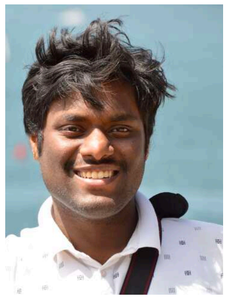
# Pearls from Pebbles: Improved Confidence Functions for Auto-labeling

12 Nov, 2024


Harit Vishwakarma
CS Ph.D. Candidate


NEURAL INFORMATION PROCESSING SYSTEMS

**Yi (Reid) Chen**
ECE Ph.D. Student



**Srinath Namburi**
CS Masters -> GE

**Sui Jiet Tay**
CS UG -> NYU


Advisors
Prof. Fred Sala
Prof. Ramya Korlakai Vinayak

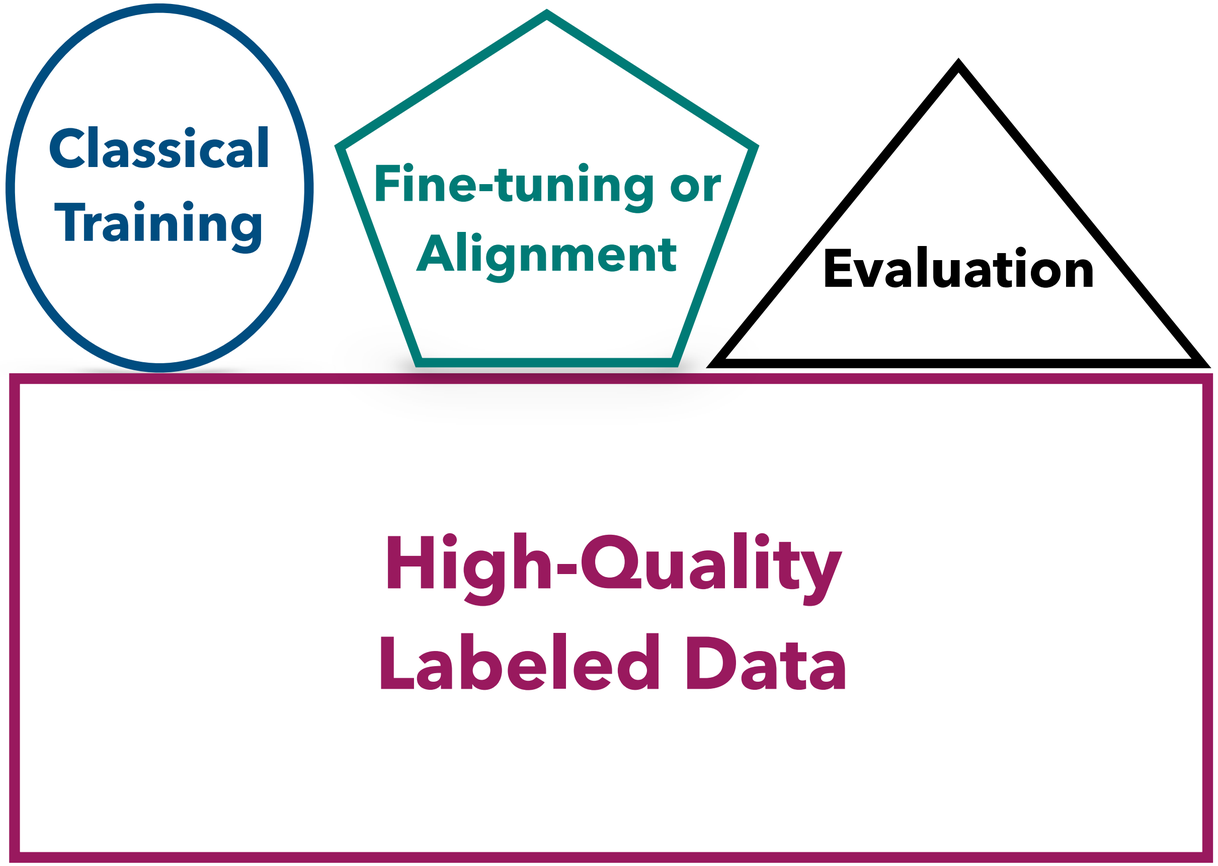

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

**Prof. Fred Sala**
**CS**

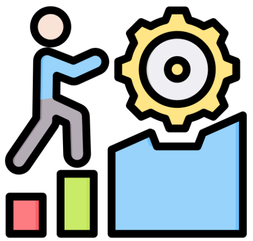
**Prof. Ramya K. Vinayak**
**ECE** + CS, Stats

