From Tuning to Guarantees: Statistically Valid Hyperparameter Selection

Amirmohammad Farzaneh

KCLIP Lab, Centre for Intelligent Information Processing Systems (CIIPS)
Department of Engineering, King's College London

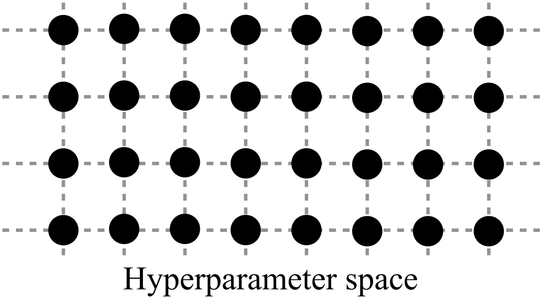
NeurIPS 2025
2 December 2025

Hyperparameter Selection

- Hyperparameter selection is a key step in the deployment of pre-trained AI models.
- **Examples:**
  - **inference** parameters, such as test-time resources, prompt templates, temperature, or decision thresholds
  - **implementation** parameters, such as arithmetic precision

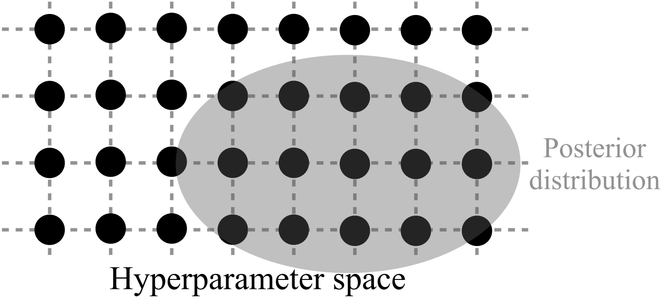
Hyperparameter Selection

- Traditional approaches are of **best effort** nature, targeting **empirical performance**.
  - No formal statistical guarantees
- **Examples:** Grid search [Bergstra and Bengio, 2012]



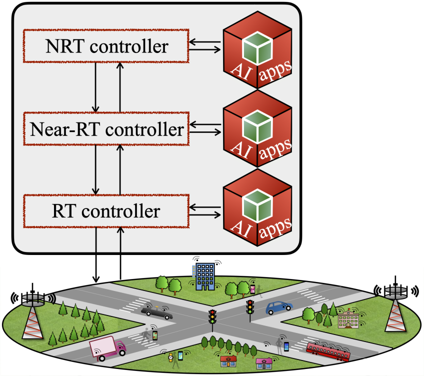
Hyperparameter Selection

- Traditional approaches are of **best effort** nature, targeting **empirical performance**:
  - No formal statistical guarantees
- **Examples:** Bayesian optimization [Snoek, Larochelle, and Adams, 2012]



Hyperparameter Selection

- The deployment of AI in sensitive domains such as healthcare [Dzau et al., 2023] and engineering [Simeone, Park, and Zecchin, 2025] requires **rigorous statistical guarantees**.
- **Example:** AI-native wireless systems in 6G (e.g., O-RAN)



Hyperparameter Selection

- **Reliability requirements** are application specific, e.g., probability of error, latency, fairness, ...
- Hyperparameter λ is said to be **reliable** if the AI model meets the reliability requirement when run with hyperparameter λ.

Hyperparameter Selection

- **Reliability requirements** are application specific, e.g., probability of error, latency, fairness, ...
- Hyperparameter λ is said to be **reliable** if the AI model meets the reliability requirement when run with hyperparameter λ.
- **Statistical validity** in hyperparameter selection:

$$\Pr[\lambda \text{ is not reliable}] \leq \delta,$$

for a user-specified $0 < \delta < 1$ ($\Pr[\cdot]$ is over data used for selection).

Discussion & Reflection

How should we think about reliability in practice?

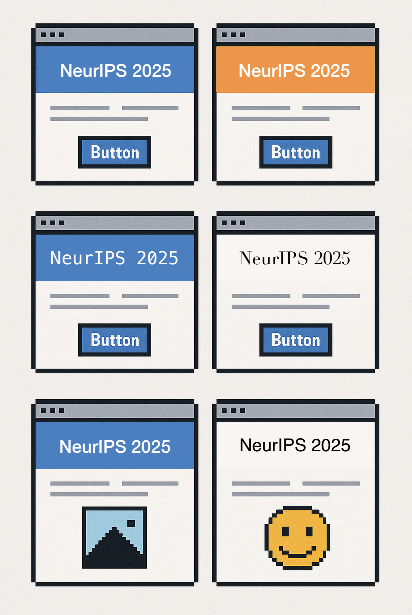
Consider an application you care about (e.g., LLM prompting, RL policies, model selection, or wireless systems), and reflect on:

- **What does it mean for a hyperparameter to be reliable?** (e.g., stable accuracy, low risk under distribution shift, robustness across seeds)
- **Which failures actually matter?** (e.g., performance drops, fairness violations, safety constraints)
- **What evidence would convince you that a hyperparameter is safe to deploy?**
- **How would you formalize these notions statistically?**

We will revisit these questions as we connect reliability to FDR, risk control, and hypothesis testing.

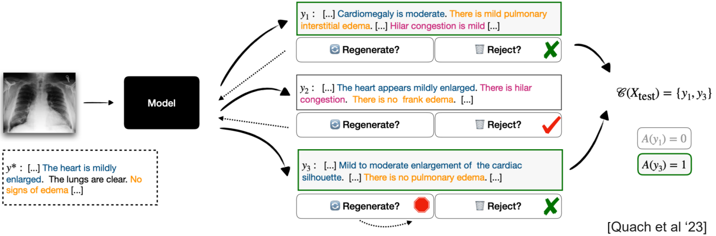
Learn-Then-Test

- **Learn-Then-Test (LTT)** [Angelopoulos, Bates, Candès, et al., 2021] formalizes statistically valid hyperparameter selection via **multiple hypothesis testing (MHT)**.



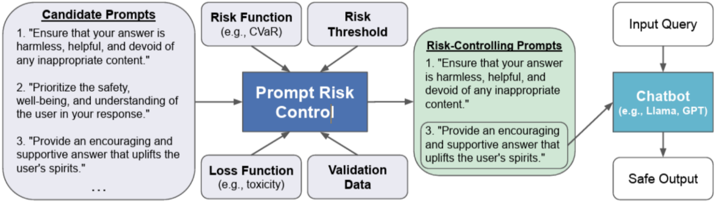
Learn-Then-Test: Some Applications

- Calibrating thresholds for **test-time scaling** (generation of multiple answers) in **LLMs** [Quach et al., 2023]



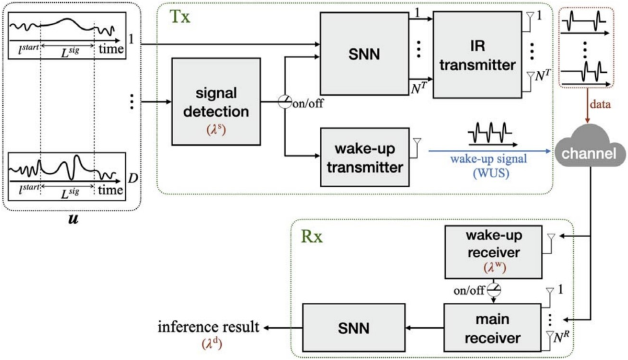
Learn-Then-Test: Some Applications

- **Recommendation systems** (learning to rank) [Angelopoulos, Krauth, et al., 2023]
- **Information retrieval** [Xu et al., 2024]
- **Prompt template** selection for LLMs [Zollo et al., 2023]



