# Ensembles in machine learning

*(simple) theory and (simple) practice*

Pierre-Alexandre Mattei

*joint work with*

***Damien Garreau, Raphaël Razafindralambo, Rémy Sun, Frédéric Precioso***

3iA Côte d'Azur
Institut interdisciplinaire d'intelligence artificielle

# Menu of the day

1. Historical intro
2. Some basic examples of ensembles used in practice
    - MC dropout
    - Deep ensembles
3. When and why ensembling works? Some empirics
4. When and why ensembling works? The case of convex losses
5. Nonconvex subtleties

- Mattei and Garreau, Are Ensembles Getting Better All the Time, JMLR 2025
- Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026
- Razafindralambo et al., Beyond Mixtures and Products for Ensemble Aggregation: A Likelihood Perspective on Generalized Means, UAI 2026

# 1

# Ensembles in stats and machine learning, a brief history

# What are ensembles?

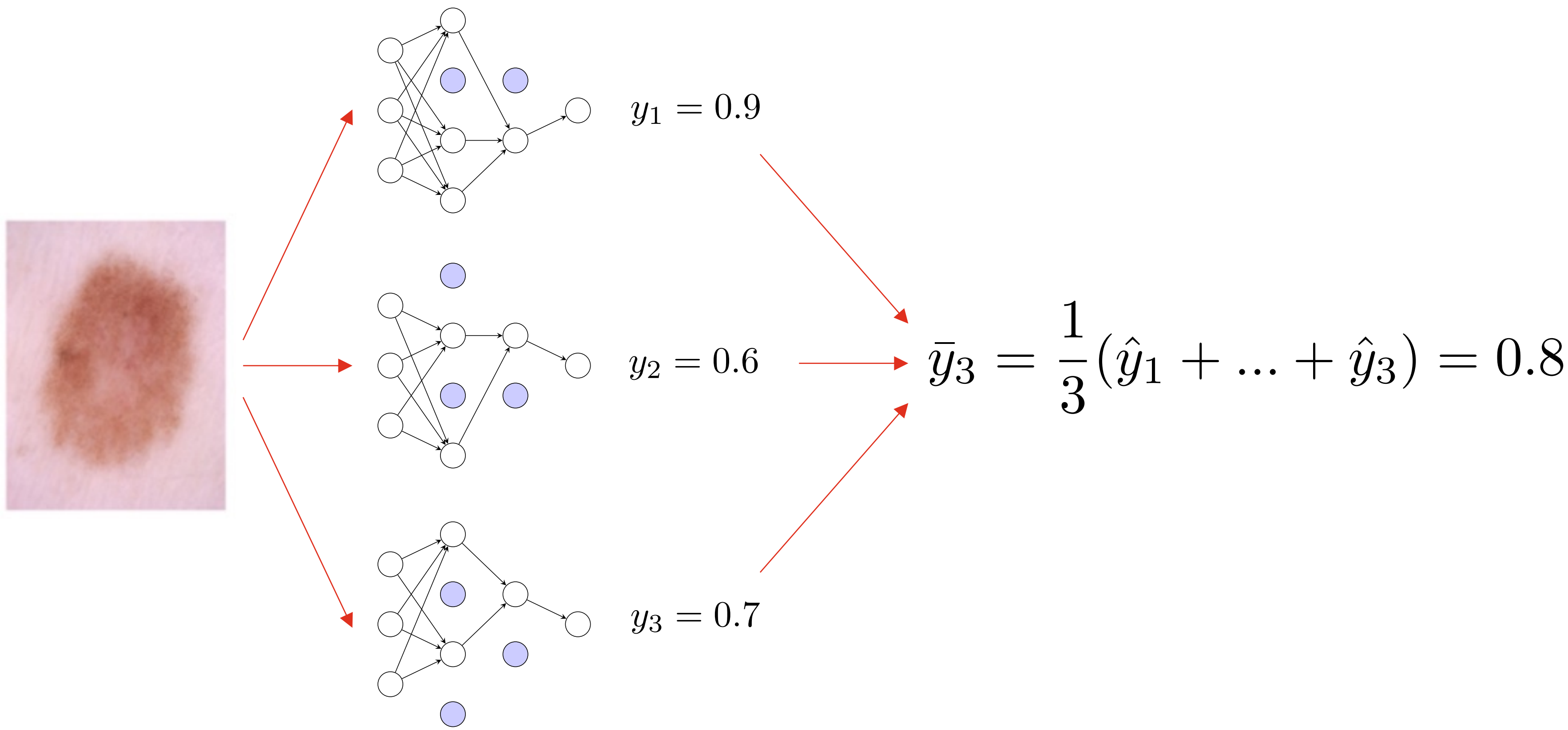
- **Ensembles combine the predictions of several base models.**

# What are ensembles?

- **Ensembles combine the predictions of several base models.**
- They can be used for any statistical task, for instance **classification** (here a dropout ensemble to classify skin lesions), but also **regression** or **density estimation**.

# What are ensembles?

- **Ensembles combine the predictions of several base models.**
- They can be used for any statistical task, for instance **classification** (here a dropout ensemble to classify skin lesions), but also **regression** or **density estimation**.



# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**
- They became very **trendy in the 90s**, in particular via the work of Leo Breiman, who came up with **bagging** and **random forests**

# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**

- They became very **trendy in the 90s**, in particular via the work of Leo Breiman, who came up with **bagging** and **random forests**

- Breiman, *Bagging predictors*, Machine Learning, 1996
  - Base models correspond to the **same algorithm trained on different bootrapped subsamples of the data**

# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**
- They became very **trendy in the 90s**, in particular via the work of Leo Breiman, who came up with **bagging** and **random forests**

- Breiman, *Bagging predictors*, Machine Learning, 1996
  - Base models correspond to the **same algorithm trained on different bootrapped subsamples of the data**
- Breiman, *Random forests*, Machine Learning, 2000
  - A variant of bagging where base models are **randomised decision trees**

# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**
- They became very **trendy in the 90s**, in particular for **neural networks**. Base models correspond to the **networks with the same architecture trained with different initialisation**

# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**
- They became very **trendy in the 90s**, in particular for **neural networks**. Base models correspond to the **networks with the same architecture trained with different initialisation**

❖ Hansen and Salamon, *Neural network ensembles*, IEEE PAMI, 1990

# What are ensembles? The 90s

- **Ensembles combine the predictions of several base models.**
- They became very **trendy in the 90s**, in particular for **neural networks**. Base models correspond to the **networks with the same architecture trained with different initialisation**

❖ Hansen and Salamon, *Neural network ensembles*, IEEE PAMI, 1990

❖ Krogh and Vedeslby, *Neural Network Ensembles, Cross Validation, and Active Learning*, NeurIPS, 1995

# What are ensembles? Trends of the 2020s

- For **tabular data**, ensembles of decision trees (random forest, boosting) still perform better than deep learning.
    - Grinsztajn, Oyallon, and Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, NeurIPS 2022

# What are ensembles? Trends of the 2020s

- For **tabular data**, ensembles of decision trees (random forest, boosting) still perform better than deep learning.
    - Grinsztajn, Oyallon, and Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, NeurIPS 2022

- Ensembling also works well together with deep learning.

# What are ensembles? Trends of the 2020s

- For **tabular data**, ensembles of decision trees (random forest, boosting) still perform better than deep learning.
  - Grinsztajn, Oyallon, and Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, NeurIPS 2022

- Ensembling also works well together with deep learning.
  - In **Monte Carlo dropout (MC dropout)**, base models are **versions of a single network with several dropout masks**
    - Gal and Ghahramani, *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, ICML 2016

# What are ensembles? Trends of the 2020s

- For **tabular data**, ensembles of decision trees (random forest, boosting) still perform better than deep learning.
    - Grinsztajn, Oyallon, and Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, NeurIPS 2022

- Ensembling also works well together with deep learning.
    - In **Monte Carlo dropout (MC dropout)**, base models are **versions of a single network with several dropout masks**
    - **Deep ensembles** are similar to the neural net ensembles of the 90s: base models are different trainings of the same deep architecture
        - Lakshminarayanan, Pritzel, Blundell, *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*, NeurIPS 2017

# What are ensembles? Trends of the 2020s

- For **tabular data**, ensembles of decision trees (random forest, boosting) still perform better than deep learning.
  - Grinsztajn, Oyallon, and Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, NeurIPS 2022

- Ensembling also works well together with deep learning.
  - In **Monte Carlo dropout (MC dropout)**, base models are **versions of a single network with several dropout masks**
  - **Deep ensembles** are similar to the neural net ensembles of the 90s: base models are different trainings of the same deep architecture

Both MC dropout and deep ensembles work extremely well especially with respect to "uncertainty aware" metrics like the cross-entropy, but are a bit less impressive in terms of accuracy.

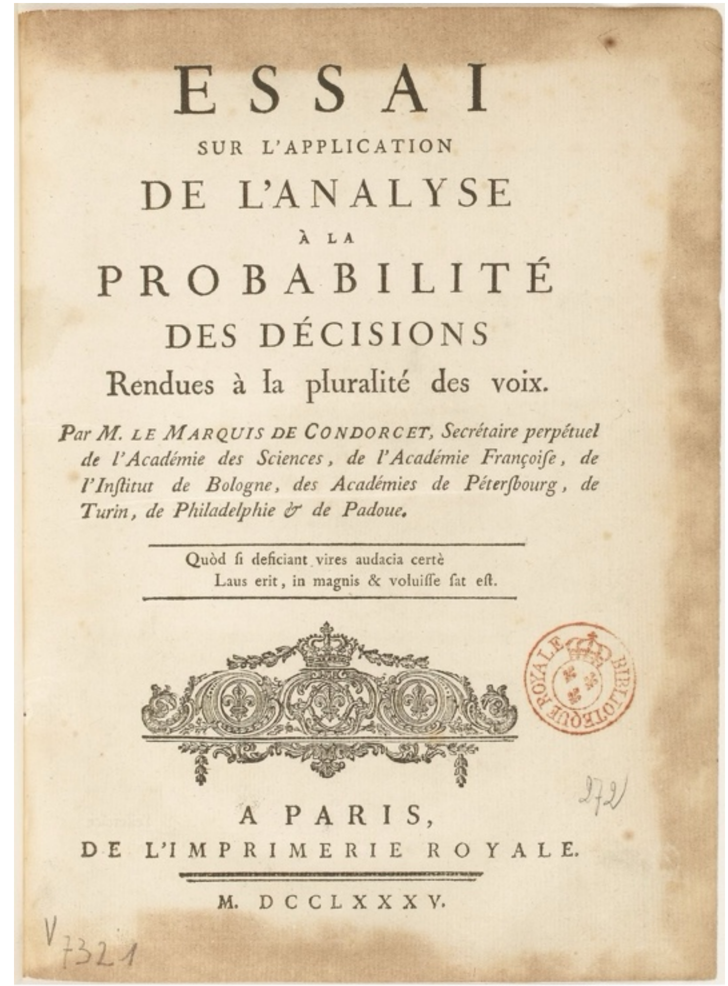
# What are ensembles? Beyond stats and ML

- The key idea behing ensembles is that **groups are collectively better at decision-making than individuals**. This is an old idea, that goes way beyond ML.

# What are ensembles? Beyond stats and ML

- The key idea behing ensembles is that **groups are collectively better at decision-making than individals**. This is an old idea, that goes way beyond ML.

- At the end of the 1700s, **Condorcet** proposed a mathematical formalisation of this argument that is somewhat close to what we'll see today. His main applications were politics and jurys.

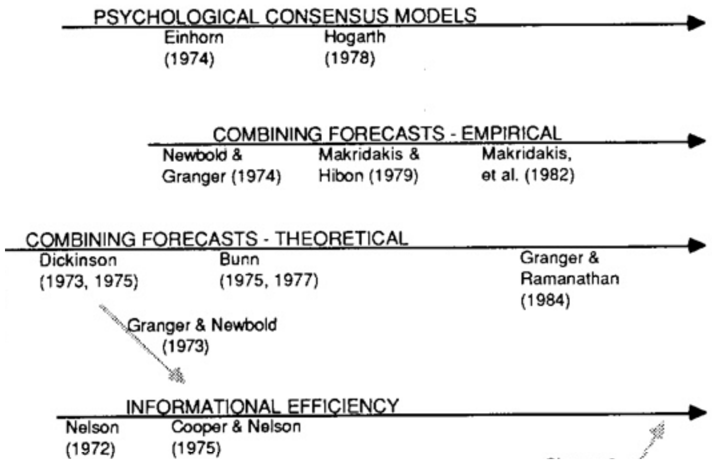



Source gallica.bnf.fr / Bibliothèque nationale de France

# What are ensembles? Beyond stats and ML

- The key idea behing ensembles is that **groups are collectively better at decision-making than individuals**. This is an old idea, that goes way beyond ML.

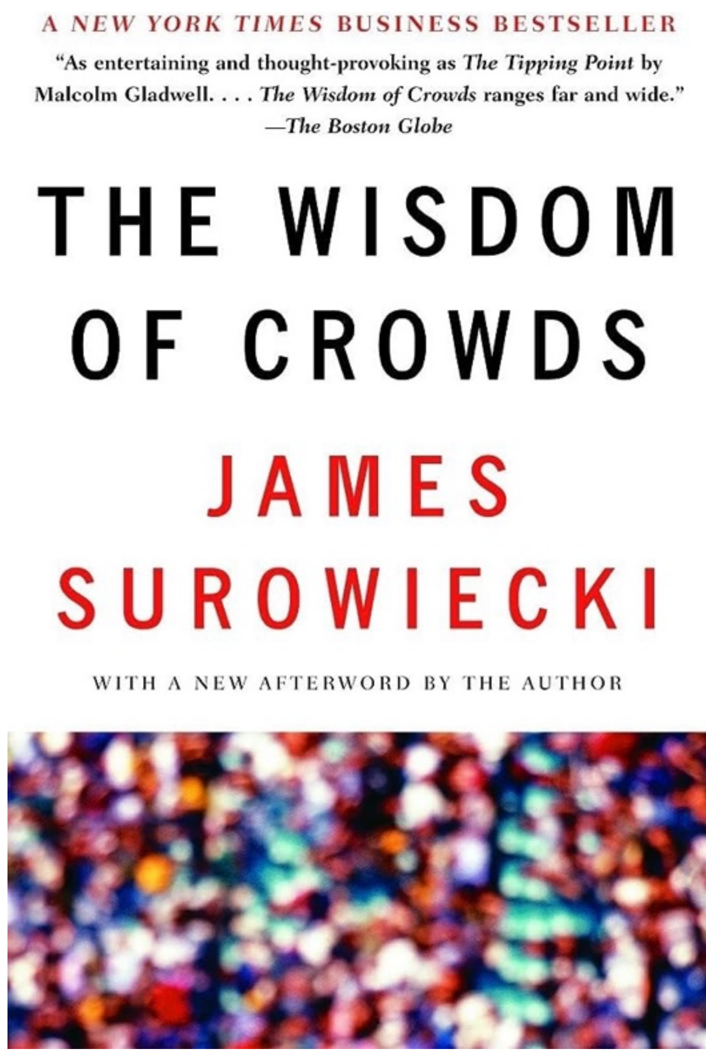
- **Economists, econometricians and forecasters** have used ensembles from the 60s.
  - Clemen, *Combining forecasts: A review and annotated bibliography*, International Journal of Forecasting, 1989



# What are ensembles? Beyond stats and ML

- The key idea behing ensembles is that **groups are collectively better at decision-making than individals**. This is an old idea, that goes way beyond ML.

- The phrase « wisdom of crowds », popularised by Surowiecki's bestselling book, is often used to summarise this idea



# What ensembles will we be looking at today

- Today, we'll study ensembles that are slight variations of the same model, e.g. **random forests**, **deep ensembles**, **MC dropout**, **bagging**, test-time data augmentations…

# What ensembles will we be looking at today

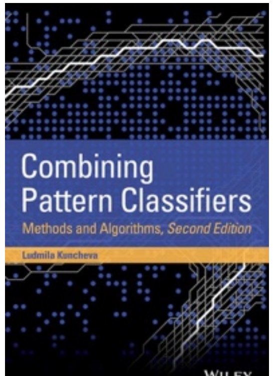
- Today, we'll study ensembles that are slight independent variations of the same model, e.g. **random forests**, **deep ensembles**, **MC dropout**, **bagging**, test-time data augmentations…

- While this is very general, other ensembles that do not satisfy this property exist, but we will not be looking at them today: **boosting**, **Bayesian model averaging**, ensembles of different physical models for weather forecasting…
  - Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms (2nd edition)*, Wiley, 2014

# What ensembles will we be looking at today

- Today, we'll study ensembles that are slight independent variations of the same model, e.g. **random forests**, **deep ensembles**, **MC dropout**, **bagging**, test-time data augmentations…

- While this is very general, other ensembles that do not satisfy this property exist, but we will not be looking at them today: **boosting**, **Bayesian model averaging**, ensembles of different physical models for weather forecasting…
  - Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms (2nd edition)*, Wiley, 2014



# 2

## Basic but popular ensembles of neural nets

# Deep ensembles

- **Train $n$ neural nets on the same dataset, with the same training algorithm but different random seeds.**

- Weirdly enough, it generally outperforms bagging of deep nets.

