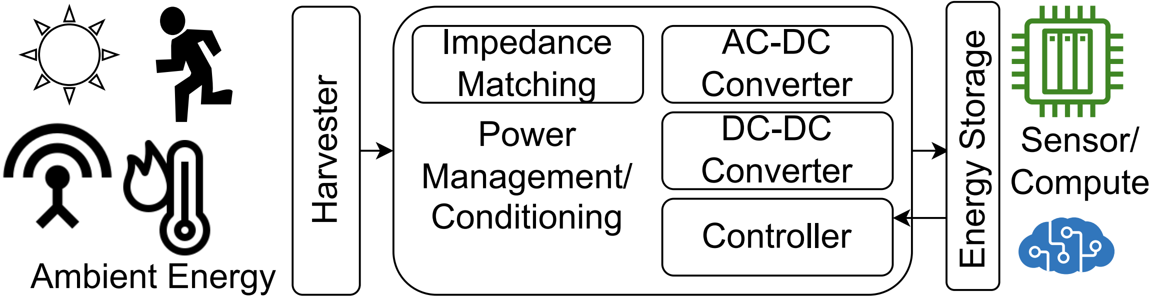
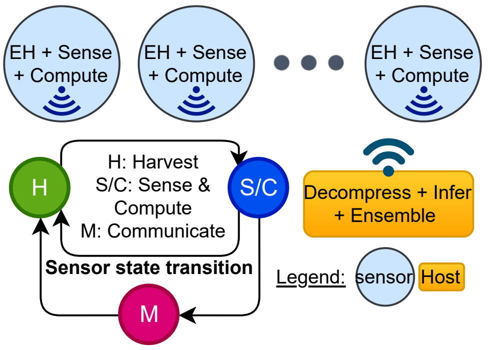
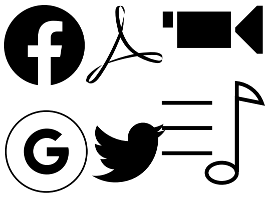
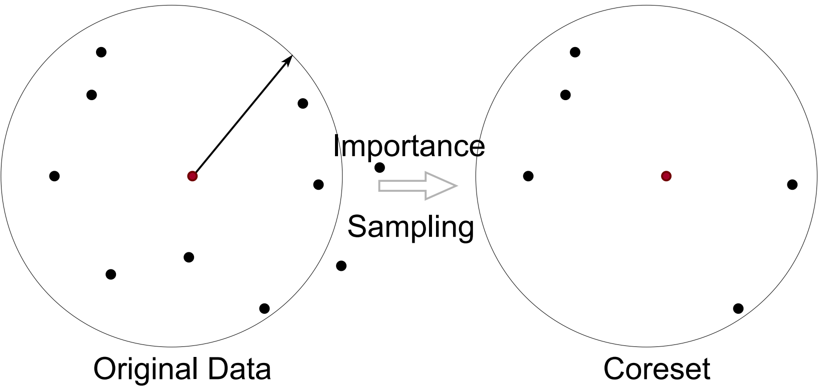
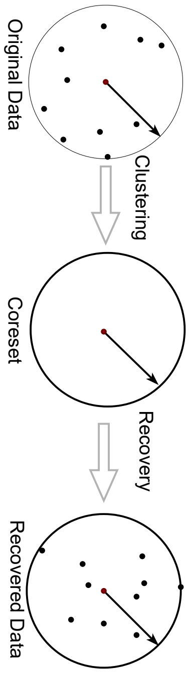
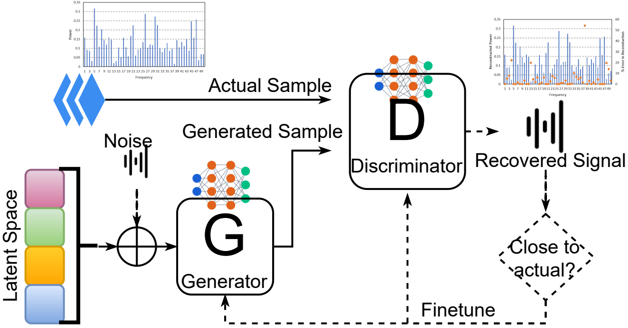

Ambient Energy
Harvester
Impedance Matching
Power Management/ Conditioning
AC-DC Converter
DC-DC Converter
Controller
Energy Storage
Sensor/ Compute

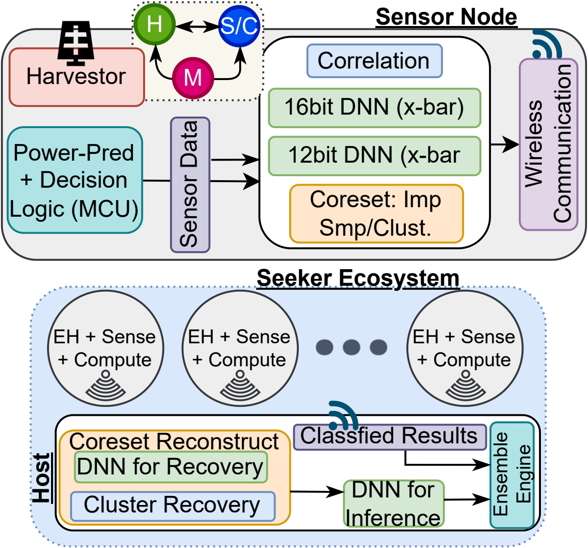

EH + Sense + Compute
EH + Sense + Compute
EH + Sense + Compute
H: Harvest
S/C: Sense & Compute
M: Communicate
H
S/C
M
Sensor state transition
Decompress + Infer + Ensemble
Legend:
sensor
Host

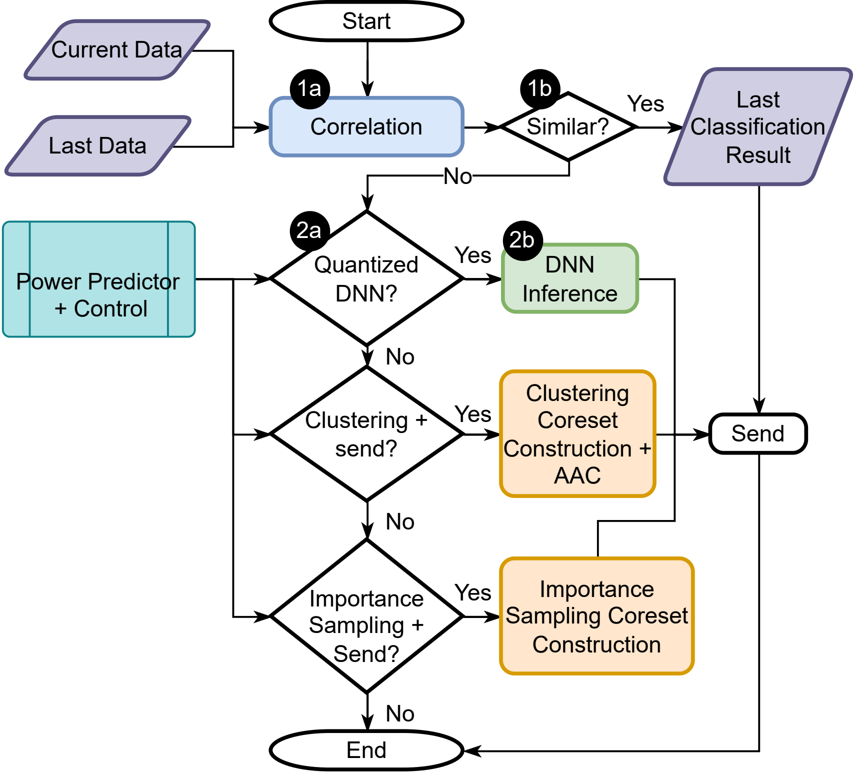


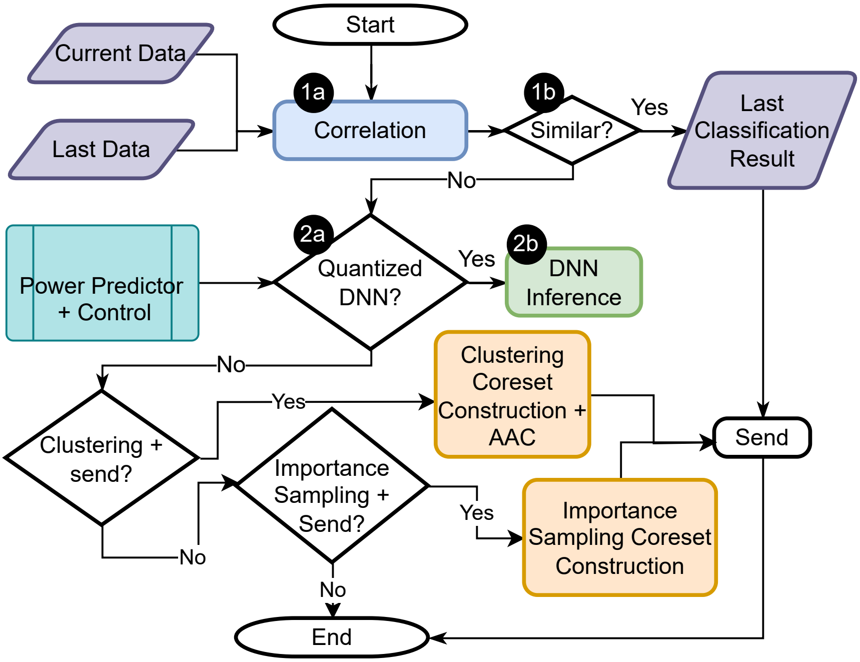

Importance
Sampling
Original Data
Coreset

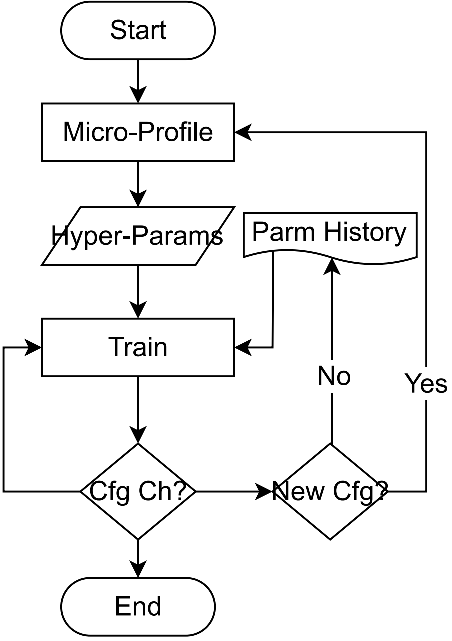

Original Data
Clustering
Coreset
Recovery
Recovered Data

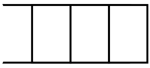

Power
Frequency
Actual Sample
Generated Sample
Noise
Latent Space
G
Generator
D
Discriminator
Reconstructed Power
% Error in Reconstruction
Frequency
Recovered Signal
Close to actual?
Finetune

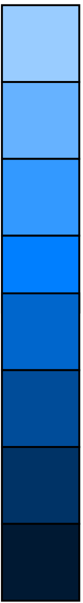

Sensor Node
H
S/C
M
Harvestor
Correlation
16bit DNN (x-bar)
12bit DNN (x-bar
Coreset: Imp Smp/Clust.
Power-Pred + Decision Logic (MCU)
Sensor Data
Wireless Communication
Seeker Ecosystem
EH + Sense + Compute
EH + Sense + Compute
EH + Sense + Compute
Host
Coreset Reconstruct
DNN for Recovery
Cluster Recovery
Classfied Results
DNN for Inference
Ensemble Engine

