A Dynamic Instance Hardness (cont.)

In this section, we conduct two additional empirical studies about DIH on data with noisy labels and a smaller DNN as an extension of the one shown in the main paper.

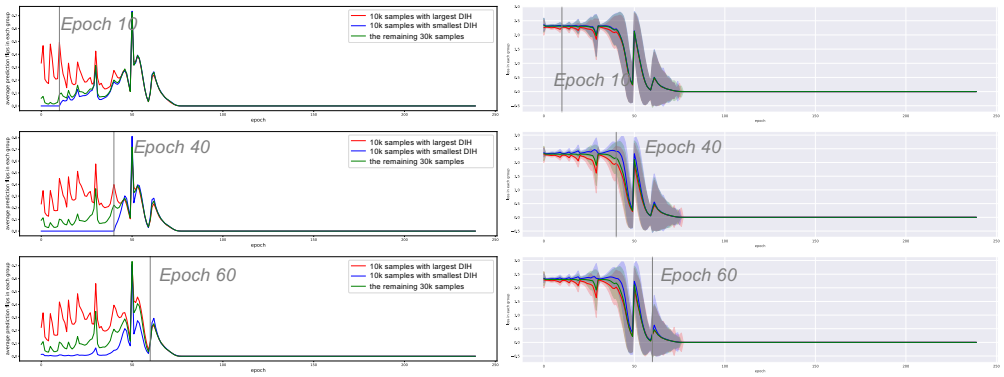

Figure 9: **LEFT:** Averaged prediction-flip and **RIGHT:** losses (mean and std.) of the three groups of samples partitioned by a DIH metric (i.e.,running mean of prediction-flip) computed at epoch 10,40 and 60 during training WideResNet-28-10 on CIFAR10 with **random labels**. In this setting, the random (but wrong) labels will be remembered very well after some training, and DIH in early stages loses the capability to predict the future DIH, i.e., they can only reflect the history but not the future. This characteristic of DIH might be helpful to detect noisy data.

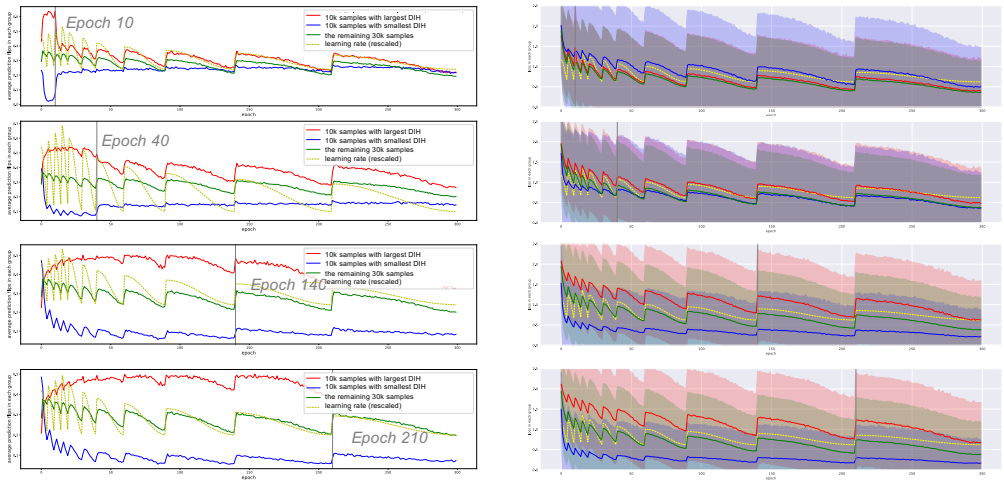

Figure 10: **LEFT:** Averaged prediction-flip and **RIGHT:** losses (mean and std.) of the three groups of samples partitioned by a DIH metric (i.e.,running mean of prediction-flip) computed at epoch 10, 40, 140 and 210 during training **a smaller CNN** on CIFAR10. It shows that the difference of memorable and forgettable samples is not sufficiently obvious until very late training epochs, e.g., after epoch-140.

First, **we conduct an empirical study of dynamic instance hardness during training a neural net on very noisy data**, as studied in [51] and [1]. In particular, we replace the ground truth labels of the training samples by random labels, and apply the same training setting used in Section 2. Then, we compute the running mean of prediction-flip for each sample at some epoch (i.e., 10, 40, 60), and partition the training samples into three groups, as we did to generate Figure 2. The result is shown in Figure 9. It shows 1) the group with the smallest prediction flip over history (left plot) is possible to have large but unchanging loss as shown in the right plot; and 2) the DIH in this case can only reflect the history but cannot predict the future. However, it also indicates that the capability of DIH to predict the future is potential to be an effective metric to distinguish noisy data or adversarial attack from real data. We will discuss it in our future work.

Table 2: Details regarding the datasets and training settings (#Feature denotes the number of features after cropping if applied), "lr_start" and "lr_target" denote the starting and target learning rate for the first episode of cosine annealing schedule, they are gradually decayed over the rest episodes.

Dataset	CIFAR10	CIFAR100	Food-101	ImageNet	STL10	SVHN
#Training	50000	50000	75750	1281167	5000	73257
#Test	10000	10000	25250	50000	8000	26032
#Feature	$(3, 32, 32)$	$(3, 32, 32)$	$(3, 224, 224)$	$(3, 224, 224)$	$(3, 96, 96)$	$(3, 32, 32)$
#Class	10	100	101	1000	10	10
#Epoch T	300	300	400	200	1200	300
BatchSize	128	128	80	256	128	128
lr_start	2×10^{-1}	2×10^{-1}	2×10^{-1}	2×10^{-1}	2×10^{-1}	2×10^{-2}
lr_target	5×10^{-4}	5×10^{-4}	1×10^{-4}	1×10^{-4}	5×10^{-4}	1×10^{-3}

Table 3: Details regarding the datasets and training settings (cont.)

Dataset	Birdsnap	FGVCaircraft	StanfordCARs	KMNIST	FMNIST
#Training	47386	6667	8144	50000	50000
#Test	2443	3333	8041	10000	10000
#Feature	$(3, 224, 224)$	$(3, 224, 224)$	$(3, 224, 224)$	$(1, 28, 28)$	$(1, 28, 28)$
#Class	500	100	196	10	10
#Epoch T	400	400	400	300	300
BatchSize	258	256	256	128	128
lr_start	4×10^{-1}	4×10^{-1}	4×10^{-1}	4×10^{-2}	4×10^{-2}
lr_target	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-3}	1×10^{-3}

Second, **we change the WideResNet to a much smaller CNN architecture with three convolutional layers**[3]. We apply the same training setting used in Section 2. Then, we compute the running mean of prediction-flip for each sample at some epoch (i.e., 10, 40, 140, 210), and partition the training samples into three groups, as we did to generate Figure 2. The result is shown in Figure 10. Compared to DIH of training deeper and wider neural nets shown in Figure 2, the memorable and forgettable samples are indistinguishable until very late stages, e.g., Epoch-140. This indicates that using DIH in earlier stage to select forgettable samples into curriculum might not be reliable when training small neural nets. We will leave explanation of this phenomenon to our future works.

