KAKURENBO: Adaptively Hiding Samples in Deep Neural Network Training

Truong Thao Nguyen(⁑), Balazs (Bali) Gerofi(⸙)(‡),
Edgar Josafat Martinez-Noriega(⁑) , Francois Trahay(⟊), Mohamed Wahib(‡)

(⁑) National Institute of Advanced Industrial Science and Technology (AIST), Japan

(⟊) Télécom SudParis, Institut Polytechnique de Paris, France

(‡) RIKEN Center for Computational Science, Japan

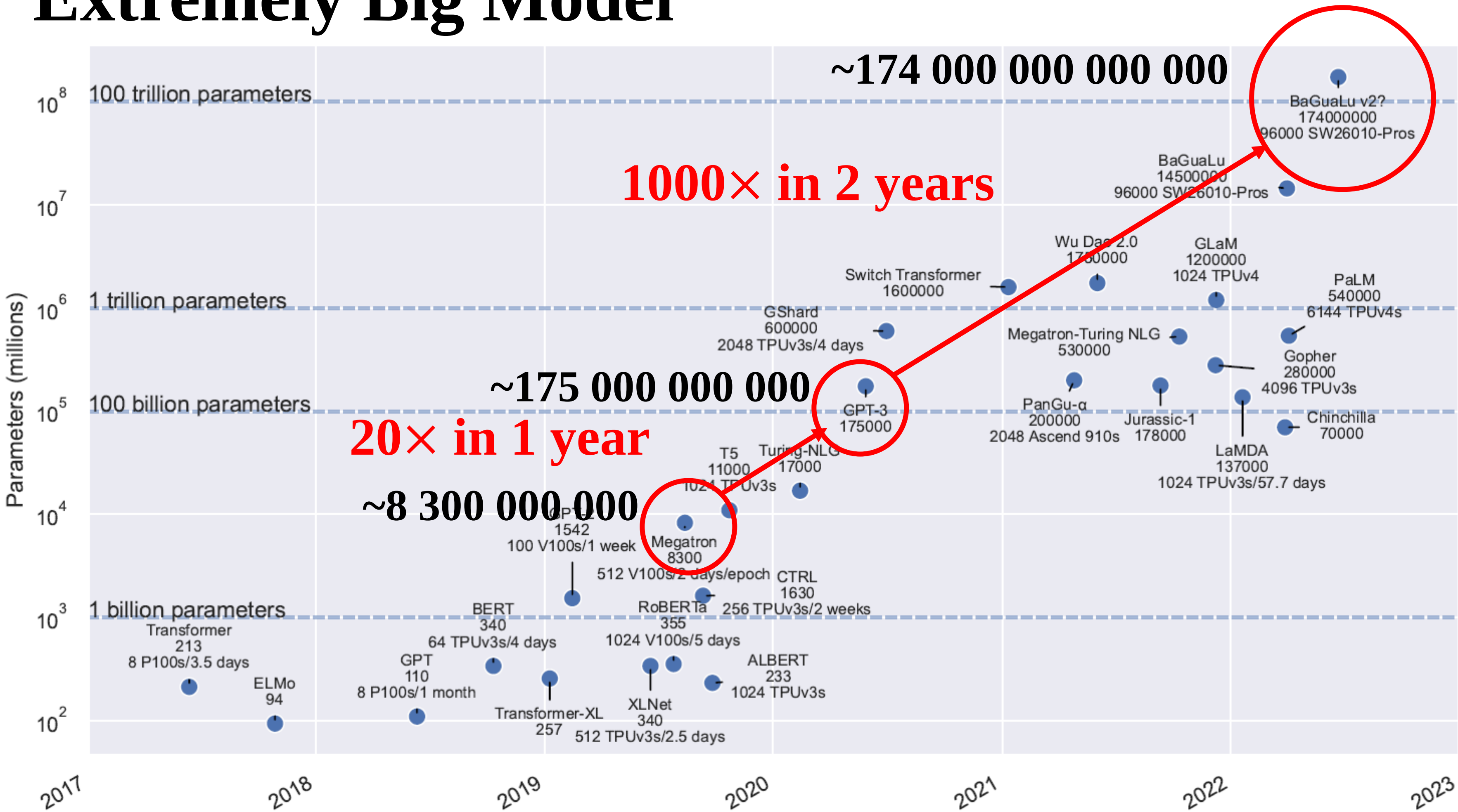
(⸙) Intel Corporation USA

Overview

- Explosion of Deep Learning (DL)
  - Effectiveness in a variety of applications
- Training big model on a large dataset become a trend
  - Example: <u>pre-trained Attention-based Models + large dataset</u>
- Drawback:
  - Long training time/cost
    - T5 ($1.3M), GPT-3 ($4.6M) AlphaGo ($35M)
  - Stress non-compute parts of supercomputers
    - Enormous pressure on the I/O subsystem

