Prediction–Powered Active Testing

Anonymous Authors[1]

Abstract

Active testing estimates model risk from a large unlabelled test pool under a limited labelling budget. We propose **Prediction–Powered Active Testing (PPAT)**, which combines the LURE estimator with a control variate built from cheap black–box predictions on the full pool, yielding an unbiased estimator with reduced variance. On real regression benchmarks, PPAT remains unbiased and consistently achieves lower variance than both LURE and random sampling.

1. Introduction

Active learning (Settles, 2009) reduces the cost of training, but in many applications the cost of *evaluation* is just as important. In many domains, large unlabelled test pools are often easy to obtain, while ground–truth labels remain expensive and require expert annotation. Active testing (Kossen et al., 2021; 2022; Berrada et al., 2025) addresses this bottleneck by adaptively sampling which test points to label while correcting for the bias introduced by adaptive sampling. Crucially, adaptive sampling can achieve lower variance than random sampling (Kossen et al., 2021). while remaining unbiased.

In many settings, such as image classification (Radford et al., 2021) and language modelling (Brown et al., 2020; Wang et al., 2021), we also have access to a powerful black–box predictor that can produce cheap predictions on the entire test pool. A natural question here is: *can we leverage these predictions to produce more accurate, label–efficient estimates of our risk?* A naive approach would be to replace the true, unknown labels with the predictions of the black–box model. This, however, generally results in biased estimates of the risk. Motivated by the recent framework of *Prediction–Powered Inference* (Angelopoulos et al., 2023a), we instead treat the predictions of a black–box model as a *control variate* (Glasserman, 2004).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

We propose *Prediction–Powered Active Testing* (PPAT), which combines the unbiased importance–weighted estimator (LURE) of active testing with a control variate built from the proxy loss obtained by replacing true labels with black–box predictions. This yields an estimator of a residualised loss, controlled by a tuning parameter λ that determines how strongly the proxy is used. We show that for suitable λ, PPAT achieves lower variance than the LURE estimator and discuss how λ can be chosen in practise. Specifically, we show that naive online estimation of λ can introduce bias, and therefore consider both simple heuristic choices and an importance–weighted plug–in estimator.

Beyond the estimator itself, PPAT also changes *what* should be sampled. Rather than reusing active testing proposals designed for the raw loss, we derive proposals tailored to the residualised objective induced by λ. This leads to a λ–dependent oracle proposal and a practical surrogate acquisition rule that samples points expected to be most informative for reducing the variance of our estimator. Empirically, on real regression datasets, our approach remains unbiased and consistently achieves lower variance than both LURE and random sampling.

2. Background

Let $f : \mathcal{X} \to \mathcal{Y}$ be a *fixed* model, with inputs $\mathbf{x} \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$, and $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ a non-negative loss function. Given a finite unlabelled test pool $D_{\text{test}} = \{\mathbf{x}_i\}_{i=1}^N$, with $\mathbf{x}_i$ drawn i.i.d from some distribution $\mathbb{P}_{\mathcal{X}}$, *active testing* (Kossen et al., 2021; 2022; Berrada et al., 2025) aims to estimate the true risk,

$$r = \mathbb{E}[\mathcal{L}(f(\mathbf{x}, y)],$$

in a sample–efficient way.

For a labelling budget $M \ll N$, the goal is to label a subset of points $\mathcal{D}_{\text{test}}^{\text{observed}} \subseteq \mathcal{D}_{\text{test}}$ of size M that can be used to estimate r. The quality of the resulting estimator depends on its *bias* and *variance*. Let $\ell_i = \mathcal{L}(f(x_i), y_i)$ and i_m be the index of the point labelled at step m. A naive approach would be to randomly sample points from $\mathcal{D}_{\text{test}}$ without replacement and calculate the subsample empirical risk:

$$\widehat{R}_{\text{iid}} = \frac{1}{M} \sum\nolimits_{i_m \in \mathcal{D}_{\text{test}}^{\text{observed}}} \ell_{i_m} \tag{1}$$

While this estimate is unbiased, its variance is typically high when $M \ll N$ (Kossen et al., 2021; 2022; Berrada et al., 2025).

2.1. Active Testing with LURE

To reduce the variance of the naive estimate $\widehat{R}_{\text{iid}}$, Kossen et al. (2021) *actively select* the most useful points from $\mathcal{D}_{\text{test}}$ and consider an importance–weighted estimate of the risk. We briefly summarise their approach below.

Active sampling. At each acquisition round m, let $S_m \subseteq \{1, \ldots, N\}$ denote the remaining indices. Let $\mathcal{F}_{m-1}$ be the σ-algebra generated by all information available before round m (e.g. D_{test}, past queried indices and labels, surrogate model). Kossen et al. (2021) actively samples indices from the pool according to a proposal

$$q_m(\cdot) := \mathbb{P}(i_m = \cdot \mid \mathcal{F}_{m-1}) \quad \text{supported on } S_m,$$

At each round, the sampled index $i_m \sim q_m$ is queried for its label, y_{i_m}, and its loss, ℓ_{i_m}, is included in the training set $D_{\text{train}} = \{\ell_{i_1}, \ldots, \ell_{i_m}\}$.

The LURE estimator. To estimate the risk, Kossen et al. (2021) use the importance–weighted estimate of (Farquhar et al., 2021) (LURE). Define the LURE importance–weight v_m for point i_m as

$$v_m = 1 + \frac{N-M}{N-m}\left(\frac{1}{(N-m+1)\, q_m(i_m)} - 1\right), \quad (2)$$

then the LURE estimator of the risk is given by:

$$\widehat{R}_{\text{LURE}} = \frac{1}{M}\sum_{m=1}^{M} v_m\, \ell_{i_m}. \quad (3)$$

This is an unbiased estimate of the true risk and has strictly *lower variance* than sampling randomly without replacement under the proposal $q_m^\star(i) \propto \ell_i$ (Farquhar et al., 2021). In practise, the true labels are not known, and so the oracle proposal $q_m^\star(i)$ is approximated with

$$q_m^{\text{AT}}(i) \propto \mathbb{E}_{\pi_m(y|x_i)}[\,\mathcal{L}(f(x_i), y)\,], \quad (4)$$

where $\pi_m(y \mid x)$ is a *surrogate model* used to approximate $p(y \mid x)$ at step m (Kossen et al., 2021).

