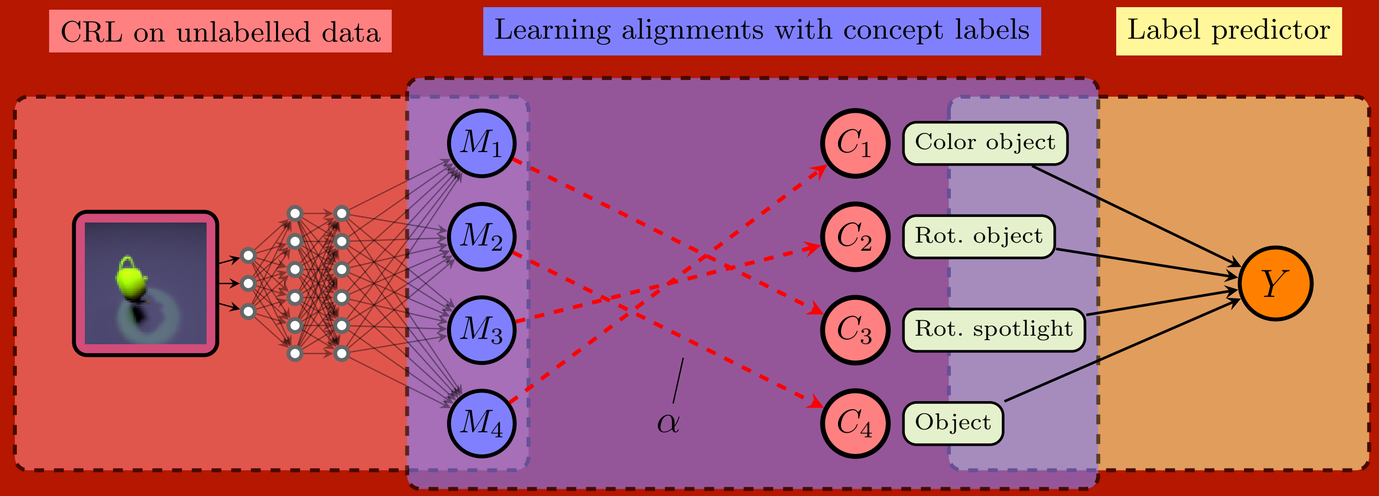
Sample-efficient Learning of Concepts with Theoretical Guarantees:

from Data to Concepts without Interventions



Joint Work

- This presentation is based on work done in collaboration with:

Dr. Sara Magliacane
University of Amsterdam

Dr. Tim van Erven
University of Amsterdam

Outline

- Concept-based Models
- Causal Representation Learning
- Combining Concepts and Causal Variables
- Alignment learning
- Experimental Results

Concept-Based Models

Need for high-level concepts

- Attribution on in input level often too *low level*:
  - Individual pixels don't tell us enough
  - Even groups of pixels might be uninformative
- What is needed are higher level features:
  - Which we call concepts

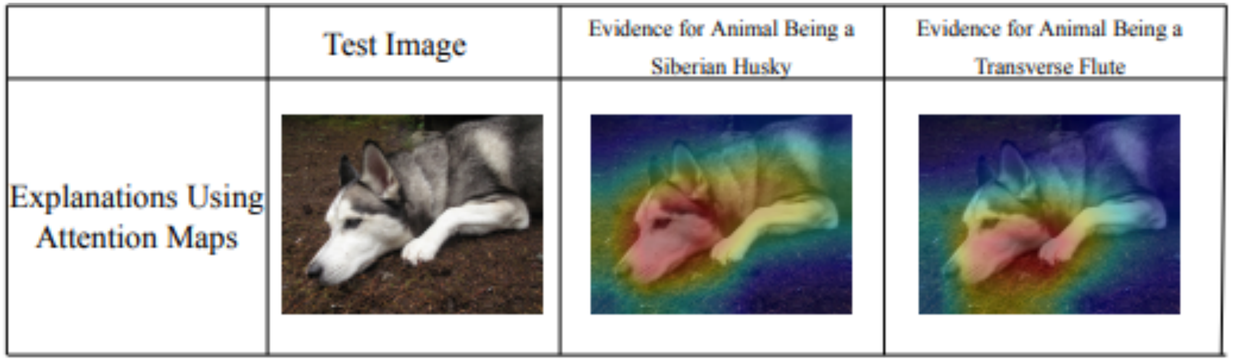


Figure from [Rudin, 2019]

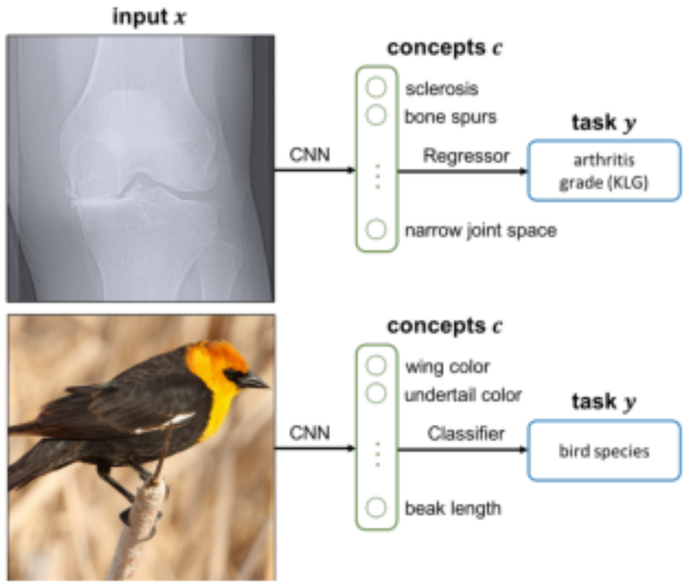
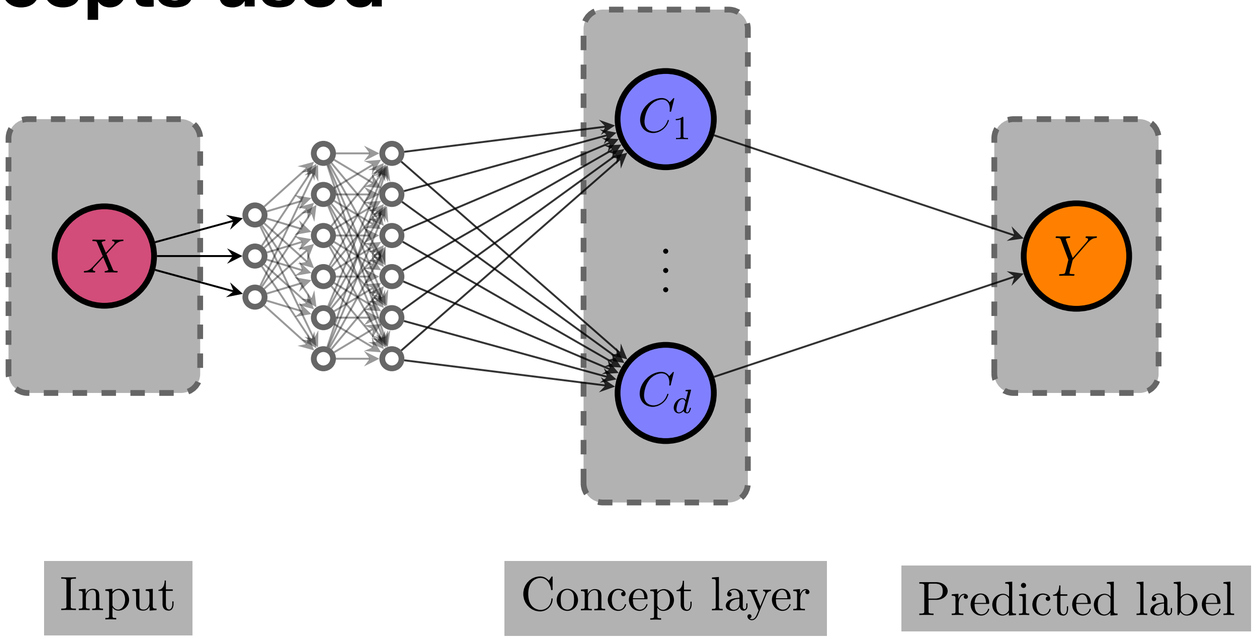


