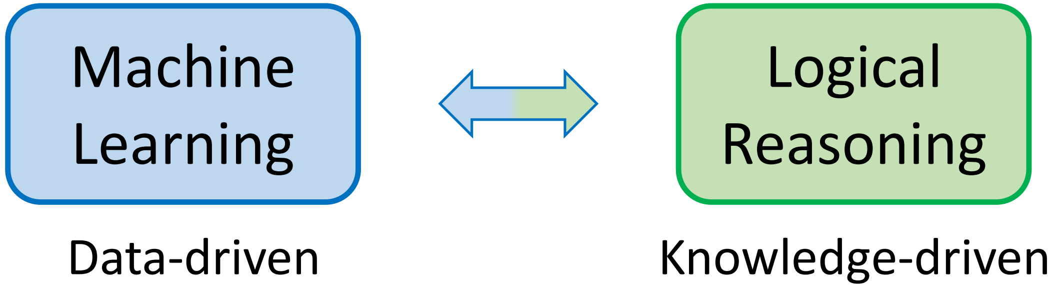
# Fast Abductive Learning by Similarity-based Consistency Optimization

**Yu-Xuan Huang**, Wang-Zhou Dai, Le-Wen Cai, Stephen Muggleton, Yuan Jiang

# Introduction

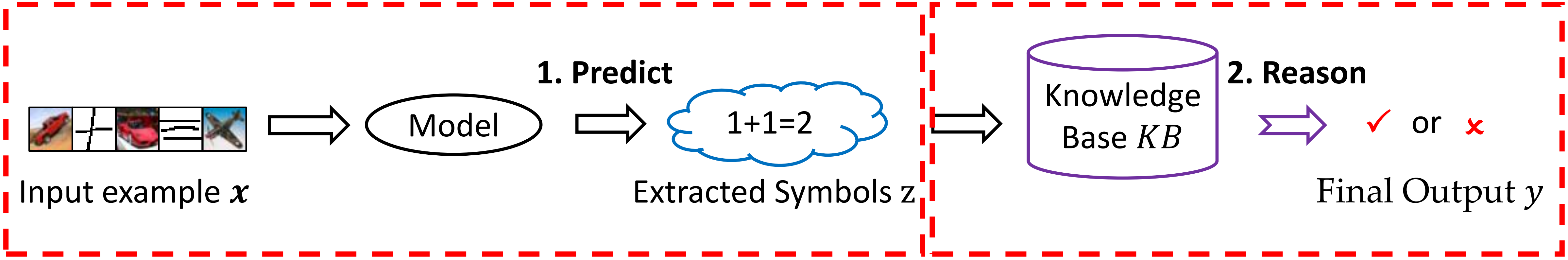
# Integration of machine learning and logical reasoning