Moreover, we provide a comparison of the smoothness between DIH and instantaneous loss on individual samples in Figure 1. It shows that the DIH is a smooth and consistent measure of the learning/memorization progress on individual samples. In contrast, the frequently used instantaneous loss is much noisier, so selecting training samples according to it will lead to unstable behaviors during training. In Figure 11, we also provide a comparison of DIH and instantaneous loss on the two groups of samples in Figure 4, which shows a similar phenomenon.

B Experiments (cont.)

We use cosine annealing learning rate schedule for multiple episodes. The switching epoch between each two consecutive episode for different datasets are listed below.

- CIFAR10, CIFAR100, SVHN, KMNIST, FMNIST:
  $(5, 10, 15, 20, 30, 40, 60, 90, 140, 210, 300)$;
- STL10: $(20, 40, 60, 80, 120, 160, 240, 360, 560, 840, 1200)$
  $= 4 \times (5, 10, 15, 20, 30, 40, 60, 90, 140, 210, 300)$;
- ImageNet: $(5, 10, 15, 20, 30, 45, 75, 120, 200)$;
- Food-101, Birdsnap, FGVC-Aircraft, StanfordCars:
  $(10, 20, 30, 40, 60, 90, 150, 240, 400) = 2 \times (5, 10, 15, 20, 30, 45, 75, 120, 200)$;

[3]The "v3" network from https://github.com/jseppanen/cifar_lasagne.

We report how the test accuracy changes with the number of training batches for each method, and the wall-clock time for all the 11 datasets in Figure 15-18.

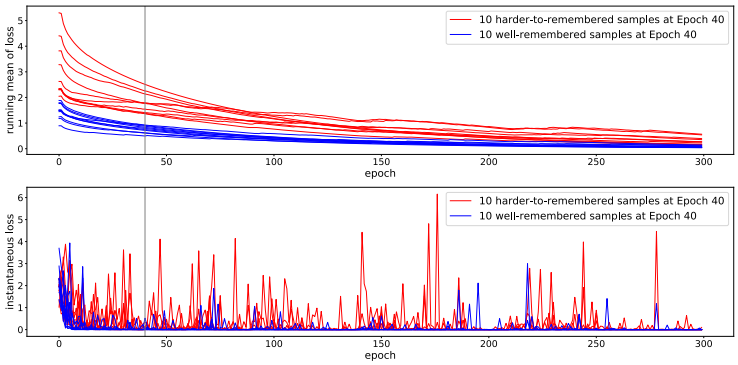

Figure 11: **Top:** DIH (running mean of loss) vs. **Bottom:** instantaneous loss of 10 samples randomly selected from the top 10k samples with the largest(red) and the smallest(blue) DIH at epoch 40 of training of WideResNet-28-10 on CIFAR10 (the same as Figure 4). It shows that for each individual sample from the two groups, DIH smoothly decreases while the corresponding instantaneous loss is much noisier.

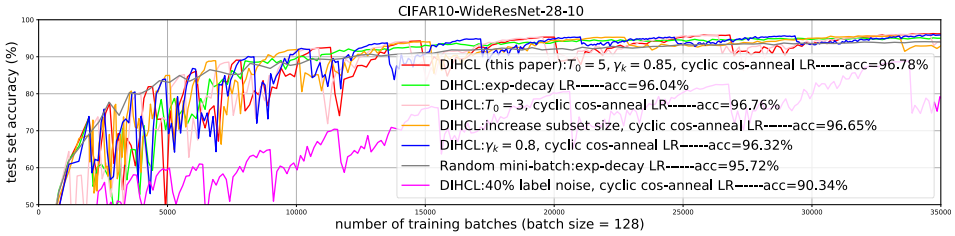

Figure 12: Large Figure 8. Comparison of DIHCL variants for training WideResNet-28-10 on CIFAR10.

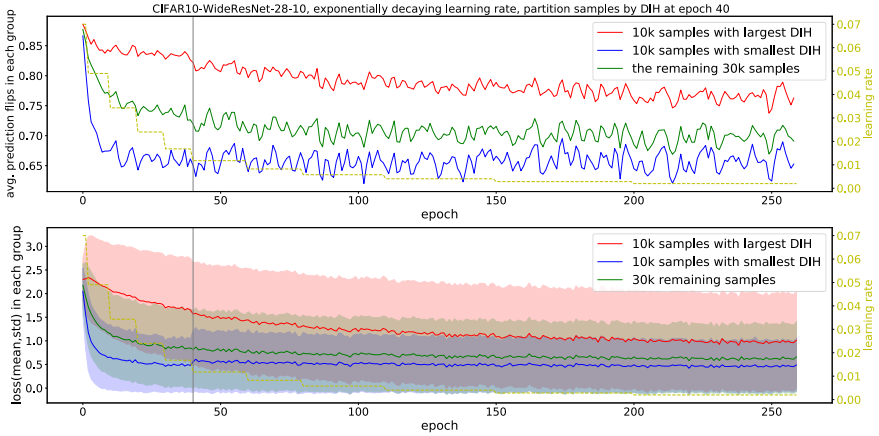

Figure 13: **TOP:** Averaged prediction-flip and **RIGHT:** losses (mean and std.) of the three groups of samples partitioned by a DIH metric (i.e., running mean of prediction flip) computed at epoch 40 when using **exponential decaying learning rate** (instead of cyclic cosine annealing rate) across episodes (cycles). DIH exhibits similar properties on identifying hard and easy samples for neural nets to learn.

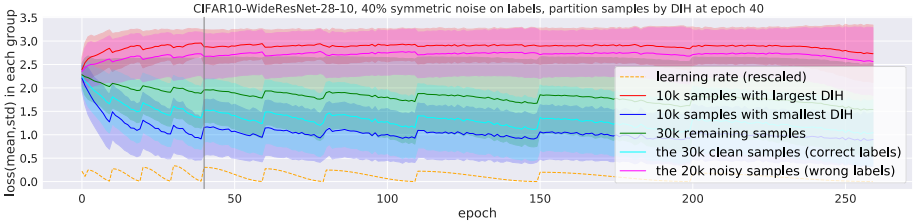

Figure 14: Losses (mean and std.) of the three groups of samples partitioned by a DIH metric (i.e., running mean of prediction flip) computed at epoch 40 when 40% of labels are randomly changed to another wrong class (i.e., **40% symmetric noises on labels**. We also show the losses on the clean samples with correct labels and noisy samples with wrong labels, where the former exhibit lower DIH than the latter. Hence, DIH is robust to label noises and can identify the hard and easy samples, which are mainly composed of the clean and noisy data respectively in this scenario.