Learn-Then-Test: Some Applications

- **Telecommunication systems** [Simeone, Park, and Zecchin, 2025]
- **Neuromorphic computing** [Jiechen Chen et al., 2024]



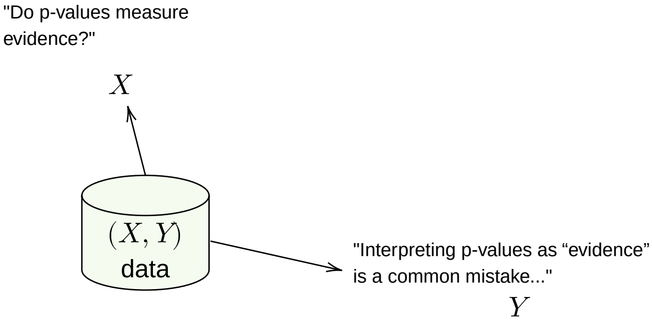
Overview

- Learn-Then-Test
- Generalizing Learn-Then-Test:
  - Beyond the Average Risk
  - Adaptive Hyperparameter Selection
  - Incorporating Prior Information
  - Selection with Autoevaluation
- Conclusions

Learn-Then-Test

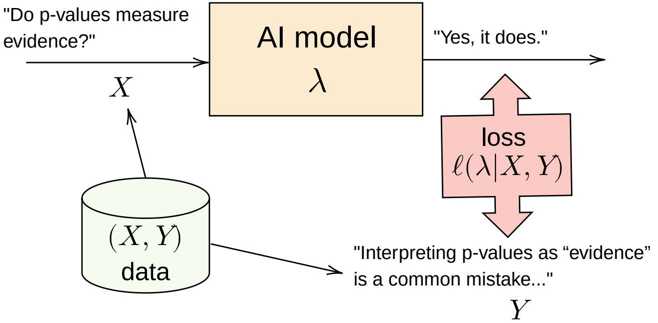
Reliability and Statistical Validity

- Data (X, Y) with X = input and Y = output



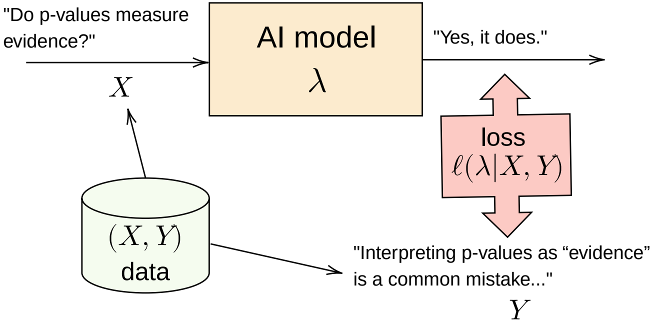
Reliability and Statistical Validity

- $\ell(\lambda|X, Y)$ = **loss** at data point (X, Y) under hyperparameter λ (e.g., 0-1 loss)



Reliability and Statistical Validity

- $\ell(\lambda|X, Y)$ = **loss** at data point (X, Y) under hyperparameter λ (e.g., 0-1 loss)



- $R(\lambda) = \mathbb{E}_{(X,Y)}[\ell(\lambda|X, Y)]$ = **average risk** (e.g., probability of error)

Examples of Losses in Different Applications

Reliability depends on how we measure failure. Different applications call for different choices of loss $\ell(\lambda \mid X, Y)$:

- **Classification / LLM prompting** 0–1 loss, misclassification rate, token-level error, or a scoring loss (e.g., NLL/perplexity).
- **Regression / Forecasting** Squared error $(Y - \hat{Y})^2$, absolute error $|Y - \hat{Y}|$, pinball loss for quantiles.
- **Structured Prediction / Detection** mAP drop, IoU-based penalties, object-miss penalties, calibration errors (ECE/MCE).
- **Reinforcement Learning / Control** Negative return, constraint violation counts, regret, safety-critical failure events.
- **LLM-based agents / LLM judging** Preference loss, pairwise defeat probability (Bradley–Terry), task-success or factuality loss.
- **Wireless / Telecom Scheduling** Throughput deficit, latency violation, outage probability, or risk of violating QoS thresholds.

Once the loss is fixed, reliability means keeping its expected value $R(\lambda)$ acceptably low.

Reliability and Statistical Validity

- **Reliability condition:**

$$\lambda \text{ is reliable } \Leftrightarrow R(\lambda) \leq \alpha,$$

where α is a user-specified threshold

Reliability and Statistical Validity

- **Reliability condition:**

$$\lambda \text{ is reliable } \Leftrightarrow R(\lambda) \leq \alpha,$$

where α is a user-specified threshold

- **Statistical validity** in hyperparameter selection:

$$\Pr\big[R(\lambda) > \alpha \text{ for a selected } \lambda\big] \leq \delta.$$

Reliability and Statistical Validity

- **Reliability condition:**

$$\lambda \text{ is reliable } \Leftrightarrow R(\lambda) \leq \alpha,$$

  where α is a user-specified threshold
- **Statistical validity** in hyperparameter selection:

$$\Pr\big[R(\lambda) > \alpha \text{ for a selected } \lambda\big] \leq \delta.$$

Parameter	**Significance**
α	Reliability requirement: $R(\lambda) \leq \alpha$
δ	Maximum allowed failure rate for the hyperparameter selection strategy

Conventional Hyperparameter Selection

- Given some **held-out data** $\mathcal{D} = \{(X, Y)\}$, conventional methods directly optimize standard **empirical risk estimates**

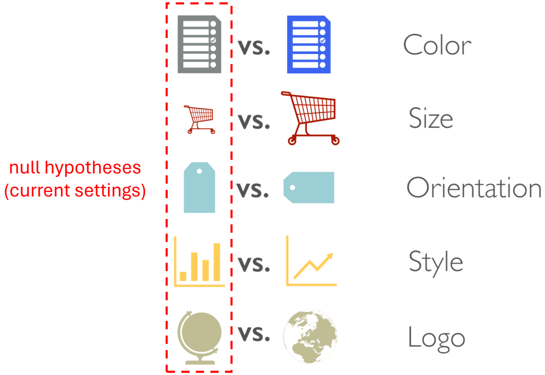
$$\hat{R}(\lambda|\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \ell(\lambda|X, Y)$$

  over hyperparameter λ.
- Statistical validity not guaranteed.

Learn-Then-Test

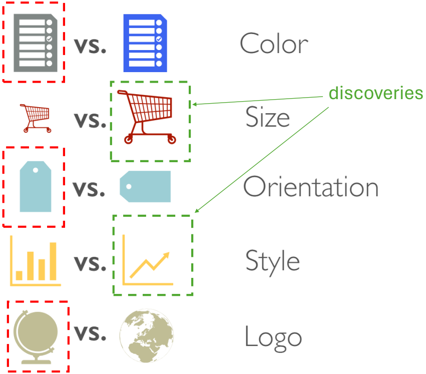
- LTT provides reliability guarantees by framing hyperparameter selection as **multiple hypothesis testing** (MHT).
- MHT underlies scientific discovery, A/B testing, ...

- LTT provides reliability guarantees by framing hyperparameter selection as **multiple hypothesis testing** (MHT).
- MHT underlies scientific discovery, A/B testing, ...
- MHT considers simultaneously a number of **binary hypotheses**.