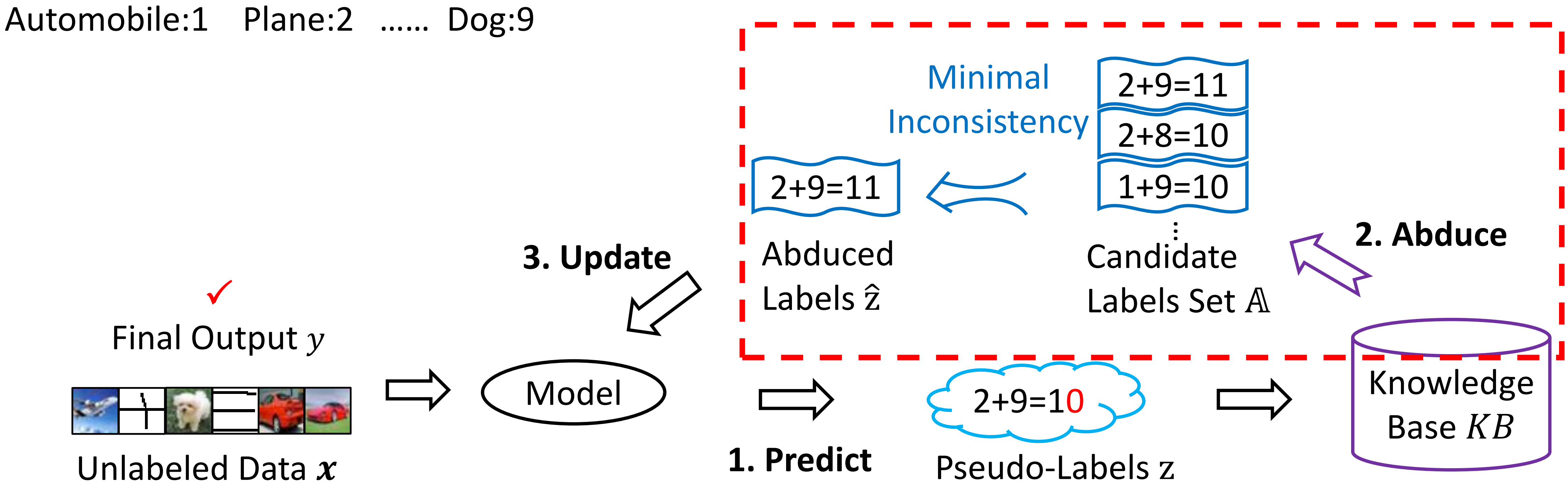

1. End-to-end models
   - Approximate logical calculus with differentiable functions
   - Demand a large number of labeled data
2. Hybrid modeling of dual systems
   - Abductive Learning (ABL)

# Abductive Learning -- Inference

Automobile:1    Plane:2    ……  Dog:9



# Abductive Learning -- Learning




- Leverage full-featured logical reasoning to reduce the requirement for labeled data

# Abduction

- Abduction (Abductive reasoning): a basic form of logical inference that **seeks the most likely explanation** for observations based on background knowledge

- A non-deterministic process that may have **multiple answers**

# Consistency measure

- Previous:

2. Minimal inconsistency based on NN's output

2+9=11

2+8=10

⋮

Consistent label set

1. Consistency checking

9+2=03

0+8=10

2+9=11

2+8=10

⋮

2+9=11

Abduced Labels $\hat{z}$

9+2=03

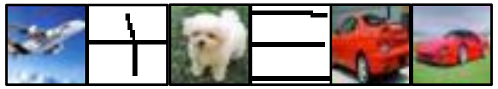
0+8=10

2+9=11

2. Consistency checking

1. Random walk by NN's (unreliable) probability

- Do not consider feature space
- Unreliable when under-trained



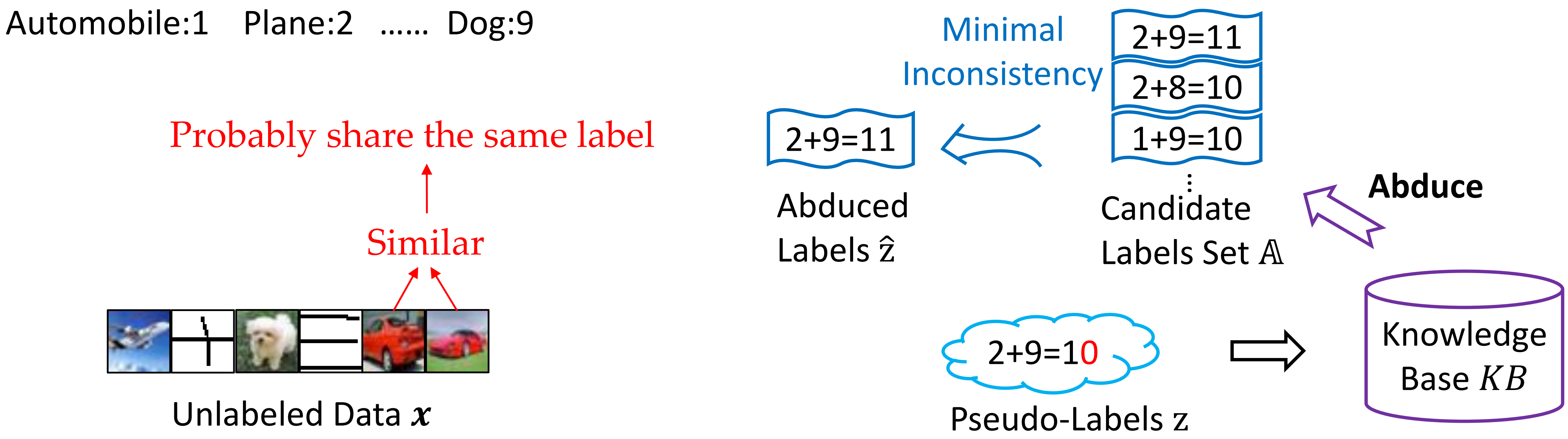
- Good measure:

Model's output + Background knowledge + Input features

# Similarity-based Consistency Measure

# Similarity-based Consistency Measure

- Idea:
  - Samples in the **same** category are **similar** in feature space
  - Samples of **different** classes are **dissimilar**



# Consistency Optimization Problem

- Given the input data $\boldsymbol{x}$, final output $y$, candidate labels set $\mathbb{A}$
- Problem formalization

$$\max_{\bar{\boldsymbol{z}} \in \mathbb{A}} \quad \text{SimilarityScore}(\boldsymbol{x}, \bar{\boldsymbol{z}})$$

- Consistency

$$\text{SimilarityScore}(\boldsymbol{x}, \bar{\boldsymbol{z}}) = \frac{1}{|\boldsymbol{x}|} \sum_{x_i \in \boldsymbol{x}} \left( \text{InterclassDis}(x_i, \bar{\boldsymbol{z}}) - \text{IntraclassDis}(x_i, \bar{\boldsymbol{z}}) \right)$$

$$\text{InterclassDis}(x_i, \bar{\boldsymbol{z}}) = \frac{1}{|\mathbb{D}_{i,\bar{\boldsymbol{z}}}|} \sum_{x_j \in \mathbb{D}_{i,\bar{\boldsymbol{z}}}} \text{Dis}(x_i, x_j),$$

the set of instances whose labels are *different* from $x_i$'s

$$\text{IntraclassDis}(x_i, \bar{\boldsymbol{z}}) = \frac{1}{|\mathbb{S}_{i,\bar{\boldsymbol{z}}}|} \sum_{x_j \in \mathbb{S}_{i,\bar{\boldsymbol{z}}}} \text{Dis}(x_i, x_j),$$

the set of instances whose labels are the *same* as $x_i$'s

# Similarity

**Final problem**

$$\max_{\bar{z} \in \mathbb{A}} \quad \frac{1}{|\boldsymbol{x}|} \sum_{x_i \in \boldsymbol{x}} \left( \frac{1}{|\mathbb{D}_{i,\bar{z}}|} \sum_{x_j \in \mathbb{D}_{i,\bar{z}}} \operatorname{Dis}(x_i, x_j) - \frac{1}{|\mathbb{S}_{i,\boldsymbol{z}}|} \sum_{x_j \in \mathbb{S}_{i,\bar{z}}} \operatorname{Dis}(x_i, x_j) \right).$$

- The higher the similarity, the smaller the distance

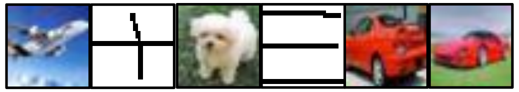
$$\operatorname{Dis}(x_i, x_j) = \operatorname{Distance}(\phi(x_i), \phi(x_j))$$

- $\phi$ is the feature map function: e.g., neural network for images or normalization function for tabular data
- We can obtain $\phi$ by unsupervised learning, or use perception classifier's embedding layer

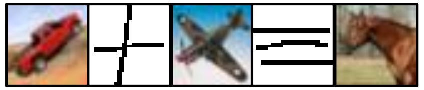
# Abductive Learning with Similarity (ABLSim)

# Borrow more samples

- It could be challenging to calculate the intra-class distance due to limited instances
- We borrow some more samples to conduct the abductive reasoning



Input examples $\boldsymbol{x}^{\langle 1 \rangle}$



Input examples $\boldsymbol{x}^{\langle 2 \rangle}$

Input examples $\boldsymbol{x}^{\langle 3 \rangle}$

no other sample whose revised label is 2

| Candidate Labels Set $\mathbb{A}^{\langle 1 \rangle}$ | Candidate Labels Set $\mathbb{A}^{\langle 2 \rangle}$ | Candidate Labels Set $\mathbb{A}^{\langle 3 \rangle}$ |
|---|---|---|
| $\boxed{2}$+9=11 | 1+3=4 | 1+1=2 |
| 2+8=10 | 1+2=3 | 1+2=3 |
| 1+9=10 | 0+4=4 | 2+1=3 |
| ⋮ | ⋮ | ⋮ |

