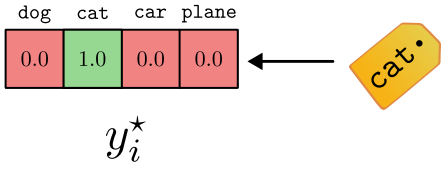
Label ambiguity in crowdsourcing for classification and expert feedback

Tanguy Lefort
IMAG, Univ Montpellier, CNRS
INRIA, LIRMM,

Supervised by
Benjamin Charlier
Alexis Joly
and **Joseph Salmon**

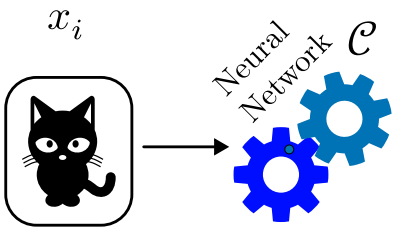
How to train your classifier

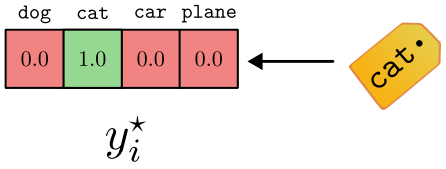
x_i



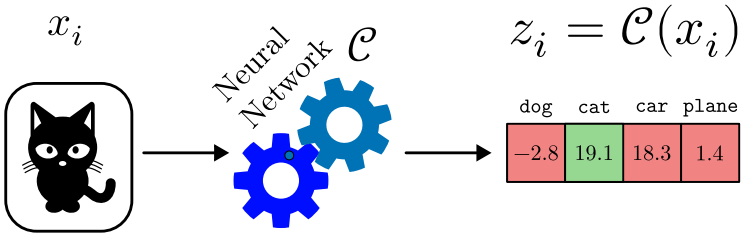
$y_i^\star$

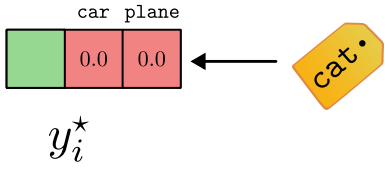
How to train your classifier



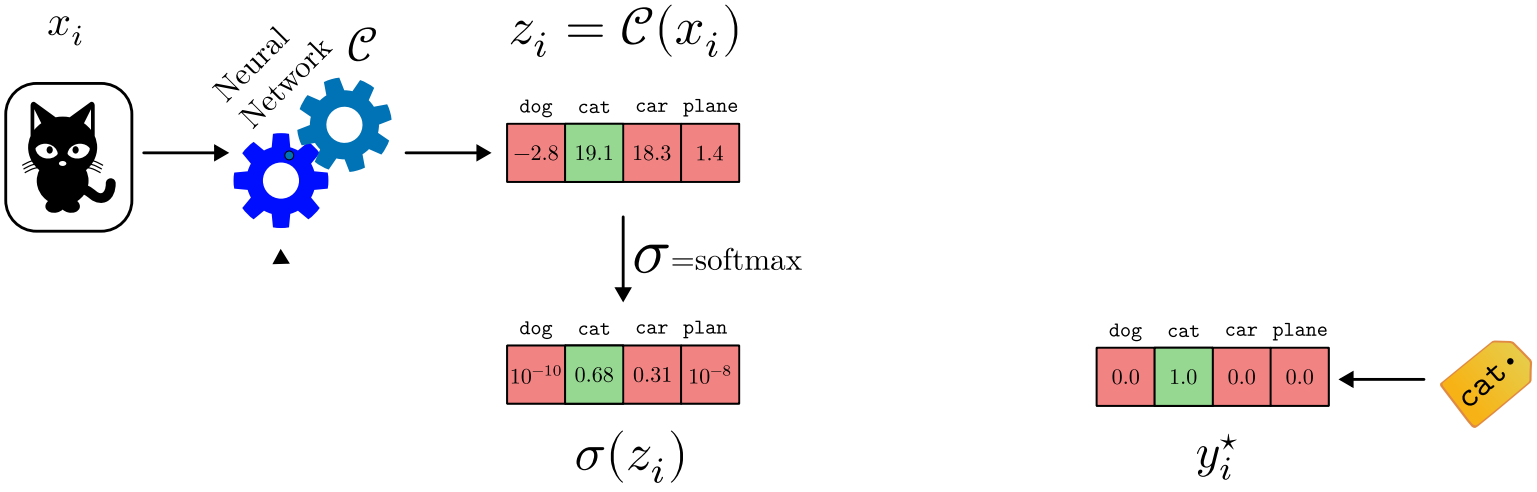


How to train your classifier

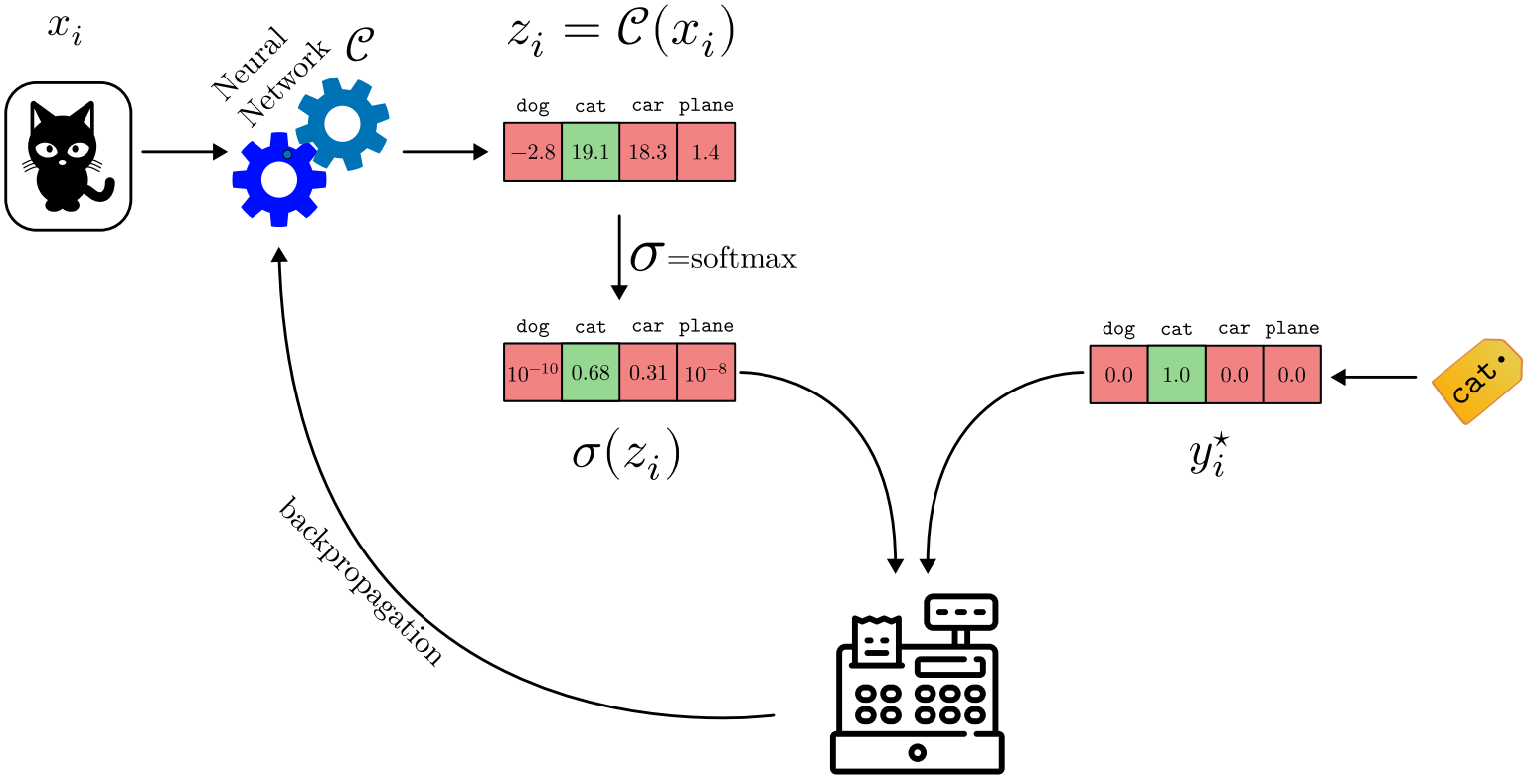




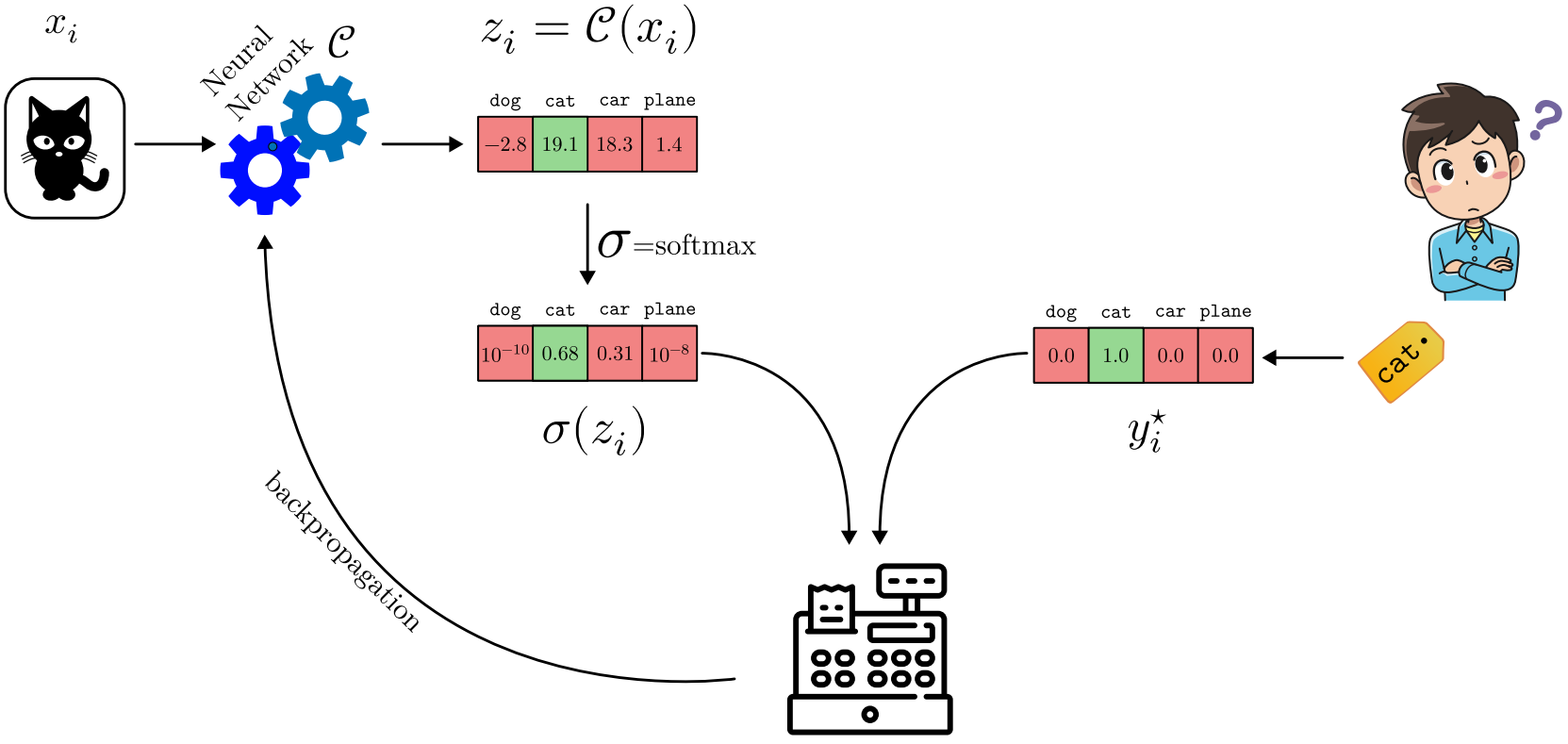
How to train your classifier



How to train your classifier



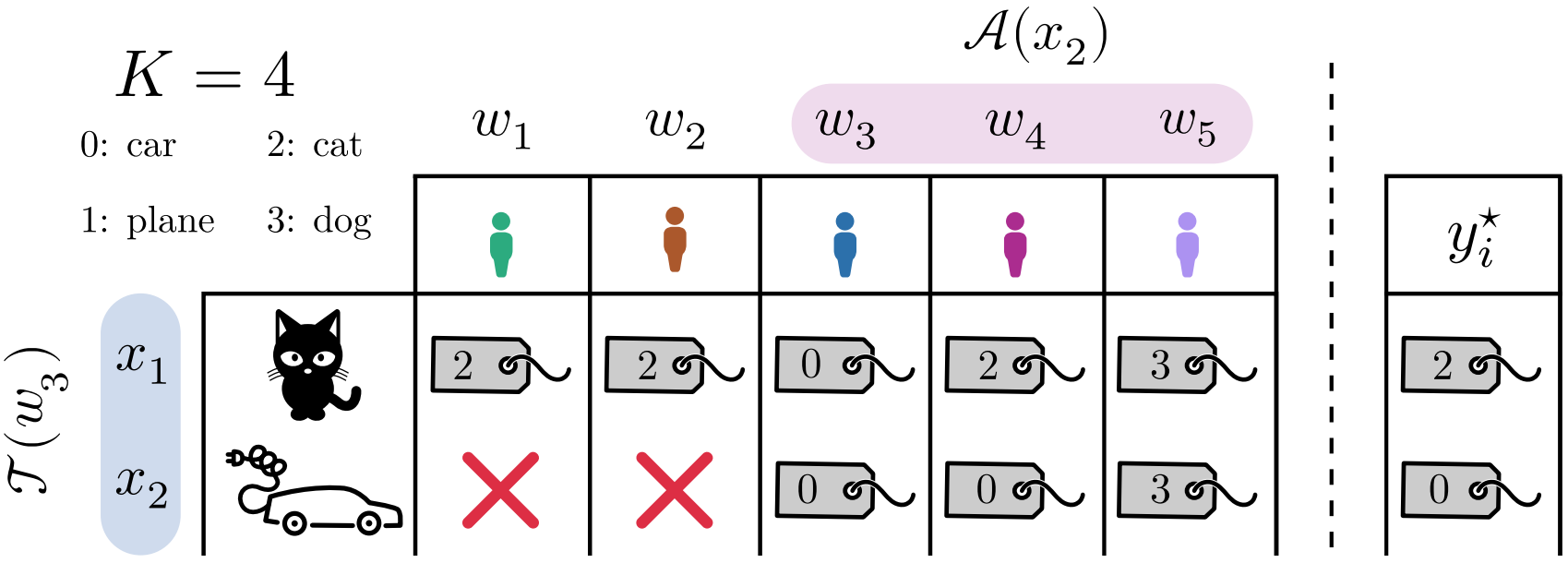
How to train your classifier



Ask citizens to label our data

Framework and notations

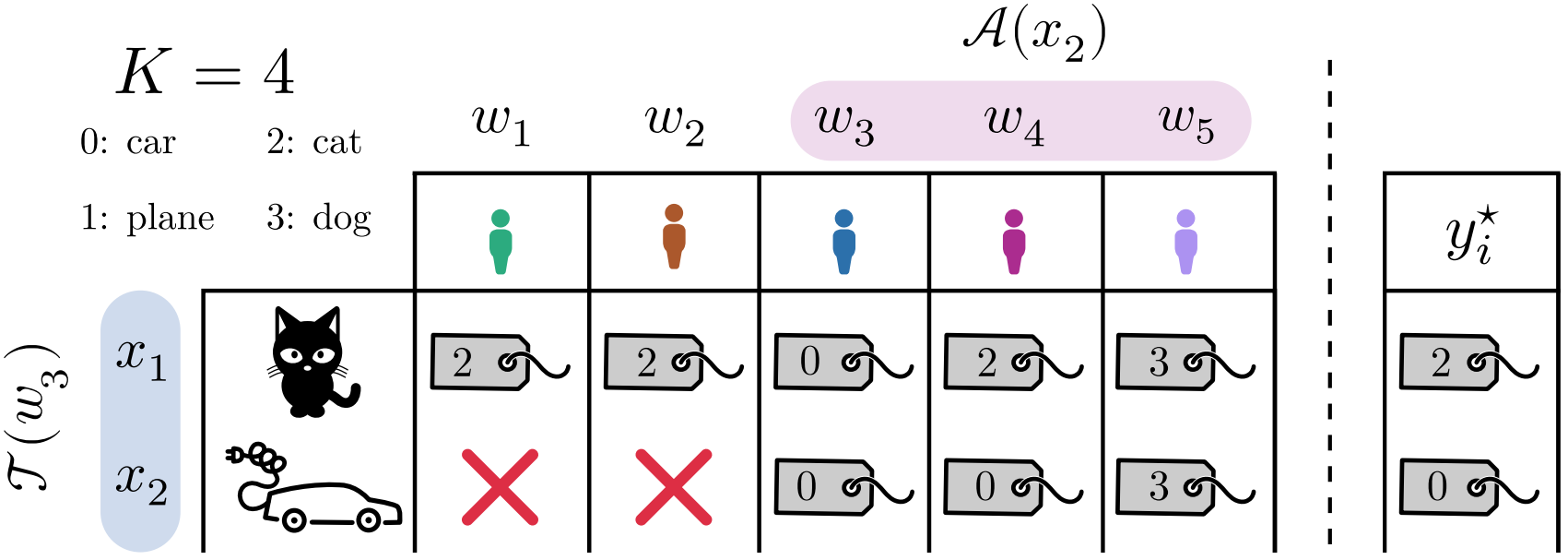
- Workers sort a given task into one of the K classes



Ask citizens to label our data

Framework and notations

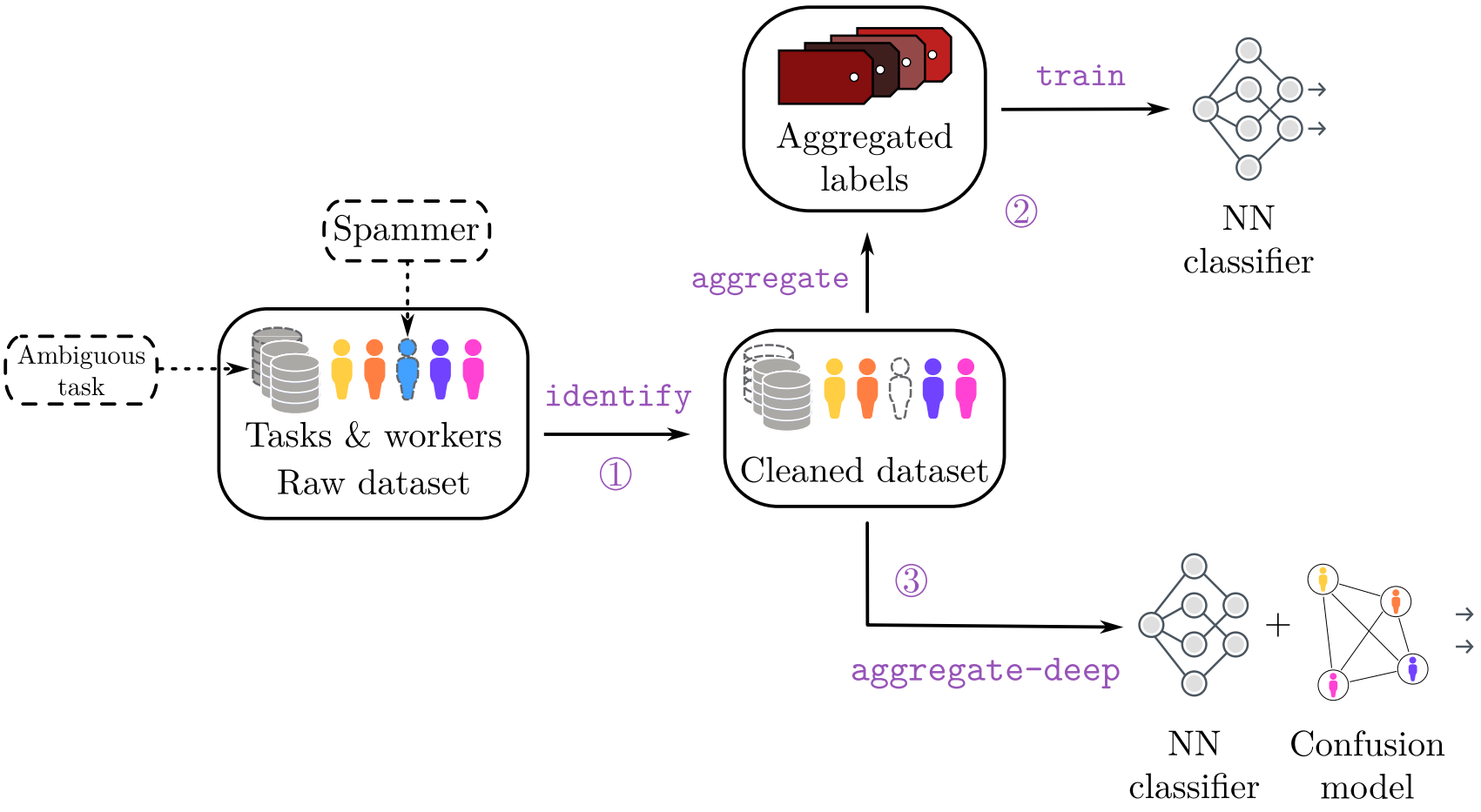
- Workers sort a given task into one of the K classes



- $y_i^{(j)} \in [K] :=$ answer of worker j to task i

From the data to the classifier

The pipeline



Main contributions

- Can we improve performance by leveraging better-quality data

(1) T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

(2) T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Main contributions

- Can we improve performance by leveraging better-quality data
- Can we standardize crowdsourcing dataset's tools in `python` for reproducibility?

(1) T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

(2) T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Main contributions

- Can we improve performance by leveraging better-quality data
- Can we standardize crowdsourcing dataset's tools in `python` for reproducibility?
- Release a real, large-scaled dataset with `Pl@ntNet`

(1) T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

(2) T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Main contributions

- Can we improve performance by leveraging better-quality data
  - Creation of the **WAUM**[1]: a metric to identify ambiguous images
- Can we standardize crowdsourcing dataset's tools in `python` for reproducibility?
- Release a real, large-scaled dataset with `Pl@ntNet`