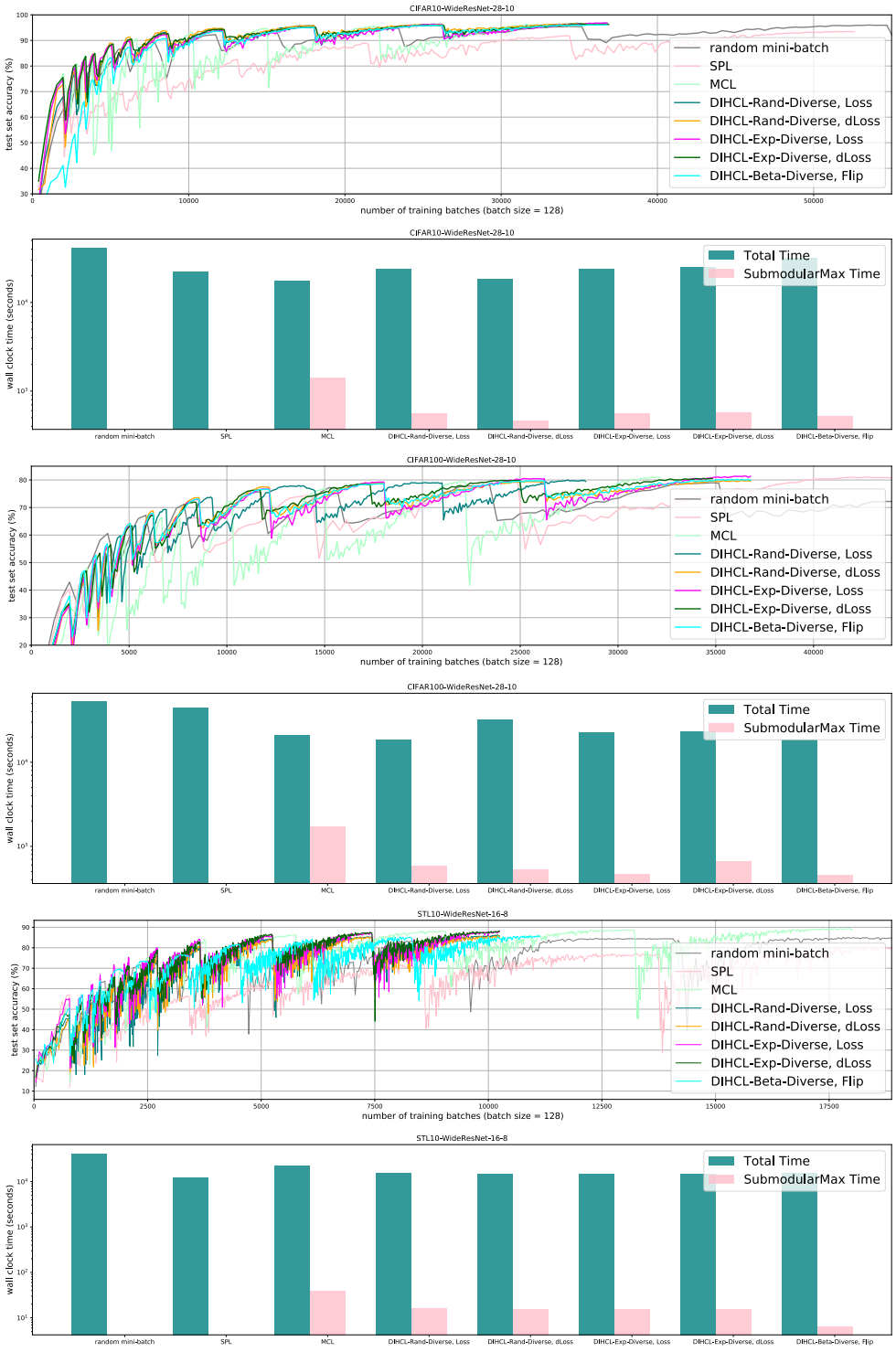

Figure 15: Training DNNs by using DIHCL (and its variants), SPL [25], MCL [52], and random mini-batch SGD on 3 datasets, i.e., CIFAR10, CIFAR100 and STL-10. We use "Diverse" to denote DIHCL that further reduces S_t by applying submodular maximization for Eq. (3). We report how the test accuracy changes with the number of training batches for each method, and the (**log-scale**) wall-clock time for 1) the entire training and 2) the submodular maximization part in DIHCL with diversity and MCL.

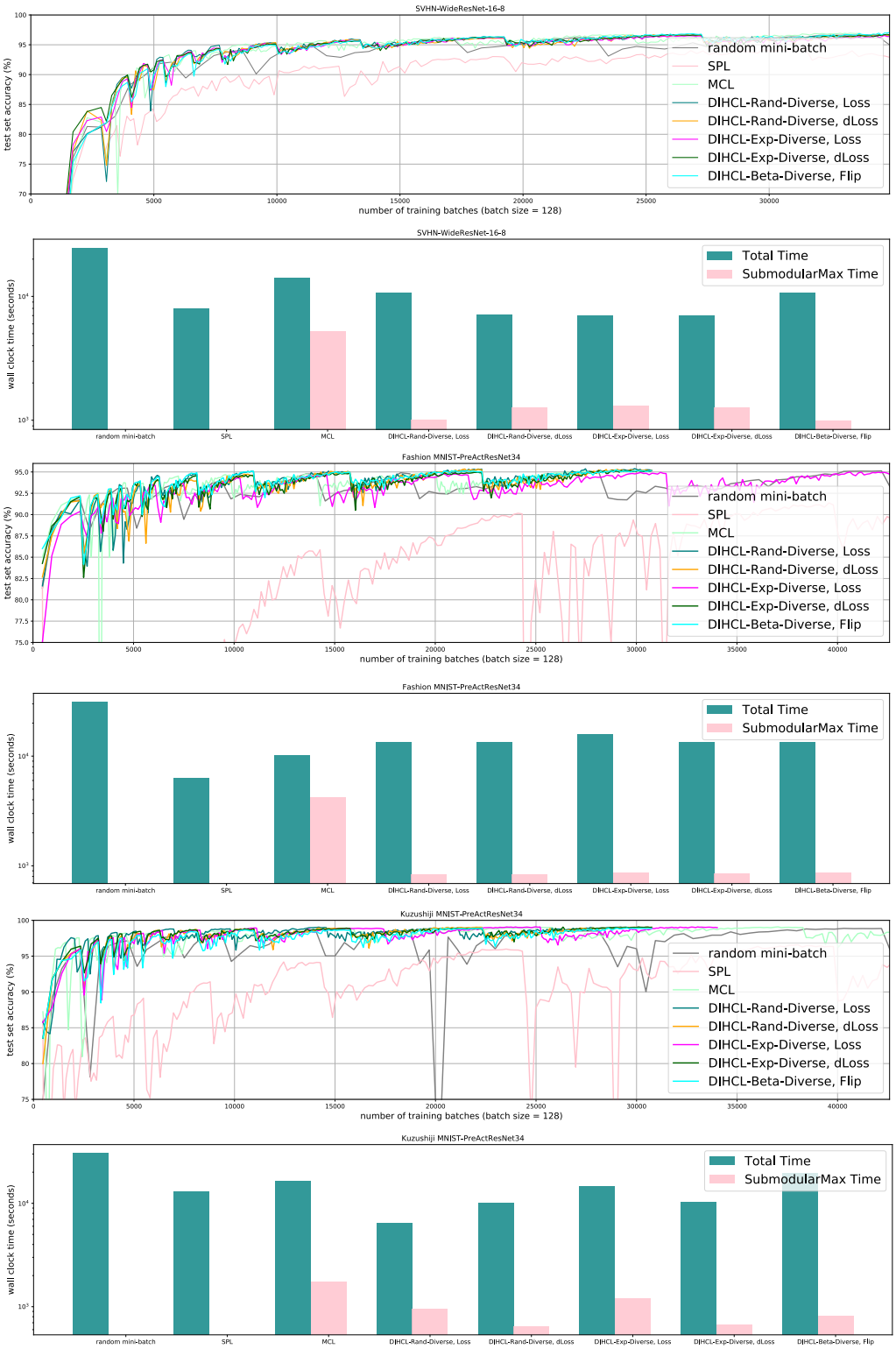

Figure 16: Training DNNs by using DIHCL (and its variants), SPL [25], MCL [52], and random mini-batch SGD on 3 datasets, i.e., SVHN, Fashion MNIST and Kuzushiji MNIST. We use "Diverse" to denote DIHCL that further reduces S_t by applying submodular maximization for Eq. (3). We report how the test accuracy changes with the number of training batches for each method, and the (**log-scale**) wall-clock time for 1) the entire training and 2) the submodular maximization part in DIHCL with diversity and MCL.

