{\mathbf{o}}_r = G_r^{(1)}(\mathbf{x}_r^{(1)}) + G_r^{(2)}(\mathbf{x}_r^{(2)}).$$

We treat  $\mathbf{x}_r$  as the state of the low-level SCM  $L_r$  and the two learned blocks as its cells  $\mathbf{V}^{(1)}, \mathbf{V}^{(2)}$ . The high-level SCM  $H$  keeps only  $Z^{(1)} = s_1$  and  $Z^{(2)} = s_2$ ; omitting  $u_1, u_2$  makes each cell map  $\pi_b$  many-to-one.

**Protocol.** We train one reference autoencoder  $(E_0, G_0)$  and ten additional autoencoders  $(E_r, G_r)$  that may receive an intervention transported from the reference model. After training, we freeze every model and retain for transport only those that reconstruct held-out observations with normalized mean squared error (NMSE) at most 0.01. The architecture, training settings, data splits, and definition of NMSE are given in § C, and we provide the code as supplementary material.

We do not initially know which learned block corresponds to  $Z^{(1)}$  and which corresponds to  $Z^{(2)}$ . We therefore use fresh examples in which each observation  $\mathbf{o}$  is paired with its known concept values  $(Z^{(1)}, Z^{(2)}) = (s_1, s_2)$ , and encode  $\mathbf{o}$  with  $E_0$ . Let  $\pi_{bj}$  denote the affine map fitted from learned block  $\mathbf{V}^{(b)}$  to concept  $Z^{(j)}$ . We fit all four maps and compare the two possible one-to-one assignments:  $(\pi_{11}, \pi_{22})$  and  $(\pi_{12}, \pi_{21})$ . On held-out examples, we choose the assignment whose two maps best predict  $Z^{(1)}$  and  $Z^{(2)}$ . After indexing the blocks by their selected concepts, these maps become the cell maps  $\pi_1$  and  $\pi_2$ , defining

the empirical reference abstraction

$$\tau(\mathbf{x}_0) := (\pi_1(\mathbf{x}_0^{(1)}), \pi_2(\mathbf{x}_0^{(2)})), \quad (7)$$

which approximates  $(s_1, s_2)$ . The nuisance variables  $u_1, u_2$  are not used to fit these maps.

We estimate  $T_r = \sigma_r \circ D_r$  by encoding the same observations with  $E_0$  and  $E_r$ , giving paired codes  $\mathbf{x}_0 = E_0(\mathbf{o})$  and  $\mathbf{x}_r = E_r(\mathbf{o})$ . We fit affine maps between all pairs of blocks and choose the one-to-one pairing with the lowest held-out code error. This gives both the estimated permutation  $\hat{\sigma}_r$  and the fitted within-block maps  $\hat{D}_r^{(b)}$ . We then index recipient blocks by their matched reference blocks, so that  $\hat{D}_r^{(b)}(\mathbf{x}_0^{(b)}) \approx \mathbf{x}_r^{(b)}$ . In the original output order,  $\hat{T}_r = \hat{\sigma}_r \circ \text{Diag}(\hat{D}_r^{(1)}, \hat{D}_r^{(2)})$ .

The purpose of  $H$  is to express interventions in the concepts we understand, but the edit must ultimately be applied to  $L$ . The high-level intervention  $\text{do}(Z^{(b)}=s_b^*)$  does not have a unique low-level implementation because  $\pi_b$  is many-to-one. Given a base observation, we encode  $\mathbf{x}_0 = E_0(\mathbf{o}_{\text{base}})$  and keep its other block fixed while choosing  $\mathbf{x}_0^{(b)*} \in \pi_b^{-1}(s_b^*)$  closest to the current block  $\mathbf{x}_0^{(b)}$ . Thus  $H$  specifies the fiber, while the nearest-point rule uses information unavailable in  $H$  to select one low-level representative. Replacing block  $b$  gives the edited reference code  $\mathbf{x}_0^*$ , which decodes as  $\hat{\mathbf{o}}_0^* = G_0(\mathbf{x}_0^*)$ . We encode the same base observation as  $\mathbf{x}_r = E_r(\mathbf{o}_{\text{base}})$  and replace its corresponding block by

$$\mathbf{x}_r^{(b)*} := \hat{D}_r^{(b)}(\mathbf{x}_0^{(b)*}), \quad (8)$$

and decode the edited recipient code as  $\hat{\mathbf{o}}_r^* = G_r(\mathbf{x}_r^*)$ . The closed-form affine projection used to choose  $\mathbf{x}_0^{(b)*}$  is given in § C.

The known map used to generate the synthetic observations is used only to evaluate the edits. For the intervention  $\text{do}(Z^{(b)}=s_b^*)$ , we construct

$$\mathbf{o}^{\text{gen}} = A_b \phi(s_b^*, u_b) + A_{3-b} \phi(s_{3-b}, u_{3-b}),$$

where  $u_b$  and  $(s_{3-b}, u_{3-b})$  come from the base observation. Thus only  $s_b$  is changed, setting it to  $s_b^*$ . We compare both the reference output  $\hat{\mathbf{o}}_0^*$  and every recipient output  $\hat{\mathbf{o}}_r^*$  with this same observation. We measure agreement using the NMSE between each decoded output and  $\mathbf{o}^{\text{gen}}$ . Because the generator is injective, a low NMSE indicates that the edit sets  $s_b$  to  $s_b^*$  while preserving its nuisance value and the other factor pair. These factor values and the generator are used only for evaluation, not to choose or transport the edit. Full experimental details and additional diagnostics are in § C.

