When are Post-hoc Conceptual Explanations Identifiable? (Supplementary material)

Tobias Leemann[1,2,†] **Michael Kirchhof**[1,†] **Yao Rong**[1,2] **Enkelejda Kasneci**[2] **Gjergji Kasneci**[2]

[1]University of Tübingen, Tübingen, Germany
[2]Technical University of Munich, Munich, Germany
[†]equal contribution

A ADDITIONAL RELATED WORK

Orthogonality constraints and disentanglement for generative models. In the context of generative adversarial networks (GANs) [Goodfellow et al., 2014], the problem of analyzing and discovering interpretable directions has be studied recently by Voynov and Babenko [2020]. Ren et al. [2022] propose a contrastive approach to discover interpretable directions using pretrained generative models. Wei et al. [2021] have proposed an orthogonality regularization of the Jacobian, which resulted in more interpretable generative abilities. Ramesh et al. [2018] constrain the right-singular vectors of a generator Jacobian to be unit directions, which corresponds to column-wise orthogonal generator Jacobians. We go beyond these works by providing rigorous results on identifiability and by extending the scope to a encoder-only models.

B PROOFS

B.1 ROTATIONS DESTROY ORTHOGONALITY LEMMA

We start by first proving an auxiliary lemma. We show that orthogonality of Jacobians, i.e., $\boldsymbol{J}_f \boldsymbol{J}_f^\top = \boldsymbol{S}$ with a diagonal matrix $\boldsymbol{S}$ will be destroyed in the general case when a rotation $\boldsymbol{R}$ is applied, such that $\boldsymbol{J}_{Rf} \boldsymbol{J}_{Rf}^\top = \boldsymbol{R}\boldsymbol{J}_f \boldsymbol{J}_f^\top \boldsymbol{R}^\top = \boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top$ is not a diagonal matrix anymore.

Lemma B.1 (Rotations destroy orthogonality patterns.) *Let $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ be a diagonal matrix, $\boldsymbol{S} = \operatorname{diag}(\boldsymbol{s})$ with diagonal entries $\boldsymbol{s} > 0$ and $s_i \neq s_j, \forall i \neq j$, i.e., all diagonal entries of $\boldsymbol{S}$ are different and positive. Let $\boldsymbol{R} \in \mathbb{R}^{K\times K}$ be any rotation matrix with $\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}$. If $\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top$ is a diagonal matrix, $\boldsymbol{R}$ must a signed permutation matrix (a permutation matrix where entries can be ± 1).*

Proof. With $\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top = \operatorname{diag}(\lambda_1, \dots \lambda_K)$, we have for each unit vector $\boldsymbol{e}^{(i)}$, $i = 1, \dots, K$, that

$$\boldsymbol{R}\boldsymbol{S}\boldsymbol{R}^\top \boldsymbol{e}^{(i)} = \lambda_i \boldsymbol{e}^{(i)} \ . \tag{2}$$

We can represent $\boldsymbol{R}$ by its rows, $\boldsymbol{R} = [\boldsymbol{r}_1, \dots, \boldsymbol{r}_K]^\top$ where each $\boldsymbol{r}_i \in \mathbb{R}^K$. In this notation, $\boldsymbol{R}^T \boldsymbol{e}^{(i)} = \boldsymbol{r}_i$, i.e., multiplication of the transpose with a unit vector will select the row $\boldsymbol{r}_i$. This results in

$$\boldsymbol{R}\boldsymbol{S}\boldsymbol{r}_i = \lambda_i \boldsymbol{e}^{(i)} \tag{3}$$

Because $\boldsymbol{R}$ is invertible and square, we can left-multiply the equation by $\boldsymbol{R}^\top$. Using $\boldsymbol{R}^T \boldsymbol{e}^{(i)} = \boldsymbol{r}_i$ again, we arrive at

$$\boldsymbol{S}\boldsymbol{r}_i = \lambda_i \boldsymbol{r}_i. \tag{4}$$

This implies that all $\boldsymbol{r}_i$ are eigenvectors of the matrix $\boldsymbol{S}$ with the eigenvalues λ_i. By the initial assumption, $\boldsymbol{S}$ is a diagonal matrix with all-different entries s_i. The eigenvectors of such a matrix are only scaled unit vectors $\boldsymbol{e}^{(j)}$. Thus, each $\boldsymbol{r}_i$ will be

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

a scaled unit-vector. The constraint of $\boldsymbol{R}$ being an orthogonal matrix enforces the $\boldsymbol{r}_i$ to be mutually different unit vectors with length 1. Therefore, $\boldsymbol{R}$ necessarily has the form of a signed permutation. □

Note that the converse is also true. If $\boldsymbol{R}$ is a signed permutation matrix, $\boldsymbol{RSR}^\top$ will be diagonal.

B.2 PCA ENSURES IDENTIFIABILITY (THEOREM 3.1)

Theorem B.1 (PCA identifiability, Theorem 3.1) *Let $z_k, k = 1, \dots, K$, be uncorrelated random variables with non-zero and unequal variancecs. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ is an orthonormal matrix. If an orthonormal post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ results in mutually uncorrelated components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a matrix where $|s_{ii}| = 1$ for $i \in 1, \dots K$.*

Proof. Since both $\boldsymbol{M}$ and $\boldsymbol{D}$ are orthogonal, $\boldsymbol{MD} = \boldsymbol{Q}$ is also orthogonal. Our post-hoc transformation resulted in uncorrelated components, i.e., $\mathrm{Cov}(\boldsymbol{Qx}) = \boldsymbol{Q}\,\mathrm{Cov}(\boldsymbol{x})\boldsymbol{Q}^\top\boldsymbol{\Gamma}$ is diagonal, where $\boldsymbol{\Gamma}$ is some diagonal matrix. Thus, $\boldsymbol{Q}\,\mathrm{Cov}(\boldsymbol{x})\boldsymbol{Q}^\top$ is diagonal, too. We also know that our original components are uncorrelated with unequal variances, i.e., $\mathrm{Cov}(\boldsymbol{x}) = \mathrm{diag}(\boldsymbol{s})$ with $\boldsymbol{s} > 0$ and $s_i \neq s_j, \forall i \neq j$. Our helper Lemma B.1 then implies that $\boldsymbol{Q}$ must be a signed permutation. Thus, $\boldsymbol{z}' := \boldsymbol{Me} = \boldsymbol{MDz} = \boldsymbol{Qz} =: \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a matrix where $|s_{ii}| = 1$ for $i \in 1, \dots K$. □

B.3 ICA ENSURES IDENTIFIABILITY (THEOREM 3.2)

Theorem B.2 (ICA identifiability, Theorem 3.2) *Let $z_k, k = 1, \dots, K$, be independent random variables with non-zero variances where at most one component is Gaussian. Let $\boldsymbol{e} = \boldsymbol{Dz}$, where $\boldsymbol{D} \in \mathbb{R}^{K\times K}$ has full rank. If a post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ results in mutually independent components $(z'_1, \dots, z'_K) = \boldsymbol{z}' = \boldsymbol{Me}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a scaling matrix.*

Proof. (1) We know that $\boldsymbol{z}' = MDz =: \boldsymbol{C}'z$. Let us start with an additional assumption that both $\boldsymbol{z}'$ and $\boldsymbol{z}$ have unit variances. Then, by Comon [1994, App. A .1], $\boldsymbol{C}'$ must be orthonormal.

Let us recall the following result

Theorem B.3 (Theorem 11 from Comon [1994]) *Let $\boldsymbol{x}$ be a vector with independent components, of which at most one is Gaussian, and whose densities are not reduced to a point-like mass. Let $\boldsymbol{C}$ be an orthogonal $K \times K$ matrix and $\boldsymbol{z}$ the vector $\boldsymbol{z} = \boldsymbol{Cx}$. Then the following three properties are equivalent:*

1. *The components z_i are pairwise independent.*
2. *The components z_i are mutually independent.*
3. *$\boldsymbol{C} = \boldsymbol{SP}$ where $\boldsymbol{S}$ is diagonal, $\boldsymbol{P}$ is a permutation.*

Since $\boldsymbol{z}$ fulfills the conditions of this this theorem and $\boldsymbol{z}'$ has mutually independent entries, we know that $\boldsymbol{C}' = \boldsymbol{SP}$.

(2) We now allow arbitrary variances, i.e., $\mathrm{Cov}(\boldsymbol{z}') = \boldsymbol{\Lambda}$ and $\mathrm{Cov}(\boldsymbol{z}) = \boldsymbol{\Gamma}$ where both covariance matrices are positive diagonal matrices. $\boldsymbol{z}' = \boldsymbol{MDz} = \boldsymbol{C}'\boldsymbol{z} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{C}'\boldsymbol{\Gamma}^{1/2}\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} =: \boldsymbol{\Lambda}^{1/2}\boldsymbol{C}''\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z}$. This is equivalent to $(\boldsymbol{\Lambda}^{-1/2}\boldsymbol{z}') = \boldsymbol{C}''(\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z})$. These rescaled random vectors both have unit variances, so (1) implies that $\boldsymbol{C}'' = \boldsymbol{S}'\boldsymbol{P}'$. We can plug this back into the previous equation and see that $\boldsymbol{z}' = \boldsymbol{\Lambda}^{1/2}\boldsymbol{C}''\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{S}'\boldsymbol{P}'\boldsymbol{\Gamma}^{-1/2}\boldsymbol{z} =: \boldsymbol{P}'\boldsymbol{S}''\boldsymbol{z}$. Thus, $\boldsymbol{z}' = \boldsymbol{Me} = \boldsymbol{MDz} = \boldsymbol{P}'\boldsymbol{S}''\boldsymbol{z}$, where $\boldsymbol{P}' \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S}'' \in \mathbb{R}^{K\times K}$ is a scaling matrix. □

B.4 TRANSFER LEMMA

DMA and IMA are based on structures in the Jacobian of the generative process. To be able to use them in the encoder and ultimately discover concepts, we first show that if an encoder mirrors the behavior of the generative process, up to a rotation and scale, its Jacobians must also mirror the Jacobians of the generative process.

Lemma B.2 (Transfer lemma) *Let $\boldsymbol{f}$ be a faithful encoder for the generative process $\boldsymbol{g}$ and further $\boldsymbol{f} \circ \boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{PSz}$ $\forall \boldsymbol{z} \in \mathcal{Z}$ where $\boldsymbol{P} \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix. Then $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{P}'\boldsymbol{S}'\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{z})^\top$ where $\boldsymbol{P}' \in \mathbb{R}^{K\times K}$ is a permutation and $\boldsymbol{S}' \in \mathbb{R}^{K\times K}$ is a diagonal matrix.*

Proof. Let $\boldsymbol{z} \in \mathcal{Z}$ be arbitrary. $(\boldsymbol{f} \circ \boldsymbol{g})(\boldsymbol{z}) = \boldsymbol{PSz}$ implies $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J_g}(\boldsymbol{z}) = \boldsymbol{PS}$. Since $\boldsymbol{f}$ is faithful to $\boldsymbol{g}$, $\boldsymbol{S}$ has full rank, i.e., $\boldsymbol{S} = \text{diag}(\alpha_1, \ldots, \alpha_K)$ with $\alpha_k \in \mathbb{R}_{\neq 0}, k = 1, \ldots, K$.

Now, let us write $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})) = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^\top$ with $\boldsymbol{v}_i \in \mathbb{R}^L$. Similarly, we can write $\boldsymbol{J_g}(\boldsymbol{z}) = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ with $\boldsymbol{w}_i \in \mathbb{R}^L, i = 1, \ldots, K$.

Let us focus on an individual row of $\boldsymbol{J_f}$, i.e., let $k \in \{1, \ldots, K\}$ be a fixed index of a row. Since $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J_g}(\boldsymbol{z}) = \boldsymbol{PS}$ and $\boldsymbol{P}$ is a permutation matrix with exactly one 1 per row, there is precisely one column index k' such that the k-th row and k'-th column of $\boldsymbol{PS}$ is non-zero. This setup allows drawing certain conclusions about the vector $\boldsymbol{v}_k$. Let $j = 1, \ldots, K$ denote an arbitrary column of $\boldsymbol{PS}$. Then,

(i) if $j = k'$, then $\boldsymbol{v}_k^\top \boldsymbol{w}_{k'} = \alpha_{k'} \neq 0$. In consequence, $\boldsymbol{v}_k \neq 0$, $\boldsymbol{w}_{k'} \neq 0$ and so we can decompose $\boldsymbol{v}_k = \boldsymbol{a}_k + \boldsymbol{b}_k$, where $\boldsymbol{a}_k \in \text{span}(\{\boldsymbol{w}_{k'}\}) \setminus \{0\}$ and $\boldsymbol{b}_k \in \text{span}(\{\boldsymbol{w}_{k'}\})^\perp$, where $^\perp$ denotes the orthogonal complement. Because $\text{span}(\{\boldsymbol{w}_{k'}\}) = \{\mu \boldsymbol{w}_{k'} | \mu \in \mathbb{R}\}$, we know that $\boldsymbol{a}_k = \frac{\alpha_{k'}}{\|\boldsymbol{w}_{k'}\|_2^2} \boldsymbol{w}_{k'}$.