- Again, we can average the networks as we see fit, for instance just average the outputs.

# Deep ensembles

- **Train $n$ neural nets on the same dataset, with the same training algorithm but different random seeds.**

- Weirdly enough, it generally outperforms bagging of deep nets.

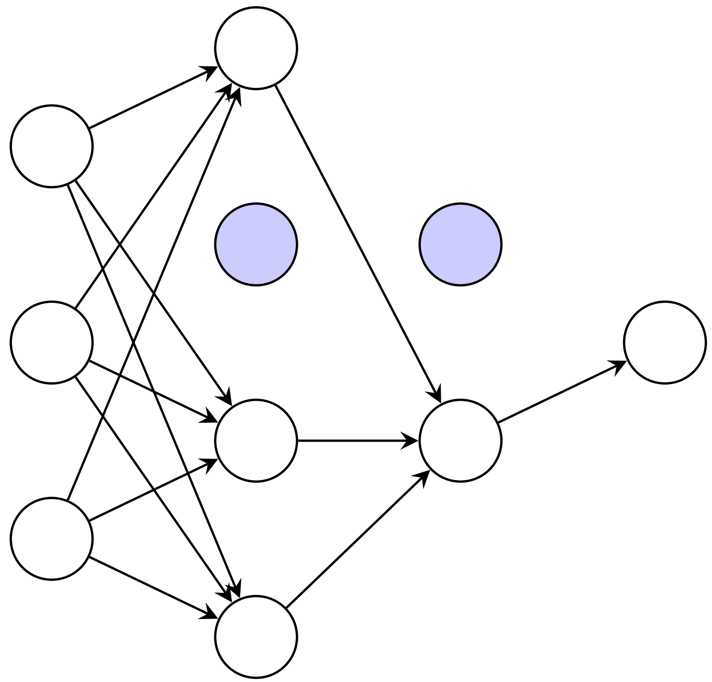
- Again, we can average the networks as we see fit, for instance just average the outputs.

- This can be used for **any deep learning task**, not only classification, but segmentation, generative modelling, regression…

- Related to Bayesian deep learning:
  - Wild et al., **A Rigorous Link between Deep Ensembles and (Variational) Bayesian Methods**, NeurIPS 2023

# MC dropout

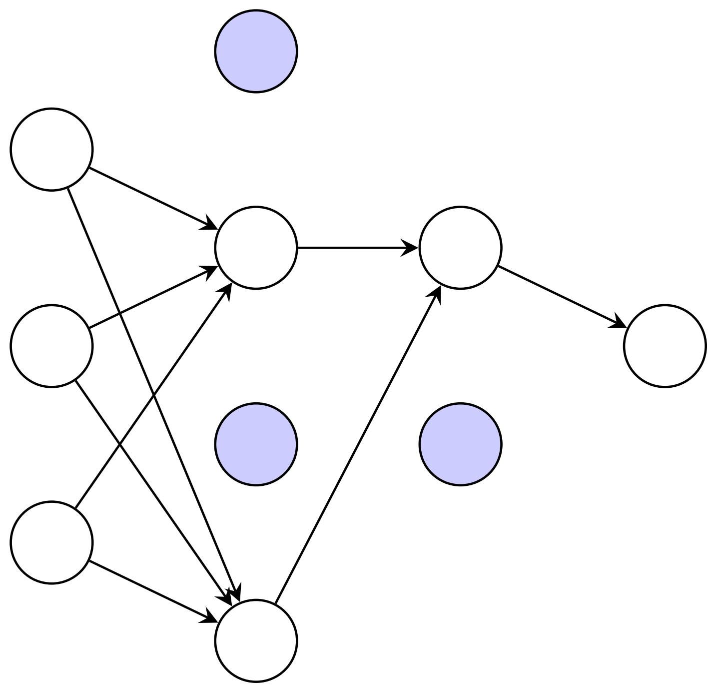
- Train a neural net once, but with dropout activated.
- **Then, average the networks obtained using different (random) dropout masks.**



# MC dropout

- Train a neural net once, but with dropout activated.

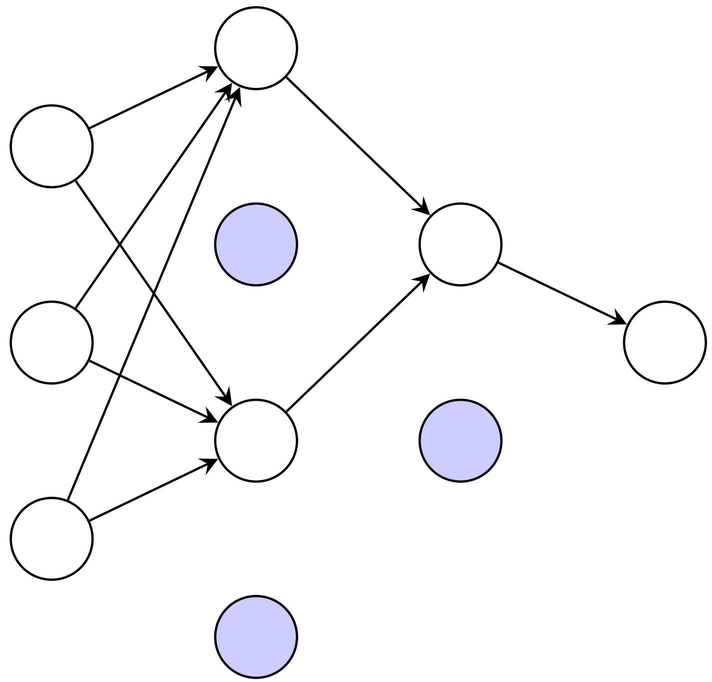
- **Then, average the networks obtained using different (random) dropout masks.**



# MC dropout

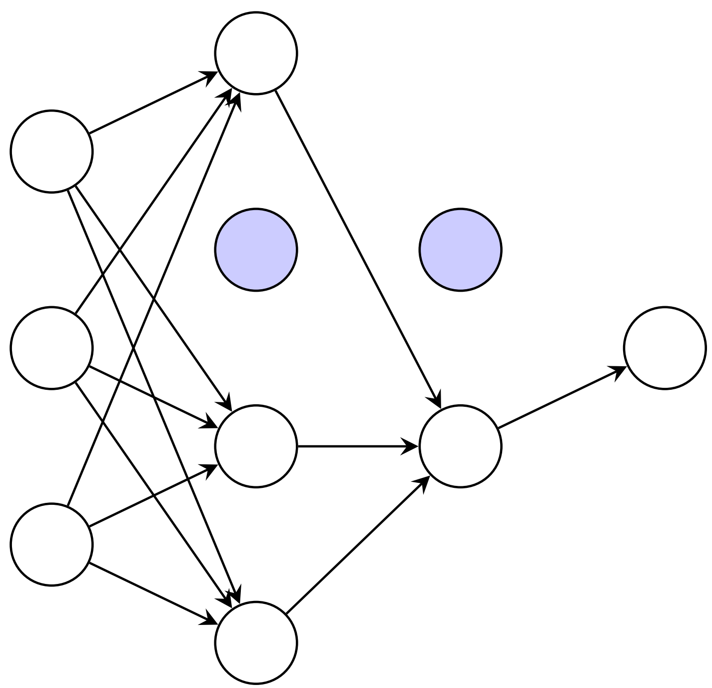
- Train a neural net once, but with dropout activated.

- **Then, average the networks obtained using different (random) dropout masks.**



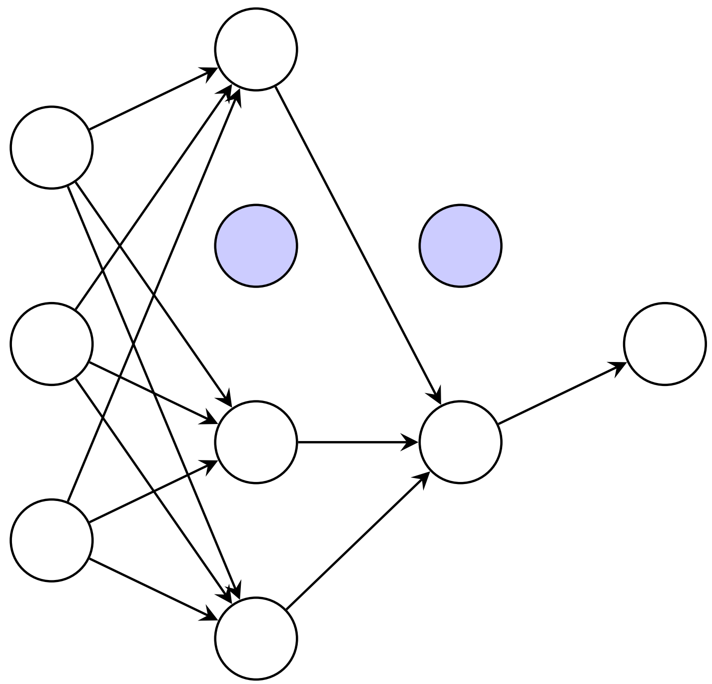
# MC dropout

- Train a neural net once, but with dropout activated.
- **Then, average the networks obtained using different (random) dropout masks.**



# MC dropout

- Train a neural net once, but with dropout activated.
- **Then, average the networks obtained using different (random) dropout masks.**



Loose relationship with Bayesian inference:

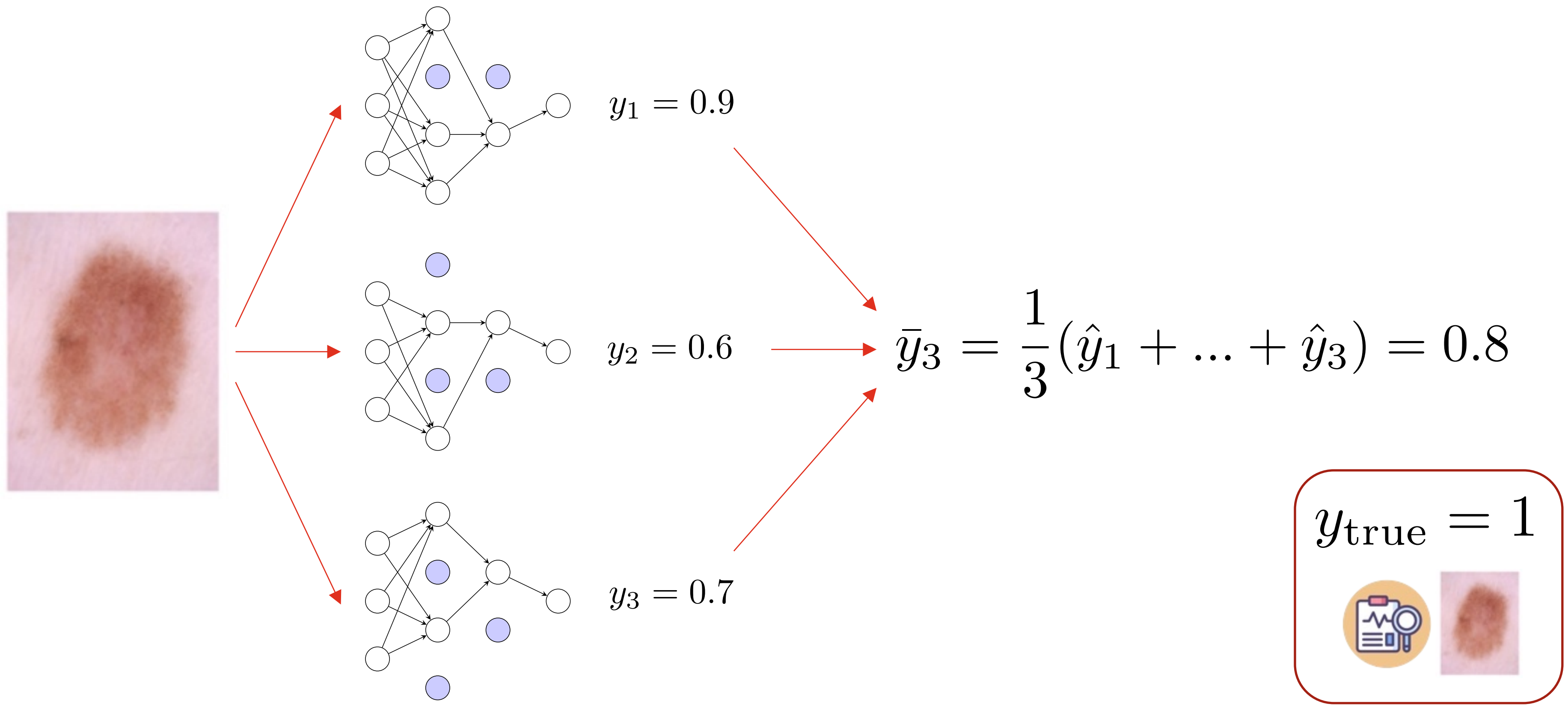
- Hron et al., Variational Bayesian dropout: pitfalls and fixes, ICML 2023

# 3

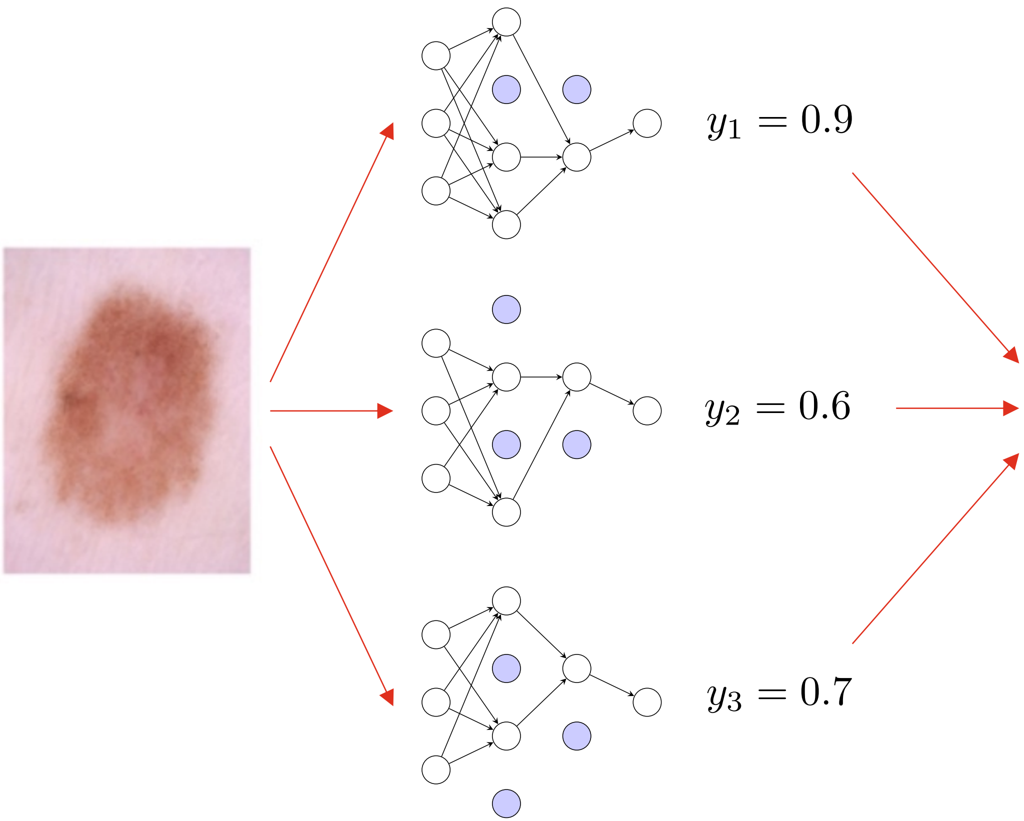
## When and why ensembles work?