**Results.** The reference model and four of the ten additional autoencoders pass the reconstruction thresh-

Table 1: **Reconstruction and concept-intervention accuracy across retained additive autoencoders.** Rows show one reference additive autoencoder and four independently trained recipient additive autoencoders, identified by training seed. “Recon.” is the normalized mean squared error (NMSE) between each model’s decoded reconstructions and held-out synthetic input observations; only models with reconstruction NMSE at most 0.01 are retained. The  $Z^{(b)}$  columns report intervention error. An intervention on  $Z^{(b)}$  is implemented as a block edit in the reference model and transported to each recipient model. Each decoded edited output is compared with the exact synthetic observation obtained by changing only  $s_b$  to the intervention value while keeping  $u_b$  and  $(s_{3-b}, u_{3-b})$  fixed. Each NMSE pools 5,000 examples; lower values are better.

| Model (seed)   | Recon.   | $Z^{(1)}$ | $Z^{(2)}$ |
|----------------|----------|-----------|-----------|
| Reference (0)  | 0.007312 | 0.007407  | 0.007378  |
| Recipient (2)  | 0.007927 | 0.011931  | 0.010487  |
| Recipient (3)  | 0.007762 | 0.009342  | 0.010307  |
| Recipient (5)  | 0.007825 | 0.010572  | 0.010985  |
| Recipient (10) | 0.007779 | 0.009505  | 0.010923  |

old; the other six fail clearly and are excluded before steering. The reference cell maps identify their concepts unambiguously, and paired codes identify the corresponding blocks in every retained recipient model. Steering succeeds in the reference model and after transport to all four recipient models: every decoded observation has NMSE at most 0.012 against the common exact generator comparison for the corresponding intervention. Tab. 1 reports the reconstruction and steering NMSE for every retained model.

This experiment is an approximate illustration of the theory; it does not establish that the learned models form an exact constructive abstraction. The accompanying supplementary code generates the synthetic data, trains the models, and reproduces the reported results from fixed seeds.

## 5 RELATED WORKS

**Causal abstractions and alignment.** Prior work introduced exact transformations and constructive causal abstractions for relating causal models at different levels (Rubenstein et al., 2017; Beckers and Halpern, 2019; Geiger et al., 2025) and applied causal abstractions to neural networks (Geiger et al., 2021). Distributed alignment search (DAS) finds a rotation of a neural representation that places each specified high-level variable in its own subspace, allowing interventions on that variable to be implemented by editing

the corresponding subspace (Geiger et al., 2024). DAS searches for an alignment to one high-level model; we instead characterize the representation changes guaranteed to preserve any constructive abstraction with the given blocks. Other work studies how abstractions can be discovered. Causal feature learning (Chalupka et al., 2017) and unsupervised causal abstraction discovery (Saulus et al., 2026) seek macro-variables or high-level models, while Li et al. (2025) use paired observations before and after unknown interventions to determine which latent variables cannot be distinguished and must therefore be represented as one higher-level variable. Our theory assumes that the low- and high-level models, blocks, and abstraction are given and asks which changes of representation preserve that abstraction.

**Block identifiability.** Independent subspace analysis (ISA) identifies latent subspaces up to their permutation and invertible transformations within them (Hyvärinen and Hoyer, 2000; Theis, 2006); additive decoders give an analogous guarantee under additive structure (Lachapelle et al., 2023). These transformations are a special case of the family  $\mathcal{T}$  characterized by Thm. 1. Causal representation learning more broadly seeks to recover latent causal variables or their structure (Schölkopf et al., 2021). Our theory starts from a given abstraction and studies whether it remains valid under the transformations left open by block identifiability.

**Feature blocks and steering.** For an SAE, each output coordinate of the encoder is one learned feature in our terminology. SAEs with different numbers of output coordinates need not learn the same features: a concept represented by one coordinate in an SAE with fewer output coordinates may be represented by several more specific coordinates in an SAE with more output coordinates (Bricken et al., 2023; Leask et al., 2025). Sparse Shift Autoencoders identify one-dimensional steering directions up to permutation and rescaling under their assumptions (Joshi et al., 2025). Block-sparse featurizers instead represent a concept using a multidimensional block and recover low-dimensional visual concept manifolds whose internal coordinates are defined only up to rotation (Fel et al., 2026). Together, these works motivate asking when steering a concept represented by a block retains its high-level causal meaning under changes of representation.

**Approximate causal abstractions.** Approximate causal abstraction allows the high-level model to reproduce the low-level system only up to a quantified error (Beckers et al., 2019), and abstraction maps can be

learned by enforcing consistency across interventional distributions (Zennaro et al., 2023). Our maximality theorem concerns exact abstractions; the synthetic experiment measures approximate steering and transport rather than proving an approximate abstraction guarantee.