(ii) if $j \neq k'$, then $\boldsymbol{v}_k^\top \boldsymbol{w}_j = 0$. With (i), it follows that $\boldsymbol{b}_k \in \text{span}\left(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K\}\right)^\perp = \text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp$.

Since $\boldsymbol{f}$ is faithful to $\boldsymbol{g}$, we know that for each $\mathbf{c} \in \text{span}(\boldsymbol{J_g}(\boldsymbol{z}))^\perp$, $\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{z}))\mathbf{c} = \mathbf{0}$ ant therefore $\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{b}_k = \mathbf{0}$ This demands that the k-th component of the product is also 0, i.e., $\mathbf{v}_k \boldsymbol{b}_k = (\boldsymbol{a}_k + \boldsymbol{b}_k)^\top \boldsymbol{b}_k = \boldsymbol{a}_k^\top \boldsymbol{b}_k + \boldsymbol{b}_k^\top \boldsymbol{b}_k = 0$. By design $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ are orthogonal such that immediately follows $\boldsymbol{b}_k = \mathbf{0}$ Hence, $\boldsymbol{v}_k = \boldsymbol{a}_k + \mathbf{0} = \frac{\alpha_{k'}}{\|\boldsymbol{w}_{k'}\|_2^2} \boldsymbol{w}_{k'} + \mathbf{0}$ for our selected row k. Globally, this means $\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z})) = \boldsymbol{P}'\boldsymbol{S}'\boldsymbol{J_g}(\boldsymbol{z})^\top$, with some scaling matrix $\boldsymbol{S}'$ and permutation matrix $\boldsymbol{P}'$. □

B.5 DISJOINT MECHANISMS ENSURE IDENTIFIABILITY (THEOREM 3.3)

Theorem B.4 (Identifiability under DMA, Theorem 3.3) *Let $\boldsymbol{g}$ have disjoint mechanisms and $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. If a full-rank post-hoc transformation $\boldsymbol{M} \in \mathbb{R}^{N \times N}$ results in disjoint rows in the Jacobian $\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}))$ for some $\boldsymbol{z} \in \mathcal{Z}$, then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K \times K}$ is a scaling matrix.*

Proof. We know that $\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{D}$ and $\boldsymbol{D}$ has full rank. Since $\boldsymbol{M}$ also has full rank, there exists a non-singular matrix $\boldsymbol{E}'$ such that $\boldsymbol{M} = \boldsymbol{E}'\boldsymbol{D}^{-1}$. We can rewrite $\boldsymbol{E}' = \boldsymbol{SE}$, where $\boldsymbol{E}$ has normalized rows and $\boldsymbol{S}$ is a diagonal matrix.

Since $\boldsymbol{D}^{-1}\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{I}$ and $\boldsymbol{g}$ is DMA, we can apply the transfer lemma (Lemma B.2). It implies that $\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z))$ has orthogonal rows.

Suppose now for contradiction that $\boldsymbol{E}$ was not a permutation matrix. This means that without loss of generality the first row must contain at least two columns whose entries are not equal to zero. Since $\boldsymbol{E}$ has full rank, there must be a second row with a non-zero entry in at least one of these columns. Since $\boldsymbol{D}^{-1}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}_a))$ has disjoint rows, $\boldsymbol{SED}^{-1}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}_a)) = \boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{g}(\boldsymbol{z}_a))$ can no longer have disjoint rows. This contradicts the assumption. Hence, $\boldsymbol{E}$ must be a permutation matrix $\boldsymbol{P}$. This give $\boldsymbol{z}' = \boldsymbol{Me} = \boldsymbol{PSD}^{-1}\boldsymbol{Dz} = \boldsymbol{PSz}$. □

B.6 INDEPENDENT MECHANISMS ENSURE IDENTIFIABILITY (THEOREM 3.4)

Theorem B.5 (Identifiability under IMA, Theorem 3.4) *Let $\boldsymbol{g}$ adhere to IMA. Let $\boldsymbol{f}$ be a faithful encoder to $\boldsymbol{g}$. Suppose we have obtained an $\boldsymbol{f}' = \boldsymbol{Mf}$ with a full-rank $\boldsymbol{M} \in \mathbb{R}^{K \times K}$ and orthogonal rows in its Jacobian $\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))$, i.e, $\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))\boldsymbol{J}_{f'}(\boldsymbol{g}(\boldsymbol{z}))^\top = \boldsymbol{\Sigma}(\boldsymbol{z})$ where $\boldsymbol{\Sigma}(\boldsymbol{z})$ is diagonal. If additionally for two points $\boldsymbol{z}_a, \boldsymbol{z}_b \in \mathcal{Z}$ and $\gamma_i \coloneqq \frac{\Sigma_{ii}(\boldsymbol{z}_b)}{\Sigma_{ii}(\boldsymbol{z}_a)}$ and $\forall i, j = 1 \ldots K, i \neq j : \gamma_i \neq \gamma_j$ (NEMR condition), then $\boldsymbol{Me} = \boldsymbol{PSz}$, where $\boldsymbol{P} \in \mathbb{R}^{K \times K}$ is a permutation and $\boldsymbol{S} \in \mathbb{R}^{K \times K}$ is a scaling matrix.*

Proof. We know that $\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{D}$ and $\boldsymbol{D}$ has full rank. Since $\boldsymbol{M}$ also has full rank, there exists a non-singular matrix $\boldsymbol{E}$ such that $\boldsymbol{M} = \boldsymbol{E}\boldsymbol{D}^{-1}$. We will now show that the solution set of $\boldsymbol{E}$ can be constrained to be a permutation and scaling operation in three steps.

(1) $\boldsymbol{J_{f'}}$ is orthogonal, i.e., $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = (\boldsymbol{M}\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{z}_a)))(\boldsymbol{M}\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{z}_a))^\top = (\boldsymbol{E}\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{E}\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))^\top = \boldsymbol{E}(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))^\top\boldsymbol{E}^\top$. Since $\boldsymbol{D}^{-1}\boldsymbol{f} \circ \boldsymbol{g} = \boldsymbol{I}$ and $\boldsymbol{g}$ is DMA, we can apply the transfer lemma (Lemma B.2) and know that $\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a))$ must have orthogonal rows, i.e., $(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))(\boldsymbol{D}^{-1}\boldsymbol{J}_f(g(z_a)))^\top = \boldsymbol{\Gamma}_a$, where $\boldsymbol{\Gamma}_a$ is some diagonal matrix with full rank. Substituting this back into the previous term, $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{E}^\top$. The same holds for $\boldsymbol{z}_b$, i.e., $\boldsymbol{\Sigma}(\boldsymbol{z}_b) = \boldsymbol{E}\boldsymbol{\Gamma}_b\boldsymbol{E}^\top$.

(2) We've seen in (1) that both $\boldsymbol{\Sigma}(\boldsymbol{z}_a)$ and $\boldsymbol{\Gamma}_a$ are the results of quadratic forms. Hence, their entries are all positive, and strictly positive because they have full rank. Thus we can define $\boldsymbol{Q} := \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{E}\boldsymbol{\Gamma}_a^{1/2}$. Due to (1), $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$, i.e., $\boldsymbol{Q}$ is orthogonal. It is easy to see that $\boldsymbol{E} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2}$. In other words, $\boldsymbol{E}$ must be a (twice) scaled orthogonal matrix.

(3) From (1) we get that

$$\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1} = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{E}^\top(\boldsymbol{E}\boldsymbol{\Gamma}_b\boldsymbol{E}^\top)^{-1} \tag{5}$$

$$\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1} = \boldsymbol{E}\boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1}\boldsymbol{E}^{-1} \tag{6}$$

$$\boldsymbol{E}^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{E} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{7}$$

Now we can insert the result from (2)

$$\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{1/2}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{8}$$

$$\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{1/2}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q} = \boldsymbol{\Gamma}_a^{1/2}\boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1}\boldsymbol{\Gamma}_a^{-1/2} \tag{9}$$

$$\boldsymbol{Q}^\top\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}\boldsymbol{Q} = \boldsymbol{\Gamma}_a\boldsymbol{\Gamma}_b^{-1} \tag{10}$$

$$\tag{11}$$

Due to the NEMR condition, $\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{\Sigma}(\boldsymbol{z}_b)^{-1}$ is a diagonal matrix with unequal positive entries. We can thus apply Lemma B.1 which implies that $\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{S}$ where $\boldsymbol{P}$ is a permutation and $\boldsymbol{S}$ a diagonal matrix. Inserting this back into (2) gives $\boldsymbol{E} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)^{-1/2}\boldsymbol{P}\boldsymbol{S}\boldsymbol{\Gamma}_a^{1/2} = \boldsymbol{P}\boldsymbol{S}'$, where $\boldsymbol{S}'$ is a diagonal matrix. Hence, $\boldsymbol{z}' = \boldsymbol{M}\boldsymbol{e} = \boldsymbol{P}\boldsymbol{S}'\boldsymbol{D}^{-1}\boldsymbol{D}\boldsymbol{z} = \boldsymbol{P}\boldsymbol{S}'\boldsymbol{z}$. □

In the next section, we discuss how the proofs can be turned into analytical solutions to discover the ground truth components.

B.7 ANALYTICAL SOLUTIONS TO CONCEPT DISCOVERY

B.7.1 Disjoint Mechanisms

Under a perfect DMA process $\boldsymbol{g}$ and a noiseless faithful encoder $\boldsymbol{f}$ to $\boldsymbol{g}$, we can compute an analytical solution for $\boldsymbol{M}$ that will result in an encoder $\boldsymbol{f}' = \boldsymbol{M}\boldsymbol{f}$ that is compliant with the *DMA criterion*, i.e., disjoint rows in its Jacobian. Suppose we are provided with a gradient matrix of $\boldsymbol{f}$, $\boldsymbol{J}_f(\boldsymbol{x}_a) \in \mathbb{R}^{K\times L}$. We propose the following steps:

1. Select a submatrix $\boldsymbol{J}_{reg} \in \mathbb{R}^{K\times K}$ of K linearly independent columns in $\boldsymbol{J}_f(\boldsymbol{x}_a)$, such that $\det(\boldsymbol{J}_{reg}) \neq 0$.
2. Compute and return $\boldsymbol{M} = \boldsymbol{J}_{reg}^{-1}$
3. This will result in $\boldsymbol{f}' = \boldsymbol{M}\boldsymbol{f}$ having disjoint rows in its Jacobian.

Proof. $\boldsymbol{J}_f(\boldsymbol{x}_a)$ must be of the form $\boldsymbol{J}_f(\boldsymbol{x}_a) = \boldsymbol{H}^{-1}\boldsymbol{J}_{f^*}(\boldsymbol{x}_a)$ for such an $\boldsymbol{M}$ to exist, where $\boldsymbol{J}_{f^*}$ is the Jacobian of an encoder $\boldsymbol{f}^*$ with disjoint rows and $\boldsymbol{H}$ has full rank. $\boldsymbol{J}_{reg}$ can be written as $\boldsymbol{J}_{reg} = \boldsymbol{H}^{-1}\boldsymbol{J}_{f^*,reg}$, where $\boldsymbol{J}_{f^*,reg}$ is a square submatrix of $\boldsymbol{J}_{f^*}$ with the same selected selected columns. The submatrix $\boldsymbol{J}_{f^*,reg}$ also will be of to be of full rank because it can be written as $\boldsymbol{H}\boldsymbol{J}_{reg}$, which are both full rank. Because of the DMA principle, $\boldsymbol{J}_{f^*,reg}$ again needs to be of the form $\boldsymbol{P}\boldsymbol{S}$ with one component active in each column. Furthermore, $\boldsymbol{M} = \boldsymbol{J}_{reg}^{-1} = (\boldsymbol{H}^{-1}\boldsymbol{P}\boldsymbol{S})^{-1} = \boldsymbol{S}^{-1}\boldsymbol{P}^{-1}\boldsymbol{H}$. As the inverses of scaling and permutation matrices have the same respective form again, $\boldsymbol{M}\boldsymbol{H}^{-1} = \boldsymbol{S}'\boldsymbol{P}'$. Therefore, $\boldsymbol{f}' = \boldsymbol{S}'\boldsymbol{P}'\boldsymbol{f}^*$, maintaining its disjoint Jacobians.