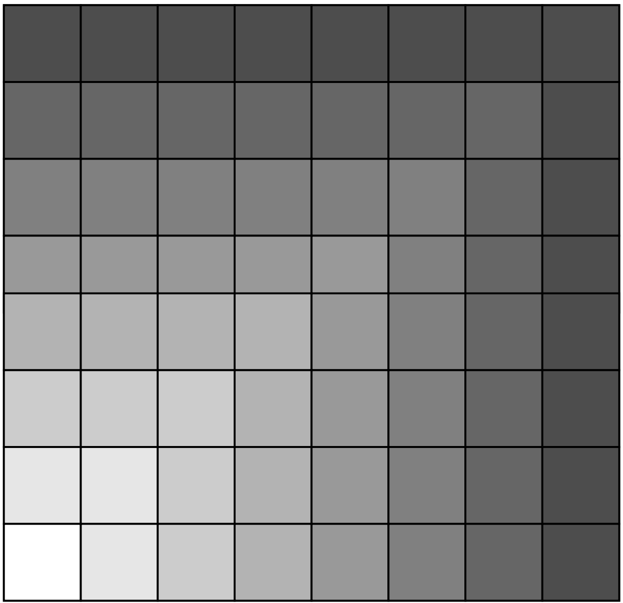

Start
Current Data
Last Data
1a
Correlation
1b
Similar?
Yes
Last Classification Result
No
Power Predictor + Control
2a
Quantized DNN?
Yes
2b
DNN Inference
No
Clustering + send?
Yes
Clustering Coreset Construction + AAC
Send
No
Importance Sampling + Send?
Yes
Importance Sampling Coreset Construction
No
End

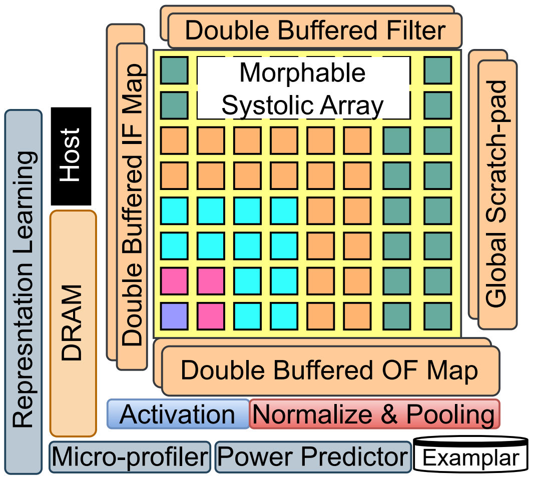

Start
Current Data
Last Data
1a
Correlation
1b
Similar?
Yes
Last Classification Result
No
Power Predictor + Control
2a
Quantized DNN?
Yes
2b
DNN Inference
No
Clustering Coreset Construction + AAC
Yes
Clustering + send?
Importance Sampling + Send?
Send
Yes
Importance Sampling Coreset Construction
No
No
End

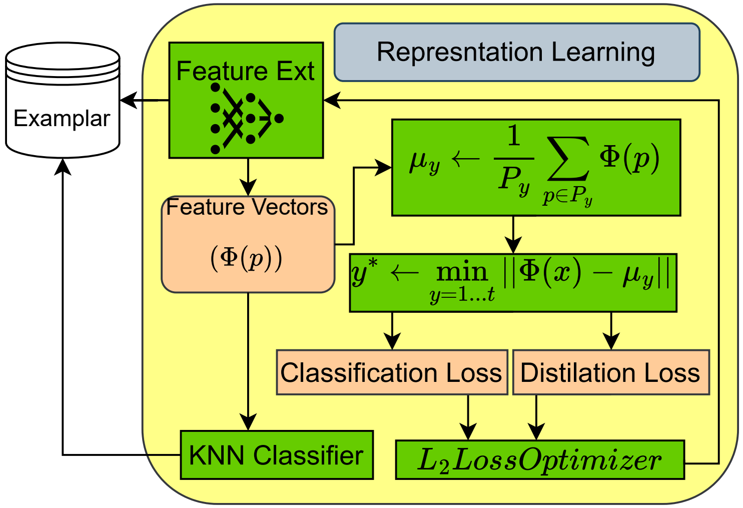

Start
Micro-Profile
Hyper-Params
Parm History
Train
No
Yes
Cfg Ch?
New Cfg?
End

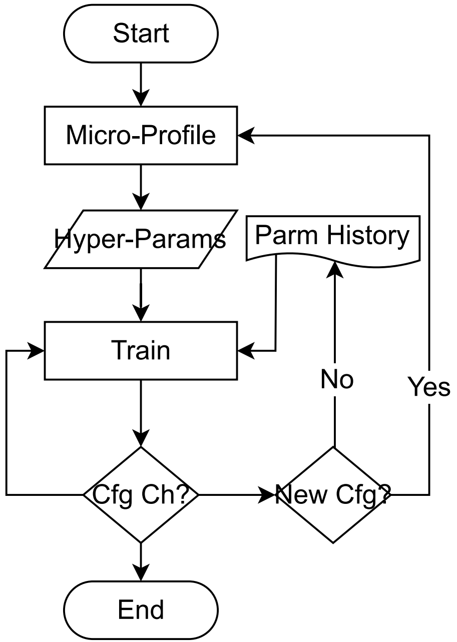


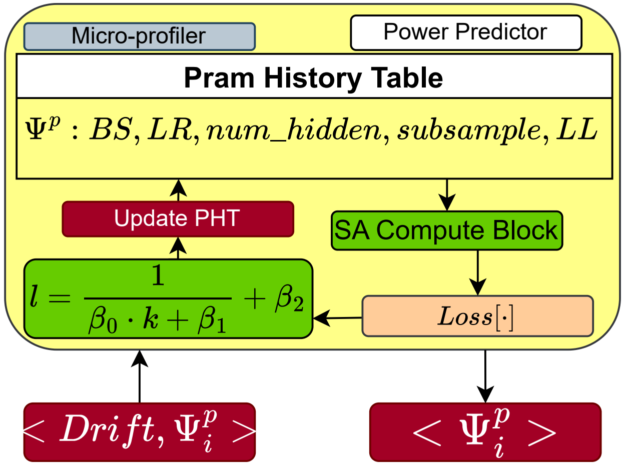


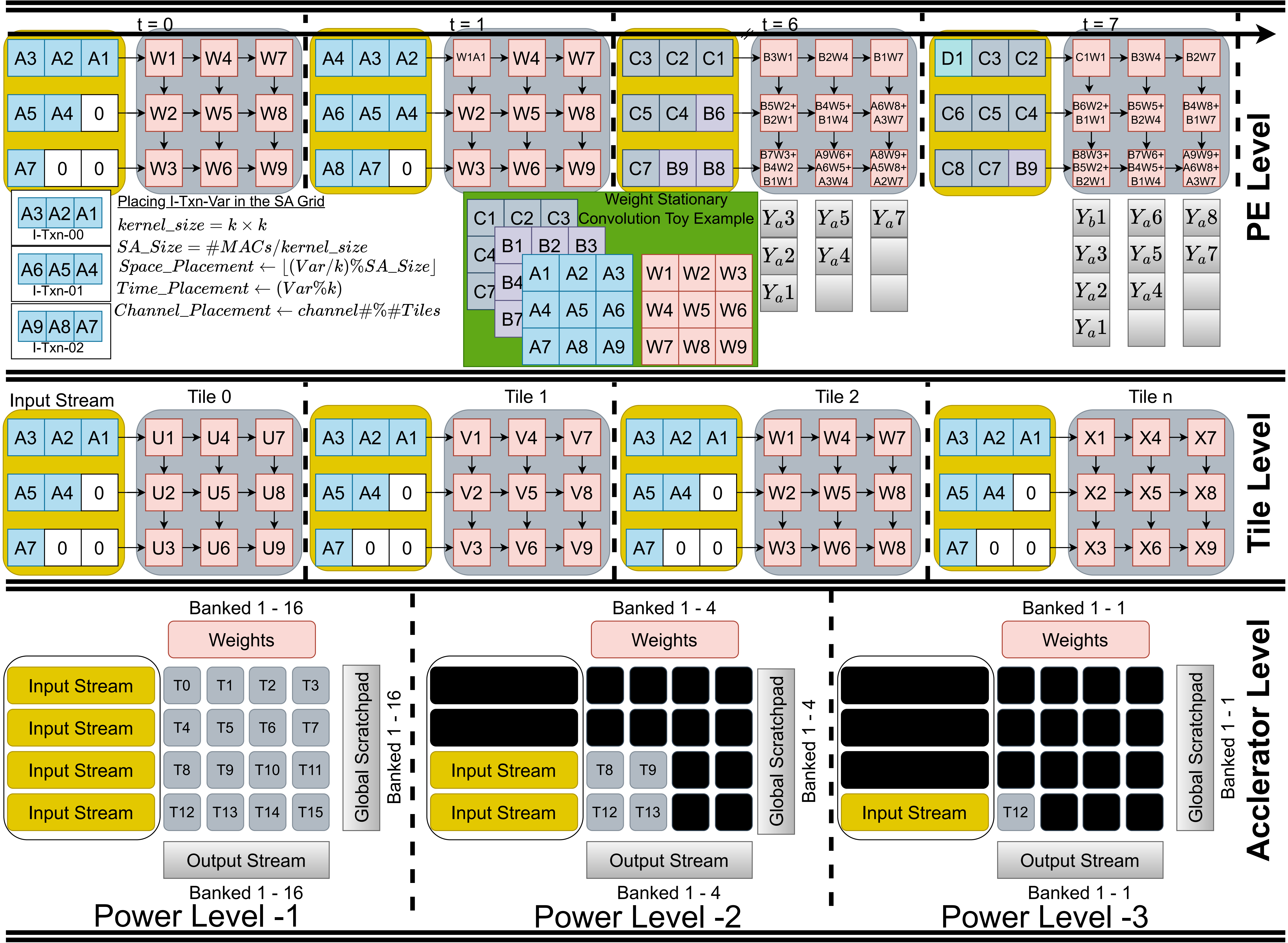


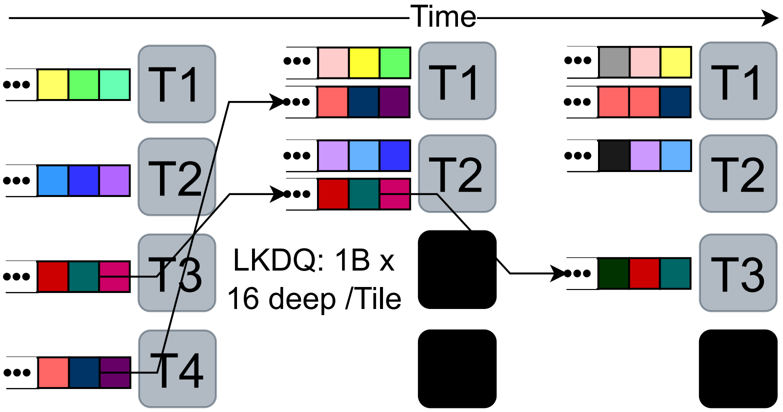

Represntation Learning
Host
DRAM
Double Buffered IF Map
Double Buffered Filter
Morphable Systolic Array
Global Scratch-pad
Double Buffered OF Map
Activation
Normalize & Pooling
Micro-profiler
Power Predictor
Examplar