Figure from [Koh et al. 2020]

XAI and the need for high-level concepts

How are concepts used



- An explanation is given as:

$$\{(h(x)_c, \psi_{cj}) \mid c \in \{1, \ldots, d\}\}$$

  - c is the concept index
  - ψ linear activations
  - $h(x)_c$ represents the strengt of concept c in the input

Concept-Based

Examples

Concept-Based Models

- Concept Bottleneck Models, [Koh et al., 2018]
- Concept Embedding Models [Zarlenga et al., 2022]
- GLANCE-Nets [Marconato et al, 2022]

Concept-Based Explanations

- (Non)-linear probing [Alain & Bengio, 2016]
- TCAV [Kim et al, 2018]
- Sparse Auto-Encoders [Sharkey & Milidge, 2022]
- Mechanistic interpretability

Causal Representation Learning

Causal Representation Learning

Brief introduction

It is often not enough to learn associations:

- Naive training of NNs known to result in learning Shortcuts
- Are a result of Spurious Correlations

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

Figure from [Beery et al, 2018]

Causal Representation Learning

Brief introduction

To overcome this, we can assume a causal setting:

- There are 2 variables $G_1 \in \{\text{Cow}, \dots\}, G_2 \in \{\text{Mountains, beach}, \dots\}$
- Nature provides some function $f(G_1, G_2) = X \in \mathbb{R}^{h \times w}$ that generates the image

CRL methods try to learn the reverse process, given a dataset of only images

Causal Representation Learning

Brief introduction

To overcome this, we can assume a causal setting:

- There are 2 variables $G_1 \in \{\text{Cow}, \dots\}, G_2 \in \{\text{Mountains, beach}, \dots\}$
- Nature provides some function $f(G_1, G_2) = X \in \mathbb{R}^{h \times w}$ that generates the image

CRL methods try to learn the reverse process, given a dataset of only images

- Learn an encoder $g_\varphi(X) = M \approx G$
- Typically a Variational Auto Encoder (VAE)

Causal Representation Learning

Caveats

- Many assumptions needed
- Even then, in general this problem is non-identifiable
- It is identifiable up to an equivalence class ($\sim$)
- The learned M are:
  - Permuted,
  - Component-wise transformations of G.

$$PT(M) = \begin{bmatrix} T_{\pi(1)}(M_{\pi(1)}) \\ T_{\pi(1)}(M_{\pi(2)}) \\ \vdots \\ T_{\pi(d)}(M_{\pi(d)}) \end{bmatrix} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_d \end{bmatrix} = G$$

Identifiability

Some definitions

The parameter f is not *identifiable*, but it is identifiable up to an equivalence relation $\sim$

$$P_\theta(X) = P'_\theta(X) \implies \theta \sim \theta'$$

Identifiability Class

The parameters f and g are equivalent according to $\overset{L}{\sim}$ if:

1. If there exists an invertible $A \in \mathbb{R}^{d \times d}$
2. And a coordinate-wise diffeomorphism:

$$T(z) = (T_1(z_1), \ldots, T_d(z_d))^\top$$

Such that,

$$f^{-1}(x) = AT(g^{-1}(x)) \text{ for all } x \in \mathcal{X}$$

If $A = P$ is a (block)-permurtation matrix we denote the relation by $\overset{P}{\sim}$ and we also call θ *disentangled*.

Causal Representation Learning

Example: CITRIS [Lippe et al. 2022]

- Data needed to train:
  - Videos
  - Binary vectors indicating if an intervention was performed
- Interventions are not necessary for inference

(a) Teapot (b) Cow (c) Head (d) Horse (e) Armadillo (f) Dragon (g) Hare

Figure 11. An example image for each object shape in the Temporal-Causal3DIdent dataset.

