NeSy Learning
Day 3: Learning Imbalances

Efthymia Tsamoura
Huawei Labs

Emile van Krieken
VU, Amsterdam

ESSLLI 2025

About this Course

✓Day 1: Introduction to NeSy

✓Day 2: Learnability

✓Day 3: Learning Imbalances in NeSy

✓Day 4: Reasoning Shortcuts

✓Day 5: Probabilistic Reasoning

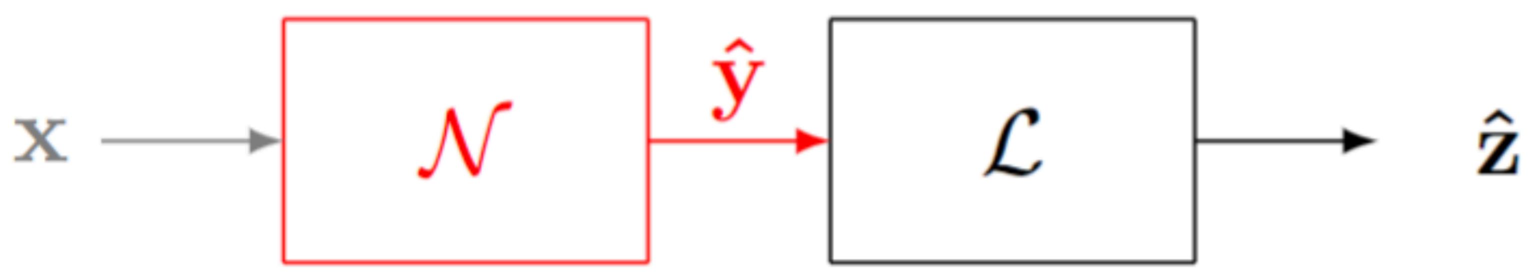
Outline of Today's Lecture

- ✓**Introduction to Learning Imbalances**
    - ✓Learning Imbalances in traditional ML
    - ✓Learning Imbalances in NeSy
- ✓**Mitigating Learning Imbalances**
    - ✓Testing time techniques
        - ✓Reduction to robust optimal transport
    - ✓Training time techniques
        - ✓Reduction to integer linear programming
- ✓**Teacher-Student NeSy**

Quick Recap of Day 2

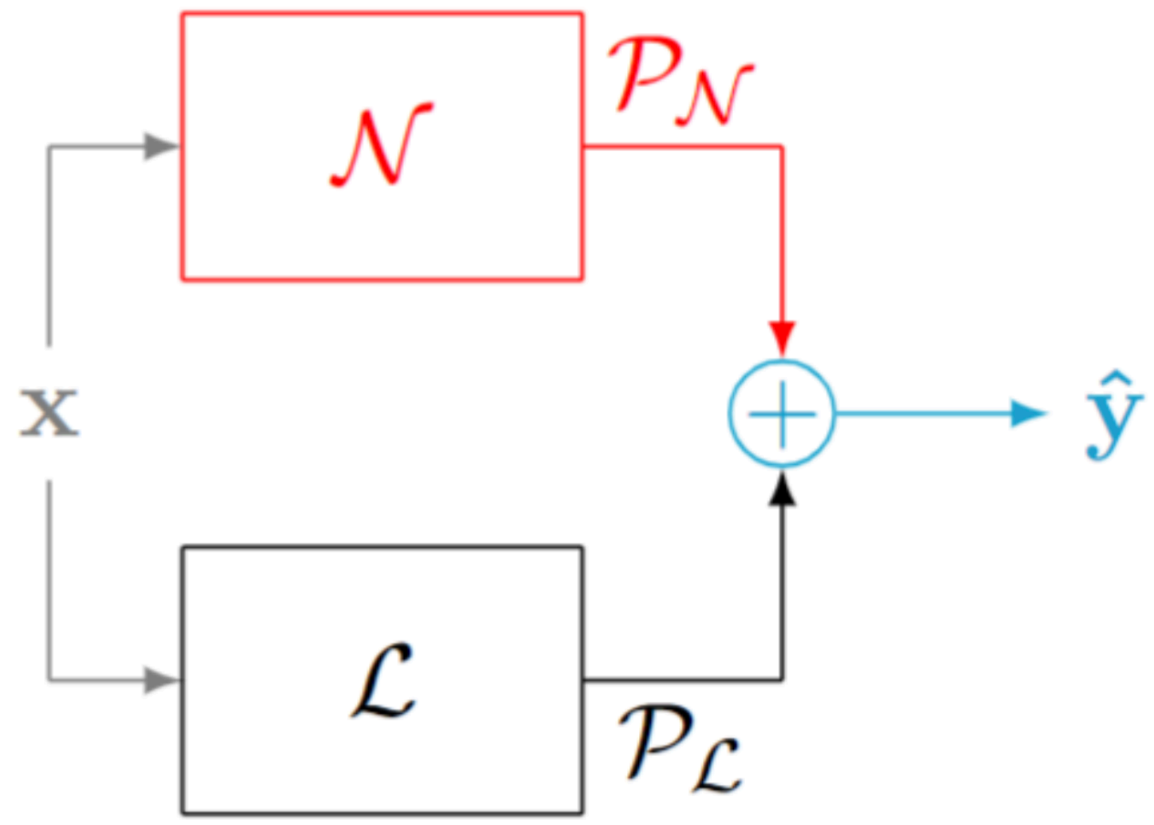
Types of Integration

Stratified



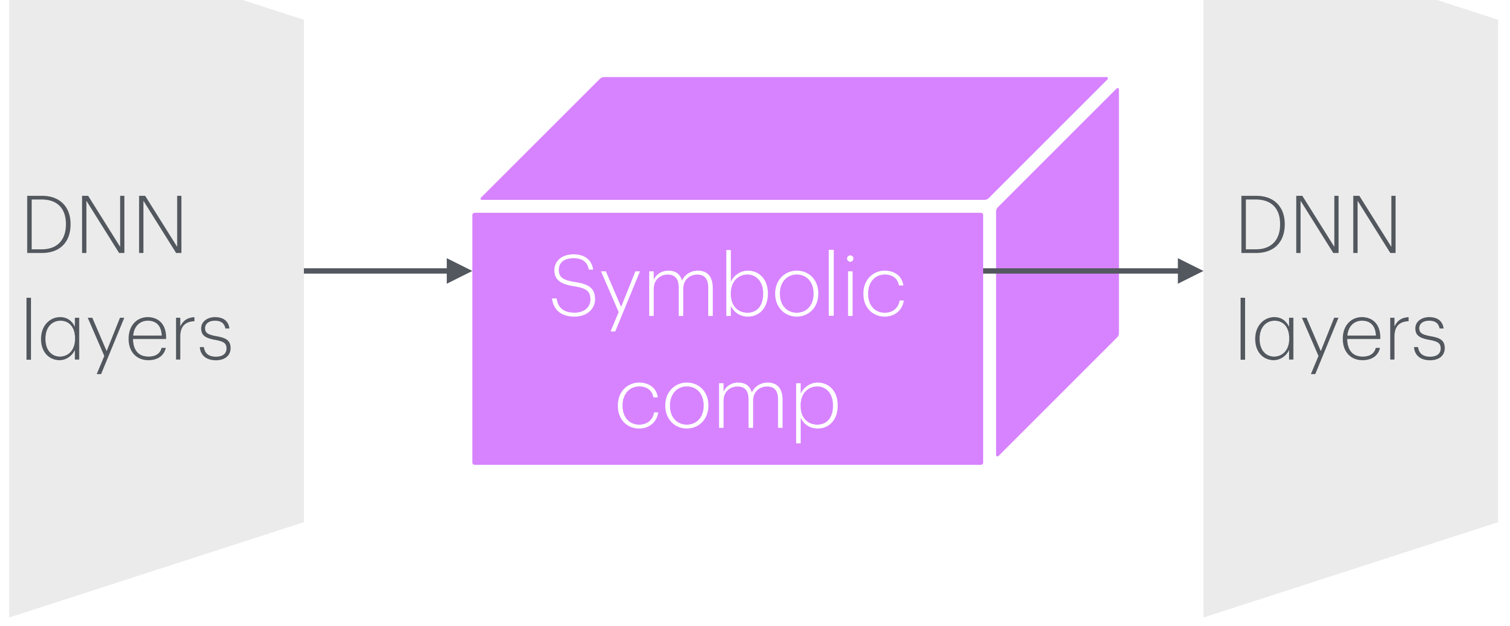
- ✓DeepProbLog [NeurIPS 2018]
- ✓ABL [NeurIPS 2019]
- ✓NeurASP [IJCAI 2020]
- ✓NeuroLog [AAAI 2021]
- ✓Scallop [NeurIPS 2021]
- ✓ENT [ICLR 2023]
- ✓DeepSoftLog [NeurIPS 2023]
- ✓ISED [NeurIPS 2023]
- ✓Dolphin [arXiv 2024]

Teacher-Student



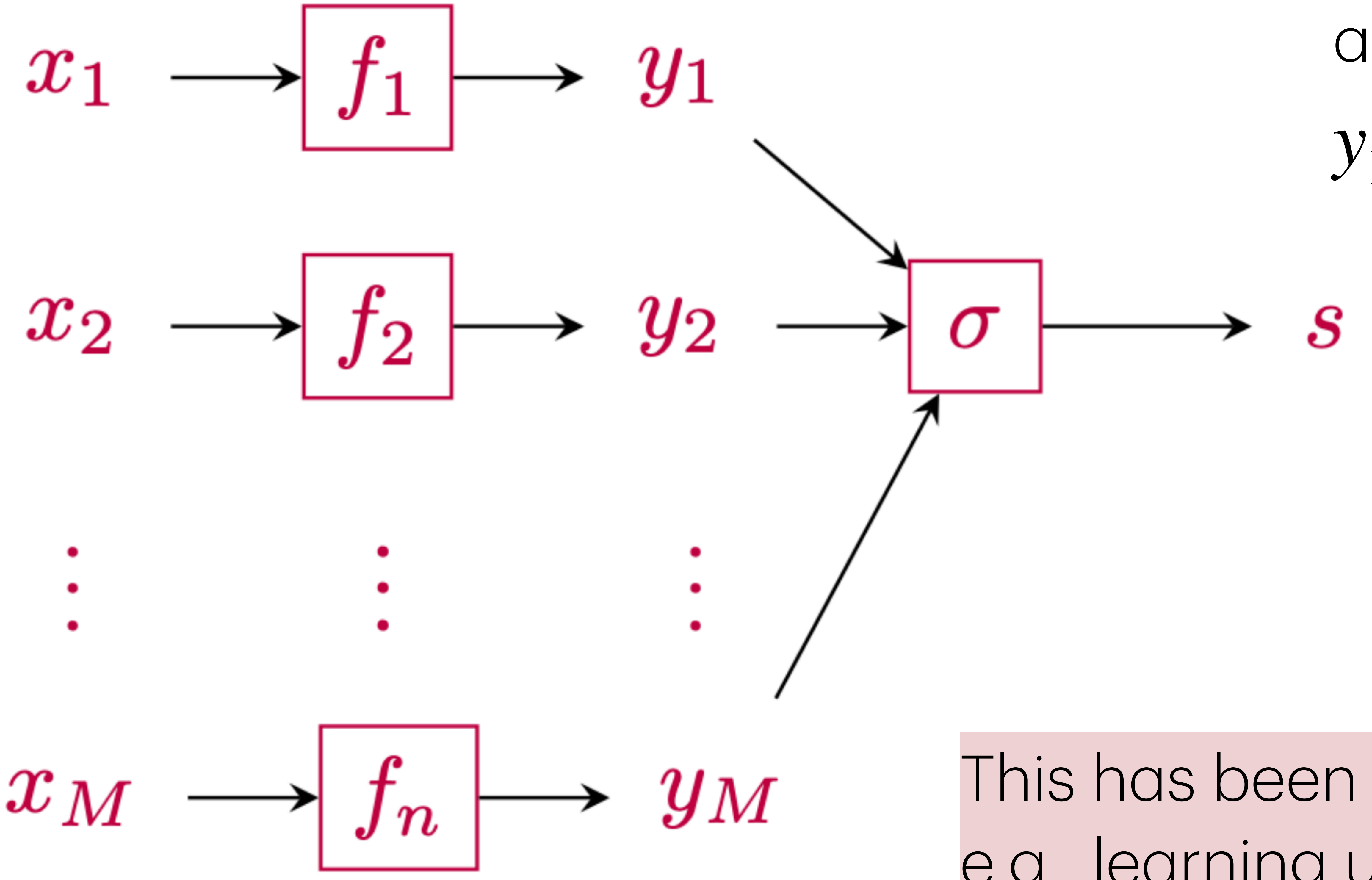
- ✓T-S, ACL [EMNLP 2016]
- ✓DPL [EMNLP 2018]
- ✓Concordia [ICML 2023]

As a layer



- ✓MIPaaL [AAAI 2020]
- ✓BB-backprop [ICLR 2020]
- ✓CombOptNet [ICML 2021]
- ✓SurCo [ICML 2023]
- ✓GenCO [ICML 2024]

Learning Setting



Problem formulation: Given the x_i's and s, **learn** the f_i's. The gold labels $y_1, \ldots, y_M$ are **unknown**.

This has been an open problem in other relevant fields, e.g., learning under indirect supervision.

Kaifu Wang, Efthymia Tsamoura, and Dan Roth. On Learning Latent Models with Multi-Instance Weak Supervision. In NeurIPS, 2023.