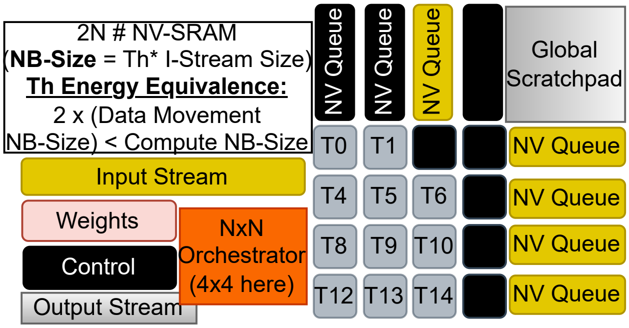

Represntation Learning
Examplar
Feature Ext
$\mu_y \leftarrow \frac{1}{P_y} \sum_{p \in P_y} \Phi(p)$
Feature Vectors
$(\Phi(p))$
$y^* \leftarrow \min_{y=1...t} ||\Phi(x) - \mu_y||$
Classification Loss
Distilation Loss
KNN Classifier
$L_2 Loss Optimizer$

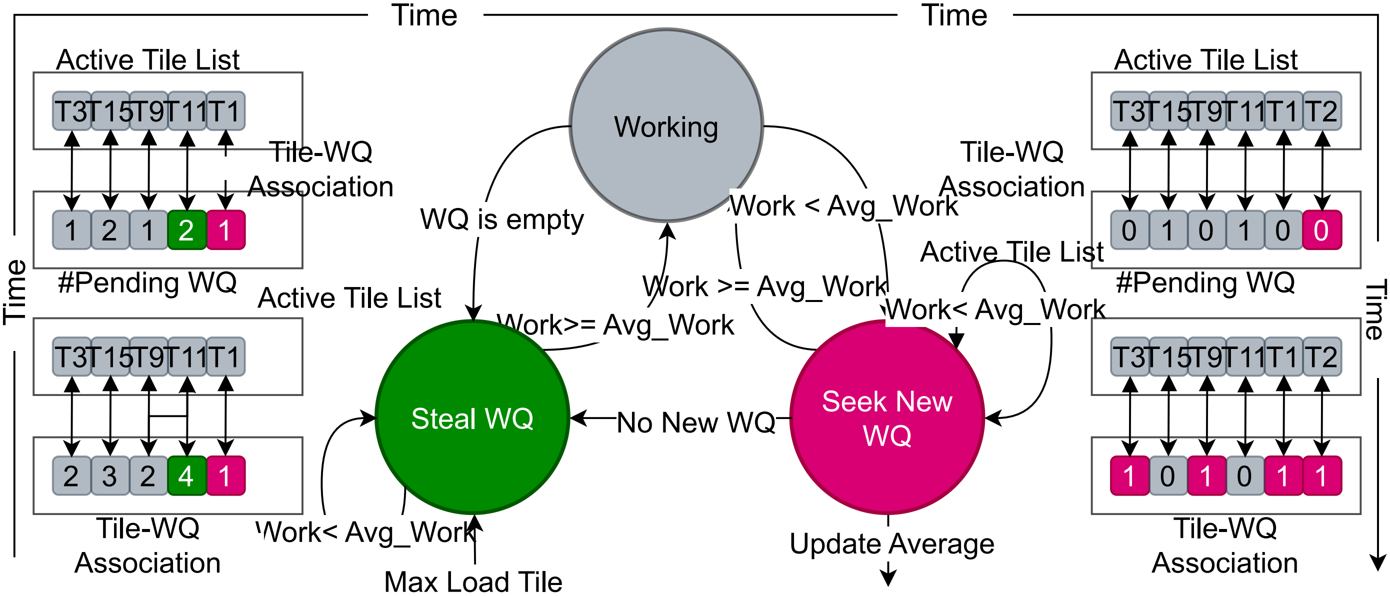

Start
Micro-Profile
Hyper-Params
Parm History
Train
No
Yes
Cfg Ch?
New Cfg?
End

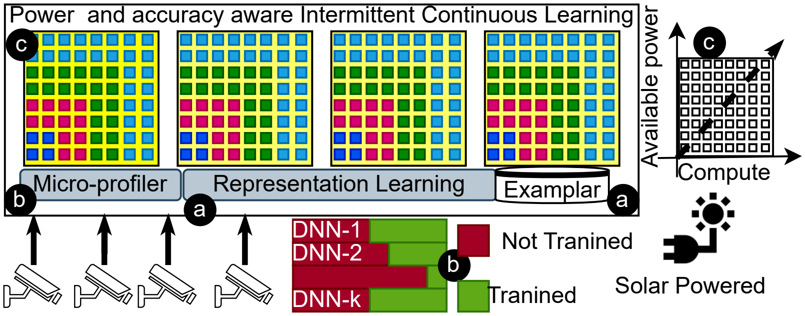

Micro-profiler
Power Predictor
Pram History Table
$\Psi^p : BS, LR, num\_hidden, subsample, LL$
Update PHT
SA Compute Block
$l = \frac{1}{\beta_0 \cdot k + \beta_1} + \beta_2$
$Loss[\cdot]$
$< Drift, \Psi_i^p >$
$< \Psi_i^p >$

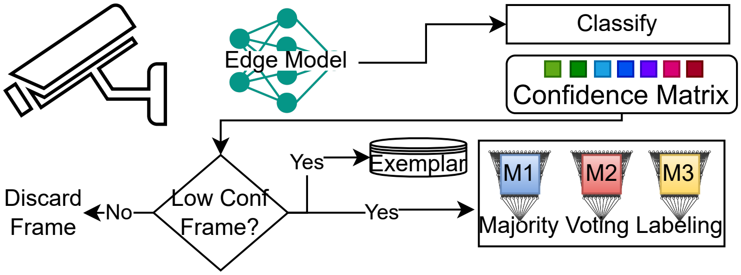

t = 0
t = 1
t = 6
t = 7
A3 A2 A1
A5 A4 0
A7 0 0
W1 W4 W7
W2 W5 W8
W3 W6 W9
A4 A3 A2
A6 A5 A4
A8 A7 0
W1A1
C3 C2 C1
C5 C4 B6
C7 B9 B8
B3W1 B2W4 B1W7
B5W2+ B2W1
B4W5+ B1W4
A6W8+ A3W7
B7W3+ B4W2 B1W1
A9W6+ A6W5+ A3W4
A8W9+ A5W8+ A2W7
D1 C3 C2
C6 C5 C4
C8 C7 B9
C1W1 B3W4 B2W7
B6W2+ B1W1
B5W5+ B2W4
B4W8+ B1W7
B8W3+ B5W2+ B2W1
B7W6+ B4W5+ B1W4
A9W9+ A6W8+ A3W7
Y_a3 Y_a2 Y_a1
Y_a5 Y_a4
Y_a7
Y_b1 Y_a3 Y_a2 Y_a1
Y_a6 Y_a5 Y_a4
Y_a8 Y_a7
PE Level
A3 A2 A1 I-Txn-00
A6 A5 A4 I-Txn-01
A9 A8 A7 I-Txn-02
Placing I-Txn-Var in the SA Grid
$kernel\_size = k \times k$
$SA\_Size = \#MACs/kernel\_size$
$Space\_Placement \leftarrow \lfloor (Var/k)\%SA\_Size \rfloor$
$Time\_Placement \leftarrow (Var\%k)$
$Channel\_Placement \leftarrow channel\#\%\#Tiles$
Weight Stationary Convolution Toy Example
C1 C2 C3 C4 C7
B1 B2 B3 B4 B7
A1 A2 A3 A4 A5 A6 A7 A8 A9
W1 W2 W3 W4 W5 W6 W7 W8 W9
Input Stream
Tile 0
Tile 1
Tile 2
Tile n
U1 U4 U7 U2 U5 U8 U3 U6 U9
V1 V4 V7 V2 V5 V8 V3 V6 V9
W1 W4 W7 W2 W5 W8 W3 W6 W8
X1 X4 X7 X2 X5 X8 X3 X6 X9
Tile Level
Banked 1 - 16
Weights
Input Stream
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15
Global Scratchpad
Banked 1 - 16
Output Stream
Banked 1 - 16
Power Level -1
Banked 1 - 4
Weights
Input Stream
T8 T9 T12 T13
Global Scratchpad
Banked 1 - 4
Output Stream
Banked 1 - 4
Power Level -2
Banked 1 - 1
Weights
Input Stream
T12
Global Scratchpad
Banked 1 - 1
Output Stream
Banked 1 - 1
Power Level -3
Acclerator Level

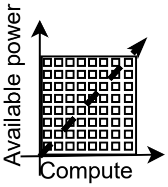

Time
T1
T2
T3
T4
LKDQ: 1B x
16 deep /Tile
T1
T2
T1
T2
T3

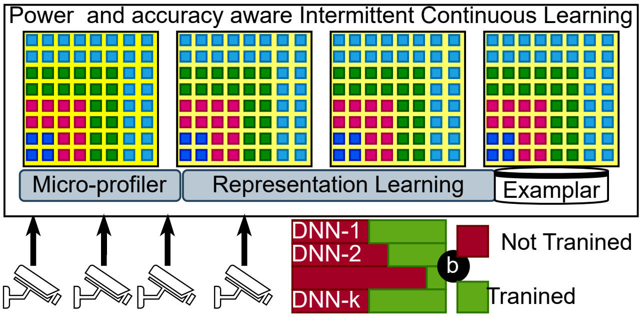

2N # NV-SRAM
(NB-Size = Th* I-Stream Size)
Th Energy Equivalence:
2 x (Data Movement
NB-Size) < Compute NB-Size
NV Queue
NV Queue
NV Queue
Global
Scratchpad
T0
T1
NV Queue
T4
T5
T6
NV Queue
T8
T9
T10
NV Queue
T12
T13
T14
NV Queue
Input Stream
Weights
Control
NxN
Orchestrator
(4x4 here)
Output Stream

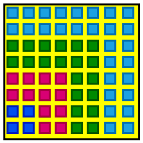

Time
Time
Active Tile List
T3 T15 T9 T11 T1
Tile-WQ
Association
1 2 1 2 1
#Pending WQ
Time
Active Tile List
T3 T15 T9 T11 T1
2 3 2 4 1
Tile-WQ
Association
Working
WQ is empty
Work < Avg_Work
Work >= Avg_Work
Work>= Avg_Work
Active Tile List
Active Tile List
Work< Avg_Work
Steal WQ
No New WQ
Seek New
WQ
Work< Avg_Work
Max Load Tile
Update Average
Active Tile List
T3 T15 T9 T11 T1 T2
Tile-WQ
Association
0 1 0 1 0 0
#Pending WQ
Time
T3 T15 T9 T11 T1 T2
1 0 1 0 1 1
Tile-WQ
Association