[1] T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

[2] T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Main contributions

- Can we improve performance by leveraging better-quality data
  - Creation of the **WAUM**[1]: a metric to identify ambiguous images

- Can we standardize crowdsourcing dataset's tools in `python` for reproducibility?
  - **peerannot**[2]: available via `pip`/github

    `https://peerannot.github.io`

- Release a real, large-scaled dataset with `Pl@ntNet`

[1] T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

[2] T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Main contributions

- Can we improve performance by leveraging better-quality data
  - Creation of the **WAUM**[1]: a metric to identify ambiguous images

- Can we standardize crowdsourcing dataset's tools in `python` for reproducibility?
  - **peerannot**[2]: available via `pip`/github

    `https://peerannot.github.io`

- Release a real, large-scaled dataset with `Pl@ntNet`
  - Creation and evaluation of a **new benchmark dataset**

---

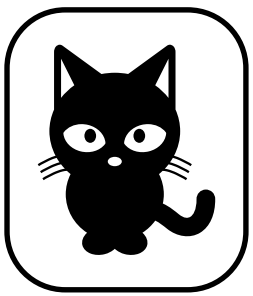
[1] T. Lefort et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

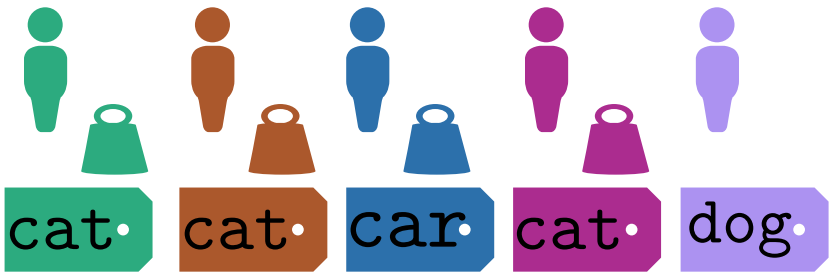
[2] T. Lefort et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.

Classical aggregation strategy
(Weighted) Majority Votes

$$\hat{y}_i^{\text{WMV}} = \underset{k \in [K]}{\text{argmax}} \sum_{j \in \mathcal{A}(x_i)} \text{⚖}_j \, \mathbb{1}(y_i^{(j)} = k)$$

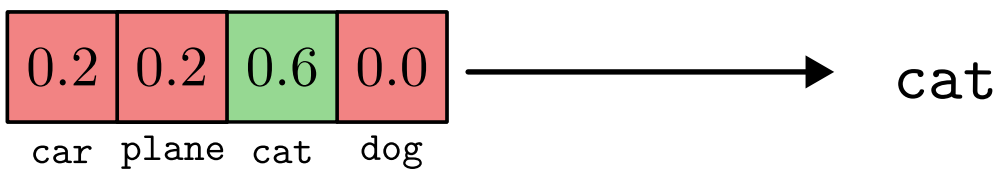
For example with equal weights:





0: car 2: cat

1: plane 3: dog



Classical aggregation strategy
(Weighted) Majority Votes

Many existing weight choices:

- WAWA[3]:

$$\text{weight}(w_j) = \text{Accuracy}(\{y_i^{(j)}\}_i, \{\hat{y}_i{}^{\text{MV}}\}_i)$$

- Shapley based[4]

- Matrix completion (MACE)[5] ...

Pros: "simple" weight can scale to large datasets and be easy to interpret
Cons: Can not capture worker skills in detail

---

[3] `https://success.appen.com/hc/en-us/articles/202703205-Calculating-Worker-Agreement-with-Aggregate-Wawa`

[4] T. Lefort et al. (July 2024b). "Weighted majority vote using Shapley values in crowdsourcing". In: *CAp 2024 - Conférence sur l'Apprentissage Automatique*. Lille, France.

[5] D. Hovy et al. (2013). "Learning whom to trust with MACE". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1120–1130.

