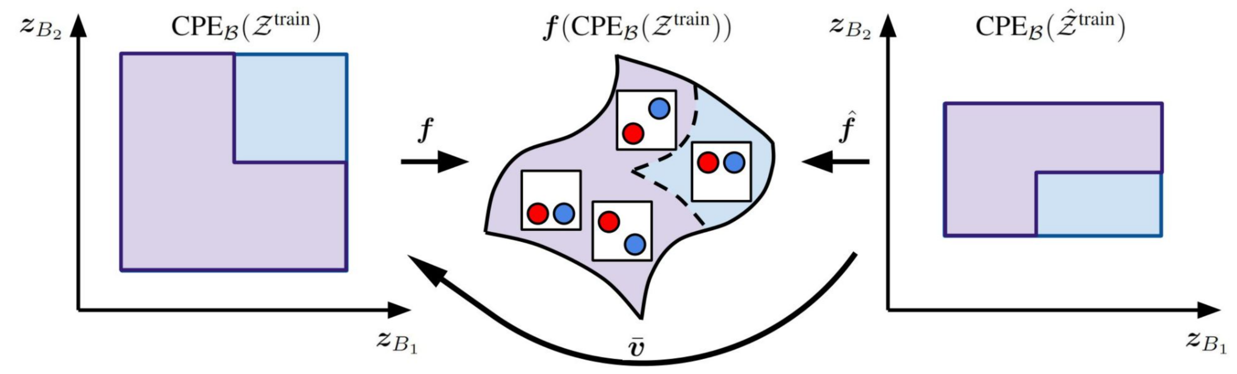
Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation

Sébastien Lachapelle*, Divyat Mahajan*, Ioannis Mitliagkas & Simon Lacoste-Julien

Mila · Université de Montréal · SAMSUNG Samsung Advanced Institute of Technology AI Lab Montreal

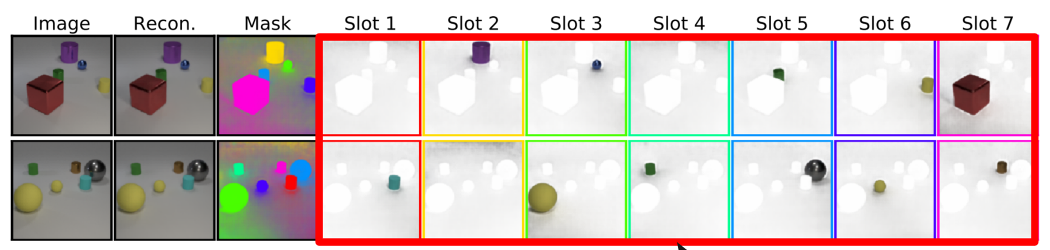
Contributions

We introduce **additive decoders**: a simple architecture **similar to object-centric decoders** for which we can prove both **disentanglement and extrapolation guarantees**.



Motivations

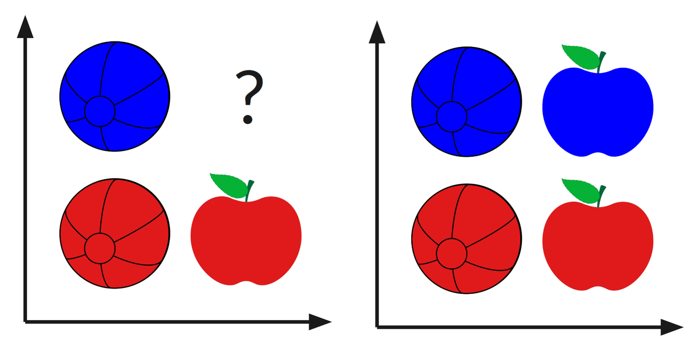
- Explain why **object-centric representation learning (OCRL)** methods (such as Slot Attention [2]) perform **disentanglement** when they are **trained only to reconstruct**, without supervision.



Source: *Object-Centric Learning with Slot Attention* by F. Locatello et al., (2020)

There's no rigorous mathematical explanation for why this is happening...

- Understanding when and why **compositional generalization** is possible.



Unidentifiability in representation learning

- Decoder: $\hat{f}: \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ $(d_x >> d_z)$
- Encoder: $\hat{g}: \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$
- Assume they solve the reconstruction problem, i.e. $\mathbb{E}^{\mathrm{train}}\|x - \hat{f}(\hat{g}(x))\|_2^2 = 0$
- By taking $\tilde{f} := \hat{f} \circ v$ and $\tilde{g} := v^{-1} \circ \hat{g}$ where v is some invertible map, we also solve the reconstruction problem…
- … but $\hat{g}$ and $\tilde{g}$ might have drastically different representations.
- This is a problem if we hope to learn a disentangled representation.
- **Our solution:** restrict $\hat{f}$ to be additive!
- Closely related to nonlinear **independent component analysis (ICA)** [1].

