Rethinking Early Stopping: Refine, Then Calibrate

Eugène Berta, David Holzmüller, Michael I. Jordan, Francis Bach



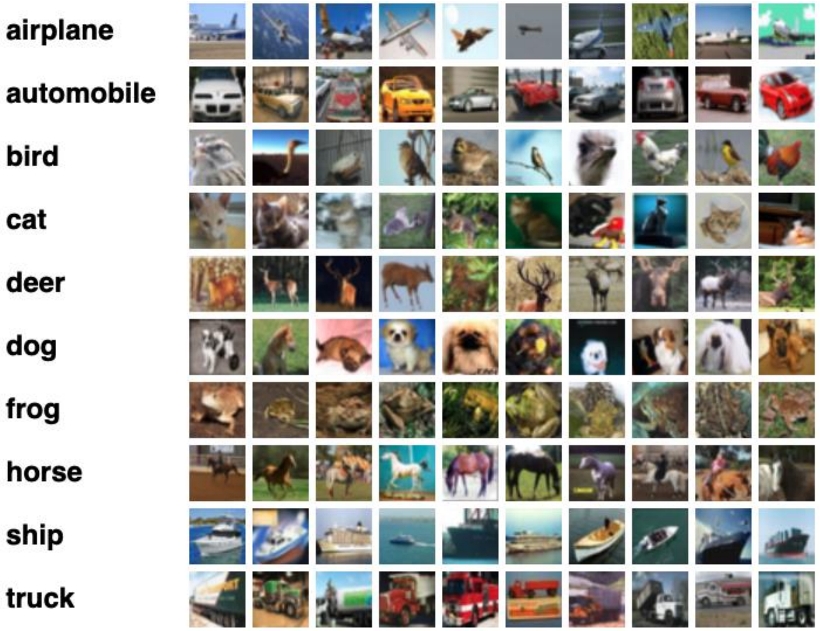

Outline

- Motivating example
- Loss function decomposition in classification
- Proposed method
- Empirical results
- A theoretical analysis: logistic regression in the high dimensional Gaussian data model

Motivating example

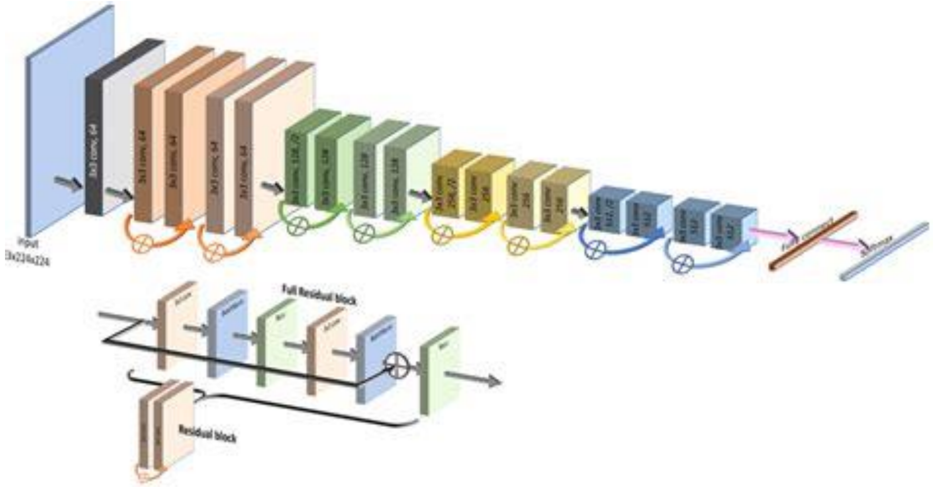
Dataset D

Images, tabular, text…



Machine learning classifier f

logistic regression, boosted trees, neural net…



Predictions

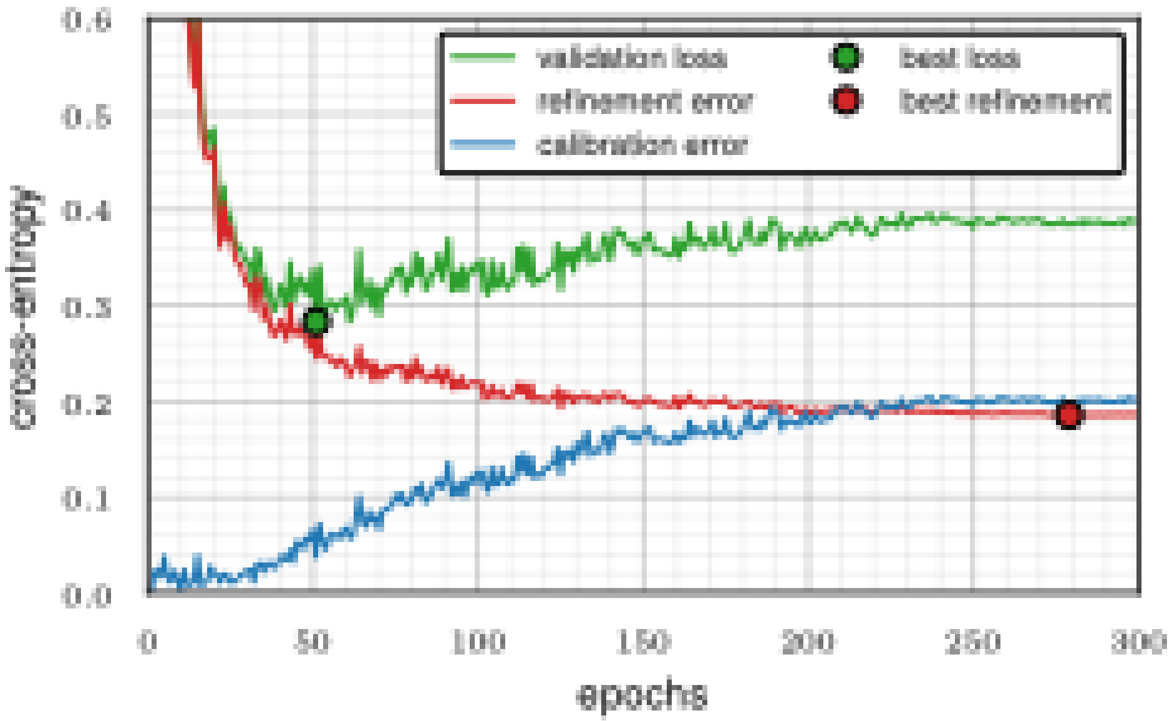
0.02	**airplane**
0.9	**automobile**
0.	**bird**
0.005	**cat**
0.	**deer**
0.005	**dog**
0.	**frog**
0.	**horse**
0.02	**ship**
0.05	**truck**

