Curriculum Learning with Dynamic Instance Hardness & Neural Networks Memorization

TIANYI ZHOU, SHENGJIE WANG, JEFF A. BILMES, MELODI LAB@UW ECE

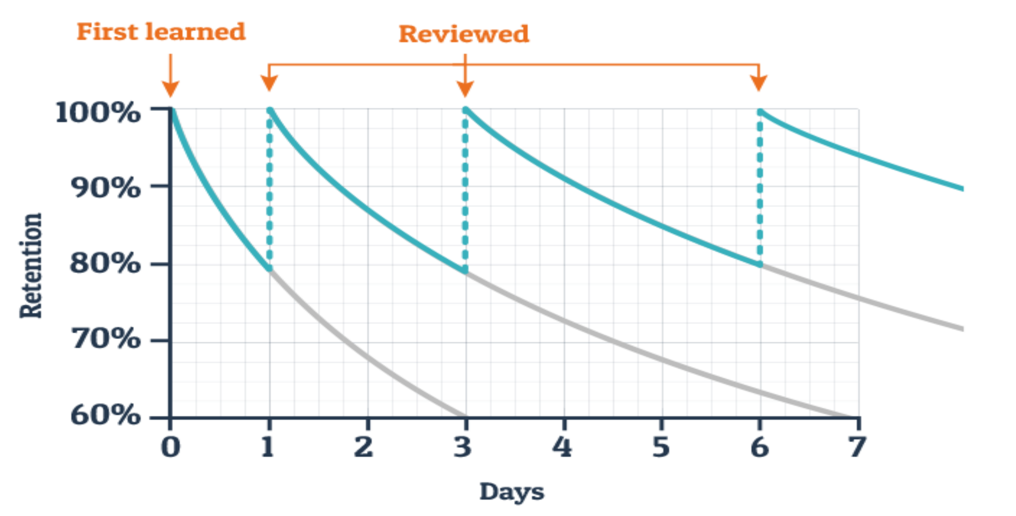
MELODI MachinE Learning, Optimization, & Data Interpretation @ UW

CONIX RESEARCH CENTER

Dynamic Instance Hardness and Memorization

Human forget things fast, but repeatedly revisiting the same things helps learning and improves the persistence of memory.

Typical Forgetting Curve for Newly Learned Information



How to measure the memorization of ML models (e.g., Deep Neural Nets) on each sample? Dynamic Instance Hardness!

iteration t. We define dynamic instance hardness (DIH) as a running average over any instantaneous instance hardness, computed recursively as follows:

$$r_{t+1}(i) = \begin{cases} \gamma \times a_t(i) + (1-\gamma) \times r_t(i) & \text{if } i \in S_t \\ r_t(i) & \text{else} , \end{cases} \quad (1)$$

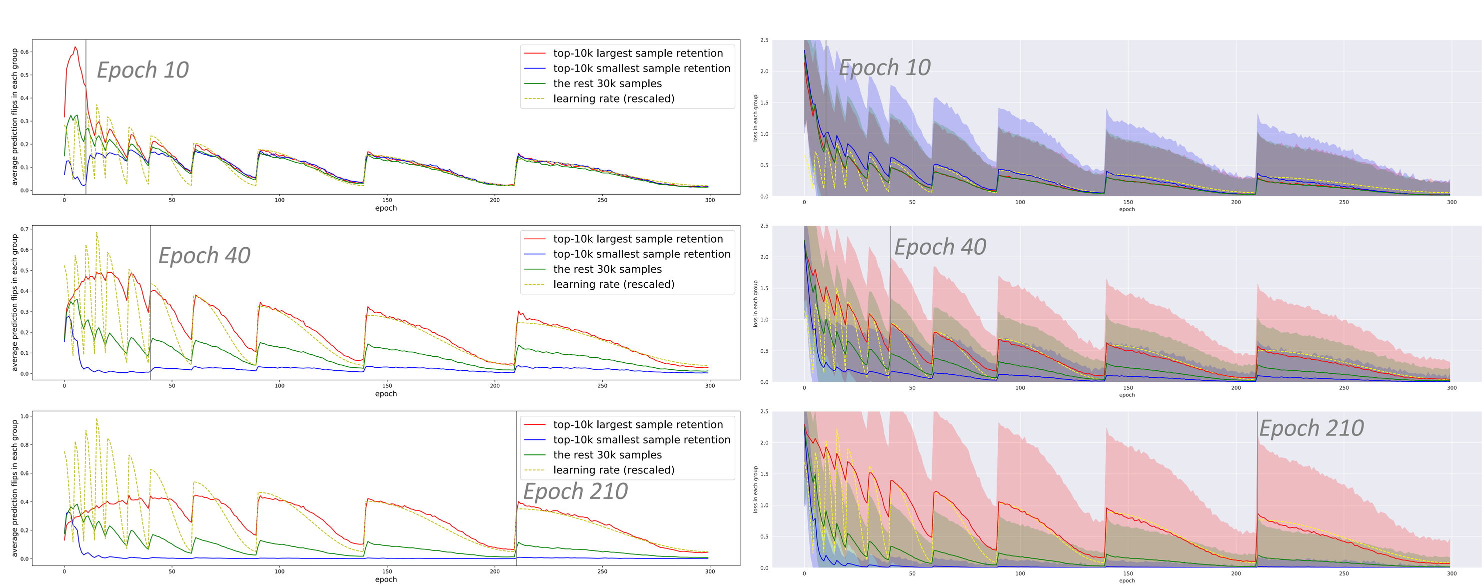
(A) Loss $\ell(y_i, F(x_i; w_t))$, where $\ell(\cdot,\cdot)$ is the loss function and $F(\cdot; w)$ is the model with parameters w;