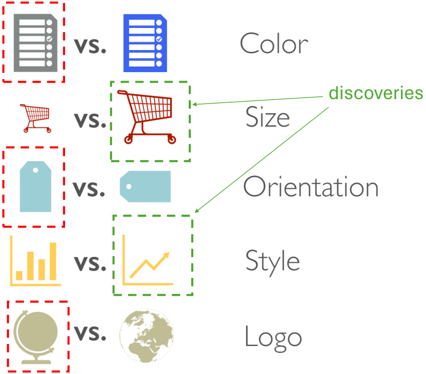
Learn-Then-Test

- Each hypothesis is tested by collecting data (e.g., click-through rates)...
- ... producing a decision for the **null hypothesis** or for the **alternative hypothesis (discovery)**.



Learn-Then-Test

- Each hypothesis is tested by collecting data (e.g., click-through rates)...
- ... producing a decision for the **null hypothesis** or for the **alternative hypothesis (discovery)**.



- **Goal:** Control **error rate** metrics such as the **family-wise error rate (FWER)**:

$$\Pr[\text{any false discovery}] \leq \delta$$

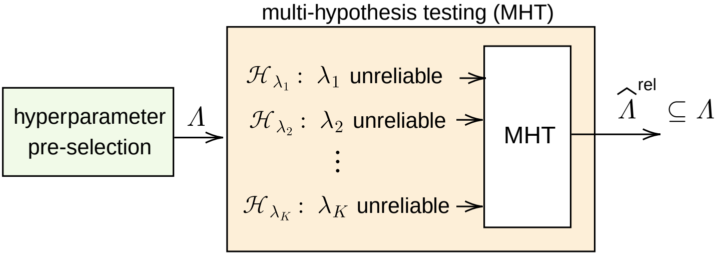
Learn-Then-Test

- ① Determine a set Λ of **candidate hyperparameters** using any methodology, such as grid search [Bergstra and Bengio, 2012] or Bayesian optimization [Snoek, Larochelle, and Adams, 2012].

Learn-Then-Test

- ① Determine a set Λ of **candidate hyperparameters** using any methodology, such as grid search [Bergstra and Bengio, 2012] or Bayesian optimization [Snoek, Larochelle, and Adams, 2012].
- ② Apply MHT for hyperparameter selection within set Λ:
  - Test a **null hypothesis** for each candidate hyperparameter λ:

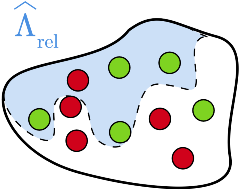
$$\mathcal{H}_\lambda : \ \lambda \text{ is unreliable, i.e., } R(\lambda) > \alpha$$



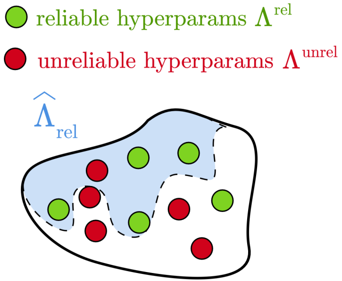
Learn-Then-Test

reliable hyperparams Λ^{rel}

unreliable hyperparams Λ^{unrel}



Learn-Then-Test



- By using MHT, the subset $\hat{\Lambda}^{\text{rel}}$ of selected hyperparameters satisfies **error rate** control guarantees.
- Notably, the **family-wise error rate** (FWER) guarantee coincides with the **statistical validity** condition

$$\Pr\big[R(\lambda) > \alpha \text{ for a selected } \lambda\big] \leq \delta.$$

Multiple Hypothesis Testing

- Given some held-out data $\mathcal{D} = \{(X, Y)\}$, evaluate standard **empirical risk estimates**

$$\hat{R}(\lambda|\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \ell(\lambda|X, Y)$$

  for every $\lambda \in \Lambda$.

Multiple Hypothesis Testing

- Given some held-out data $\mathcal{D} = \{(X, Y)\}$, evaluate standard **empirical risk estimates**

$$\hat{R}(\lambda|\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \ell(\lambda|X, Y)$$

  for every $\lambda \in \Lambda$.
- A larger value of the **estimated reliability margin** $(\alpha - \hat{R}(\lambda|\mathcal{D}))^+$ provides more evidence that λ is reliable.

Multiple Hypothesis Testing

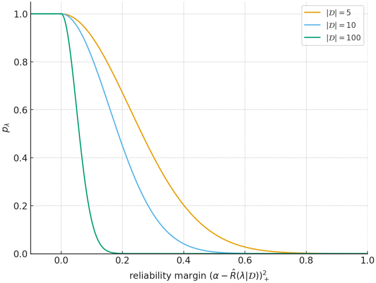
- Given some held-out data $\mathcal{D} = \{(X, Y)\}$, evaluate standard **empirical risk estimates**

$$\hat{R}(\lambda|\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \ell(\lambda|X, Y)$$

  for every $\lambda \in \Lambda$.
- A larger value of the **estimated reliability margin** $(\alpha - \hat{R}(\lambda|\mathcal{D}))^+$ provides more evidence that λ is reliable.
- Thus, a statistic of the form

$$p_\lambda = e^{-2|\mathcal{D}|(\alpha-\hat{R}(\lambda|\mathcal{D}))^2_+}.$$

  will tend to be small when λ is reliable.

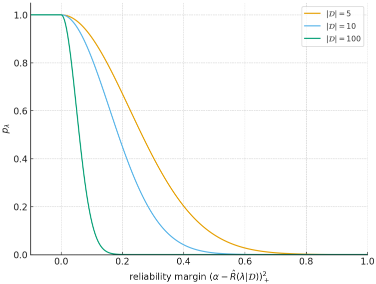


Multiple Hypothesis Testing

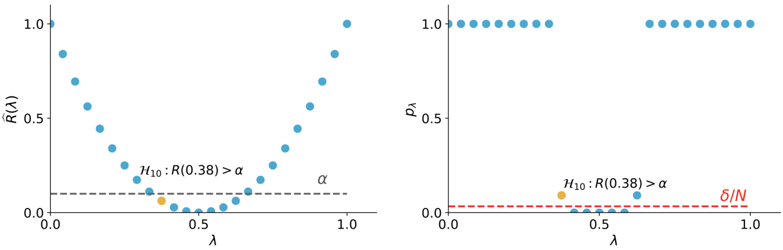
- If the loss is bounded within $[0, 1]$, the statistic p_λ is a **p-value** for the null hypothesis $\mathcal{H}_\lambda$.
- A small p-value p_λ indicates evidence that the hyperparameter λ is reliable:

$$\Pr[p_\lambda \leq \delta \mid \lambda \text{ is unreliable}] \leq \delta,$$

for $0 < \delta < 1$.



Why Empirical Risk Alone Is Not Reliable

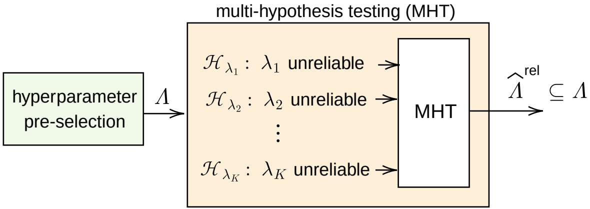


- Empirical risk $\hat{R}(\lambda)$ can underestimate the true risk due to finite-sample noise.
- This may cause an **unreliable** hyperparameter to falsely appear safe.
- LTT converts these empirical risk gaps into **valid** p**-values**, ensuring unreliable hyperparameters cannot "slip through" by chance.