Motivating example

Model fitting

training, hyper-parameter search…

$$\boxed{\min_{f \in \mathcal{F}} \mathrm{Risk}_D(f)}$$



Training a ResNet-18 on CIFAR-10. We plot the cross-entropy loss on the validation set, with its calibration and refinement error terms.

What is this decomposition?

Is there a better way to train classifiers?

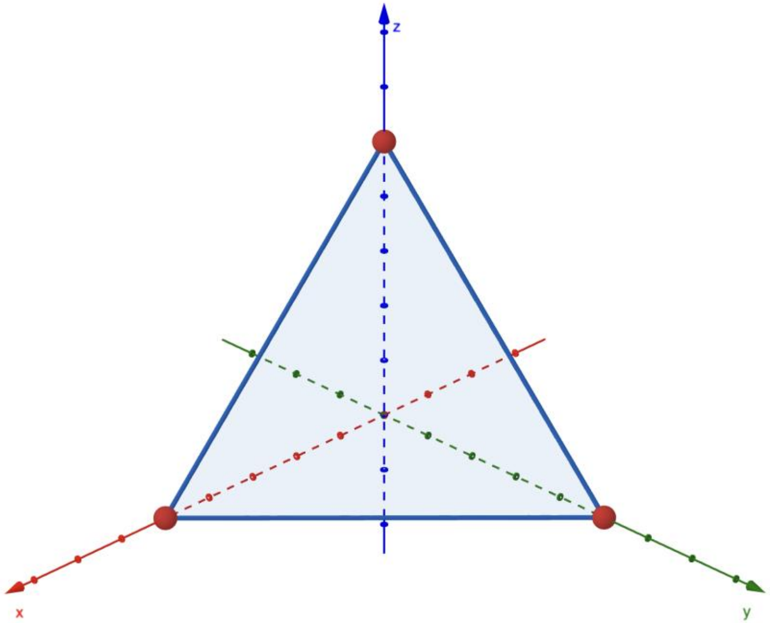
Proper loss functions in classification

Predictions in $\Delta_k = \{p \in [0,1]^k | \mathbf{1}^\top p = 1\}$, labels in $\mathcal{Y}_k = \{y \in \{0,1\}^k | \mathbf{1}^\top y = 1\}$.

Evaluated with loss functions $\ell : \Delta_k \times \mathcal{Y}_k \to \mathbb{R}_+$,

such as:

- The Brier score $\ell(p, y) = \|y - p\|_2^2$
- The log-loss $\ell(p, y) = -\sum_{i=1}^{k} y_i \log(p_i)$



The probability simplex (blue triangle) and label space (red dots) for k=3.

We overload the notation: $\ell(p, q) = \mathbb{E}_{y \sim q}[\ell(p, y)]$

A natural requirement is that $\ell(q, q) \leq \ell(p, q), \forall p, q$.

Then, ℓ is called proper (**log-loss and brier are proper losses**).

Gneiting, T., & Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association.*

Decomposition of proper losses

In machine learning, we usually have $(X, Y) \sim \mathcal{D}$.

We make predictions $p = f(X)$ with a model $f : \mathcal{X} \to \Delta_k$.

In this setting, for any proper loss,

$$\mathrm{Risk}_{\mathcal{D}}(f) = \mathbb{E}_{\mathcal{D}}[\ell(f(X), Y)] = \mathbb{E}_{\mathcal{D}}[d_\ell(f(X), C)] + \mathbb{E}_{\mathcal{D}}[e_\ell(C)]$$

with $\underbrace{d_\ell(p, q) = \ell(p, q) - \ell(q, q)}_{\ell\text{-divergence}}$, $\underbrace{e_\ell(q) = \ell(q, q)}_{\ell\text{-entropy}}$, and $\underbrace{C = \mathbb{E}_{\mathcal{D}}[Y | f(X)]}_{\text{Calibrated scores}}$.

Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society.*

Kull, M., & Flach, P. (2015). Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. *Machine Learning and Knowledge Discovery in Databases: European Conference.*

Decomposition of proper losses

$$\underbrace{\mathbb{E}_{\mathcal{D}}[\ell(f(X),Y)]}_{\text{Risk}} = \underbrace{\mathbb{E}_{\mathcal{D}}[d_\ell(f(X),C)]}_{\text{Calibration error}} + \underbrace{\mathbb{E}_{\mathcal{D}}[e_\ell(C)]}_{\text{Refinement error}}$$

Risk: How good are my predictions?

=

Calibration error: is my model over/under confident?