(B) Loss change $|\ell(y_i, F(x_i; w_t)) - \ell(y_i, F(x_i; w_{t-1}))|$ between two consecutive time steps;

(C) Prediction flip $|\mathbb{1}[\hat{y}_i^t = y_i] - \mathbb{1}[\hat{y}_i^{t-1} = y_i]|$, where $\hat{y}_i^t$ is the prediction of sample i in step t, e.g., $\mathrm{argmax}_j\, F(x_i; w_t)[j]$ for classification.

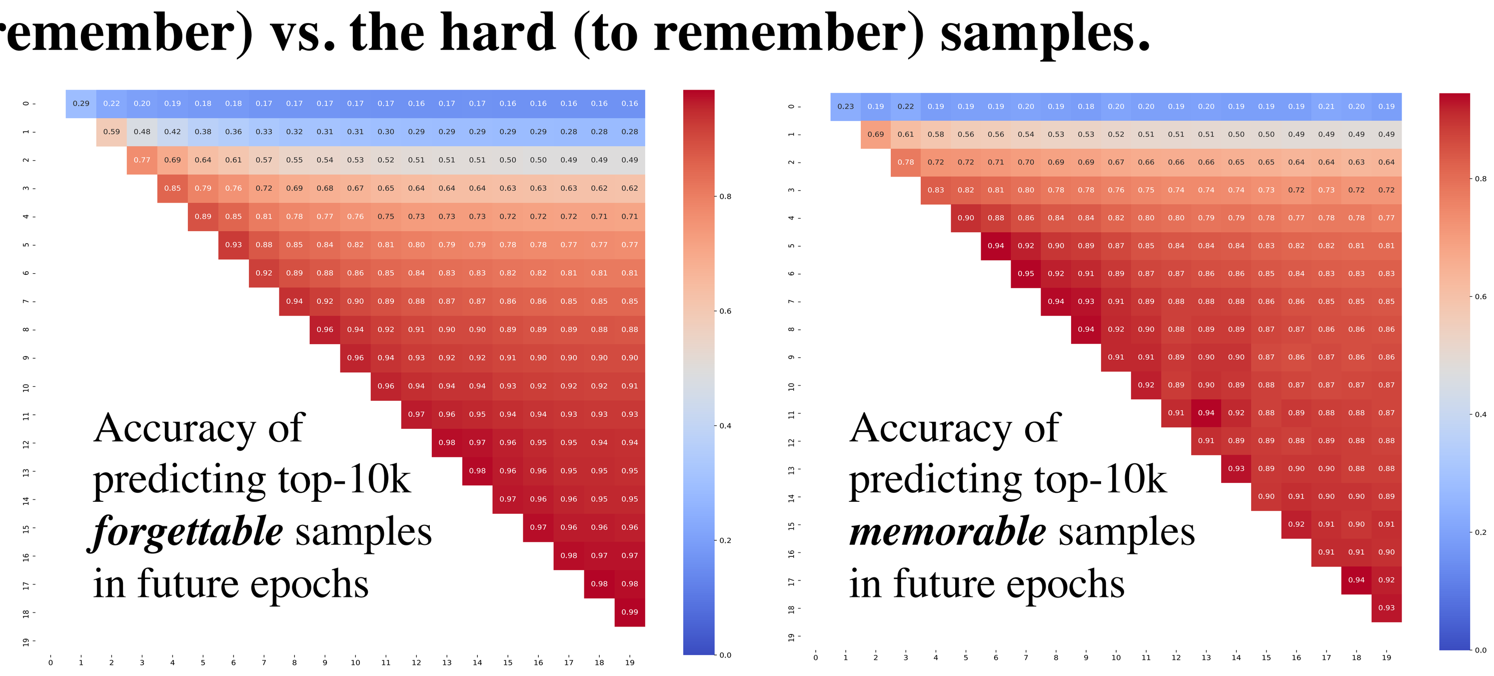
Three Observations of Dynamic Instance Hardness

