Uncertainty-Aware Concept Bottleneck Models with Enhanced Interpretability

Haifei Zhang, Patrick Barry, Eduardo Brandao

Jean Monnet University
Hubert Curien Laboratory, France

September 15, 2025
Porto, Portugal

Image classification using black-box models

Input image

Black-Box Model

$\hat{y}$

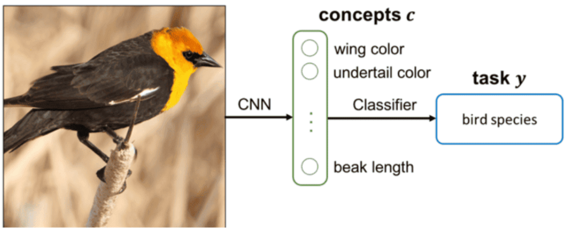
Output

We get the prediction, but we don't understand why!

Concept Bottleneck Models (CBMs)

Core idea: predict labels of images through human-understandable concepts as intermediate reasoning.

1. **Concept encoder:** Image $\boldsymbol{x}$ $\rightarrow$ CNN backbone $\rightarrow$ Concept activation probabilities $\hat{\boldsymbol{c}}$.
2. **Task predictor:** $\hat{c} \rightarrow$ Interpretable classifier $\rightarrow \hat{y}$.

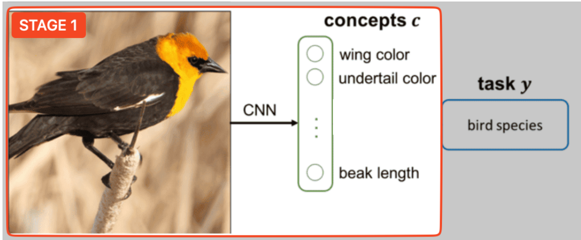


- **Conventional approach for stage 2:** Logistic regression.
  - ✓ Good balance between performance and interpretability.
  - ✗ Coefficient values are abstract and unintuitive to understand.
  - ✗ Difficult to capture uncertainty propagation from the concept prediction to the label prediction.
- **Our Contribution: A novel interpretable classifier for stage 2 that can capture uncertainty.**

Concept Bottleneck Models (CBMs)

Core idea: predict labels of images through human-understandable concepts as intermediate reasoning.

1. **Concept encoder:** Image $\boldsymbol{x}$ $\rightarrow$ CNN backbone $\rightarrow$ Concept activation probabilities $\hat{\boldsymbol{c}}$.
2. **Task predictor:** $\hat{c}$ $\rightarrow$ Interpretable classifier $\rightarrow$ $\hat{y}$.

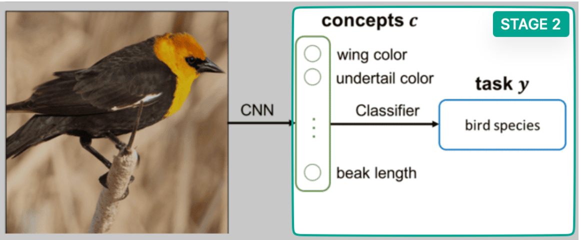


- **Conventional approach for stage 2:** Logistic regression.
  - ✓ Good balance between performance and interpretability.
  - ✗ Coefficient values are abstract and unintuitive to understand.
  - ✗ Difficult to capture uncertainty propagation from the concept prediction to the label prediction.
- **Our Contribution: A novel interpretable classifier for stage 2 that can capture uncertainty.**

Concept Bottleneck Models (CBMs)

Core idea: predict labels of images through human-understandable concepts as intermediate reasoning.

1. **Concept encoder:** Image $\boldsymbol{x}$ $\rightarrow$ CNN backbone $\rightarrow$ Concept activation probabilities $\hat{\boldsymbol{c}}$.
2. **Task predictor:** $\hat{\boldsymbol{c}} \rightarrow$ Interpretable classifier $\rightarrow \hat{y}$.