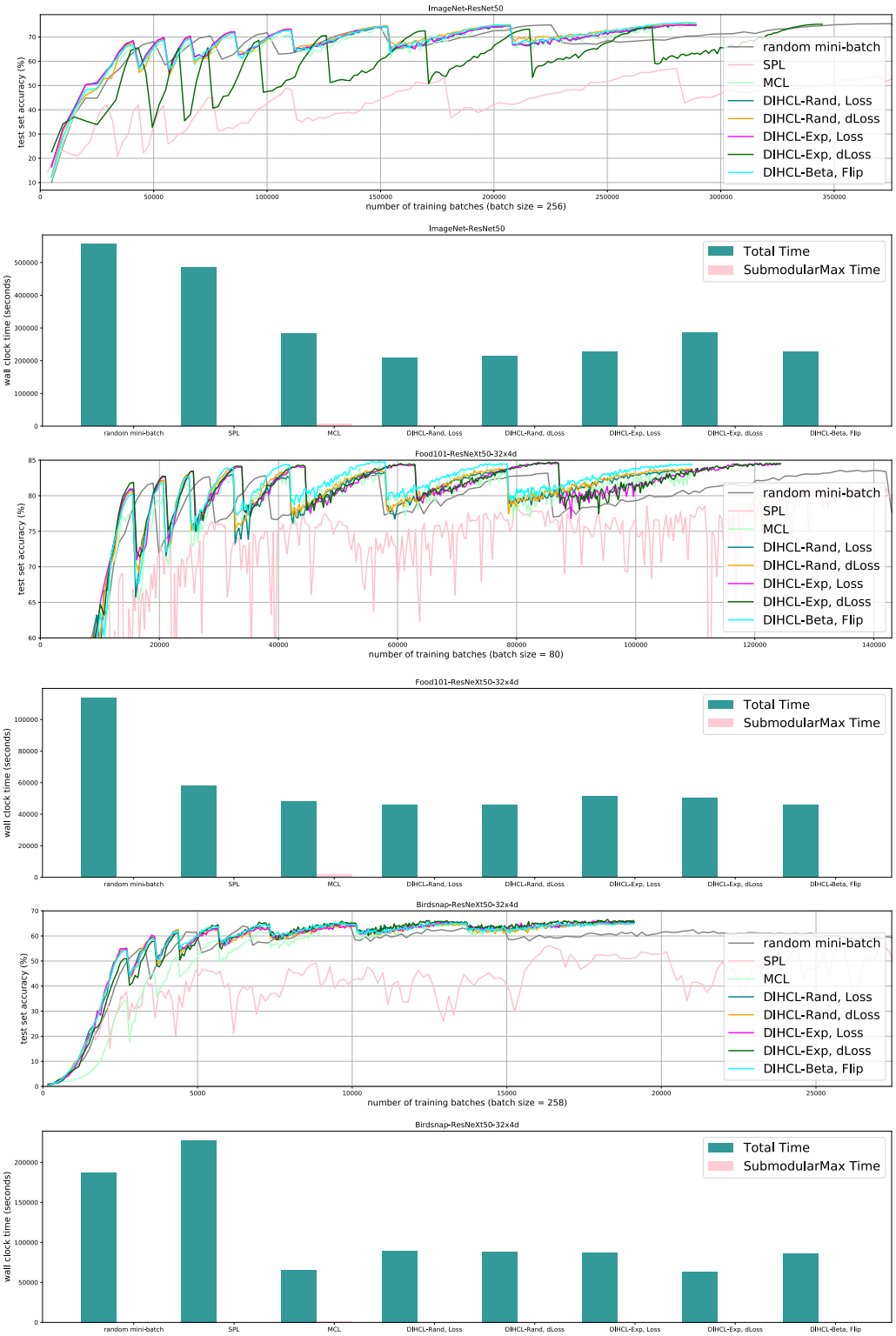

Figure 17: Training DNNs by using DIHCL (and its variants), SPL [25], MCL [52], and random mini-batch SGD on 3 datasets, i.e., ImageNet, Food-101 and Birdsnap. We report how the test accuracy changes with the number of training batches for each method, and the wall-clock time for 1) the entire training and 2) the submodular maximization part in MCL.

