Confidence Functions for Auto-labeling

27 Nov, 2024

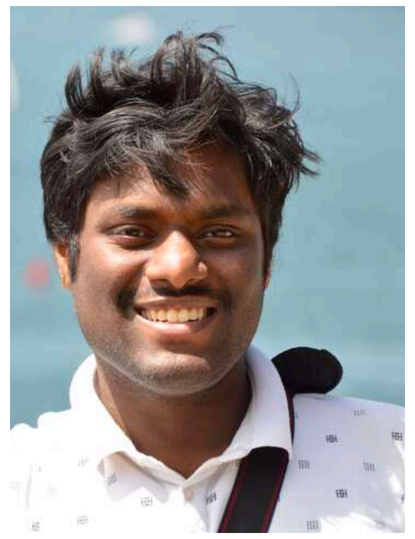
Harit Vishwakarma
CS Ph.D. Candidate

NEURAL INFORMATION PROCESSING SYSTEMS 2024

Advisors
Prof. Fred Sala
Prof. Ramya Korlakai Vinayak

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Yi (Reid) Chen
ECE Ph.D. Student



Srinath Namburi
CS Masters -> GE

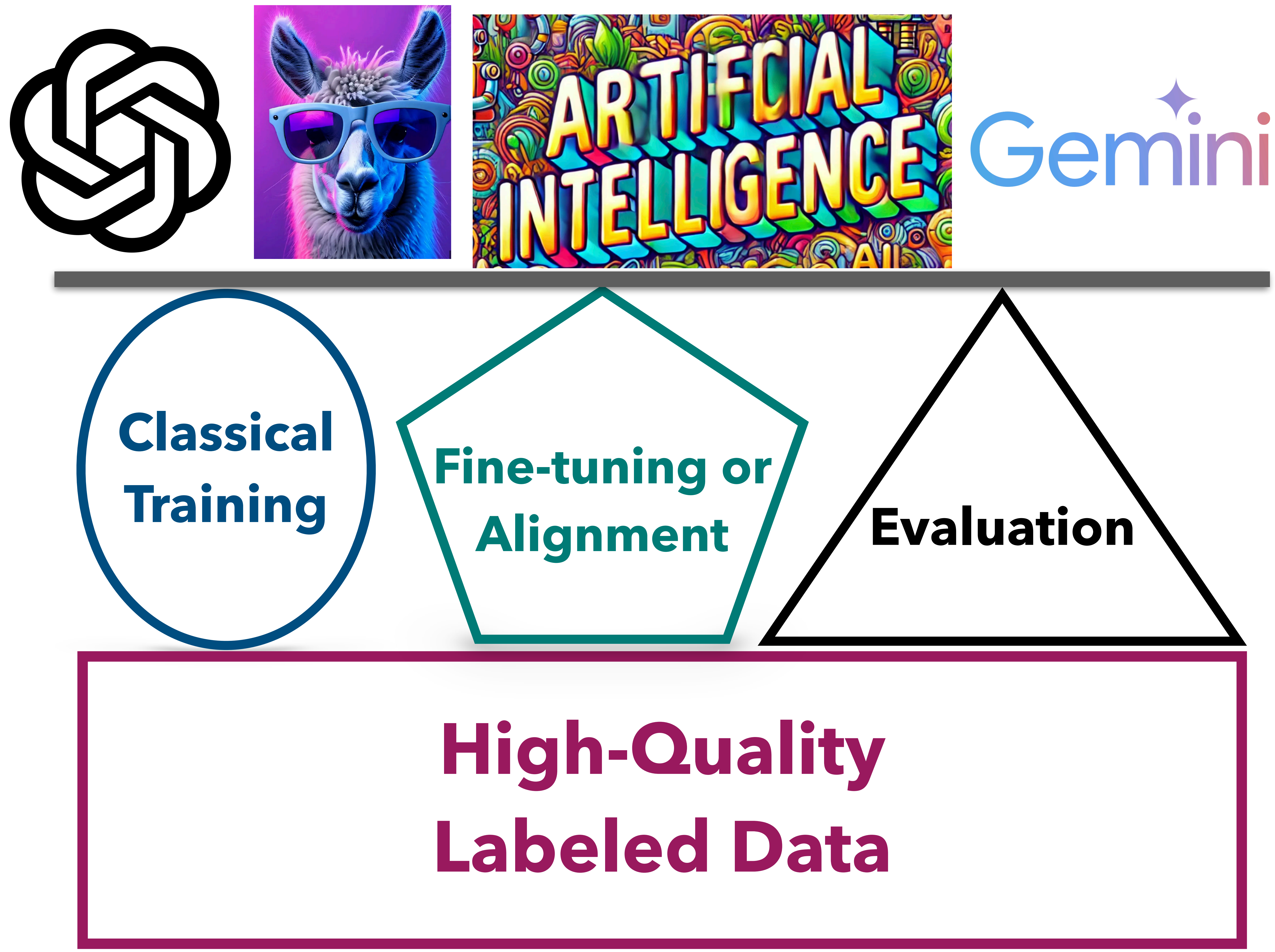
Sui Jiet Tay
CS UG -> NYU

Prof. Fred Sala
CS

Prof. Ramya K. Vinayak
ECE + CS, Stats

Labeled Data Bottleneck

High-quality labeled data is essential for safe and reliable AI



Collecting it is Costly, Time Consuming & Laborious.

Data Labeling costs a lot of time and money

IMAGENET Deng et. Al. 2009

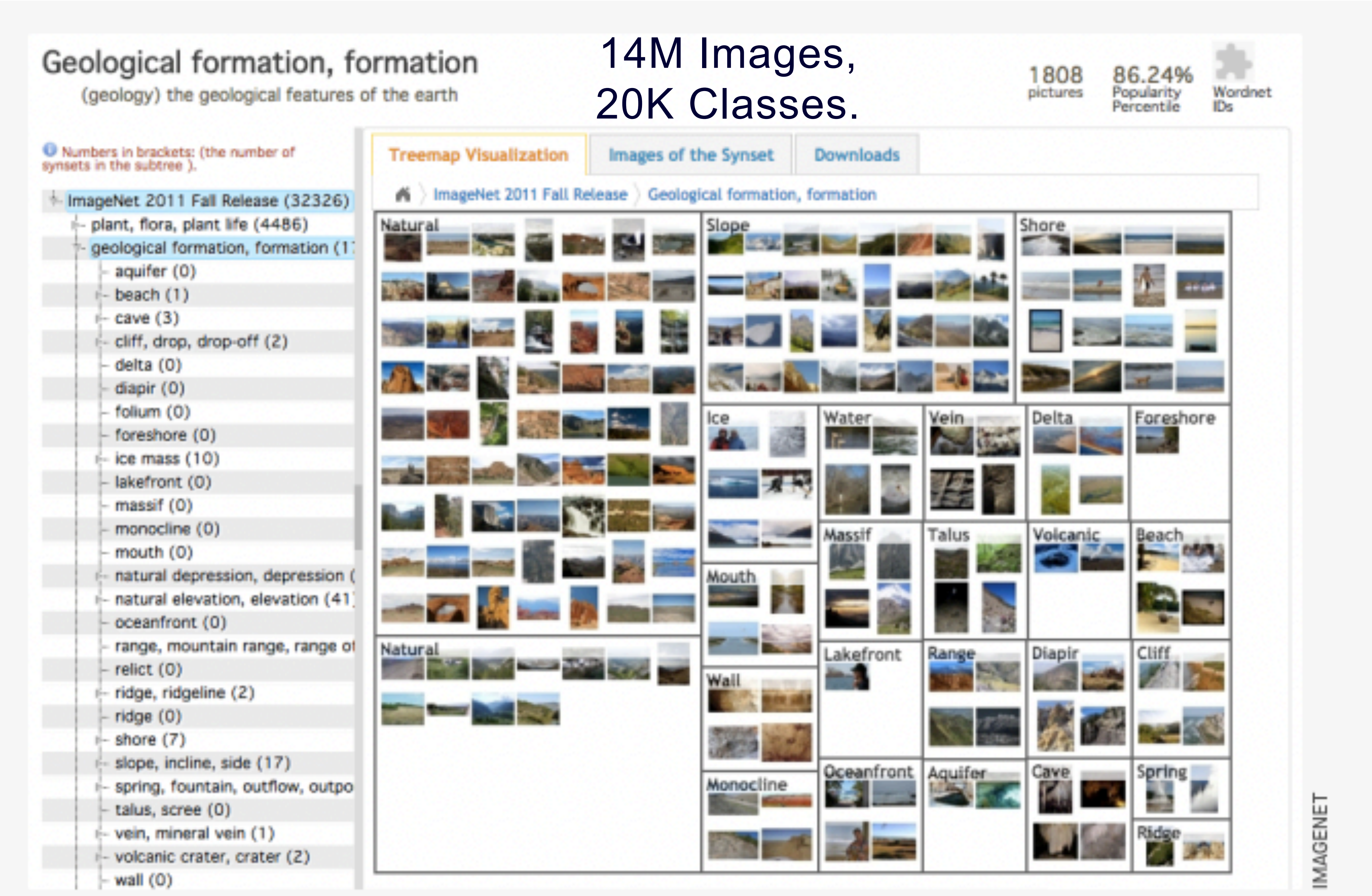
Crowdsourcing is widely used to get labels

Took multiple years and a lot of human effort

Wisdom of Crowd

amazon mechanical turk

and many others…



A screenshot of the ImageNet database online

Takes a lot of time and money to get labels.