3. Prediction–Powered Active Testing

We now describe our approach, Prediction–Powered Active Testing (PPAT), which leverages a cheap proxy labeller g to reduce the variance of the LURE estimator in a similar spirit to (Angelopoulos et al., 2023a;b). Assume we can evaluate a cheap proxy labeller $g : \mathcal{X} \to \mathcal{Y}$ on *all* test inputs, yielding proxy labels $\tilde{y}_i = g(\mathbf{x}_i)$. Define the *proxy loss* $\tilde{\ell}_i = L(f(\mathbf{x}_i), \tilde{y}_i)$ and its pool mean $\bar{\ell}_{\text{proxy}} = \frac{1}{N}\sum_{i=1}^{N} \tilde{\ell}_i$. Moreover, define a centred proxy statistic

$$c_i = \tilde{\ell}_i - \bar{\ell}_{\text{proxy}}, \qquad \text{so that} \qquad \frac{1}{N}\sum_{i=1}^{N} c_i = 0. \quad (5)$$

3.1. PPAT Estimator

Fix $\lambda \in \mathbb{R}$, a proposal $q_m(\cdot)$ (which we will later discuss in more detail) and define the *residual loss* $z_i(\lambda) = \ell_i - \lambda c_i$. Our estimator is given by:

$$\widehat{R}_{\text{PPAT}}(\lambda) = \frac{1}{M}\sum_{m=1}^{M} v_m\, z_{i_m}(\lambda), \quad (6)$$

where the weights v_m are defined in (2). This estimator is the LURE estimator but applied to $z_i(\lambda)$ instead of ℓ_i, and thus inherits its unbiasedness.

Proposition 3.1. *(Unbiasedness) Let $r = \mathbb{E}[\mathcal{L}(f(x), y)]$. For any fixed $\lambda \in \mathbb{R}$ we have:*

$$\mathbb{E}\Big[\widehat{R}_{\text{PPAT}}(\lambda)\Big] = r$$

More importantly, it can be shown that this estimator achieves lower variance than $\widehat{R}_{\text{LURE}}$ for a range of λ values.

Proposition 3.2. *(Variance of $\widehat{R}_{\text{PPAT}}$) Let $\widehat{C} = \frac{1}{M}\sum_{m=1}^{M} v_m c_{i_m}$, then $\widehat{R}_{\text{PPAT}}(\lambda) = \widehat{R}_{\text{LURE}} - \lambda\widehat{C}$ and we have*

$$\begin{aligned}\text{Var}\Big(\widehat{R}_{\text{PPAT}}(\lambda)\Big) &= \text{Var}(\widehat{R}_{\text{LURE}}) \\ &\quad - 2\lambda\,\text{Cov}(\widehat{R}_{\text{LURE}}, \widehat{C}) + \lambda^2 \text{Var}(\widehat{C}). \quad (7)\end{aligned}$$

Thus, $\text{Var}\Big(\widehat{R}_{\text{PPAT}}(\lambda)\Big) < \text{Var}(\widehat{R}_{\text{LURE}})$ *whenever* $\lambda \in (\min\{0, 2\lambda^\star\}, \max\{0, 2\lambda^\star\})$*, where*

$$\lambda^\star = \frac{\text{Cov}(\widehat{R}_{\text{LURE}}, \widehat{C})}{\text{Var}(\widehat{C})}. \quad (8)$$

In particular, $\lambda^\star$ minimises $\text{Var}\Big(\widehat{R}_{\text{PPAT}}(\lambda)\Big)$.

Note that under λ^*, the variance of the PPAT estimator can also be written as $\text{Var}(\widehat{R}_{\text{LURE}})(1 - \rho^2_{\widehat{R}_{\text{LURE}},\widehat{C}})$, where $\rho^2_{\widehat{R}_{\text{LURE}},\widehat{C}}$ is the correlation between $\widehat{R}_{\text{LURE}}$ and $\widehat{C}$. Here, a larger correlation results in a larger variance reduction.

3.2. Estimation of λ

In most settings, λ^*, and consequently $(\min\{0, 2\lambda^\star\}, \max\{0, 2\lambda^\star\})$, is unknown a priori. Moreover, unfortunately using an estimate $\widehat{\lambda}$ of $\lambda^\star$ based

on the collected labelled data will bias the PPAT estimator (see Prop. E.3). Nonetheless, it is possible to choose λ *heuristically* without introducing additional bias and still achieve variance reduction. It is also possible to find a *plug–in* estimate of λ based on the optimal control variate coefficient on the pool.

Heuristic specification of λ. A natural default is $\lambda = 1$ – this also mirrors the choice in (Angelopoulos et al., 2023a). This choice leads to variance reduction when the proxy loss is on roughly the same scale as the true loss and is highly correlated with it, such that $\mathrm{Var}(\widehat{C}) \approx \mathrm{Var}(\widehat{R}_{\mathrm{LURE}})$ and $\rho_{\widehat{R}_{\mathrm{LURE}},\widehat{C}} \approx 1$. By contrast, $\lambda = 0.5$ is a more conservative shrinkage choice when the proxy remains informative but is either less strongly correlated with the true loss or somewhat more variable; for example, it is consistent with either $\rho_{\widehat{R}_{\mathrm{LURE}},\widehat{C}} \approx 0.5$ and comparable variances, or $\rho_{\widehat{R}_{\mathrm{LURE}},\widehat{C}} \approx 1$ and $\mathrm{Var}(\widehat{C}) \approx 4\,\mathrm{Var}(\widehat{R}_{\mathrm{LURE}})$.

Plug–in estimate of λ. A direct estimator of $\lambda^\star$ for a given test pool is not practical as it requires running multiple active testing trials to estimate sample covariances under a given proposal. A more tractable, alternative target is the finite–pool control–variate coefficient given by:

$$\lambda^\dagger = \arg\min_{\lambda\in\mathbb{R}} \frac{1}{N}\sum_{i=1}^{N}(\ell_i - \lambda c_i)^2 = \frac{\frac{1}{N}\sum_{i=1}^{N}\ell_i c_i}{\frac{1}{N}\sum_{i=1}^{N} c_i^2}, \quad (9)$$

A natural plug–in estimator for this is obtained by replacing the pool moments with their importance–weighted estimates:

$$\hat{\mu}_{\ell c} = \frac{1}{M}\sum_{m=1}^{M} v_m \ell_{i_m} c_{i_m}, \quad (10)$$

$$\hat{\mu}_{c^2,M} = \frac{1}{M}\sum_{m=1}^{M} v_m c_{i_m}^2, \quad (11)$$

$$\hat{\lambda}_M = \frac{\hat{\mu}_{\ell c,M}}{\hat{\mu}_{c^2,M}}. \quad (12)$$

See Appendix E for further discussion.

3.3. Sampling Proposals

All of the previous results hold for any given $\mathcal{F}_{m-1}$–measurable proposal $q_m(\cdot)$ supported on S_m. In practise, we would like to choose q_m to minimise $\mathrm{Var}(\widehat{R}_{\mathrm{PPAT}}(\lambda))$. As the exact finite–horizon variance–minimising proposal is generally intractable, we instead consider a myopic proxy based on the residualised losses (see Appendix C for further details).

Oracle proposal for residualised losses. At round m, a natural oracle objective is to choose q_m so as to minimise the importance–sampling proxy

$$\sum_{i\in S_m} \frac{z_i(\lambda)^2}{q_m(i)} \quad (13)$$

subject to

$$\sum_{i\in S_m} q_m(i) = 1, \quad q_m(i) > 0.$$

A standard Lagrange multiplier argument yields

$$q^\star_{m,\lambda}(i) = \frac{|z_i(\lambda)|}{\sum_{j\in S_m}|z_j(\lambda)|}, \qquad i \in S_m. \quad (14)$$

This oracle proposal prioritises points whose residualised loss is large in magnitude. Intuitively, these are points for which the true loss ℓ_i is expected to differ most strongly from the centred proxy term λc_i. This proposal is, however, not available in practice since it depends on the unknown labels through ℓ_i.