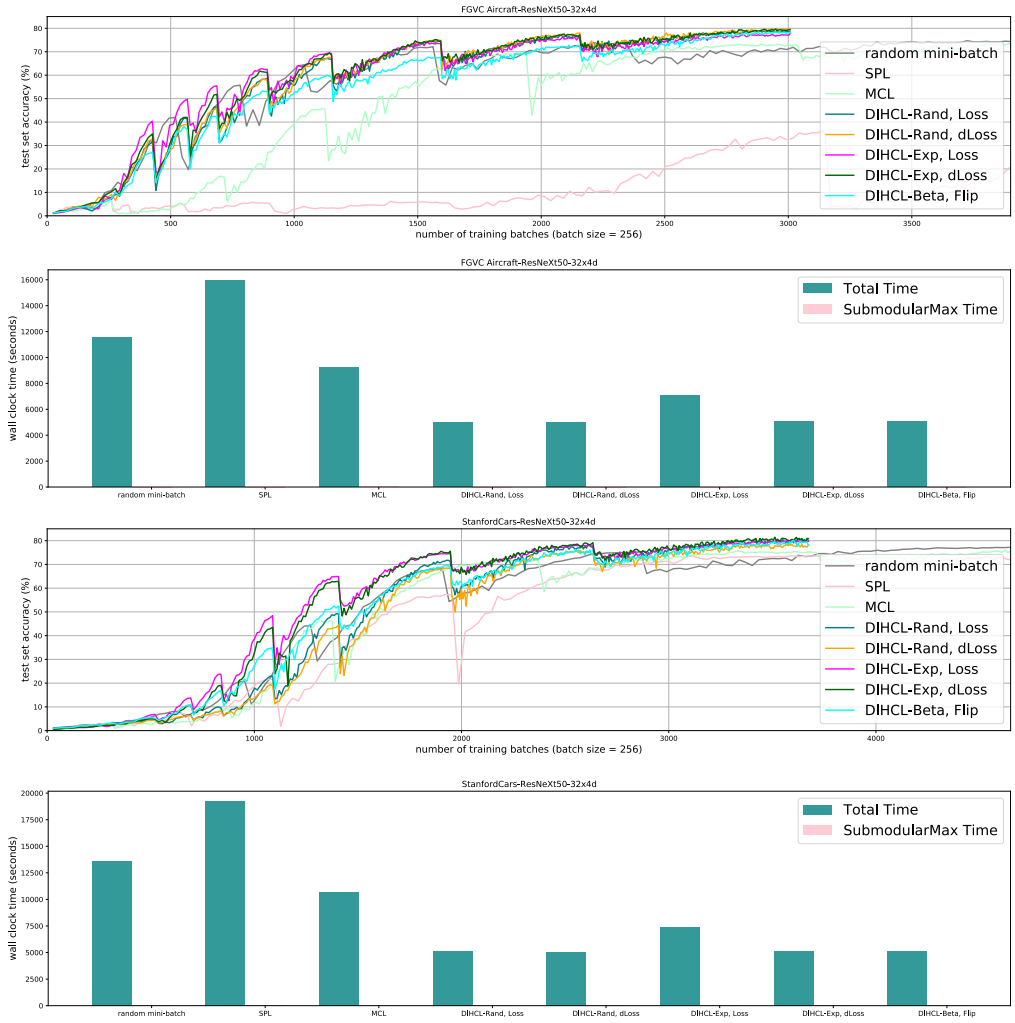

Figure 18: Training DNNs by using DIHCL (and its variants), SPL [25], MCL [52], and random mini-batch SGD on 2 datasets, i.e., FGVC Aircraft and Stanford Cars. We report how the test accuracy changes with the number of training batches for each method, and the wall-clock time for 1) the entire training and 2) the submodular maximization part in MCL.

Table 4: Test Acc (mean±variance) over 5 trials on two CIFAR datasets. It shows that the performance of DIHCL is stable and does not suffer from high variance.

Curriculum	CIFAR10	CIFAR100
DIHCL-Rand, Loss	96.74 ± 0.04	80.80 ± 0.16
DIHCL-Rand, dLoss	96.75 ± 0.06	80.73 ± 0.21
DIHCL-Exp, Loss	97.07 ± 0.11	82.31 ± 0.24
DIHCL-Exp, dLoss	96.44 ± 0.10	81.35 ± 0.27
DIHCL-Beta, Flip	96.48 ± 0.04	81.13 ± 0.18

C Theoretical Formulation and Proofs

C.1 Problem Formulation

A curriculum is a sequence of selected subsets of V, where each subset is a batch of samples used to update the model in each training step, i.e., $(S_1, S_2, \ldots, S_T)$. Let each subset $S \subseteq V$ have a characteristic vector $e_S \in \{0,1\}^n$ where $e_S(i) = 1$ if $i \in S$ and $e_S(i) = 0$ if $i \notin S$. We can form a compact multi-set representation of the curriculum $S_{1:t} \triangleq \sum_{\ell=1}^{t} e_{S_t}$, which is a non-negative integer valued vector of length $|V|$ where $S_{1:t}[i]$ counts how many times sample i has been selected in the T training steps. Formally, $S_{1:t}$ is a point on the non-negative integer lattice $\mathbb{Z}^V_{\geq 0}$. We formulate the goal of a curriculum learning method to be constrainedly maximizing an objective function $f : \mathbb{Z}^V_{\geq 0} \to \mathbb{R}_{\geq 0}$:

$$\max_{S_{1:T}:\forall t\in[T], S_t\subseteq V, |S_t|\leq k_t} f(S_{1:T}), \tag{7}$$

where k_t is a limit on the size of the set of samples selected at time t. The function $f(S_{1:T})$ evaluates the quality of a curriculum $S_{1:T}$. Hence, it depends on the model, the data, and the optimization algorithm, and thus covers all the information needed to design an optimal curriculum. However, it is intractable to estimate since it measures the quality of all possible training sequences (exponential, i.e., $T^{|V|+1}$) and so we cannot directly maximize $f(S_{1:T})$ even if we had it.

As a surrogate, partial information about $f(\cdot)$ might be available at each step in the form of the marginal gain of each sample i, i.e.,

$$f(i|S_{1:t-1}) \triangleq f(e_i + S_{1:t-1}) - f(S_{1:t-1}). \tag{8}$$

$f(i|S_{1:t-1})$ can be seen to measure how informative sample i is to future training given the historical curriculum $S_{1:t-1}$. Recalling the properties of DIH, it is perhaps not unreasonable to make an assumption that $r_t(i) \approx f(i|S_{1:t-1})$, since $r_t(i)$ depends on the historical curriculum and (as empirically argued above) can reflect the hardness of sample i into the future. For other CL methods, by contrast, $f(i|S_{1:t-1})$ can be assumed to be the instantaneous hardness $a_t(i)$.

Given the partial observations $f(i|S_{1:t-1})$, curriculum learning can be formulated as online optimization that aims to maximize $f(S_{1:T})$: at every step t, we select a subset of samples $S_t \subseteq V$ of size $|S_t| \leq k_t$ to train the model, observing only marginal gains $f(i|S_{1:t-1})$ for each i. We can therefore define the following objective, which is an approximation of Eq. 7 that ignores the dependency between samples within each subset S_t.

$$\max_{S_{1:T}:\forall t\in[T], S_t\subseteq V, |S_t|\leq k_t} g(S_{1:T}) \triangleq \sum_{t=1:T} \sum_{i\in S_t} f(i|S_{1:t-1}) \tag{9}$$

For simplicity, we slightly overload the notation $S_{1:T}$ for function $g(\cdot)$ so that we retain information about the subset selected at every time step (i.e., we can extract S_t for $1 \leq t \leq T$ from $S_{1:T}$). Compared to Eq. 7, Eq. 9 is a more tractable problem since it only requires partial information about $f(\cdot)$, and we can solve it by sequentially determining S_t from $t = 1$ to $t = T$ since selecting S_t only depends on the historically selected $S_{1:t-1}$. But directly solving Eq. (9) requires inference over all the n training samples in every step, as most existing curriculum learning methods do. For example, at step t, we need to know $f(i|S_{1:t-1})$ for all $i \in V$ to select S_t. Since we're assuming $r_t(i) \approx f(i|S_{1:t-1})$, for $i \in S_{t-1}$, no extra inference cost is needed to compute $f(i|S_{1:t-1})$ since the gradient computation for training in step $(t-1)$ already includes inference steps, so the predictions and losses are free byproducts. However, for $i \notin S_{t-1}$, extra inference is required and could be expensive when training DNNs since the training in step $(t-1)$ does not include such inference calculations.

Is it possible to mitigate or eliminate the inference costs for $i \notin S_{t-1}$? A simple solution is to keep using the stale marginal gain for any $i \notin S_{t-1}$ computed whenever it was most recently at some earlier step $\tau_t(i) < t-1$. In other words, denote $\tau_t(i) < t-1$ as the most recent step before $t-1$ when i was selected. Then we use $f(i|S_{1:\tau_t(i)})$ instead of $f(i|S_{1:t-1})$ when selecting such training samples for step t. However, this approximation cannot be applied to instantaneous hardness measures since they can change drastically between steps, as shown in Figure 1. By contrast, as discussed in Section 2, it is safer to keep a stale DIH for $i \notin S_{t-1}$ since DIH is smooth and hence more consistent between steps and decreases during training.

We will revisit this formulation below where we will show that although DIHCL uses a stale marginal gain to optimize an approximation (Eq. 9) of the original problem (Eq. 7), it achieves an approximation bound (as given in Corollary 1) to the global optimal solution of Eq. (7), if one is allowed to make further assumptions on $f(\cdot)$.