## 6 DISCUSSION

**Limitations.** Our result assumes deterministic SCMs  $L$  and  $H$ , a fixed block partition and its correspondence to the high-level variables, and a given exact constructive abstraction  $(\tau, \omega)$ . It applies to hard interventions on complete blocks and does not show how these assumptions can be learned from data. Our synthetic experiment studies a controlled and approximate setting: the additive architecture matches the synthetic generator, concept-labelled observations are used to fit the cell maps  $\pi_b$ , observations encoded by different additive autoencoders are used to fit transformations between their representations, and only autoencoders that reconstruct accurately are retained. The experiment therefore illustrates concept-directed steering and its transport in a simplified setting, but it does not establish an exact abstraction or provide a general method for discovering one.

**Future work.** The main open problems are to recover the blocks and  $(\tau, \omega)$  under explicit identifiability assumptions; extend the maximality result to stochastic SCMs, soft interventions, and quantitatively approximate abstractions; and evaluate concept-directed steering and transport in real neural representations, where exact generator comparisons are unavailable.

**Conclusion.** Under a given block-aligned constructive abstraction, the ambiguity left by block-identifiability does not destroy the high-level causal interpretation. Block-diagonal reparameterizations, up to compatible coordinate permutations, are the maximal class guaranteed to preserve every such abstraction under the transported alignment; transformations outside this class may preserve a particular abstraction but have no universal guarantee (Thm. 1). In a controlled synthetic setting, we illustrate this connection by translating high-level concept interventions into edits of entire learned blocks and transporting those edits between independently trained representations (§ 4).

### AI use statement

Generative AI tools were used to assist with implementing, debugging, and testing the experimental code and to suggest limited edits to the manuscript for clarity and concision. All reported results were pro-

duced by executing the code, and the authors reviewed the AI-assisted code, experimental outputs, and manuscript edits. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

## REFERENCES

- Sander Beckers and Joseph Y. Halpern. Abstracting Causal Models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019. URL <http://arxiv.org/abs/1812.03789>. arXiv:1812.03789.
- Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. Approximate Causal Abstraction. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019. URL <http://arxiv.org/abs/1906.11583>. arXiv:1906.11583.
- Trenton Bricken, Adly Templeton, Joshua Batten, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL <https://transformer-circuits.pub/2023/monosemantic-features>.
- Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. Causal feature learning: an overview. *Behaviormetrika*, 44(1):137–164, 2017. doi: 10.1007/s41237-016-0008-2. Verified against Crossref 2026-09-05.
- Thomas Fel, Matthew Kowal, Mozes Jacobs, Dron Hazra, Usha Bhalla, Lee Sharkey, Lucius Bushnaq, Satchel Grant, Tal Haklay, Thomas Icard, Can Rager, Michael Pearce, Daniel Wurgaft, Aiden Swann, Fenil Doshi, Siddharth Boppana, Curt Tigges, Nick Cammarata, Thomas Serre, Vasudev Shyam, Owen Lewis, Thomas McGrath, Jack Merullo, Ekdeep Singh Lubana, and Atticus Geiger. Structuring Sparsity: Block-Sparse Featureizers Capture Visual Concept Manifolds, June 2026. URL <http://arxiv.org/abs/2606.25234>. arXiv:2606.25234 [cs].
- Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal Abstractions of Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL <http://arxiv.org/abs/2106.02997>. arXiv:2106.02997.
- Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, pages 160–187. PMLR, March 2024. URL <https://proceedings.mlr.press/v236/geiger24a.html>. ISSN: 2640-3498.
- Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL <https://www.jmlr.org/papers/v26/geiger25a.html>. arXiv:2301.04709; journal version of the 2023 preprint.
- Aapo Hyvärinen and Patrik Hoyer. Emergence of Phase- and Shift-Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces. *Neural Computation*, 12(7):1705–1720, 2000. doi: 10.1162/089976600300015312. Verified against Crossref 2026-09-05.
- Aapo Hyvärinen, Hiroaki Sasaki, and Richard E. Turner. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. *arXiv:1805.08651 [cs, stat]*, February 2019. URL <http://arxiv.org/abs/1805.08651>. arXiv:1805.08651.
- Shruti Joshi, Andrea Dittadi, Sébastien Lachapelle, and Dhanya Sridhar. Identifiable Steering via Sparse Autoencoding of Multi-Concept Shifts, February 2025. URL <http://arxiv.org/abs/2502.12179>. arXiv:2502.12179 [cs].
- Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvärinen. Variational Autoencoders and Nonlinear ICA: A Unifying Framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, June 2020. URL <http://proceedings.mlr.press/v108/khemakhem20a.html>. ISSN: 2640-3498.
- Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation, July 2023. URL <http://arxiv.org/abs/2307.02598>. arXiv:2307.02598 [cs, stat].
- Patrick Leask, Bart Bussmann, Michael T. Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse Autoencoders Do Not Find Canonical Units of Analysis. In *International Conference on*