Figure reproduced from Angelopoulos, Bates, Candès, et al., 2021.

Multiple Hypothesis Testing

- We have a set of p-values p_λ, one for each hyperparameter $\lambda \in \Lambda$.



- **Examples of FWER-controlling MHT procedures:**
  - Bonferroni correction

$$\lambda \in \hat{\Lambda}^{\text{rel}} \text{ if } p_\lambda \leq \frac{\delta}{|\Lambda|}$$

  - Fixed sequence testing

Error Rates in Multiple Hypothesis Testing

- When testing many hypotheses simultaneously (e.g., many hyperparameters), different types of errors can occur.
- Two common error metrics:
  - **FWER (Family-Wise Error Rate)**: Probability of making *at least one* false discovery.
  - **FDR (False Discovery Rate)**: Expected *proportion* of false discoveries among all discoveries.
- These metrics control different risks, and are used for different applications.

Family-Wise Error Rate (FWER)

- In hyperparameter selection, a "false discovery" means:

  Selecting a hyperparameter λ that is actually unreliable.

- **FWER** measures the probability of making *any* such mistake:

$$\mathsf{FWER} = \mathsf{Pr}\left(\exists\lambda \in \hat{\Lambda}^{\mathrm{rel}} : R(\lambda) > \alpha\right).$$

- **Goal of LTT:** Ensure

$$\mathsf{FWER} \leq \delta,$$

  which *exactly matches* the notion of statistical validity in hyperparameter selection.
- Very conservative: tries to avoid even a single unreliable choice.

False Discovery Rate (FDR)

- In set-valued prediction problems (e.g., multi-label classification), we may output several items at once.
- **FDR** measures the *expected proportion* of incorrect predictions:

$$\mathrm{FDR}(T) = \mathbb{E}\left[\frac{\#\{\text{false predictions}\}}{\max\{1, \#\{\text{predictions}\}\}}\right].$$

- Unlike FWER, FDR does *not* penalize a single mistake heavily.
- FDR is appropriate when:
  - We produce many items at once (e.g., a set of labels),
  - and we care about the average purity of the set.
- **Examples of notable FDR-controlling procedures:**
  - **Benjamini–Hochberg (BH) procedure** [Benjamini and Hochberg, 1995]
  - **Benjamini–Yekutieli (BY)** [Benjamini and Yekutieli, 2001]

Example 1: Multi-Label Classification with FDR Control

- Task: **multi-label classification** on MS COCO.
  - Each image X has a set of labels $Y \subseteq \{1, \ldots, K\}$.
  - Base model $\hat{f}(x) \in [0,1]^K$ outputs class-wise probabilities.
- We predict a **set of labels**

$$T_\lambda(x) = \{z \in \{1, \ldots, K\} : \hat{f}_z(x) \geq \lambda\}$$

and we want to control the **false discovery rate** (FDR) of wrong labels in $T_\lambda(X)$.

- LTT chooses λ so that FDR is controlled at level (α, δ).

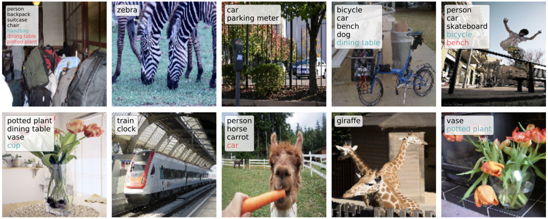
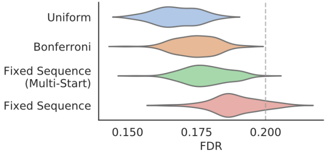


Figure reproduced from Angelopoulos, Bates, Candès, et al., 2021.

Example 1: Multi-Label Classification with FDR Control

- Dataset: MS COCO, $n = 4000$ calibration points, 1000 validation points.
- Candidate thresholds: $\Lambda = \{0, 0.001, \ldots, 1\}$.
- LTT compares different MHT procedures:
  - Uniform (no multiple testing correction)
  - Bonferroni
  - Fixed-sequence testing
  - Fixed-sequence testing with multiple starting points
- Goal: control FDR at level $\alpha = 0.2$ with failure probability $\delta = 0.1$.



Method	50%	75%	90%	99%	99.9%
Uniform	3	4	6	11	13
Bonferroni	3	4	7	11	14
Fixed Sequence (Multi-Start)	3	4	7	11	14
Fixed Sequence	3	5	7	12	14

Figure reproduced from Angelopoulos, Bates, Candès, et al., 2021.

Interactive LTT Notebook & Code

The following GitHub repo contains a simple, self-contained implementation of Learn-Then-Test (LTT) that you can run immediately on Colab or locally.

https://github.com/amirfar76/neurips25-valid-hparam-tutorial

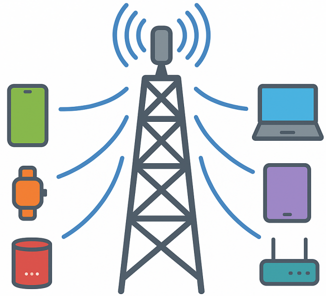
Generalizing Learn-Then-Test: Beyond the Average Risk

Beyond the Average Risk

- LTT controls the **average risk** $R(\lambda) = \mathbb{E}_{(X,Y)}[\ell(\lambda|X,Y)]$.
- In some applications, it may be preferable to control:
  - a quantile of the loss $\ell(\lambda|X,Y)$: **quantile risk**
  - functionals of the data distribution: **information-theoretic measures**

Controlling the Quantile Risk: Example

- Wireless scheduling problem:

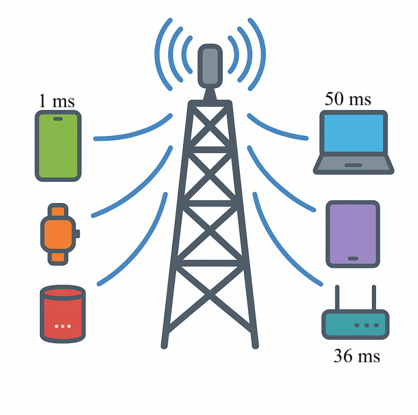


- **Reliability requirement using the average risk:** Ensure that the average user delay is less than 10 ms

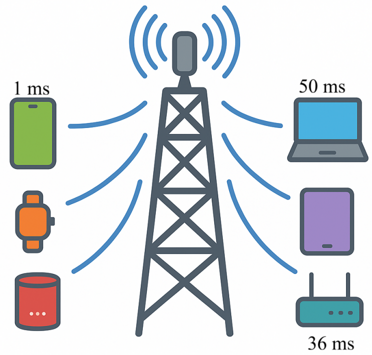
$$R(\lambda) \leq \alpha = 10 \text{ ms}$$

with a probability no smaller than 90% ($\delta = 0.1$)

Controlling the Quantile Risk: Example



Controlling the Quantile Risk: Example



- **Reliability requirement using the quantile risk:**
  - At least 90% of the users must have a delay smaller than 10 ms

$$R_q(\lambda) \leq \alpha = 10 \text{ ms}$$

with $q = 0.1$

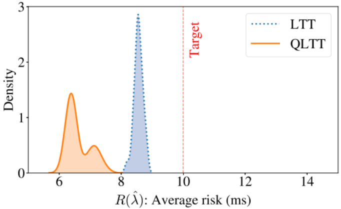
Quantile Learn-Then-Test

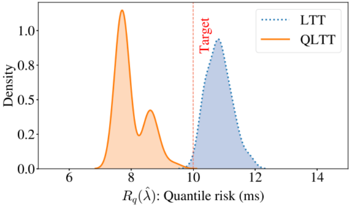
- **Quantile LTT** (QLTT) [Farzaneh, Park, and Simeone, 2024] extends LTT to control the quantile risk.
- **Invert a confidence interval** on the desired quantile to obtain a p-value.