$f : \mathbb{Z}^V_{\geq 0} \to \mathbb{R}_{\geq 0}$ on ground set V is defined over an integer lattice. The diminishing return (DR) property of f is the following inequality $0 \leq \forall x \leq y$:

$$f(x + e_i) - f(x) \geq f(y + e_i) - f(y), \tag{10}$$

Where e_i is a one-hot vector with all zeros except for a single one at the ith position. We assume f is normalized and monotone, i.e., $f(0) = 0$ and $f(x) \leq f(y), \forall 0 \leq x \leq y$. W.l.o.g. we also assume the max singleton gain is bounded by 1, i.e., $\max_i f(e_i) \leq 1$. We can think that f takes input as a multi-set, and the gain of an item diminishes as its counter increases in the multi-set.

C.2 Proofs

In the setting of selecting mini-batches for training machine learning models, suppose the mini-batch size is k, the training set is V, and at every time step t, we select $S_t \subseteq V$ with $|S_t| = k$, and only observe the gains on the selected subset (e.g., for neural networks, we update the running mean of training losses during the forward pass of the chosen mini-batch, or DIH type (A)). At every time step of selecting a mini-batch, we observe $f(i|S_{1:t-1})\ \forall i \in S_t$. Let $n = |V|$, $m = \frac{n}{k}$, and for simplicity assume $n \mod k = 0$. We define function g to reflect the observed gains from f as we select data samples at each training step:

$$g(S_{1:t}) = \sum_{t'=1:t} \sum_{i \in S_{t'}} f(i|S_{1:t'-1}) \tag{11}$$

For simplicity, we slightly overload the notation of $S_{1:T}$ for function $g(\cdot)$ so that we retain information about the subset selected at every time step (i.e., we can extract S_t for $1 \leq t \leq T$ from $S_{1:T}$). Note that g is permutation-variant for $k > 1$, i.e., for different ordering in $S_{1:t}$, g gives different values.

Theorem 1. *For $f : \mathbb{Z}^V_{\geq 0} \to \mathbb{R}_{\geq 0}$ on ground set V satisfying the DR property, compared to any solution $S^*_{1:T}$, $S_{1:T}$, the solution of* DIHCL-Greedy*, achieves*

$$g(S_{1:T}) + c_{f,m} \geq \max\left\{\frac{1 - e^{-1}}{k}, \frac{k}{2n}\right\} g(S^*_{1:T}), \tag{5}$$

Where $c_{f,m} \triangleq m \min_{A_{1:m}} g(A_{1:m})$ such that $\bigcup_{i=1}^m A_i = V$, and $|A_i| = k$.

To bridge $S_{1:T}$ with $S^*_{1:T}$, we first connect $S_{1:T}$ to the greedy solution with singleton gain oracle, but uses the history of sequence of $(S_1, S_2, \ldots, S_{T-1})$, which we denote by $(\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_T)$:

$$\hat{S}_t = \operatorname*{argmax}_{S \subseteq V, |S| = k} \sum_{i \in S} f(i|S_{1:t-1}). \tag{12}$$

Note we denote any set with subscript 0 (at time step 0) as an empty set, i.e. $S_0 = \emptyset$, $\hat{S}_0 = \emptyset$, and etc.. We define the observed gain values on the singleton gain oracle with history of $(S_1, S_2, \ldots, S_{T-1})$ as:

$$g(\hat{S}_{1:T}|S_{1:T-1}) = \sum_{t=1:T} \sum_{i \in \hat{S}_t} f(i|S_{1:t-1}) \tag{13}$$

Firstly, we derive a lower bound of $g(S_{1:T})$ in terms of $g(\hat{S}_{1:T}|S_{1:T-1})$.

Lemma 1. $g(S_{1:T}) + c_{f,m} \geq g(\hat{S}_{1:T}|S_{1:T-1})$.

Proof. Define $\zeta(i, A_{1:t})$ to return the subsequence of $A_{1:t}$ that starts from A_1 and ends at $A_{t'}$ where $A_{t'}$ is the last set in the whole sequence that contains the element i, i.e., $\zeta(i, A_{1:t}) = \operatorname{argmax}_{A_{1:t'}} t' \mathbb{1}_{i \in A_{t'}}$. When i is not present in the whole sequence $A_{1:t}$, $\zeta(i, A_{1:t})$ returns $\emptyset$.

By definitions of $c_{f,m}$ and $g(\hat{S}_{1:m}|S_{1:m-1})$, we have $c_{f,m} \geq mf(V) \geq g(\hat{S}_{1:m}|S_{1:m-1})$ due to the diminishing return (DR) property.

For $T \leq m$, Lemma 1 is true because of the above inequality.

For $T \geq m + 1$, we compare the previous gains of elements in S_t to the current gains of elements in $\hat{S}_t$:

$$
\begin{aligned}
g(S_{1:T}) + c_{f,m} &\geq g(S_{1:T}) + g(\hat{S}_{1:m}|S_{1:m-1}) && (14)\\
&\geq \sum_{t=m+1:T} \sum_{i \in S_t} f(i|\zeta(i, S_{1:t-1})) + g(\hat{S}_{1:m}|S_{1:m-1}) && (15)\\
&\geq \sum_{t=m+1:T} \sum_{i \in \hat{S}_t} f(i|\zeta(i, S_{1:t-1})) + g(\hat{S}_{1:m}|S_{1:m-1}) && (16)\\
&\geq \sum_{t=m+1:T} \sum_{i \in \hat{S}_t} f(i|S_{1:t-1}) + g(\hat{S}_{1:m}|S_{1:m-1}) && (17)\\
&= g(\hat{S}_{1:T}|S_{1:T-1}) && (18)
\end{aligned}
$$

Eq. 14 and Eq. 17 hold due to the diminishing return property and Eq. 16 is a result of the greedy step (i.e., S_t is optimal when conditioning on $\zeta(i, S_{1:t-1})$). Note that we are guaranteed to find an element in the sequence history ($|\zeta(i, S_{1:t-1})| > 0$ in Eq. 15 and Eq. 16) since we sweep the ground set V in the first m steps of solution $S_{1:m}$.

□

Remarks. In the proof, we ignore the gain at the T step, i.e., $\sum_{i \in S_T} f(i|S_{1:T-1})$ as such gain can potentially be zero. In other words, $g(S_{1:T-1}) + c_{f,m} \geq g(\hat{S}_{1:T}|S_{1:T-1})$. For the case that f is modular, i.e., $f(x + e_i) = f(x) + f(e_i)$, and for only k elements in V, the function evaluations are non-zero, the bound meets in equality: $g(S_{1:T-1}) + c_{f,m} = g(\hat{S}_{1:T}|S_{1:T-1})$. The idea is that we have to sweep all elements in the ground set before we identify the non-zero-valued elements.

Next, we find a lower bound of $g(\hat{S}_{1:T}|S_{1:T-1})$ in terms of $g(S^*_{1:T})$.

Lemma 2. $g(\hat{S}_{1:T}|S_{1:T-1}) \geq \frac{1-e^{-1}}{k} g(S^*_{1:T})$.