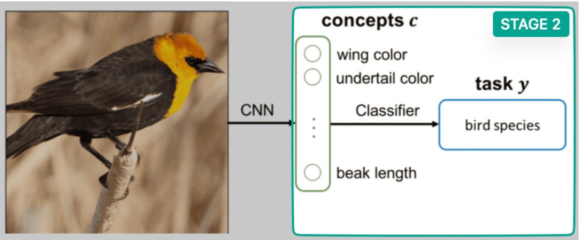


- **Conventional approach for stage 2:** Logistic regression.
  - ✓ Good balance between performance and interpretability.
  - ✗ Coefficient values are abstract and unintuitive to understand.
  - ✗ Difficult to capture uncertainty propagation from the concept prediction to the label prediction.
- **Our Contribution: A novel interpretable classifier for stage 2 that can capture uncertainty.**

Concept Bottleneck Models (CBMs)

Core idea: predict labels of images through human-understandable concepts as intermediate reasoning.

1. **Concept encoder:** Image $\boldsymbol{x}$ $\rightarrow$ CNN backbone $\rightarrow$ Concept activation probabilities $\hat{\boldsymbol{c}}$.
2. **Task predictor:** $\hat{\boldsymbol{c}} \rightarrow$ Interpretable classifier $\rightarrow \hat{y}$.

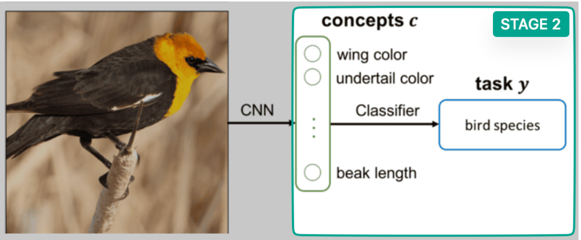


- **Conventional approach for stage 2:** Logistic regression.
  - ✓ Good balance between performance and interpretability.
  - ✗ Coefficient values are abstract and unintuitive to understand.
  - ✗ Difficult to capture uncertainty propagation from the concept prediction to the label prediction.
- **Our Contribution:** A novel interpretable classifier for stage 2 that can capture uncertainty.

Concept Bottleneck Models (CBMs)

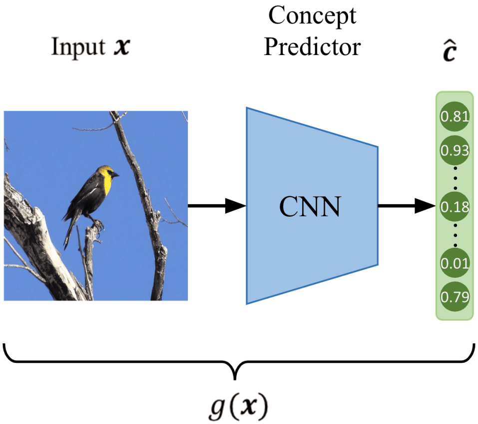
Core idea: predict labels of images through human-understandable concepts as intermediate reasoning.

1. **Concept encoder:** Image $\boldsymbol{x}$ $\rightarrow$ CNN backbone $\rightarrow$ Concept activation probabilities $\hat{\boldsymbol{c}}$.
2. **Task predictor:** $\hat{\boldsymbol{c}} \rightarrow$ Interpretable classifier $\rightarrow \hat{y}$.



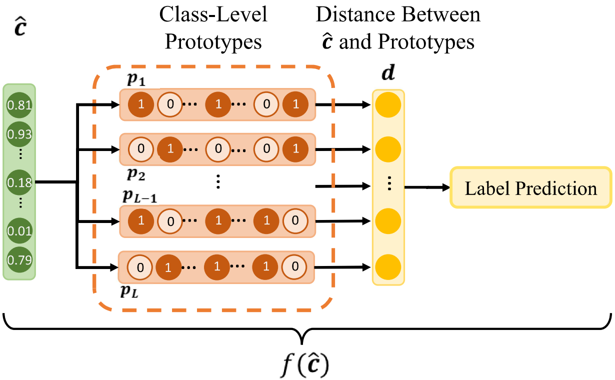
- **Conventional approach for stage 2:** Logistic regression.
  - ✓ Good balance between performance and interpretability.
  - ✗ Coefficient values are abstract and unintuitive to understand.
  - ✗ Difficult to capture uncertainty propagation from the concept prediction to the label prediction.
- **Our Contribution: A novel interpretable classifier for stage 2 that can capture uncertainty.**

Architecture of proposed Class-Level Prototype Classifier (CLPC)



Training of stage 1 to Learn $g(\boldsymbol{x})$: Fine-tune pre-trained CNN to predict concepts $\hat{\boldsymbol{c}}$.