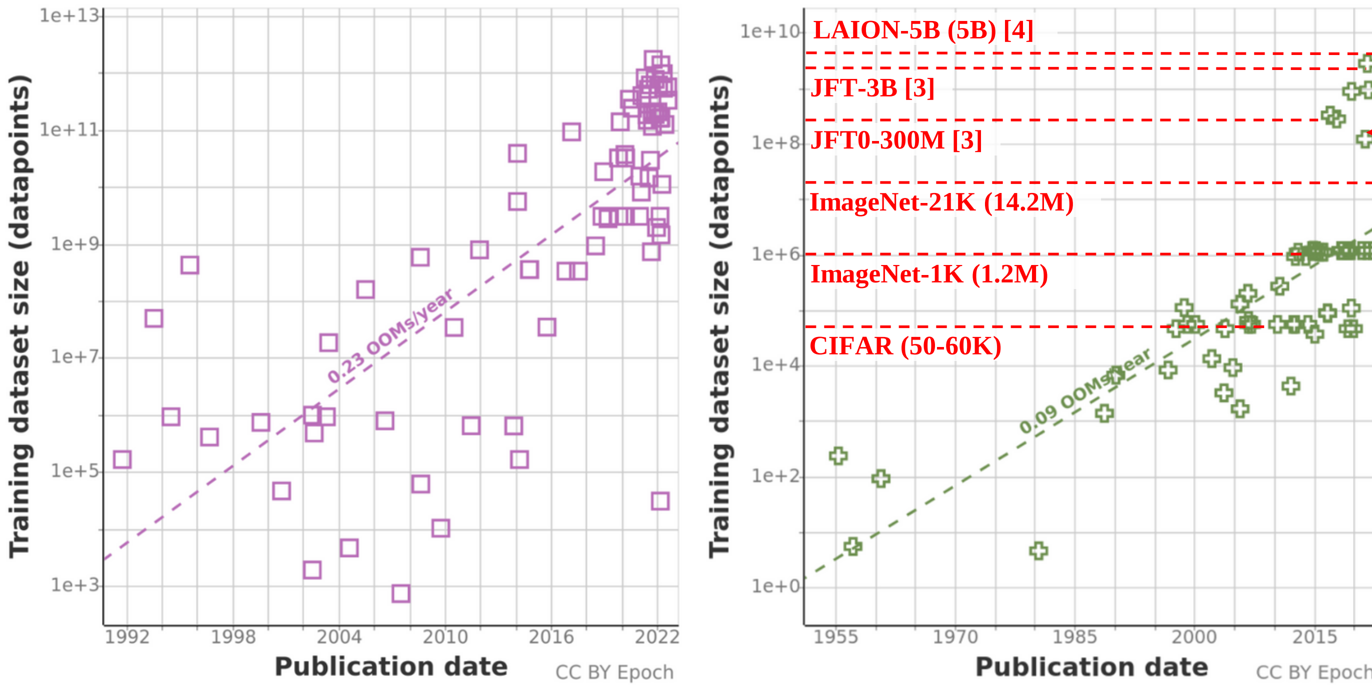
➔ **Our Target**: **Accelerating DNN training**, i.e., reducing training time and cost, while maintaining the accuracy

Extremely Big Model



Credit: Mohamed Wahib@RIKEN

Trends in Training Dataset Sizes



Synthetic datasets,
e.g., Fractal (100M) [2]

Training process becomes longer and more costly

Training datasets for language (left) and vision (right) [1]

Our Target: **Accelerating DNN training**, i.e., reducing training time and cost, while maintaining the accuracy

[1]
[2]
Vi
[3]
[4]
Systems 35 (2022)

Research Approaches

Biased with-replacement sampling

- not all samples are of the same importance
- Training with more importance samples can lead to faster convergence

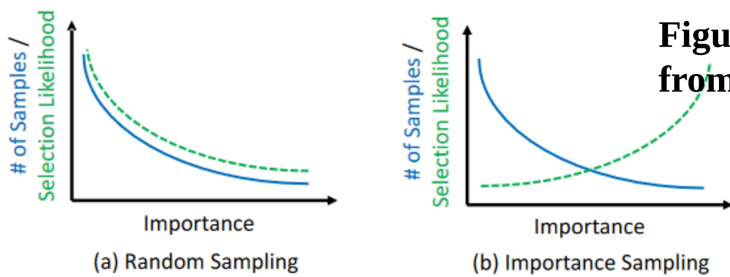


Figure from [5]

Figure 4: Comparison between (a) random sampling and (b) importance sampling.

Data pruning

- Prune selected samples from dataset
- Same accuracy when training with pruned dataset