***Observation I*: DNNs have very different training dynamics on samples with small and large DIH.**



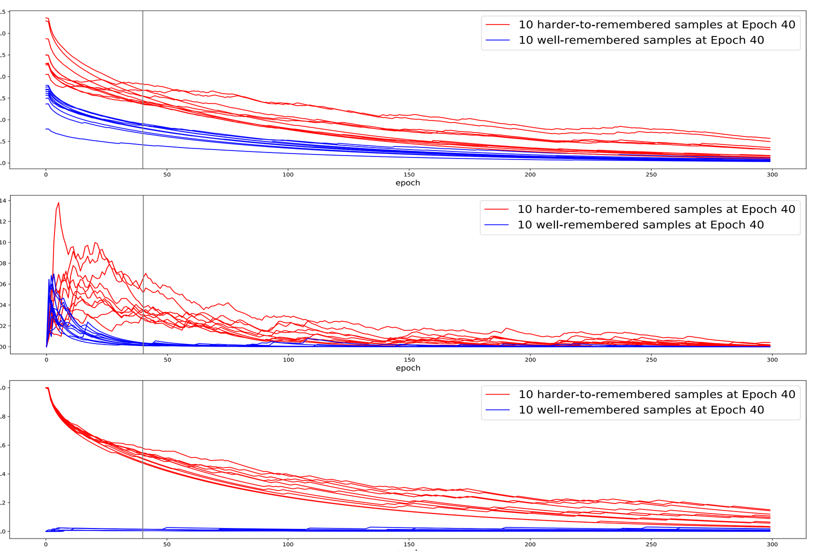
- We split the training set into three groups at epoch 10/40/210, according to **DIH** computed over history.
- The plots show how the prediction flip (LEFT) and loss (RIGHT) of samples from the three groups change during training.

***Observation II*: DIH in early epochs suffices to identify the easy (to remember) vs. the hard (to remember) samples.**



- LEFT: We compute the overlap between the top-10k group with the **largest r** at epoch i and j for every i and every j > i.
- RIGHT: We compute the overlap between the top-10k group with the **smallest r** at epoch i and j for every i and every j > i.

***Observation III*: DIH metrics decrease during training for both easy and hard samples:**
Neural nets improves its memorization on all data during training.