Classical aggregation strategy

Dawid and Skene[6]

- Represent workers from their pairwise confusions matrix $\pi^{(j)} \in \mathbb{R}^{K \times K}$
- Probabilistic model on their answers:

$$y^{(j)} | y^\star \sim \text{Multinomial}(\pi^{(j)}_{y^\star, \bullet})$$

with $\pi^{(j)}_{k,\ell} = \mathbb{P}(\text{worker } j \text{ answers } \ell \text{ with unknown truth} k)$

Pros:

- Finer modelisation
- Can use adversarial workers

Cons:

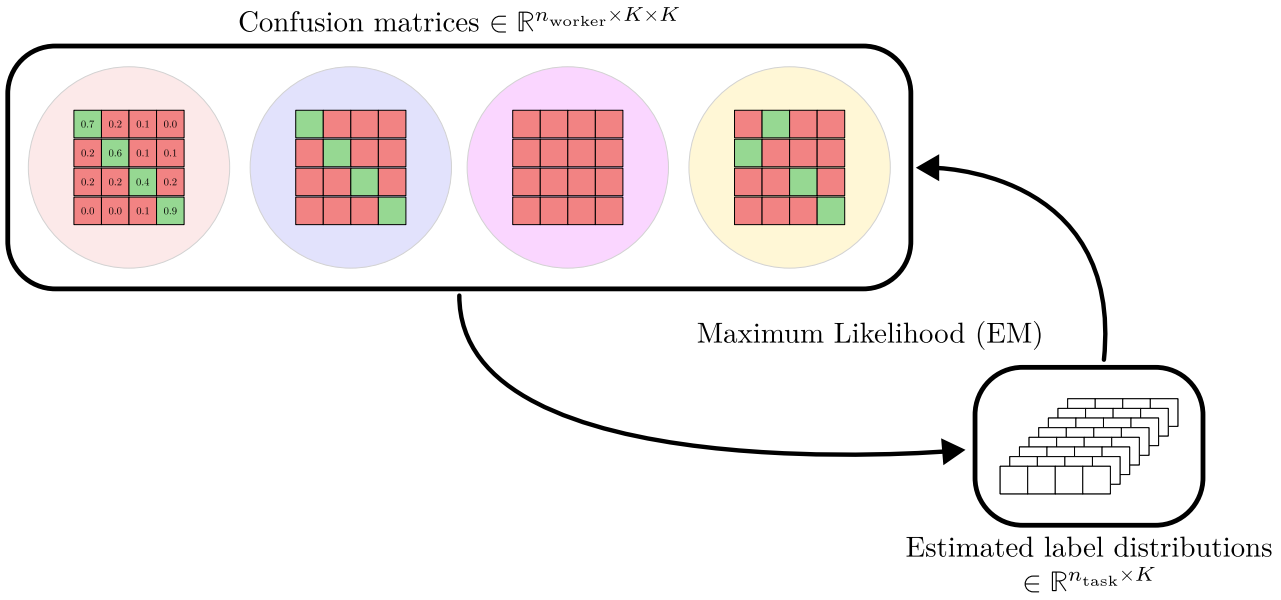
- Memory issue: $n_{\texttt{worker}} \times K^2$ parameters to estimate only the confusion matrices

[6] A. Dawid and A. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.

Classical aggregation strategy

Dawid and Skene – Model

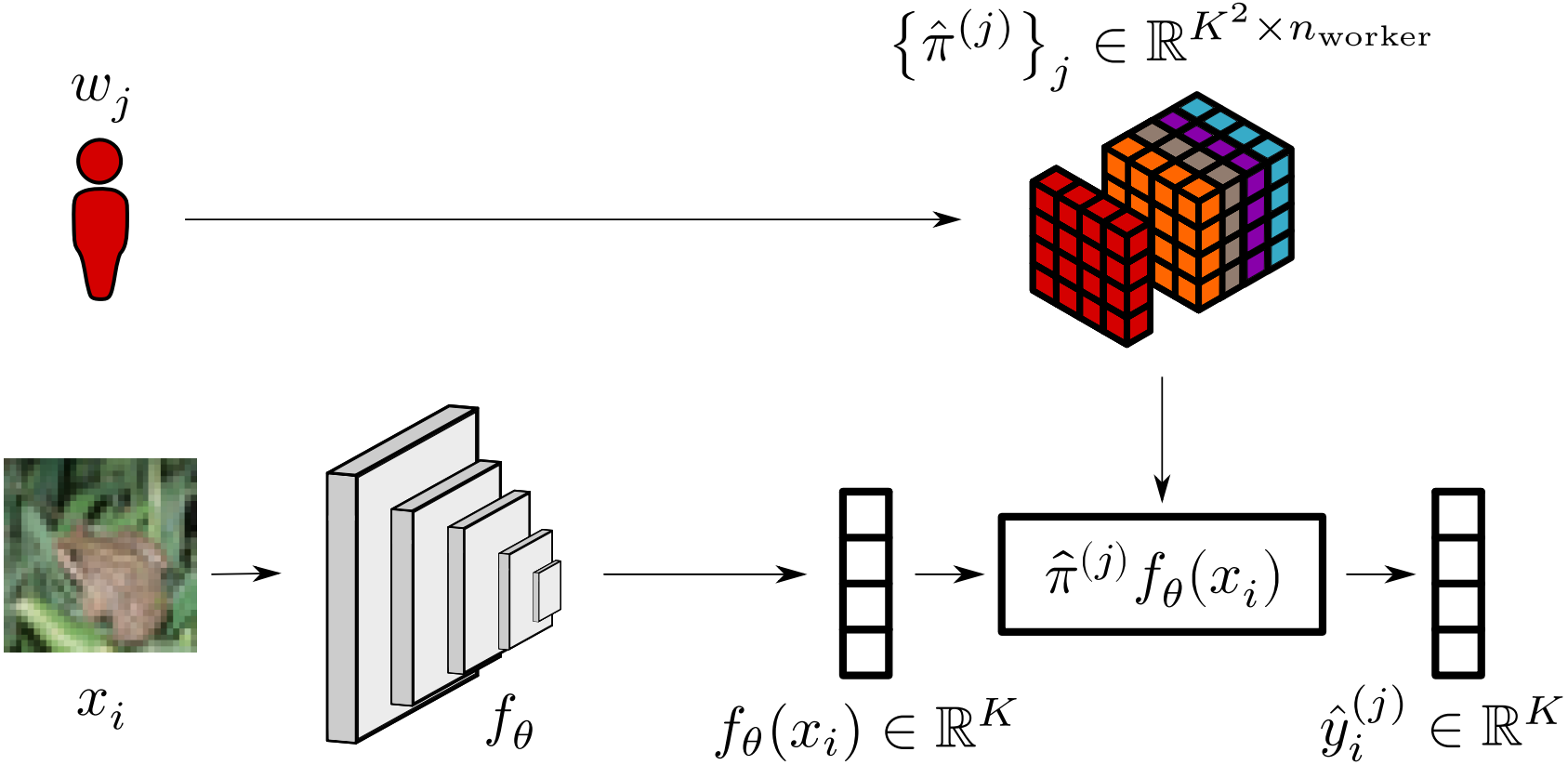
Probabilistic model $\longrightarrow$ Likelihood (to maximize)



Classical deep-learning strategy

CrowdLayer[7]

- Idea: put the DS confusion matrix in a neural network as a new layer

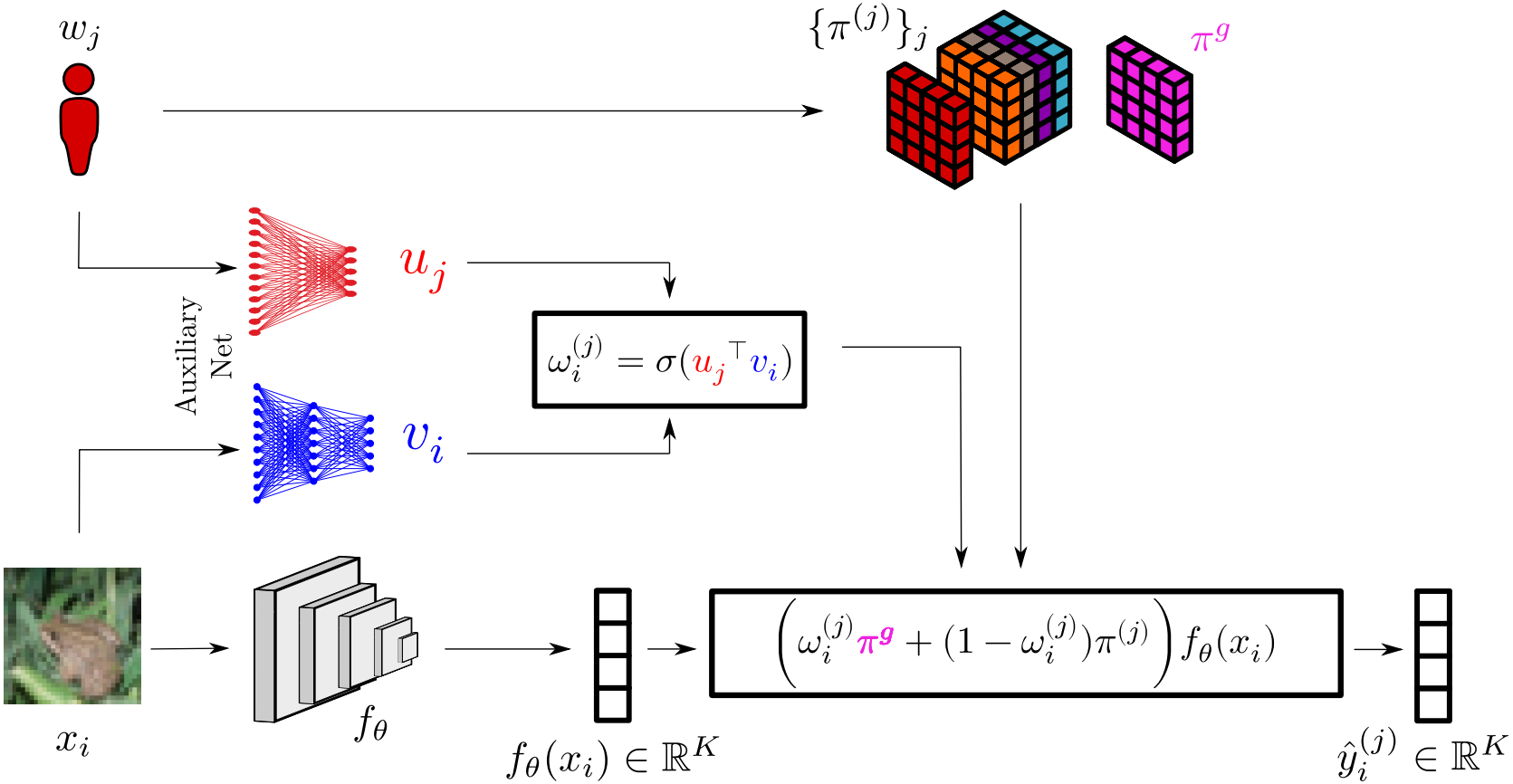


[7] F. Rodrigues and F. Pereira (2018). "Deep learning from crowds". In: *AAAI*. vol. 32.