Architecture of proposed Class-Level Prototype Classifier (CLPC)



Training of stage 2 to learn $f(\hat{c})$:

- Assign each class a single **binary-valued prototype** in the concept space.
- Predict label by measuring the distance between **concept activations** $\hat{c}$ and the **prototypes**.

What are prototypes in our setting?

Learnable binary-valued vectors representing **ideal concepts** for a class.

Plane

Car

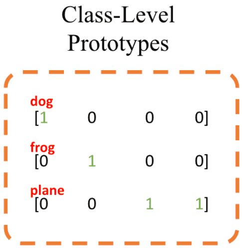
Frog

Dog

	Plane	Car	Frog	Dog
Ears	0	0	0	1
Hairy	0	0	0	1
Wings	1	0	0	0
⋮	⋮	⋮	⋮	⋮
Wet	0	0	1	0
Wheels	1	1	0	0
Metallic	1	1	0	0

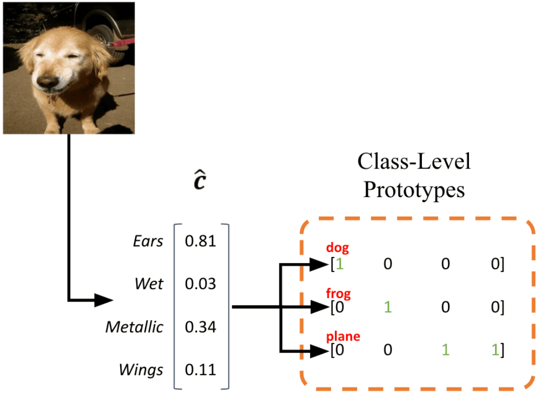
Example of inference

For illustration objective, we consider only 4 concepts: *Ears, Wet, Metallic, Wings*



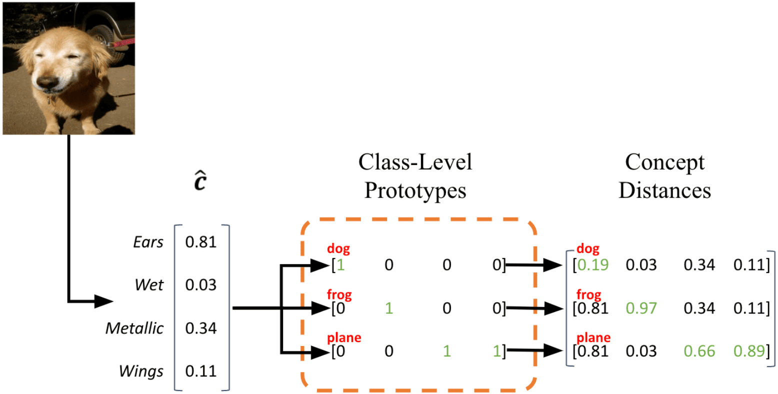
Example of inference

For illustration objective, we consider only 4 concepts: *Ears, Wet, Metallic, Wings*