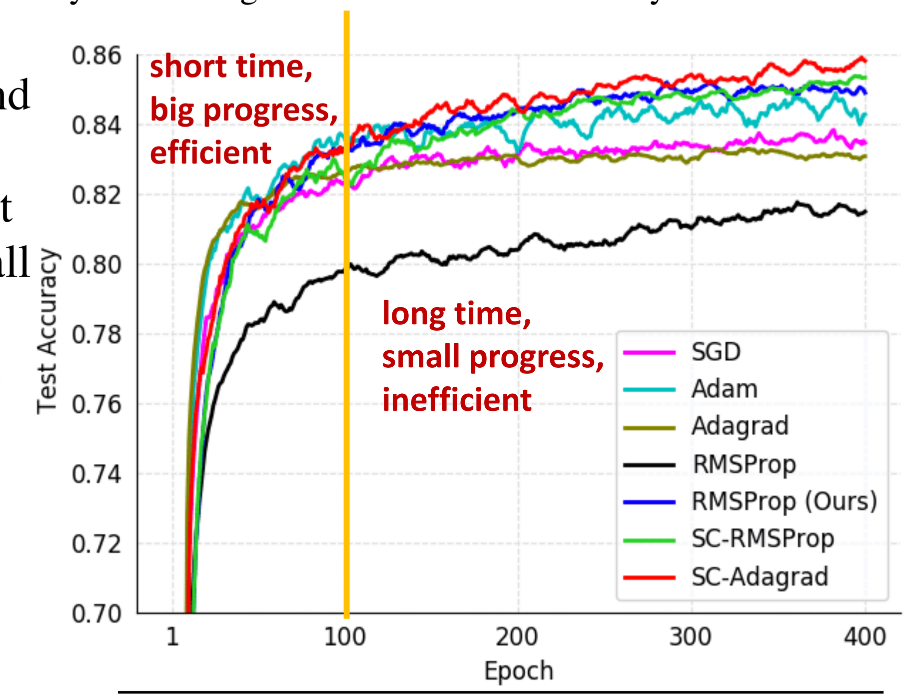


Curriculum Learning for better Memorization

A free curriculum to train ML models:

- **Idea:** train ***forgettable*** samples more frequently and spend less efforts on ***memorable*** samples.
- **Efficiency:** the 1st curriculum learning method that does not require an extra forward propagation on all data to determine the training set for the next step.
- **Robustness:** DIH changes smoothly comparing to instantaneous feedback such as loss.
- **Provable:** we can formulate the problem of optimizing a curriculum as an online optimization of an unknown diminishing return (submodular) function under mild assumptions, and derive the near-optimality guarantee.

Why do learning curves of ML models always look like this?



Empirical advantages:

- **Converge faster in early-stage:** achieve reasonably good performance in a shorter time.
- **Higher final accuracy and better generalization:** avoid overfitting on memorable samples and focus on forgettable samples.
- **More efficient:** we achieve 2-5x speedup empirically. It can reduce communication costs for ML over networks.
- **Simple to implement:** record byproduct of back-propagation to update DIH.

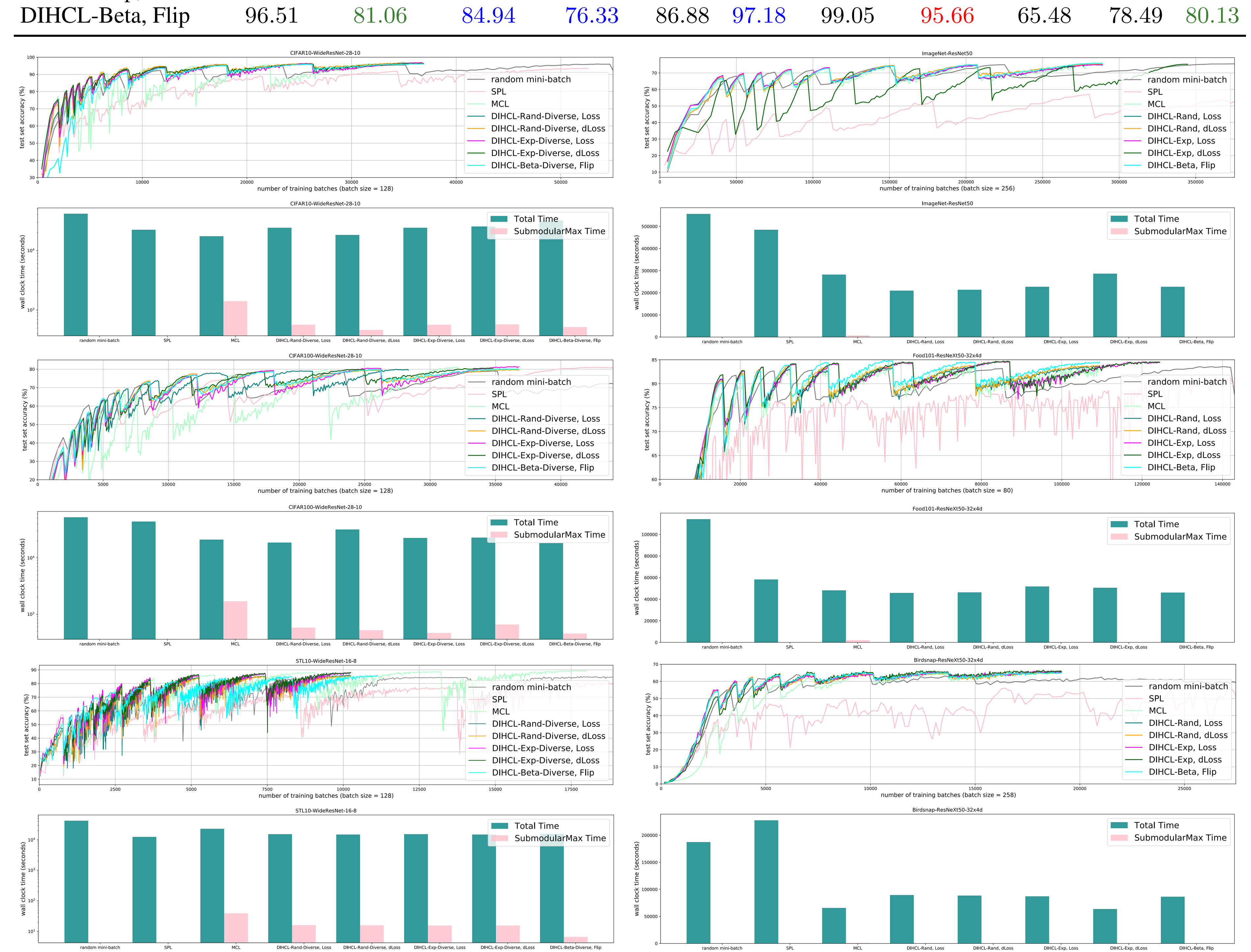
Algorithm 1 DIH Curriculum Learning (DIHCL-Greedy)