Classical deep-learning strategy

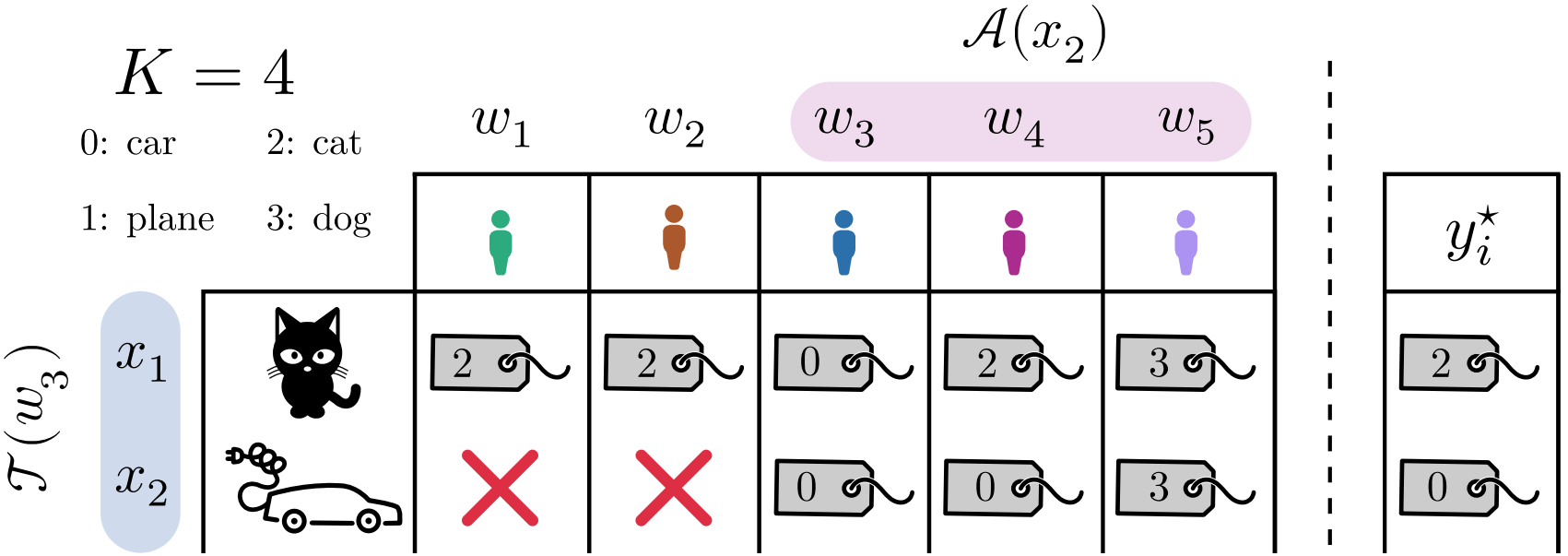
CoNAL[8]

- Idea: CrowdLayer + global and local confusions

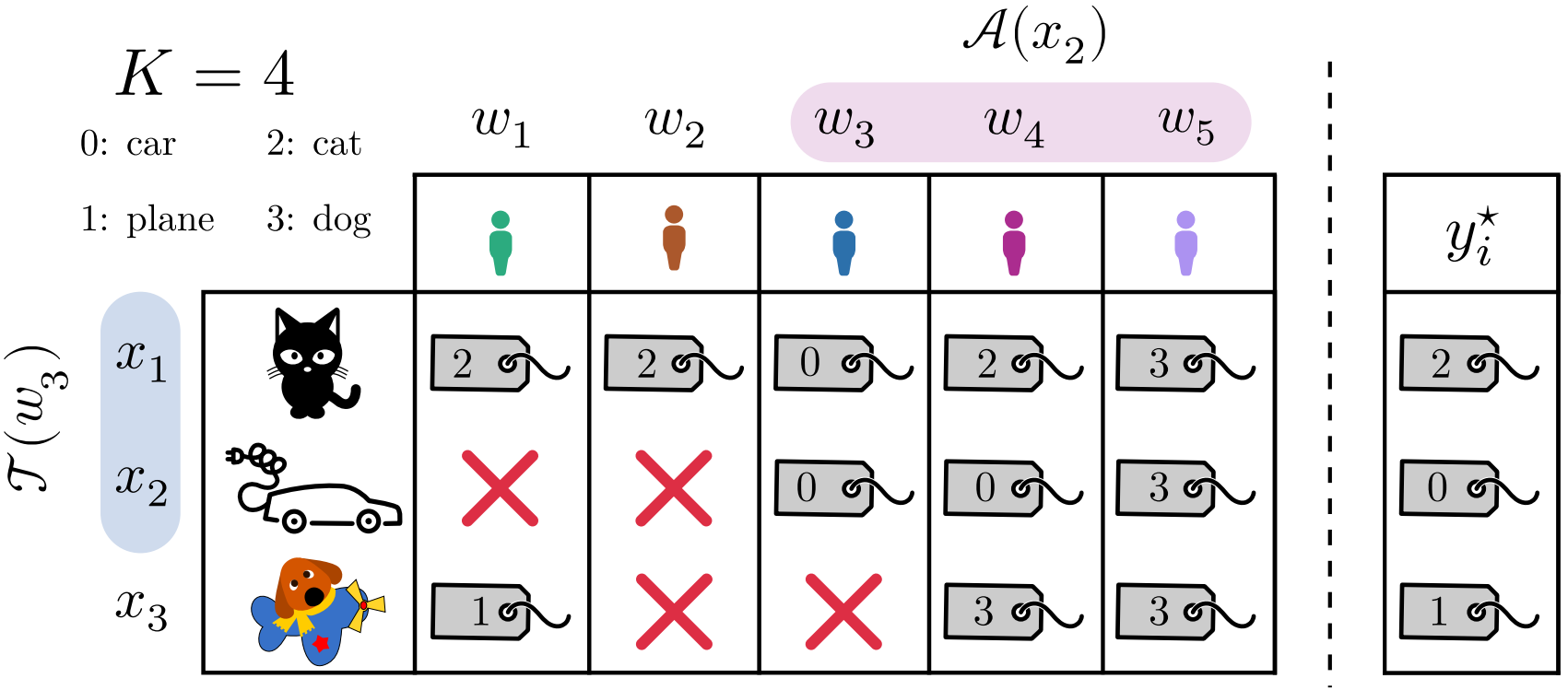


[8] Z. Chu, J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". In: *AAAI*, pp. 5832–5840.

When images have underlying ambiguity



When images have underlying ambiguity



Ambiguity in classical supervised setting
Area Under the Margin (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$

- AUM[9]: monitor margin during training
- Classifier: at epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of scores (logits)

Average = Stability

Margin between scores: content of Hinge loss

$$\mathrm{AUM}(x_i, y_i) = \frac{1}{T} \sum_{t=1}^{T} \left[\mathcal{C}^{(t)}(x_i)_{y_i} - \max_{\ell \neq y_i} \mathcal{C}^{(t)}(x_i)_\ell \right]$$

Score of assigned label

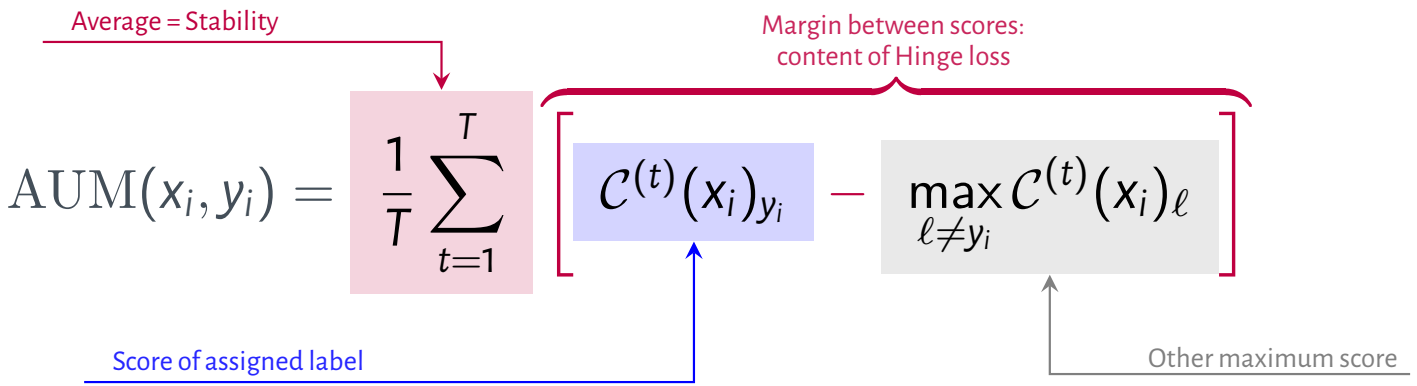
Other maximum score

[9] G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.

Ambiguity in classical supervised setting
Area Under the Margin (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$

- AUM[9]: monitor margin during training
- Classifier: at epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of scores (logits)



Challenging for crowdsourcing:

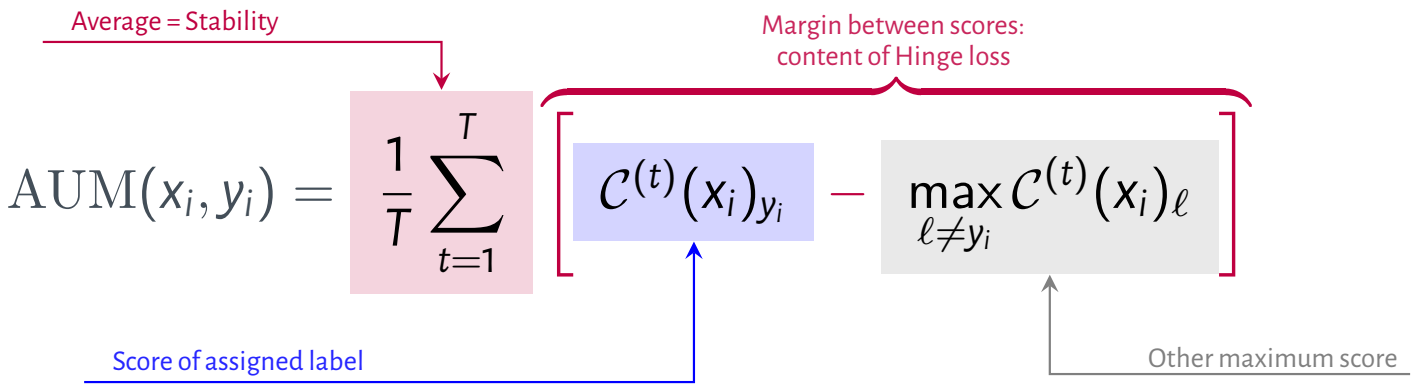
- No single y_i, multiple $y_i^{(j)}$: one for each worker w_j answering task x_i

[9] G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.

Ambiguity in classical supervised setting
Area Under the Margin (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$

- AUM[9]: monitor margin during training
- Classifier: at epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of scores (logits)



Challenging for crowdsourcing:

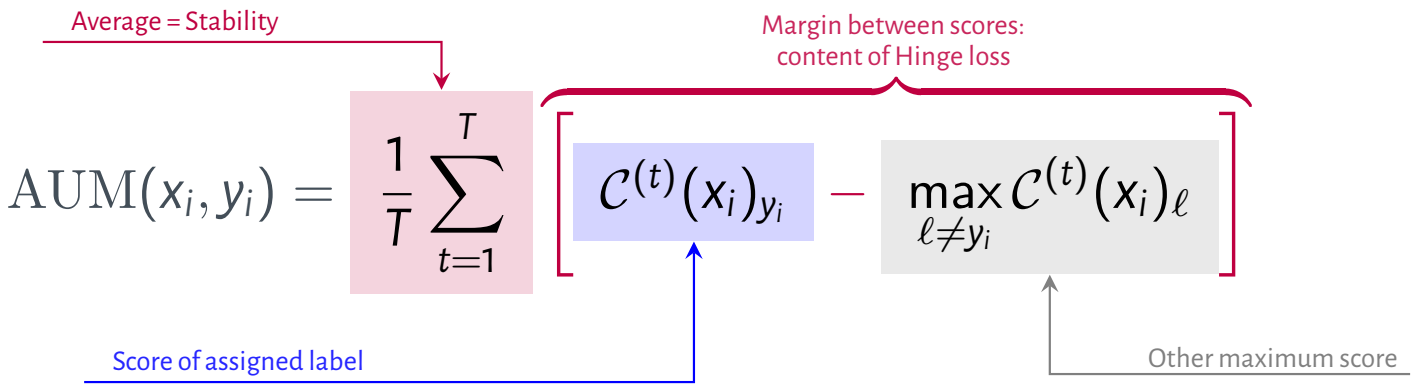
- No single y_i, multiple $y_i^{(j)}$: one for each worker w_j answering task x_i
  - …so $\mathcal{C}^{(t)}(x_i)_{y_i}$ does not exist

[9] G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.

Ambiguity in classical supervised setting
Area Under the Margin (AUM)

Goal: identify issues in classical datasets $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [K]$

- AUM[9]: monitor margin during training
- Classifier: at epoch $t \in [T]$, $\mathcal{C}^{(t)}(x_i) \in \mathbb{R}^K$ a vector of scores (logits)



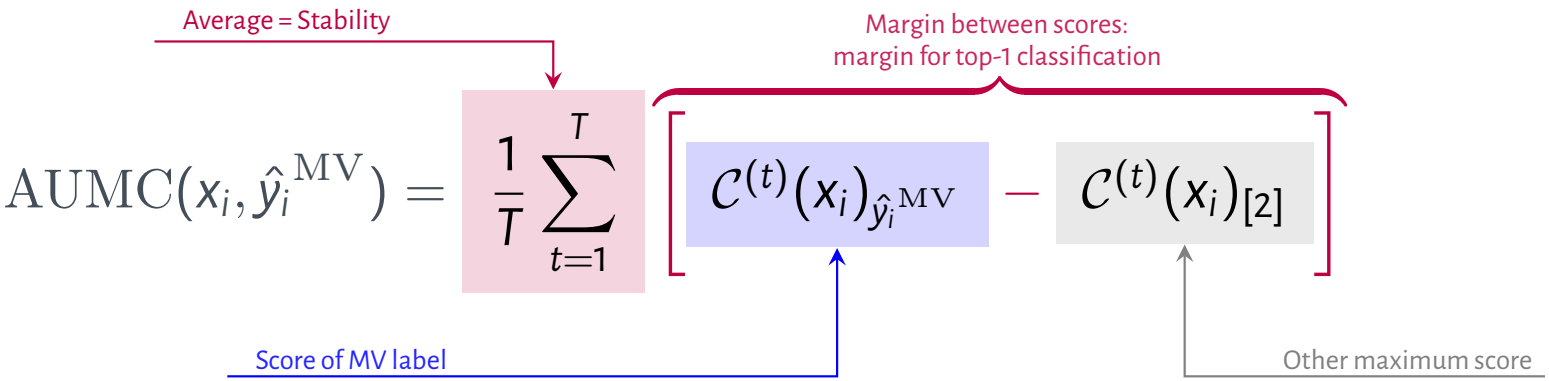
Challenging for crowdsourcing:

- No single y_i, multiple $y_i^{(j)}$: one for each worker w_j answering task x_i
  - …so $\mathcal{C}^{(t)}(x_i)_{y_i}$ does not exist
  - …and same issue with $\ell \neq y_i$.