B.7.2 Independent Mechanisms

Suppose we are given matrices $\boldsymbol{\Sigma}(\boldsymbol{z}_a) = \boldsymbol{J}_f(\boldsymbol{x}_a)\boldsymbol{J}_f(\boldsymbol{x}_a)^\top = \boldsymbol{D}^{-1}\boldsymbol{\Gamma}_a(\boldsymbol{D}^{-1})^\top$ and $\boldsymbol{\Sigma}(\boldsymbol{z}_b) = \boldsymbol{J}_f(\boldsymbol{x}_b)\boldsymbol{J}_f(\boldsymbol{x}_b)^\top$. We then apply the following steps

1. $\boldsymbol{U} = \text{inverse}(\text{cholesky}(\boldsymbol{\Sigma}(\boldsymbol{z}_a))$
2. $\boldsymbol{V} = \text{eigenvectors}(\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_b)\boldsymbol{U}^\top)$
3. return $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U}$

Algorithm 1: DMA concept discovery with SGD.

Input: encoder $\boldsymbol{f}$, images $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$
Jacobians $\leftarrow$ Gradient($\boldsymbol{f}$, $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$).detach()
$M \leftarrow K$-dim identity matrix
for L *epochs,* $\boldsymbol{J_f}(\boldsymbol{x}) \in$ *Jacobians* **do**
  $\boldsymbol{U} \leftarrow |\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{x})|$ `// No absolute value operation here for IMA`
  $\boldsymbol{U} \leftarrow$ row-normalize $\boldsymbol{U}$
  loss $\leftarrow ||\boldsymbol{U}\,\boldsymbol{U}^\top - \boldsymbol{I}_K||_F$
  loss.backwards() `// Optimize` M
end
return M

Algorithm 2: DMA concept discovery with SGD (determinant loss).

Input: encoder $\boldsymbol{f}$, images $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$
Jacobians $\leftarrow$ Gradient($\boldsymbol{f}$, $\{\boldsymbol{x}_n\}_{n=1,\dots,N}$).detach()
$M \leftarrow K$-dim identity matrix
for L *epochs,* $\boldsymbol{J_f}(\boldsymbol{x}) \in$ *Jacobians* **do**
  $\boldsymbol{U} \leftarrow |\boldsymbol{M}\boldsymbol{J_f}(\boldsymbol{x})|$ `// No absolute value operation here for IMA`
  $\boldsymbol{V} \leftarrow \boldsymbol{U}\boldsymbol{U}^\top$
  loss $\leftarrow \log\left(\prod_i V_{ii}\right) - \log\det\left(\boldsymbol{V}\right)$
  loss.backwards() `// Optimize` M
end
return M

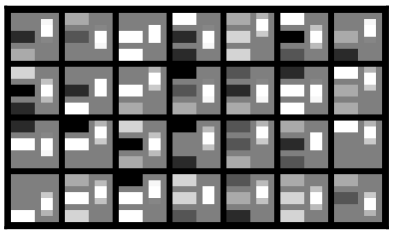
The first step implies that $\boldsymbol{U}^{-1}\boldsymbol{U}^{-\top} = \boldsymbol{\Sigma}(\boldsymbol{z}_a)$ and that $\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{U}^\top = \boldsymbol{I}$. We have thus identified the matrix $\boldsymbol{E}$ from step (2) of the identifiability proof, which has the form $\boldsymbol{U} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{M}$. In step two we compute $\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_b)\boldsymbol{U}^\top = \boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{\Gamma}_b\boldsymbol{\Gamma}_a^{-1/2}\boldsymbol{Q}^\top\boldsymbol{\Lambda}^{1/2} = \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^\top$, where $\boldsymbol{R}$ holds the eigenvalues. Accordingly, by left and right multiplying with $\boldsymbol{V}$, we observe that $(\boldsymbol{V}^\top\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{z}_b)(\boldsymbol{V}^\top\boldsymbol{U})^\top = \boldsymbol{R}$, i.e., $(\boldsymbol{V}^\top\boldsymbol{U})$ solves the orthogonality problem for $\boldsymbol{\Sigma}(\boldsymbol{z}_b)$. We can easily verify that $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U}$ is also a solution for $\boldsymbol{\Sigma}(\boldsymbol{z}_a)$ by computing $\boldsymbol{V}^\top\boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{z}_a)\boldsymbol{U}^\top\boldsymbol{V} = \boldsymbol{I}$. By the identifiability result, $\boldsymbol{H} = \boldsymbol{V}^\top\boldsymbol{U} = \boldsymbol{\Lambda}\boldsymbol{P}\boldsymbol{M}$, a scaled and permuted version of $\boldsymbol{D}^{-1}$, if the additional gradient ratio condition is fulfilled with $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$.

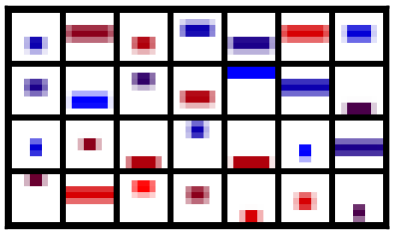
B.8 ALGORITHMS

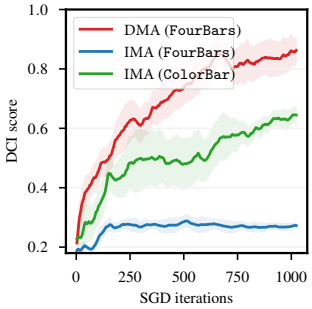
We present the SGD optimization for DMA in Algorithm 1. Note that the algorithm for IMA optimization via SGD can be obtained by just omitting the absolute value operation in the line indicated by the comment. For the smaller toy datasets, we experiment with a version of the algorithm that uses the determinant (see Algorithm 2), similar to the objective put forward by Gresele et al. [2021]. As the determinant operation is hard to backpropagate through and might be unstable, we recommend Algorithm 1 for real-world applications and observed no significant performance differences on the datasets studied in this work.

B.9 EXTENDING GRADIENTS TO GENERAL ATTRIBUTIONS

We make an initial attempt to generalize our method, considering gradients as a simple form of attribution method. Intuitively, $\boldsymbol{J_f} = \nabla_{\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}))$ contains input gradients (termed grad in the remainder) which can be thought of as a simple form of attribution for each component [Simonyan et al., 2013, Shah et al., 2021]. Thus, on a more general level, our proposed approach optimizes for the disjointness of attributions. Thus, we may use other forms of *homogeneous attributions* in place of $\boldsymbol{J_f}$. These are local attribution methods $A_{\boldsymbol{f}} : \mathbb{R}^L \to \mathbb{R}^{K\times L}$ for the encoder $\boldsymbol{f}$ with $A_{\boldsymbol{Mf}}(\boldsymbol{x}) = \boldsymbol{M}A_{\boldsymbol{f}}(\boldsymbol{x})$ that map an instance $\boldsymbol{x}$ to a matrix of attributions for each latent dimension. Besides the above input gradients, this class contains other popular methods such as integrated gradients (IG) [Sundararajan et al., 2017] and smoothed gradients (SG) [Smilkov et al.,







(a) Random samples in the `FourBars` dataset. (b) Random samples in the `ColorBar` dataset. (c) Disentangling gradients of synthetic datasets with SGD.

Figure 7: Random samples drawn from the synthetic datasets (a,b). On the `FourBars` dataset, IMA fails to iterate towards a disentangled solution, because the non-equal magnitudes condition is violated. However, IMA converges on the `ColorBar` dataset, although at a slower rate (c)

2017] (because these methods are linear in $\boldsymbol{f}$). Thus, we can formulate a generalized *disjoint attributions objective*:

$$\min_{\boldsymbol{M}} \sum_{n=1}^{N} \left|\left| \left|\overline{\boldsymbol{M}A_f(x)}\right| \left|\overline{\boldsymbol{M}A_f(x)}\right|^\top - I_K \right|\right|_F^2 . \tag{12}$$

We indicate the row-normalization operation by the overbar, and denote by $|\cdot|$ the element-wise absolute values operation. Without the absolute value operation this results in the *independent attributions objective*.

C EXPERIMENTAL DETAILS

We report the most important implementation details for our experiments in this section. Please confer the actual implementation available online[1] for full information.

C.1 SYNTHETIC DATASETS

We show random samples from both datasets in Figure 7. We provide an additional graphics with the behavior on the synthetic datasets in Figure 7c. They show that SGD exhibits a convergence behavior as predicted by our theory and comparable to the analytical solutions (shown in the main paper).

C.2 ARCHITECTURES

For the disentanglement models, we use the implementations provided by the open source library `disentanglement-pytorch`[2]. For the evaluation measures, we use the implementation of `disentanglement_lib`[3] with their respective default parameters. We use a simple encoder and decoder architecture, that consists of five and six feed-forward convolutional layers respectively and relies on the ReLU activation function.

C.3 CORRELATED SAMPLING

In this paper, we use two methods to introduce correlations between the ground truth components. Both methods rely on proportional resampling: We first draw a batch that has multiple times the final batch size (we use factors from 3-6

[1] https://github.com/tleemann/identifiable_concepts
[2] https://github.com/amir-abdi/disentanglement-pytorch
[3] https://github.com/google-research/disentanglement_lib

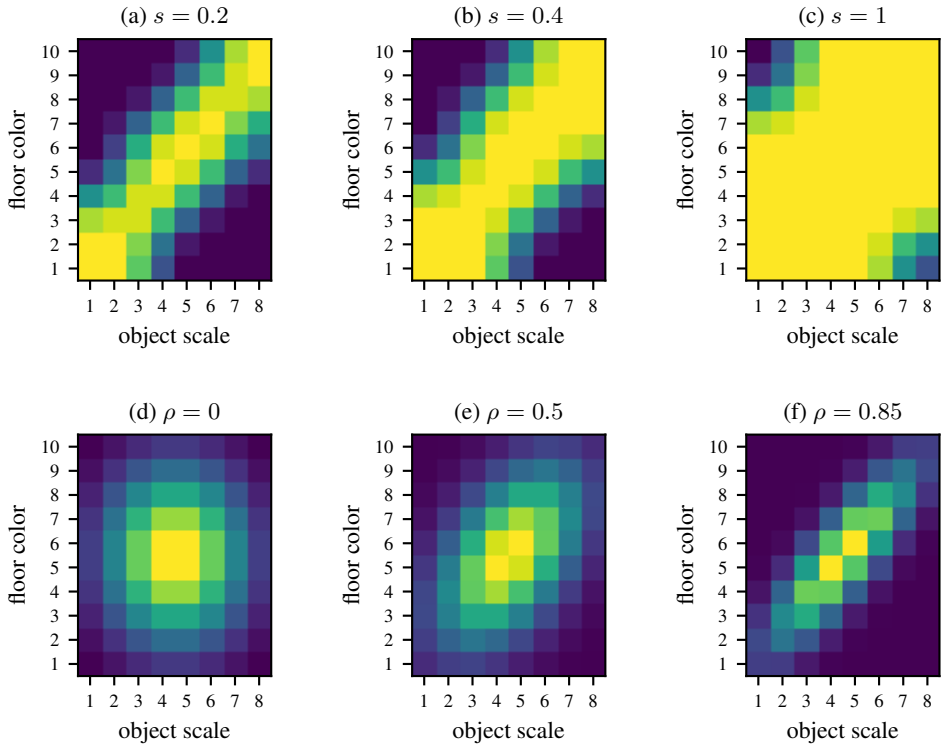


Figure 8: Exemplary correlated densities of the components floor color and object scale under the correlated sampling setup of Gresele et al. [2021] (a – c) and with our Gaussian sampling (d – f). The correlation strength is indicated on top. Purple denotes a low and yellow a high density.

depending on the non-uniformity of the distribution), then compute the (non-normalized) probability of each sample under a given distribution over the component values, and then resample a final batch (with replacement) proportional to these probabilities.