- Learning Representations (ICLR)*, 2025. URL <https://openreview.net/forum?id=9ca9eHNrdH>. arXiv:2502.04878.
- Xiushi Li, Sékou-Oumar Kaba, and Siamak Ravanbakhsh. On the Identifiability of Causal Abstractions. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 3241–3249. PMLR, 2025. URL <https://proceedings.mlr.press/v258/li25g.html>.
- Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. *arXiv:2006.15055 [cs, stat]*, October 2020. URL <http://arxiv.org/abs/2006.15055>. arXiv: 2006.15055.
- Aaron Mueller, Andrew Lee, Shruti Joshi, Ekdeep Singh Lubana, Dhanya Sridhar, and Patrik Reizinger. From isolation to entanglement: When do interpretability methods identify and disentangle known concepts?, December 2025. URL <http://arxiv.org/abs/2512.15134>. arXiv:2512.15134 [cs].
- Patrik Reizinger, Yash Sharma, Matthias Bethge, Bernhard Schölkopf, Ferenc Huszár, and Wieland Brendel. Jacobian-based Causal Discovery with Nonlinear ICA. *Transactions on Machine Learning Research*, April 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=2Yo9xqR6Ab>.
- Paul K. Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M. Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. Causal Consistency of Structural Equation Models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*. arXiv, July 2017. doi: 10.48550/arXiv.1707.00819. URL <http://arxiv.org/abs/1707.00819>. arXiv:1707.00819 [cs, stat].
- Théo Saulus, Simon Lacoste-Julien, and Dhanya Sridhar. Unsupervised Causal Abstractions Discovery, June 2026. URL <http://arxiv.org/abs/2606.19594>. arXiv:2606.19594.
- Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward Causal Representation Learning. *Proceedings of the IEEE*, 109(5):612–634, February 2021. doi: 10.1109/JPROC.2021.3058954. URL <http://arxiv.org/abs/2102.11107>. arXiv: 2102.11107 version: 1.
- Shohei Shimizu, Patrik O Hoyer, Aapo Hyvarinen, and Antti Kerminen. A Linear Non-Gaussian Acyclic Model for Causal Discovery. page 28, 2006.
- Fabian J. Theis. Towards a General Independent Subspace Analysis. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 1361–1368, 2006. URL <https://proceedings.neurips.cc/paper/2006/hash/20479c788fb27378c2c99eadcf207e7f-Abstract.html>. Verified against DBLP 2026-09-05.
- Fabio Massimo Zennaro, Máté Drávucz, Geanina Apachitei, W. Dhammika Widanage, and Theodoros Damoulas. Jointly learning consistent causal abstractions over multiple interventional distributions. In *Conference on Causal Learning and Reasoning (CLear)*, volume 213 of *Proceedings of Machine Learning Research*, pages 88–121, 2023. URL <https://proceedings.mlr.press/v213/zennaro23a.html>. Author list corrected 2026-09-05 against the PMLR page; arXiv:2301.05893.

---

# When Do Learned Blocks Preserve Causal Meaning?

## Block Identifiability and Constructive Causal Abstractions:

### Supplementary Materials

---

## A TERMINOLOGY

Tab. 2 collects the terms the paper builds on with their notation, a one-line reading, and the place each is defined.

Table 2: Core terminology and notation used in the paper.

| Term                                      | Notation                           | Meaning                                                                                                                                                   | Defined in                                                |
|-------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| Representation<br>(feature vector)        | $\mathbf{x} = g(\mathbf{o})$       | Featurizer output whose coordinates form $\mathbf{V}_L$                                                                                                   | § 2                                                       |
| Activation vector                         | $\mathbf{o}$                       | Activation read from the network and supplied to the featurizer $g$                                                                                       | § 2                                                       |
| Featurizer                                | $g$                                | Maps network activations to $\mathbf{x}$ ; examples include SAEs and block-sparse featurizers                                                             | § 2                                                       |
| Learned feature                           | $X \in \mathbf{V}_L$               | Coordinate variable of the featurizer output                                                                                                              | § 2                                                       |
| Concept                                   | $Z^{(b)} \in \mathbf{V}_H$         | High-level variable, possibly encoded across several learned coordinates                                                                                  | § 2                                                       |
| Low-level model                           | $L, \mathbf{V}_L$                  | Representation-level SCM on the featurizer output, whose coordinates carry the block partition                                                            | § 2 and Assum. 1                                          |
| High-level model                          | $H, \mathbf{V}_H$                  | SCM over the human-interpretable concepts                                                                                                                 | § 2                                                       |
| Deterministic SCM                         | $M$                                | Signature $\Sigma$ plus mechanisms $\{F_X\}$ ; $\text{Solve}(M)$ are its fixed points                                                                     | § 2; Geiger et al. (2025)                                 |
| Hard intervention                         | $I$ (on $L$ ), $J$ (on $H$ )       | Replaces the mechanisms of a variable set by constants; identified with its overwrite map; the empty intervention is the identity                         | § 2                                                       |
| Intervention algebra                      | $\Psi$                             | Composition algebra isomorphic to hard interventions; here, hard interventions on zero or more complete blocks                                            | § 2; Assum. 1                                             |
| Exact transformation                      | $(\tau, \omega)$                   | State and intervention maps commuting with $\text{Solve}$                                                                                                 | eq. (1); Rubenstein et al. (2017); Geiger et al. (2025)   |
| Bijjective translation                    | $T : L \rightarrow L_T$            | Coordinate change with transported interventions $I^T = T \circ I \circ T^{-1}$ ; exact relative to the transported intervention algebra                  | Defn. 3; Geiger et al. (2025)                             |
| Constructive abstraction                  | —                                  | Exact transformation whose state and intervention maps are induced by an alignment                                                                        | Defn. 4; Beckers and Halpern (2019); Geiger et al. (2025) |
| Block partition                           | $\mathbf{V}^{(b)}$                 | Disjoint grouping of the low-level variables into $B$ blocks                                                                                              | eq. (4)                                                   |
| Alignment cell, cell map                  | $\mathbf{V}^{(b)}, \pi_b$          | The block from which $Z^{(b)}$ is computed and the surjection producing its value                                                                         | Assum. 1                                                  |
| Block-diagonal reparameterization         | $D \in \mathcal{D}$                | Bijection acting within each block, mixing none                                                                                                           | Defn. 5                                                   |
| Transformation family                     | $T \in \mathcal{T}$                | Maps $T = \sigma \circ D$ with compatible $\sigma$ and $D \in \mathcal{D}$                                                                                | Defn. 8 and Thm. 1                                        |
| Transported model                         | $L_T$                              | Mechanisms conjugated through $T$ , so $T : L \rightarrow L_T$ is a bijjective translation                                                                | Defn. 3                                                   |
| Compatible permutation, transported cells | $\sigma, \sigma(\mathbf{V}^{(b)})$ | Permutation of coordinates with matching value domains, carrying each block to a possibly interleaved cell; whole-block permutations are the special case | Defn. 7                                                   |