[9] G. Pleiss et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.

Going to the crowdsourcing setting
AUMC

Naive extension: Use the MV label $\hat{y}_i^{\mathrm{MV}}$ instead of the unknown y_i + use previous work of margins' properties [10]



Issue:

- Lose all worker-related information
- Sensitive to poorly performing workers

[10] M. Lapin, M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". In: *CVPR*, pp. 1468–1477.

Going to the crowdsourcing setting
WAUM

Weighted Areas Under the Margins:

- Scale effects in the scores discarded, need normalization [11]

[11] C. Ju, A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.

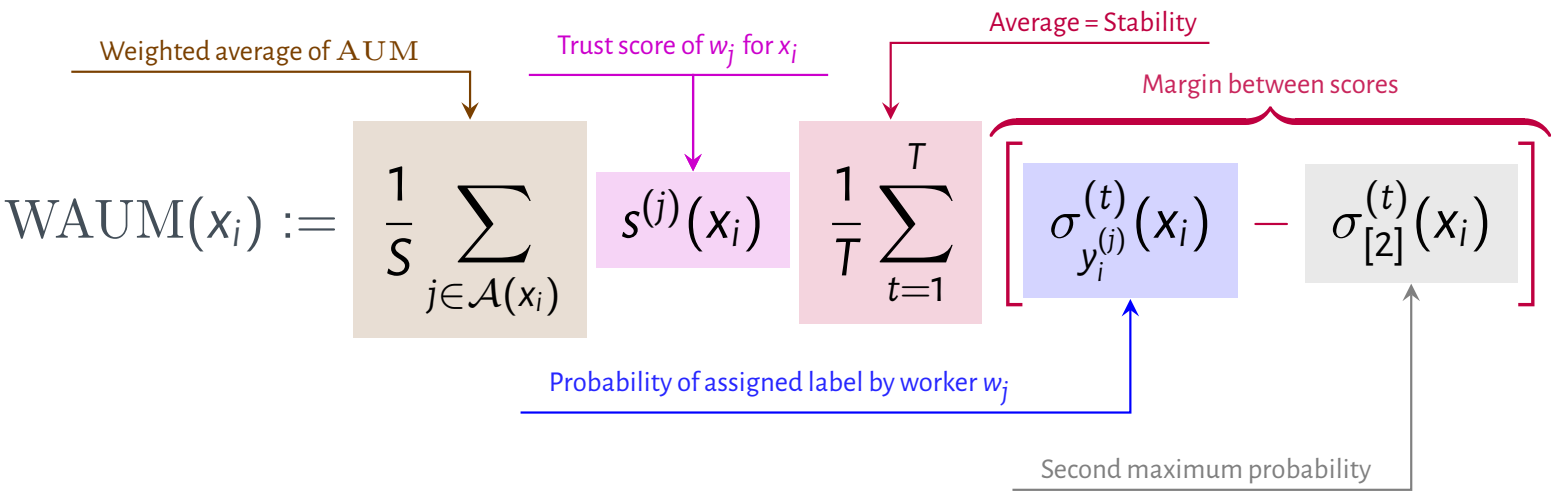
Going to the crowdsourcing setting
WAUM

Weighted Areas Under the Margins:

- Scale effects in the scores discarded, need normalization [11]

With:

- $\sigma(x_i) = \sigma(\mathcal{C}(x_i)) \in \Delta_{K-1}$ (simplex of dim $K-1$)
- Softmax ordered: $\sigma_{[1]}(x_i) \geq \cdots \geq \sigma_{[K]}(x_i) > 0$



[11] C. Ju, A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.

Weights in the WAUM

Leverage both tasks and labels

Our chosen worker/task score:

- Consider a score of the form: worker term $\times$ task term (similar to GLAD[12])
- Use multidimensionality of DS confusion matrices
- Use a neural network as a control agent to measure task difficulty[13]

[12] J. Whitehill et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. vol. 22.

[13] M. Servajean et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.

Weights in the WAUM

Leverage both tasks and labels

Our chosen worker/task score:

- Consider a score of the form: worker term $\times$ task term (similar to GLAD[12])
- Use multidimensionality of DS confusion matrices
- Use a neural network as a control agent to measure task difficulty[13]

$$s^{(j)}(x_i) = \left\langle \operatorname{diag}(\hat{\pi}^{(j)}) \mid \sigma^{(T)}(x_i) \right\rangle \in [0,1]$$

Worker j overall ability

Difficulty of task i

[12] J. Whitehill et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise". In: *NeurIPS*. vol. 22.

[13] M. Servajean et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\mathrm{AUM}(x_i, y_i^{(j)})$

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\mathrm{AUM}(x_i, y_i^{(j)})$
- Compute trust scores $s^{(j)}(x_i)$

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\mathrm{AUM}(x_i, y_i^{(j)})$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j \in \mathcal{A}(x_i)} s^{(j)}(x_i) \mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j' \in \mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

Computing the WAUM

The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\mathrm{AUM}(x_i, y_i^{(j)})$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j \in \mathcal{A}(x_i)} s^{(j)}(x_i) \mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j' \in \mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.01$)
- **Estimate confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset

Computing the WAUM

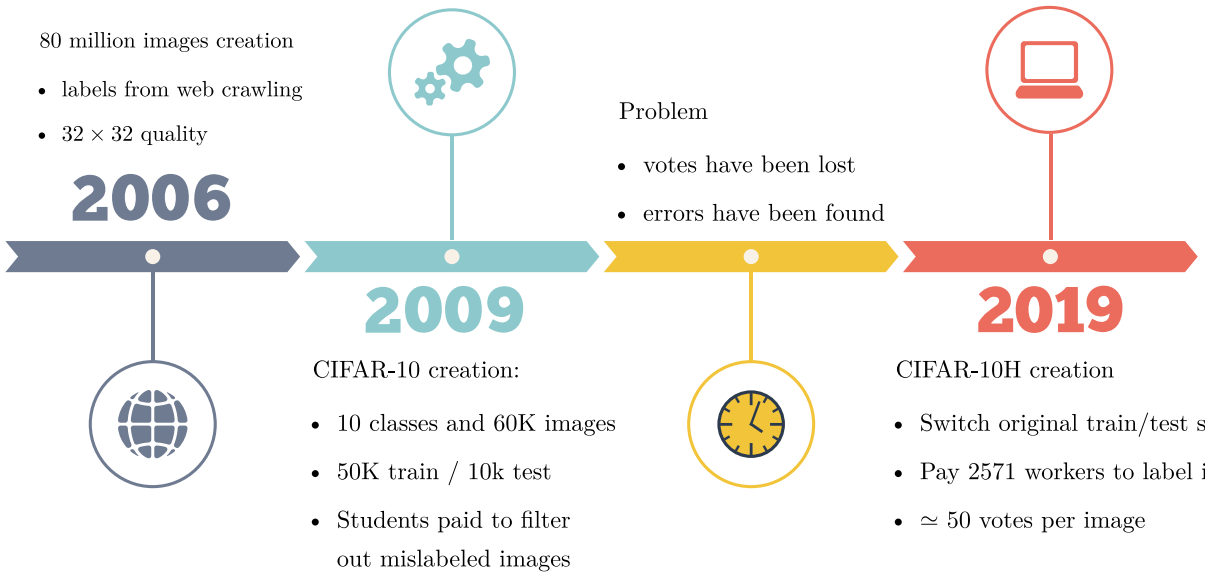
The pipeline summarized

- Estimate confusion matrices $\pi^{(j)} \in \mathbb{R}^{K \times K}$, for all $j \in [n_{\texttt{worker}}]$
- Train a network on all crowdsourced task/label pairs: $(x_i, y_i^{(j)})$
- Compute all $\mathrm{AUM}(x_i, y_i^{(j)})$
- Compute trust scores $s^{(j)}(x_i)$
- For each task compute $\mathrm{WAUM}(x_i) = \dfrac{\sum_{j \in \mathcal{A}(x_i)} s^{(j)}(x_i) \mathrm{AUM}(x_i, y_i^{(j)})}{\sum_{j' \in \mathcal{A}(x_i)} s^{(j')}(x_i)}$

Usage (for learning):

- **Prune** x_i's with $\mathrm{WAUM}(x_i)$ below quantile q_α (say $\alpha = 0.01$)
- **Estimate confusion matrices** $\hat{\pi}^{(j)}$ on pruned training dataset
- **Aggregate** labels and **Train** a classifier on the newly pruned dataset

Presenting CIFAR-10H[14] dataset



Labels: cat, dog, car, plane, bird, horse, frog, deer, ship, truck

[14] J. C. Peterson et al. (2019). "Human Uncertainty Makes Classification More Robust". In: *ICCV*, pp. 9617–9626.

Presenting CIFAR-10H[14] dataset

2006

80 million images creation

- labels from web crawling
- 32×32 quality

2009

CIFAR-10 creation:

- 10 classes and 60K images
- 50K train / 10k test
- Students paid to filter out mislabeled images

Problem

- votes have been lost
- errors have been found

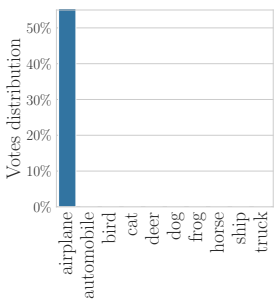
2019

CIFAR-10H creation

- Switch original train/test sets
- Pay 2571 workers to label images
- $\simeq$ 50 votes per image

Labels: cat, dog, car, plane, bird, horse, frog, deer, ship, truck

Image #7681
CIFAR-10 label: airplane



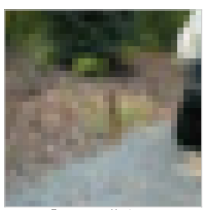
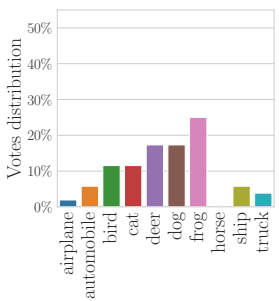


Image #6750
CIFAR-10 label: deer



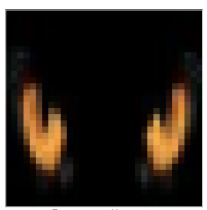
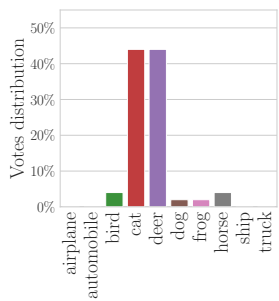


Image #9246
CIFAR-10 label: cat



[14] J. C. Peterson et al. (2019). "Human Uncertainty Makes Classification More Robust". In: *ICCV*, pp. 9617–9626.

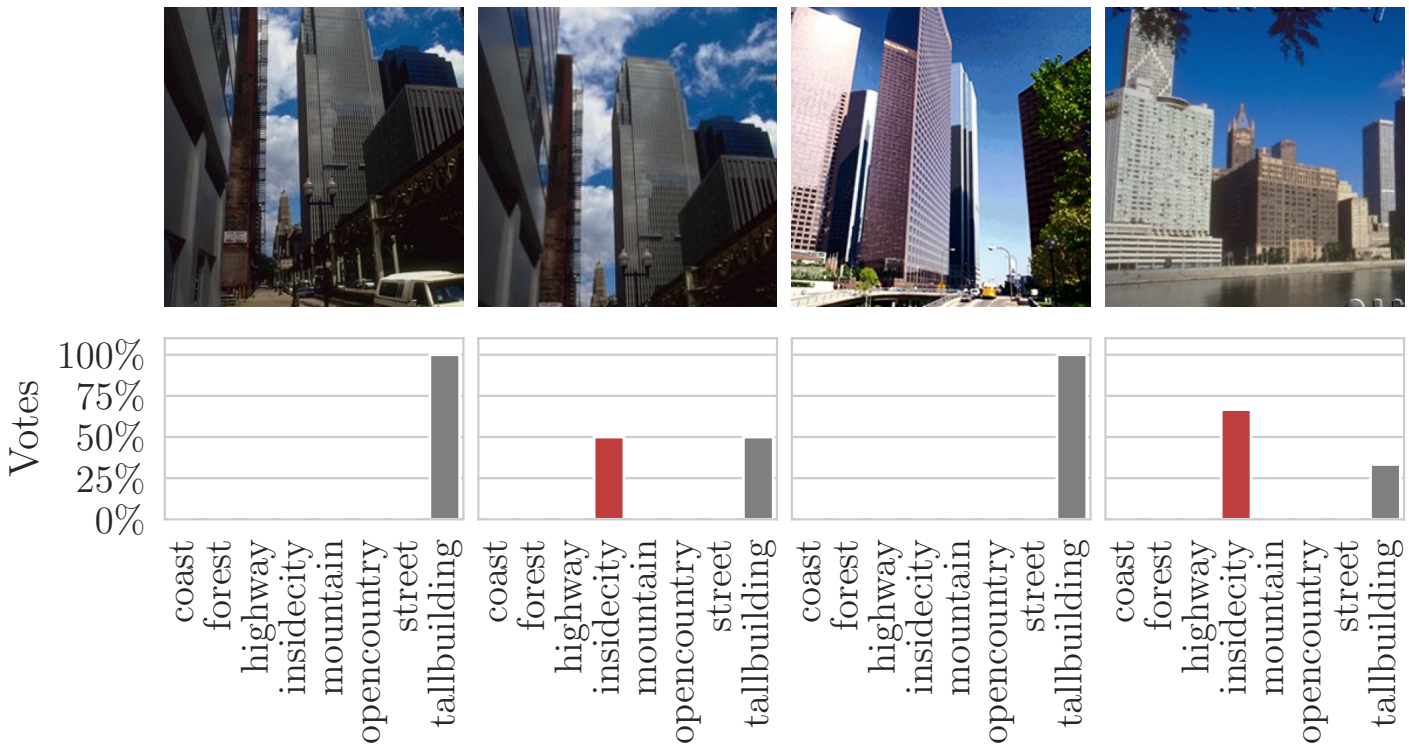
Presenting LabelMe dataset[15]

- 1000 training / 500 validation / 1188 test images
- 59 workers: each task has up to 3 votes
- 8 classes: highway, insidecity, tallbuilding, street, forest, coast, mountain, opencountry

[15] F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.