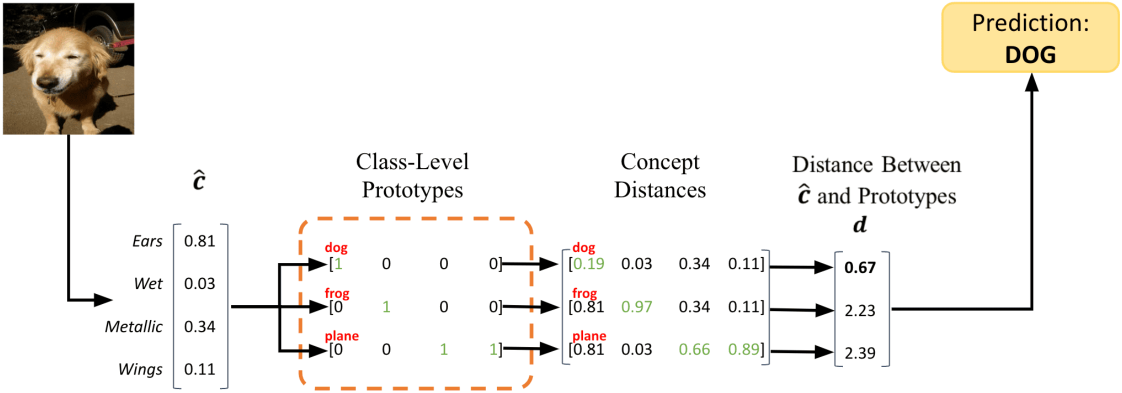
Example of inference

For illustration objective, we consider only 4 concepts: *Ears, Wet, Metallic, Wings*



Example of inference

For illustration objective, we consider only 4 concepts: *Ears, Wet, Metallic, Wings*



Learning prototypes

The loss function learns prototypes that are accurate, sparse, and determinate.

Learning prototypes

The loss function learns prototypes that are accurate, sparse, and determinate.

Total loss:

$$\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_b \mathcal{L}_b \qquad \text{(Total Loss)}$$

Learning prototypes

The loss function learns prototypes that are accurate, sparse, and determinate.

Total loss:

$$\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_b \mathcal{L}_b \qquad \text{(Total Loss)}$$

Loss components: for training set $\{\hat{\boldsymbol{c}}^{(i)}, y^{(i)}\}_{i=1}^{N}$, class label set $\{1, \ldots, L\}$ and concept set $\{1, \ldots, K\}$

$$\mathcal{L}_p = \frac{1}{N} \sum_{i=1}^{N} \left(d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_{y^{(i)}}) - \frac{1}{L-1} \sum_{j \neq y^{(i)}} d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_j) \right) \qquad \text{(Prototype Loss)}$$

Learning prototypes

The loss function learns prototypes that are accurate, sparse, and determinate.

Total loss:

$$\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_b \mathcal{L}_b \qquad \text{(Total Loss)}$$

Loss components: for training set $\{\hat{\boldsymbol{c}}^{(i)}, y^{(i)}\}_{i=1}^{N}$, class label set $\{1, \ldots, L\}$ and concept set $\{1, \ldots, K\}$

$$\mathcal{L}_p = \frac{1}{N} \sum_{i=1}^{N} \left(d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_{y^{(i)}}) - \frac{1}{L-1} \sum_{j \neq y^{(i)}} d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_j) \right) \qquad \text{(Prototype Loss)}$$

$$\mathcal{L}_s = \sum_{j=1}^{L} ||\boldsymbol{p}_j||_1 \qquad \text{(Sparsity Loss)}$$

Learning prototypes

The loss function learns prototypes that are accurate, sparse, and determinate.

Total loss:

$$\mathcal{L} = \mathcal{L}_p + \lambda_s \mathcal{L}_s + \lambda_b \mathcal{L}_b \tag{Total Loss}$$

Loss components: for training set $\{\hat{\boldsymbol{c}}^{(i)}, y^{(i)}\}_{i=1}^{N}$, class label set $\{1, \ldots, L\}$ and concept set $\{1, \ldots, K\}$