| Term                  | Notation        | Meaning                                                                                                                  | Defined in            |
|-----------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------|-----------------------|
| Fibers                | $\pi_b^{-1}(z)$ | The low-level cell values mapped to the same high-level value; equal fibers make two abstractions differ by a relabeling | § 3.2 and Corollary 1 |
| Block-identifiability | —               | Recovery of blocks up to their permutation and invertible within-block maps; ISA gives the linear within-subspace case   | § 1                   |

## B PROOFS

This section contains the proof of the main theorem and a secondary corollary.

### B.1 Universal maximality of block-diagonal invariance

*Proof of Thm. 1. (b)⇒(a).* Assume  $T = \sigma \circ D$  with  $D = \text{Diag}(D^{(1)}, \dots, D^{(B)})$ , and fix a target  $H$  and  $(\tau, \omega) \in \mathcal{A}(L, H)$ . Constructiveness of the transported pair requires two properties: an alignment with cells  $\sigma(\mathbf{V}^{(b)})$  inducing  $\tau \circ T^{-1}$ , and exactness of  $(\tau \circ T^{-1}, \omega^T) : L_T \rightarrow H$ .

*Step 1: the transported alignment.* Let  $\pi_b : \text{Val}_{\mathbf{V}^{(b)}} \rightarrow \text{Val}_{Z^{(b)}}$  be the original cell maps, so

$$\tau(\mathbf{x}) = (\pi_b(\mathbf{x}_{\mathbf{V}^{(b)}}))_{b=1}^B. \quad (9)$$

For  $\mathbf{y} = T(\mathbf{x})$ ,  $\sigma^{-1}$  moves the value in  $\sigma(\mathbf{V}^{(b)})$  back to  $\mathbf{V}^{(b)}$ , and  $(D^{(b)})^{-1}$  undoes the within-block change of coordinates:

$$(T^{-1}(\mathbf{y}))_{\mathbf{V}^{(b)}} = (D^{(b)})^{-1}(\sigma_b^{-1}(\mathbf{y}_{\sigma(\mathbf{V}^{(b)})})). \quad (10)$$

Define the cell map on the transported cell  $\sigma(\mathbf{V}^{(b)})$  by

$$\pi_b^T := \pi_b \circ (D^{(b)})^{-1} \circ \sigma_b^{-1}. \quad (11)$$

The cells  $\sigma(\mathbf{V}^{(b)})$  partition  $\mathbf{V}_L$  (Defn. 7); each  $\pi_b^T$  is surjective because  $\pi_b$  is and the remaining factors are bijective; and substituting the block formula for  $T^{-1}$  into  $\tau$  gives

$$(\tau \circ T^{-1})(\mathbf{y}) = (\pi_b^T(\mathbf{y}_{\sigma(\mathbf{V}^{(b)})}))_{b=1}^B, \quad (12)$$

the required alignment-induced form.