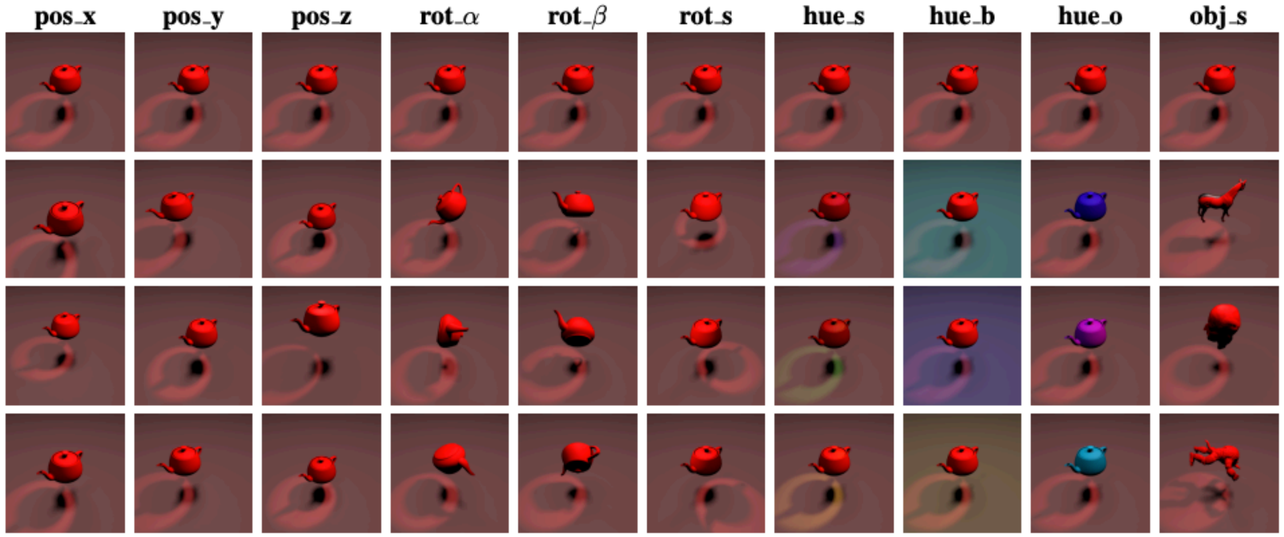


Figure 12. Visualizing the different factors of variation in the Temporal Causal3DIdent dataset. Each column represents one dimension of a causal factor, and the different rows show the original image (first row) with only the corresponding causal factor being changed.

Causal Representation Learning

Example: CITRIS [Lippe et al. 2022]

- Data needed to train:
  - Videos
  - Binary vectors indicating if an intervention was performed
- Interventions are not necessary for inference

(a) Teapot (b) Cow (c) Head (d) Horse (e) Armadillo (f) Dragon (g) Hare

Figure 11. An example image for each object shape in the Temporal-Causal3DIdent dataset.

Table 10. Experimental results for the Temporal-Causal3DIdent 7 shapes dataset, including standard deviations over 3 seeds. See Table 1 for a detailed discussion on the table and metrics.

	Triplet evaluation distances ↓											Correlation metrics			
	pos_x	pos_y	pos_z	rot_α	rot_β	rot_s	hue_s	hue_b	hue_o	obj_s	Mean	R^2 diag ↑	R^2 sep ↓	Spearman diag ↑	Spearman sep ↓
Oracle	0.08	0.06	0.08	0.06	0.09	0.04	0.04	0.01	0.04	0.00	0.05	-	-	-	-
SlowVAE	0.44	0.25	0.41	0.69	0.75	0.25	0.57	0.10	0.14	0.37	0.40	0.61	0.23	0.59	0.27
(stds)	±0.035	±0.030	±0.021	±0.012	±0.003	±0.003	±0.064	±0.001	±0.004	±0.040	±0.018	±0.053	±0.005	±0.038	±0.006
iVAE*	0.26	0.23	0.34	0.58	0.65	0.10	0.31	0.02	0.09	0.14	0.27	0.80	0.29	0.77	0.28
(stds)	±0.013	±0.038	±0.015	±0.051	±0.043	±0.011	±0.003	±0.000	±0.001	±0.039	±0.002	±0.002	±0.013	±0.001	±0.019
CITRIS-VAE	0.15	0.13	0.23	0.54	0.71	0.07	0.05	0.02	0.06	0.18	0.21	0.89	0.10	0.88	0.12
(stds)	±0.002	±0.000	±0.001	±0.009	±0.009	±0.002	±0.002	±0.000	±0.002	±0.050	±0.004	±0.001	±0.007	±0.002	±0.006
CITRIS-NF	**0.12**	**0.08**	**0.11**	**0.09**	**0.14**	**0.05**	**0.05**	**0.02**	**0.06**	**0.00**	**0.07**	**0.98**	**0.04**	**0.97**	**0.08**
(stds)	±0.001	±0.000	±0.001	±0.000	±0.004	±0.001	±0.001	±0.004	±0.002	±0.000	±0.001	±0.000	±0.003	±0.000	±0.007

Combining Concepts and Causal Variables

Combining Concepts and Causal variables

Main idea

We will assume that the causal variable and interpretable concepts are the same!

Problem: The learned causal variables are permuted and transformed

Combining Concepts and Causal variables

Complete setting

- Unobserved causal variables

$$G = (G_1, \ldots, G_d)$$

- Observation X is generated by some function f

$$X = f(G) + \epsilon$$

- Assume the G_i variables correspond to interpretable concepts,

$$G_i = C_i$$

- From CRL we get a g_φ that maps X to learned causal variables

$$M = (M_1, \ldots, M_d)$$

We don't know which C_i corresponds to which M_j, and we don't know what their relation is

We will learn an alignment map,

$$\alpha : M \mapsto C$$

to resolve this issue

Identifiability theory

Combined with concepts

Assumptions

We are given 2 functions, g_ψ and $g_{\psi'}$, learned by the human and the machine respectively with

- g_ψ and $g_{\psi'}$ both diffeomorphisms
- $g_\psi \overset{P}{\sim} f^{-1}$ and $g_{\psi'} \overset{P}{\sim} f^{-1}$

Hence, also $g_\psi \overset{P}{\sim} g_{\psi'}$ and we see that the following relation holds:

$$PT(M) = \begin{bmatrix} T_{\pi(1)}(M_{\pi(1)}) \\ T_{\pi(1)}(M_{\pi(2)}) \\ \vdots \\ T_{\pi(d)}(M_{\pi(d)}) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_d \end{bmatrix} = C$$