# Labeled Data Bottleneck

## Need for high-quality labeled data is perpetual



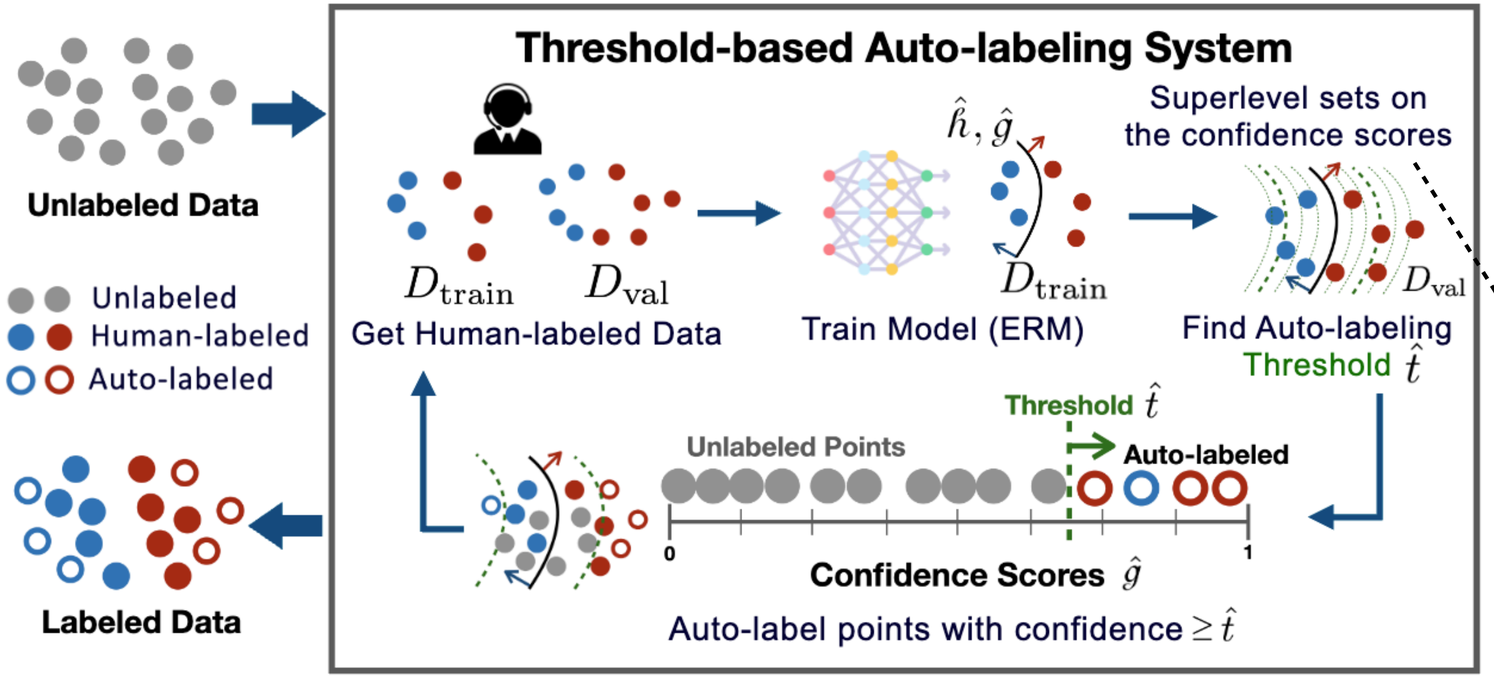

## Collecting it is Costly, Time Consuming & Laborious.



# A Promising Solution: Threshold-based Auto-labeling (TBAL)

Commercial technique getting used in practice (e.g. Amazon Sagemaker Groundtruth)

Auto-labels points on which model's **confidence scores** are above a **threshold**