Practical surrogate proposal. To obtain a tractable acquisition rule, suppose that at round m we have a surrogate predictive distribution $\pi_m(y \mid x)$ for the unknown label. We then define the score

$$s_{m,i}(\lambda) = \left(\mathbb{E}_{\pi_m(\cdot|x_i)}\left[\left(\mathcal{L}(f(\mathbf{x}_i), y) - \lambda c_i\right)^2\right]\right), \quad (15)$$

and set

$$q_m^{\mathrm{PPAT}}(i) \propto s_{m,i}(\lambda), \qquad i \in S_m, \quad (16)$$

followed by normalisation over S_m. See Appendix C.3 for further details.

4. Experiments

We evaluate **PPAT** on four real regression datasets, where we compare with **LURE** and **Random** sampling. We describe our experimental setup and results briefly below with full details in Appendix B.

Experimental setup. We use the following UCI regression datasets (Dheeru & Karra Taniskidou, 2017): `airfoil`, `concrete`, `parkinsons`, `pumadyn32nm`. We split each dataset into two halves, where the first half is used to train the model f that we want to evaluate, and the second half is used to sample $n_{\text{pool}} = 100$ points for our test pool. For f, we fit a Gaussian process (GP) model with a RBF kernel.

Both **LURE** and **PPAT** require a surrogate model $\pi_m(y \mid \mathbf{x})$ for their sampling proposals; we use a Bayesian ridge regression model for this surrogate, where we update the surrogate after every acquired test point and initialise it with the same data used to train f. For their sampling

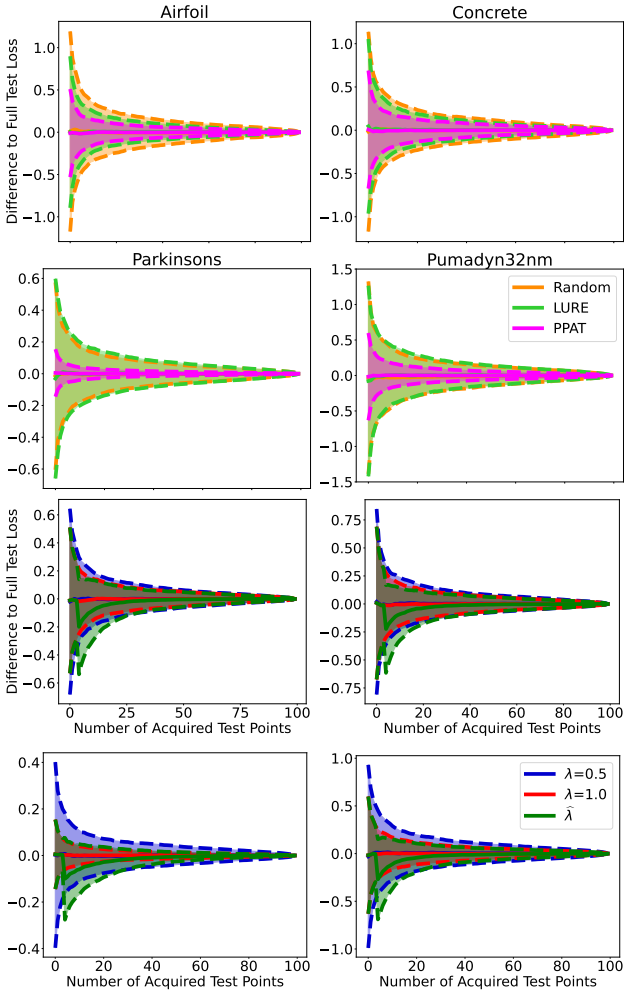

Figure 1. Results on four UCI regression datasets. Top row compares PPAT with Random and LURE. Bottom rows shows **PPAT** with $\lambda \in \{0.5, 1.0, \widehat{\lambda}\}$. All plots show the mean difference to the full test loss on the pool $\pm$ standard deviation over 1000 trials.

proposals, we use (4) and (16) respectively. For the proxy model g, we use the TabPFN–2.5 (Grinsztajn et al., 2025) regression foundation model which has been trained purely on synthetic data, thus avoiding any train–test leakage.

For **PPAT** we use a default of $\lambda = 1$ and also compare with the different choices of λ described in Section 3.2. For our plug–in estimator $\widehat{\lambda}$, we only use $\widehat{\lambda}$ during *evaluation* and fix $\lambda = 1$ for the sampling proposal. Moreover, to avoid the large errors in the estimation of λ that can occur at earlier steps, we only use the plug–in estimator after step $m = 5$. All experiments are run for 1000 independent trials on a NVIDIA GeForce RTX 2080 Ti with 12GB of vRAM.

Results. From Figure 1, we see that **PPAT** with the default choice of $\lambda = 1$ achieves lower variance than both **LURE** and **Random** across all the tested regression datasets, while also remaining unbiased as expected. We also see that both $\lambda = 1.0, 0.5$ serve as sensible default choices, with $\lambda = 1.0$ generally achieving the lowest variance. For the plug–in estimate $\widehat{\lambda}$, we find that it has variance comparable with $\lambda = 1.0$, with its bias decreasing as m increases. Importantly, all these choices of λ outperform both **LURE** and **Random** as can be seen from Figure 2.

Together, these findings demonstrate the effectiveness of **PPAT** for active testing. Moreover, they further support the choices of λ discussed in Section 3.2.

References

Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I., and Zrnic, T. Prediction-powered inference. *Science*, 382 (6671):669–674, 2023a.

Angelopoulos, A. N., Duchi, J. C., and Zrnic, T. Ppi++: Efficient prediction-powered inference. *arXiv preprint arXiv:2311.01453*, 2023b.

Berrada, G., Kossen, J., Smith, F. B., Razzak, M., Gal, Y., and Rainforth, T. Scaling up active testing to large language models. *arXiv preprint arXiv:2508.09093*, 2025.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Farquhar, S., Gal, Y., and Rainforth, T. On statistical bias in active learning: How and when to fix it. *arXiv preprint arXiv:2101.11665*, 2021.

Fisch, A., Maynez, J., Hofer, R., Dhingra, B., Globerson, A., and Cohen, W. W. Stratified prediction-powered inference for effective hybrid evaluation of language models. *Advances in Neural Information Processing Systems*, 37: 111489–111514, 2024.

Glasserman, P. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer, New York, 2004.

Grinsztajn, L., Flöge, K., Key, O., Birkel, F., Jund, P., Roof, B., Jäger, B., Safaric, D., Alessi, S., Hayler, A., et al.

Tabpfn-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Huber, P. J. *Robust Statistics*, pp. 1248–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_594. URL https://doi.org/10.1007/978-3-642-04898-2_594.

Kossen, J., Farquhar, S., Gal, Y., and Rainforth, T. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, pp. 5753–5763. PMLR, 2021.