*Step 2: exactness.* Let  $I \in \text{dom}(\omega) = \Psi$  set the complete blocks indexed by  $S \subseteq \{1, \dots, B\}$  to values  $\mathbf{i}_b$ . Its conjugate  $I^T$  sets each transported cell  $\sigma(\mathbf{V}^{(b)})$ ,  $b \in S$ , to

$$\mathbf{i}_b^T := \sigma_b(D^{(b)}(\mathbf{i}_b)), \quad (13)$$

and the transported cell map assigns this cell value the same macro-value:

$$\pi_b^T(\mathbf{i}_b^T) = \pi_b(\mathbf{i}_b). \quad (14)$$

Hence  $\omega^T$ , as defined in Defn. 6, coincides with the intervention map induced by the transported alignment (Defn. 4). Since  $T : L \rightarrow L_T$  is a bijective translation (Defn. 3), and bijective translations are exact (Geiger et al., 2025),

$$\text{Solve}(L_T^{I^T}) = T(\text{Solve}(L^I)), \quad (15)$$

so

$$(\tau \circ T^{-1})(\text{Solve}(L_T^{I^T})) = \tau(\text{Solve}(L^I)) \quad (16)$$

$$= \text{Solve}(H^{\omega(I)}) \quad (17)$$

$$= \text{Solve}(H^{\omega^T(I^T)}), \quad (18)$$

using the transported-solution equation, exactness of  $(\tau, \omega)$ , and  $\omega^T(I^T) = \omega(I)$ . The transported pair is exact and alignment-induced, hence constructive, proving (a).

(a) $\Rightarrow$ (b). Fix the compatible permutation  $\sigma$  from the theorem statement. We construct a block-diagonal bijection  $D$  with  $T = \sigma \circ D$  via the identity block abstraction: a coarse abstraction may discard the information needed to recover  $T$ , whereas the identity abstraction retains the complete value of every block.

*Step 1: the identity abstraction lies in the quantified family.* Define

$$\tau_{\text{id}}(\mathbf{x}) = (\mathbf{x}_{\mathbf{V}^{(b)}})_{b=1}^B, \quad (19)$$

which packages each complete block as one product-valued macro-variable and loses no information. Let  $H_{\text{id}}$  be the SCM with mechanisms conjugated through  $\tau_{\text{id}}$  (Defn. 3), with interventions transported accordingly. The pair is a bijective translation, hence exact, and is induced by the cells  $\mathbf{V}^{(b)}$  with identity cell maps, so it lies in  $\mathcal{A}(L, H_{\text{id}})$ ; because condition (a) ranges over every target  $H$ , it applies to this abstraction.

*Step 2: blockwise factorization of  $T^{-1}$ .* Condition (a) says  $\tau_{\text{id}} \circ T^{-1}$  is constructive under the fixed transported cells  $\sigma(\mathbf{V}^{(b)})$ , so it supplies surjective cell maps

$$\pi_b^T : \text{Val}_{\sigma(\mathbf{V}^{(b)})} \rightarrow \text{Val}_{\mathbf{V}^{(b)}} \quad (20)$$

whose outputs are the complete original block values. Equating the  $b$ -th component of the alignment-induced state map with the  $b$ -th component of  $\tau_{\text{id}} \circ T^{-1}$  gives

$$(T^{-1}(\mathbf{y}))_{\mathbf{V}^{(b)}} = \pi_b^T(\mathbf{y}_{\sigma(\mathbf{V}^{(b)})}) \quad (21)$$

for every  $b$ : each output block of  $T^{-1}$  depends only on its assigned transported input block.

*Step 3: the block factors are bijective.* Each  $\pi_b^T$  is surjective as a cell map in the transported alignment. For injectivity, suppose  $\mathbf{u} \neq \mathbf{u}'$  but  $\pi_b^T(\mathbf{u}) = \pi_b^T(\mathbf{u}')$ ; completing  $\mathbf{u}$  and  $\mathbf{u}'$  to total states  $\mathbf{y} \neq \mathbf{y}'$  that agree on every other transported cell, the component equations give  $T^{-1}(\mathbf{y}) = T^{-1}(\mathbf{y}')$ , contradicting injectivity of  $T^{-1}$ . Hence every  $\pi_b^T$  is bijective.

*Step 4: assembling  $D$ .* With the bijection  $\sigma_b : \text{Val}_{\mathbf{V}^{(b)}} \rightarrow \text{Val}_{\sigma(\mathbf{V}^{(b)})}$ , define

$$(D^{(b)})^{-1} := \pi_b^T \circ \sigma_b : \text{Val}_{\mathbf{V}^{(b)}} \rightarrow \text{Val}_{\mathbf{V}^{(b)}}, \quad (22)$$

and let  $D^{-1} := \text{Diag}((D^{(1)})^{-1}, \dots, (D^{(B)})^{-1})$ . Comparing  $D^{-1} \circ \sigma^{-1}$  with  $T^{-1}$  on each block:

$$((D^{-1} \circ \sigma^{-1})(\mathbf{y}))_{\mathbf{V}^{(b)}} = (D^{(b)})^{-1}(\sigma_b^{-1}(\mathbf{y}_{\sigma(\mathbf{V}^{(b)})})) \quad (23)$$

$$= \pi_b^T(\mathbf{y}_{\sigma(\mathbf{V}^{(b)})}) \quad (24)$$

$$= (T^{-1}(\mathbf{y}))_{\mathbf{V}^{(b)}}, \quad (25)$$

by the definition of  $(D^{(b)})^{-1}$  and the block-factorization equation of Step 2. The maps agree on every block, so  $T^{-1} = D^{-1} \circ \sigma^{-1}$  and  $T = \sigma \circ D$ , which is condition (b). Exactness is not used in this algebraic factorization; it is needed in Step 1 to place the identity abstraction in the family quantified over by condition (a).  $\square$

## B.2 Uniqueness up to macro value relabeling

Two surjective cell maps with the same fibers identify the same low-level values and therefore differ only by a relabeling of their high-level values. The following corollary applies this observation to abstractions related by a block reparameterization.