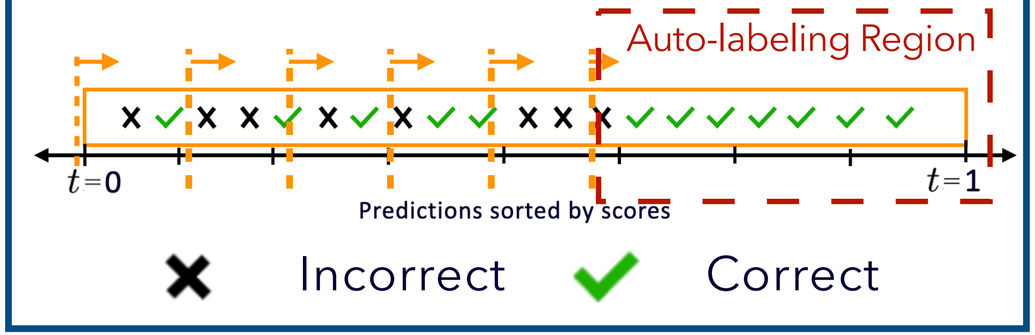

**Standard Procedure**

Model: Neural Nets

Training: Min. Cross Entropy with SGD

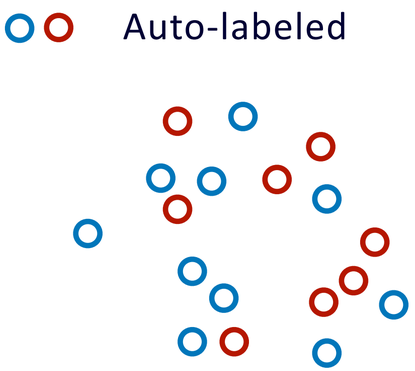
Scores (g): Softmax Outputs

Use **scores** and **validation data** to find threshold



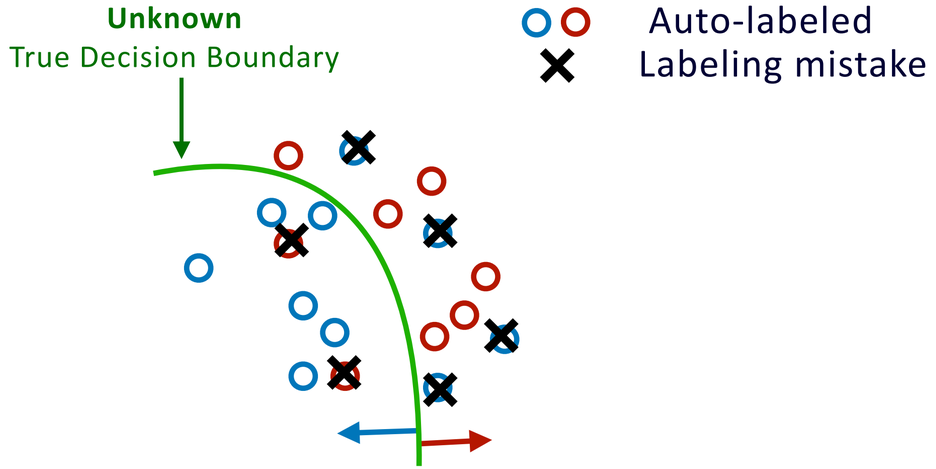
# Quality and Quantity of Auto-labeled Data

$N$ Number of **unlabeled** points




$A$ Set of **auto-labeled** points

$N_a$ Number of auto-labeled points




$M_a$ Number of labeling mistakes

## Quantity

**Auto-labeling Coverage**

$$\hat{\mathcal{P}} = \frac{N_a}{N}$$

Good Stuff
maximize this

## Quality

**Auto-labeling Error**

$$\hat{\mathcal{E}} = \frac{M_a}{N_a}$$

Bad Stuff
minimize this

**There are Trade-offs between Coverage and Error**

Need to guarantee $\leq \epsilon_a$

# Factors Affecting TBAL Performance

Assume human labels are always correct (no noise).

1. Amount of validation data used for threshold estimation.

   Less val. data $\Longrightarrow$ High variance in threshold estimation $\Longrightarrow$ low coverage or high error.

   Promises and Pitfalls of Threshold-based Auto-labeling, **V**LSV, NeurIPS' 23 (spotlight).

2. Confidence scores on which threshold is estimated.

   Poor/overconfident scores $\Longrightarrow$ low coverage or high error.