Kossen, J., Farquhar, S., Gal, Y., and Rainforth, T. Active surrogate estimators: An active learning approach to label-efficient model evaluation. *Advances in Neural Information Processing Systems*, 35:24557–24570, 2022.

Owen, A. B. *Monte Carlo theory, methods and examples*. 2013. URL http://statweb.stanford.edu/~owen/mc/.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Settles, B. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

Wang, S., Liu, Y., Xu, Y., Zhu, C., and Zeng, M. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4195–4205, 2021.

Zrnic, T. and Candès, E. J. Active statistical inference. *arXiv preprint arXiv:2403.03208*, 2024a.

Zrnic, T. and Candès, E. J. Cross-prediction-powered inference. *Proceedings of the National Academy of Sciences*, 121(15):e2322083121, 2024b.

A. Related Work

Active Testing Kossen et al. (2021) introduced active testing by framing label-efficient model evaluation as an adaptive sampling problem. Their approach combines task-specific acquisition rules with the unbiased LURE estimator developed by Farquhar et al. (2021). Unlike active learning, the acquisition strategy in active testing should be tailored to reducing the variance of the evaluation estimator, which leads to proposals based on expected loss. Kossen et al. (2022) subsequently proposed Active Surrogate Estimators (ASE), which also exploit predictions over the full pool similar to our approach, but in a fundamentally different way: instead of correcting an importance–weighted estimator with observed labels, ASE learns a surrogate for the conditional loss and then imputes losses across the pool via an interpolation–based estimate. This can be label-efficient empirically, but it does not retain the unbiasedness of LURE, since surrogate misspecification or finite–sample estimation error enters the final risk estimate directly. By contrast, our method keeps the unbiased active-testing estimator and uses proxy predictions on the pool only as a control variate, so that black–box predictions reduce variance without introducing any additional bias. More recently, Berrada et al. (2025) showed how to scale active testing with LURE to LLM evaluation. There, they consider using fixed surrogates built via in–context learning, where they allow the surrogate to be smaller than the target model and to be used for acquisition without target–model predictions. Our approach is complementary: it can incorporate the same in–context learning surrogates, but additionally leverages cheap predictions from a black–box model to improve the estimator itself.

Relatedly, Zrnic & Candès (2024a) consider adaptive label sampling for statistical inference. They propose both batch and sequential methods that provide valid confidence intervals for M–estimation problems. Their framework could in principle be specialised to risk estimation (as risk estimation is a particular M–estimation problem), but it is not framed around model evaluation. Moreover, their sequential formulation is closer to online selective labelling than to the pool–based acquisition rules considered in active testing: labels are acquired point–by–point based on past information, rather than by selecting among the full remaining pool. This is well suited to their goal of asymptotically valid inference, but differs from our setting, where we focus on pool–based risk estimation and use black–box predictions as control variates within an unbiased active–testing estimator.

Prediction–Powered Inference Prediction–powered inference (PPI, (Angelopoulos et al., 2023a)) was originally introduced with the goal of finding tighter confidence intervals in the semi–supervised setting by leveraging machine learning predictions. PPI has since been extended in several directions. PPI++ (Angelopoulos et al., 2023b) proposes a more computationally efficient, loss–based formulation together with a power–tuning parameter λ. Stratified PPI (Fisch et al., 2024) improves efficiency via data stratification, while Cross–PPI (Zrnic & Candès, 2024b) shows how the predictor can be trained within the same prediction–powered pipeline. Our work is closely related in spirit, but differs in both objective and setting: rather than semi–supervised inference with i.i.d. labelled and unlabelled data, we study active risk estimation over a finite test pool, using black-box predictions as a control variate inside an importance–weighted active testing estimator.

B. Experimental Setup

Here, we describe the experimental setup for the experiments in Section 4 in more detail.

Datasets. For our experiments, we used the following UCI regression datasets (Dheeru & Karra Taniskidou, 2017):

- **Airfoil**: A NASA aerodynamic dataset for predicting scaled sound pressure level from airfoil operating conditions. It contains 1,503 instances and 5 input features.
- **Concrete**: A materials dataset for predicting the compressive strength of concrete from its ingredients and age. It contains 1,030 instances and 8 input features.
- **Parkinsons**: A biomedical voice dataset consisting of 5,875 recordings from 42 patients with early-stage Parkinson's disease. The UCI version contains 19 features, with UPDRS scores as regression targets.
- **Pumadyn32nm:**: a synthetic robot-arm dynamics benchmark in which the goal is to predict the angular acceleration of a robot arm link from its state variables. It contains 8,192 instances and 32 input features.

For each dataset, we standardised the features and outputs and split it into two halves, with one half being used to train the model f that we want to evaluate and the other half being used to sample our pool.

Active testing. We sampled a pool of $n_{\text{pool}} = 100$ points for each experiment and used a budget of $M = 100$. For **LURE** and **PPAT** we used the sampling proposals described in Appendix C.3. Note that the proposal for **PPAT** further simplifies when using a Bayesian ridge regression model (with Gaussian priors) for the surrogate since the predictive distribution is Gaussian.

For **PPAT** we use a default of $\lambda = 1$ and also compare with the different choices of λ described in Section 3.2. For our plug–in estimator $\widehat{\lambda}$, we only use $\widehat{\lambda}$ during *evaluation* and fix $\lambda = 1$ for the sampling proposal. Moreover, to avoid the large errors in the estimation of λ that can occur at earlier steps, we only use the plug–in estimator after step $m = 5$.

Models. For f, we fit a Gaussian process (GP) model with a RBF kernel, where we optimised the kernel hyperparameters using the marginal likelihood. For the surrogate model $\pi_m(y \mid x)$, we used a Bayesian ridge regression model with the default hyperparameters from scikit-learn (Buitinck et al., 2013). The surrogate was updated after every acquired test point and initialised with the same data used to train f.

For our black–box model g, we used TabPFN–2.5 (Grinsztajn et al., 2025). To obtain predictions on our pool, we used in–context learning, with the data used to train f serving as our context data.

Metrics. At each step m of the active testing, we report the mean difference ± 1 standard deviation between the active risk estimate and the empirical test risk.

Hardware. All experiments were run for 1000 independent trials on a NVIDIA GeForce RTX 2080 Ti with 12GB of vRAM.

C. Additional Details

C.1. Prediction–Powered Inference

Here, we give a brief overview of prediction–powered inference (Angelopoulos et al., 2023a)

Prediction–powered inference (PPI, (Angelopoulos et al., 2023a)) studies statistical inference when one has a small labelled sample together with a much larger unlabelled sample on which a black–box predictor $g : \mathcal{X} \to \mathcal{Y}$ can be evaluated. Suppose we have labelled data $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$ and unlabelled covariates $\mathcal{D}_u = \{\widetilde{\mathbf{x}}_j\}_{j=1}^N$, with $(\mathbf{x}_i, y_i)$ and $\widetilde{\mathbf{x}}_j$ drawn i.i.d. from $\mathbb{P}_{\mathcal{X}\times\mathcal{Y}}$ and $\mathbb{P}_{\mathcal{X}}$ respectively, and $K \ll N$. In the simplest mean–estimation setting, the goal is to estimate the population mean, $\theta^\star = \mathbb{E}[y]$, using the labels from $\mathcal{D}_l$ and the predictions of g on $\mathcal{D}_u$.