PAC-Learnability

A problem instance is PAC-learnable if there exists an algorithm $\mathscr{A}$ such that for any two user parameters ϵ and δ the following holds under any input distribution:

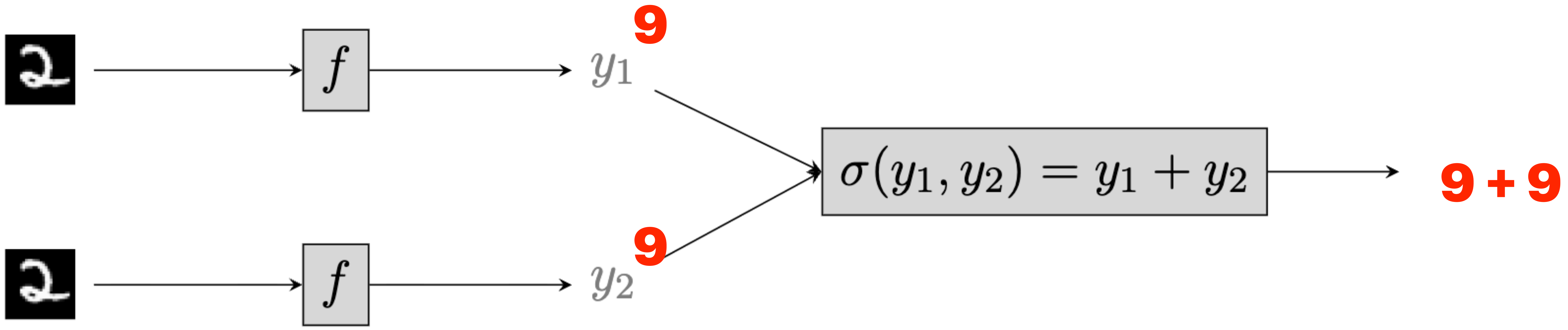
✓ with probability at least 1-δ

✓ the learned classifier f misclassifies an input with probability $\leq \epsilon$

✓ when given at least $m_{\epsilon,\delta}$ samples

Polynomial in ϵ and δ

PAC-Learnability: Known and Deterministic σ



Reasoning: if for any **(y,…,y)** and **(y′,…,y′)**, we have σ**(y,…,y)** $\neq$ σ**(y′,…,y′)**, then the classification errors are **not concealed.**

Suppose the following:

✓ All mass is concentrated in  with gold label 2

✓ f misclassifies  as **9**. Hence, the gold labels are **(2,2)**, but f outputs **(9,9)**

Question: are f's classification errors concealed or not?

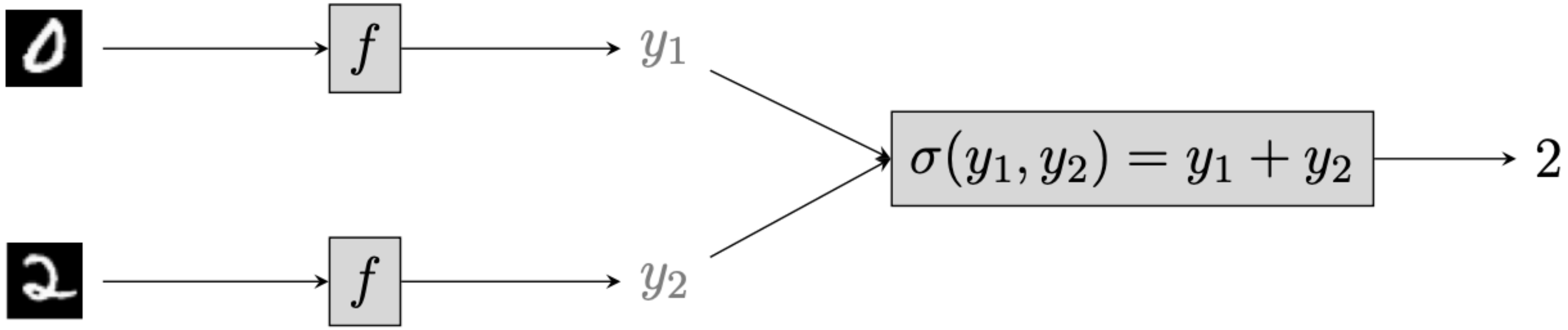
Answer: no, since **2 + 2** $\neq$ **9+9**

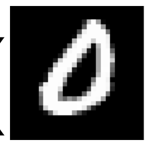
Relevant Settings

✓Partial label learning

✓Learning via transition matrices

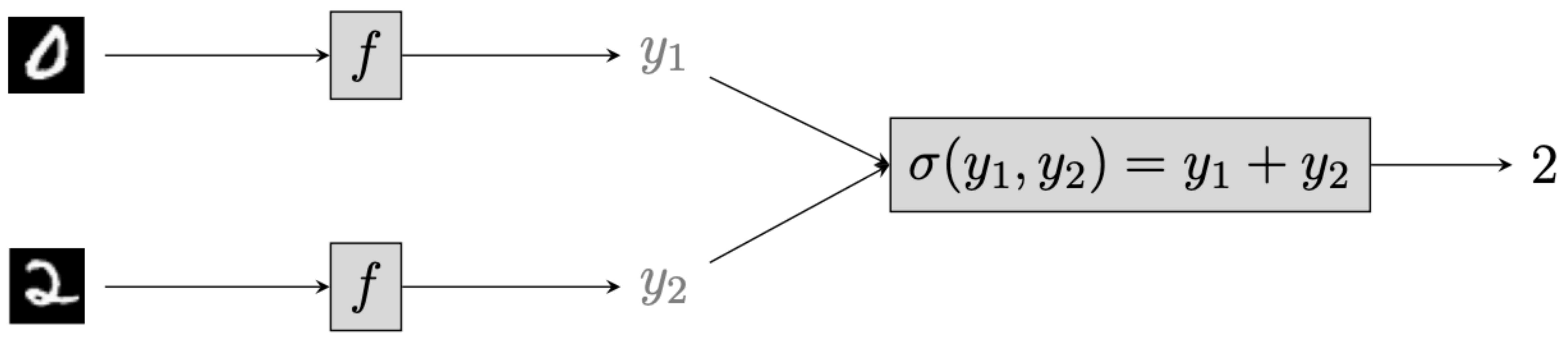
NeSy Learning



Training sample: ((, ), {(0,2), (2,0), (1,1)})

Mutually exclusive set of candidate label vectors

Transition Matrix Formulation of 2SUM



hidden labels

observed labels

$$
\mathbf{T}_1 = \mathbf{T}_2 =
\begin{array}{c}
\\
s=0 \\ s=1 \\ \vdots \\ s=9 \\ s=11 \\ \vdots \\ s=18
\end{array}
\begin{array}{c}
\begin{array}{cccc} \quad 0 \quad & \quad 1 \quad & \cdots & \quad 9 \quad \end{array} \\
\begin{bmatrix}
\mathbb{P}(Y=0) & 0 & \cdots & 0 \\
\mathbb{P}(Y=1) & \mathbb{P}(Y=0) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\mathbb{P}(Y=9) & \mathbb{P}(Y=8) & \cdots & \mathbb{P}(Y=0) \\
0 & 0 & \cdots & \mathbb{P}(Y=2) \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \mathbb{P}(Y=9)
\end{bmatrix}
\end{array}
$$

Learning Imbalances in NeSy

Kaifu Wang, Efthymia Tsamoura, and Dan Roth. On characterizing and mitigating imbalances in multi-instance weak supervision. CoRR, abs/2407.10000, 2024

Learning Imbalances: What Are They?

Major differences in the errors occurring when classifying instances of different classes (aka *class-specific risks*). In other words:

A classifier is much better in classifying instances of some class (e.g., cats) than classing instance of other classes (e.g., different species of birds)

Learning Imbalances in Traditional ML

- ✓Core problem in ML
  - ✓Real-world data is imbalanced
- ✓Theoretical results focus on supervised learning
  - ✓Very few theoretical results in weakly-supervised learning [Journal of Machine Learning Research, 2011]

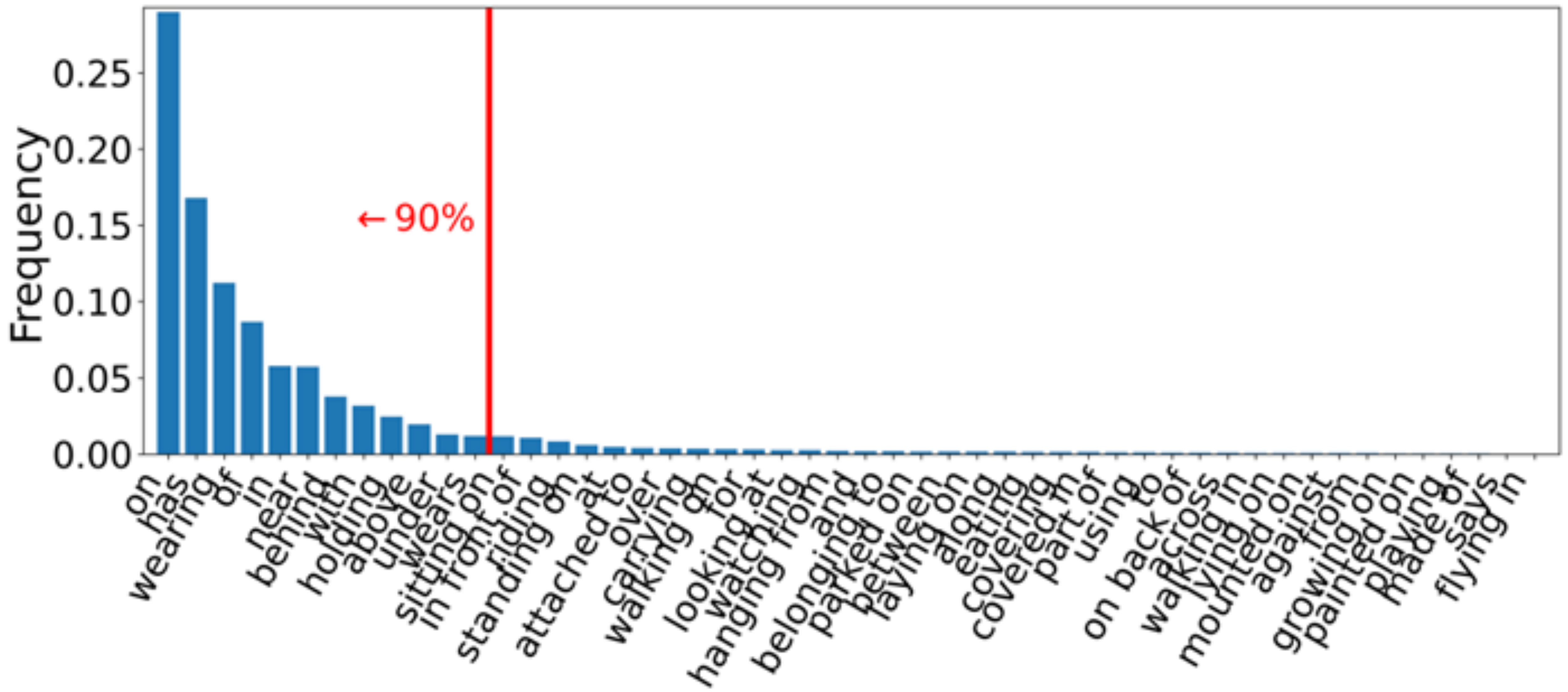


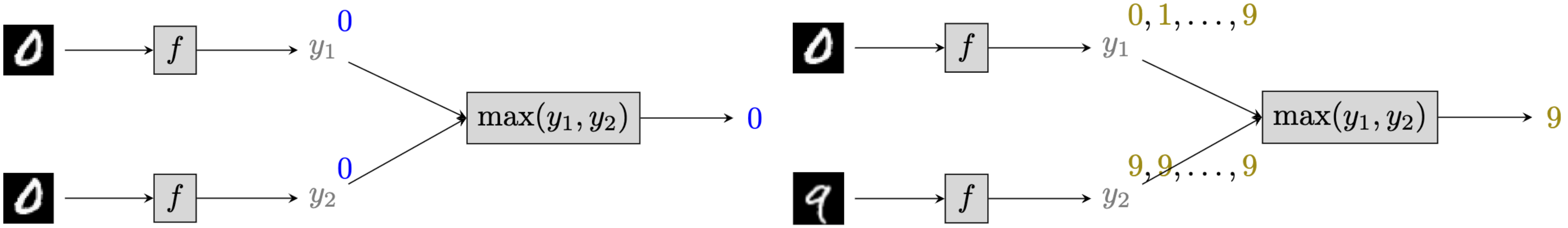
Figure. Distribution of training facts categorized per class in Visual Genome [International Journal of Computer Vision, 2017]

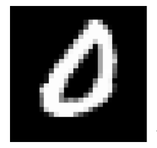
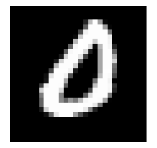
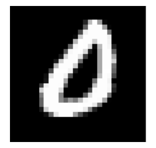
Learning Imbalances in Neurosymbolic Learning

Root of learning imbalances in traditional ML: imbalanced training data

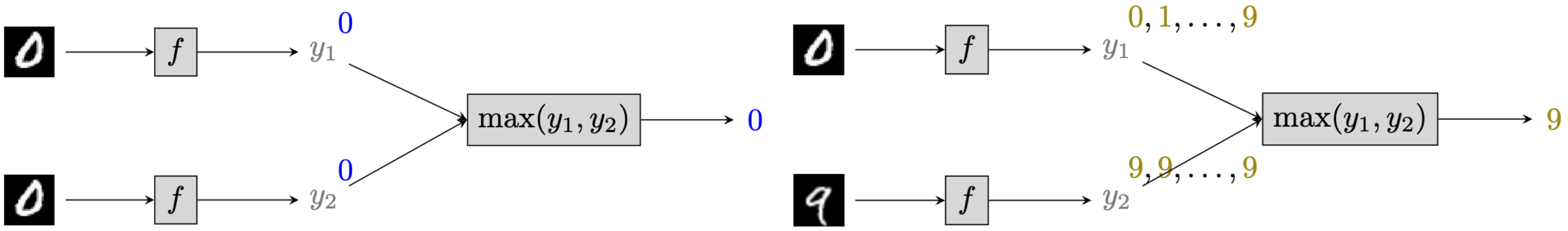
Question: Do you think that there is **another root of learning imbalances in NeSY?**

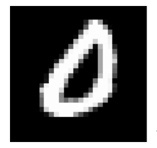
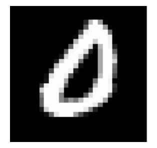
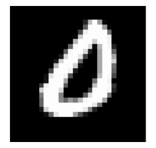
Learning Imbalances in Neurosymbolic Learning



Question: Which class is easier to learn when the number of (,,0) equals the number of (,, 9)?