The two methods differ in the probability distribution assigned to the component values. The first setting (used in Sec. 4.2) uses the approach of Träuble et al. [2021]: As visualized in Fig. 8(a) to (c), we pick two components z_1 and z_2, create the grid of possible values, and then lay a diagonal line over this grid. Along this line, we set a normal distribution with a standard deviation s. A higher s means that the distribution gives a higher probability to more component combinations of the grid, whereas a smaller s is more restrictive. Mathematically, it is defined by Träuble et al. [2021] as:

$$p(z_1, z_2) \propto \exp\left(-\frac{(z_1 - \alpha z_2)^2}{2s^2}\right), \tag{13}$$

where $\alpha = z_1^{\max}/z_2^{\max}$ brings the components to a same scale and s is similarly normalized to the maximum values that z_1 and z_2 can take. The remaining components $z_i, i > 2$, are marginalized out of this distribution and thus continue to be sampled uniformly at random.

This setting is limited to one pair of components and also introduces a non-Gaussian distribution over all components. To tackle these limitations and thus to make the distributional challenge harder, we use a different probability distribution in Sec. 4.3. Here, we lay a normal distribution over *all* components, i.e., $z \sim \mathcal{N}(\mu, \Sigma)$, where μ is centered in the middle of the possible values, i.e., $\mu = \frac{z^{\max} - z^{\min}}{2}$. Σ is similarly normalized, since we decompose it into $\Sigma = \text{diag}(\sigma^2)\Gamma$. The vector $\sigma \in \mathbb{R}^K_{>0}$ gives standard deviations for each component via $\sigma^2 = \left(\frac{\mu+0.5}{2}\right)^2$ such that the distribution stretches across the grid of possible values. Note that the $+0.5$ is because the values are assumed to be zero-indexed. Γ is a correlation matrix with 1 on its diagonal. In the first experiment in Sec. 4.3, we correlate only one pair of variables and set their corresponding off-diagonal entries in Γ to ρ. Fig. 8 (d) to (f) show the corresponding marginal distributions of these components. In the

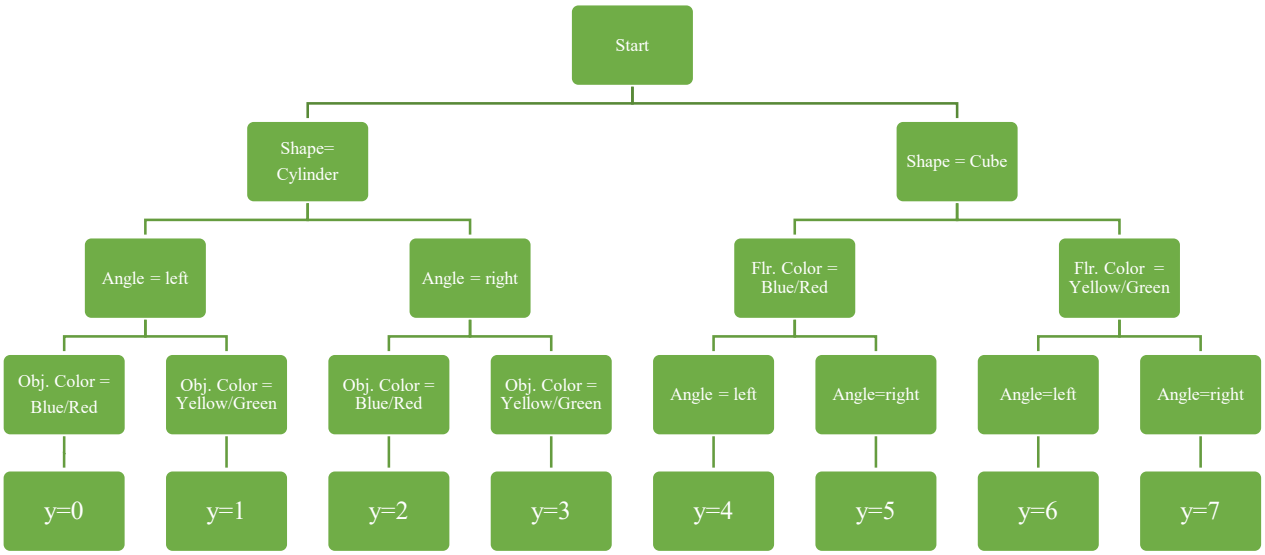


Figure 9: The decision tree setup that we use for the discriminative classification problem. Each image is assigned one out of eight class labels y according to the following decision tree.

second experiment, we fill Γ with several correlations in the following order:

$$\begin{array}{c} \\ z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 4 & 12 & 14 & 9 \end{array} \\ \begin{pmatrix} & 11 & 5 & 10 & 6 \\ & & 3 & 8 & 15 \\ & & & 13 & 7 \\ & & & & 2 \\ & & & & \end{pmatrix} \end{array} \tag{14}$$

where the component order of the rows and columns is $z_1 = \texttt{floor\_color}$, $z_2 = \texttt{background\_color}$, $z_3 = \texttt{object\_color}$, $z_4 = \texttt{object\_scale}$, $z_5 = \texttt{object\_shape}$, $z_6 = \texttt{orientation}$. Here, it is important to ascertain that the covariance matrix stays positive definite. Thus, we start with $\rho = 0.7$, check if the lowest eigenvalue of Σ is at least 0.2, and if not, reduce ρ by a factor of 0.9 until the eigenvalue fulfills this property. While technically it would be enough to have the smallest eigenvalue anywhere above 0, we found that 0.2 helps in numerical stability, for instance when inverting the covariance matrix to compute the multivariate normal distribution density.

C.4 DISCRIMINATIVE SETUP

The decision tree that is used to generate the class distribution is shown in Figure 9. It relies on 4 (binarized) components. We trained a simple CNN classifier for this problem using the cross-entropy loss. In addition to the classification loss terms, we add a regularizer $\|\boldsymbol{z}\|_2^2$, which constrains the latent codes to not grow arbitrarily large, during training. To create a realistic setup, we subsample the dataset to follow a normal distribution as shown in Fig. 8d. We also add label noise near the decision boundary: For objects which have an orientation that is nearly centered, we follow each branch (left/right) with a probability of 50 %. With increasing left-orientedness, the probability of following the left branch increases to almost 100 % in form of a sigmoid function over the actual orientation. We follow the same procedure for the remaining features. We train the classifier for 10k iterations at a batch size of 24 and verify that it reaches an accuracy close to the best-possible one taking the mislabeled samples into account. We add correlations by increasing the chance of the the factors *obj. color* and *floor color* taking the same binary value. We use our disjoint attributions approach to find a $H \in \mathbb{R}^{4\times 6}$ matrix that should map the 6-dimensional latent space of the model to the four binary concepts that are used in the classification task. For the unit directions, we take the first four unit directions of the latent space, for PCA and ICA, we take the most prominent four components discovered for the evaluation with the four annotated ground truth concepts.

C.5 EVALUATION SCORES

Several scores to quantify disentanglement have been proposed in the literature and often emphasize a different aspect of disentanglement [Sepliarskaia et al., 2019]. Among the most common scores is the Disentanglement-Completeness-Informativenss score (DCI) by Eastwood and Williams [2018]. In their work, they propose a metric to measure Disentanglement, that relies on training predictors $\hat{z}_j = f_j(e)$ to predict each individual ground truth component z_j from the learned latent representation e. Furthermore, they compute normalized importance weights P_{ik} that quantify how important learned component e_i is for predicting the ground component z_k. The disentanglement metric computes a row-wise entropy over the P-matrix, which assigns a score of 1, if the learned component e_i is useful for predicting only a single factor and as score of 0, if it is equally useful for predicting all factors. Other commonly used metrics include the Mutual Information Gap (MIG) [Chen et al., 2018], Separated Attribute Predictability (SAP) [Kumar et al., 2018] and the FactorVAE metric [Kim and Mnih, 2018]. However, it is unclear which of these metrics (or if any) also provide useful results in the correlated setting Träuble et al. [2021]. Therefore, to compute the reliable evaluations, we train the model (and the post-processing methods such as PCA, ICA, IMA, DMA) on the correlated dataset, but compute the metrics on samples from the full, *uncorrelated* datasets to avoid distortion in our scores. Träuble et al. noted that the DCI scores were able to discover entanglement between 2 variables [Träuble et al., 2021, Figure 11, Appendix], whereas most other metrics failed even in this case. Therefore, we mainly rely on this score for our experiments but also report results corresponding to Sec. 4.2 for the other scores that show a similar picture in this appendix (Appendix D.4).

C.6 CUB EXPERIMENTS

CUB-200-2011 is a fine-grained dataset containing a total of 11,788 images of 200 bird species (5994 for training and 5794 for testing). We trained a ResNet-50 with two fully-connected (fc) layers (the second fc layer served as a bottleneck layer and took 2048-dim feature vectors as input and output 512-dim ones) on CUB for 100 epochs using a SGD optimizer with an initial learning rate of 0.001. The input images were center cropped to 224×224 pixels. Trained on a standard cross-entropy loss, the ResNet achieved a classification accuracy of on average 77.47% on five random seeds, indicating proper training. After training the classifier, we applied our proposed method to discover components in the embedding space.

CUB provides no ground-truth components since it is a real-world dataset. It does, however, contain 312 attributes semantically describing the bird classes, e.g., wing color or beak shape. These attributes have no guarantee to be complete, but they offer 312 interpretable components. This allows for an attempt to quantify whether our discovered components are interpretable and meaningful by comparing whether they match some of these interpretable ones.

Formally, we are given a set of image feature embeddings $\{\boldsymbol{e}_n\}_{n=1,\dots,N}$, $\boldsymbol{e}_n \in \mathbb{R}^L$ and a matrix $\boldsymbol{H} = (\boldsymbol{h}_1, \dots, \boldsymbol{h}_K) \in \mathbb{R}^{L \times K}$ that contains the directions of discovered components ($L = 512$, $K = 30$). A score s_n^k of n-th image for the k-th discovered component can be calculated by projecting the feature embeddings on that component direction, i.e., $s_n^k = \langle \boldsymbol{e}_n, \boldsymbol{h}_k \rangle$. One pitfall is that s_n^k can be negative, indicating, e.g., a non-black bird for the component "primary color: black", but this opposite attribute is usually encoded in a separate attribute in CUB, e.g., "primary color: white". Thus, we separate the negative and positive values into two components (where we set values of the opposite sign to 0), resulting in $2 \cdot K$ positive scores for each image.

To compare these component scores with the attributes, we make use of the numerical attribute values provided in CUB. First, we average the $2 \cdot K$ component values of all images of a class, to be comparable with the class-wise attributes provided by CUB. This gives us a numerical $2 \cdot K$ dimensional component description and a 312 dimensional attribute description per class. Now, we match the discovered components to the attributes. We compare each discovered component to each attributes via the Spearman's rank correlation coefficient and consider the attribute with the highest score to match the component. These are the matches used in Sec. 4.5. We further use the (average) Spearman's rank correlation across all components to their best-matching attributes to quantify how well the components match to interpretable attributes in Appendix D.7.

C.7 HYPERPARAMETERS FOR THE DISENTANGLEMENT MODELS

We orient our hyperparameter ranges by the works of Träuble et al. [2021], Locatello et al. [2019]. The exact ranges are provided in Tab. 4. We find the best hyperparameters in the ranges for each correlation strength/dataset/model triple separately. Then we train five models from independent seeds to run our experiments. We use the Adam optimizer for all model with a learning rate of 10^{-4}, batch size of 64 and train for 300k iterations (equiv. to 40 epochs on Shapes3D).

Model	Ranges
BetaVAE	$\beta \in \{1, 2, 4, 6, 8, 16\}$
FactorVAE	$\gamma \in \{5, 8, 10, 20, 30, 40, 50, 100\}$
BetaTCVAE	$\beta \in \{1, 2, 4, 6, 8, 10\}$
DIPVAEI	$\lambda_{od} \in \{1, 2, 5, 10, 20, 50\}$

Table 4: The hyperparameter ranges considered in this work.