Re-create ImageNet using Mturk: $300,000.00

The Future Of Data Labeling: Bridging Gaps In AI's Supply Chain

Trevor Koverko Former Forbes Councils Member
Forbes Technology Council
COUNCIL POST | Membership (Fee-Based)

June, 2024

Growth

The data labeling industry has witnessed remarkable growth in recent years, transitioning from a niche sector to an indispensable component of the broader artificial intelligence and machine learning landscape. According to a report by Grand View Research, the global data labeling market is anticipated to reach an astounding $17 billion by 2030, boasting a compound annual growth rate (CAGR) of 28.9% from 2023 to 2030. This surge can be attributed to the escalating demand for AI and ML applications across diverse sectors including healthcare, finance, retail and transportation.

https://www.forbes.com/councils/forbestechcouncil/2024/06/17/the-future-of-data-labeling-bridging-gaps-in-ais-supply-chain/

https://www.grandviewresearch.com/press-release/global-data-collection-labeling-market

Data labeling market projections $17B by 2030

↑

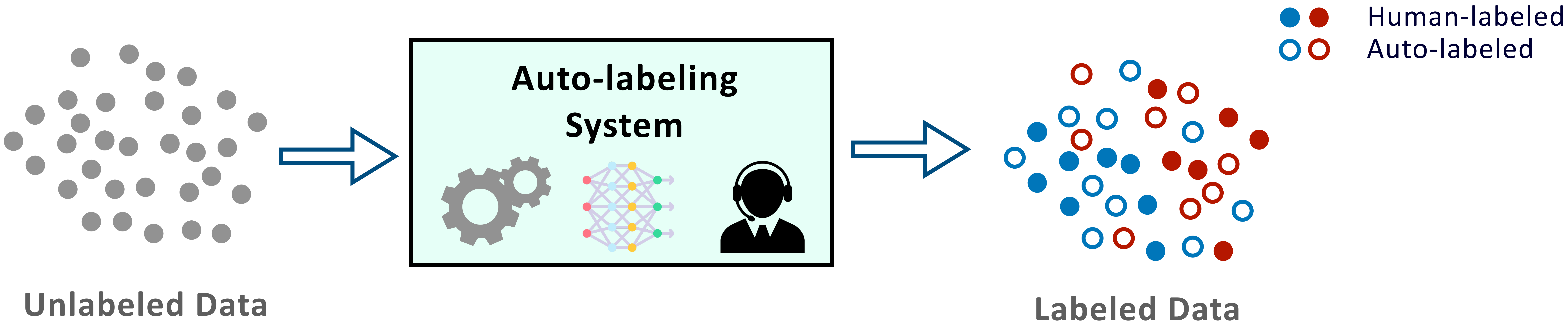
Increasing Demand for High-quality labeled data

↑

Growth of AI

Auto-labeling at lower costs and in less time

A broad set of techniques to create **labeled datasets** using **classifiers** and **human inputs.**



Weak Supervision

Snorkel

[1] Lifting Weak Supervision to Structured Prediction

Vishwakarma, Roberts, Sala; **NeurIPS 2022**

[2] Universalizing Weak Supervision

Shin, Li, **Vishwakarma**, Roberts, Sala; **ICLR 2022**

Threshold-based Auto-labeling

aws Amazon SageMaker Ground Truth

[3] Promises and Pitfalls of Threshold-based Auto-labeling

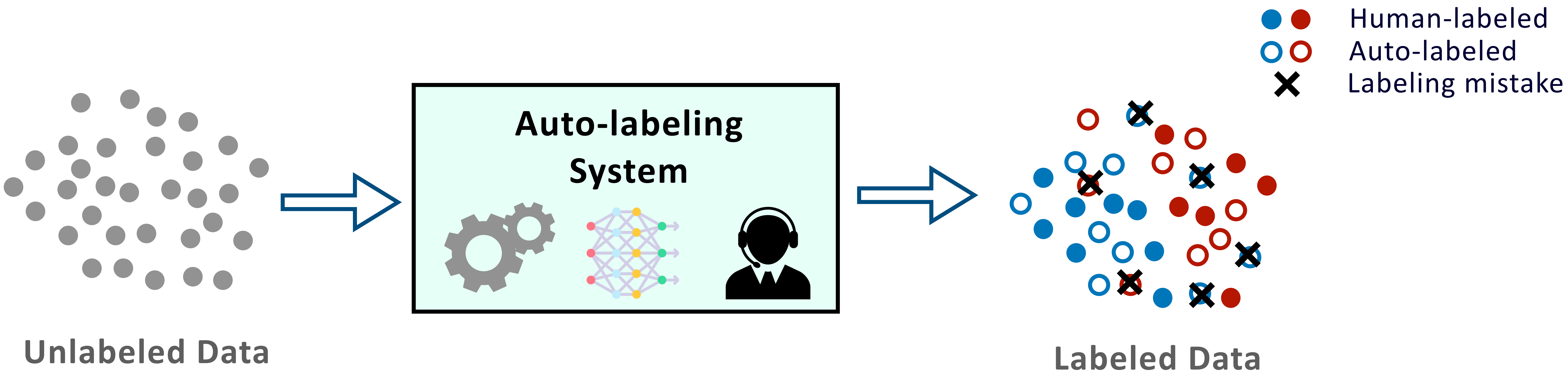
Vishwakarma, Lin, Sala, Vinayak ; **NeurIPS 2023 (Spotlight)**

[4] Pearls from Pebbles: Improved Confidence Functions for Auto-labeling

Vishwakarma, Chen, Tay, Srinath, Sala, Vinayak ; **NeurIPS 2024**

Auto-labeling Techniques can Help!

A broad set of techniques to create **labeled datasets** using **classifiers** and **human inputs.**



The output dataset may have labeling errors.

The impact of these errors is significant:

a. Datasets are static and have long shelf-life.

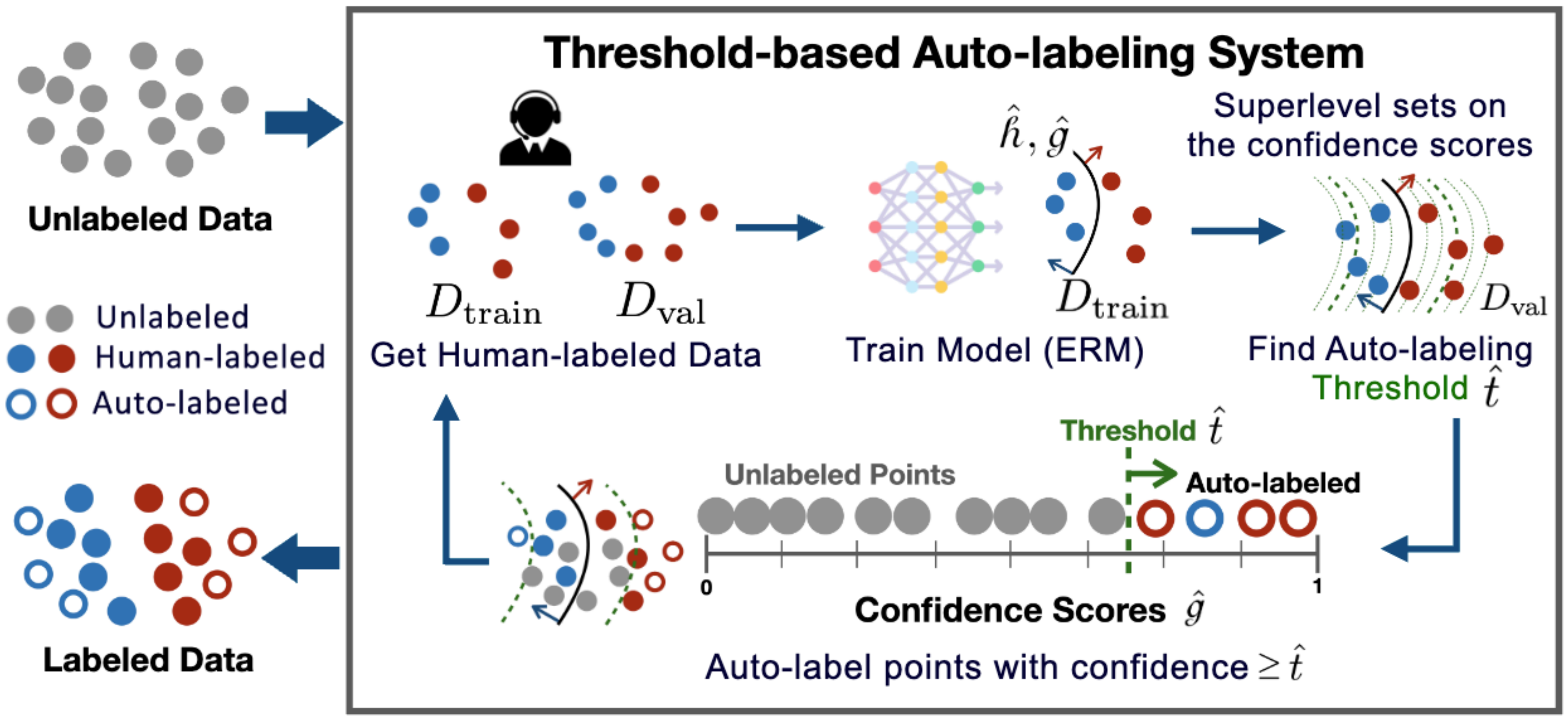
b. Multiple models are trained on the same dataset.