# Borrow more samples

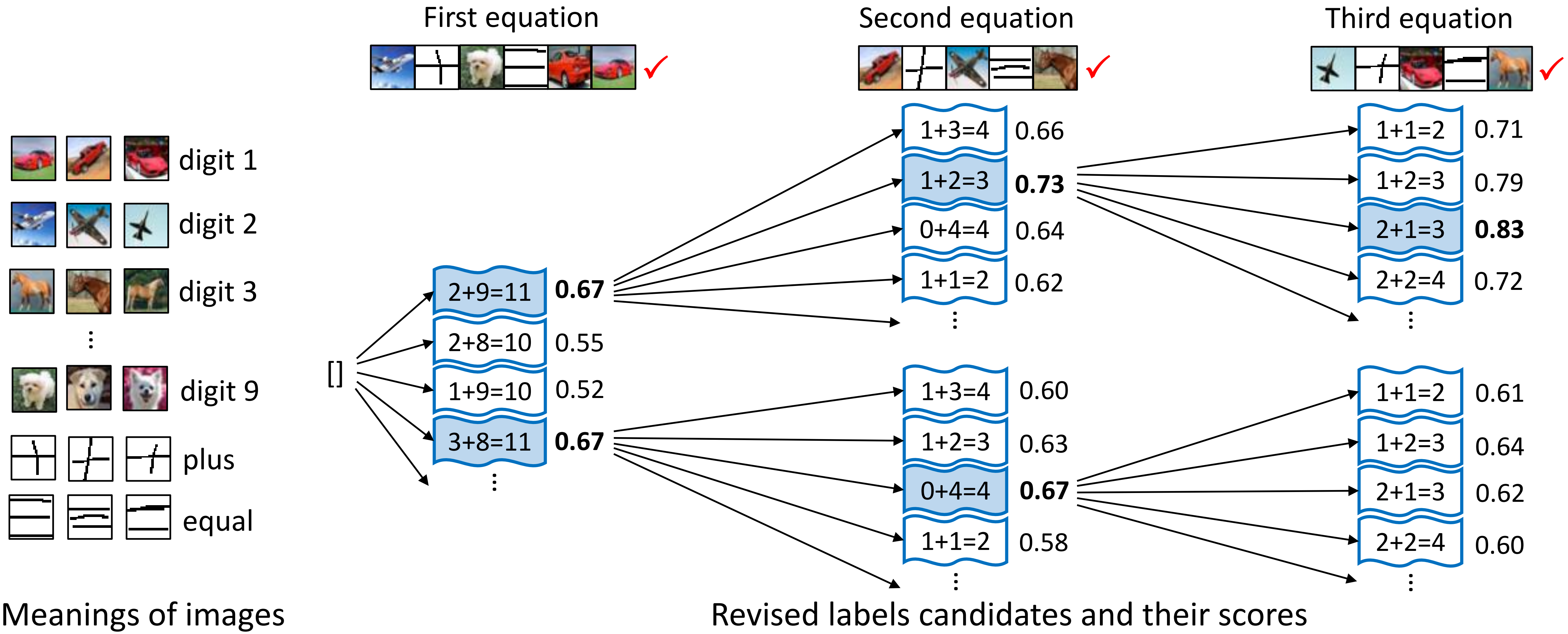
- The abduction problem

$$\begin{aligned} \max_{\bar{\boldsymbol{Z}}} \quad & \mathrm{Score}(\boldsymbol{X}, \bar{\boldsymbol{Z}}), \\ s.t. \quad & \boldsymbol{X} = (\boldsymbol{x}^{\langle 1 \rangle}, \boldsymbol{x}^{\langle 2 \rangle}, \cdots, \boldsymbol{x}^{\langle m \rangle}), \\ & \bar{\boldsymbol{Z}} \in \mathbb{A}^{\langle 1 \rangle} \times \mathbb{A}^{\langle 2 \rangle} \times \cdots \times \mathbb{A}^{\langle m \rangle}, \\ & \mathbb{A}^{\langle k \rangle} = \{\bar{\boldsymbol{z}} \mid KB \cup \bar{\boldsymbol{z}} \models y^{\langle k \rangle}\}. \end{aligned}$$