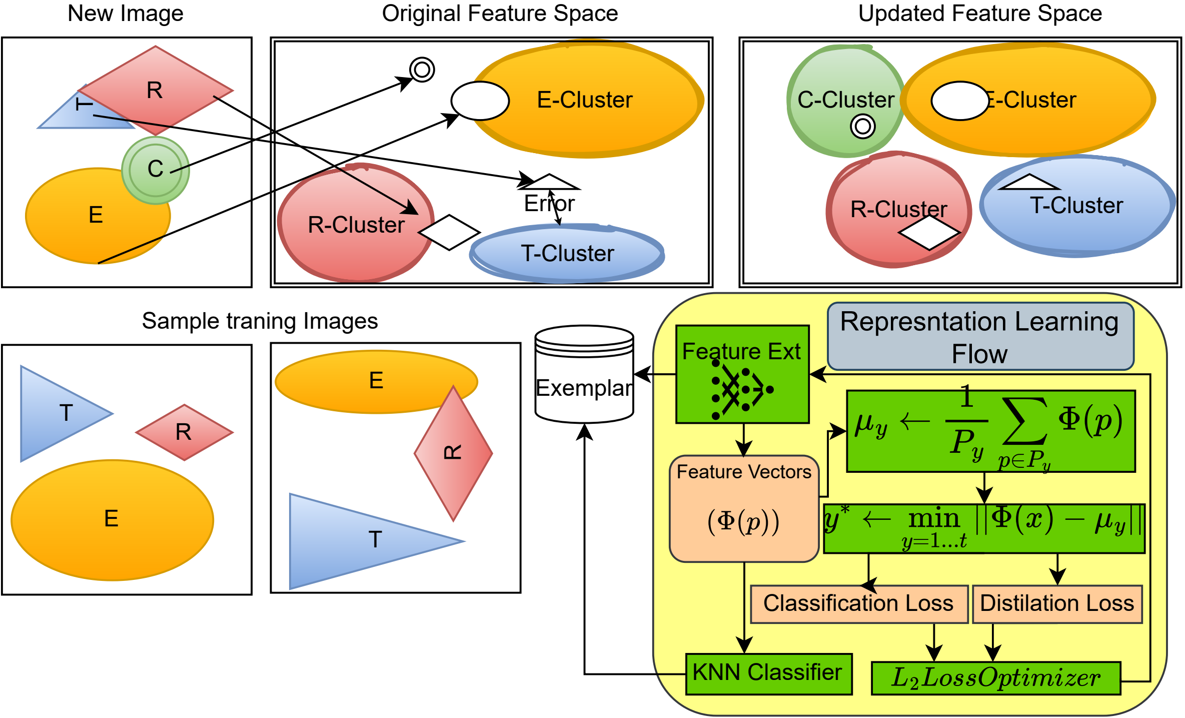

Power and accuracy aware Intermittent Continuous Learning
c
Micro-profiler
Representation Learning
Examplar
a
b
a
Available power
Compute
c
DNN-1
DNN-2
DNN-k
b
Not Tranined
Tranined
Solar Powered

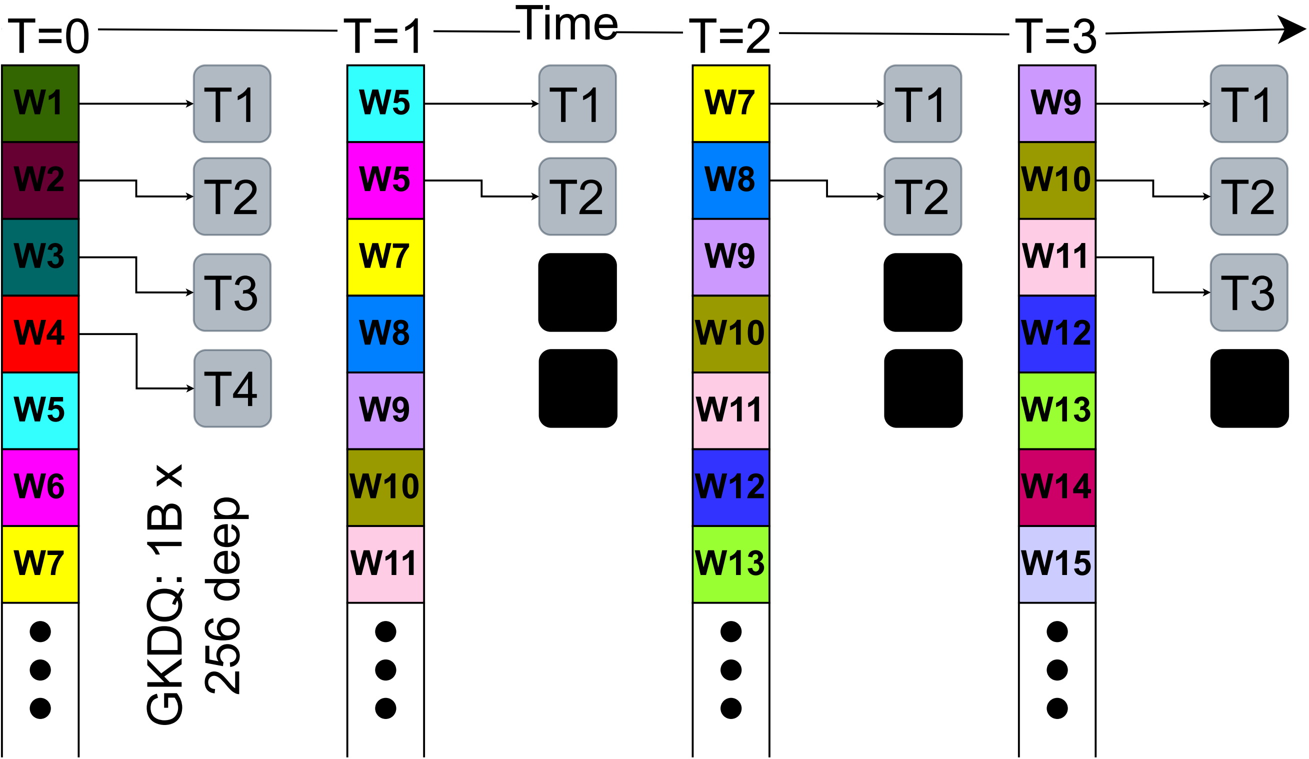

Edge Model
Classify
Confidence Matrix
Low Conf Frame?
No
Discard Frame
Yes
Exemplar
Yes
M1
M2
M3
Majority Voting Labeling

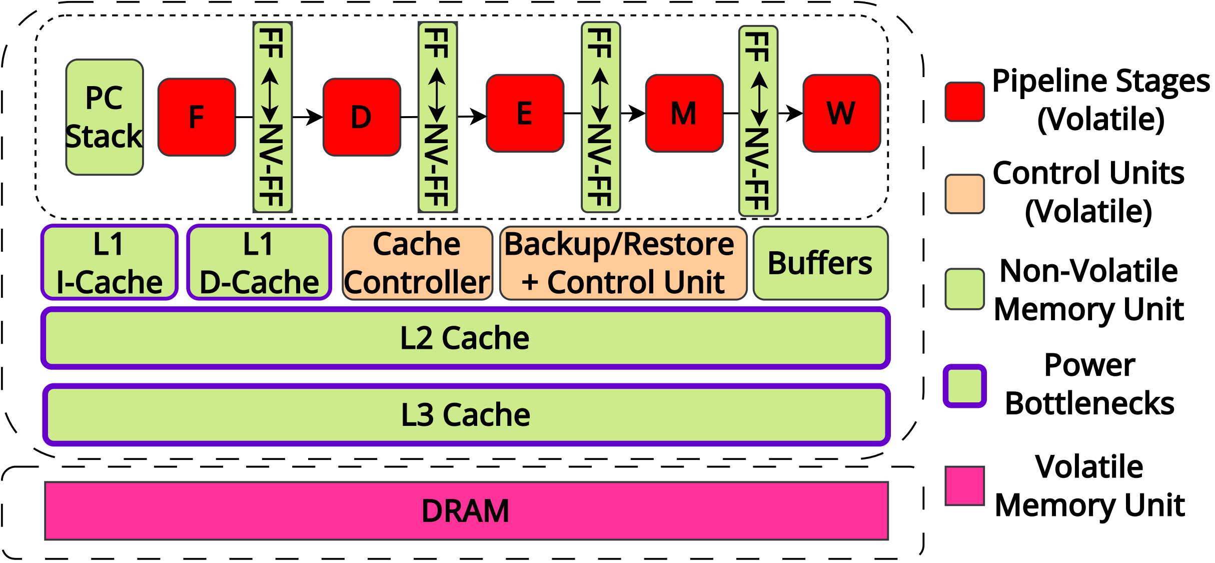

Available power
Compute

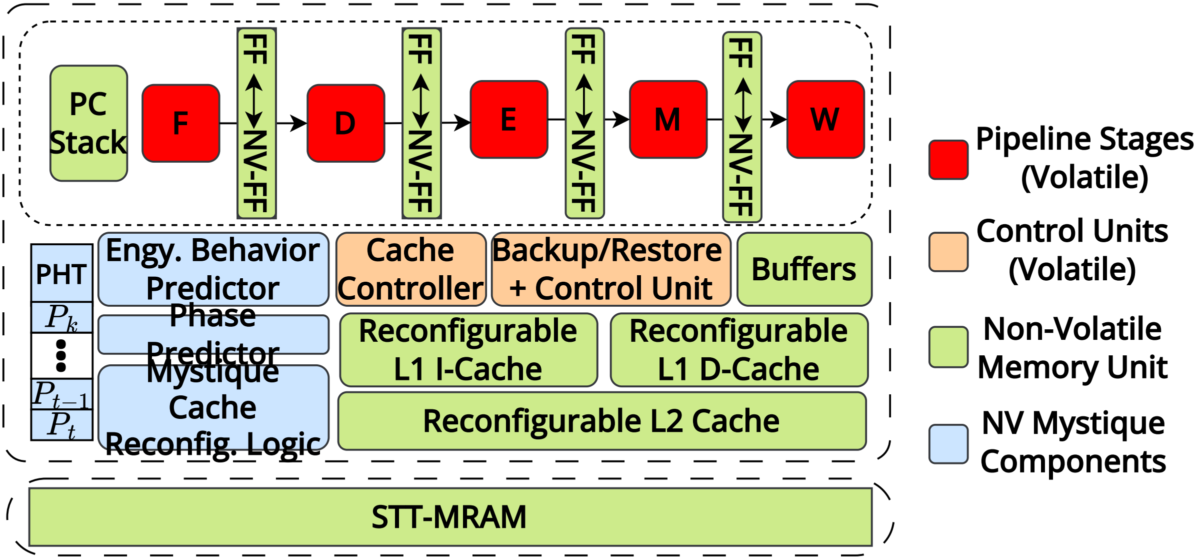

Power and accuracy aware Intermittent Continuous Learning
Micro-profiler
Representation Learning
Examplar
DNN-1
DNN-2
DNN-k
b
Not Tranined
Tranined