With a permutation

$$\pi \colon \{1, \ldots, d\} \to \{1, \ldots, d\}$$

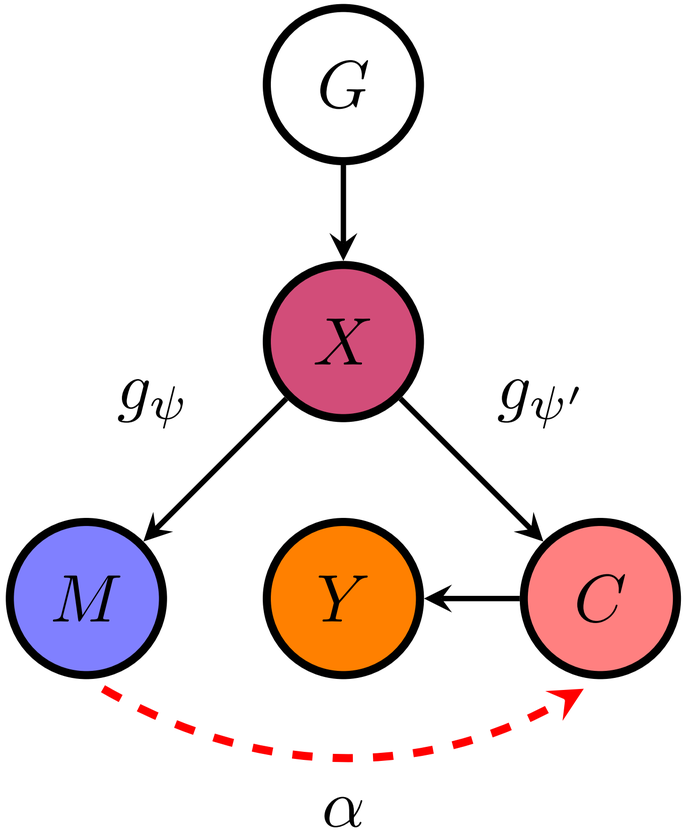
Identifiability theory

Combined with concepts

Learning $\alpha: M \mapsto C$

$$PT(M) = \begin{bmatrix} T_{\pi(1)}(M_{\pi(1)}) \\ T_{\pi(1)}(M_{\pi(2)}) \\ \vdots \\ T_{\pi(d)}(M_{\pi(d)}) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_d \end{bmatrix} = C$$

Involves learning a permutation and a 1-dimensional mapping



Alignment Learning

Alignment Learning

Intuition

For now imagine 2 variables and 2 outcomes:

$$\begin{aligned} Y_1 &= \beta_2 X_2 \\ Y_2 &= \beta_1 X_1 \end{aligned} \quad \rightarrow \quad \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 & \beta_2 \\ \beta_1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Alignment Learning

Intuition

For now imagine 2 variables and 2 outcomes:

$$\begin{aligned} Y_1 &= \beta_2 X_2 \\ Y_2 &= \beta_1 X_1 \end{aligned} \quad \rightarrow \quad \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 & \beta_2 \\ \beta_1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Regressing $X = (X_1, X_2)$ simultaneously on to $Y = (Y_1, Y_2)$ would give

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \widehat{\beta}_1^1 & \widehat{\beta}_2^1 \\ \widehat{\beta}_1^2 & \widehat{\beta}_2^2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \text{ with } \widehat{\beta}_1^1 \approx 0,\ \widehat{\beta}_2^2 \approx 0$$

Alignment Learning

Intuition

For now imagine 2 variables and 2 outcomes:

$$\begin{aligned} Y_1 &= \beta_2 X_2 \\ Y_2 &= \beta_1 X_1 \end{aligned} \quad \rightarrow \quad \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 0 & \beta_2 \\ \beta_1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Regressing $X = (X_1, X_2)$ simultaneously on to $Y = (Y_1, Y_2)$ would give

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} \hat{\beta}_1^1 & \hat{\beta}_2^1 \\ \hat{\beta}_1^2 & \hat{\beta}_2^2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \text{ with } \hat{\beta}_1^1 \approx 0,\ \hat{\beta}_2^2 \approx 0$$

- Using **Lasso Regression** we can force the irrelevant terms to zero
- Build the permutation from non-zero estimated parameters

Alignment Learning

General method

Assume that the mapping T can be described using a feature map $\varphi : \mathbb{R} \to \mathbb{R}^p$:

$$\varphi(M) = [\varphi(M_1)^\top, \ldots, \varphi(M_d)^\top]^\top \in \mathbb{R}^{pd}$$

$$C_i = \varphi(M)\beta_i^\star + \epsilon_i \quad \beta_i^\star \in \mathbb{R}^{pd}$$

$$C_i = T(M) + \epsilon_i$$

The optimal β_i^* will have a ***sparse structure!***

-> The variable C_i only relates to $M_{\pi(i)} = M_j$

-> Only the parameters that affect $\varphi(M_j)$ are non-zero

$$(\beta_i^\star)^{\pi(i)} \neq 0$$

We will employ a ***Group Lasso*** approach for each coordinate!

Alignment Learning

Method

$\Phi \in \mathbb{R}^{n \times pd}$ stacked transformed data matrix

$\mathbf{C}_i \in \mathbb{R}^n$ stacked data of Concept i

Regression problem

The estimator per dimension given by:

$$\widehat{\beta}_i = \underset{\beta \in \mathbb{R}^{dp}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{C}_i - \Phi\beta\|^2 + \lambda \sum_{j=1}^{d} \|\beta^j\|$$

$$= \underset{\beta \in \mathbb{R}^{dp}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{C}_i - \Phi\beta\|^2 + \|\beta\|_{2,1}$$

$$\beta_i^j = ((\beta_i)_k \mid k = (j-1)p, \ldots, jp)$$

Group lasso regularisation ensures that many groups will be 0

Intuitively, largest $\|\widehat{\beta}_i^j\|$ should correspond with the only non-zero group.

Alignment Learning

Method

$\Phi \in \mathbb{R}^{n \times pd}$ stacked transformed data matrix

$\mathbf{C}_i \in \mathbb{R}^n$ stacked data of Concept i

Regression problem

The estimator per dimension given by:

$$\widehat{\beta}_i = \underset{\beta \in \mathbb{R}^{dp}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{C}_i - \Phi\beta\|^2 + \lambda \sum_{j=1}^{d} \|\beta^j\|$$

$$= \underset{\beta \in \mathbb{R}^{dp}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{C}_i - \Phi\beta\|^2 + \|\beta\|_{2,1}$$

In Python has a user-friendly implementation!

`group_lasso.GroupLasso.fit()`

$$\beta_i^j = ((\beta_i)_k \mid k = (j-1)p, \ldots, jp)$$

Group lasso regularisation ensures that many groups will be 0

Intuitively, largest $\|\widehat{\beta}_i^j\|$ should correspond with the only non-zero group.

Alignment Learning

Method

Getting the correct matching

We could build $\widehat{\pi}(i) = \underset{j=1,\ldots,d}{\operatorname{argmax}} \| \widehat{\beta}_i^j \|$

However, this will likely not be a correct permutation. So, we solve an assignment problem:

$$\widehat{\pi} = \underset{\pi \in \Pi}{\operatorname{argmax}} \sum_{i=1}^{d} \sum_{j=1}^{d} \| \widehat{\beta}_i^{\pi(i)} \|$$

Alignment Learning

Method

Getting the correct matching

We could build $\hat{\pi}(i) = \underset{j=1,\ldots,d}{\operatorname{argmax}} \|\hat{\beta}_i^j\|$

However, this will likely not be a correct permutation. So, we solve an assignment problem:

$$\hat{\pi} = \underset{\pi \in \Pi}{\operatorname{argmax}} \sum_{i=1}^{d} \sum_{j=1}^{d} \|\hat{\beta}_i^{\pi(i)}\|$$

This optimisation can be done efficiently and easily!