- Combinatorial optimization problem where the search space of grows exponentially with $m$
- ABLSim uses **beam search** to solve this optimization problem greedily

# Beam Search (Example)

- Beam width $b = 2$



# Beam Search (Algorithm)

**Algorithm 1** ABLSim Learning

**Input:** Unlabeled data $\boldsymbol{X} = (\boldsymbol{x}^{\langle 1 \rangle}, \boldsymbol{x}^{\langle 2 \rangle}, \cdots, \boldsymbol{x}^{\langle m \rangle})$; Final output $y = (y^{\langle 1 \rangle}, y^{\langle 2 \rangle}, \cdots, y^{\langle m \rangle})$;
Current model $f$; Knowledge base $KB$; Beam width $b$

**Output:** Model $f$

**for** $t = 1$ **to** $T$ **do**
    $\mathbb{A} \leftarrow []$ # the candidate labels
    **for** $k = 1$ **to** $m$ **do**
        $\boldsymbol{z}^{\langle k \rangle} \leftarrow f(\boldsymbol{x}^{\langle k \rangle})$ # generate pseudo-labels
        $\mathbb{A}^{\langle k \rangle} \leftarrow \text{Abduce}(KB, \boldsymbol{z}^{\langle k \rangle}, y^{\langle k \rangle})$ # abduce all consistent revised pseudo-labels
        $\mathbb{A} \leftarrow \mathbb{A} \times \mathbb{A}^{\langle k \rangle}$ # Cartesian product
        $\boldsymbol{x} \leftarrow \boldsymbol{X}[1:k]$
        $\boldsymbol{score} \leftarrow []$ # the score of each candidate labels
        **for** $\bar{z} \in \mathbb{A}$ **do**
            $\boldsymbol{score}.append(\text{Score}(\boldsymbol{x}, \bar{z}))$ # get the score of candidate labels according to Eq. (12)
        **end for**
        $\mathbb{A} \leftarrow \text{TopN}(\mathbb{A}, \boldsymbol{score}, b)$ # select the top-k score candidate labels
    **end for**
    $\bar{\boldsymbol{Z}} \leftarrow \text{TopN}(\mathbb{A}, \boldsymbol{score}, 1)$ # select the best candidate labels
    $f \leftarrow \text{Update}(f, \boldsymbol{X}, \bar{\boldsymbol{Z}})$ # update model using abduced labels $\bar{\boldsymbol{Z}}$
**end for**

- Could be accelerated by GPU and parallel computations

# Combing Different Consistency Measures

- The confidence score

$$\mathrm{ConfidenceScore}(\boldsymbol{x}, \bar{\boldsymbol{z}}) = \frac{1}{|\boldsymbol{x}|} \prod_{x_i \in \boldsymbol{x}} \mathrm{Confidence}(x_i, \bar{z}_i)$$

- The final score for ABLSim's consistency measure

$$\mathrm{Score}(\boldsymbol{x}, \bar{\boldsymbol{z}}) = \theta \cdot \mathrm{SimilarityScore}(\boldsymbol{x}, \bar{\boldsymbol{z}}) + (1-\theta) \cdot \mathrm{ConfidenceScore}(\boldsymbol{x}, \bar{\boldsymbol{z}})$$