**Corollary 1** (Uniqueness up to macro value relabeling). *Let  $L_2 = (L_1)_D$  for  $D \in \mathcal{D}$ . Let  $(\tau_1, \omega_1) : L_1 \rightarrow H_1$  and  $(\tau_2, \omega_2) : L_2 \rightarrow H_2$  be constructive abstractions with the same cells  $\mathbf{V}^{(b)}$  and cell maps  $\pi_{1,b}, \pi_{2,b}$ , and suppose that for every  $b$  the fibers of  $\pi_{2,b}$  and of  $\pi_{1,b} \circ (D^{(b)})^{-1}$  coincide on  $\text{Val}_{\mathbf{V}^{(b)}}$ . Then there is a unique blockwise bijection  $\phi = (\phi_b)_b$ ,  $\phi_b : \text{Val}_{Z_1^{(b)}} \rightarrow \text{Val}_{Z_2^{(b)}}$ , with  $\tau_2 = \phi \circ \tau_1 \circ D^{-1}$  and  $\omega_2(I^D) = \phi \circ \omega_1(I) \circ \phi^{-1}$  for every  $I \in \Psi$ , and  $H_1$  and  $H_2$  have  $\phi$ -related solution sets under every intervention in  $\omega_1(\Psi)$ . If moreover  $\pi_{2,b} = \pi_{1,b} \circ (D^{(b)})^{-1}$  for every  $b$ , then  $\phi = \text{id}$  and the two macro-models have the same solution sets under every such intervention.*

*Proof of Corollary 1.* By Thm. 1 with  $\sigma = \text{id}$ ,  $(\tau_1 \circ D^{-1}, \omega_1^D)$  is constructive  $L_2 \rightarrow H_1$  with cell maps  $\pi_{1,b} \circ (D^{(b)})^{-1}$ . Fix  $b$ : this map and  $\pi_{2,b}$  are surjections of  $\text{Val}_{\mathbf{V}^{(b)}}$  with the same fibers, so  $\phi_b(z) := \pi_{2,b}(\mathbf{u})$  for any  $\mathbf{u} \in (\pi_{1,b} \circ (D^{(b)})^{-1})^{-1}(z)$  is a well-defined bijection, and it is the only map with  $\pi_{2,b} = \phi_b \circ \pi_{1,b} \circ (D^{(b)})^{-1}$  because the latter map is onto. Hence  $\tau_2 = \phi \circ \tau_1 \circ D^{-1}$ , and by the definition of the induced intervention map (Defn. 4) the intervention  $I^D$  setting the cells  $S$  to  $D^{(b)}(\mathbf{i}_b)$  is sent by  $\omega_2$  to  $\text{do}(Z_2^{(b)} = \phi_b(\pi_{1,b}(\mathbf{i}_b)), b \in S)$ , which is  $\phi \circ \omega_1(I) \circ \phi^{-1}$  because  $\phi$  acts coordinatewise. Exactness of both pairs then gives, for every  $I \in \Psi$ ,  $\phi(\text{Solve}(H_1^{\omega_1(I)})) = \phi(\tau_1(\text{Solve}(L_1^I))) = \tau_2(\text{Solve}(L_2^{I^D})) = \text{Solve}(H_2^{\omega_2(I^D)})$ , the empty intervention included; the last claim follows because equal cell maps force  $\phi_b = \text{id}$ .  $\square$

## C CONCEPT-DIRECTED STEERING AND TRANSPORT DETAILS

**Generator and model.** Each example draws  $s_1, u_1, s_2, u_2$  independently from  $\text{Uniform}[-1, 1]$ , lifts each pair by  $\phi(s, u) = (s, u, s^2, su, u^2)^\top$ , and forms

$$\mathbf{o} = A_1 \phi(s_1, u_1) + A_2 \phi(s_2, u_2) \in \mathbb{R}^{10}, \quad (26)$$

where  $A = [A_1 \ A_2] \in \mathbb{R}^{10 \times 10}$  is the orthogonal factor of one seeded random matrix and is fixed across all splits and models. The construction satisfies the reference-side sufficient-nonlinearity condition of Lachapelle et al. (2023); the exact learned-side conditions of that result are not claimed for finite trained networks. For model  $r$ , the encoder  $E_r$  has widths 10, 64, 64, 4, and each branch of the decoder  $G_r$  has widths 2, 64, 64, 10, with tanh hidden activations and linear outputs. Training minimizes reconstruction mean squared error (MSE) with Adam, using batch size 1024, learning rate  $10^{-3}$ , and weight decay  $5 \times 10^{-4}$ . Training runs for at most 2000 epochs and stops after 200 epochs without a validation-MSE improvement of at least  $10^{-7}$ ; the best validation checkpoint is retained. All runs use CPU-only local execution with four PyTorch threads. No factor label enters autoencoder training or checkpoint selection.