Threshold-based Auto-labeling (TBAL)

Auto-labels with **accuracy guarantees**!

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

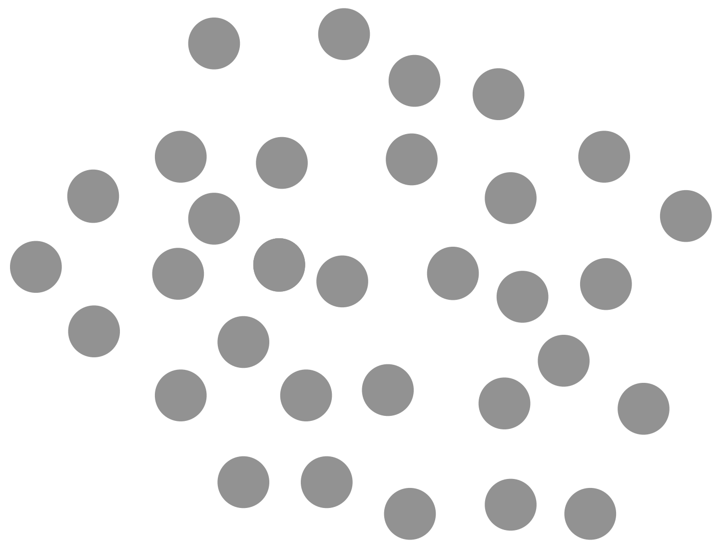
Auto-labels points on which model's **confidence scores** are above a **threshold**



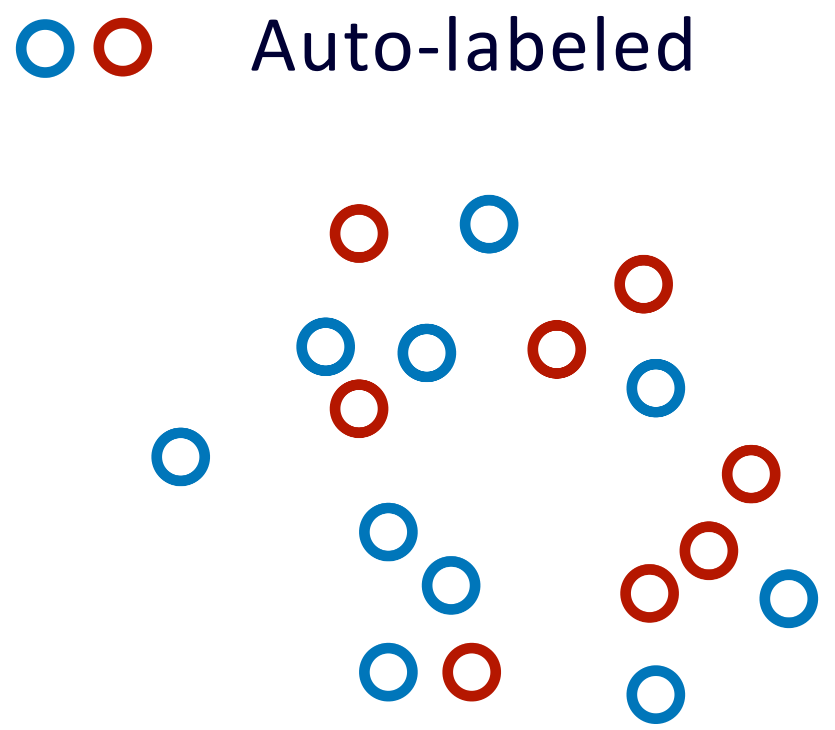
But our understanding ~~is~~ was limited!