Dataset	MPI3D-real		
Correlated components	background & object color	background & robot arm dof-1	robot arm dof-1 & robot arm dof-2
BetaVAE	0.340 ± 0.027	0.277 ± 0.026	0.300 ± 0.046
+PCA	0.116 ± 0.008	0.174 ± 0.021	0.154 ± 0.015
+ICA	0.237 ± 0.042	0.205 ± 0.023	0.180 ± 0.021
+Ours (IMA)	$\mathbf{0.355 \pm 0.033}$	$\mathbf{0.349 \pm 0.015}$	$\mathbf{0.337 \pm 0.038}$
+Ours (DMA)	0.334 ± 0.025	0.317 ± 0.028	0.278 ± 0.030
FactorVAE	$\mathbf{0.205 \pm 0.022}$	$\mathbf{0.239 \pm 0.017}$	0.171 ± 0.005
+PCA	0.179 ± 0.010	0.234 ± 0.012	0.171 ± 0.006
+ICA	0.066 ± 0.009	0.090 ± 0.006	0.073 ± 0.011
+Ours (IMA)	0.201 ± 0.019	0.226 ± 0.010	$\mathbf{0.191 \pm 0.011}$
+Ours (DMA)	0.184 ± 0.013	0.218 ± 0.016	0.180 ± 0.013
BetaTCVAE	$\mathbf{0.383 \pm 0.022}$	$\mathbf{0.359 \pm 0.026}$	$\mathbf{0.309 \pm 0.036}$
+PCA	0.356 ± 0.022	0.328 ± 0.017	0.295 ± 0.038
+ICA	0.245 ± 0.041	0.260 ± 0.024	0.170 ± 0.045
+Ours (IMA)	0.323 ± 0.025	0.316 ± 0.029	0.271 ± 0.033
+Ours (DMA)	0.327 ± 0.027	0.325 ± 0.025	0.272 ± 0.033
DipVAE	0.235 ± 0.019	0.181 ± 0.049	0.232 ± 0.040
+PCA	0.090 ± 0.005	0.088 ± 0.028	0.091 ± 0.011
+ICA	0.234 ± 0.019	0.180 ± 0.048	0.232 ± 0.041
+Ours (IMA)	0.230 ± 0.022	0.182 ± 0.048	0.230 ± 0.042
Ours (DMA)	$\mathbf{0.249 \pm 0.026}$	$\mathbf{0.188 \pm 0.049}$	$\mathbf{0.253 \pm 0.051}$

Table 5: MPI-3D dataset: Mean $\pm$ std. err. of the DCI scores (across all components of the dataset) of several models and post-hoc methods applied to their embeddings. Columns show which pair of components was correlated during training.

For the optimization of the post-hoc disentanglement problem, we use slightly different hyperparameters. We use the RMSProp optimizer with learning rate of 10^{-3} and a batch size of 48.

C.8 DETAILS ON THE INTRODUCTORY EXAMPLE

The introductory example is inspired by a real explanation generated for a missclassification of the ResNet50 model pretrained on the ImageNet [Russakovsky et al., 2015] dataset delivered with the popular `pytorch` [Paszke et al., 2017] package. Using the approach devised by Leemann et al. [2022], we use the individual neurons of the classifier's last-layer as concepts and describe them by words. We obtain the conceptual explanation shown in Figure 10. We simplify the explanation for the motivational figure and give the concepts relatable names. However, the gist of the example stays the same.

D ADDITIONAL RESULTS

D.1 RECONSTRUCTION QUALITY

As a check, we investigate the reconstruction quality of the disentanglement models. For the 3D shapes, the reconstruction is very high, but we observe some more serious reconstruction errors on the MPI-3d dataset (see Appendix D.2). Figures 11 and 12 show the original images on the left and the reconstructions of a randomly chosen BetaVAE on the right. On Shapes3D, the BetaVAE is able to reconstruct the image from its embedding representation. On MPI3D-real, it is able to reconstruct the big image parts shared across many pictures (ground, background stripe and background), but becomes blurry in the smaller and more nuanced robot arm and object shapes. This indicates that the information on these components might not be stored

Prediction: **jack-o-lantern**,
True class: **traffic light**

No.	Top Concepts (**jack-o-lantern**)	Weight	Act.
2	candles, burning, flame, fire, fires (rating: 2.28)	0.24	1.92
4	batting helmet, patella, spinal column, hindquarter, passenger car (rating: 2.09)	0.14	1.68
8	birds, flamingo, new world blackbird, duck, october (rating: 2.04)	0.22	0.74

No.	Top Concepts (**traffic light**)	Weight	Act.
1	stoplight, traffic light, traffic, boulevard, transportation system (rating: 2.04)	0.28	1.04
2	candles, burning, flame, fire, fires (rating: 2.28)	0.15	1.92
4	towers, buildings, condos, lamppost, building (rating: 1.93)	0.19	1.18

candles, burning, flame, fire, fires (rating: 2.28)

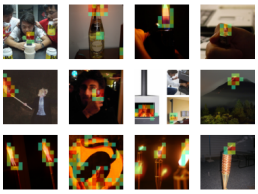


Figure 10: Original local conceptual explanation of the missclassification. We find that the most activating concept “candles, burning, flame...” activates for very dark images. This concept is also highly activated for the traffic light example. We cleared up the description of the concepts for the motivational figure.

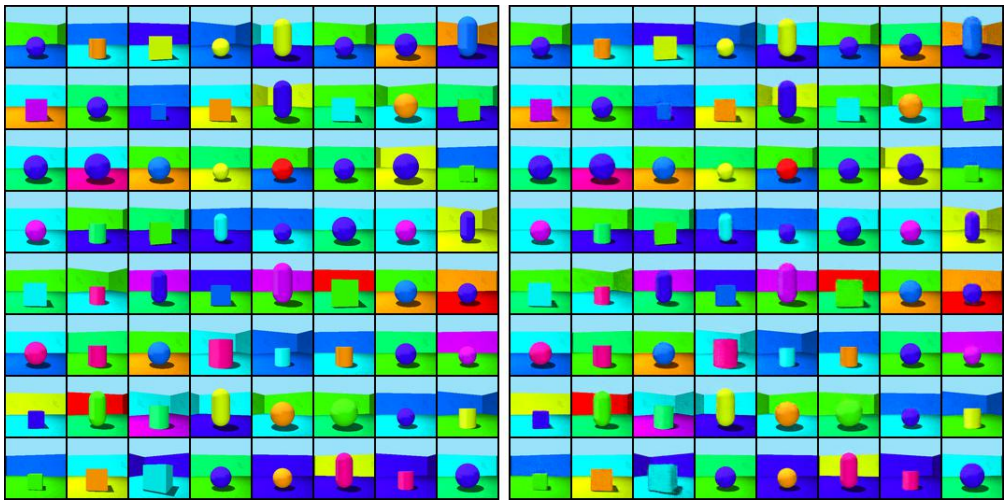


Figure 11: Random example images (left) and their reconstructions (right) of a BetaVAE on Shapes3D.

in the embedding space and is thus hardly disentanglable. A longer training (800k instead of 300k iterations) did not resolve the issue. The issue might arise, following Gondal et al. [2019], because the input images were scaled down to 64x64 pixels making the detailed objects hard to perceive, and because the same architecture as in the Shapes3D experiments was used, which might not be expressive enough.

D.2 RESULTS FOR THE MPI-3D DATASET

In addition to 3Dshapes, we use the challenging MPI3D-real dataset [Gondal et al., 2019], which consists of realistic images of a moving robot arm. It is by far more challenging, as the component is only present in a small portion of the images, and the data consists of real photographs. We report the results on this dataset in Table 5. We saw low disentanglement scores of both the base and post-hoc models on MPI3D-real compared to the performance on Shapes3D. This implies that the embedding spaces of the VAEs was not trained well. In fact, this is supported by the reconstruction quality considerations on both Shapes3D and MPI3D-real. Because our approaches are based on the given embeddings, they also struggle when they incorrectly reflect the sample.

D.3 CORRELATION STRENGTHS AND ATTRIBUTION METHODS IN FIRST EXPERIMENT

In this section we provide additional ablations for the rectification experiment in Sec. 4.2. We investigate the impact of the choice of attribution method (Appendix B.9) and the correlation strength s. The values (DCI scores) are shown in Tab. 8. As expected, our approach offers the highest gains over the baseline when the correlation is higher. Starting at $s = 0.4$, our runs start to reliably outperform the baselines. Regarding the attributions, there is no clear picture, but Grad and SG seem to yield good results more stably across runs. DMA usually outperforms IMA, which supports our theoretical results on

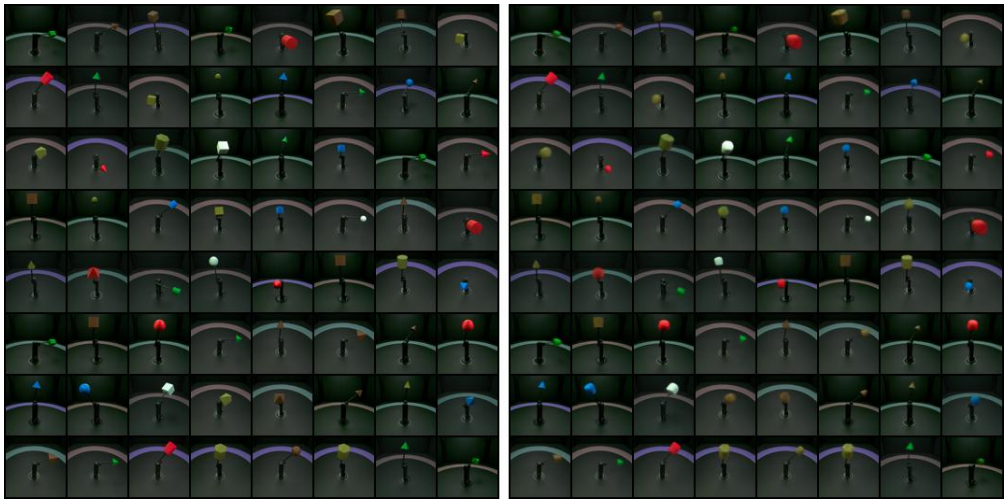


Figure 12: Random example images (left) and their reconstructions (right) of a BetaVAE on MPI3D-real.

identifiability.

D.4 FURTHER DISENTANGLEMENT METRICS

Tables 9 – 11 show the results of the experiment in Sec. 4.2 measured in the alternative metrics MIG, FactorVAE and SAP score. For MIG, we see similar results as for DCI in Table 2 and in Table 5. The results in FactorVAE and SAP score are slightly inferior but our approach still improves over the baseline in many setups. We also compute the disentanglement only on the two correlated components for the first pair of factors in Table 7. This emphasized the improvement introduced by our IMA and DMA approaches.

D.5 RUNTIMES AND FURTHER ABLATION STUDIES

Runtime. Runtime can be an important concern for algorithms in explainable AI, for instance when they are to be deployed on embedded devices. We therefore report the runtimes required to obtain the results shown in Table 2 here:

Algorithm	PCA	ICA	Ours-DMA	Ours-IMA
Runtime (sec)	316 ± 38	320 ± 44	1140 ± 97	1017 ± 121

For our SGD-based optimization, we note that the user can choose how many optimization steps are executed. In the present work, we chose 20000 steps to make sure that the optimization has converged. Using these settings, the runtime of our algorithms is approximately 3 times as high as that of the baseline. We think that this is not prohibitively more expensive. However, convergence of the optimization is usually achieved much quicker.

Effect of less SGD iterations. To ablate the behavior of our approach with a smaller runtime budget, we rerun all the approaches in Table 2 using only 8000 iterations, making the runtime approximately equal across methods. We report the DCI scores as in the original table in Table 6 and see that our DMA approach still outperforms all the baselines in 10 of 12 settings. Thus, even when runtime is an important concern in the evaluation, our approach can still yield competitive results.