   Pearls from Pebbles: Improved Confidence Functions for Auto-labeling, **V**CTNSV, NeurIPS' 24

3. More factors: noise, class proportions, querying strategies, model training etc.

   Future…

# Standard training procedure and softmax scores can be bad for auto-labeling

**Prone to the overconfidence problem**

High scores even for incorrect predictions

**Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images**

Anh Nguyen, University of Wyoming, anguyen8@uwyo.edu
Jason Yosinski, Cornell University, yosinski@cs.cornell.edu
Jeff Clune, University of Wyoming, jeffclune@uwyo.edu

**Don't Just Blame Over-parametrization for Over-confidence: Theoretical Analysis of Calibration in Binary Classification**

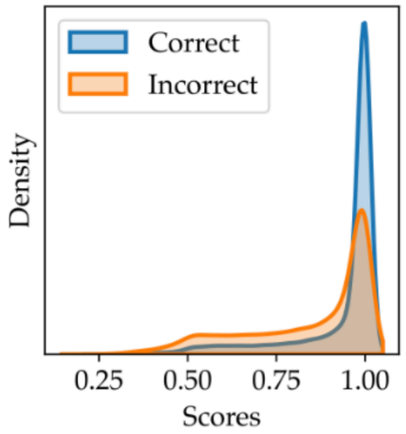
Yu Bai [1] Song Mei [2] Huan Wang [1] Caiming Xiong [1]

Szegedy et al. 2014; Nguyen et al. 2015; Hendricks & Gimpel 2017; Guo et al. 2017; Hein et al. 2018, Bai et al. 2021

## Experiment

### Run 1 round of TBAL

| | |
|---|---|
| **Data** | CIFAR-10 |
| **Model** | CNN model (5.8 M parameters) |
| **Training data** | 4000 points drawn randomly |
| **Validation data** | 1000 points drawn randomly |
| **Error Tolerance** | 5% |



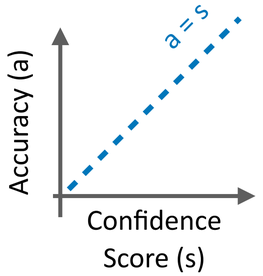

Kernel Density Estimate(KDE) of scores on the remaining unlabeled data

| | |
|---|---|
| **Test Accuracy** | 55% |
| **Coverage** | 2.9% |
| **Auto-labeling Error** | 10.1% |

# Ad-hoc Methods to Reduce Overconfidence may not help either

**Calibration**

Points where score is t, the accuracy on those points should be t




**On Calibration of Modern Neural Networks**

Chuan Guo [*1] Geoff Pleiss [*1] Yu Sun [*1] Kilian Q. Weinberger [1]

**Verified Uncertainty Calibration**

Ananya Kumar, Percy Liang, Tengyu Ma

**TOP-LABEL CALIBRATION AND MULTICLASS-TO-BINARY REDUCTIONS**

Chirag Gupta & Aaditya Ramdas

**Cut your Losses with Squentropy**

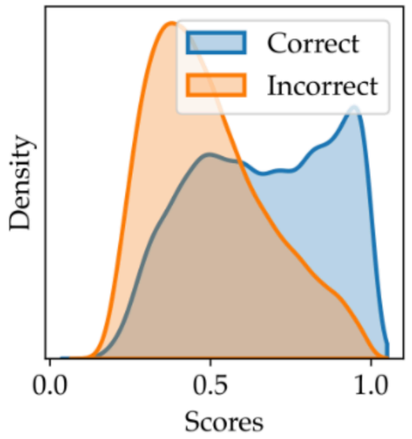
Like Hui [1 2] Mikhail Belkin [2 1] Stephen Wright [3]

Platt 1999; Zadrozny & Elkan, 2001; 2002; Guo et al. 2017; Kumar et al. 2019; Corbiére et al. (2019); Kull et al. 2019, Mukhoti et al. 2020; Gupta & Ramdas 2021; Moon et al. 2020; Zhu et al. 2022; Hui et al. 2023

## Experiment

Run 1 round of TBAL + **Temperature Scaling**

| | |
|---|---|
| **Data** | CIFAR-10 |
| **Model** | CNN model (5.8 M parameters) |
| **Training data** | 4000 points drawn randomly |
| **Validation data** | 1000 points drawn randomly |
| **Error Tolerance** | 5% |