Weighting coefficient

# Experiments

# Results

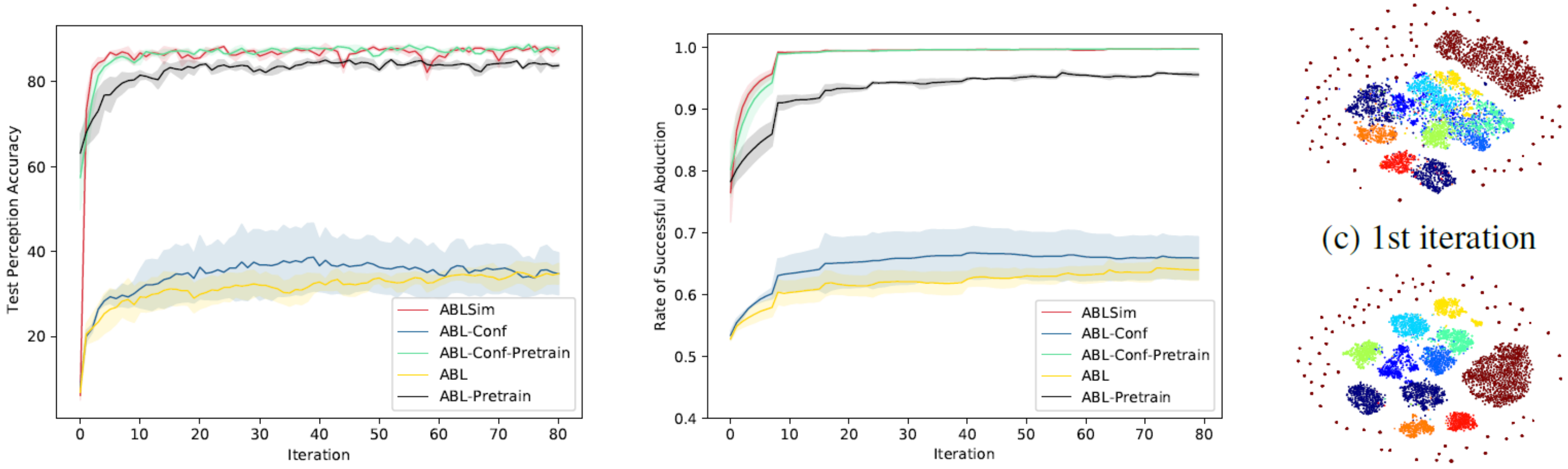
- **MNIST (CIFAR-10) Addition**
- **Handwritten Formula Recognition (HWF)**

| | Method | Addition | Addition (CIFAR) | HWF | HWF (CIFAR) |
|---|---|---|---|---|---|
| Acc / % | DeepProbLog | 96.5±0.5 | 21.6±1.7 | 32.2±0.6 | 15.2±2.6 |
| | NGS-dft | 39.9±54.1 | 38.7±35.1 | 99.6±0.2 | 23.8±6.3 |
| | NGS-opt | 98.5±0.3 | 88.7±0.8 | 99.6±0.2 | 66.0±14.5 |
| | ABLSim (ours) | **98.8±0.1** | **88.9±0.5** | **99.9±0.1** | **88.4±0.7** |
| Time / s | DeepProbLog | 396±3 | time out | time out | time out |
| | NGS-dft | time out | time out | 299±36 | time out |
| | NGS-opt | 46±4 | 6954±558 | 240±7 | time out |
| | ABLSim (ours) | **42±5** | **6066±79** | **130±4** | **7263±122** |

- ABLSim solves all tasks efficiently and achieves a higher accuracy than SOTA models

# Results

- **CIFAR-10 Decimal Equation Decipherment**




(a) Perception model test accuracy. (b) Rate of successful abduction. (c) 1st iteration (d) 10th iteration

Figure 3: Learning curves (a & b) and the t-SNE visualization of the learned embeddings (c & d).

- Converges much faster and achieves higher accuracy than other methods
- The embeddings of classes are improved after the neural net is updated with the abduced labels

# Results

- **Theft Judicial Sentencing**

Table 3: Micro-F1-score of the model, and MAE of the predicted sentence. The label rates are denoted as suffixes.

| KB | Method | F1 | MAE |
|---|---|---|---|
| N/A | PL-10 | 0.814 | 0.862 |
| N/A | Tri-10 | 0.812 | 0.840 |
| Full | SS-ABL-10 | 0.862 | 0.824 |
| Part | SS-ABL-10 | 0.833 | 0.835 |
| Part | ABLSim-10 | **0.851** | **0.828** |
| N/A | PL-50 | 0.858 | 0.832 |
| N/A | Tri-50 | 0.861 | 0.810 |
| Full | SS-ABL-50 | 0.865 | 0.788 |
| Part | SS-ABL-50 | 0.862 | 0.803 |
| Part | ABLSim-50 | **0.866** | **0.783** |

- ABLSim achieves the highest or comparable performance with weaker KB

# Conclusion

- Propose a novel consistency measure for abduction-based neuro-symbolic learning and the ABLSim method
- ABLSim significantly outperforms the state-of-the-art neuro-symbolic learning approaches in terms of speed and performance
- Future work: discover new class and new knowledge to automatically extend the knowledge base

# Q & A

Yu-Xuan Huang

huangyx@lamda.nju.edu.cn

https://www.lamda.nju.edu.cn/huangyx/

Thanks!