Robustness with respect to noise. While IMA covers a more general class of functions, we empirically observed superior performance for DMA in most experiments. We therefore hypothesize that the performance difference stems from the behavior of IMA and DMA under noisy gradients and from the approximate optimizers that we use. We conduct an ablation study to obtain further evidence for these hypotheses. We modify the `FourBars` dataset to fulfill NEMR by adding varying magnitudes of the component gradients in the rows of $\boldsymbol{J}_f(\boldsymbol{g}(\boldsymbol{x}))$. This dataset is now solvable by both IMA and DMA. We then add noise to the analytical gradients. We perform a fixed number of 500 SGD steps of Algorithm 2 and otherwise use the same optimizer parameters as in the main paper. We obtain the DCI curves across different noise levels shown in Figure 13. Without noise, both algorithms find disentangled solutions with DCI scores >0.9 (practically perfect disentanglement when

Correlated components	floor & background		orientation & background		orientation & size	
BetaVAE	0.497 ± 0.03		0.581 ± 0.04		0.491 ± 0.05	
+PCA	0.263 ± 0.03	-47%	0.310 ± 0.02	-47%	0.324 ± 0.04	-34%
+ICA	$\mathbf{0.574 \pm 0.04}$	+16%	0.540 ± 0.08	-7%	0.577 ± 0.04	+17%
+Ours (OA)	0.533 ± 0.11	+7%	0.594 ± 0.04	+2%	0.576 ± 0.03	+17%
+Ours (DA)	0.472 ± 0.14	-5%	$\mathbf{0.633 \pm 0.05}$	+9%	$\mathbf{0.617 \pm 0.03}$	+26%
FactorVAE	0.507 ± 0.11		0.502 ± 0.08		$\mathbf{0.712 \pm 0.01}$	
+PCA	0.358 ± 0.07	-29%	0.474 ± 0.05	-5%	0.556 ± 0.03	-22%
+ICA	0.294 ± 0.07	-42%	0.263 ± 0.05	-48%	0.340 ± 0.03	-52%
+Ours (OA)	0.539 ± 0.04	+6%	0.498 ± 0.03	-1%	0.568 ± 0.06	-20%
+Ours (DA)	$\mathbf{0.567 \pm 0.07}$	+12%	$\mathbf{0.531 \pm 0.04}$	+6%	0.571 ± 0.02	-20%
BetaTCVAE	0.619 ± 0.01		0.613 ± 0.04		0.659 ± 0.01	
+PCA	0.400 ± 0.03	-35%	0.421 ± 0.07	-31%	0.450 ± 0.07	-32%
+ICA	0.540 ± 0.02	-13%	0.497 ± 0.04	-19%	0.627 ± 0.02	-5%
+Ours (OA)	0.635 ± 0.04	+3%	0.648 ± 0.03	+6%	0.682 ± 0.02	+4%
+Ours (DA)	$\mathbf{0.644 \pm 0.01}$	+4%	$\mathbf{0.659 \pm 0.02}$	+8%	$\mathbf{0.724 \pm 0.02}$	+10%
DipVAE	0.631 ± 0.02		0.652 ± 0.02		0.548 ± 0.04	
+PCA	0.158 ± 0.01	-75%	0.160 ± 0.02	-75%	0.170 ± 0.02	-69%
+ICA	0.630 ± 0.02	-0%	0.651 ± 0.02	-0%	0.542 ± 0.03	-1%
+Ours (OA)	0.640 ± 0.01	+1%	0.621 ± 0.02	-5%	0.545 ± 0.05	-1%
+Ours (DA)	$\mathbf{0.683 \pm 0.01}$	+8%	$\mathbf{0.676 \pm 0.01}$	+4%	$\mathbf{0.591 \pm 0.06}$	+8%

Table 6: Using 8000 instead of 20000 SGD iterations: Mean $\pm$ std. err. of the DCI scores of post-hoc methods applied to the embedding spaces of four disentanglement architectures with different pairs of correlated variables. Our DMA method still yields competitive results even with fewer SGD steps.

Dataset	Shapes3D
Correlated factors	floor vs. background
BetaVAE	0.579 ± 0.089
+PCA	0.291 ± 0.033
+ICA	0.435 ± 0.076
+IMA-SGD	$\underline{0.738 \pm 0.072}$
+DMA-SGD	$\mathbf{0.868 \pm 0.025}$
FactorVAE	0.684 ± 0.163
+PCA	0.526 ± 0.136
+ICA	0.363 ± 0.097
+IMA-SGD	$\underline{0.779 \pm 0.063}$
+DMA-SGD	$\mathbf{0.847 \pm 0.072}$
BetaTCVAE	0.589 ± 0.005
+PCA	0.388 ± 0.046
+ICA	0.609 ± 0.065
+IMA-SGD	$\mathbf{0.876 \pm 0.027}$
+DMA-SGD	$\underline{0.754 \pm 0.127}$
DipVAE	0.615 ± 0.114
+PCA	0.429 ± 0.169
+ICA	0.585 ± 0.024
+IMA-SGD	$\mathbf{0.798 \pm 0.099}$
+DMA-SGD	$\underline{0.782 \pm 0.009}$

Table 7: Mean $\pm$ std. err. of the DCI scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The DCI is computed across **the two correlated components** of the dataset.

evaluated on traversals). When we add noise, the disentanglement scores decrease as the working assumptions now only hold approximately. At a noise level of 0.1, the actual gradients shown in Figure 3a are hard to see already with bare eyes. At each point there is a small but consistent gap between the performance of IMA and DMA, indicating that the DMA objective often finds better solutions with the standard SGD optimizer pipeline. This matches our empirical findings of the real data experiments.

D.6 QUALITATIVE RESULTS ON SHAPES3D

In this section, we want to show another traversal plot like the one in Fig. 4 and more thoroughly analyze its latent space. We chose another architecture (BetaTCVAE) and $s = 0.2$ with the usual correlated factors *floor color* and *background color*. Out of the 5 independent runs, we selected the one with the highest DCI score (of the base model) for the analysis.

Linear entanglement matrix. To study which factors are encoded in which latent dimension, we compute a matrix of linear entanglement. By our linear entanglement hypothesis, $\boldsymbol{z}' = \boldsymbol{D}\boldsymbol{z}$, where the matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_K] \in \mathbb{R}^{K \times K}$ contains the directions $\boldsymbol{d}_i \in \mathbb{R}^K$, in which the ground truth concepts are encoded. Changing the component i (entry $\boldsymbol{z}_i$) by one unit will change the resulting embedding by $\boldsymbol{d}_i$. To find these $\boldsymbol{d}_i$, we take the factors at the origin of the traversal plot and alter only a single component i. We then encode the image corresponding to that change, and measure the change in embeddings to find the linear direction $\boldsymbol{d}_i$ that the corresponding component is encoded in (to be precise, we sample several changes and take the largest eigenvector of the embedding changes covariance). Thus, we can estimate the matrix $\boldsymbol{D}$. An example is shown in Fig. 14a and provides evidence that linear entanglement is possible when training autoencoder models from correlated data.

To estimate which factors are changing when a unit direction of the (plain or post-processed) embedding space is followed (a change in $\boldsymbol{z}_i'$), we can invert the equation to $\boldsymbol{z} = \boldsymbol{D}^{-1}\boldsymbol{z}'$. The columns in $\boldsymbol{D}^{-1}$ correspond to the change in ground truth components that going one unit in the latent space coordinate i will entail. We refer to this matrix $\boldsymbol{D}^{-1}$, that shows which ground truth components will be altered by moving along one latent dimension as *linear entanglement matrix*.

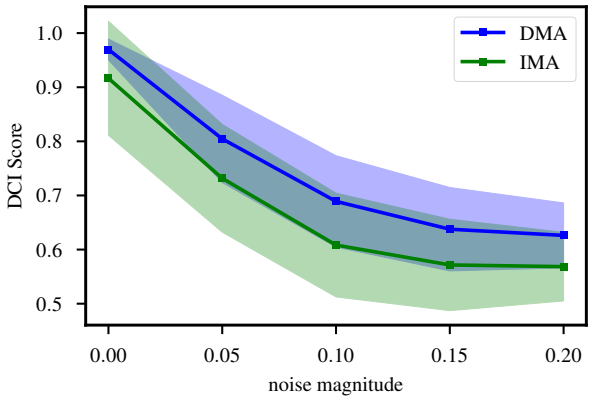


Figure 13: Robustness of optimization to noisy gradients. We use a variant of the `FourBars` dataset that can be identified both by IMA and DMA (the NEMR condition holds) and add noise of increasing magnitude to the analytical gradients. While the disentanglement scores (DCI) decrease for both methods, we observe that the performance of IMA under noise is slightly worse than that of DMA. This may be one factor contributing to the weaker overall performance of IMA as compared to DMA.

Figure 15 shows the traversals along with the corresponding linear entanglement matrices that correspond well to the changes observed. For the plain method, the components that were correlated are deeply entangled (upper line). However, our method (DMA, SG, lower line) is able to separate them well, which is testified both by the traversal and the linear disentanglement matrix.

D.7 FURTHER RESULTS ON CUB

For a quantitative evaluation, we match the discovered concepts on CUB with the annotated ground truth attributes. we report results for the quantitative comparison on CUB introduced in Appendix C.6 of our methods with PCA, ICA, and a baseline of randomly sampled directions. Furthermore, we implement ConceptSHAP [Yeh et al., 2019] and ACE [Ghorbani et al., 2019] and use them to discover concepts on CUB (using their default settings otherwise). The results of this metric are presented in Table 12.

ICA failed to discover meaningful components, while PCA was only capable of discovering very few high-variance ones in the beginning, but begins to fail for $K > 10$. This is possibly because in PCA, the directions are required to be orthogonal. Surprisingly, both PCA and ICA were not much better than the random baseline. Regarding ConceptSHAP and ACE, we find that ACE often focused on the background concepts and ConceptSHAP discovered concepts that are usually more focused on the birds but hard to localize in a fine-grained manner. Our method constantly discovered components and surpassed all three baselines. In particular, our method (DMA) lead to good performance. This leads us to the hypotheses that for high-dimensional data, the disjointness principle is required to identify solutions. Figure 16 illustrates the correlation between the ground-truth attribute representation (scores) and predicted representation by using our model (using plain gradients) for the top discovered component. The two components are clearly correlated, but more in a block-sense: Classes with low scores on the attribute received low scores on the discovered component. The same holds for high scores, but within these, we observe stronger noise, which explains why the Spearman's correlation values were imperfect. This can be due to a certain degree of arbitrage in the ground-truth attribute values of each class. Here, Fig. 17, just like Fig. 6 in the main paper, shows qualitative examples, including the ground-truth values which appear to fluctuate. We emphasize that this analysis should be viewed as an initial take on quantifying the quality of interpretable components, but that a refined benchmark is material for future work.

Model	BetaVAE			FactorVAE			BetaTCVAE			DIPVAEI		
Correlation	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$	$s = 0.2$	$s = 0.4$	$s = \infty$
unit dirs.	0.666 ±0.030	0.497 ±0.028	0.650 ±0.049	0.441 ±0.065	0.507 ±0.105	0.651 ±0.087	0.580 ±0.022	0.619 ±0.008	0.504 ±0.056	0.686 ±0.072	0.631 ±0.018	0.868 ±0.052
PCA	0.287 ±0.010	0.263 ±0.028	0.357 ±0.024	0.312 ±0.048	0.358 ±0.075	0.484 ±0.064	0.341 ±0.018	0.400 ±0.030	0.396 ±0.061	0.266 ±0.029	0.158 ±0.013	0.215 ±0.037
ICA	0.394 ±0.099	0.574 ±0.040	0.674 ±0.012	0.193 ±0.052	0.294 ±0.070	0.390 ±0.109	0.516 ±0.019	0.540 ±0.023	**0.642 ±0.007**	0.672 ±0.073	0.630 ±0.018	**0.870 ±0.049**
Grad (IMA)	0.638 ±0.067	0.617 ±0.018	0.556 ±0.109	0.478 ±0.046	0.551 ±0.040	**0.666 ±0.041**	0.548 ±0.035	0.623 ±0.021	0.551 ±0.038	0.705 ±0.062	0.644 ±0.019	0.794 ±0.043
IG (IMA)	0.702 ±0.035	0.460 ±0.128	0.578 ±0.117	0.470 ±0.035	0.511 ±0.042	0.581 ±0.066	0.619 ±0.024	0.533 ±0.006	0.612 ±0.024	0.650 ±0.072	0.605 ±0.006	0.701 ±0.045
SG (IMA)	0.677 ±0.037	0.438 ±0.127	0.609 ±0.131	0.475 ±0.042	0.561 ±0.040	0.644 ±0.055	0.533 ±0.028	0.620 ±0.021	0.559 ±0.040	0.698 ±0.060	0.642 ±0.017	0.785 ±0.046
Grad (DMA)	0.645 ±0.067	**0.641 ±0.031**	**0.690 ±0.062**	0.547 ±0.056	0.584 ±0.047	0.385 ±0.169	**0.629 ±0.033**	0.666 ±0.010	0.598 ±0.057	**0.717 ±0.059**	**0.684 ±0.009**	0.857 ±0.037
IG (DMA)	0.645 ±0.076	0.530 ±0.106	0.548 ±0.114	**0.573 ±0.046**	**0.615 ±0.045**	0.631 ±0.128	0.607 ±0.028	0.624 ±0.021	0.584 ±0.039	0.703 ±0.073	0.659 ±0.008	0.771 ±0.029
SG (DMA)	**0.711 ±0.040**	0.593 ±0.094	0.633 ±0.062	0.506 ±0.057	0.600 ±0.027	0.644 ±0.066	0.628 ±0.033	**0.670 ±0.014**	0.595 ±0.059	0.716 ±0.059	0.682 ±0.010	0.851 ±0.036