Understanding Threshold-based Auto-labeling

Quality and Quantity of Auto-labeled Data

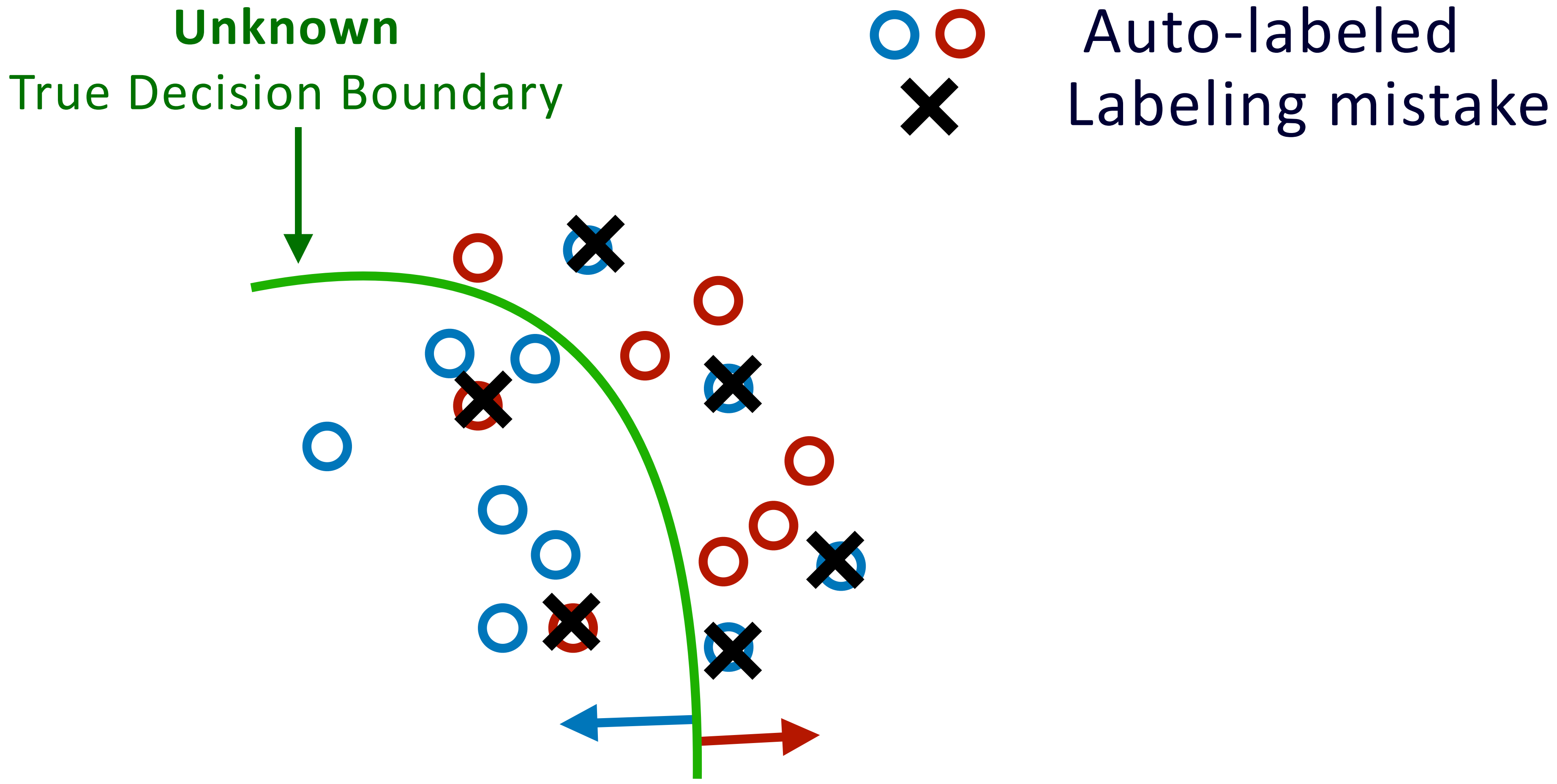


N Number of **unlabeled** points



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

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

Bad Stuff
minimize this

There are Trade-offs between Coverage and Error

Need to guarantee $\leq \epsilon_a$

Confidence Function

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

Confidence in predictions of the classifier

Depends on h but drop it for convenience

Predicted label/class

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

Confidence Score

$$g(\mathbf{x})[\hat{y}]$$

Softmax Score

Multi-class setting

0	1	2	3
0.02	0.06	0.02	**0.9**

$\hat{y} = 3 \quad g(\mathbf{x})[\hat{y}] = 0.9$

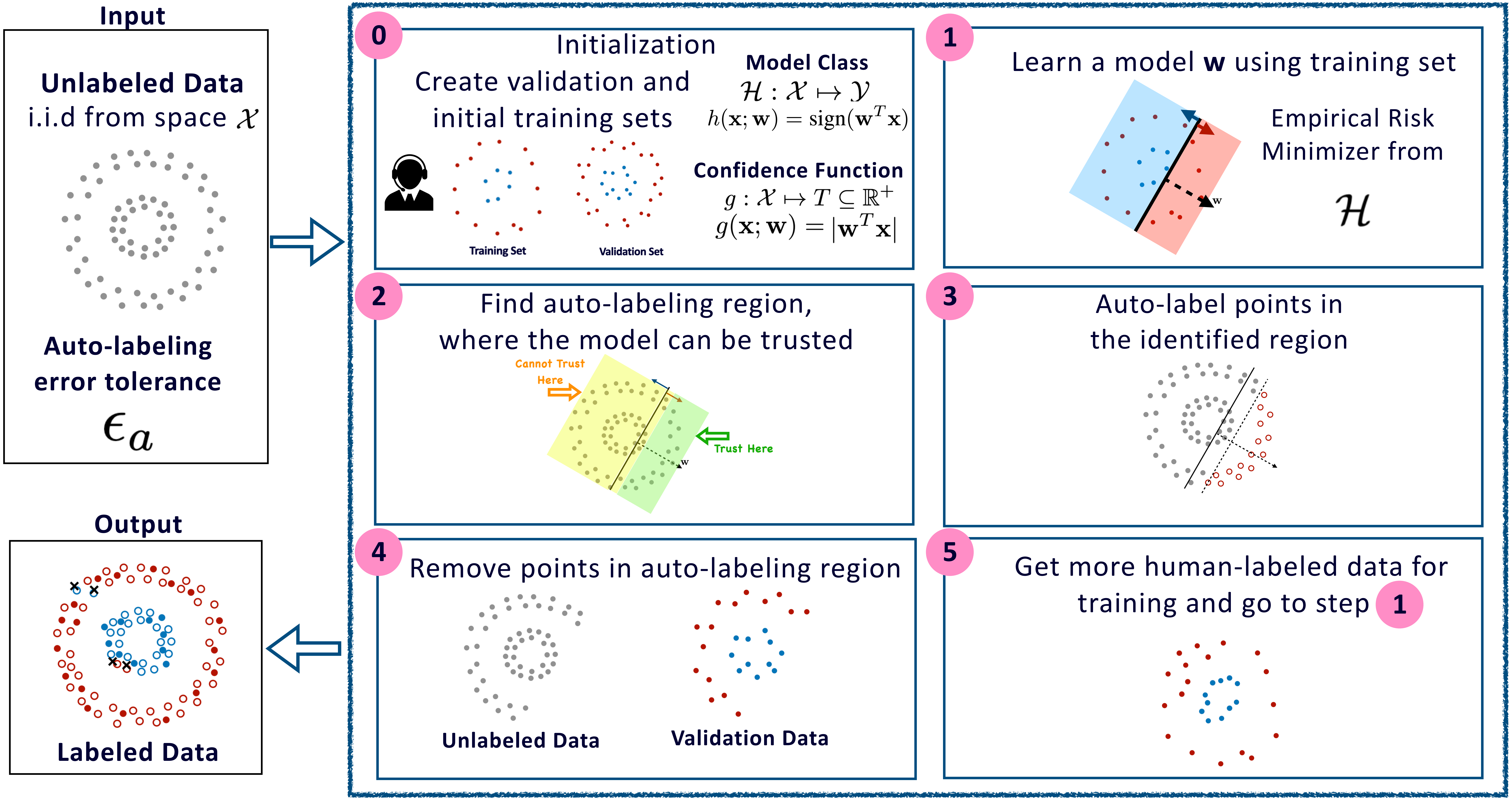
Margin Scores

Binary classes (Linear)

0	1
$-\mathbf{w}^T\mathbf{x}$	$\mathbf{w}^T\mathbf{x}$

$\hat{y} = 1 \quad g(\mathbf{x})[\hat{y}] = \mathbf{w}^T\mathbf{x}$

Threshold-based Auto-labeling Workflow (TBAL)



Step 2: Finding the auto-labeling region is crucial.

Quality

Auto-labeling Error

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

Bad Stuff
minimize this

Need to guarantee $\leq \epsilon_a$

Quantity

Auto-labeling Coverage

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

Good Stuff
maximize this

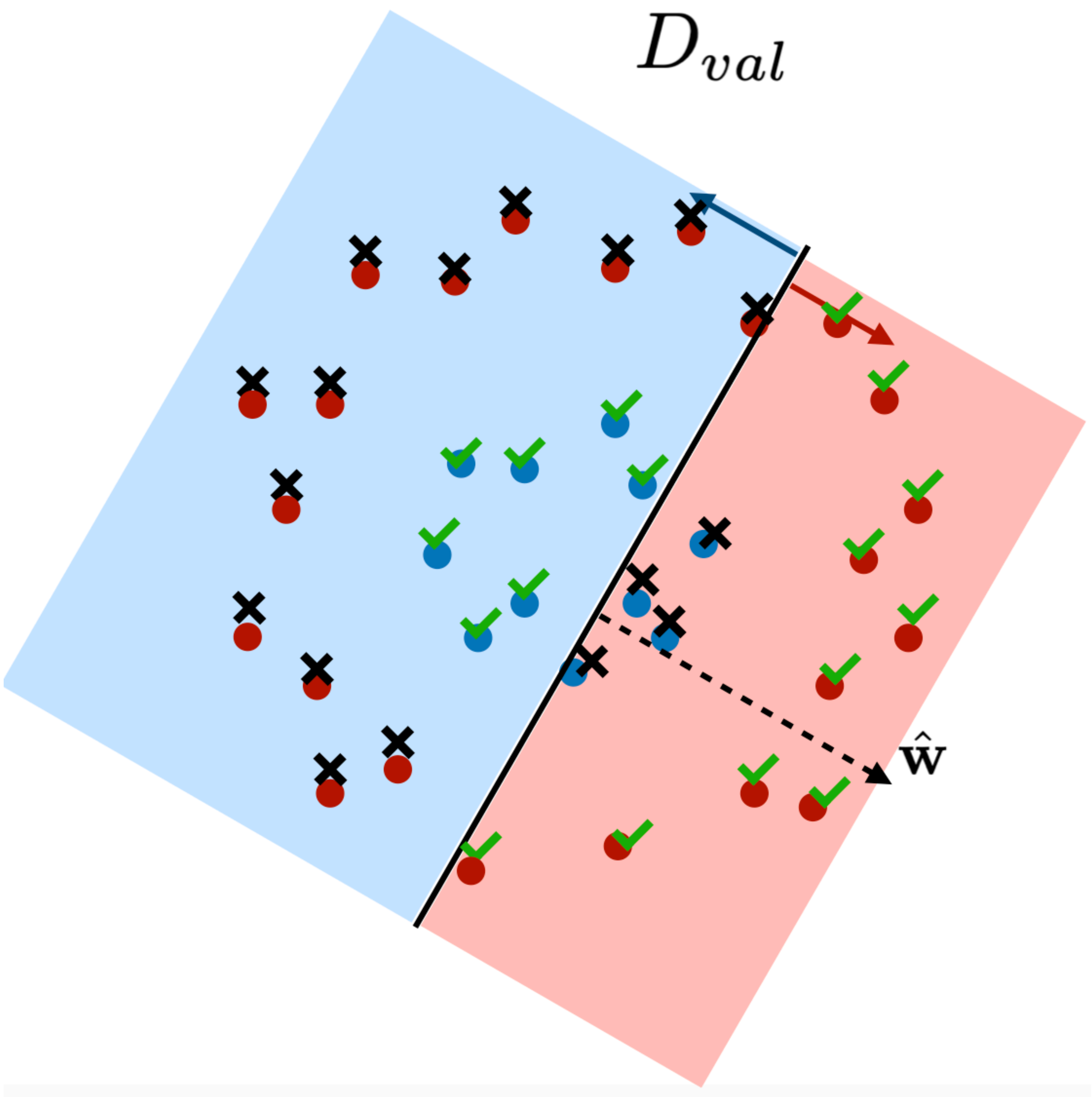
	Coverage	Error	
Case 1	High ↑	High ↑	✗
Case 2	Low ↓	Low ↓	✗
Case 3	Low ↓	High ↑	✗
Case 4	High ↑	Low ↓	✓