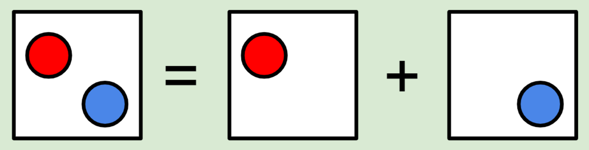
Additive decoders

Additive decoder: $x = f(z) = \sum_{B \in \mathcal{B}} f^{(B)}(z_B)$

Observation e.g. an image · Latent factors · Partition of $\{1, 2, ..., d_z\}$ e.g. $\mathcal{B} = \{\{1,2\},\{3,4\}\}$ · Subblock of z (non-overlapping)

Example: Images of moving balls

$x = f(z)$ $\quad f^{(B_1)}(z_{B_1})$ $\quad f^{(B_2)}(z_{B_2})$



$\mathcal{B} := \{\{1,2\},\{3,4\}\}$

$z_{B_1} = (z_1, z_2)$ (XY-position of red ball)

$z_{B_2} = (z_3, z_4)$ (XY-position of blue ball)

$f^{(B)}$ (Block-specific decoder)

Disentanglement guarantee (identifiability)

Theorem (informal): Assume that

1. Data generating process: $x = \sum_{B \in \mathcal{B}} f^{(B)}(z_B)$ with $z \sim \mathbb{P}_z^{\mathrm{train}}$;
2. Learned decoder is additive as well;
3. Ground-truth decoder is "sufficiently nonlinear" (see paper)
4. More regularity conditions... (like C^2-diffeomorphism)

Then, $\mathbb{E}^{\mathrm{train}}\|x - \hat{f}(\hat{g}(x))\|^2 = 0$ implies that, for all $B \in \mathcal{B}$,

$$\hat{f}^{(B)}(z_B) = f^{(\pi(B))}(v_{\pi(B)}(z_B)) + c^{(B)} \qquad \sum_{B \in \mathcal{B}} c^{(B)} = 0$$

Permutation that sends blocks to blocks, i.e., $\pi(B) \in \mathcal{B}$. · Invertible transformation

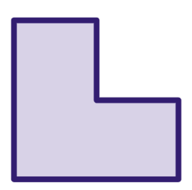
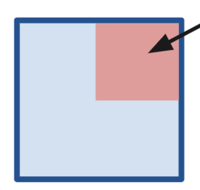
- The learned block-specific decoders imitate the ground-truth ones!
- Latent factors can be dependent and have an almost arbitrary support.

Cartesian-Product Extrapolation

$\hat{\mathcal{Z}}^{\mathrm{train}}$:= support of the learned factors **(seen during training)**

Cartesian-Product Extension:
$\mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\mathrm{train}}) := \prod_{B \in \mathcal{B}} \hat{\mathcal{Z}}_B^{\mathrm{train}}$, where $\hat{\mathcal{Z}}_B^{\mathrm{train}} := \{\hat{z}_B \mid \hat{z} \in \hat{\mathcal{Z}}^{\mathrm{train}}\}$ (projection)

For example:

$\hat{\mathcal{Z}}^{\mathrm{train}} :=$  $\qquad \mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\mathrm{train}}) :=$ 

Corollary (informal): Under the same assumptions as the previous theorem:

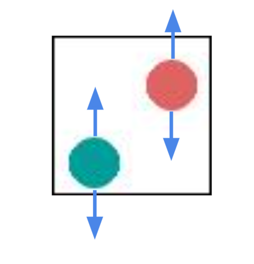
The learned decoder imitates the ground-truth not only over $\hat{\mathcal{Z}}^{\mathrm{train}}$, but over all $\mathrm{CPE}_{\mathcal{B}}(\hat{\mathcal{Z}}^{\mathrm{train}})$.

Experiments: Datasets

Scalar Latent dataset: Balls move only along y-axis

$\dim(z) = 2$

$\mathcal{B} = \{\{1\},\{2\}\}$



Evaluation Metric:

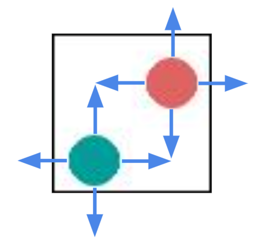
$\mathrm{LMS} = \arg\max_{\pi \in \mathfrak{S}_{\mathcal{B}}} \frac{1}{\ell} \sum_{B \in \mathcal{B}} s_{B,\pi(\tilde{B})}$

where $s_{B,\tilde{B}}$ measures how well we can predict one block from the other