*some empirics*

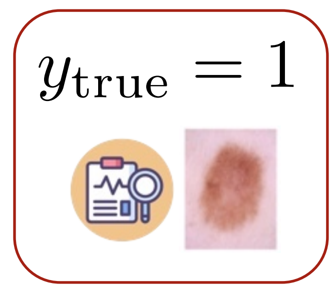
# MC dropout for medical image classification



# MC dropout for medical image classification

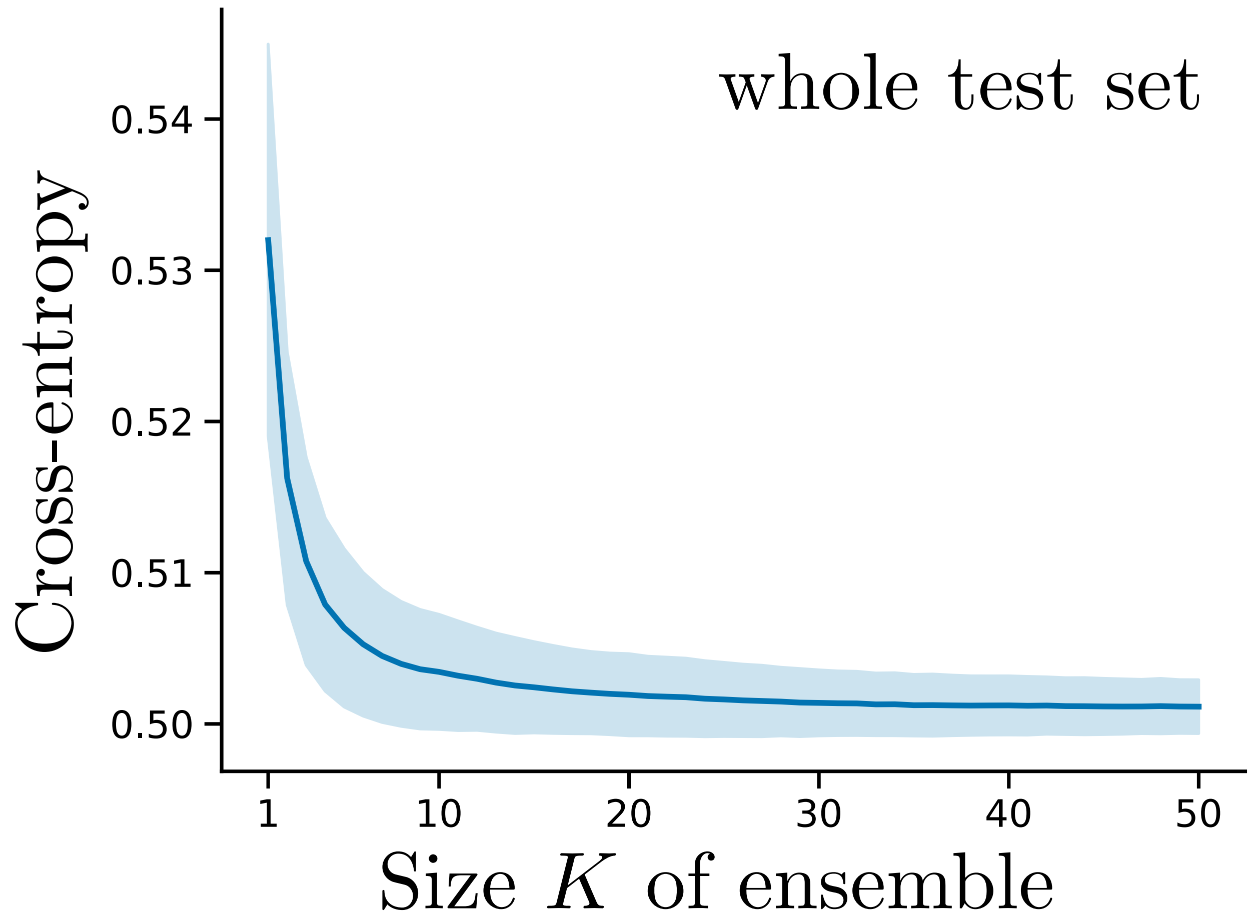



$$\bar{y}_3 = \frac{1}{3}(\hat{y}_1 + \ldots + \hat{y}_3) = 0.8$$

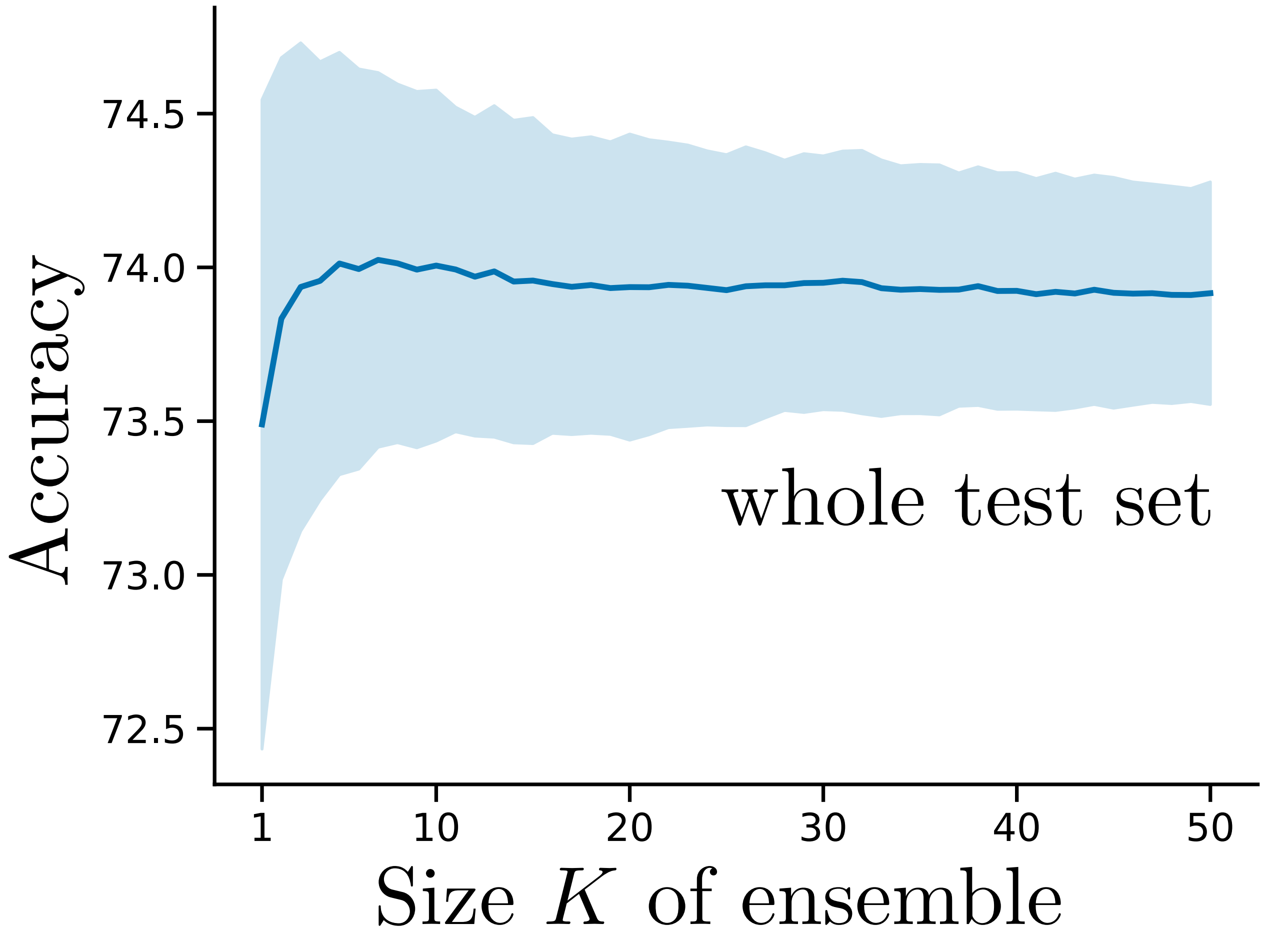



"Infinite ensemble" $\bar{y}_\infty = \mathbb{E}[y_k] = \lim \bar{y}_K$

# Ensembles seem to be getting better and better 😀



# Ensembles seem to be getting better and better? Really? 🤔



# First observations

- Ensembles seem to be **monotonically improving for the cross-entropy**, but **things seem less clear for the accuracy**.

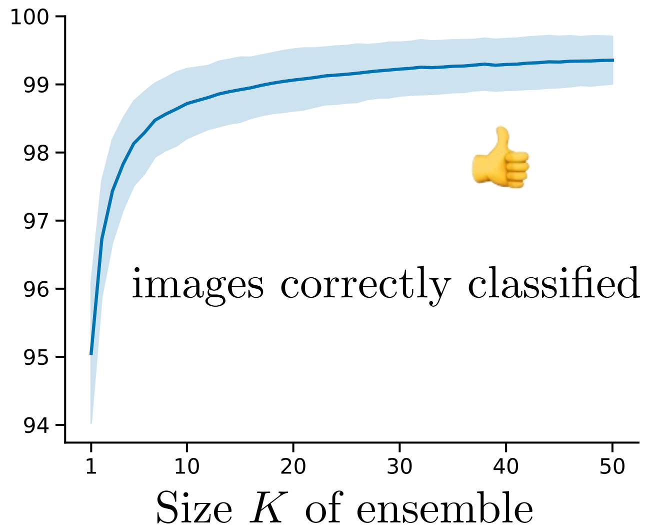
# First observations

- Ensembles seem to be **monotonically improving for the cross-entropy**, but **things seem less clear for the accuracy**.
- If we investigate, we notice that we can split the test dataset into two parts:

# First observations

- Ensembles seem to be **monotonically improving for the cross-entropy**, but **things seem less clear for the accuracy**.

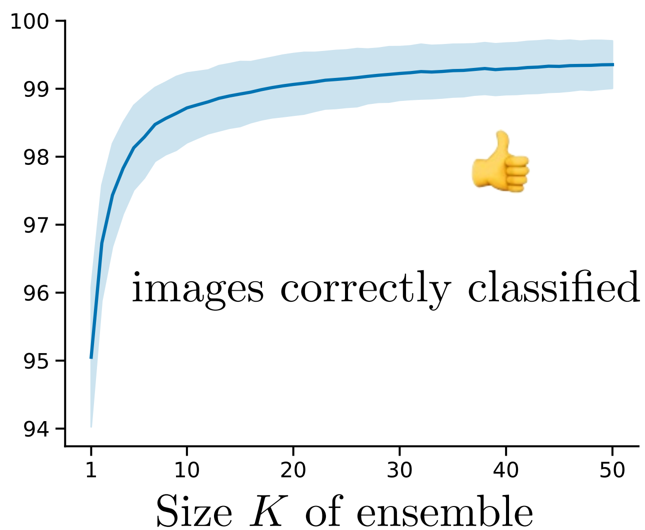
- If we investigate, we notice that we can split the test dataset into two parts:
  - One part where the **prediction of $\overline{y}_\infty$ is right, for which the accuracy increases**

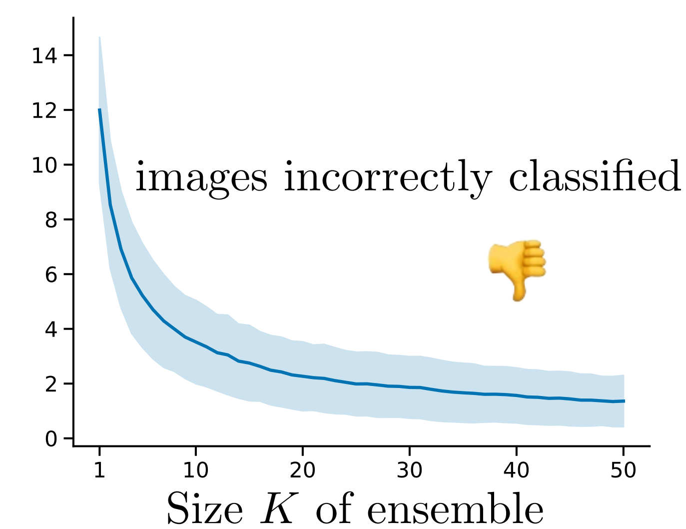


# First observations

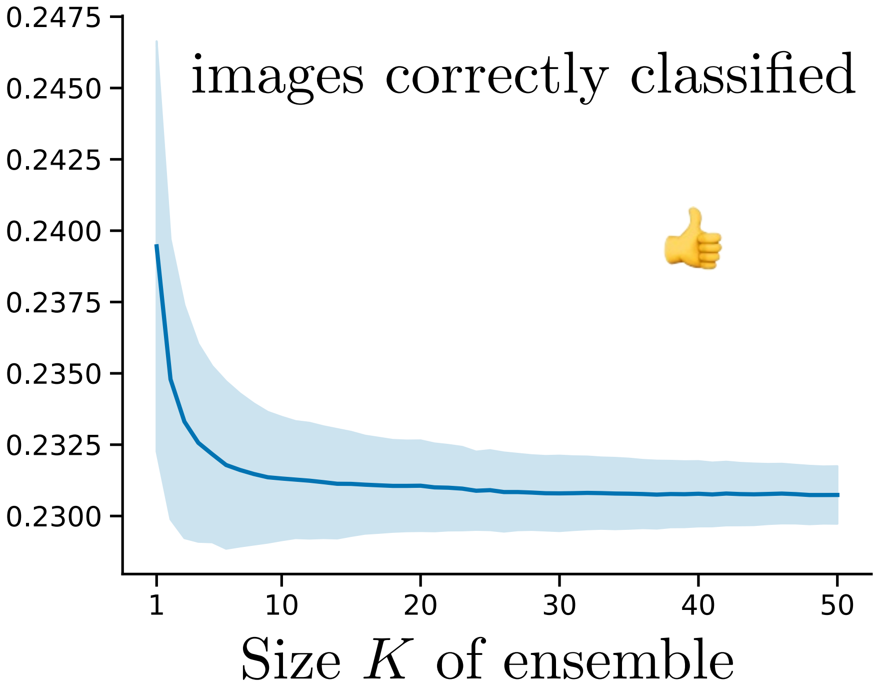
- Ensembles seem to be **monotonically improving for the cross-entropy**, but **things seem less clear for the accuracy**.

- If we investigate, we notice that we can split the test dataset into two parts:
  - ➢ One part where the **prediction of $\overline{y}_\infty$ is right, for which the accuracy increases**
  - ➢ One part where the **prediction of $\overline{y}_\infty$ is wrong, for which the accuracy decreases**



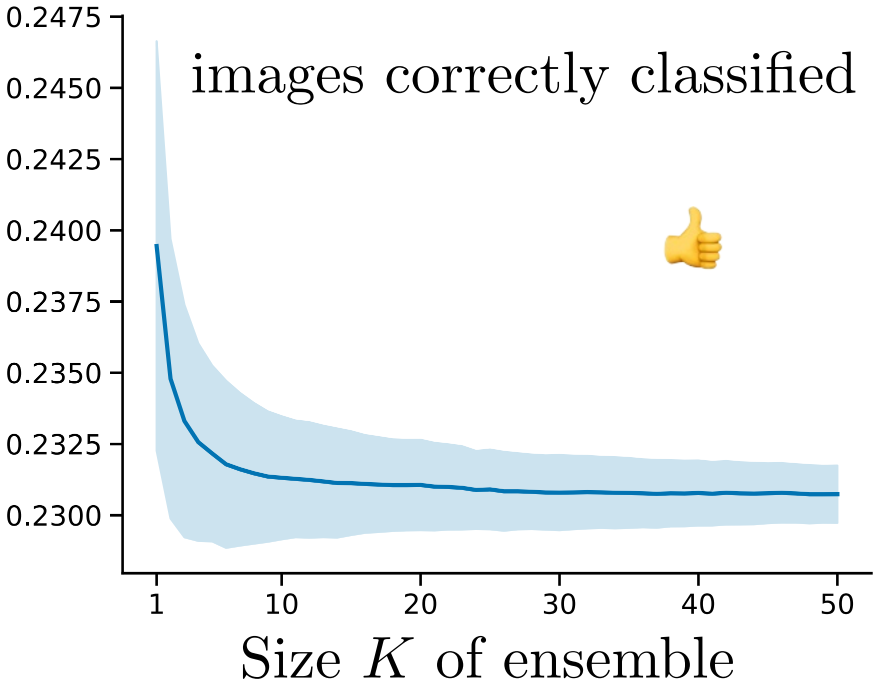



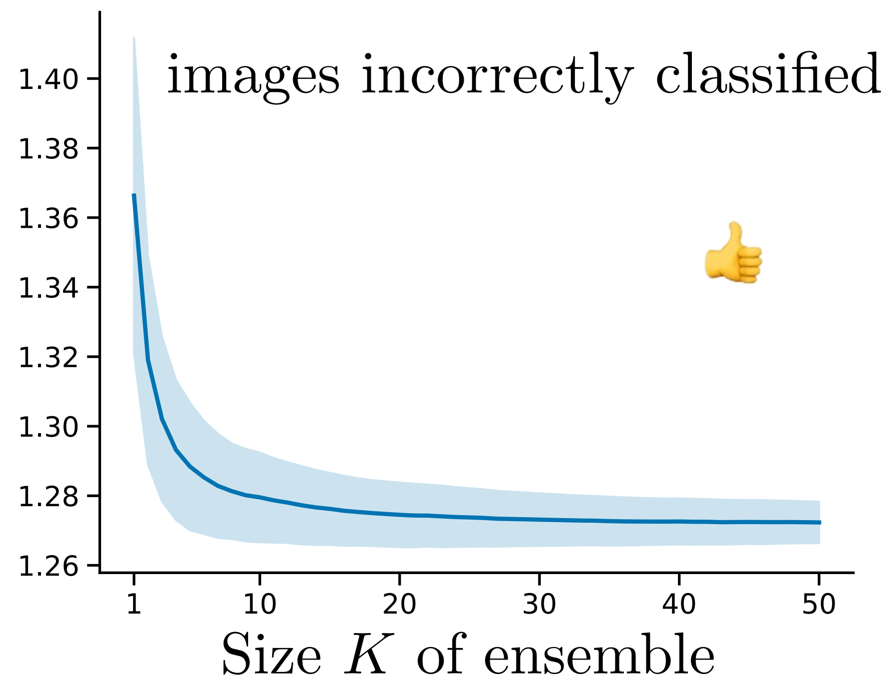
# What about the cross entropy?