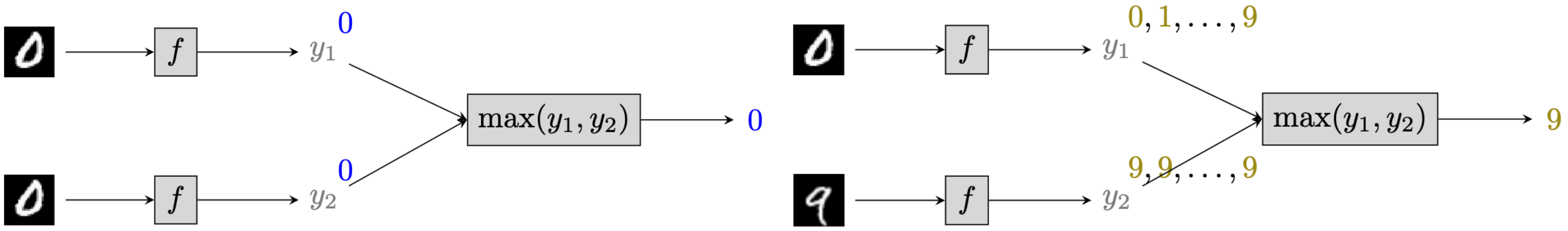
Learning Imbalances in Neurosymbolic Learning

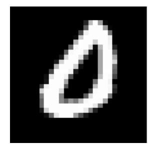


Question: Which class is easier to learn when the number of (, , 0) equals the number of (, , 9)?

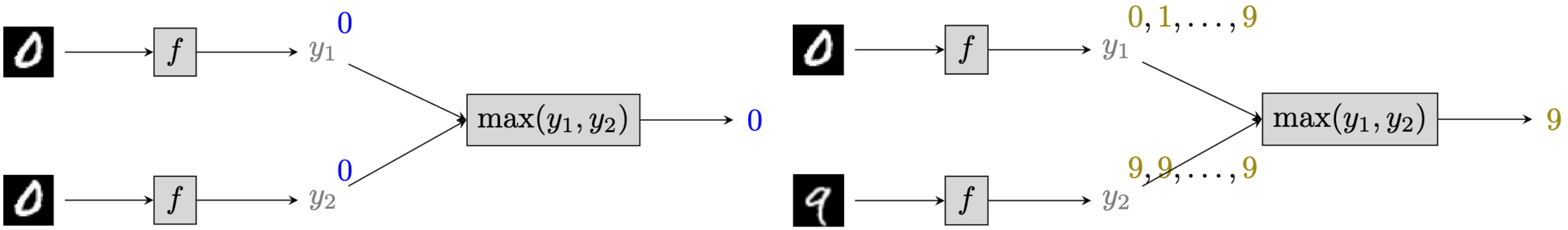
Answer: class **0** **(reduction to supervised learning)**

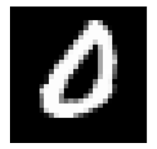
Learning Imbalances in Neurosymbolic Learning

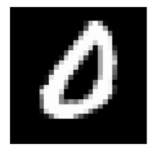
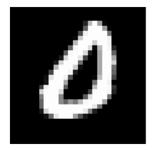
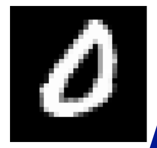
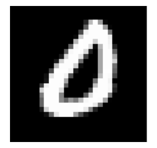


Question: Which class is easier to learn when the number of  equals the number of  and samples are formed by i.i.d. sampling?

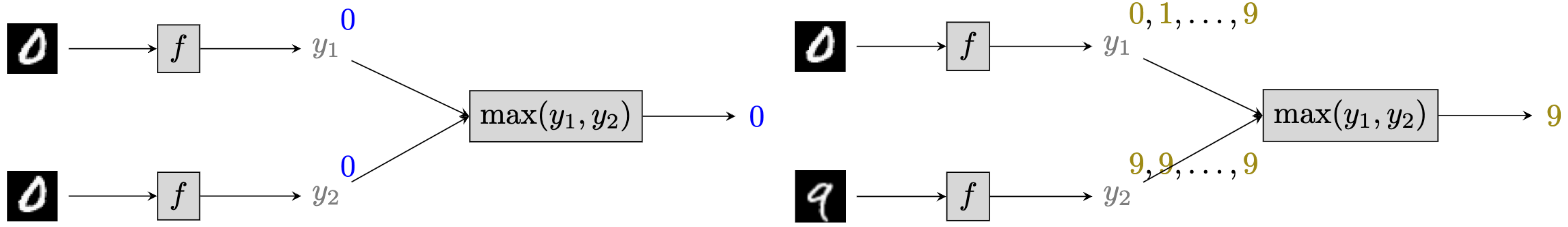
Learning Imbalances in Neurosymbolic Learning



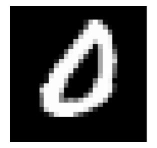
Question: Which class is easier to learn when the number of  equals the number of  and samples are formed by i.i.d. sampling?

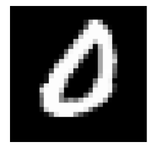
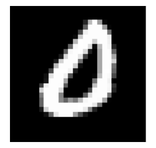
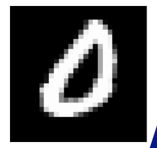
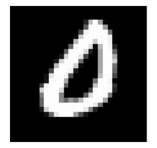
Answer: class **9 (way more samples of the form (** **,**  **, 9), (**  **,** **, 9), or (**  **,** **, 9) than (** **,**  **, 0)**

Learning Imbalances in Neurosymbolic Learning



The sampling process along with σ may lead to imbalances in the samples

Question: Which class is easier to learn when the number of  equals the number of  and samples are formed by i.i.d. sampling?

Answer: class **9 (way more samples of the form (, , 9), (, , 9), or (, , 9) than (, , 0)**

Learning Imbalances in Neurosymbolic Learning

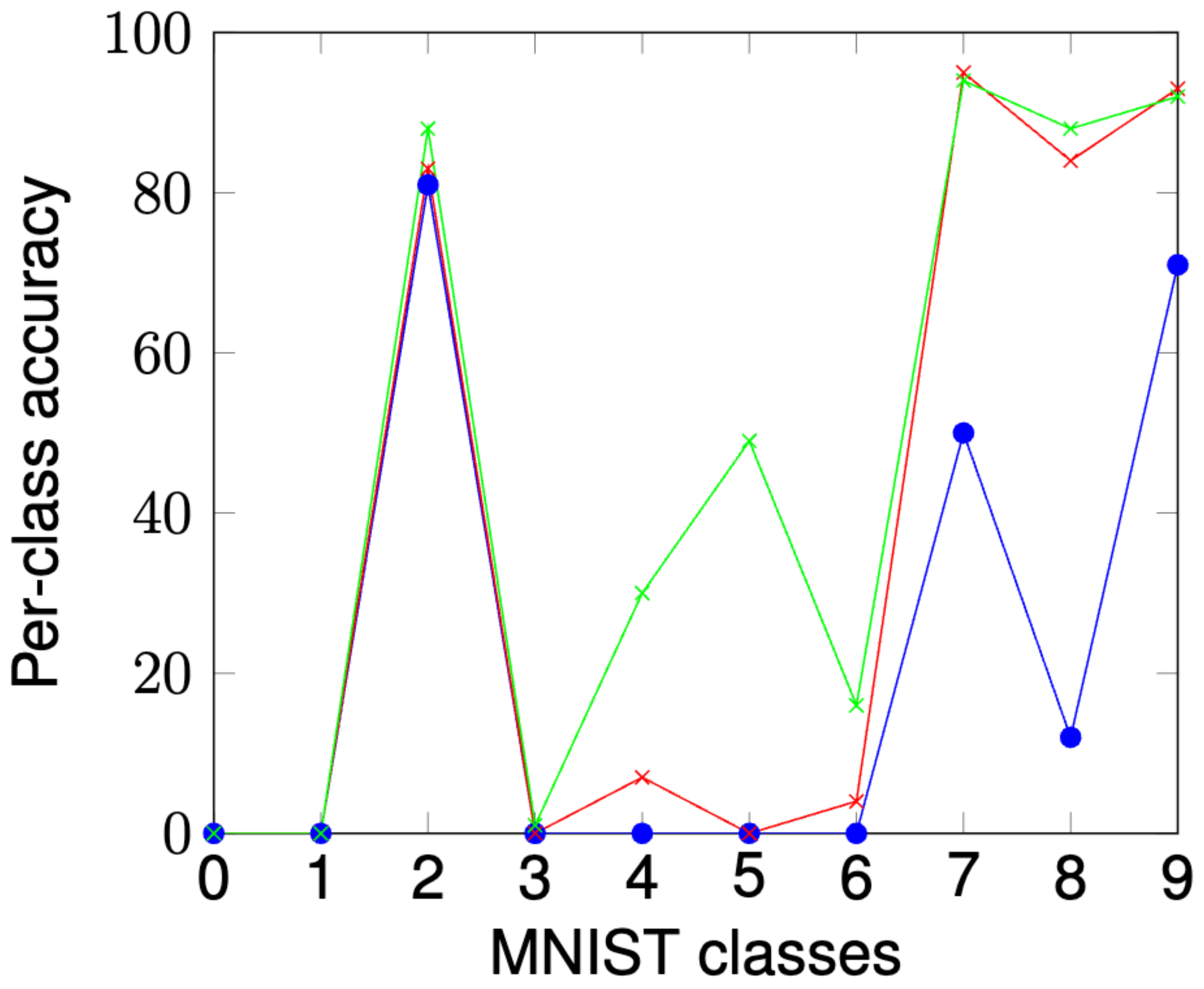


Figure: Accuracy of the MNIST classifier. Blue, red and green curves show accuracy at 20, 40 and 100 epochs. Learning converges in 100 epochs.

Learning Imbalances in Neurosymbolic Learning: Characterization

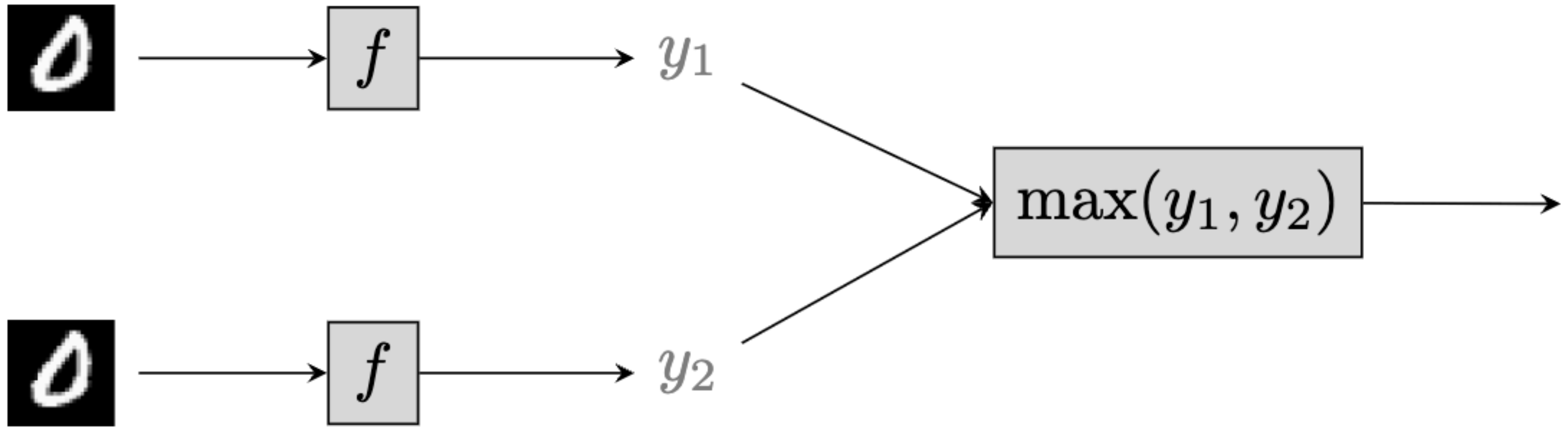
✓We bounded $R_j(f)$ via function:

Probability classifier f misclassifies an instance of class j (e.g., a zero)

Probability the overall output is wrong (e.g., target max is 9, but we output 0)

$$\Phi_{\sigma,j}(R_{\mathsf{P}}(f,\sigma))$$

Symbolic function, e.g., max



In other words, if you the **probability of obtaining a wrong overall output**, you can bound the **probability f misclassifies a specific class**