input: $\{(x_i, y_i)\}_{i=1}^n, \pi(\cdot;\eta), \eta_{1:T}, \ell(\cdot,\cdot), F(\cdot; w)$; $T, T_0; \gamma, \gamma_k \in [0,1]$
initialize: $w, \eta_1, k_1 = n, r_0(i) = 1\ \forall i \in [n]$
for $t \in \{1, \cdots, T\}$ **do**
  if $t \le T_0$ **then**
    $S_t \leftarrow [n]$;
  else
    Let $S_t = \mathrm{argmax}_{S:|S|=k_t} \sum_{i \in S} r_{t-1}(i)$;
  end if
  Apply optimization $\pi(\cdot;\eta)$ to update model:
$$w_t \leftarrow w_{t-1} + \pi\left(\nabla_w \sum_{i \in S_t} \ell(y_i, F(x_i; w_{t-1})); \eta_t\right)$$
  Compute normalized $a_t(i)$ for $i \in S_t$ using Eq. (5);
  Update DIH $r_{t+1}(i)$ using Eq. (1);
  $k_{t+1} \leftarrow \gamma_k \times k_t$;
end for

Experiments for training Deep Neural Networks

Table 1. The test accuracy (%) achieved by different methods training DNNs on 11 datasets (without pre-training). We use "Loss, dLoss, Flip" to denote the 3 choices of DIH metrics based on (A), (B), and (C) respectively. In all DIHCL variants, we apply lazier-than-lazy-greedy (Mirzasoleiman et al., 2015) for Eq. (6) on all datasets except Food-101, Birdsnap, Aircraft (FGVC Aircraft), Cars (Stanford Cars), and ImageNet. For each dataset, the best accuracy is in blue, the second best is red, and third best green.

Curriculum	CIFAR10	CIFAR100	Food-101	ImageNet	STL10	SVHN	KMNIST	FMNIST	Birdsnap	Aircraft	Cars
Rand mini-batch	96.18	79.64	83.56	75.04	86.06	96.48	98.67	95.22	64.23	74.71	78.73
SPL	93.55	80.25	81.36	73.23	81.33	96.15	97.24	92.09	63.26	68.95	77.61
MCL	96.60	80.99	84.18	75.09	88.57	96.93	99.09	95.07	65.76	75.28	76.98
DIHCL-Rand, Loss	96.76	80.77	83.82	75.41	87.25	96.81	99.10	95.69	65.62	79.00	80.91
DIHCL-Rand, dLoss	96.73	80.65	83.82	75.34	86.93	96.83	99.14	95.64	65.25	79.93	78.70
DIHCL-Exp, Loss	97.03	82.23	84.65	75.10	88.36	96.91	99.20	95.45	66.13	77.68	79.85
DIHCL-Exp, dLoss	96.40	81.42	84.75	75.62	89.41	96.80	99.18	95.50	66.59	79.72	81.48
DIHCL-Beta, Flip	96.51	81.06	84.94	76.33	86.88	97.18	99.05	95.66	65.48	78.49	80.13



ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

TIANYIZH@UW.EDU