+

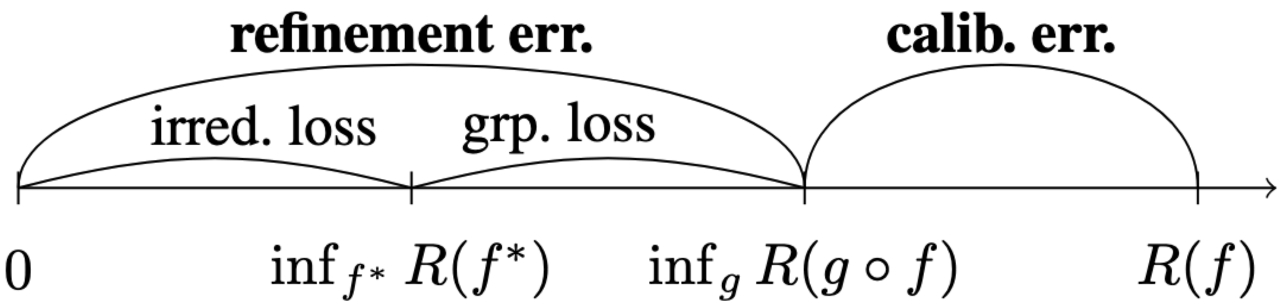
Refinement error: how well does my model separates classes? (accuracy, AUROC)

Proper loss ℓ	Divergence d_ℓ	Entropy e_ℓ
Logloss $-\sum_i y_i \log(p_i)$	KL divergence $\sum_i q_i \log \frac{q_i}{p_i}$	Shannon entropy $-\sum_i q_i \log q_i$
Brier score $\|y-p\|_2^2$	Squared distance $\|p-q\|_2^2$	Gini index $\sum_i q_i(1-q_i)$

A new variational decomposition

Theorem: Refinement error: $\mathcal{R}_\ell(f) = \min_g \mathrm{Risk}_\mathcal{D}(g \circ f)$

Calibration error: $\mathcal{K}_\ell(f) = \mathrm{Risk}_\mathcal{D}(f) - \min_g \mathrm{Risk}_\mathcal{D}(g \circ f)$



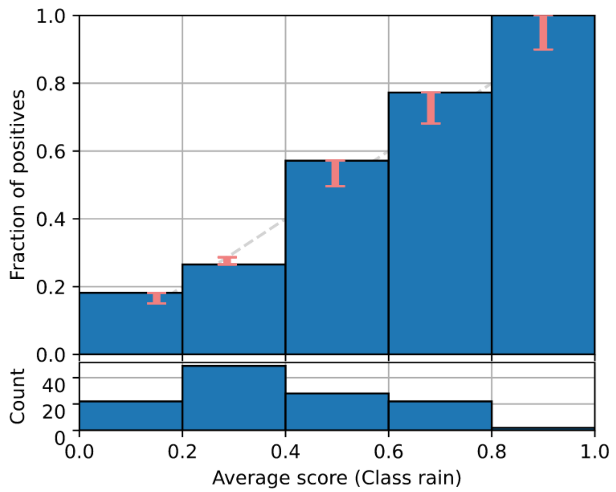
Calibration in the ML literature

$$\mathbb{E}_{\mathcal{D}}[d(f(X), \mathbb{E}[Y|f(X)])]$$

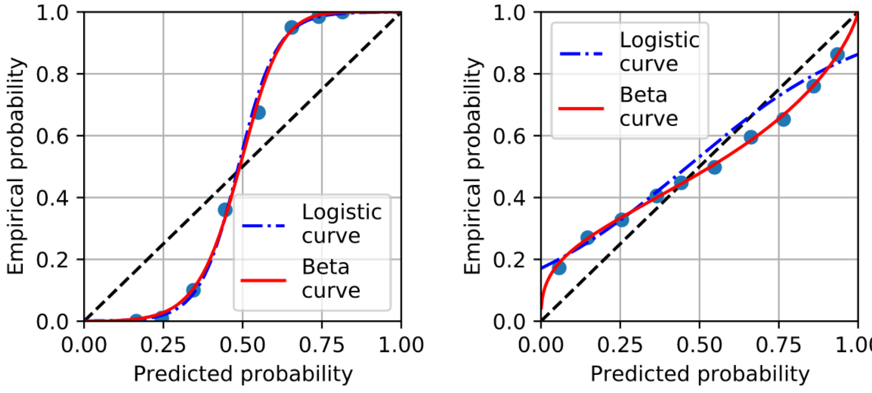
D_2 L_1 😱?

Binning

- Biased
- Inconsistent
- Parameter dependent
- Multiclass?

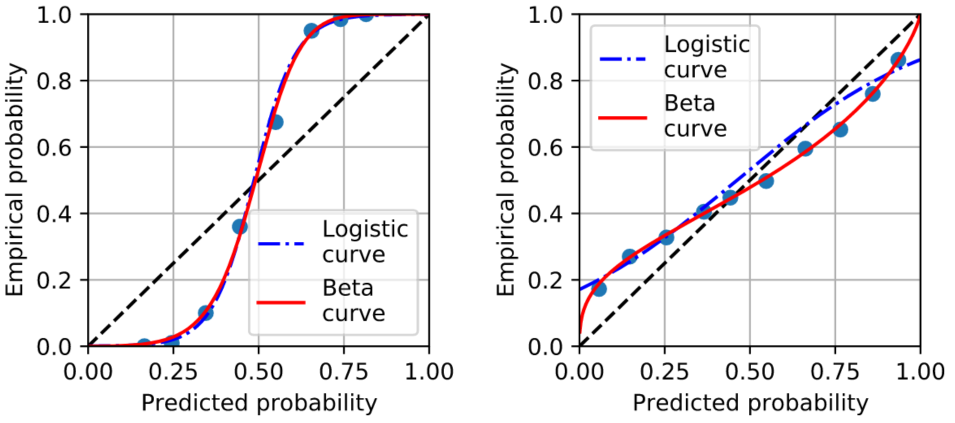


Post-hoc calibration:



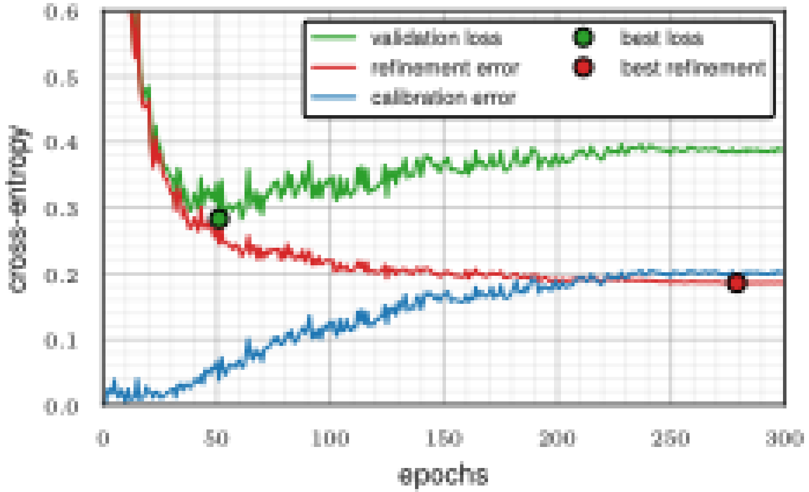
(a) Underconfidence (b) Overconfidence

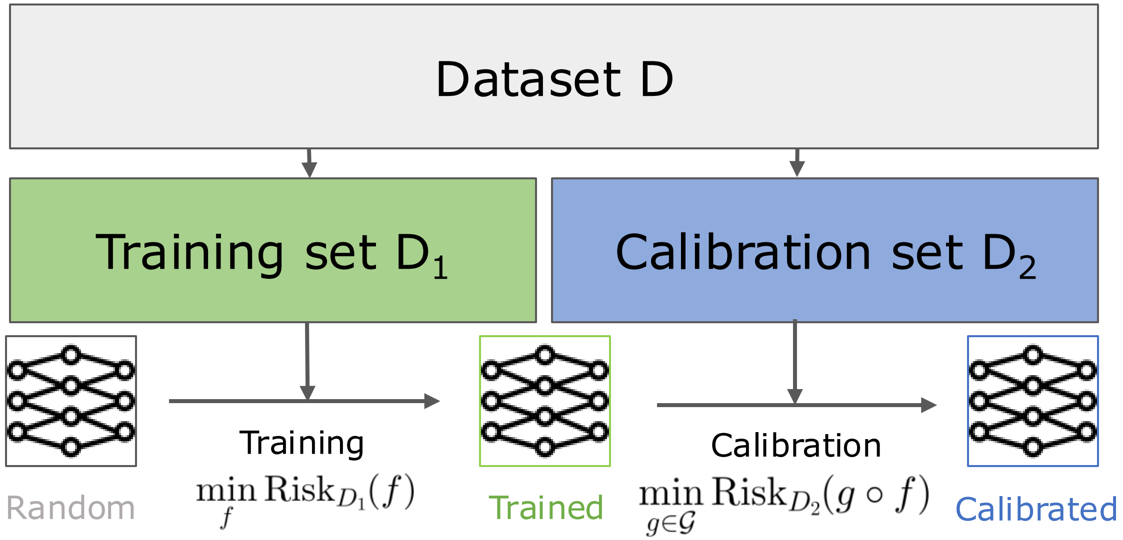
Post-hoc calibration

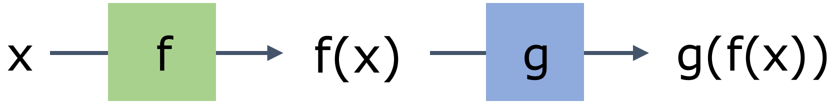


(a) Underconfidence

(b) Overconfidence







Post-hoc calibration

Isotonic regression

$$\min_{g \nearrow} \mathrm{Risk}_{D_2}(g \circ f)$$

✅ Preserves the ROC convex hull.
✅ Theoretical guarantees.
❌ Ill defined in the multi-class case.