`scipy.optimize.linear_sum_assignment`

Permutation Learning

Kernelized version

Most of the time we cannot assume that T can be decomposed. What can we do then?

Kernel Methods!

It is possible to extend the Group Lasso to a Kernelized version

Permutation Learning

Kernelized version

The concepts are now modelled by

$$C_i = \beta_i^\star(M) + \epsilon_i,$$

where $\beta_i^\star$ is a function of the form:

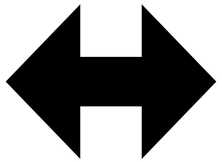
$$\beta^\star(M) = \sum_{j=1}^{d} \beta^j(M_j) \quad \text{only} \quad \beta^{\pi(j)} \neq 0$$

Each j has an associated kernel:

$$\kappa_j(M_j, M_j') = \left\langle \varphi_j(M_j), \varphi_j(M_j') \right\rangle$$

Infinite dimensional viewpoint

$$\min_{\beta^1,\ldots,\beta^d} \frac{1}{n}\|\mathbf{C}_i - \beta(\mathbf{M})\|^2 + \lambda \sum_{i=1}^{d} \|\beta^j\|_{\mathscr{H}_j}$$



Finite dimensional solution

$$\inf_{c^1,\ldots,c^n \in \mathbb{R}^n} \frac{1}{n}\|\mathbf{C}_i - \sum_{j=1}^{d} K_j c^j\|^2 + \lambda \sum_{j=1}^{d} \|c^j\|_{K_j}$$

$$K_j^{\ell k} = \kappa_j(M_j^{(\ell)}, M_j^{(k)})$$

Permutation Learning

Kernelized version

The concepts are now modelled by

$$C_i = \beta_i^\star(M) + \epsilon_i,$$

where $\beta_i^\star$ is a function of the form:

$$\beta^\star(M) = \sum_{j=1}^{d} \beta^j(M_j) \quad \text{only} \quad \beta^{\pi(j)} \neq 0$$

Each $\beta^j \in \mathscr{H}_j$ RKHS, with associated kernel:

$$\kappa_j(M_j, M_j') = \left\langle \varphi_j(M_j), \varphi_j(M_j') \right\rangle$$

Infinite dimensional viewpoint

Finite dimensional solution

$$\min_{\beta^1,\ldots,\beta^d} \frac{1}{n}\|\mathbf{C}_i - \beta(\mathbf{M})\|^2 + \lambda \sum_{i=1}^{d} \|\beta^j\|_{\mathscr{H}_j}$$

$$\Leftrightarrow$$

$$\inf_{c^1,\ldots,c^n \in \mathbb{R}^n} \frac{1}{n}\|\mathbf{C}_i - \sum_{j=1}^{d} K_j c^j\|^2 + \lambda \sum_{j=1}^{d} \|c^j\|_{K_j}$$

$$K_j^{\ell k} = \kappa_j(M_j^{(\ell)}, M_j^{(k)})$$

This we cannot do easily in Python

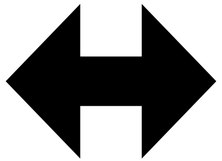
This we can do easily in Python

Permutation Learning

Kernelized version

Infinite dimensional viewpoint

$$\min_{\beta^1,\dots,\beta^d} \frac{1}{n}\|\mathbf{C}_i - \beta(\mathbf{M})\|^2 + \lambda \sum_{i=1}^{d} \|\beta^j\|_{\mathscr{H}_j}$$



Finite dimensional solution

$$\inf_{c^1,\dots,c^n \in \mathbb{R}^n} \frac{1}{n}\|\mathbf{C}_i - \sum_{j=1}^{d} K_j c^j\|^2 + \lambda \sum_{j=1}^{d} \|c^j\|_{K_j}$$

$$K_j^{\ell k} = \kappa_j(M_j^{(\ell)}, M_j^{(k)})$$

Getting the permutation

$$\widehat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \sum_{i=1}^{d} \sum_{j=1}^{d} \|\widehat{c}_i^{\pi(j)}\|$$

Permutation Learning

Algorithms summarised

Algorithm 1 Estimating the permutation using linear regression with Group Lasso regularization

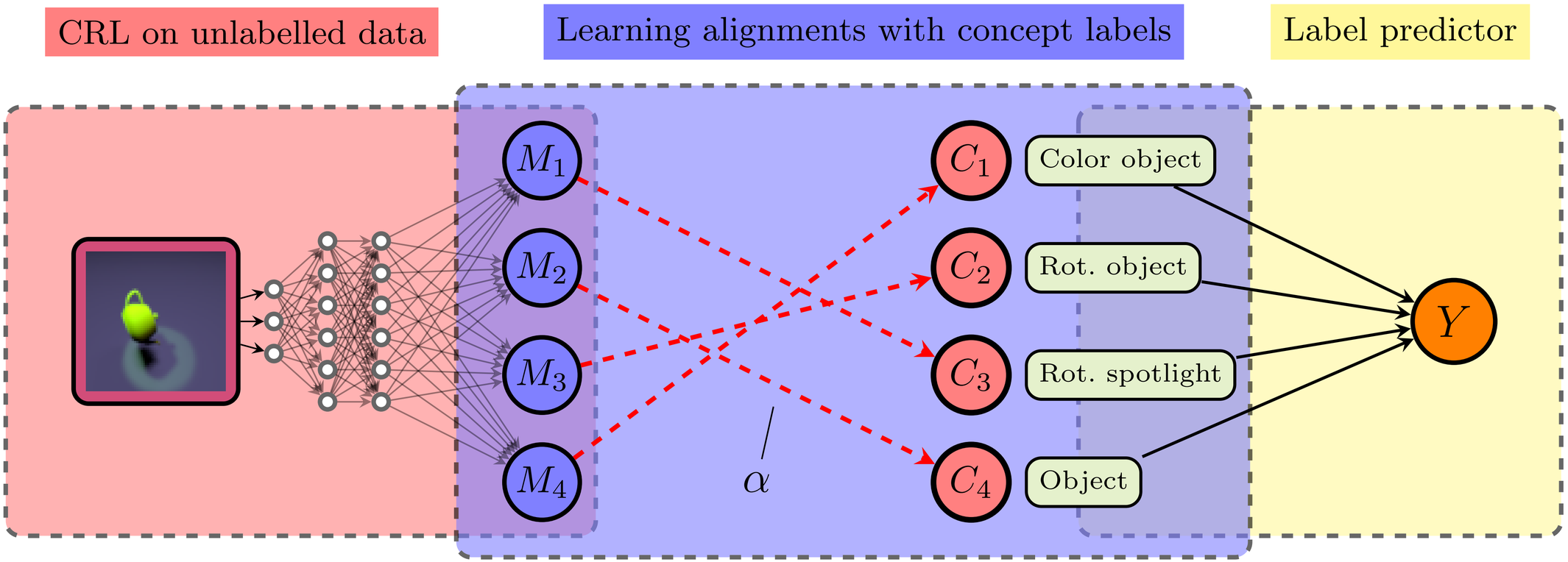
Input: regularization parameter $\lambda > 0$
Data: $\{(C^{(\ell)}, M^{(\ell)})\}_{\ell=1}^{n}$
for $i = 1, \dots, d$ **do**
  $\widehat{\beta}_i \leftarrow \underset{\beta \in \mathbb{R}^{dp}}{\arg\min} \|\mathbf{C}_i - \Phi\beta\|^2 + \lambda\sqrt{p}\|\beta\|_{2,1}$
end for
$\widehat{\pi} \leftarrow \underset{\pi \in \Pi}{\arg\max} \sum_{i=1}^{d} \|\widehat{\beta}_i^{\pi(i)}\|$

