Beyond Best Effort: How to Ensure Reliability in AI-Based Wireless Systems

Osvaldo Simeone
(with thanks to Kfir Cohen, Sangwoo Park, Matteo Zecchin, Jiechen Chen, H. Vincent Poor and Petar Popovski)

King's College London

BalkanCom 2024

AI in Next-Generation Wireless Systems

BalkanCom 2024

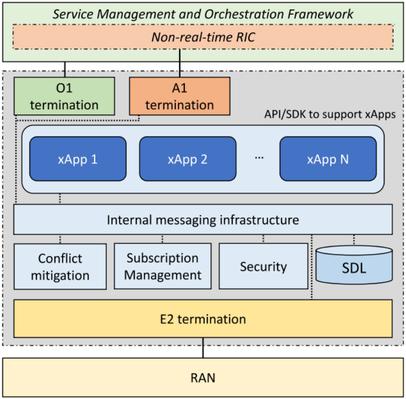
Seventh International Balkan Conference on Communications and Networking

Ljubljana, Slovenia, June 3-6, 2024

AI-Enabled Fluid Connectivity

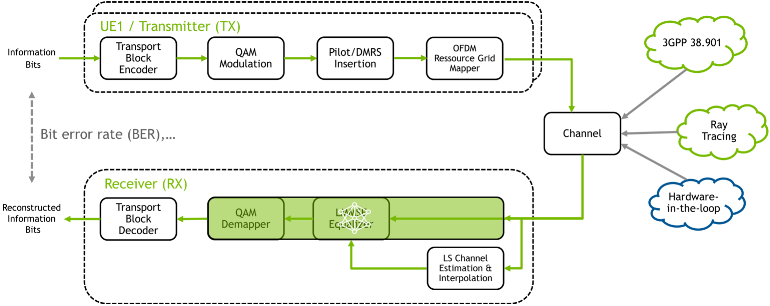
AI in Next-Generation Wireless Systems

- **AI-based "apps"** are key components of next-generation wireless architectures [Bonati et al, '23]



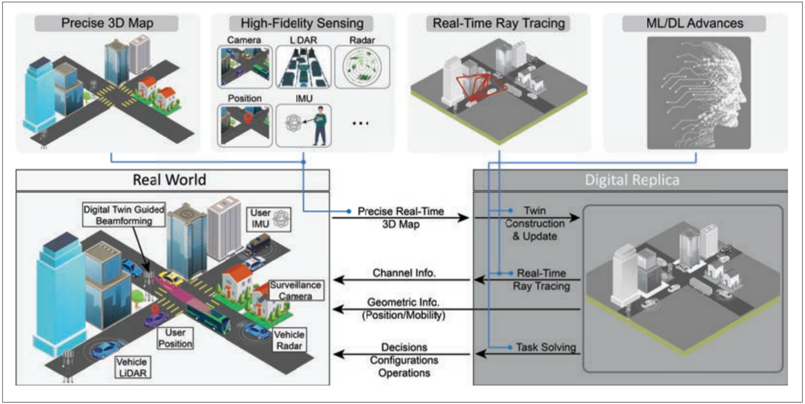
AI in Next-Generation Wireless Systems

- AI apps for **decision making**, e.g., decoding at the PHY [Cammerer et al, '23]



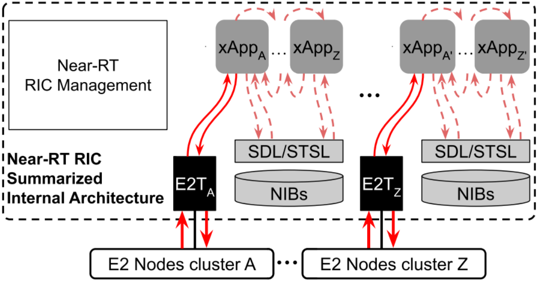
AI in Next-Generation Wireless Systems

- AI apps for **simulation**, e.g., **digital twins** [Alkhateeb et al, '23] [Ruah et al, '23]



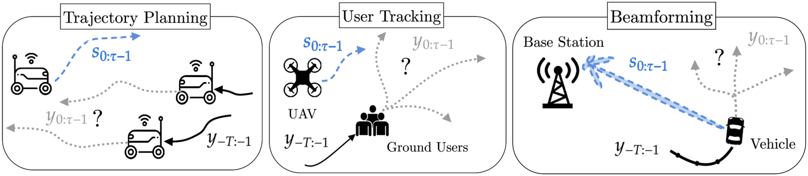
AI in Next-Generation Wireless Systems

- AI apps are typically arranged into **functional graphs**, in which outputs from one app feed into another app [Almeida et al, '24] [Mungari et al '24]



AI in Next-Generation Wireless Systems

- Example: **prediction-based optimization or control** [Lindemann et al, '22] [Zecchin et al, '24]



Collision avoidance **Coverage requirement** **Comm. rate requirement**

AI in Next-Generation Wireless Systems

- Current deployments of AI apps are **best effort**, lacking the theoretical backing of conventional model-based solutions

Given pre-trained AI apps, can we ensure reliability at deployment time (irrespective of the quality of the underlying AI apps)?

1. How to ensure reliability of an AI app used for **decision making**?
2. How to ensure reliability of an AI app used for **prediction-based optimization or control**?
3. How to ensure end-to-end reliability of **composite** AI modules?

AI in Next-Generation Wireless Systems

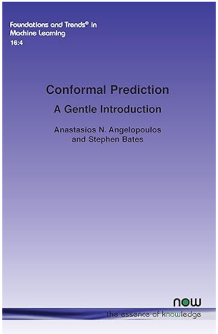
- Current deployments of AI apps are **best effort**, lacking the theoretical backing of conventional model-based solutions

Given pre-trained AI apps, can we ensure reliability at deployment time (irrespective of the quality of the underlying AI apps)?

1. How to ensure reliability of an AI app used for **decision making**?
2. How to ensure reliability of an AI app used for **prediction-based optimization or control**?
3. How to ensure end-to-end reliability of **composite** AI modules?

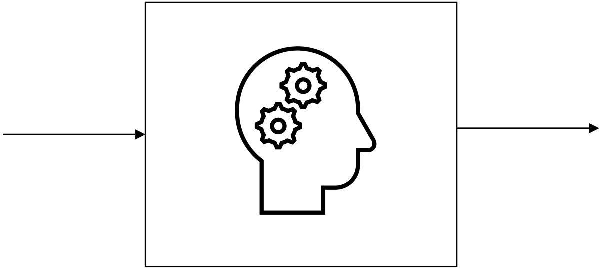
AI in Next-Generation Wireless Systems

1. How to ensure reliability of a single AI app used for **decision making**?
   - **Conformal prediction**
2. How to ensure reliability of an AI app used for **prediction-based optimization or control**?
   - **Conformal risk control**
3. How to ensure end-to-end reliability of **composite** AI modules?
   - **Learn then test**