# What about the cross entropy?

- Dividing the dataset into the same parts is harmless for the monotonicity of the cross-entropy. It seems that **the cross-entropy is always getting better and better!**



# The behaviour we just saw is typical

- These observations are incredibly universal, and were previously highlighted for random forests by Probst and Boulesteix (2018) who looked at **308 data sets!**
    - Probst and Boulesteix, *To Tune or Not to Tune the Number of Trees in Random Forests*, JMLR 2018

# The behaviour we just saw is typical

- These observations are incredibly universal, and were previously highlighted for random forests by Probst and Boulesteix (2018) who looked at 308 data sets!

    - Probst and Boulesteix, *To Tune or Not to Tune the Number of Trees in Random Forests*, JMLR 2018

- Their empirical conclusions are:

    - **The cross-entropy and the Brier score seem to be getting better all the time as the ensemble grows**

# The behaviour we just saw is typical

- These observations are incredibly universal, and were previously highlighted for random forests by Probst and Boulesteix (2018) who looked at 308 data sets!

    - Probst and Boulesteix, *To Tune or Not to Tune the Number of Trees in Random Forests*, JMLR 2018

- Their empirical conclusions are:

    - **The cross-entropy and the Brier score seem to be getting better all the time as the ensemble grows**
    - **The accuracy and the AUC have a more subtle behaviour that can be non-monotonic**

# The behaviour we just saw is typical

- These observations are incredibly universal, and were previously highlighted for random forests by Probst and Boulesteix (2018) who looked at 308 data sets!

    - Probst and Boulesteix, *To Tune or Not to Tune the Number of Trees in Random Forests*, JMLR 2018

- Their empirical conclusions are:

    - **The cross-entropy and the Brier score seem to be getting better all the time as the ensemble grows**
    - **The accuracy and the AUC have a more subtle behaviour that can be non-monotonic**

Our goal is to provide a theoretical explaination for these behaviours.

# The behaviour we just saw is typical **and not just for classification!**

- We observe similar patterns for **regression**
  - **MSE, MAE, Huber losses are getting better all the time**
  - **Some other weird losses (e.g. for robust regression) may be nonmotonic**

# The behaviour we just saw is typical **and not just for classification!**

- Things are also similar for **generative modelling with diffusion models**

# The behaviour we just saw is typical **and not just for classification!**

- Things are also similar for **generative modelling with diffusion models**

$$\mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$$

$$\left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \boxed{\nabla_{\mathbf{x}} \log q_t(\mathbf{x})}\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$$

- $\mathbf{f}(\mathbf{x}, t) \in \mathbb{R}^d$: drift coefficient.
- $g(t) \in \mathbb{R}$: diffusion coefficient.
- $t \in [0, T]$.
- $\mathrm{d}\mathbf{x}, \mathrm{d}t$: small change in $\mathbf{x}, t$.
- $\mathbf{w}$: Brownian motion.



❖ Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

# The behaviour we just saw is typical **and not just for classification!**

- Things are also similar for **generative modelling with diffusion models**

- The loss function is essentially a **MSE to learn an approximation of the score** $\nabla_x \log\, q_t(x)$

$$L_{\mathrm{DDSM}}(\boldsymbol{s}_{\boldsymbol{\theta}}) := \mathbb{E}_{t\sim[0,T]}\left(\lambda_t \mathbb{E}_{q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)q_0(\mathbf{x}_0)} \|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\|^2\right)$$


❖ Vincent, A connection between score matching and denoising autoencoders, Neural Computation, 2011
❖ Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

# The behaviour we just saw is typical **and not just for classification!**

- Things are also similar for **generative modelling with diffusion models**

- The loss function is essentially a **MSE to learn an approximation of the score** $\nabla_x \log q_t(x)$

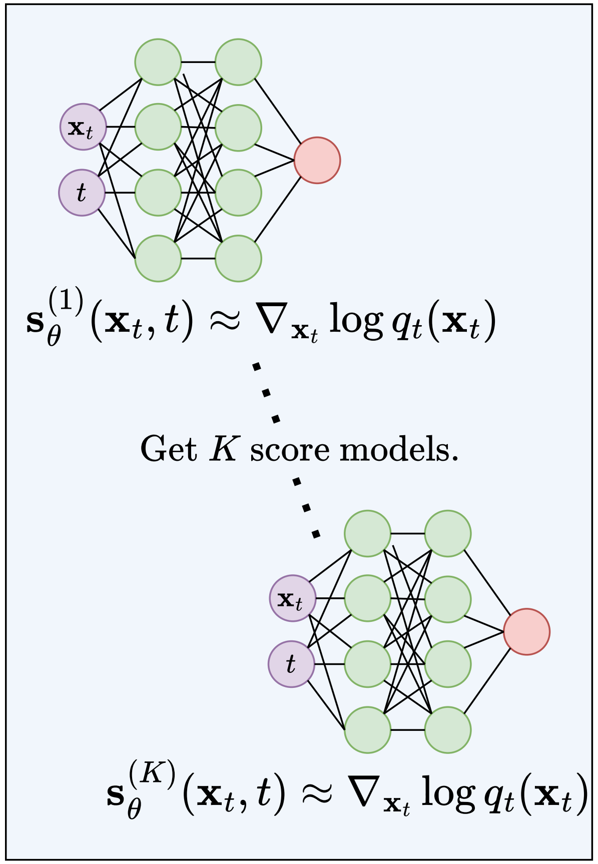
$$L_{\text{DDSM}}(\boldsymbol{s}_{\boldsymbol{\theta}}) := \mathbb{E}_{t\sim[0,T]}\left(\lambda_t \mathbb{E}_{q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)q_0(\mathbf{x}_0)} \|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\|^2\right)$$

*…so in a sense, this is quite similar to regression!*


❖ Vincent, A connection between score matching and denoising autoencoders, Neural Computation, 2011
❖ Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

# The behaviour we just saw is typical **and not just for classification!**

- How can we build ensembles of diffusion models?

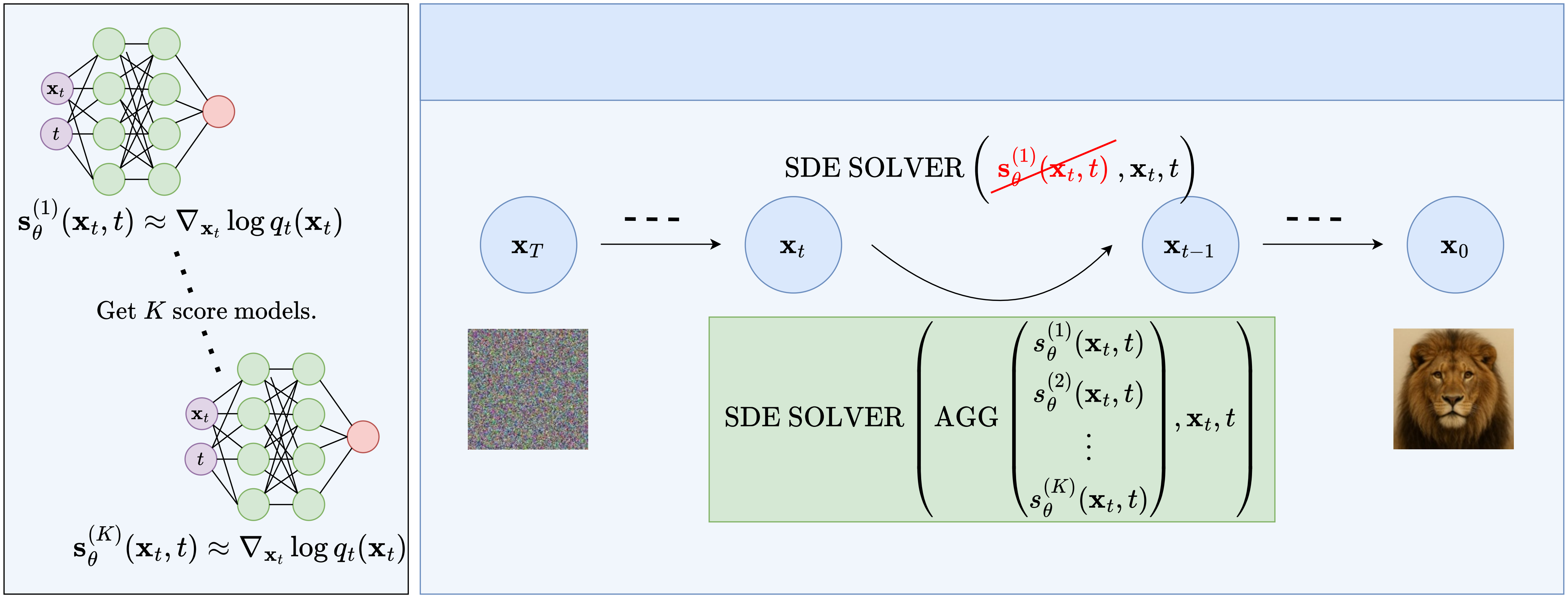





❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

# The behaviour we just saw is typical **and not just for classification!**

- How can we build ensembles of diffusion models?

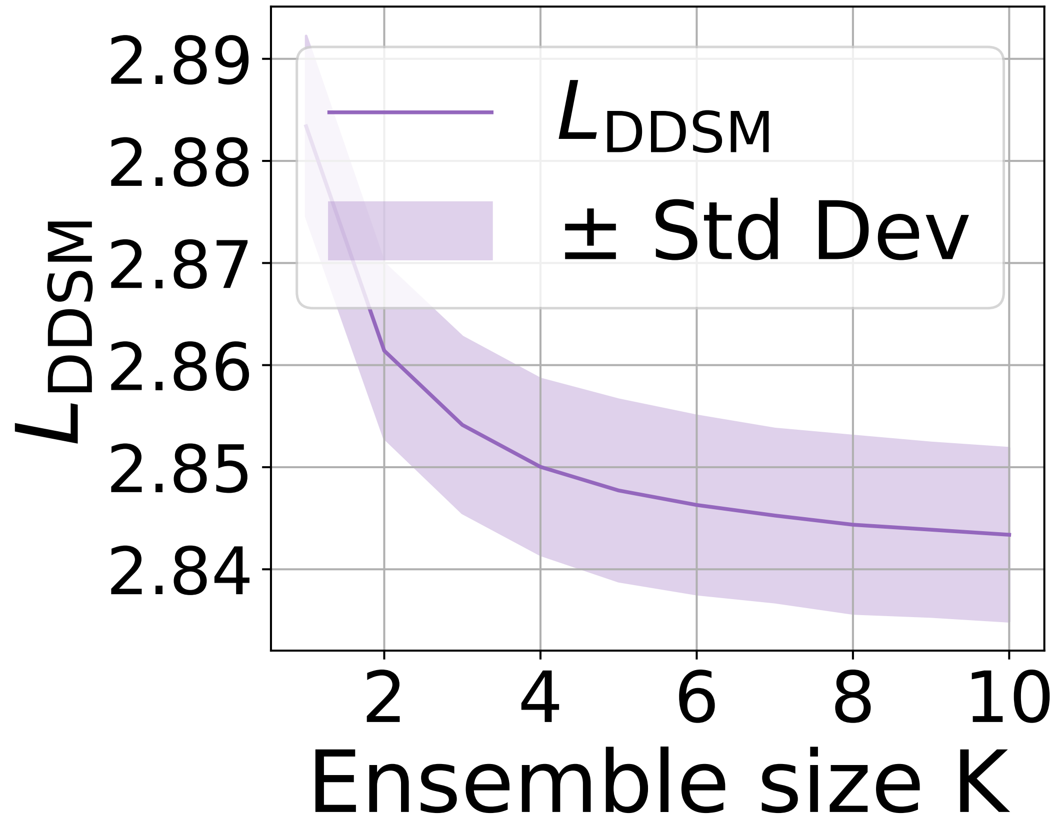





❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

## The behaviour we just saw is typical **and not just for classification!**

- **The standard MSE loss is getting better all the time** (e.g. here on CIFAR 10)

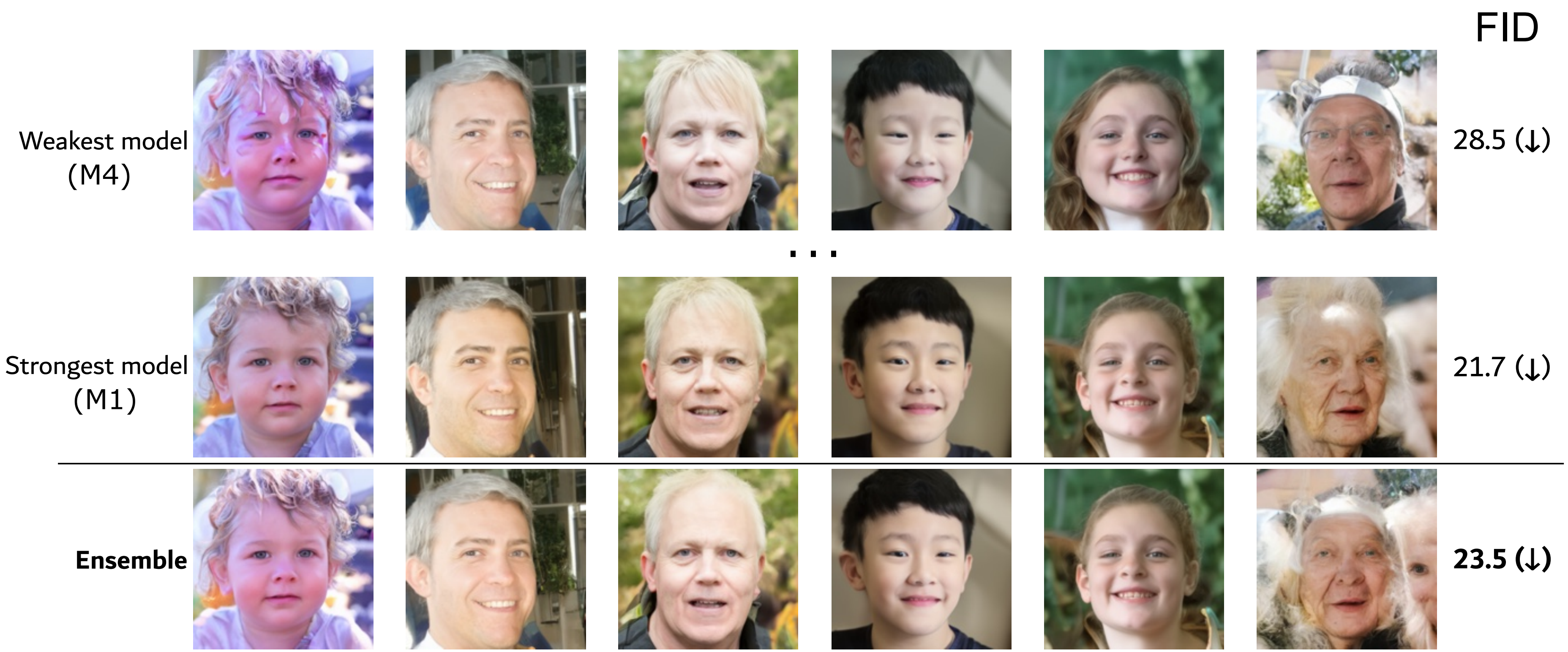





❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

# The behaviour we just saw is typical **and not just for classification!**

- **But ensembling barely affects image quality** 😭 (e.g. here on FFHQ)