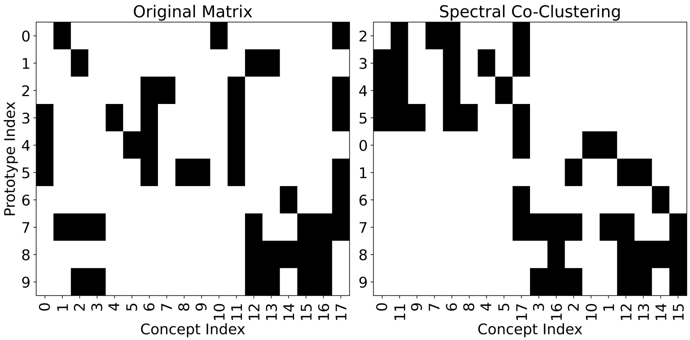
$$\mathcal{L}_p = \frac{1}{N} \sum_{i=1}^{N} \left(d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_{y^{(i)}}) - \frac{1}{L-1} \sum_{j \neq y^{(i)}} d(\hat{\boldsymbol{c}}^{(i)},\ \boldsymbol{p}_j) \right) \tag{Prototype Loss}$$

$$\mathcal{L}_s = \sum_{j=1}^{L} ||\boldsymbol{p}_j||_1 \tag{Sparsity Loss}$$

$$\mathcal{L}_b = \sum_{j=1}^{L} \sum_{k=1}^{K} (1 - p_{jk}) \cdot p_{jk} \tag{Binary Loss}$$

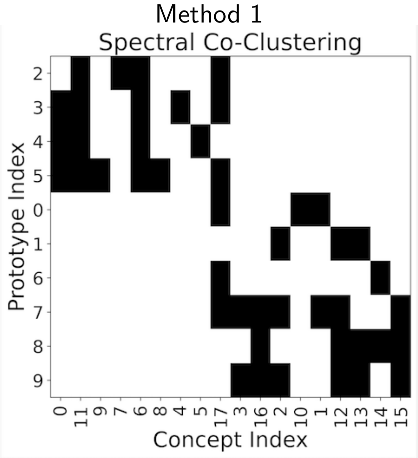
Global explanation: what have we learned?

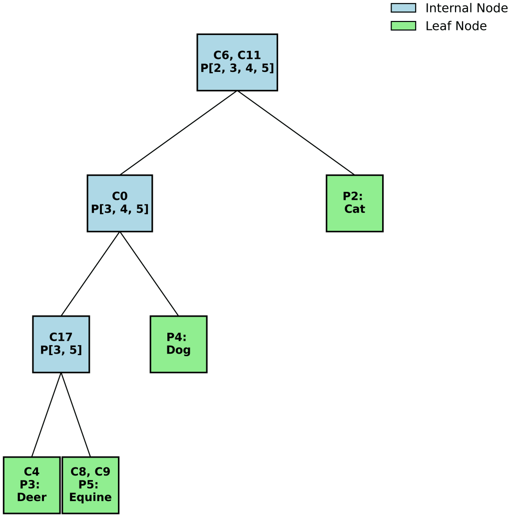
Clustering prototypes and concepts.



Global explanation: what have we learned?

Prototype tree

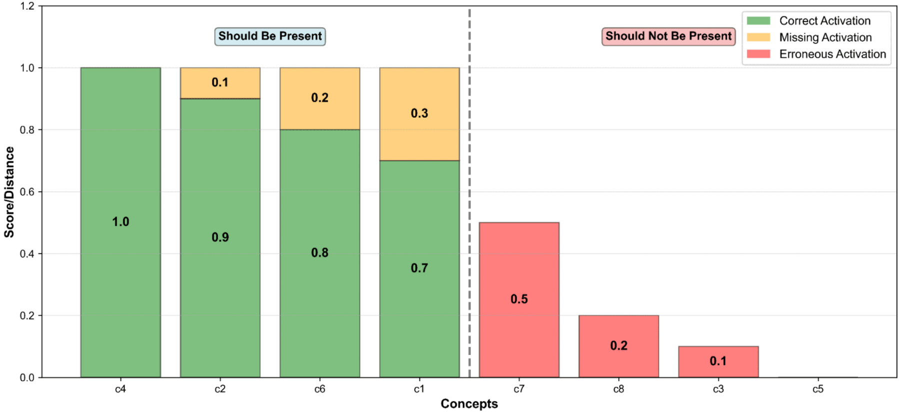




Local explanation: why do we get this prediction?