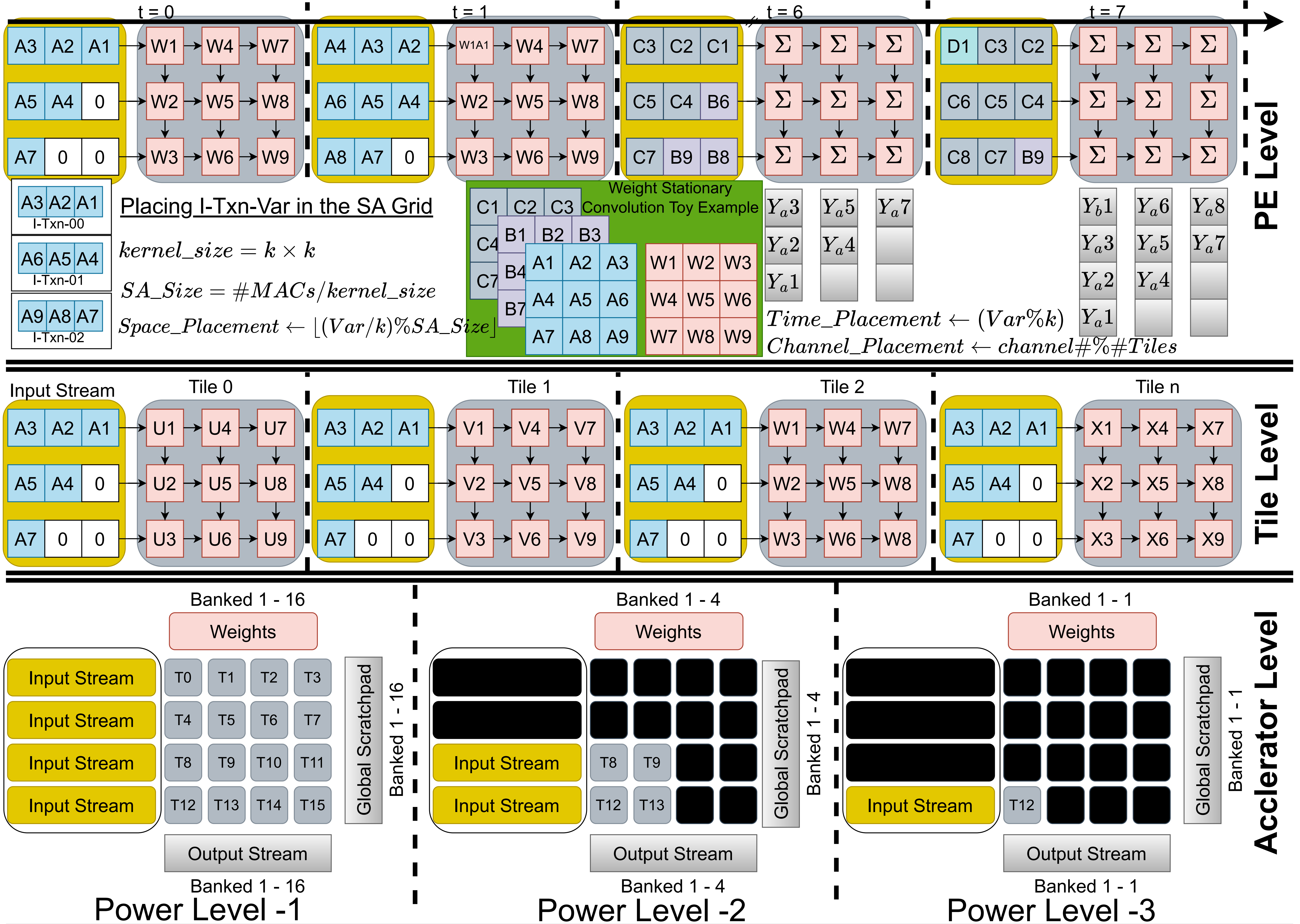


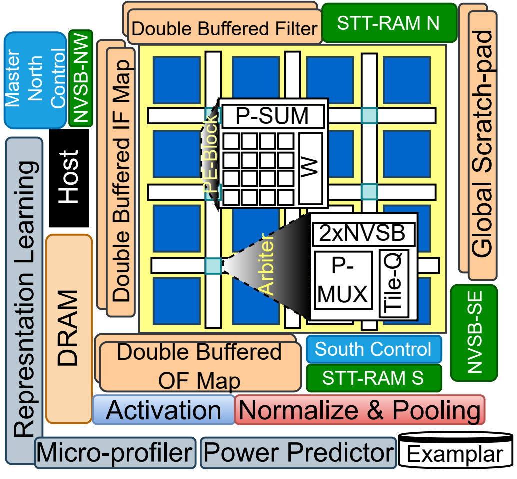

New Image
Original Feature Space
Updated Feature Space
R
T
C
E
E-Cluster
R-Cluster
Error
T-Cluster
C-Cluster
E-Cluster
R-Cluster
T-Cluster
Sample traning Images
T
R
E
E
R
T
Exemplar
Represntation Learning Flow
Feature Ext
$\mu_y \leftarrow \frac{1}{P_y} \sum_{p \in P_y} \Phi(p)$
Feature Vectors
$(\Phi(p))$
$y^* \leftarrow \min_{y=1 \ldots t} ||\Phi(x) - \mu_y||$
Classification Loss
Distilation Loss
KNN Classifier
$L_2 Loss Optimizer$

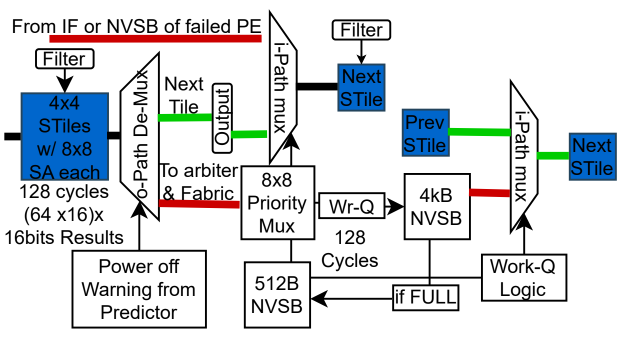

Time
T=0
T=1
T=2
T=3
W1
W2
W3
W4
W5
W6
W7
T1
T2
T3
T4
GKDQ: 1B x 256 deep
W5
W5
W7
W8
W9
W10
W11
T1
T2
W7
W8
W9
W10
W11
W12
W13
T1
T2
W9
W10
W11
W12
W13
W14
W15
T1
T2
T3

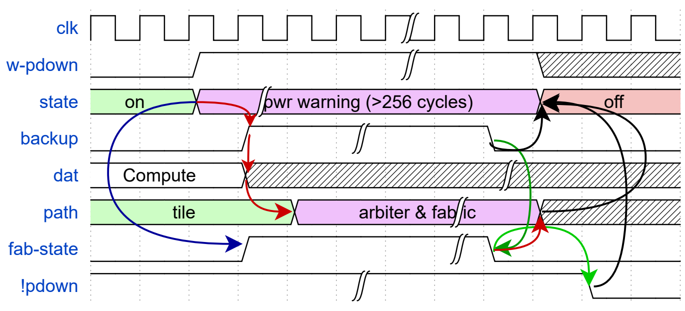

PC Stack
F
FF ↔ NV-FF
D
FF ↔ NV-FF
E
FF ↔ NV-FF
M
FF ↔ NV-FF
W
L1 I-Cache
L1 D-Cache
Cache Controller
Backup/Restore + Control Unit
Buffers
L2 Cache
L3 Cache
DRAM
Pipeline Stages (Volatile)
Control Units (Volatile)
Non-Volatile Memory Unit
Power Bottlenecks
Volatile Memory Unit

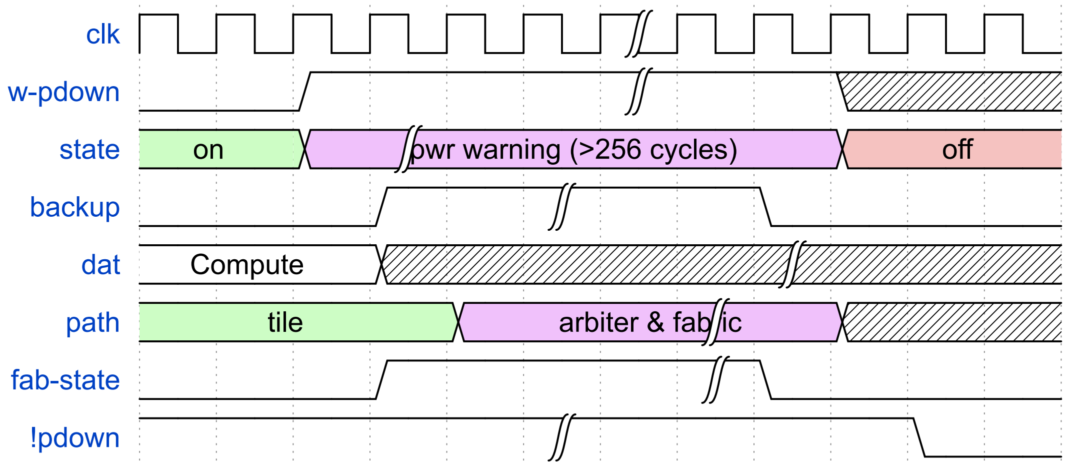

PC Stack
F
FF ↔ NV-FF
D
FF ↔ NV-FF
E
FF ↔ NV-FF
M
FF ↔ NV-FF
W
PHT
P_k
P_{t-1}
P_t
Engy. Behavior Predictor
Phase Predictor
Mystique Cache Reconfig. Logic
Cache Controller
Backup/Restore + Control Unit
Buffers
Reconfigurable L1 I-Cache
Reconfigurable L1 D-Cache
Reconfigurable L2 Cache
STT-MRAM
Pipeline Stages (Volatile)
Control Units (Volatile)
Non-Volatile Memory Unit
NV Mystique Components