Quantile Learn-Then-Test

- **Quantile LTT** (QLTT) [Farzaneh, Park, and Simeone, 2024] extends LTT to control the quantile risk.
- **Invert a confidence interval** on the desired quantile to obtain a p-value.

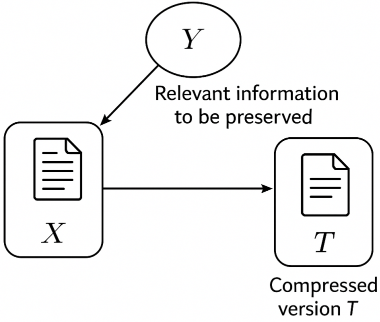
$q = 0.1$





Controlling Information-Theoretic Measures: Information Bottleneck Problem

- X = input
- Y = target variable
- $(X, Y) \sim P_{XY}$, with P_{XY} **unknown**
- **Goal:** Extract a **compressed representation** $T \sim P_{T|X}$ of X that is sufficiently **relevant** for predicting Y.



Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Mutual information:

$$I(A;B) = \mathbb{E}_{P_{AB}}\left[\log_2\left(\frac{P_{AB}}{P_A \cdot P_B}\right)\right]$$

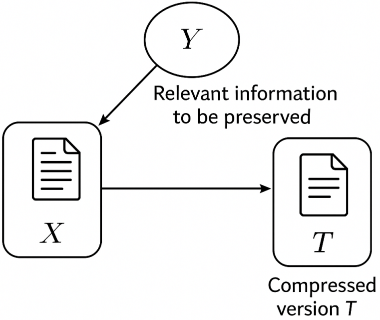
Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Mutual information:

$$I(A;B) = \mathbb{E}_{P_{AB}}\left[\log_2\left(\frac{P_{AB}}{P_A \cdot P_B}\right)\right]$$

- **Information Bottleneck (IB) problem** [Tishby, Pereira, and Bialek, 2001; Zaidi, Estella-Aguerri, and Shamai, 2020]:

$$\begin{aligned} &\underset{P_{T|X}}{\text{minimize}} \quad I(X;T) && \text{(size in bits)} \\ &\text{subject to} \quad I(T;Y) \geq \alpha && \text{(relevance)} \end{aligned}$$



Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Given data $\mathcal{D} = \{(X, Y)\} \underset{\text{i.i.d.}}{\sim} P_{XY}$, one can produce the estimates $\hat{I}(X;T)$ and $\hat{I}(T;Y)$.

Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Given data $\mathcal{D} = \{(X, Y)\} \underset{\text{i.i.d.}}{\sim} P_{XY}$, one can produce the estimates $\hat{I}(X; T)$ and $\hat{I}(T; Y)$.
- Introduce a **Lagrange multiplier** $\lambda > 0$ to tackle the **unconstrained problem** [Alemi et al., 2017]

$$\underset{P_{T|X}}{\text{minimize}} \quad \underbrace{\hat{I}(X; T)}_{\text{size}} - \lambda \underbrace{\hat{I}(T; Y)}_{\text{relevance}}.$$

Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Given data $\mathcal{D} = \{(X, Y)\} \underset{\text{i.i.d.}}{\sim} P_{XY}$, one can produce the estimates $\hat{I}(X;T)$ and $\hat{I}(T;Y)$.
- Introduce a **Lagrange multiplier** $\lambda > 0$ to tackle the **unconstrained problem** [Alemi et al., 2017]

$$\underset{P_{T|X}}{\text{minimize}} \quad \underbrace{\hat{I}(X;T)}_{\text{size}} - \lambda \underbrace{\hat{I}(T;Y)}_{\text{relevance}}.$$

- **Reliability requirement:** Select the hyperparameter λ to guarantee the **relevance constraint**

$$I(T;Y) \geq \alpha$$

Controlling Information-Theoretic Measures: Information Bottleneck Problem

- Given data $\mathcal{D} = \{(X, Y)\} \underset{\text{i.i.d.}}{\sim} P_{XY}$, one can produce the estimates $\hat{I}(X;T)$ and $\hat{I}(T;Y)$.
- Introduce a **Lagrange multiplier** $\lambda > 0$ to tackle the **unconstrained problem** [Alemi et al., 2017]

$$\underset{P_{T|X}}{\text{minimize}} \quad \underbrace{\hat{I}(X;T)}_{\text{size}} - \lambda \underbrace{\hat{I}(T;Y)}_{\text{relevance}}.$$

- **Reliability requirement:** Select the hyperparameter λ to guarantee the **relevance constraint**

$$I(T;Y) \geq \alpha$$

- Since the joint distribution P_{XY} is not known, one can only ensure the **probabilistic relevance constraint**:

$$\Pr\left[I(T;Y) < \alpha\right] \leq \delta$$

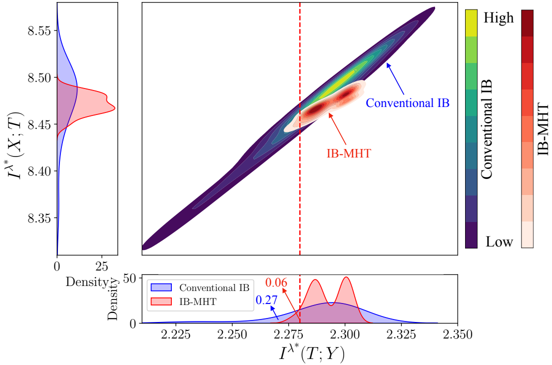
where δ is a user defined probability

Controlling Information-Theoretic Measures: Information Bottleneck Problem

- **IB-MHT** [Farzaneh and Simeone, 2024] inverts a confidence interval on the mutual information to obtain p-values for each candidate Lagrange multiplier, allowing the use of LTT.

Controlling Information-Theoretic Measures: Information Bottleneck Problem

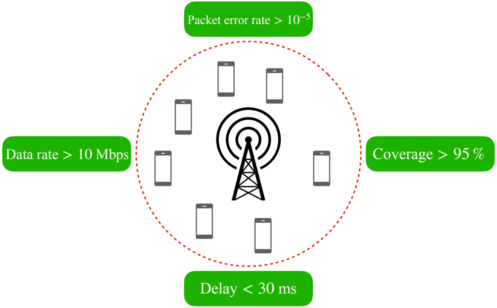
- **IB-MHT** [Farzaneh and Simeone, 2024] inverts a confidence interval on the mutual information to obtain p-values for each candidate Lagrange multiplier, allowing the use of LTT.
- **Image processing example:** Joint distributions of the mutual informations $I(T;Y)$ and $I(X;T)$ obtained by using a conventional IB solver [Alemi et al., 2017] and IB-MHT ($\delta = 0.1$)



Generalizing Learn-Then-Test: Multi-Objective Optimization

Multi-Objective Optimization

- LTT only controls a single risk function $R(\lambda)$.
- In some applications, there are multiple risk functions to be controlled.