Temperature scaling

$$\min_{\alpha \in \mathbb{R}} \mathrm{Risk}_{D_2}(g_\alpha \circ f)$$

Where $g_\alpha(p) = \mathrm{Softmax}(\alpha \log(p))$

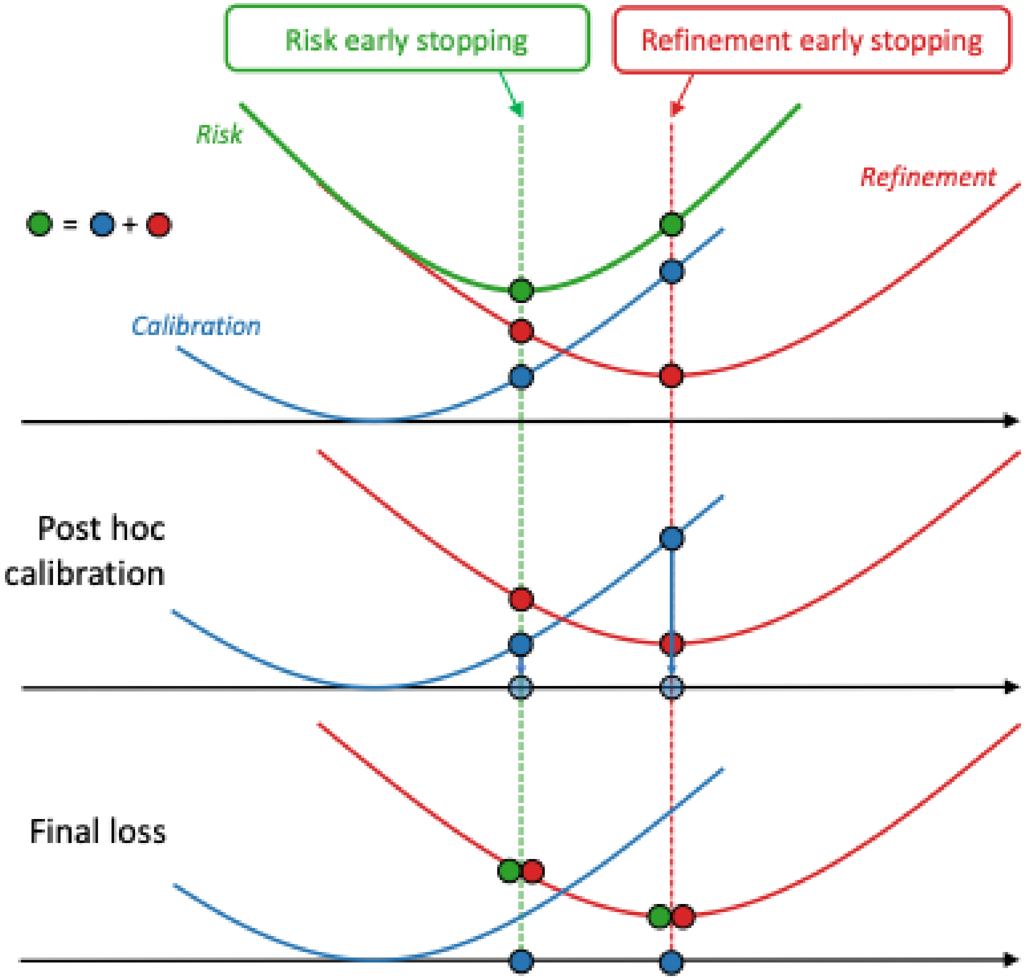
✅ Preserves refinement error.
✅ Inherently multi-class.
❌ No theoretical guarantees?

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. *International conference on Knowledge discovery and data mining*.

Berta, E., Bach, F. & Jordan, M.. (2024). Classifier Calibration with ROC-Regularized Isotonic Regression. *International Conference on Artificial Intelligence and Statistics*.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *International conference on machine learning*.

Our method: Refine, Then Calibrate



Early stopping	Training minimizes	Post hoc minimizes
Risk	Cal. + Ref.	Cal.
Refinement	Ref.	Cal.

How can we estimate refinement?

Validation accuracy? Area under the ROC curve?

$$\mathcal{R}_\ell(f) = \min_g \mathrm{Risk}_{\mathcal{D}}(g \circ f)$$

$$\mathcal{R}_\ell(f) \simeq \min_{g \in \mathcal{G}} \mathrm{Risk}_{D_2}(g \circ f)$$

$$\downarrow$$

Validation loss after post-hoc calibration.

Choosing $\mathcal{G}$

Large $\mathcal{G}$?

e.g. Isotonic regression

- ✅ little bias in our estimator
- ❌ over-fitting the validation set D_2

Small $\mathcal{G}$?

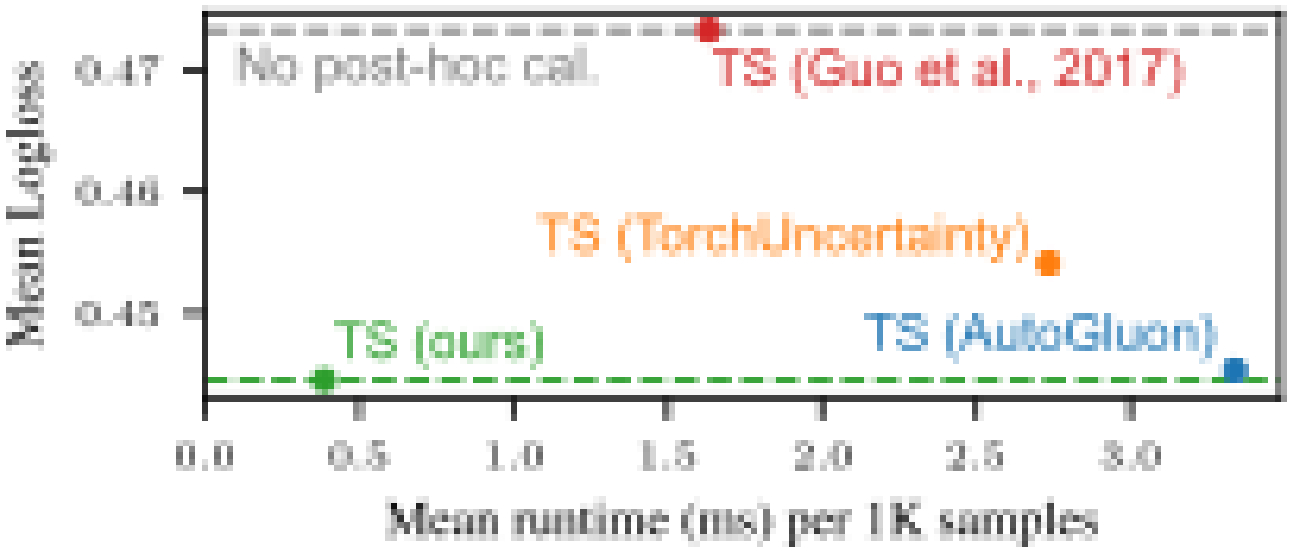
e.g. Temperature scaling

- ✅ robust to over-fitting
- ❌ biased estimator? Unless close to $g^*(f(X)) = \mathbb{E}_{\mathcal{D}}[Y|f(X)]$

We evaluate **TS-refinement** = validation loss after temperature scaling

⚠️ Could be any other refinement estimator.