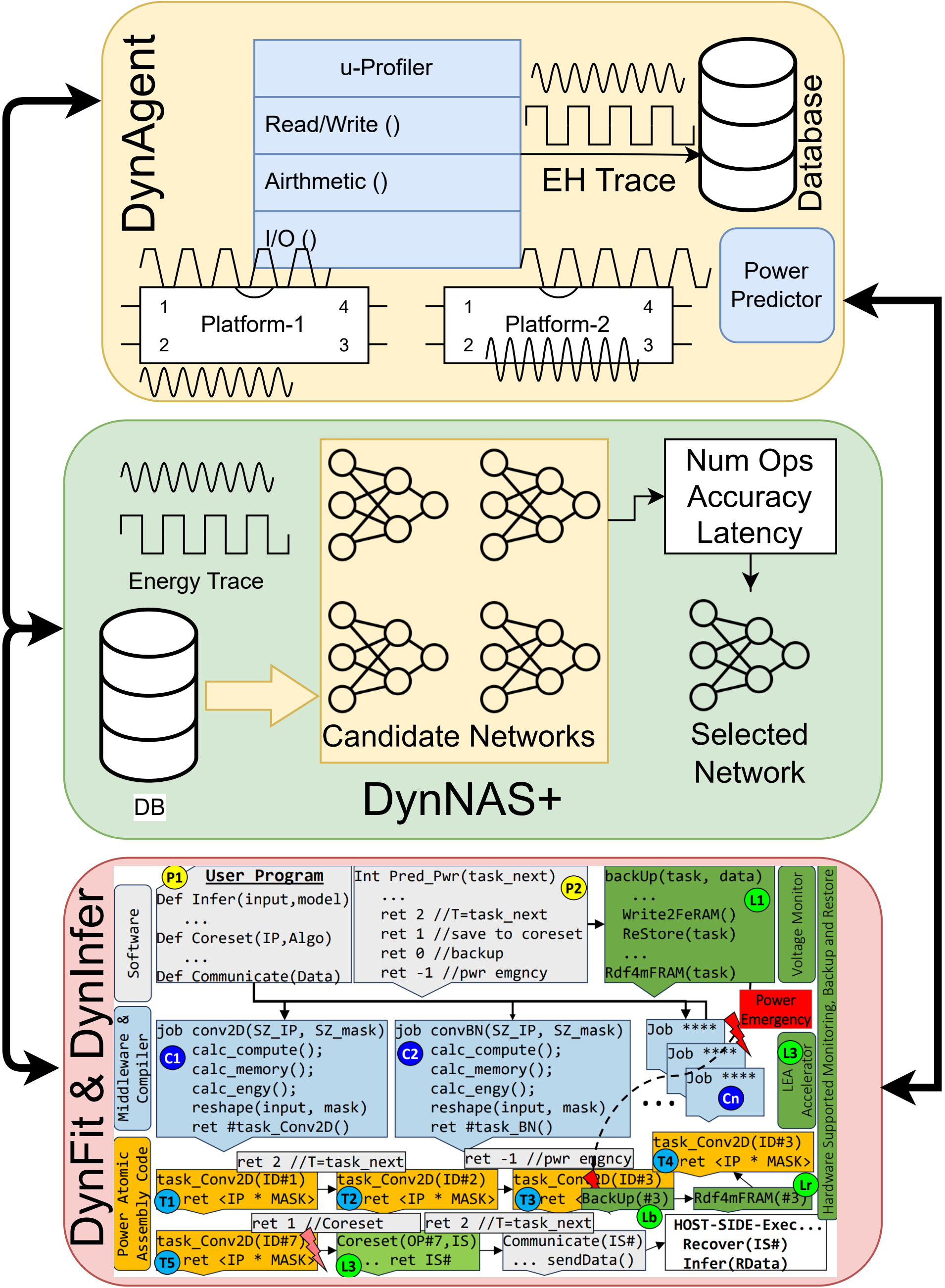

t = 0
t = 1
t = 6
t = 7
PE Level
Placing I-Txn-Var in the SA Grid
I-Txn-00
I-Txn-01
I-Txn-02
$kernel\_size = k \times k$
$SA\_Size = \#MACs/kernel\_size$
$Space\_Placement \leftarrow \lfloor (Var/k)\%SA\_Size \rfloor$
Weight Stationary Convolution Toy Example
$Time\_Placement \leftarrow (Var\%k)$
$Channel\_Placement \leftarrow channel\#\%\#Tiles$
Y_a1 Y_a2 Y_a3 Y_a4 Y_a5 Y_a6 Y_a7 Y_a8 Y_b1
Input Stream
Tile 0
Tile 1
Tile 2
Tile n
Tile Level
Banked 1 - 16
Banked 1 - 4
Banked 1 - 1
Weights
Input Stream
Global Scratchpad
Output Stream
T0 T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15
Power Level -1
Power Level -2
Power Level -3
Acclerator Level


Master North Control
NVSB-NW
Double Buffered Filter
STT-RAM N
Global Scratch-pad
Host
Represntation Learning
DRAM
Double Buffered IF Map
P-SUM
W
Arbiter
2xNVSB
P-MUX
Tile-Q
NVSB-SE
Double Buffered OF Map
South Control
STT-RAM S
Activation
Normalize & Pooling
Micro-profiler
Power Predictor
Examplar


From IF or NVSB of failed PE
Filter
4x4 STiles w/ 8x8 SA each
128 cycles (64 x16)x 16bits Results
o-Path De-Mux
Next Tile
Output
To arbiter & Fabric
i-Path mux
8x8 Priority Mux
Filter
Next STile
Wr-Q
128 Cycles
4kB NVSB
Prev STile
i-Path mux
Next STile
Power off Warning from Predictor
512B NVSB
if FULL
Work-Q Logic


clk
w-pdown
state
on
pwr warning (>256 cycles)
off
backup
dat
Compute
path
tile
arbiter & fabric
fab-state
!pdown


clk
w-pdown
state
on
pwr warning (>256 cycles)
off
backup
dat
Compute
path
tile
arbiter & fabric
fab-state
!pdown


DynAgent
u-Profiler
Read/Write ()
Airthmetic ()
I/O ()
EH Trace
Database
Power Predictor
1
4
Platform-1
2
3
1
4
Platform-2
2
3
Energy Trace
DB
Candidate Networks
Num Ops
Accuracy
Latency
Selected Network
DynNAS+
DynFit & DynInfer
Software
Middleware & Compiler
Power Atomic Assembly Code
P1
P2
L1
C1
C2
Cn
L3
T1
T2
T3
T4
T5
Lb
Lr
Voltage Monitor
Power Emergency
LEA Accelerator
Hardware Supported Monitoring, Backup and Restore
Job ****
Job ****
Job ****
. . .
ret 2 //T=task_next
ret -1 //pwr emgncy
task_Conv2D(ID#1)
ret <IP * MASK>
task_Conv2D(ID#2)
ret <IP * MASK>
task_Conv2D(ID#3)
ret <IP * MASK>
BackUp(#3)
Rdf4mFRAM(#3)
task_Conv2D(ID#3)
ret <IP * MASK>
ret 1 //Coreset
ret 2 //T=task_next
task_Conv2D(ID#7)
ret <IP * MASK>
Coreset(OP#7,IS)
.. ret IS#
Communicate(IS#)
... sendData()

```
User Program
Def Infer(input,model)
   ...
Def Coreset(IP,Algo)
   ...
Def Communicate(Data)
```

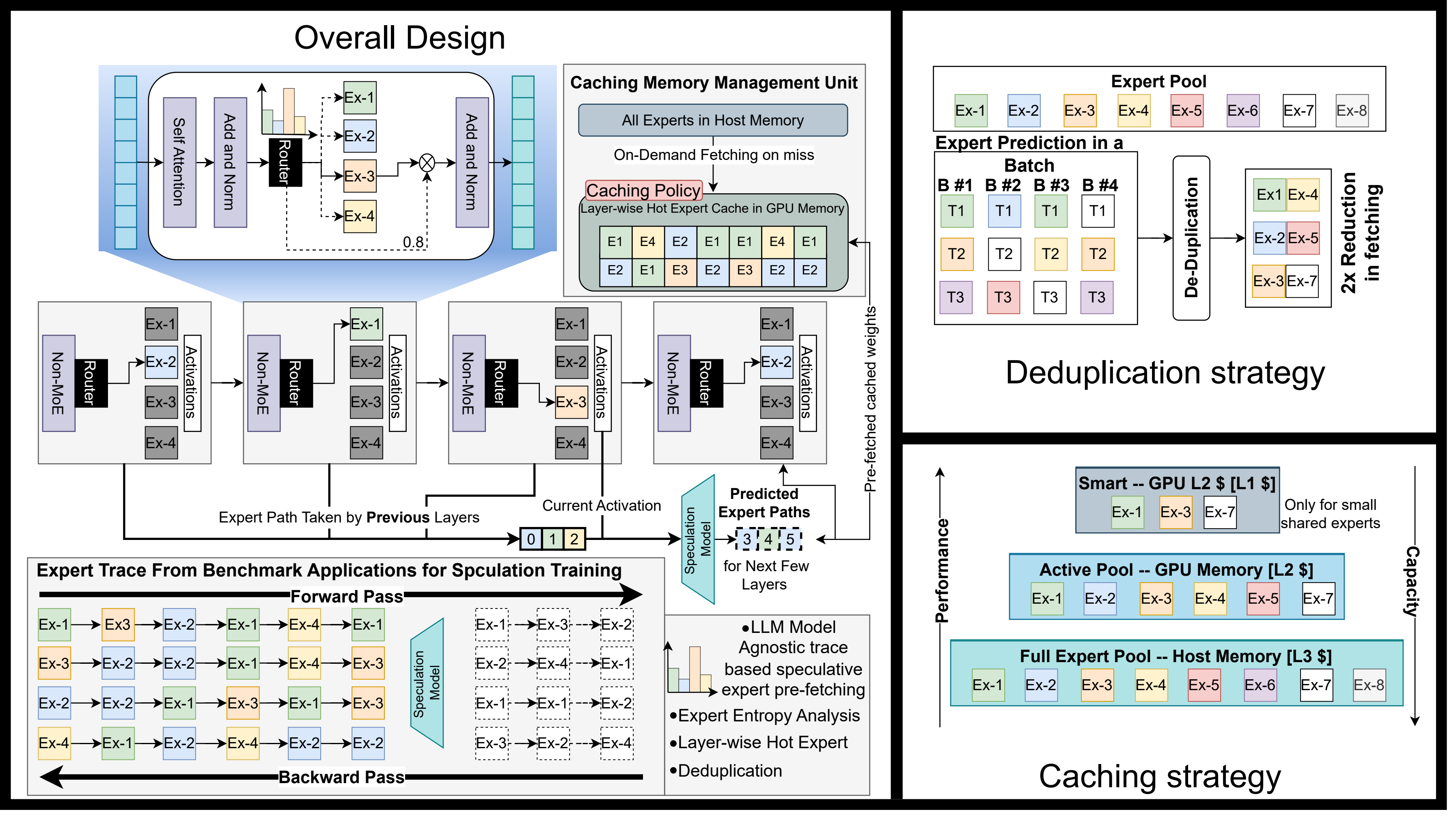
```
Int Pred_Pwr(task_next)
   ...
   ret 2 //T=task_next
   ret 1 //save to coreset
   ret 0 //backup
   ret -1 //pwr emgncy
```

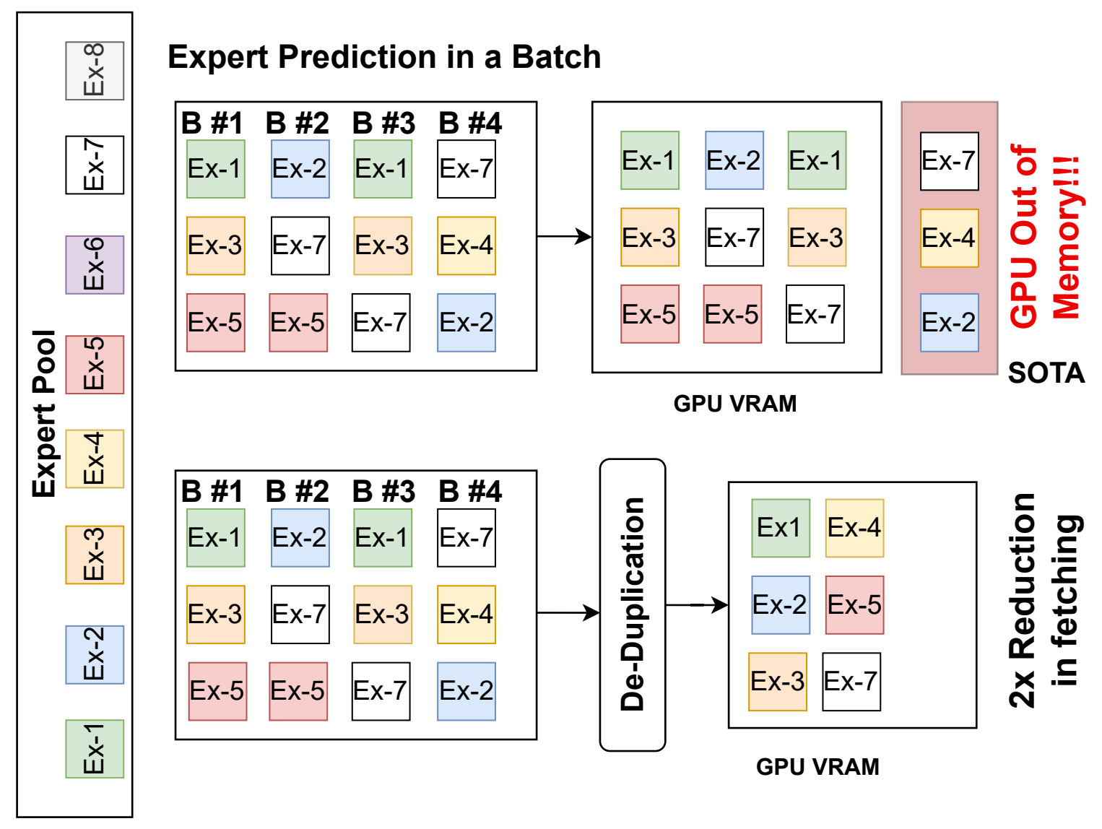
```
backUp(task, data)
   ...
  Write2FeRAM()
  ReStore(task)
  ...
Rdf4mFRAM(task)
```