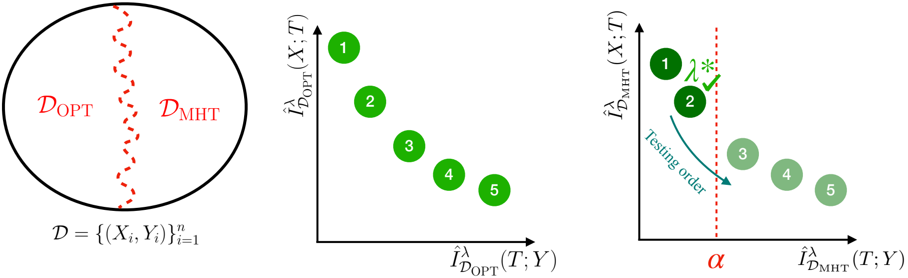


$$\min_{\lambda \in \Lambda} \{R_{L_c+1}(\lambda), R_{L_c+2}(\lambda), \dots, R_L(\lambda)\}$$

$$\text{subject to } R_l(\lambda) < \alpha_l \text{ for all } 1 \leq l \leq L_c,$$

Pareto Testing

- To address this, Pareto testing (PT) [Laufer-Goldshtein et al., 2023] takes the following steps.



① Split data set $\mathcal{D}$

② Estimate Pareto frontier using $\mathcal{D}_{\text{OPT}}$

③ Sequential FWER-controlling MHT using $\mathcal{D}_{\text{MHT}}$

Interactive PT Notebook & Code

The following GitHub repo contains a simple, self-contained implementation of Pareto Testing (PT) that you can run immediately on Colab or locally.

https://github.com/amirfar76/neurips25-valid-hparam-tutorial

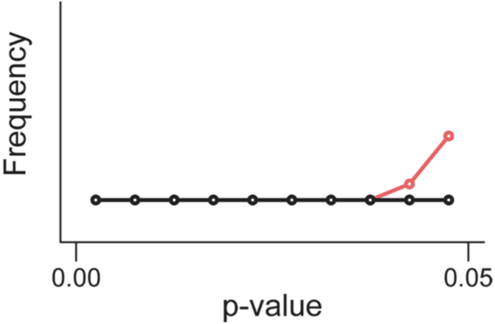
Generalizing Learn-Then-Test: Adaptive Hyperparameter Selection

Adaptive Hyperparameter Selection

- LTT is a **batch** method operating on a **fixed data set**.
- In practice, collecting data and evaluating the loss of a model can be **expensive**: it may require interactions with the real world, simulations, optimizations, ...
- It is thus useful to carry out testing **sequentially**, **discarding** less performing hyperparameters and **terminating** as soon as possible.

Adaptive Hyperparameter Selection

- LTT does not support adaptive hyperparameter selection:
- p-values do not allow for **optional continuation**: **p-hacking** [Head et al., 2015].



Adaptive Hyperparameter Selection

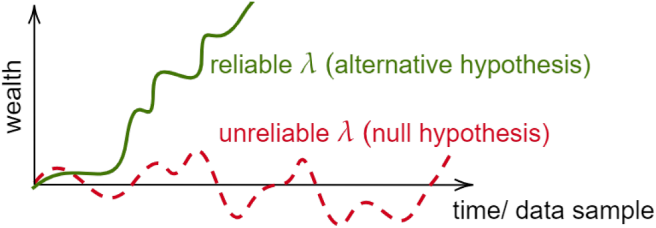
- **E-values** are a (nonparametric, composite) generalization of likelihood ratios [Shafer and Vovk, 2019, Ramdas and Wang, 2024], which are more robust than p-values.

Adaptive Hyperparameter Selection

- **E-values** are a (nonparametric, composite) generalization of likelihood ratios [Shafer and Vovk, 2019, Ramdas and Wang, 2024], which are more robust than p-values.
- Specifically, the sequential extension of e-values, known as **e-processes**, supports **optional continuation** (by Ville's theorem).

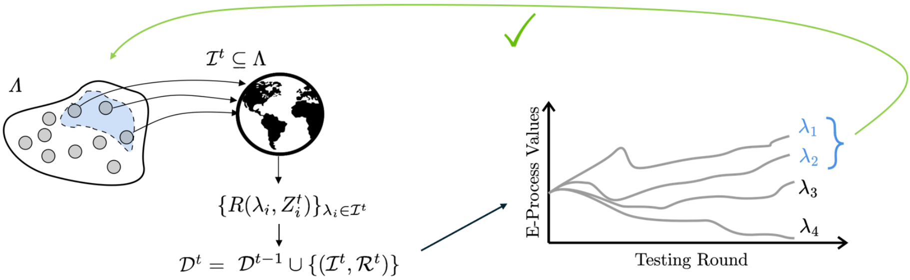
Adaptive Hyperparameter Selection

- **E-values** are a (nonparametric, composite) generalization of likelihood ratios [Shafer and Vovk, 2019, Ramdas and Wang, 2024], which are more robust than p-values.
- Specifically, the sequential extension of e-values, known as **e-processes**, supports **optional continuation** (by Ville's theorem).
- An e-process has the **game-theoretic** interpretation of accumulated wealth for a strategy **betting** on the reliability of a hyperparameter.



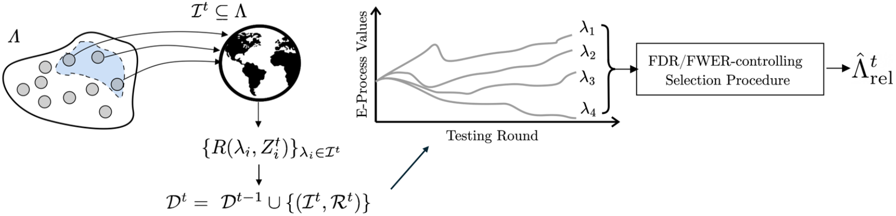
Adaptive Hyperparameter Selection

- **Adaptive LTT (aLTT)** [Zecchin, Park, and Simeone, 2024]
  - Test a subset $\mathcal{I}^t \subseteq \Lambda$ of hyperparameters at each testing round t
  - Update **e-process** for each hyperparameter $\lambda \in \mathcal{I}^t$
  - Stop, producing a set Λ^{rel}, or continue



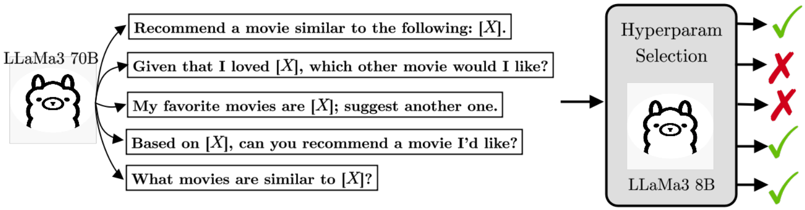
Adaptive Hyperparameter Selection

- Thanks to the **anytime validity** properties of e-processes, aLTT supports **optional continuation** and **adaptive termination**, while guaranteeing FWER control.