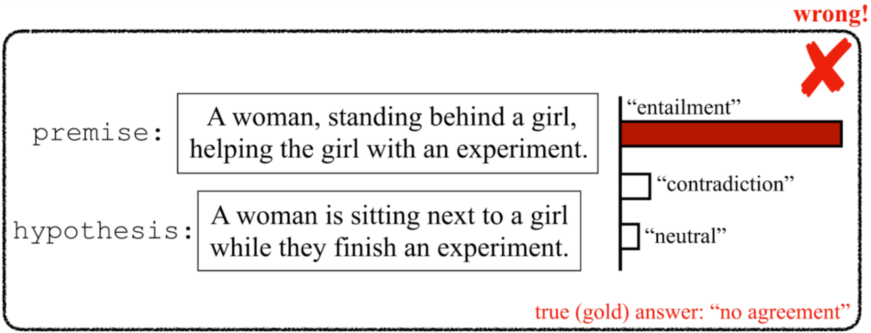
Reliable AI-Based Decision Making

Reliable AI-Decision Making



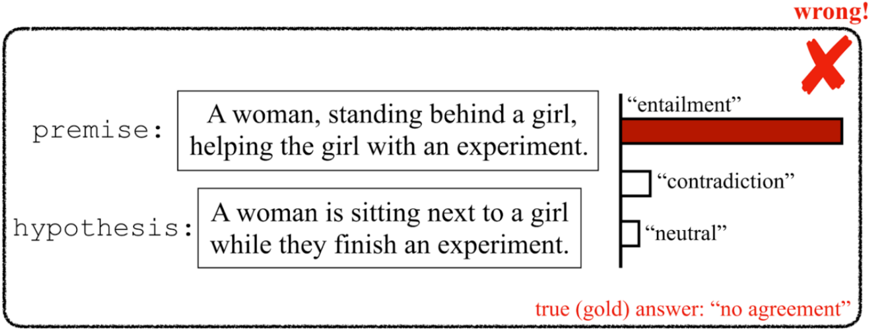
Can We Trust an AI App?

- AI models assign, implicitly or explicitly, a **confidence level** to different possible outputs
- **Reliability via calibration**: If AI confidence = true accuracy $\Longrightarrow$ ask a second opinion, refuse to make a decision, ...
- But AI models are **overconfident**: confidence > true accuracy



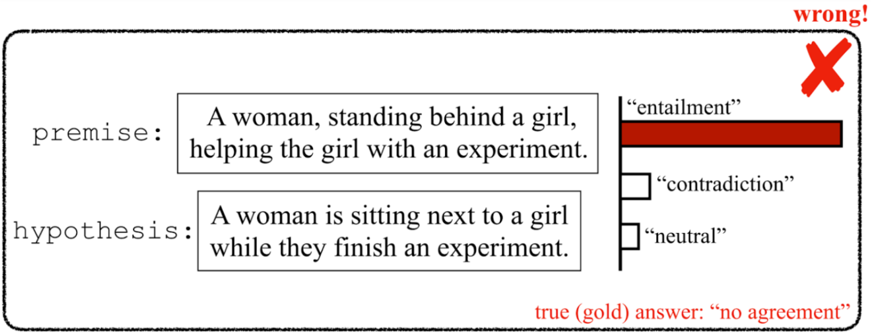
Can We Trust an AI App?

- AI models assign, implicitly or explicitly, a **confidence level** to different possible outputs
- **Reliability via calibration**: If AI confidence = true accuracy $\Longrightarrow$ ask a second opinion, refuse to make a decision, ...
- But AI models are **overconfident**: confidence > true accuracy



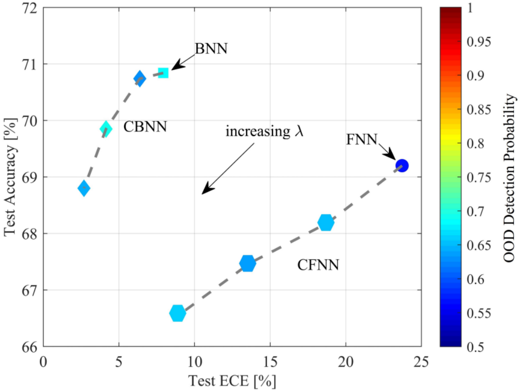
Can We Trust an AI App?

- AI models assign, implicitly or explicitly, a **confidence level** to different possible outputs
- **Reliability via calibration**: If AI confidence = true accuracy $\Longrightarrow$ ask a second opinion, refuse to make a decision, ...
- But AI models are **overconfident**: confidence > true accuracy



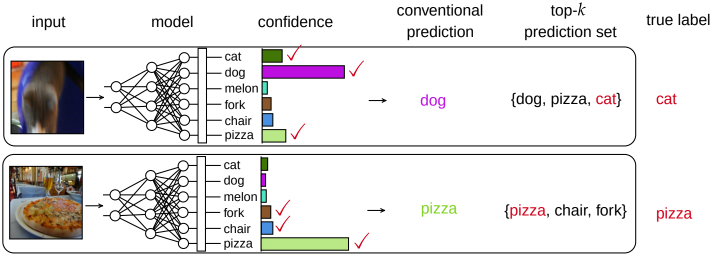
Can We Trust an AI App?

- There is typically a **trade-off** between calibration and accuracy [Huang et al, '24][Tao et al, '23][Kamran and Wien '21]
- (Expected calibration error (ECE) = expected gap between confidence and accuracy)



Reliability via Set Prediction

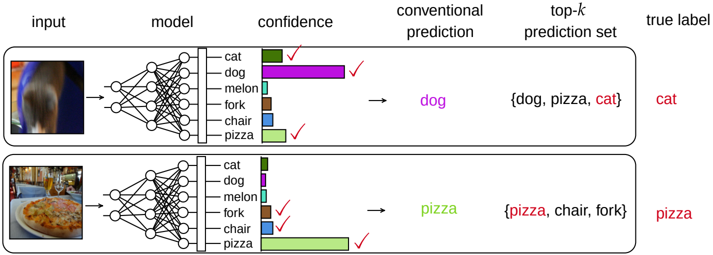
- One way to alleviate this problem is via **top-k set prediction**



- **Reliability via coverage?** No, the predicted set may not contain the true output with some desired probability
- **Non-adaptive** set sizes

Reliability via Set Prediction

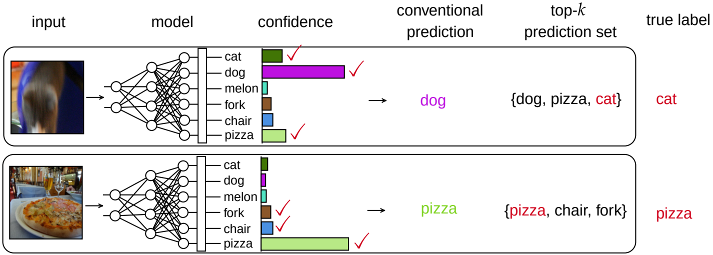
- One way to alleviate this problem is via **top-k set prediction**



- **Reliability via coverage?** No, the predicted set may not contain the true output with some desired probability
- **Non-adaptive** set sizes