Block Latent dataset: Balls move only along both x, y axis

$\dim(z) = 4$

$\mathcal{B} = \{\{1,2\},\{3,4\}\}$

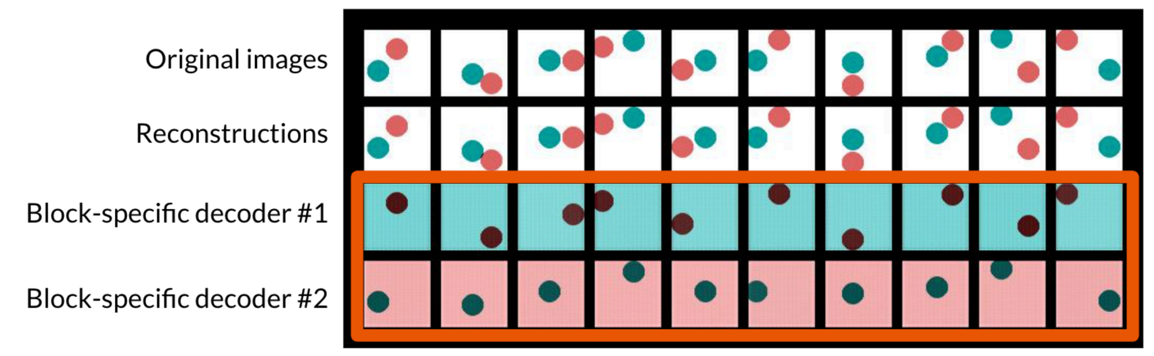


- *Independent Case:* $z_{B_1} \perp z_{B_2}$
- *Dependent Case:* $z_{B_1} \not\perp z_{B_2}$

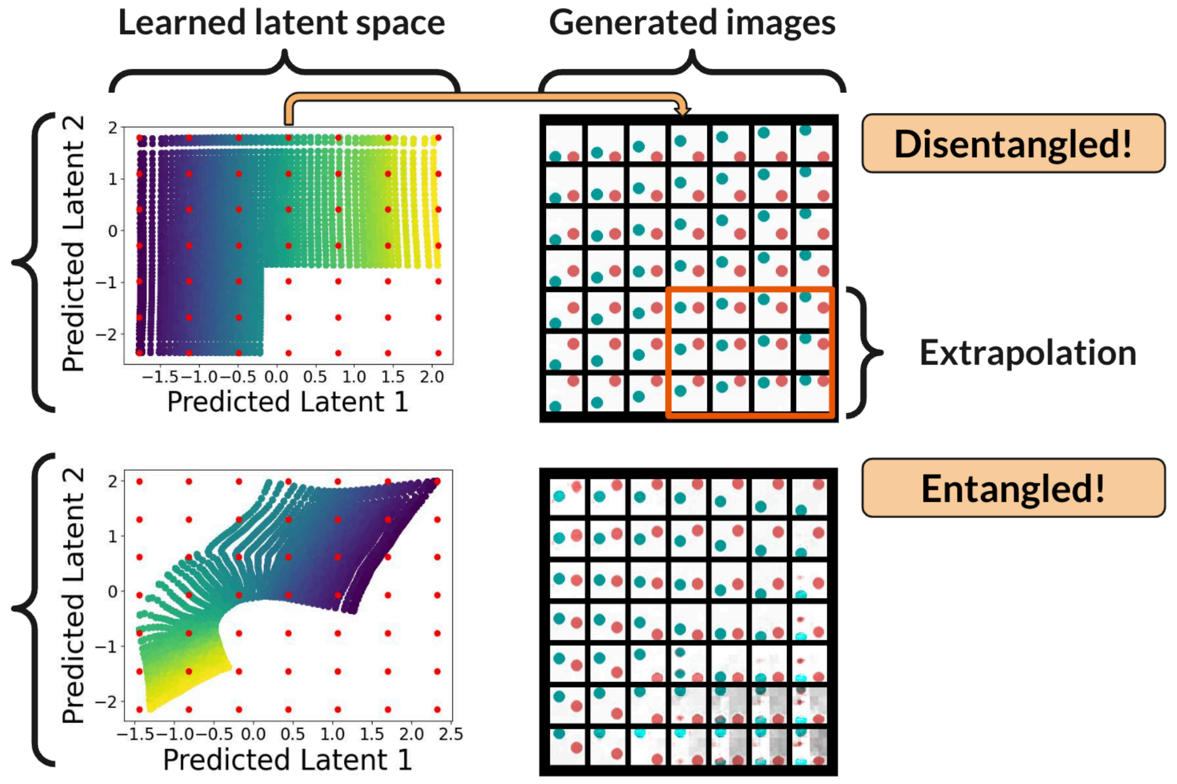
Experiments: Disentanglement

	Scalar Latent	Block Latent Independent	Block Latent Dependent
Non-Additive Decoder	70.6 (5.2)	53.9 (7.6)	78.1 (2.9)
Additive Decoder	**91.5 (3.6)**	**92.2 (4.9)**	**99.9 (0.0)**

Modified MCC score (Higher = more disentangled)



Experiments: Extrapolation



[1] A. Hyvärinen, H. Sasaki, and R. E. Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In AISTATS. PMLR, 2019.

[2] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In Advances in Neural Information Processing Systems, 2020.