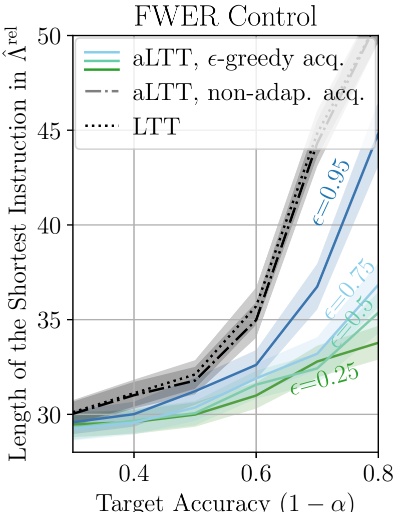


Example

- Prompt template selection from a set Λ of LLM-generated candidate prompts.
- **Reliability requirement:** Average error rate of the answer
- **Post-selection optimization:** Minimize average prompt length



Example

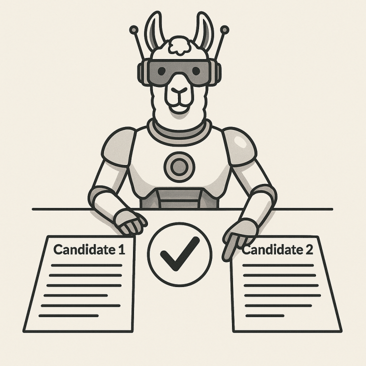


FWER Control
Length of the Shortest Instruction in $\hat{\Lambda}^{\text{rel}}$
50
45
40
35
30
aLTT, ϵ-greedy acq.
aLTT, non-adap. acq.
LTT
ϵ=0.95
ϵ=0.75
ϵ=0.5
ϵ=0.25
0.4
0.6
0.8
Target Accuracy $(1-\alpha)$

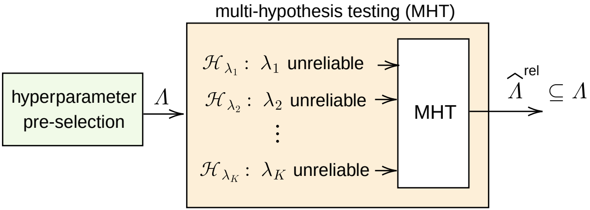
Generalizing Learn-Then-Test: Incorporating Prior Information

Incorporating Prior Information

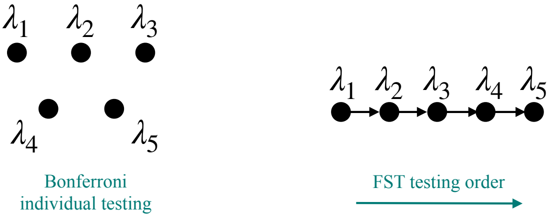
- In some applications, one may have prior information about the **relative reliability level** of different hyperparameters.
- **Examples**:
  - a higher resource utilization may imply higher accuracy
  - use **LLM-as-a-judge**



Incorporating Prior Information

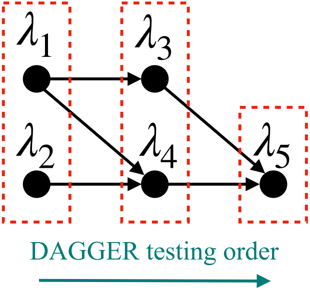


- Prior information can inform the MHT step, e.g., via **fixed-sequence testing** (FST):



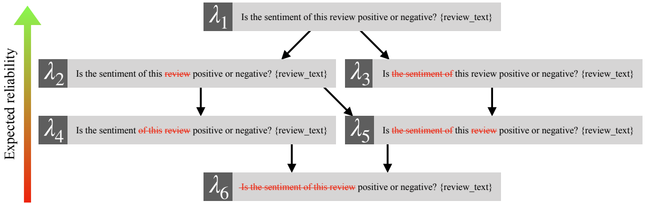
Reliability Graph-based Pareto Testing

- However, prior information – **possibly extracted from separate data** – may be much richer than a simple linear ordering.
- **Reliability graph-based Pareto testing** (RG-PT) [Farzaneh and Simeone, 2025] operates MHT over a directed acyclic graph (via DAGGER [Ramdas, Jianbo Chen, et al., 2019]).



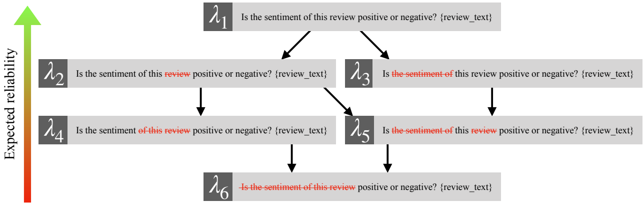
Example

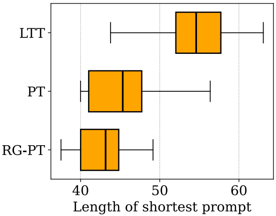
- Prompt template optimization



Example

- Prompt template optimization





(Pareto testing [Laufer-Goldshtein et al., 2023])

Generalizing Learn-Then-Test: Hyperparameter Selection with Autoevaluation

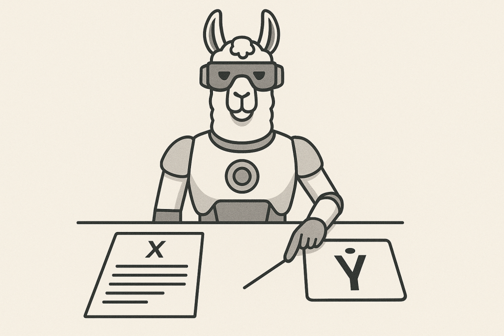
Reliable Hyperparameter Selection via Autoevaluation

- LTT requires held-out data (X, Y) to estimate the risk (and compute p-values or e-values).
- However, labels Y may be scarce.

Reliable Hyperparameter Selection via Autoevaluation

- LTT requires held-out data (X, Y) to estimate the risk (and compute p-values or e-values).
- However, labels Y may be scarce.
- **R-AutoEval** [Boyeau et al., 2024, Eyre and Madras, 2024, Schneider et al., 2024] allows for reliable hyperparameter selection via LTT by assuming:
  - limited labeled data
  - abundant unlabeled data

- R-AutoEval incorporates **synthetic labels** through **prediction-powered inference** (PPI) [Angelopoulos, Bates, Fannjiang, et al., 2023]

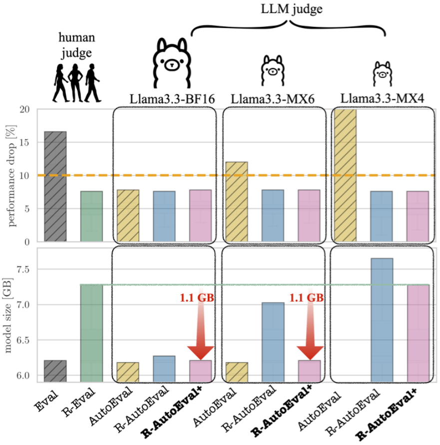


$$\begin{aligned}\text{PPI risk estimate } &= \text{estimate with unlabeled data}\\ &\quad - \text{bias correction using labeled data}\end{aligned}$$

R-AutoEval+

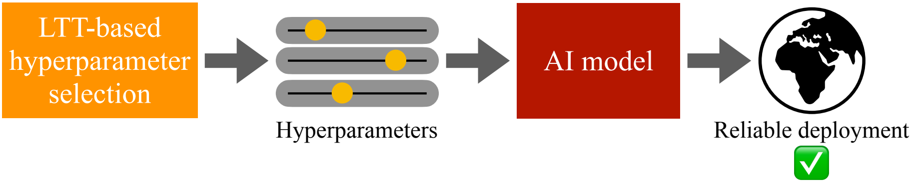
- **R-AutoEval+** [Park, Zecchin, and Simeone, 2025] **dynamically tunes its reliance on synthetic data**, reverting to conventional methods based only on labeled data when the autoevaluator is insufficiently accurate.