Reliability via Set Prediction

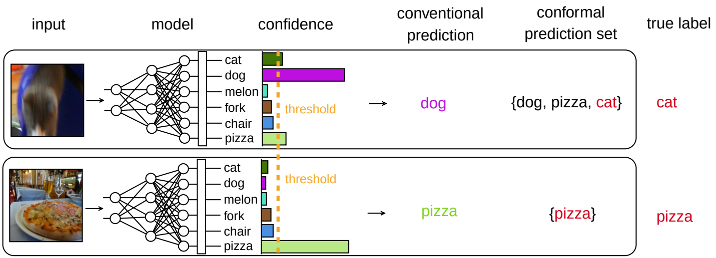
- One way to alleviate this problem is via **top-k set prediction**



- **Reliability via coverage?** No, the predicted set may not contain the true output with some desired probability
- **Non-adaptive** set sizes

Reliability via Set Prediction

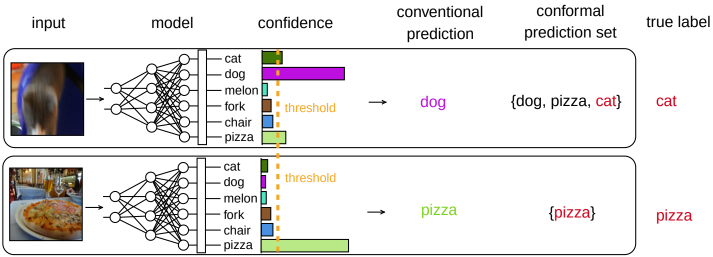
- Alternatively, create prediction sets by including all outputs with confidence **above a threshold**



- **Adaptive** set sizes
- Applicable also to **continuous** outputs (regression)
- Reliability via coverage?

Reliability via Set Prediction

- Alternatively, create prediction sets by including all outputs with confidence **above a threshold**



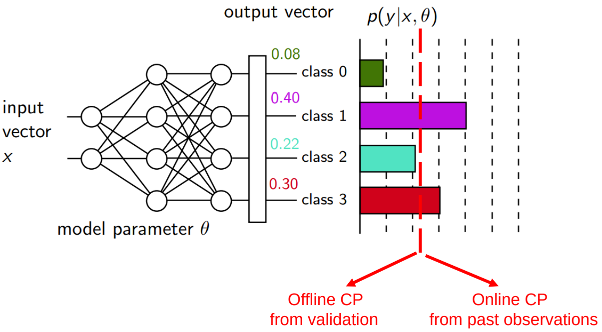
- **Adaptive** set sizes
- Applicable also to **continuous** outputs (regression)
- **Reliability via coverage?**

Conformal Prediction

- **Conformal prediction** guarantees **reliability via coverage**

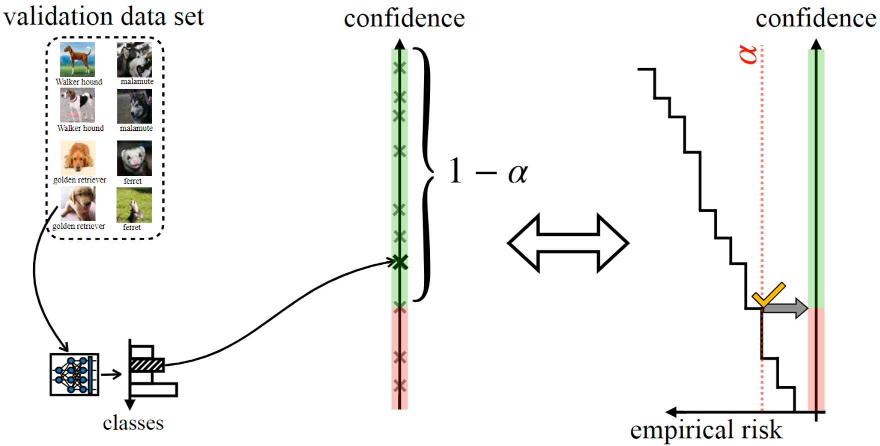
$$\Pr[\text{true output} \in \text{predicted set}] \geq 1 - \alpha$$

for any **user-defined** miscoverage level α



Offline Conformal Prediction

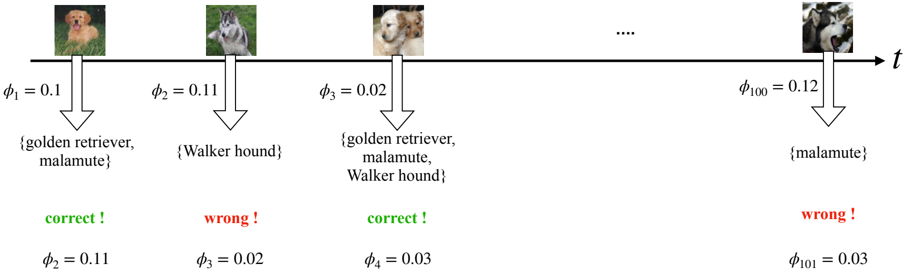
- Selects threshold based on **validation data**
- Guarantees coverage for **exchangeable** data (e.g., i.i.d.) [Vovk et al, '05]



Online Conformal Prediction

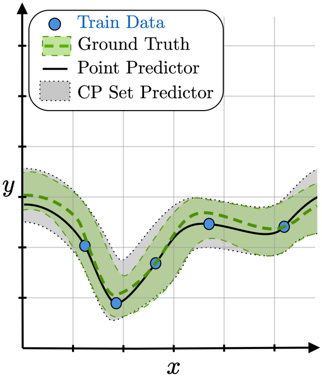
- Adjusts the threshold adaptively based on past errors to minimize the regret [Gibbs and Candes,' 21] [Feldman et al '22]
- Guarantees coverage on average **over time** (see also [Angelopoulos et al '24])

$$\phi_{t+1} = \phi_t + \gamma(\alpha - \mathrm{err}_t)$$



Calibration vs. Informativeness

- **Calibration is guaranteed**, irrespective of the quality of the AI model
- But, if the AI model is poor, the resulting predicted set may be **uniformative** [Zecchin et al, '24] [Park et al '24]

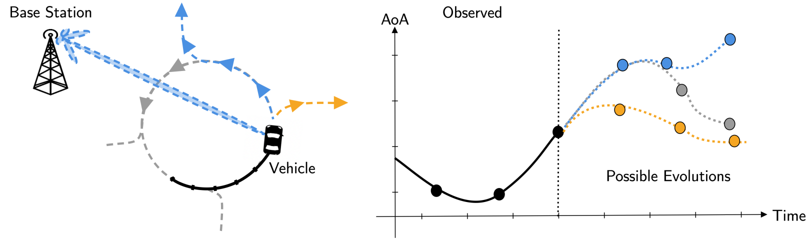


Applications

- Conformal prediction can be **wrapped** around the use of any AI app to ensure **reliability via coverage**
- Examples of use cases [Cohen et al, '23]
  - List demodulation, list decoding
  - Modulation classification
  - Channel prediction
  - Device tracking