Proof. For $T' < T$, we compare $g(S^*_{1:T})$ with $g(\hat{S}_{1:T}|S_{1:T-1})$:

$$
\begin{aligned}
\frac{1}{k} g(S^*_{1:T}) &= \frac{1}{k} \sum_{t=1:T} \sum_{i \in S^*_t} f(i|S^*_{1:t-1}) && (19)\\
&\leq \frac{1}{k} \sum_{t=1:T} k \times \max_{i \in S^*_t} f(i|S^*_{1:t-1}) && (20)\\
&\leq \frac{1}{k} k f(S^*_{1:T}) && (21)\\
&\leq f(S^*_{1:T'} + S_{1:T}) && (22)\\
&\leq f(S_{1:T'}) + \sum_{i \in S^*_{1:T}} f(i|S_{1:T'}) && (23)\\
&\leq f(S_{1:T'}) + T(\sum_{i \in \hat{S}_{T'+1}} f(i|S_{1:T'})) && (24)\\
&= f(S_{1:T'}) + T(g(\hat{S}_{1:T'+1}|S_{1:T'}) - g(\hat{S}_{1:T'}|S_{1:T'-1})) && (25)\\
&\leq g(\hat{S}_{1:T'}|S_{1:T'-1}) + T(g(\hat{S}_{1:T'+1}|S_{1:T'}) - g(\hat{S}_{1:T'}|S_{1:T'-1})) && (26)
\end{aligned}
$$

From Eq. 20 to Eq. 21, we use $\sum_{t=1:T} \max_{i \in S^*_t} f(i|S^*_{1:t-1}) \leq \sum_{t=1:T} f(S^*_t|S^*_{1:t-1}) = f(S^*_{1:T})$. Eq. 23 is due to DR property and Eq. 24 is a result of greedy selection. Also note that for Eq. 22, $S^*_{1:T'} + S_{1:T} = \sum_{l=1}^{T'} e_{S^*_l} + \sum_{l=1}^{T} e_{S_l}$.

By rearranging Eq. 26, we have $\frac{1}{T}(\frac{1}{k} g(S^*_{1:T}) - g(\hat{S}_{1:T'}|S_{1:T'-1}))) \leq g(g(\hat{S}_{1:T'+1}|S_{1:T'}) - g(\hat{S}_{1:T'}|S_{1:T'-1})$, i.e., every time step, we reduce the gap to $1/k$ of the of optimal solution by at least $\frac{1}{T}$. Therefore $g(\hat{S}_{1:T}|S_{1:T-1}) \geq \frac{1-e^{-1}}{k} g(S^*_{1:T})$. □

Remarks. We will show that there is a hard case with $1/k$ factor. Suppose f is a set cover function ($f(i|A) = 0$ if $i \in A$) and $|V| = k^2$. The ground set V is partitioned into k groups $V =$

$V_1 \cup V_2 \cup \ldots \cup V_k$ with k elements in each group, such that $f(a) = 1 \, \forall a \in V$, $f(a|b) = 0 \, \forall a, b \in V_i$, and $f(\{a, b\}) = f(a) + f(b) \, \forall a \in V_i, b \in V_j, i \neq j$. For the first time step, $g(\hat{S}_1|\emptyset)$ gets a gain of k which is equal to $g(S_1^*)$. However, S_1 may select one element from each of the group since we are doing the ground set sweeping exploration, and all the rest gains will be zero conditioned on S_1. The optimal solution, on the other hand, can select all k elements from one group at a time, and get a value of k^2 in the end.

Combine Lemma 1 and Lemma 2, we get the first factor $\frac{1-e^{-1}}{k}$ for the bound in Theorem 1.

Lemma 3. $g(\hat{S}_{1:T}|S_{1:T-1}) \geq \frac{k}{2n} g(S_{1:T}^*)$.

Proof. We will first connect $g(\hat{S}_{1:T}|S_{1:T-1})$ with the solution that selects the entire ground set V at every step, i.e., $g(V_{1:T}) = g((V, V, \ldots, V))$.

$$
\begin{aligned}
g(\hat{S}_{1:T}|S_{1:T-1}) &\geq \frac{k}{n} g(V_{1:T}|S_{1:T-1}) && (27)\\
&\geq \frac{k}{n} g(V_{1:T}|V_{1:T-1}) && (28)\\
&= \frac{k}{n} \sum_{t=1:T} \sum_{i \in V} f(i|V_{1:t-1}) && (29)\\
&\geq \frac{k}{n} f(V_{1:T}) && (30)
\end{aligned}
$$

For Eq. 27, we use the fact that $\hat{S}_{1:T}$ achieve the top k gains selected by the greedy process in each step. Next, we will bound any solution $g(S_{1:T}^*)$ by $g(V_{1:T}|V_{1:T-1})$. Firstly, we will need to partition $S_{1:T}^*$ into two parts: (1) for the first part, we collect all the new elements introduced at every time step t that do not exist in $S_{1:t-1}^*$, i.e., $\tilde{S}_{1:T}^* = (S_1^* \setminus \emptyset, S_2^* \setminus \cup(S_{1:1}^*), S_3^* \setminus \cup(S_{1:2}^*), \ldots, S_T^* \setminus \cup(S_{1:T-1}^*))$, where $\tilde{S}_t^* \triangleq S_t^* \setminus \cup(S_{1:t-1}^*)$ and $\cup(S_{1:t}) \triangleq \bigcup_{i=1:t} S_i$, which is the set union on all elements in the multiset (you can think it sets all the counters in the multiset with values ≥ 1 to ones), and "$\setminus$" is the set minus operation. Therefore, $\tilde{S}_{1:T}^*$ contains every element in $S_{1:T}^*$ exactly once, i.e., every element in $S_{1:T}^*$ only appears once in $\tilde{S}_{1:T}^*$, and at many time steps, $\tilde{S}_t^*$ might be empty; (2) the other part contains all the rest elements, i.e., $S_{1:T}^* - \tilde{S}_{1:T}^* = (S_1^* \setminus \tilde{S}_1^*, S_2^* \setminus \tilde{S}_2^*, \ldots, S_T^* \setminus \tilde{S}_T^*)$. We bound the two parts as follows:

$$
\begin{aligned}
g(S_{1:T}^*) &= \sum_{t=1:T} \sum_{i \in S_t^*} f(i|S_{1:t-1}^*) && (31)\\
&= \sum_{t=1:T} \sum_{i \in \tilde{S}_t^*} f(i|S_{1:t-1}^*) + \sum_{t=1:T} \sum_{i \in (S_t^* \setminus \tilde{S}_t^*)} f(i|S_{1:t-1}^*) && (32)\\
&\leq \sum_{i \in V} f(i) + \sum_{t=1:T} \sum_{i \in (S_t^* \setminus \tilde{S}_t^*)} f(i|S_{1:t-1}^*) && (33)\\
&\leq \sum_{i \in V} f(i) + f(S_{1:T}^*) && (34)\\
&\leq g(V_{1:T}|V_{1:T-1}) + f(V_{1:T}) && (35)
\end{aligned}
$$