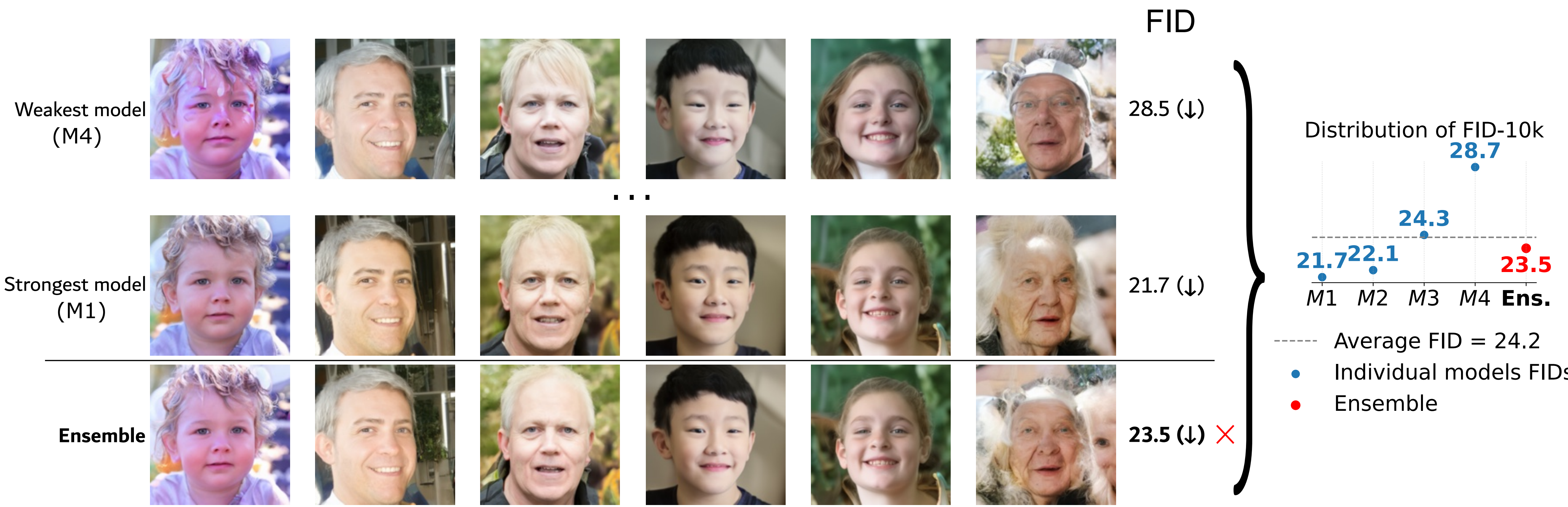





❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

# The behaviour we just saw is typical **and not just for classification!**

- **But ensembling barely affects image quality** 😭 (e.g. here on FFHQ)

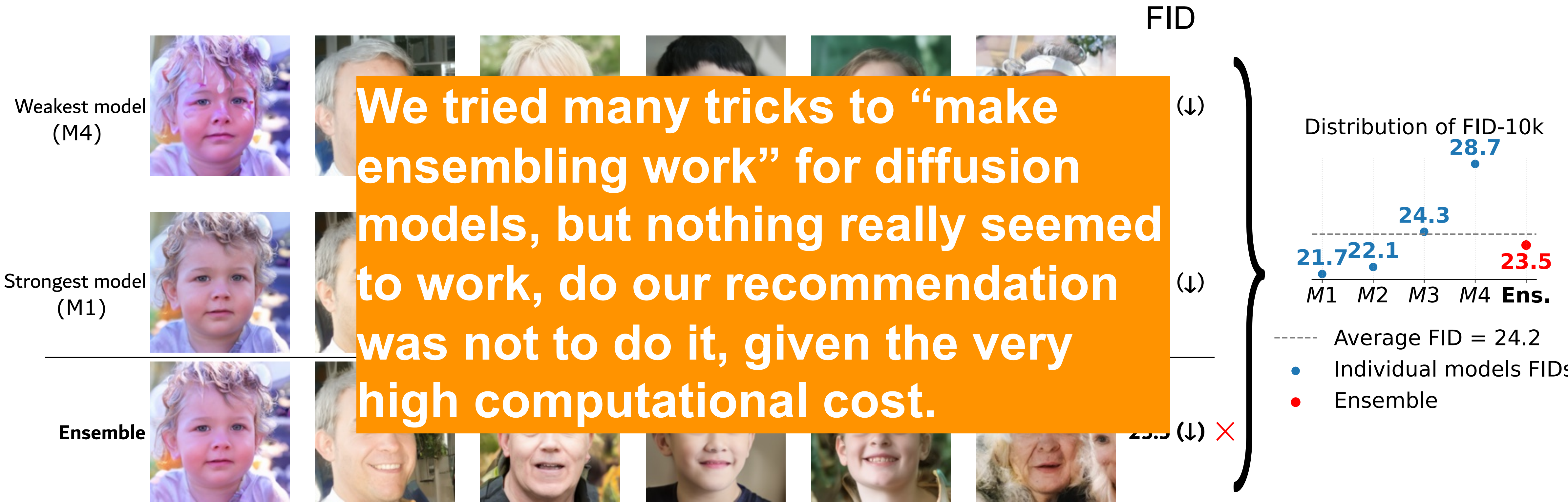





❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

# The behaviour we just saw is typical **and not just for classification!**

- **But ensembling barely affects image quality** 😭 (e.g. here on FFHQ)




**We tried many tricks to "make ensembling work" for diffusion models, but nothing really seemed to work, do our recommendation was not to do it, given the very high computational cost.**



❖ Razafindralambo et al., When Are Two Scores Better Than One? Investigating Ensembles of Diffusion Models, TMLR 2026

# What differentiates these losses?

- **Q: In what way are the cross-entropy/Brier score different from the classification error/AUC?**

# What differentiates these losses?

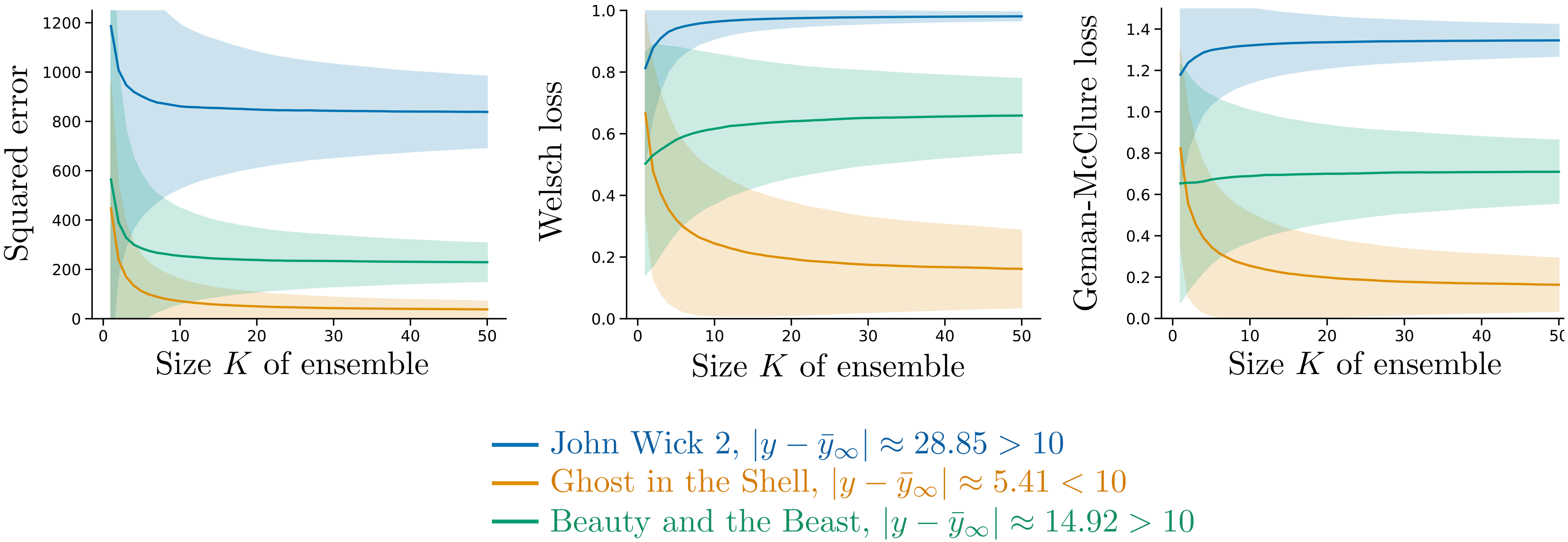
- **Q: In what way are the cross-entropy/Brier score different from the classification error/AUC?**
  - **Cross-entropy, Brier score and MSE are convex**, which is not the case of classification error/AUC/FID.

# What differentiates these losses?

- **Q: In what way are the cross-entropy/Brier score different from the classification error/AUC?**
  - **Cross-entropy, Brier score and MSE are convex**, which is not the case of classification error/AUC/FID.
- Our main contribution is to show that this convexity divide is responsible for the results we just saw. We will essentially show that
  - **For convex losses, ensembles are getting better all the time**
  - **For convex losses,  ensembles are improving over “good” data points and getting worse over “bad” data points.**

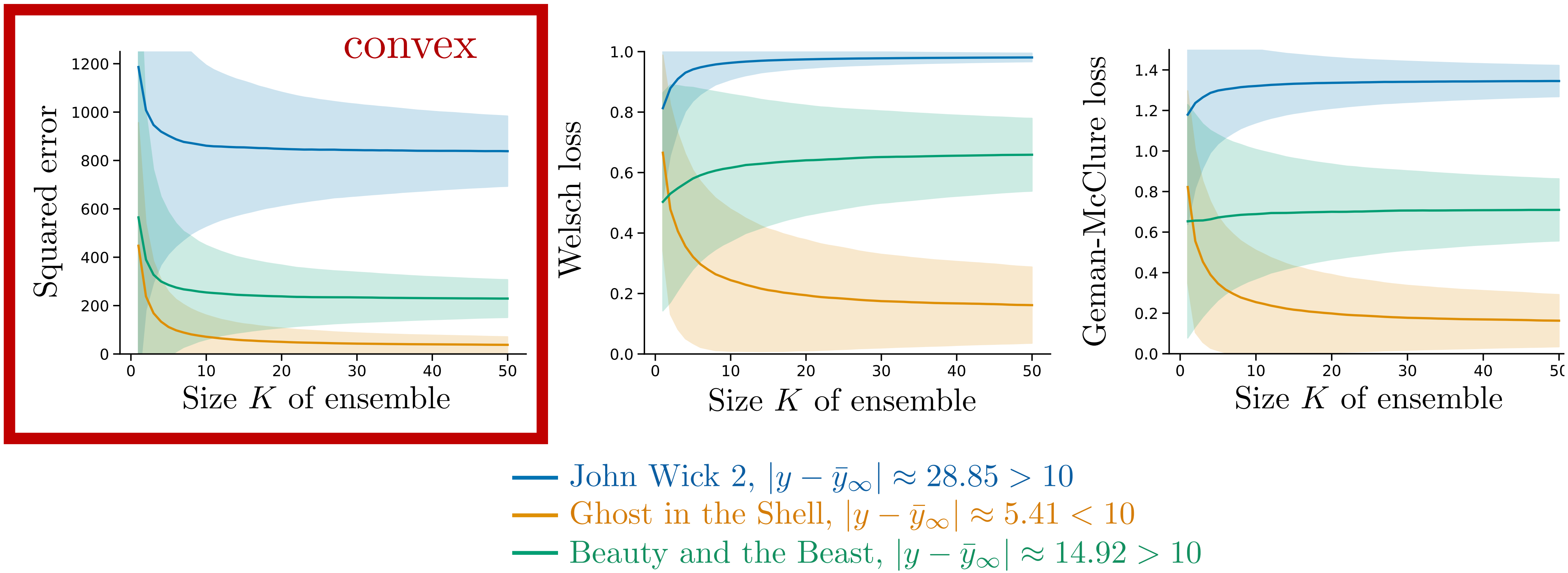
# This goes beyond ML!

- While we only talked about machine learning so far, **these insights are true as long as there is an ensemble and a loss function**, for instance in regression, parameter estimation, or collective intelligence.
- Here a crowd was asked to **predict the ratings of upcoming movies**



# This goes beyond ML!

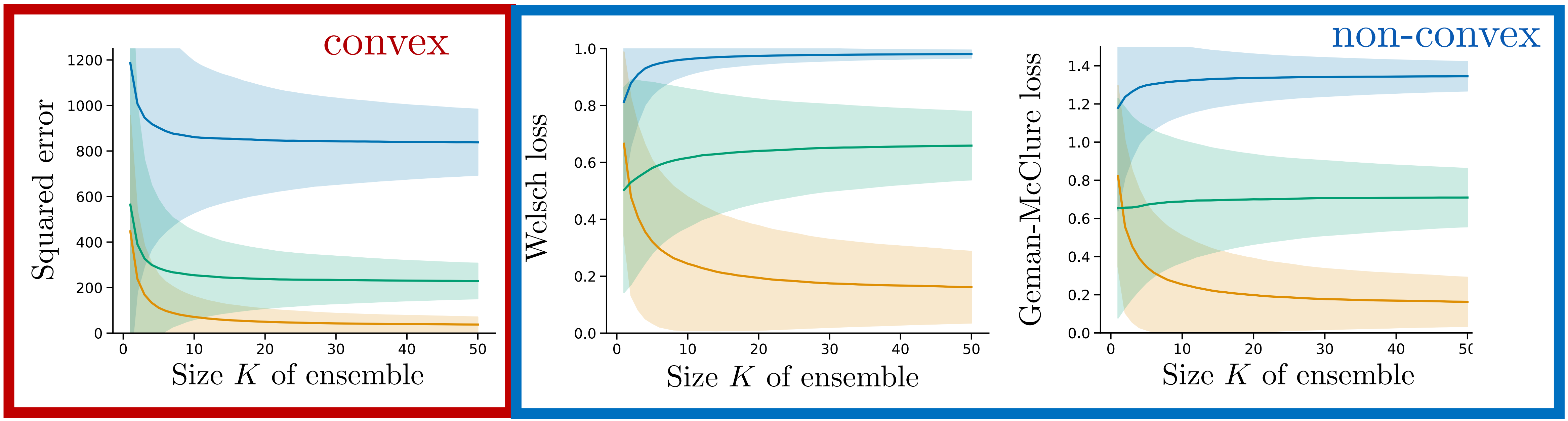
- While we only talked about machine learning so far, **these insights are true as long as there is an ensemble and a loss function**, for instance in regression, parameter estimation, or collective intelligence.
- Here a crowd was asked to **predict the ratings of upcoming movies**



*Simoiu et al., Studying the "Wisdom of Crowds" at Scale, HCOMP, 2019*

# This goes beyond ML!

- While we only talked about machine learning so far, **these insights are true as long as there is an ensemble and a loss function**, for instance in regression, parameter estimation, or collective intelligence.
- Here a crowd was asked to **predict the ratings of upcoming movies**




John Wick 2, $|y - \bar{y}_\infty| \approx 28.85 > 10$

Ghost in the Shell, $|y - \bar{y}_\infty| \approx 5.41 < 10$

Beauty and the Beast, $|y - \bar{y}_\infty| \approx 14.92 > 10$

# 4

For convex losses, ensembles are getting better all the time

# Formalising the problem

- We have predictions $\hat{y}_1, \ldots, \hat{y}_K \in C$ (e.g. the predictive probabilities of $K$ neural nets for a single given image), and we want to look at the **average prediction**

$$\overline{y}_K = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_k$$

- We also assume that we have a **loss function** $L : C \to \mathbb{R}$

- Since we want to have results on the influence of the predictions at the **data point level**, we assume that everything is fixed except the $\hat{y}s$ .