Example

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
Prototype	1	1	0	1	0	1	0	0
$\hat{c}$	0.7	0.9	0.1	1.0	0.0	0.8	0.5	0.2
Uncertainty	0.3	0.1	0.1	0.0	0.0	0.2	0.5	0.2



Concept intervention: how can we get correct prediction?

Intervene 1-by-1 on the most "impactful" concepts to correct wrong label predictions.

- Conventional concept ordering strategy: Feature-importance-based.
  However, **the error in concept prediction is not considered**.
- Our proposed concept ordering strategy: Gain-based.
  Consider both the importance of concepts and the error in concept prediction.
  - For Logistic Regression:

$$\text{LR-Gain}_k = w_{j^*k} \cdot \left(\mathbb{1}(w_{j^*k} > 0) - \hat{c}_k\right).$$

  - For CLPC:

$$\text{CLPC-Gain}_k = |p_{j^*k} - \hat{c}_k|.$$

Where j^* and k are indices for true class and concepts, respectively.

If $w_{j^*k} < 0$ or $p_{j^*k} = 0$, set $\hat{c}_k$ to 0. In contrast, if $w_{j^*k} > 0$ or $p_{j^*k} = 1$, set $\hat{c}_k$ to 1.

Experimental setup

Image Datasets

- **CUB (Birds)**: 200 classes, 112 concepts
- **Derm7pt (Skin Lesions)**: 5 classes, 19 concepts
- **RIVAL10 (Objects)**: 10 classes, 18 concepts

Evaluations

- Baseline: Logistic regression
- Experiments:
  1. Classification accuracy
  2. Conformal prediction
  3. Robustness to concept noise
  4. Concept intervention efficiency

Classification accuracy

Table: Classification accuracy results

Dataset	Concept Acc (%)	Ave. $\|\|\boldsymbol{p}_j\|\|$	Accuracy (%)		Δ (%)
			LR	CLPC	
CUB	94.86	21.95/112	**76.46**	76.01	*-0.45*
Derm7pt	88.38	6.59/19	**66.33**	64.81	*-1.52*
RIVAL10	99.71	4.50/18	**99.17**	98.96	*-0.21*

Key takeaway

CLPC has competitive classification accuracy compared to logistic regression.

Conformal Prediction (CP)

What is CP?

A framework that yields reliable **set-valued or empty predictions** with guaranteed error rates.

Table: Conformal prediction performance (*error rate = 5%*)

Dataset	Set Acc (%)		Set Size		Reject Ratio (%)	
	LR	CLPC	LR	CLPC	LR	CLPC
CUB	92.12	**94.97**	1	1	**29.5**	53.30
Derm7pt	87.34	**94.43**	**2.15**	3.38	0	0
RIVAL10	**99.96**	99.92	1	1	**5.07**	5.37

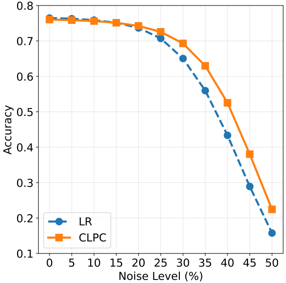
Key takeaway

CLPC is more sensitive and cautious in the face of uncertainty.

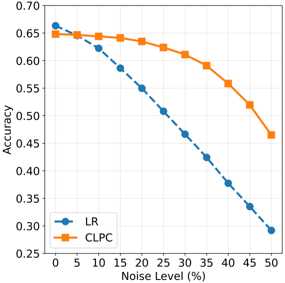
Robustness to concept noise

Inject noise by randomly flipping $\alpha\%$ concepts:

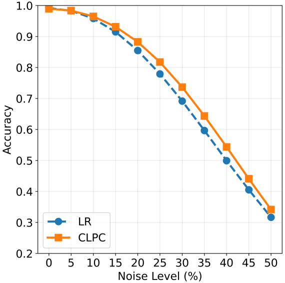
- concept activation score $\leq 0.5 \rightarrow$ random value in $(0.5, 1]$;
- concept activation score $> 0.5 \rightarrow$ random value in $[0, 0.5]$.