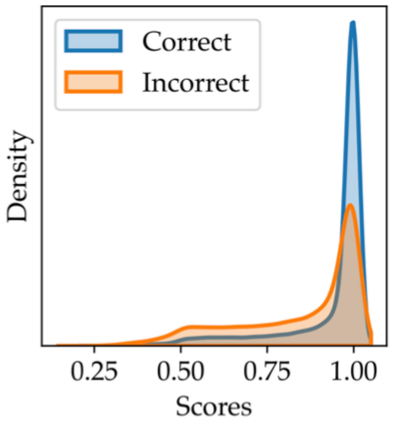

Kernel Density Estimate(KDE) of scores on the remaining unlabeled data

| | |
|---|---|
| **Test Accuracy** | 55% |
| **Coverage** | 4.9% |
| **Auto-labeling Error** | 14.1% |

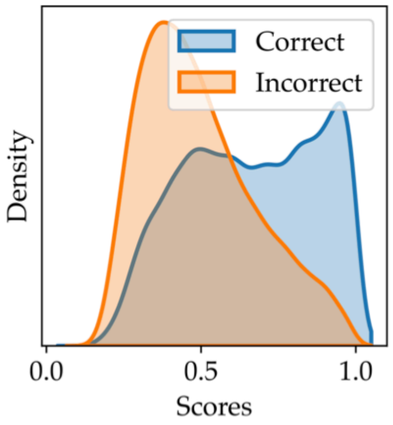
What are the right choices of scores and how do we get them?

**We propose Colander, a principled method to learn confidence scores tailored for TBAL.**

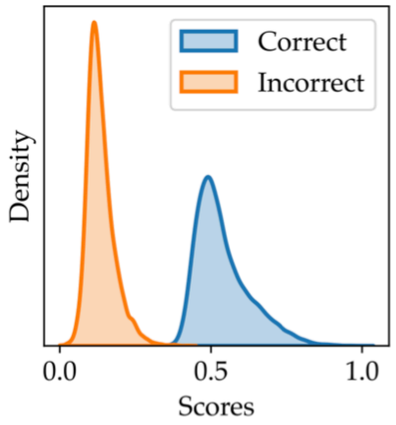
# Colander boosts coverage significantly



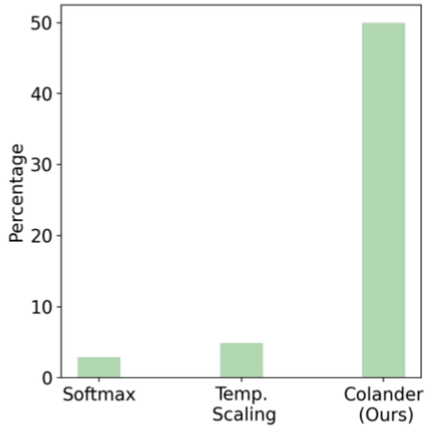

(a) Softmax



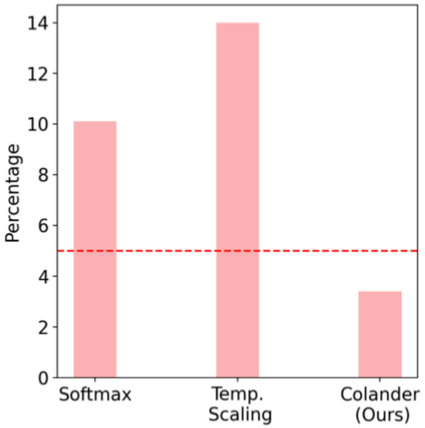

(b) Temp. Scaling




(c) `Colander` (Ours)




(d) Coverage




(e) Auto-labeling error

| | |
|---|---|
| **Data** | CIFAR-10 |
| **Model** | CNN model (5.8 M parameters) |
| **Training data** | 4000 points drawn randomly |
| **Validation data** | 1000 points drawn randomly |
| **Error Tolerance** | 5% |