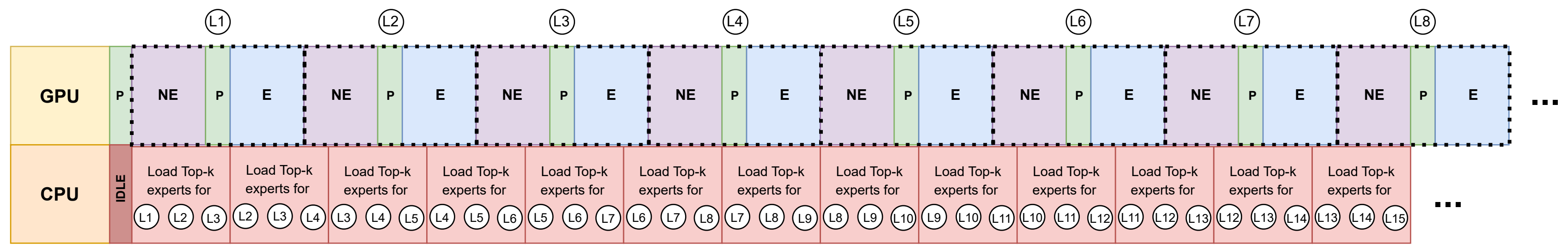
```
job conv2D(SZ_IP, SZ_mask)
    calc_compute();
    calc_memory();
    calc_engy();
    reshape(input, mask)
    ret #task_Conv2D()
```

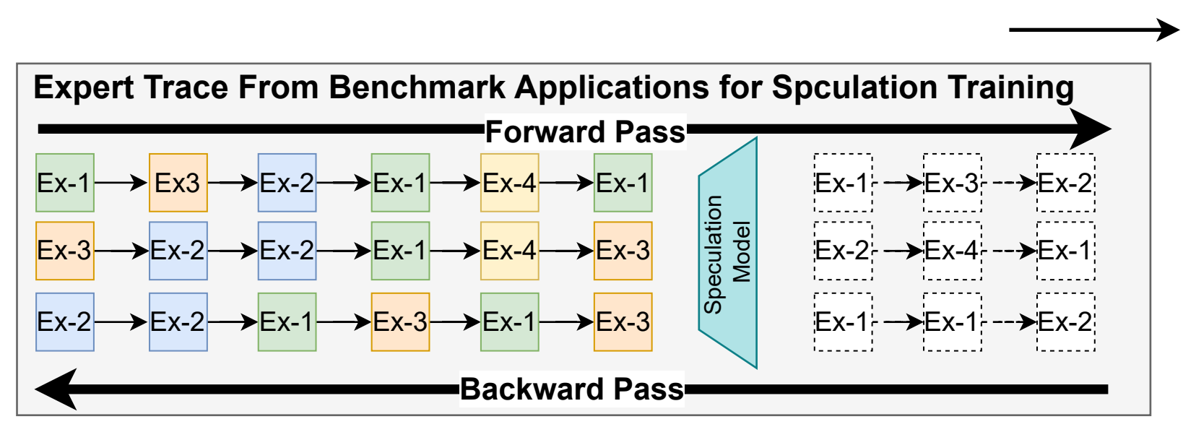
```
job convBN(SZ_IP, SZ_mask)
    calc_compute();
    calc_memory();
    calc_engy();
    reshape(input, mask)
    ret #task_BN()
```

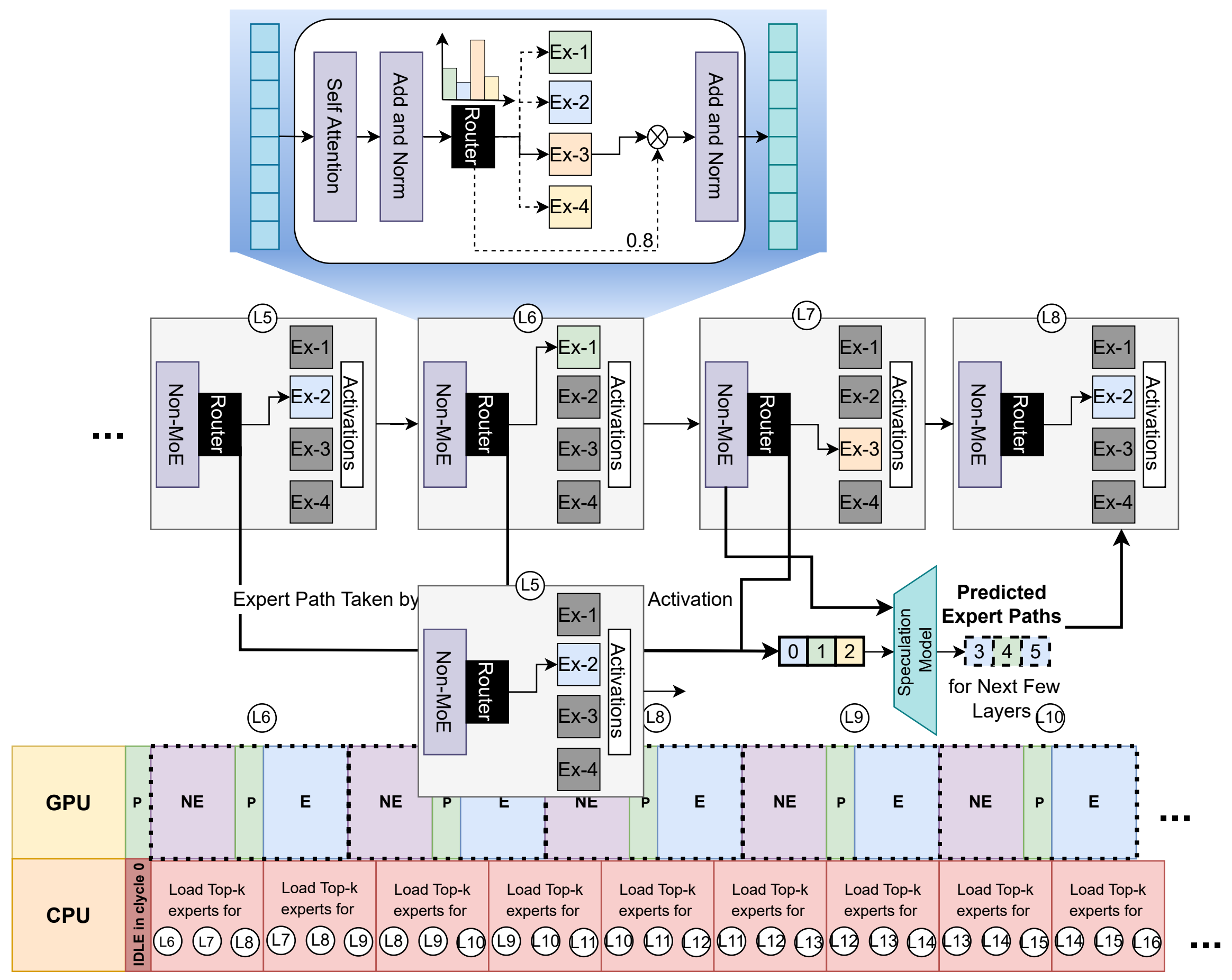
```
HOST-SIDE-Exec...
 Recover(IS#)
 Infer(RData)
```

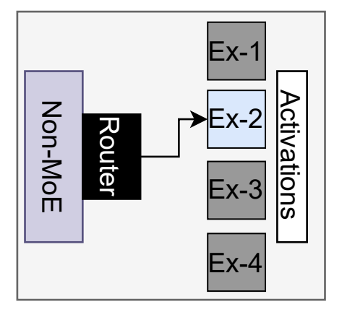

Overall Design
Caching Memory Management Unit
All Experts in Host Memory
On-Demand Fetching on miss
Caching Policy
Layer-wise Hot Expert Cache in GPU Memory
Self Attention
Add and Norm
Router
Ex-1
Ex-2
Ex-3
Ex-4
0.8
Non-MoE
Activations
Pre-fetched cached weights
Expert Path Taken by Previous Layers
Current Activation
Speculation Model
Predicted Expert Paths
for Next Few Layers
Expert Trace From Benchmark Applications for Spculation Training
Forward Pass
Backward Pass
•LLM Model Agnostic trace based speculative expert pre-fetching
•Expert Entropy Analysis
•Layer-wise Hot Expert
•Deduplication
Expert Pool
Expert Prediction in a Batch
B #1
B #2
B #3
B #4
T1
T2
T3
De-Duplication
2x Reduction in fetching
Deduplication strategy
Smart -- GPU L2 $ [L1 $]
Only for small shared experts
Active Pool -- GPU Memory [L2 $]
Full Expert Pool -- Host Memory [L3 $]
Performance
Capacity
Caching strategy