Table 8: Mean ± std. err. of the DCI score of the experiments in Sec. 4.2 for the first correlated component pair (*floor* vs *background* color) in Shapes3D, as an ablation study with further correlations strengths and attribution methods (see Appendix B.9). We observe only small differences between attribution methods, with plain Grad and SG performing best in the DMA setting.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	0.309 ± 0.031	0.426 ± 0.043	0.335 ± 0.059	0.232 ± 0.022	0.185 ± 0.031	$\mathbf{0.196 \pm 0.034}$
+PCA	0.111 ± 0.031	0.101 ± 0.009	0.092 ± 0.031	0.095 ± 0.010	0.105 ± 0.023	0.123 ± 0.033
+ICA	0.360 ± 0.040	0.324 ± 0.054	0.277 ± 0.036	0.155 ± 0.025	0.163 ± 0.014	0.071 ± 0.014
+Ours (IMA)	0.511 ± 0.029	0.437 ± 0.044	0.502 ± 0.030	$\mathbf{0.239 \pm 0.021}$	$\mathbf{0.229 \pm 0.022}$	0.187 ± 0.039
+Ours (DMA)	$\mathbf{0.594 \pm 0.023}$	$\mathbf{0.485 \pm 0.057}$	$\mathbf{0.545 \pm 0.034}$	0.193 ± 0.036	0.092 ± 0.038	0.080 ± 0.015
FactorVAE	0.297 ± 0.084	0.319 ± 0.076	$\mathbf{0.423 \pm 0.018}$	0.079 ± 0.001	0.103 ± 0.020	0.080 ± 0.010
+PCA	0.202 ± 0.057	0.135 ± 0.028	0.235 ± 0.036	$\mathbf{0.111 \pm 0.006}$	$\mathbf{0.122 \pm 0.011}$	$\mathbf{0.107 \pm 0.009}$
+ICA	0.199 ± 0.061	0.106 ± 0.025	0.078 ± 0.021	0.018 ± 0.008	0.061 ± 0.015	0.069 ± 0.015
+Ours (IMA)	$\mathbf{0.337 \pm 0.033}$	$\mathbf{0.322 \pm 0.056}$	0.288 ± 0.092	0.070 ± 0.014	0.086 ± 0.018	0.039 ± 0.014
+Ours (DMA)	0.276 ± 0.036	0.217 ± 0.064	0.213 ± 0.036	0.046 ± 0.021	0.045 ± 0.016	0.048 ± 0.015
BetaTCVAE	0.333 ± 0.008	0.400 ± 0.046	0.402 ± 0.017	$\mathbf{0.279 \pm 0.025}$	$\mathbf{0.223 \pm 0.030}$	0.201 ± 0.039
+PCA	0.249 ± 0.033	0.145 ± 0.039	0.184 ± 0.062	0.265 ± 0.019	0.203 ± 0.028	$\mathbf{0.213 \pm 0.035}$
+ICA	0.390 ± 0.031	0.276 ± 0.043	0.346 ± 0.072	0.199 ± 0.040	0.158 ± 0.038	0.170 ± 0.033
+Ours (IMA)	0.484 ± 0.025	0.490 ± 0.033	0.526 ± 0.036	0.092 ± 0.029	0.071 ± 0.029	0.041 ± 0.014
+Ours (DMA)	$\mathbf{0.525 \pm 0.014}$	$\mathbf{0.540 \pm 0.021}$	$\mathbf{0.620 \pm 0.024}$	0.120 ± 0.037	0.122 ± 0.044	0.075 ± 0.028
DipVAE	0.493 ± 0.032	0.481 ± 0.020	0.433 ± 0.044	0.138 ± 0.020	0.099 ± 0.040	$\mathbf{0.143 \pm 0.045}$
+PCA	0.063 ± 0.006	0.086 ± 0.027	0.108 ± 0.014	0.054 ± 0.016	0.042 ± 0.011	0.064 ± 0.010
+ICA	0.495 ± 0.032	0.438 ± 0.053	0.224 ± 0.026	0.138 ± 0.023	0.096 ± 0.040	0.139 ± 0.047
+Ours (IMA)	0.512 ± 0.042	0.425 ± 0.036	0.465 ± 0.049	$\mathbf{0.146 \pm 0.019}$	$\mathbf{0.105 \pm 0.033}$	0.136 ± 0.049
+Ours (DMA)	$\mathbf{0.591 \pm 0.028}$	$\mathbf{0.546 \pm 0.017}$	$\mathbf{0.497 \pm 0.060}$	0.133 ± 0.029	0.094 ± 0.036	0.125 ± 0.045

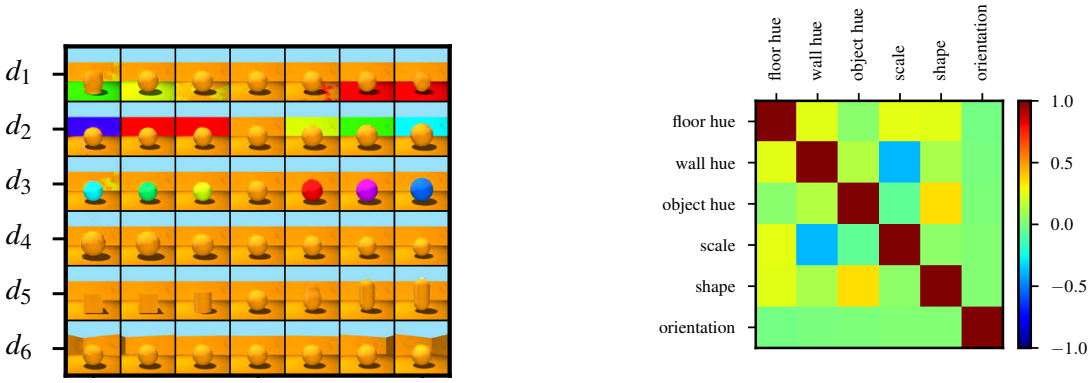
Table 9: Mean ± std. err. of the Mutual-Information Gap (MIG) scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The MIG is computed across all components of the dataset.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	$\mathbf{0.834 \pm 0.022}$	$\mathbf{0.839 \pm 0.053}$	0.828 ± 0.011	0.557 ± 0.032	0.490 ± 0.044	0.412 ± 0.022
+PCA	0.722 ± 0.060	0.689 ± 0.047	0.716 ± 0.035	0.393 ± 0.037	0.452 ± 0.031	0.398 ± 0.031
+ICA	0.797 ± 0.036	0.775 ± 0.083	0.794 ± 0.022	0.385 ± 0.100	0.262 ± 0.061	0.251 ± 0.031
+Ours (IMA)	0.767 ± 0.108	0.808 ± 0.060	$\mathbf{0.832 \pm 0.022}$	0.565 ± 0.022	0.504 ± 0.036	0.443 ± 0.027
+Ours (DMA)	0.813 ± 0.087	0.829 ± 0.068	0.826 ± 0.029	$\mathbf{0.567 \pm 0.024}$	$\mathbf{0.525 \pm 0.042}$	$\mathbf{0.444 \pm 0.027}$
FactorVAE	0.636 ± 0.045	0.622 ± 0.064	0.595 ± 0.050	$\mathbf{0.354 \pm 0.016}$	$\mathbf{0.389 \pm 0.015}$	0.342 ± 0.006
+PCA	0.627 ± 0.071	$\mathbf{0.680 \pm 0.027}$	$\mathbf{0.652 \pm 0.024}$	0.330 ± 0.018	0.388 ± 0.022	$\mathbf{0.353 \pm 0.016}$
+ICA	0.619 ± 0.059	0.446 ± 0.146	0.200 ± 0.148	0.277 ± 0.013	0.242 ± 0.082	0.304 ± 0.017
+Ours (IMA)	$\mathbf{0.663 \pm 0.022}$	0.661 ± 0.028	0.644 ± 0.051	0.347 ± 0.007	0.386 ± 0.020	0.337 ± 0.013
+Ours (DMA)	0.646 ± 0.026	0.637 ± 0.023	0.619 ± 0.026	0.330 ± 0.015	0.375 ± 0.016	0.335 ± 0.013
BetaTCVAE	0.676 ± 0.012	0.814 ± 0.052	0.877 ± 0.015	0.445 ± 0.044	0.379 ± 0.021	0.346 ± 0.020
+PCA	0.761 ± 0.035	0.738 ± 0.063	0.794 ± 0.037	$\mathbf{0.505 \pm 0.040}$	$\mathbf{0.425 \pm 0.012}$	0.389 ± 0.008
+ICA	0.834 ± 0.004	0.761 ± 0.051	0.806 ± 0.051	0.149 ± 0.099	0.168 ± 0.053	0.057 ± 0.035
+Ours (IMA)	0.837 ± 0.004	0.849 ± 0.015	$\mathbf{0.879 \pm 0.013}$	0.463 ± 0.048	0.401 ± 0.018	0.399 ± 0.019
+Ours (DMA)	$\mathbf{0.842 \pm 0.000}$	$\mathbf{0.854 \pm 0.017}$	0.878 ± 0.013	0.460 ± 0.046	0.399 ± 0.018	$\mathbf{0.399 \pm 0.014}$
DipVAE	$\mathbf{0.826 \pm 0.006}$	0.839 ± 0.006	0.785 ± 0.033	$\mathbf{0.517 \pm 0.046}$	$\mathbf{0.473 \pm 0.046}$	0.430 ± 0.013
+PCA	0.671 ± 0.019	0.603 ± 0.064	0.653 ± 0.039	0.431 ± 0.028	0.373 ± 0.027	0.344 ± 0.021
+ICA	0.826 ± 0.006	0.831 ± 0.007	0.749 ± 0.027	0.434 ± 0.042	0.423 ± 0.027	0.424 ± 0.012
+Ours (IMA)	0.824 ± 0.007	0.812 ± 0.018	0.785 ± 0.029	0.503 ± 0.044	0.471 ± 0.035	0.436 ± 0.021
+Ours (DMA)	0.822 ± 0.006	$\mathbf{0.850 \pm 0.012}$	$\mathbf{0.809 \pm 0.045}$	0.505 ± 0.040	0.459 ± 0.040	$\mathbf{0.448 \pm 0.026}$

Table 10: Mean $\pm$ std. err. of the FactorVAE scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The FactorVAE score is computed across all components of the dataset.