Algorithm 2 Estimating the permutation using kernels

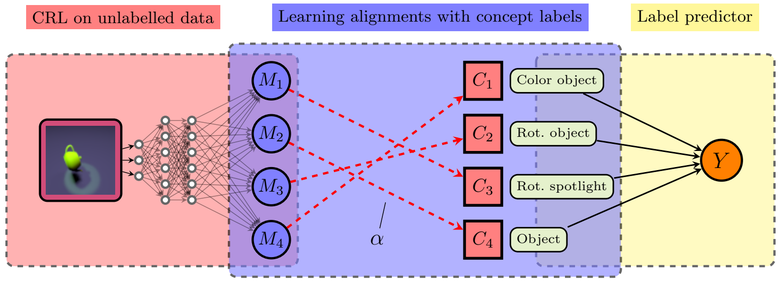
Input: reg. parameter $\lambda > 0$, kernels $\kappa_1, \dots, \kappa_d$
Data: $\{(C^{(\ell)}, M^{(\ell)})\}_{\ell=1}^{n}$
for $j = 1, \dots, d$ **do**
  $(K_j)_{\ell k} \leftarrow \kappa_j(M^{(\ell)}, M^{(k)})$
  $L_j \leftarrow \text{CholeskyDecomposition}(K_j)$
end for
for $i = 1, \dots d$ **do**
  $\widehat{\gamma}_i \leftarrow \underset{\gamma \in \mathbb{R}^{np}}{\arg\min} \frac{1}{n}\|\mathbf{C}_i - \sum_{j=1}^{d} L_j\gamma^j\|^2 + \lambda\|\gamma\|_{2,1}$
end for
$\widehat{\pi} \leftarrow \underset{\pi \in \Pi}{\arg\max} \sum_{i=1}^{d} \|\widehat{\gamma}_i^{\pi(i)}\|$

New Concept-Based Framework

New Concept-Based framework



New Concept-Based framework

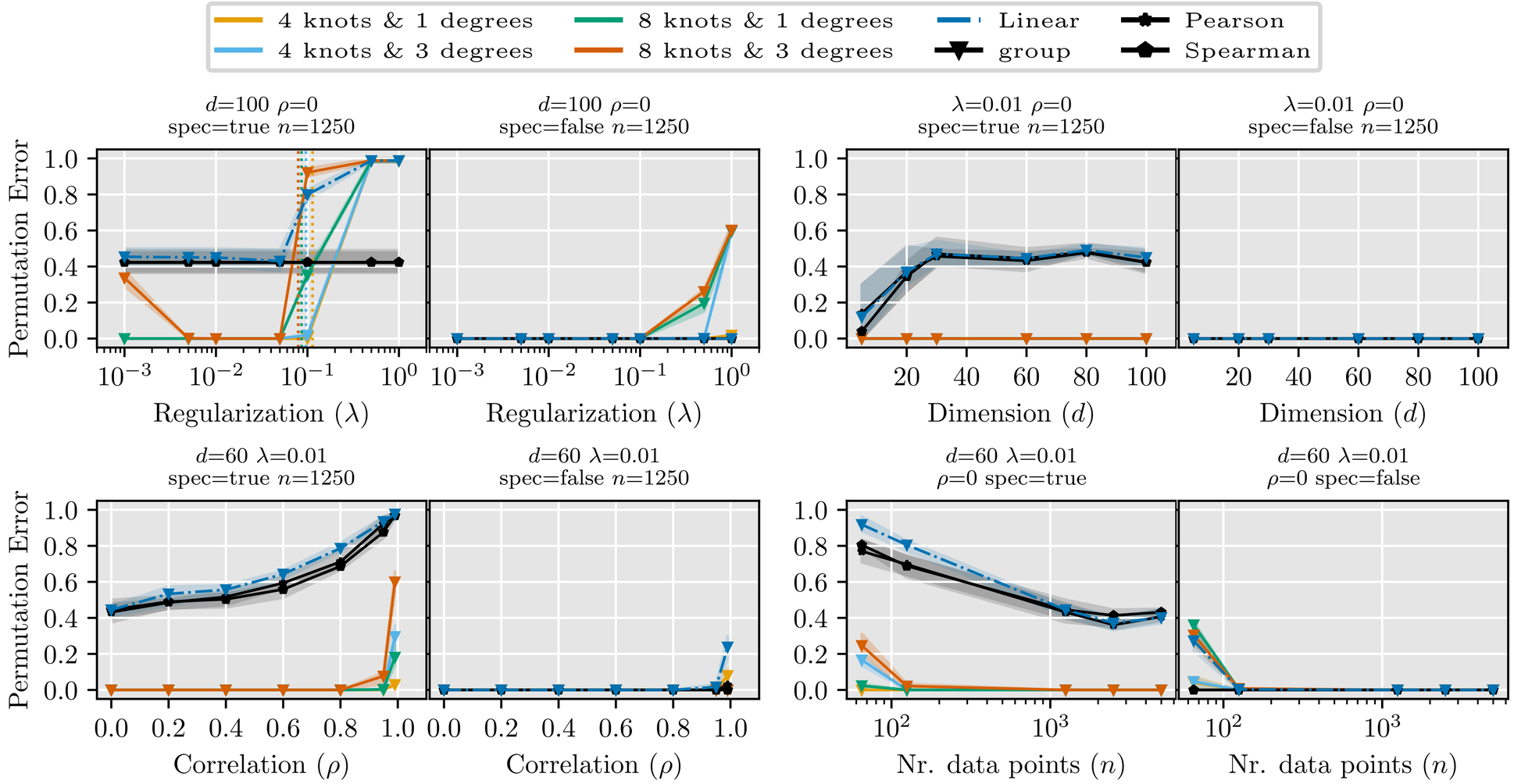


Recipe

1. Pre-train an encoder using unlabelled data
2. Gather some (X, C) pairs to align the learned learned causal variables with the interpretable concepts.
3. Learn any downstream task using (X, Y) pairs.