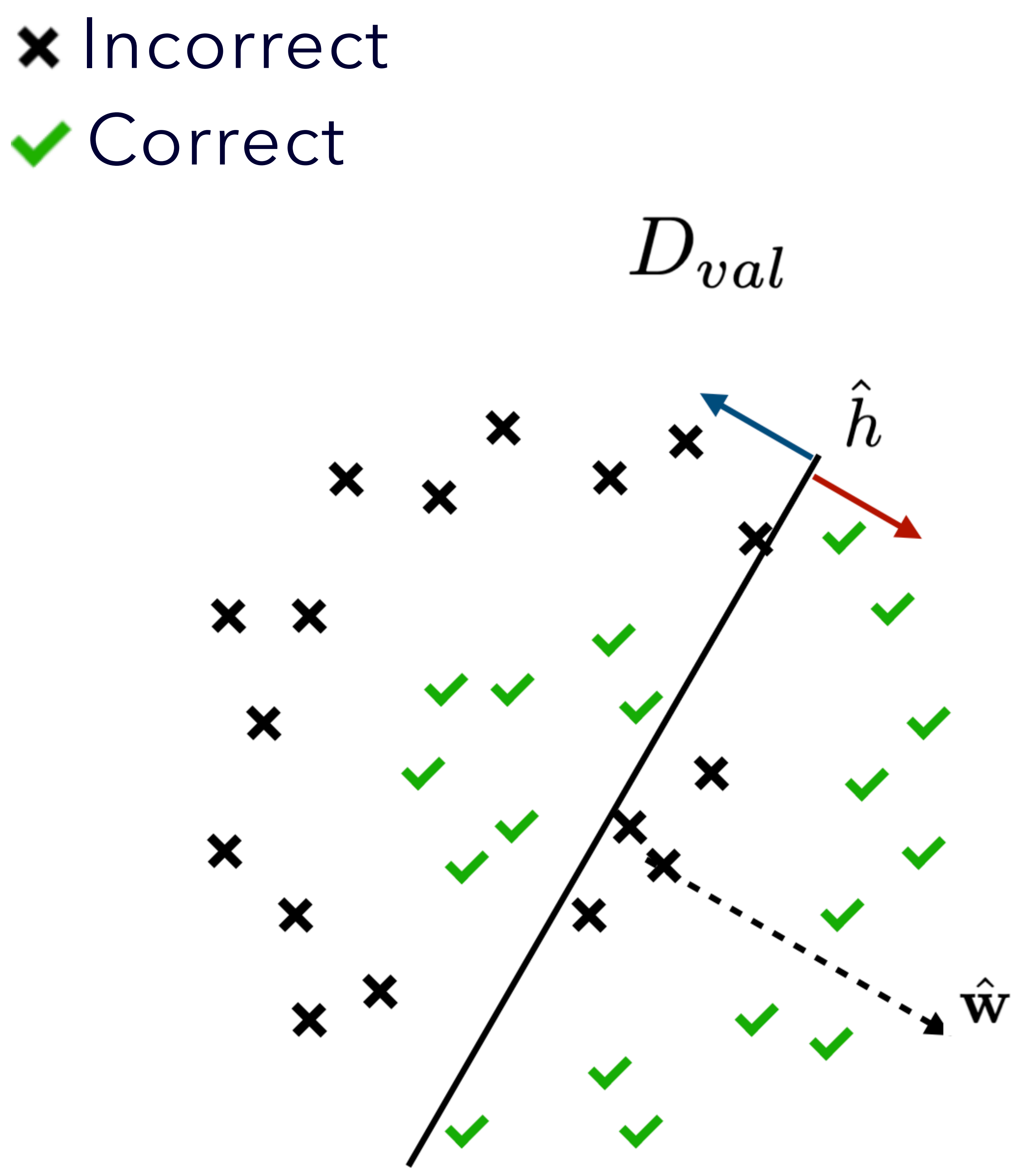
Use **validation data** and **confidence scores** to find the auto-labeling region.

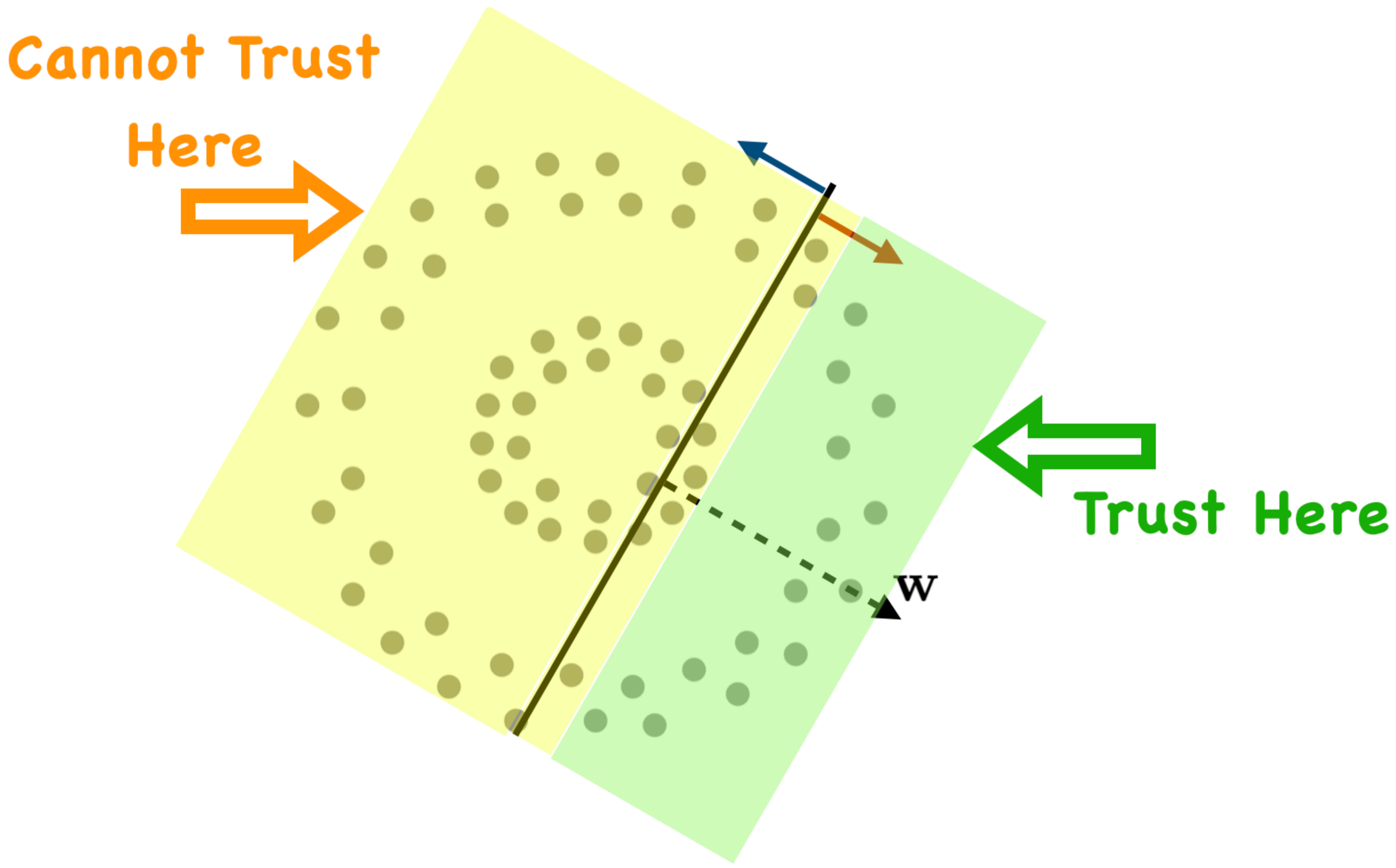
TBAL Workflow: Step 2

Find the Auto-labeling region

On the **validation data** we know where the **classifier** is **correct** and **incorrect**.