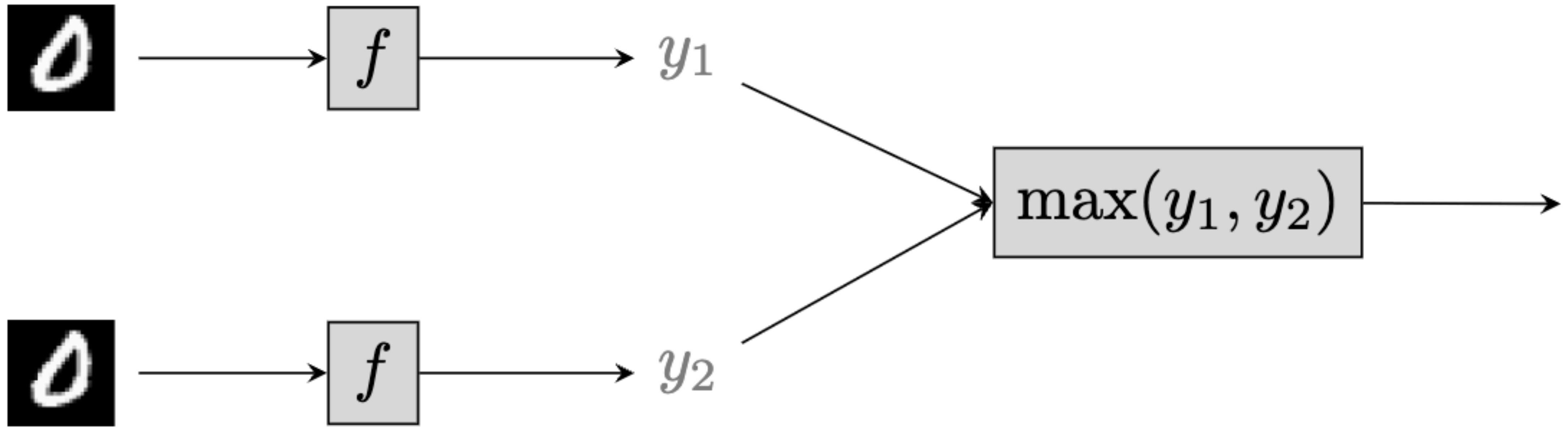
Learning Imbalances in Neurosymbolic Learning: Characterization

Proposition 3.3. *Let $d_{[\mathcal{F}]}$ be the Natarajan dimension of $[\mathcal{F}]$. Given a confidence level $\delta \in (0,1)$, we have that $R_j(f) \leq \Phi_{\sigma,j}(\widetilde{R}_{\mathsf{P}}(f;\sigma,\mathcal{T}_{\mathsf{P}},\delta))$ with probability $1-\delta$ for any $j \in [c]$, where*

$$\widetilde{R}_{\mathsf{P}}(f;\sigma,\mathcal{T}_{\mathsf{P}},\delta) = \widehat{R}_{\mathsf{P}}(f;\sigma,\mathcal{T}_{\mathsf{P}}) + \sqrt{\frac{2\log(em_{\mathsf{P}}/2d_{[\mathcal{F}]}\log(6Mc^2d_{[\mathcal{F}]}/e))}{m_{\mathsf{P}}/2d_{[\mathcal{F}]}\log(6Mc^2d_{[\mathcal{F}]}/e)}} + \sqrt{\frac{\log(1/\delta)}{2m_{\mathsf{P}}}} \tag{3}$$

Empirical error in the overall output

Number of training samples



In other words, if you the **probability of obtaining a wrong overall output**, you can bound the **probability f misclassifies a specific class**

Learning Imbalances in Neurosymbolic Learning: Characterization

✓We bounded $R_j(f)$ via function:

Probability classifier f misclassifies an instance of class j (e.g., a zero)

Probability the overall output is wrong (e.g., target max is 9, but we output 0)

$$\Phi_{\sigma,j}(R_{\mathsf{P}}(f,\sigma))$$

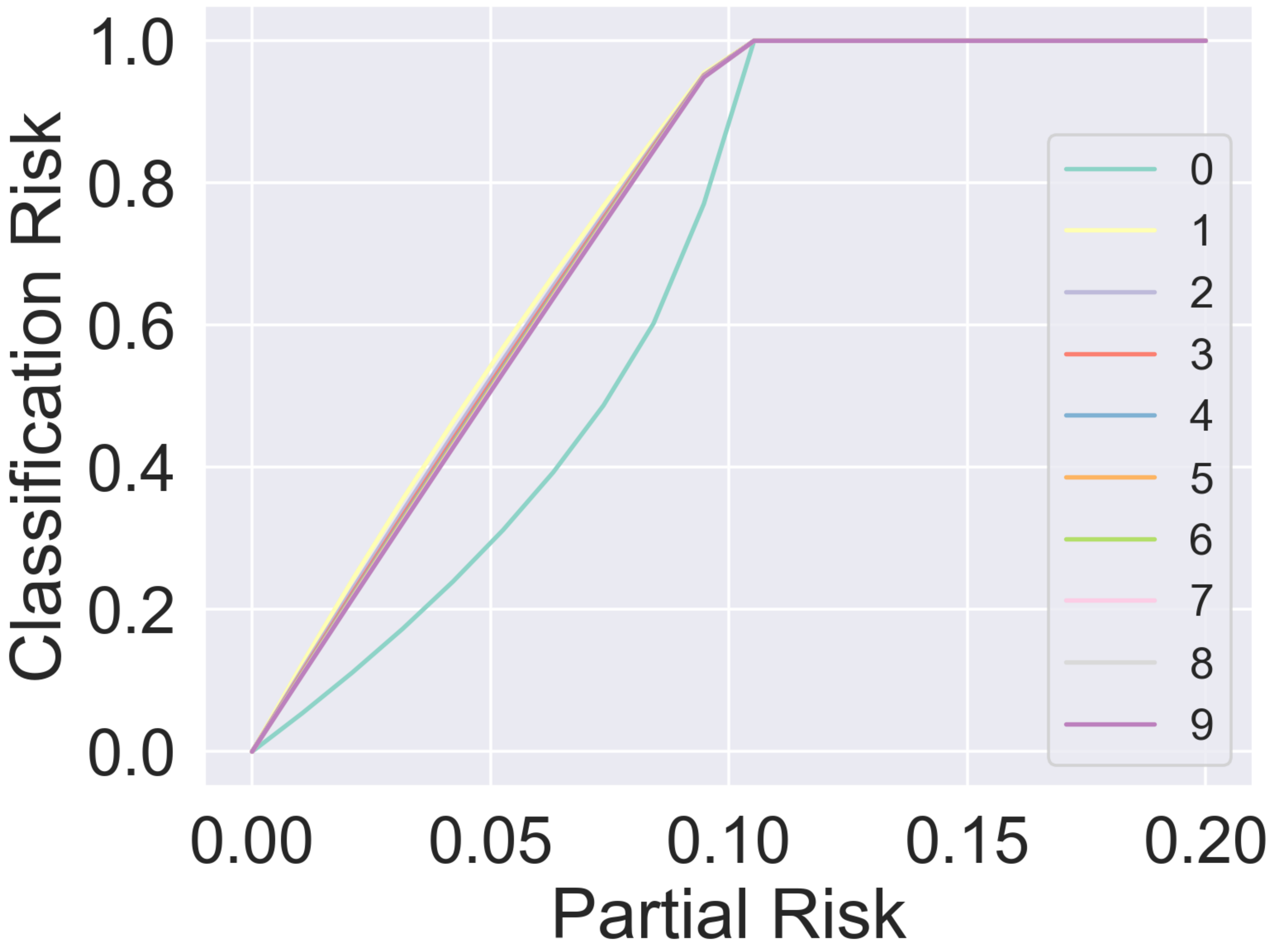
Symbolic function, e.g., max

✓This bound is computed via solving a quadratic program

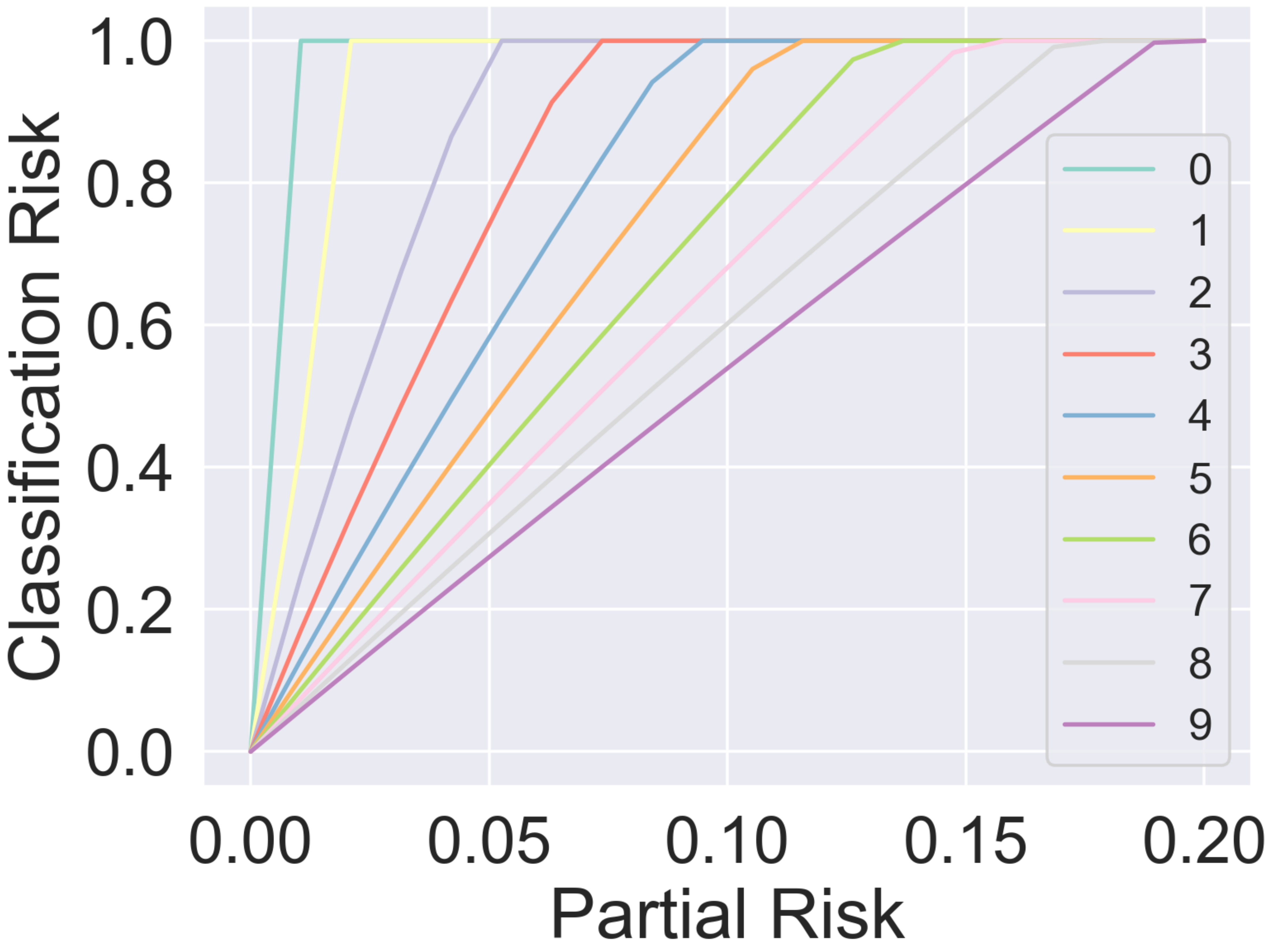
✓Does not make any assumptions on $\boldsymbol{\sigma}$

Learning Imbalances in Neurosymbolic Learning: Characterization

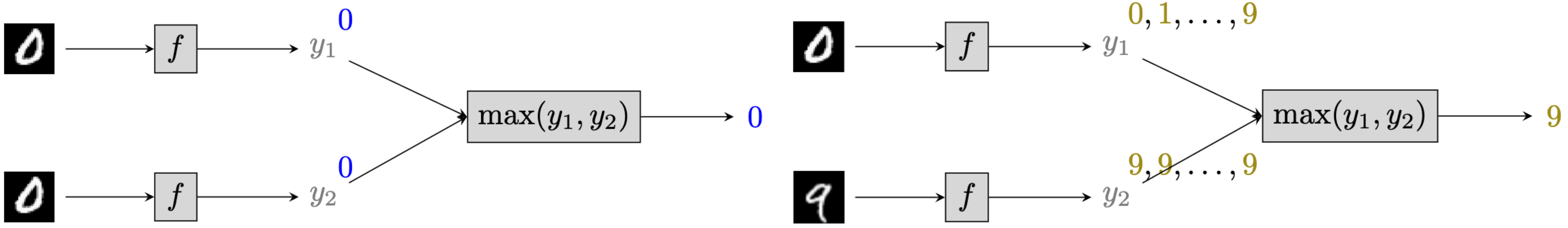
Easier to learn **0** (reduction to supervised learning)



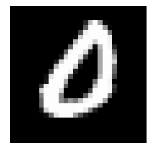
Easier to learn **9** (more samples giving supervision on 9)



Learning Imbalances in Neurosymbolic Learning



Existing results in **ML: equally difficult to learn class 0 and 9**

Question: Which class is easier to learn when the number of  equals the number of  and samples are formed by i.i.d. sampling?

Mitigating Imbalances in NeSy

Kaifu Wang, Efthymia Tsamoura, and Dan Roth. On characterizing and mitigating imbalances in multi-instance weak supervision. CoRR, abs/2407.10000, 2025

Mitigating Imbalances

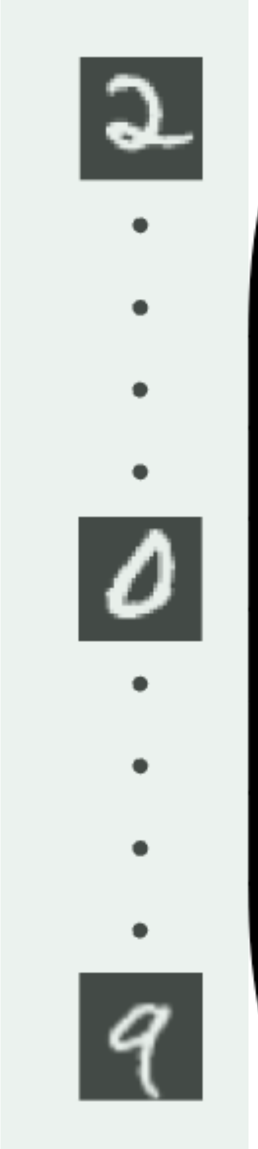
Objective: Enforce the prior distribution (common approach in ML):

✓ Give more importance to minority classes during training

✓ Encourage the model to predict minority classes during testing

Mitigating Imbalances During Testing

Prediction matrix $\mathbf{P}$



$$\begin{array}{c} y=0 \cdots\cdots y=9 \\ \begin{pmatrix} 0.1 & \cdots\cdots & 0.05 \\ \vdots & & \vdots \\ 0.7 & \cdots\cdots & 0.01 \\ \vdots & & \vdots \\ 0.01 & \cdots\cdots & 0.8 \end{pmatrix} \end{array}$$

Predictions for the i-th test sample

Testing sample

Rationale: Given a (gold) label distribution $\hat{r}$, correct the predictions $\mathbf{P}$ to $\mathbf{P}'$, so that $\mathbf{P}'$ adheres to $\hat{r}$.