Dataset	Shapes3D			MPI3D-real		
Correlated factors	floor vs. background	orientation vs. background	orientation vs. size	background vs. object color	background vs. robot arm dof-1	robot arm dof-1 vs. robot arm dof-2
BetaVAE	0.086 ± 0.003	0.119 ± 0.004	0.100 ± 0.005	0.127 ± 0.014	0.098 ± 0.015	$\mathbf{0.092 \pm 0.025}$
+PCA	0.047 ± 0.005	0.062 ± 0.006	0.066 ± 0.006	0.027 ± 0.005	0.055 ± 0.008	0.037 ± 0.006
+ICA	0.007 ± 0.001	0.013 ± 0.001	0.019 ± 0.004	0.017 ± 0.006	0.007 ± 0.002	0.004 ± 0.001
+Ours (IMA)	$\mathbf{0.099 \pm 0.026}$	0.114 ± 0.008	0.112 ± 0.007	$\mathbf{0.131 \pm 0.011}$	$\mathbf{0.113 \pm 0.005}$	0.082 ± 0.024
+Ours (DMA)	0.094 ± 0.020	$\mathbf{0.127 \pm 0.012}$	$\mathbf{0.114 \pm 0.013}$	0.107 ± 0.025	0.059 ± 0.024	0.037 ± 0.013
FactorVAE	0.072 ± 0.006	0.059 ± 0.006	$\mathbf{0.064 \pm 0.001}$	0.059 ± 0.004	0.066 ± 0.008	0.054 ± 0.003
+PCA	0.060 ± 0.006	$\mathbf{0.066 \pm 0.004}$	0.057 ± 0.004	$\mathbf{0.065 \pm 0.008}$	$\mathbf{0.076 \pm 0.004}$	$\mathbf{0.071 \pm 0.003}$
+ICA	0.013 ± 0.002	0.008 ± 0.001	0.006 ± 0.002	0.002 ± 0.000	0.002 ± 0.001	0.001 ± 0.000
+Ours (IMA)	$\mathbf{0.077 \pm 0.012}$	0.052 ± 0.005	0.054 ± 0.017	0.054 ± 0.006	0.059 ± 0.006	$\mathbf{0.036 \pm 0.015}$
+Ours (DMA)	0.071 ± 0.014	0.053 ± 0.012	0.040 ± 0.010	0.041 ± 0.017	0.043 ± 0.015	0.044 ± 0.013
BetaTCVAE	0.052 ± 0.002	0.107 ± 0.013	0.096 ± 0.016	$\mathbf{0.151 \pm 0.017}$	$\mathbf{0.133 \pm 0.007}$	$\mathbf{0.117 \pm 0.011}$
+PCA	0.073 ± 0.004	0.075 ± 0.011	0.107 ± 0.015	0.148 ± 0.018	0.125 ± 0.009	0.109 ± 0.007
+ICA	0.015 ± 0.000	0.010 ± 0.001	0.011 ± 0.002	0.011 ± 0.004	0.005 ± 0.002	0.004 ± 0.002
+Ours (IMA)	0.105 ± 0.003	0.119 ± 0.012	$\mathbf{0.130 \pm 0.023}$	0.055 ± 0.017	0.059 ± 0.016	0.056 ± 0.003
+Ours (DMA)	$\mathbf{0.108 \pm 0.005}$	$\mathbf{0.127 \pm 0.013}$	0.109 ± 0.017	0.071 ± 0.020	0.072 ± 0.010	0.051 ± 0.015
DipVAE	0.083 ± 0.004	0.084 ± 0.003	0.070 ± 0.002	0.056 ± 0.011	0.039 ± 0.013	0.057 ± 0.016
+PCA	0.027 ± 0.003	0.034 ± 0.006	0.043 ± 0.004	0.023 ± 0.004	0.030 ± 0.008	0.022 ± 0.005
+ICA	0.006 ± 0.001	0.003 ± 0.002	0.030 ± 0.002	0.011 ± 0.005	0.005 ± 0.003	0.005 ± 0.002
+Ours (IMA)	0.089 ± 0.012	0.082 ± 0.005	0.077 ± 0.002	$\mathbf{0.060 \pm 0.008}$	$\mathbf{0.047 \pm 0.010}$	$\mathbf{0.061 \pm 0.016}$
+Ours (DMA)	$\mathbf{0.114 \pm 0.003}$	$\mathbf{0.105 \pm 0.008}$	$\mathbf{0.084 \pm 0.007}$	0.051 ± 0.008	0.043 ± 0.012	0.054 ± 0.016

Table 11: Mean $\pm$ std. err. of the SAP scores of four post-hoc methods applied to the embedding spaces of four disentanglement models on two datasets with different pairs of correlated variables. The SAP score is computed across all components of the dataset.



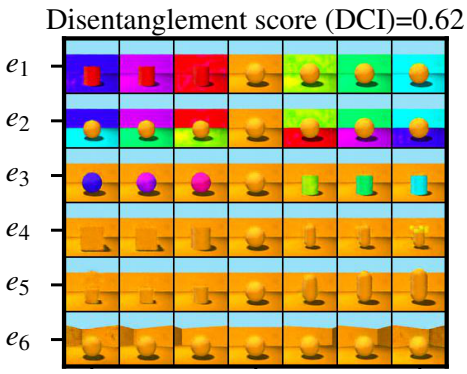
(a) optimal linear directions (traversal plot)

(b) Cosines of the linear directions

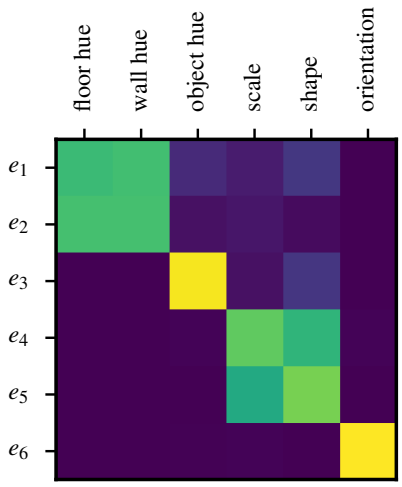
Figure 14: Empirical results for linear entanglement. For the model shown in Fig. 4 (trained on correlated data), we observe almost perfect linear entanglement, i.e., that $f \circ g = D$: (a) There exist linear directions d_1 to d_6 in f's embedding space that encode the individual components. (b) However, these directions are not necessarily orthogonal; they can be entangled as testified by non-zero cosine distances between them. See Fig. 15 for additional results.

Num. components	K=1	K=10	K=20	K=30
PCA	**0.789** $\pm$ 0.024	0.602 $\pm$ 0.007	0.497 $\pm$ 0.005	0.440 $\pm$ 0.006
ICA	0.515 $\pm$ 0.028	0.442 $\pm$ 0.005	0.412 $\pm$ 0.006	0.390 $\pm$ 0.007
ACE [Ghorbani et al., 2019]	0.623 $\pm$ 0.012	0.579 $\pm$ 0.010	0.550 $\pm$ 0.008	0.527 $\pm$ 0.007
ConceptSHAP [Yeh et al., 2019]	0.655 $\pm$ 0.014	0.596 $\pm$ 0.006	0.568 $\pm$ 0.008	0.545 $\pm$ 0.006
Ours-IMA,Grad	0.657 $\pm$ 0.025	0.601 $\pm$ 0.009	0.564 $\pm$ 0.009	0.535 $\pm$ 0.008
Ours-DMA,Grad	0.701 $\pm$ 0.045	**0.626** $\pm$ 0.029	**0.585** $\pm$ 0.028	**0.559** $\pm$ 0.011

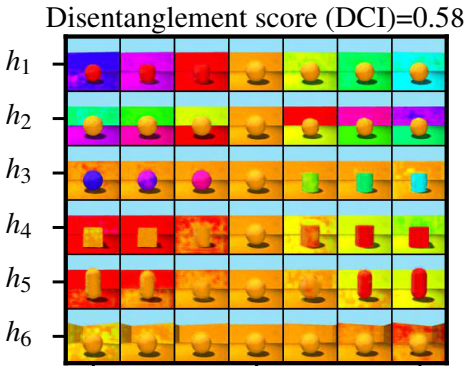
Table 12: Quantitative comparison of discovered components using our methods, PCA, ICA and a random baseline. Mean correlation score of top-K (K in column) discovered components are shown in (mean ± std.) for five runs.



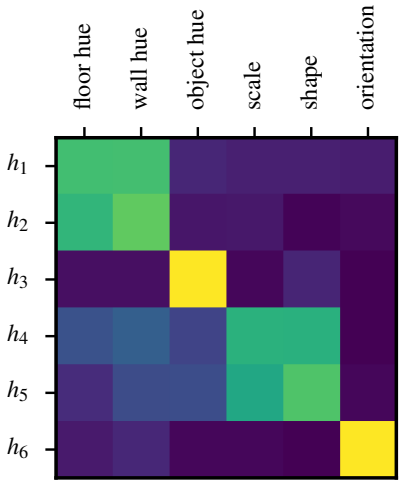
(a) BetaTCVAE



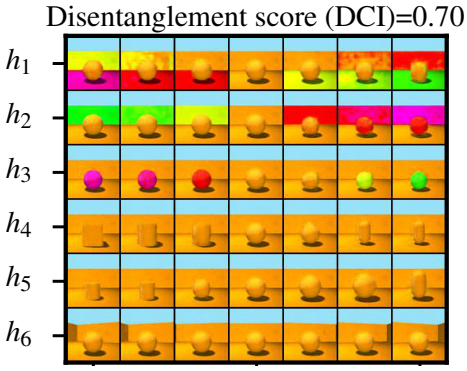
(b) Corresponding linear entanglement matrix



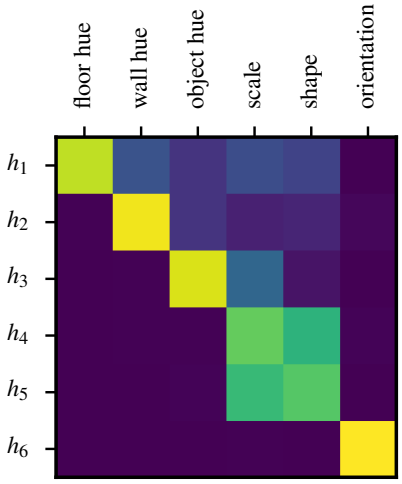
(c) BetaTCVAE + ICA



(d) Corresponding linear entanglement matrix



(e) BetaTCVAE + Ours (DMA, SG)



(f) Corresponding linear entanglement matrix

Figure 15: Traversal plots from another model (BetaTCVAE) trained on the correlated dataset. As for all traversal plots in this paper, we manually permuted the dimensions to match across plots. In addition, we compute a matrix of linear entanglement that shows which ground truth factors is changed when moving into a certain direction (brightness corresponds to maginitude of change). While none of the post-hoc methods manages to disentangle shape and size (most likely due to their non-linear encoding), our model resolves the linearly entangled factors *floor hue* and *wall hue* fairly well, which can also be seen from the entanglement matrix.

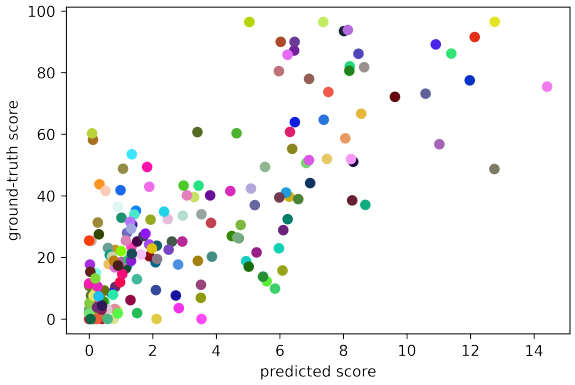


Figure 16: Correlation between ground-truth attribute scores and our predicted scores for the best matched component. Each dot represents a class.

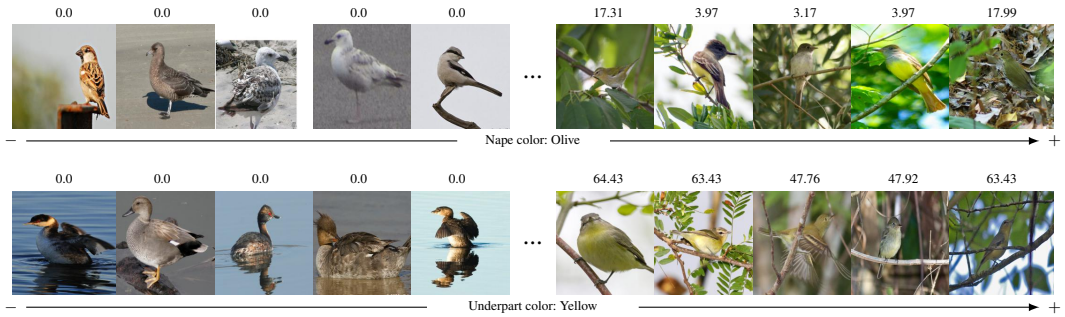


Figure 17: Examples of discovered components on CUB. The corresponding ground-truth attribute is shown under images and the ground-truth value of each image is depicted above the image. "+/−" indicate the positive/negative direction along the discovered concept.

References

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, volume 32, pages 9277–9286, 2019.

Muhammad Waleed Gondal, Manuel Wuthrich, 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. *Advances in Neural Information Processing Systems*, 32, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

Tobias Leemann, Yao Rong, Stefan Kraft, Enkelejda Kasneci, and Gjergji Kasneci. Coherence evaluation of visual concepts with objects and language. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Aditya Ramesh, Youngduck Choi, and Yann LeCun. A spectral regularizer for unsupervised disentanglement. *arXiv preprint arXiv:1812.01161*, 2018.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Anna Sepliarskaia, Julia Kiseleva, Maarten de Rijke, et al. Evaluating disentangled representations. *arXiv preprint arXiv:1910.05587*, 2019.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2046–2059, 2021. URL https://proceedings.neurips.cc/paper/2021/file/0fe6a94848e5c68a54010b61b3e94b0e-Paper.pdf.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, ICML*, 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR, 2021.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.

Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6721–6730, 2021.

Chih-Kuan Yeh, Been Kim, Sercan O Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.