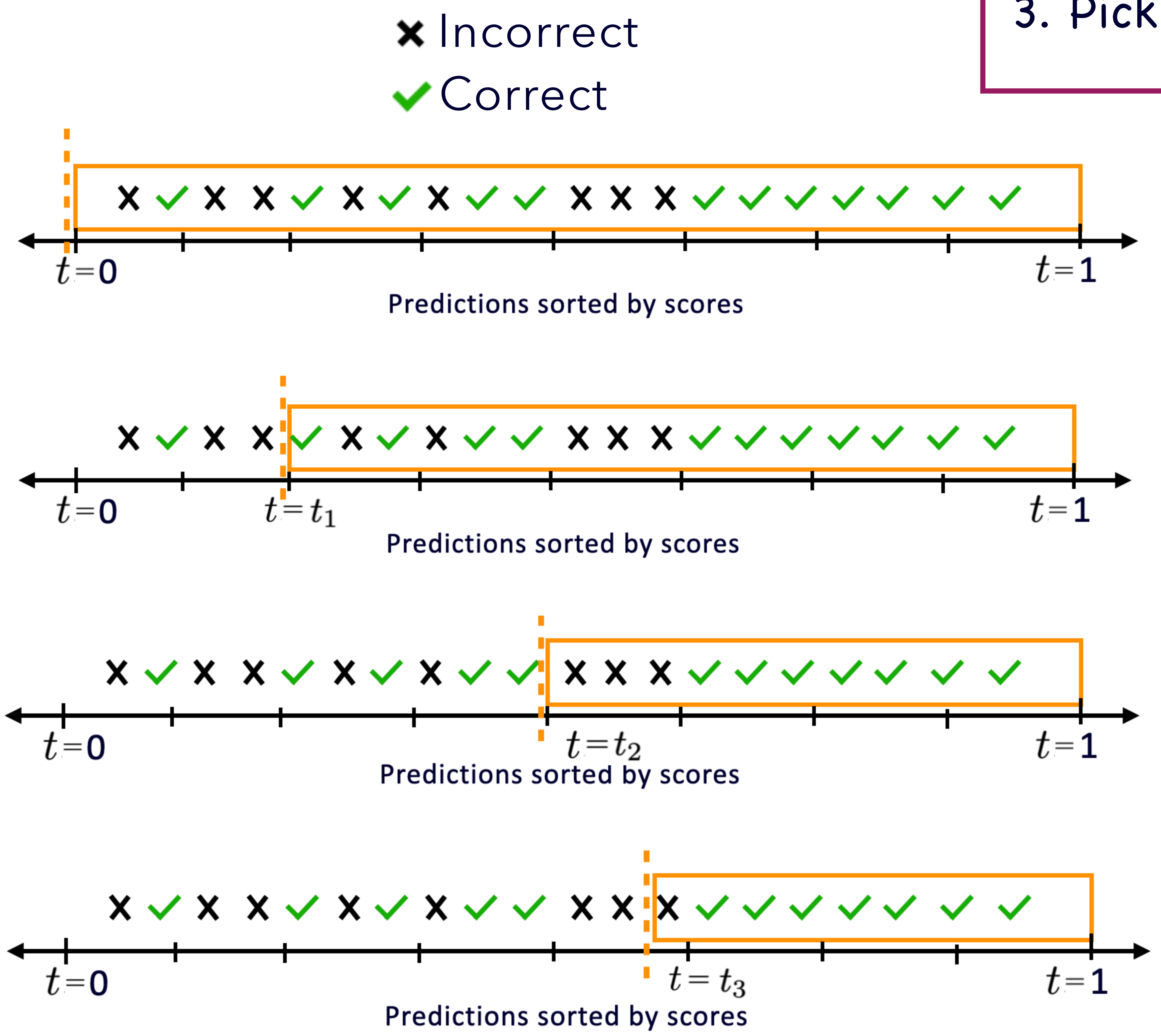


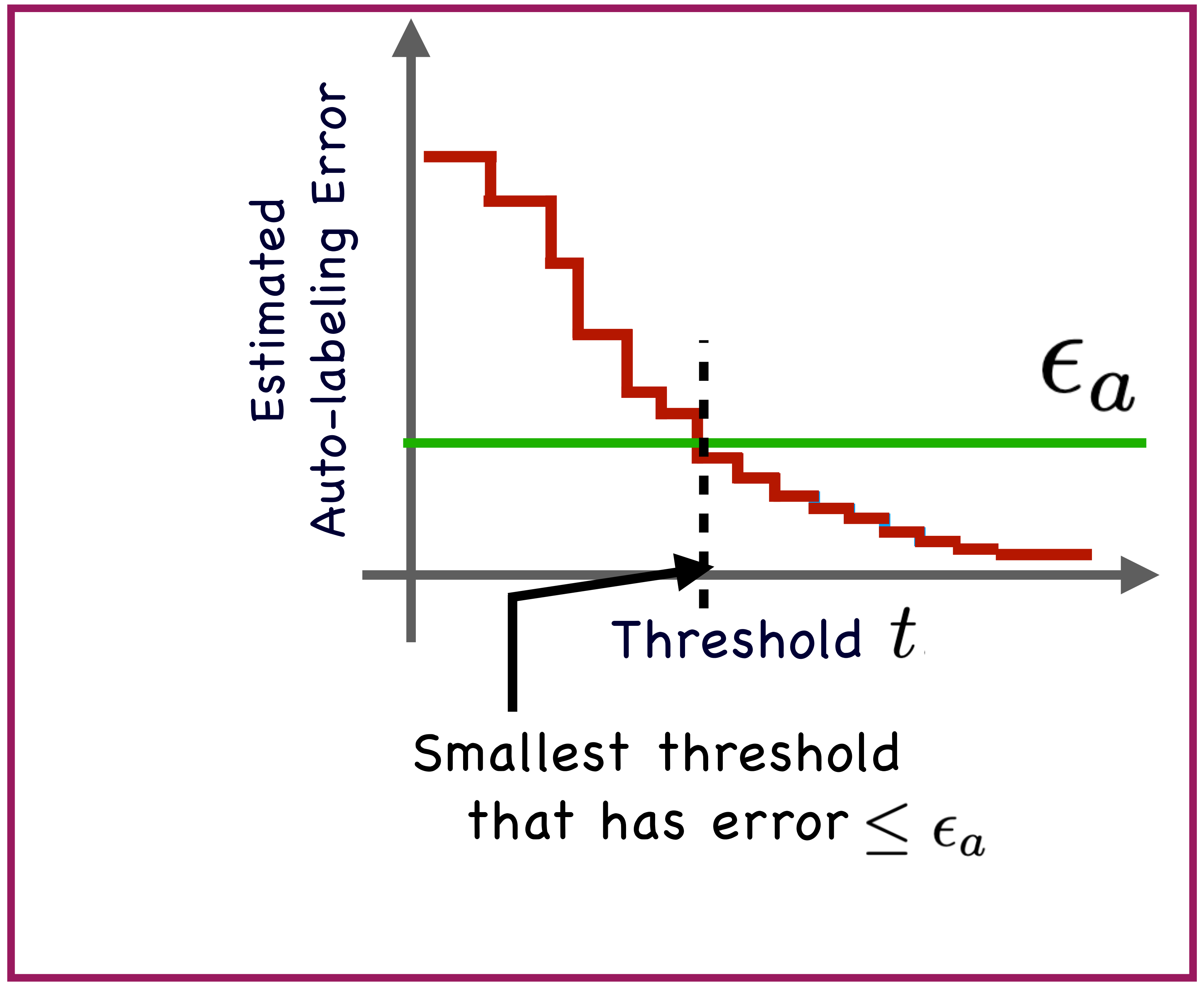


TBAL Workflow: Step 2

Find the Auto-labeling region

1. Order points based on the **Confidence scores.**
2. Estimate the auto-labeling error at several thresholds.
3. Pick the smallest threshold having error at most ϵ_a





The hope

Factors Affecting TBAL Performance

Assume human labels are always correct (no noise).

1. Amount of validation data used for threshold estimation.

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

NeurIPS' 23 (spotlight).

2. Confidence scores on which threshold is estimated.

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

NeurIPS' 24.

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

Future…

We studied TBAL and the role of validation data set

Promises and Pitfalls of Threshold-based Auto-labeling

Harit Vishwakarma
hvishwakarma@cs.wisc.edu
University of Wisconsin-Madison

Heguang Lin
hglin@seas.upenn.edu
University of Pennsylvania

Frederic Sala
fredsala@cs.wisc.edu
University of Wisconsin-Madison

Ramya Korlakai Vinayak
ramya@ece.wisc.edu
University of Wisconsin-Madison

NeurIPS, 2023 (**Spotlight**)

More details in the paper.

https://arxiv.org/abs/2211.12620v2

Long talk on
MLOpt Youtube Channel
https://www.youtube.com/@UWMadisonMLOPTIdeaSeminar

Thanks to AmFam and DSI

TL;DR

Theoretical and empirical results,

TBAL can produce accurately labeled dataset,
provided there is sufficient validation data.

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.

NeurIPS' 23 (spotlight).

2. Confidence scores on which threshold is estimated.

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

Today's Focus

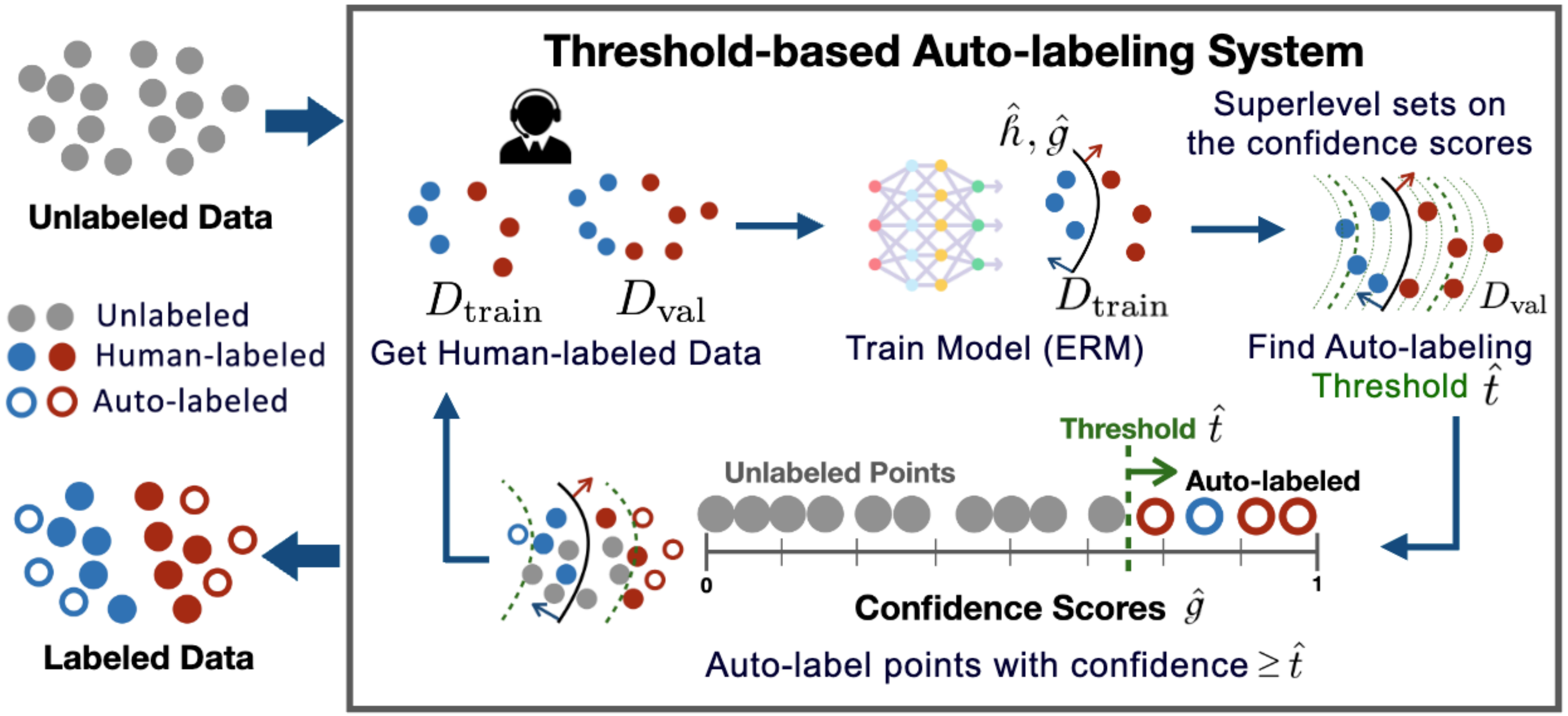
NeurIPS' 24.