# Formalising the problem

- We have predictions $\hat{y}_1, \ldots, \hat{y}_K \in C$ (e.g. the predictive probabilities of $K$ neural nets for a single given image), and we want to look at the **average prediction**

$$\overline{y}_K = \frac{1}{K} \sum_{k=1}^{K} \hat{y}_k$$

- We also assume that we have a **loss function** $L : C \to \mathbb{R}$

- Since we want to have results on the influence of the predictions at the **data point level**, we assume that everything is fixed except the $\hat{y}s$ .

- We are going to assume that $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable (weaker than i.i.d.)

# Formalising the problem

- We have predictions $\hat{y}_1, \ldots, \hat{y}_K \in C$ (e.g. the predictive probabilities of $K$ neural nets for a single given image), and we want to look at the **average prediction**

$$\overline{y}_K = \frac{1}{K}\sum_{k=1}^{K} \hat{y}_k$$

- We also assume that we have a **loss function** $L : C \to \mathbb{R}$

- Since we want to have results on the influence of the predictions at the **data point level**, we assume that everything is fixed except the $\hat{y}s$.

- We are going to assume that $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable (weaker than i.i.d.) with mean

$$\overline{y}_\infty = \mathbb{E}\left[\hat{y}_1\right] = \cdots = \mathbb{E}\left[\hat{y}_K\right]$$

## Form

**All these assumptions are true for random forests, deep ensembles, bagging,…**

- We have predictions $\hat{y}_1, \ldots, \hat{y}_K \in C$ (e.g. the predictive probabilities of $K$ neural nets for a single given image), and we want to look at the **average prediction**

$$\overline{y}_K = \frac{1}{K}\sum_{k=1}^{K} \hat{y}_k$$

- We also assume that we have a **loss function** $L : C \to \mathbb{R}$

- Since we want to have results on the influence of the predictions at the **data point level**, we assume that everything is fixed except the $\hat{y}s$.

- We are going to assume that $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable (weaker than i.i.d.) with mean

$$\overline{y}_\infty = \mathbb{E}\left[\hat{y}_1\right] = \cdots = \mathbb{E}\left[\hat{y}_K\right]$$

# Refresher: Jensen's inequality

- If we assume that the loss is convex, we have **convexity and averages**, so using Jensen's seems like a good idea.
- Reminder: in its general form, Jensen's states that $f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$ when $f$ is convex and these expectations exist.
- A useful version is its **finite form**

$$f\left(\frac{1}{K}\sum_{k=1}^{K} x_k\right) \leq \frac{1}{K}\sum_{k=1}^{K} f\left(x_k\right)$$

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k)$$

# Jensen’s and ensembles, old school classics

- Using the finite form of Jensen’s gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k)$$

**Loss of the ensemble**

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k)$$

**Loss of the ensemble**

**Average loss of the individual models**

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K}\sum_{k=1}^{K} L(\hat{y}_k) \leq \max_{k\in\{1,\ldots,K\}} L(\hat{y}_k)$$

Loss of the ensemble

Average loss of the individual models

Loss of the worst model

## Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k) \leq \max_{k \in \{1,\ldots,K\}} L(\hat{y}_k)$$

**Loss of the ensemble**

**Average loss of the individual models**

**Loss of the worst model**

**For this, we don't even need to assume that the predictions are random variables!**

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k) \leq \max_{k \in \{1,\ldots,K\}} L(\hat{y}_k)$$

- If $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable with mean $\overline{y}_\infty = \mathbb{E}\left[\hat{y}_1\right] = \cdots = \mathbb{E}\left[\hat{y}_K\right]$, Jensen's further gives

$$L(\overline{y}_\infty) \leq \mathbb{E}\left[L\left(\overline{y}_K\right)\right]$$

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K}\sum_{k=1}^{K} L(\hat{y}_k) \leq \max_{k\in\{1,\ldots,K\}} L(\hat{y}_k)$$

- If $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable with mean $\overline{y}_\infty = \mathbb{E}\left[\hat{y}_1\right] = \cdots = \mathbb{E}\left[\hat{y}_K\right]$, Jensen's further gives

$$L(\overline{y}_\infty) \leq \mathbb{E}\left[L\left(\overline{y}_K\right)\right]$$

**Loss of the "infinite" ensemble**

# Jensen's and ensembles, old school classics

- Using the finite form of Jensen's gives

$$L(\overline{y}_K) \leq \frac{1}{K} \sum_{k=1}^{K} L(\hat{y}_k) \leq \max_{k \in \{1,\ldots,K\}} L(\hat{y}_k)$$

- If $\hat{y}_1, \ldots, \hat{y}_K$ are exchangeable with mean $\overline{y}_\infty = \mathbb{E}\left[\hat{y}_1\right] = \cdots = \mathbb{E}\left[\hat{y}_K\right]$, Jensen's further gives

$$L(\overline{y}_\infty) \leq \mathbb{E}\left[L\left(\overline{y}_K\right)\right]$$

**Loss of the "infinite" ensemble**

**Average loss of a finite ensemble**

# Jensen's and ensembles, old and new

- The arguments of the previous slides were already used in the ensemble papers from the 90s (Michael Perrone's 1993 PhD thesis, Krogh and Vedelsby, 1995, Breiman, 1996).

- When $L$ is the squared error, this is just another way to say that **ensembling reduces the variance**

- The summary was that **an ensemble with a single model is worse than one with a finite number of models $K \geq 2$, which is in turn worse than one with an infinite number of models.**

# Jensen's and ensembles, old and new

- The arguments of the previous slides were already used in the ensemble papers from the 90s (Michael Perrone's 1993 PhD thesis, Krogh and Vedelsby, 1995, Breiman, 1996).

- When $L$ is the squared error, this is just another way to say that **ensembling reduces the variance**

- The summary was that **an ensemble with a single model is worse than one with a finite number of models $K \geq 2$, which is in turn worse than one with an infinite number of models.**

- But this says nothing about the question that really interests us : **Is it always true that an ensemble of $K$ models performs better than an ensemble of $K - 1$ models?**

- Actually, playing around with Jensen's inequality allows to have a simple answer to that question.

# Playing around with Jensen's

- Our goal is to use Jensen's to show that ensembles are getting better, i.e. that

$$\mathbb{E}[L(\overline{y}_K)] \leq \mathbb{E}[L(\overline{y}_{K-1})]$$

# Playing around with Jensen's

- Our goal is to use Jensen's to show that ensembles are getting better, i.e. that

$$\mathbb{E}[L(\overline{y}_K)] \leq \mathbb{E}[L(\overline{y}_{K-1})]$$

- It would be neat to write down $\overline{y}_K$ **as some form of expectation of** $\overline{y}_{K-1}$ and directly use Jensen's.

# Playing around with Jensen's

- Our goal is to use Jensen's to show that ensembles are getting better, i.e. that

$$\mathbb{E}[L(\overline{y}_K)] \leq \mathbb{E}[L(\overline{y}_{K-1})]$$

- It would be neat to write down $\overline{y}_K$ **as some form of expectation of** $\overline{y}_{K-1}$ and directly use Jensen's. This is actually not too hard, by noting that

$$\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k$$

# Playing around with Jensen's

- Our goal is to use Jensen's to show that ensembles are getting better, i.e. that

$$\mathbb{E}[L(\overline{y}_K)] \leq \mathbb{E}[L(\overline{y}_{K-1})]$$

- It would be neat to write down $\overline{y}_K$ **as some form of expectation of** $\overline{y}_{K-1}$ and directly use Jensen's. This is actually not too hard, by noting that

$$\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k = \frac{1}{K}\sum_{j=1}^{K}\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k$$

- Using Jensen's then gives us

$$L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right) \leq \frac{1}{K}\sum_{j=1}^{K}L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)$$

# Playing around with Jensen's

- We saw that $$L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right) \leq \frac{1}{K}\sum_{j=1}^{K} L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)$$

# Playing around with Jensen's

- We saw that $L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right) \leq \frac{1}{K}\sum_{j=1}^{K} L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)$

- Averaging this expression gives

$$\mathbb{E}\left[L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right)\right] \leq \frac{1}{K}\sum_{j=1}^{K}\mathbb{E}\left[L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)\right]$$

## Playing around with Jensen's

- We saw that $L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right) \leq \frac{1}{K}\sum_{j=1}^{K} L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)$

- Averaging this expression gives

$$\mathbb{E}\left[L\left(\frac{1}{K}\sum_{k=1}^{K}\hat{y}_k\right)\right] \leq \frac{1}{K}\sum_{j=1}^{K}\mathbb{E}\left[L\left(\frac{1}{K-1}\sum_{k\neq j}\hat{y}_k\right)\right]$$

Equals $\mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$ because of exchangeability

# Ensembles get better for convex losses

- We have just shown that, when the predictions are exchangeable and the loss is convex, **ensembles are monotonically getting better**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \leq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

# Ensembles get better for convex losses

- We have just shown that, when the predictions are exchangeable and the loss is convex, **ensembles are monotonically getting better**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \leq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

  - Possible to have strict monotonicity when the loss is strongly convex

# Ensembles get better for convex losses

- We have just shown that, when the predictions are exchangeable and the loss is convex, **ensembles are monotonically getting better**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \leq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

  - Possible to have strict monotonicity when the loss is strongly convex
  - A version of this result has been published a long time ago in a different context
    - Marshall and Proschan, *An inequality for convex functions involving majorization*, Journal of Mathematical Analysis and Applications, 1965

# Ensembles get better for convex losses

- We have just shown that, when the predictions are exchangeable and the loss is convex, **ensembles are monotonically getting better**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \leq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

  - Possible to have strict monotonicity when the loss is strongly convex
  - A version of this result has been published a long time ago in a different context
    - Marshall and Proschan, *An inequality for convex functions involving majorization*, Journal of Mathematical Analysis and Applications, 1965
  - In the ML/stats litterature, versions of this result with specific losses (MSE, cross-entropy) and additional assumptions have been published (Probst and Boulesteix, 2018)

# Ensembles get better for convex losses

- We have just shown that, when the predictions are exchangeable and the loss is convex, **ensembles are monotonically getting better**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \leq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

- Possible to have strict monotonicity when the loss is strongly convex
- A version of this result has been published a long time ago in a different context
  - Marshall and Proschan, *An inequality for convex functions involving majorization*, Journal of Mathematical Analysis and Applications, 1965
- In the ML/stats litterature, versions of this result with specific losses (MSE, cross-entropy) and additional assumptions have been published (Probst and Boulesteix, 2018)
- Also related to **the monotonicity of IWAE bounds**, as first noticed by this paper
  - Noh et al., *Regularizing deep neural networks by noise: Its interpretation and optimization*, NeurIPS 2017

# Beyond the standard average?

- Can we extend these results to **more general aggregation functions?**
- This is difficult, and had to be done in a case-by-case basis. Let's look at an example that looks simple enough.

# Beyond the standard average?

- Can we extend these results to **more general aggregation functions?**
- This is difficult, and had to be done in a case-by-case basis. Let's look at an example that looks simple enough.

- We focus on **probabilistic prediction: our models are densities $\boldsymbol{p_1, \ldots, p_K}$.** A fairly general way to aggregate them is by using their **normalised generalised mean**, for a given $r \in \mathbb{R}$

$$\mathring{p}_K \propto \left( \frac{1}{K} \sum_{k=1}^{k} p_k^r \right)^{1/r}$$

❖ Razafindralambo et al., Beyond Mixtures and Products for Ensemble Aggregation: A Likelihood Perspective on Generalized Means, UAI 2026

# Is there a « wisdom of crowds » for the normalised generalised mean?

- When $\boldsymbol{r = 0}$, the normalised generalised mean corresponds to the **geometric mean** (a.k.a. product of experts).
  - Averaging the logits of deep nets
  - Averaging the scores of score-based models (e.g. diffusion models)

- When $\boldsymbol{r}$**=1**, this is just the **standard average** (a.k.a. mixture of experts).
  - Averaging the probabilities of deep nets

## Is there a « wisdom of crowds » for the normalised generalised mean?

- When $\boldsymbol{r = 0}$, the normalised generalised mean corresponds to the **geometric mean** (a.k.a. product of experts).
  - Averaging the logits of deep nets
  - Averaging the scores of score-based models (e.g. diffusion models)

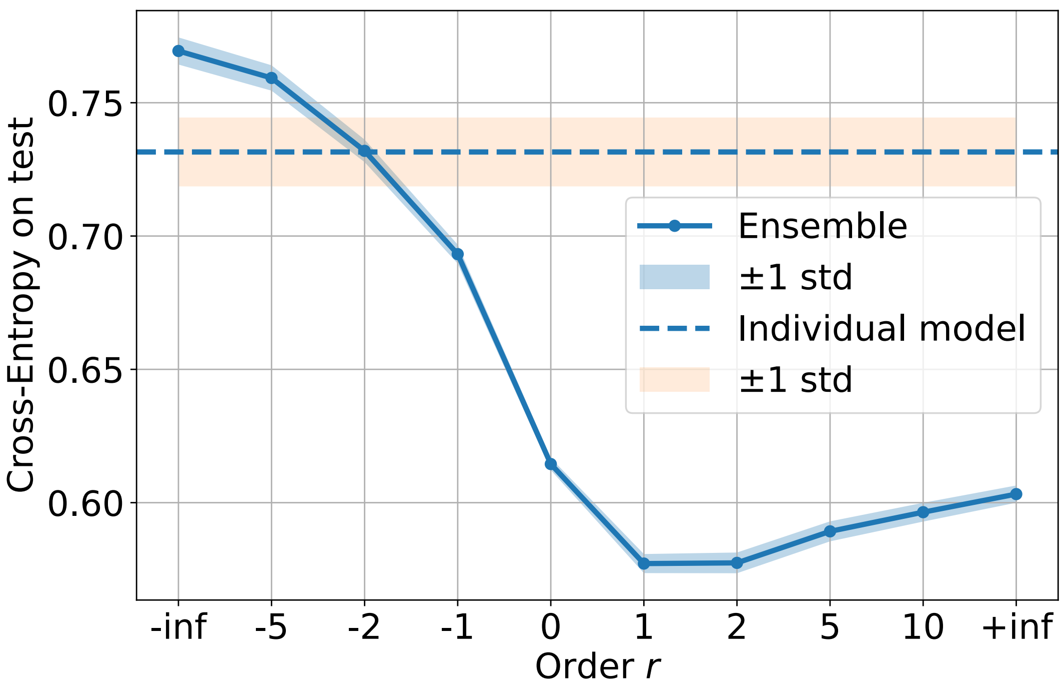
- When $\boldsymbol{r}$**=1**, this is just the **standard average** (a.k.a. mixture of experts).
  - Averaging the probabilities of deep nets

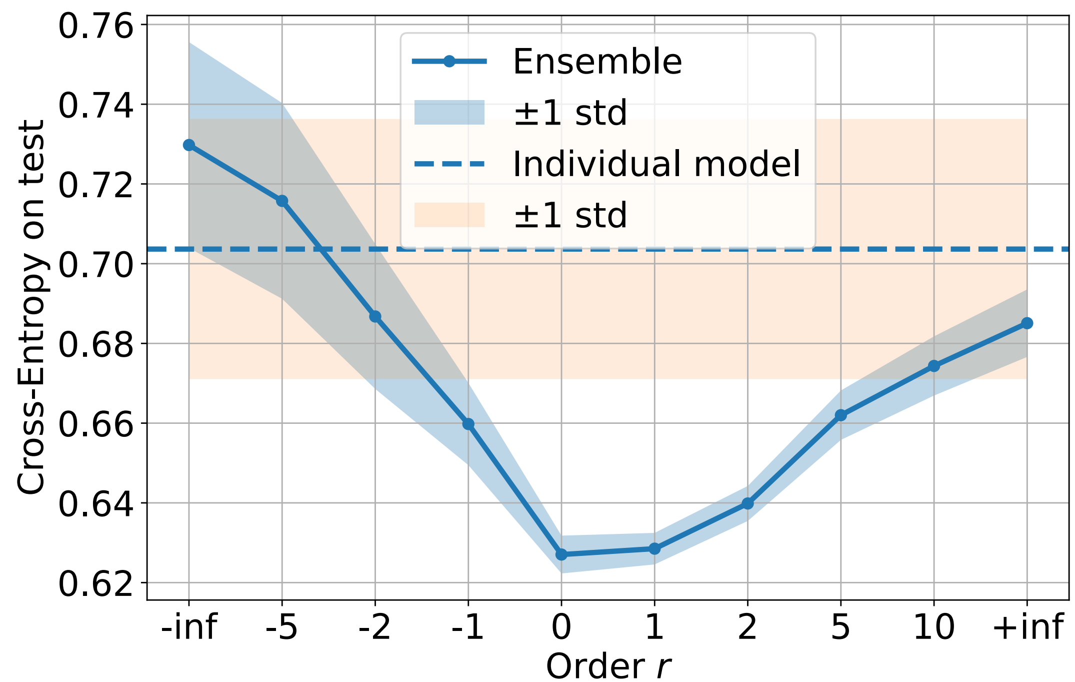
- Actually, **these are the two extreme aggregations that always work!** We showed that

$$r \in [0,1] \implies \forall x \in \mathcal{X},\ \log \mathring{p}_{K,r}(x) \geq \frac{1}{K}\sum_{k=1}^{K} \log p_k(x)$$

- When $r \notin [0,1]$, we find counter-examples where this inequality fails!