**Data splits and fitted maps.** The experiment uses 20,000/2,000 observations for autoencoder training and validation and 5,000 held-out observations for reconstruction eligibility. The reference cell maps use separate calibration, validation, and test splits of sizes 2,000, 2,000, and 5,000. The cross-model maps use 5,000/2,000 paired, unlabelled observations for fitting and validation. For each of 5,000 further base observations, we independently sample one intervention value  $s_1^*$  and one intervention value  $s_2^*$  from  $\text{Uniform}[-1, 1]$ . This gives 5,000 steering cases per concept, or 10,000 interventions in total; each corresponding edit is evaluated in the reference model and four retained recipient models.

For each possible pairing of reference block  $b$  and concept  $j$ , an affine candidate map

$$\pi_{bj}(\mathbf{x}_0^{(b)}) = \mathbf{w}_{bj}^\top \mathbf{x}_0^{(b)} + c_{bj} \quad (27)$$

is fitted by least squares. The one-to-one assignment with the highest validation  $R^2$  is retained, and its two maps become  $\pi_1$  and  $\pi_2$ . For every possible reference–recipient block pair, an affine map is fitted in standardized coordinates by least squares. The one-to-one pairing with the lowest worst-block validation NMSE estimates  $\hat{\sigma}_r$  and is retained. After indexing each recipient block by its matched reference block, the selected within-block map  $\hat{D}_r^{(b)}$  satisfies  $\hat{D}_r^{(b)}(\mathbf{x}_0^{(b)}) \approx \mathbf{x}_r^{(b)}$ . This second fit sees paired codes of the same observations but no factor values.

**Selecting a point in the fiber.** Let  $\mathbf{x}_0^{(b)}$  be the intervened block of the reference code for the base observation. For the affine  $\pi_b$ , the closest point in its fiber over  $s_b^*$  is

$$\mathbf{x}_0^{(b)*} = \mathbf{x}_0^{(b)} + \frac{s_b^* - \pi_b(\mathbf{x}_0^{(b)})}{\|\mathbf{w}_b\|^2} \mathbf{w}_b, \quad \pi_b(\mathbf{x}_0^{(b)*}) = s_b^*. \quad (28)$$

Replacing this block produces the reference code  $\mathbf{x}_0^*$  and decoded observation  $\hat{\mathbf{o}}_0^* = G_0(\mathbf{x}_0^*)$ . Recipient model  $r$  receives  $\mathbf{x}_r^{(b)*} = \hat{D}_r^{(b)}(\mathbf{x}_0^{(b)*})$ , producing  $\mathbf{x}_r^*$  and  $\hat{\mathbf{o}}_r^* = G_r(\mathbf{x}_r^*)$ . This Euclidean projection is a convenient way to choose one representative of the fiber; it is not a unique inverse of  $\pi_b$ .

**Evaluation.** For each intervention, we construct one exact generator observation by replacing only the intervened concept value in the factors of the base observation:

$$\mathbf{o}^{\text{gen}} = A_b \phi(s_b^*, u_b) + A_{3-b} \phi(s_{3-b}, u_{3-b}), \quad (29)$$

where  $u_b$  and the second factor pair come from the base observation. The same  $\mathbf{o}^{\text{gen}}$  is used to evaluate the complete reference and recipient observations by computing the NMSE between  $\widehat{\mathbf{o}}_r^*$  and  $\mathbf{o}^{\text{gen}}$  for  $r = 0$  and every retained recipient. Because the generator is injective and  $\phi$  contains the original factor coordinates, this common comparison tests whether the edit sets  $s_b$  to  $s_b^*$  while preserving its nuisance value and the other factor pair.

For predictions  $\widehat{\mathbf{y}}_1, \dots, \widehat{\mathbf{y}}_N$  and comparisons  $\mathbf{y}_1, \dots, \mathbf{y}_N$ , we define

$$\text{NMSE}(\widehat{\mathbf{y}}, \mathbf{y}) := \frac{\sum_{n=1}^N \|\widehat{\mathbf{y}}_n - \mathbf{y}_n\|^2}{\sum_{n=1}^N \|\mathbf{y}_n - \bar{\mathbf{y}}\|^2}, \quad \bar{\mathbf{y}} := \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n. \quad (30)$$

For cross-model map validation, each recipient coordinate is standardized using the corresponding training-set mean and standard deviation before applying this definition.

The thresholds were fixed before the run: reconstruction NMSE at most 0.01; reference-readout and decoded-concept  $R^2$  at least 0.95; cross-model map NMSE at most 0.05; untargeted  $s$  and  $u$   $R^2$  at least 0.95; generator-validity residual at most three times the reconstruction floor; and final common-comparison NMSE at most 0.02.

**Numerical results.** The reference model reconstructs with NMSE 0.007312. Four of ten additional autoencoders pass the same gate, with NMSE between 0.007762 and 0.007927; the other six lie between 0.099071 and 0.100609. The two selected reference cell maps have test  $R^2$  between 0.999602 and 0.999618, while each block has no predictive power for the other concept. The selected cross-model maps have validation NMSE between 0.001985 and 0.005839; the alternative block pairings have NMSE near 1.

All retained models satisfy the reconstruction criterion, and every predeclared calibration, map-validation, steering, preservation, validity, and common-observation check passes.