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

Future…

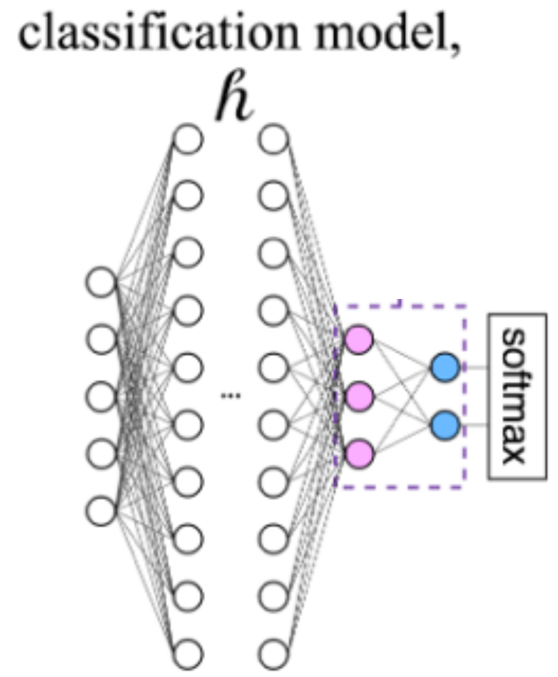
Recall the Standard Workflow for TBAL

Recap of TBAL workflow



Standard Training Procedure (Vanilla)

Pick your favorite Neural Net
(MLP, CNN, RNN, Transformer, …)



Minimize the Cross-Entropy Loss on training data using SGD

Use softmax scores for auto-labeling

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

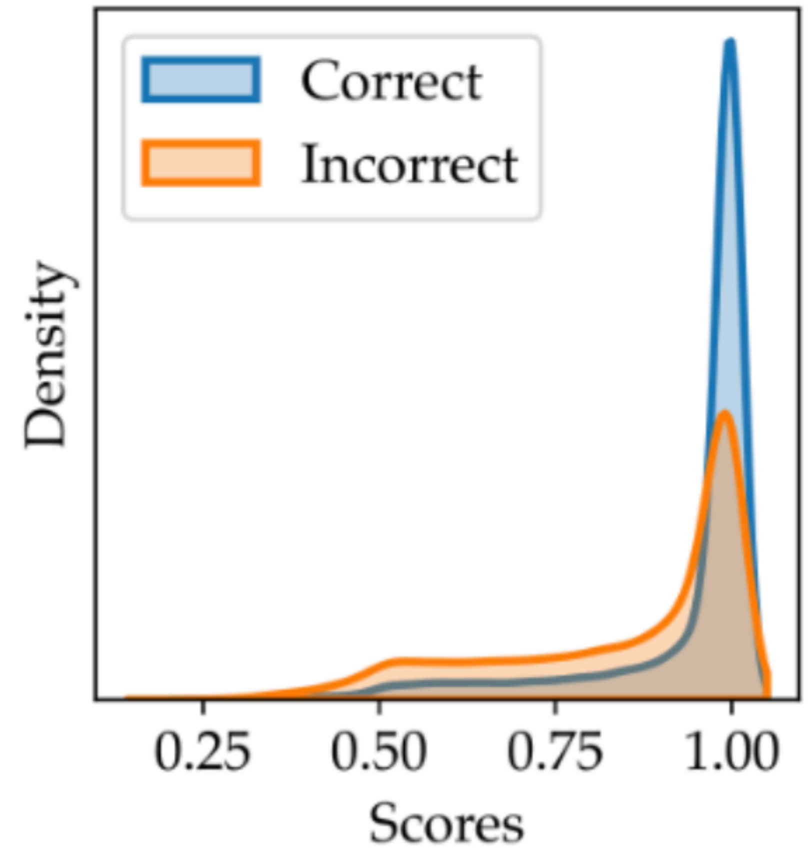
Yu Bai [1] Song Mei [2] Huan Wang [1] Caiming Xiong [1]

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%



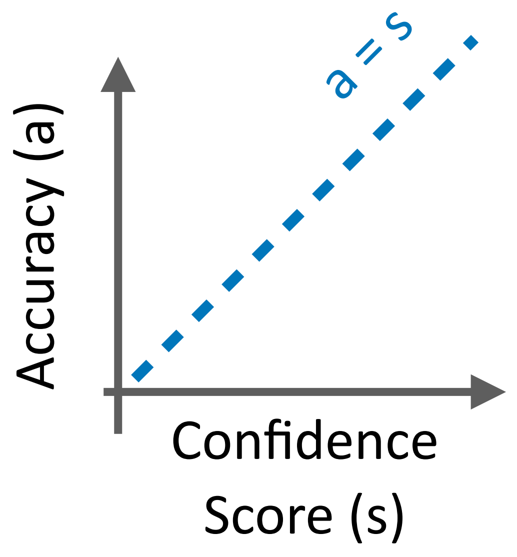
Kernel Density Estimate(KDE) of scores on the remaining unlabeled data

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

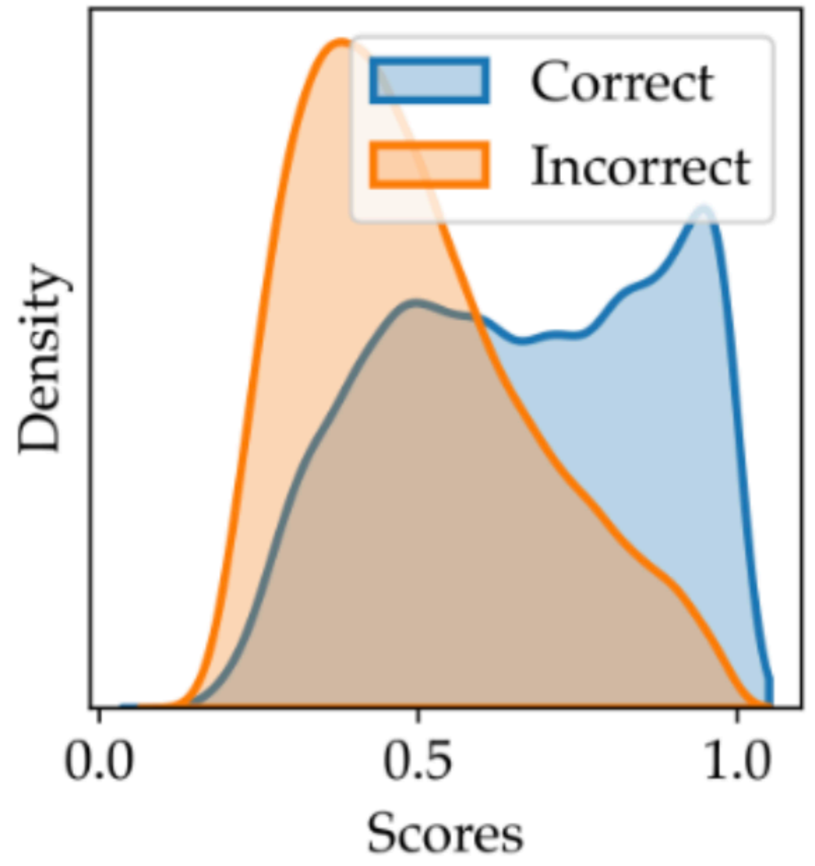
Like Hui [1 2] Mikhail Belkin [2 1] Stephen Wright [3]

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%



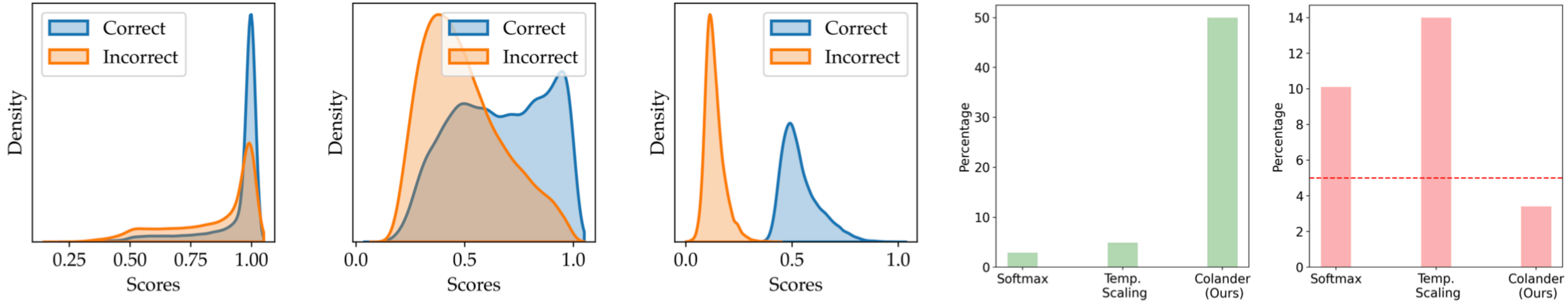
Kernel Density Estimate(KDE) of scores on the remaining unlabeled data

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 $\leq \epsilon_a$

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

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

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

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

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

Address Two Challenges

Do not know the true quantities

Efficient method to solve the optimization

Learn scores in practice using empirical estimates and smooth surrogates.