- **R-AutoEval+** [Park, Zecchin, and Simeone, 2025] **dynamically tunes its reliance on synthetic data**, reverting to conventional methods based only on labeled data when the autoevaluator is insufficiently accurate.
- Post-training quantization of LLMs

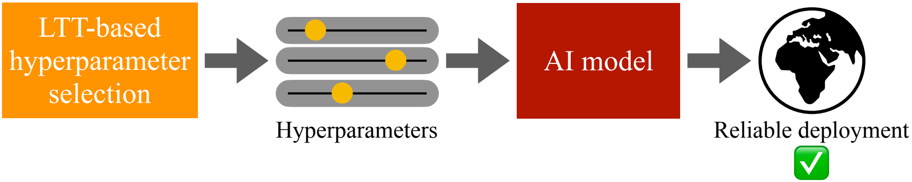


Conclusions

Conclusion



- Hyperparameter selection is a key step in the deployment of pre-trained AI models.
- Multiple hypothesis testing provides a principled statistical framework to ensure statistical guarantees (Learn-Then-Test).
- This talk has reviewed several practical extensions:
  - quantile risk and information-theoretic measures
  - adaptive selection
  - prior information for structured testing
  - autoevaluation



- Hyperparameter selection is a key step in the deployment of pre-trained AI models.
- Multiple hypothesis testing provides a principled statistical framework to ensure statistical guarantees (Learn-Then-Test).
- This talk has reviewed several practical extensions:
  - quantile risk and information-theoretic measures
  - adaptive selection
  - prior information for structured testing
  - autoevaluation
- Future work:
  - extension to training-time hyperparameter setting (e.g., freeze-thaw)
  - large-scale deployment in engineering settings (e.g., O-RAN)

Live Q&A with Anastasios Angelopoulos

University of California, Berkeley
Author of the original Learn–Then–Test (LTT) framework

people.eecs.berkeley.edu/~angelopoulos

We will now switch to the live Zoom session.

Please feel free to ask questions about LTT, conformal prediction, multiple hypothesis testing, or broader reliability topics.

Acknowledgments

- This work was supported by the European Union's Horizon Europe project CENTRIC (101096379), by the Open Fellowships of the EPSRC (EP/W024101/1), and by the EPSRC project (EP/X011852/1).

References I

Alemi, Alexander A et al. (2017). "Deep Variational Information Bottleneck". In: *Proc. International Conference on Learning Representations*.

Angelopoulos, Anastasios N, Stephen Bates, Emmanuel J Candès, et al. (2021). "Learn then test: Calibrating predictive algorithms to achieve risk control". In: *arXiv preprint arXiv:2110.01052*.

Angelopoulos, Anastasios N, Stephen Bates, Clara Fannjiang, et al. (2023). "Prediction-powered inference". In: *Science* 382.6671, pp. 669–674.

Angelopoulos, Anastasios N, Karl Krauth, et al. (2023). "Recommendation systems with distribution-free reliability guarantees". In: *Conformal and Probabilistic Prediction with Applications*. PMLR, pp. 175–193.

Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Benjamini, Yoav and Daniel Yekutieli (2001). "The control of the false discovery rate in multiple testing under dependency". In: *Annals of statistics*, pp. 1165–1188.

References II

- Bergstra, James and Yoshua Bengio (2012). "Random search for hyper-parameter optimization.". In: *Journal of machine learning research* 13.2.
- Boyeau, Pierre et al. (2024). "Autoeval done right: Using synthetic data for model evaluation". In: *arXiv preprint arXiv:2403.07008*.
- Chen, Jiechen et al. (2024). "Neuromorphic split computing with wake-up radios: Architecture and design via digital twinning". In: *IEEE Transactions on Signal Processing*.
- Dzau, Victor J et al. (2023). *Achieving the promise of artificial intelligence in health and medicine: Building a foundation for the future*.
- Eyre, Benjamin and David Madras (2024). "Auto-evaluation with few labels through post-hoc regression". In: *arXiv preprint arXiv:2411.12665*.
- Farzaneh, Amirmohammad, Sangwoo Park, and Osvaldo Simeone (2024). "Quantile Learn-Then-Test: Quantile-Based Risk Control for Hyperparameter Optimization". In: *IEEE Signal Processing Letters*.
- Farzaneh, Amirmohammad and Osvaldo Simeone (2024). "Statistically Valid Information Bottleneck via Multiple Hypothesis Testing". In: *arXiv preprint arXiv:2409.07325*.

References III

Farzaneh, Amirmohammad and Osvaldo Simeone (2025). "Multi-Objective Hyperparameter Selection via Hypothesis Testing on Reliability Graphs". In: *arXiv preprint arXiv:2501.13018*.

Head, Megan L et al. (2015). "The extent and consequences of p-hacking in science". In: *PLoS biology* 13.3, e1002106.

Laufer-Goldshtein, Bracha et al. (2023). "Efficiently controlling multiple risks with Pareto testing". In: *Proc. International Conference on Learning Representations*.

Park, Sangwoo, Matteo Zecchin, and Osvaldo Simeone (2025). "Adaptive Prediction-Powered AutoEval with Reliability and Efficiency Guarantees". In: *arXiv preprint arXiv:2505.18659*.

Quach, Victor et al. (2023). "Conformal language modeling". In: *arXiv preprint arXiv:2306.10193*.

Ramdas, Aaditya, Jianbo Chen, et al. (2019). "A sequential algorithm for false discovery rate control on directed acyclic graphs". In: *Biometrika* 106.1, pp. 69–86.

References IV

Ramdas, Aaditya and Ruodu Wang (2024). "Hypothesis testing with e-values". In: *arXiv preprint arXiv:2410.23614*.

Schneider, Lennart et al. (2024). "Hyperband-based Bayesian optimization for black-box prompt selection". In: *arXiv preprint arXiv:2412.07820*.

Shafer, Glenn and Vladimir Vovk (2019). *Game-theoretic foundations for probability and finance*. John Wiley & Sons.

Simeone, Osvaldo, Sangwoo Park, and Matteo Zecchin (2025). "Conformal calibration: Ensuring the reliability of black-box ai in wireless systems". In: *arXiv preprint arXiv:2504.09310*.

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams (2012). "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25.

Tishby, Naftali, Fernando Pereira, and William Bialek (2001). "The Information Bottleneck Method". In: *Proc. Allerton Conference on Communication, Control and Computation*.

Xu, Yunpeng et al. (2024). "Two-stage Risk Control with Application to Ranked Retrieval". In: *arXiv preprint arXiv:2404.17769*.

References V

- Zaidi, Abdellatif, Inaki Estella-Aguerri, and Shlomo Shamai (2020). "On the information bottleneck problems: Models, connections, applications and information theoretic views". In: *Entropy* 22.2, p. 151.
- Zecchin, Matteo, Sangwoo Park, and Osvaldo Simeone (2024). "Adaptive learn-then-test: Statistically valid and efficient hyperparameter selection". In: *arXiv preprint arXiv:2409.15844*.
- Zollo, Thomas P et al. (2023). "Prompt risk control: A rigorous framework for responsible deployment of large language models". In: *arXiv preprint arXiv:2311.13628*.