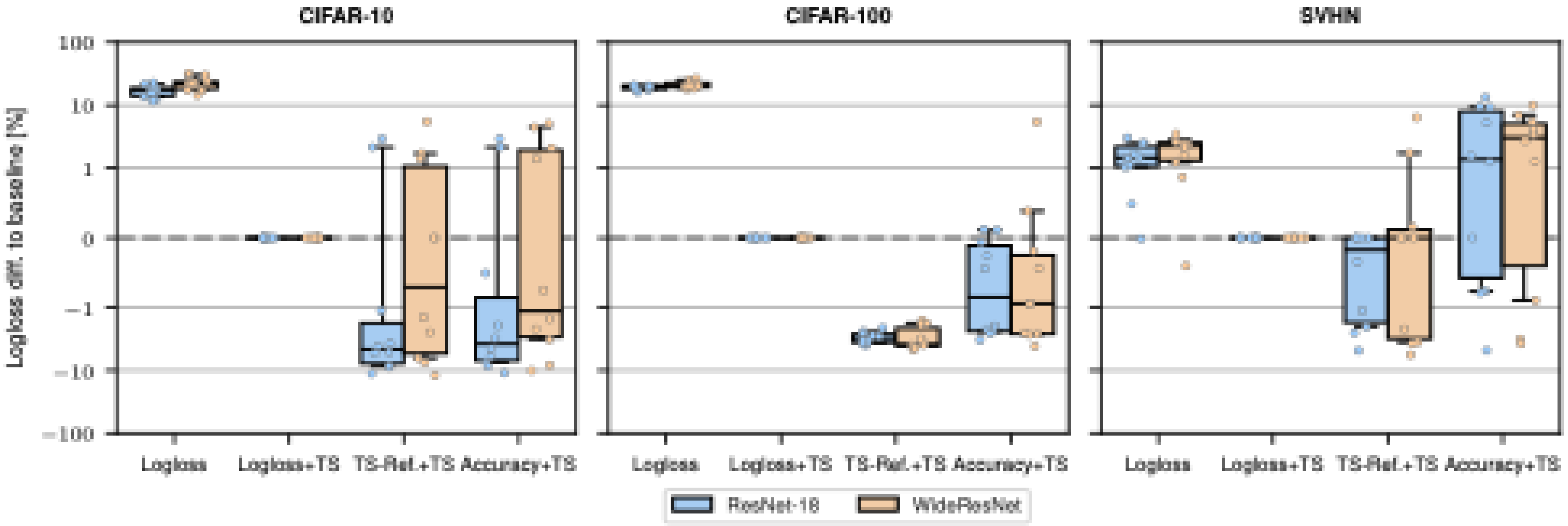
Use the best implementation, ours!



Runtime versus mean benchmark scores of different TS implementations.
Runtimes are averaged over validation sets with 10K+ samples. Evaluation is on XGBoost models trained with default parameters, using the epoch with the best validation accuracy.

github.com/dholzmueller/probmetrics

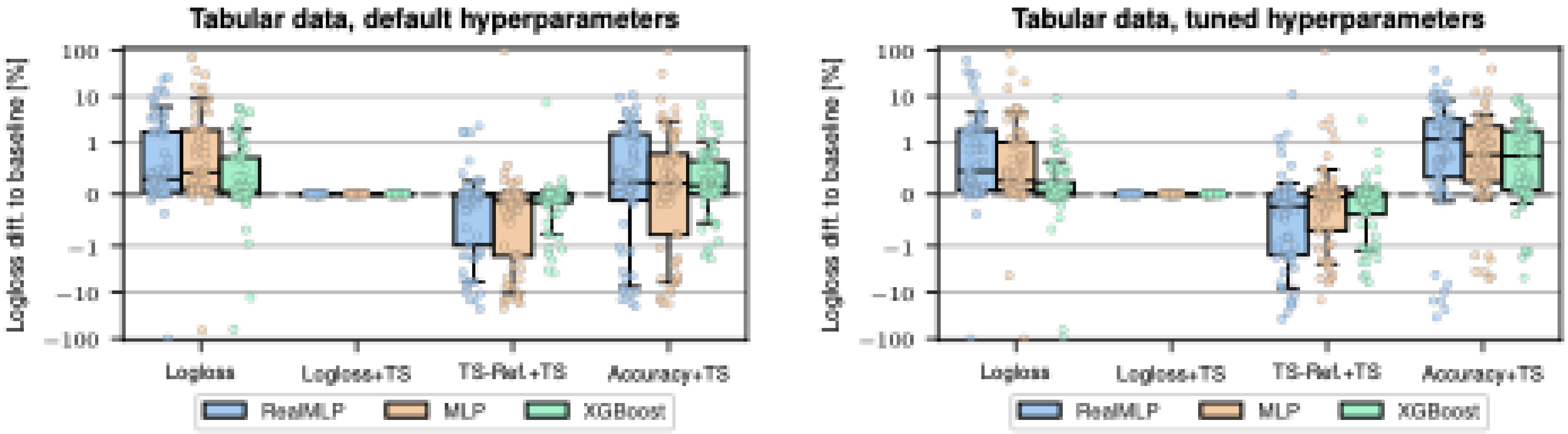
Results – computer vision



Relative differences in test log-loss (lower is better) between logloss+TS and other procedures on vision datasets.
"+TS" indicates temperature scaling applied to the final model. Each dot represents a training run on one dataset. Box-plots show the 10%, 25%, 50%, 75%, and 90% quantiles. Relative differences (y-axis) are plotted using a log scale.