(a) CUB



(b) Derm7pt

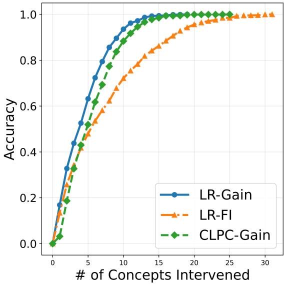


(c) RIVAL10

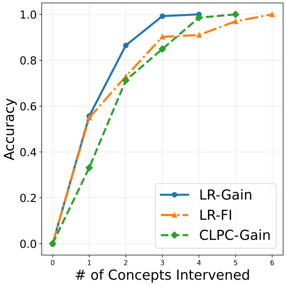
Key takeaway

CLPC is more robust to noise in concept prediction and thus more reliable for low-quality input images.

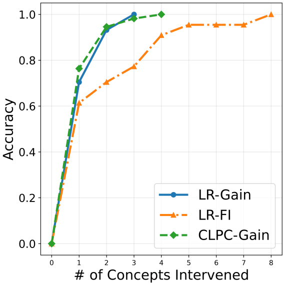
Concept intervention efficiency



(a) CUB



(b) Derm7pt



(c) RIVAL10

Key takeaway

Gain-based strategies are more efficient than the feature-importance-based strategy.

Conclusion

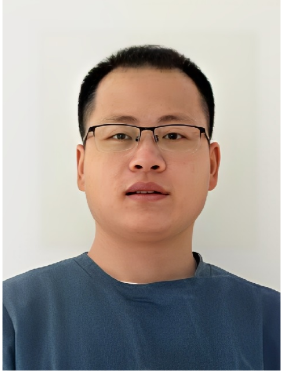
Our proposed CLPC model has:

- Competitive performance as conventional interpretable models;
- Enhanced global and local interpretability;
- Natural capability to capture uncertainty propagation from concepts to labels;
- Strong robustness to noise in concept predictions.

Future work:

- Learn multiple prototypes per class.
- Investigate concept leakage present in the model.
- Conduct user-centred evaluations to validate model interpretability.

Our team

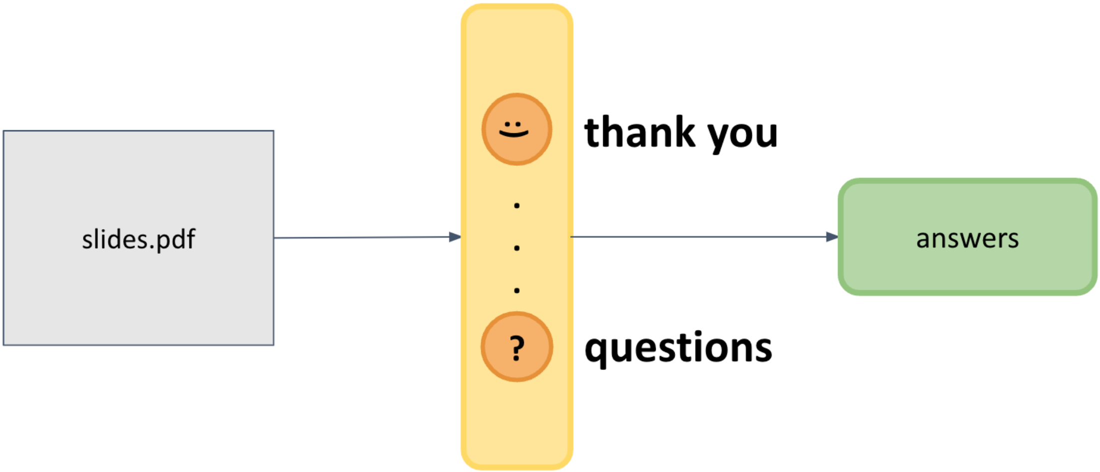


Haifei Zhang
Associate Professor

Patrick Barry
Master Student

Eduardo Brandao
Associate Professor

Lab page: https://laboratoirehubertcurien.univ-st-etienne.fr/en/index.html


slides.pdf
thank you
questions
answers