For mean estimation, the key quantity is the *rectifier* $\Delta = \mathbb{E}[g(\mathbf{x}) - y]$. Indeed, noting that population mean can be written as $\theta^\star = \mathbb{E}[g(x)] - \Delta$, PPI provides the following estimate of $\theta^\star$:

$$\widehat{\theta}_{\text{PP}} = \underbrace{\frac{1}{N}\sum_{j=1}^{N} g(\widetilde{\mathbf{x}}_j)}_{\tilde{\theta}_g} - \underbrace{\frac{1}{K}\sum_{i=1}^{K}\big(g(\mathbf{x}_i) - y_i\big)}_{\hat{\Delta}} \tag{17}$$

$$= \frac{1}{K}\sum_{i=1}^{K} y_i - \left(\frac{1}{K}\sum_{i=1}^{K} g(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N} g(\widetilde{\mathbf{x}}_j)\right) \tag{18}$$

This estimator is unbiased and, crucially, when the predictions are accurate and $\hat{\Delta} \approx 0$, we have $\widehat{\theta}_{\text{PP}} \approx \tilde{\theta}_g$, which has much lower variance than the classical empirical estimate of the mean based on $\mathcal{D}_l$. Moreover, from the equivalent form (18), we can see that this estimator treats the predictions of g as a *control variate* (Glasserman, 2004).

PPI++ (Angelopoulos et al., 2023b) extends this estimator by including a *power tuning parameter* $\lambda \in \mathbb{R}$ to control the effect of the predictions. Concretely, they propose the following estimator

$$\widehat{\theta}_{\text{PP++}} = \frac{1}{K}\sum_{i=1}^{K} y_i - \lambda\left(\frac{1}{K}\sum_{i=1}^{K} g(\mathbf{x}_i) - \frac{1}{N}\sum_{j=1}^{N} g(\widetilde{\mathbf{x}}_j)\right) \tag{19}$$

The variance of this estimator is given by

$$\mathrm{Var}\Big(\widehat{\theta}_{\mathrm{PP++}}\Big) = \frac{1}{K}\mathrm{Var}(y) - \frac{2\lambda}{K}\mathrm{Cov}\big(y, g(\mathbf{x})\big) + \lambda^2\left(\frac{1}{K} + \frac{1}{N}\right)\mathrm{Var}\big(g(\mathbf{x})\big), \tag{20}$$

which has strictly lower variance than the empirical estimate for $\lambda \in (\min\{0, 2\lambda^\star\}, \max\{0, 2\lambda^\star\})$, where $\lambda^\star = \mathrm{Cov}\big(y, g(\mathbf{x})\big)/\big(1 + \frac{K}{N}\big)\,\mathrm{Var}\big(g(\mathbf{x})\big)$. Under λ^*, the variance can be written as

$$\mathrm{Var}\Big(\widehat{\theta}^{\lambda^\star}_{\mathrm{PP++}}\Big) = \frac{1}{K}\mathrm{Var}(y)\left(1 - \frac{\rho^2_{y,g}}{1 + \frac{K}{N}}\right), \tag{21}$$

where $\rho_{y,g}$ is the correlation between the true labels y and the predictions $g(\mathbf{x})$. Thus, we see that the more correlated the predictions are with the true labels, the larger the variance reduction.

Although the above discussion is stated for mean estimation, the same principle extends more broadly for M-estimation problems (Huber, 2011; Angelopoulos et al., 2023a;b).

C.2. Oracle Sampling Proposal for PPAT

Here, we provide further details on the derivation of the oracle proposal described in Section 3.3. Condition on $\mathcal{F}_{m-1}$ and let S_m denote the remaining pool at round m. If we were to estimate the mean on the remaining–pool,

$$\bar{z}_{S_m}(\lambda) = \frac{1}{|S_m|}\sum_{i \in S_m} z_i(\lambda),$$

from a single draw $i_m \sim q_m$, the standard unbiased importance–sampling (Owen, 2013) estimator would be

$$\widehat{z}_m(q) = \frac{z_{i_m}(\lambda)}{|S_m|\, q_m(i_m)}.$$

Its conditional expectation is

$$\mathbb{E}[\widehat{z}_m(q) \mid \mathcal{F}_{m-1}] = \frac{1}{|S_m|}\sum_{i \in S_m} z_i(\lambda) = \bar{z}_{S_m}(\lambda),$$

and its conditional variance is

$$\mathrm{Var}(\widehat{z}_m(q) \mid \mathcal{F}_{m-1}) = \frac{1}{|S_m|^2}\left(\sum_{i \in S_m}\frac{z_i(\lambda)^2}{q_m(i)} - \Big(\sum_{i \in S_m} z_i(\lambda)\Big)^2\right).$$

Since the second term does not depend on q_m, minimizing this variance over possible proposals q_m is equivalent to solving

$$\min_{q_m} \quad \sum_{i \in S_m}\frac{z_i(\lambda)^2}{q_m(i)} \qquad \text{s.t.} \qquad \sum_{i \in S_m} q_m(i) = 1, \;\; q_m(i) > 0,$$

which is precisely the objective in (13). A standard Lagrange multiplier argument then gives

$$q^\star_{m,\lambda}(i) \propto |z_i(\lambda)|, \qquad i \in S_m.$$

The absolute value appears because the variance depends on $z_i(\lambda)^2$: for acquisition, only the magnitude of the residualised loss matters, not its sign. Intuitively, points with large $|z_i(\lambda)|$ are precisely those for which the true loss is poorly explained by the proxy term λc_i, and hence are the points that contribute most to the variance of the residualised estimator. By contrast, if λc_i already explains most of ℓ_i, then $z_i(\lambda)$ is small and there is less to be gained by querying that point.

Note that (13) is a myopic proxy[1] rather than the exact finite–horizon variance of the full LURE estimator. The latter is generally difficult to compute as sampling is done sequentially without replacement.

[1] More specifically, it is the exact one–step variance objective for unbiased importance sampling of the residualised loss.

C.3. Simplification of Sampling Proposals for the MSE Loss

Here, we describe how the sampling proposals for LURE (4) and PPAT (15) can be further simplified for the MSE loss. We also discuss the expected behaviour of the PPAT and LURE proposals.

LURE. For the MSE loss,

$$\mathcal{L}(f(\mathbf{x}_i), y) = (f(\mathbf{x}_i) - y)^2.$$

Let

$$f_i := f(\mathbf{x}_i), \qquad \mu_{m,i} := \mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}[y], \qquad \sigma^2_{m,i} := \mathrm{Var}_{\pi_m(\cdot|\mathbf{x}_i)}(y).$$

Then the LURE proposal in (4) simplifies as

$$q_m^{\mathrm{AT}}(i) \propto \mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}\left[(f_i - y)^2\right] = \mathrm{Var}_{\pi_m(\cdot|\mathbf{x}_i)}(y) + \Big(\mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}[y] - f_i\Big)^2 = \sigma^2_{m,i} + (\mu_{m,i} - f_i)^2.$$

Thus, under the MSE loss, the LURE proposal has the usual bias–variance decomposition form: it favours points for which either the surrogate is uncertain about the label, or the surrogate mean disagrees strongly with the prediction of the model being evaluated.

