Towards Reliable, Efficient, and Robust AI for Communications

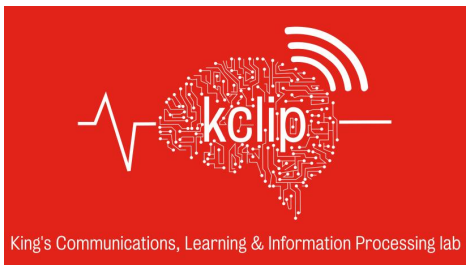
Osvaldo Simeone

King's College London

2022 IEEE SPS - EURASIP Summer School
30/8/2022

King's Communications, Learning & Information Processing Lab

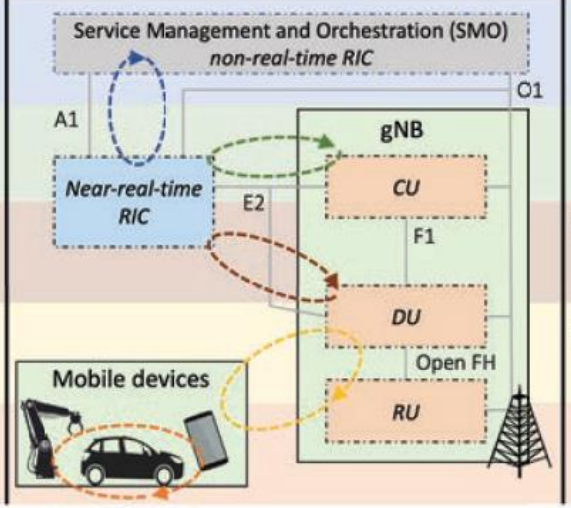
- Joint work with K. Cohen and S. Park (KCL), as well as M. Zecchin, M. Kountouris, and D. Gesbert (EURECOM)



Motivation

The Role of AI in 6G & Beyond

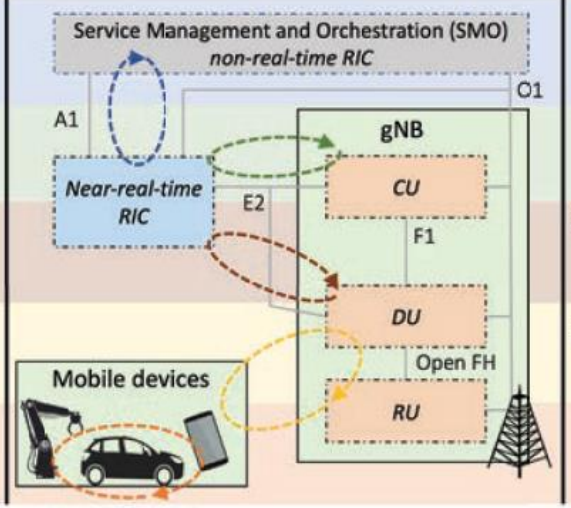
- AI is playing an increasingly significant role in **engineering**.
- As a case in point, **next-generation communication systems** will leverage AI at all layers of the protocol stack.
- This imposes **new requirements** on the performance of AI.

Control and learning objective	Scale	Input data	Timescale	Architecture	Challenges and limitations
Policies, models, slicing	> 1000 devices	Infrastructure-level KPIs	Non-real-time > 1 s		Orchestration of very many near-real-time RICs and CUs/DUs/RUs
User Session Management e.g., load balancing, handover	> 100 devices	CU-level KPIs e.g., number of sessions, PDCP traffic	Near-real-time 10-1000 ms		Process streams from multiple CUs and sessions
Medium Access Management e.g., scheduling policy, RAN slicing	> 100 devices	MAC-level KPIs e.g., PRB utilization, buffering	Near-real-time 10-1000 ms		Operate at small time scales, make decisions involving several DUs/UEs
Radio Management e.g., resource scheduling, beamforming	~10 devices	MAC/PHY-level KPIs e.g., PRB utilization, channel estimation	Real-time < 10 ms		Deployment of AI/ML models at the DU is not supported
Device DL/UL Management e.g., modulation, interference, blockage detection	1 device	I/Q samples	Real-time < 1 ms		Require device- and/or RU-level standardization

[Bonati et al '21]

The Role of AI in 6G & Beyond

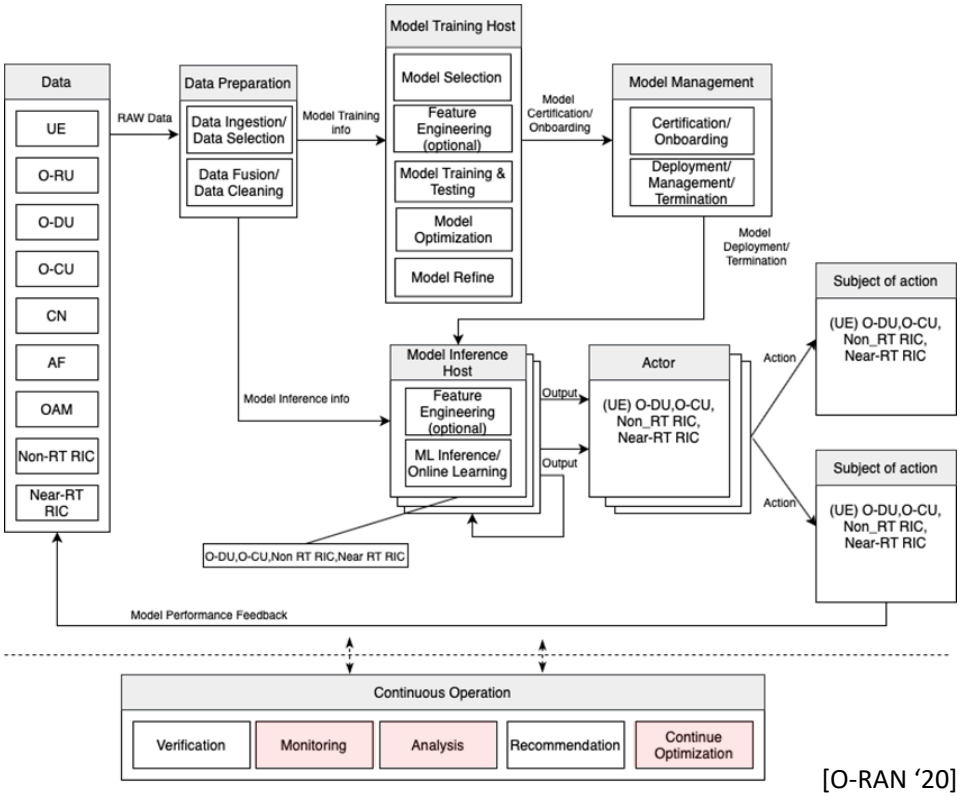
- AI is playing an increasingly significant role in **engineering**.
- As a case in point, **next-generation communication systems** will leverage AI at all layers of the protocol stack.
- This imposes **new requirements** on the performance of AI.

Control and learning objective	Scale	Input data	Timescale	Architecture	Challenges and limitations
Policies, models, slicing	> 1000 devices	Infrastructure-level KPIs	Non-real-time > 1 s		Orchestration of very many near-real-time RICs and CUs/DUs/RUs
User Session Management e.g., load balancing, handover	> 100 devices	CU-level KPIs e.g., number of sessions, PDCP traffic	Near-real-time 10-1000 ms		Process streams from multiple CUs and sessions
Medium Access Management e.g., scheduling policy, RAN slicing	> 100 devices	MAC-level KPIs e.g., PRB utilization, buffering	Near-real-time 10-1000 ms		Operate at small time scales, make decisions involving several DUs/UEs
Radio Management e.g., resource scheduling, beamforming	~10 devices	MAC/PHY-level KPIs e.g., PRB utilization, channel estimation	Real-time < 10 ms		Deployment of AI/ML models at the DU is not supported
Device DL/UL Management e.g., modulation, interference, blockage detection	1 device	I/Q samples	Real-time < 1 ms		Require device- and/or RU-level standardization

[Bonati et al '21]

Requirements for AI in Engineering

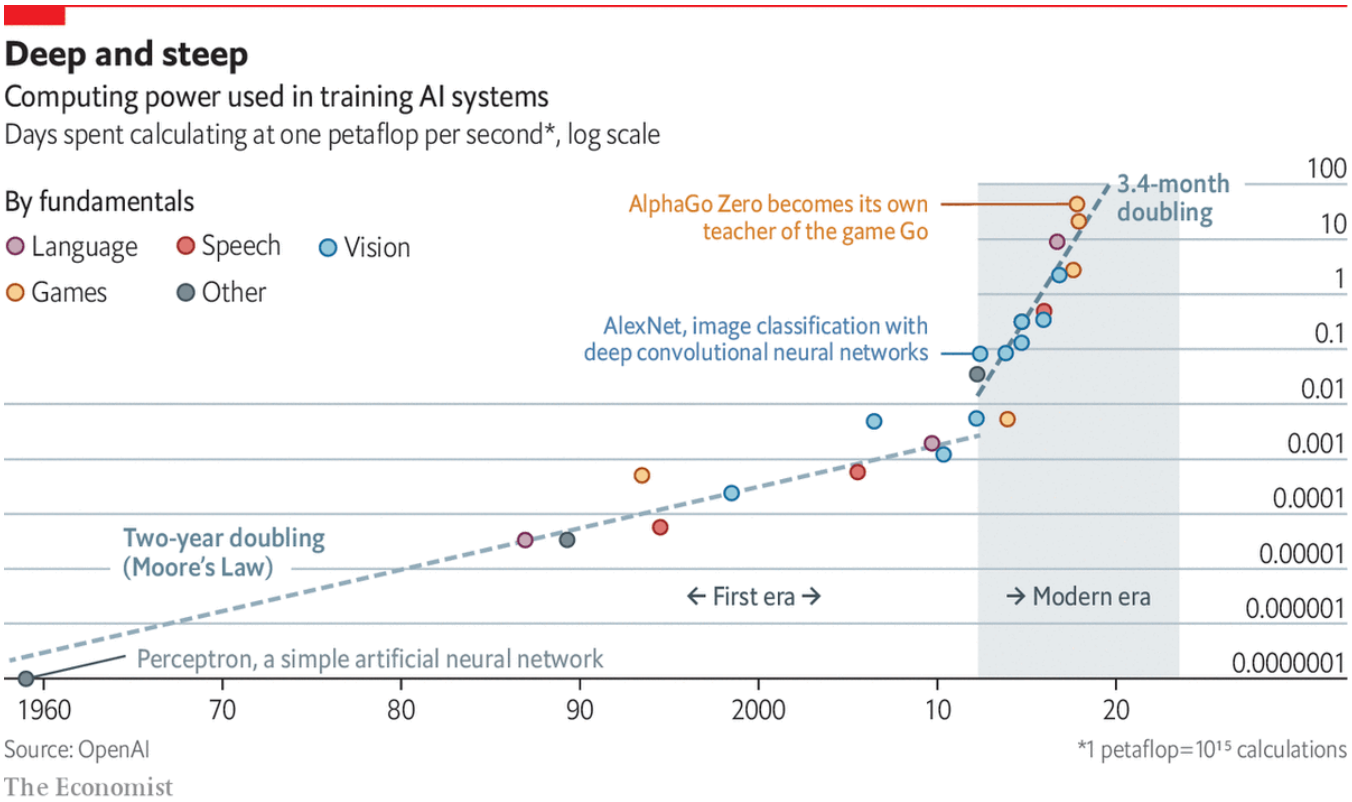
- In many engineering problems, **accuracy** should be weighted against:
- 1) **reliability**, or **calibration**, providing a faithful quantification of the uncertainty of the AI's decisions, e.g., for **monitoring**;
- 2) **sample efficiency**, enabling fast **adaptation**



[O-RAN '20]

Requirements for AI in Engineering

- 3) **robustness** to deviations from design assumptions
- 4) **hardware efficiency**, facilitating implementation on mobile/edge devices



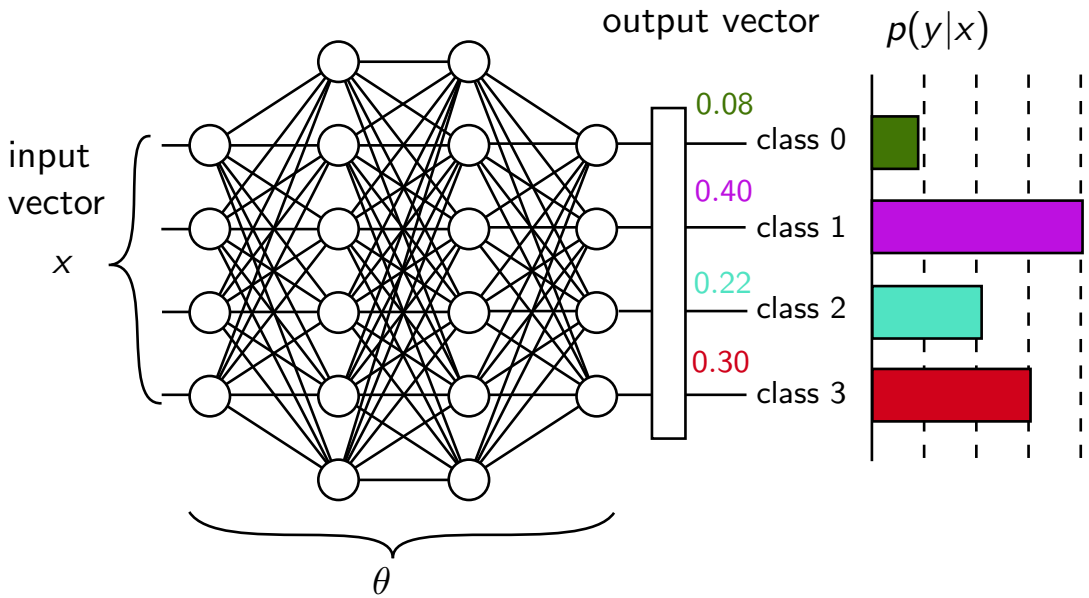
This Talk

- **Reliable** AI:
  - Bayesian learning
- **Reliable and robust** AI:
  - Robust Bayesian learning
- **Sample-efficient** AI:
  - Meta-learning
- **Reliable and sample-efficient** AI:
  - Bayesian meta-learning
- **Hardware-efficient** AI:
  - Neuromorphic computing
  - Quantum computing

Reliable AI:
Bayesian Learning

Predictive Uncertainty

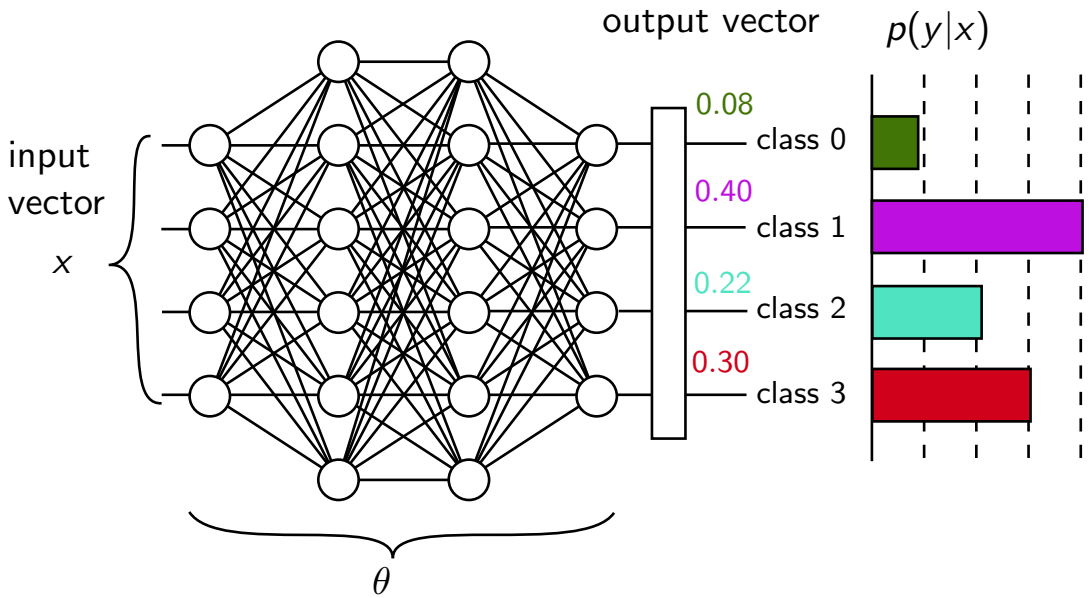
- Discriminative **probabilistic models** $p(y|x, \theta)$ output hard decisions and confidence levels.



- **Hard decision**: $\hat{y}(x|\theta) = 1$ (class with largest score)
- **Confidence level**: $\text{conf}(x|\theta) = p(\hat{y}(x|\theta)|x, \theta) = 0.4$ (self reported)
- How **reliable** is the estimate of predictive uncertainty reported by the model?

Predictive Uncertainty

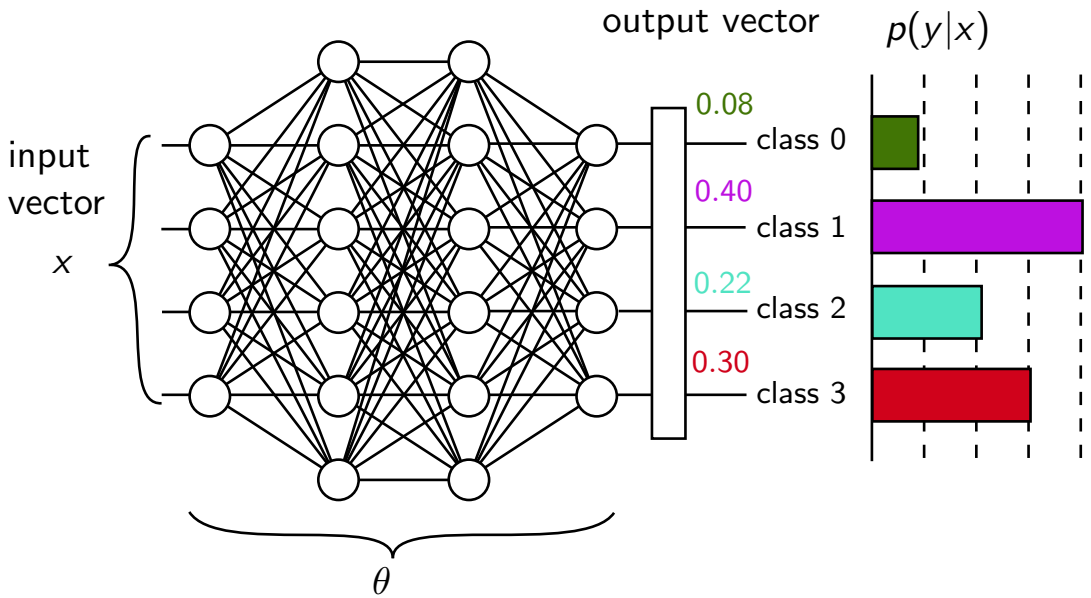
- Discriminative **probabilistic models** $p(y|x,\theta)$ output hard decisions and confidence levels.



- **Hard decision**: $\hat{y}(x|\theta) = 1$ (class with largest score)
- **Confidence level**: $\text{conf}(x|\theta) = p(\hat{y}(x|\theta)|x,\theta) = 0.4$ (self reported)
- How **reliable** is the estimate of predictive uncertainty reported by the model?

Predictive Uncertainty

- Discriminative **probabilistic models** $p(y|x,\theta)$ output hard decisions and confidence levels.



- **Hard decision**: $\hat{y}(x|\theta) = 1$ (class with largest score)
- **Confidence level**: $\text{conf}(x|\theta) = p(\hat{y}(x|\theta)|x,\theta) = 0.4$ (self reported)
- How **reliable** is the estimate of predictive uncertainty reported by the model?

Quantifying Calibration

- Assume that the data is generated from some ground-truth **population distribution** $P(x, y)$.
- In practice, this can be estimated based on validation/ test data.
- The **accuracy** of a probabilistic model $p(y|x, \theta)$ on input x is

$$\mathrm{acc}(x|\theta) = P(\hat{y}(x|\theta)|x)$$

- A probabilistic model $p(y|x, \theta)$ is **reliable**, or **well calibrated**, if

$$\mathrm{conf}(x|\theta) \approx \mathrm{acc}(x|\theta),$$

or

$$\text{confidence level} \approx \text{accuracy}$$

Quantifying Calibration

- Assume that the data is generated from some ground-truth **population distribution** $P(x, y)$.
- In practice, this can be estimated based on validation/ test data.
- The **accuracy** of a probabilistic model $p(y|x, \theta)$ on input x is

$$\mathrm{acc}(x|\theta) = P(\hat{y}(x|\theta)|x)$$

- A probabilistic model $p(y|x, \theta)$ is **reliable**, or **well calibrated**, if

$$\mathrm{conf}(x|\theta) \approx \mathrm{acc}(x|\theta),$$

or

$$\text{confidence level} \approx \text{accuracy}$$

Quantifying Calibration

- Assume that the data is generated from some ground-truth **population distribution** $P(x, y)$.
- In practice, this can be estimated based on validation/ test data.
- The **accuracy** of a probabilistic model $p(y|x, \theta)$ on input x is

$$\mathrm{acc}(x|\theta) = P(\hat{y}(x|\theta)|x)$$

- A probabilistic model $p(y|x, \theta)$ is **reliable**, or **well calibrated**, if

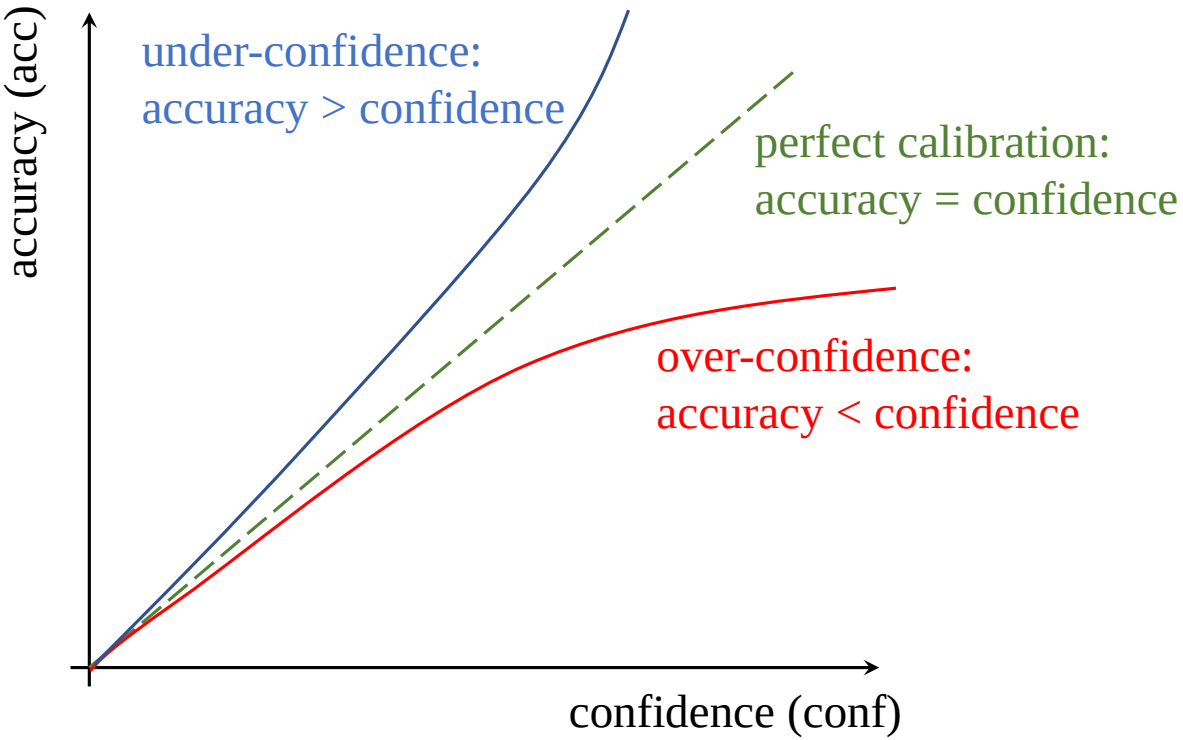
$$\mathrm{conf}(x|\theta) \approx \mathrm{acc}(x|\theta),$$

or

$$\text{confidence level} \approx \text{ accuracy}$$

Reliability Diagrams

- Given a discriminative model $p(y|x, \theta)$, how can we quantify its reliability?
- **Reliability diagrams** plot accuracy vs. confidence, providing a visual depiction of calibration performance.[1]



[1] C. Guo, et al, "On calibration of modern neural networks," ICML 2017.

Reliability Diagrams

- Mathematically, reliability diagrams are based on the evaluation of the accuracy

$$\mathrm{acc}(c|\theta) = P(\hat{y}(x|\theta)|\mathrm{conf}(x|\theta) = c)$$

  as a function of the confidence level c.
- A **well-calibrated** predictor has $\mathrm{acc}(c|\theta) \approx c$; it is **over-confident** if $\mathrm{acc}(c|\theta) < c$; and **under-confident** if $\mathrm{acc}(c|\theta) > c$.
- In practice, the reliability diagram $\mathrm{acc}(c|\theta)$ vs. c is plotted by binning the confidence levels c.

Reliability Diagrams

- Mathematically, reliability diagrams are based on the evaluation of the accuracy

$$\mathrm{acc}(c|\theta) = P(\hat{y}(x|\theta)|\mathrm{conf}(x|\theta) = c)$$

  as a function of the confidence level c.
- A **well-calibrated** predictor has $\mathrm{acc}(c|\theta) \approx c$; it is **over-confident** if $\mathrm{acc}(c|\theta) < c$; and **under-confident** if $\mathrm{acc}(c|\theta) > c$.
- In practice, the reliability diagram $\mathrm{acc}(c|\theta)$ vs. c is plotted by binning the confidence levels c.

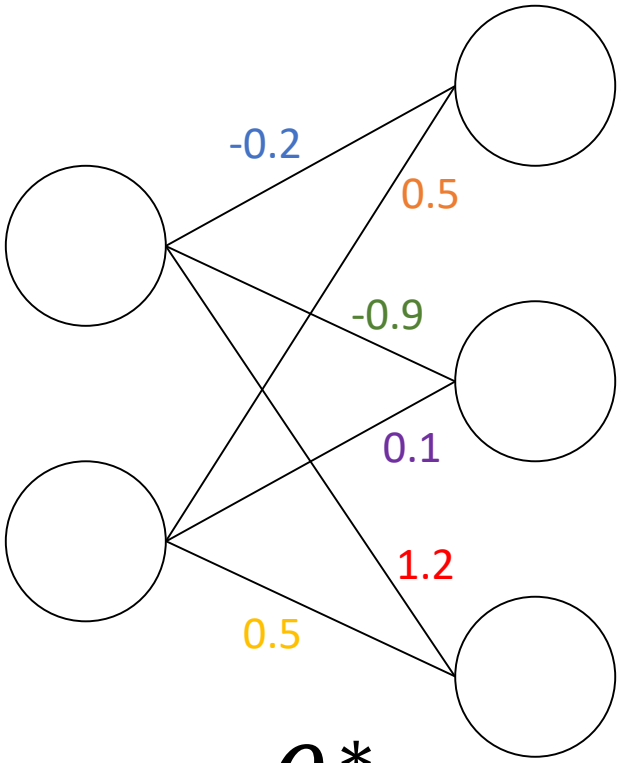
Expected Calibration Error

- The **expected calibration error (ECE)** provides a scalar measure of reliability.
- The ECE is the weighted average of the differences between accuracy and confidence levels, i.e.,

$$\mathrm{ECE} = \int_{c=0}^{1} |\mathrm{acc}(c) - c| \mathrm{d}c$$

Conventional Frequentist Learning

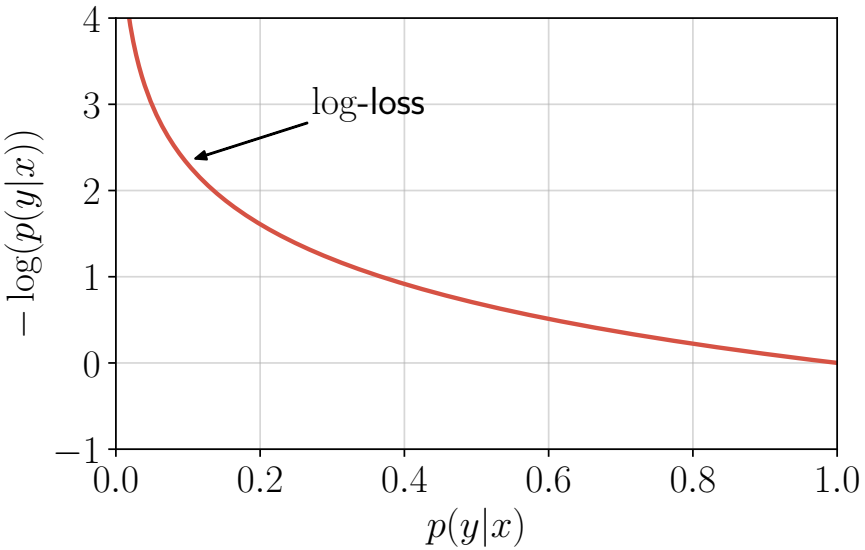
- **Frequentist** learning (e.g., standard deep learning):
  - Based on training data $\mathcal{D}$, optimization of a single model parameter vector θ
  - Decision based on a single model $p(y|x, \theta^*)$



Conventional Frequentist Learning

- Fix a loss function $\ell(x, y|\theta)$.
- Typical choice is the **log-loss**

$$\ell(x, y|\theta) = -\log p(y|x, \theta).$$



Conventional Frequentist Learning

- To optimize θ, frequentist learning tackles the **empirical risk minimization (ERM)** problem

$$\min_{\theta} \underbrace{\left\{ L_{\mathcal{D}}(\theta) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \ell(x, y|\theta) \right\}}_{\text{training loss}}$$

- The training loss is an empirical approximation of the **population loss**

$$L_{\mathrm{p}}(\theta) = \mathrm{E}_{(x,y)\sim P(x,y)}[\ell(x, y|\theta)]$$

- When data availability is limited, we have

$$L_{\mathcal{D}}(\theta) \neq L_{p}(\theta),$$

  and hence there is non-negligible **epistemic uncertainty**.
- This may cause the trained predictor $p(y|x, \theta^*)$ to be poorly calibrated.

Conventional Frequentist Learning

- To optimize θ, frequentist learning tackles the **empirical risk minimization (ERM)** problem

$$\min_{\theta} \underbrace{\left\{ L_{\mathcal{D}}(\theta) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \ell(x, y|\theta) \right\}}_{\text{training loss}}$$

- The training loss is an empirical approximation of the **population loss**

$$L_{\mathrm{p}}(\theta) = \mathrm{E}_{(x,y)\sim P(x,y)}[\ell(x, y|\theta)]$$

- When data availability is limited, we have

$$L_{\mathcal{D}}(\theta) \neq L_{p}(\theta),$$

  and hence there is non-negligible **epistemic uncertainty**.
- This may cause the trained predictor $p(y|x, \theta^*)$ to be poorly calibrated.

Conventional Frequentist Learning

- To optimize θ, frequentist learning tackles the **empirical risk minimization (ERM)** problem

$$\min_{\theta} \underbrace{\left\{ L_{\mathcal{D}}(\theta) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \ell(x, y|\theta) \right\}}_{\text{training loss}}$$

- The training loss is an empirical approximation of the **population loss**

$$L_{\mathrm{p}}(\theta) = \mathrm{E}_{(x,y)\sim P(x,y)}[\ell(x, y|\theta)]$$

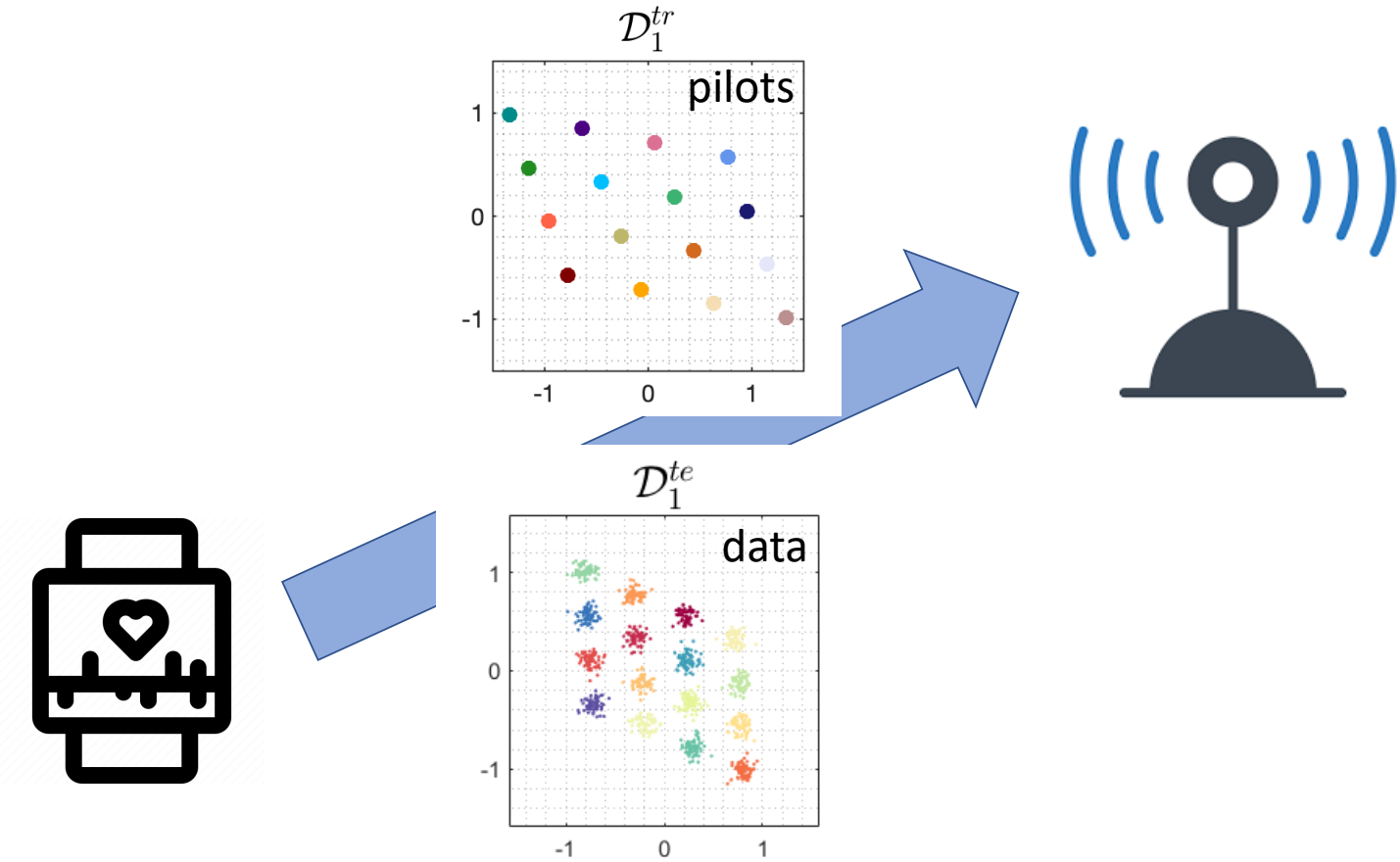
- When data availability is limited, we have

$$L_{\mathcal{D}}(\theta) \neq L_{p}(\theta),$$

  and hence there is non-negligible **epistemic uncertainty**.
- This may cause the trained predictor $p(y|x, \theta^*)$ to be poorly calibrated.

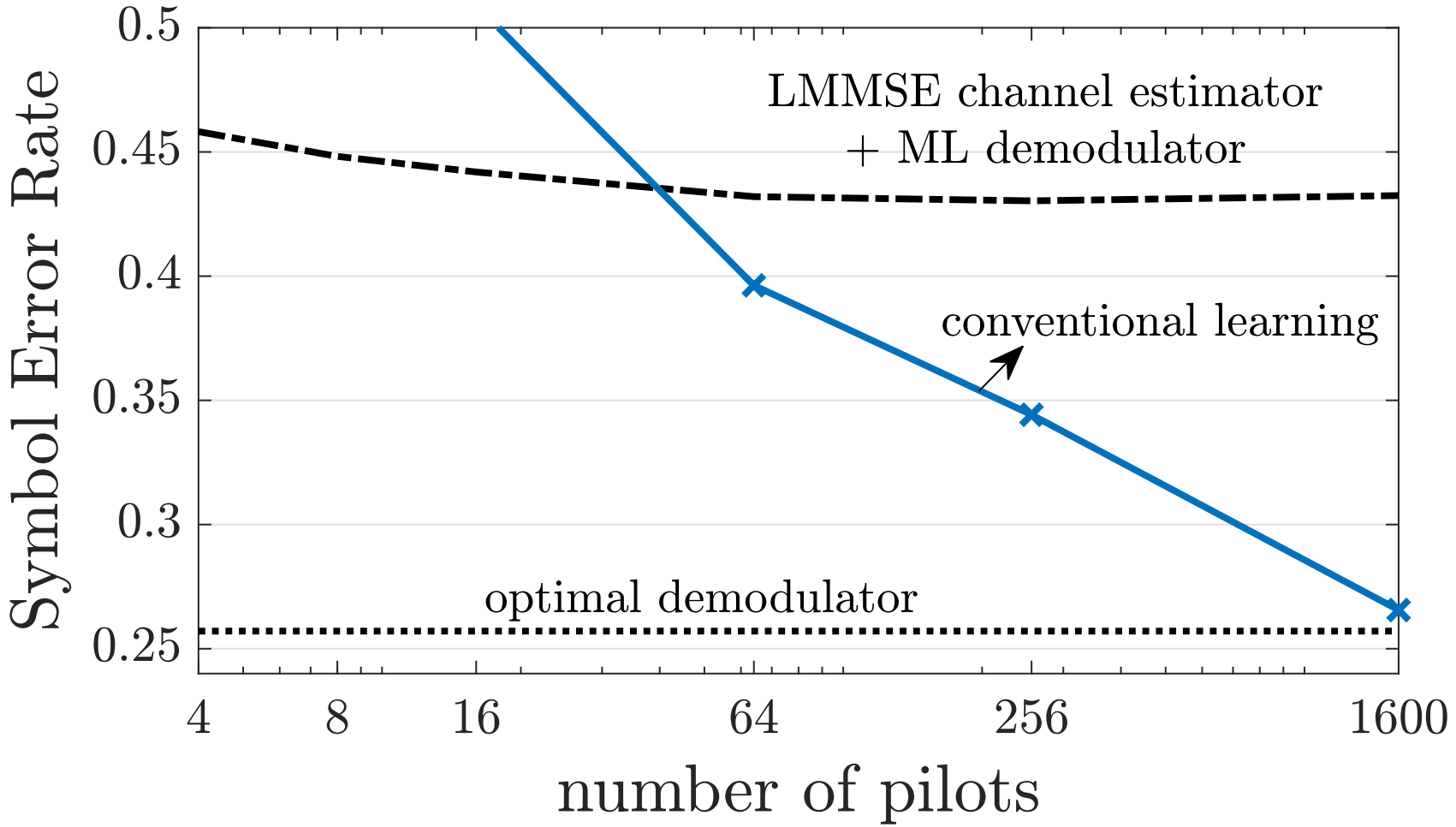
Application to Demodulation

- Short-packet transmission with I/Q imbalance[2]



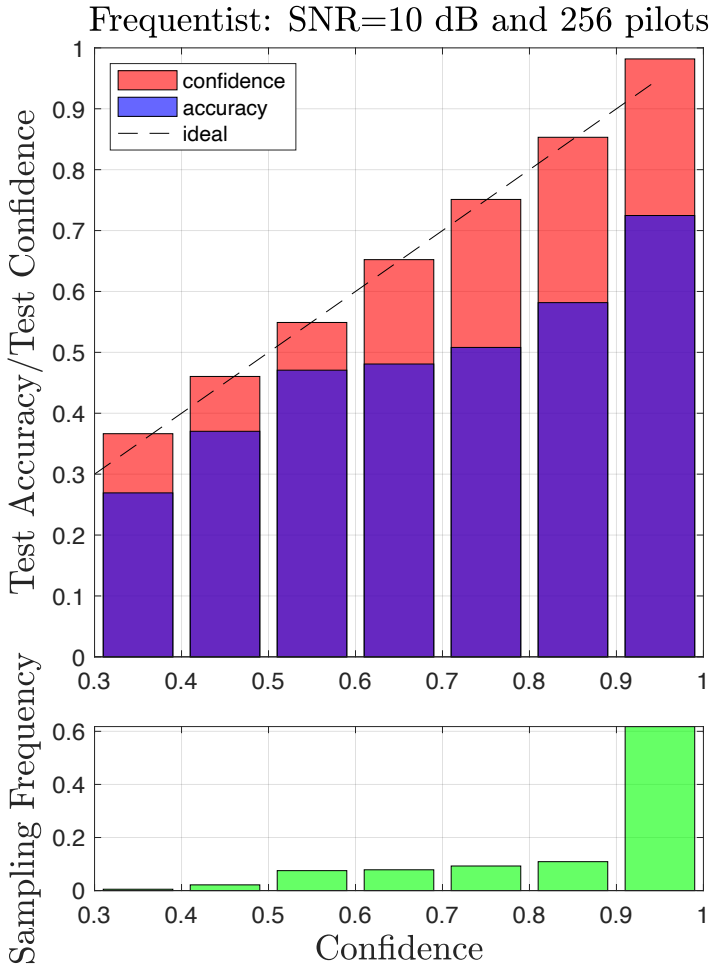
2 K. Cohen, S. Park, and O. Simeone, "Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning," in Proc. WSA, 2021.

Application to Demodulation



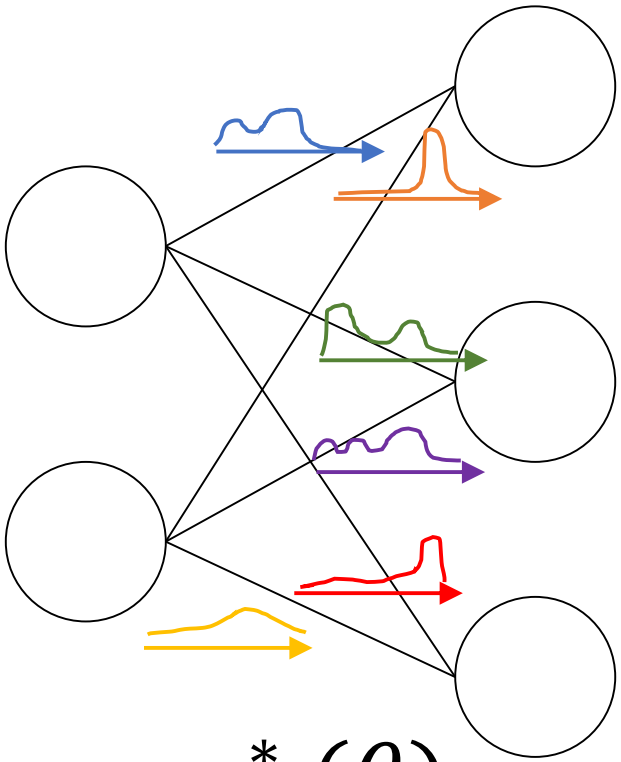
Application to Demodulation

- By ignoring epistemic uncertainty, frequentist learning yields poorly calibrated decisions.



Bayesian Learning

- **Bayesian** learning:
  - Optimization of a distribution $q(\theta)$ in the model parameter space
  - Distribution $q(\theta)$ encodes **epistemic uncertainty**.

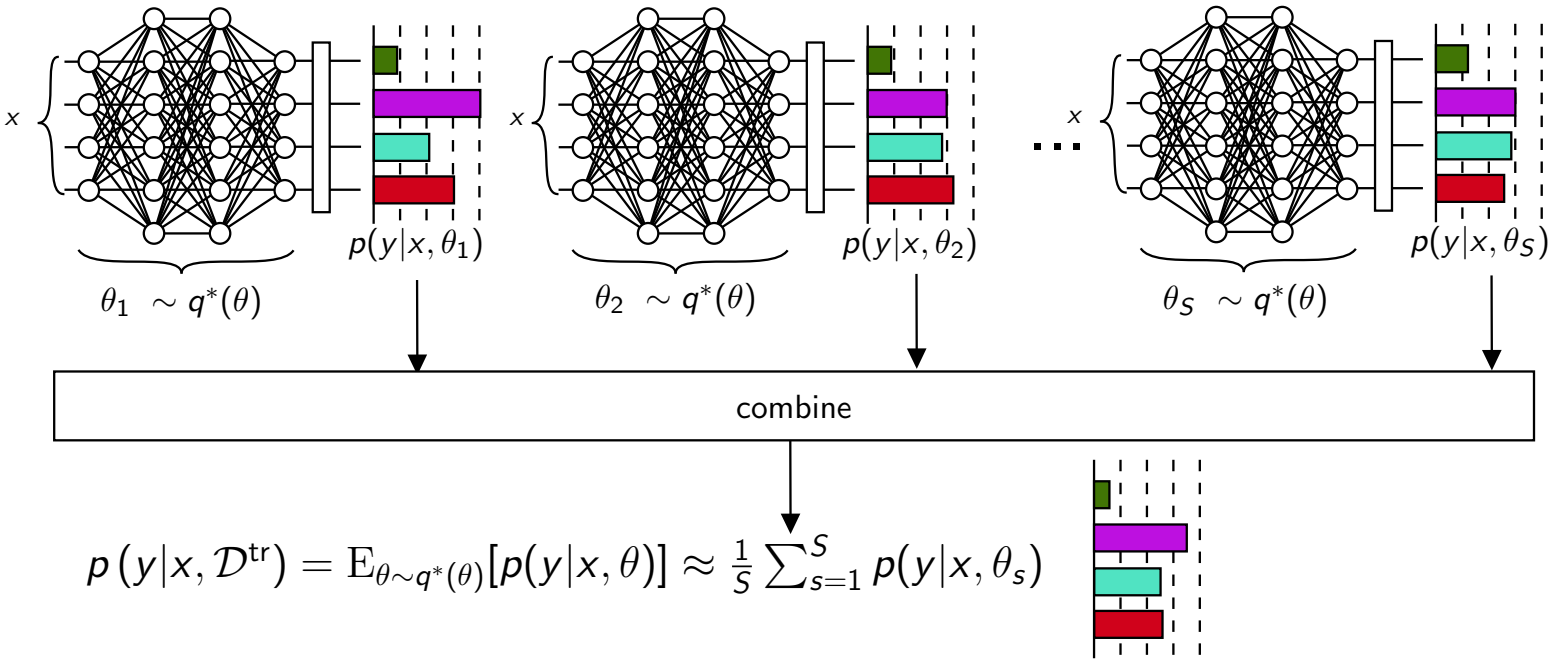


Bayesian Learning

- Decision obtained via **ensembling**, i.e., via

$$\mathrm{E}_{\theta \sim q^*(\theta)} \left[p(y|x, \theta) \right],$$

  accounting for the "opinions" of multiple models.
- In practice, the average is done over S i.i.d. model parameters $\theta \sim q^*(\theta)$.



Bayesian Learning

- At test time, we have
  - **Hard decision**: $\hat{y}(x|q^*) = \arg\max_y \mathrm{E}_{\theta\sim q^*(\theta)}\left[p(y|x,\theta)\right]$ (class with largest average score)
  - **Confidence level**:

$$\mathrm{conf}(x|q^*) = \mathrm{E}_{\theta\sim q^*(\theta)}[p(\hat{y}(x|q^*)|x,\theta)]$$

- The confidence level accounts for **epistemic uncertainty** via the **disagreement** among models.[3]

[3] N. Houlsby, et al, "Bayesian active learning for classification and preference learning," arXiv:1112.5745, 2011.

Bayesian Learning

- At test time, we have
  - **Hard decision**: $\hat{y}(x|q^*) = \arg\max_y \mathrm{E}_{\theta\sim q^*(\theta)}\left[p(y|x,\theta)\right]$ (class with largest average score)
  - **Confidence level**:

$$\mathrm{conf}(x|q^*) = \mathrm{E}_{\theta\sim q^*(\theta)}[p(\hat{y}(x|q^*)|x,\theta)]$$

- The confidence level accounts for **epistemic uncertainty** via the **disagreement** among models.[3]

[3] N. Houlsby, et al, "Bayesian active learning for classification and preference learning," arXiv:1112.5745, 2011.

Bayesian Learning

- Assuming a prior distribution $p(\theta)$, Bayesian learning minimizes the **free energy**

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

- If no constraints are imposed on the distribution $q(\theta)$, the optimal solution $q^*(\theta)$ is given by the standard posterior distribution

$$p(\theta|\mathcal{D}) \propto p(\theta)\exp\left(-L_{\mathcal{D}}(\theta)\right) = \underbrace{p(\theta)}_{\text{prior}} \left(\prod_{n=1}^{N} \underbrace{p(y_n|x_n,\theta)}_{\text{likelihood}}\right).$$

Bayesian Learning

- Assuming a prior distribution $p(\theta)$, Bayesian learning minimizes the **free energy**

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

- If no constraints are imposed on the distribution $q(\theta)$, the optimal solution $q^*(\theta)$ is given by the standard posterior distribution

$$p(\theta|\mathcal{D}) \propto p(\theta)\exp\left(-L_{\mathcal{D}}(\theta)\right) = \underbrace{p(\theta)}_{\text{prior}} \left(\prod_{n=1}^{N} \underbrace{p(y_n|x_n,\theta)}_{\text{likelihood}}\right).$$

Generalized Bayesian Learning and Frequentist Learning

- More generally, we can define **generalized Bayesian learning** as minimizing the free energy[4,5]

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

- With $\beta = 0$, the problem reduces to frequentist learning, which outputs a single model parameter vector θ^*.
- This criterion is well justified by **PAC Bayes theory**, which derives it as an upper bound on the population loss.[6]

4 J. Knoblauch, et al, "Generalized variational inference: Three arguments for deriving new posteriors," arXiv:1904.02063, 2019.

5 O. Simeone, "Machine Learning for Engineers", Cambridge University Press, 2022.

6 P. Alquier, "User-friendly introduction to PAC-Bayes bounds," arXiv preprint, 2021.

Generalized Bayesian Learning and Frequentist Learning

- More generally, we can define **generalized Bayesian learning** as minimizing the free energy[4,5]

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

- With $\beta = 0$, the problem reduces to frequentist learning, which outputs a single model parameter vector θ^*.
- This criterion is well justified by **PAC Bayes theory**, which derives it as an upper bound on the population loss.[6]

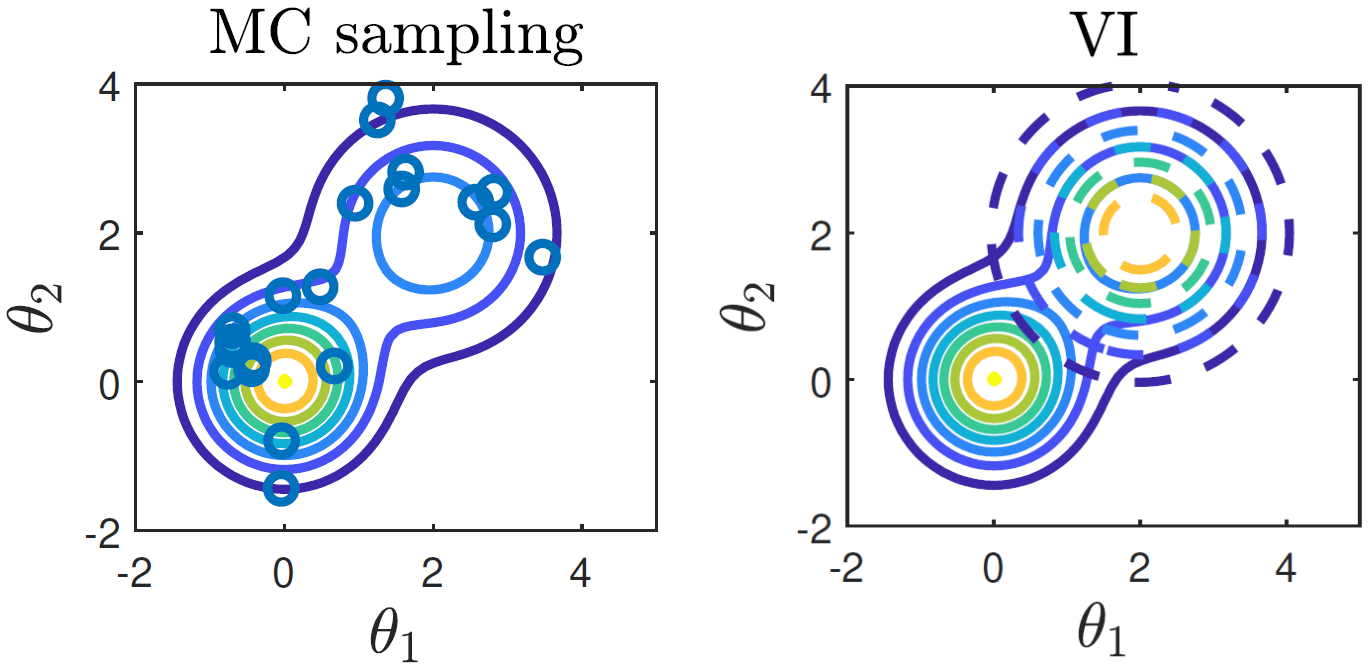
4 J. Knoblauch, et al, "Generalized variational inference: Three arguments for deriving new posteriors," arXiv:1904.02063, 2019.

5 O. Simeone, "Machine Learning for Engineers", Cambridge University Press, 2022.

6 P. Alquier, "User-friendly introduction to PAC-Bayes bounds," arXiv preprint, 2021.

Bayesian Learning in Practice

- Exact minimization of the free energy is practically infeasible.
- Practical solutions are based on variational inference (VI) or Monte Carlo (MC) sampling.



Variational Inference (VI)

- VI tackle the minimization of the free energy

$$F_{\mathcal{D}}(\varphi) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta|\varphi)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta|\varphi)||p_0(\theta)\right)}_{\text{information-theoretic regularization}}$$

  over a distribution $q(\theta|\varphi)$ within a given parametric family.

- A common choice is the Gaussian variational posterior

$$q(\theta|\varphi) = \mathcal{N}(\theta|\mu, \Sigma),$$

  for which optimization is done over parameters $\varphi = \{\mu, \Sigma\}$, typically via gradient descent (GD).

Variational Inference (VI)

- VI tackle the minimization of the free energy

$$F_{\mathcal{D}}(\varphi) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta|\varphi)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta|\varphi)||p_0(\theta)\right)}_{\text{information-theoretic regularization}}$$

  over a distribution $q(\theta|\varphi)$ within a given parametric family.
- A common choice is the Gaussian variational posterior

$$q(\theta|\varphi) = \mathcal{N}(\theta|\mu, \Sigma),$$

  for which optimization is done over parameters $\varphi = \{\mu, \Sigma\}$, typically via gradient descent (GD).

Reliability of Bayesian Learning

- If the model is **well specified**, i.e., if the joint distribution of training and test data satisfies

$$P(\mathcal{D}, (x, y)) \approx \underbrace{\mathrm{E}_{\theta \sim p(\theta)} \left[\prod_{(x,y) \in \mathcal{D} \cup \{(x,y)\}} p(x, y|\theta) \right]}_{p(\mathcal{D}, (x,y))},$$

  then Bayesian learning is **reliable**, or **well calibrated**:

$$P(y|x, \mathcal{D}) \approx p(y|x, \mathcal{D}) = \mathrm{E}_{\theta \sim p(\theta|\mathcal{D})}[p(y|x, \theta)]$$

- Hence, calibration in Bayesian learning does not hinge on the availability of a large data set, but "only" on a well-specified model.

Reliability of Bayesian Learning

- If the model is **well specified**, i.e., if the joint distribution of training and test data satisfies

$$P(\mathcal{D}, (x, y)) \approx \underbrace{\mathrm{E}_{\theta \sim p(\theta)} \left[\prod_{(x,y) \in \mathcal{D} \cup \{(x,y)\}} p(x, y|\theta) \right]}_{p(\mathcal{D}, (x,y))},$$

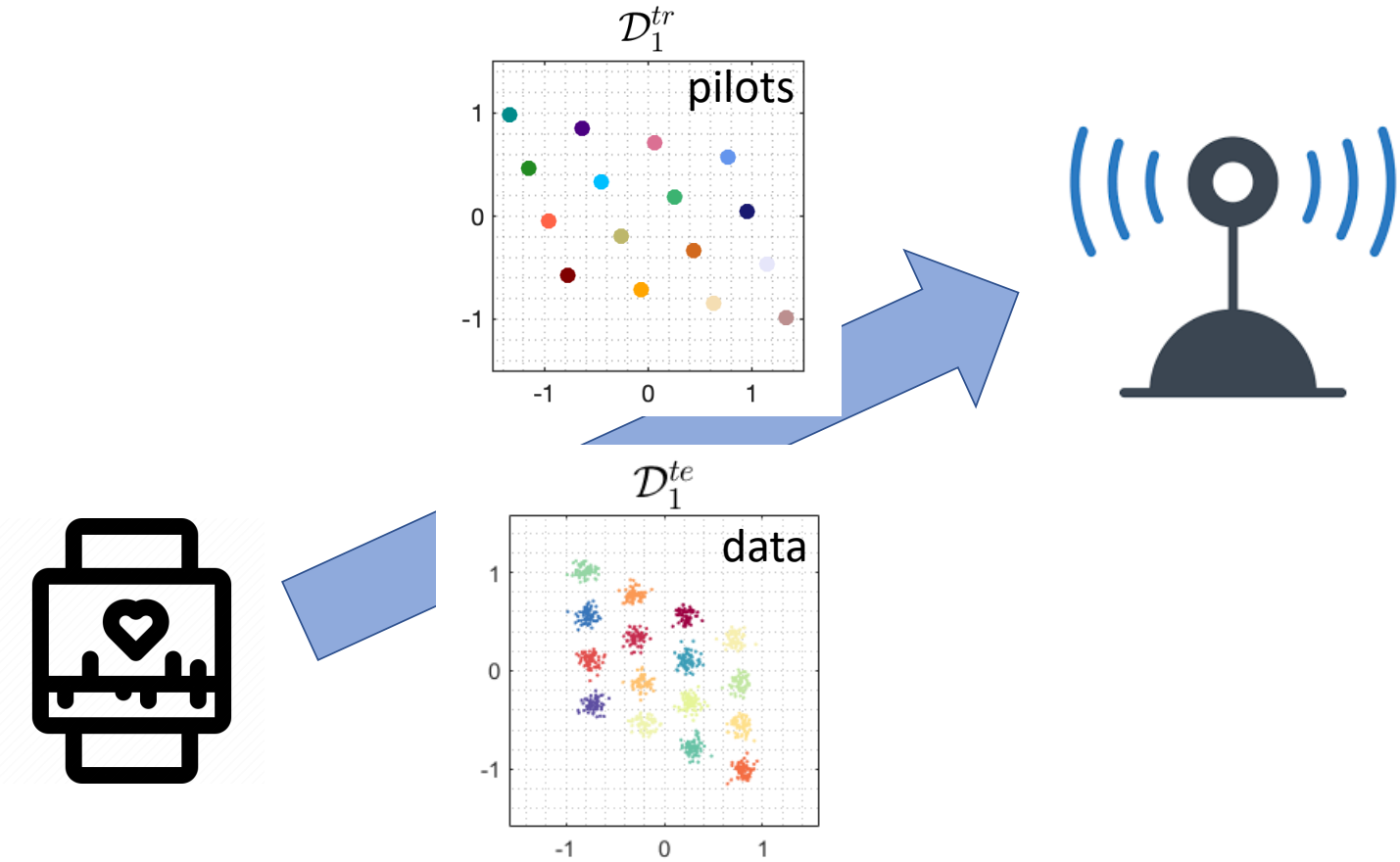
  then Bayesian learning is **reliable**, or **well calibrated**:

$$P(y|x, \mathcal{D}) \approx p(y|x, \mathcal{D}) = \mathrm{E}_{\theta \sim p(\theta|\mathcal{D})}[p(y|x, \theta)]$$

- Hence, calibration in Bayesian learning does not hinge on the availability of a large data set, but "only" on a well-specified model.

Application to Demodulation

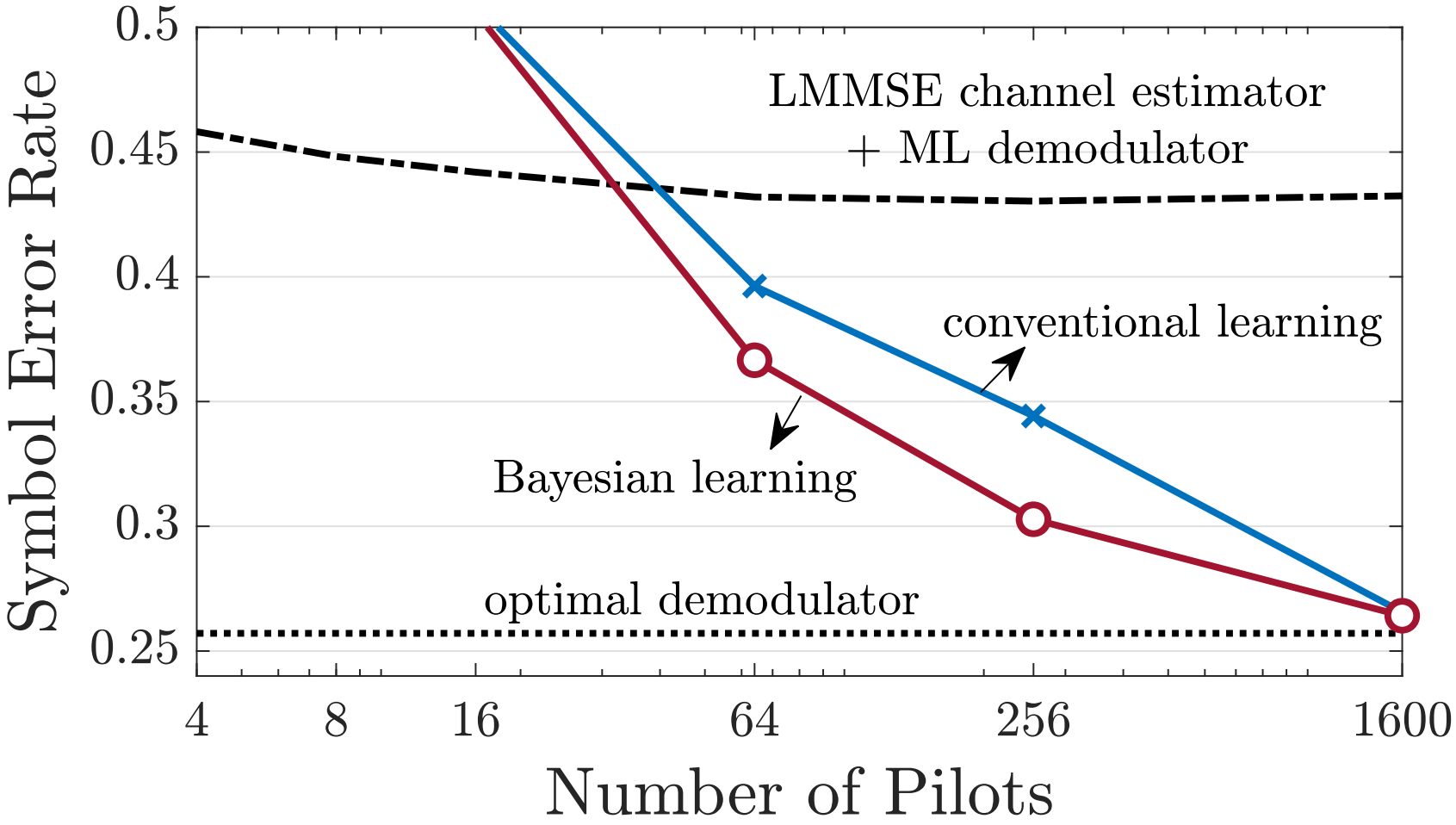
- Short-packet transmission with I/Q imbalance[7]



7 K. Cohen, S. Park, and O. Simeone, "Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning," in Proc. WSA, 2021.

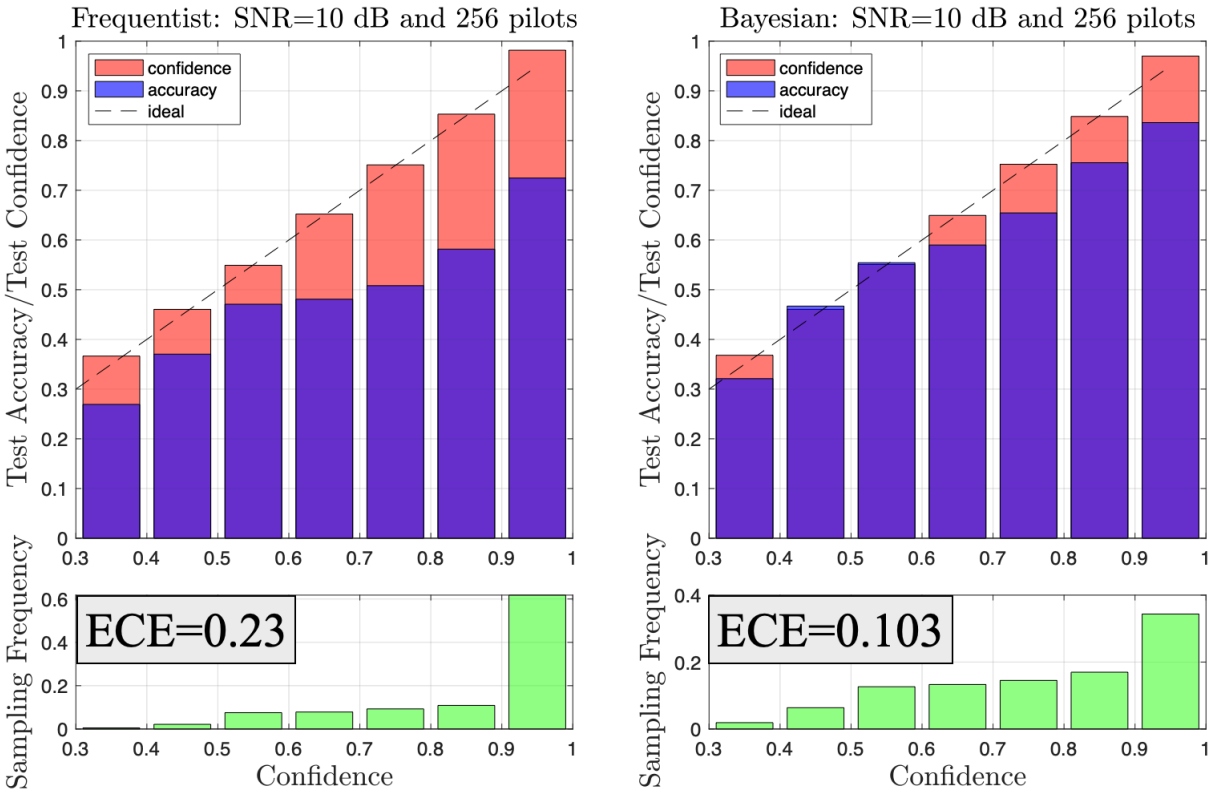
Application to Demodulation

- Frequentist and Bayesian learning yields similar accuracy levels.



Application to Demodulation

- Frequentist learning yields **overconfident** decisions, while Bayesian learning produces **well-calibrated** outputs.



Robust and Reliable AI: Robust Bayesian Learning

Robust Bayesian Learning

- Both frequentist and Bayesian learning assume that training and test data are generated from the same distribution.
  - What if there are **outliers** in the training data?
  - Example: Training data may be affected by interference or malicious reporting.
- Bayesian learning is well calibrated only if the model is well specified.
  - What if the model is **misspecified**?
  - Example: Light-weight models deployed on resource constrained devices

Robust Bayesian Learning

- Both frequentist and Bayesian learning assume that training and test data are generated from the same distribution.
  - What if there are **outliers** in the training data?
  - Example: Training data may be affected by interference or malicious reporting.
- Bayesian learning is well calibrated only if the model is well specified.
  - What if the model is **misspecified**?
  - Example: Light-weight models deployed on resource constrained devices

Limitations of Bayesian Learning: Model Misspecification

- Choosing $\beta \neq 1$ in generalized Bayesian learning can partly address the problem of misspecification:
  - This is related to the "**cold posterior** problem"[8]
- But the next example suggests that this may not be enough.

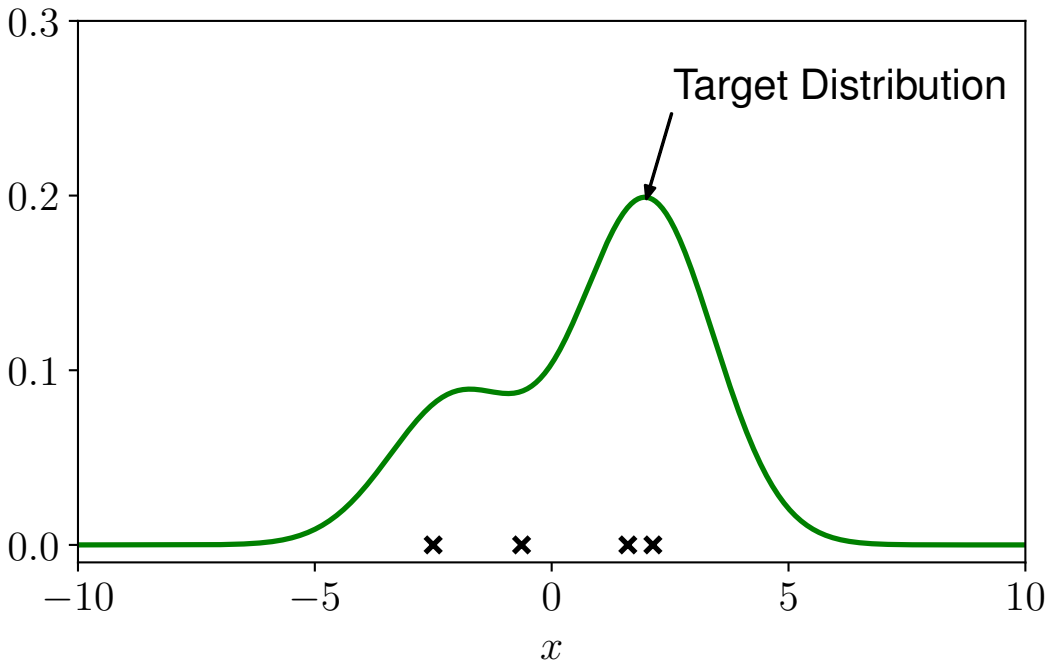
[8] K. Pitas and J. Arbel, "Cold Posteriors through PAC-Bayes," arXiv:2206.11173.

A Toy Example

- Consider a density estimation problem with an underlying data distribution that is a mixture of Gaussians (e.g., a fading channel with blocking for mmwave or THz communications):

$$P(x) = 0.7\mathcal{N}(x|2,2) + 0.3\mathcal{N}(x|-2,2).$$

- Assume a Gaussian likelihood function: $p(x|\theta) = \mathcal{N}(x|\theta,1)$



A Toy Example

- In frequentist learning, we would optimize the mean θ, obtaining the predictive distribution $p(x|\theta^*) = \mathcal{N}(x|\theta^*, 1)$.
- Bayesian learning minimizes the free energy, obtaining a distribution $q^*(\theta)$.
- Then, the predictive data distribution produced by Bayesian learning is evaluated via ensembling, i.e., via the average across multiple models

$$\mathrm{E}_{\theta\sim q^*(\theta)}[p(x|\theta)] = \mathrm{E}_{\theta\sim q^*(\theta)}[\mathcal{N}(x|\theta, 1)].$$

- So, Bayesian learning has the capacity to capture a multi-modal Gaussian distribution.

A Toy Example

- In frequentist learning, we would optimize the mean θ, obtaining the predictive distribution $p(x|\theta^*) = \mathcal{N}(x|\theta^*, 1)$.
- Bayesian learning minimizes the free energy, obtaining a distribution $q^*(\theta)$.
- Then, the predictive data distribution produced by Bayesian learning is evaluated via ensembling, i.e., via the average across multiple models

$$\mathrm{E}_{\theta\sim q^*(\theta)}[p(x|\theta)] = \mathrm{E}_{\theta\sim q^*(\theta)}[\mathcal{N}(x|\theta, 1)].$$

- So, Bayesian learning has the capacity to capture a multi-modal Gaussian distribution.

A Toy Example

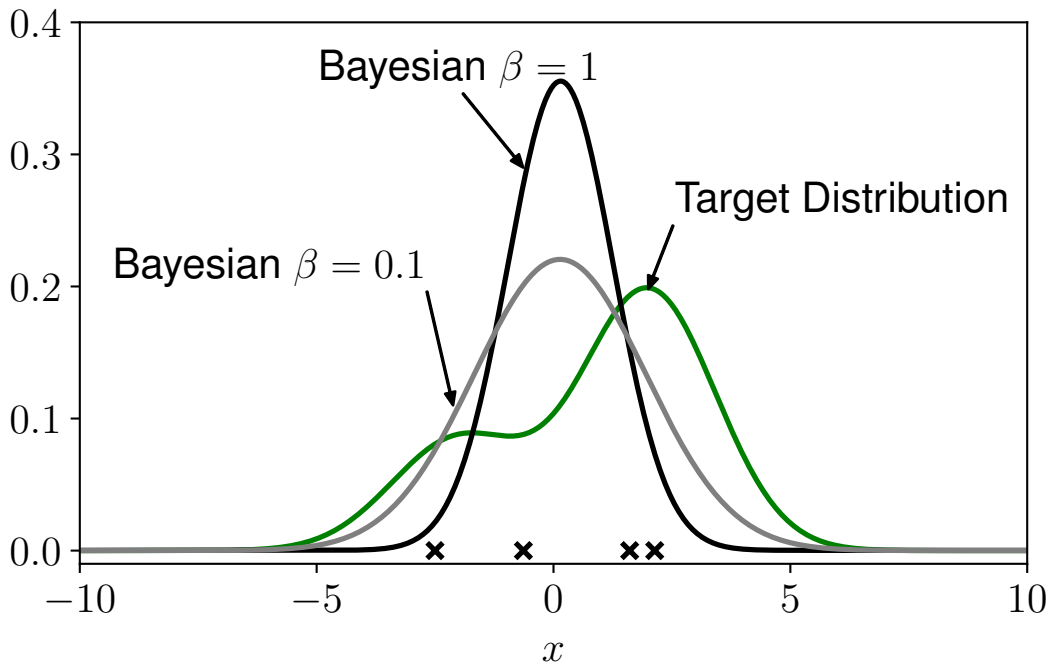
- In frequentist learning, we would optimize the mean θ, obtaining the predictive distribution $p(x|\theta^*) = \mathcal{N}(x|\theta^*, 1)$.
- Bayesian learning minimizes the free energy, obtaining a distribution $q^*(\theta)$.
- Then, the predictive data distribution produced by Bayesian learning is evaluated via ensembling, i.e., via the average across multiple models

$$\mathrm{E}_{\theta\sim q^*(\theta)}[p(x|\theta)] = \mathrm{E}_{\theta\sim q^*(\theta)}[\mathcal{N}(x|\theta, 1)].$$

- So, Bayesian learning has the capacity to capture a multi-modal Gaussian distribution.

A Toy Example

- The model class is **misspecified** since it is not possible to capture both modes of the data distribution using a *single* Gaussian model.
- In this scenario, generalized Bayesian learning presents poor generalization, even with $\beta \neq 1$.



Generalized Bayesian Learning and Misspecification

- Why is generalized Bayesian learning failing to match the ensemble $\mathrm{E}_{\theta\sim q^*(\theta)}[p(x|\theta)]$ to the true data distribution $P(x)$?
- The key issue is that the free energy

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

includes the "ensemble" training loss

$$L_{\mathcal{D}}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log\left(p(x|\theta)\right)$$

and not the loss of the ensemble.

Generalized Bayesian Learning and Misspecification

- Why is generalized Bayesian learning failing to match the ensemble $\mathrm{E}_{\theta\sim q^*(\theta)}[p(x|\theta)]$ to the true data distribution $P(x)$?
- The key issue is that the free energy

$$F_{\mathcal{D}}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p(\theta)\right)}_{\text{information-theoretic regularization}}$$

includes the "ensemble" training loss

$$L_{\mathcal{D}}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log\left(p(x|\theta)\right)$$

and not the loss of the ensemble.

$(m, 1)$-Robust (Generalized) Bayesian Learning

- To overcome the limitations of (generalized) Bayesian learning, it was recently proposed to use a multi-sample version of the free energy: $(m, 1)$-**robust Bayesian learning**.
- The m-**sample free energy** is defined as[9]

$$F_{\mathcal{D}}^{m}(q(\theta)) = N\mathrm{E}_{\theta_1,\ldots,\theta_m \sim q(\theta)^{\otimes m}}\left[L_{\mathcal{D}}^{m}(\theta)\right] + \beta \mathrm{KL}\left(q(\theta)||p_0(\theta)\right)$$

where the training loss

$$L_{\mathcal{D}}^{m}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log\left(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i)\right)$$

explicitly captures the log-loss of a mixture of m models drawn from $q(\theta)$.

- Like the free energy, it can be justified via the analysis of generalization based on **PAC Bayes** theory.

9 W. Morningstar, et al "PAC ^{m}-Bayes: Narrowing the Empirical Risk Gap...", NeurIPS 2021.

$(m, 1)$-Robust (Generalized) Bayesian Learning

- To overcome the limitations of (generalized) Bayesian learning, it was recently proposed to use a multi-sample version of the free energy: $(m, 1)$-**robust Bayesian learning**.
- The m-**sample free energy** is defined as[9]

$$F_{\mathcal{D}}^{m}(q(\theta)) = N\mathrm{E}_{\theta_1,\ldots,\theta_m \sim q(\theta)^{\otimes m}}\left[L_{\mathcal{D}}^{m}(\theta)\right] + \beta \mathrm{KL}\left(q(\theta)||p_0(\theta)\right)$$

where the training loss

$$L_{\mathcal{D}}^{m}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log\left(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i)\right)$$

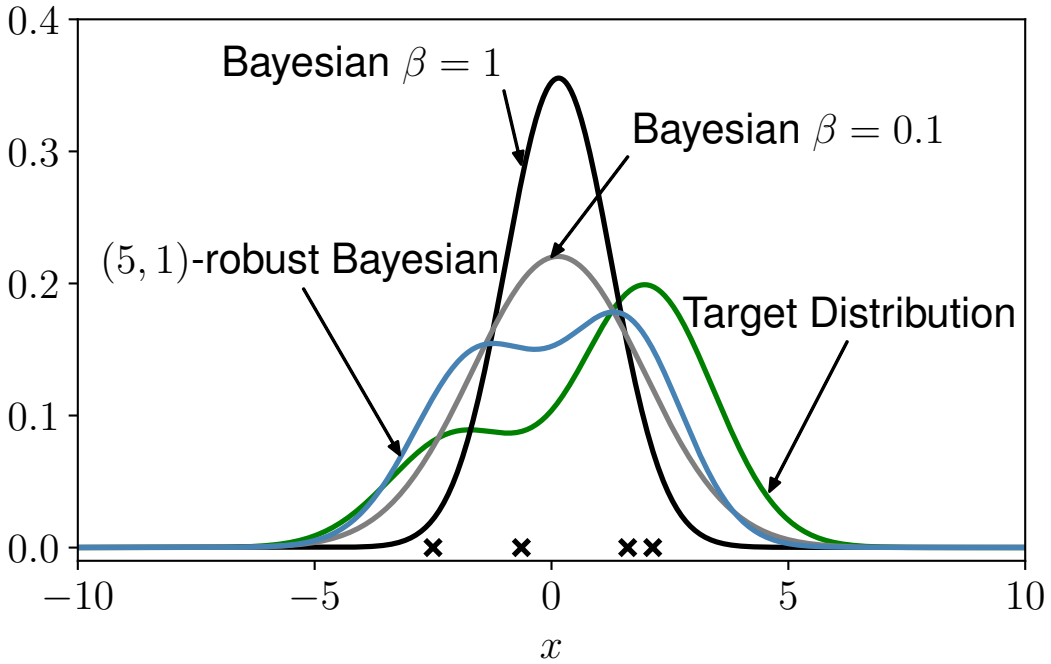
explicitly captures the log-loss of a mixture of m models drawn from $q(\theta)$.

- Like the free energy, it can be justified via the analysis of generalization based on **PAC Bayes** theory.

9 W. Morningstar, et al "PAC ^{m}-Bayes: Narrowing the Empirical Risk Gap...", NeurIPS 2021.

Toy Example (Continued)

- $(m, 1)$-robust Bayesian learning is clearly better able to capture the multi-modal properties of the ground-truth distribution $P(x)$.



Limitations of Bayesian Learning: Outliers

- Training data often contains **outliers** – anomalous data points that do not follow the same distribution of test data
  - Errors due to human labeling, measuring tools failures or interference, adversarial examples, ...

- Outliers can be modelled via the gross-error model: Given a contamination ratio $\epsilon \in (0, 1]$, the sampling distribution is[10]

$$\tilde{P}(x) = \epsilon \underbrace{Q(x)}_{\substack{\text{out-of-distribution} \\ \text{measure (OOD)}}} + (1 - \epsilon) \underbrace{P(x)}_{\substack{\text{in-distribution} \\ \text{measure (ID)}}}$$

[10] P. J. Huber, "Robust estimation of a location parameter," The Annals of Mathematical Statistics, 1964.

Limitations of Bayesian Learning: Outliers

- Training data often contains **outliers** – anomalous data points that do not follow the same distribution of test data
  - Errors due to human labeling, measuring tools failures or interference, adversarial examples, ...

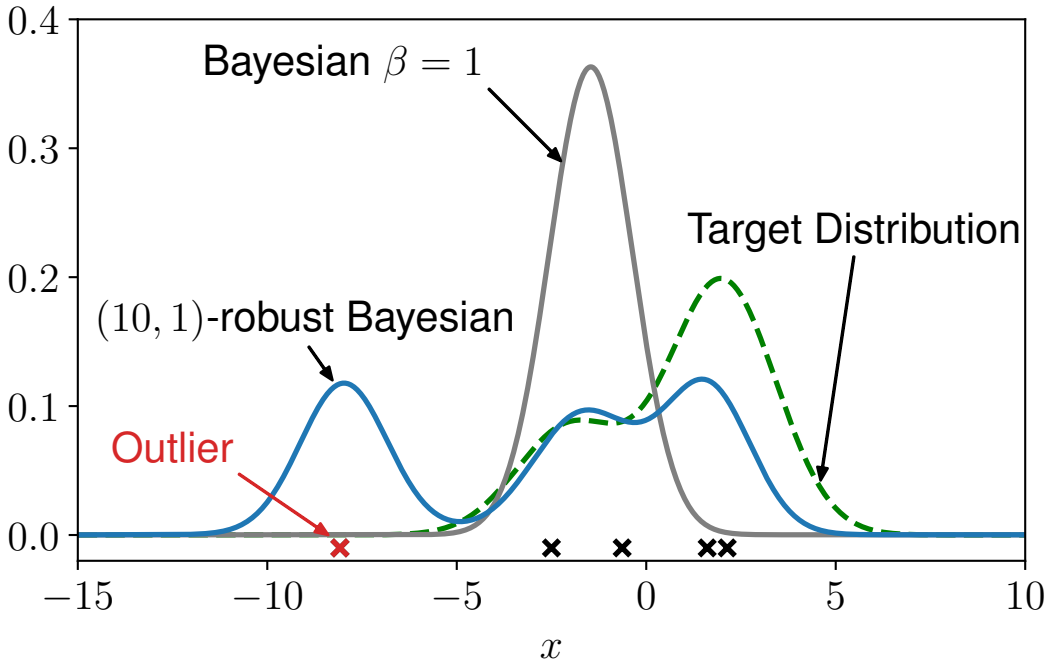
- Outliers can be modelled via the gross-error model: Given a contamination ratio $\epsilon \in (0, 1]$, the sampling distribution is[10]

$$\tilde{P}(x) = \epsilon \underbrace{Q(x)}_{\substack{\text{out-of-distribution} \\ \text{measure (OOD)}}} + (1 - \epsilon) \underbrace{P(x)}_{\substack{\text{in-distribution} \\ \text{measure (ID)}}}$$

[10] P. J. Huber, "Robust estimation of a location parameter," The Annals of Mathematical Statistics, 1964.

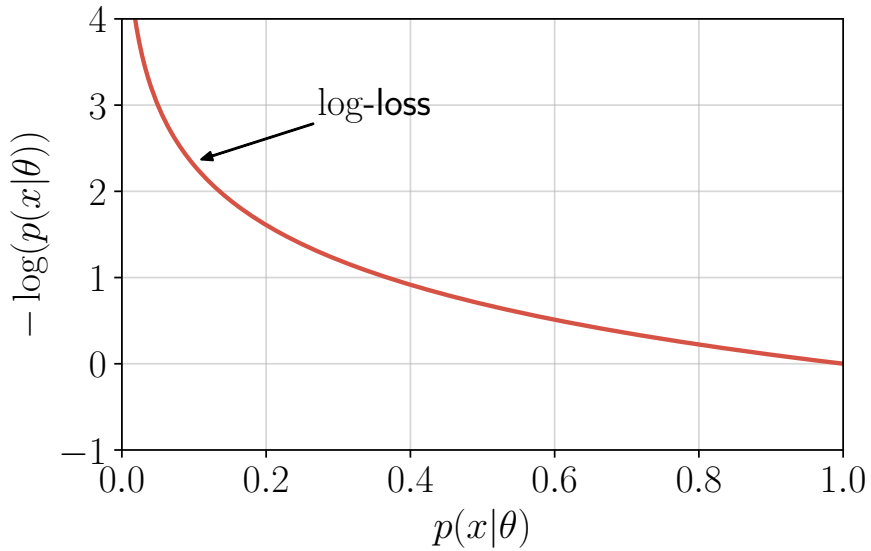
Toy Example (Continued)

- While more robust to misspecification, $(m, 1)$-robust Bayesian learning is significantly affected by outliers.



Reconsidering the Log-Loss

- What is the cause of the lack of robustness of existing free energy metrics?
- The free energy relies on the standard log-loss $-\log p(x|\theta)$, which penalizes very strongly models that do not cover well all data points, including outliers.

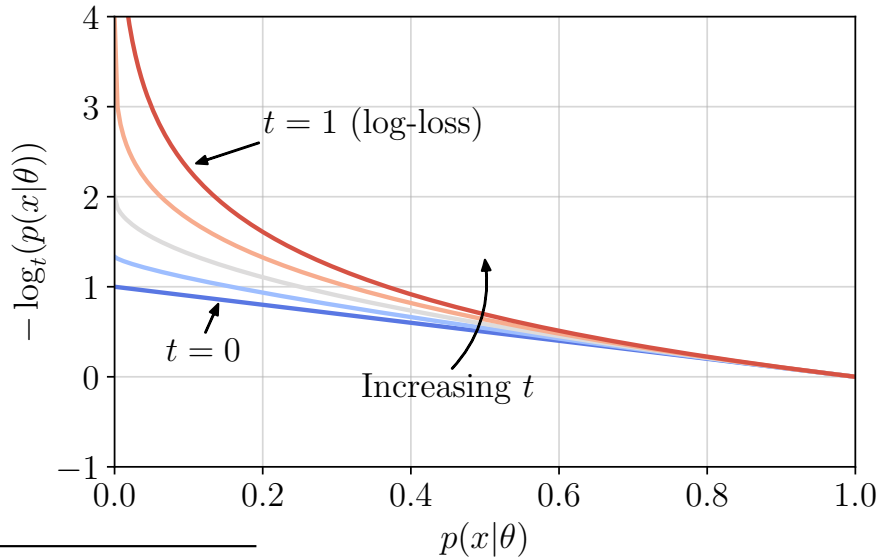


Beyond the Log-Loss: t-Log-Loss

- The t-**log-loss**, for $t \in [0, 1)$, is defined as[11,12]

$$-\log_t(p) := -\frac{1}{1-t}\left(p^{1-t} - 1\right) \quad \text{for } x > 0,$$

  - For $t \to 1$ recovers the standard log-loss
  - Since we have $-\log_t(p) \leq (1-t)^{-1}$, outliers have a bounded influence when t is sufficiently small.



11 C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," Journal of Statistical Physics, 1988.

12 T. Sypherd, et al, "A loss function for robust classification..." arXiv:1906.02314, 2019.

(m, t)-Robust (Generalized) Bayesian Learning

- (m, t)-**robust Bayesian learning** minimizes the (m, t)-**free energy criterion**:[13]

$$F_{\mathcal{D}}^{m,t}(q(\theta)) = N\mathrm{E}_{\theta_1,\ldots,\theta_m \sim q(\theta)^{\otimes m}}\left[L_{\mathcal{D}}^{m,t}(\theta)\right] + \mathrm{KL}\left(q(\theta)||p_0(\theta)\right)$$

  where

$$L_{\mathcal{D}}^{m,t}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log_t\left(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i)\right)$$

  replaces the log-loss with the t-log-loss.

- The criterion has two tuning knobs:
  - the generalized logarithm parameter $t \in [0, 1)$, which determines the robustness to outliers;
  - and the number constituent models $m \geq 1$ in the ensemble, which determines the robustness to misspecification.

13 M. Zecchin, et al, "Robust PAC^m..." arXiv:2203.01859, 2022.

(m, t)-Robust (Generalized) Bayesian Learning

- (m, t)-**robust Bayesian learning** minimizes the (m, t)-**free energy criterion**:[13]

$$F_{\mathcal{D}}^{m,t}(q(\theta)) = N\mathrm{E}_{\theta_1,\ldots,\theta_m \sim q(\theta)^{\otimes m}}\left[L_{\mathcal{D}}^{m,t}(\theta)\right] + \mathrm{KL}\left(q(\theta)||p_0(\theta)\right)$$

  where

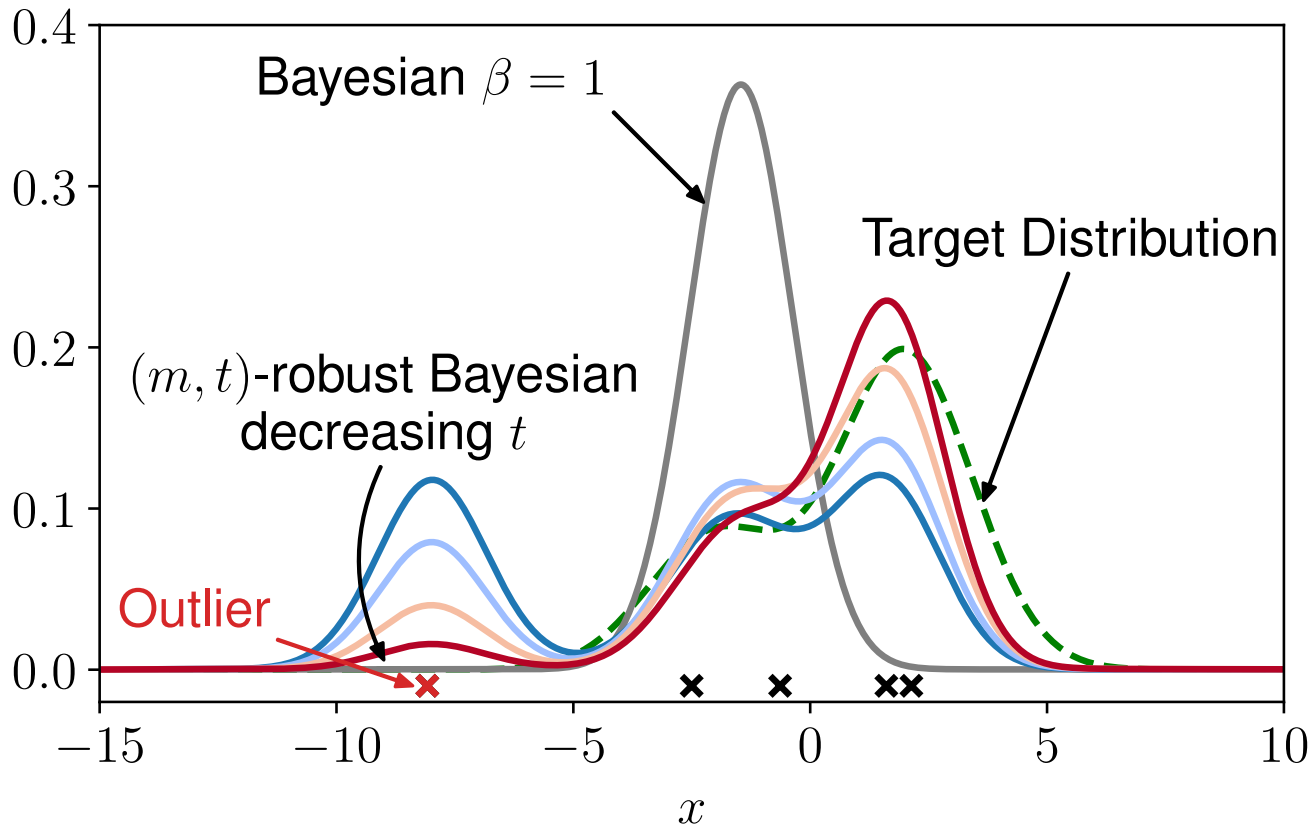
$$L_{\mathcal{D}}^{m,t}(\theta) = -\frac{1}{N}\sum_{x\in\mathcal{D}} \log_t\left(\frac{1}{m}\sum_{i=1}^{m} p(x|\theta_i)\right)$$

  replaces the log-loss with the t-log-loss.
- The criterion has two tuning knobs:
  - the generalized logarithm parameter $t \in [0, 1)$, which determines the robustness to outliers;
  - and the number constituent models $m \geq 1$ in the ensemble, which determines the robustness to misspecification.

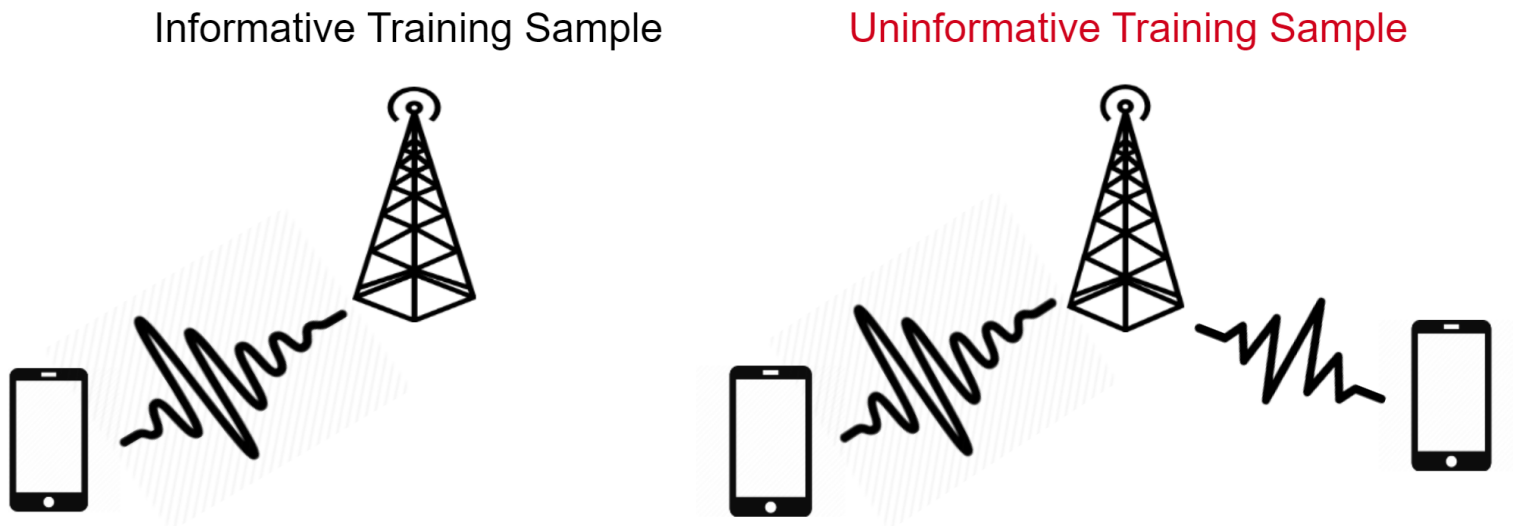
13 M. Zecchin, et al, "Robust PAC^m..." arXiv:2203.01859, 2022.

Toy Example (Continued)

- (m, t)-robust Bayesian learning is able to tackle both model misspecification and the presence of outliers.



Robust Bayesian Learning: Automatic Modulation Classification



- Determine the modulation type y associated to a received based-band signal vector x.
- Interference leads to uninformative training samples with ambiguous labels, i.e., outliers.

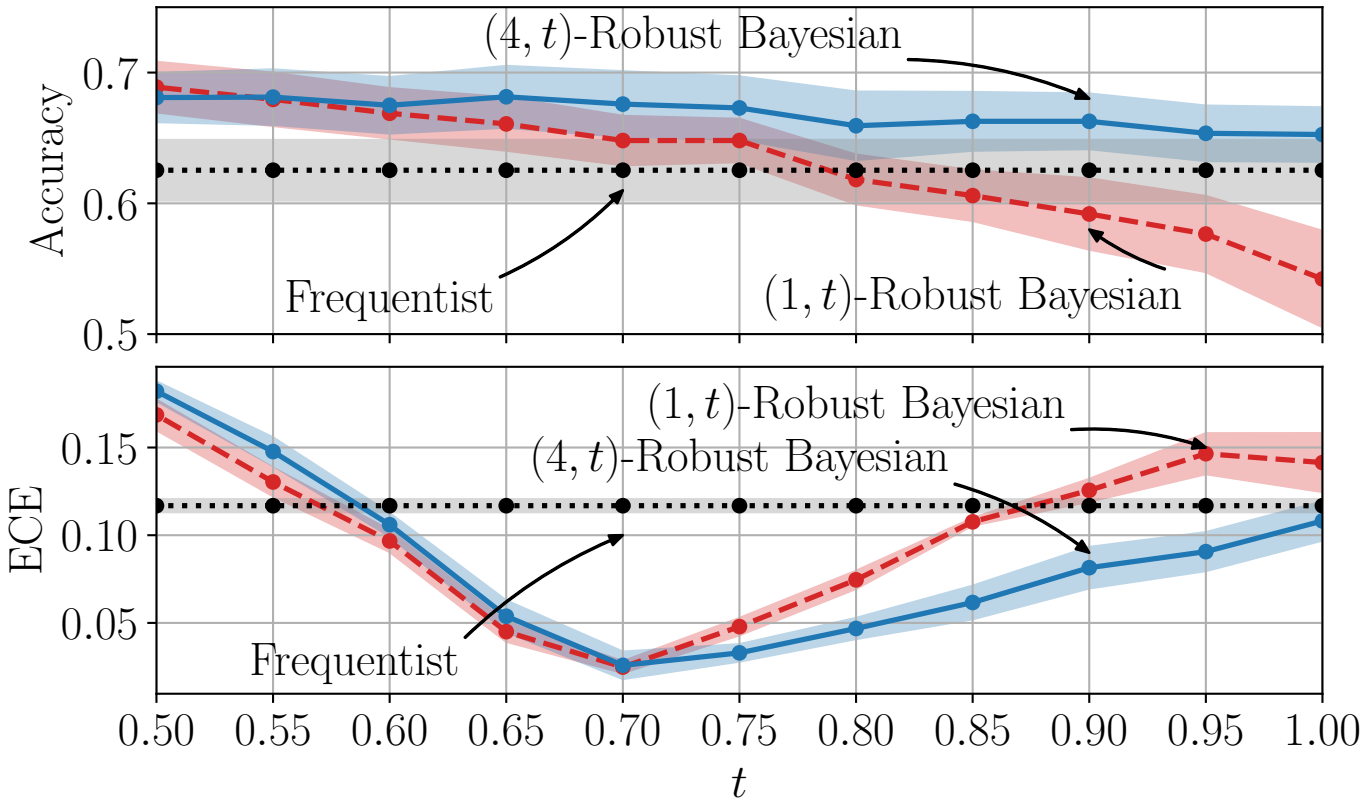
Robust Bayesian Learning: Automatic Modulation Classification

- The model is a neural network classifier comprising two convolutional layers and two linear layers.
- The dataset is the *DeepSIG: RadioML 2016.10A*[14] data set with 30% of the samples affected by interference.
- Testing is done on a clean data set.
- We evaluate the final model in terms of *accuracy* and *calibration*.

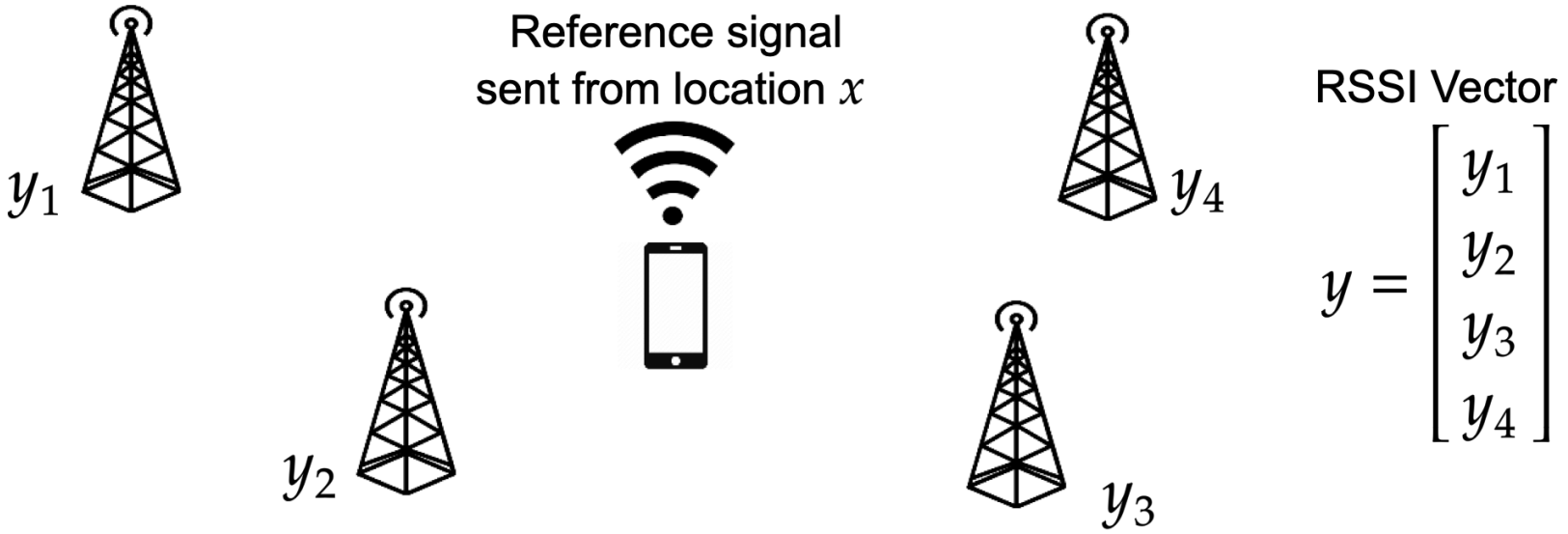
14 T. j O'Shea, et al, "Convolutional Radio Modulation Recognition Networks " arXiv:1602.04105, 2016.

Robust Bayesian Learning: Automatic Modulation Classification

- Robust Bayesian learning can improve calibration for $t < 1$, while also enhancing accuracy with $m > 1$ ($\beta = 0.01$).

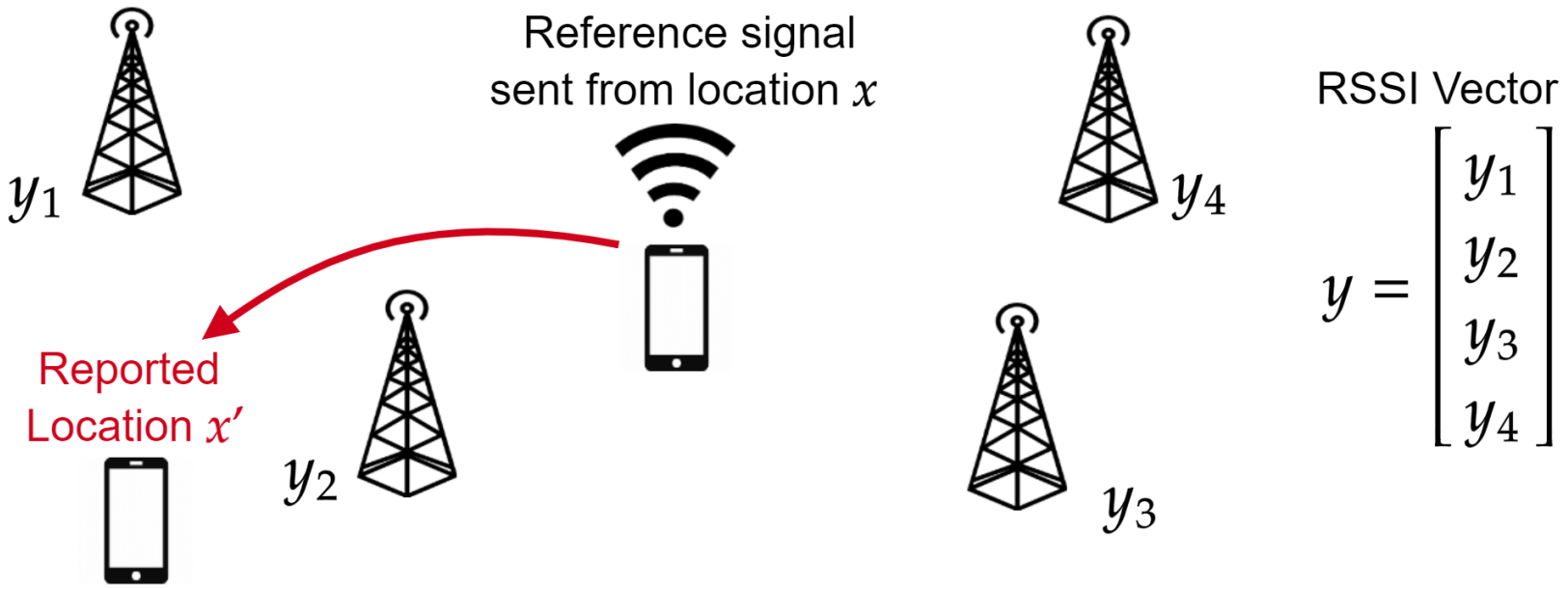


Robust Bayesian Learning: RSSI Based Localization



- Determine the location y of a transmitter based on received signal strength indicator (RSSI) vector x measured at different base stations.

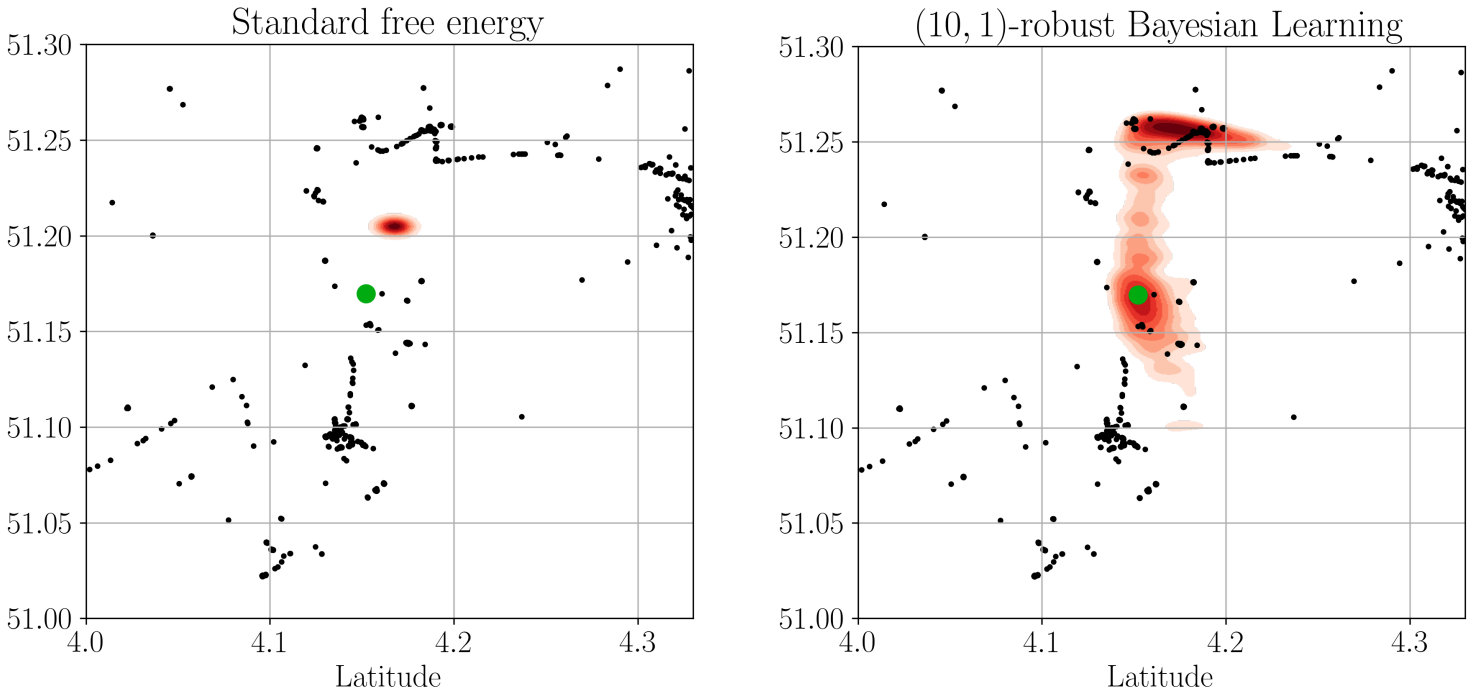
Robust Bayesian Learning: RSSI Based Localization



- **Outliers** are modelled by replacing an ϵ-fraction of the true labels y with a random location (e.g., malicious or inaccurate reporting).

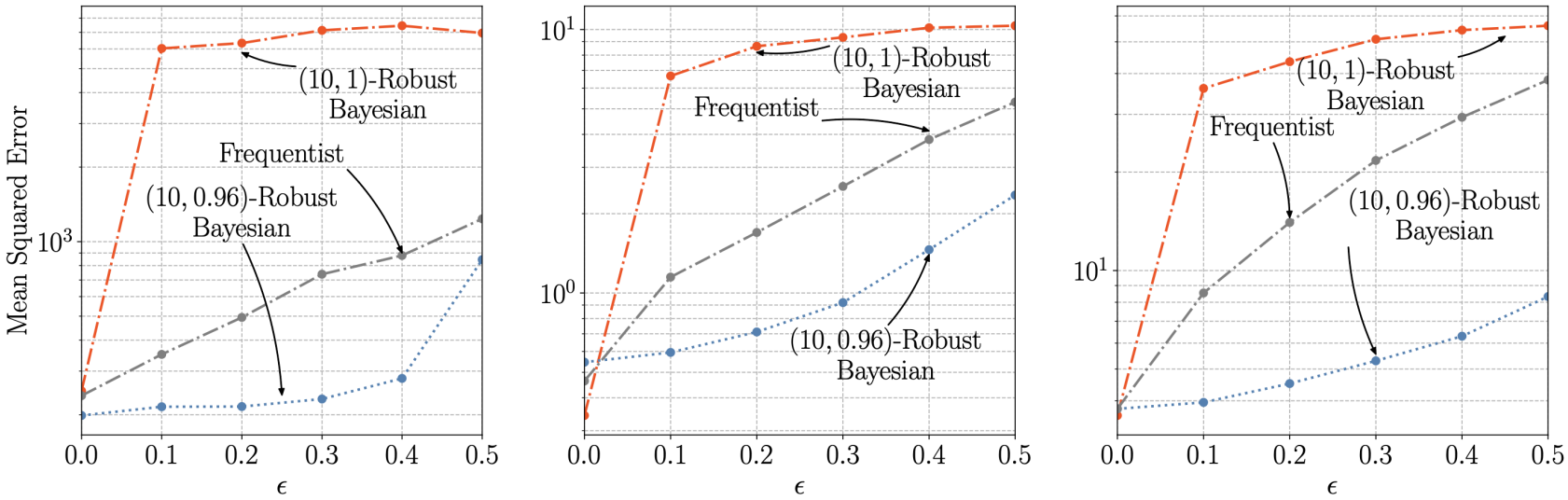
Robust Bayesian Learning: RSSI Based Localization

- We consider a model class $p(y|x,\theta) = \mathcal{N}(y|f_\theta(x), 0.01)$ where $f_\theta(x)$ is the output of a neural network.
- The model class is misspecified whenever the device location conditioned on the RSSI vector is not Gaussian distributed.
- $(m, 1)$-robust Bayesian learning mitigates model misspecification.



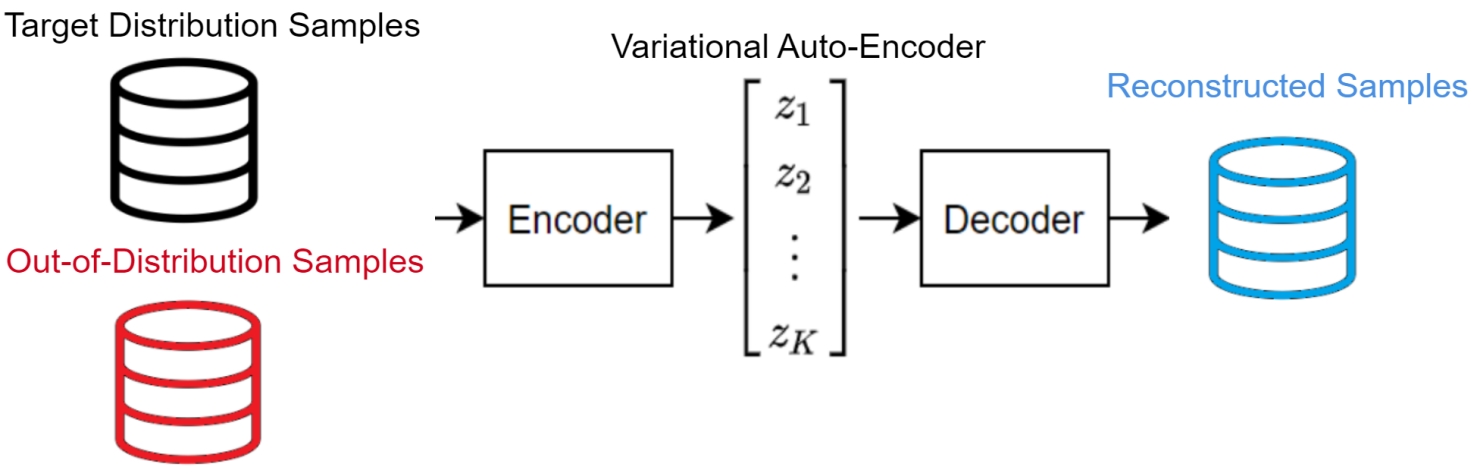
Robust Bayesian Learning: RSSI Based Localization

- (m, t)-robust Bayesian learning with $t < 1$ mitigates performance degradation due to outliers.



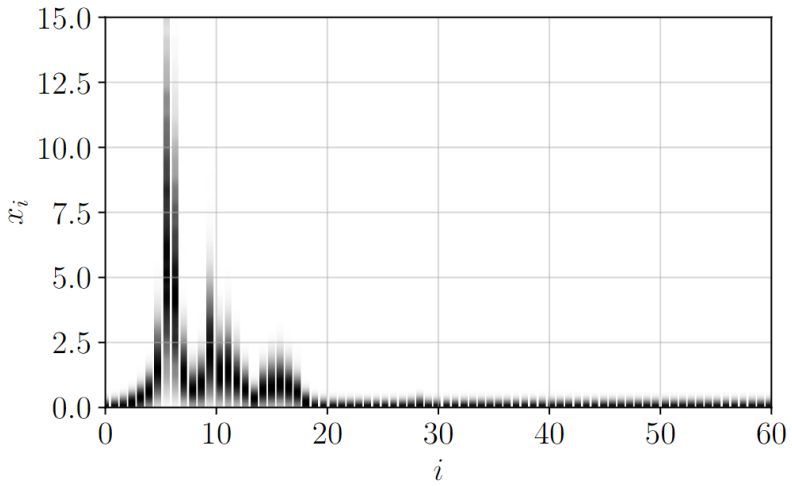
(a) *SigfoxRural* (b) *UTSIndoor* (c) *UJIIndoor*

Robust Bayesian Learning: Channel Simulation

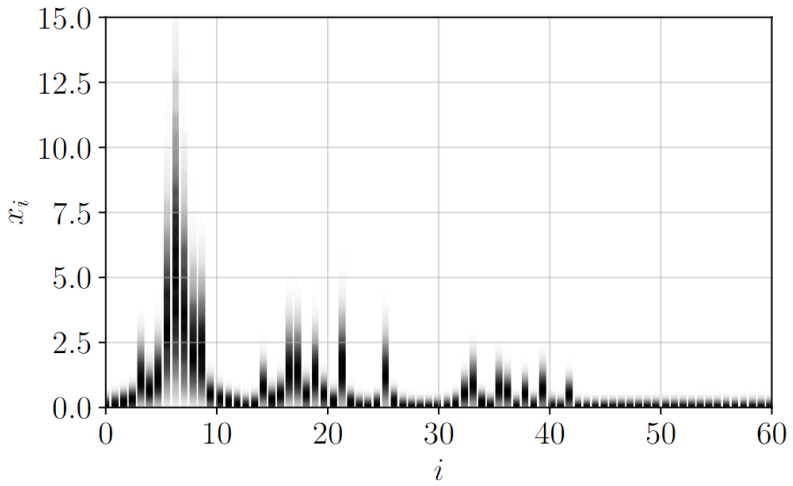


- Given a training dataset of channel responses x, train a generative model that is able to simulate new samples approximately distributed as the target channel model.
- We consider a training dataset comprising **outliers** from a different channel model.

Robust Bayesian Learning: Channel Simulation



(a) TDL-A $\tau = 100$ns

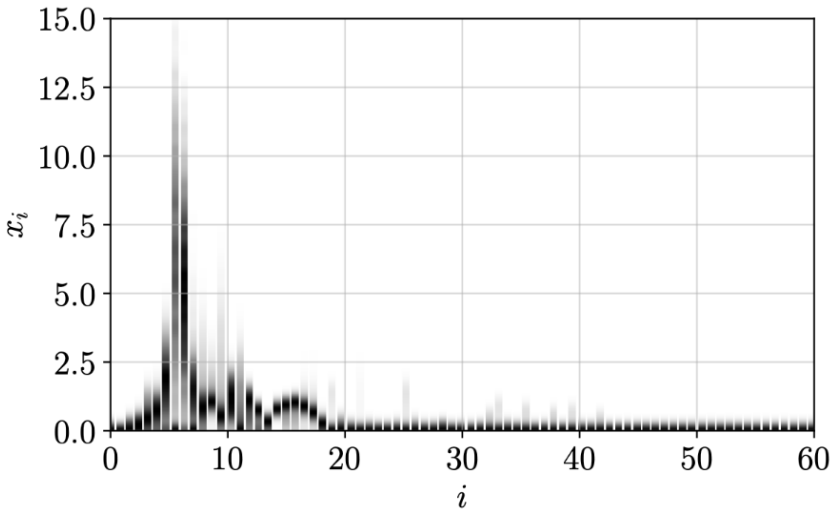


(b) TDL-A $\tau = 300$ns

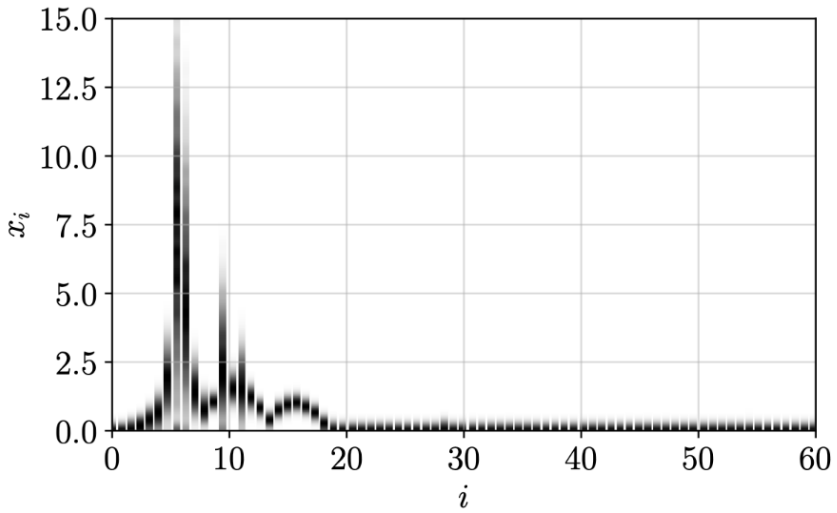
- Target (ID) distribution: TDL-A channel model with delay $\tau = 100$ ns
- Outliers (OOD) distribution: TDL-A channel model with a longer delay spread $\tau = 300$ ns

Robust Bayesian Learning: Channel Simulation

- We train a **variational autoencoder (VAE)** using the corrupted data set with $\epsilon = 0.2$, and use the generative model to generate new samples.



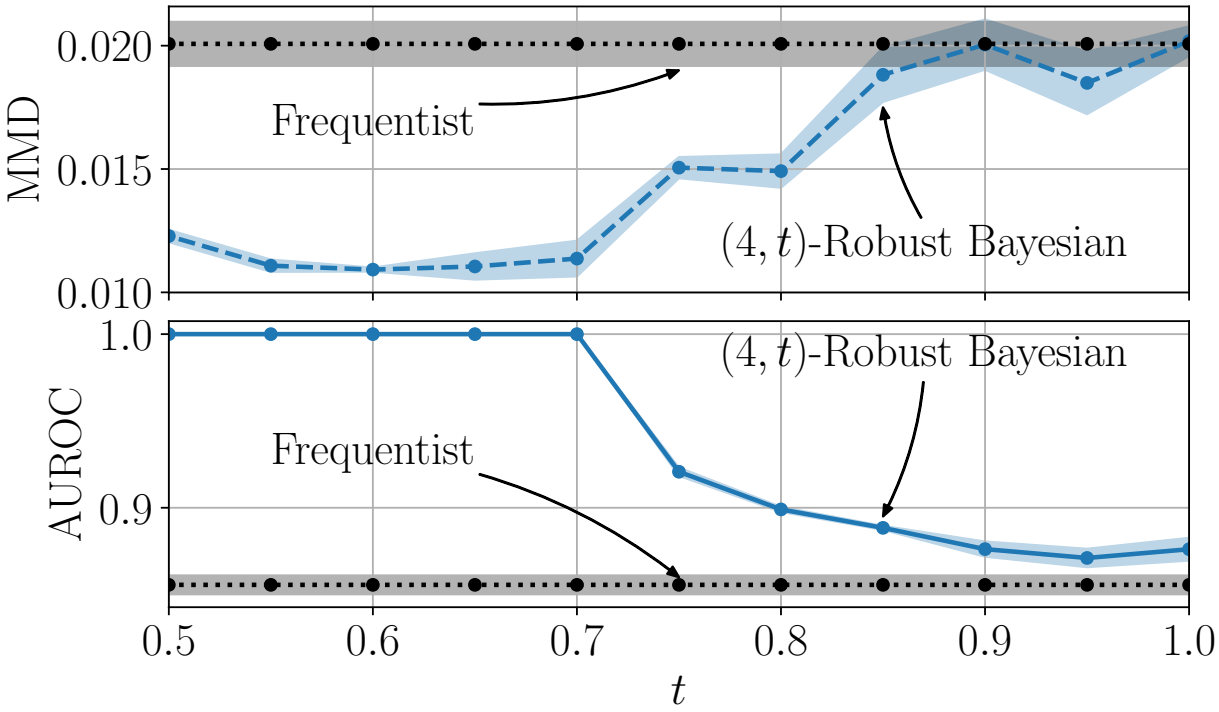
(c) Frequentist Learning



(d) $(4, 0.7)$-Robust Bayesian Learning

Robust Bayesian Learning: Channel Simulation

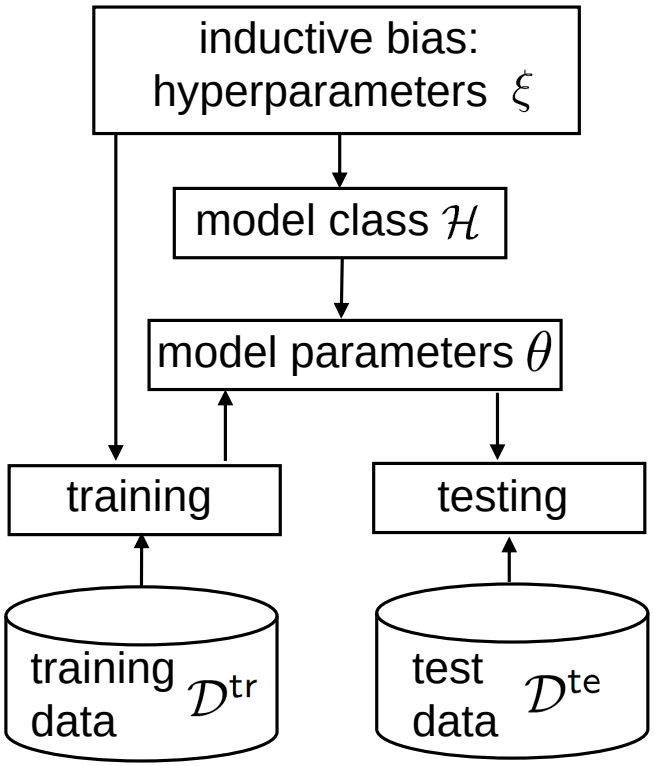
- Performance in terms of maximum mean discrepancy (MMD) between true and generated distributions, and in terms of area under the receiver operating curve (AUROC):
  - (m, t)-robust Bayesian learning with $t < 1$ yields higher accuracy in the generative model and better out-of-distribution detection capabilities.



Sample-efficient AI: Meta-Learning

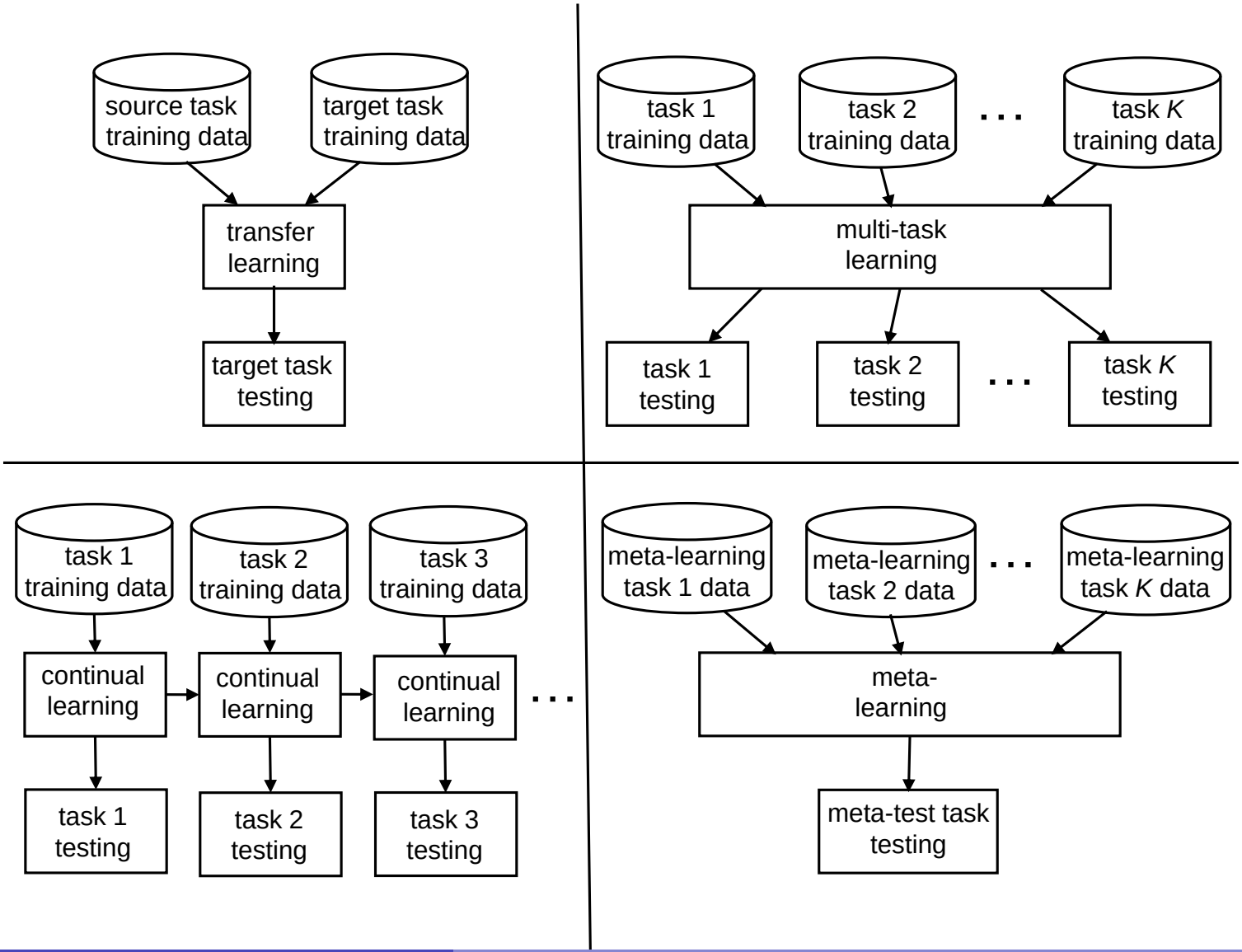
Conventional Learning

- Conventional machine learning may require excessive training data, particularly in settings with time-varying conditions
- Meta-learning provide tools to reduce sample complexity by transferring knowledge from other learning tasks



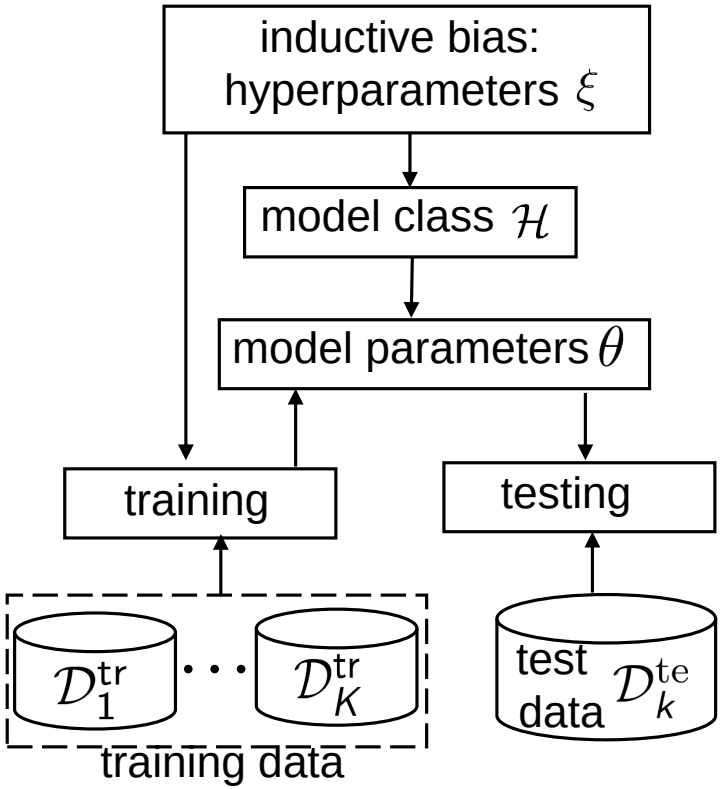
Transferring Knowledge Across Tasks

- There are several ways to formalize the problem of knowledge transfer across tasks.



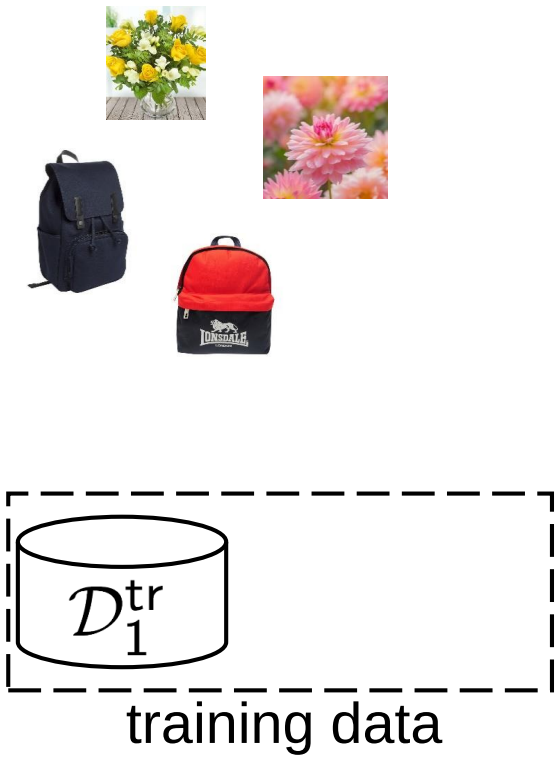
Joint Learning

- For reference, let us first consider joint learning as a simplified form of multi-task learning.
- Joint learning trains a shared model across K tasks, and tests the model on any one of the K tasks.



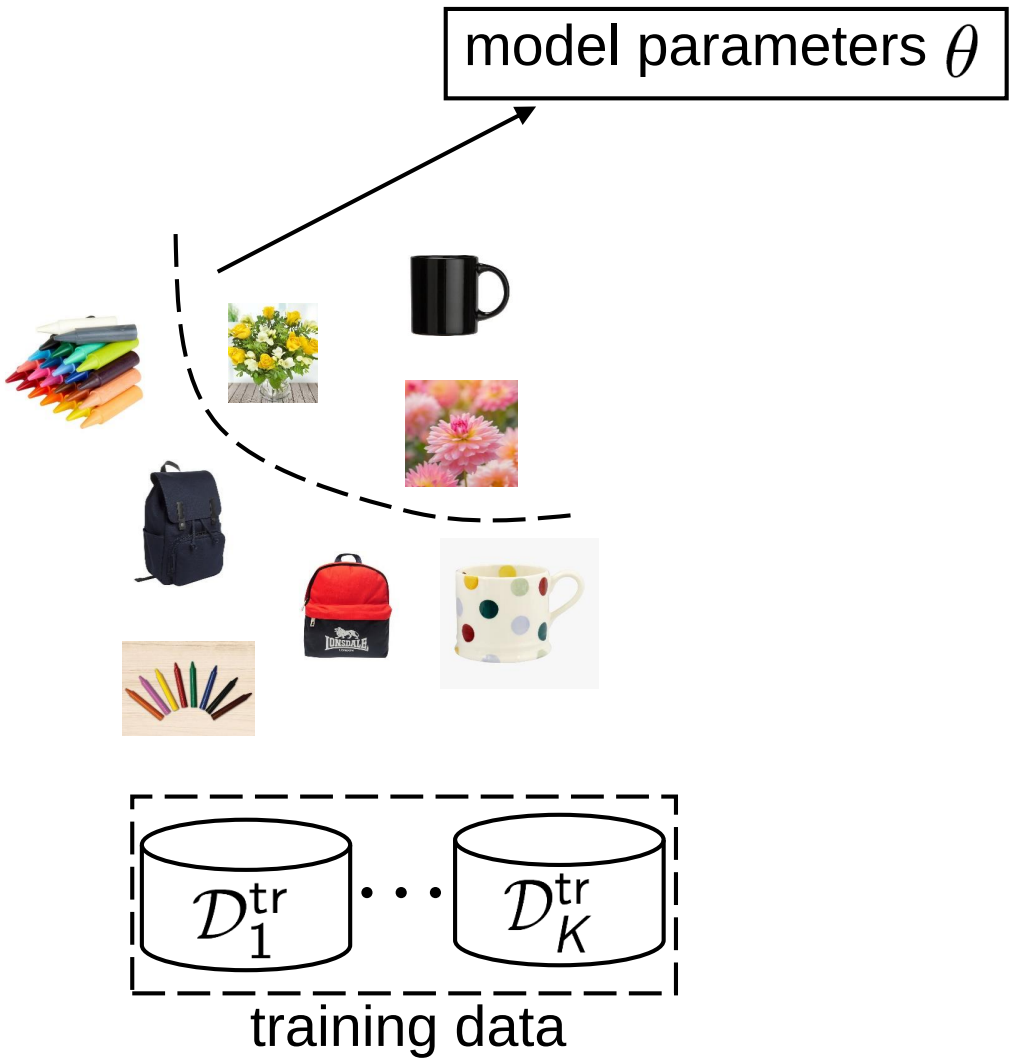
Joint Learning: Example

- Binary image classification task



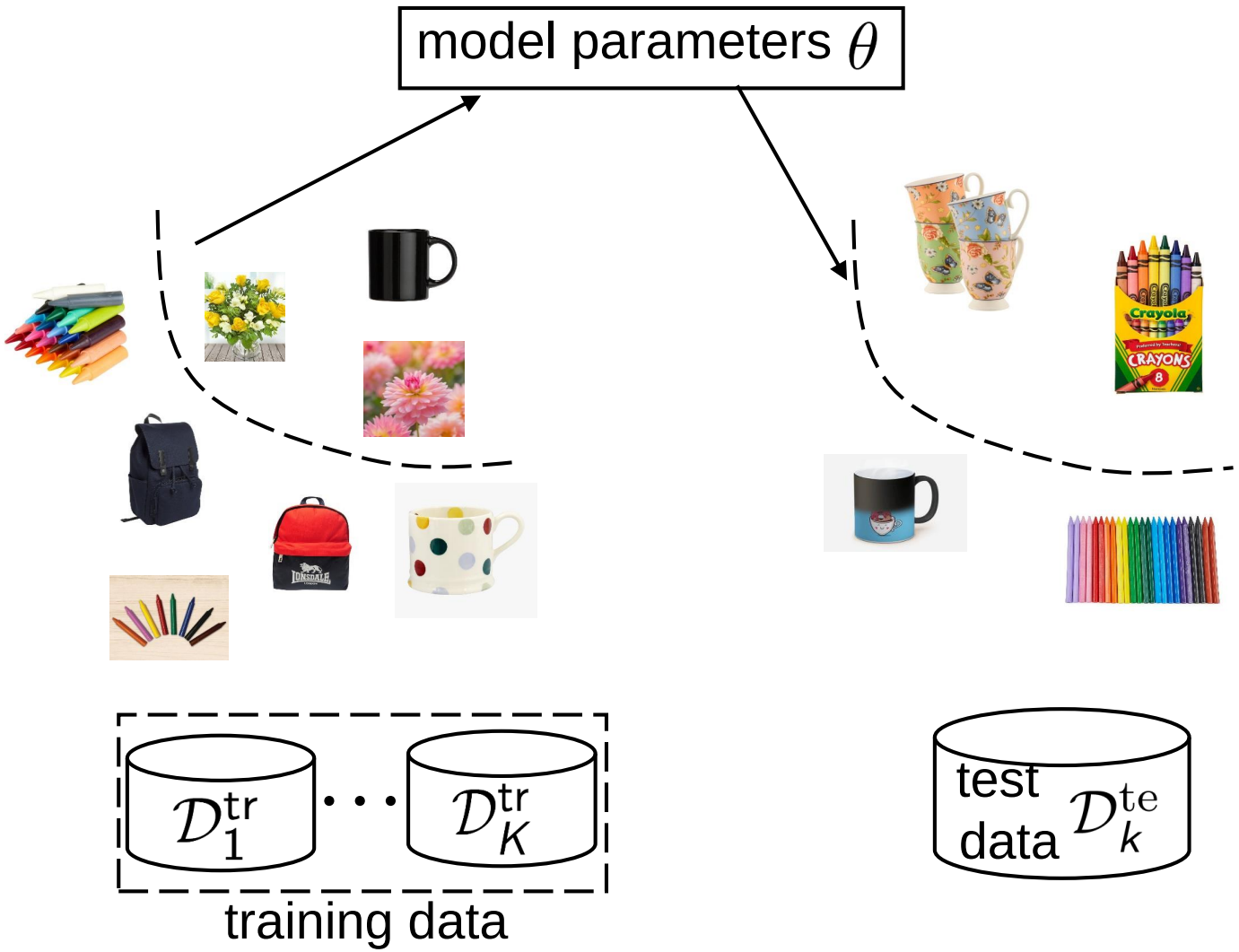
Joint Learning: Example

- $K = 2$ binary image classification tasks



Joint Learning: Example

- $K = 2$ binary image classification tasks



Joint Learning

- Joint learning addresses the problem of minimizing the joint training loss across the K tasks

$$L_{\{\mathcal{D}_k\}_{k=1}^K}(\theta) = \frac{1}{K}\sum_{k=1}^{K} L_{\mathcal{D}_k^{\text{tr}}}(\theta)$$

  over the shared model parameter θ.

- Key advantage:
  - By pooling together data from K tasks, the overall size of the training set is $K \cdot N$, which may be large even when the available data per task is limited, i.e., when N is small.

Joint Learning

- Joint learning addresses the problem of minimizing the joint training loss across the K tasks

$$L_{\{\mathcal{D}_k\}_{k=1}^K}(\theta) = \frac{1}{K}\sum_{k=1}^{K} L_{\mathcal{D}_k^{\text{tr}}}(\theta)$$

  over the shared model parameter θ.

- Key advantage:
  - By pooling together data from K tasks, the overall size of the training set is $K \cdot N$, which may be large even when the available data per task is limited, i.e., when N is small.

Joint Learning

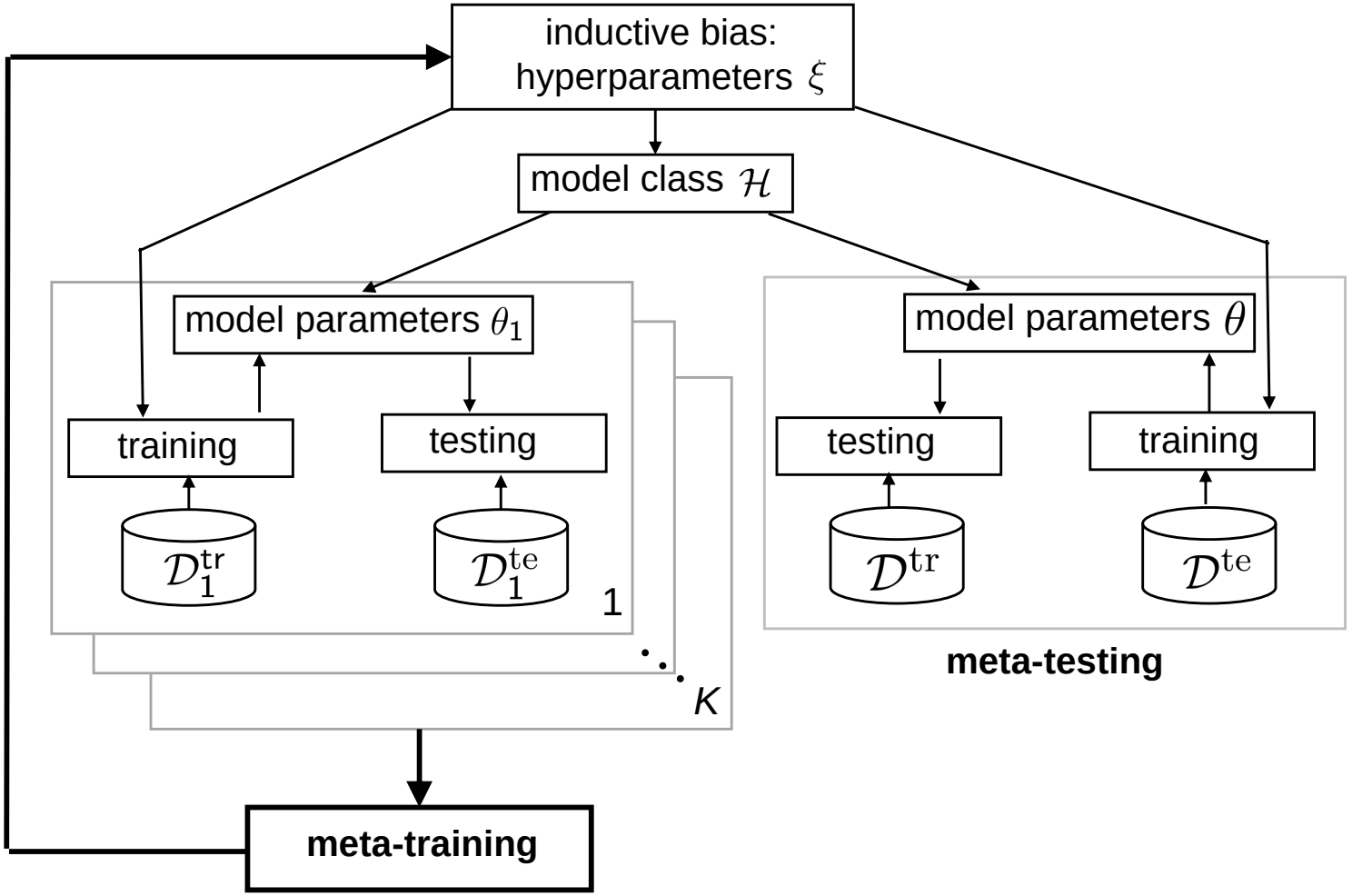
- But joint learning has two potentially critical shortcomings:
  - The jointly trained model only works if there is a single model parameter θ that "works well" for all tasks.
  - Even if there is a single model parameter θ that yields desirable test results on all K tasks, this does not guarantee that the same is true for a new task.
- By focusing on training a common model, joint learning is not designed to enable adaptation based on training data for a new task.
- One could use the jointly trained model parameter θ to initialize the training process on a new task (a process known as **fine-tuning**), but there is no guarantee that this would yield a desirable outcome.

Joint Learning

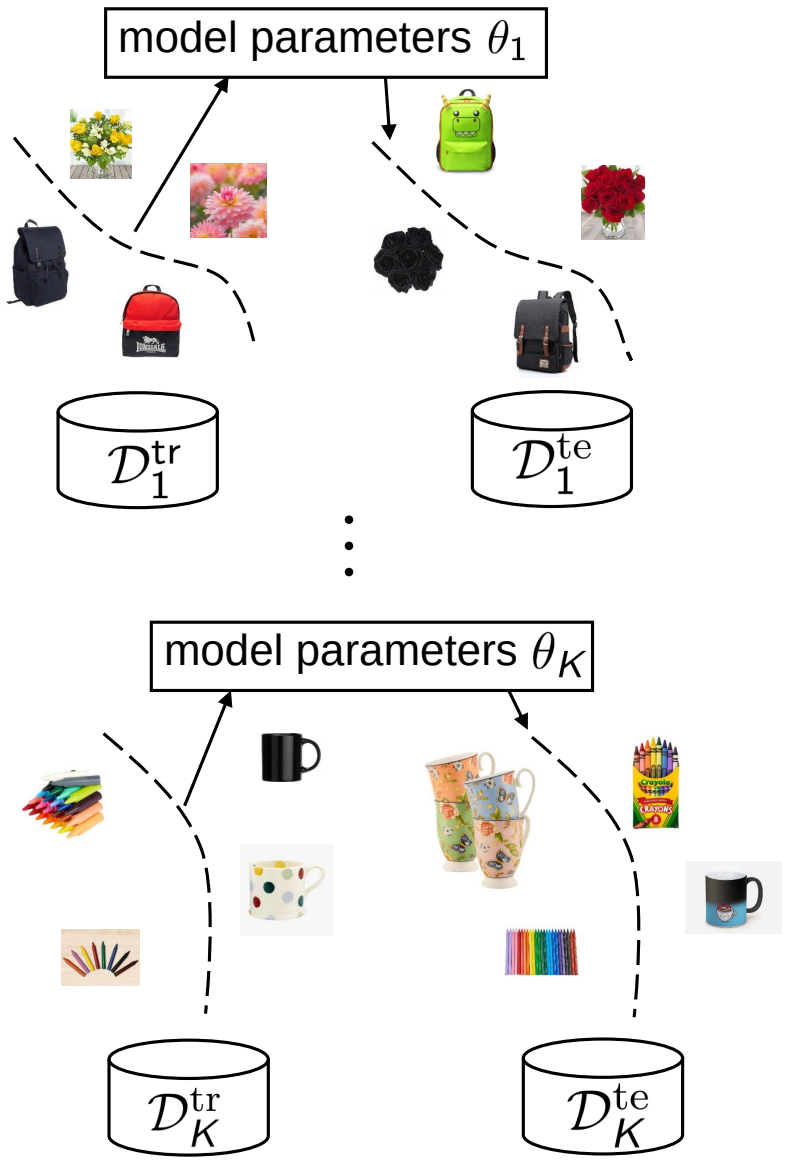
- But joint learning has two potentially critical shortcomings:
  - The jointly trained model only works if there is a single model parameter θ that "works well" for all tasks.
  - Even if there is a single model parameter θ that yields desirable test results on all K tasks, this does not guarantee that the same is true for a new task.
- By focusing on training a common model, joint learning is not designed to enable adaptation based on training data for a new task.
- One could use the jointly trained model parameter θ to initialize the training process on a new task (a process known as **fine-tuning**), but there is no guarantee that this would yield a desirable outcome.

Meta-Learning

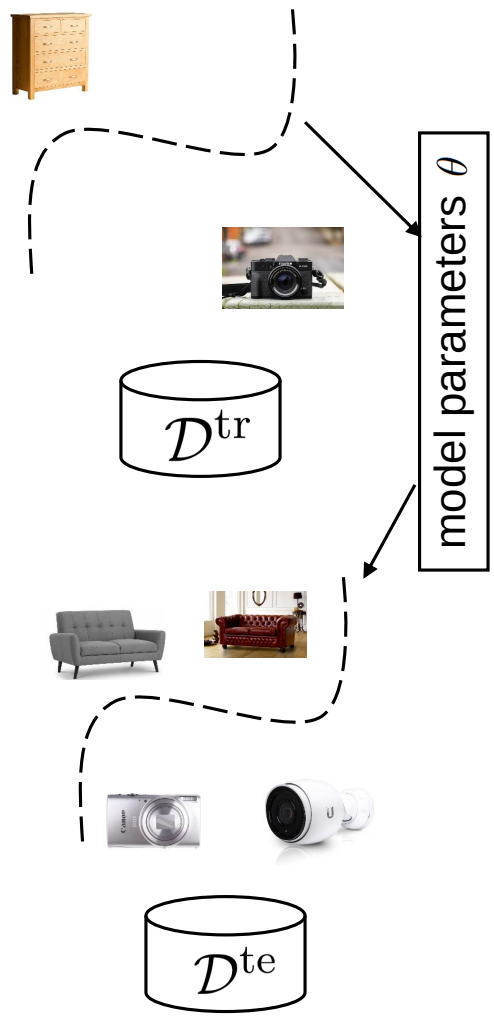
- Meta-learning optimizes **shared hyperparameters**, while enabling **adaptation of the model parameters** for each task.



Meta-Learning: Example



meta-training



meta-testing

Meta-Learning

- Let us fix a training algorithm mapping a training set $\mathcal{D}$ to a model parameter vector θ

$$\theta = \theta^{\mathrm{tr}}(\mathcal{D}|\xi).$$

- The algorithm depends on a hyperparameter vector ξ.
- As an example, $\theta^{\mathrm{tr}}(\mathcal{D}|\xi)$ can be the output of I iterations of gradient descent (GD) given initialization $\theta^{(1)}$ and learning rates $\gamma^{(i)}$, determined by hyperparameters $\xi = (\theta^{(1)}, \{\gamma^{(i)}\}_{i=1}^{I})$.

Meta-Learning

- Let us fix a training algorithm mapping a training set $\mathcal{D}$ to a model parameter vector θ

$$\theta = \theta^{\mathrm{tr}}(\mathcal{D}|\xi).$$

- The algorithm depends on a hyperparameter vector ξ.
- As an example, $\theta^{\mathrm{tr}}(\mathcal{D}|\xi)$ can be the output of I iterations of gradient descent (GD) given initialization $\theta^{(1)}$ and learning rates $\gamma^{(i)}$, determined by hyperparameters $\xi = (\theta^{(1)}, \{\gamma^{(i)}\}_{i=1}^{I})$.

Meta-Learning

- Meta-learning aims at minimizing the average test loss on the meta-training tasks

$$\mathcal{L}_{\{\mathcal{D}_k^{\text{tr}}\}_{k=1}^K}(\xi) = \frac{1}{K}\sum_{k=1}^{K} L_{\mathcal{D}_k^{\text{te}}}(\theta^{\text{tr}}(\mathcal{D}_k^{\text{tr}}|\xi)).$$

- The resulting minimization problem
  - is over the hyperparameter vector ξ and not over a shared model parameter θ;
  - and accounts for the performance of model parameters adapted to the data of each task k.

Meta-Learning

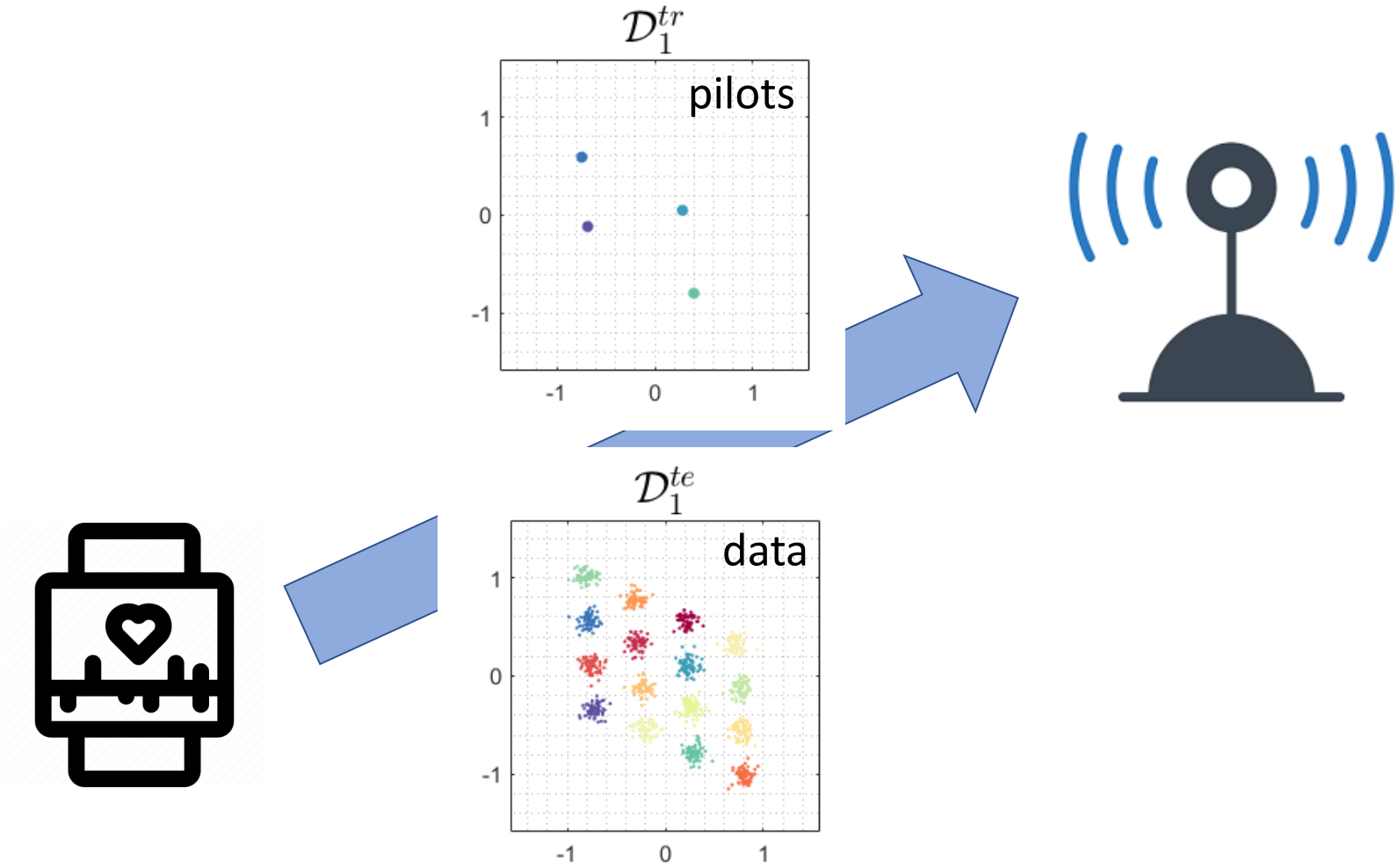
- Meta-learning aims at minimizing the average test loss on the meta-training tasks

$$\mathcal{L}_{\{\mathcal{D}_k^{\text{tr}}\}_{k=1}^K}(\xi) = \frac{1}{K}\sum_{k=1}^{K} L_{\mathcal{D}_k^{\text{te}}}(\theta^{\text{tr}}(\mathcal{D}_k^{\text{tr}}|\xi)).$$

- The resulting minimization problem
  - is over the hyperparameter vector ξ and not over a shared model parameter θ;
  - and accounts for the performance of model parameters adapted to the data of each task k.

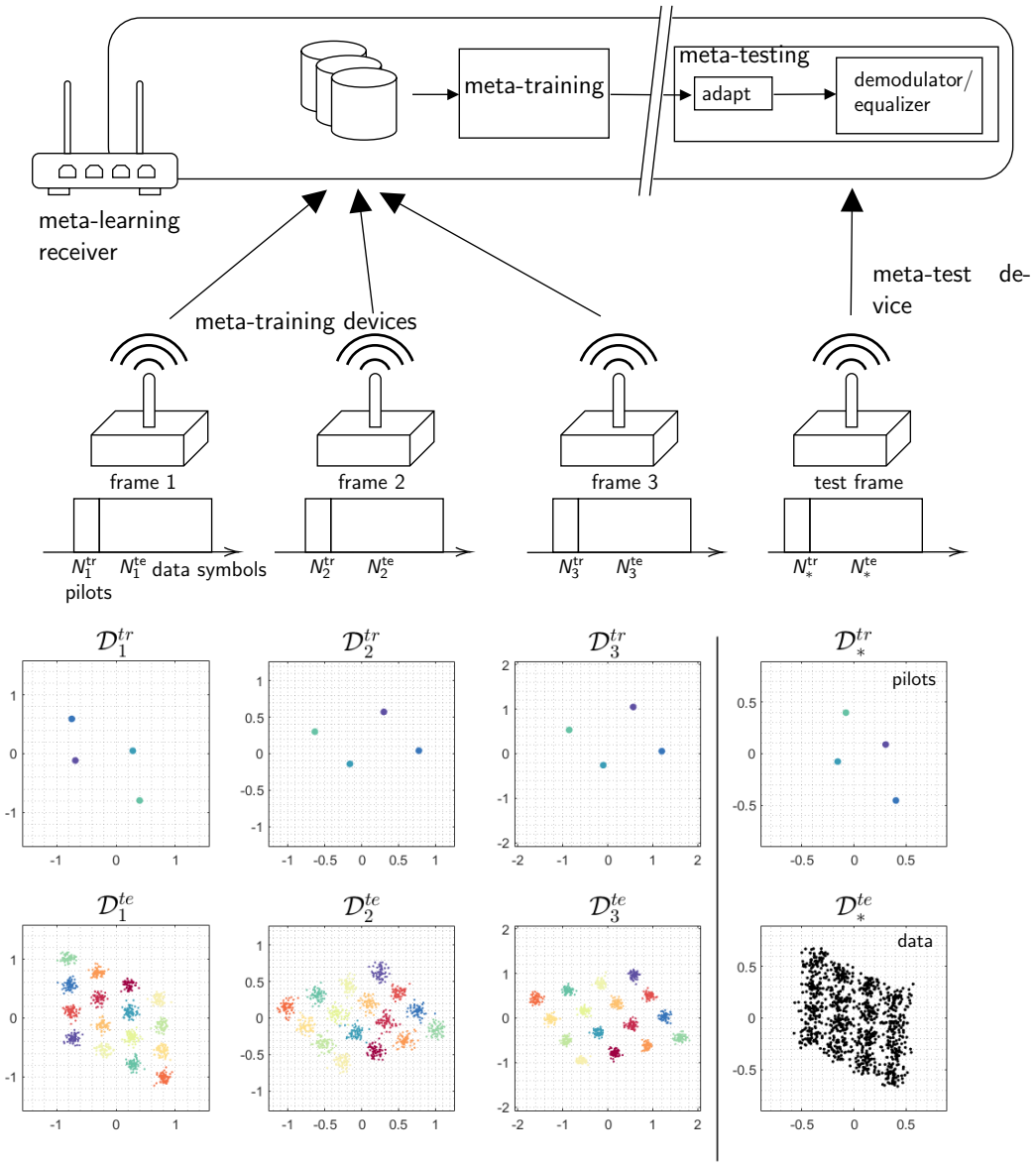
Application to Demodulation

- Short-packet transmission with I/Q imbalance[15]



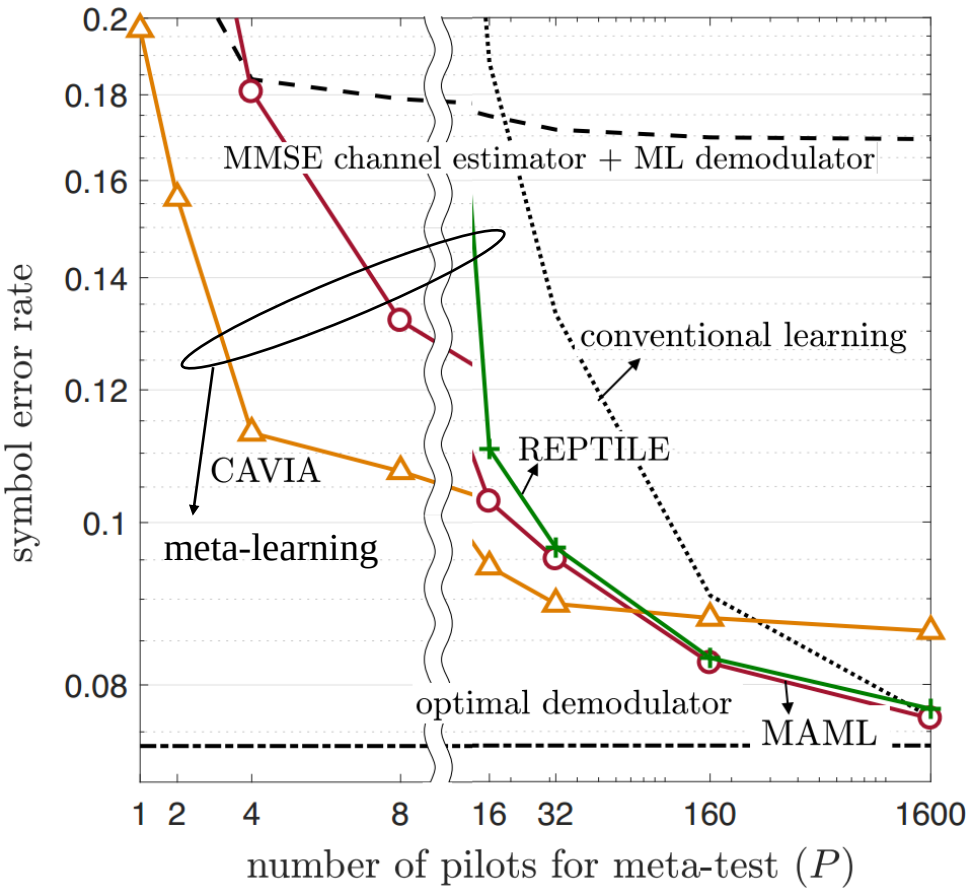
15 S. Park, H. Jang, and O. Simeone, "Learning to demodulate from few pilots via offline and online meta-learning," IEEE Transactions on Signal Processing, 2020.

Application to Demodulation



Application to Demodulation

- Symbol error rate vs. number of pilots at run time



- Meta-learning can reduce pilot requirements.

Reliable and Sample-Efficient AI: Bayesian Meta-Learning

Integrating Bayesian Learning and Meta-Learning

- Recall that, given a prior $p_0(\theta)$, for each learning task k, Bayesian learning aims at minimizing the *free energy*

$$F_{\mathcal{D}_k}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}_k}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p_0(\theta)\right)}_{\text{information-theoretic regularization}}$$

- With VI, minimization is done over the parameters φ of a variational distribution $q(\theta|\varphi)$ (e.g., Gaussian).
- Hyperparameters ξ may determine
  - the prior $p_0(\theta|\xi)$
  - the optimizer over φ (e.g., initialization)

Integrating Bayesian Learning and Meta-Learning

- Recall that, given a prior $p_0(\theta)$, for each learning task k, Bayesian learning aims at minimizing the *free energy*

$$F_{\mathcal{D}_k}(q(\theta)) = N \underbrace{\mathrm{E}_{\theta\sim q(\theta)}[L_{\mathcal{D}_k}(\theta)]}_{\text{average training loss}} + \beta \underbrace{\mathrm{KL}\left(q(\theta)||p_0(\theta)\right)}_{\text{information-theoretic regularization}}$$

- With VI, minimization is done over the parameters φ of a variational distribution $q(\theta|\varphi)$ (e.g., Gaussian).
- Hyperparameters ξ may determine
  - the prior $p_0(\theta|\xi)$
  - the optimizer over φ (e.g., initialization)

Integrating Bayesian Learning and Meta-Learning

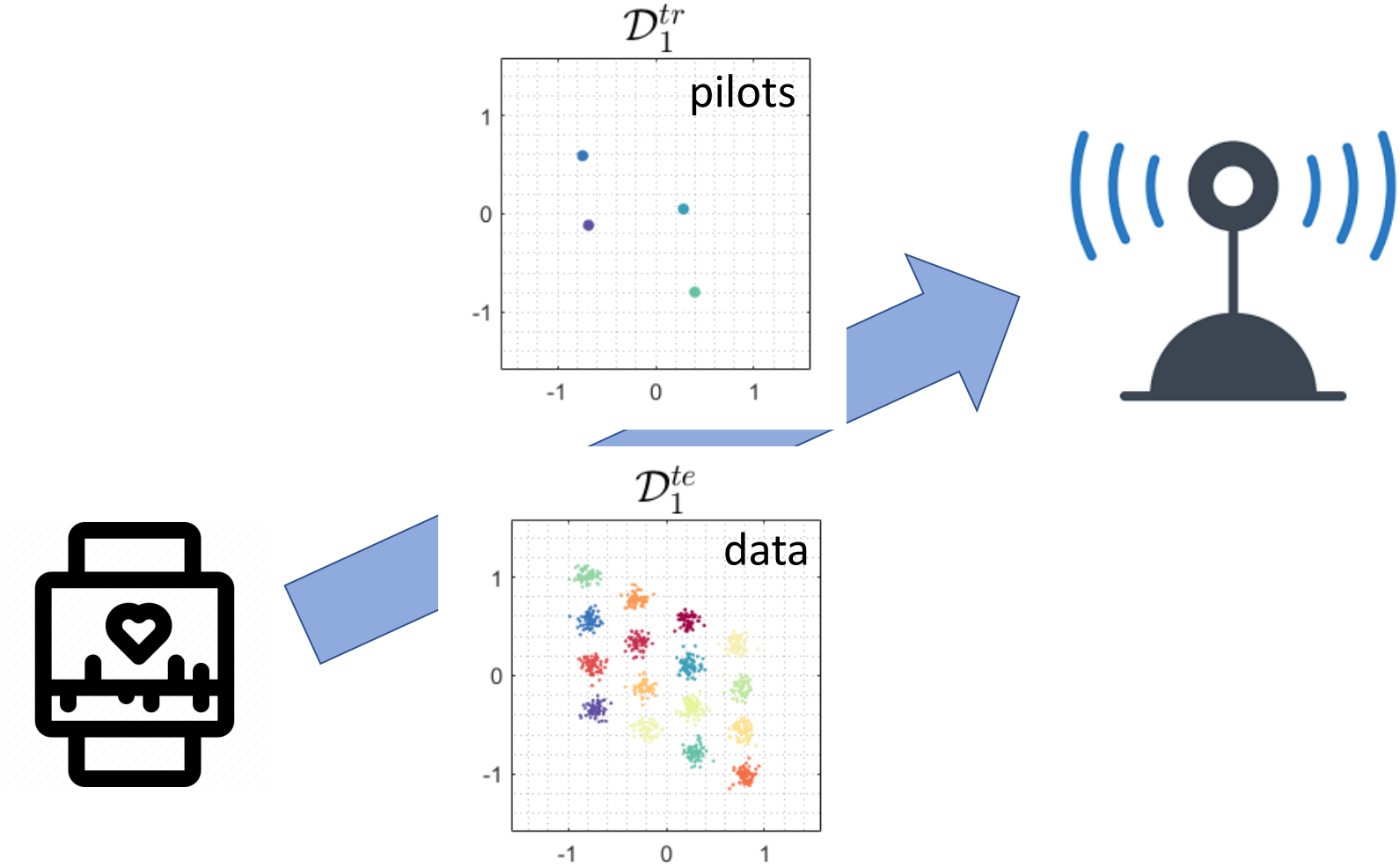
- Accordingly, we obtain a (variational) posterior distribution $q^{\mathrm{tr}}(\theta|\mathcal{D}_k^{\mathrm{tr}}, \xi)$ for task k given data $\mathcal{D}_k^{\mathrm{tr}}$ and hyperparameter vector ξ.
- Given data from K tasks, Bayesian meta-learning can be formulated as the minimization of the aggregate average training loss

$$\mathcal{F}_{\{\mathcal{D}_k\}_{k=1}^K}(\xi) = \frac{1}{K}\sum_{k=1}^{K} \mathrm{E}_{\theta\sim q^{\mathrm{tr}}(\theta|\mathcal{D}_k^{\mathrm{tr}},\xi)}[L_{\mathcal{D}_k^{\mathrm{te}}}(\theta)],$$

  where $q^{\mathrm{tr}}(\theta|\mathcal{D}_k^{\mathrm{tr}}, \xi)$ is a (variational) posterior optimized for task k given data $\mathcal{D}_k^{\mathrm{tr}}$ and hyperparameter vector ξ.
- Similar training-test splits have been applied as discussed for frequentist learning.

Application to Demodulation

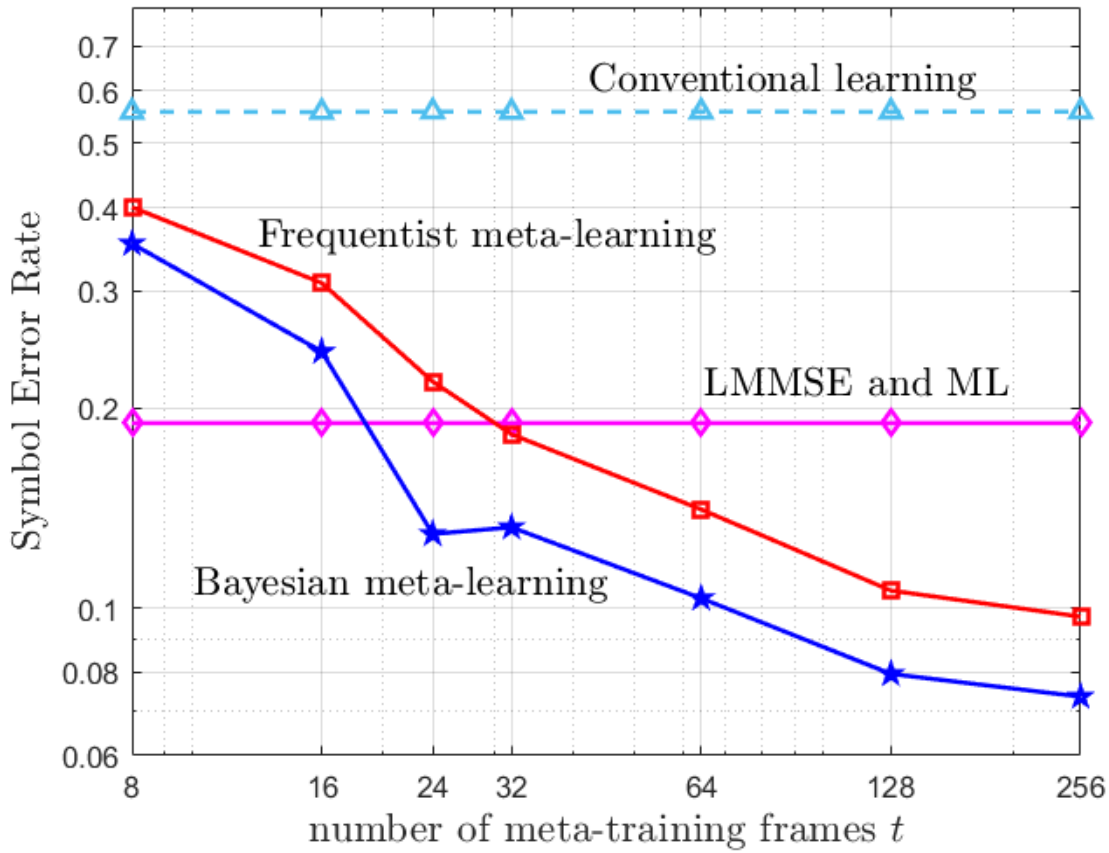
- Short-packet transmission with I/Q imbalance[16]



16 S. Park, H. Jang, and O. Simeone, "Learning to demodulate from few pilots via offline and online meta-learning," IEEE Transactions on Signal Processing, 2020.

Application to Demodulation

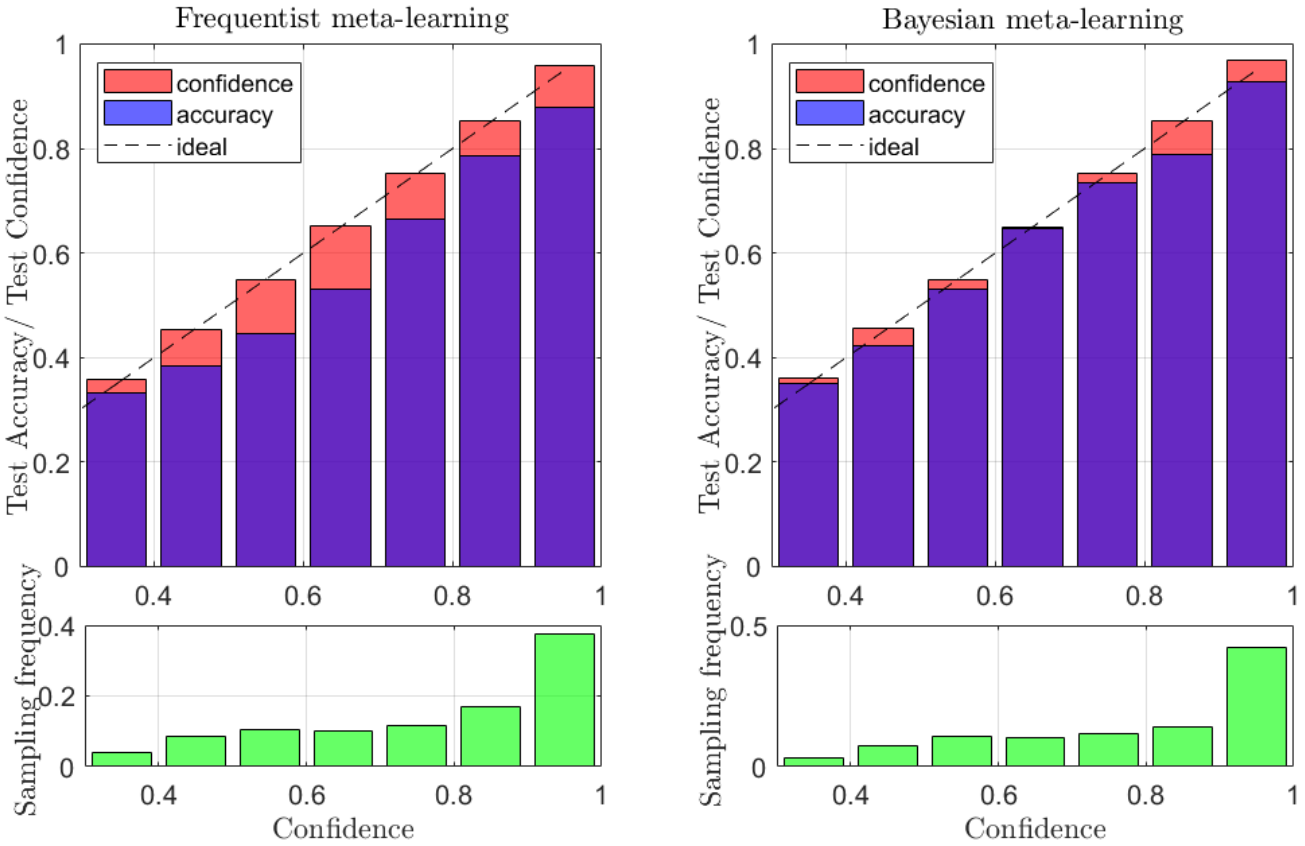
- Symbol error rate vs. number of meta-training frames[17]
- $\mathrm{SNR} = 18\mathrm{dB}$, 16 meta-frames and 8 pilots



17 K. Cohen, et al, "Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning," in Proc. WSA, 2021.

Application to Demodulation

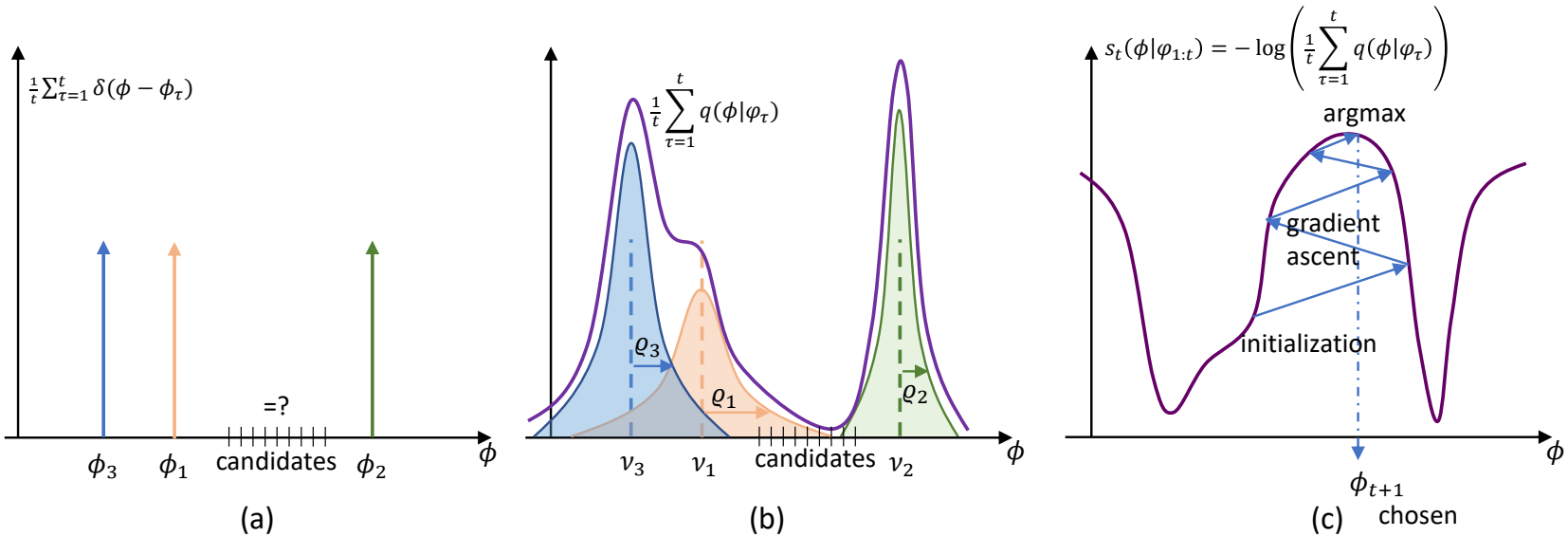
- Reliability plots (with 16 meta-training frames, 8 pilots and SNR=18 dB).
- Bayesian meta-learning is better calibrated.



Active Meta-learning

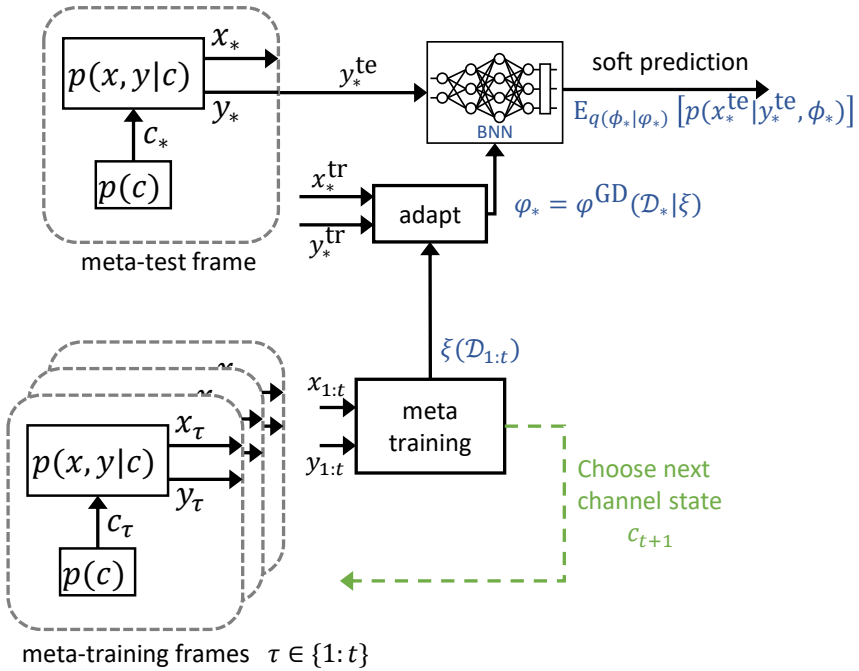
- We have assumed that previous tasks are collected at random...
- ... can we instead optimize the sequence of tasks so as to improve the efficiency of meta-learning in terms of number of tasks?

Frequentist vs. Bayesian Active Meta-Learning



- Frequentist meta-learning relies on point estimates, and is hence unable to score as-of-yet unexplored model parameters.
- Bayesian meta-learning can associate a score to each model parameter vector.
- This can be used to select the next task if we have a mechanism to map model parameters to a task.

Bayesian Active Meta-learning: Application to Equalization



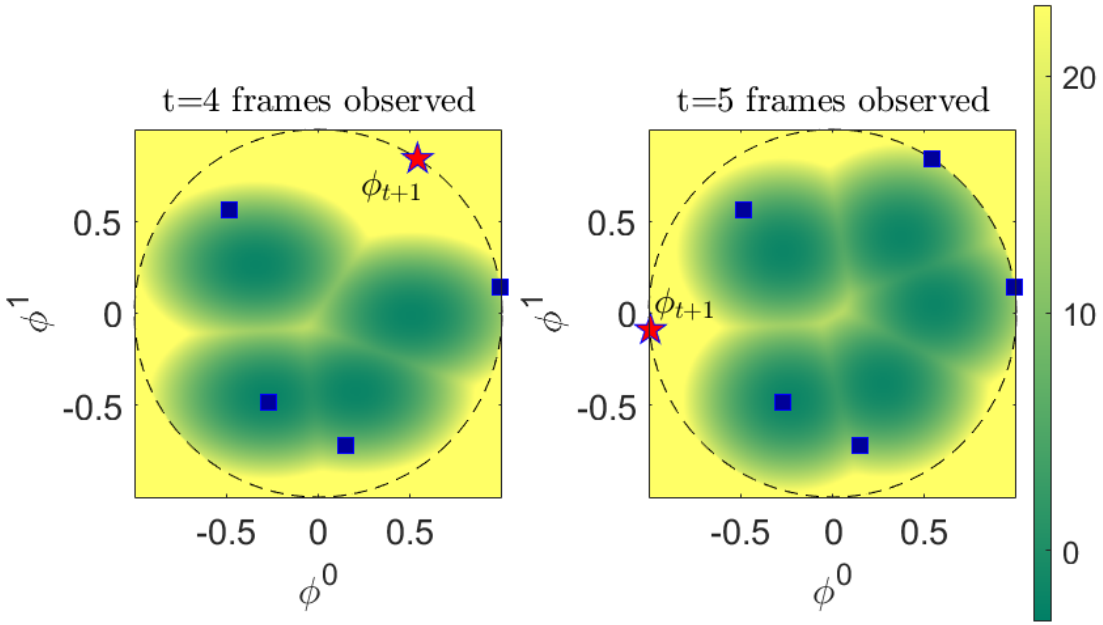
- We have access to a channel simulator and we can generate data at will in a sequential fashion.
- SIMO channel

$$y_\tau[i] = c_\tau x[i] + z_\tau[i]$$

with $c_\tau \sim \mathcal{N}(0, I_2)$

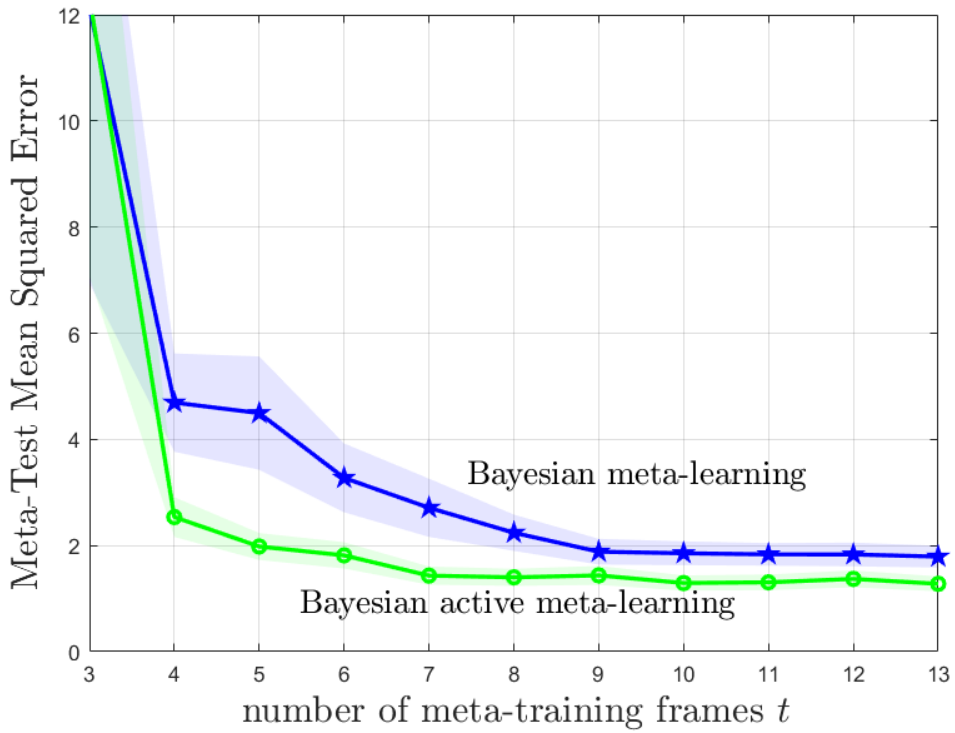
- Linear soft equalizer $\phi \in \mathbb{R}^2$ via $\hat{x}_\tau[i] = \phi_\tau^\top y_\tau[i]$ with $\mathcal{N}(\hat{x}_\tau[i], \sigma^2)$

Bayesian Active Meta-learning: Application to Equalization



- Scoring function $s_t(\phi|\varphi_{1:t}) := -\log\left(\frac{1}{t}\sum_{\tau=1}^{t} q(\phi|\varphi_\tau)\right)$
- Squares represent previously selected model parameter $\{\phi_\tau\}_{\tau=1}^{t}$.

Bayesian Active Meta-Learning: Some Results

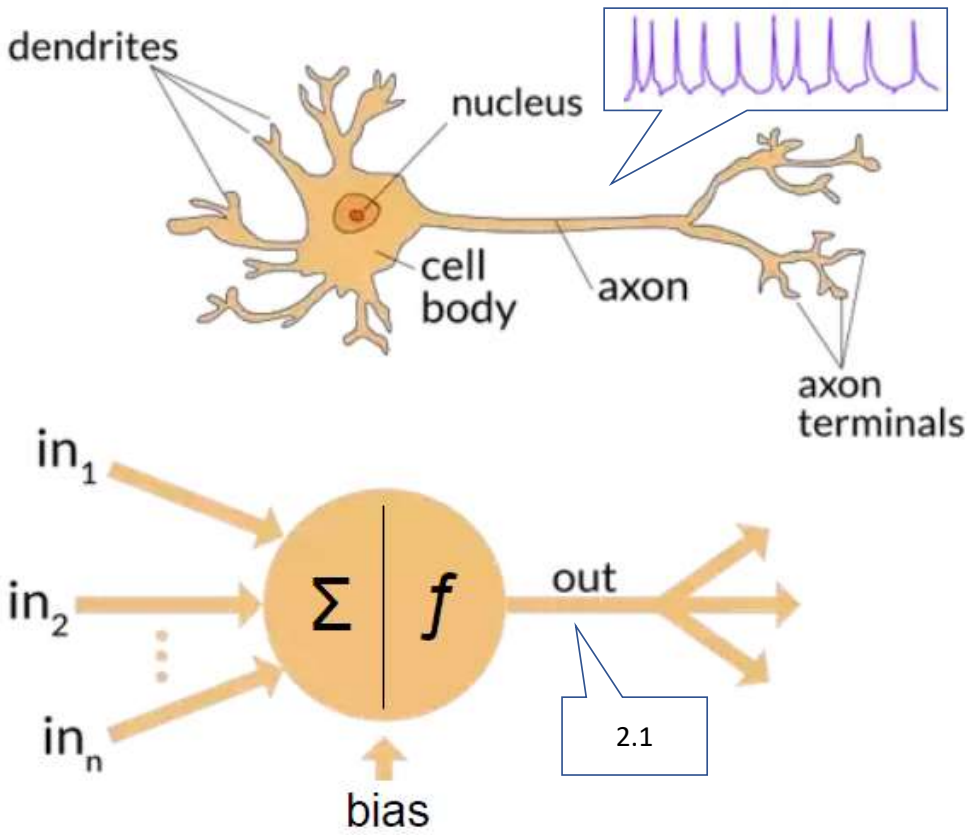


- Active meta-learning requires far few meta-training frames.
- Reduced randomness due to structural (rather than random) selection of channels.

Hardware-Efficient AI: Neuromorphic Computing

Neuromorphic Computing

- **Spiking Neural Networks (SNNs)** replace static neurons with spiking, dynamic, neuronal models.[18]



18 H. Jang, O. Simeone, B. Gardner, and A. Gruning, "An Introduction to Spiking Neural Networks," IEEE Signal Processing Magazine, 2019.

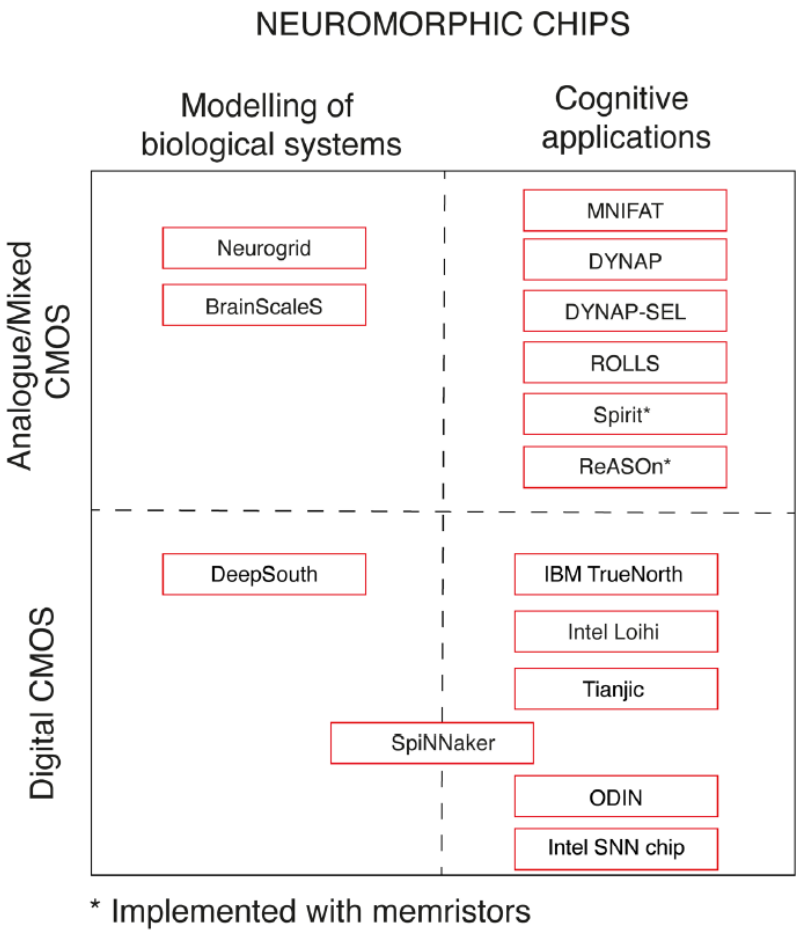
Neuromorphic Computing: Promises

- The goal is to harness the **efficiency** of biological systems, as well as their capacity for **adaptation**.
- Target use case: **device/ edge AI** (e.g., at mobile devices or DUs).

[INRC '21]

Spiking Neural Networks

- Current neuromorphic computing platforms and algorithms implement SNNs using digital or mixed analog-digital technology.



PROPERTIES:

- In-memory computing
- Fine-grained parallelism
- Learning in hardware
- Event based and asynchronous communication
- Reduced precision
- Spike-based processing
- Adaptability
- Leveraging noise and stochasticity

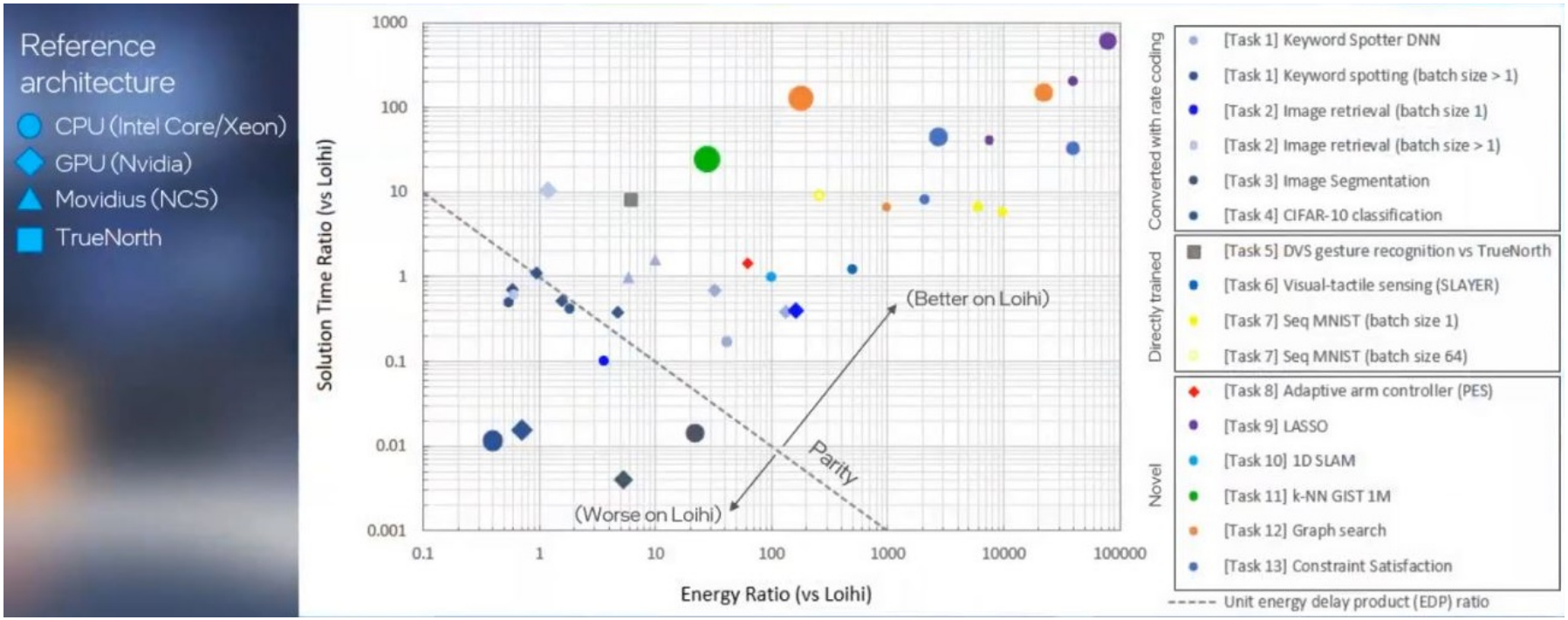
.
.
.

- Brain-inspired

[Mehonic and Kenyon '21]

Spiking Neural Networks

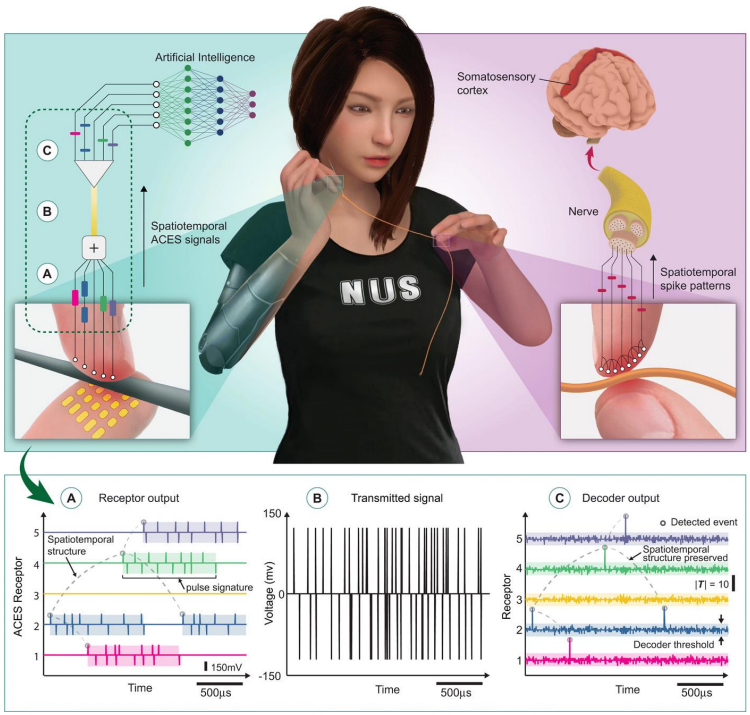
- **Orders of magnitude** gains in terms of **latency and energy** have already been shown when selecting suitable workloads.



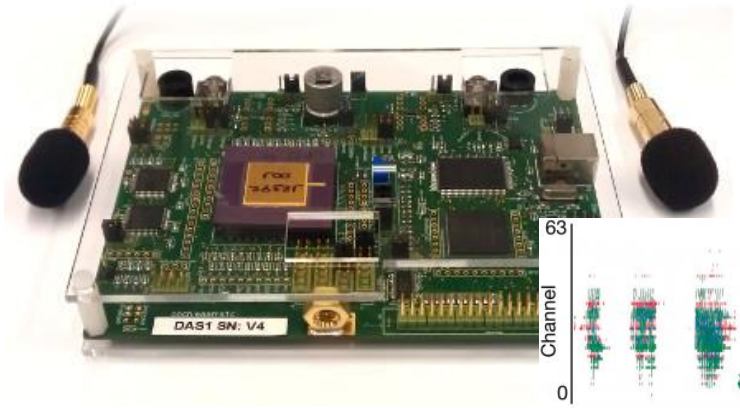
[INRC '21]

Neuromorphic Sensing

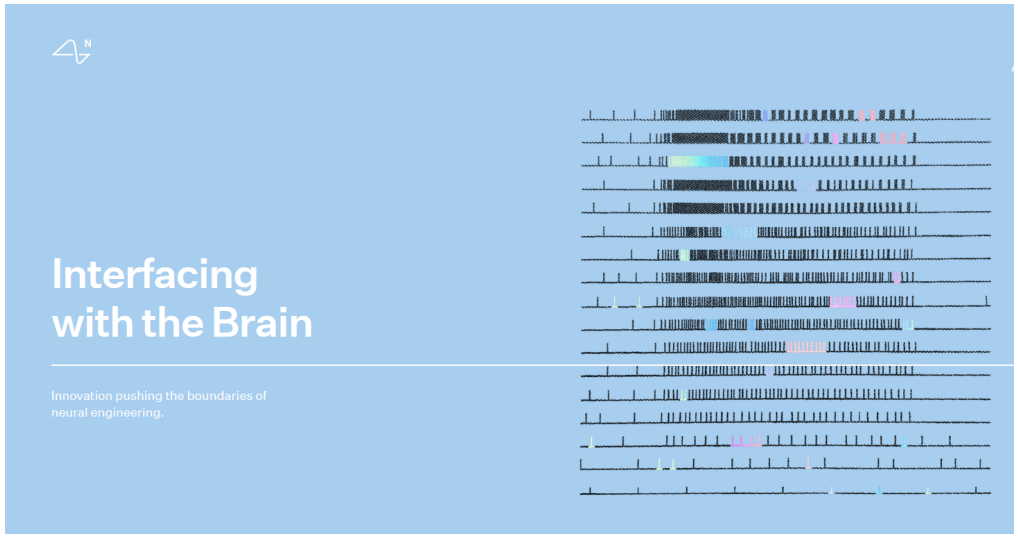
- Silicon cochlea, touch sensor, brain interface,...



[Lee et al '19]

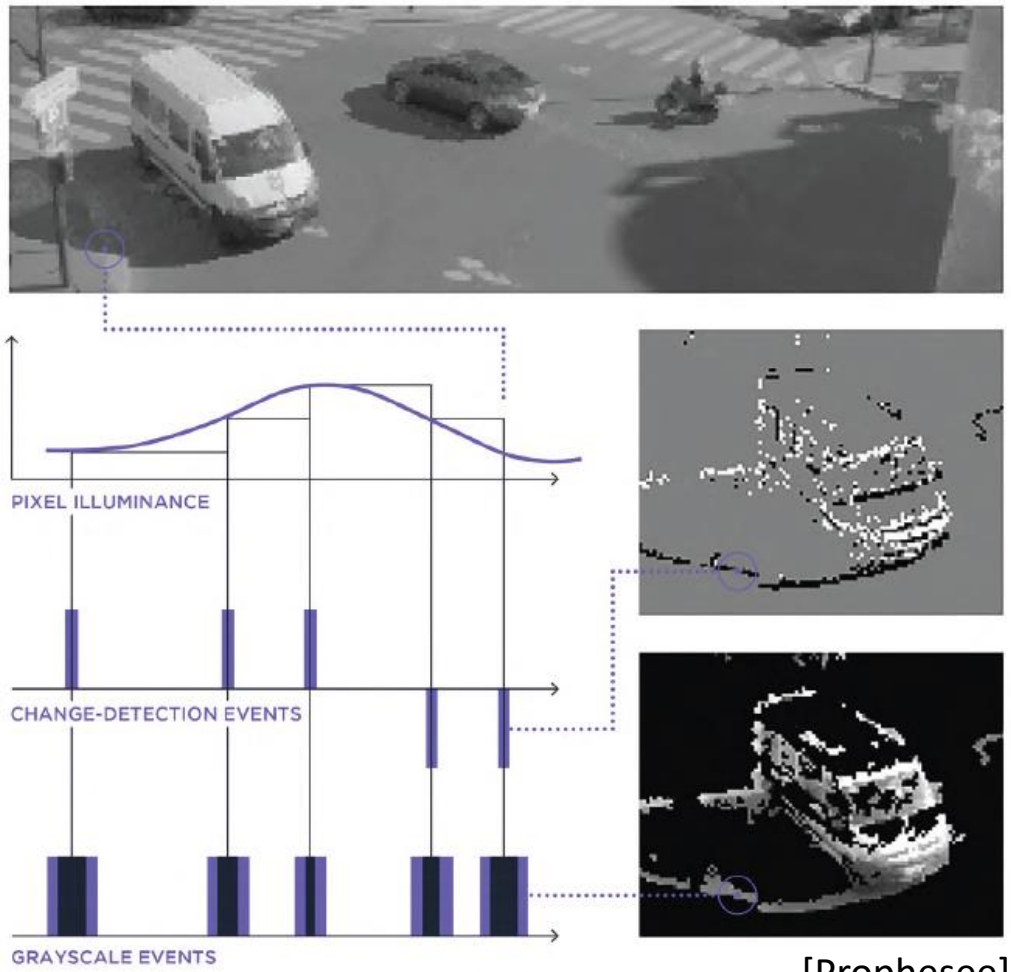


iniLabs



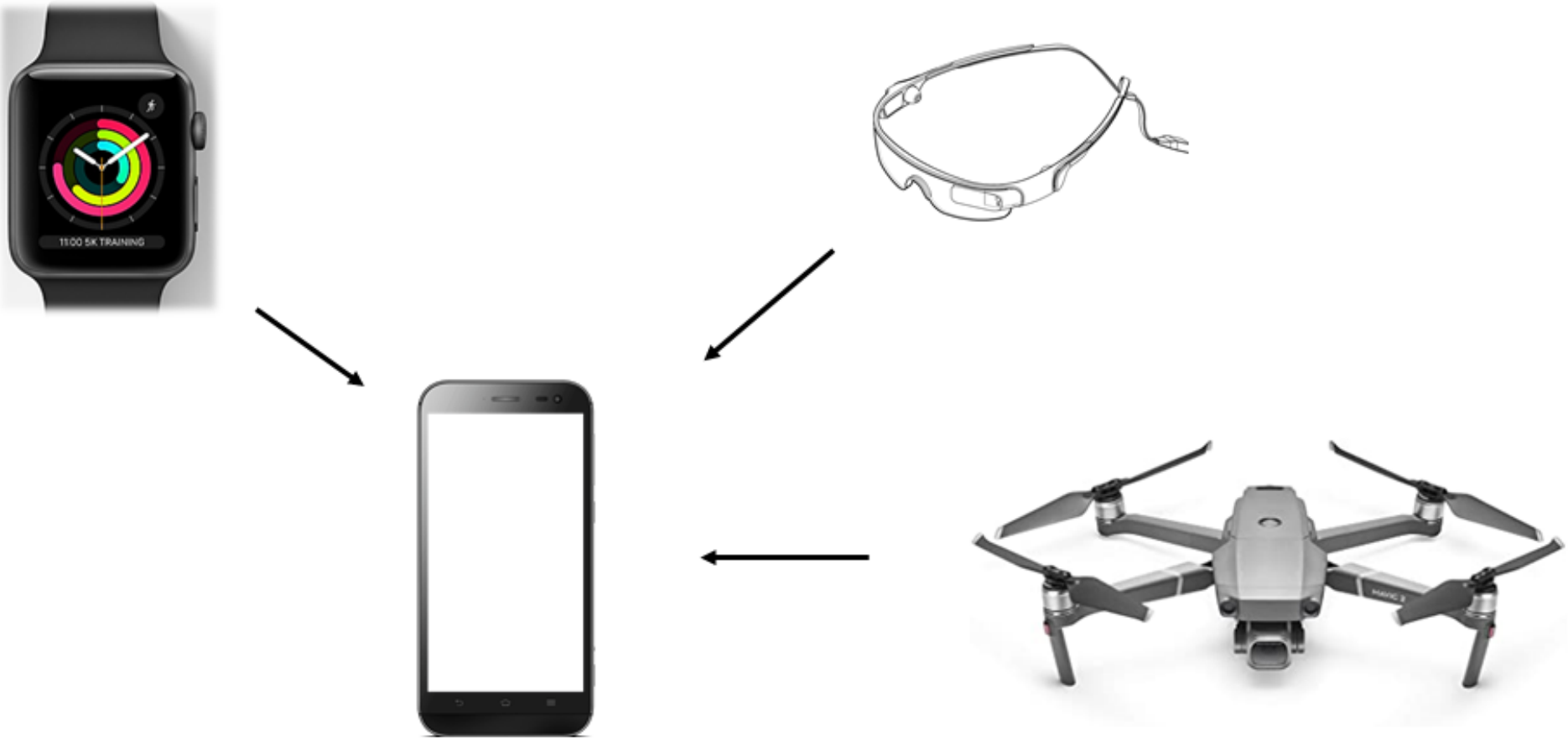
Neuromorphic Sensing

- ... and Dynamic Vision Sensor (DVS): iniVation, AiCTX, Prophesee

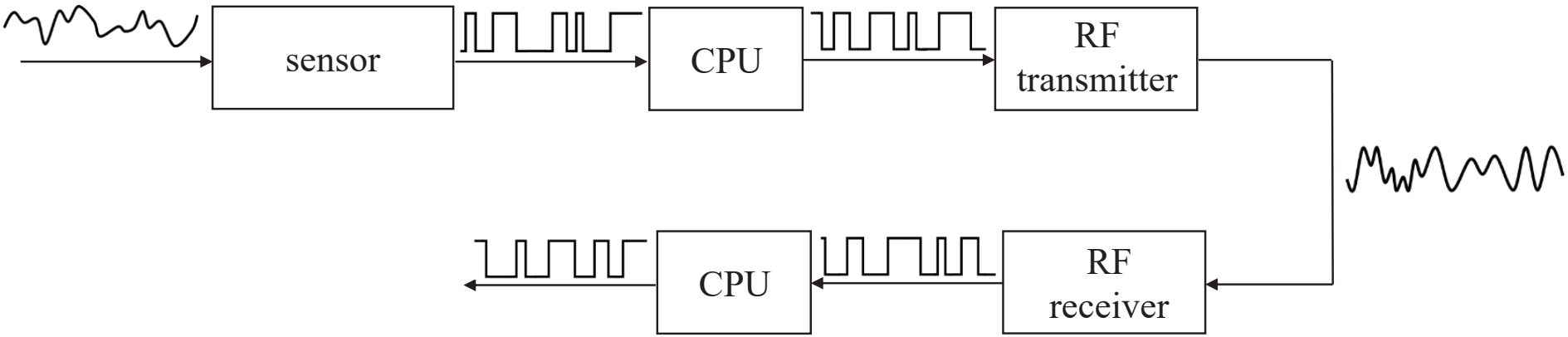


[Prophesee]

Remote Sensing and Inference

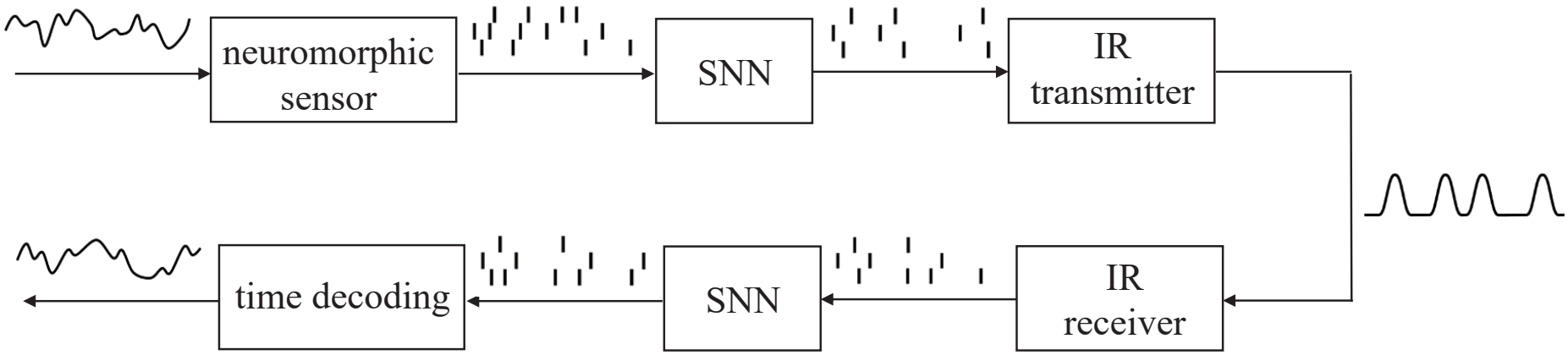


Conventional Frame-Based Digital Solution



- Digital sensing, computing, and communications:
  - High energy consumption for always-on operation, particularly when activity is sparse
  - Latency caused by frame-based transmission

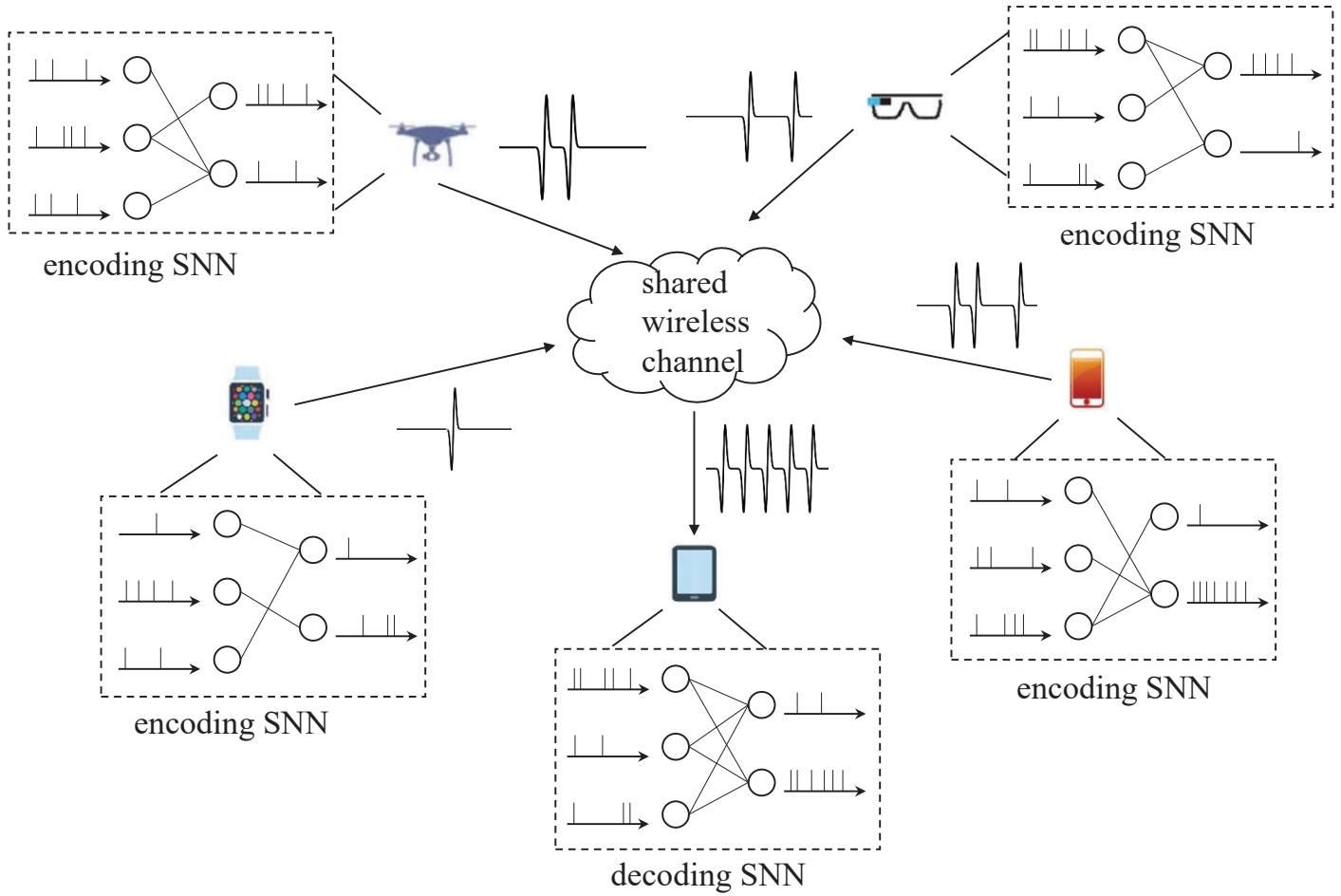
NeuroComm



- Neuromorphic sensing, computing, and communications:
  - replace digital sensing with neuromorphic sensing
  - replace digital processors with neuromorphic processors
  - replace digital communications with impulse radio

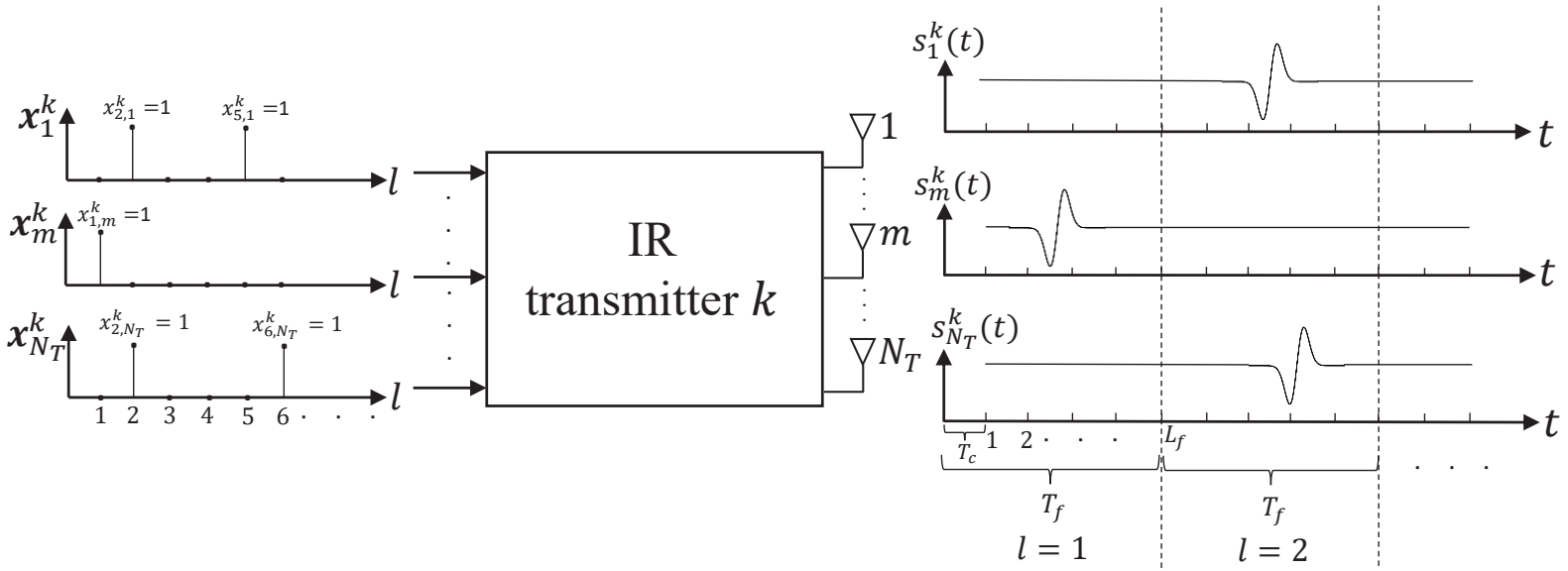
NeuroComm

- Semantics-driven energy consumption:
  - Compute and communication energy spent only when significant events (spikes) are sensed



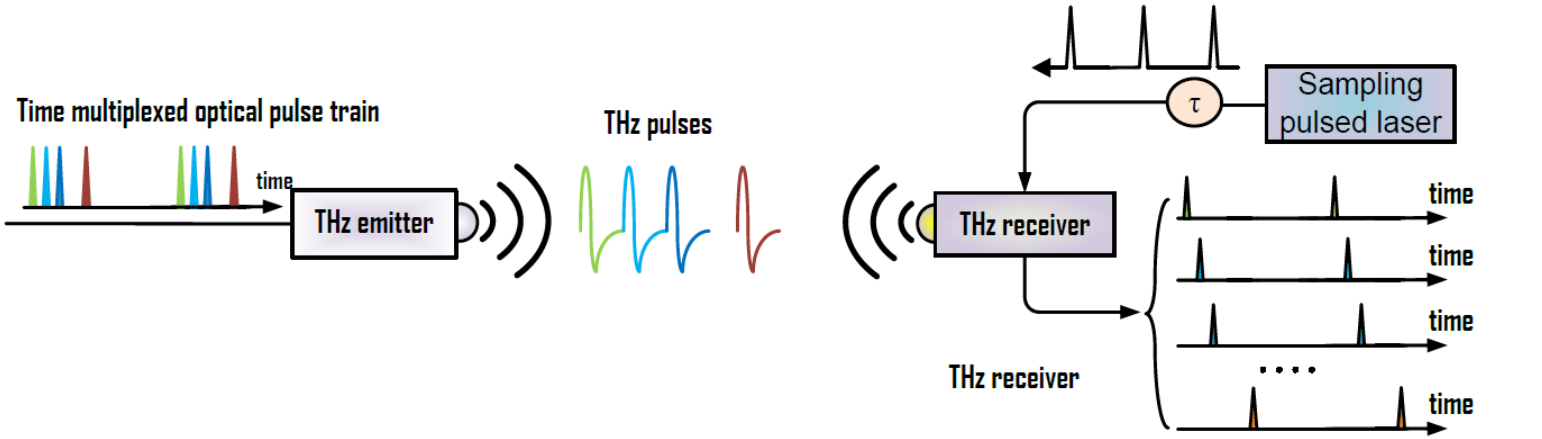
NeuroComm

- Low latency:
  - NeuroComm maps directly spikes into radio pulses.



NeuroComm and 6G

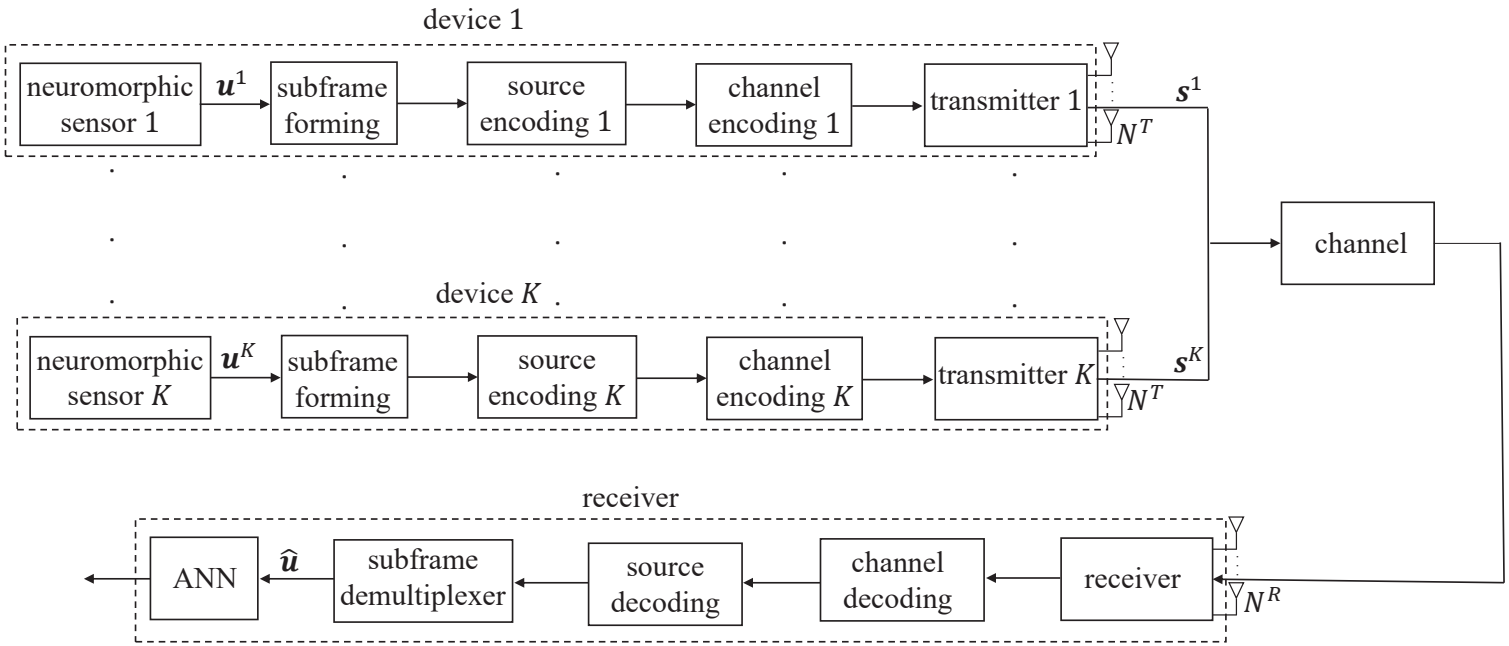
- Impulse radio is a candidate for beyond-5G systems in the Terahertz range.
- Used for extremely low-power transmission in the IEEE 802.15.4z standard.



[Yu et al '15]

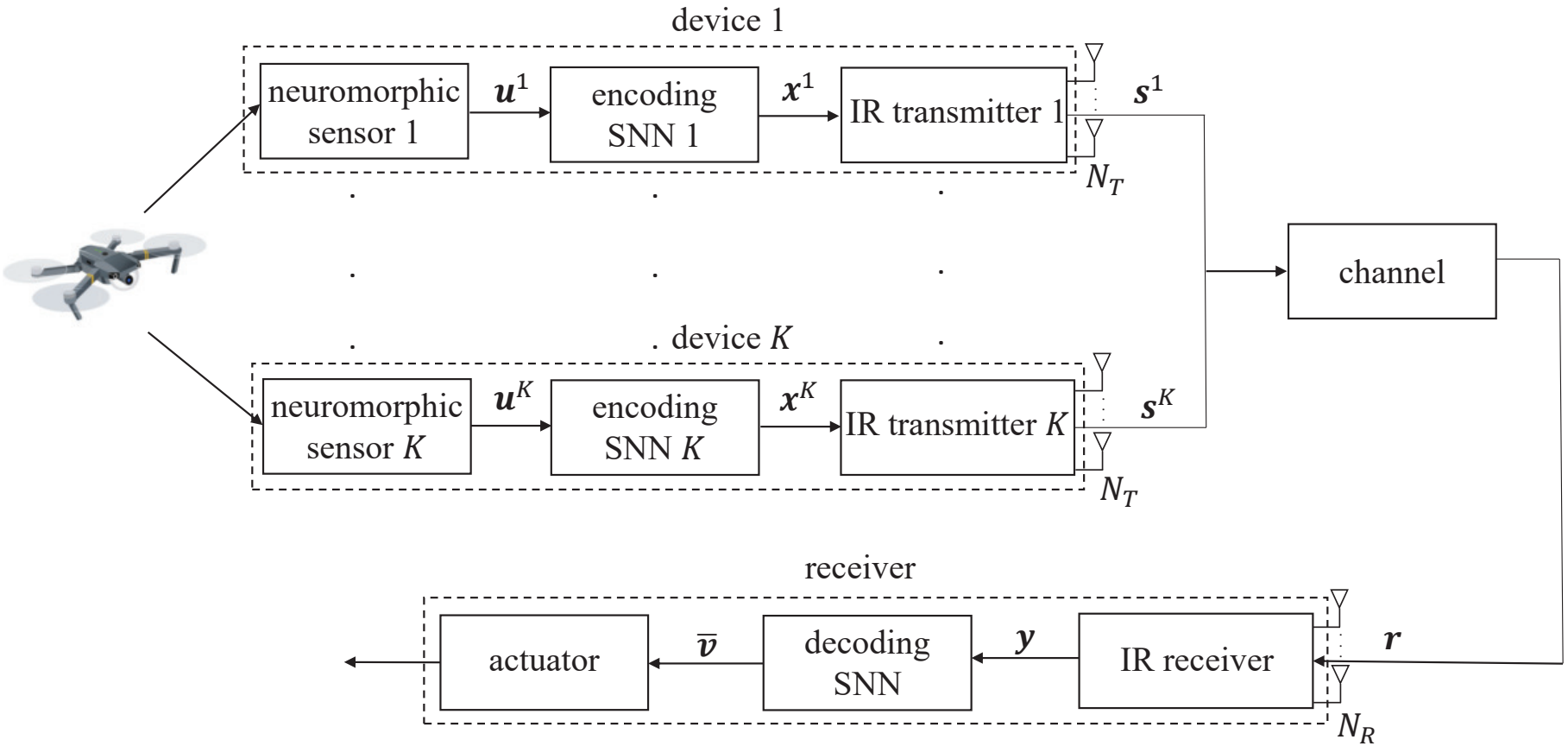
NeuroComm as Neuromorphic Joint Source-Channel Coding

- NeuroComm implements a form of joint source-channel coding that is tailored to the application.
- This contrast with the standard solution based on separate source-channel coding.



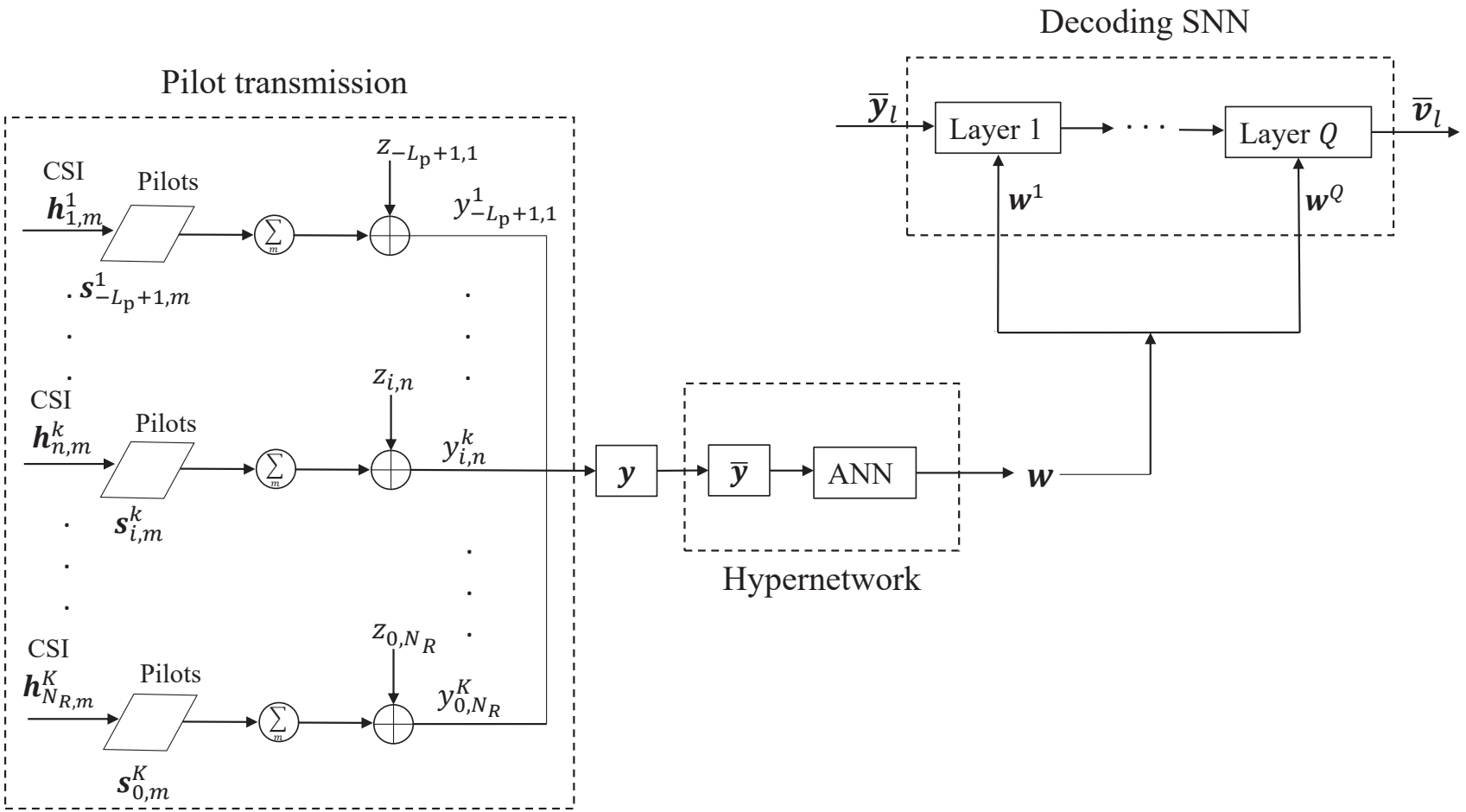
Optimizing NeuroComm

- Need to optimize operation of encoding and decoding SNNs



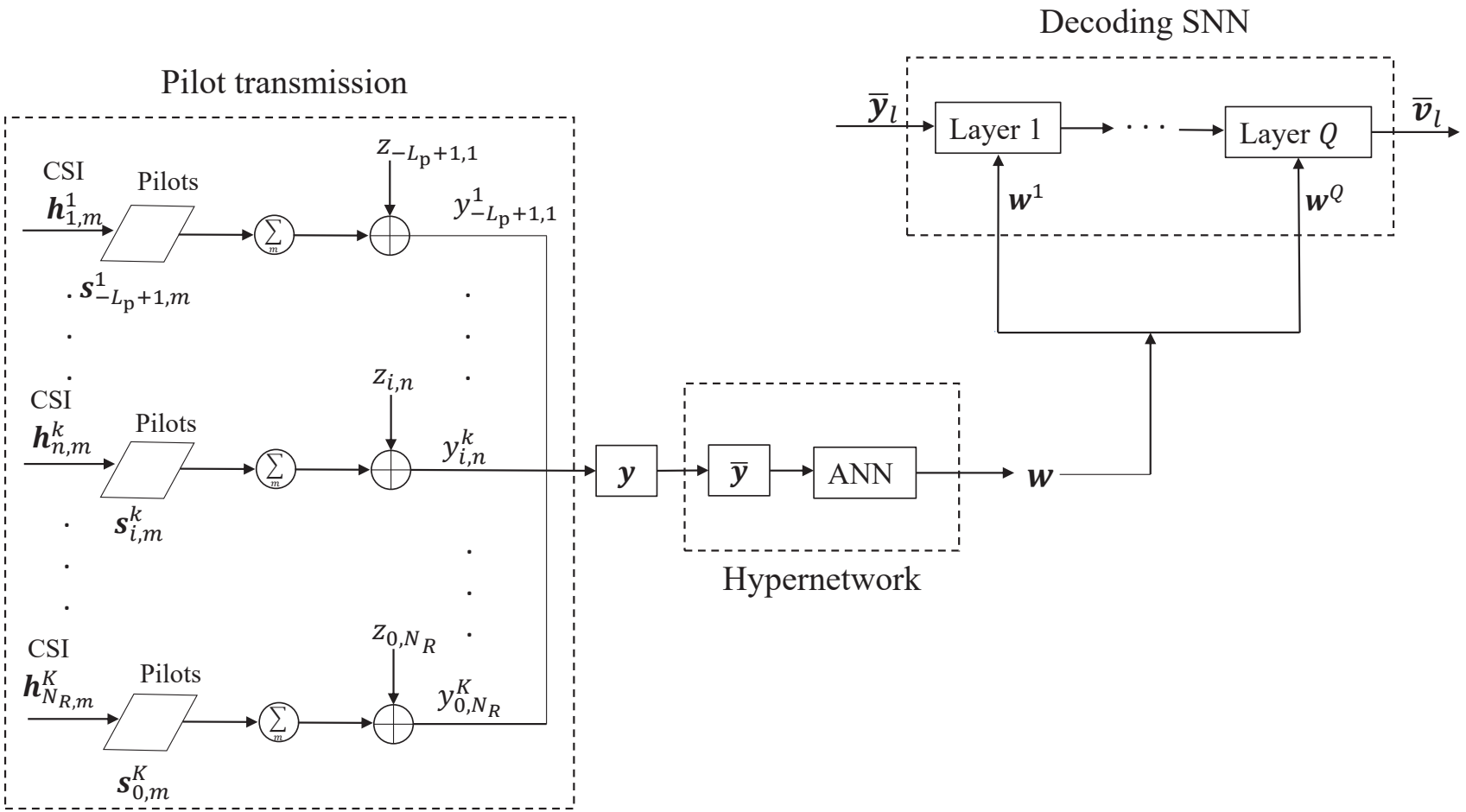
Hyper-NeuroComm

- End-to-end supervised learning using data collected from multiple channel realizations.
- While the transmitters are assumed to have no CSI, the receiver uses a hypernetwork to map pilots to the weights of the decoding SNN.



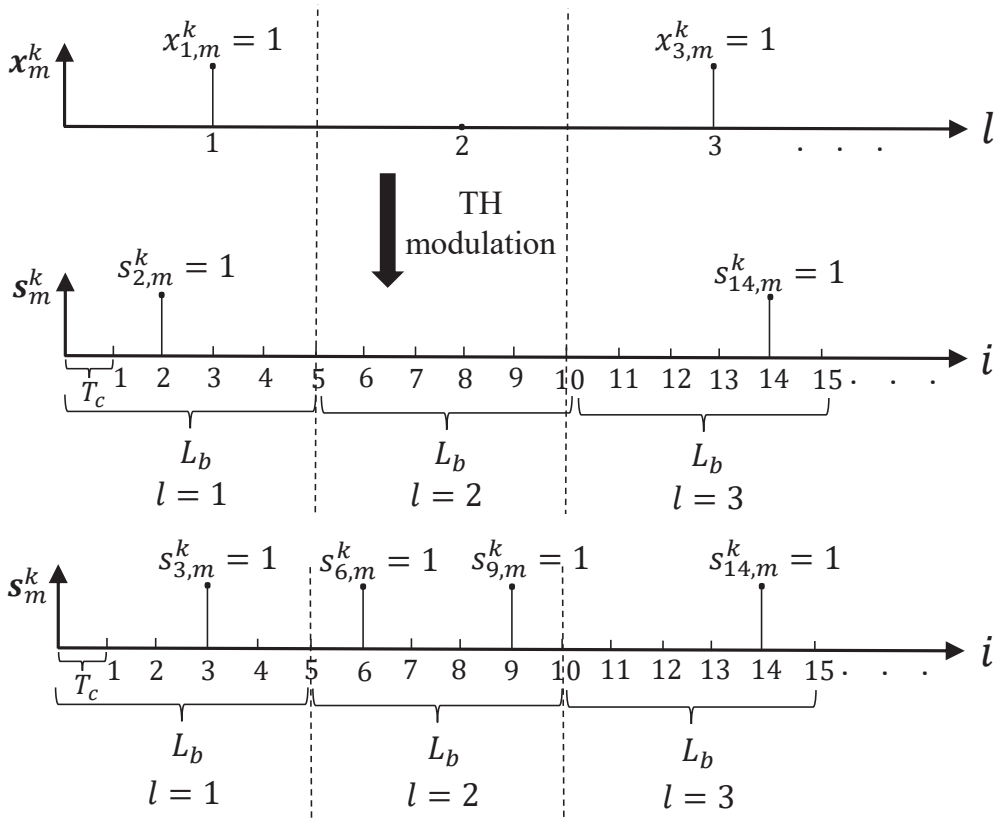
Hyper-NeuroComm

- End-to-end supervised learning using data collected from multiple channel realizations.
- While the transmitters are assumed to have no CSI, the receiver uses a hypernetwork to map pilots to the weights of the decoding SNN.



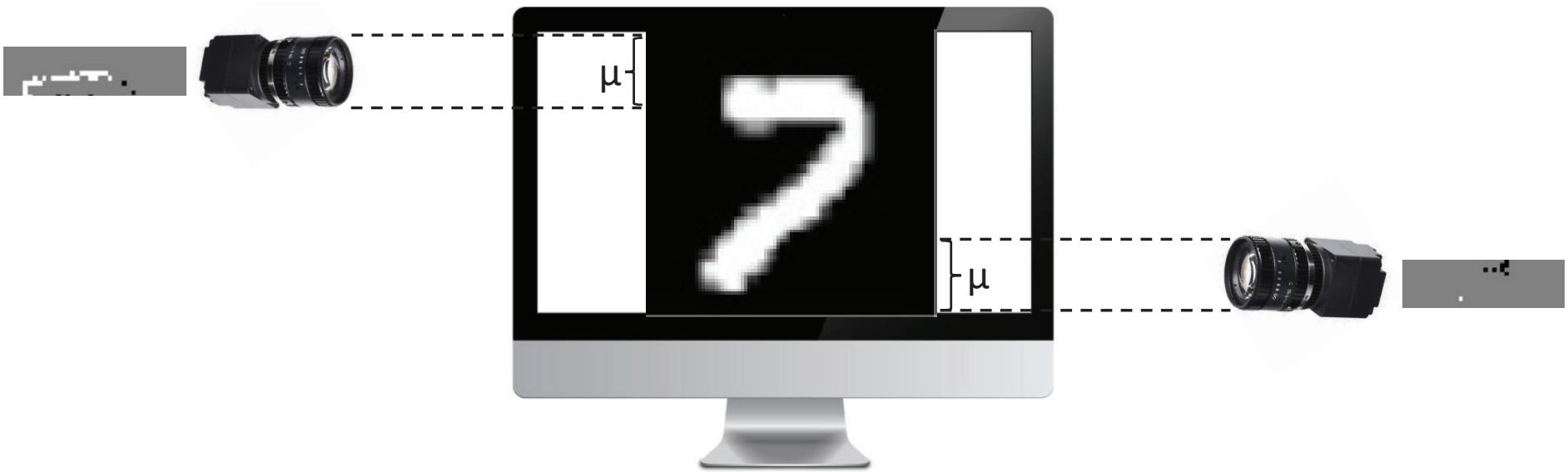
Time Hopping vs. Learned Time Hopping

- To accommodate multiple users and antennas, one can use standard time hopping (TH)...
- ... or let the encoding SNNs optimize the use of additional channel uses (Learned TH).



Experiments

- MNIST-DVS data set with each sensor observing a different fraction μ of the input.
- Communication over multipath fading channels.

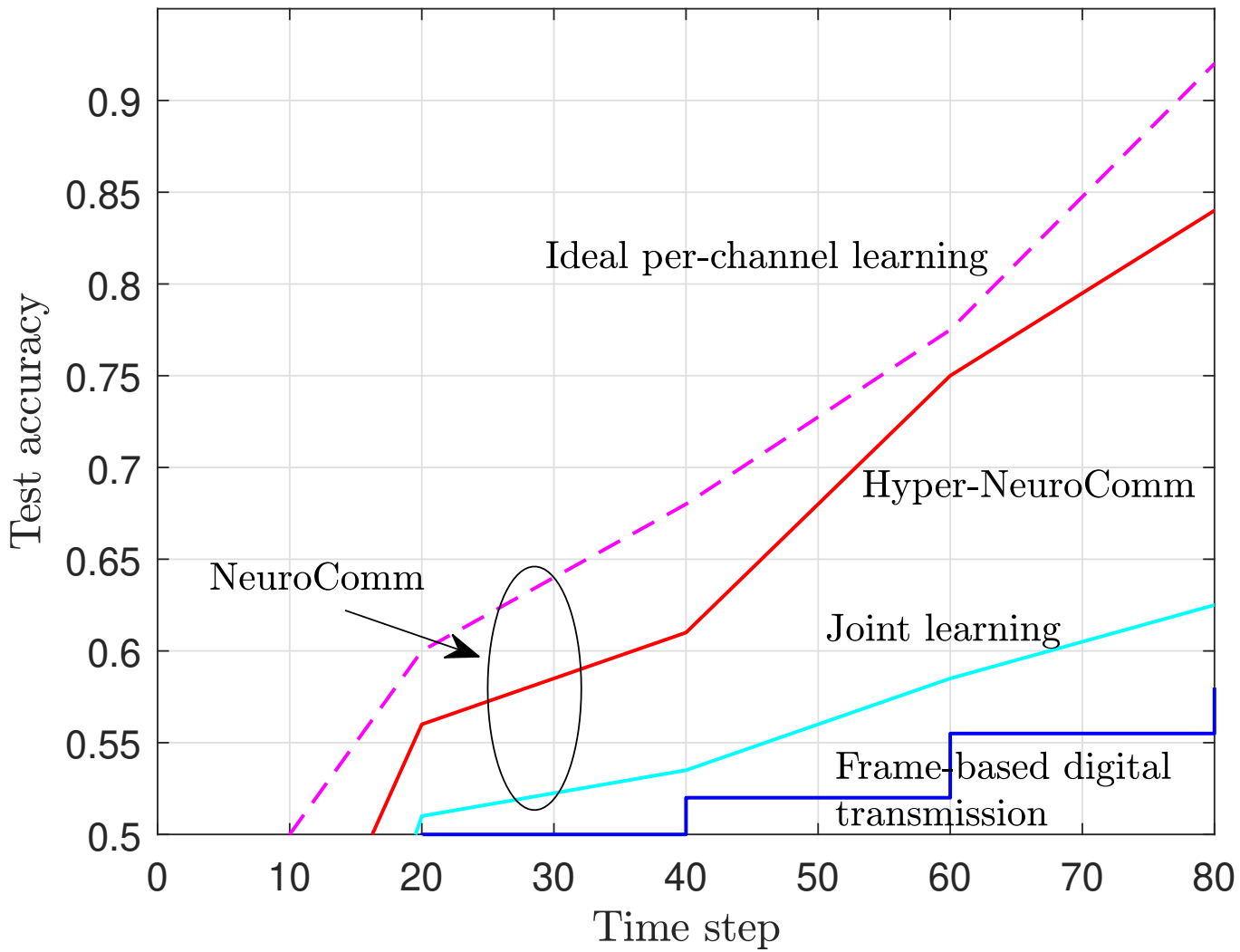


Experiments

- Benchmarks:
  - Frame-based digital transmission + ALOHA
    - ★ VQ-VAE [Van den Oord, 2017]
    - ★ LDPC decoding
  - NeuroComm with ideal per-channel learning
  - NeuroComm with joint learning (no hypernetwork)

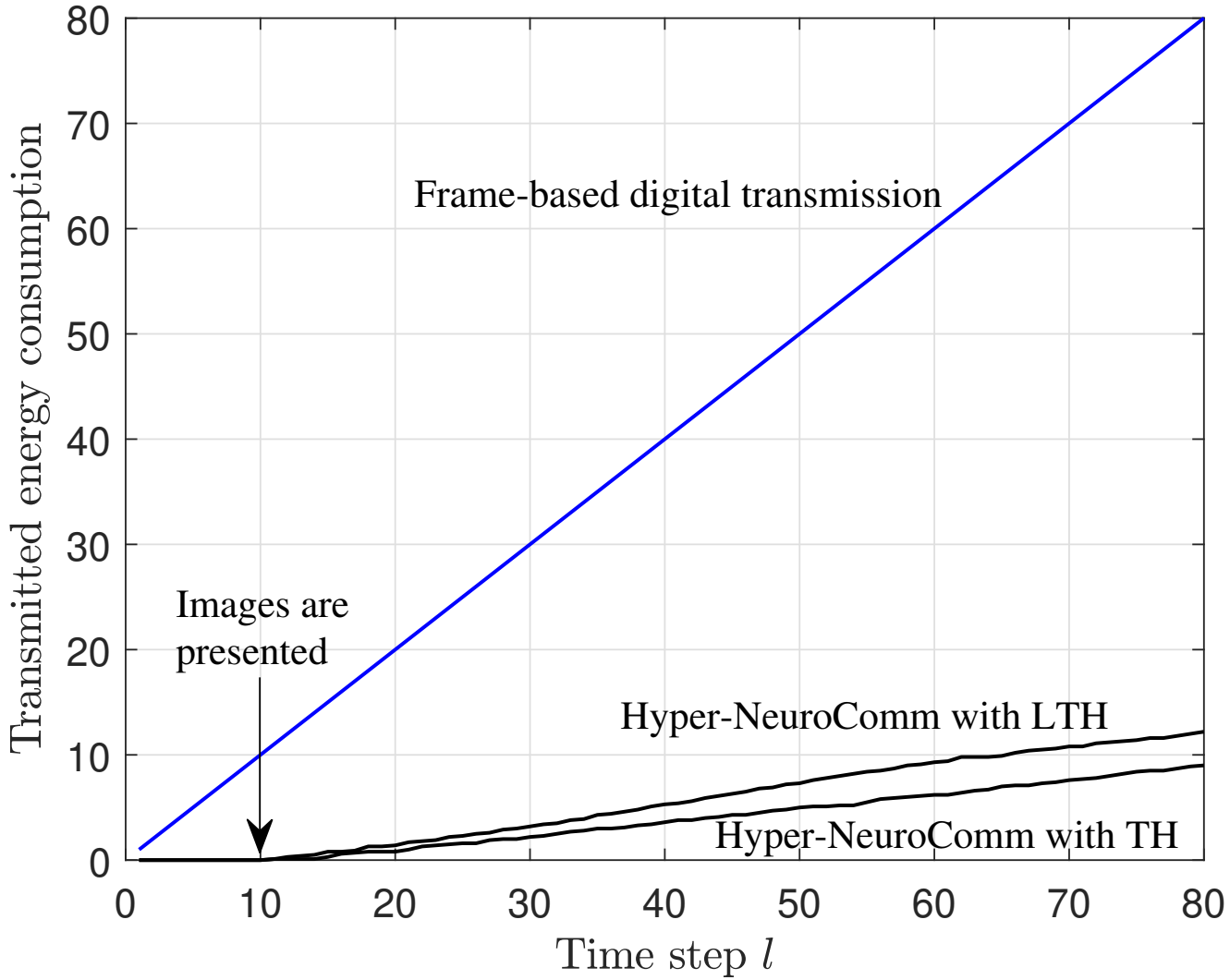
Experiments

- $K = 1$, LTH, SNR= 10 dB



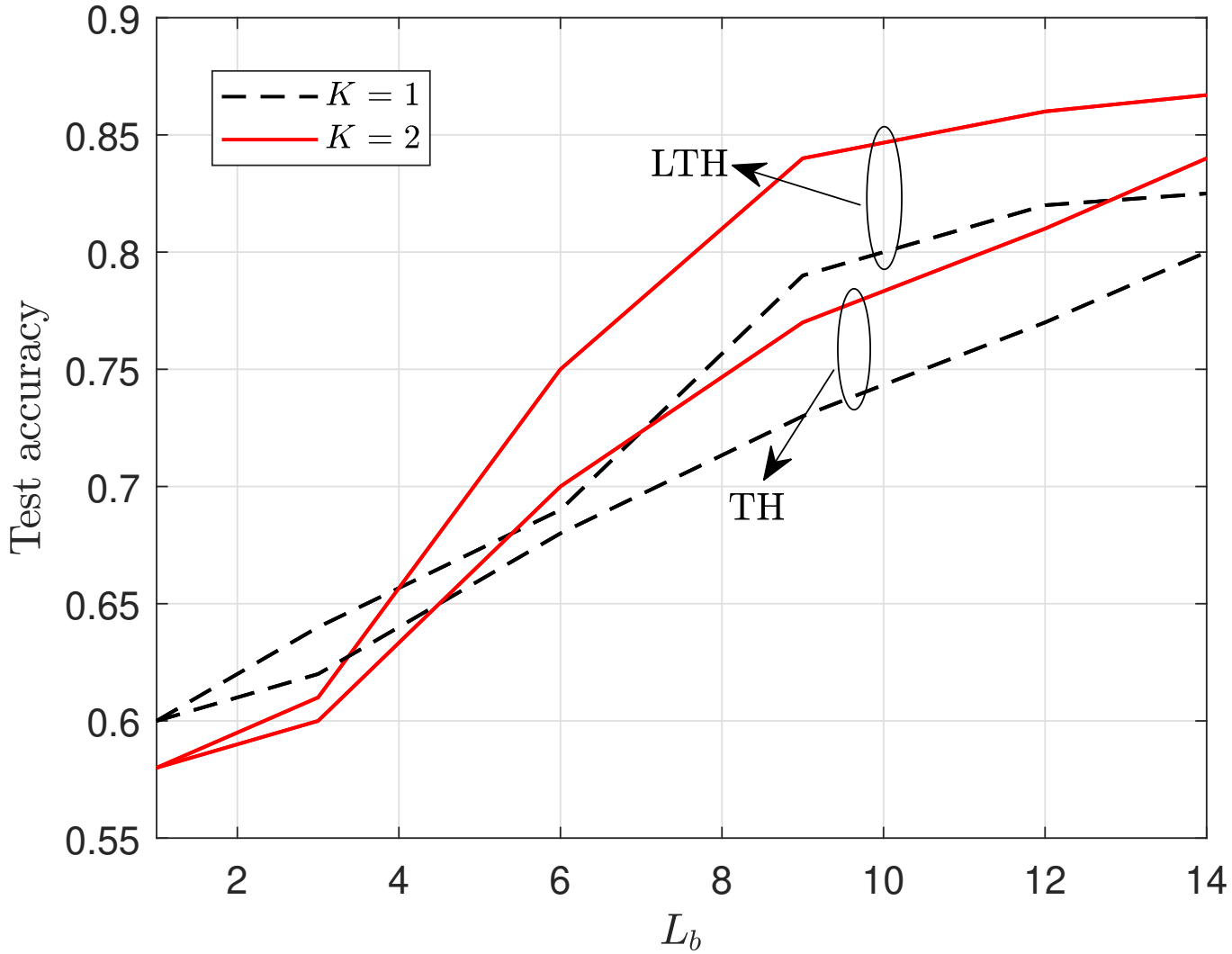
Experiments

- $K = 1$, LTH, SNR$=$ 10 dB



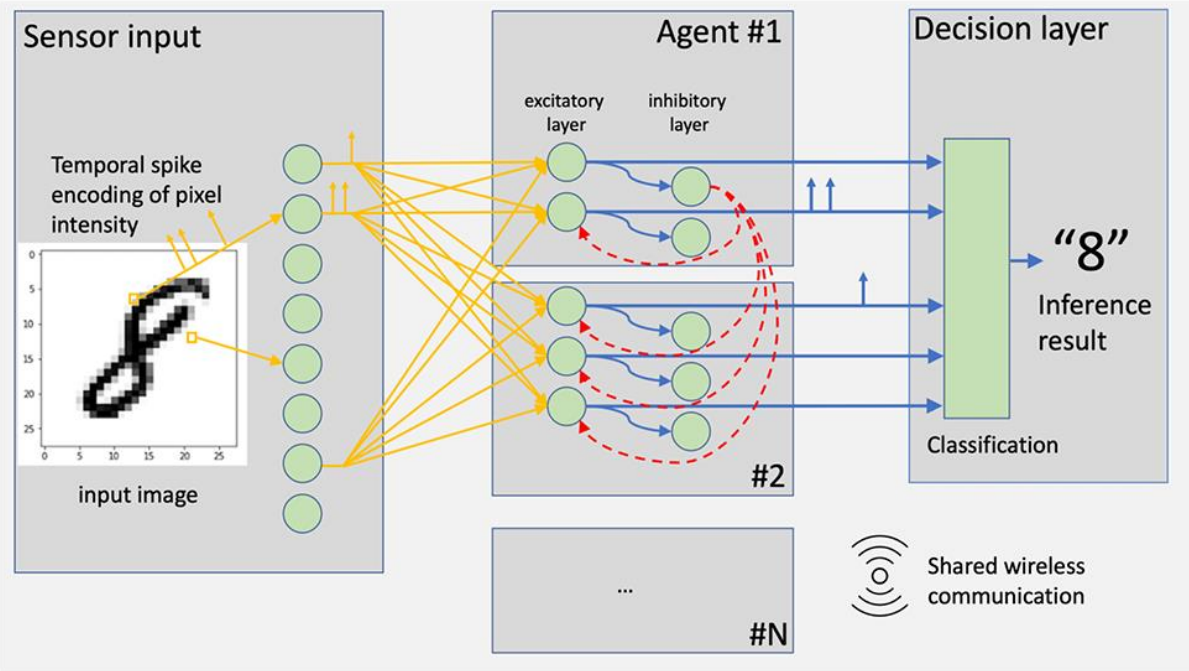
Experiments

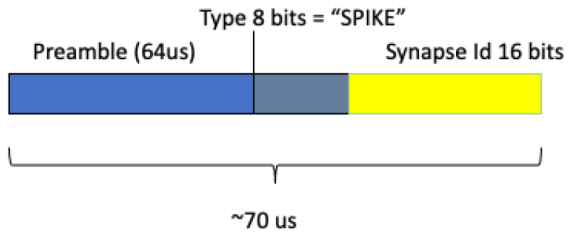
- Bandwith expansion factor L_b ($\mu = 0.75$, SNR$=$ 10 dB)



A Prototype

- An application-layer protocol for the distributed implementation of an SNN using packet-based transmission
- Impact of packet losses for channel-agnostic training





[Borsos et al, 2022]

Hardware-Efficient AI(?): Quantum Computing

Hype and Promises

The Tell

Quantum computing will be the smartphone of the 2020s, says Bank of America strategist

Published: Dec. 12, 2019 at 2:40 p.m. ET

By Chris Matthews

12

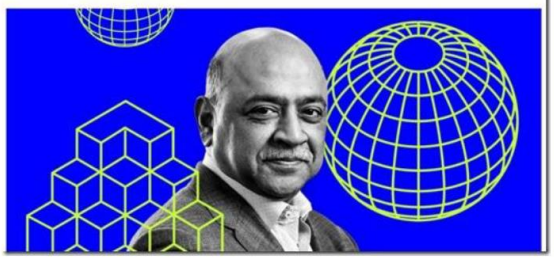
Exponentially more computing power may revolutionize health care and cybersecurity

FAST COMPANY

09-28-21 | FAST COMPANY INNOVATION FESTIVAL

IBM CEO: Quantum computing will take off 'like a rocket ship' this decade

But Arvind Krishna says that some hard quantum physics problems await as the market pushes for larger and larger quantum systems.



CLASSIQ Q.A.D PLATFORM INSIGHTS

‹ Back to Insights

PRESS RELEASES

CLASSIQ RESEARCH REVEALS THAT QUANTUM COMPUTING IS THE FUTURE – AND IT'S COMING SOON

22 September, 2021

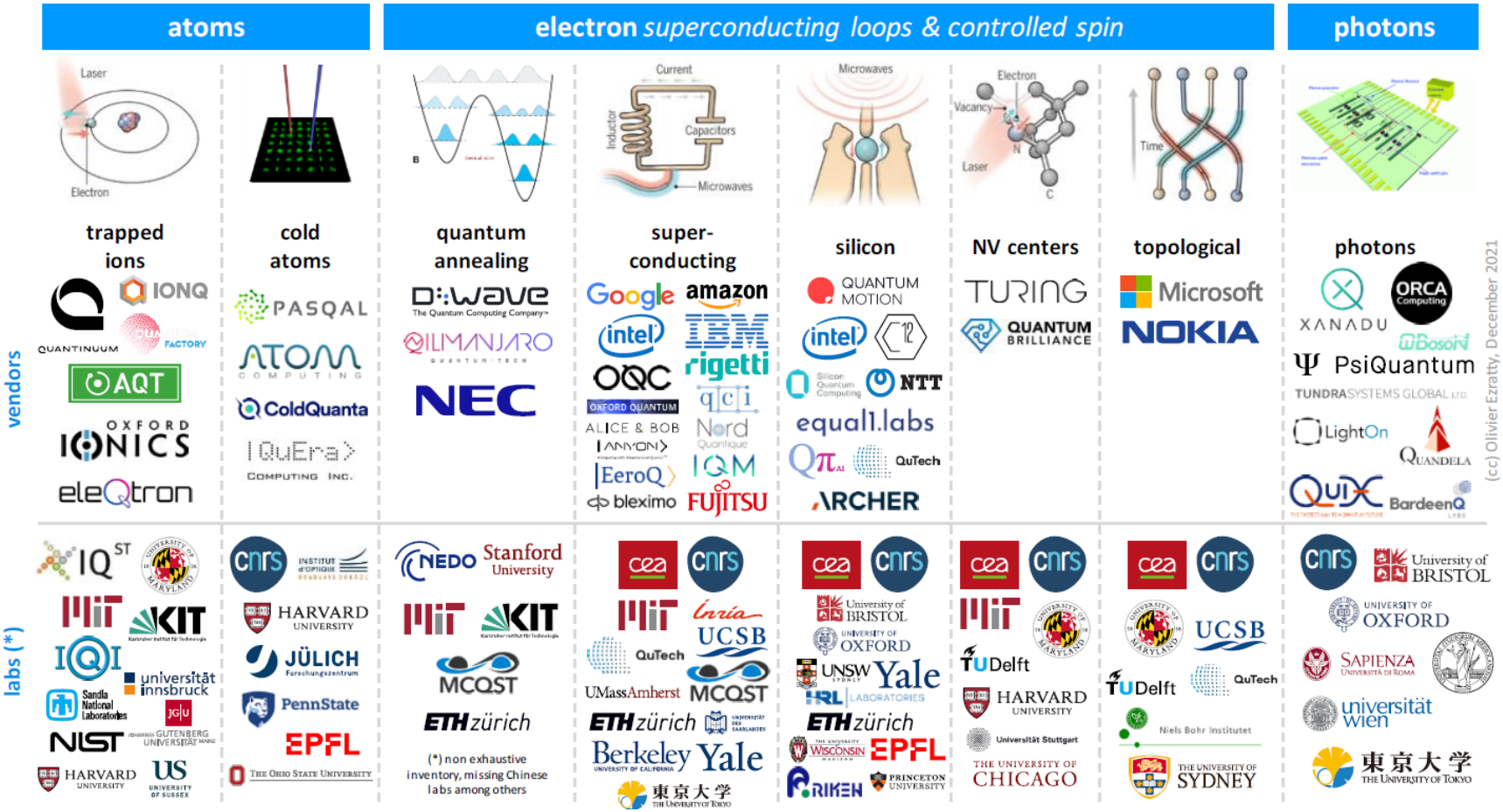
Quantum Computing Paranoia Creates a New Industry

Even though quantum computers don't exist yet, security companies are preparing to protect against them.

by Tom Simonite January 30, 2017

MIT Technology Review

Players

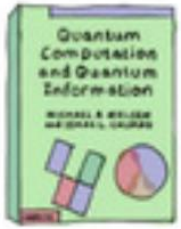


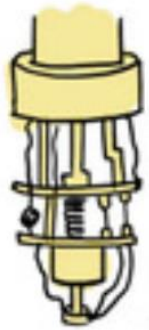
[O Ezratty '22]

Quantum Algorithms and Today's Technology

- The traditional design of quantum algorithms assumes **large and reliable** quantum computers.
- Quantum machine learning is emerging as a programming paradigm suited for current **noisy intermediate-scale quantum (NISQ) computers**.[19]

	Fault-tolerant	**Near-term**
# qubits	millions	10-1000
errors	corrected	mitigated
use	Shor, Grover, HHL...	variational circuits
research	computational complexity	run it and see
available	in 5-30 years?	now

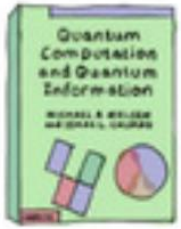




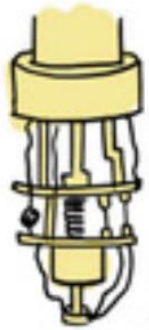
[M. Schuld '21]

Quantum Algorithms and Today's Technology

- The traditional design of quantum algorithms assumes **large and reliable** quantum computers.
- Quantum machine learning is emerging as a programming paradigm suited for current **noisy intermediate-scale quantum (NISQ) computers**.[19]



	Fault-tolerant	Near-term
# qubits	millions	10-1000
errors	corrected	mitigated
use	Shor, Grover, HHL...	variational circuits
research	computational complexity	run it and see
available	in 5-30 years?	now



[M. Schuld '21]

Quantum Circuit

- A **quantum algorithm** is specified by a quantum circuit operating on a set of n qubits.
- A **quantum circuit** consists of a sequence of **quantum gates** that are applied sequentially and **in place** to the n qubits...

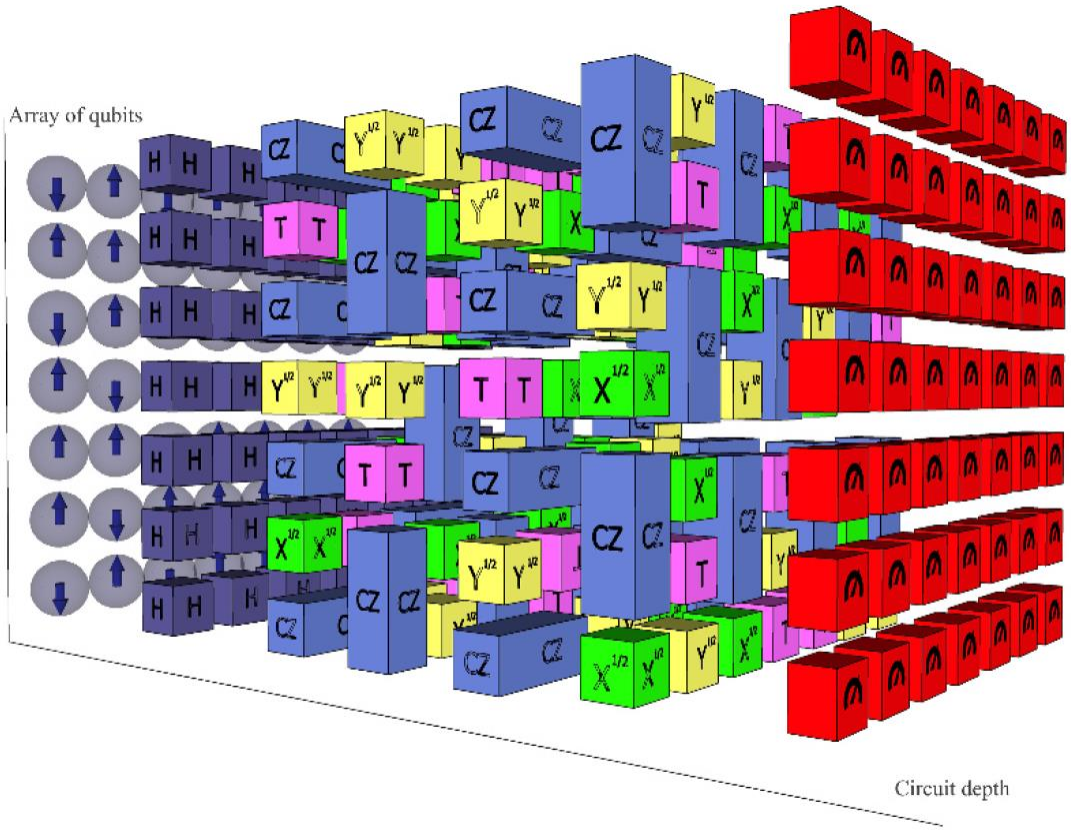


Figure 3.4: 3-D Quantum circuit diagram Source: Google [Hidary '19]

Quantum Circuit

- ... followed by **measurements** that convert the state of the n qubits into n classical bits.

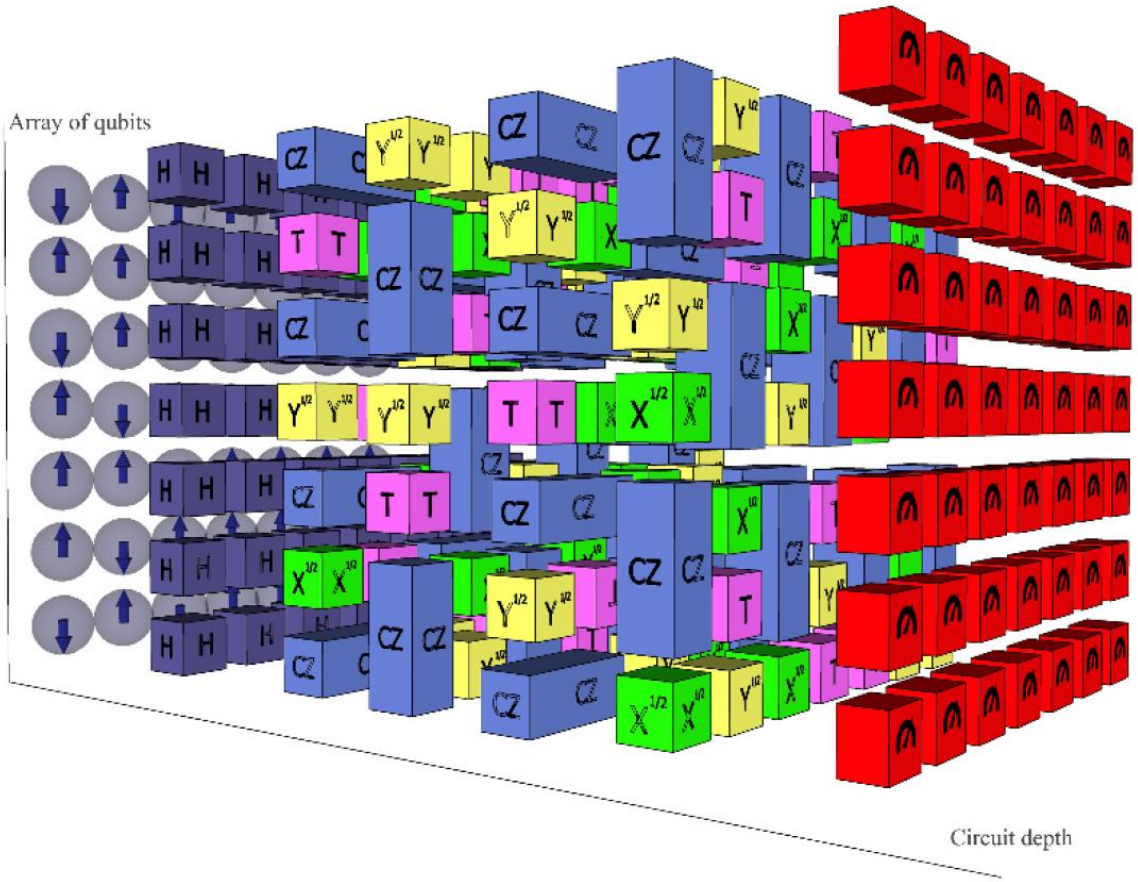
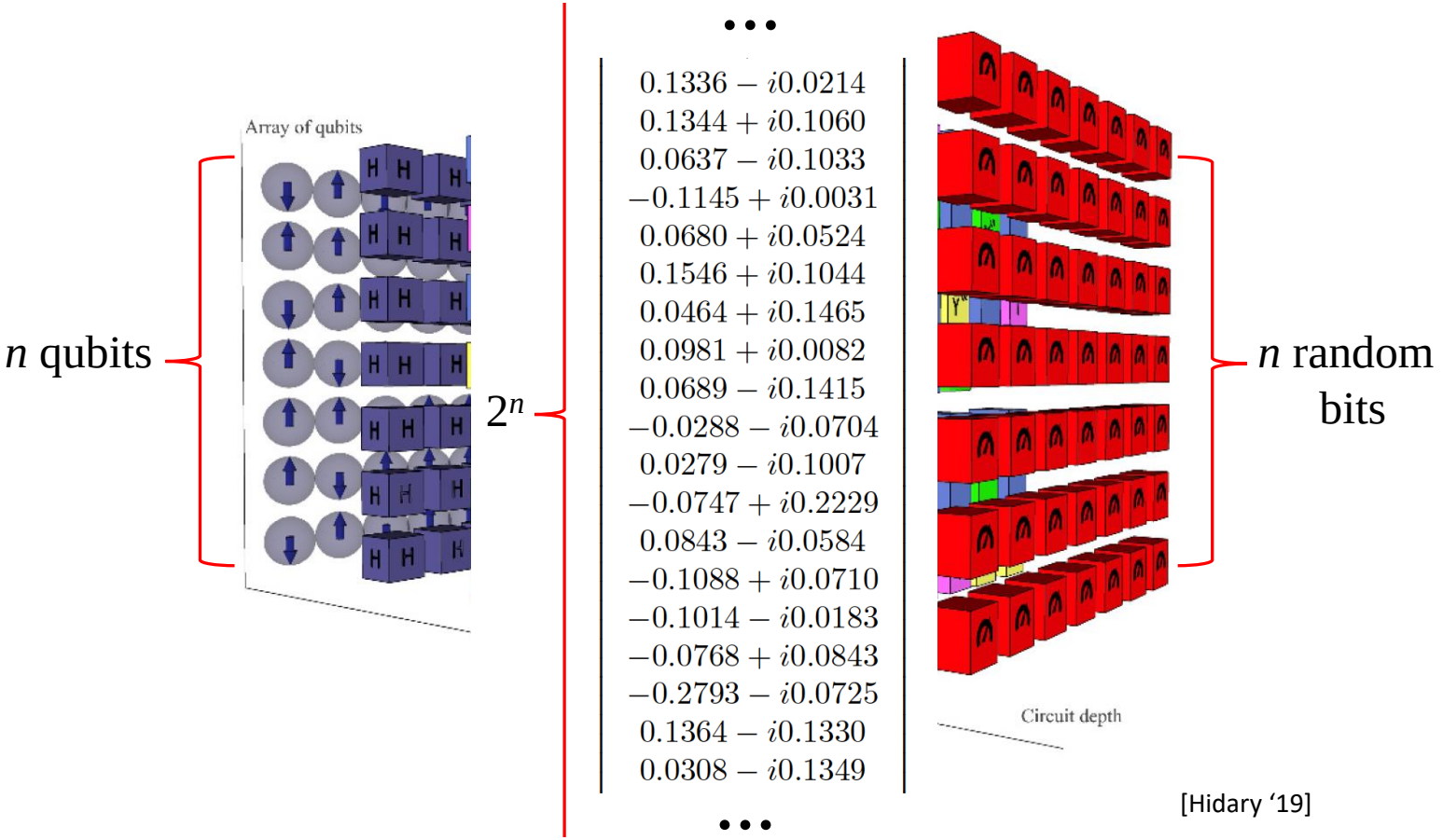


Figure 3.4: 3-D Quantum circuit diagram Source: Google [Hidary '19]

Quantum Circuit

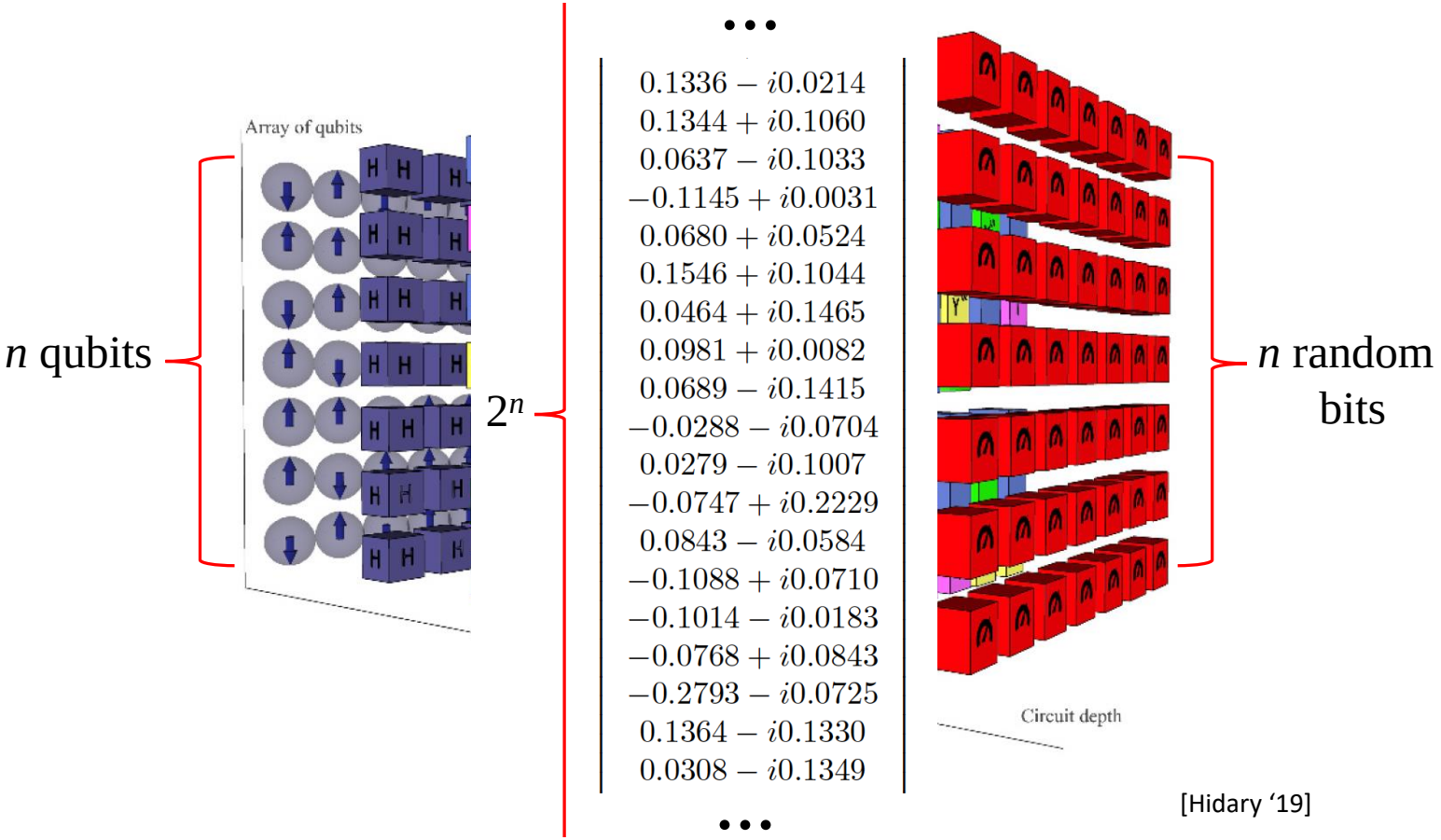
- The state of n qubits is described by a 2^n-**dimensional complex (amplitude) vector**.
- Quantum measurements are inherently **random**: **"collapse"** of the waveform.



[Hidary '19]

Quantum Circuit

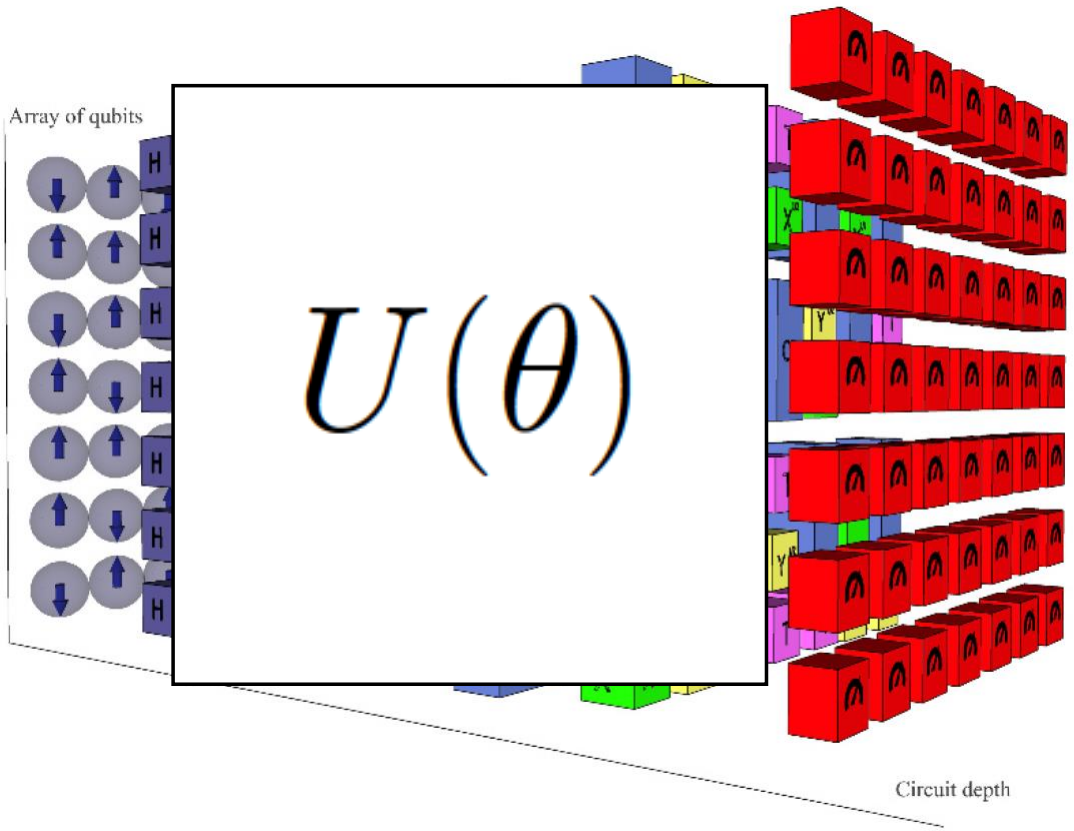
- The state of n qubits is described by a 2^n-**dimensional complex (amplitude) vector**.
- Quantum measurements are inherently **random**: **"collapse"** of the waveform.



[Hidary '19]

Parameterized Quantum Circuit

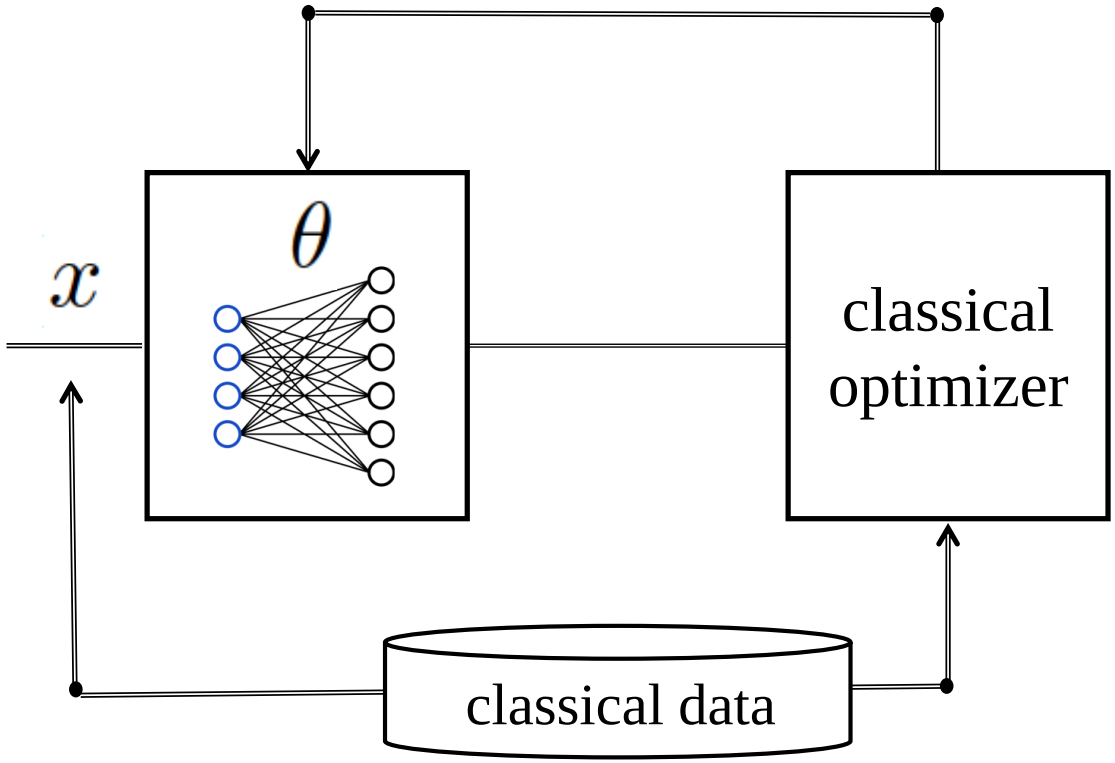
- A **parameterized quantum circuit** (PQC) is defined by a fixed sequence of quantum gates whose operation depends on a vector of **classical parameters** θ.
- PQCs are also known as **quantum neural networks**.



[Hidary '19]

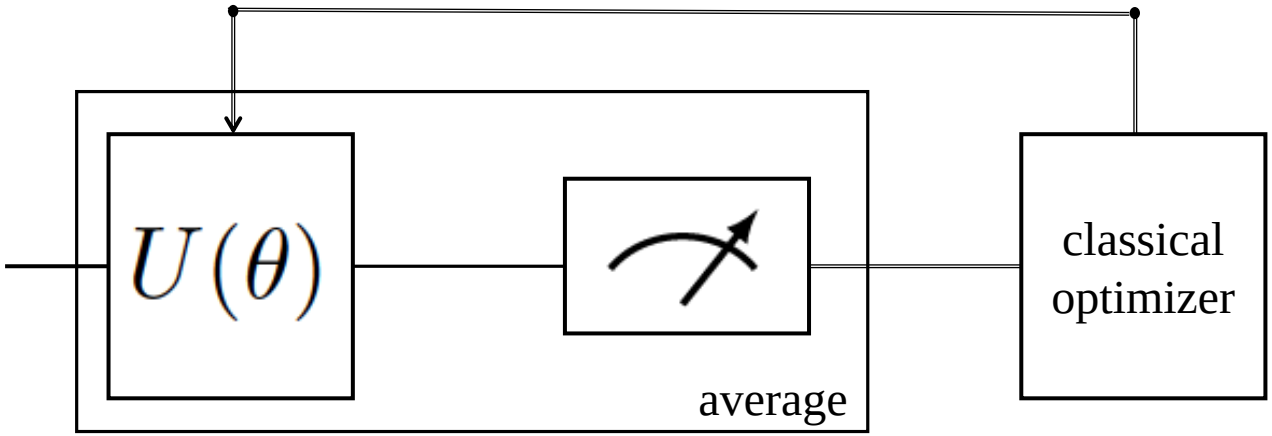
Classical Machine Learning

- Classical machine learning relies on **parameterized functions** $f(x|\theta)$, e.g., neural networks.
- The parameters θ are optimized by comparing the model output with classical data.



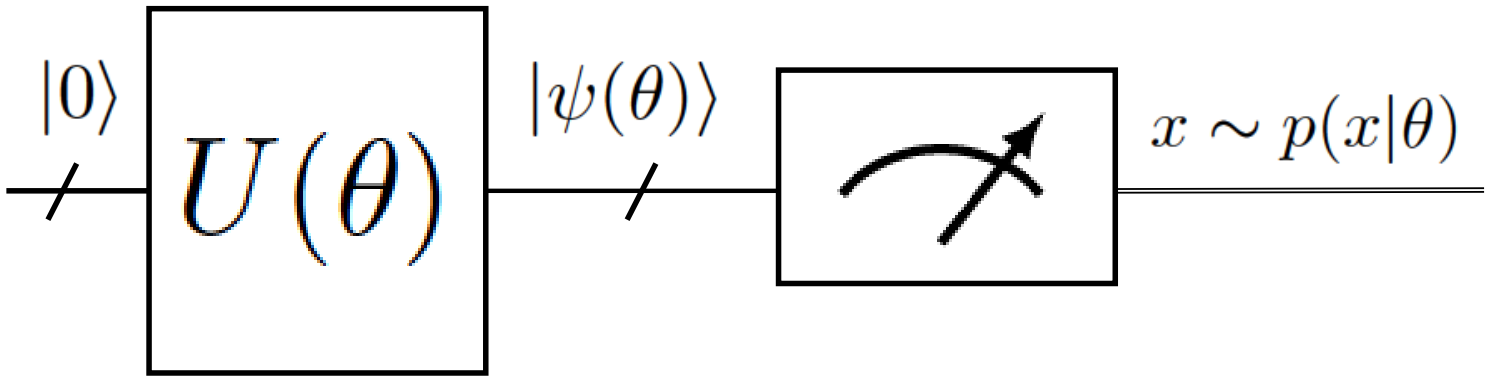
Quantum Machine Learning

- In **quantum machine learning**, the parameters of the PQC $U(\theta)$ are designed using classical optimization based on measurements of the output of the PQC and (possibly) data.
- By keeping the quantum computer in the loop, the classical optimizer can account for the **non-idealities and limitations** of quantum operations.



Unsupervised Generative Learning

- The measurement of the output of a PQC on n qubits produces a **random n-bit string** $x \sim p(x|\theta)$ by Born's rule.
- **Born machine**: Generative model for binary strings x implemented via a PQC[20]



20 B. Coyle B, et al, "The Born supremacy: Quantum advantage and training of an Ising Born machine," Quantum Information, 2020.

Unsupervised Generative Learning

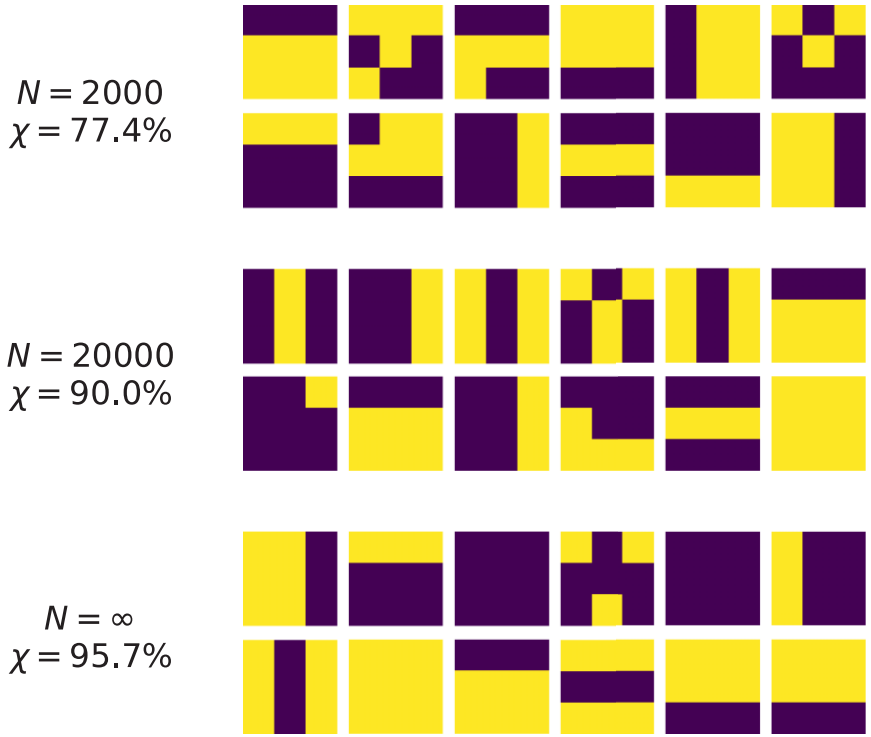


FIG. 4. 3×3 BARS-AND-STRIPES samples generated from the QCBMs. The circuit parameters used here are from the final stages of Adam training with different batch sizes N in Fig. 3(a). χ is the rate of generating valid samples in the training dataset. For illustrative purposes, we only show 12 samples for each situation with batch size N.

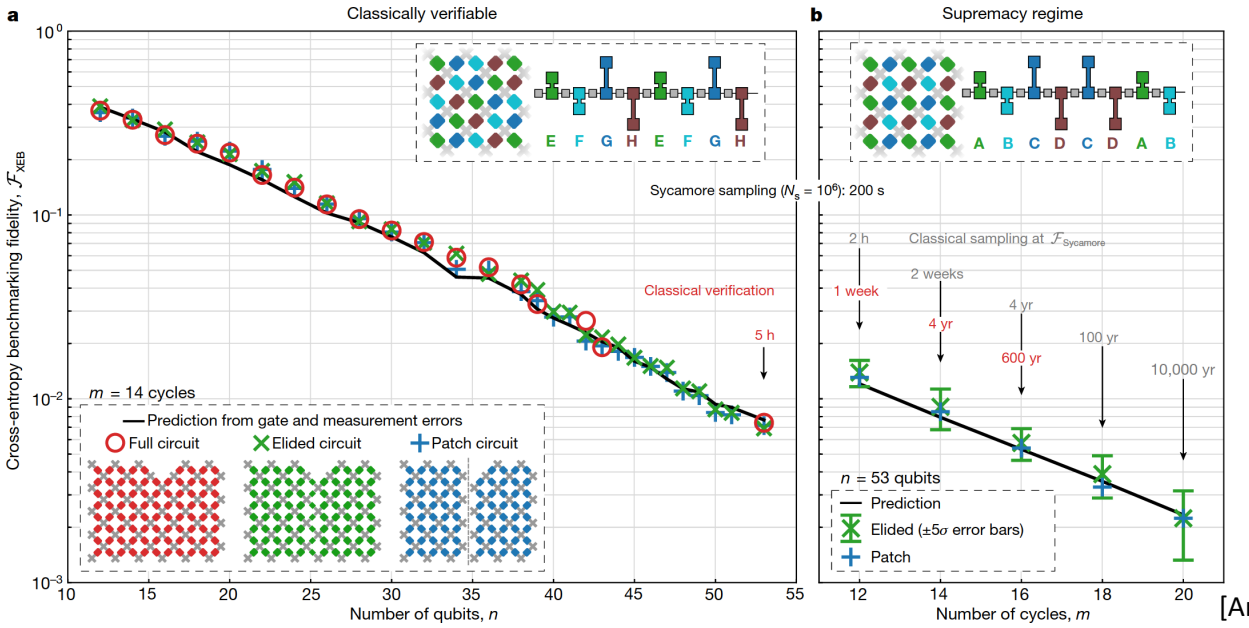
[Liu and Wang '18]

Quantum Circuits as Samplers

- Current claims of **quantum supremacy/ advantage** rest on the capacity of quantum circuits to generate samples from **joint discrete distributions** in a more efficient manner than classical devices[21].

Article

Quantum supremacy using a programmable superconducting processor



[Arute et al '19]

21 X. Gao et al, "Enhancing generative models via quantum correlations," arXiv 2021.

Quantum Machine Learning

- Quantum machine learning can efficiently implement probabilistic models[22].
- Applications to probabilistic inference[23], Bayesian learning[24], meta-learning,...

22 R. Sweke, et al, "On the quantum versus classical learnability of discrete distributions," Quantum, 2021.

23 I. Nikoloska, and O. Simeone, "Training Hybrid Classical-Quantum Classifiers via Stochastic Variational Optimization", arXiv:2201.08629, 2022.

24 M. Benedetti, et al, "Variational inference with a quantum computer," Physical Review Applied, 2021.

Challenges

- **Architecture:**
  - What are the "right" building blocks for quantum machine learning models?
  - How to scale up classical input and/or output data?
  - How to integrate classical and quantum machine learning models?
- **Optimization:**
  - How to improve the performance of gradient descent in the presence of barren plateaus?
  - How to account for "quantum noise"?
- **Theory:**
  - What are the data requirements for quantum machine learning, particularly for generative modeling?

For More...

- O. Simeone, An Introduction to Quantum Machine Learning for Engineers, https://arxiv.org/abs/2205.09510

Conclusions

Conclusions

- Importance of reliability, efficiency, and robustness in AI for engineering
- Reliable AI via Bayesian learning
- Robust and reliable AI via robust Bayesian learning
- Efficient AI via meta-learning
- Reliable and efficient AI via Bayesian meta-learning
- Hardware-efficient AI via neuromorphic and quantum (?) computing
- Directions for research:
  - Formal reliability guarantees
  - Formal robustness guarantees, including under more general conditions
  - Integration with model-driven design
  - Applications to open networking
  - Use cases for quantum and neuromorphic computing in communications

Acknowledgements

This work has been supported by Intel via the Intel Neuromorphic Research Community (INRC) and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 725731)