# Interactive Disentanglement: Learning Concepts by Interacting with their Prototype Representations

# Supplementary Materials

## 1 Hyperparameters and model details

### 1.1 Interactive Concept Swapping Networks

In our experiments, the prototype slots were initialized randomly from a truncated Gaussian distribution with mean $\mu = 0$, variance $\sigma^2 = 0.5$, minimum $a = -1$, and maximum $b = 1$. The encoder used in our experiments was a convolutional neural network with residual connections and ReLU activations. Each read-out encoder is a linear layer with LeakyReLU activations. Lastly, the decoder architecture was again a neural network with transposed convolutions and also here residual layers.
In the standard ECR experiments with iCSNs, $J = 3$, $Z = 512$, $Q = 128$, $K = 6$ for each $j \in [1, ..., J]$, $N = 128$ and $\tau$ was decreased every 1000 epochs with steps $[2., 0.5, 0.1, 0.01, 0.001, 0.0001, 0.00001]$ over 8000 epochs in total.
Notably, group normalization as proposed by Wu *et al.* [2] was applied after extracting the concept encodings via the read-out encoders (performed in `collectedReadOutEncoders` in Alg. 1).
Pseudo code can be found in Alg. 1 and Alg. 2.

### 1.2 Baseline models

For the experiments with Cat-VAE, the softmax temperature was set to $\tau = 0.1$ and each categorical distribution had $k = 6$ categories. The number of latent variables was set to 3 for Cat-VAE, $\beta$-VAE, Ada-VAE, and VAE runs. For all baselines, the encoder and decoder consisted of a convolutional and transposed convolutional network. In all experiments, $\beta = 4.$, except for Ada-VAE, where $\beta = 1.$, as recommended by Locatello *et al.* [1]. All baseline models were trained for 2000 epochs.

### 1.3 Linear probing

The linear models for probing the latent representations of the different model configurations were a decision tree and logistic regression model. The max depth of the decision tree was set to 8. The logistic regression model was run with parameters $C = 0.316$ and the maximum number of iterations at 1000. Both the decision tree and logistic regression model were trained with a fixed random seed.

## 2 Details on simulated interactions

The simulated user interactions were performed via an $L_2$ regulatory loss term on the latent codes $y$.
In case a user tells an iCSN *not* to use a specific prototype slot of the superordinate concept $j$ and slot $k$, this loss corresponds to: $MSE(y_{j \cdot k+k}, \mathbf{0})$, where $y_{j \cdot k+k}$ corresponds to the value of $y$ at position $j \cdot k + k$ and $\mathbf{0}$ being a vector of length $N$.
When a user provides a subset of examples with corresponding desired prototype slot IDs the loss term corresponds to: $MSE(y^{subset}_{j \cdot k+k}, \mathbf{1})$ with $\mathbf{1}$ of length equal to the number of samples in *subset*. The subset of examples in our simulated interactions were identified via the ground truth labels, e.g., for identifying the subset of images containing a pentagon. The interactions for learning a novel basic concept followed the same procedure.
To allow the model to update is latent space via interactions we increased $\tau = 0.00001$ back to $\tau = 0.0001$.

## References


[1] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning (ICML)*, 2020.

[2] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision (ECCV)*, 2018.

**Algorithm 1:** Interactive Concept Swapping Network – pair images forward pass

**Input :** Image pair $x \in \mathbb{R}^D$, $x' \in \mathbb{R}^D$, known share IDs v.
**Output:** Image reconstructions $\hat{x} \in \mathbb{R}^D$, $\hat{x}' \in \mathbb{R}^D$, and latent codes $y \in [0,1]^{J \cdot K}$, $y' \in [0,1]^{J \cdot K}$

// Forward pass through initial encoder. $z \in \mathbb{R}^Z$
$z \leftarrow f(x)$
$z' \leftarrow f(x')$
// Forward pass through $J$ read-out encoders. $\phi \in \mathbb{R}^{J \times Q}$
$\phi \leftarrow$ `collectedReadOutEncoders`$(z)$
$\phi' \leftarrow$ `collectedReadOutEncoders`$(z')$
// Compute the distance of each concept encoding to all prototype slots of its corresponding category $j$.
$y \leftarrow$ `computeProtoDistance`$(\phi, \mathrm{v})$
$y' \leftarrow$ `computeProtoDistance`$(\phi', \mathrm{v})$
// Reconstruct the images from the prototype distance codes.
$\hat{y} \leftarrow g(y)$
$\hat{y}' \leftarrow g(y')$

**Algorithm 2:** computeProtoDistance

**Input :** Concept encodings $\phi \in \mathbb{R}^{J \times Q}$, $\phi' \in \mathbb{R}^{J \times Q}$
**Given :** Set of prototype slot codebooks $\Theta := [\mathrm{P}_1, ..., \mathrm{P}_J] \in \mathbb{R}^{J \times Q \times K}$, softmax temperature $\tau$, and share IDs v.
**Output:** Latent codes $y \in [0,1]^{J \cdot K}$, $y' \in [0,1]^{J \cdot K}$

// For every superordinate concept category
**for** $j \leftarrow 0$ **to** $J - 1$ **do**
  // Dot-product between concept encoding and all prototype slots from codebook $\mathrm{P}_j$.
  $s_j \leftarrow$ `softmaxDotProduct`$(\phi_j, \mathrm{P}_j)$
  $s'_j \leftarrow$ `softmaxDotProduct`$(\phi'_j, \mathrm{P}_j)$
  // Compute normalizing weighted softmax.
  $\Pi_j \leftarrow$ `softmaxNormTau`$(s_j, \tau)$
  $\Pi'_j \leftarrow$ `softmaxNormTau`$(s'_j, \tau)$
**end for**
// Swap the distance codes at the position corresponding to the shared IDs.
$y \leftarrow [\Pi_1, ..., \boldsymbol{\Pi}'_{\mathbf{v}}, ..., \Pi_J]$
$y' \leftarrow [\Pi'_1, ..., \boldsymbol{\Pi}_{\mathbf{v}}, ..., \Pi'_J]$