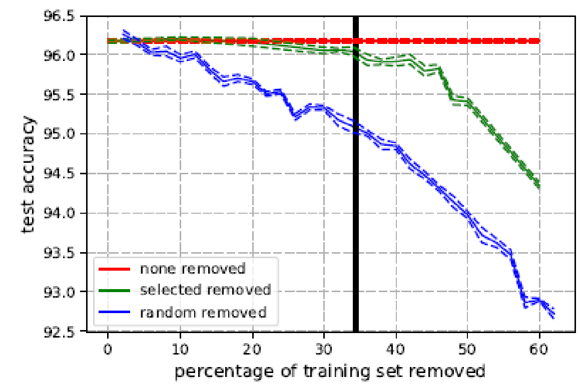
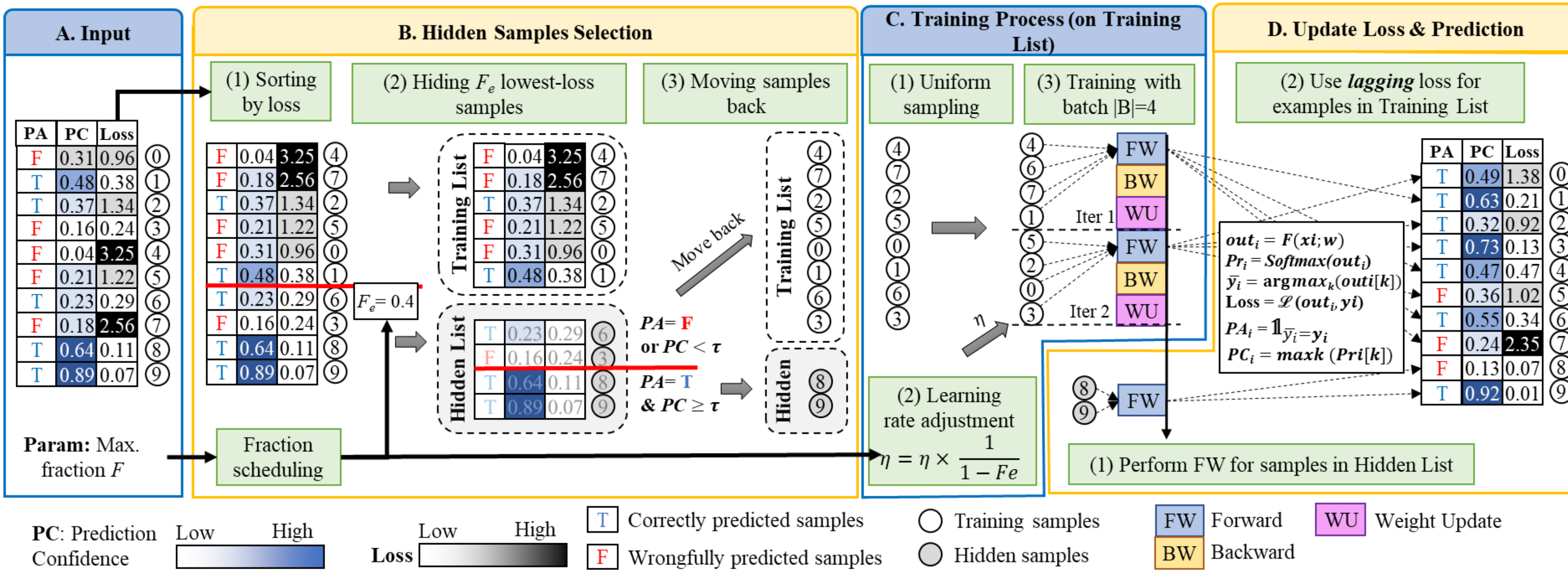


Figure from [6]

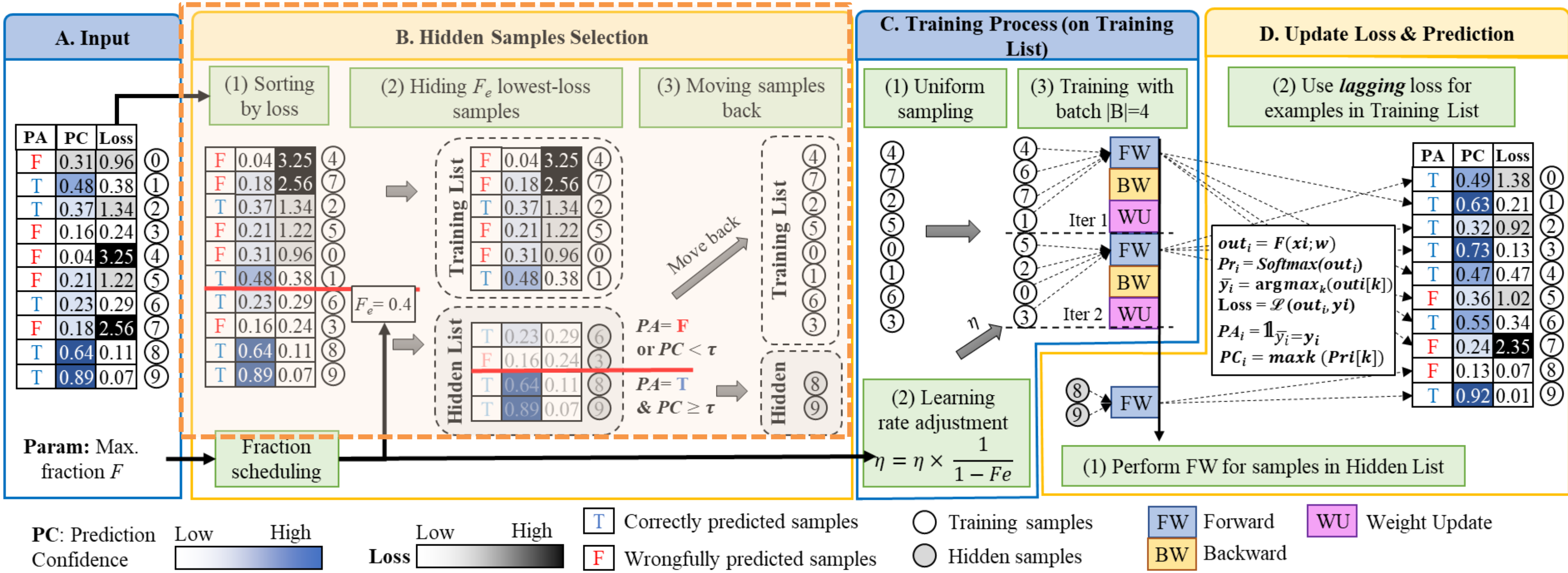
Real-time, adaptively, exclude samples with the least impact from the dataset during the training (Hiding samples)