New Concept-Based framework

Some Experimental Results



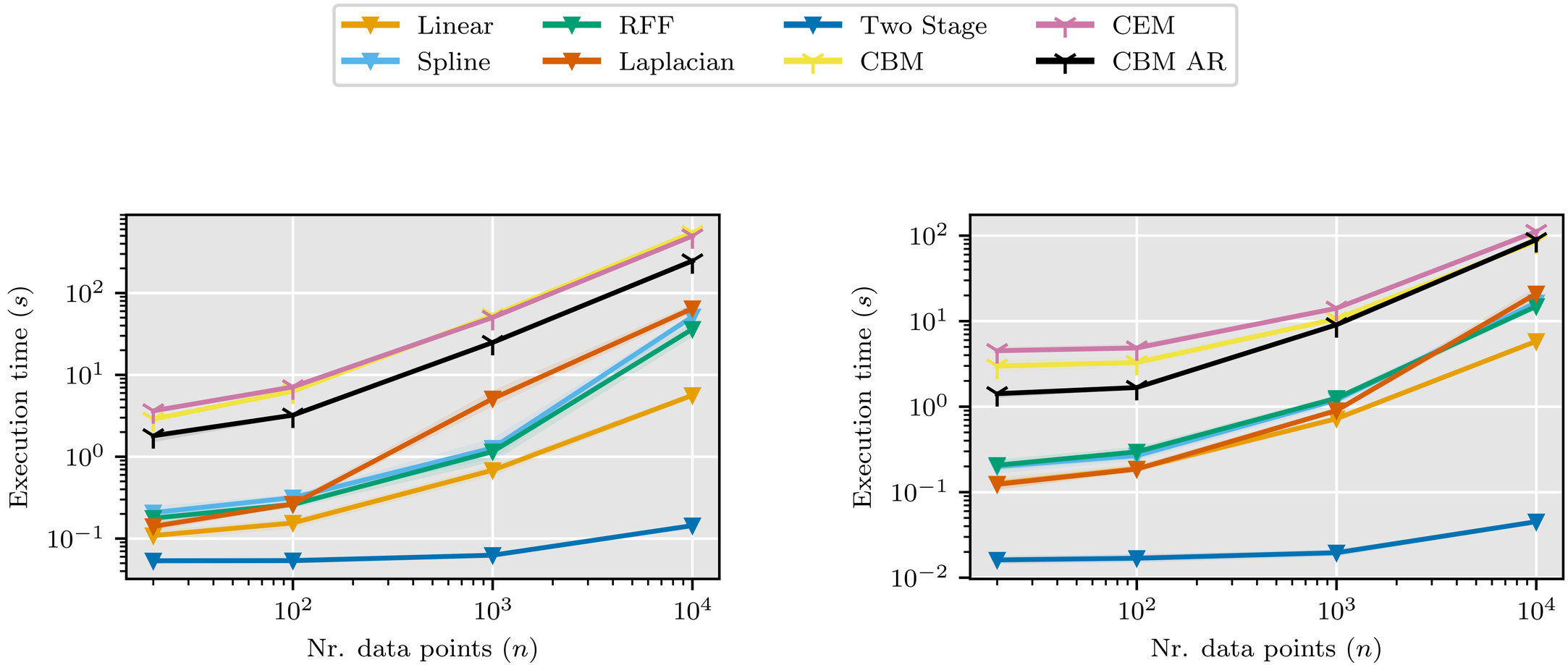
New Concept-Based framework

Some Experimental Results

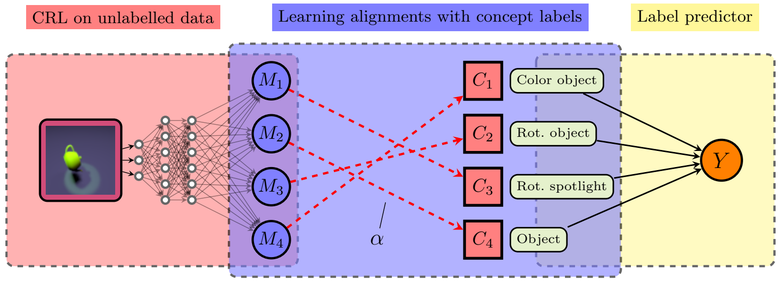
		Label Acc. ↑ (n)				OIS ↓ (n)			
Model	Method	20	100	1000	10000	20	100	1000	10000
Action Sparsity Dataset									
DMS-VAE	Linear	**0.83 ± 0.01**	0.86 ± 0.00	0.85 ± 0.00	0.86 ± 0.00	**0.77 ± 0.03**	0.81 ± 0.01	0.83 ± 0.01	0.84 ± 0.01
	Spline	0.83 ± 0.01	**0.86 ± 0.00**	0.85 ± 0.00	0.86 ± 0.00	0.72 ± 0.03	**0.83 ± 0.02**	**0.85 ± 0.01**	0.86 ± 0.01
	RFF	0.82 ± 0.00	0.85 ± 0.00	0.85 ± 0.00	0.86 ± 0.00	0.77 ± 0.02	0.82 ± 0.02	0.84 ± 0.01	0.85 ± 0.01
	Laplacian	0.81 ± 0.01	0.85 ± 0.00	0.85 ± 0.00	0.86 ± 0.00	0.75 ± 0.03	0.82 ± 0.02	0.84 ± 0.01	0.85 ± 0.01
iVAE	Linear	0.66 ± 0.01	0.68 ± 0.00	0.70 ± 0.00	0.70 ± 0.00	0.73 ± 0.03	0.79 ± 0.02	0.81 ± 0.01	0.83 ± 0.01
	Spline	0.63 ± 0.01	0.68 ± 0.00	0.70 ± 0.00	0.70 ± 0.00	0.69 ± 0.02	0.76 ± 0.02	0.81 ± 0.01	0.83 ± 0.01
	RFF	0.59 ± 0.01	0.66 ± 0.01	0.68 ± 0.00	0.69 ± 0.00	0.59 ± 0.04	0.70 ± 0.02	0.78 ± 0.01	0.81 ± 0.01
	Laplacian	0.59 ± 0.01	0.65 ± 0.01	0.68 ± 0.00	0.68 ± 0.00	0.60 ± 0.04	0.70 ± 0.02	0.78 ± 0.01	0.79 ± 0.01
CBM		0.64 ± 0.01	0.73 ± 0.01	0.85 ± 0.00	0.91 ± 0.00	0.60 ± 0.03	0.60 ± 0.02	0.73 ± 0.01	0.88 ± 0.01
CEM		0.64 ± 0.02	0.72 ± 0.01	0.86 ± 0.00	0.91 ± 0.00	0.66 ± 0.02	0.69 ± 0.03	0.82 ± 0.01	0.88 ± 0.01
HardCBM		0.66 ± 0.01	0.73 ± 0.01	0.85 ± 0.00	**0.91 ± 0.00**	0.56 ± 0.03	0.61 ± 0.01	0.68 ± 0.01	**0.89 ± 0.00**
Temporal Causal3DIdent Dataset									
CITRISVAE	Linear	**0.81 ± 0.01**	0.85 ± 0.00	0.85 ± 0.00	0.85 ± 0.00	0.74 ± 0.06	0.75 ± 0.06	0.80 ± 0.04	0.81 ± 0.04
	Spline	0.80 ± 0.01	**0.87 ± 0.00**	**0.88 ± 0.00**	**0.89 ± 0.00**	0.74 ± 0.06	**0.80 ± 0.04**	**0.82 ± 0.03**	**0.83 ± 0.03**
	RFF	0.71 ± 0.01	0.81 ± 0.01	0.84 ± 0.01	0.85 ± 0.01	0.70 ± 0.06	0.76 ± 0.05	0.79 ± 0.04	0.81 ± 0.04
	Laplacian	0.76 ± 0.01	0.84 ± 0.01	0.87 ± 0.00	0.88 ± 0.00	0.72 ± 0.06	0.78 ± 0.04	0.80 ± 0.03	0.82 ± 0.03
iVAE	Linear	0.81 ± 0.01	0.84 ± 0.00	0.85 ± 0.00	0.85 ± 0.00	0.74 ± 0.06	0.76 ± 0.05	0.78 ± 0.05	0.80 ± 0.04
	Spline	0.77 ± 0.01	0.84 ± 0.00	0.86 ± 0.00	0.87 ± 0.00	0.73 ± 0.06	0.78 ± 0.04	0.79 ± 0.04	0.81 ± 0.04
	RFF	0.75 ± 0.01	0.81 ± 0.01	0.85 ± 0.00	0.86 ± 0.00	0.69 ± 0.06	0.75 ± 0.05	0.78 ± 0.04	0.80 ± 0.04
	Laplacian	0.74 ± 0.01	0.81 ± 0.00	0.85 ± 0.00	0.86 ± 0.00	0.69 ± 0.07	0.76 ± 0.05	0.78 ± 0.04	0.80 ± 0.04
CBM		0.58 ± 0.01	0.65 ± 0.01	0.73 ± 0.00	0.82 ± 0.00	0.57 ± 0.02	0.53 ± 0.03	0.68 ± 0.04	0.78 ± 0.05
CEM		0.61 ± 0.01	0.64 ± 0.01	0.73 ± 0.00	0.82 ± 0.00	0.64 ± 0.04	0.67 ± 0.03	0.72 ± 0.05	0.78 ± 0.05