Challenges:

✓The technique should be lightweight

✓$\mathbf{P}'$ should be close enough to $\mathbf{P}$

✓$\mathbf{P}'$ should not strictly abide to $\hat{r}$ (to tolerate noise)

Mitigating Imbalances During Testing

Rationale: Given a (gold) label distribution $\hat{r}$, correct the predictions $\mathbf{P}$ to $\mathbf{P}'$, so that $\mathbf{P}'$ adheres to $\hat{r}$.

Challenges:

✓The technique should be lightweight

✓$\mathbf{P}'$ should be close enough to $\mathbf{P}$

✓$\mathbf{P}'$ should not strictly abide to $\mathbf{P}$ (to tolerate noise)

$$\min_{\mathbf{P}' \in \mathbb{R}_+^{n \times c}, \mathbf{P}'\mathbf{1}_c = \mathbf{1}_n} \langle -\log(\mathbf{P}), \mathbf{P}' \rangle + \tau KL(\mathbf{P}'\mathbf{1}_n \,||\, n\hat{r})$$

$\mathbf{P}'$ should induce a valid distribution

Mitigating Imbalances During Testing

$$\min_{\mathbf{P}' \in \mathbb{R}_+^{n \times c}, \mathbf{P}'\mathbf{1}_c = \mathbf{1}_n} \langle -\log(\mathbf{P}), \mathbf{P}' \rangle + \tau KL(\mathbf{P}'\mathbf{1}_n || n\hat{r})$$

$\mathbf{P}'$ should induce a valid distribution

- ✓ Formulation is a robust semi-constrained optimal transport (RSOT) problem instance
- ✓ Approximate the optimal solution using the robust semi-Sinkhorn algorithm [NeurIPS, 2021]

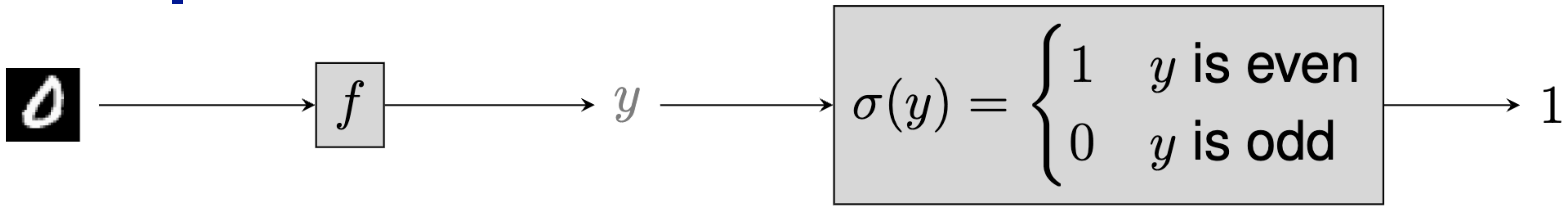
$$\min_{\mathbf{P}' \in \mathbb{R}_+^{n \times c}, \mathbf{P}'\mathbf{1}_c = \mathbf{1}_n} \langle -\log(\mathbf{P}), \mathbf{P}' \rangle + \tau KL(\mathbf{P}'\mathbf{1}_n || n\hat{r}) + \eta H(\mathbf{P}')$$

Entropic regularization term to find solutions in PTIME

Mitigating Imbalances During Training

Problem: We will first focus on the case where we have one input instance at a time

Example

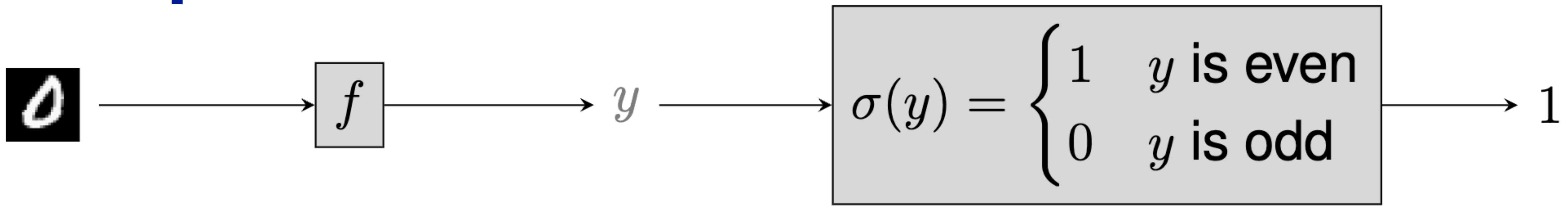


Training sample: (, {0,2,4,6,8})

Mitigating Imbalances During Training

Problem: We will first focus on the case where we have one input instance at a time

Example



Training sample: (, {0,2,4,6,8})

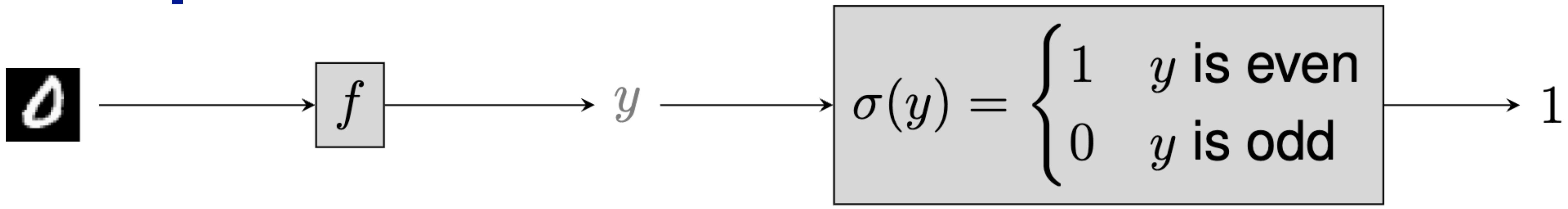
Prediction matrix $\mathbf{P}$

$$
\begin{array}{c}
y=0 \cdots\cdots y=9 \\
\begin{pmatrix}
0.1 & \cdots & 0.05 \\
\vdots & & \vdots \\
0.7 & \cdots & 0.01 \\
\vdots & & \vdots \\
0.01 & \cdots & 0.8
\end{pmatrix}
\end{array}
$$

Mitigating Imbalances During Training

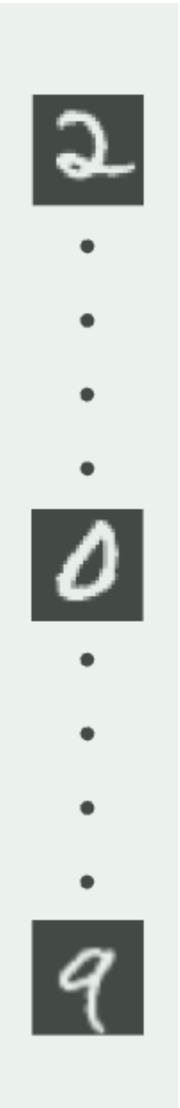
Problem: We will first focus on the case where we have one input instance at a time

Example



Training sample: (, {0,2,4,6,8})

Prediction matrix $\mathbf{P}$



$$\begin{array}{c} y=0 \;\cdots\cdots\; y=9 \\ \begin{pmatrix} 0.1 & \cdots & 0.05 \\ \vdots & & \vdots \\ 0.7 & \cdots & 0.01 \\ \vdots & & \vdots \\ 0.01 & \cdots & 0.8 \end{pmatrix} \end{array}$$

Rationale: Given a (gold) label distribution $\hat{r}$, correct the predictions $\mathbf{P}$ to $\mathbf{Q}$, so that $\mathbf{Q}$ adheres to $\hat{r}$.

Challenges:

✓The technique should be lightweight

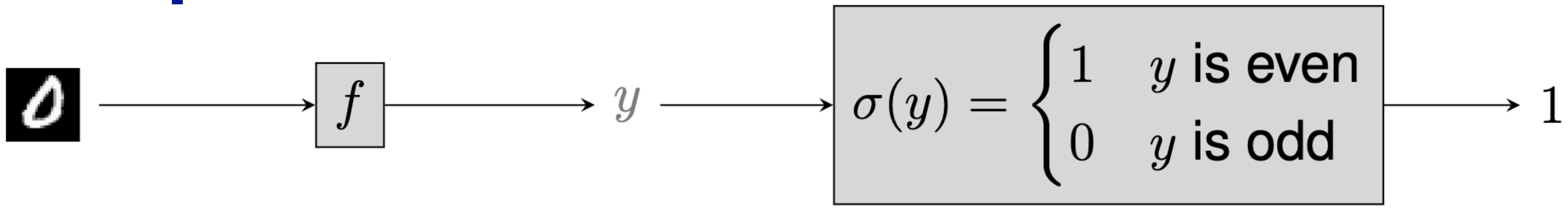
✓$\mathbf{Q}$ should be close enough to $\mathbf{P}$

✓The predictions should satisfy σ

Mitigating Imbalances During Training

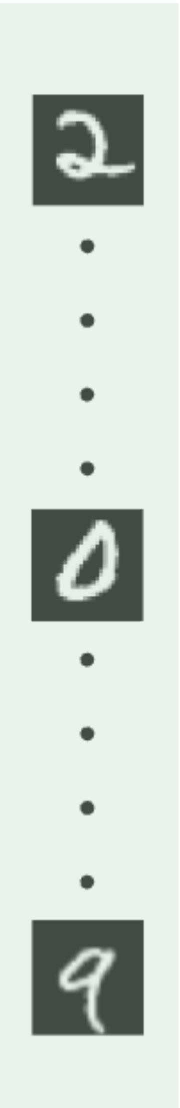
Problem: We will first focus on the case where we have one input instance at a time

Example



Training sample: (, {0,2,4,6,8})

Prediction matrix $\mathbf{P}$



$$
\begin{array}{c}
y=0 \cdots\cdots y=9 \\
\begin{pmatrix}
0.1 & \cdots & 0.05 \\
\vdots & & \vdots \\
0.7 & \cdots & 0.01 \\
\vdots & & \vdots \\
0.01 & \cdots & 0.8
\end{pmatrix}
\end{array}
$$

$$\min_{\mathbf{Q}} \langle \mathbf{Q}, -\log(\mathbf{P}) \rangle$$

s.t. a cell should be empty if not a valid label

Challenges:

✓The technique should be lightweight

✓$\mathbf{Q}$ should be close enough to $\mathbf{P}$

✓The predictions should satisfy σ

Mitigating Imbalances During Training

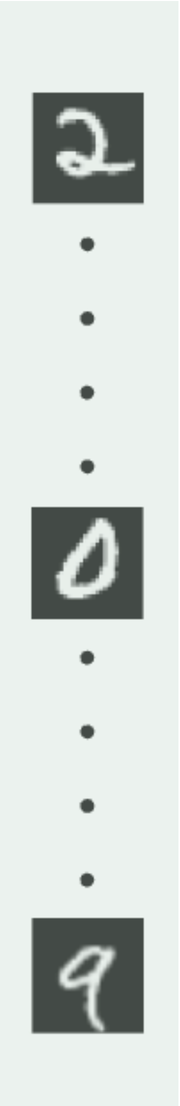
Problem: We will first focus on the case where we have one input instance at a time

Example

$$\longrightarrow \boxed{f} \longrightarrow y \longrightarrow \boxed{\sigma(y) = \begin{cases} 1 & y \text{ is even} \\ 0 & y \text{ is odd} \end{cases}} \longrightarrow 1$$

Training sample: (, {0,2,4,6,8})

Prediction matrix $\mathbf{P}$



$$\begin{array}{c} y = 0 \cdots\cdots y = 9 \\ \begin{pmatrix} 0.1 & \cdots & 0.05 \\ \vdots & & \vdots \\ 0.7 & \cdots & 0.01 \\ \vdots & & \vdots \\ 0.01 & \cdots & 0.8 \end{pmatrix} \end{array}$$

Question: Do we need additional constraints?

$$\min_{\mathbf{Q}} \langle \mathbf{Q}, -\log(\mathbf{P}) \rangle$$

s.t. a cell should be empty if not a valid label

Challenges:

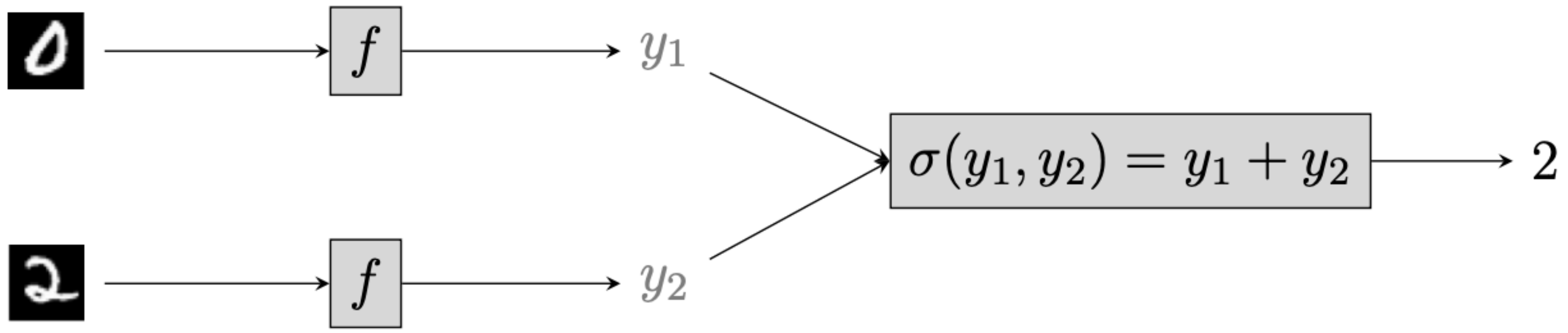
✓The technique should be lightweight

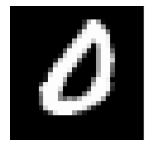
✓$\mathbf{Q}$ should be close enough to $\mathbf{P}$