[5] Zeng, Xiao, Ming Yan, and Mi Zhang. "Mercury: Efficient on-device distributed dnn training via stochastic importance sampling." In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, pp. 29-41. 2021.
[6] Toneva, Mariya, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. "An empirical study of example forgetting during deep neural network learning." arXiv preprint arXiv:1812.05159 (2018).

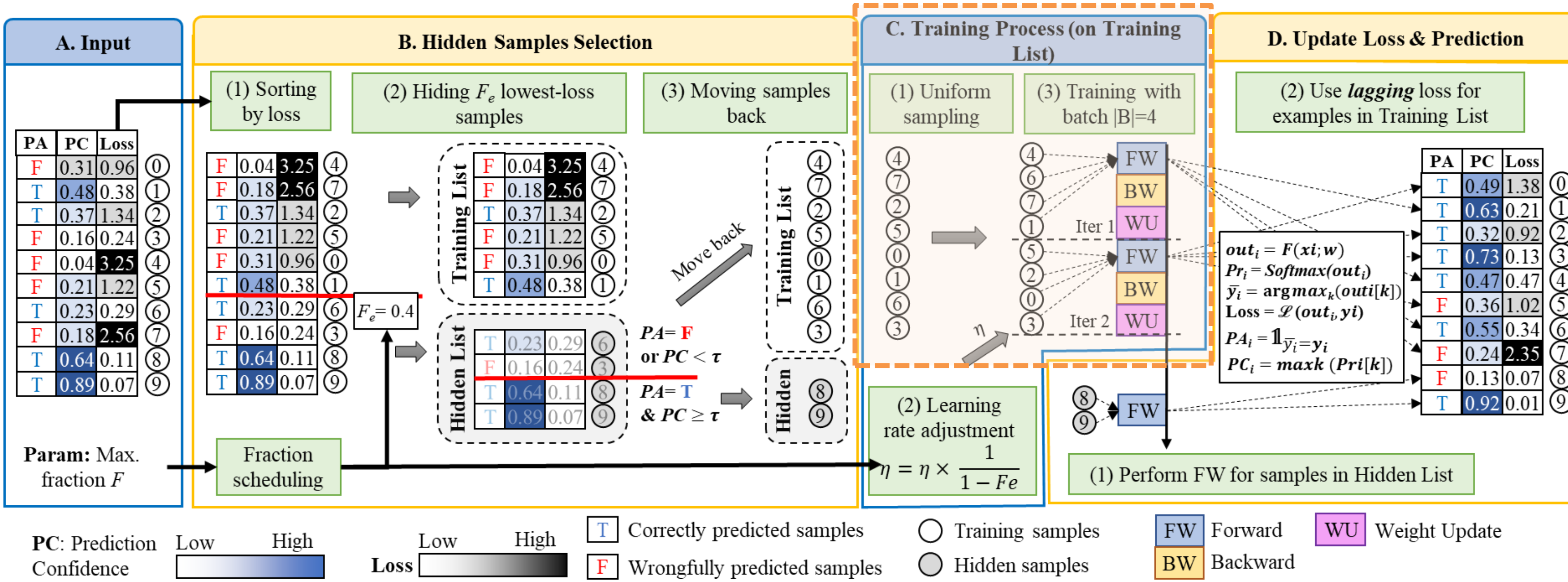
KAKURENBO


A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use *lagging* loss for examples in Training List
PA PC Loss
F_e= 0.4
Training List
Hidden List
Move back
PA= F or $PC < \tau$
PA= T & $PC \geq \tau$
Hidden
Iter 1
Iter 2
FW
BW
WU
η
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1-Fe}$
$out_i = F(xi; w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = \mathrm{argmax}_k(outi[k])$
$\mathrm{Loss} = \mathscr{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\ (Pri[k])$
(1) Perform FW for samples in Hidden List
Param: Max. fraction F
Fraction scheduling
PC: Prediction Confidence
Low
High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