# Is there a « wisdom of crowds » for the normalised generalised mean?



# 5

# What happens for nonconvex losses?

# Can we reuse what we did for non-convex losses?

- There is an instance of nonconvex loss that can be tackled by the convex theory! Is we have a **concave loss then ensembles are always getting worse:**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \geq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

- Of course concave loss do not exist in real life, but this still gives us some insights about what’s going on.

# Can we reuse what we did for non-convex losses?

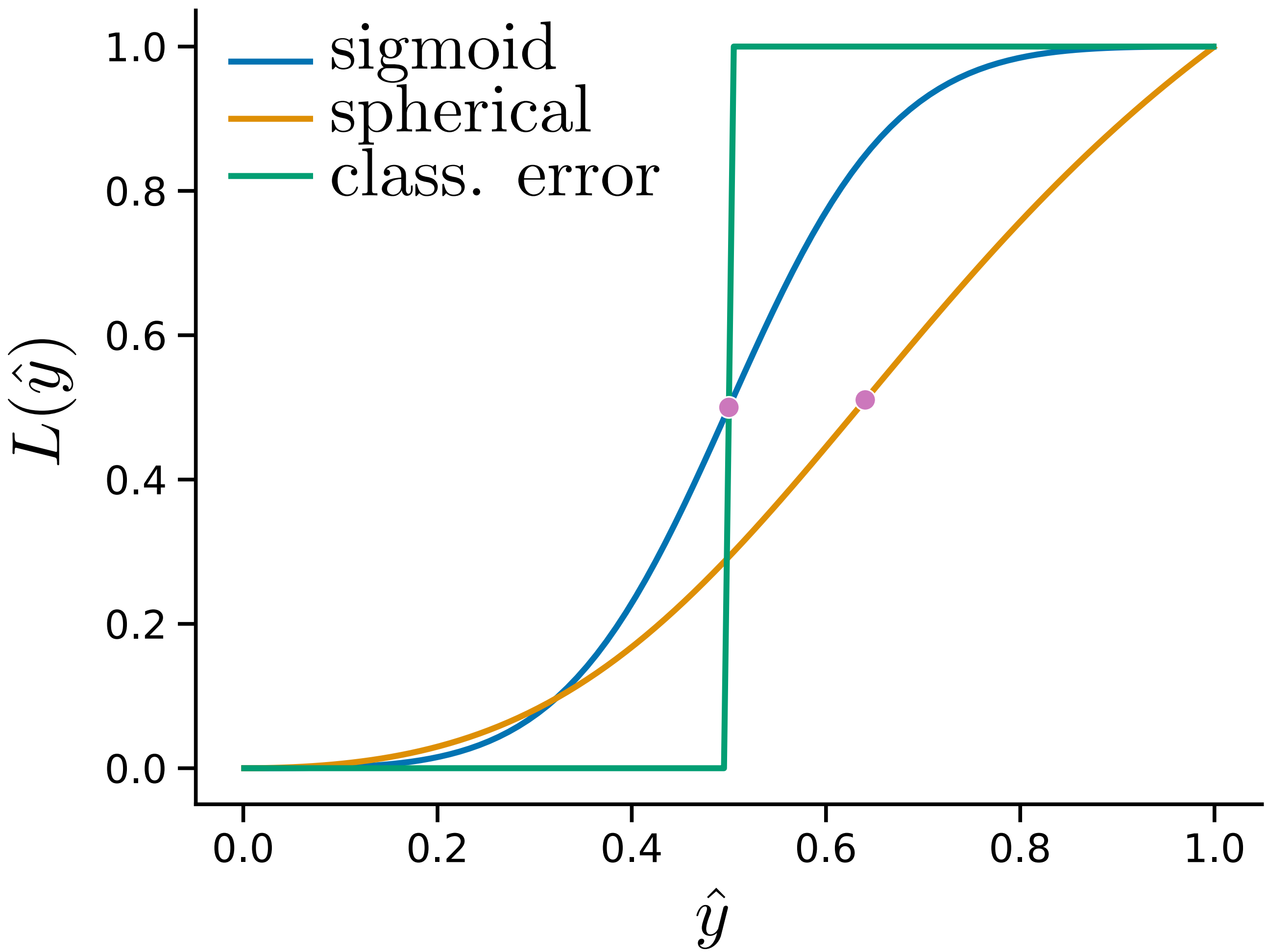
- There is an instance of nonconvex loss that can be tackled by the convex theory! Is we have a **concave loss then ensembles are always getting worse:**

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] \geq \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]$$

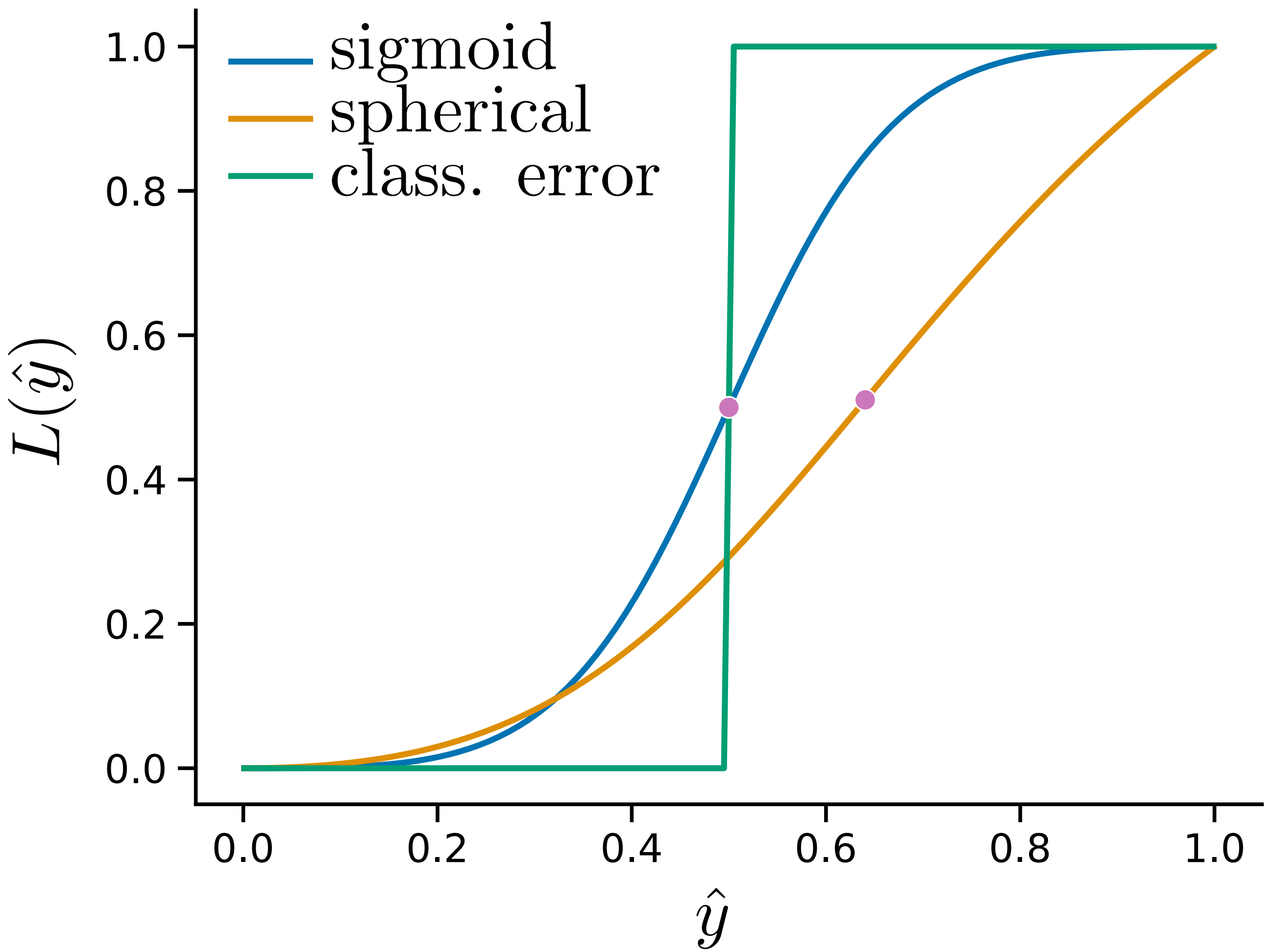
- Of course concave loss do not exist in real life, but this still gives us some insights about what's going on.

- Indeed, some losses are convex in some part of the prediction space, and convex in another part. **If our predictions mostly end up in the concave part, we can expect our ensembles to get worse!**

# Some nonconvex losses for binary classification

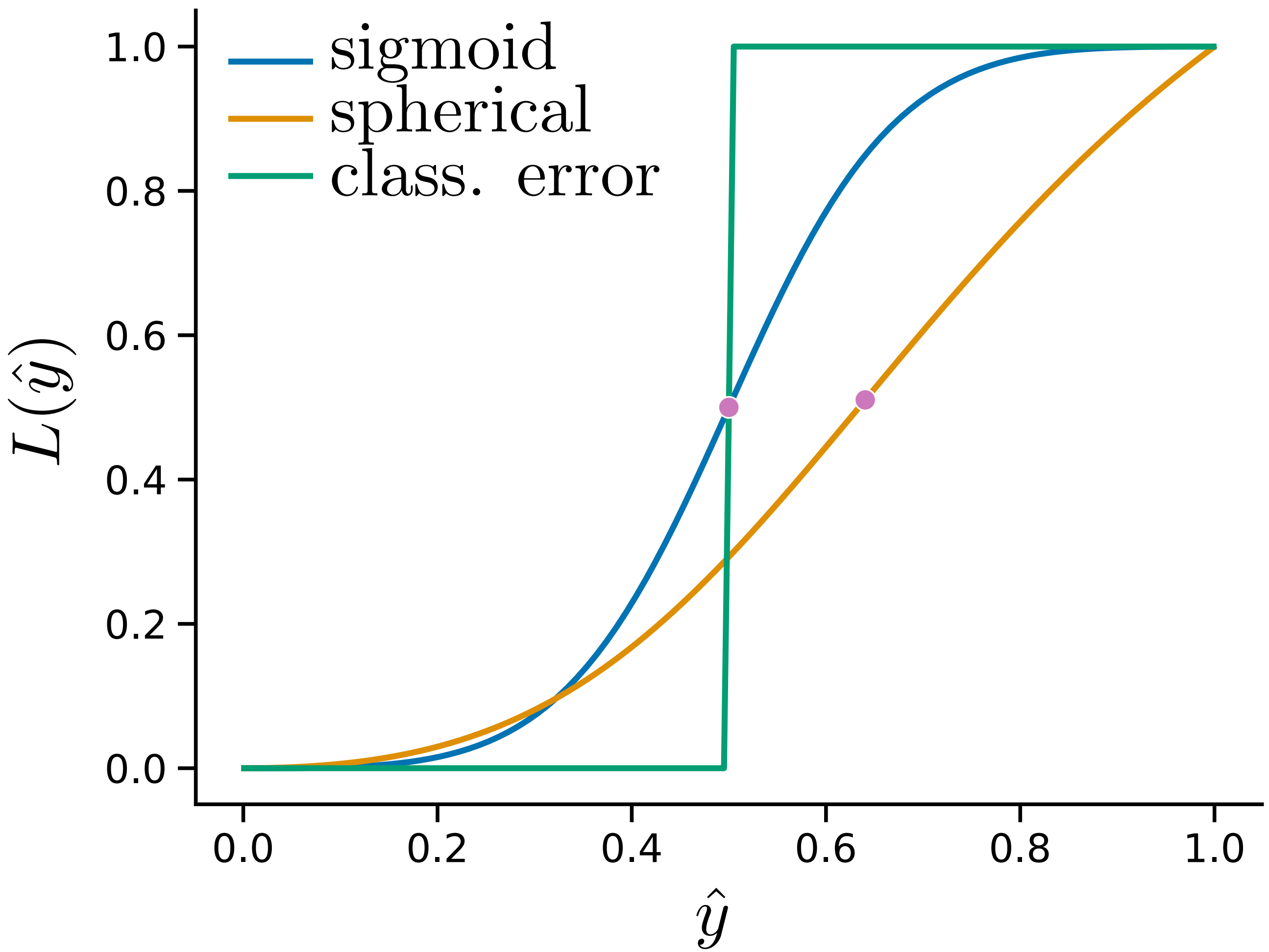


# Some nonconvex losses for binary classification




- The smooth losses are **convex when predictions are « right »** (above the purple inflexion point) and **concave when they are « wrong »**.

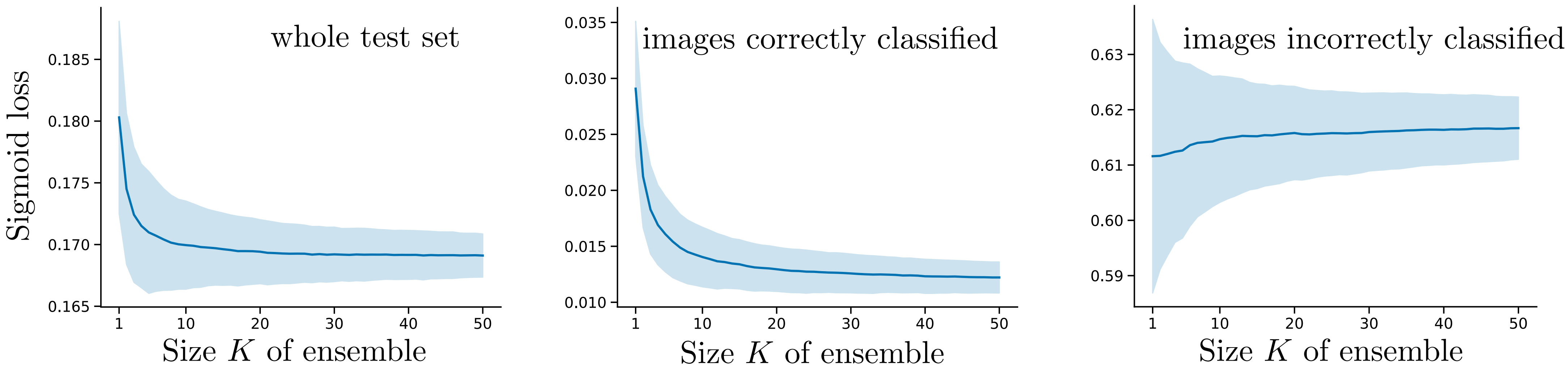
# Some nonconvex losses for binary classification