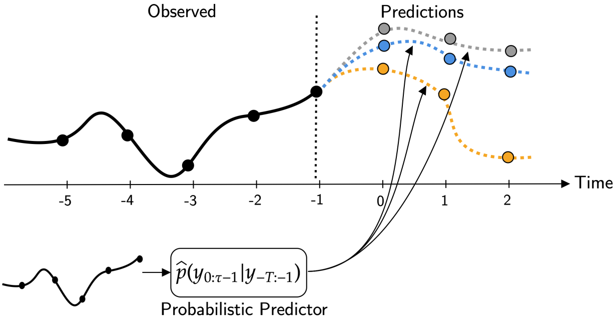
Example

- Predict the **angle of arrival (AoA)** of the line-of-sight path between a base station and a moving vehicle
- The evolution of $y_{0:\tau-1}$ conditioned on $y_{-T:-1}$ is **multimodal** due to the unknown vehicle future route



Time Series Prediction

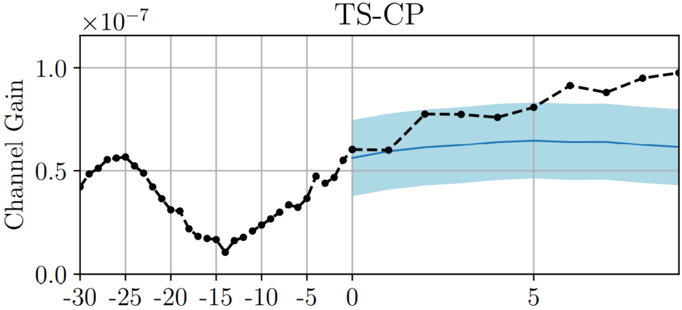
- Provided the past T samples of a time series $y_{-T}, \ldots, y_{-1}$, predict the next τ samples $y_0, \ldots, y_{\tau-1}$
- Assume the availability of a **probabilistic sequence model** (e.g., transformer) $\hat{p}(y_{0:\tau-1}|y_{-T:-1})$
- We wish to obtain a **reliable set predictor** from an **arbitrary** probabilistic sequence model



Time Series Prediction

- Previous work used a **single prediction** $\hat{y}_{0:\tau-1}$ to evaluate confidence as [Lindemann et al '23]

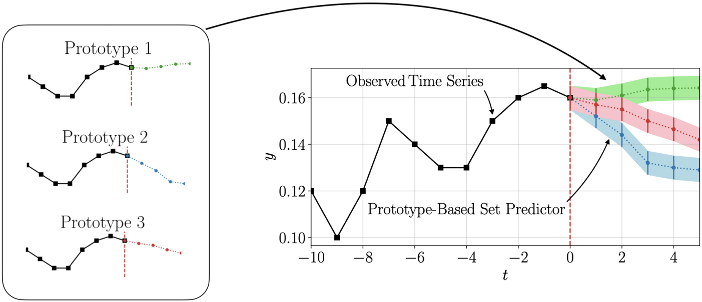
$$-||y_{0:\tau-1} - \hat{y}_{0:\tau-1}||$$



Time Series Prediction

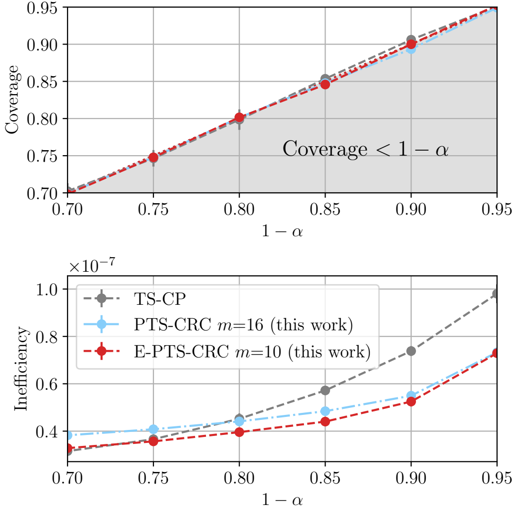
- Sample a number of **prototypes** $\mathcal{P}^m = \{\hat{y}^i_{0:\tau-1}\}_{i=1}^m$ from the probabilistic model $\hat{p}(y_{0:\tau-1}|y_{-T:-1})$
- Use the **confidence score** [Zecchin et al, '24]

$$- \min_{\hat{y}_{0:\tau-1} \in \mathcal{P}^m} ||y_{0:\tau-1} - \hat{y}_{0:\tau-1}||$$



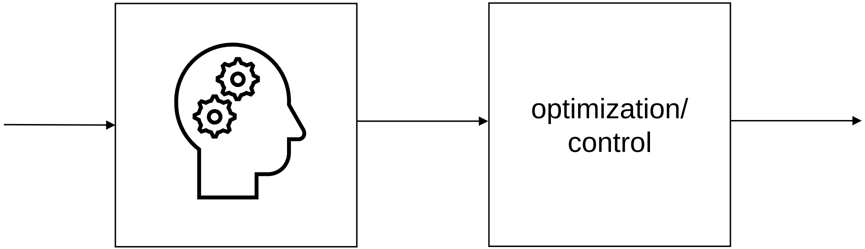
Time Series Prediction

- Channel prediction: The performance depends on the predictor and on the function used to evaluate the confidence



Reliable AI for Prediction-Based Optimization and Control

Reliable AI for Prediction-Based Optimization and Control



Prediction-Based Optimization

- Consider constrained optimization problems of the form

$$\begin{aligned} &\underset{x}{\text{maximize}}\ U(x) && \text{(utility)} \\ &\text{subject to}\ \mathsf{E}_y[R(x,y)] \leq \alpha && \text{(reliability constraint)} \end{aligned}$$

where the **target variable** y is **unknown** and must be predicted

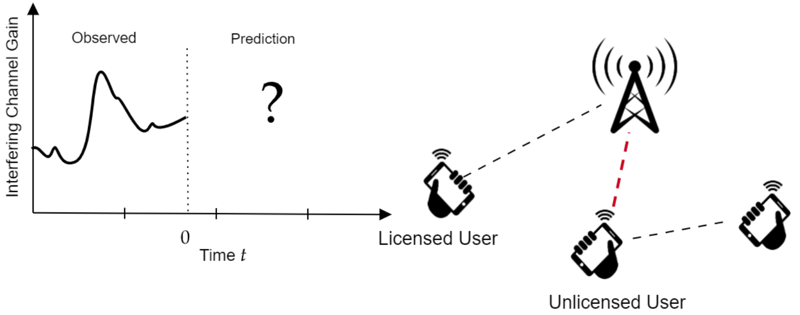
Example

- **Power allocation** for **unlicensed** user subject to an average **interference** constraint for a licensed user:

$$\underset{x}{\text{maximize}}\ U(x) \qquad \text{(unlicensed user rate)}$$

$$\text{subject to } \mathrm{E}_y[R(x,y)] \leq \alpha \qquad \text{(interference constraint)}$$