KAKURENBO

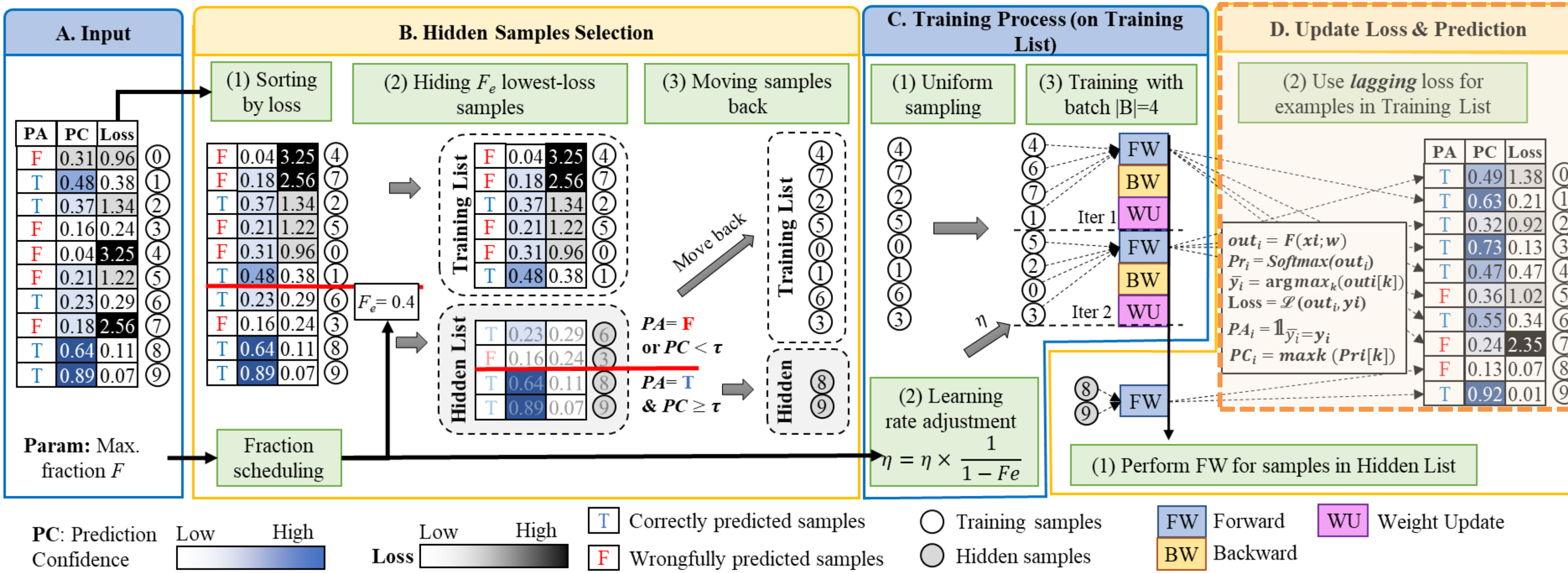

A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
(1) Perform FW for samples in Hidden List
Fraction scheduling
Param: Max. fraction F
$F_e = 0.4$
Training List
Hidden List
Move back
PA= F or $PC < \tau$
PA= T & $PC \geq \tau$
Hidden
Iter 1
Iter 2
η
FW
BW
WU
PA
PC
Loss
$out_i = F(xi; w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = \mathrm{argmax}_k(outi[k])$
$\mathrm{Loss} = \mathcal{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\,(Pri[k])$
PC: Prediction Confidence
Low
High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

KAKURENBO



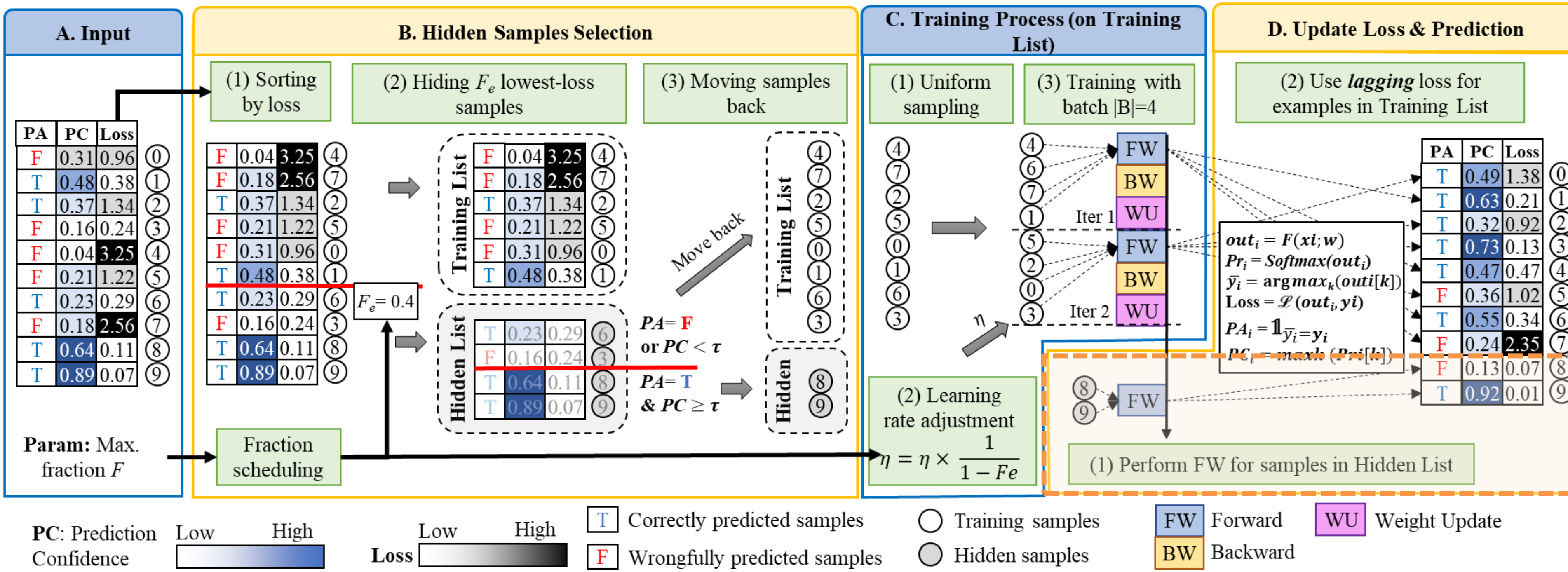
A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
PA
PC
Loss
Training List
Hidden List
Move back
Hidden
F_e= 0.4
PA= F or PC < τ
PA= T & PC ≥ τ
Iter 1
Iter 2
FW
BW
WU
η
$out_i = F(xi;w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = argmax_k(outi[k])$
$Loss = \mathscr{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\ (Pri[k])$
Param: Max. fraction F
Fraction scheduling
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
(1) Perform FW for samples in Hidden List
PC: Prediction Confidence
Low
High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

