Online Learning for Hierarchical Inference

Sharayu Moharir
IIT Bombay

Joint work with Ghina Al-Atat, Puranjay Datta, and Jaya Prakash Champati

Online Learning for HI

Online Learning for HI

Motivation

Our Setting

Main Results

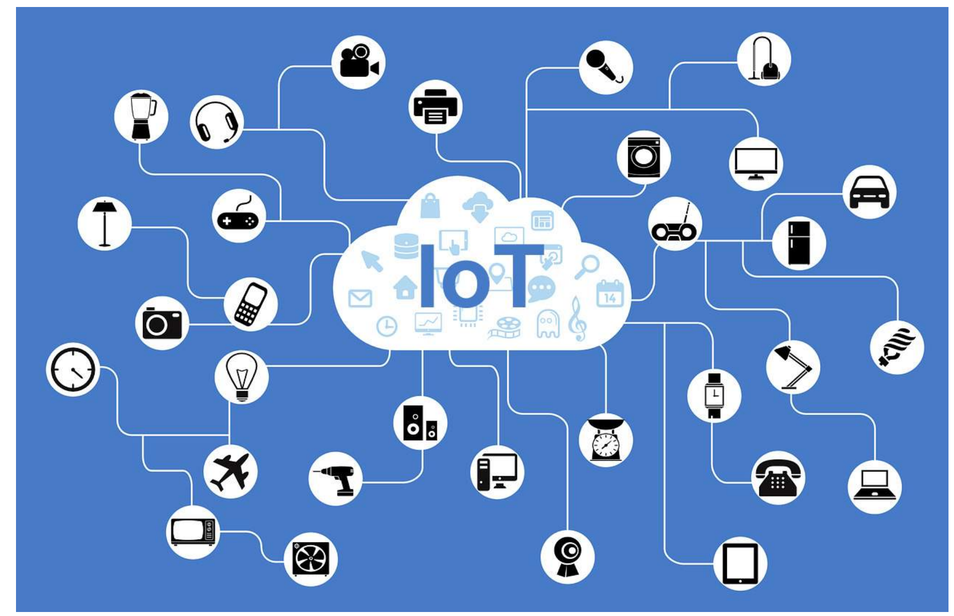
Background: Prediction with Experts

Our Algorithms and Guarantees

Numerical Results

Conclusions

Inference in CPSs



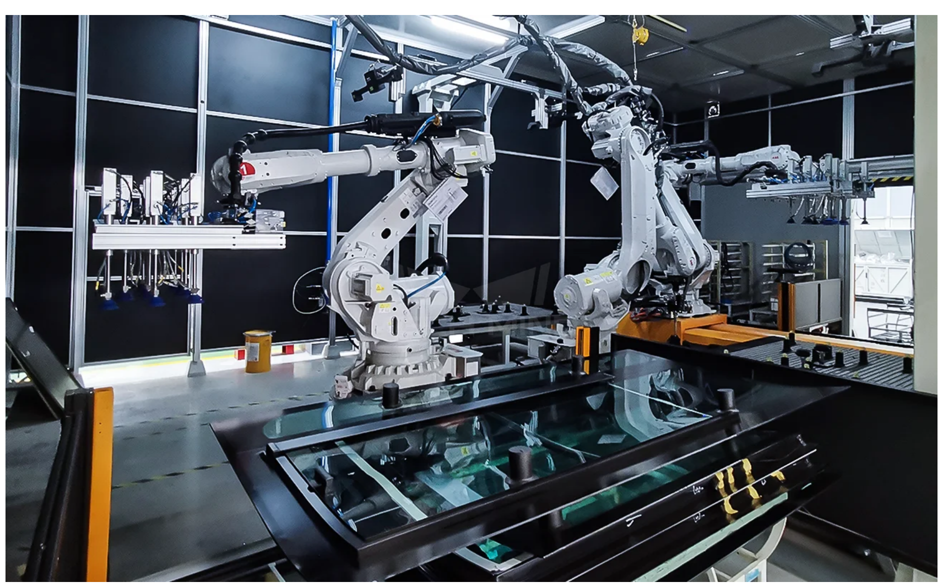
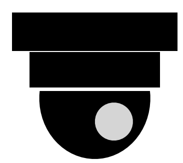


Image: https://www.1home.io/blog/what-is-a-smart-home/
Image: https://www.mech-mind.com/blog/definition-benefits-of-factory-automation-system.html

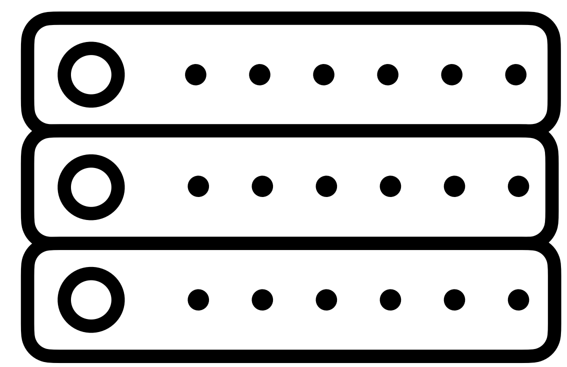
Image: https://tecadmin.net/what-is-iot-internet-of-things/
Image: https://www.medicalbuyer.co.in/global-remote-healthcare-market-to-reach-usd-59-7b/

System Components



End-Device

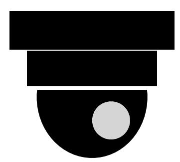
- Limited memory
- Limited compute power
- Periodically collects samples
- Needs inference on each sample to make control decisions



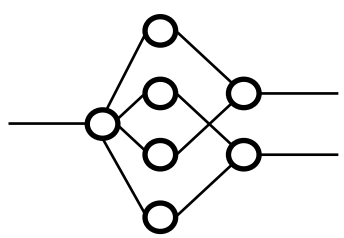
Edge-Server

- Large memory
- High compute power
- Not co-located with ED
- Can communicate with ED via a wireless channel

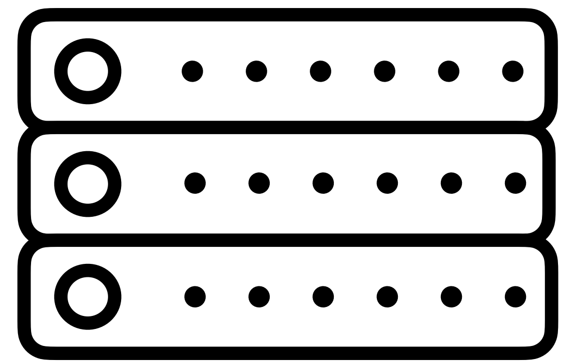
System Components



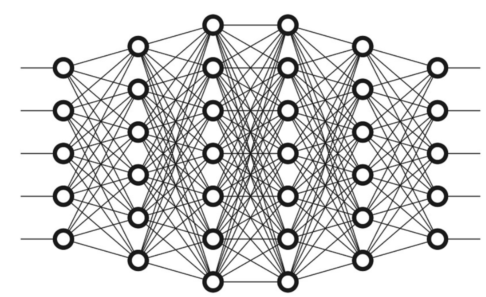
End-Device



Local DL

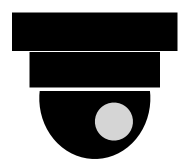


Edge-Server

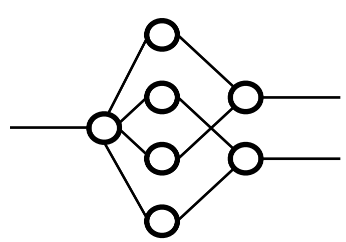


Remote DL

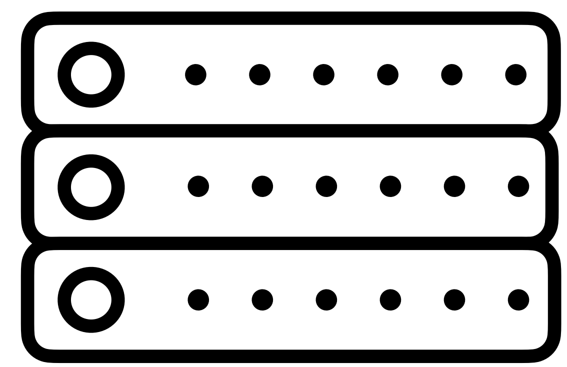
System Components



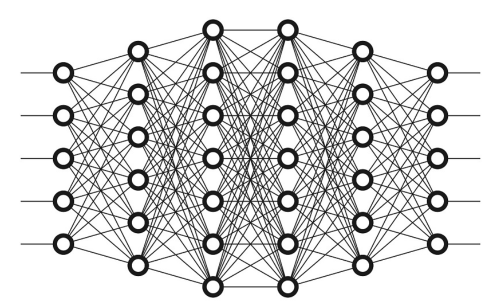
End-Device



Local DL



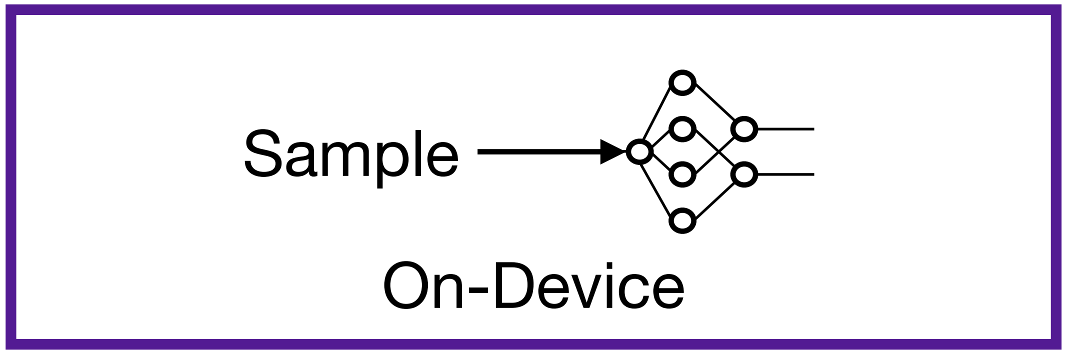
Edge-Server



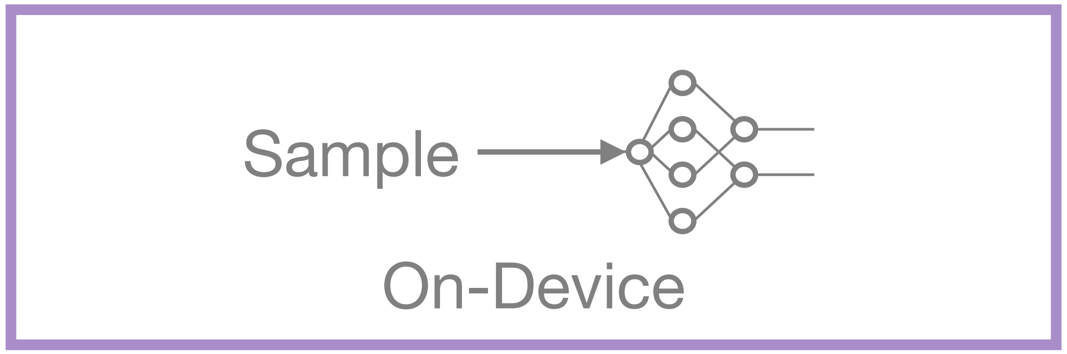
Remote DL

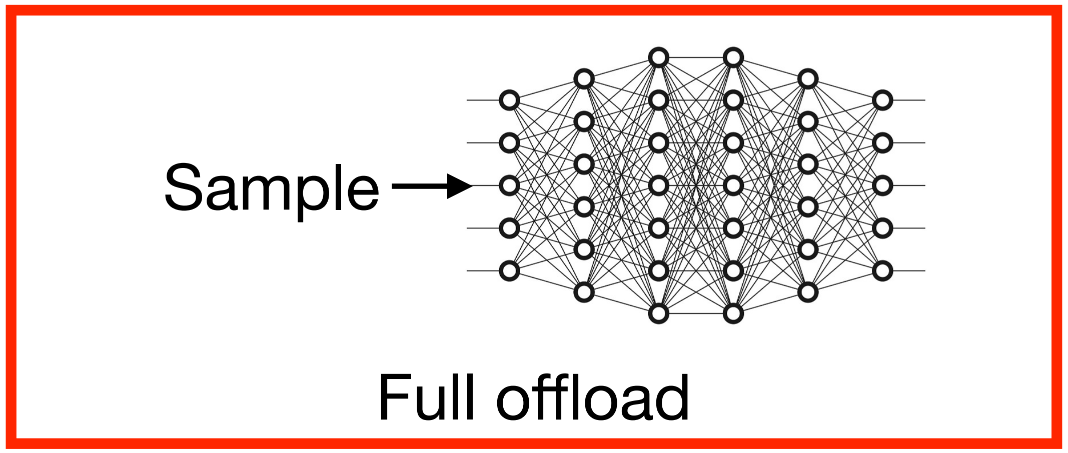
🤔 Where should we do the inference?

Candidate Strategies

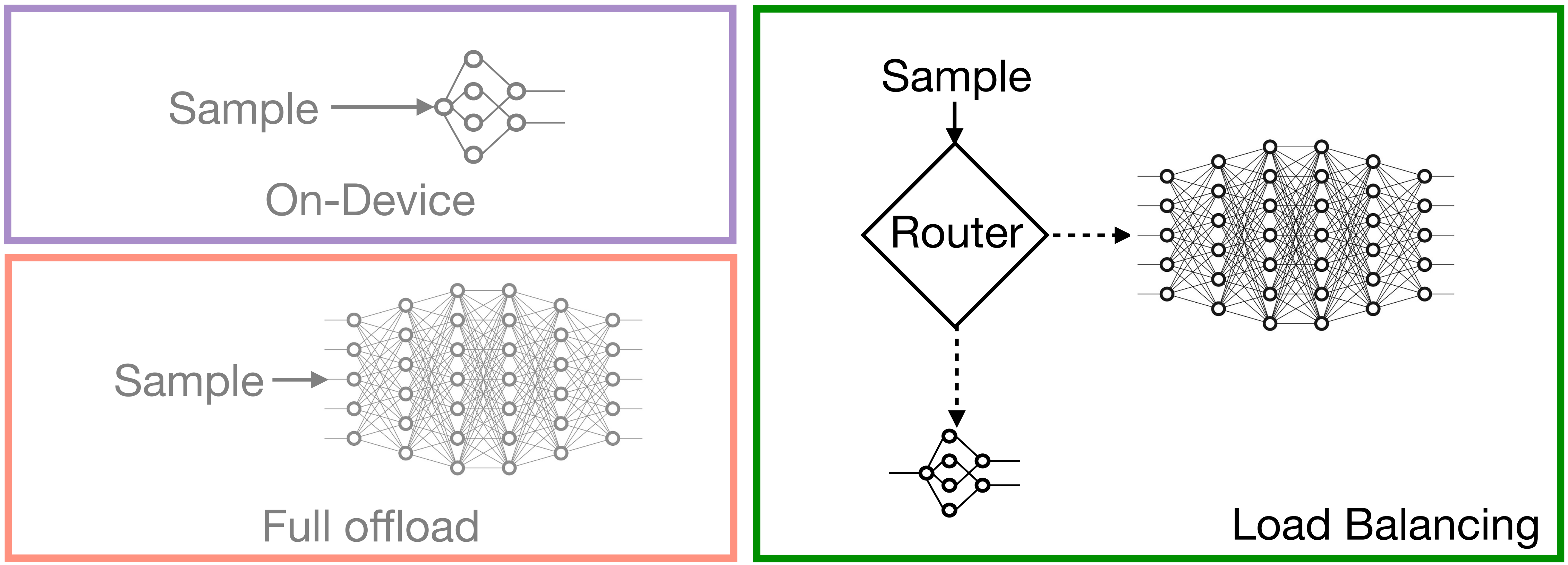


Candidate Strategies

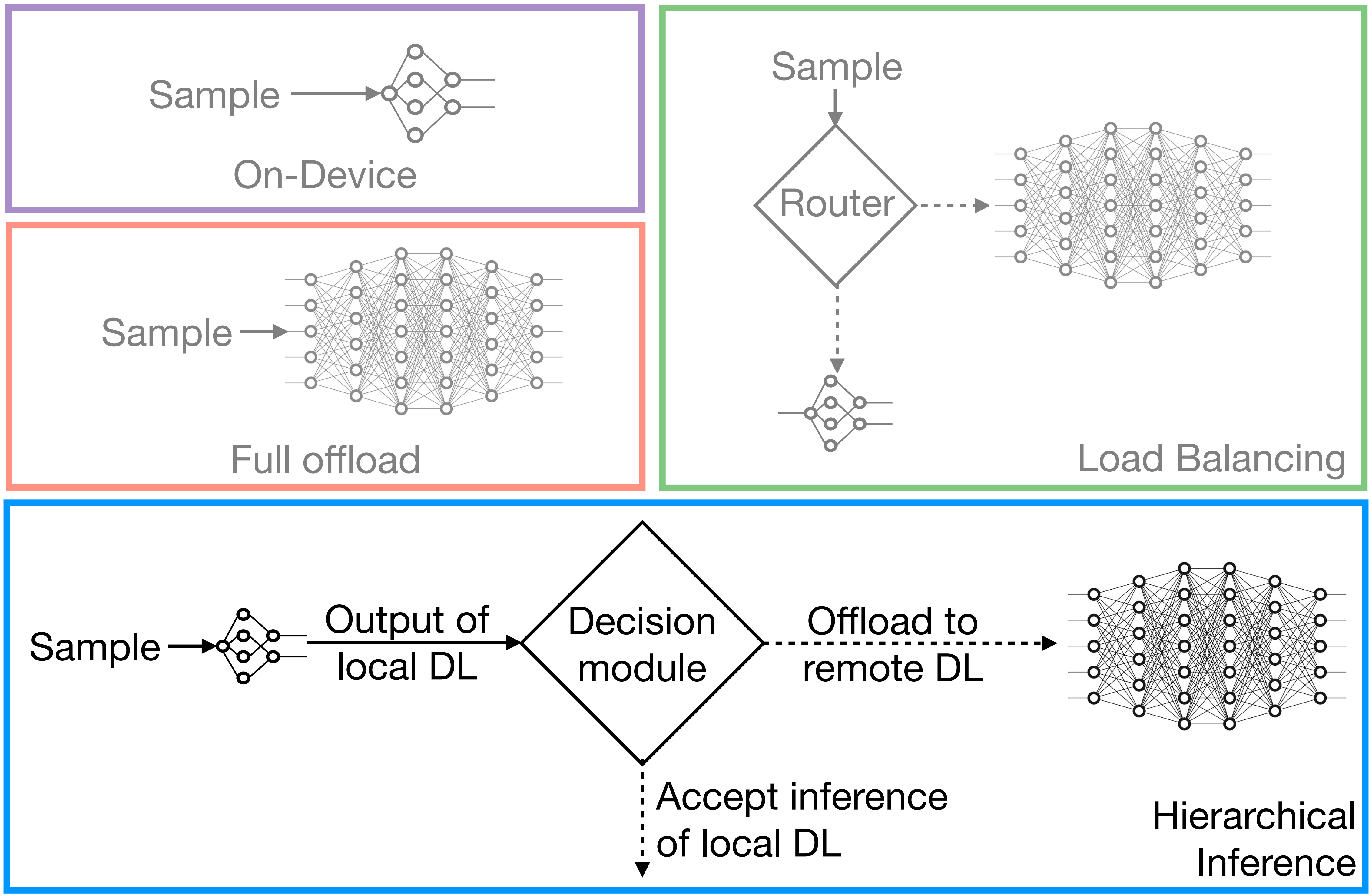




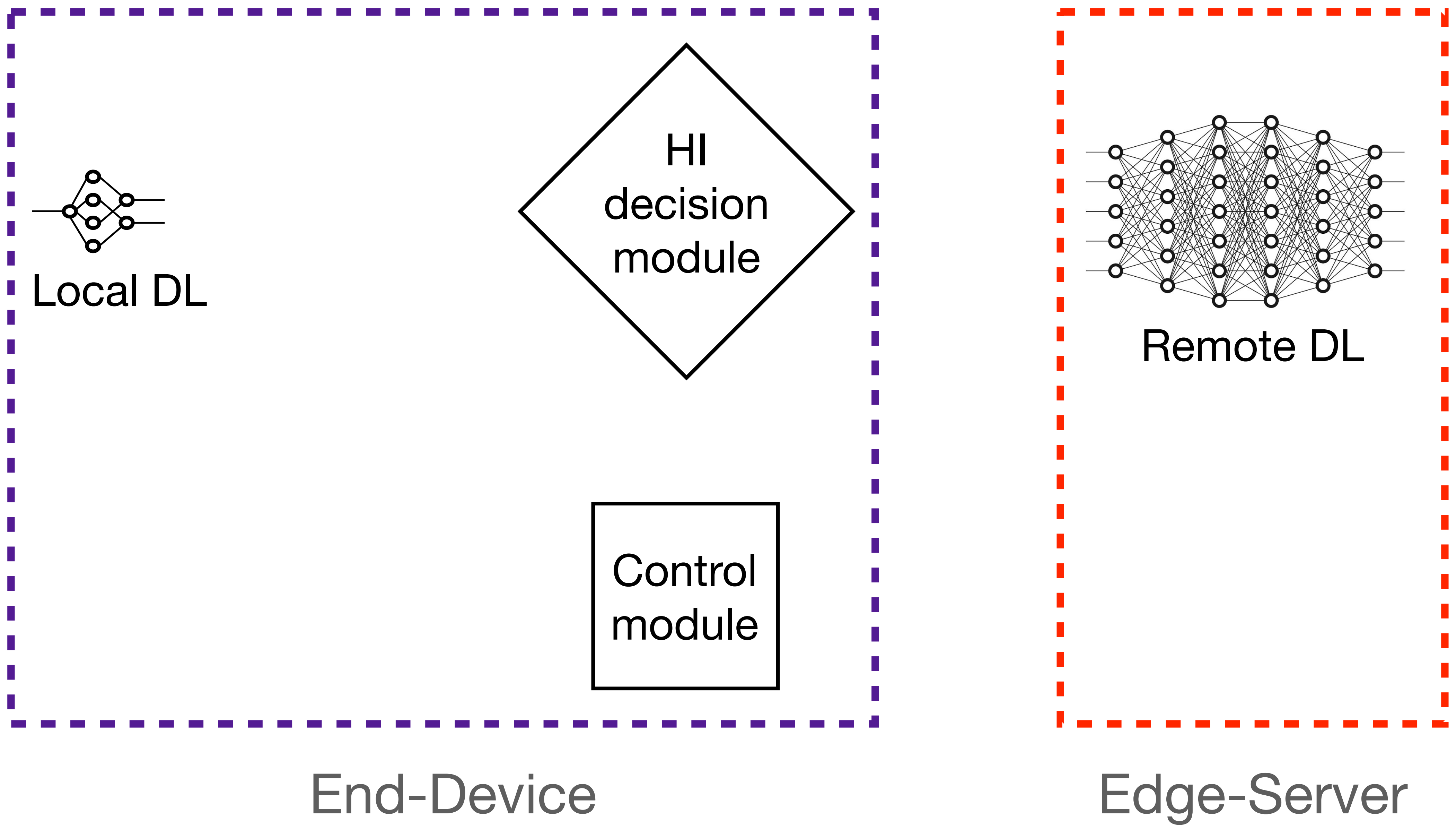
Candidate Strategies



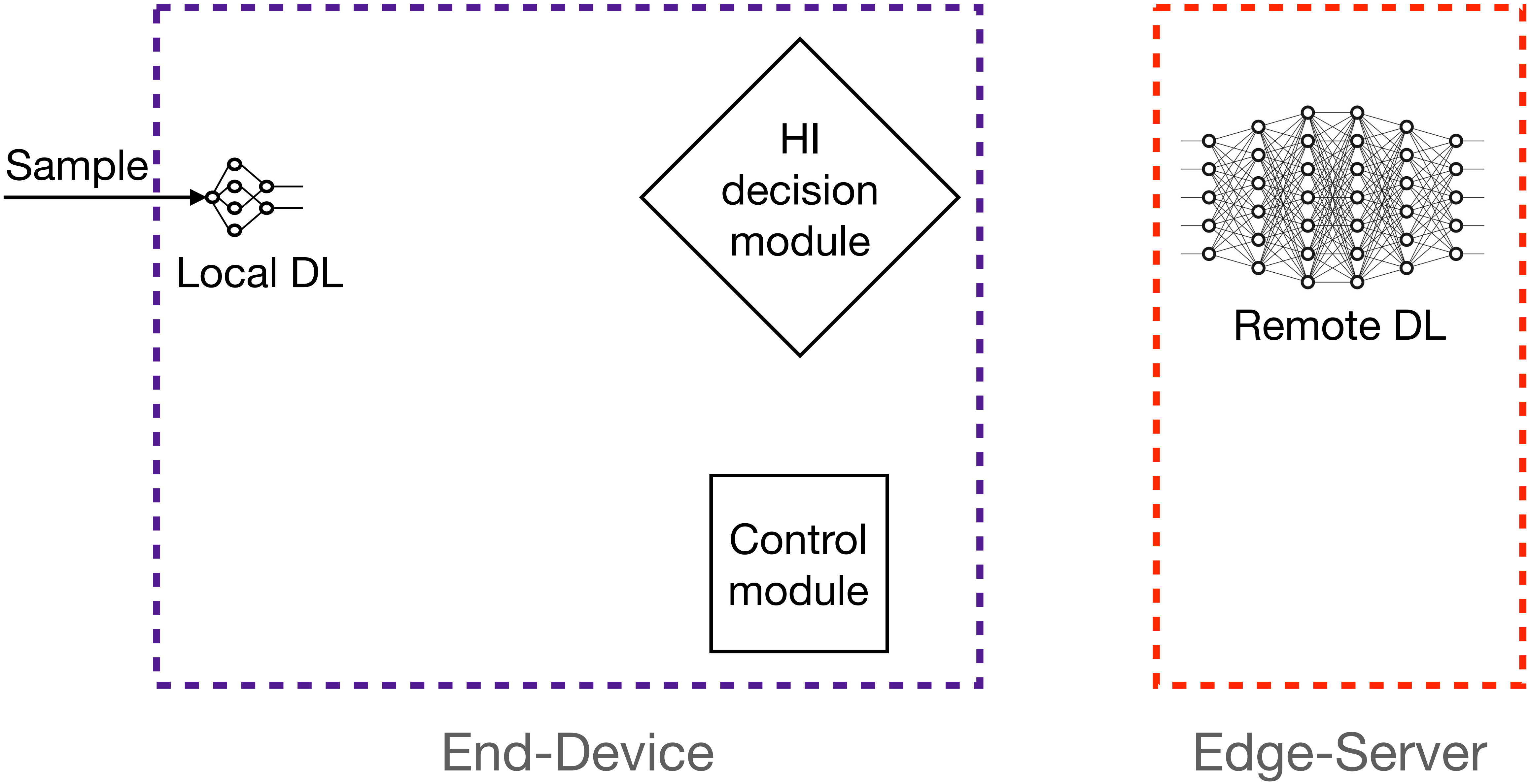
Candidate Strategies



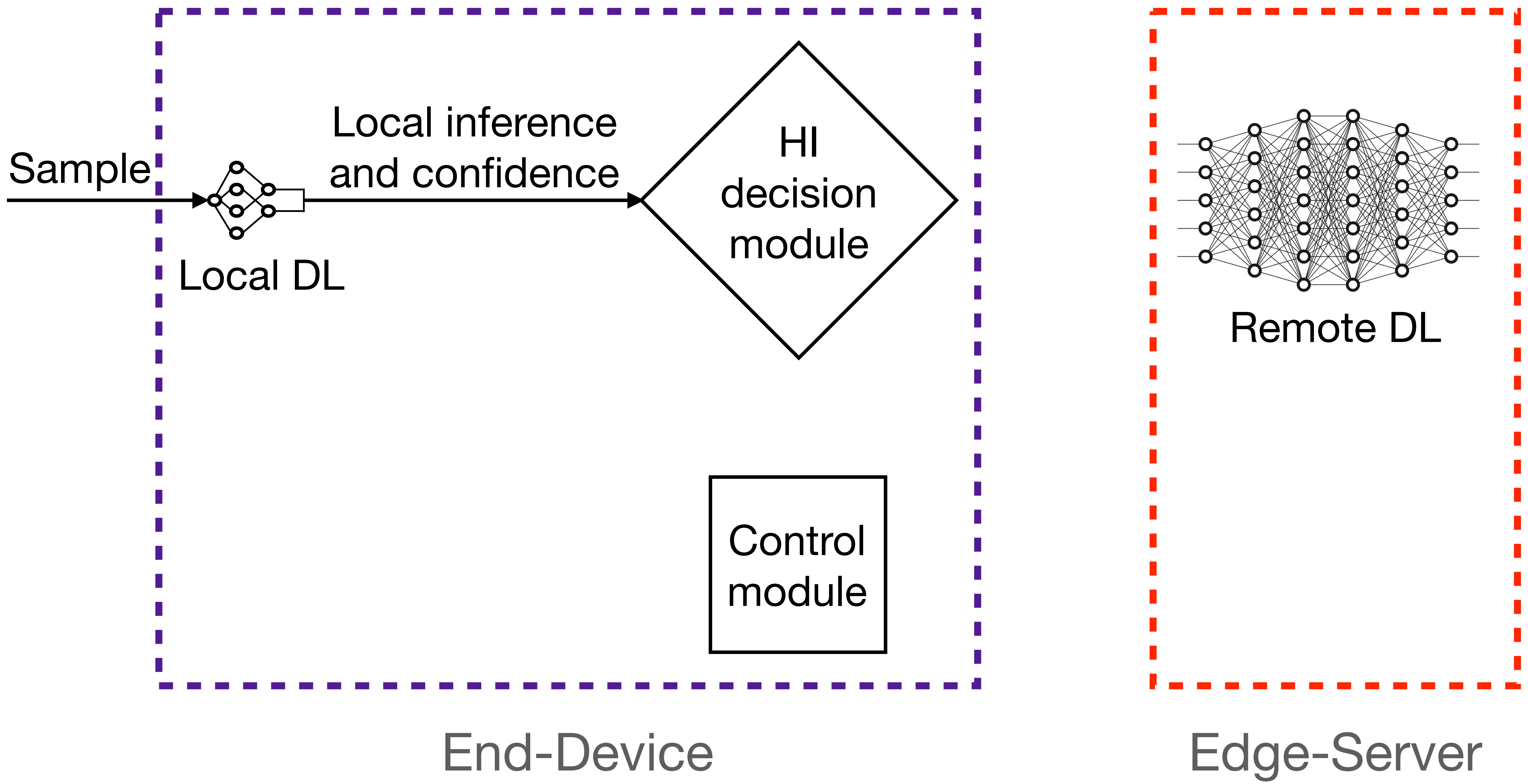
HI Schematic



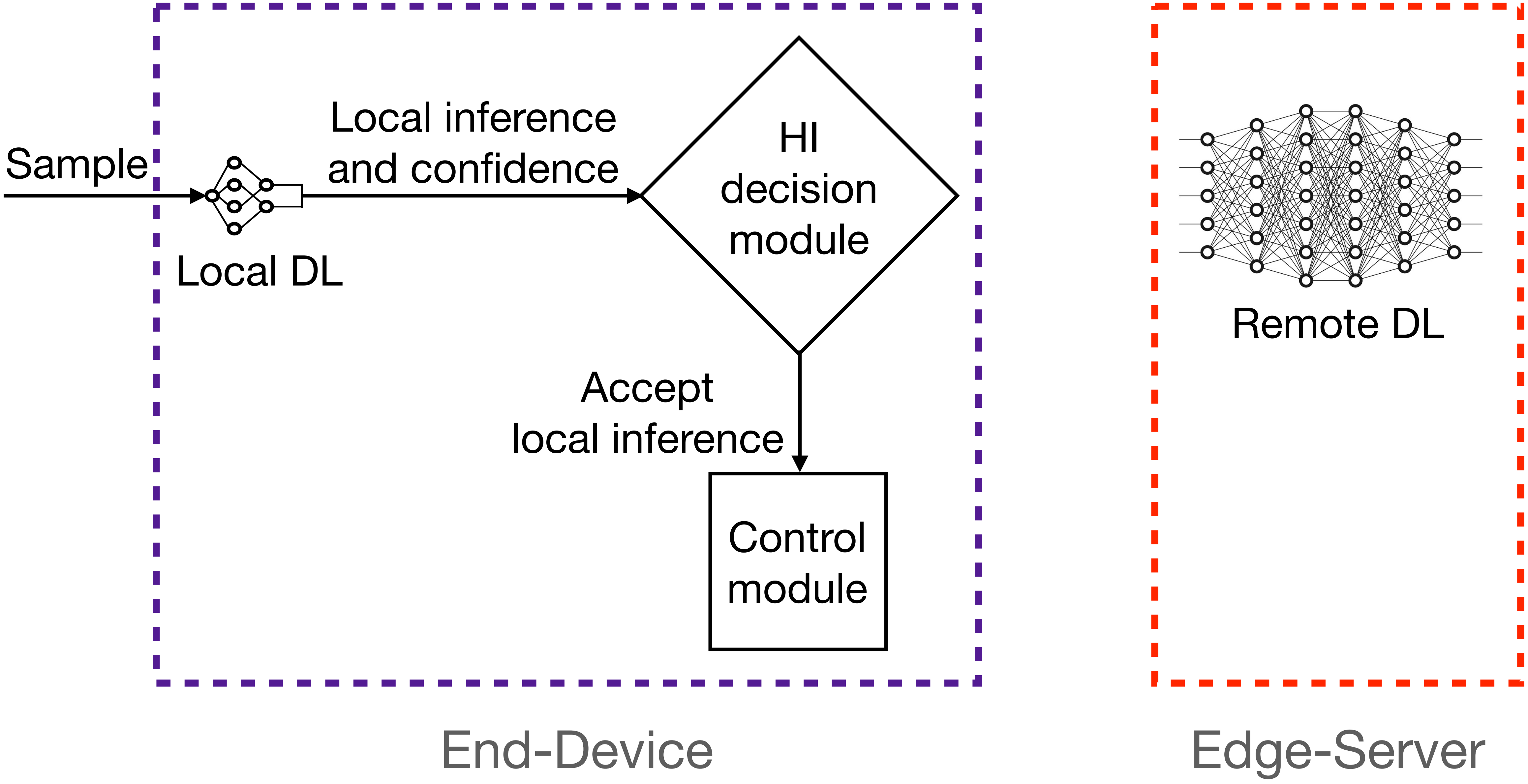
HI Schematic



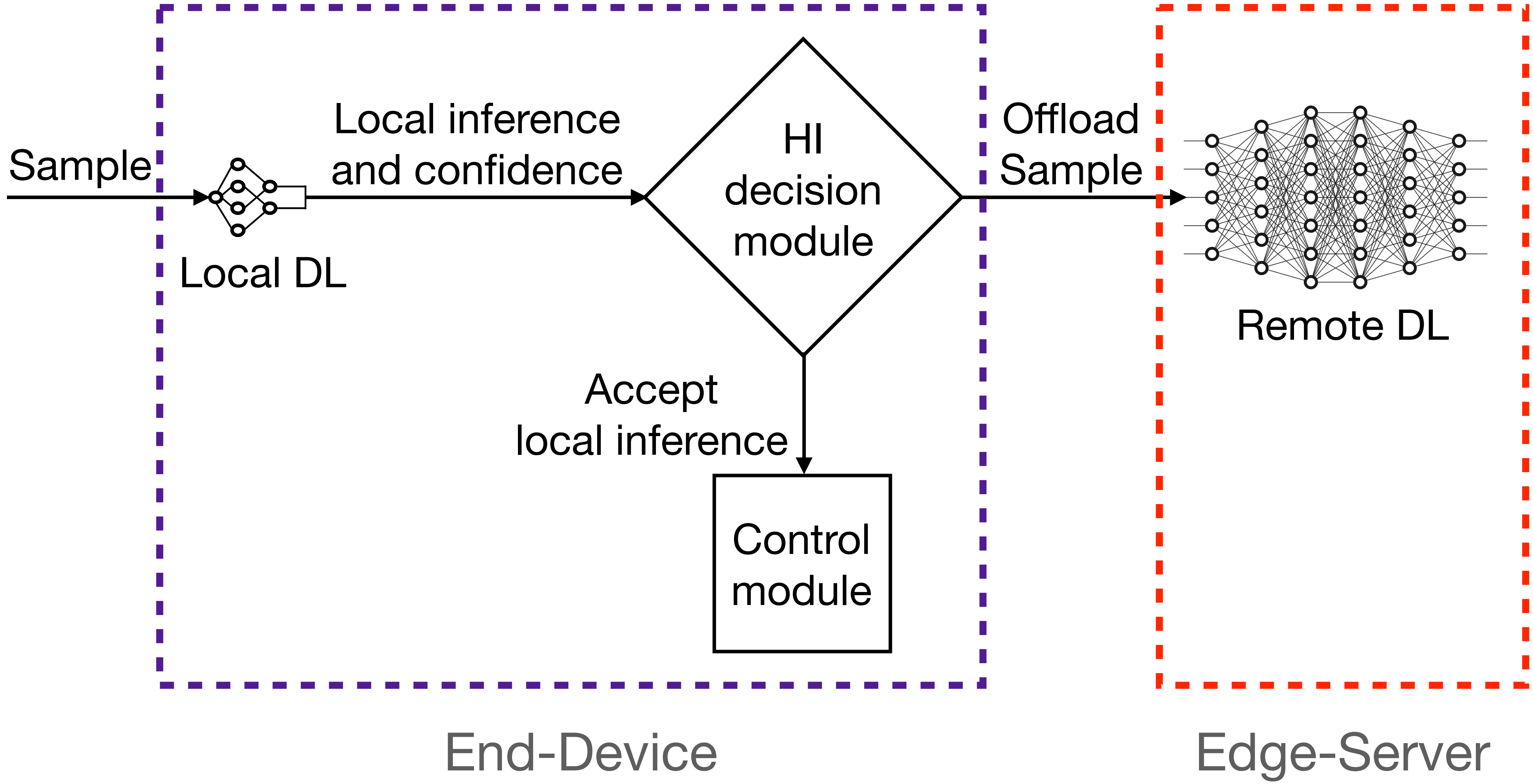
HI Schematic



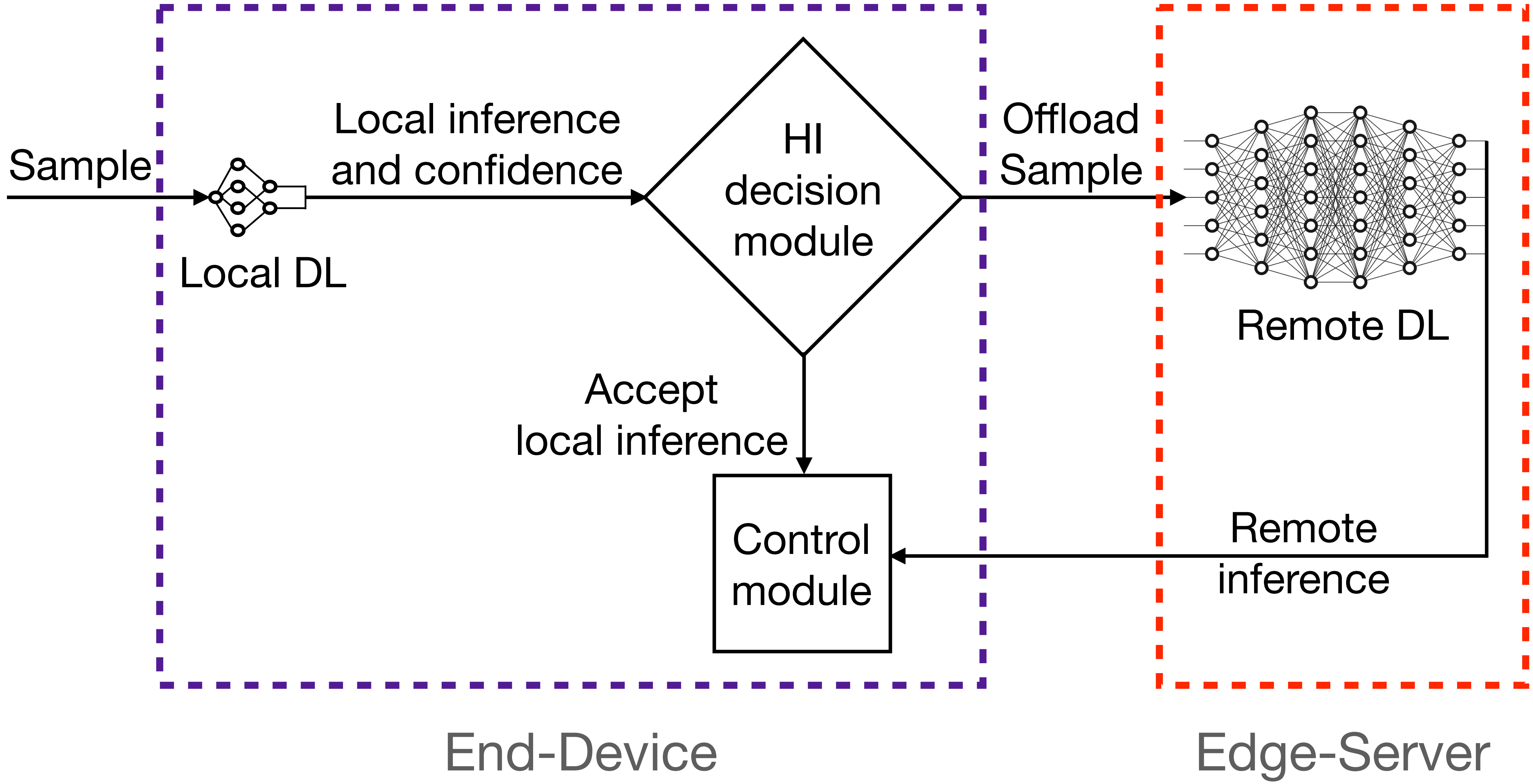
HI Schematic



HI Schematic



HI Schematic



Confidence

- Measure of the confidence the model has in its inference
- DL model outputs a score for each candidate class
- Sample typically classified into class with highest score
- Confidence = $f($score vector for that sample$)$

Confidence

- Measure of the confidence the model has in its inference
- DL model outputs a score for each candidate class
- Sample typically classified into class with highest score
- Confidence = f(score vector for that sample)

  Example: max soft-max value

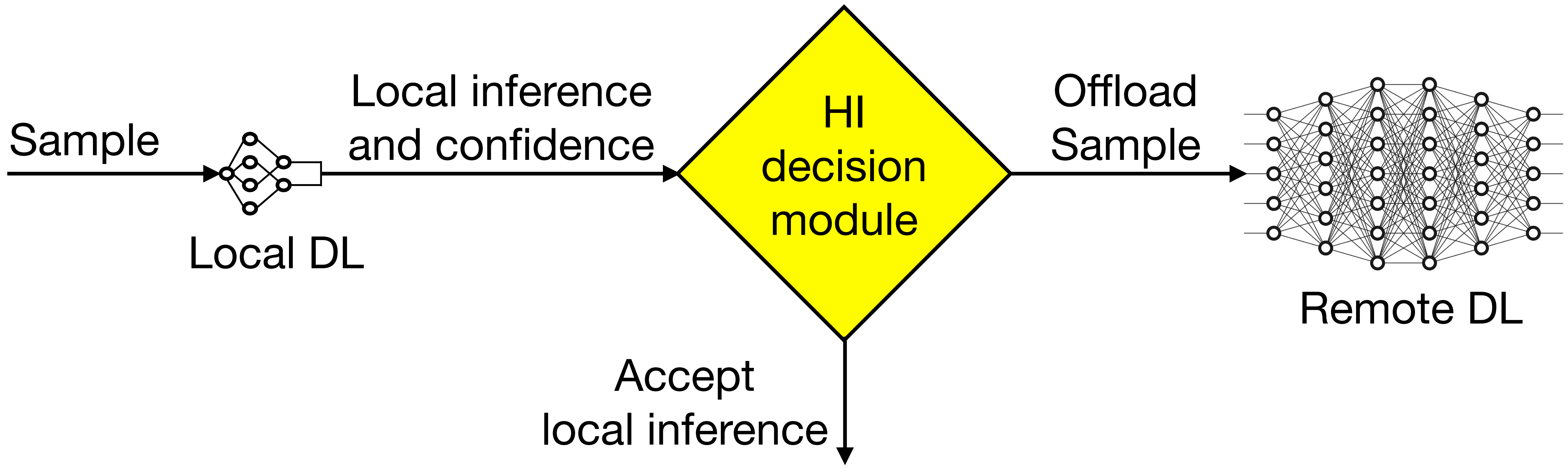
  $\mathscr{C}$: set of classes

  $s(c)$: score for a sample for class c

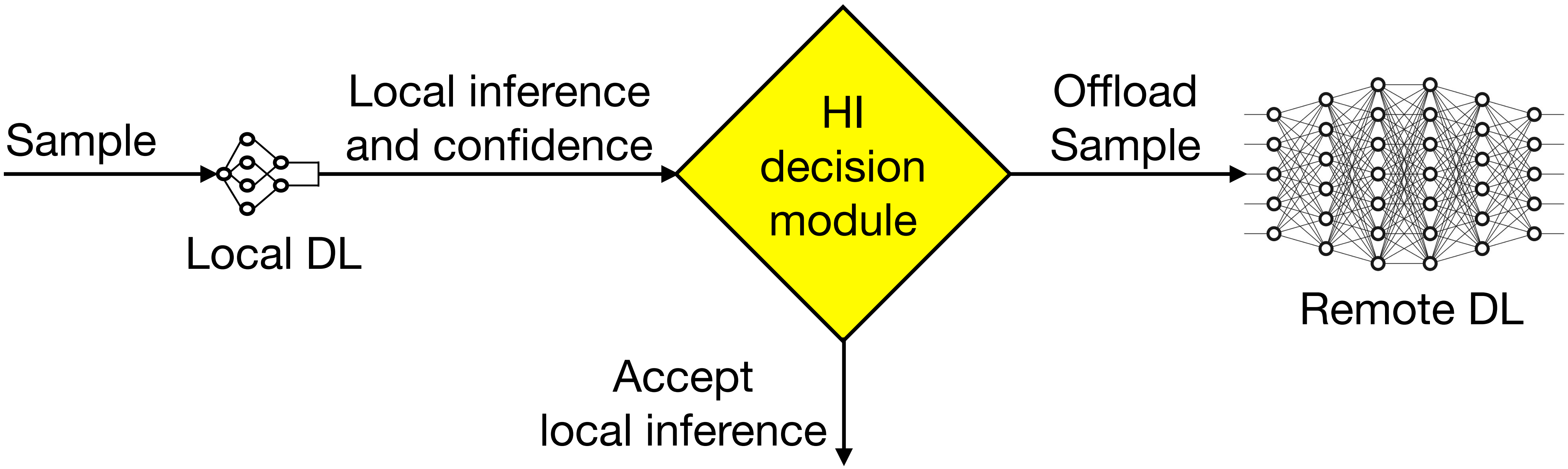
  Output: $\arg\max_{c \in \mathscr{C}} s(c)$

  Confidence: $\max_{c \in \mathscr{C}} \frac{e^{s(c)}}{\sum_{u \in \mathscr{C}} e^{s(u)}}$

Design Challenge in HI



Design Challenge in HI



💡 Offload if confidence is low, i.e., below a threshold

🤔 What should be the threshold?

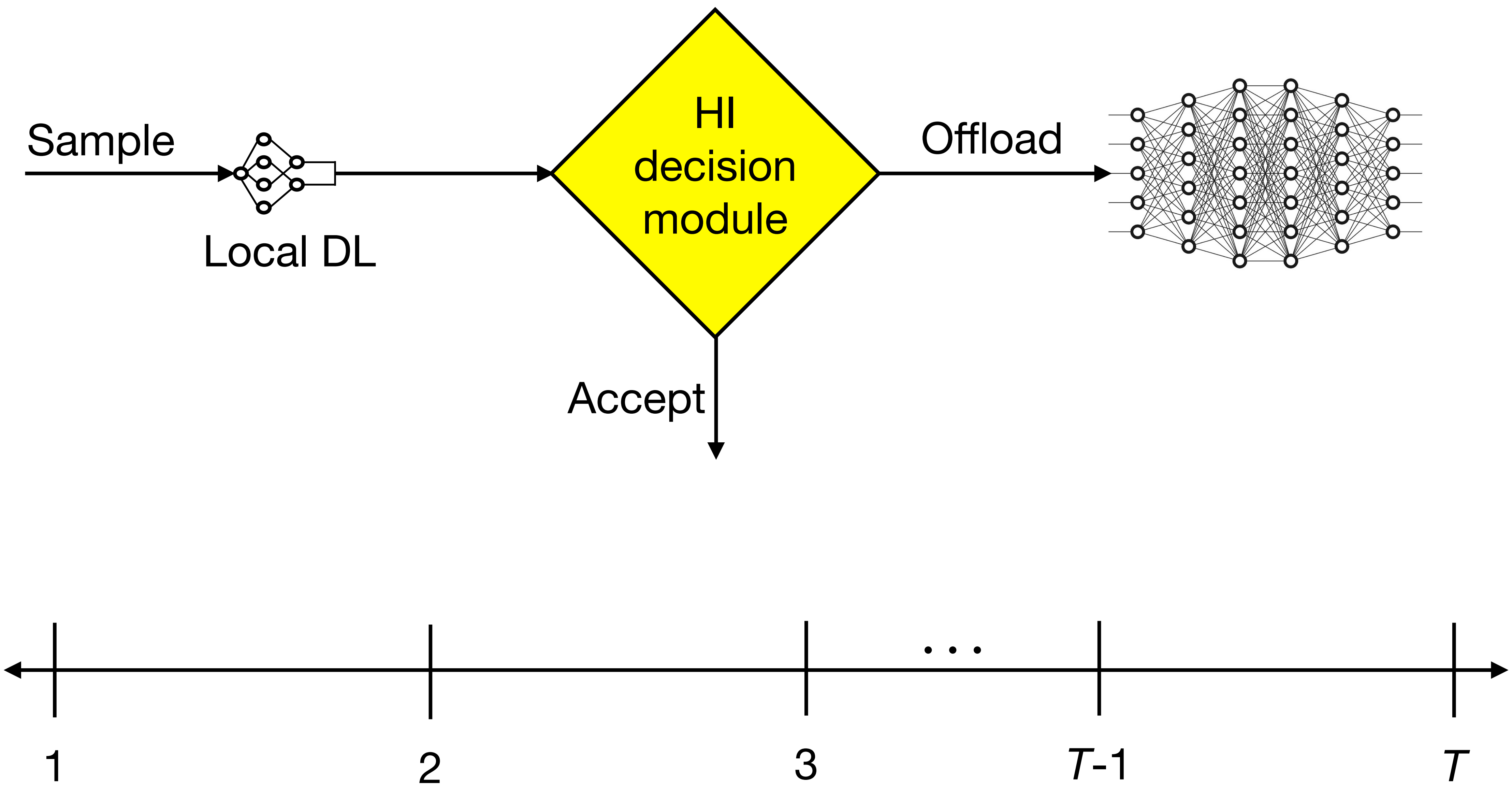
🤷‍♀️ Depends on system parameters and performance metric(s)

Prior Work on HI

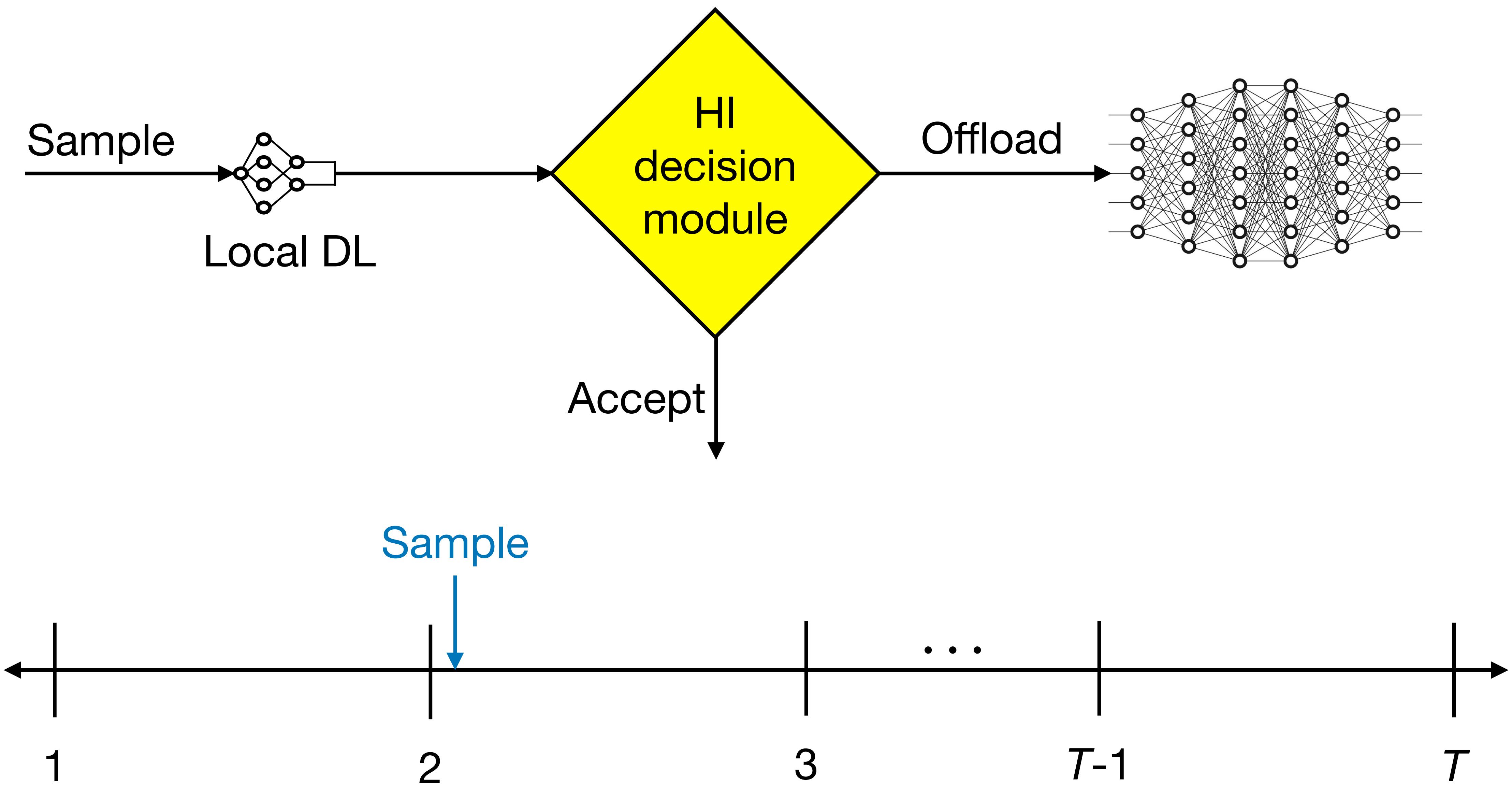
- Multiple use cases (Al-Atat, et al., *MobiSys* 2023)
- Threshold selection
  - based on transmission energy constraint of ED (Nikoloska, et al., *IEEE Communication Letters* 2023)
  - linear regression on two highest soft-max values to obtain threshold (Behera, et al., *ACM MobiCom* 2023)
- Online learning for finding optimal threshold, dataset dependent regret bound (Moothedath, et al., *IEEE TMLCN* 2024)
- Multiple EDs (Beytur, et al., *IEEE INFOCOM* 2024)

Online Learning for HI

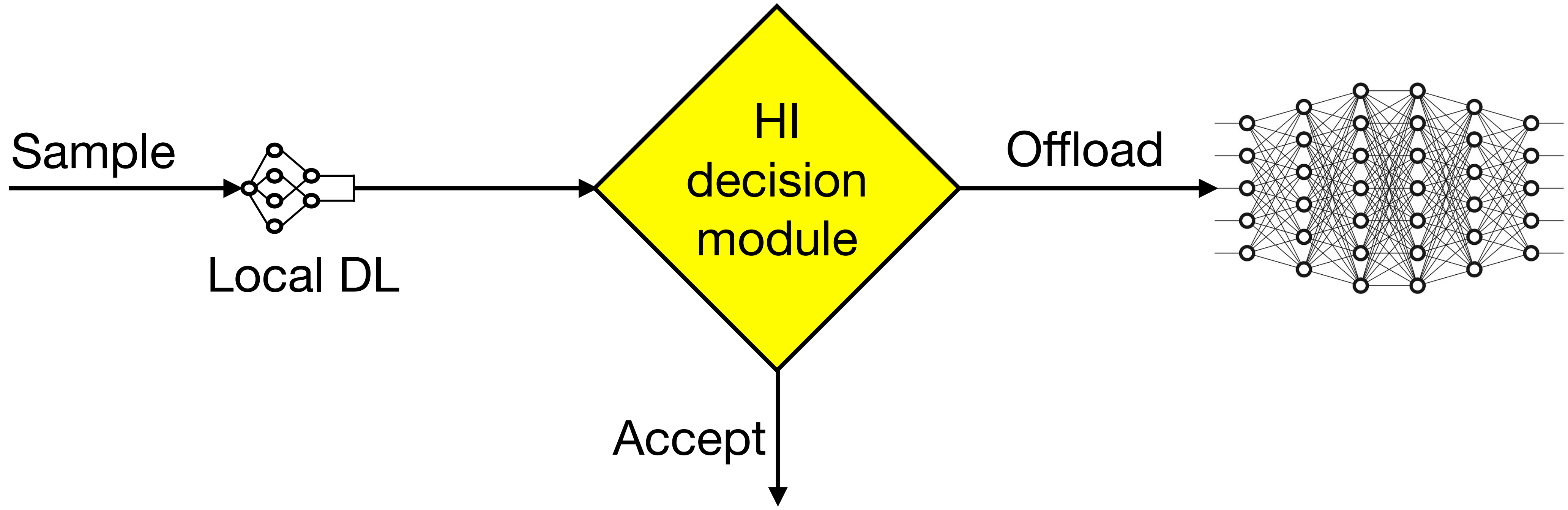
Sequence of Events

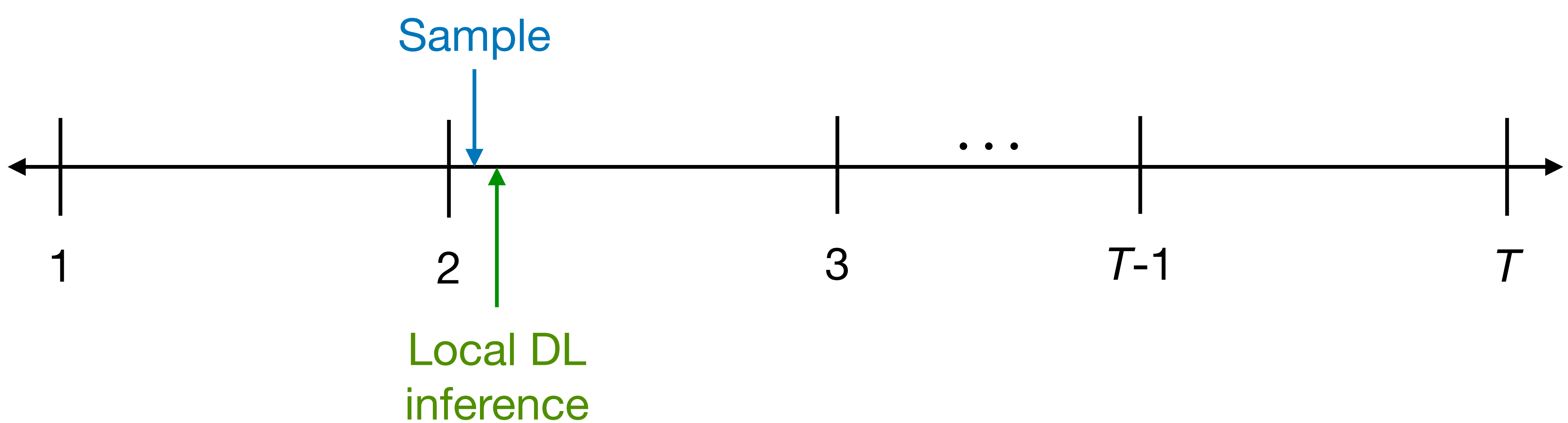


Sequence of Events

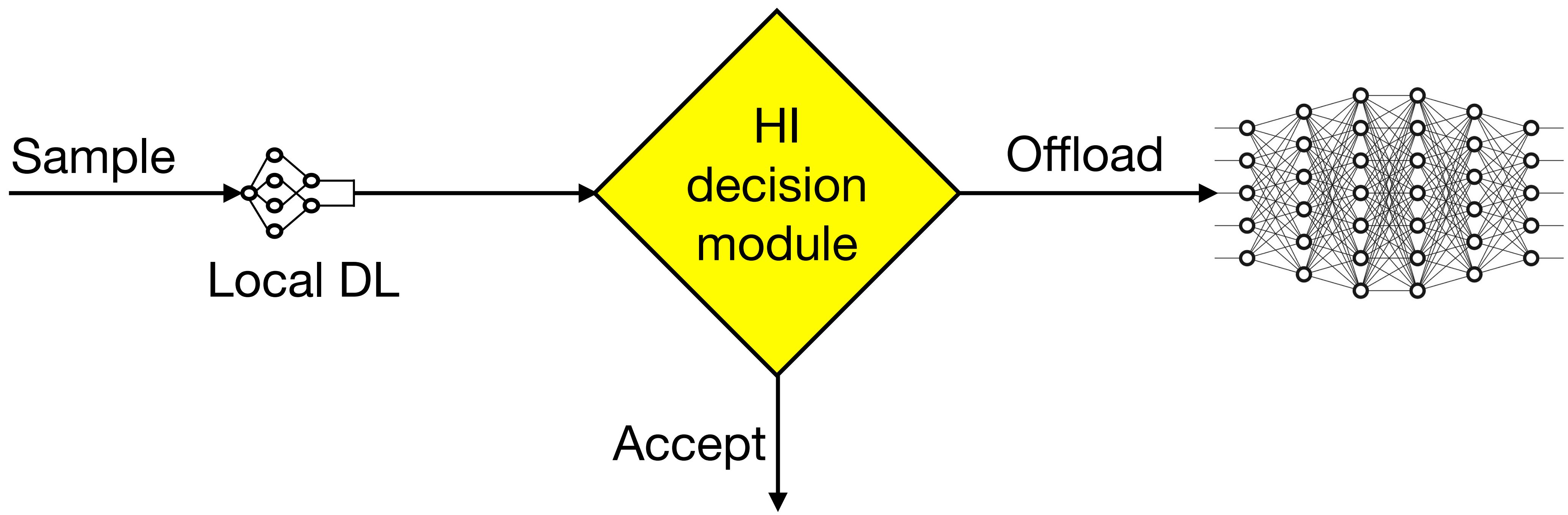


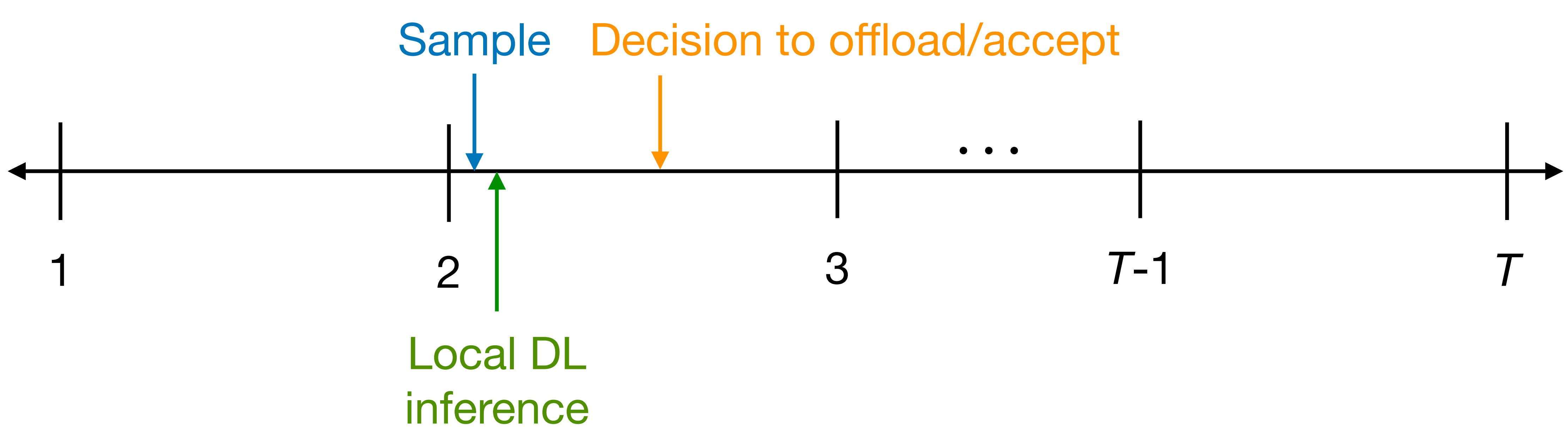
Sequence of Events



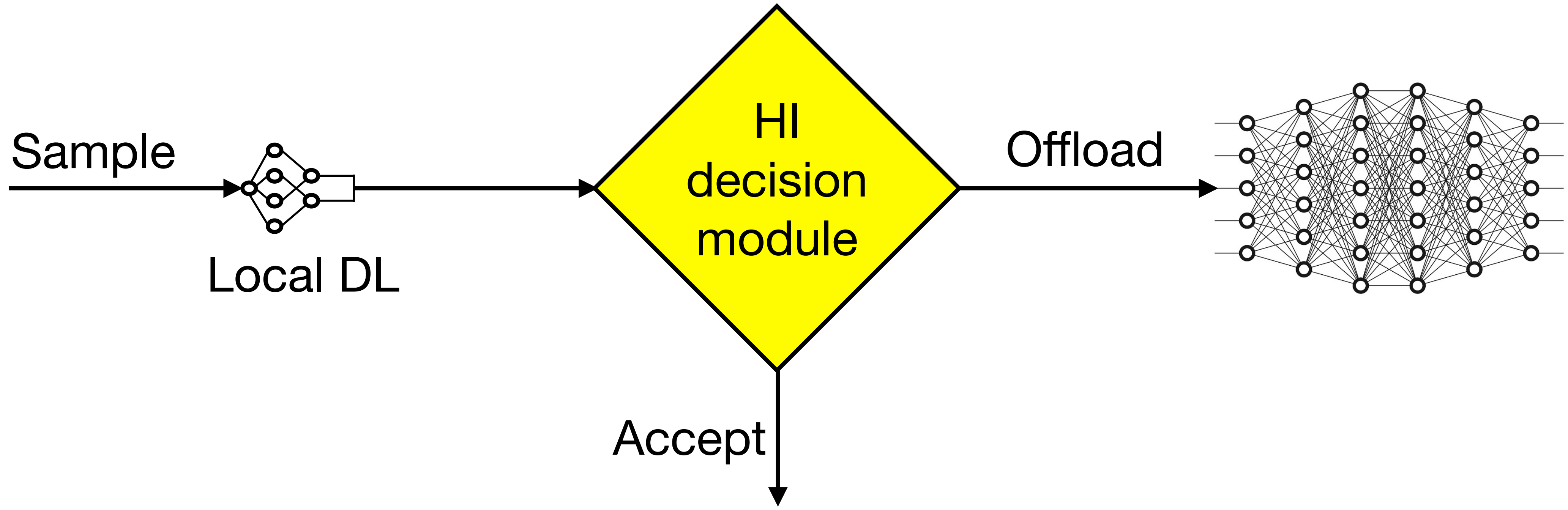


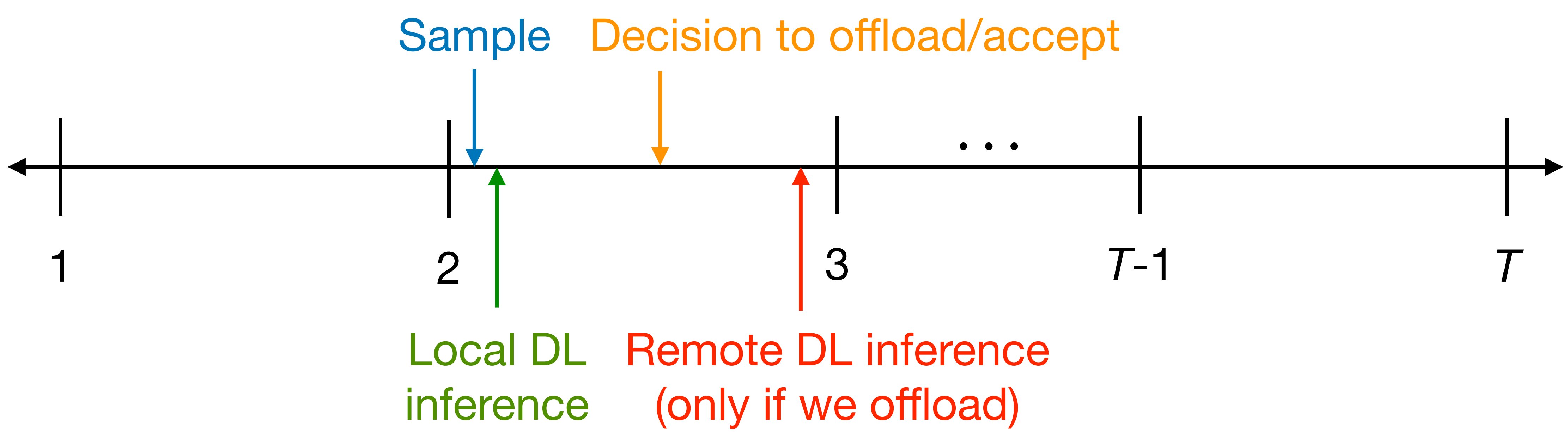
Sequence of Events





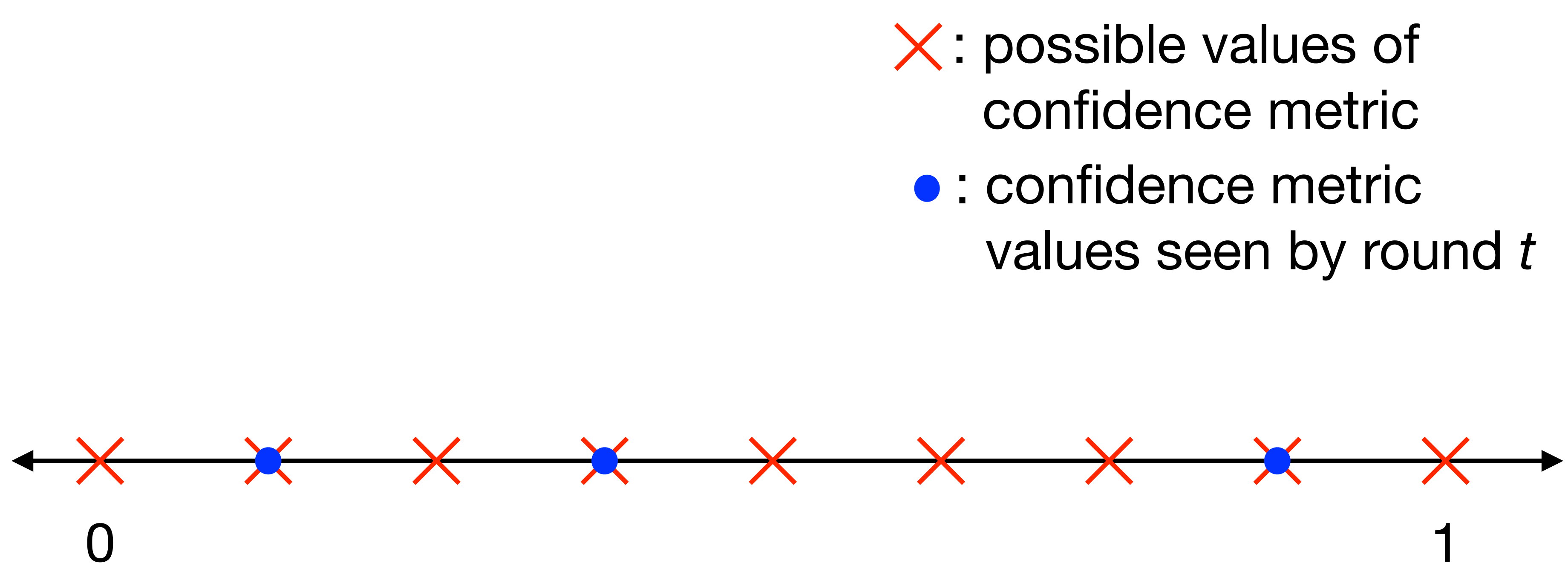
Sequence of Events



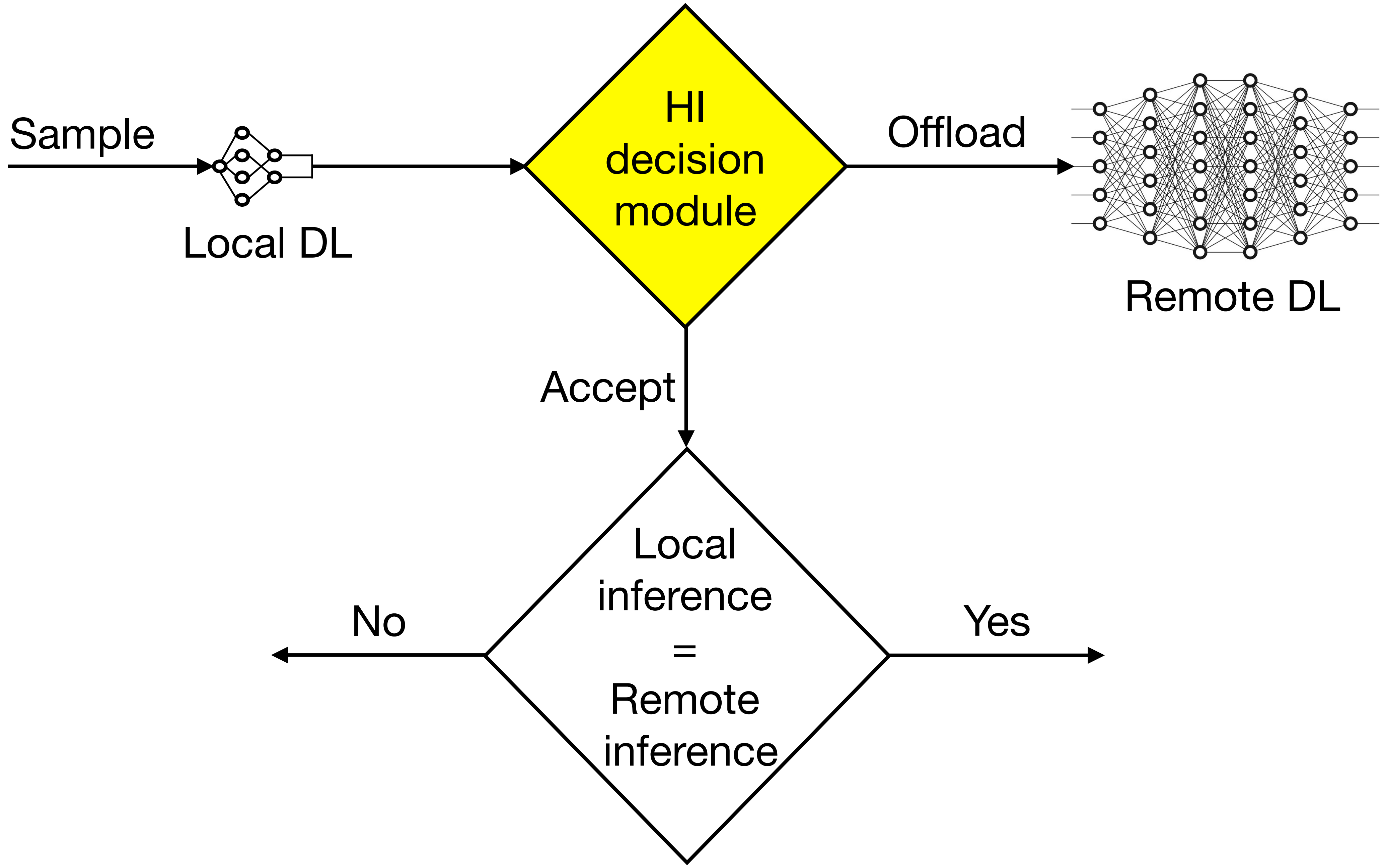


Confidence Values

- Measure of the confidence the model has in its inference
  - Examples: max soft-max value
- Belong to a discrete set
- Stochastic, generated i.i.d. across time (distribution unknown)

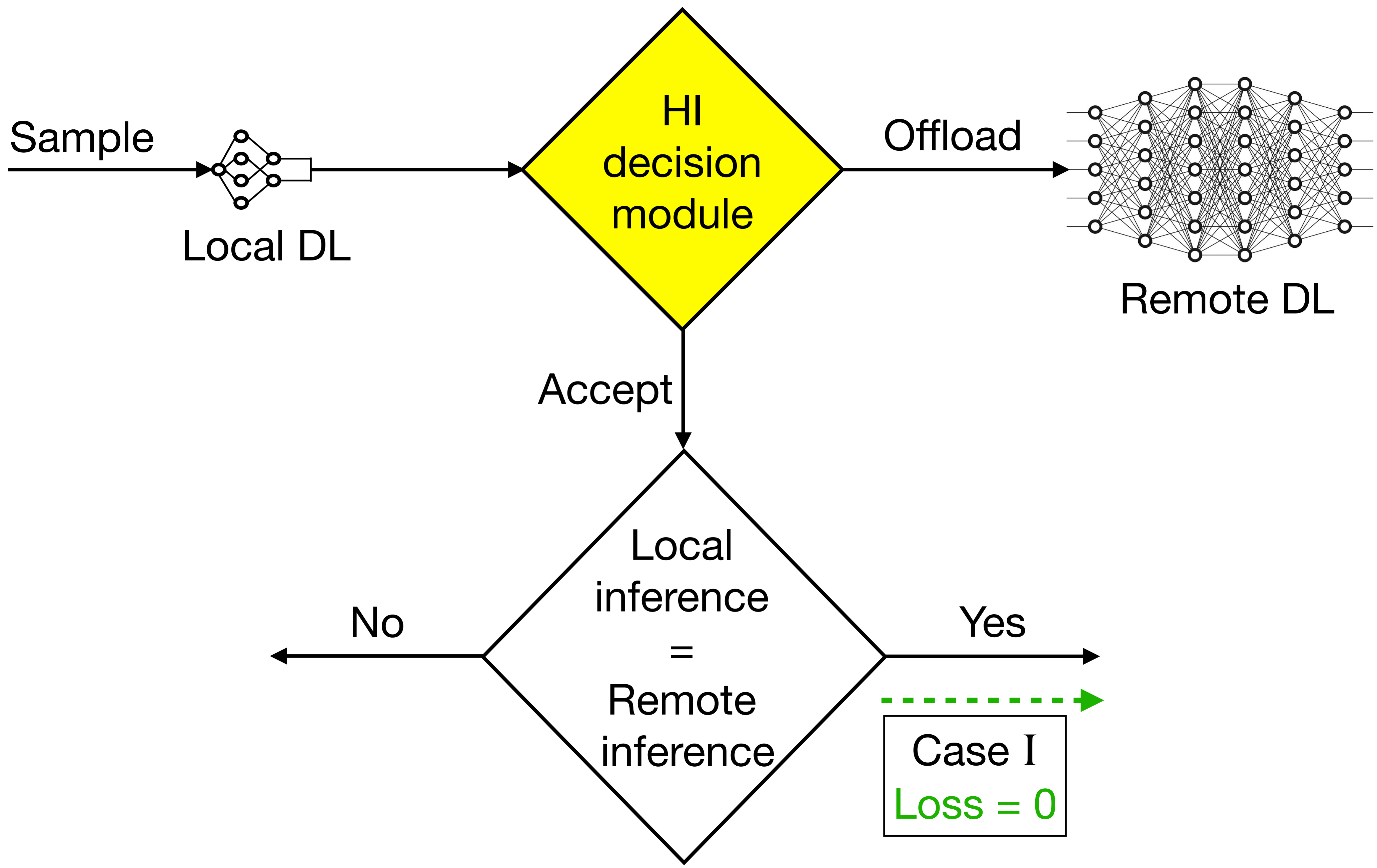


Loss Model

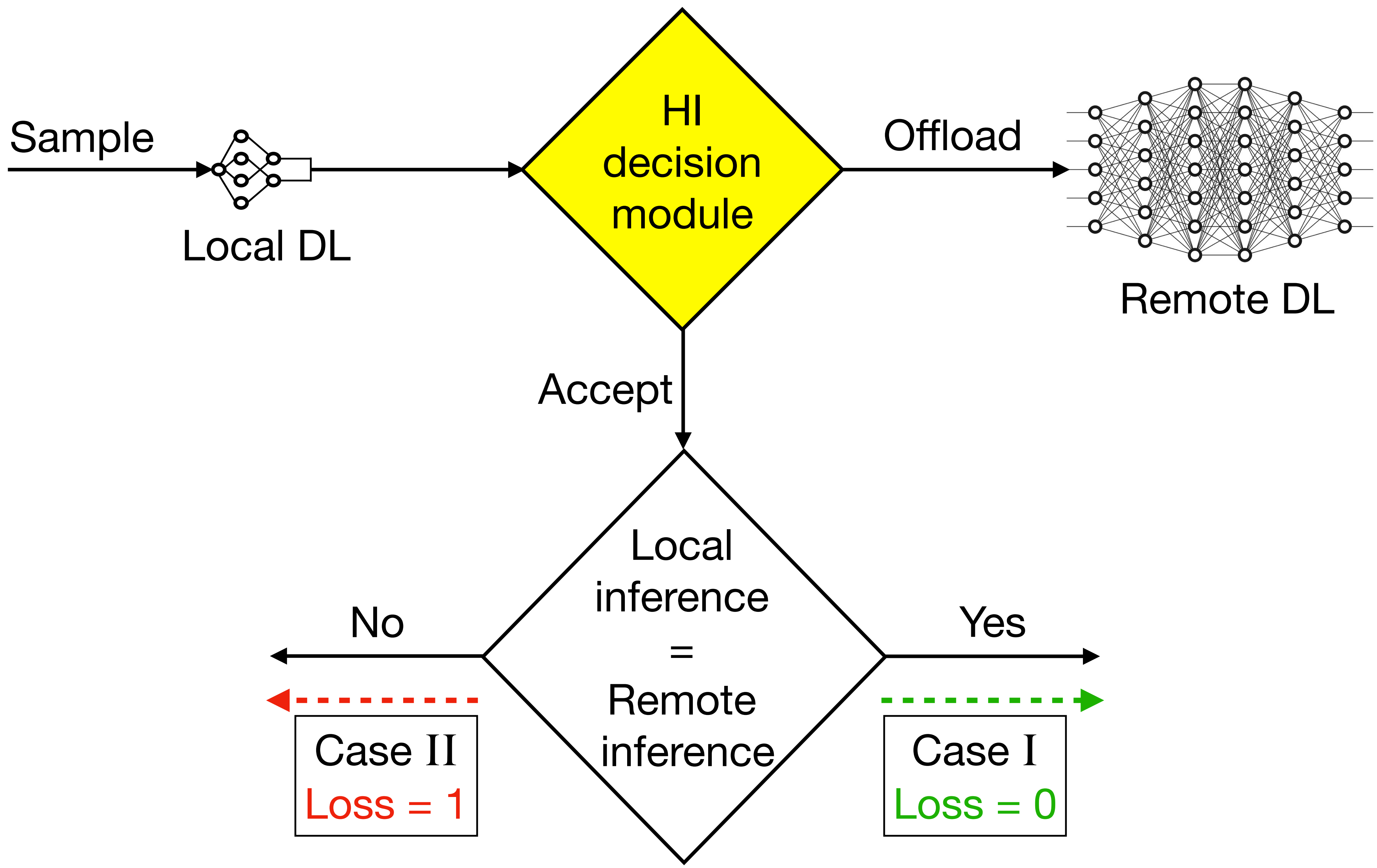


Sample
Local DL
HI decision module
Offload
Remote DL
Accept
Local inference = Remote inference
No
Yes

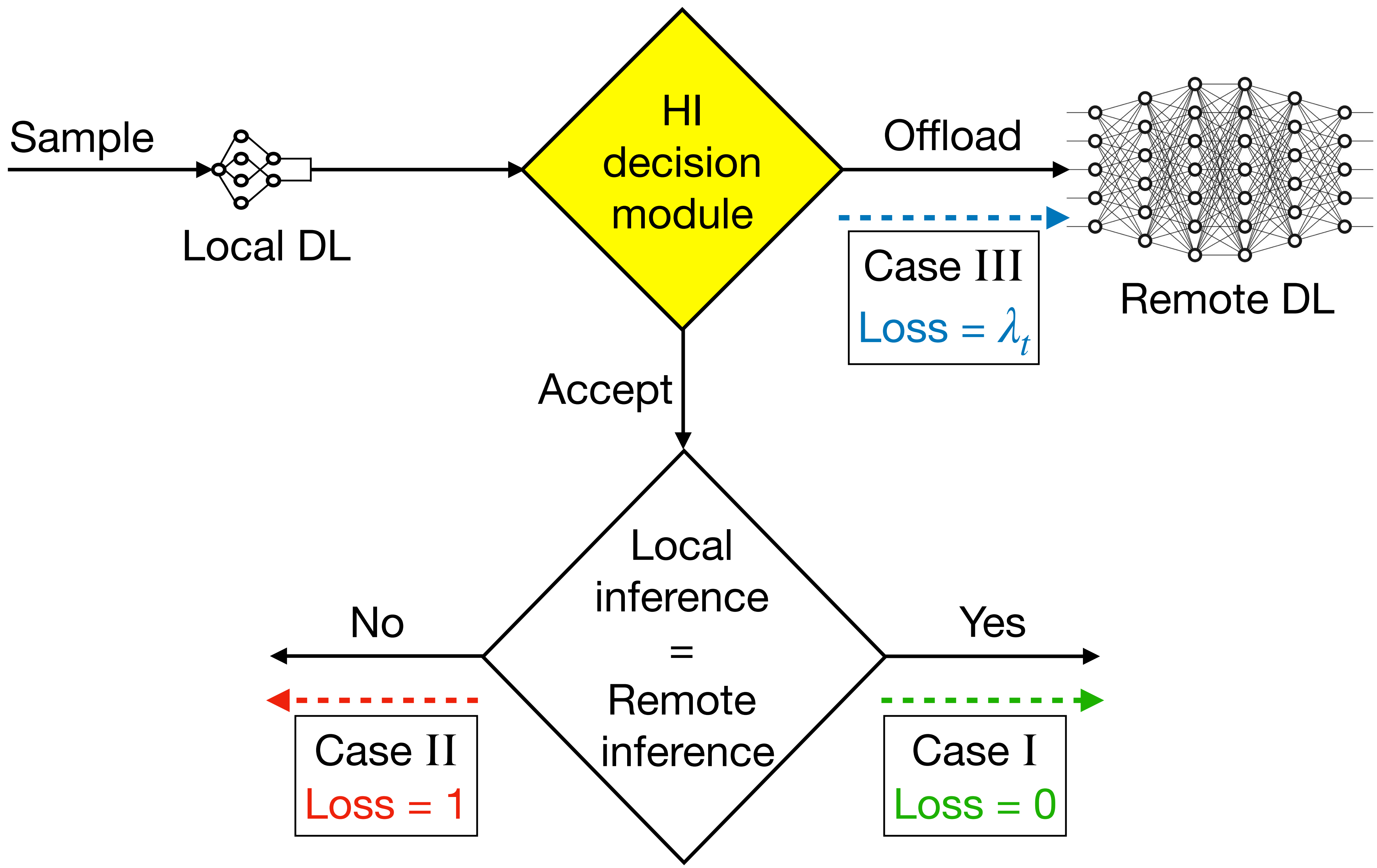
Loss Model


Sample
Local DL
HI
decision
module
Offload
Remote DL
Accept
Local
inference
=
Remote
inference
No
Yes
Case I
Loss = 0

Loss Model

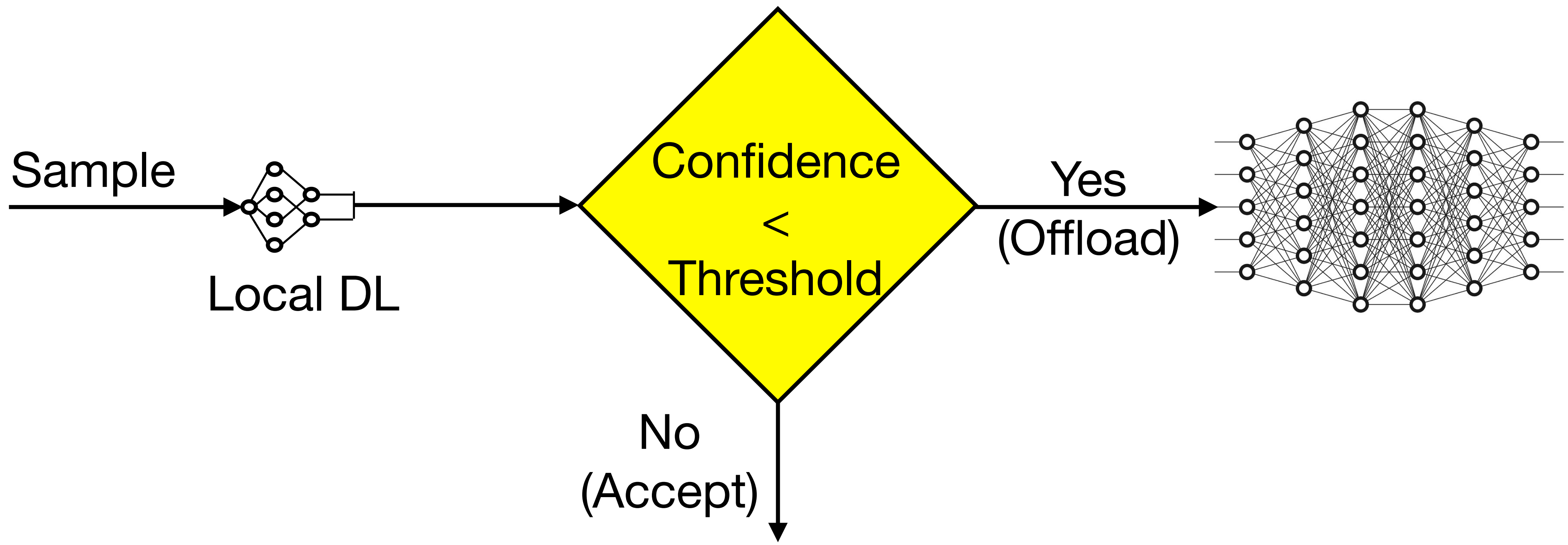

Sample
Local DL
HI decision module
Offload
Remote DL
Accept
Local inference = Remote inference
No
Yes
Case II
Loss = 1
Case I
Loss = 0

Loss Model



Sample
Local DL
HI decision module
Offload
Remote DL
Case III
Loss = λ_t
Accept
Local inference = Remote inference
No
Yes
Case II
Loss = 1
Case I
Loss = 0

Threshold Policies

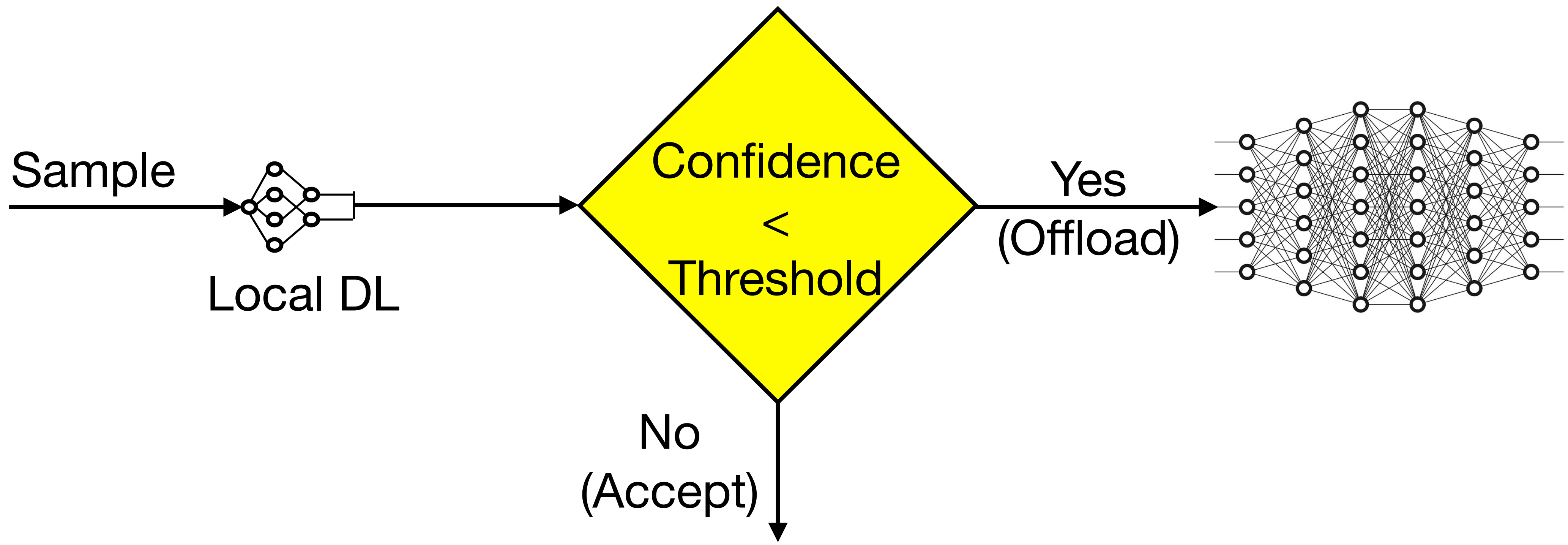


Key idea:

- Scalar parameter (threshold)
- Offload $\iff$ confidence < threshold
- Threshold can vary over time

Fixed threshold policies: threshold time-invariant

Performance Metric



Baseline: Optimal fixed threshold policy (OPT)

Candidate policy P: Attempts to learn the optimal threshold

$$\text{Regret}_{\text{P}}(T) = \sum_{t=1}^{T} \mathbb{E}[\text{Loss}_{\text{P}}(t)] - \sum_{t=1}^{T} \mathbb{E}[\text{Loss}_{\text{OPT}}(t)]$$

Online Learning for HI

Motivation

Our Setting

Main Results

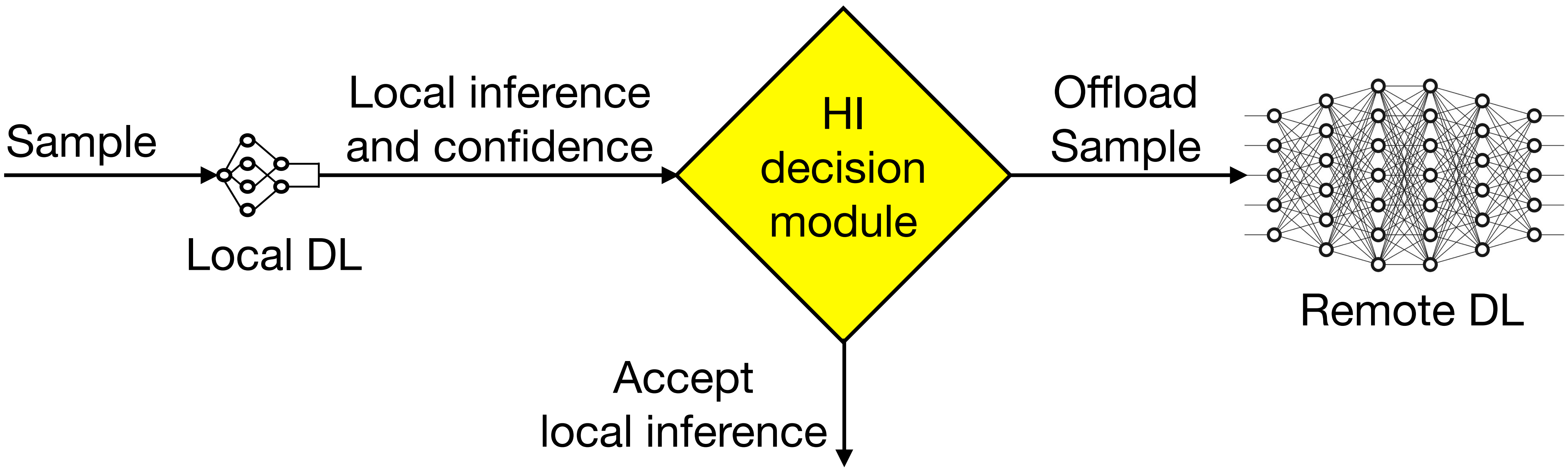
Background: Prediction with Experts

Our Algorithms and Guarantees

Numerical Results

Conclusions

Main Results



💡 Offload if confidence is low, i.e., below a threshold

🤔 What should be the threshold?

🥳 Online algorithms to choose threshold with sub-linear regret

Online Learning for HI

Prediction with Experts

- K experts
- Time divided into rounds
- Algorithmic task: choosing 1 expert per round
- K-dimensional loss vector in each round
  - Adversarial losses
  - Loss revealed after expert chosen
  - System incurs loss of chosen expert
- Static policy: chooses the same expert in all rounds
- Baseline: optimal static policy
- Goal: minimize cumulative regret in rounds 1 to T

loss

0.3

0.2

0.4

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1,\ 2, \cdots,\ T$

Choose expert k with probability $\propto w_k(t)$

Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1,\ 2, \cdots,\ T$

Choose expert k with probability $\propto w_k(t)$

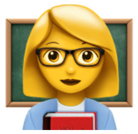
Receive loss vector $[l_1(t), l_2(t),\ \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$

weight

1 

1 

1 

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1,\ 2, \cdots,\ T$

  Choose expert k with probability $\propto w_k(t)$

  Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

  Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$

weight

1 🧑‍🏫

1 👩‍🏫

✅ 1 👨‍🏫

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1,\ 2, \cdots,\ T$

Choose expert k with probability $\propto w_k(t)$

Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$

	weight		loss
	1	🧑‍🏫	0.3
	1	👩‍🏫	0.2
✅	1	👨‍🏫	0.4

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \le k \le K$

For $t = 1,\ 2, \cdots,\ T$

Choose expert k with probability $\propto w_k(t)$

Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$				$t = 2$	
weight		loss		weight	
1	🧑‍🏫	0.3		$e^{-0.3\eta}$	🧑‍🏫
1	👩‍🏫	0.2	→	$e^{-0.2\eta}$	👩‍🏫
✅ 1	👨‍🏫	0.4		$e^{-0.4\eta}$	👨‍🏫

The Hedge Algorithm

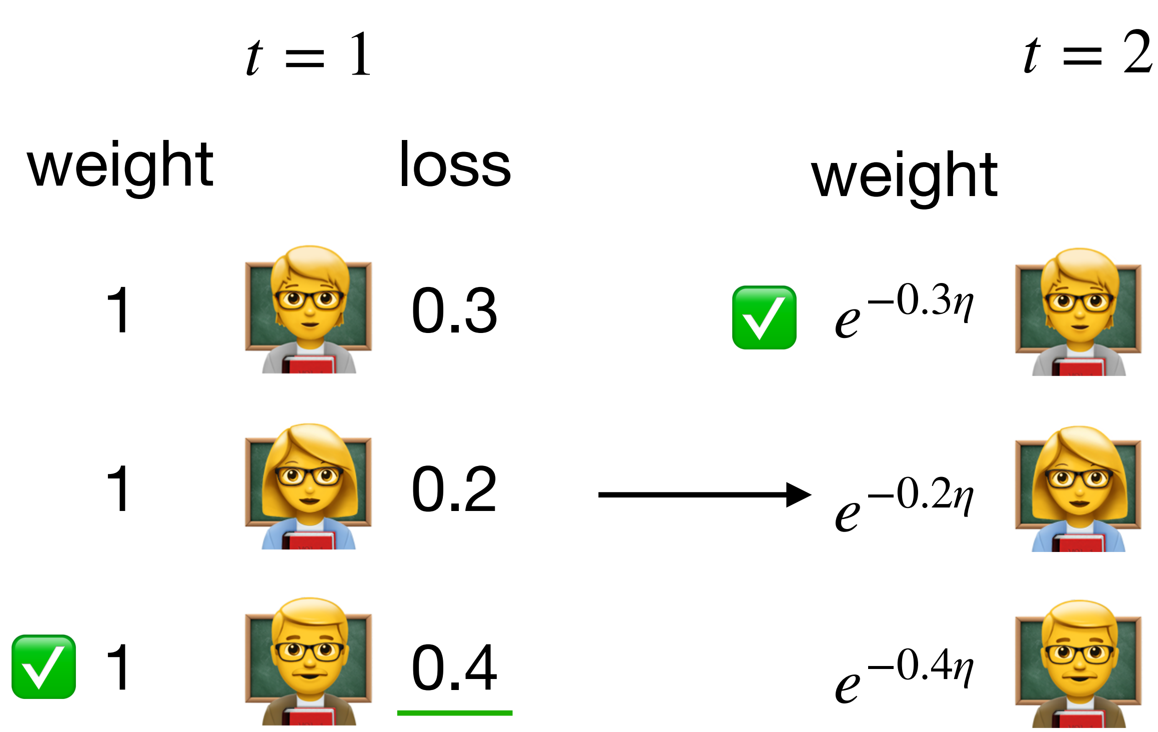
Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1,\ 2, \cdots,\ T$

Choose expert k with probability $\propto w_k(t)$

Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$



The Hedge Algorithm

> Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$
>
> For $t = 1, 2, \cdots, T$
>
>   Choose expert k with probability $\propto w_k(t)$
>
>   Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$
>
>   Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$ weight	$t = 1$ loss		$t = 2$ weight	$t = 2$ loss
1	0.3		✅ $e^{-0.3\eta}$	0.2
1	0.2	$\longrightarrow$	$e^{-0.2\eta}$	0.4
✅ 1	0.4		$e^{-0.4\eta}$	0.7

The Hedge Algorithm

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1, 2, \cdots, T$

    Choose expert k with probability $\propto w_k(t)$

    Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

    Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

$t = 1$

	weight		loss
	1	🧑‍🏫	0.3
	1	👩‍🏫	0.2
✅	1	👨‍🏫	<u>0.4</u>

$t = 2$

	weight		loss
✅	$e^{-0.3\eta}$	🧑‍🏫	<u>0.2</u>
	$e^{-0.2\eta}$	👩‍🏫	0.4
	$e^{-0.4\eta}$	👨‍🏫	0.7

$t = 3$

	weight		loss
✅	$e^{-0.5\eta}$	🧑‍🏫	<u>0.3</u>
	$e^{-0.6\eta}$	👩‍🏫	0.1
	$e^{-1.1\eta}$	👨‍🏫	0.3

Hedge for HI?

Initialize weights $w_k(1) = 1$ for $1 \leq k \leq K$

For $t = 1, 2, \cdots, T$

Choose expert k with probability $\propto w_k(t)$

Receive loss vector $[l_1(t), l_2(t), \cdots, l_K(t)]$

Update $w_k(t+1) = w_k(t)\exp(-\eta l_k(t))$

- Thresholds as experts
- Loss of expert k = loss incurred if the system chooses threshold k

Online Learning for HI

Motivation

Our Setting

Main Results

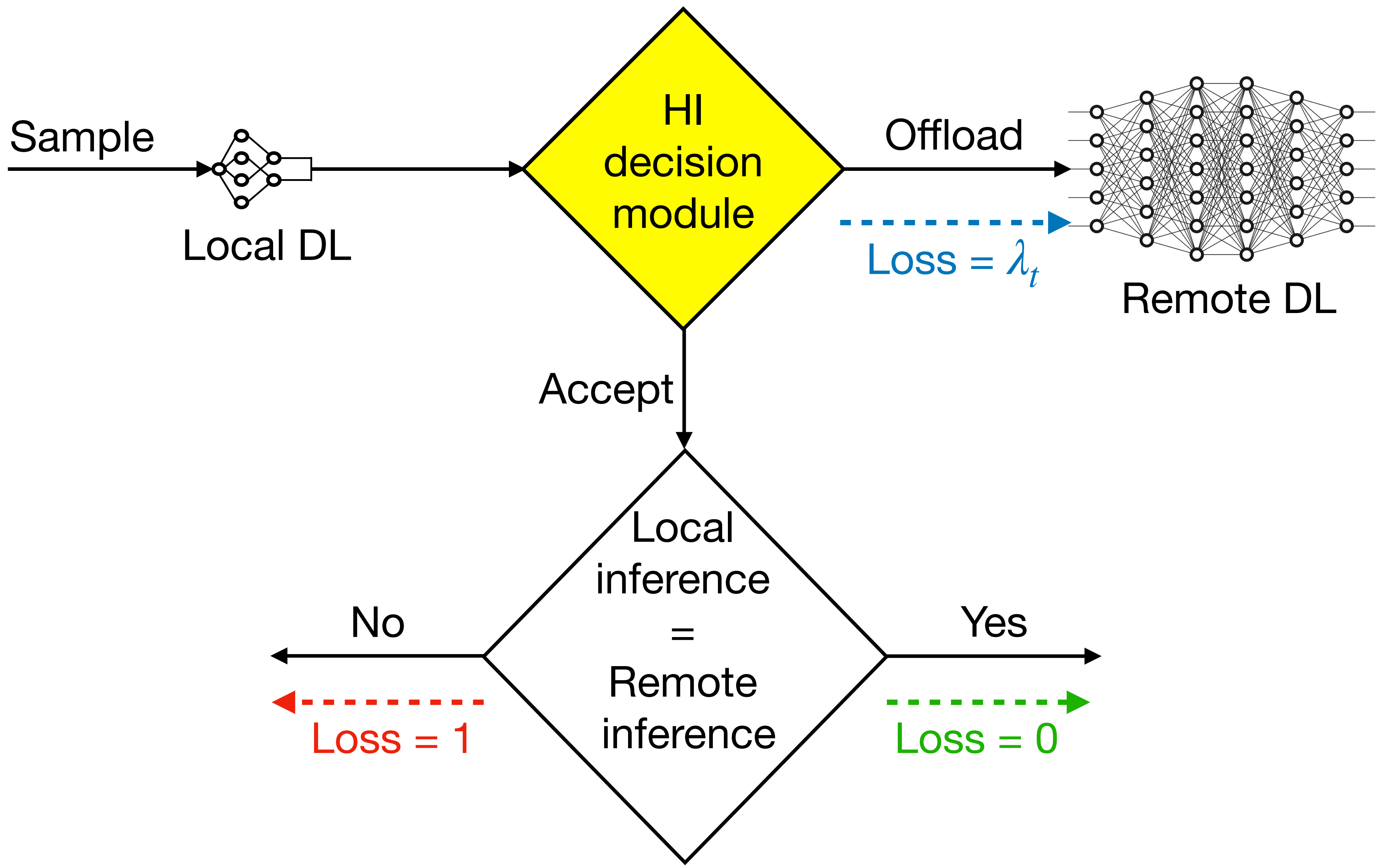
Background: Prediction with Experts

Our Algorithms and Guarantees

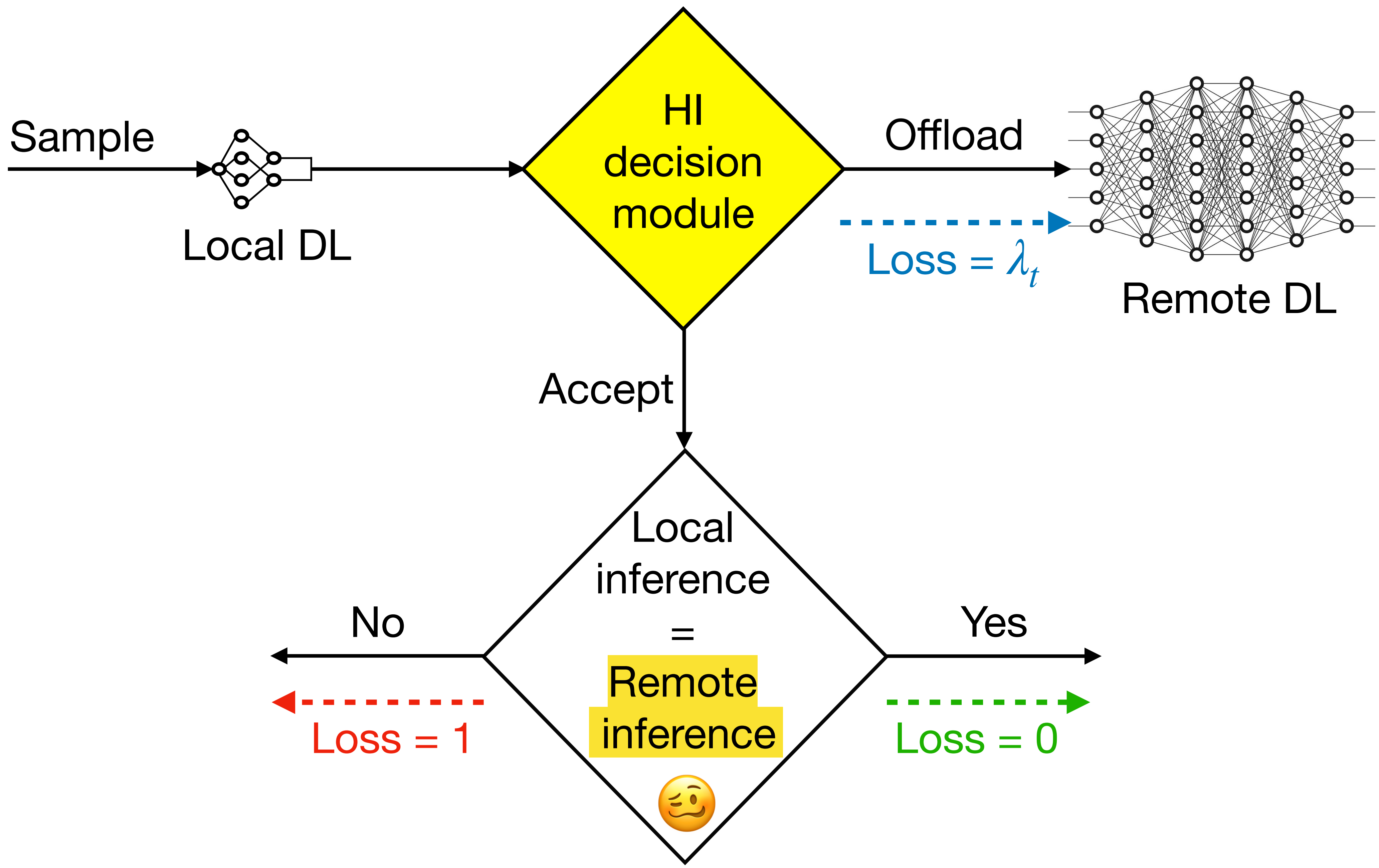
Numerical Results

Conclusions

Recall: Loss Model

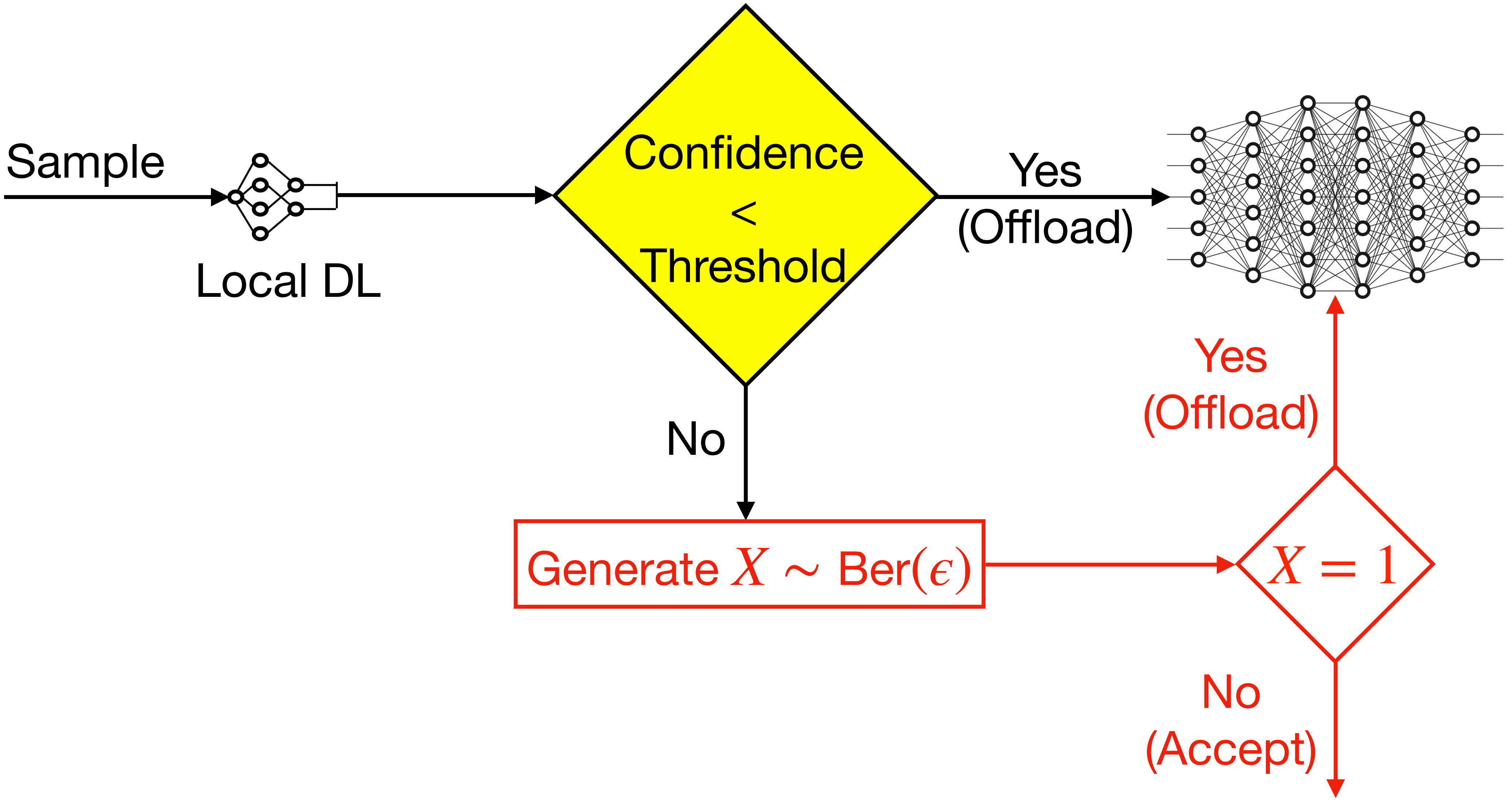


Hedge for HI?

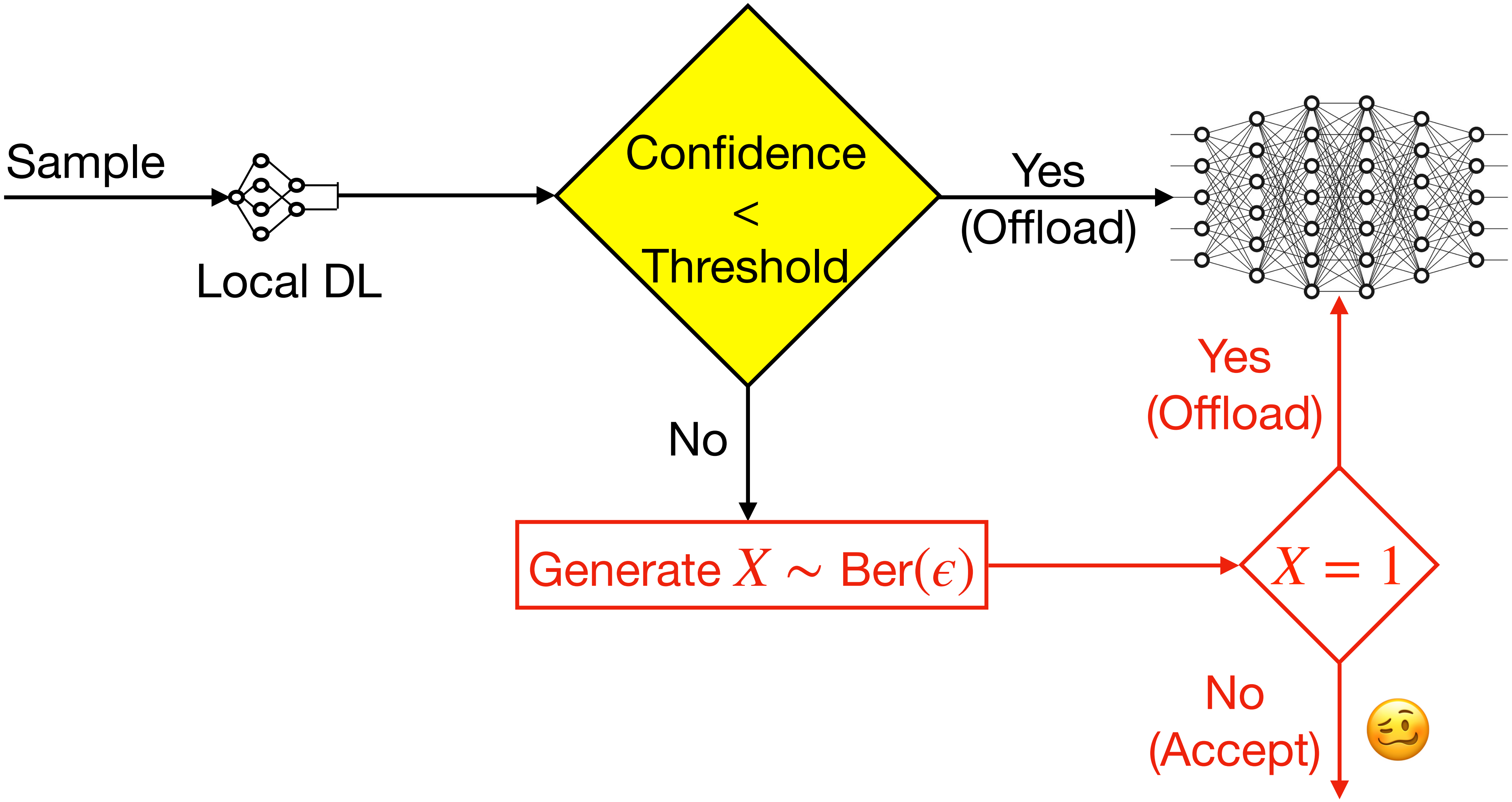


🥴 If we accept, loss vector only partially revealed

Tweak 1

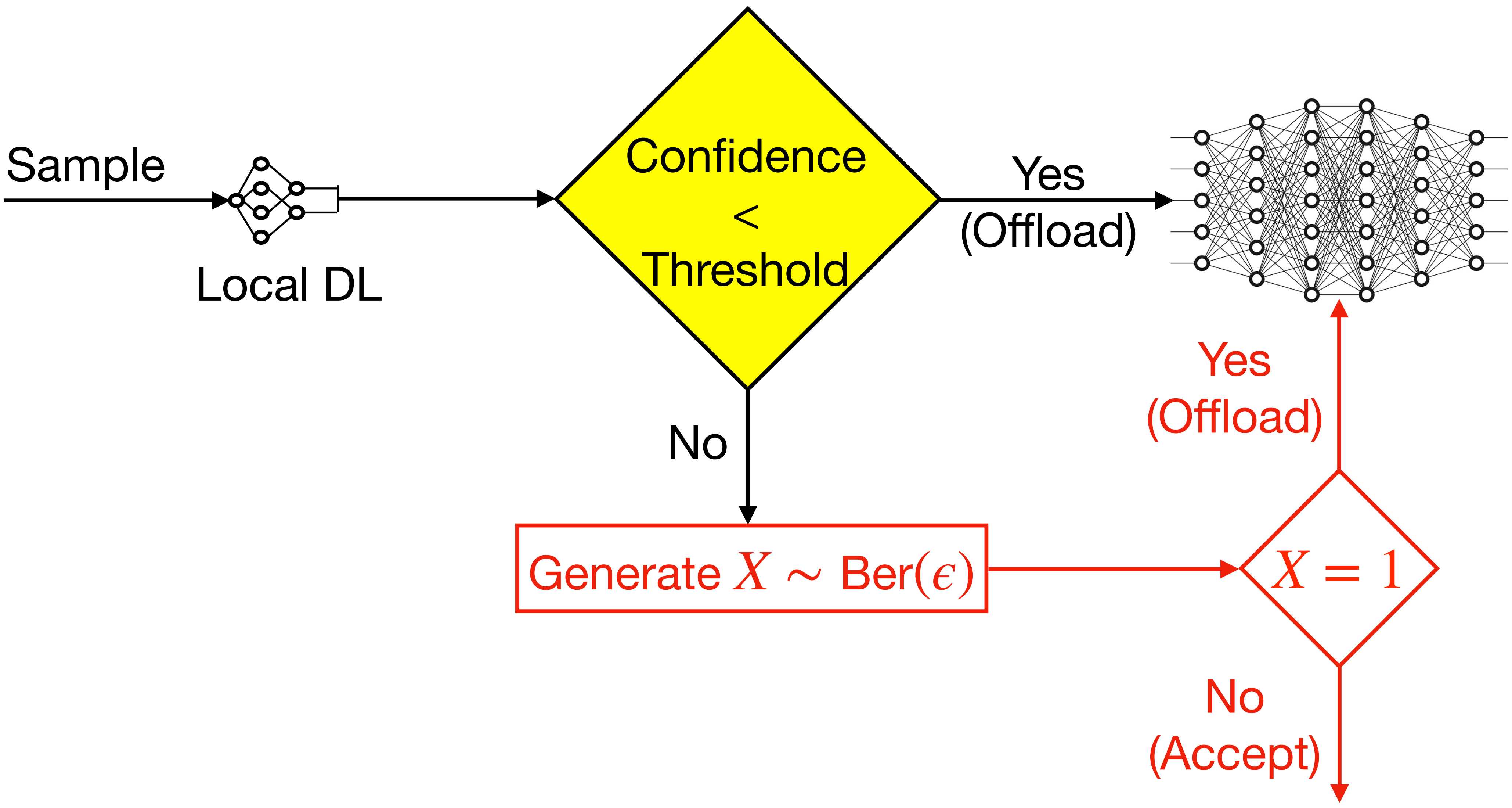

Sample
Local DL
Confidence
<
Threshold
Yes
(Offload)
No
Generate $X \sim \mathrm{Ber}(\epsilon)$
$X = 1$
Yes
(Offload)
No
(Accept)

Tweak 1

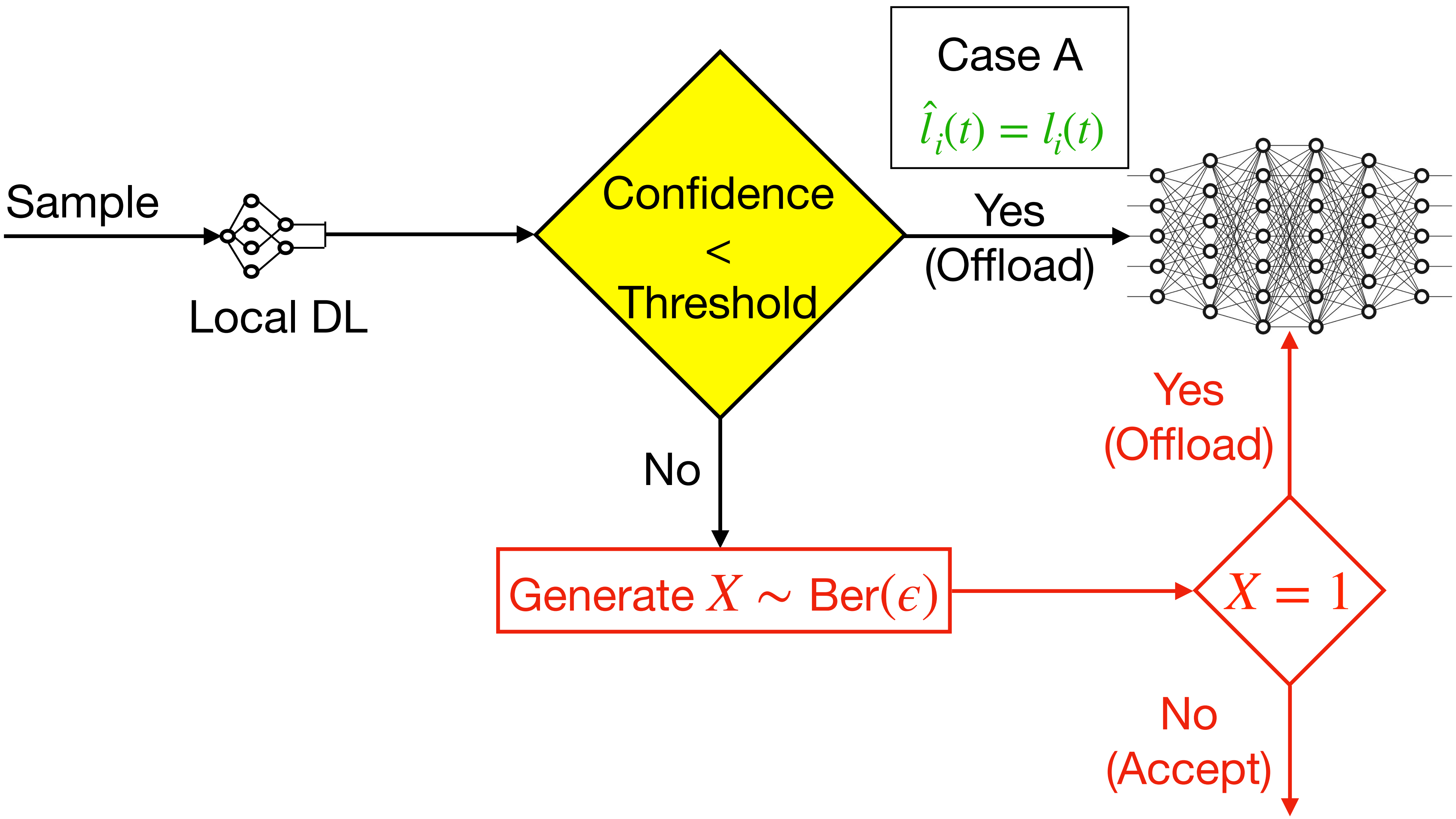


🥴 If we accept, loss vector only partially revealed

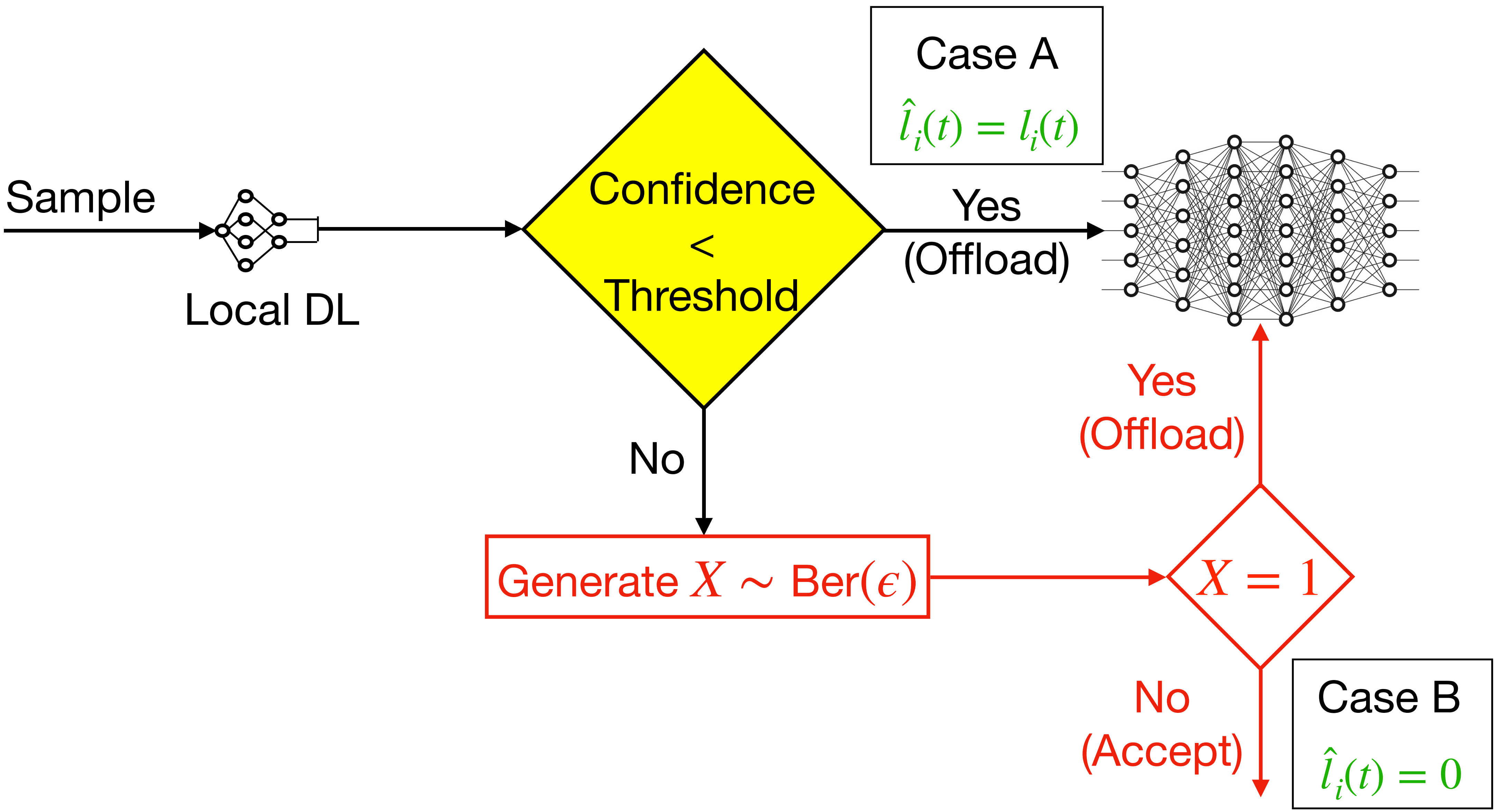
Tracking Loss



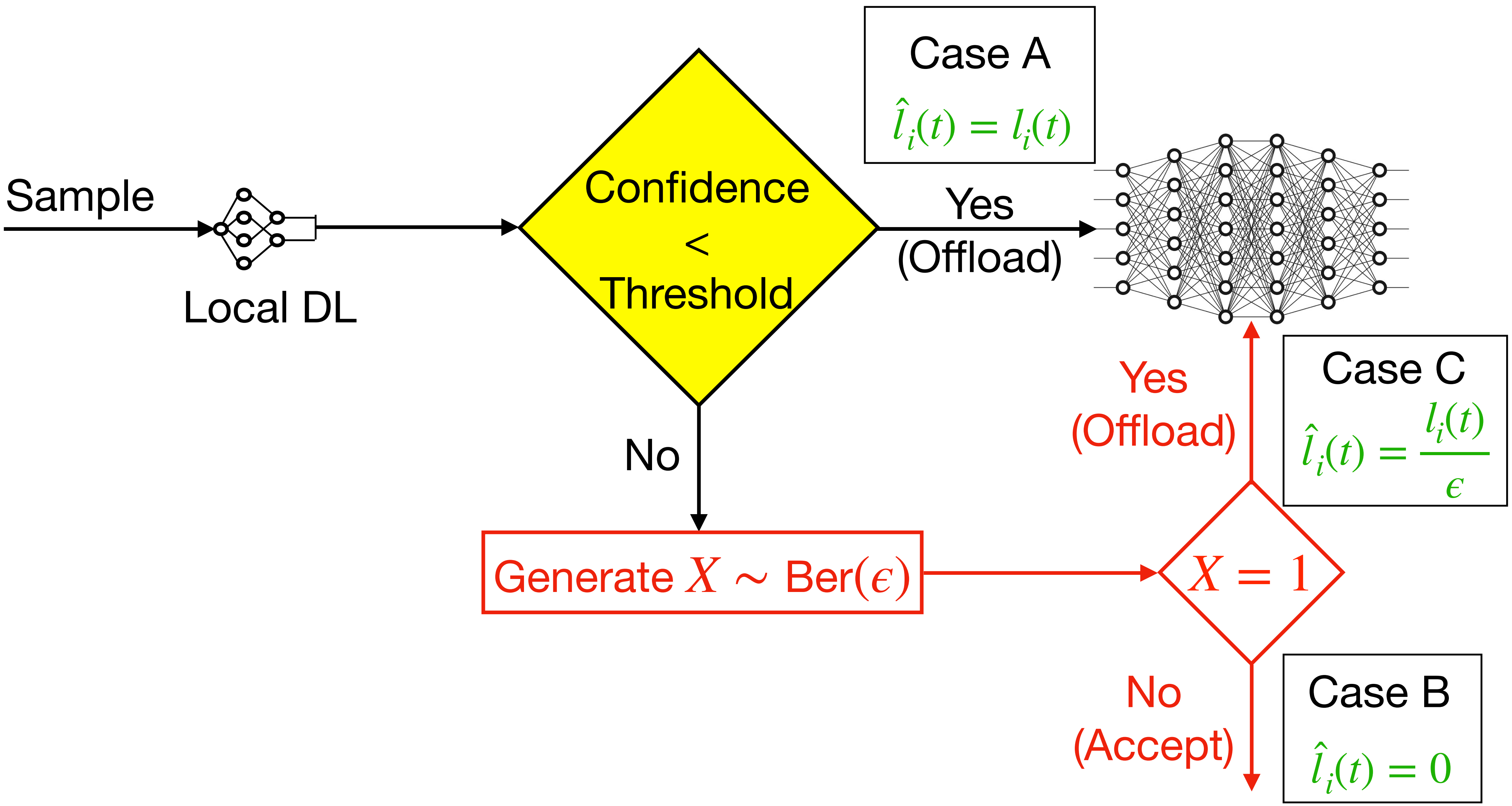
Tracking Loss



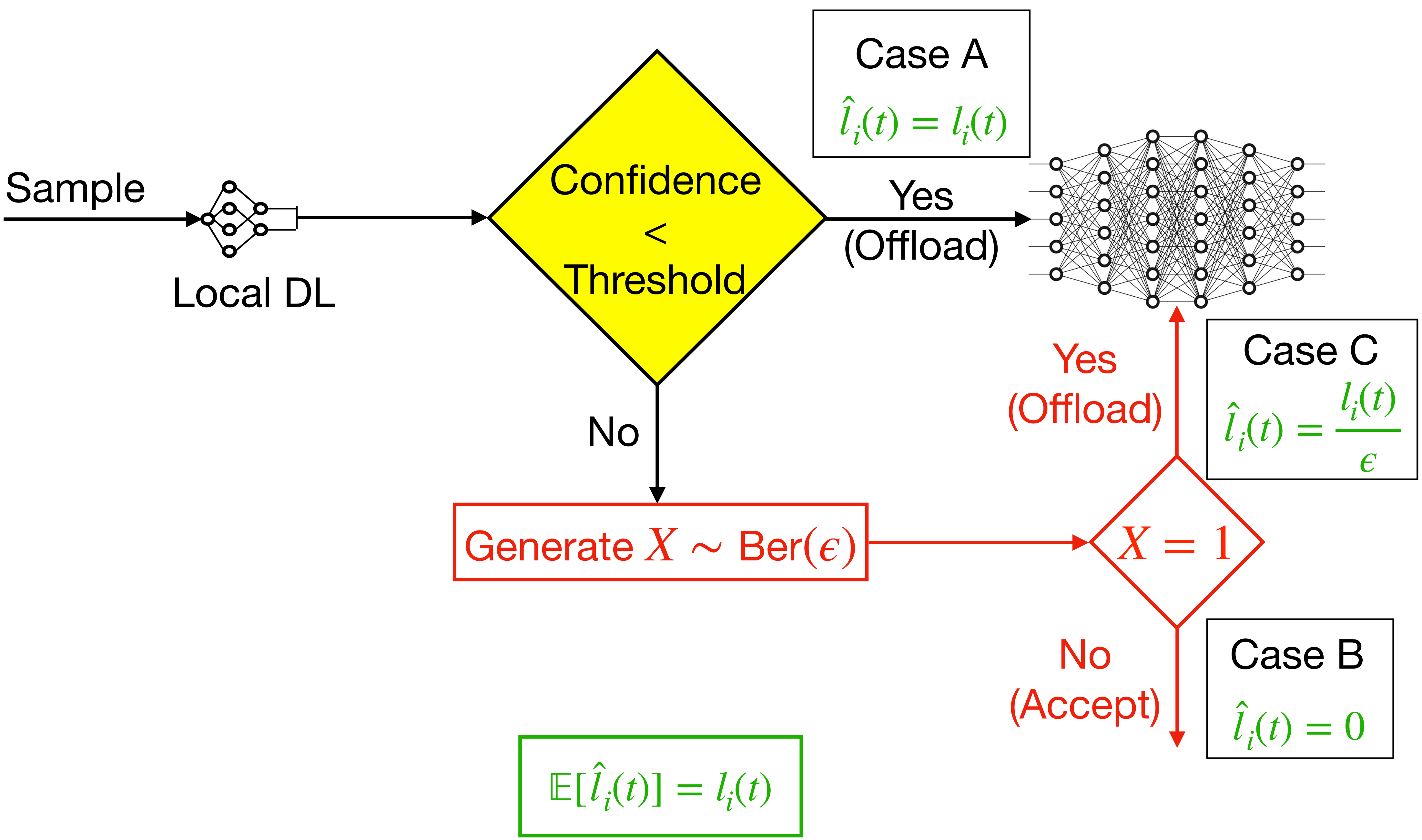
Tracking Loss



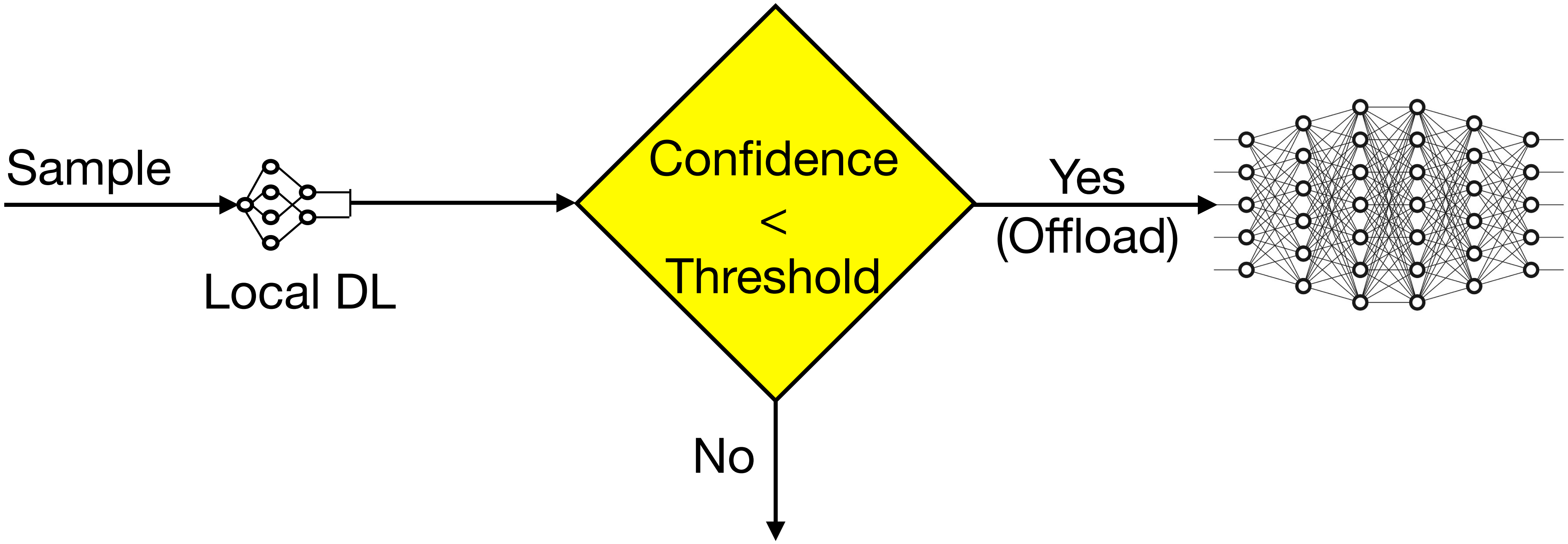
Tracking Loss



Tracking Loss



Hedge for HI?



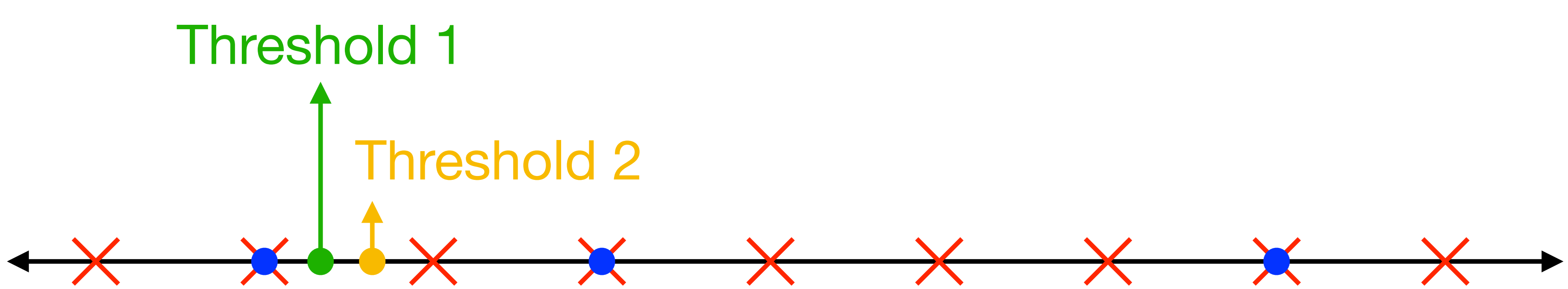
🥴 K may be very large (algorithm to be implemented on the ED)

Structural Properties of HI

Recall: In round *t*, if confidence < threshold, offload; else accept

: possible values of confidence metric

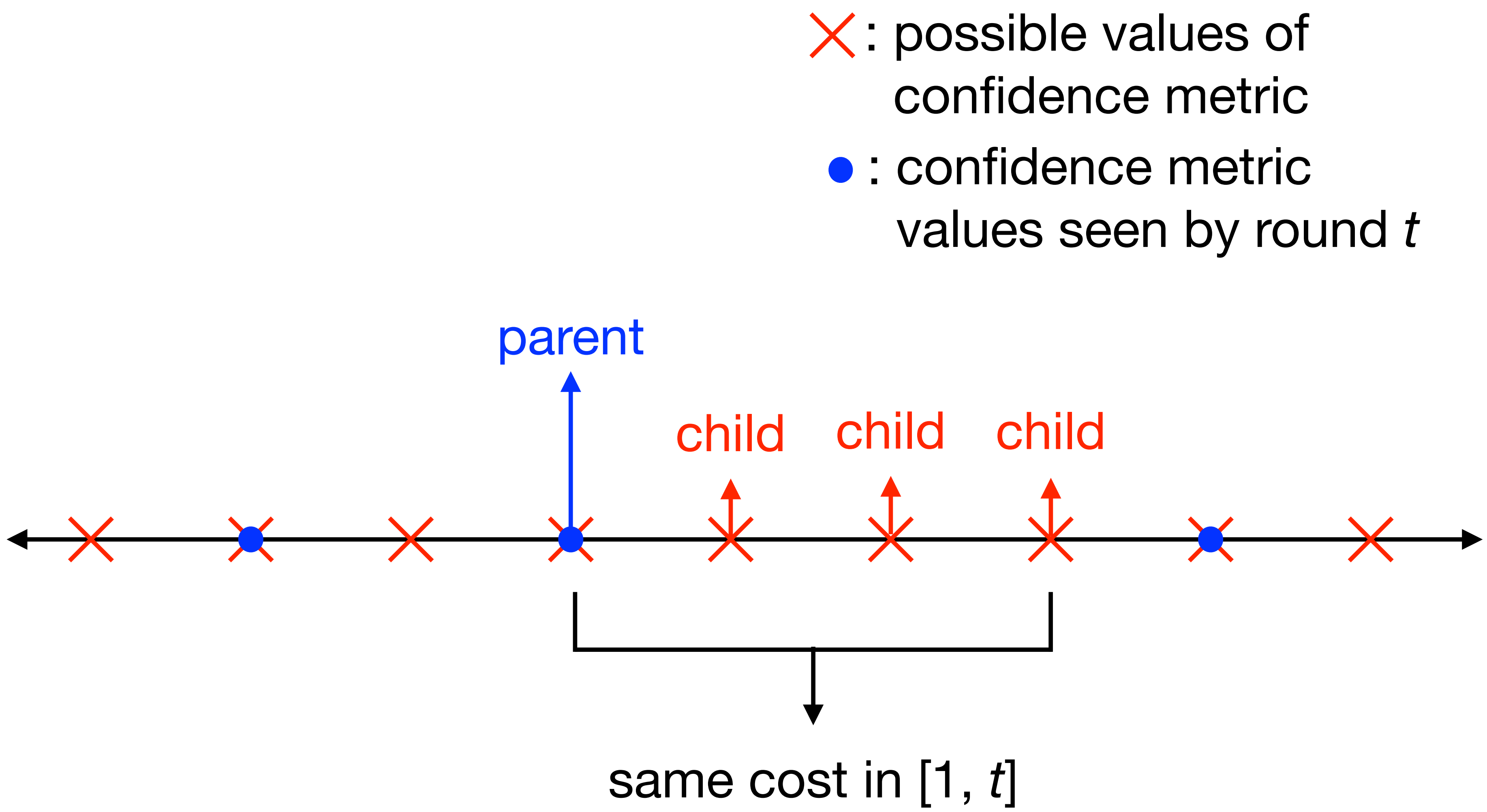
: confidence metric values seen by round *t*



- Both thresholds have the same sample path cost
- Can limit set of thresholds to a discrete set

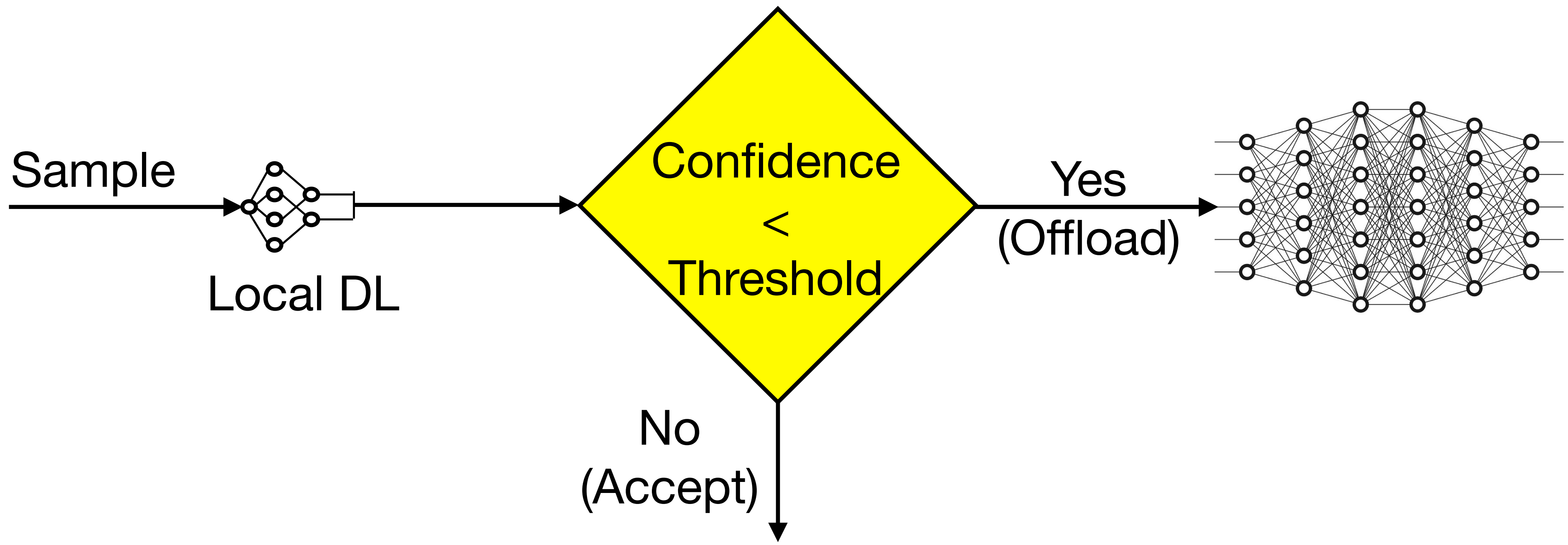
Structural Properties of HI

Recall: In round *t*, if confidence < threshold, offload; else accept



Tweak 2

🥴 K may be very large (algorithm to be implemented on the ED)



- Maintain a growing set of experts (thresholds)
- If the confidence value in round t is “new”, add to set of experts
- New expert inherits cumulative loss of its parent

Our Algorithm: Hedge-HI

1️⃣ Maintain a growing set of experts

- Set of experts = set of confidence values seen
- New expert inherits cumulative loss of its parent

2️⃣ Forced exploration with probability ϵ

- Compute $\hat{l}_i(t)$: an unbiased estimator of the loss

3️⃣ Use Hedge to choose an expert (threshold) using $\hat{l}_i(t)$s

Performance of Hedge-HI

- Recall input parameters:
  - Learning rate η
  - Forced exploration probability ϵ
- N_T: number of experts at the end of round T

Theorem (sub-linear regret)

For $\eta = \left(\frac{\mathbb{E}[N_T]}{T}\right)^{\frac{2}{3}}$ and $\epsilon = \sqrt{\frac{\eta}{2}}$, Hedge-HI has $O\left(T^{\frac{2}{3}}(\mathbb{E}[N_T])^{\frac{1}{3}}\right)$ regret.

Performance of Hedge-HI

- Recall input parameters:
  - Learning rate η
  - Forced exploration probability ϵ
- N_T: number of experts at the end of round T

Theorem (sub-linear regret)

For $\eta = \left(\frac{\mathbb{E}[N_T]}{T}\right)^{\frac{2}{3}}$ and $\epsilon = \sqrt{\frac{\eta}{2}}$, Hedge-HI has $O\left(T^{\frac{2}{3}}(\mathbb{E}[N_T])^{\frac{1}{3}}\right)$ regret.

upper bounded by a constant for our setting

Performance of Hedge-HI

- Recall input parameters:
  - Learning rate η
  - Forced exploration probability ϵ
- N_T: number of experts at the end of round T

not known

Theorem (sub-linear regret)

For $\eta = \left(\frac{\mathbb{E}[N_T]}{T}\right)^{\frac{2}{3}}$ and $\epsilon = \sqrt{\frac{\eta}{2}}$, Hedge-HI has $O\left(T^{\frac{2}{3}}(\mathbb{E}[N_T])^{\frac{1}{3}}\right)$ regret.

upper bounded by a constant for our setting

Limitations of Hedge-HI

1️⃣ Maintain a growing set of experts

- Set of experts = set of confidence values seen
- New expert inherits cumulative loss of its parent

2️⃣ Forced exploration with probability ϵ

- Compute $\hat{l}_i(t)$: an unbiased estimator of the loss

3️⃣ Use Hedge to choose an expert (threshold) using $\hat{l}_i(t)$s

Limitations of Hedge-HI

1️⃣ Maintain a growing set of experts
- Set of experts = set of confidence values seen
- New expert inherits cumulative loss of its parent

2️⃣ Forced exploration with probability ϵ
- Compute $\hat{l}_i(t)$: an unbiased estimator of the loss

3️⃣ Use Hedge to choose an expert (threshold) using $\hat{l}_i(t)$s

💡 Freeze the set of experts after round $\tau(< T)$.

Hedge-HI-Restart

Input: τ, η (learning rate), ϵ (forced exploration probability)

For $t \leq \tau$, use Hedge-HI

Freeze set of experts after round τ, reset all weights to 1

For $t > \tau$, use Hedge-HI on the frozen set of experts

τ = 🤔

Large τ: large number of experts, increase in complexity

Small τ: may miss the optimal expert, increase in regret

Hedge-HI-Restart

- Recall input parameters:
  - τ
  - Learning rate η
  - Forced exploration probability ϵ
- N_t: number of experts at the end of round t
- Assumption: $\mathbb{P}(\text{confidence}(t) = \text{optimal threshold}) = \nu \in (0,1]$

Theorem (sub-linear regret)

$\exists \tau,\ \eta,$ and ϵ, such that Hedge-HI-Restart has $O\left(T^{\frac{2}{3}}(\mathbb{E}[\log N_T])^{\frac{1}{3}}\right)$ regret.

Summary

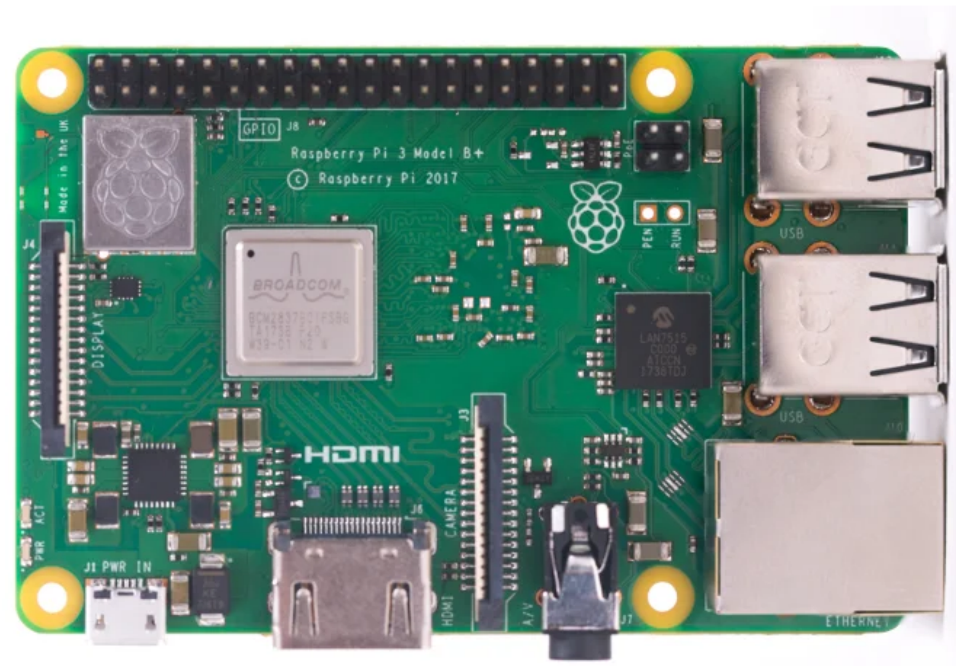
	Hedge-HI	**Hedge-HI-Restart**
Expert Set	Can continue growing	Stops growing after round τ
Regret	$O\left(T^{\frac{2}{3}}(\mathbb{E}[N_T])^{\frac{1}{3}}\right)$	$O\left(T^{\frac{2}{3}}(\mathbb{E}[\log N_T])^{\frac{1}{3}}\right)$

Limitations

- Guarantee for Hedge-HI-Restart holds only under an assumption
- Need to know $\mathbb{E}[N_T]$

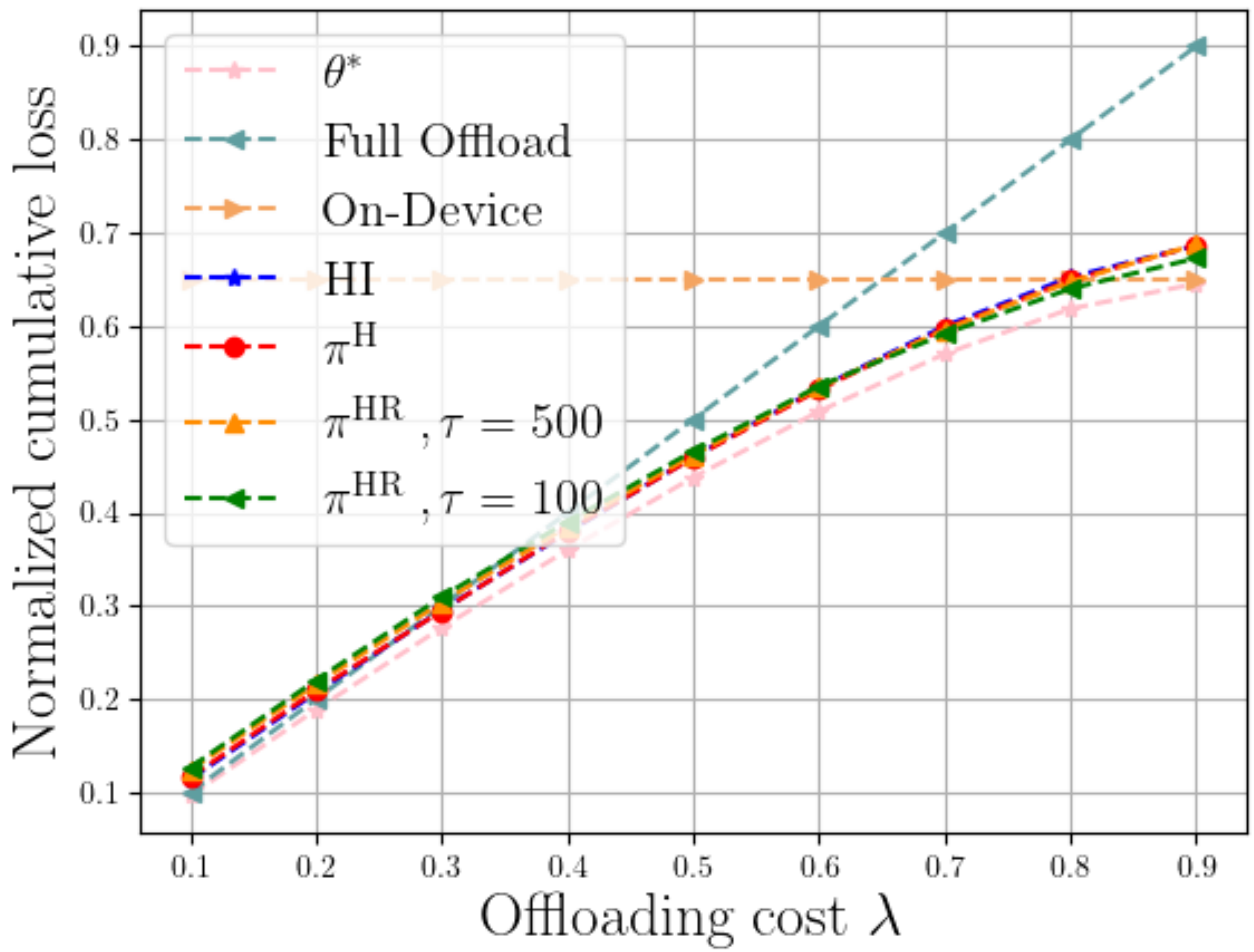
Online Learning for HI

Simulation Framework

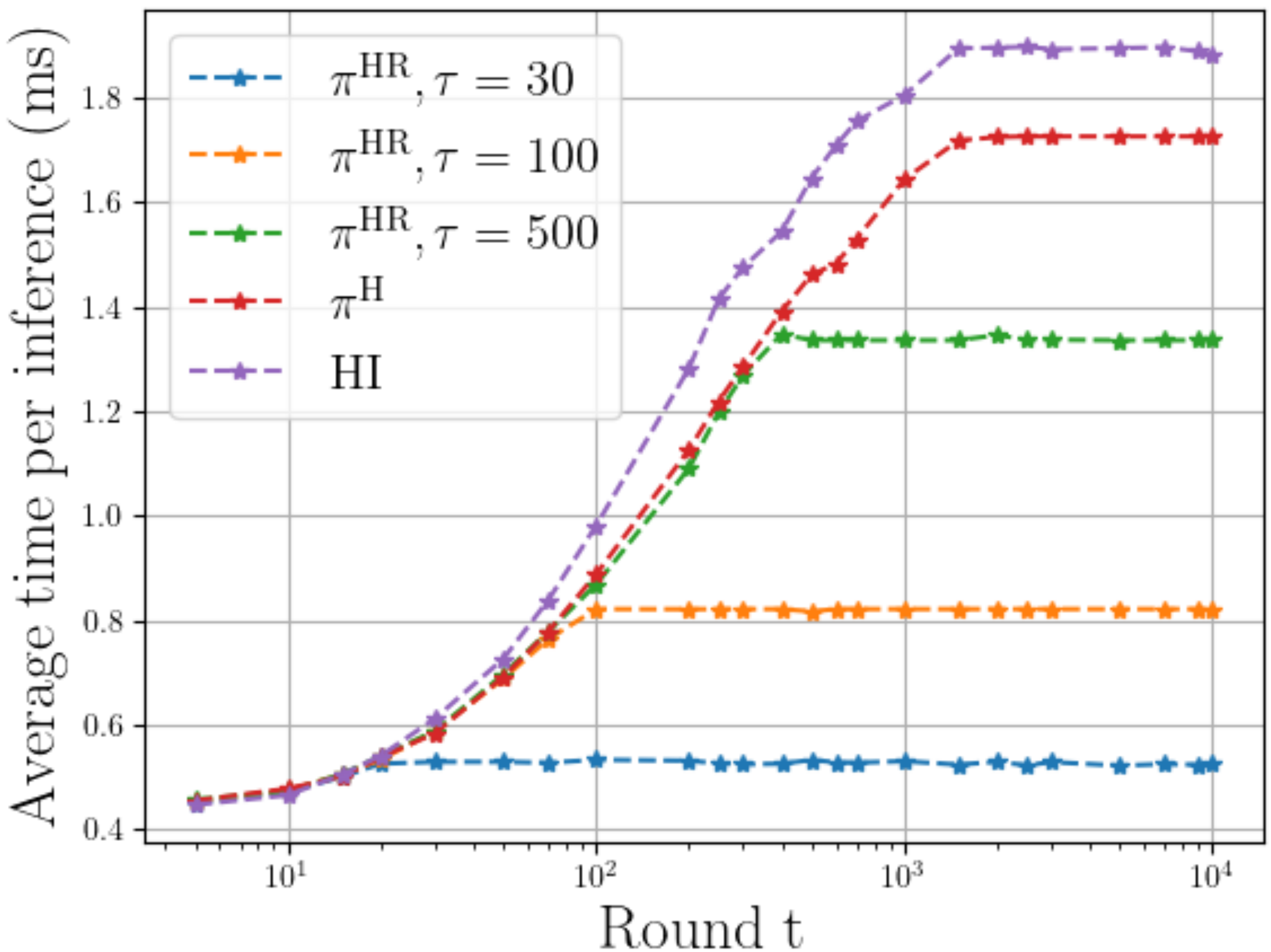


- Imagenet dataset (1000 classes, 50000 samples)
- S-ML: 8-bit quantized MobileNet tflite model
  - width parameter 0.25
  - size 500 KB
  - accuracy 35%
  - confidence 8-bit (256 unique values)
- T = 10000
- Use $\mathbb{E}[N_T]$ = 256

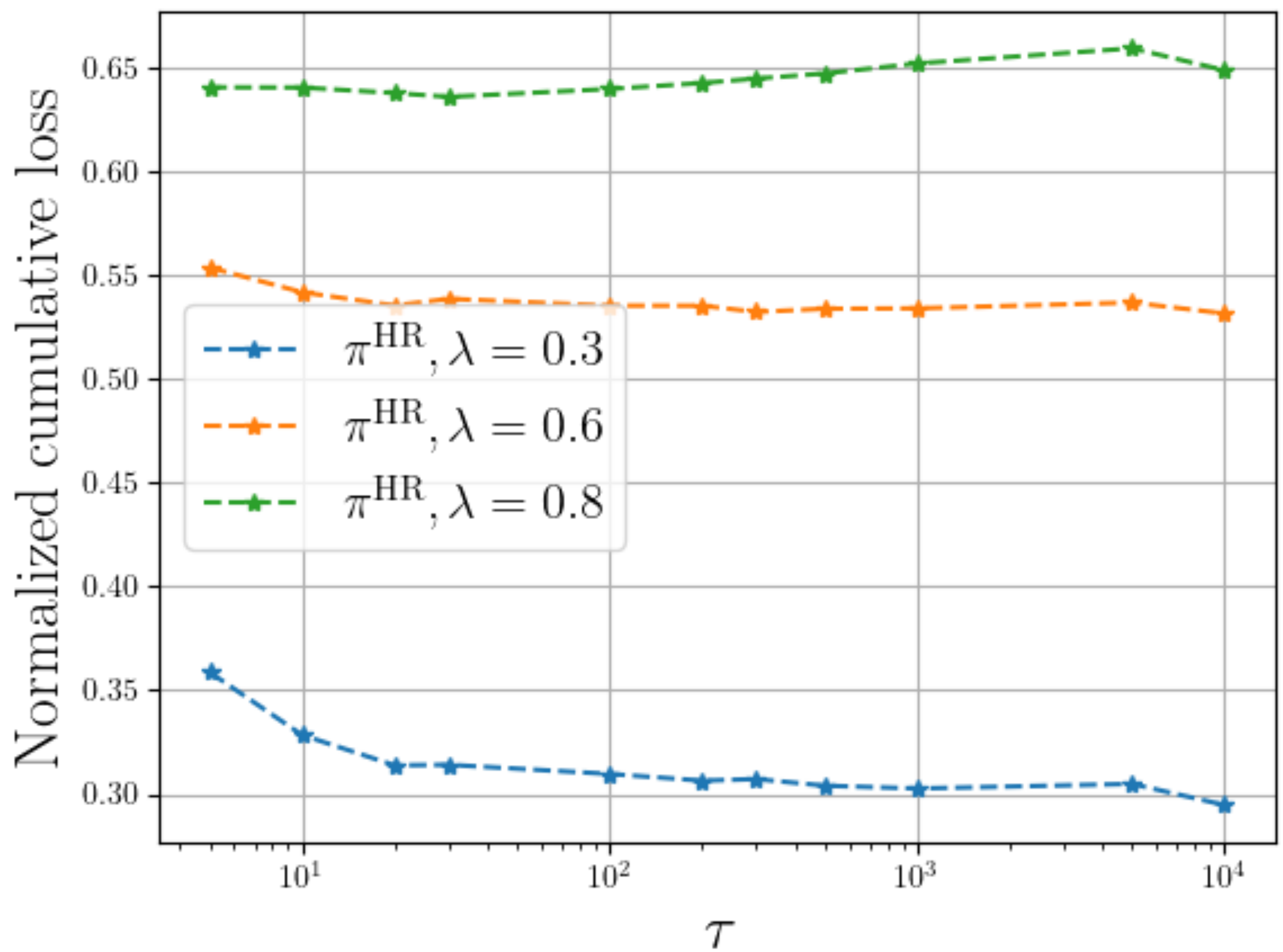
Cumulative Loss



Average Runtime



Hedge-HI Restart(τ)



Online Learning for HI

Motivation

Our Setting

Main Results

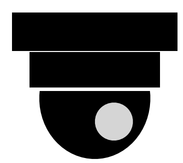
Background: Prediction with Experts

Our Algorithms and Guarantees

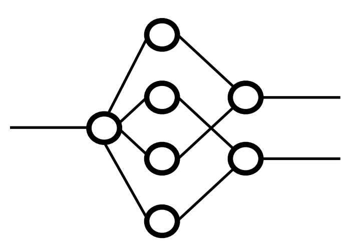
Numerical Results

Conclusions

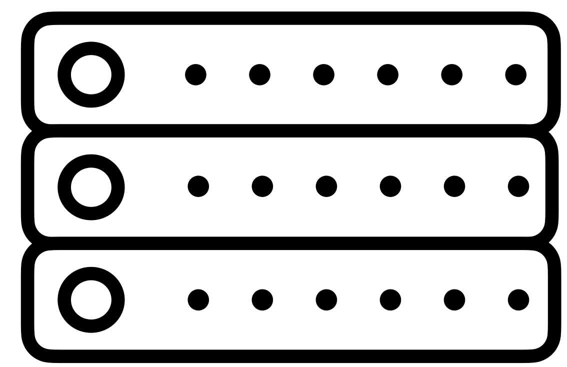
System Components



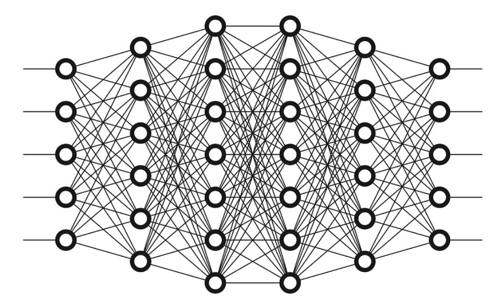
End-Device



Local DL



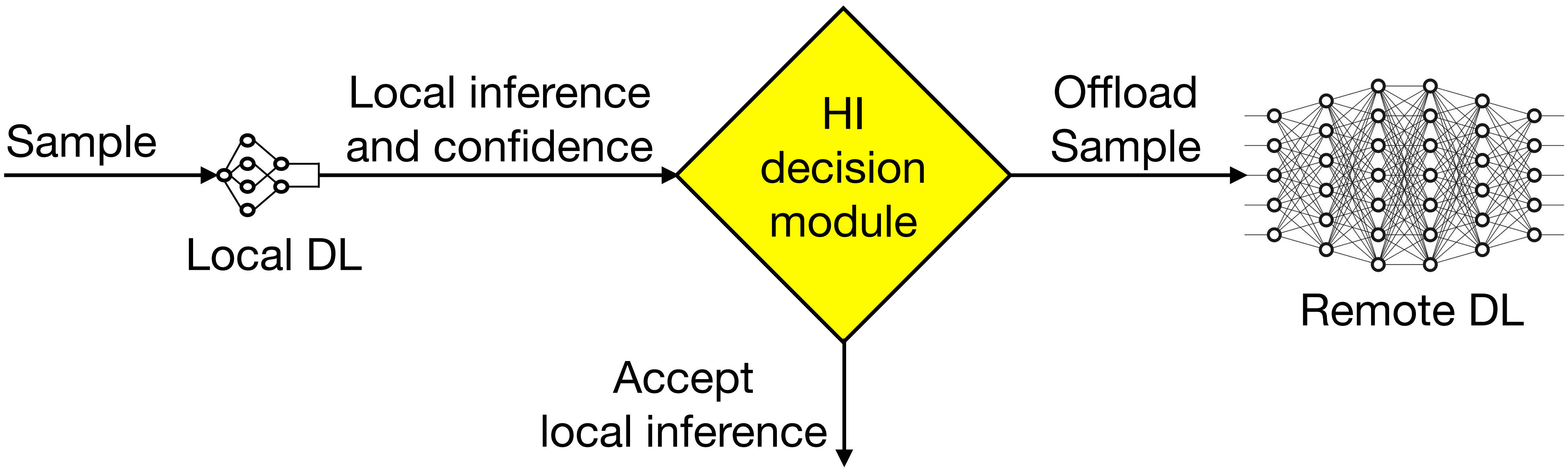
Edge-Server



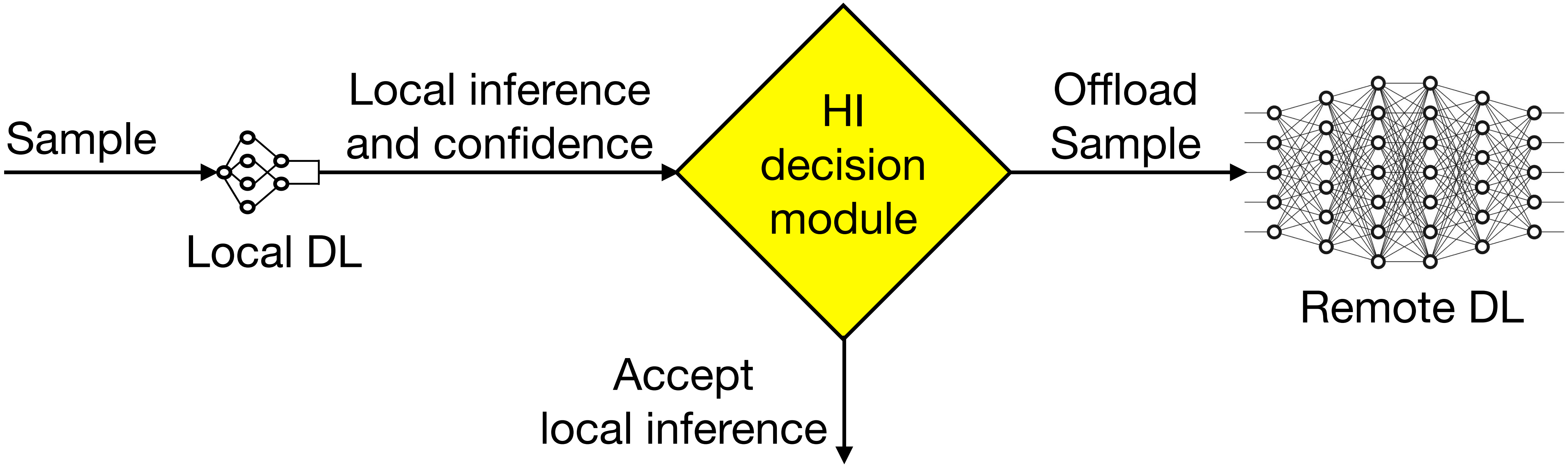
Remote DL

Where should we do the inference?

Hierarchical Inference



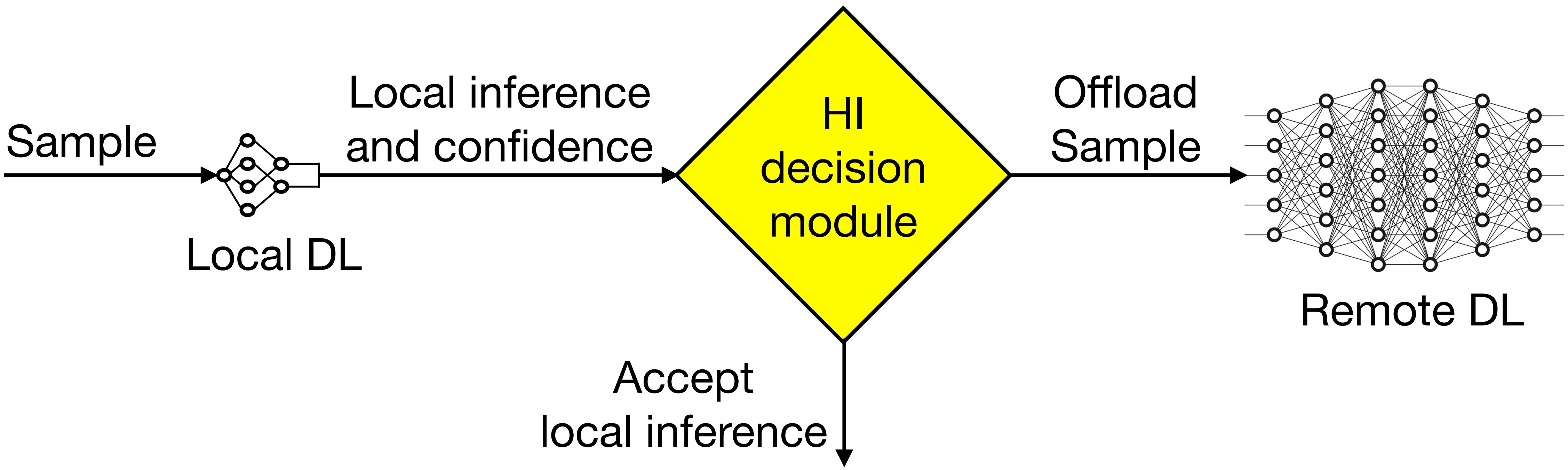
Hierarchical Inference



💡 Offload if confidence is low, i.e., below a threshold

🤔 What should be the threshold?

Our Contributions



💡 Offload if confidence is low, i.e., below a threshold

🤔 What should be the threshold?

🥳 Variants of Hedge to choose threshold with sub-linear regret

Prediction with Experts

1. Branching Experts (Gofer et al., *COLT* 2013)
    - New experts revealed over time
    - N_T finite
    - Cumulative loss of new expert close to that of an existing expert
    - Algorithm with $O\left(\sqrt{TN_T}\right)$ regret

2. Lifelong Learning with Branching Experts (Wu et al., *ACML* 2021)
    - New experts revealed over time
    - N_T finite
    - Adversarial and stochastic losses

Hedge-HI minus forced exploration is order-optimal

Online Learning for Hierarchical Inference*

Thanks!

*to appear in *ACM MobiHoc* 2024