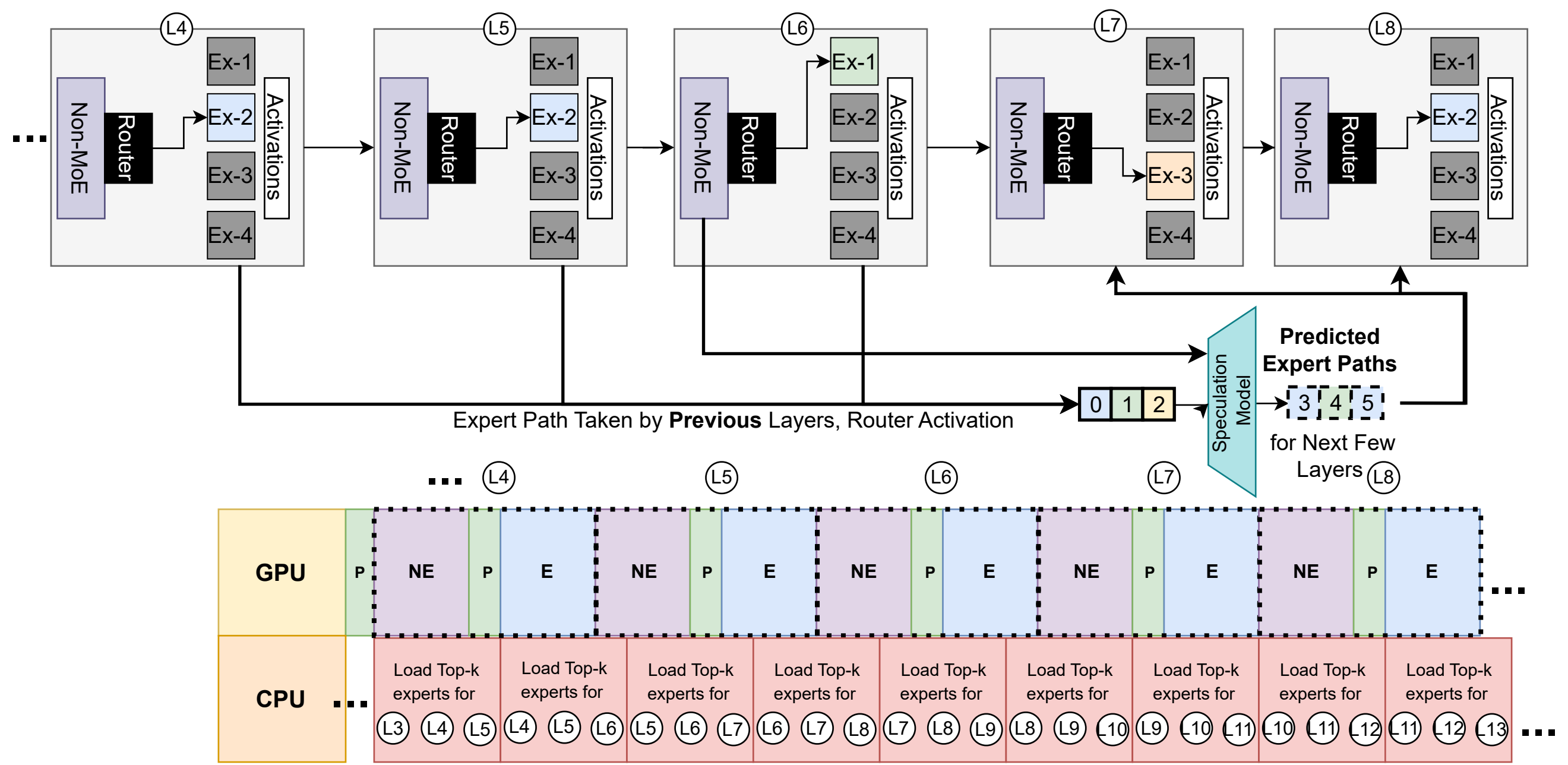

Expert Pool
Ex-1
Ex-2
Ex-3
Ex-4
Ex-5
Ex-6
Ex-7
Ex-8
Expert Prediction in a Batch
B #1
B #2
B #3
B #4
GPU VRAM
GPU Out of Memory!!!
SOTA
De-Duplication
2x Reduction in fetching

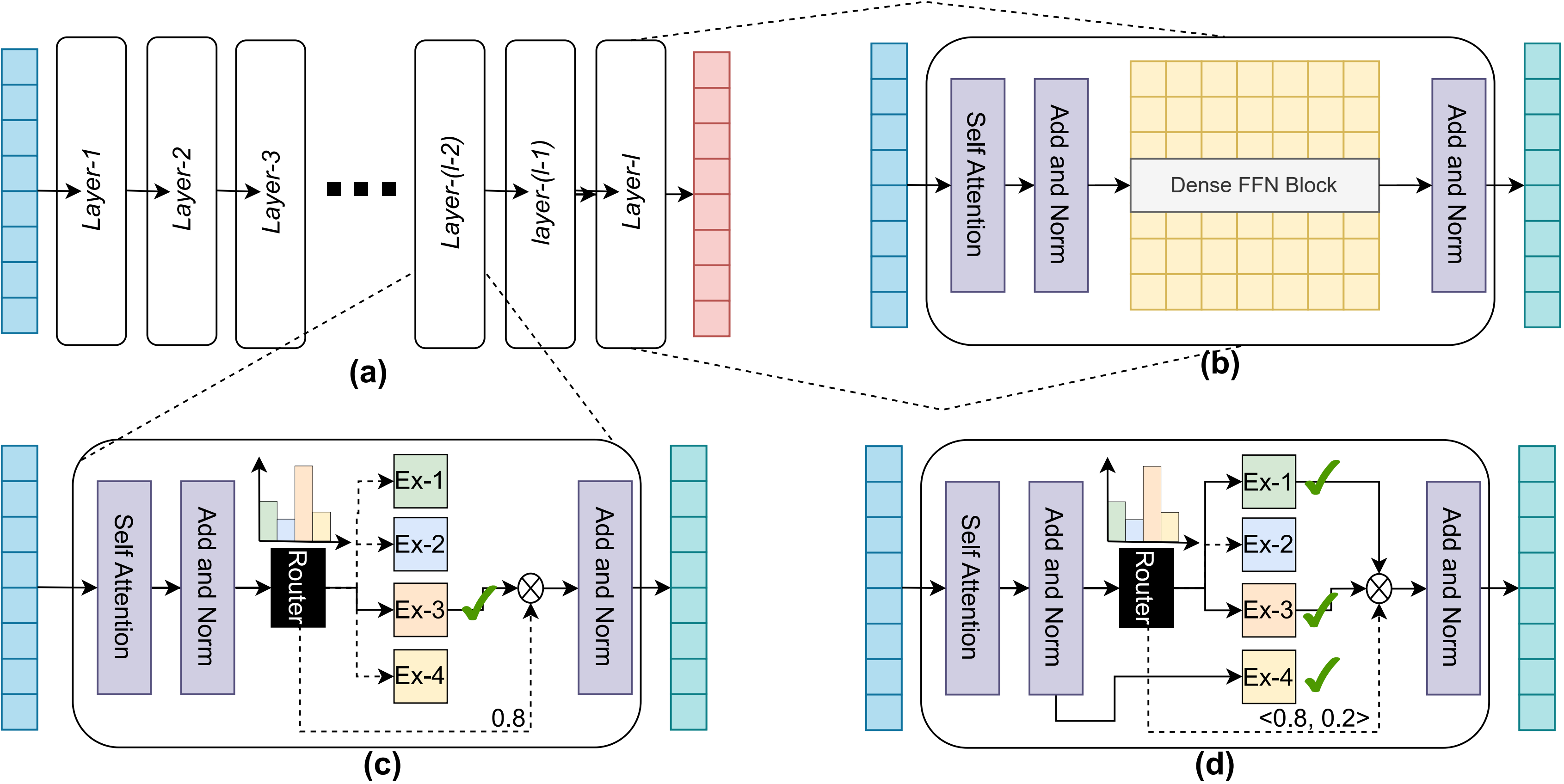

GPU
CPU
IDLE
P
NE
E
Load Top-k experts for
L1
L2
L3
L4
L5
L6
L7
L8
...

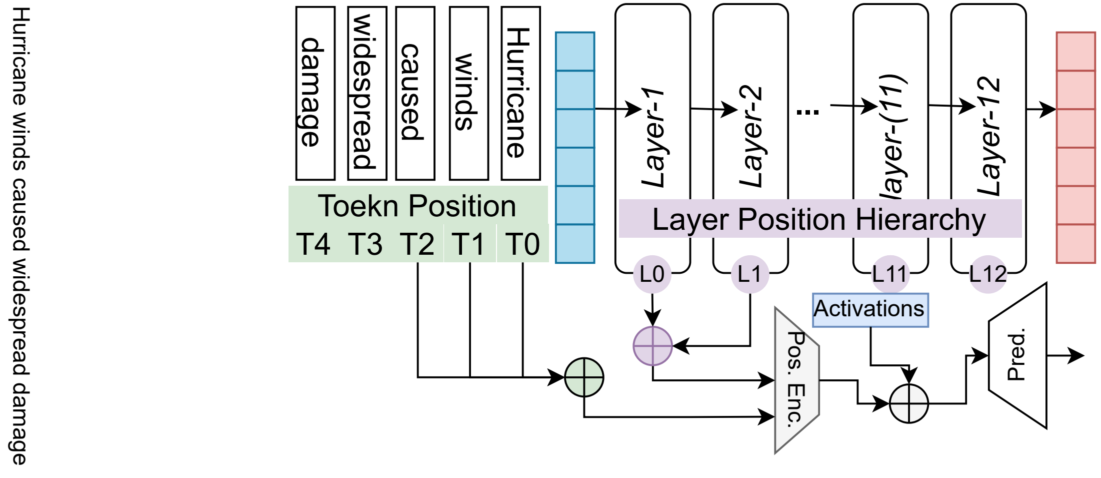

Expert Trace From Benchmark Applications for Spculation Training
Forward Pass
Speculation Model
Backward Pass

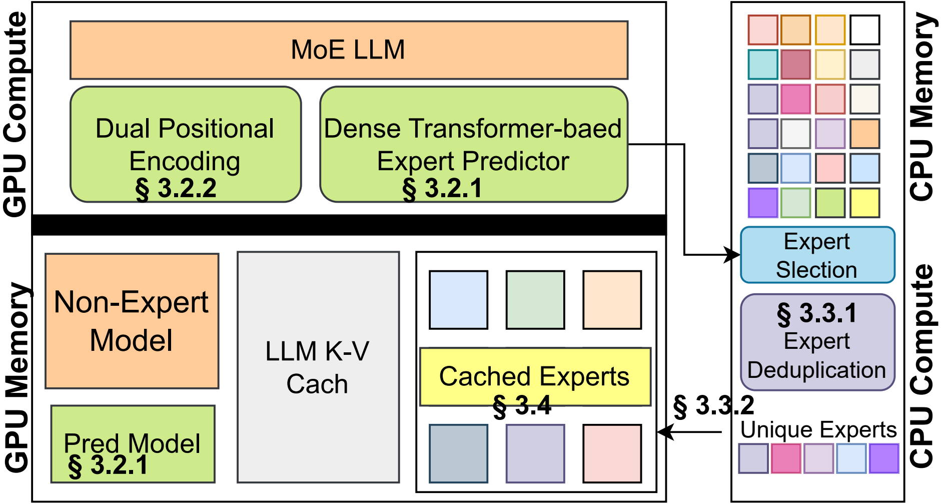

Self Attention
Add and Norm
Router
Ex-1
Ex-2
Ex-3
Ex-4
0.8
Non-MoE
Activations
L5
L6
L7
L8
Expert Path Taken by
Activation
Speculation Model
Predicted Expert Paths
for Next Few Layers
GPU
CPU
P
NE
E
Load Top-k experts for
...


Non-MoE
Router
Ex-1
Ex-2
Ex-3
Ex-4
Activations


L4
L5
L6
L7
L8
Non-MoE
Router
Ex-1
Ex-2
Ex-3
Ex-4
Activations
Expert Path Taken by Previous Layers, Router Activation
Speculation Model
Predicted Expert Paths
for Next Few Layers
GPU
CPU
P
NE
E
Load Top-k experts for
...


Layer-1
Layer-2
Layer-3
Layer-(l-2)
layer-(l-1)
Layer-l
(a)
Self Attention
Add and Norm
Dense FFN Block
Add and Norm
(b)
Self Attention
Add and Norm
Router
Ex-1
Ex-2
Ex-3
Ex-4
0.8
Add and Norm
(c)
Self Attention
Add and Norm
Router
Ex-1
Ex-2
Ex-3
Ex-4
<0.8, 0.2>
Add and Norm
(d)


Hurricane winds caused widespread damage
damage
widespread
caused
winds
Hurricane
Toekn Position
T4
T3
T2
T1
T0
Layer-1
Layer-2
...
Layer-(11)
Layer-12
Layer Position Hierarchy
L0
L1
L11
L12
Activations
Pos. Enc.
Pred.


GPU Compute
MoE LLM
Dual Positional Encoding
§ 3.2.2
Dense Transformer-baed Expert Predictor
§ 3.2.1
GPU Memory
Non-Expert Model
Pred Model
§ 3.2.1
LLM K-V Cach
Cached Experts
§ 3.4
CPU Memory
Expert Slection
§ 3.3.1
Expert Deduplication
§ 3.3.2
Unique Experts
CPU Compute