PPAT. For PPAT, recall that

$$c_i = \tilde{\ell}_i - \bar{\ell}_{\mathrm{proxy}}, \qquad \tilde{\ell}_i = (f_i - g(\mathbf{x}_i))^2.$$

Moreover, note that the proposal (15) admits the decomposition

$$\mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}\Big[\big(\mathcal{L}(f(\mathbf{x}_i), y) - \lambda c_i\big)^2\Big] = \mathrm{Var}_{\pi_m(\cdot|\mathbf{x}_i)}(\mathcal{L}(f(\mathbf{x}_i), y)) + \big(\mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}[\mathcal{L}(f(\mathbf{x}_i), y)] - \lambda c_i\big)^2. \tag{22}$$

This shows that the PPAT proposal favours points for which either the surrogate is uncertain about the loss, or the surrogate mean loss is far from the proxy term λc_i. In this sense, the acquisition rule targets points that are expected to be most informative for estimating the residualised quantity $z_i(\lambda)$, and therefore for reducing the variance of $\widehat{R}_{\mathrm{PPAT}}(\lambda)$.

Under the MSE loss, (22) becomes

$$\mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}\Big[\big((f_i - y)^2 - \lambda c_i\big)^2\Big] = \mathrm{Var}_{\pi_m(\cdot|\mathbf{x}_i)}\big((f_i - y)^2\big) + \big(\mathbb{E}_{\pi_m(\cdot|\mathbf{x}_i)}[(f_i - y)^2] - \lambda c_i\big)^2.$$

Using the standard bias–variance decomposition, the second term simplifies to

$$\big(\sigma^2_{m,i} + (\mu_{m,i} - f_i)^2 - \lambda c_i\big)^2.$$

To simplify the first term, let

$$a_{m,i} = f_i - \mu_{m,i}, \qquad z = y - \mu_{m,i},$$

so that $\mathbb{E}[z] = 0$, $\mathrm{Var}(z) = \sigma^2_{m,i}$, and

$$(f_i - y)^2 = (a_{m,i} - z)^2.$$

Hence we have

$$\mathrm{Var}\big((f_i - y)^2\big) = \mathrm{Var}(z^2 - 2a_{m,i}z) = \big(\mathbb{E}[z^4] - \sigma^4_{m,i}\big) + 4a^2_{m,i}\sigma^2_{m,i} - 4a_{m,i}\mathbb{E}[z^3].$$

Now, assuming the surrogate predictive distribution is Gaussian[2], i.e. $y \mid \mathbf{x}_i \sim \mathcal{N}(\mu_{m,i}, \sigma^2_{m,i})$, then $\mathbb{E}[z^3] = 0$ and $\mathbb{E}[z^4] = 3\sigma^4_{m,i}$, giving

$$\mathrm{Var}_{\pi_m(\cdot|\mathbf{x}_i)}\big((f_i - y)^2\big) = 2\sigma^4_{m,i} + 4\sigma^2_{m,i}(\mu_{m,i} - f_i)^2.$$

Therefore, under the MSE loss and a Gaussian surrogate, the PPAT proposal simplifies to

$$q_m^{\mathrm{PPAT}}(i) \propto 2\sigma^4_{m,i} + 4\sigma^2_{m,i}(\mu_{m,i} - f_i)^2 + \big(\sigma^2_{m,i} + (\mu_{m,i} - f_i)^2 - \lambda c_i\big)^2.$$

[2]This is the case for our experiments where we use a Bayesian ridge regression model with Gaussian priors as the surrogate.

C.4. Plug–in Estimate of λ

A direct consistent estimator of the exact finite–sample $\lambda^\star$ in (24) is generally not possible, since

$$\lambda^\star = \frac{\mathrm{Cov}(\hat{R}_{\mathrm{LURE}}, \hat{C})}{\mathrm{Var}(\hat{C})}$$

depends on the sample covariance of the final random pair $(\hat{R}_{\mathrm{LURE}}, \hat{C})$, which we cannot estimate unless we are able to run the active testing procedure multiple times. A natural, tractable alternative is the optimal control variate parameter $\lambda^\dagger$ on the entire pool:

$$\lambda^\dagger = \arg\min_{\lambda \in \mathbb{R}} \frac{1}{N} \sum_{i=1}^{N} (\ell_i - \lambda c_i)^2 = \frac{\frac{1}{N}\sum_{i=1}^{N} \ell_i c_i}{\frac{1}{N}\sum_{i=1}^{N} c_i^2}.$$

Since $\frac{1}{N}\sum_{i=1}^{N} c_i = 0$, this is equivalently

$$\lambda^\dagger = \frac{\widehat{\mathrm{Cov}}(\ell, c)}{\widehat{\mathrm{Var}}(c)},$$

where $\widehat{\mathrm{Cov}}(\ell, c)$ and $\widehat{\mathrm{Var}}(c)$ are the sample covariance and variance on the pool respectively. A plug–in estimator can be obtained by replacing the pool moments with their LURE estimates:

$$\hat{\mu}_{\ell c, M} = \frac{1}{M} \sum_{m=1}^{M} v_m \ell_{i_m} c_{i_m}, \qquad \hat{\mu}_{c^2, M} = \frac{1}{M} \sum_{m=1}^{M} v_m c_{i_m}^2, \qquad \hat{\lambda}_M = \frac{\hat{\mu}_{\ell c, M}}{\hat{\mu}_{c^2, M}}.$$

By the properties of LURE we have:

$$\mathbb{E}[\hat{\mu}_{\ell c, M} \mid D_{\mathrm{test}}] = \frac{1}{N} \sum_{i=1}^{N} \ell_i c_i, \qquad \mathbb{E}[\hat{\mu}_{c^2, M} \mid D_{\mathrm{test}}] = \frac{1}{N} \sum_{i=1}^{N} c_i^2.$$

It is possible to show, under regularity conditions which these LURE estimates are mean–square consistent (e.g. bounded second moments and $\frac{1}{N}\sum_i c_i^2$ bounded away from 0), that

$$\hat{\mu}_{\ell c, M} \xrightarrow{p} \frac{1}{N} \sum_{i=1}^{N} \ell_i c_i, \qquad \hat{\mu}_{c^2, M} \xrightarrow{p} \frac{1}{N} \sum_{i=1}^{N} c_i^2,$$

and therefore, by Slutsky's theorem,

$$\hat{\lambda}_M \xrightarrow{p} \lambda^\dagger.$$

D. Additional Results

D.1. Comparing PPAT with Different Values of λ

Here, we repeat the plot in the first row of Figure 1 but for $\lambda = 0.5$ and $\lambda = \widehat{\lambda}$, where $\widehat{\lambda}$ is computed as in Section 3.2.