Run 1 round of TBAL +
**Temperature Scaling** or **Colander**

# How does Colander work?

# The Optimal Confidence Functions for TBAL

In any round, given the classifier $h$

We want to find function $g$ that can,

a) Give maximum coverage

b) Ensure auto-labeling error $\leq \epsilon_a$

$$\hat{y} := h(\mathbf{x})$$

$$\textit{confidence function } g : \mathcal{X} \to \Delta^k$$

Depends on $h$

but drop it for convenience

Hypothetically, if we know true distribution and labels,

Coverage $\mathscr{P}(g, \mathbf{t} \mid h) := \mathbb{P}_{\mathbf{x}}\big(g(\mathbf{x})[\hat{y}] \geq \mathbf{t}[\hat{y}]\big),$

Auto-labeling Error $\mathscr{E}(g, \mathbf{t} \mid h) := \mathbb{P}_{\mathbf{x}}\big(y \neq \hat{y} \mid g(\mathbf{x})[\hat{y}] \geq \mathbf{t}[\hat{y}]\big).$

$$\underset{g \in \mathcal{G}, \mathbf{t} \in T^k}{\arg\max} \quad \mathscr{P}(g, \mathbf{t} \mid h) \;\; \text{s.t.} \;\; \mathscr{E}(g, \mathbf{t} \mid h) \leq \epsilon_a. \quad \textbf{(P1)}$$

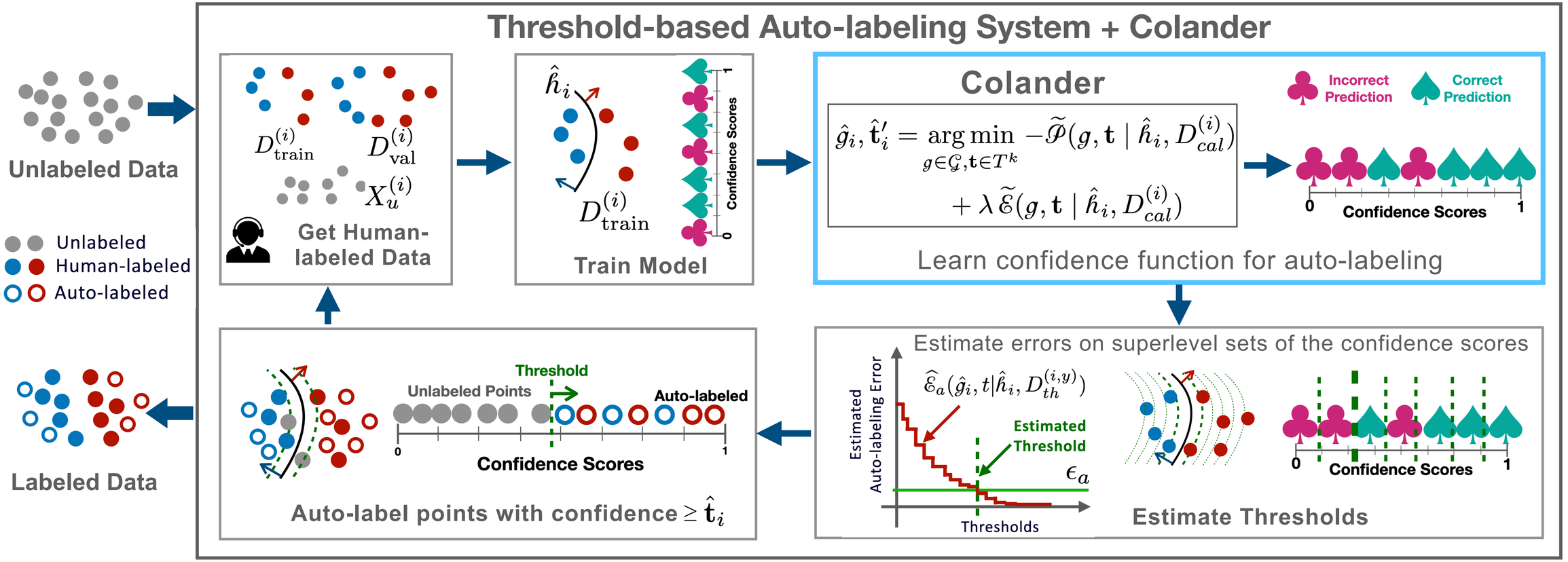
$g^\star \quad \mathbf{t}^\star$

Practical Version

Estimate using part of validation data

Use smooth surrogates and solve using SGD.