github.com/eugeneberta/RefineThenCalibrate-Vision

Results – tabular data



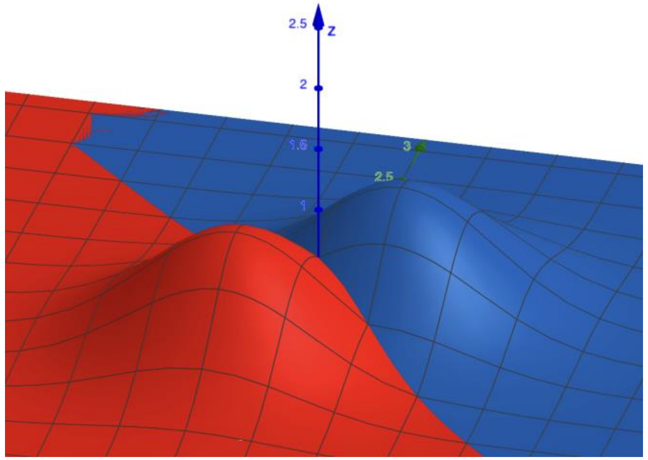
Relative differences in test logloss (lower is better) between logloss+TS and other procedures on tabular datasets.
"+TS" indicates temperature scaling applied to the final model. Each dot represents one dataset with 10K+ samples. Percentages are clipped to [−100, 100] due to one outlier with almost zero loss. Box-plots show the 10%, 25%, 50%, 75%, and 90% quantiles. Relative differences (y-axis) are plotted using a log scale.

github.com/dholzmueller/pytabkit

Theoretical analysis: the Gaussian data model

Gaussian data model:

$$X \in \mathbb{R}^p, Y \in \{-1, 1\} \begin{cases} X \sim \mathcal{N}(\mu, \Sigma) \text{ if } Y = 1 \\ X \sim \mathcal{N}(-\mu, \Sigma) \text{ if } Y = -1 \end{cases}$$



Linear classifier:

$$f(X) = \sigma(w^\top X)$$ with $$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

In this well studied setting, $w^* = 2\Sigma^{-1}\mu$

Theoretical analysis: the Gaussian data model

The error rate writes $\mathrm{err}(w) = \Phi(-a_w/2)$ with, $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-\frac{t^2}{2})dt$

And $\underbrace{a_w = \frac{\langle w, w^* \rangle_\Sigma}{\|w\|_\Sigma}}_{\text{Expertise level}}$ with $\underbrace{\|w\|_\Sigma = \sqrt{w^\top \Sigma w}}_{\text{Confidence level}}$, $\langle w, w^* \rangle_\Sigma = w^\top \Sigma w^*$

Theorem 5.1. *For proper loss ℓ, the calibration and refinement errors of our model are*

$$\mathcal{K}_\ell(w) = \mathbb{E}\Big[d_\ell\Big(\sigma\Big(\|w\|_\Sigma\Big(z + \frac{a_w}{2}\Big)\Big), \sigma\Big(a_w\Big(z + \frac{a_w}{2}\Big)\Big)\Big)\Big]$$
$$\mathcal{R}_\ell(w) = \mathbb{E}\Big[e_\ell\Big(\sigma\Big(a_w\Big(z + \frac{a_w}{2}\Big)\Big)\Big)\Big],$$

where the expectation is taken on $z \sim \mathcal{N}(0,1)$.

Theorem 5.2. *The re-scaled weight vector $w_s \leftarrow sw$ with $s = \langle w, w^* \rangle_\Sigma / \|w\|_\Sigma^2$ yields null calibration error $\mathcal{K}(w_s) = 0$ while preserving the refinement error $\mathcal{R}(w_s) = \mathcal{R}(w)$.*

Theoretical analysis: regularized logistic regression in high dimension

The weight vector learned with regularized logistic regression:

$$\min_{w \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i w^\top X_i)) + \frac{\lambda}{2} \|w\|^2$$

Has the following distr. when $n, p \to \infty$ with a constant ratio,

$$w_\lambda \sim \mathcal{N}\left(\eta(\lambda I_p + \tau\Sigma)^{-1}\mu, \frac{\gamma}{n}(\lambda I_p + \tau\Sigma)^{-1}\Sigma(\lambda I_p + \tau\Sigma)^{-1} \right)$$

We deduce **Proposition 6.1.** *For $n, p \to \infty$,*

$$\langle w_\lambda, w^* \rangle_\Sigma \xrightarrow{P} \mathbb{E}_{\sigma \sim F}\left[\frac{2\eta c^2}{\lambda + \tau\sigma} \right],$$

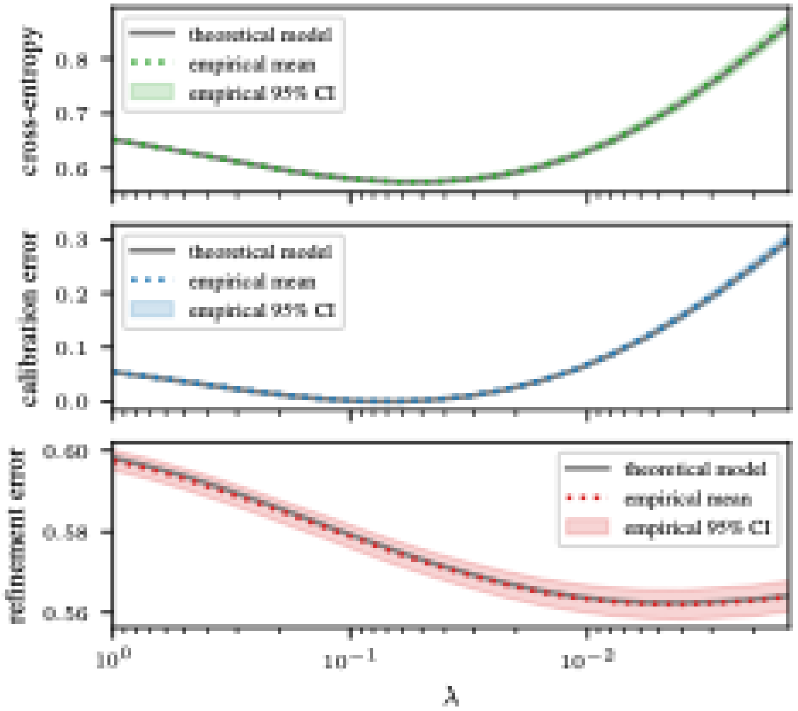
$$\|w_\lambda\|_\Sigma^2 \xrightarrow{P} \mathbb{E}_{\sigma \sim F}\left[\frac{\gamma r \sigma^2 + \eta^2 c^2 \sigma}{(\lambda + \tau\sigma)^2} \right],$$