- The target variable y is the channel gain of the licensed user



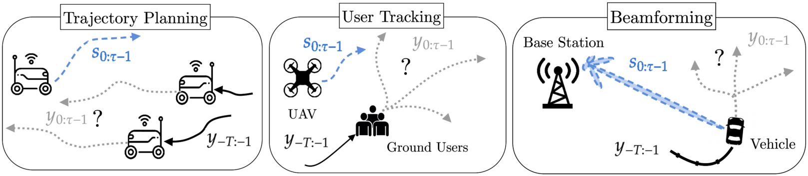
Prediction-Based Control

- Choose a sequence of **actions** $x_{0:\tau-1}$ to control the **state** $s_{0:\tau-1}$ of a dynamical system so that

$$\underset{x_{0:\tau-1}}{\text{maximize}} \; U(s_{0:\tau-1}) \qquad \text{(utility)}$$

$$\text{subject to } \mathbb{E}_{y_{0:\tau-1}}[R(s_{0:\tau-1}, y_{0:\tau-1})] \leq \alpha \qquad \text{(reliability constraint)}$$

for some **unknown target process** $y_{0:\tau-1}$



Collision avoidance **Coverage requirement** **Comm. rate requirement**

Prediction-Based Optimization and Control

- A conventional **best-effort** prediction-based optimization or control would replace the target with a prediction $\hat{y}$

$$
\begin{aligned}
&\underset{x}{\text{maximize}}\ U(x) && \text{(utility)}\\
&\text{subject to } R(x,\hat{y}) \leq \alpha && \text{(reliability constraint)}
\end{aligned}
$$

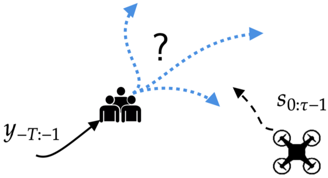
- However, this does not guarantee reliability

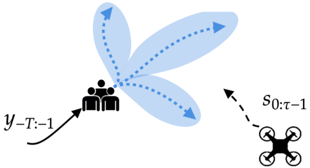
Prediction-Based Control and Control

- With a conformal prediction-based predicted set, the **average** constraint can be turned into a **worst-case** constraint [Lindemann et al '23] [Zecchin et al '24]

$$\begin{aligned} &\underset{x}{\text{maximize}}\ U(x) \\ &\text{subject to} \max_{y \in \text{predicted set}} R(x,y) \leq \beta \end{aligned}$$

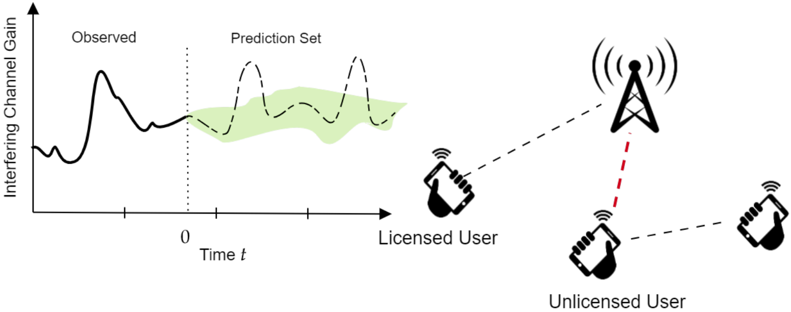
where β is a function of α





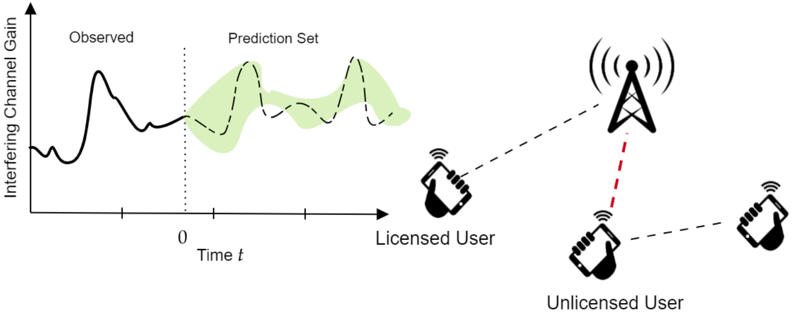
Example

- **Reliability via coverage** may not provide an ideal solution when used for prediction-based optimization or control



Example

$$L(\text{y}, \text{predicted set}) = 1(y \notin \text{predicted set}) \quad \underbrace{\ell(y)}_{\text{increasing}}$$



Conformal Risk Control

- **Conformal risk control** generalizes conformal prediction by ensuring the reliability requirement [Angelopoulos, et al '22] [Cohen et al '24]

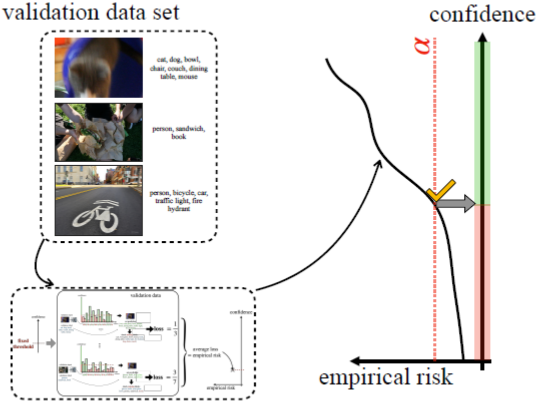
$$\mathrm{E}[L(\text{true output}, \text{predicted set})] \leq \alpha$$

  as long as the loss function L is **decreasing** as the predicted set grows
- Note that the conformal prediction miscoverage loss is a special case:

$$L(\text{true output}, \text{predicted set}) = 1(\text{true output} \notin \text{predicted set})$$

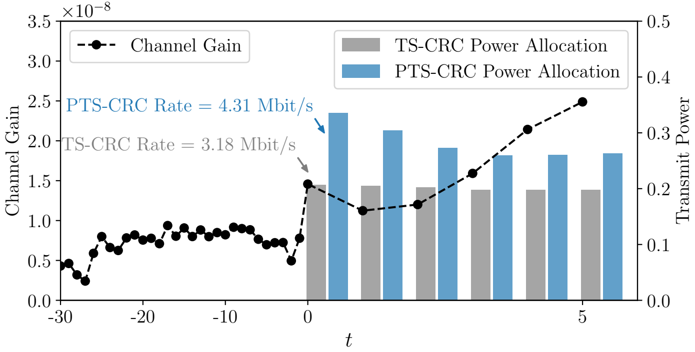
Conformal Risk Control

- As for conformal prediction, conformal risk control can be implemented **offline** or **online**



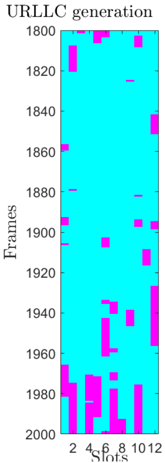
Example

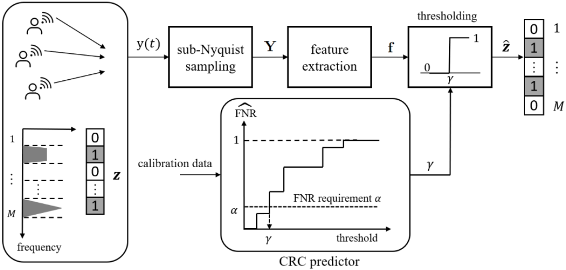
- The performance level in terms of utility depends on the quality of the predictor and on the confidence function



Example

- **Proactive scheduling** for URLLC [Cohen, et al '23] and spectrum sensing [Lee et al '24]

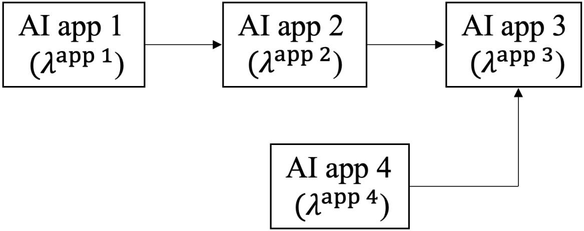




Reliable Composition of AI Models

Reliable Composition of AI Models

- Graph of **pre-trained AI apps**, each with free **hyperparameters** (e.g., temperature, module selection, fine-tuning learning rate, complexity-fidelity trade-off)



- How to select a hyperparameter vector λ so as to guarantee **end-to-end reliability** (with minimal data requirements)?

$$\Pr[R(\lambda) \leq \alpha] \geq 1 - \delta$$

Reliable Composition of AI Models

- Conformal risk control is not directly applicable, since it applies to a **single** hyperparameter λ and to a monotonic loss $R(\lambda)$
- **Conventional** approach: Use **validation** data to estimate the risk as $\hat{R}(\lambda)$, and then choose vector λ as

$$\underset{\lambda}{\text{minimize}}\ \hat{R}(\lambda)$$

- This may lead to **overfitting**, failing to satisfy end-to-end reliability
- Furthermore, it is not applicable if evaluating requires $\hat{R}(\lambda)$ real-world testing

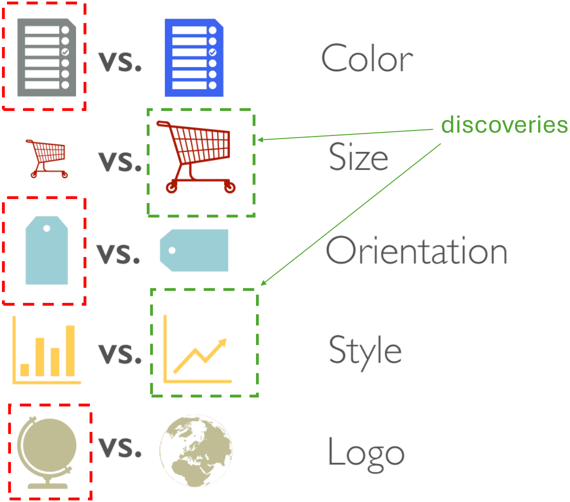
Multi-Hypothesis Testing

- Hyperparameter selection as **scientific discovery** or **A/B testing** (multi-hypothesis testing) [Angelopoulos et al '22]

Multi-Hypothesis Testing

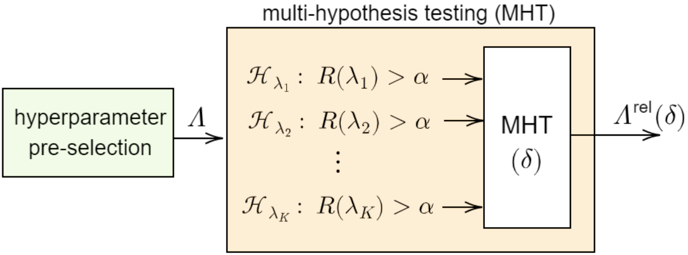
- **Family-wise error rate (FWER) control**:

$$\Pr[\text{no false discovery}] \geq 1 - \delta$$



Learn Then Test

- **Learn then test:** Test one hypothesis for each candidate hyperparameter vector λ [Angelopoulos et al '22]
- FWER guarantees that all selected hyperparameters are reliable with probability $\geq 1-\delta$



$$\Pr\left[\max_{\lambda\in\Lambda^{\text{rel}}(\delta)} R(\lambda) \leq \alpha\right] \geq 1-\delta$$

Learn Then Test via Fixed-Sequence Testing

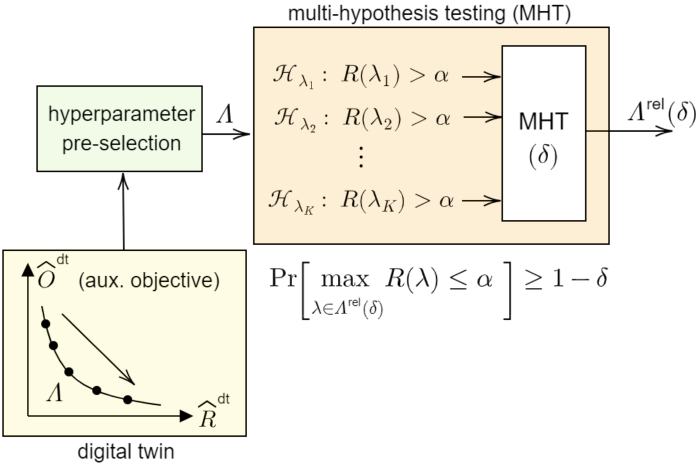
- **Input:** Pre-selected subset of hyperparameters Λ
- **Order** the hyperparameter in any way
- **Set** $j = 1$
- **Repeat** until reliability check is violated
  - **Estimate** risk as $\hat{R}(\lambda^{(j)})$ based on N validation data points
  - **Reliability check:**

$$\hat{R}(\lambda^{(j)}) \leq \alpha - \sqrt{\frac{-\ln(\delta)}{2N}}$$

  - If checked, add $\lambda^{(j)}$ to $\Lambda^{\text{rel}}(\delta)$
  - $j = j + 1$

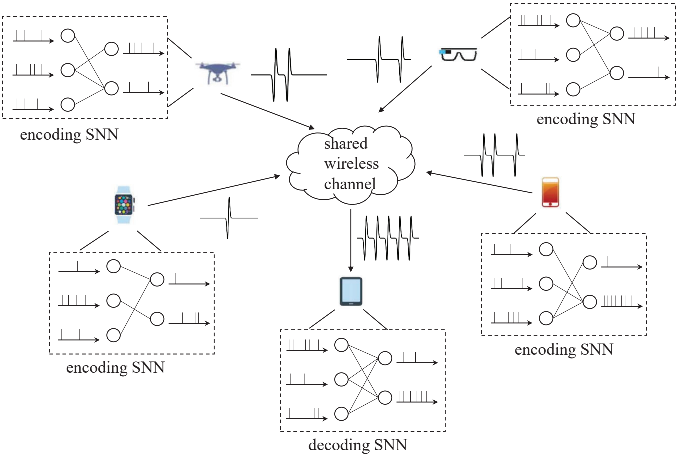
Digital Twin-Based Pareto Testing

- How to pre-select hyperparameters and the testing order? **Pareto testing** [Laufer-Goldshtein et al '22] via a **digital twin** [Chen et al '24]



$$\Pr\left[\max_{\lambda \in \Lambda^{\text{rel}}(\delta)} R(\lambda) \leq \alpha\right] \geq 1 - \delta$$

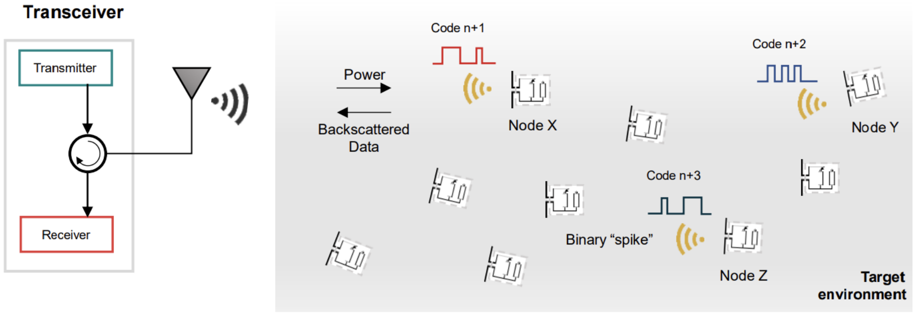
Neuromorphic Communications



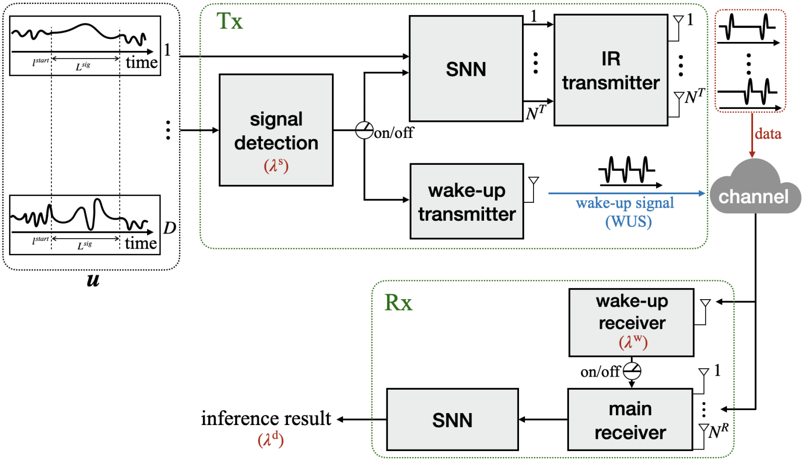
- **Neuromorphic communication** integrates neuromorphic sensing, impulse radio communications, and neuromorphic computing [Skatchkovsky et al '21] [Chen et al '23]

Neuromorphic Communications

- **Hardware implementation** showcases potential **scaling** to thousands of nodes [Lee at al '24]



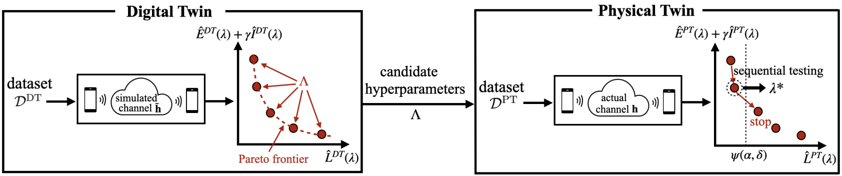
Neuromorphic Communications



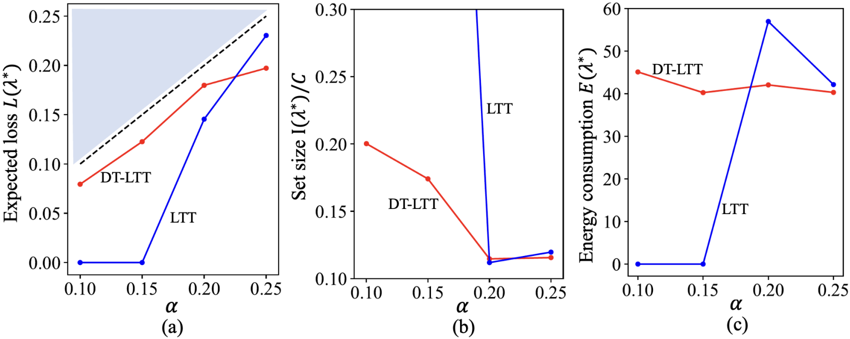
- Neuromorphic communications with a **wake-up radio** [Chen et al '24]
- **Hyperparameters:** thresholds for sensing, wake-up radio signal detection, and decision making

Neuromorphic Communications

- Based on simulations, the digital twin determines an estimated **energy-risk Pareto boundary** [Laufer-Goldshtein et al '22]
- Lean the test is applied sequentially starting from the lowest estimated risk



Digital Twin-Based Learn Then Test



Conclusions

Conclusions

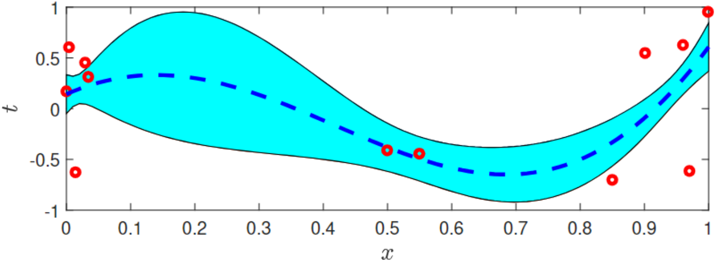
- Recent advances in statistics enable the **post-hoc calibration** of pre-trained AI model, ensuring reliability for
  - decision making
  - model-based optimization and prediction
  - composition of AI models
- Conformal prediction, conformal risk control, and learn then test are easily wrapped around existing AI models
- Directions for research:
  - In-depth exploration of other use cases for wireless systems
  - Information-theoretic analysis
  - Decentralized implementations

Acknowledgments

This work was supported by the European Union's Horizon Europe project CENTRIC (101096379), by an Open Fellowship of the EPSRC (EP/W024101/1), by the EPSRC project EP/X011852/1, and by Project REASON, a UK Government funded project under the Future Open Networks Research Challenge (FONRC) sponsored by the Department of Science Innovation and Technology (DSIT).

Extra Slides

Reliability via Set Prediction



Reliability via Set Prediction

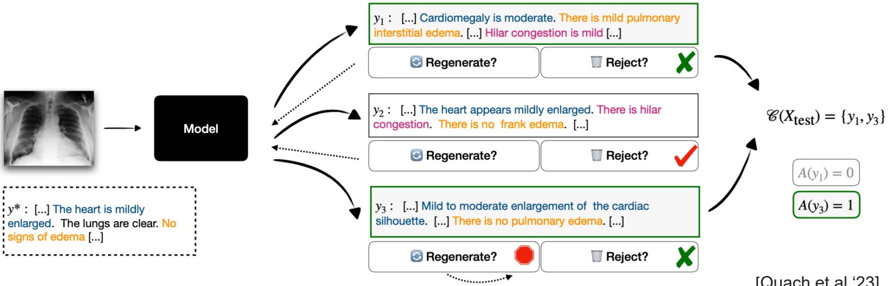
{ fox squirrel 0.99 }

{ fox squirrel 0.82, gray fox 0.03, bucket 0.02, rain barrel 0.02 }

{ marmot 0.30, fox squirrel 0.22, mink 0.18, weasel 0.16, beaver 0.03, polecat 0.01 }

[Angelopoulos et al '23]

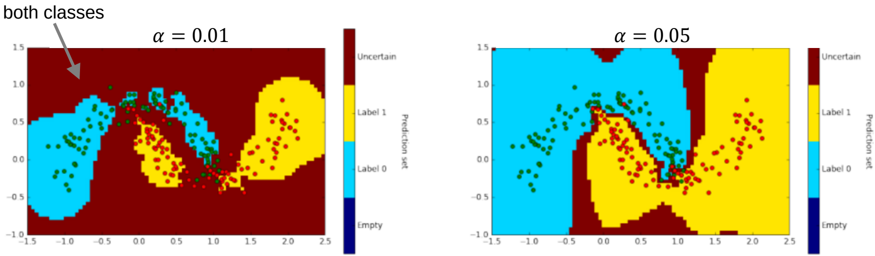
Reliability via Set Prediction



[Quach et al '23]

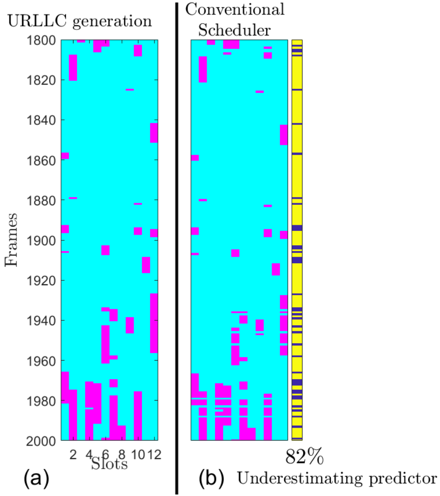
Offline Conformal Prediction

- Example [Toccaceli, '19]



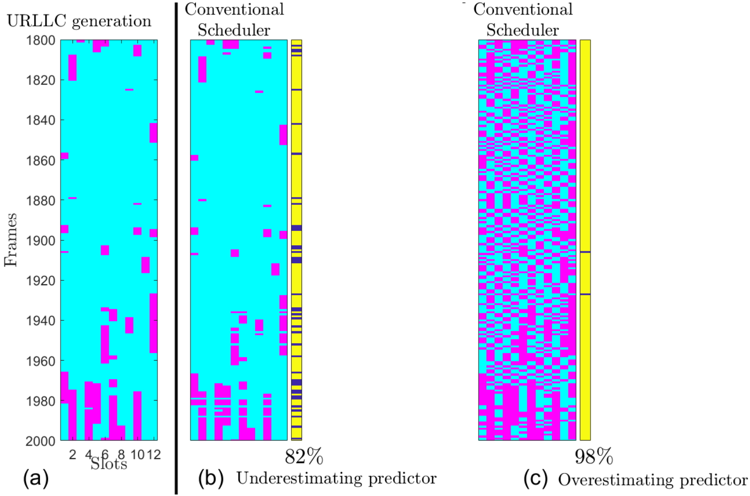
Example

- Using conventional prediction-based optimization, the output of the scheduler may be **unreliable**...



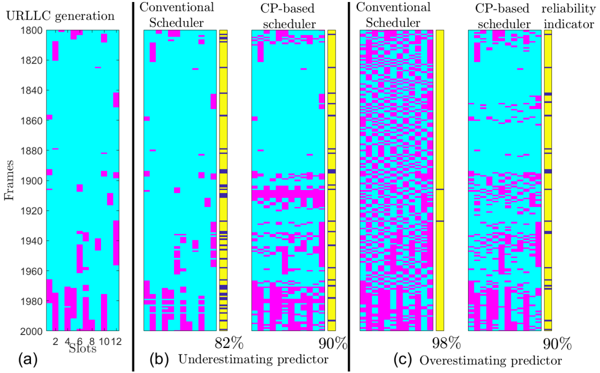
Example

- ... or **inefficient**

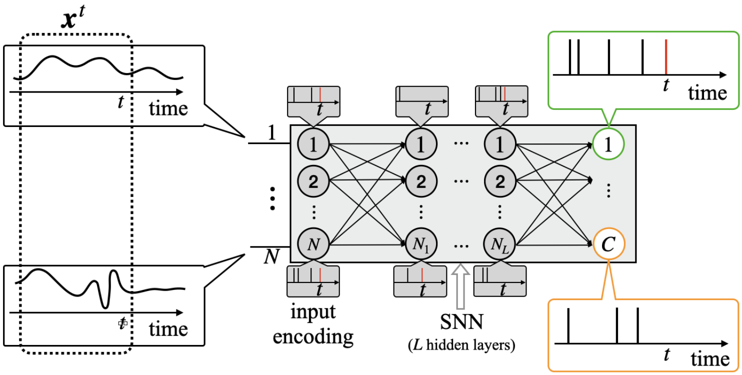


Example

- Using conformal risk control guarantees **reliable and efficient** resource allocation, irrespective of the calibration of the predictor [Cohen et al '23]



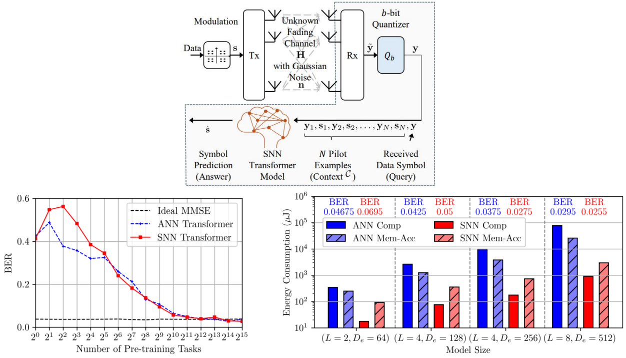
Neuromorphic Computing



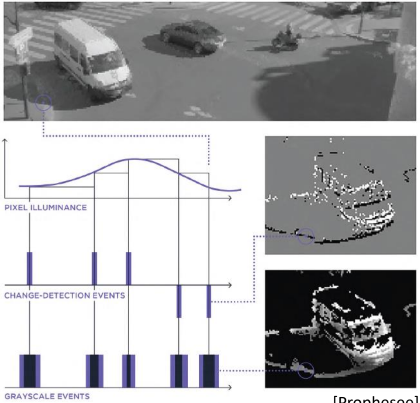
- **Neuromorphic computing** implements **spiking neural networks (SNNs)**
- SNNs leverage **sparsity** to reduce processing energy [Davies et al '23]

Neuromorphic Computing

- E.g., neuromorphic transformer for in-context learning for MIMO demodulation [Song et al '24]



Neuromorphic Sensing



[Prophesee]

- Neuromorphic computing is particularly effective when implemented on data captured by **neuromorphic sensors**, such as DVS cameras

Example

- **Beam tracking** using vision-based prediction [Imran et al '24]

$$L(\text{true mask}, \text{predicted set}) = \text{fraction of missed pixels}$$