KAKURENBO

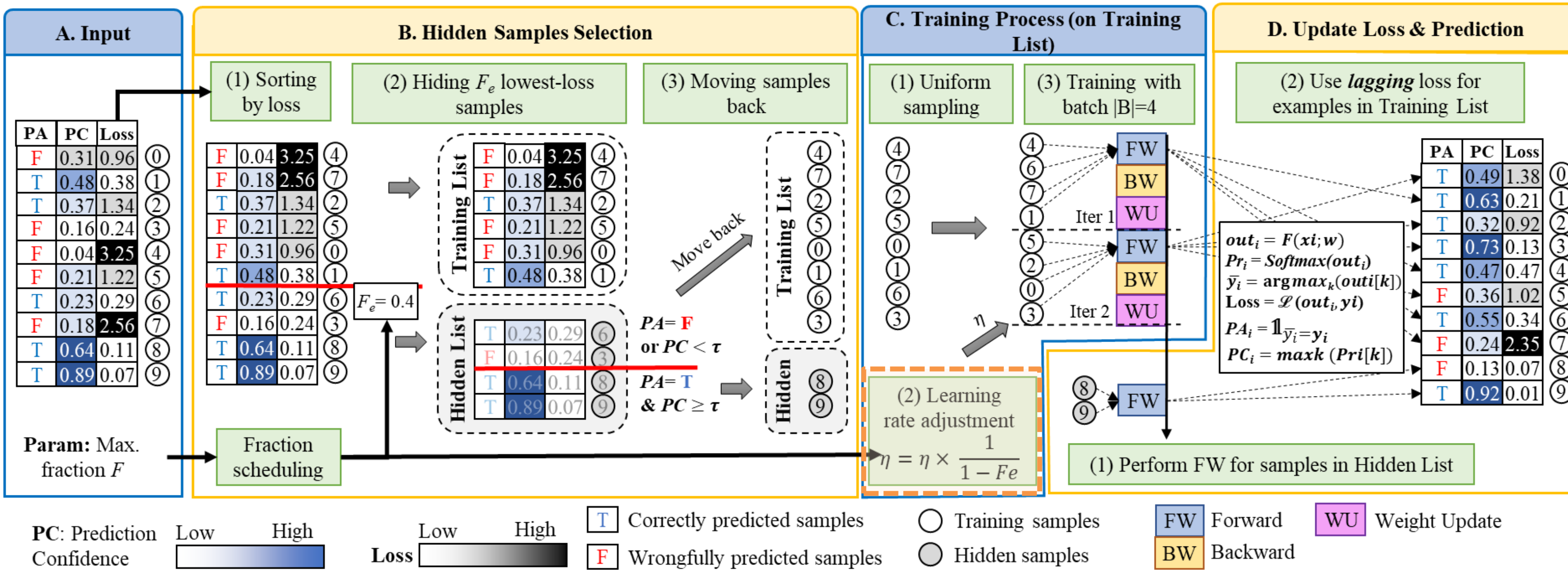


A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
PA PC Loss
Param: Max. fraction F
Fraction scheduling
F_e = 0.4
Training List
Hidden List
Move back
PA= F or PC < τ
PA= T & PC ≥ τ
Hidden
Iter 1
Iter 2
FW
BW
WU
η
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
(1) Perform FW for samples in Hidden List
$out_i = F(xi; w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = argmax_k(outi[k])$
$Loss = \mathcal{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\ (Pri[k])$
PC: Prediction Confidence
Low
High
Loss
Correctly predicted samples
Wrongfully predicted samples
Training samples
Hidden samples
Forward
Backward
Weight Update