✓The predictions should satisfy σ

Mitigating Imbalances During Training

Problem: We will first focus on the generic case



Training sample: ((,) {(0,2), (2,0), (1,1)})

Prediction matrix $\mathbf{P}_1$

$$\begin{array}{ccc} y=0 & \cdots & y=9 \\ \end{array}$$
$$\begin{pmatrix} 0.1 & \cdots & 0.05 \\ \vdots & & \vdots \\ 0.7 & \cdots & 0.01 \\ \vdots & & \vdots \\ 0.01 & \cdots & 0.8 \end{pmatrix}$$

Prediction matrix $\mathbf{P}_2$

$$\begin{array}{ccc} y=0 & \cdots & y=9 \\ \end{array}$$
$$\begin{pmatrix} 0.02 & \cdots & 0.8 \\ \vdots & & \vdots \\ 0.3 & \cdots & 0.07 \\ \vdots & & \vdots \\ 0.2 & \cdots & 0.005 \end{pmatrix}$$

Mitigating Imbalances During Training

Prediction matrix $\mathbf{P}_1$

$$\begin{array}{c} y=0 \cdots\cdots y=9 \\ \begin{pmatrix} 0.1 & \cdots\cdots & 0.05 \\ \vdots & & \vdots \\ 0.7 & \cdots\cdots & 0.01 \\ \vdots & & \vdots \\ 0.01 & \cdots\cdots & 0.8 \end{pmatrix} \end{array}$$

Prediction matrix $\mathbf{P}_2$

$$\begin{array}{c} y=0 \cdots\cdots y=9 \\ \begin{pmatrix} 0.02 & \cdots\cdots & 0.8 \\ \vdots & & \vdots \\ 0.3 & \cdots\cdots & 0.07 \\ \vdots & & \vdots \\ 0.2 & \cdots\cdots & 0.005 \end{pmatrix} \end{array}$$

Rationale: Given a (gold) label distribution $\hat{r}$, correct the predictions $\mathbf{P}_i$ to $\mathbf{Q}_i$, so that $\mathbf{Q}_i$ adheres to $\hat{r}$.

Challenges:

✓The technique should be lightweight

✓$\mathbf{Q}_i$ should be close enough to $\mathbf{P}_i$

✓$\mathbf{Q}_i$ should not strictly abide to $\hat{r}$ (to tolerate noise)

✓The predictions should satisfy σ

Mitigating Imbalances During Training

Challenges:

✓The technique should be lightweight

✓$\mathbf{Q}_i$ should be close enough to $\mathbf{P}_i$

✓$\mathbf{Q}_i$ should not strictly abide to $\hat{r}$ (to tolerate noise)

✓The predictions should satisfy σ

Reduction to **integer linear programming**

Integer linear programming formulation of NeSy

objective

$$\min_{(\mathbf{Q}_1,\ldots,\mathbf{Q}_M)} \sum_{i=1}^{M} \langle -\log(\mathbf{P}_i), \mathbf{Q}_i \rangle,$$

s.t.

$$\begin{array}{rll}
\sum_{t=1}^{R_\ell}[\alpha_{\ell,t}] & \geq 1, & \ell \in [n] \\
-|\varphi_{\ell,t}|[\alpha_{\ell,t}] + \sum_{k=1}^{|\varphi_{\ell,t}|}[\varphi_{\ell,t,k}] & \geq 0, & \ell \in [n], t \in [R_\ell] \\
-\sum_{k=1}^{|\varphi_{\ell,t}|}[\varphi_{\ell,t,k}] + [\alpha_{\ell,t}] & \geq (1 - |\varphi_{\ell,t}|), & \ell \in [n], t \in [R_\ell] \\
\sum_{j=1}^{c}[q_{\ell,i,j}] & = 1, & \ell \in [n], i \in [M] \\
[q_{\ell,i,j}] & \in [0,1], & \ell \in [n], i \in [M], j \in [c] \\
|\mathbf{Q}_i \cdot \mathbf{1}_n - n\hat{\mathbf{r}}| & \leq \epsilon, & i \in [M]
\end{array}$$

NeSy to Integer Linear Programming

Constraints in NeSy can be expressed as formulas in **disjunctive normal (DNF)** form

1st image is **1**, 2nd image is **1**

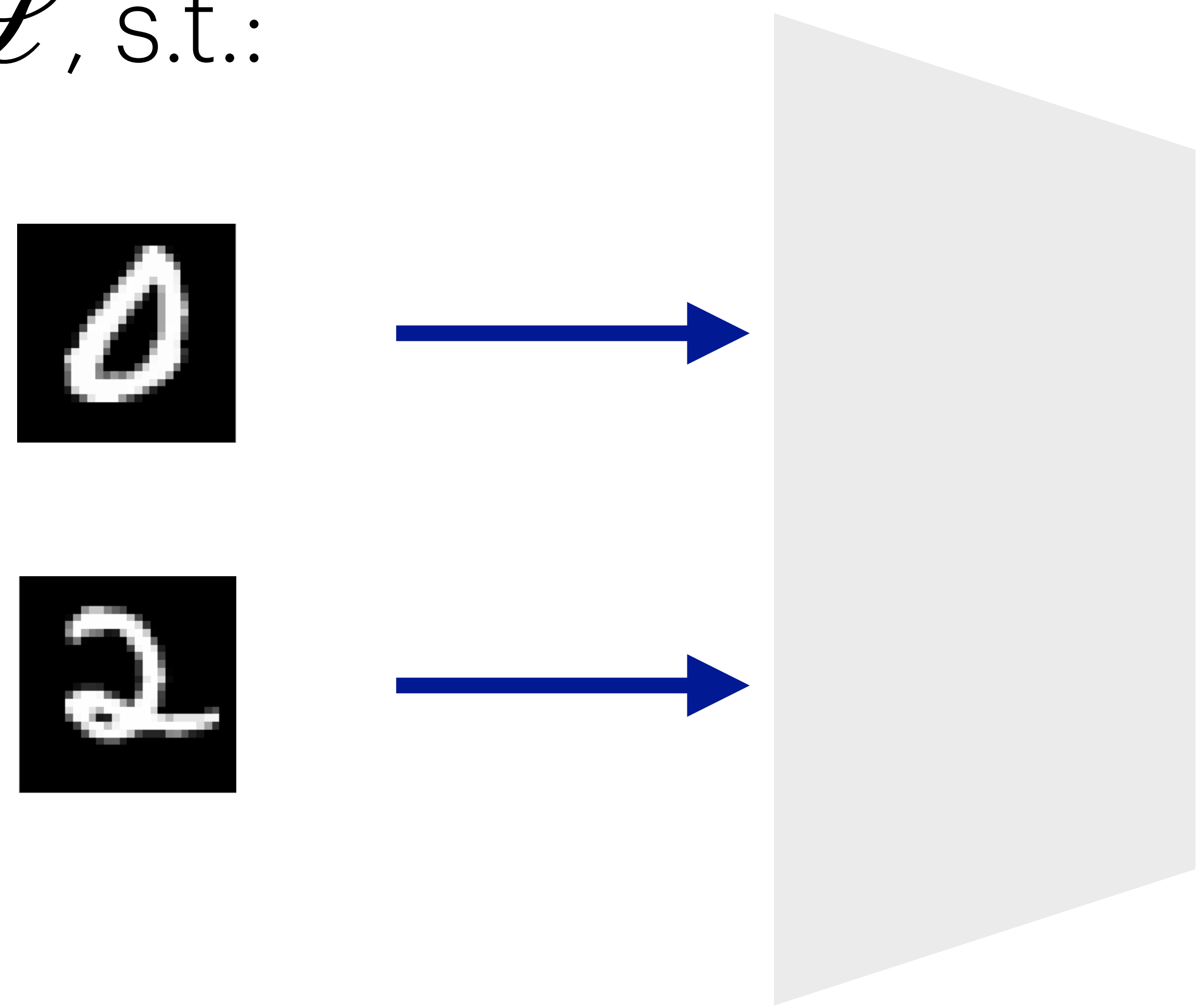
Example: Convert $(\mathbf{x}_{1,0} \wedge \mathbf{x}_{2,2}) \vee (\mathbf{x}_{1,1} \wedge \mathbf{x}_{2,1}) \vee (\mathbf{x}_{1,2} \wedge \mathbf{x}_{2,0})$

1st image is **0**, 2nd image is **2**

Goal: Train the DNN knowing that the input MNIST images sum up to **2**.

Goal: Convert a Boolean constraint ϕ to a set of linear equations $\mathscr{L}$, s.t.:

✓ ϕ becomes true if and only if $\mathscr{L}$ is satisfied



NeSy to Integer Linear Programming

Example: Convert $(\mathbf{x}_{1,0} \wedge \mathbf{x}_{2,2}) \vee (\mathbf{x}_{1,1} \wedge \mathbf{x}_{2,1}) \vee (\mathbf{x}_{1,2} \wedge \mathbf{x}_{2,0})$

Step 1: Convert to CNF (via the Tseytin transformation), i.e., formulas of this form:

$\Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$, each Φ_i is a disjunction of (negated) Boolean variables

Step 2: Translate each clause to a linear constraint using some predefined rules

Rules to Convert a Boolean Constraint to an ILP

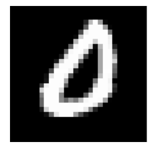
Boolean constraint	**Linear constraint**	
$X_1 \land X_2 \land \ldots \land X_n$	$\sum_i [X_i] = n$	
$X_1 \lor X_2 \lor \ldots \lor X_n$	$\sum_i [X_i] \geq 1$	Non-Mutually exclusive
$X_1 \lor X_2 \lor \ldots \lor X_n$	$\sum_i [X_i] = 1$	Mutually exclusive
$\neg X$	$1 - [X]$	

Rules to Convert a Boolean Constraint to an ILP

Boolean constraint	**Linear constraint**	
$X_1 \wedge X_2 \wedge \ldots \wedge X_n$	$\sum_i [X_i] = n$	
$X_1 \vee X_2 \vee \ldots \vee X_n$	$\sum_i [X_i] \geq 1$	Non-Mutually exclusive
$X_1 \vee X_2 \vee \ldots \vee X_n$	$\sum_i [X_i] = 1$	Mutually exclusive
$\neg X$	$1 - [X]$	

Test: Convert the CNF formula $X_1 \wedge (\neg X_2 \vee \neg X_3)$ into the corresponding integer linear program

Computing the Marginals of the Hidden Labels

- ✓**Statistically consistent** technique to compute the gold hidden label ratios $\boldsymbol{r}$
  - ✓**Problem (via example)**
    - ✓**Given** samples of the form (, , max = 9)
    - ✓**Compute** the distribution of the instances in each class

Not covered in this lecture

Kaifu Wang, Efthymia Tsamoura, and Dan Roth. On characterizing and mitigating imbalances in multi-instance weak supervision. CoRR, abs/2407.10000, 2025