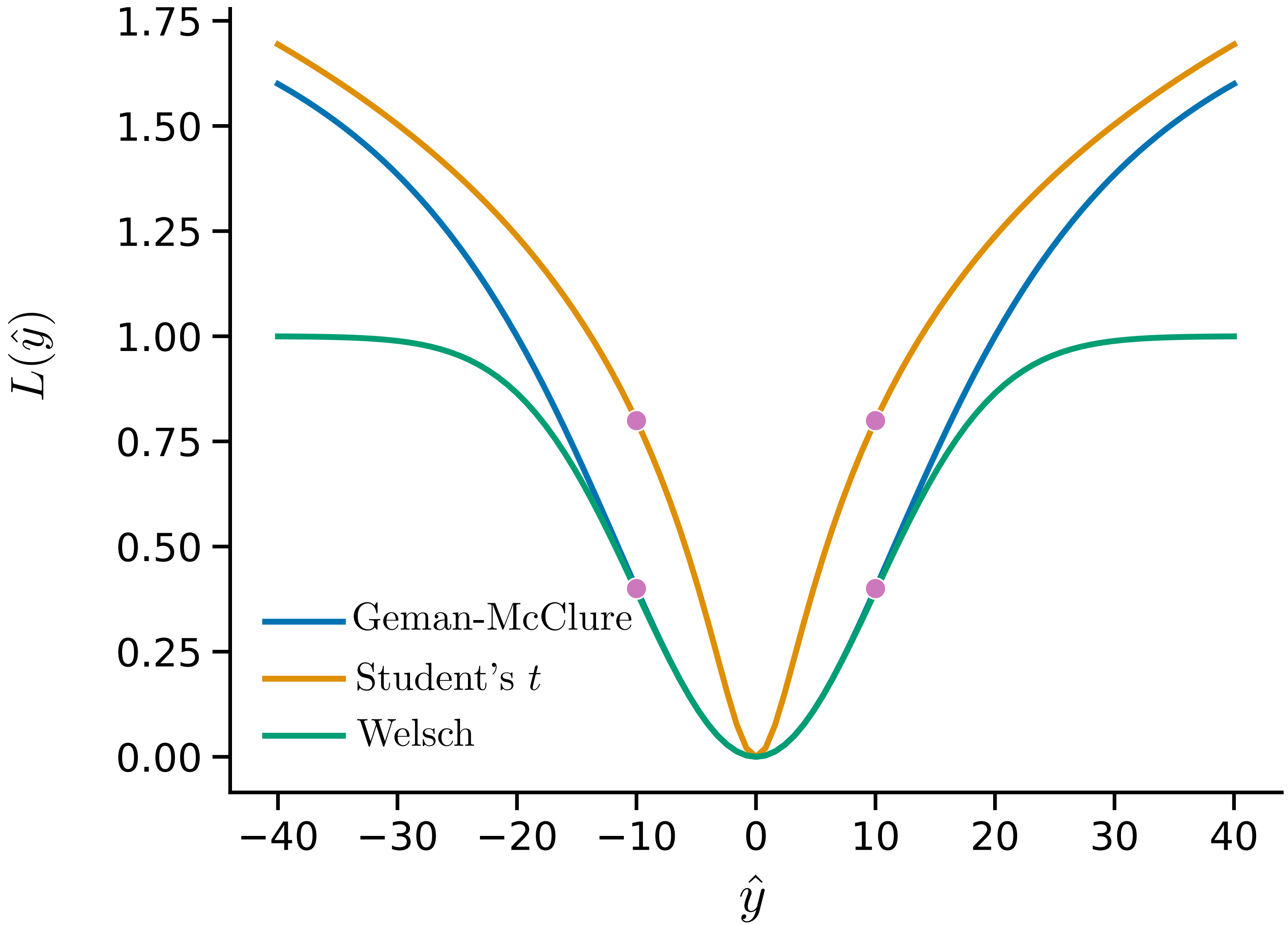

- The smooth losses are **convex when predictions are « right »** (above the purple inflexion point) and **concave when they are « wrong ».**
- The classification error can be approximated by the sigmoid loss

# The sigmoid loss behaves roughly like the classification error

Good ensembles get better, bad ones get worse

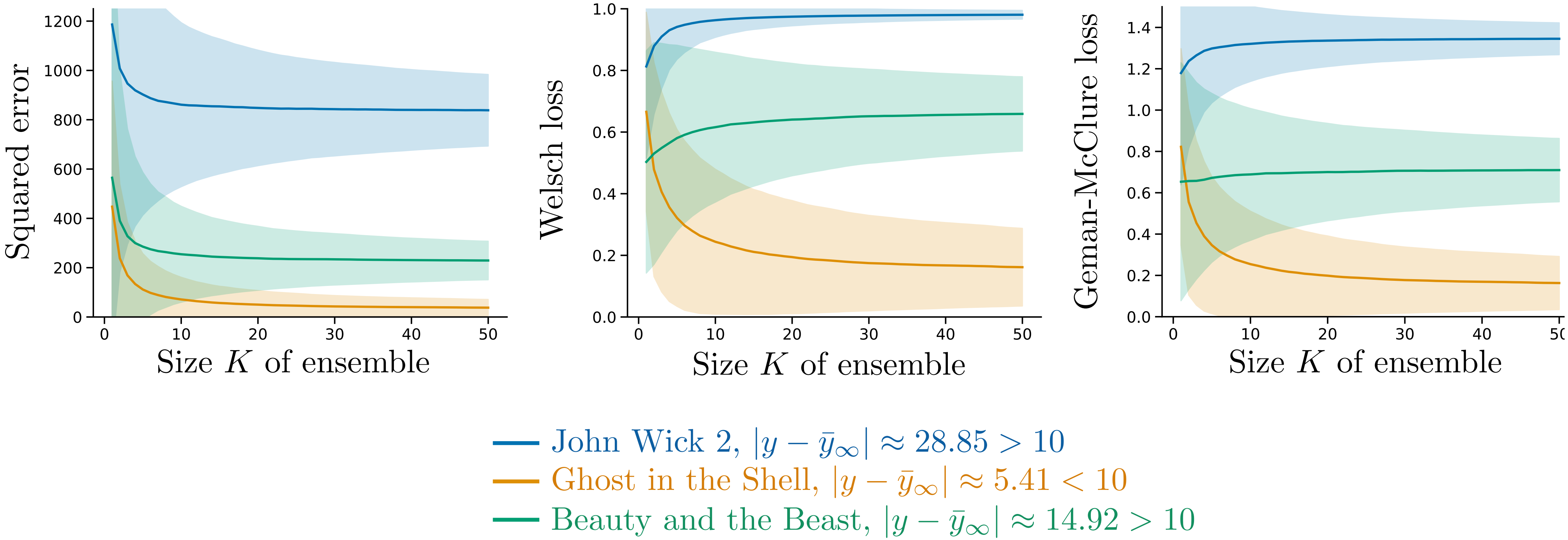


# Some nonconvex losses for regression




- The smooth losses are **convex when predictions are « right »** (above the purple inflexion point) and **concave when they are « wrong »**.
- The classification error can be approximated by the sigmoid loss
- Same insights for regressions losses.

# Again, good ensembles get better, bad ones get worse for nonconvex losses



# We can formalise this!

**Theorem.** Let $\hat{y}_1, \ldots, \hat{y}_K \in C$ be nondegenerate i.i.d. random variables whose first 5 moments are finite, and $L$ be a function with continuous and bounded partial derivatives of order up to 5, with Hessian matrix $H$. Then

# We can formalise this!

**Theorem.** Let $\hat{y}_1, \ldots, \hat{y}_K \in C$ be nondegenerate i.i.d. random variables whose first 5 moments are finite, and $L$ be a function with continuous and bounded partial derivatives of order up to 5, with Hessian matrix $H$. Then

1. If $H(\overline{y}_\infty) \succ 0$, then the ensemble is eventually getting better: for $K$ large enough,
$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] < \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right], \tag{1}$$

# We can formalise this!

**Theorem.** Let $\hat{y}_1, \ldots, \hat{y}_K \in C$ be nondegenerate i.i.d. random variables whose first 5 moments are finite, and $L$ be a function with continuous and bounded partial derivatives of order up to 5, with Hessian matrix $H$. Then

1. If $H(\overline{y}_\infty) \succ 0$, then the ensemble is eventually getting better: for $K$ large enough,
$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] < \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right], \tag{1}$$

2. If $H(\overline{y}_\infty) \prec 0$, then the ensemble is eventually getting worse: for $K$ large enough,
$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] > \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]. \tag{2}$$

## We can formalise this!

**Theorem.** Let $\hat{y}_1, \ldots, \hat{y}_K \in C$ be nondegenerate i.i.d. random variables whose first 5 moments are finite, and $L$ be a function with continuous and bounded partial derivatives of order up to 5, with Hessian matrix $H$. Then

1. If $H(\overline{y}_\infty) \succ 0$, then the ensemble is eventually getting better: for $K$ large enough,
$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] < \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right], \tag{1}$$

2. If $H(\overline{y}_\infty) \prec 0$, then the ensemble is eventually getting worse: for $K$ large enough,
$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] > \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]. \tag{2}$$



The proof essentially relies on a fourth order Taylor expansion of the loss.

# What about the classification error?

- Smooth nonconvex losses are not that popular. The only truly popular nonconvex loss is the classification error.

# What about the classification error?

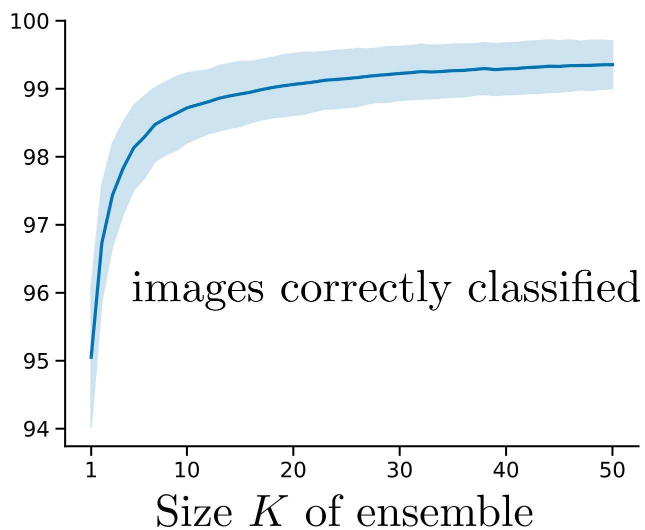
- Smooth nonconvex losses are not that popular. The only truly popular nonconvex loss is the classification error.

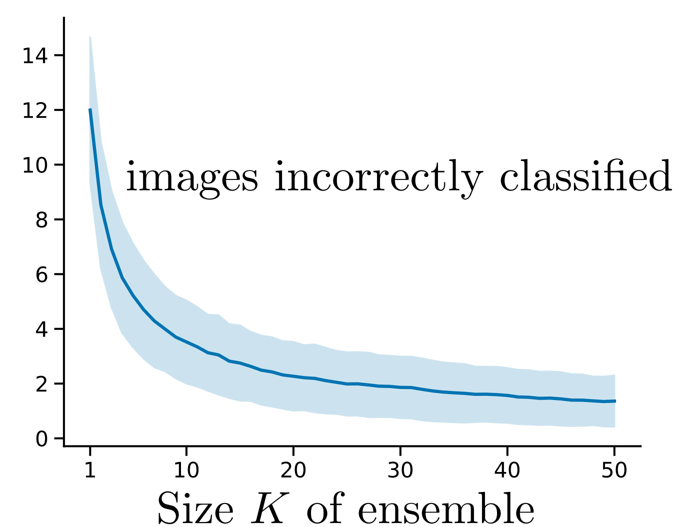
- Since the accuracy is well-approximated by the sigmoid loss, the previous result seems to indicate that **the accuracy will be increasing for points well classified**, and **decreasing for points misclassified**.

# What about the classification error?

- Smooth nonconvex losses are not that popular. **The only truly popular nonconvex loss is the classification error.**

- Since the accuracy is well-approximated by the sigmoid loss, the previous result seems to indicate that **the accuracy will be increasing for points well classified**, and **decreasing for points misclassified**. In practice, we saw in the beginning that this was indeed the case



# Can we show the previous conjecture?

- Surprisingly, things get very weird for the classification error.

- **We were looking for a quick and natural proof**
  - using the fact that the sigmoid approximates the classification error
  - using the fact that the result is true if the $\hat{y}s$ are Gaussian and then use the central limit theorem (or Berry-Esseen, or large deviations?)

- **Nothing worked.** And for good reason. Indeed, we found that there is a very simple counter–example that dates back to Condorcet (1785) !

# Condorcet's counter-example for binary classification

- For binary classification when the true label is 0, the average error is just

$$\mathbb{E}[L(\overline{y}_k)] = \mathbb{P}(\overline{y}_K \geq 0.5)$$

- Now, let $\hat{y}_1, \cdots, \hat{y}_K \sim \mathcal{B}(\overline{y}_\infty)$

- We would very like $\mathbb{E}[L(\overline{y}_k)]$ to be decreasing if $\overline{y}_\infty < 0.5$ and increasing if $\overline{y}_\infty > 0.5$

# Condorcet's counter-example for binary classification

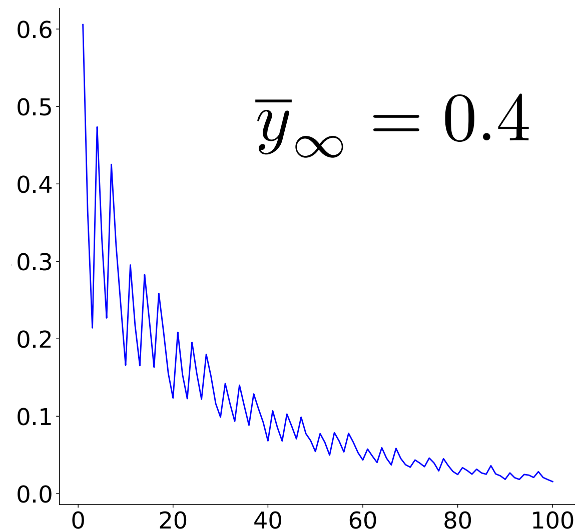
- For binary classification when the true label is 0, the average error is just

$$\mathbb{E}[L(\overline{y}_k)] = \mathbb{P}(\overline{y}_K \geq 0.5)$$

- Now, let $\hat{y}_1, \cdots, \hat{y}_K \sim \mathcal{B}(\overline{y}_\infty)$
- We would very like $\mathbb{E}[L(\overline{y}_k)]$ to be decreasing if $\overline{y}_\infty < 0.5$ and increasing if $\overline{y}_\infty > 0.5$

**but…**

**Non-monotonicity seems to be there because of the potential ties, ie when $\overline{y}_K = 0.5$**



# We did manage to do something!

Let $\hat{y}_1, \ldots, \hat{y}_K \in \mathbb{R}^{n_{\text{Cl}}}$ be i.i.d. random variables. Then

1. If the prediction is asymptotically correct (Weird assumption 1), then the ensemble is eventually getting better: for $K$ large enough,

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] < \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right], \tag{1}$$

2. If the prediction is asymptotically completely incorrect (Weird assumption 2), then the ensemble is eventually getting ~~better~~ worse: for $K$ large enough,

$$\mathbb{E}\left[L\left(\overline{y}_K\right)\right] > \mathbb{E}\left[L\left(\overline{y}_{K-1}\right)\right]. \tag{2}$$



Proof idea: use strong large deviation theorems: Bahadur-Ranga Rao-Petrov for binary classification and Joutard for multiclass.

# Our basic (kinda new) tool

**Theorem 1 (monotonicity of tail probabilities, univariate)** *Let* $X_1, \ldots, X_n$ *be i.i.d. random variables with finite expectation* $\mu$*, and let* $\varepsilon > 0$*. Assume furthermore that*

1. $\mathbb{E}\left[\mathrm{e}^{tX_1}\right] < +\infty$ *for all* $t \in \mathbb{R}$*;*

2. $\mathbb{P}\left(X_1 > \mu + \epsilon\right) > 0$*;*

3. $X_1$ *is absolutely continuous with respect to the Lebesgue measure;*

*or, alternatively to (3),*

*(3bis)* $X_1$ *is a lattice random variable and* $\mathbb{P}\left(X_1 = \mu + \epsilon\right) > 0$*.*

*Then,* $\mathbb{P}\left(\overline{X}_n \geq \mu + \varepsilon\right)$ *and* $\mathbb{P}\left(\overline{X}_n > \mu + \varepsilon\right)$ *are both strictly decreasing for all* $n$ *large enough.*



Similar result in the multivariate case but much more complicated...

# Conclusion

- From a practical perspective, our results strengthen and confirm the message of Probst and Boulesteix: **ensemble size should not be tuned, and ensembles should be as large as possible.**

- **This work was solely about the « pure variance reduction effect » of generic ensembles**. Of course, this is a **very small part of the story**

- Theoretical work on specific kinds of ensembles is also very important
  - Scornet and Hooker, **Theory of Random Forests: A Review**, HAL-05006431, 2025
  - Wild et al., **A Rigorous Link between Deep Ensembles and (Variational) Bayesian Methods**, NeurIPS 2023
  - Hron et al., **Variational BayesiaVariational Bayesian dropout: pitfalls and fixes**, ICML 2023