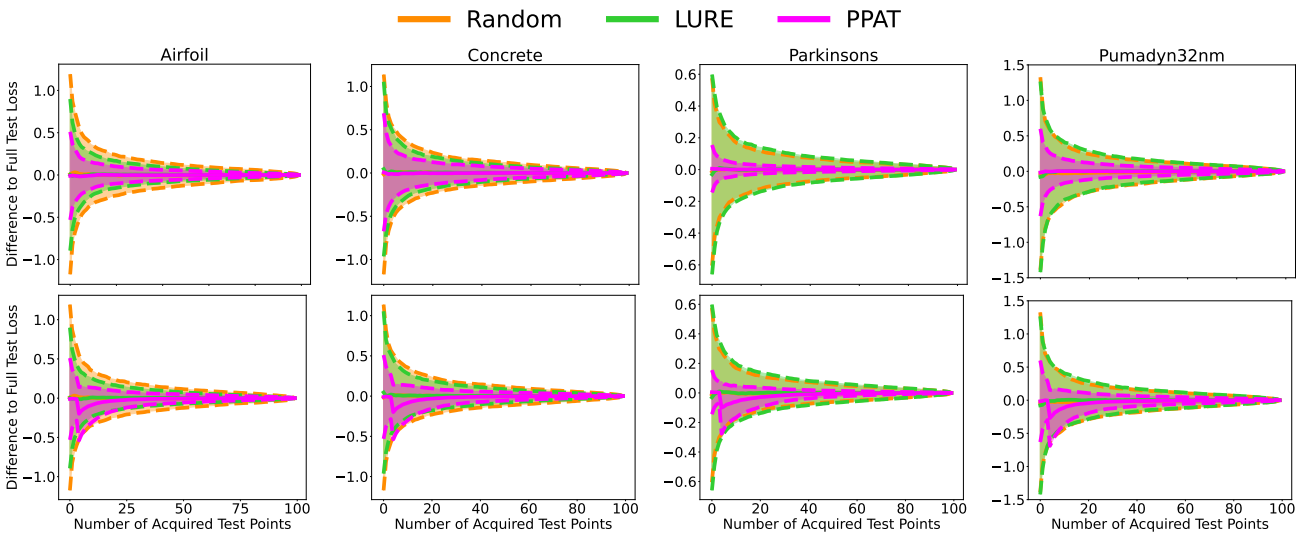

Figure 2. Results on four UCI regression datasets. Top row compares PPAT with Random and LURE, where for PPAT we use $\lambda = 0.5$. Bottom row shows the same but with $\lambda = \widehat{\lambda}$. All plots show the mean difference to the full test loss on the pool $\pm$ standard deviation over 1000 trials.

E. Proofs

Proposition E.1. *(Unbiasedness) Let $r = \mathbb{E}[\mathcal{L}(f(x), y)]$. For any fixed $\lambda \in \mathbb{R}$ we have:*

$$\mathbb{E}\Big[\widehat{R}_{\mathrm{PPAT}}(\lambda)\Big] = r$$

Proof. Fix $\lambda \in \mathbb{R}$ and define $z_i(\lambda) = \ell_i - \lambda c_i, i = 1, \ldots, N$. By definition,

$$\widehat{R}_{\mathrm{PPAT}}(\lambda) = \frac{1}{M} \sum_{m=1}^{M} v_m z_{i_m}(\lambda).$$

That is, $\widehat{R}_{\mathrm{PPAT}}(\lambda)$ is exactly the LURE estimator applied to

$$\{z_1(\lambda), \ldots, z_N(\lambda)\}.$$

By the unbiasedness of LURE we have

$$\mathbb{E}\Big[\widehat{R}_{\mathrm{PPAT}}(\lambda) \,\Big|\, \mathcal{D}_{\text{test}}\Big] = \frac{1}{N} \sum_{i=1}^{N} z_i(\lambda).$$

Now

$$\frac{1}{N} \sum_{i=1}^{N} z_i(\lambda) = \frac{1}{N} \sum_{i=1}^{N} (\ell_i - \lambda c_i) = \frac{1}{N} \sum_{i=1}^{N} \ell_i - \lambda \frac{1}{N} \sum_{i=1}^{N} c_i.$$

Since $c_i = \tilde{\ell}_i - \bar{\ell}_{\text{proxy}}$, we have

$$\frac{1}{N} \sum_{i=1}^{N} c_i = 0,$$

and therefore

$$\frac{1}{N} \sum_{i=1}^{N} z_i(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \ell_i.$$

Hence

$$\mathbb{E}\Big[\widehat{R}_{\mathrm{PPAT}}(\lambda)\,\Big|\mathcal{D}_{\text{test}}\Big] = \frac{1}{N}\sum_{i=1}^{N}\ell_i.$$

Taking expectations over the random draw of the pool and using the tower law then gives us the desired result:

$$\mathbb{E}\Big[\widehat{R}_{\mathrm{PPAT}}(\lambda)\Big] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\ell_i\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\ell_i] = r.$$

□

Proposition E.2. *(Variance of $\widehat{R}_{\mathrm{ppat}}$) Let $\widehat{C} = \frac{1}{M}\sum_{m=1}^{M} v_m c_{i_m}$, then $\widehat{R}_{\mathrm{PPAT}}(\lambda) = \widehat{R}_{\mathrm{PPAT}} - \lambda\widehat{C}$ and we have*

$$\mathrm{Var}\Big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\Big) = \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) - 2\lambda\,\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C}) + \lambda^2\mathrm{Var}(\widehat{C}). \tag{23}$$

Thus, $\mathrm{Var}\Big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\Big) < \mathrm{Var}(\widehat{R}_{\mathrm{LURE}})$ *whenever* $\lambda \in (\min\{0, 2\lambda^\star\}, \max\{0, 2\lambda^\star\})$*, where*

$$\lambda^\star = \frac{\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C})}{\mathrm{Var}(\widehat{C})}. \tag{24}$$

In particular, $\lambda^\star$ minimises $\mathrm{Var}\Big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\Big)$*, with the variance in this case being*

$$\mathrm{Var}\Big(\widehat{R}_{\mathrm{PPAT}}(\lambda^\star)\Big) = \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) - \frac{\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C})^2}{\mathrm{Var}(\widehat{C})} \;\le\; \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}), \tag{25}$$

with strict inequality whenever $\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C}) \neq 0$.