Presenting LabelMe dataset[15]

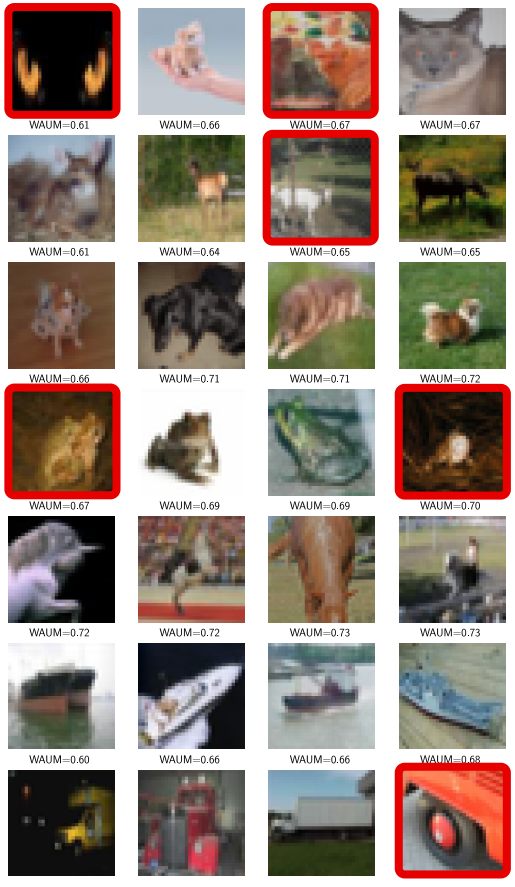
- 1000 training / 500 validation / 1188 test images
- 59 workers: each task has up to 3 votes
- 8 classes: highway, insidecity, tallbuilding, street, forest, coast, mountain, opencountry



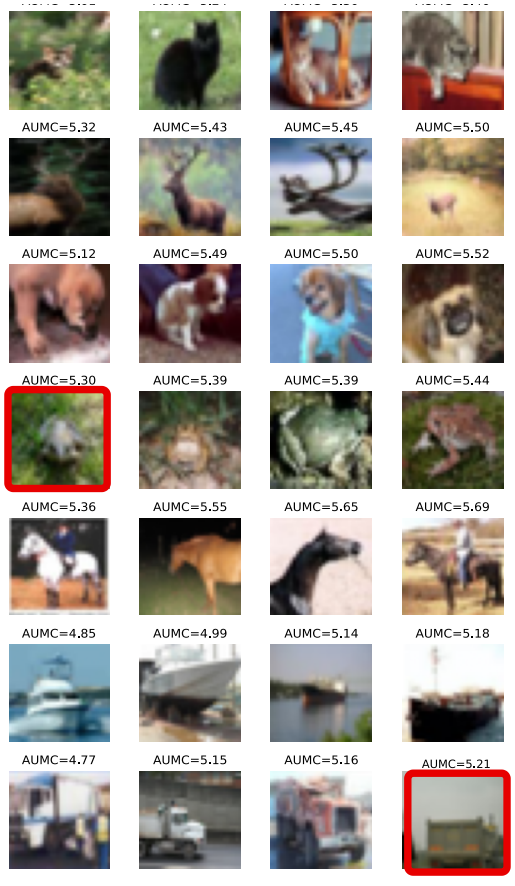
[15] F. Rodrigues, F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.

Qualitative results

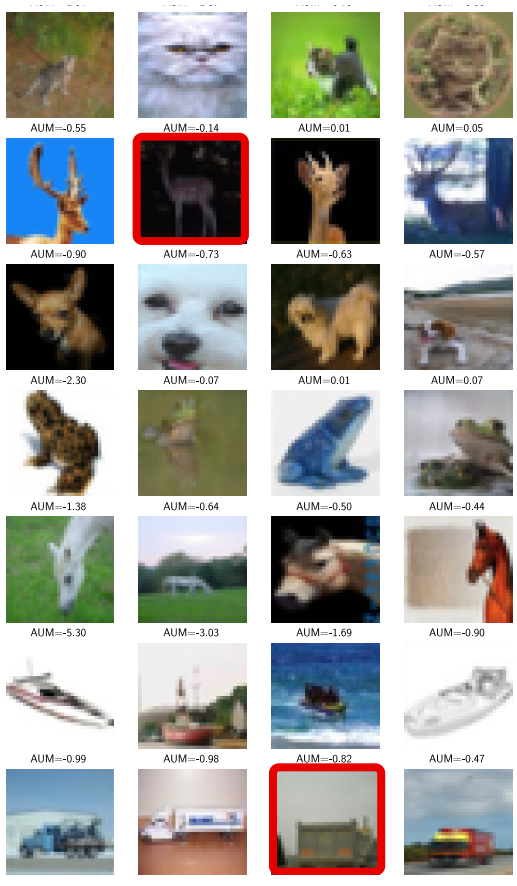
WAUM



AUMC

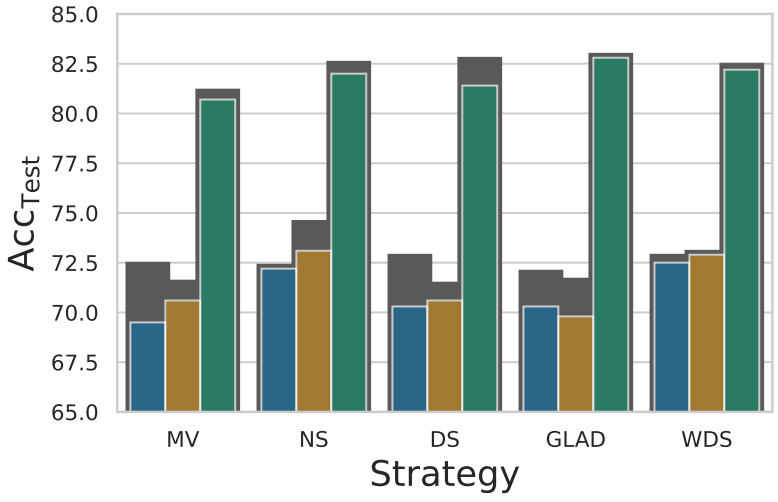


AUM

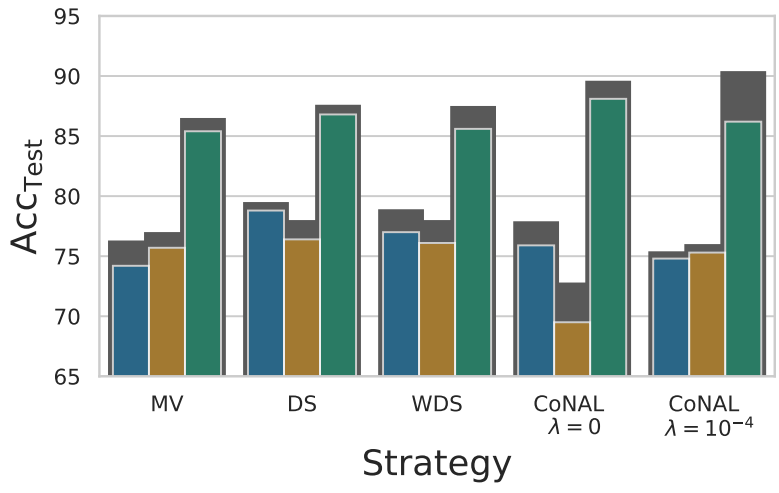


Ablation study

CIFAR-10H



LabelMe



In short

- Introduced the WAUM to find ambiguous images
- Better quality data can improve performance

In short

- Introduced the WAUM to find ambiguous images
- Better quality data can improve performance

Towards large-scale problems

- DS model and confusion matrices do not scale
- What is currently done in large-scale settings?
- Can we evaluate their performance?

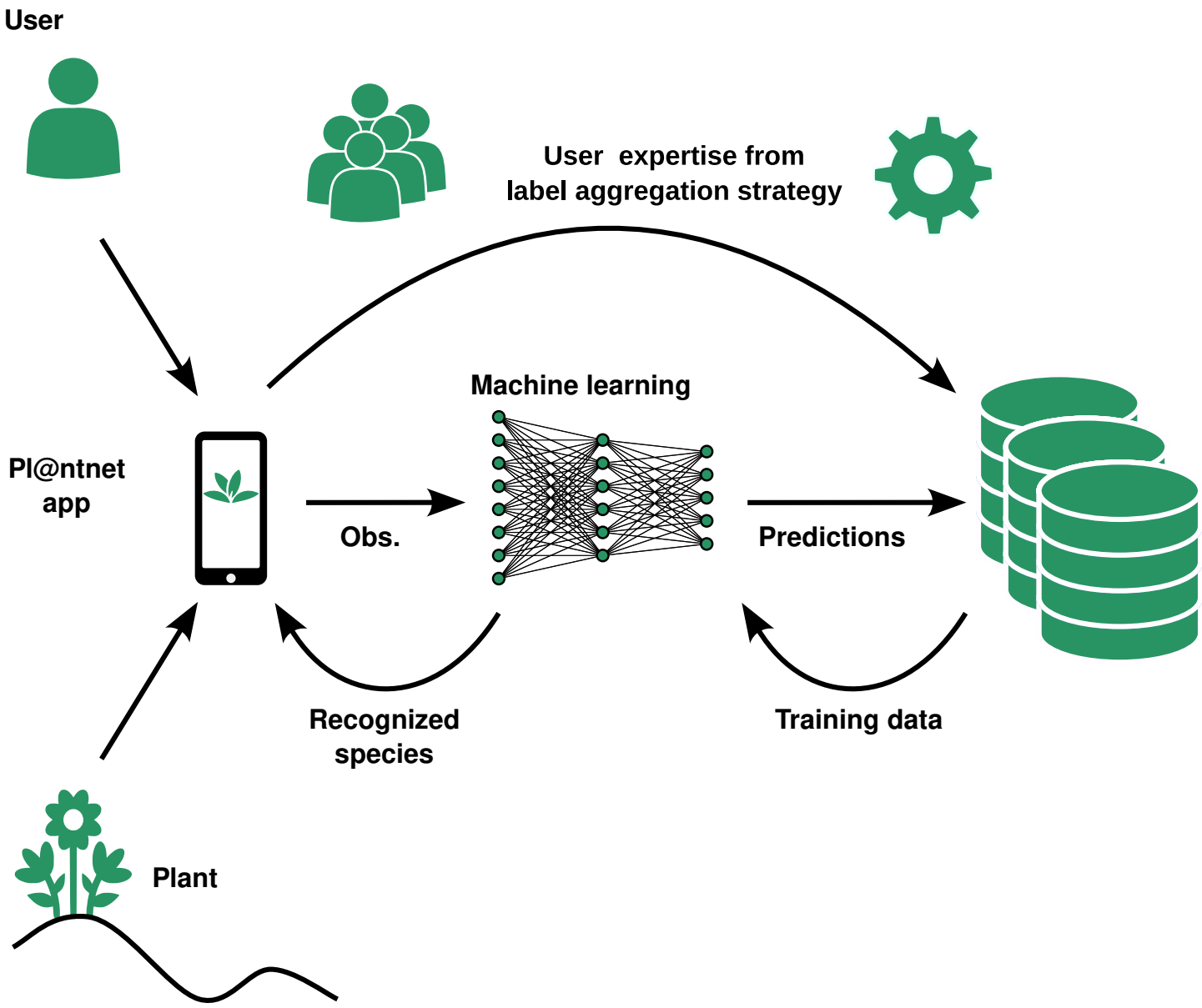
In short

- Introduced the WAUM to find ambiguous images
- Better quality data can improve performance