KAKURENBO

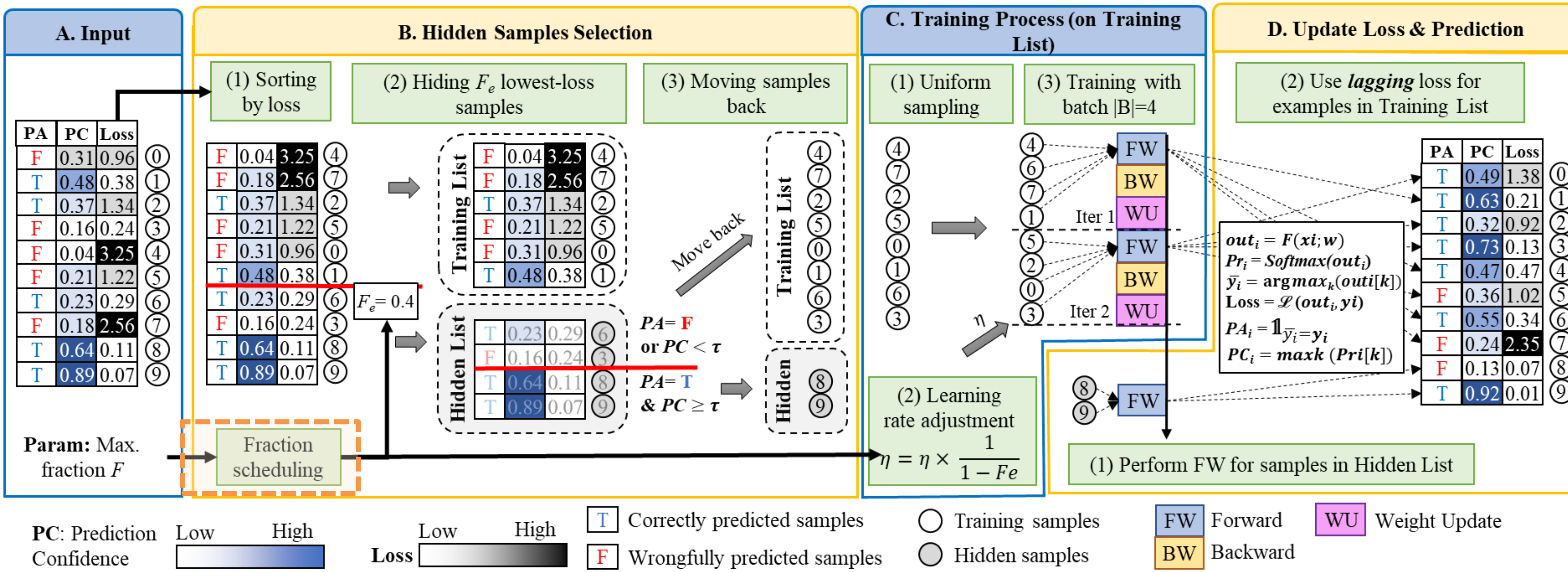

A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
PA PC Loss
Param: Max. fraction F
Fraction scheduling
F_e= 0.4
Training List
Hidden List
Move back
PA= F or PC < τ
PA= T & PC ≥ τ
Hidden
Iter 1
Iter 2
FW
BW
WU
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
$out_i = F(xi;w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = \mathrm{argmax}_k(outi[k])$
$\mathrm{Loss} = \mathcal{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
(1) Perform FW for samples in Hidden List
PC: Prediction Confidence
Low High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

KAKURENBO



A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
PA PC Loss
Training List
Hidden List
Move back
Hidden
F_e= 0.4
PA= F or PC < τ
PA= T & PC ≥ τ
Iter 1
Iter 2
FW
BW
WU
Param: Max. fraction F
Fraction scheduling
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
(1) Perform FW for samples in Hidden List
$out_i = F(xi; w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = \mathrm{argmax}_k(outi[k])$
$Loss = \mathcal{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\ (Pri[k])$
PC: Prediction Confidence
Low
High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

KAKURENBO



A. Input
B. Hidden Samples Selection
C. Training Process (on Training List)
D. Update Loss & Prediction
(1) Sorting by loss
(2) Hiding F_e lowest-loss samples
(3) Moving samples back
(1) Uniform sampling
(3) Training with batch |B|=4
(2) Use lagging loss for examples in Training List
PA
PC
Loss
Training List
Hidden List
Move back
Hidden
F_e= 0.4
PA= F or PC < τ
PA= T & PC ≥ τ
Iter 1
Iter 2
FW
BW
WU
η
$out_i = F(xi; w)$
$Pr_i = Softmax(out_i)$
$\bar{y}_i = argmax_k(outi[k])$
$Loss = \mathscr{L}(out_i, yi)$
$PA_i = \mathbb{1}_{\bar{y}_i = y_i}$
$PC_i = maxk\ (Pri[k])$
Param: Max. fraction F
Fraction scheduling
(2) Learning rate adjustment
$\eta = \eta \times \frac{1}{1 - Fe}$
(1) Perform FW for samples in Hidden List
PC: Prediction Confidence
Low
High
Loss
T Correctly predicted samples
F Wrongfully predicted samples
Training samples
Hidden samples
FW Forward
BW Backward
WU Weight Update