Address Two Challenges

~~Do not know the true quantities~~

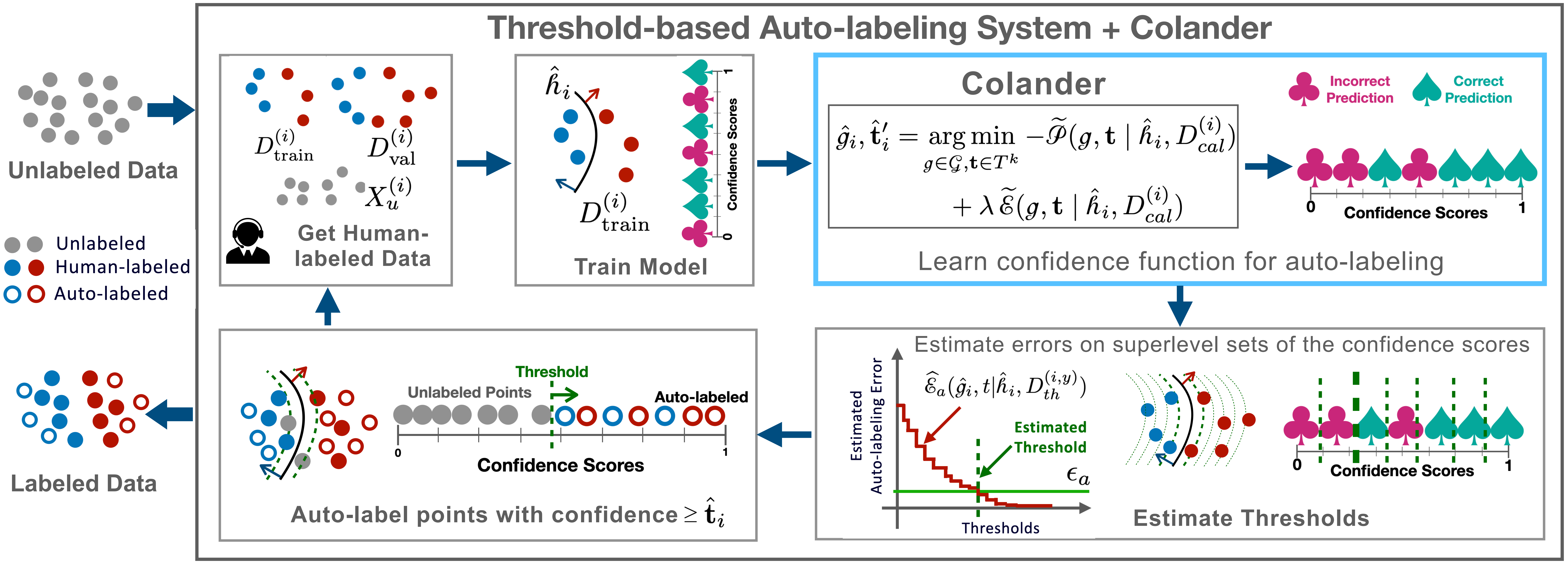
Estimate using part of validation data

~~Efficient method to solve opt.~~

Replace 0-1 variables by sigmoids.

Solve it using gradient-based methods
SGD, Adam etc.

Updated workflow of TBAL



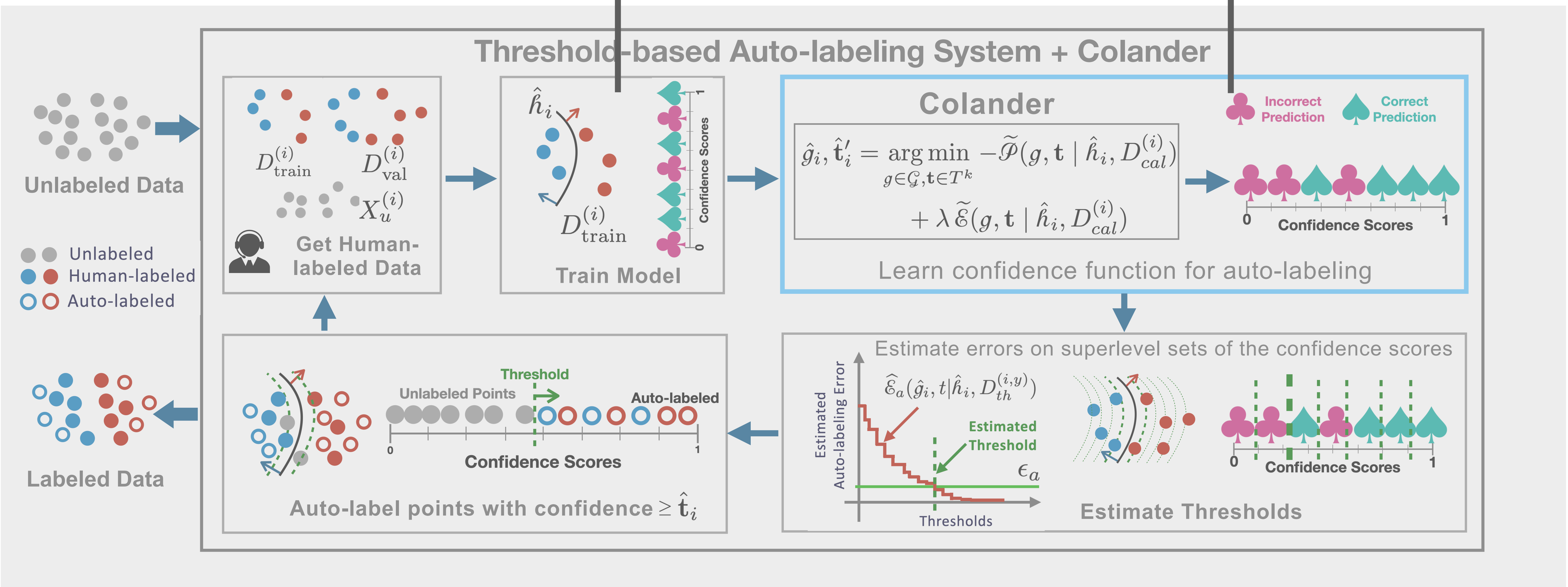
Experiments Setup and Results

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$\pm$0.4	52.0$\pm$1.2	7.8$\pm$0.3	36.2$\pm$0.8
TS	8.3$\pm$0.6	66.6$\pm$1.4	13.3$\pm$0.1	44.9$\pm$1.0
Dirichlet	7.8$\pm$0.6	64.0$\pm$1.3	14.1$\pm$0.3	42.5$\pm$0.7
SB	7.8$\pm$0.7	63.0$\pm$2.9	13.0$\pm$0.5	45.2$\pm$2.0
Top-HB	8.2$\pm$0.8	66.5$\pm$2.2	13.7$\pm$0.1	45.9$\pm$1.4
AdaTS	7.4$\pm$0.6	64.7$\pm$2.6	14.0$\pm$0.3	46.1$\pm$0.7
Ours	**3.3$\pm$0.8**	**82.9$\pm$0.4**	**0.6$\pm$0.2**	**66.5$\pm$0.7**

Results with Squentropy Train-time Method

(See paper for full results)

Takeaways

TBAL is a useful technique for creating labeled datasets with accuracy guarantees.

Common choices of scores, (**softmax scores and calibration)** can lead to poor auto-labeling performance.

We proposed **Colander** a principled method to learn the **optimal confidence functions for TBAL** and show that it boosts the performance significantly.

Future works

Reduce validation and calibration data requirements

Study factors such as label noise, class proportions, querying strategies,

Thank You

:-)

arXiv

Pearls from Pebbles: Improved Confidence Functions for Auto-labeling

Harit Vishwakarma hvishwakarma@cs.wisc.edu
Reid (Yi) Chen reid.chen@wisc.edu
Sui Jiet Tay sstay2@wisc.edu
Satya Sai Srinath Namburi sgnamburi@cs.wisc.edu
Frederic Sala fredsala@cs.wisc.edu
Ramya Korlakai Vinayak ramya@ece.wisc.edu
University of Wisconsin-Madison, WI, USA

https://arxiv.org/pdf/2404.16188

Paper

Poster@NeurIPS

Wed 11

4:30 - 6:30 PM

Thanks to American Family Insurance

Questions and Feedback

\end{talk}