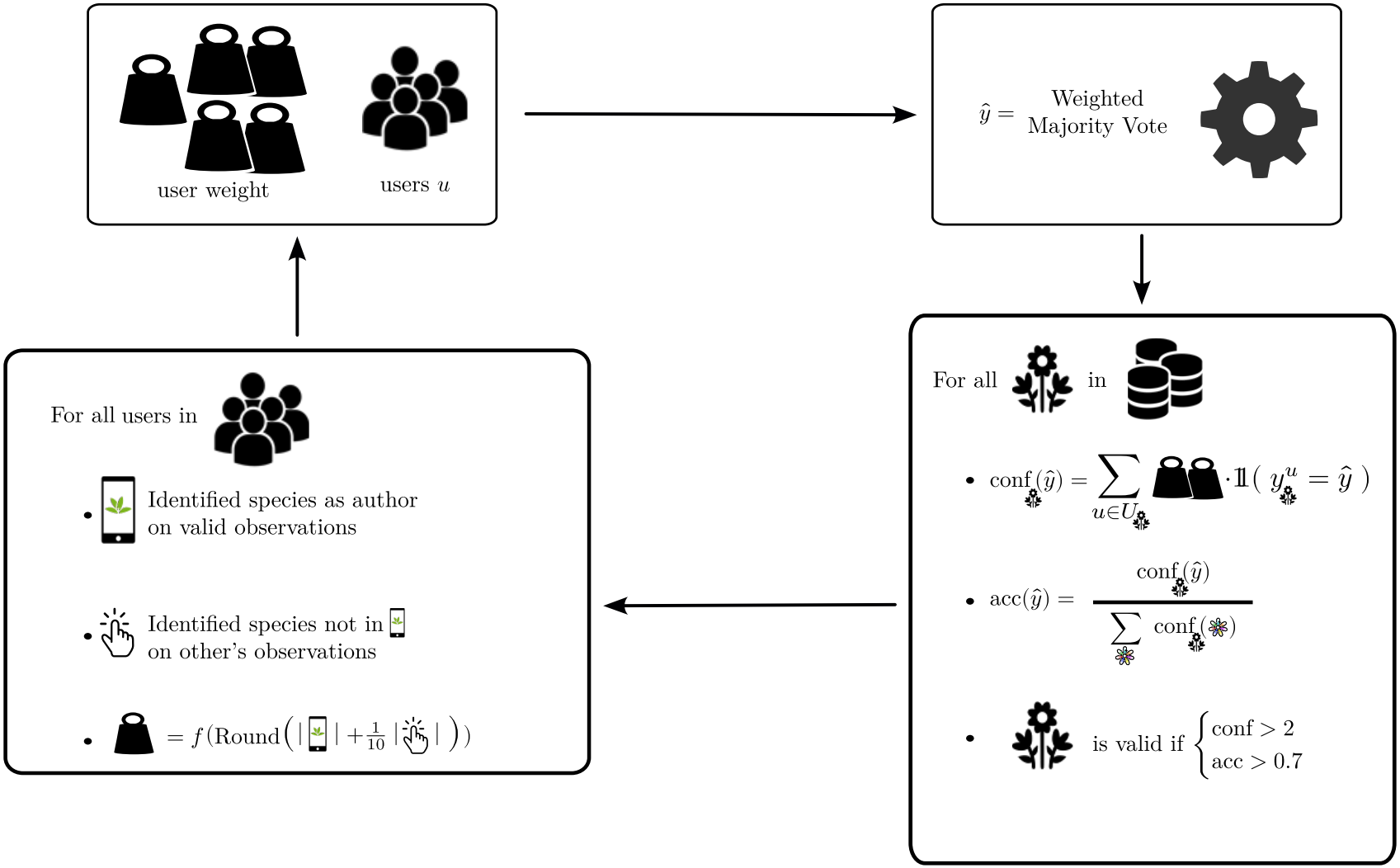
Towards large-scale problems

- DS model and confusion matrices do not scale
- What is currently done in large-scale settings?
- Can we evaluate their performance?
  - To evaluate we need the dataset!

Presenting Pl@ntNet pipeline



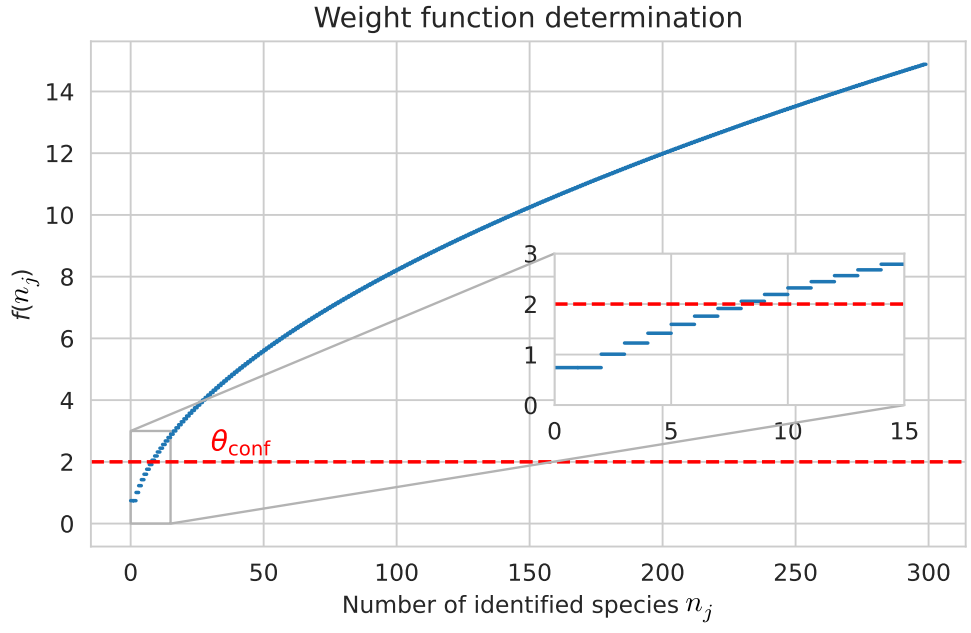
Pl@ntNet aggregation strategy



Pl@ntNet aggregation strategy

Weight function

$$f(n_j) = n_j^{\alpha} - n_j^{\beta} + \gamma \text{ with } \begin{cases} \alpha & = 0.5 \\ \beta & = 0.2 \\ \gamma & \simeq 0.74 \end{cases}$$

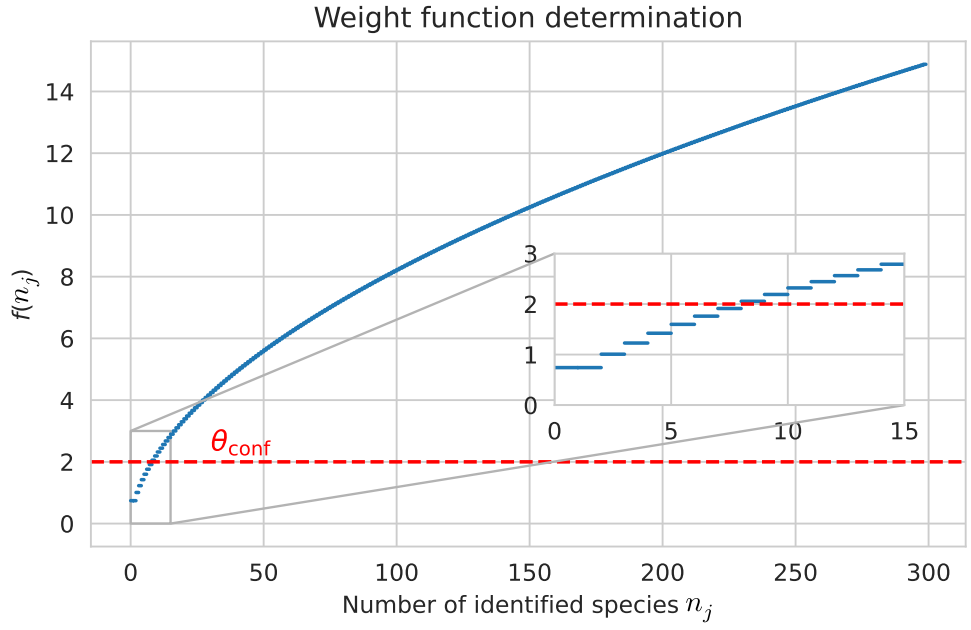


- With 8 identified species one becomes self-validating

Pl@ntNet aggregation strategy

Weight function

$$f(n_j) = n_j^{\alpha} - n_j^{\beta} + \gamma \text{ with } \begin{cases} \alpha & = 0.5 \\ \beta & = 0.2 \\ \gamma & \simeq 0.74 \end{cases}$$

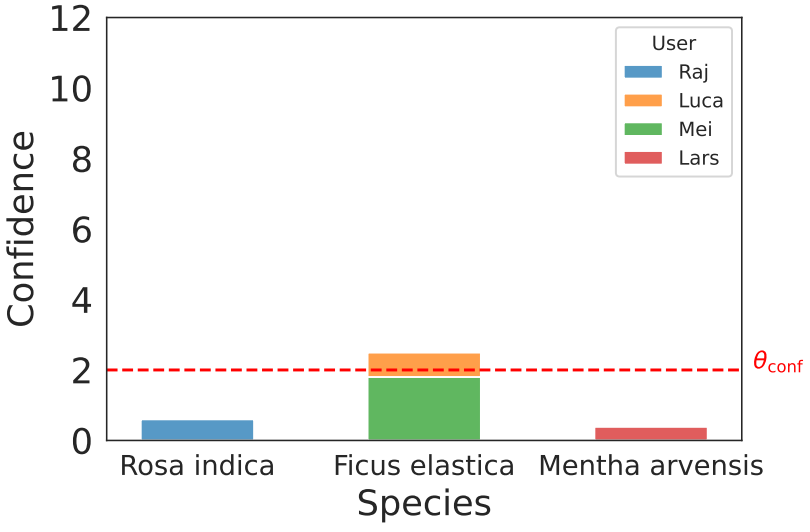


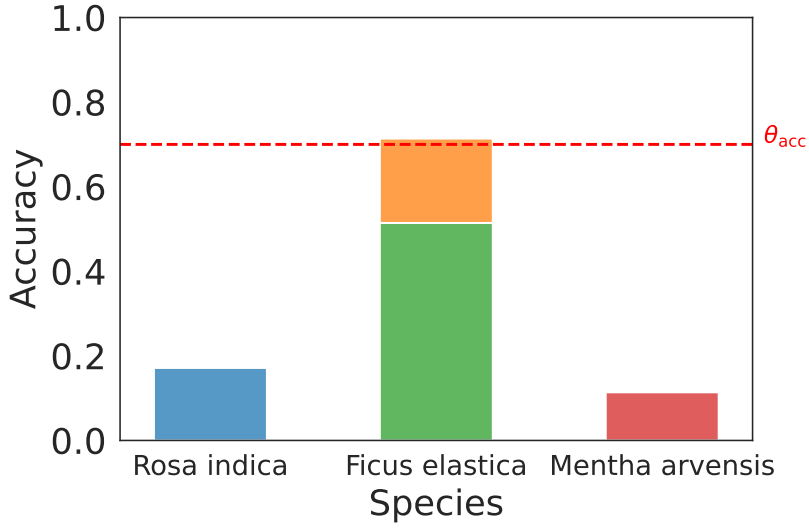
- With 8 identified species one becomes self-validating
- But observations can be invalidated at any time in the future