where the convergence is in probability.

Mai, X., Liao, Z., & Couillet, R. (2019). A large scale analysis of logistic regression: Asymptotic performance and new insights. *International Conference on Acoustics, Speech and Signal Processing*.

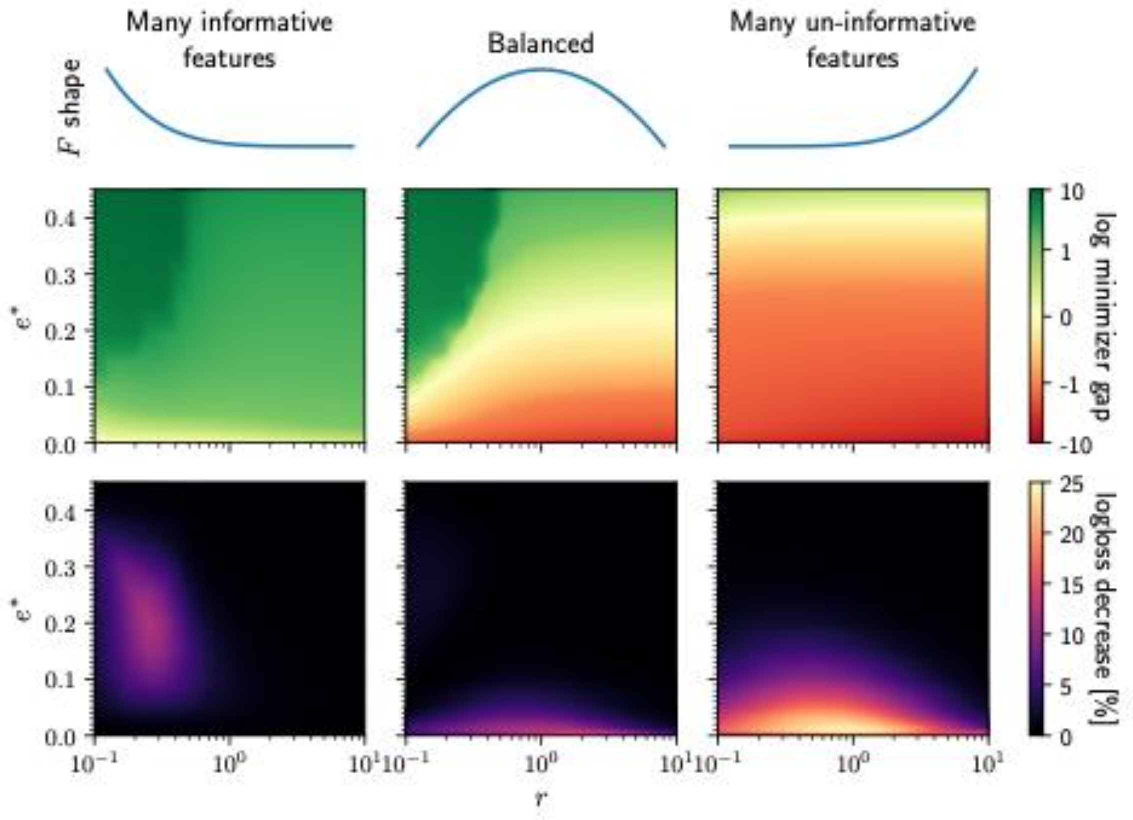
Theoretical analysis: regularized logistic regression in high dimension

We provide an efficient solver for the problem of computing calibration and refinement errors, under our specific mathematical model, see github.com/eugeneberta/RefineThenCalibrate-Theory



Cross-entropy, calibration and refinement errors when λ varies. The spectral distribution F is uniform, e∗ = 10%, r = 1/2 . We fit a logistic regression on 2000 random samples from our data model, we compute the resulting calibration and refinement errors and plot 95% error bars after 50 seeds.

Theoretical analysis: regularized logistic regression in high dimension



Influence of problem parameters on calibration and refinement minimizers. *First row: spectral distribution shape. Second row: log gap between the two minimizers. In green regions, calibration is minimized earlier, while in red regions it is refinement. Third row: relative logloss gain (%) obtained with refinement early stopping.*

github.com/eugeneberta/RefineThenCalibrate-Theory

Conclusion

- New refinement estimator for classification
  - Need a family of functions (monotonic or increasing)
  - Going beyond?
- Selecting the best epoch and hyperparameters based on refinement error
  - Calibrated classifiers with lower loss!
  - Improvements in most examples with little changes
  - Online code
- (simple) theoretical analysis: logistic regression in the high dimensional Gaussian data model
  - Going beyond?

The end, thanks for listening!

Read the full paper:

Use our method on your favorite classification task:

Buy my new book: