Energy-Free Learning
for Lifelong Embedded Intelligence

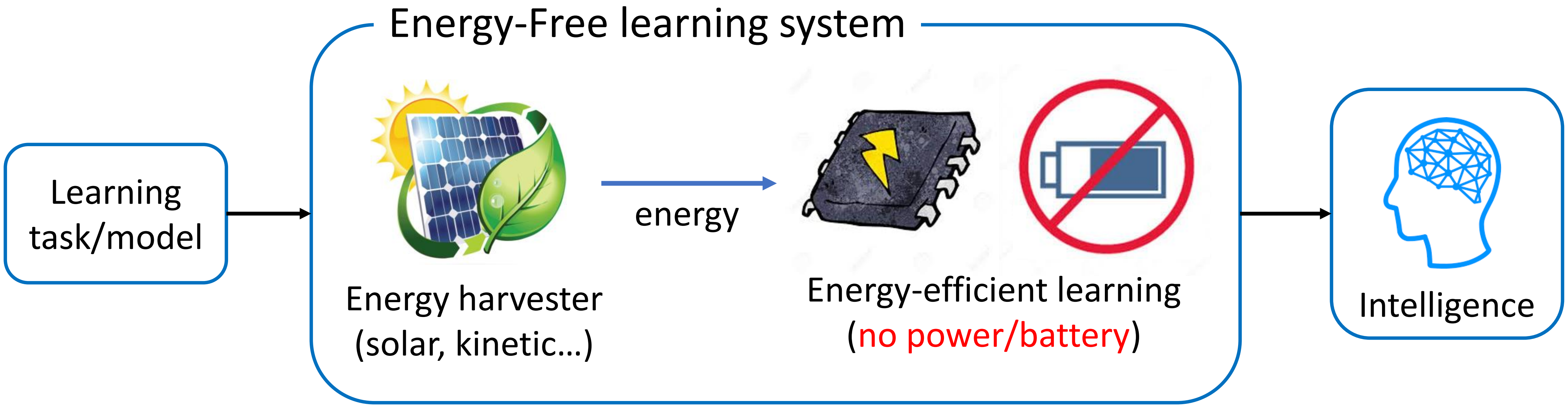
Seulki Lee

Smart and Connected Systems Group

UNC Chapel Hill

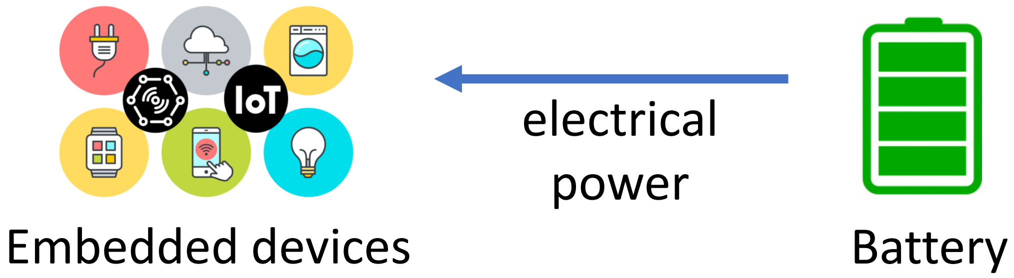
What am I trying to do?

- Create a lifelong learning system using harvested energy for embedded intelligence.
  - It keeps learning and improving its intelligence over time in its lifetime.
  - The learning task can be updated, changed or evolved.

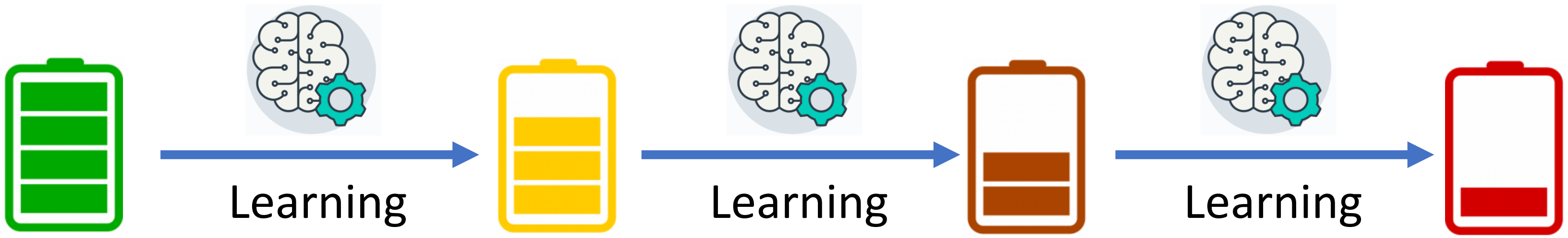


Motivation

- Mobile devices have limited power (battery).
  - At present, they almost all rely on some kind of battery that eventually runs down.



- "Machine Learning (ML)" requires a large amount of power.
  - It drains a battery quickly.



Energy harvesting

- A device able to generate power could, in principle, operate forever.
  - Need to run in their lifetime.
  - Once deployed, inaccessible to change or recharge a battery.

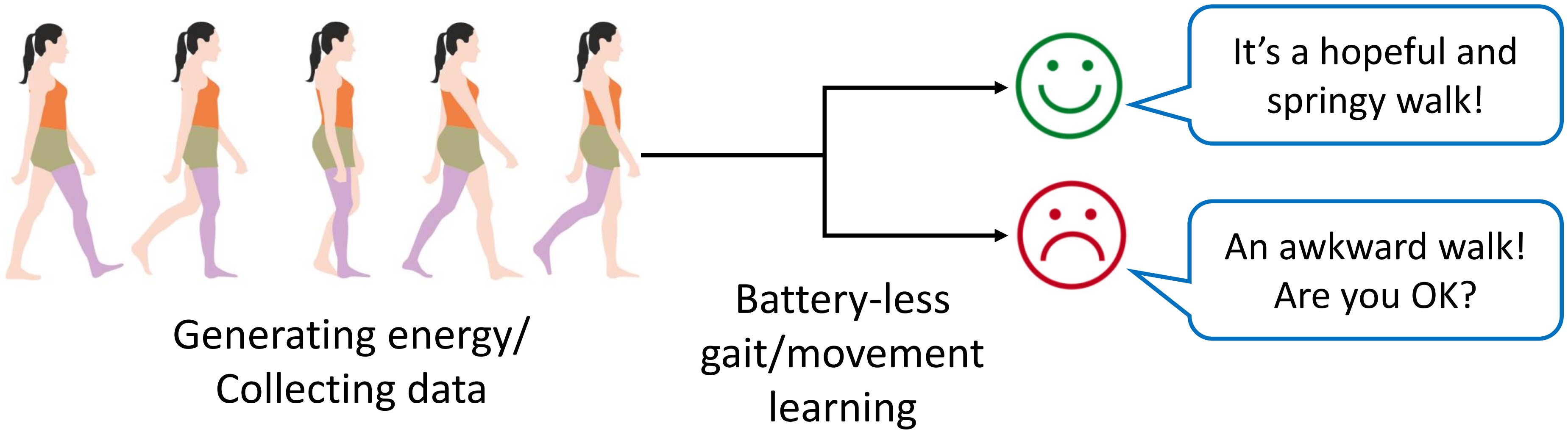
Implantable medical devices

Wildlife monitoring

Remote sensing

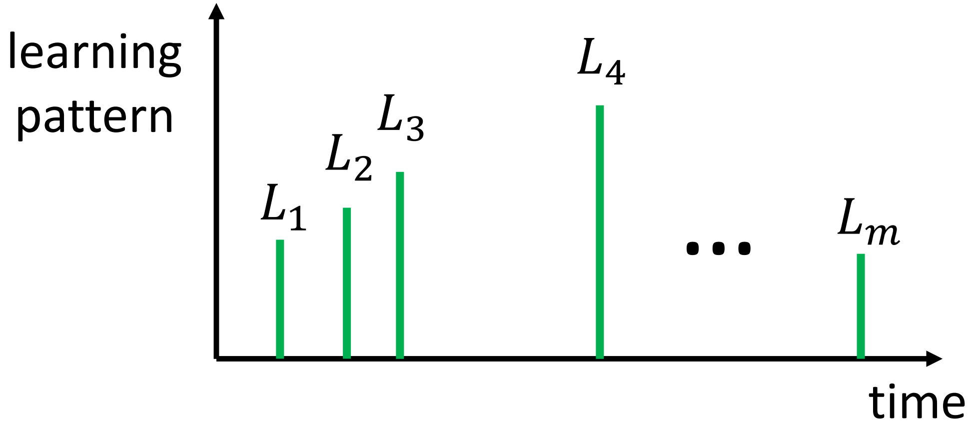
Example: Energy-harvesting + learning ability

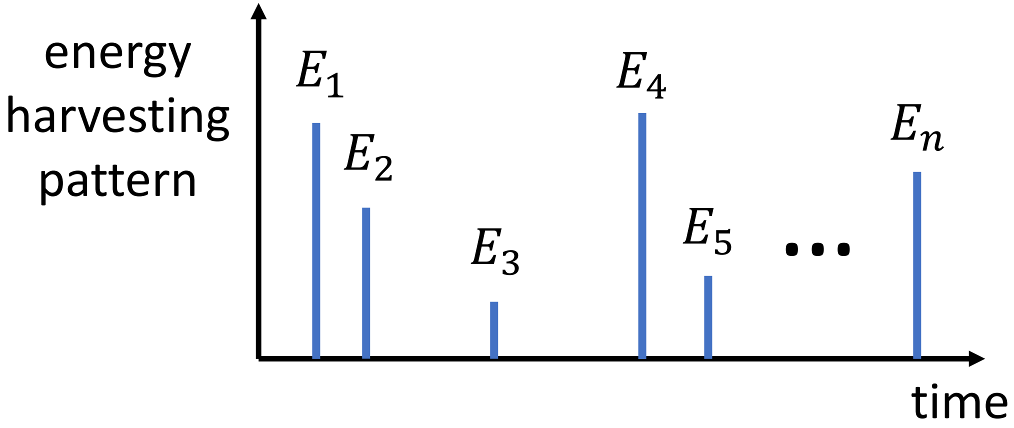
- An energy-free learning system in shoes
  - A piezoelectric harvester generates energy for every step.
  - Not only harvesting energy but also learning a walking pattern.
  - Detect abnormal gait or unusual movement of a user.



Energy harvesting and learning

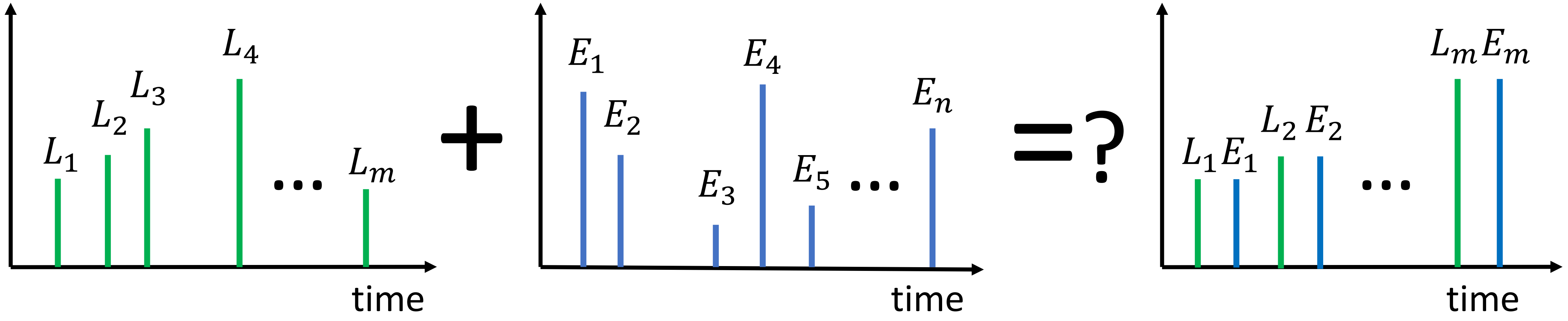
- ***Observation 1***: Learning does not happen all the time. Systems learn intermittently in its lifetime.
- ***Example***: 1) learning examples come unpredictably and some are useless to learn, 2) a learning goal is already met.
- ***Observation 2***: Energy harvesters generate lifelong energy in an intermittent manner.
- ***Example***: 1) sunny/rainy day for a solar panel, 2) slow or no movement for a human-kinetic harvester





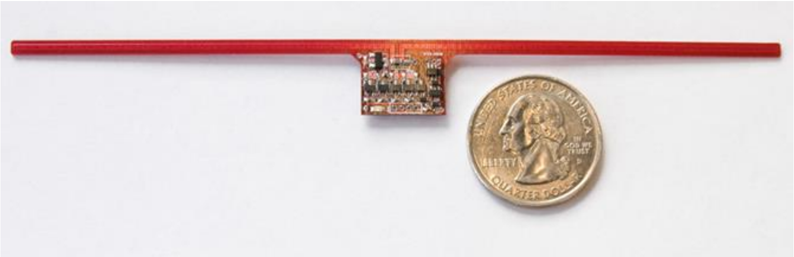
A pipedream

- ***Idea***: Can we leverage intermittently-harvested energy for power-constrained systems, especially for lifelong learning which is also performed intermittently?
  - Can we match learning and energy pattern intelligently?
    - ***Example***: skip a less-important learning example based on energy.
  - If not, what is the best way of doing it?



How does it get done today?

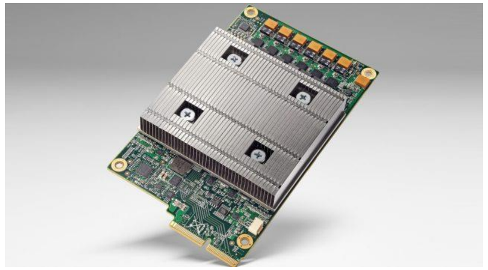
- State-of-the-art energy harvesting systems
  - Wireless Identification Sensing Platform
  - Piezoelectric step counter



- Limitations
  - No learning ability: most are simple sensing/computing platforms.
  - Short-term computation: immediate-results focused.
  - No estimation of execution time.

How does it get done today?

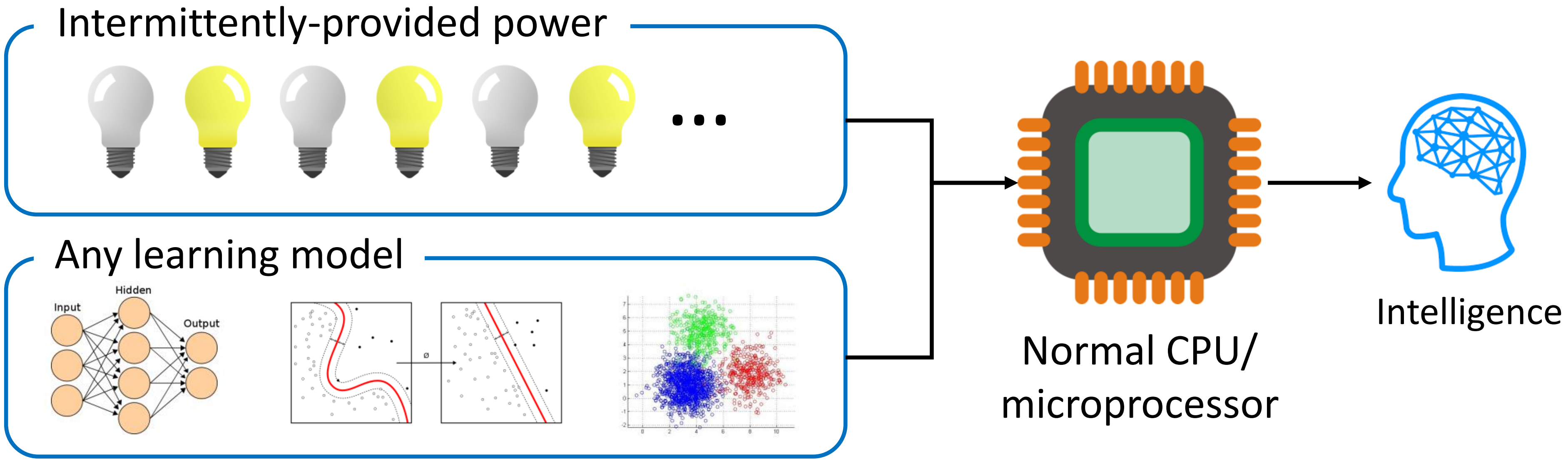
- State-of-the-art embedded machine learning
  - Embedded GPU
  - Tensor Processing Unit (TPU)
  - Special-purpose Unit (VPU)



- Limitations
  - Embedded machine learning usually rely on special hardware.
  - They are not available for all embedded systems.
  - GPU: expensive, TPU: hard to get, VPU: no general-purpose.
  - Without them, an embedded system can hardly learn by itself.

What is new about your approach?

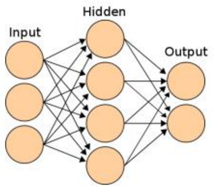
- Designing of '*Intermittent learning model*'
  - Perform a learning task using intermittently-harvested energy.
  - No restriction on learning task/algorithm.
  - No learning-purpose hardware (no GPU, no TPU): It runs on a general-purpose computing unit like CPU or microprocessor.



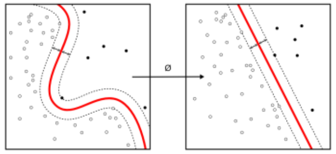
What is new about your approach?

- Providing an expected learning performance
  - Looking at whether a learning task is learnable with harvested energy.
  - If learnable, provide a reasonable estimation of expected learning performance.

Learning model A



Learning model B



Is it learnable?

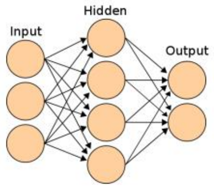
Yes, it is expected to complete its learning at time $\boldsymbol{t}$ with $\boldsymbol{X}$**%** of learning accuracy.

No, it requires exponential computation that cannot be performed with harvested energy.

What is new about your approach?

- Fitting a learning task into resource-constrained condition
  - Harvested energy + small memory + low-computational capacity.
  - Finding an energy-efficient/lightweight way of performing a large learning task/model.
  - Should not degrade learning performance.

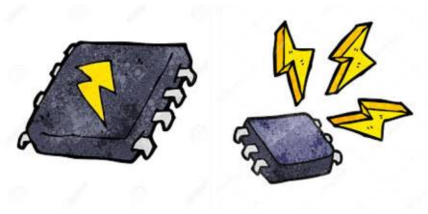
Learning model A



requires huge memory/computation/energy

accelerated learning/
lightweight learning

Energy-Free learning system



small memory/
slow computing unit/
harvested-energy

What difference it will make?

- Battery-less lifelong systems will keep learning persistently.
  - Millions of embedded devices with limited power-supply will be able to learn.
  - Systems will improve its intelligence over time by lifelong learning.
- Learning will be performed on the spot, not in a remote cloud system.
  - Issues caused by learning in a remote system like security, privacy or communication will be solved.
  - Intelligent IoT environment can be built locally.
- Dumb systems will turn into smart ones
  - A dumb system will become smart by having the ability of learning if an energy-free learning component is added to it.
  - No additional energy/overhead required to the system.

Problem Statement

- Perform any learning model/task L in a sustainable/persistent manner given intermittently-provided energy E which achieves comparable learning performance within an expected completion/execution time with reasonable certainty P.

Problem Statement

- Perform any learning model/task L in a sustainable/persistent manner given intermittently-provided energy E which achieves comparable learning performance within an expected completion/execution time with reasonable certainty P.

1) Intermittent learning

2) Achieving comparable learning performance

3) Providing an expected completion time

Problem 1) and 3)

- Perform any learning model/task L in a sustainable/persistent manner given intermittently-provided energy E which achieves comparable learning performance within an expected completion/execution time with reasonable certainty P.

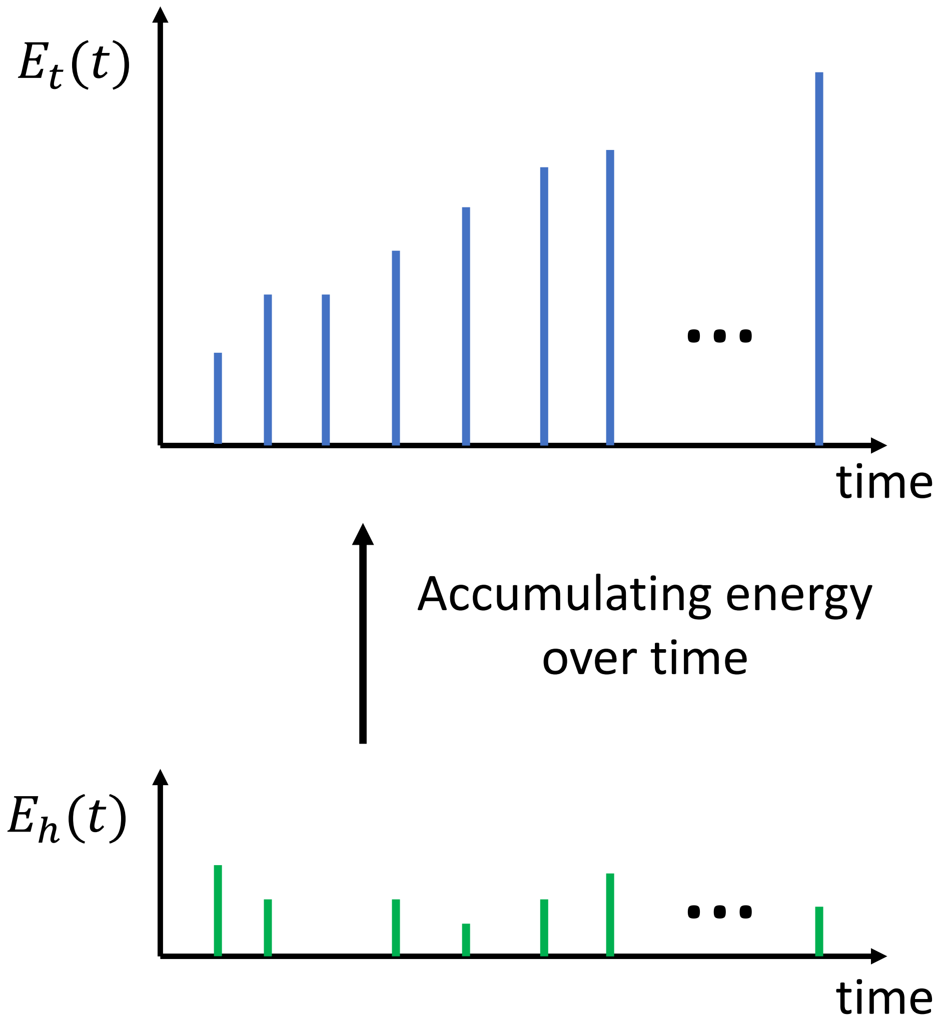
1) Intermittent learning

2) Achieving comparable learning performance

3) Providing an expected completion time

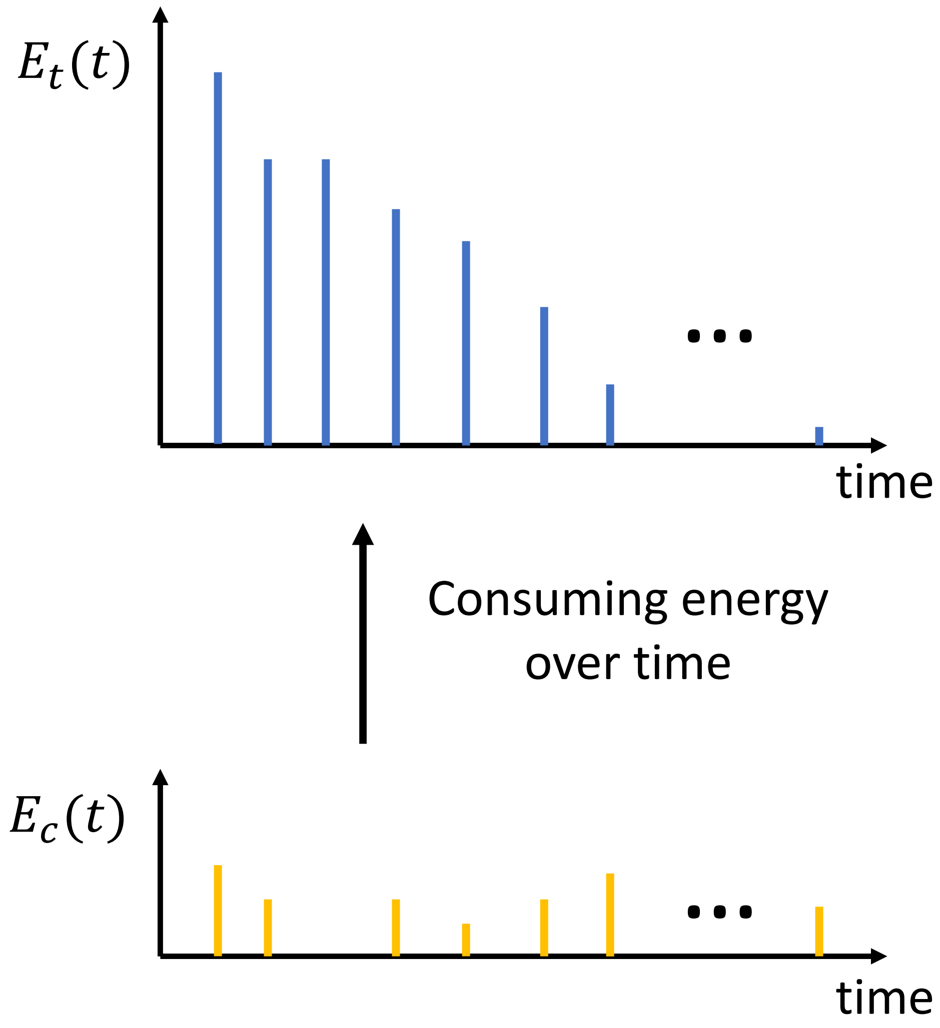
Energy-harvesting model

- Energy-harvesting model
  - $E_t(t)$ – Total available energy at time t
  - $E_h(t)$ – Newly harvested energy at time t
  - $E_t(t) = E_t(t-1) + E_h(t)$ or
  - $E_t(t) = \sum_{i=1}^{t} E_h(i)$
  - max($E_t(t)$), max($E_h(t)$) for all $t \geq 1$



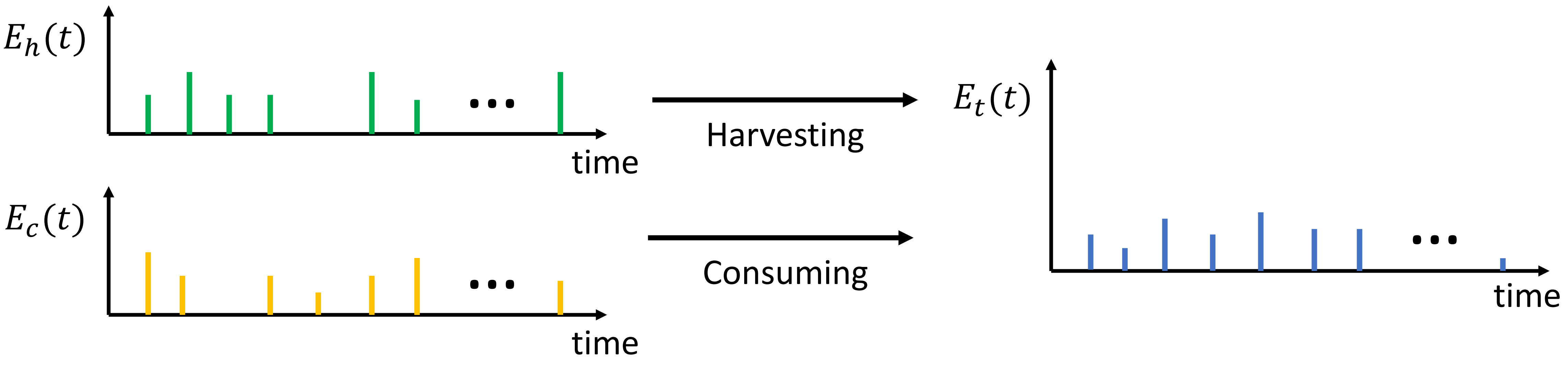
Energy-consuming model

- Energy-consuming model
  - $E_c(t)$ – Energy consumed at time t
  - $E_t(t) = E_t(t-1) - E_c(t)$ or
  - $E_t(t) = E_t(0) - \sum_{i=1}^{t} E_h(i)$
  - max($E_c(t)$) for all $t \geq 1$



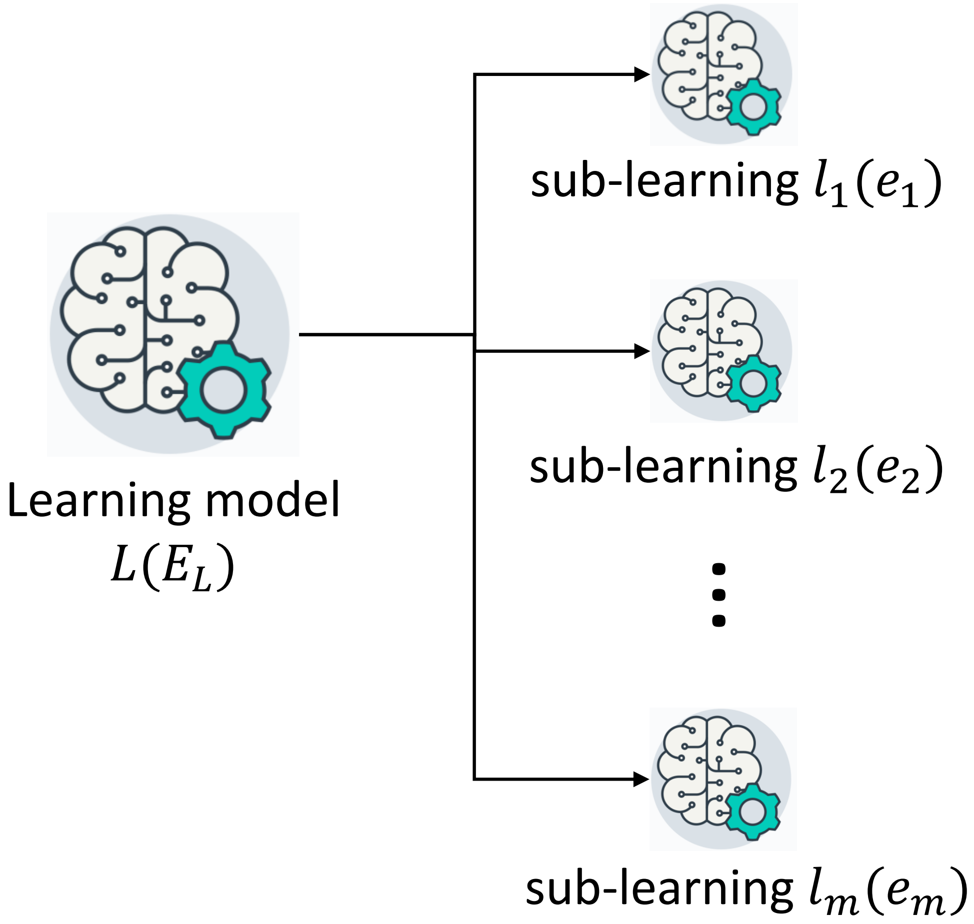
Harvesting-consuming energy model

- Harvesting and consuming happen at the same time
  - $E_t(t) = E_t(t-1) + E_h(t) - E_c(t)$ or
  - $E_t(t) = \sum_{i=1}^{t} E_h(i) - \sum_{i=1}^{t} E_c(i)$



Intermittent learning

- Given a learning model L:
  - L is decomposed into sub-learning tasks: $L = \{l_1, l_2, \ldots, l_m\}$.
  - Each sub-learning task l_i consumes e_i amount of energy: $E_L = \{e_1, e_2, \ldots, e_m\}$ where $E_L = \sum_{i=1}^{m} e_m$.
  - Intermittently perform l_i when $E_t(t) \geq e_i$ for all $1 \leq i \leq m$.
  - Keep the latest learning state consistently between l_{i-1} and l_i for all $1 \leq i \leq m$.

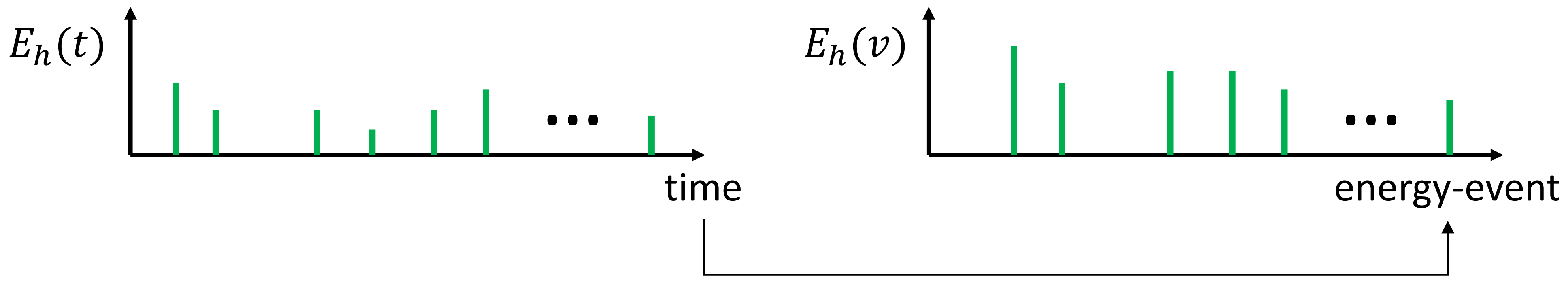


Can we tell when L will be completed?

- A learning model L is completed if its all sub-learning tasks l_i complete.
- If $E_t(t)$ or $E_h(t)$ is predictable for future time t, we can provide an expected completion time of L.
- However, making a prediction of $E_t(t)$ or $E_h(t)$ is impossible.
- Does it mean that completion time of learning L cannot be provided?

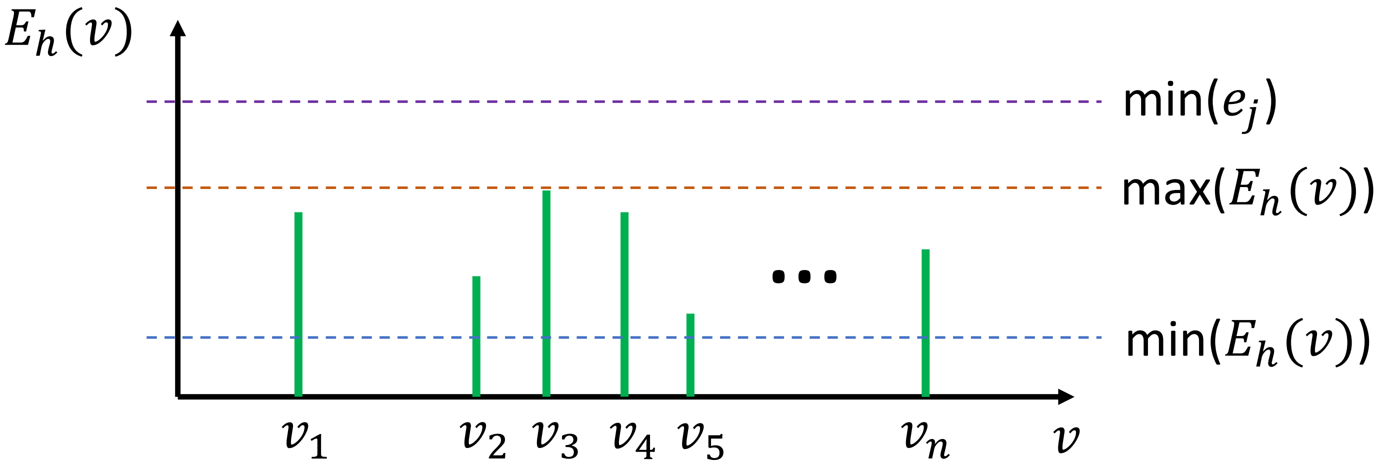
Moving from time to energy-event

- Instead of predicting $E_t(t)$ or $E_h(t)$ in terms of time, do it based on a new concept called '*energy-event*'.
  - ***Definition***: An energy-event v is an action of energy-harvesting that consequently generates $E_h(v)$ amount of energy.
  - ***Example***: 1) making a step for a pressure-harvester in shoes, 2) absorbing sunlight for 1 second with a solar panel.
  - A prediction is made based on energy-event, not time.



Properties of an energy-event

- Observations and assumptions
  - Each energy-event v harvests different amount of energy: $E_h(v_i) \neq E_h(v_j)$ for all $i \neq j$.
  - $E_h(v_i)$ comes within some common lower and upper bound usually given from physical capacity of a harvester: $\min(E_h(v)) \leq E_h(v_i) \leq \max(E_h(v))$.
  - $E_h(v_i) \leq \min(e_j)$ for all i, j where e_j is required energy for a learning task l_j.

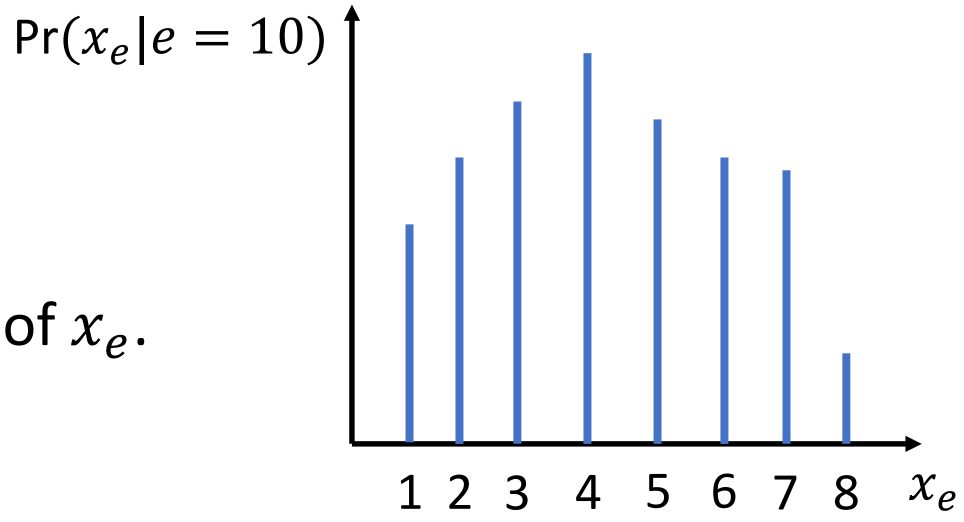


Probabilistic approach

- Thus, $E_h(v_i)$ will show statistical pattern within boundaries.
- If $E_h(v_i)$ can be statistically inferred, completion time of a learning L can be expected.
- If consecutive energy-events $E_h(v_i), E_h(v_{i+1}), \ldots, E_h(v_{i+n})$ are given, the total amount of energy harvested from those energy-events can be also obtained.

Bayesian statistical inference

- We are interested in a number of consecutive energy-events v's that collectively generate energy e.
  - ***Definition***: n_e is a random variable from distribution $f(n_e|e)$ which indicates the smallest number of consecutive v's for harvesting energy e.
  - $f(n_e|e)$=Pr$(n_e|e)$.
  - We'd like to infer distribution $f(n_e|e)$.
  - Let x_e be observations of $f(n_e|e)$.
  - Then, $\hat{f}(x_e|e)$ be a sample distribution of x_e.



Inference of $\hat{f}(x_e|e)$

- If $\hat{f}(x_e|e)$ can be expressed with population parameter θ: $\hat{f}(x_e|e,\theta)$…
  - Find θ that provides the highest probability.
  - $\theta \mapsto \hat{f}(x_e|e,\theta)$
  - Maximum Likelihood estimation of θ: $\hat{\theta}_{ML}(x_e) = \underset{\theta}{\operatorname{argmax}}\, \hat{f}(x_e|e,\theta)$
- If a prior distribution g over θ exists…
  - $\theta \mapsto \hat{f}(\theta|x_e,e) = \frac{\hat{f}(x_e|e,\theta)g(\theta|e)}{\hat{f}(x_e,e)}$
  - Maximum A Posteriori estimation of θ:
  - $\hat{\theta}_{MAP}(x_e) = \underset{\theta}{\operatorname{argmax}}\, \hat{f}(\theta|x_e,e)$

$$= \underset{\theta}{\operatorname{argmax}} \frac{\hat{f}(x_e|e,\theta)g(\theta|e)}{\cancel{\hat{f}(x_e,e)}} = \underset{\theta}{\operatorname{argmax}}\, \hat{f}(x_e|e,\theta)g(\theta|e)$$

How to optimize θ?

- Expectation Maximization
  - ***Expectation step (E step)***: calculate $Q(\theta|\theta^{(t)}) = E[\log L\left(\hat{f}(x_e|e,\theta)\right)]$.
  - ***Maximization step (M step)***: find the parameters θ that maximize:
    $$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}).$$
  - Repeat E and M step: monotonically converges to a local minimum.

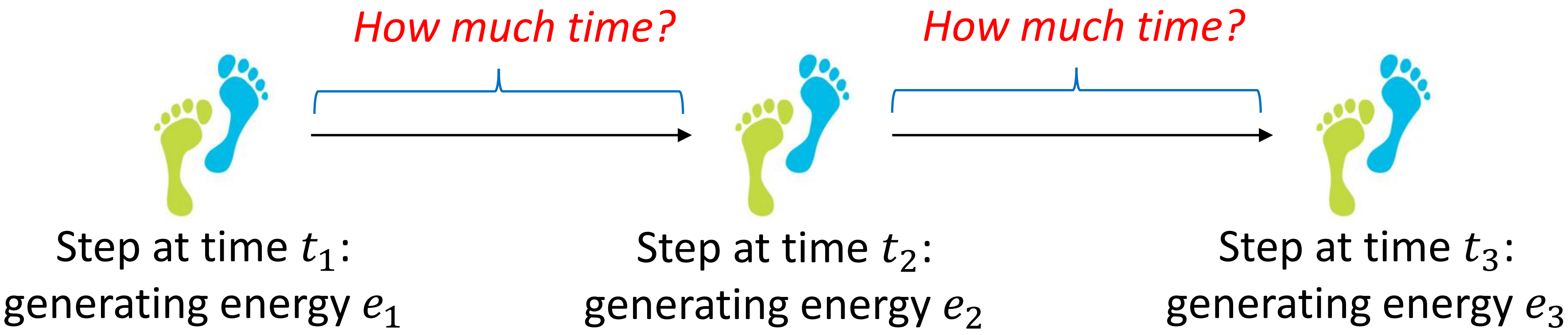
- MCMC (Markov Chain Monte Carlo)
  - Sampling from a probability distribution based on constructing a Markov chain.
  - Metropolis–Hastings algorithm or Gibbs sampling.

Providing expected completion time

- Recall: $\Pr(x_e|e) = \hat{f}(x_e|e, \theta)$
- Now that θ is known, x_e for harvesting energy e with the highest probability P can be obtained.
  - $x_e = \underset{x_e}{\operatorname{argmax}} \Pr(x_e|e) = \underset{x_e}{\operatorname{argmax}}(\hat{f}(x_e|e, \theta))$
  - $P = \max_{x_e}(\Pr(x_e|e)) = \max_{x_e}(\hat{f}(x_e|e, \theta))$
- Finally, we can claim:
  - A learning model $L = \{l_1, l_2, \dots, l_m\}$ consuming $E_L = \{e_1, e_2, \dots, e_m\}$ amount of energy is expected to complete its learning task after x_e number of energy-events with probability P.
  - Also, an expected number of energy-events can be provided: $E[x_e]$.

Problem of energy-event approach

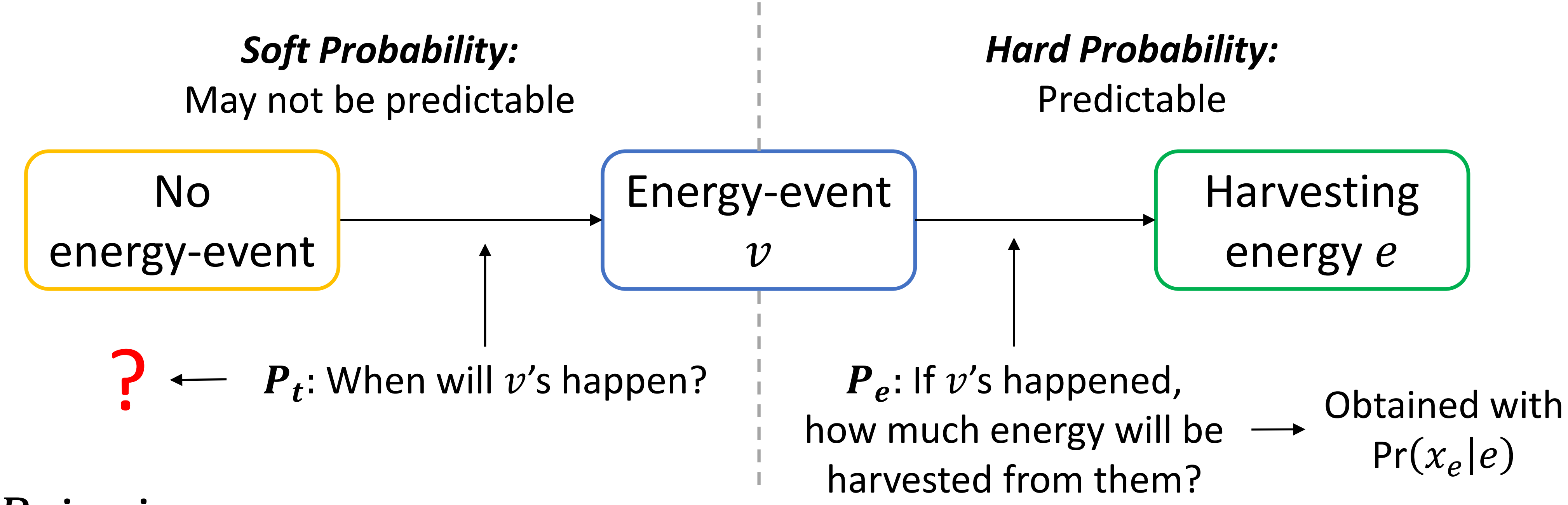
- Limitation
  - $\Pr(x_e|e)$ does not provide when the next energy-event v will happen.
  - Not intuitive: It is not expressed in terms of time.
  - ***Example:*** how do we know when a person will make next step (v) that would generates energy?



- Thus, only depending on energy-event is not enough…

Holistic view

- Flow of energy harvesting with time and energy-event



- If P_t is given…
  - The time expected to complete a learning task can be given with $\Pr(x_e|e)$.
  - But obtaining P_t is difficult.

Problem 2)

- Perform any learning model/task L in a sustainable/persistent manner given intermittently-provided energy E which achieves comparable learning performance within an expected completion/execution time with reasonable certainty P.

1) Intermittent learning

2) Achieving comparable learning performance

3) Providing an expected completion time

Is a learning model L learnable?

- Some class C of target concepts is learnable if…
  - Each target concept in C can be learned from a polynomial number of training examples.
  - The processing time per example is also polynomially bounded.

Learning performance criteria

- ***Sample complexity***: How many training examples are needed for a learner to converge (with high probability) to a successful hypothesis?

- ***Computational complexity***: How much computational effort is needed for a learner to converge (with high probability) to a successful hypothesis?

- ***Mistake bound***: How many training examples will the learner misclassify before converging to a successful hypothesis?

PAC-learnable (Computational learning theory)

- ***PAC***: Probably Approximately Correct learning
  - ***Definition***: Consider a concept class C defined over a set of instances X of length n and a learner L using hypothesis space H. C is **PAC-learnable** by L using H if for all $c \in C$, distributions D over X, ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will with probability at least $(1 - \delta)$ output a hypothesis h $\in H$ such that $error_D(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, n, and $size(c)$. ***- Leslie Valiant, 1984 –***

- With high probability $(1 - \delta)$ (the "probably" part), the selected function will have low generalization error ϵ (the "approximately correct" part).

Learning complexity

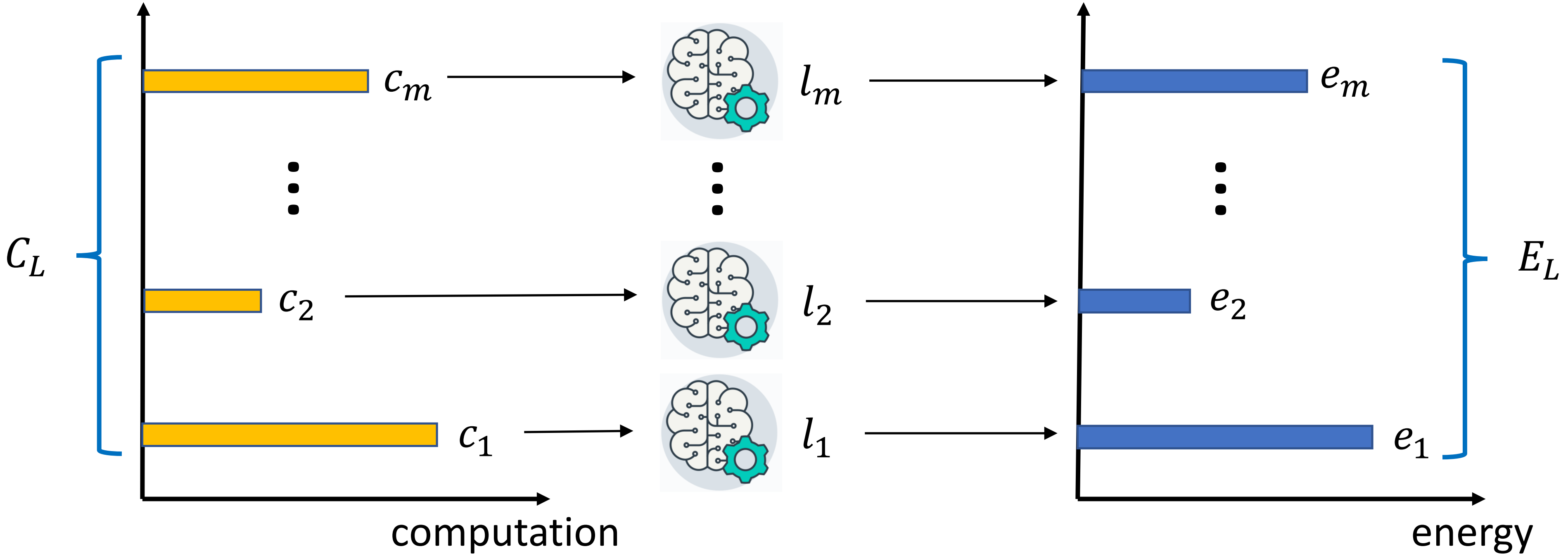
- Sample complexity of a PAC-learnable learning model
  - $m \geq \frac{1}{\epsilon}\left(4\log_2\frac{2}{\delta} + 8VC(H)\log_2\frac{13}{\epsilon}\right)$ - ***Blumer, 1989***
  - m: the number of training example required to achieve PAC learning.

  - ***Definition***: The **Vapnik-Chervonenkis dimension,** $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of H shattered by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$.

- The complexity grows only polynomially with $1/\epsilon$, $1/\delta$, the size of the instances, and the size of the target concept if it is PAC-learnable.

Construction of L and E_L from C_L

- Total computation $C_L = \{c_1, c_2, \ldots, c_m\}$ for a learning model L can be provided from the PAC-learnable analysis.
  - Thus, a learning model $L = \{l_1, l_2, \ldots, l_m\}$ and its consequential energy consumption $E_L = \{e_1, e_2, \ldots, e_m\}$ can be constructed based on C_L.



Other constraints

- Embedded systems have other resource constraints besides energy
  - Small memory and low computational capacity.
  - Usually, they cannot perform L as it is even if sufficient energy is given.
  - Thus, the learning model L should be reduced to fit into them.

Learning model L
with C_L and M_L

Requires computation C_L
and M_L amount of memory

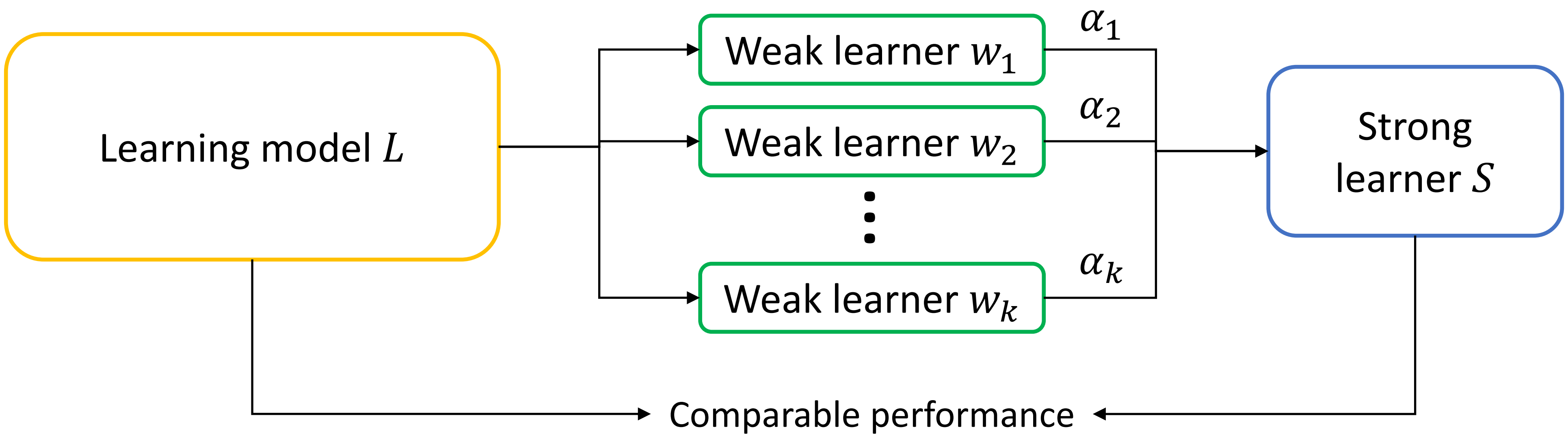
L can't fit into A
since $C_L \gg C_A$
and
$M_L \gg M_A$

Energy-Free
learning system A
with C_A and M_A

Computational capability of C_A
and M_A amount of memory

AdaBoost – *Schapire, 2012*

- Construct a number of weak learners w_i that perform same learning task as L but use less resource.
  - Each w_i uses only the amount of resource available in the system.
  - A strong learner S can be built by adaptively boosting learning ability of w_i.
  - S would eventually show comparable learning performance to L.



Thank you!