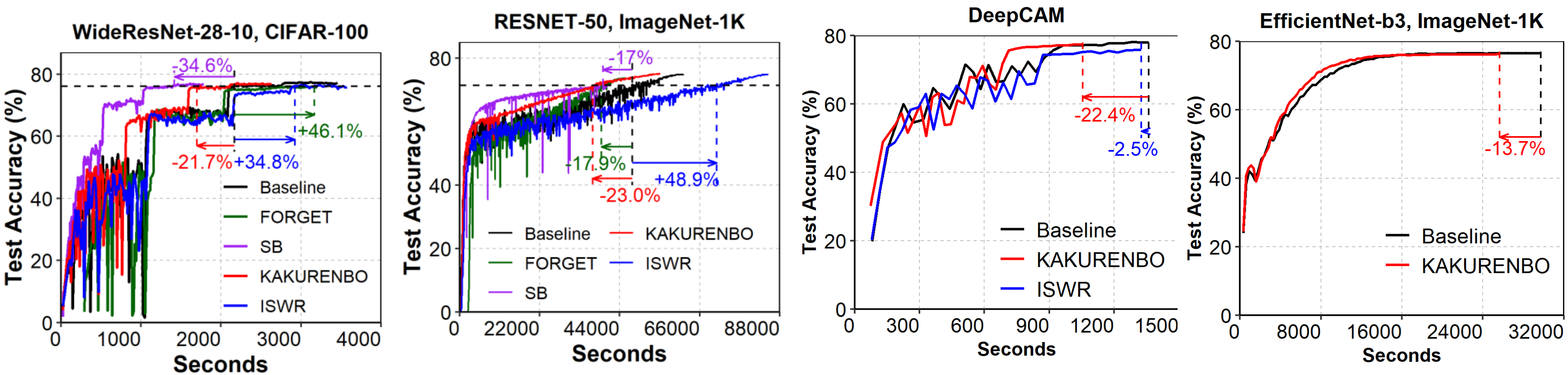
Evaluation Setting

- Strategies:
  - Baseline
  - ISWR: Importance Sampling with Replacement
  - FORGET: pruning technique
  - SB: Selective Backprop
  - KAKURENBO (ours)
- Datasets and Models
- Default setting:
  - F=30%,
  - Threshold τ=0.7

Table 7: Datasets and Models Used in Experiments (* Down-stream training using the pre-trained model).

Model	Dataset	#Samples	#Epoch	#GPUs	minibatch (per GPU)	Task
Resnet50 [36]	ImageNet-1K [19]	1.2M	100	32	64	Image Classification
EfficientNet-b3 [37]					32	
WideResNet-28-10 [39]	CIFAR-100 [41]	50K	200	32	32	Image Classification
DeepCAM [4]	DeepCAM [4]	∼ 122K	35	1024	1	Image Segmentation
DeiT-Tiny-224 [42]	Fractal-3K [6]	3M	80	32	16	Image Classification
	(*) CIFAR-10 [41]	50K	1000	8	96	
	(*) CIFAR-100 [41]	50K	1000	8	96	

Evaluation Results



Setting	CIFAR-100 WRN-28-10		ImageNet-1K ResNet-50		ImageNet-1K EfficientNet-b3		DeepCAM	
	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.
Baseline	77.49		74.89		76.63		78.14	
ISWR	76.51	(-0.98)	74.91	(+0.02)	N/A		75.75	(-2.39)
FORGET	76.14	(-1.35)	73.70	(-1.20)	N/A		N/A	
SB	77.03	(-0.46)	71.37	(-3.52)	N/A		N/A	
KAKURENBO	77.21	(-0.28)	75.15	(+0.26)	76.23	(-0.5)	77.42	(-0.9)

Max testing accuracy (Top-1) in percentage of KAKURENBO in the comparison with those of the Baseline and other SOTA methods. Diff. represent the gap to the Baseline.

Impact of KAKURENBO in Transfer Learning

Model: DeiT-Tiny-224
Pretrained with Fractal-3K (3Millions of images)

	Dataset	Metrics	Baseline	ISWR	FORGET	SB	KAKUR.
Up stream	Fractal-3K	Loss	3.26	3.671	3.27	4.18	3.59
		Time (min)	623	719	533	414	529
		Impr.	-	(+15.4%)	(-14.4%)	(-33.5%)	(-15.1%)
Down stream	CIFAR-10	Acc. (%)	95.03	95.79	95.85	93.59	95.28
		Diff.	-	(+0.76)	(+0.82)	(-1.44)	(+0.25)
	CIFAR-100	Acc. (%)	79.69	79.62	79.95	76.98	79.35
		Diff.	-	(-0.07)	(+0.26)	(-2.71)	(-0.34)