Outline of Today's Lecture

✓**Introduction to Learning Imbalances**

✓Learning Imbalances in traditional ML

✓Learning Imbalances in NeSy

✓**Mitigating Learning Imbalances**

✓Testing time techniques

✓Reduction to robust optimal transport

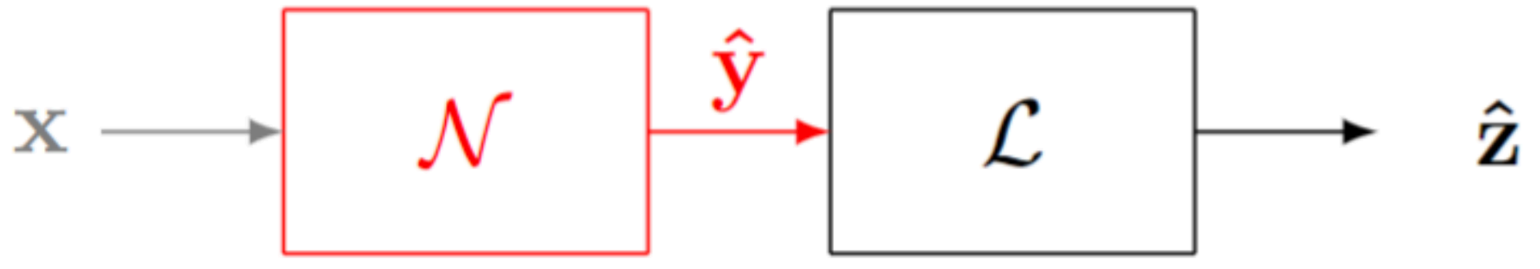
✓Training time techniques

✓Reduction to integer linear programming

✓**Teacher-Student NeSy**

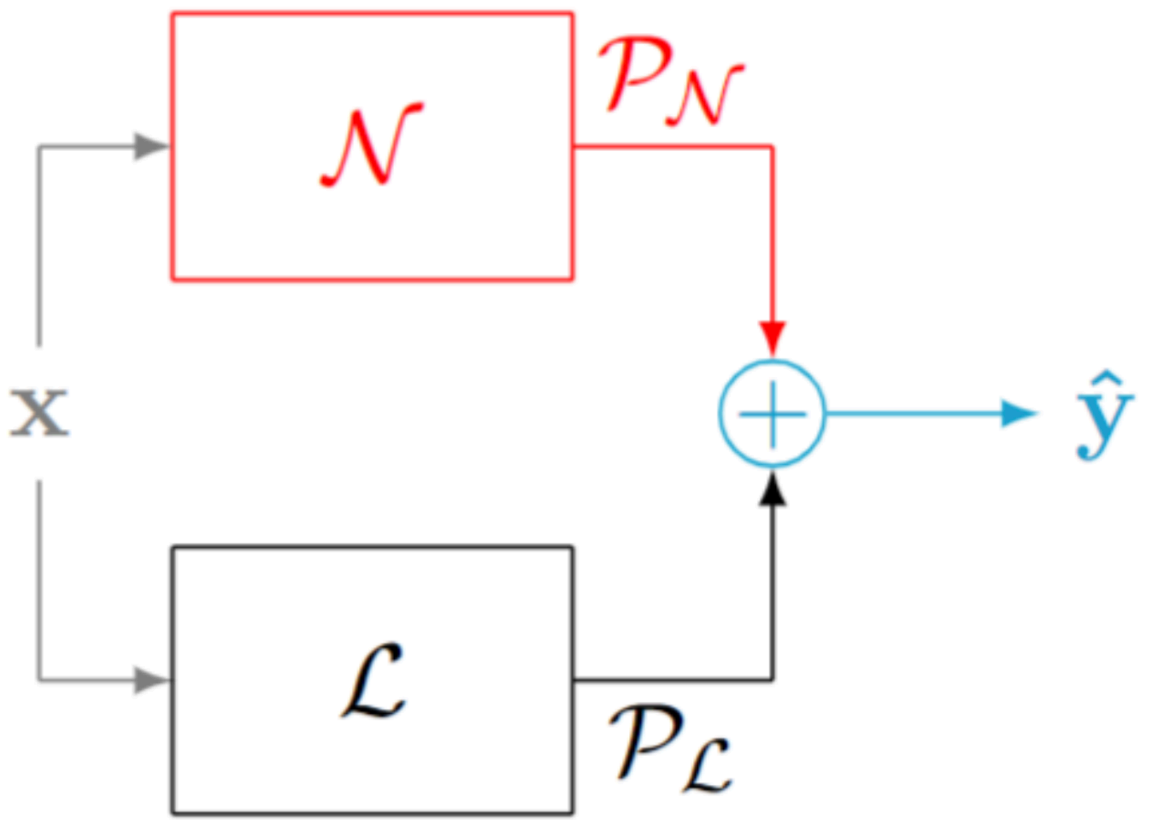
Types of Integration

Stratified



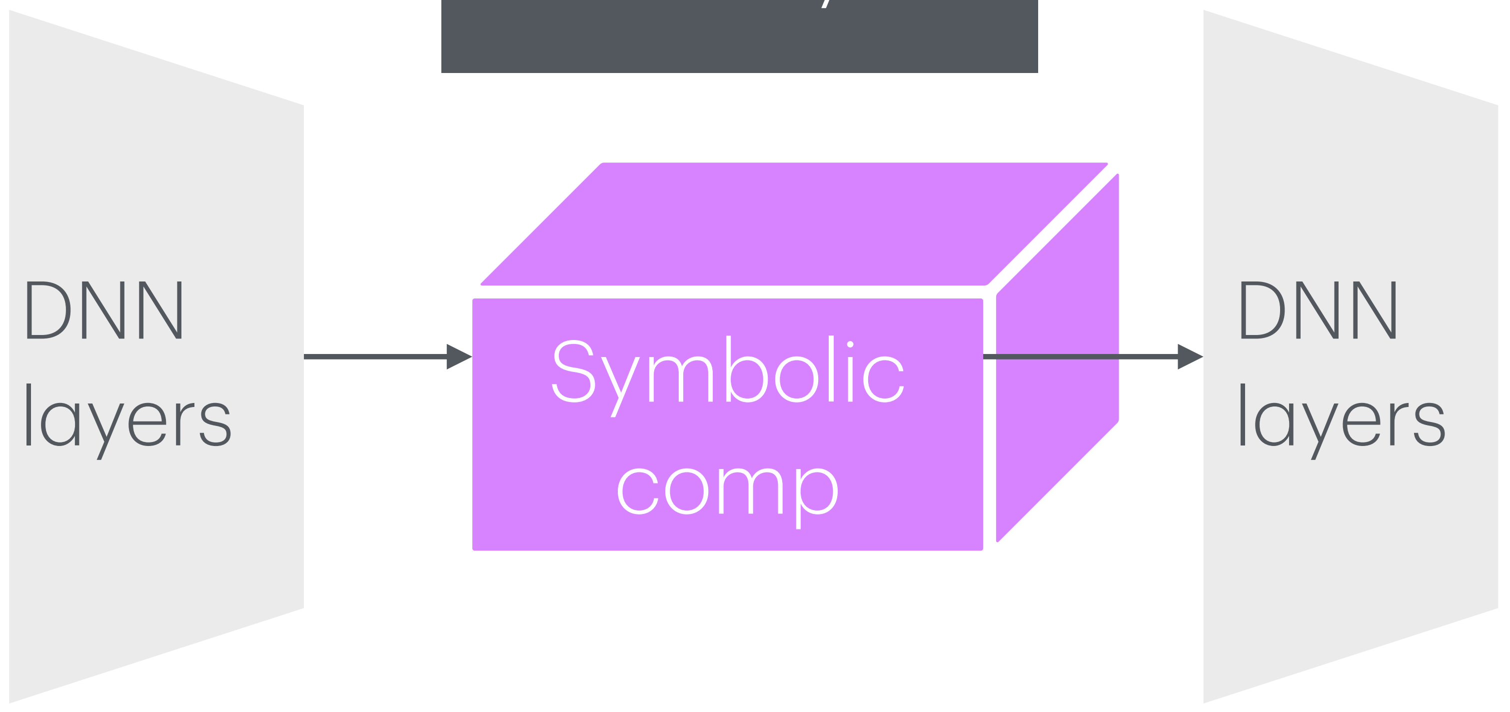
- ✓DeepProbLog [NeurIPS 2018]
- ✓ABL [NeurIPS 2019]
- ✓NeurASP [IJCAI 2020]
- ✓NeuroLog [AAAI 2021]
- ✓Scallop [NeurIPS 2021]
- ✓ENT [ICLR 2023]
- ✓DeepSoftLog [NeurIPS 2023]
- ✓ISED [NeurIPS 2023]
- ✓Dolphin [arXiv 2024]

Teacher-Student



- ✓T-S, ACL [EMNLP 2016]
- ✓DPL [EMNLP 2018]
- ✓Concordia [ICML 2023]

As a layer



- ✓MIPaaL [AAAI 2020]
- ✓BB-backprop [ICLR 2020]
- ✓CombOptNet [ICML 2021]
- ✓SurCo [ICML 2023]
- ✓GenCO [ICML 2024]

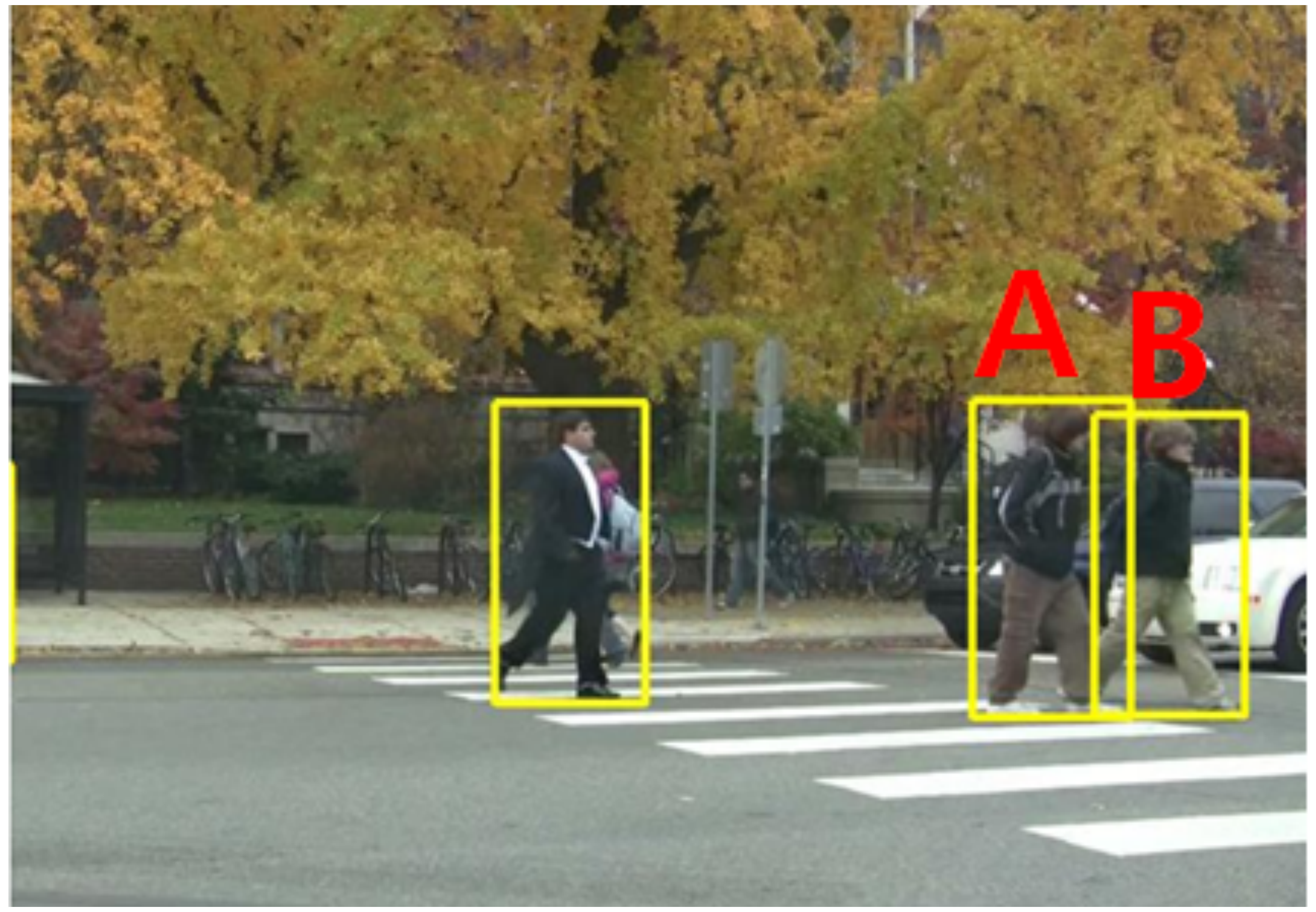
Knowledge Distillation

Traditional ML: Distill knowledge from a complex deep network to a small one

NeSy: Distill knowledge from a **logical theory** into a **deep network**

Knowledge Distillation: State-of-the-art

- ✓ Inability to express complex relationships between the input and the output data, (Hu et al., 2016a;b), (Wang & Poon, 2018)
- ✓ Problems with the optimization leading to vacuum supervision



Task. Understand the activity of a group of actors in a video

Rules of this form, could not be supported

$$\text{DOING(A, activity)} \wedge \text{CLOSE(A,B)} \xrightarrow{0.75} \text{DOING(B, activity)}$$

Concordia

✓ Supports general uncertain theories in first-order

✓ Offers plug-and-play interface

✓ Integrates symbolic with neural components without relying on the independence assumption

✓ Can use the deep network predictions as priors

Leon Jonathan Feldstein, Jurcius Modestas and Efthymia Tsamoura. Parallel neurosymbolic integration with Concordia. In ICML, 2023.

Concordia: Architecture

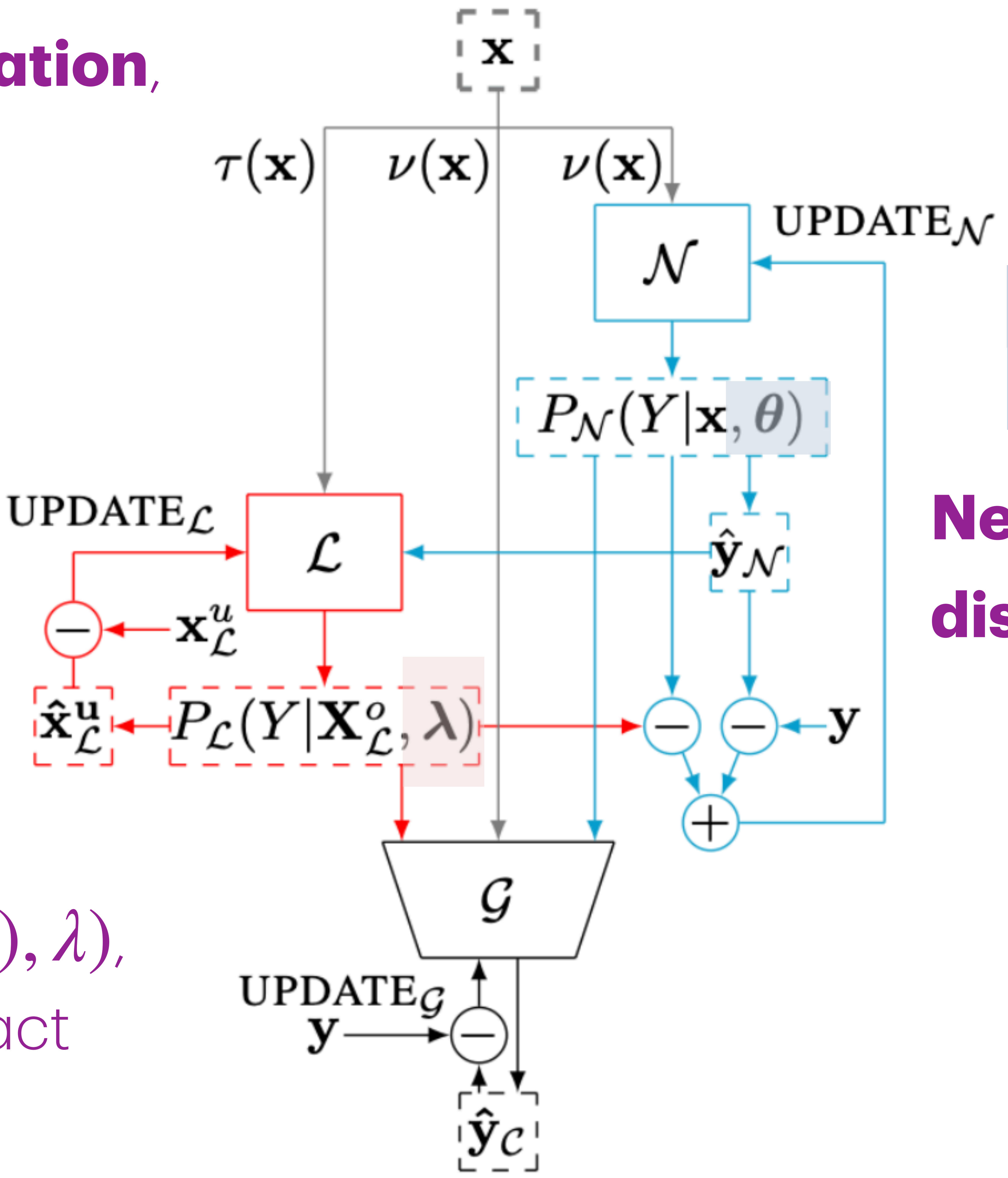
Raw data

Symbolic representation, e.g., DOING(A, joking)