Pl@ntNet aggregation strategy

Examples

Initial setting

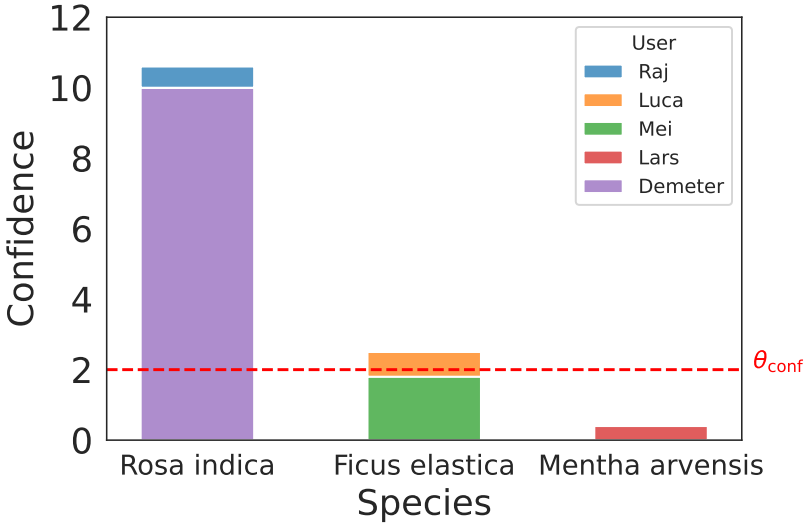


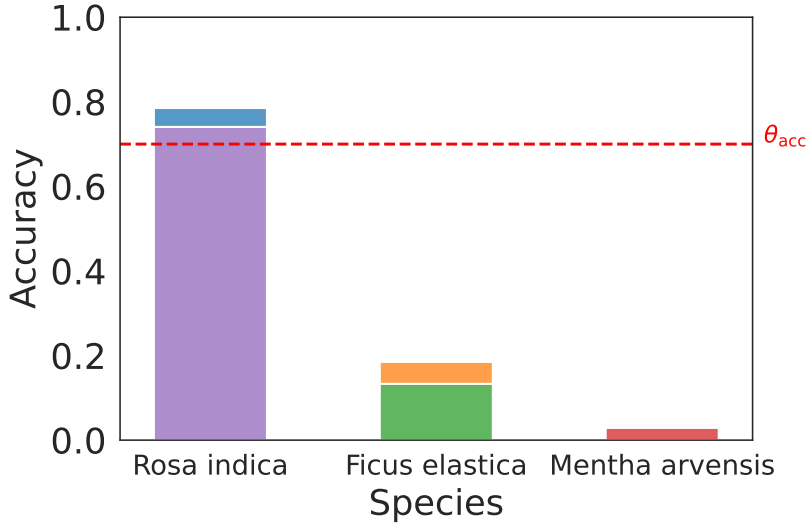


Pl@ntNet aggregation strategy

Examples

Label switch

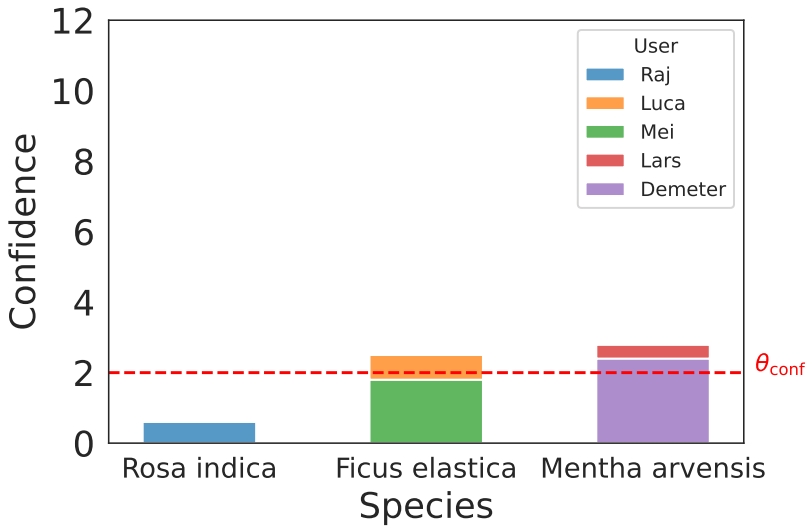


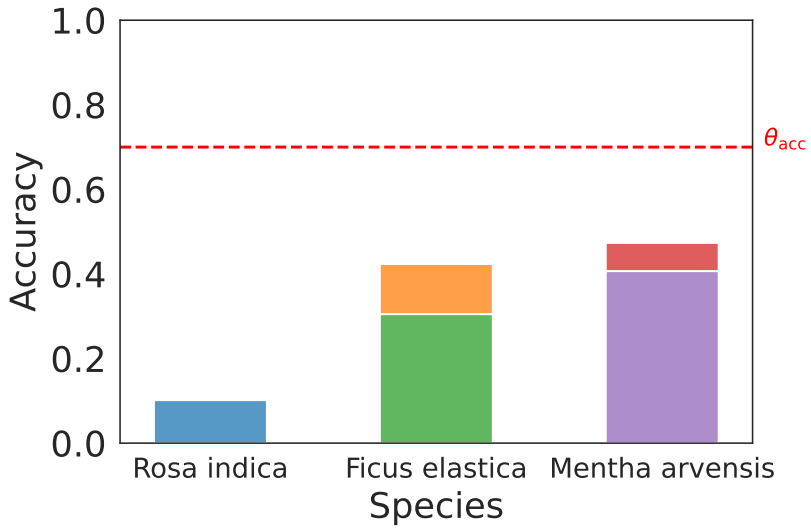


Pl@ntNet aggregation strategy

Examples

Invalidate





Realeasing a new dataset

- South Western European flora obs since 2017
- 823 000 users answered more than 11000 species
- 6 700 000 observations
- 9 000 000 votes casted
- **Imbalance**: 80% of observations are represented by 10% of total votes

Realeasing a new dataset

- South Western European flora obs since 2017
- 823 000 users answered more than 11000 species
- 6 700 000 observations
- 9 000 000 votes casted
- **Imbalance**: 80% of observations are represented by 10% of total votes

- Extraction of 98 experts (TelaBotanic + prior knowledge – thanks to Pierre Bonnet)

Compared strategies

- **Majority Vote** (MV)

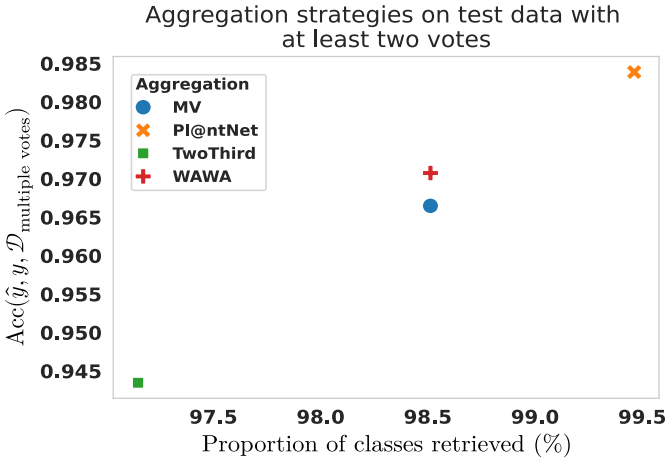
Compared strategies

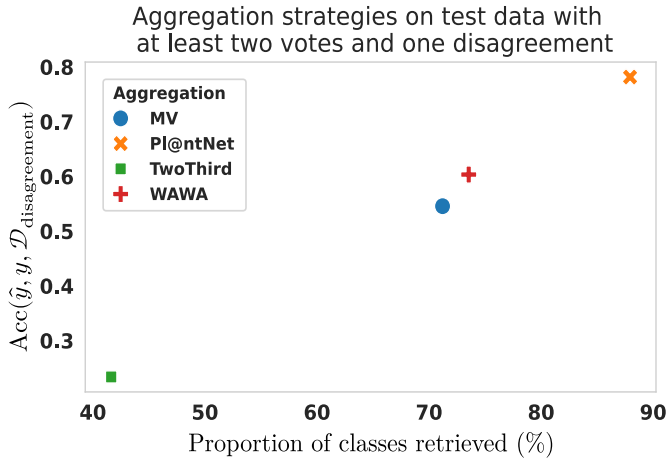
- **Majority Vote** (MV)
- **Worker agreement with aggregate (WAWA)** (**appen**)
  - Majority vote
  - Weight user by how much they agree with the majority
  - Weighted majority vote

Compared strategies

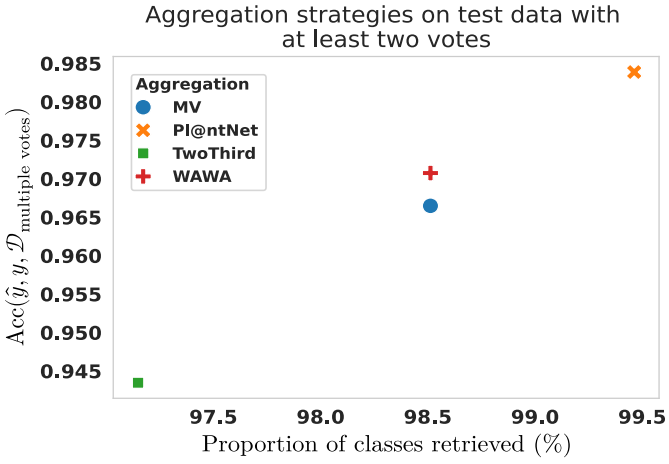
- **Majority Vote** (MV)
- **Worker agreement with aggregate (WAWA)** (**appen**)
  - Majority vote
  - Weight user by how much they agree with the majority
  - Weighted majority vote
- **iNaturalist**
  - Need 2 votes
  - 2/3 of agreements

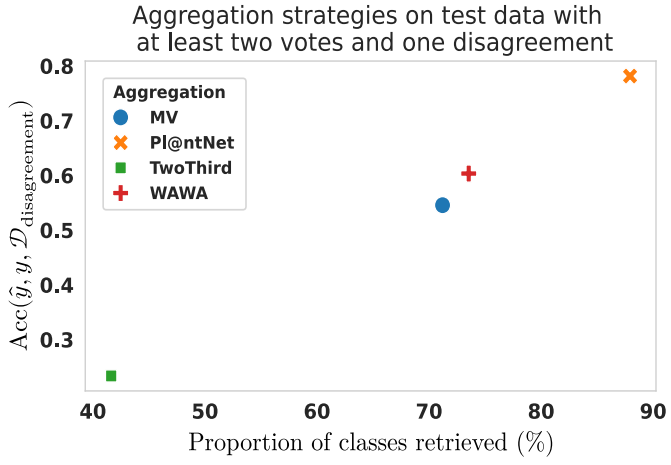
Results


Aggregation strategies on test data with at least two votes
Aggregation
MV
Pl@ntNet
TwoThird
WAWA
0.985
0.980
0.975
0.970
0.965
0.960
0.955
0.950
0.945
97.5
98.0
98.5
99.0
99.5
Proportion of classes retrieved (%)
$\mathrm{Acc}(\hat{y}, y, \mathcal{D}_{\text{multiple votes}})$


Aggregation strategies on test data with at least two votes and one disagreement
Aggregation
MV
Pl@ntNet
TwoThird
WAWA
0.8
0.7
0.6
0.5
0.4
0.3
40
50
60
70
80
90
Proportion of classes retrieved (%)
$\mathrm{Acc}(\hat{y}, y, \mathcal{D}_{\text{disagreement}})$

Results





In short

- Pl@ntNet aggregation performs better overall
- TwoThird is highly impacted by their reject threshold
- In ambiguous settings (right), strategies weighting users are better

More experiments

- AI vote's integration

- Penalizing user's mistakes

More experiments

- AI vote's integration
  - include AI's vote in the aggregation
  - Risks of model collapse
  - Results suggest using AI's vote if the score is above a threhsold $\theta_{\text{score}} = 0.7$ can improve performance
- Penalizing user's mistakes

More experiments

- AI vote's integration
  - include AI's vote in the aggregation
  - Risks of model collapse
  - Results suggest using AI's vote if the score is above a threhsold $\theta_{\text{score}} = 0.7$ can improve performance
- Penalizing user's mistakes
  - users with large number of erroneous votes would be penalized in weight
  - Currently this does not improve performance
  - Taking the Accuracy of each user in the weight function shows however close performance from the current strategy

Conclusion and perspectives
Key points

- We released a new dataset for large-scale crowdsourced data: `https://zenodo.org/records/10782465`
- We evaluated the current label aggregation in Pl@ntNet against other scaling ones and show better performance overall

- We released a new dataset for large-scale crowdsourced data: `https://zenodo.org/records/10782465`
- We evaluated the current label aggregation in Pl@ntNet against other scaling ones and show better performance overall

- There is a need to *better collect votes* and not *waste* expert knowledge $\rightarrow$ recommendation system
- We worked on the aggregation of species determination, users can enter other information – the observation is a flower/organ/*etc.*

Thank you

Peerannot library

`https://peerannot.github.io`

References I

- Chu, Z., J. Ma, and H. Wang (2021). "Learning from Crowds by Modeling Common Confusions.". In: *AAAI*, pp. 5832–5840.
- Dawid, A. and A. Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *J. R. Stat. Soc. Ser. C. Appl. Stat.* 28.1, pp. 20–28.
- Hovy, D. et al. (2013). "Learning whom to trust with MACE". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1120–1130.
- Ju, C., A. Bibaut, and M. van der Laan (2018). "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *J. Appl. Stat.* 45.15, pp. 2800–2818.
- Lapin, M., M. Hein, and B. Schiele (2016). "Loss functions for top-k error: Analysis and insights". In: *CVPR*, pp. 1468–1477.
- Lefort, T. et al. (2022). "Identify ambiguous tasks combining crowdsourced labels by weighting Areas Under the Margin". In: *arXiv preprint arXiv:2209.15380*.

References II

- Lefort, T. et al. (May 17, 2024a). "Peerannot: Classification for Crowdsourced Image Datasets with Python". In: *Computo*.
- — (July 2024b). "Weighted majority vote using Shapley values in crowdsourcing". In: *CAp 2024 - Conférence sur l'Apprentissage Automatique*. Lille, France.
- Peterson, J. C. et al. (2019). "Human Uncertainty Makes Classification More Robust". In: *ICCV*, pp. 9617–9626.
- Pleiss, G. et al. (2020). "Identifying mislabeled data using the area under the margin ranking". In: *NeurIPS*.
- Rodrigues, F. and F. Pereira (2018). "Deep learning from crowds". In: *AAAI*. Vol. 32.
- Rodrigues, F., F. Pereira, and B. Ribeiro (2014). "Gaussian process classification and active learning with multiple annotators". In: *ICML*. PMLR, pp. 433–441.
- Servajean, M. et al. (2017). "Crowdsourcing thousands of specialized labels: A Bayesian active training approach". In: *IEEE Transactions on Multimedia* 19.6, pp. 1376–1391.

References III

 Whitehill, J. et al. (2009). “Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise”. In: *NeurIPS*. Vol. 22.