HardCBM		0.60 ± 0.01	0.65 ± 0.01	0.74 ± 0.00	0.83 ± 0.00	0.57 ± 0.05	0.53 ± 0.02	0.63 ± 0.03	0.77 ± 0.05

New Concept-Based framework

Some Experimental Results



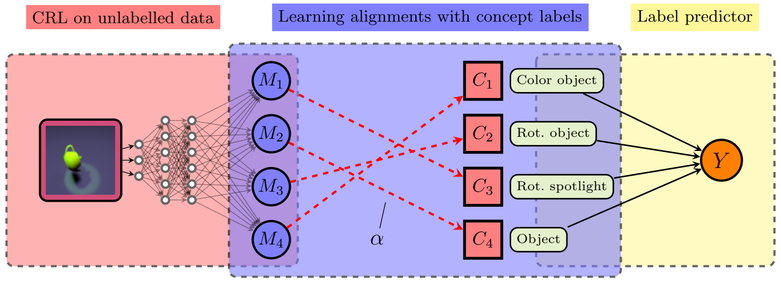
New Concept-Based framework



Some Upsides

1. Our framework is CRL agnostic, so use your favourite method!
2. Pre-training can be done with unlabelled data!
3. Concept part more efficiently trained than CBM methods
4. Concept classification and down stream task often better when not a lot data is available.
5. Provably is robust against shortcut learning and spurious correlations.

New Concept-Based framework



Some Downsides (or opportunities for future work)

1. CRL methods cannot handle > 30 causal variables
2. CRL mostly works with real valued causal variables vs CBM uses discrete values
3. Discrete CRL does exist, but at the moment requires even more assumptions

Thank you for your attention!

References

- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence, 1(5), 206-215.
- Beery, S., Van Horn, G., & Perona, P. (2018). Recognition in terra incognita. In Proceedings of the European conference on computer vision (ECCV) (pp. 456-473).
- Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. (2020). Concept bottleneck models. In International Conference on Machine Learning, ICML, Proceedings of Machine Learning Research. PMLR.
- Zarlenga, M. E., Barbiero, P., Ciravegna, G., Marra, G., Giannini, F., Diligenti, M., Shams, Z., Precioso, F., Melacci, S., Weller, A., et al. (2022). Concept embedding models: beyond the accuracy-explainability trade-off. In Advances in Neural Information Processing Systems, NeurIPS
- Marconato, E., Passerini, A., and Teso, S. (2022). Glancenets: Interpretable, leak-proof concept-based models. In Neural Information Processing Systems, NeurIPS.
- Marconato, E., Passerini, A., and Teso, S. (2023). Interpretability is in the mind of the beholder: A causal framework for human-interpretable representation learning. Entropy.
- Alain, G. (2016). Understanding intermediate layers using linear classifier probes. arXiv preprint arXiv:1610.01644.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In International Conference on Machine Learning, ICML, Proceedings of Machine Learning Research. PMLR.
- Sharkey, L., Braun, D., and Millidge, B. Taking features out of superposition with sparse autoencoders, 2023
- Bühlmann, P. and Van De Geer, S. (2011). Statistics for high-dimensional data: methods, theory and applications. Springer Science & Business Media.
- Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. Journal of Machine Learning Research, 9(40):1179–1225.