Neural representation, e.g., tensor representation of a bounding box

(Unknown) parameters of the network

(Unknown) weights of the logical theory



Network as a probability distribution , $P_{\mathcal{N}}(X\,|\,\nu(\mathbf{x}), \theta)$

Logic as a probability distribution $P_{\mathcal{L}}(X\,|\,\tau(\mathbf{x}), \lambda)$, e.g., probability of this fact DOING(A, joking)

Key Ideas: (1) Represent the problem in symbolic form. (2) Treat logic as a conditional distribution

Concordia: Architecture

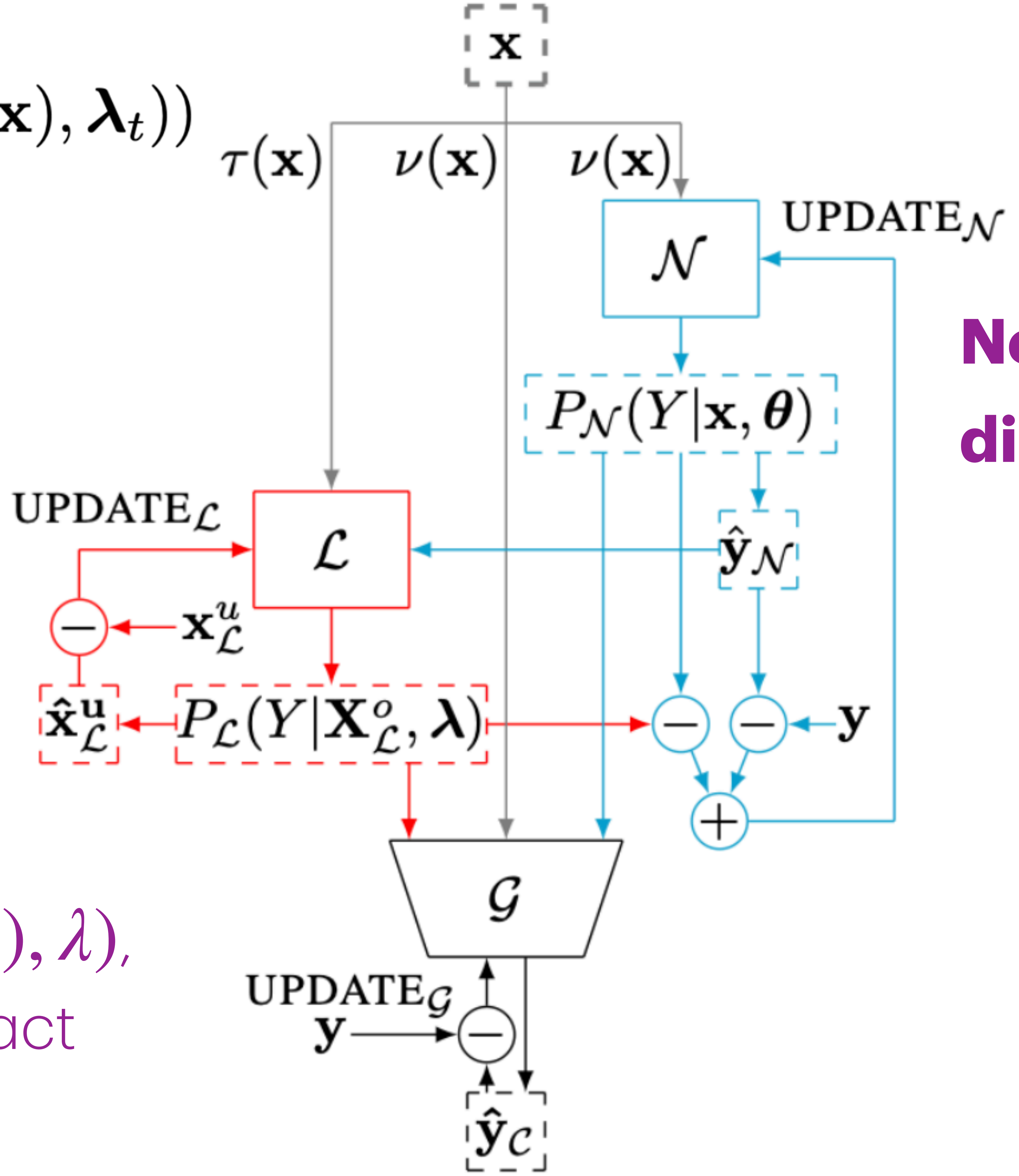
Updating θ at step $t+1$**:**

Predicted outputs, e.g., Person A is running

Gold output: Person A is standing

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \ell(\hat{\mathbf{y}}, \mathbf{y}) + KL(P_{\mathcal{N}}(Y|\mathbf{X}_{\mathcal{N}} = \nu(\mathbf{x}), \boldsymbol{\theta}_t) || P_{\mathcal{L}}(Y|\mathbf{X}^o_{\mathcal{L}} = \tau(\mathbf{x}), \boldsymbol{\lambda}_t))$$

Question: Can Concordia support unsupervised learning?



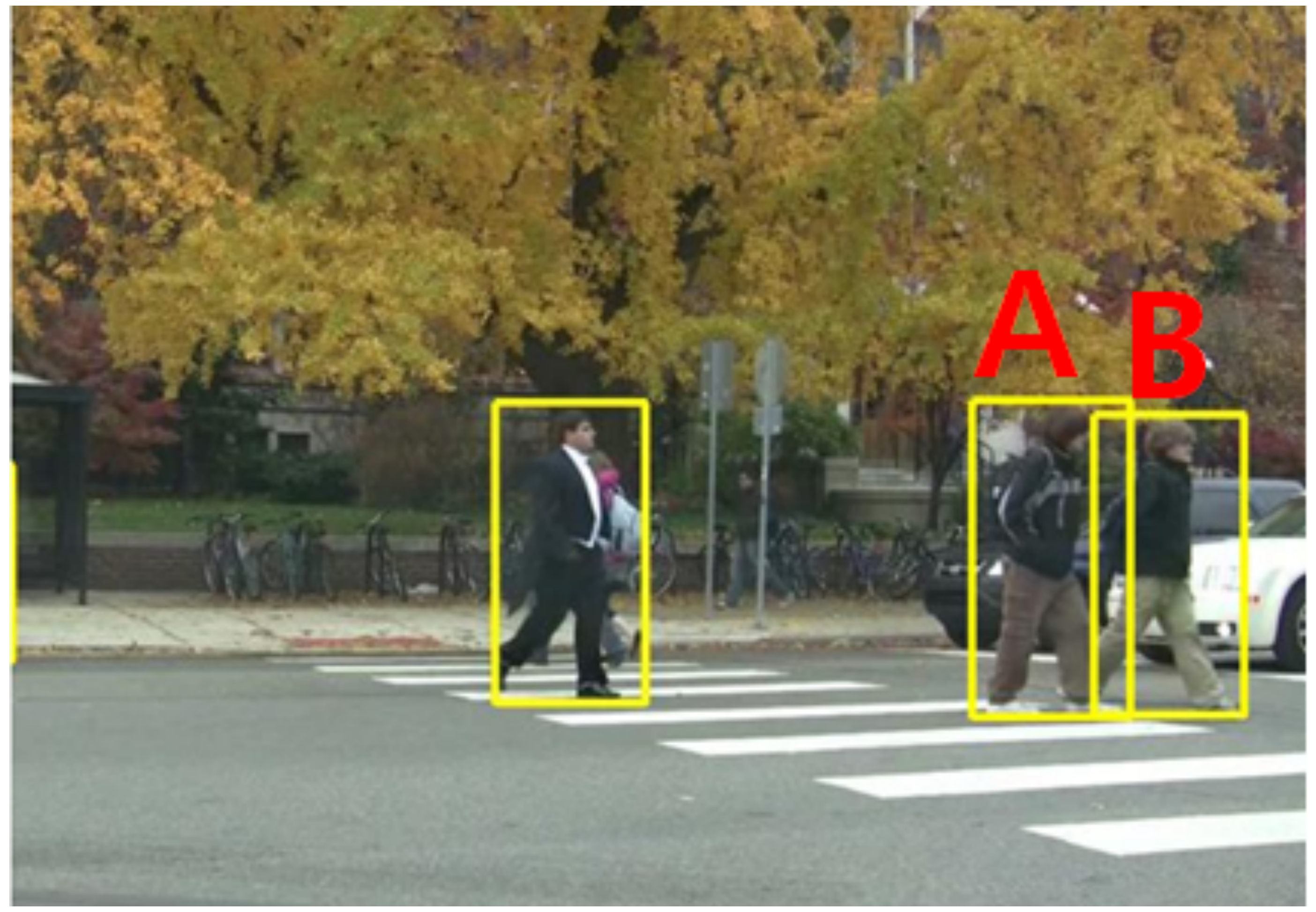
Network as a probability distribution, $P_{\mathcal{N}}(X\,|\,\nu(\mathbf{x}), \theta)$

Logic as a probability distribution $P_{\mathcal{L}}(X\,|\,\tau(\mathbf{x}), \lambda)$, e.g., probability of this fact DOING(A, joking)

Logic as a Probability Distribution

- ✓ Many uncertain logics give this ability, e.g.,
  - ✓ ProbLog — to be covered in subsequent lectures
  - ✓ Markov Logic Networks
  - ✓ Probabilistic Soft Logic

Empirical Results



Task. Understand the activity of a group of actors in a video

Accuracy over 5 runs

Model	Avg (%)	Max (%)	Min (%)
ACD+$\mathcal{L}$ [17]	86.00	-	-
MobileNet	90.07	91.36	89.61
IARG(MobileNet) [14]	90.18	92.39	87.55
Concordia(MobileNet, $\mathcal{L}$)	**90.73**	**93.19**	**89.54**
Inception	89.72	91.83	86.84
IARG(Inception) [14]	88.88	91.67	85.33
Concordia(Inception, $\mathcal{L}$)	**92.75**	**93.34**	**92.31**

$\lambda_1 : \text{FRAME}(B, F) \wedge \text{FLABEL}(F, A) \rightarrow \text{DOING}(B, A)$ The activity of an actor is the same with the activity of the frame

$\lambda_2 : \text{DOING}(B_1, A) \wedge \text{CLOSE}(B_1, B_2) \rightarrow \text{DOING}(B_2, A)$ Two actors close to each other perform the same activity

$\lambda_3 : \text{SEQUENCE}(B_1, B_2) \wedge \text{CLOSE}(B_1, B_2) \rightarrow \text{SAME}(B_1, B_2)$

$\lambda_4 : \text{DOING}(B_1, A) \wedge \text{SAME}(B_1, B_2) \rightarrow \text{DOING}(B_2, A)$ If the actor within two bounding boxes is the same, then she likely performs the same activity

$\lambda_5 : \text{DNN}(B, A) \rightarrow \text{DOING}(B, A)$

The actor does what the networks predicts

Outline of Today's Lecture

- ✓**Introduction to Learning Imbalances**
  - ✓Learning Imbalances in traditional ML
  - ✓Learning Imbalances in NeSy
- ✓**Mitigating Learning Imbalances**
  - ✓Testing time techniques
    - ✓Reduction to robust optimal transport
  - ✓Training time techniques
    - ✓Reduction to integer linear programming
- ✓**Teacher-Student NeSy**

More Cool Research

- ✓**Trigger Graphs:** Exact & scalable probabilistic reasoning over hundreds over millions of facts [VLDB 2021, SIGMOD 2023]
  - ✓Relies on redundancy-free reasoning and provenance circuits
- ✓**SPECTRUM:** Rule mining under formal guarantees in the order of seconds over millions of facts [AAAI 2023, arXiv 2025]
  - ✓Involves addressing a tough problem in graph theory
- ✓**Concordia:** Neurosymbolic teacher-student learning [ICML 2023]
  - ✓First over general first-order theories
- ✓**SO-Chase:** Goal-driven QA over expressive ontologies under formal guarantees [AAAI 2018, arXiv 2024]
  - ✓Fixes incompleteness errors in relevant SOTA
- ✓**NGP:** Neurosymbolic scene graph generation [AAAI 2023]
  - ✓SOTA over all deep neural baselines up to 2024

Thank you!

- ✓ More info can be found at: **https://tsamoura.github.io/**