# Updated workflow of TBAL



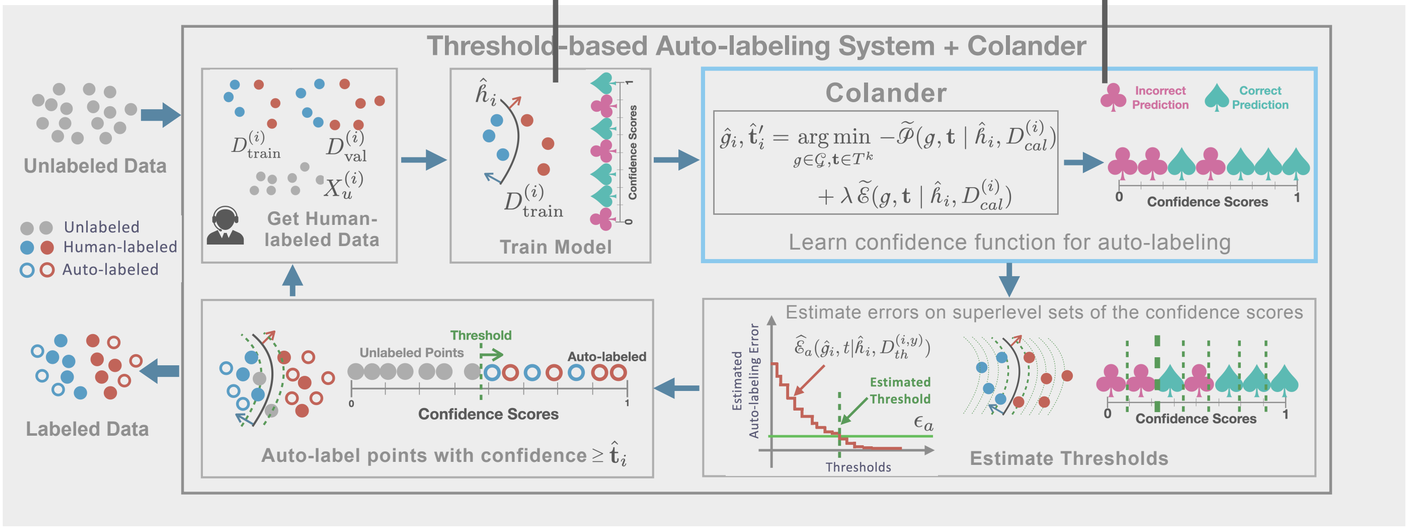
# Experiments Setup and Results

**Train-time**

1. Vanilla
2. CRL (Moon et al. 2020)
3. FMFP (Zhu et al. 2022)
4. Squentropy (Hui et al. 2023)

**Post-hoc**

1. Colander (Ours)
2. Temperature Scaling ( Guo et al. 2017)
3. Histogram Binning ( Gupta & Ramdas, 2021)
4. Scaling Binning ( Kumar et al. 2019)
5. Dirichlet ( Kull et al. 2019)




**Cross product, resulting in 20 methods.**

With Colander, TBAL achieves significantly high coverage while respecting the error constraint.

| | 20 Newsgroups | | Tiny-ImageNet | |
|---|---|---|---|---|
| | **Err (↓)** | **Cov (↑)** | **Err (↓)** | **Cov (↑)** |
| Softmax | 4.6±0.4 | 52.0±1.2 | 7.8±0.3 | 36.2±0.8 |
| TS | 8.3±0.6 | 66.6±1.4 | 13.3±0.1 | 44.9±1.0 |
| Dirichlet | 7.8±0.6 | 64.0±1.3 | 14.1±0.3 | 42.5±0.7 |
| SB | 7.8±0.7 | 63.0±2.9 | 13.0±0.5 | 45.2±2.0 |
| Top-HB | 8.2±0.8 | 66.5±2.2 | 13.7±0.1 | 45.9±1.4 |
| AdaTS | 7.4±0.6 | 64.7±2.6 | 14.0±0.3 | 46.1±0.7 |
| **Ours** | **3.3**±0.8 | **82.9**±0.4 | **0.6**±0.2 | **66.5**±0.7 |

Results with Squentropy Train-time Method

(See paper for full results)

# Thank You

Paper

Poster

Wed 11

4:30 - 6:30 PM