Proof. Let $\widehat{C} := \frac{1}{M}\sum_{m=1}^{M} v_m c_{i_m}$ Then, by linearity of the estimator,

$$\widehat{R}_{\mathrm{PPAT}}(\lambda) = \frac{1}{M}\sum_{m=1}^{M} v_m(\ell_{i_m} - \lambda c_{i_m}) = \frac{1}{M}\sum_{m=1}^{M} v_m\ell_{i_m} - \lambda\frac{1}{M}\sum_{m=1}^{M} v_m c_{i_m} = \widehat{R}_{\mathrm{LURE}} - \lambda\widehat{C}.$$

Therefore,

$$\begin{aligned}\mathrm{Var}\big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\big) &= \mathrm{Var}\big(\widehat{R}_{\mathrm{LURE}} - \lambda\widehat{C}\big)\\ &= \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) - 2\lambda\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C}) + \lambda^2\mathrm{Var}(\widehat{C}),\end{aligned}$$

Assume now that $\mathrm{Var}(\widehat{C}) > 0$, and define $\lambda^\star = \mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C})/\mathrm{Var}(\widehat{C})$. Rearranging the previous expression then gives

$$\mathrm{Var}\big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\big) - \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) = \mathrm{Var}(\widehat{C})\,\lambda(\lambda - 2\lambda^\star).$$

Since $\mathrm{Var}(\widehat{C}) > 0$, this is negative iff

$$\lambda(\lambda - 2\lambda^\star) < 0,$$

which holds exactly when

$$\lambda \in \big(\min\{0, 2\lambda^\star\},\, \max\{0, 2\lambda^\star\}\big).$$

This proves the range of λ for which PPAT has lower variance than LURE. To find the $\lambda^\star$ that achieves the minimum possible variance, we first note that:

$$\begin{aligned}\mathrm{Var}\big(\widehat{R}_{\mathrm{PPAT}}(\lambda)\big) &= \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) - 2\lambda\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C}) + \lambda^2\mathrm{Var}(\widehat{C})\\ &= \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) + \mathrm{Var}(\widehat{C})\big(\lambda - \lambda^\star\big)^2 - \frac{\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}},\widehat{C})^2}{\mathrm{Var}(\widehat{C})}.\end{aligned}$$

Clearly, the variance is minimised at $\lambda = \lambda^\star$, with minimum value

$$\mathrm{Var}\big(\widehat{R}_{\mathrm{PPAT}}(\lambda^\star)\big) = \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}) - \frac{\mathrm{Cov}(\widehat{R}_{\mathrm{LURE}}, \widehat{C})^2}{\mathrm{Var}(\widehat{C})} \leq \mathrm{Var}(\widehat{R}_{\mathrm{LURE}}).$$

The inequality is strict wheneverCov$(\widehat{R}_{\mathrm{LURE}}, \widehat{C}) \neq 0$. □

Proposition E.3. *(Bias from estimating λ) Let $r = \mathbb{E}[\mathcal{L}(f(x), y)]$ and let $\widehat{\lambda}$ be an $\mathcal{F}_M$–measurable estimate of $\lambda^\star$ Then we have that*

$$\mathbb{E}[\widehat{R}_{\mathrm{PPAT}}(\widehat{\lambda})] - r = -\mathbb{E}[\mathrm{Cov}(\widehat{\lambda}, \widehat{C} \mid D_{test})] \tag{26}$$

Proof. Let $\widehat{R} = \frac{1}{N}\sum_{i=1}^{N} \ell_i$ be the empirical risk on $\mathcal{D}_{\text{test}}$ using the true labels, and note that

$$\widehat{R}_{\mathrm{PPAT}}(\lambda) = \widehat{R}_{\mathrm{LURE}} - \lambda\widehat{C}. \tag{27}$$

Moreover, we have:

$$\mathbb{E}\left[\widehat{R}_{\mathrm{LURE}} \mid D_{\text{test}}\right] = \widehat{R}, \quad \mathbb{E}\left[\widehat{C} \mid D_{\text{test}}\right] = \frac{1}{N}\sum_{i=1}^{N} c_i = 0, \tag{28}$$

where the expectation on the left follows from properties of LURE (Farquhar et al., 2021) and the expectation on the right follows from the definition of the c_is. Now, applying the linearity of expectations to (26) gives:

$$\mathbb{E}[\widehat{R}_{\mathrm{PPAT}} \mid D_{\text{test}}] = \widehat{R} - \mathbb{E}[\widehat{\lambda}\widehat{C} \mid D_{\text{test}}],$$

Now, by the definition of covariance and (28) we can write the expectation on the right hand side as:

$$\begin{aligned}\mathbb{E}[\widehat{\lambda}\widehat{C} \mid D_{\text{test}}] &= \mathrm{Cov}(\widehat{\lambda}, \widehat{C} \mid D_{\text{test}}) + \mathbb{E}[\widehat{\lambda} \mid D_{\text{test}}]\mathbb{E}[\hat{C} \mid D_{\text{test}}] \\ &= \mathrm{Cov}(\widehat{\lambda}, \widehat{C} \mid D_{\text{test}}).\end{aligned}$$

Hence

$$\mathbb{E}[\widehat{R}_{\mathrm{PPAT}} \mid D_{\text{test}}] = \widehat{R} - \mathrm{Cov}(\widehat{\lambda}, \widehat{C} \mid D_{\text{test}}),$$

and taking expectations on both sides gives the desired result. □

The above result shows that unless $\widehat{\lambda}$ has zero covariance with $\widehat{C}$, which generally is not the case when $\widehat{\lambda}$ is $\mathcal{F}_M$–measurable, then our estimator is biased.

Remark. Note that the same result will hold if $\widehat{\lambda}$ is $\mathcal{F}_m$–measurable, for $1 \leq m \leq M$.

F. Algorithm Block

Algorithm 1 Prediction–Powered Active Testing (PPAT)

Require: Test pool $\{\mathbf{x}_i\}_{i=1}^N$, model f, loss $\mathcal{L}$, proxy labeller g, budget M, surrogate π_1, fixed λ

for $i = 1, \ldots, N$ **do**
  $\tilde{y}_i \leftarrow g(\mathbf{x}_i), \quad \tilde{\ell}_i \leftarrow \mathcal{L}(f(\mathbf{x}_i), \tilde{y}_i)$
end for
$\bar{\ell}_{\text{proxy}} \leftarrow \frac{1}{N}\sum_{i=1}^N \tilde{\ell}_i, \quad c_i \leftarrow \tilde{\ell}_i - \bar{\ell}_{\text{proxy}}$
$S_1 \leftarrow \{1, \ldots, N\}$
for $m = 1, \ldots, M$ **do**
  for $i \in S_m$ **do**
    $s_{m,i}(\lambda) \leftarrow \mathbb{E}_{y\sim\pi_m(\cdot|x_i)}\Big[\big(\mathcal{L}(f(x_i), y) - \lambda c_i\big)^2\Big]$
  end for
  $q_m(i) \propto s_{m,i}(\lambda)$ for $i \in S_m$
  Sample $i_m \sim q_m(\cdot)$, query y_{i_m}, and set $\ell_{i_m} \leftarrow \mathcal{L}(f(\mathbf{x}_{i_m}), y_{i_m})$
  $v_m \leftarrow 1 + \frac{N-M}{N-m}\left(\frac{1}{(N-m+1)q_m(i_m)} - 1\right)$
  Update surrogate π_{m+1} using the newly observed pair $(\mathbf{x}_{i_m}, y_{i_m})$
  $S_{m+1} \leftarrow S_m \setminus \{i_m\}$
end for
return $\widehat{R}_{\text{PPAT}}(\lambda) = \frac{1}{M}\sum_{m=1}^M v_m\big(\ell_{i_m} - \lambda c_{i_m}\big)$