From Eq. 32 to Eq. 33, we use the fact $\sum_{t=1:T} \sum_{i \in \tilde{S}_t^*} f(i|S_{1:t-1}^*) \leq \sum_{t=1:T} \sum_{i \in \tilde{S}_t^*} f(i) \leq \sum_{i \in V} f(i)$ since $\tilde{S}_T^*$ contains one instance of every element in $S_{1:T}^*$ and removing the conditioning part would make the gains larger (guaranteed by diminishing return property). To get Eq. 34, we reduce the conditioning part of $f(i|S_{1:t-1}^*)$ in Eq. 33 by using the following inequality: for $A_1 \subseteq A_2 \subseteq V$, denote $A_3 = A_2 \setminus A_1$ and let $A_1 = \{i_1, i_2, \cdots, i_{|A_1|}\}$, by diminishing return property of $f(\cdot)$, we have:

$$
\begin{aligned}
\sum_{i \in A_1} f(i|A_2) \leq f(i_1|A_3) + f(i_2|\{i_1\} \cup A_3) + f(i_3|\{i_1, i_2\} \cup A_3) + \\
\ldots + f(i_{|A_1|}|A_2 \setminus \{i_{|A_1|}\}) = f(A_2|A_3). \qquad (36)
\end{aligned}
$$

According to the pre-defined partition, we pick out the first occurrence of every element into $\tilde{S}_{1:T}^*$, every remaining element $i \in (S_t^* \setminus \tilde{S}_t^*)$ is guaranteed to find itself in its conditioning history $S_{1:t-1}^*$

and therefore, we may use the inequality described in Eq. 36 to bound the second term in Eq. 33 by $f(S^*_{1:T})$ (letting $A_1 = S^*_t \setminus \tilde{S}^*_t$ and $A_2 = S^*_{1:t-1}$ and applying the inequality from $t = 1$ to T sequentially). To make it more concrete, for example, at step $t = 2$, by using Eq. 36, we have:

$$\sum_{i\in(S^*_2\setminus\tilde{S}^*_2)} f(i|S^*_1) \le f(i_1) + f(i_2|i_1) + f(i_3|i_1,i_2) + \ldots + f(i_{|S^*_2\setminus\tilde{S}^*_2|}|S^*_2 \setminus \tilde{S}^*_2 \setminus \{i_{|S^*_2\setminus\tilde{S}^*_2|}\}) \tag{37}$$

$$= f(S^*_2 \setminus \tilde{S}^*_2); \tag{38}$$

At time step $t = 3$, we have:

$$\sum_{i\in(S^*_3\setminus\tilde{S}^*_3)} f(i|S^*_{1:2}) \le f(i_1|S^*_2 \setminus \tilde{S}^*_2) + f(i_2|\{i_1\} \cup (S^*_2 \setminus \tilde{S}^*_2)) + f(i_3|\{i_1,i_2\} \cup (S^*_2 \setminus \tilde{S}^*_2)) + \ldots + f(i_{|S^*_3\setminus\tilde{S}^*_3|}|(S^*_3 \setminus \tilde{S}^*_2 \setminus \{i_{|S^*_3\setminus\tilde{S}^*_3|}\}) \cup (S^*_2 \setminus \tilde{S}^*_2)) \tag{39}$$

$$= f(S^*_3 \setminus \tilde{S}^*_3|S^*_2 \setminus \tilde{S}^*_2). \tag{40}$$

Hence, we have the inequality between Eq. 33 and Eq. 34.

To get Eq. 35 from Eq. 34, we use the fact $\sum_{i\in V} f(i) \le g(V_{1:T}|V_{1:T-1})$ because $g(V_{1:T}|V_{1:T-1})$ contains $\sum_{i\in V} f(i)$ at step $t = 1$, and the second term in Eq. 35 is due to the fact that $f(\cdot)$ is monotone non-decreasing.

Finally, we combine Eq. 30 and Eq. 35, we get $2g(\hat{S}_{1:T}|S_{1:T-1}) \ge \frac{2k}{n} f(V_{1:T}) \ge \frac{k}{n} g(S^*_{1:T})$.

□

By combining Lemma 1 and Lemma 3, we get the second factor $\frac{k}{2n}$ for the bound in Theorem 1.

Remarks. The first factor $\frac{1-e^{-1}}{k}$ dominates when k is relatively small compared to n. Recall the hard case example above on the $1/k$ factor. We can generalize it to any $k < n$ by (almost) equally distribute the n elements into the k groups described in the hard case. Then, for $n < k^2$, the optimal solution gets n in the end while the greedy solution gets k, so the ratio is $\frac{k}{n}$. For $n \ge k^2$, the optimal solution still gets k^2 while the greedy solution gets k, so the ratio is $\frac{1}{k}$. In both scenarios, our bounds match the hard example up to constant factors.

Corollary 1. *With the assumptions in Theorem 1, we have*

$$f(S_{1:T}) + \frac{1}{k} c_{f,m} \ge \frac{1}{k} \max\left\{\frac{1-e^{-1}}{k}, \frac{k}{2n}\right\} f(S^*_{1:T}). \tag{6}$$

Proof.

$$f(S_{1:T}) + \frac{1}{k} c_{f,m} \ge \frac{1}{k}(g(S_{1:T}) + c_{f,m}) \tag{41}$$

$$\ge \frac{1}{k} \max\{\frac{1-e^{-1}}{k}, \frac{k}{2n}\} g(S^*_{1:T}) \tag{42}$$

$$\ge \frac{1}{k} \max\{\frac{1-e^{-1}}{k}, \frac{k}{2n}\} f(S^*_{1:T}) \tag{43}$$

□

We mentioned a few weighted sampling method to replace the greedy step. Here, we apply a random sampling procedure similar to the lazier-than-lazy approach [32]: we sample a subset $R_j \subseteq V \setminus S_{t,j-1}$ of size $\frac{n}{k} \log \frac{1}{\epsilon}$, and then choose the top-gain element from R_j and add it to $S_{t,j-1}$ to from $S_{t,j}$. We denote such sampling based greedy as DIHCL-Greedy-random.

Theorem 2. *For $f : \mathbb{Z}^V_{\ge 0} \to \mathbb{R}_{\ge 0}$ on ground set V with DR property, compared to any solution $S^*_{1:T}$, $S_{1:T}$, the solution of DIHCL-Greedy-random, achieves*

$$\mathbb{E}[g(S_{1:T})] + c_{f,m} \ge (1 - (1 - \frac{1-\epsilon}{k})^k) \frac{1-e^{-1}}{k} g(S^*_{1:T}) \tag{44}$$

$$\ge \frac{(1-e^{-1}-\epsilon)(1-e^{-1})}{k} g(S^*_{1:T}). \tag{45}$$

Proof. We can think the selection of every S_t is a greedy process of k steps, with S_t as the optimal solution. Suppose up to step j, we select the set $S_{t,j}$. We first bound the probability that the sampled set has some intersection with the optimal set S_t.

$$\Pr[R_j \cap (S_t \setminus S_{t,j-1}) \neq \emptyset] \geq 1 - (1 - \frac{|S_t \setminus S_{t,j-1}|}{|V \setminus S_{t,j-1}|})^{|R|} \tag{46}$$

$$\geq 1 - (1 - \frac{|S_t \setminus S_{t,j-1}|}{n})^{|R|} \tag{47}$$

$$\geq 1 - e^{-\frac{|R|}{n}|S_t \setminus S_{t,j-1}|} \tag{48}$$

$$\geq (1 - e^{-\frac{|R|k}{n}})\frac{|S_t \setminus S_{t,j-1}|}{k} \tag{49}$$

$$\tag{50}$$

In step j, we denote the selected item by v_j. We can then get the expected gain given the probability that there is some intersection:

$$\mathbb{E}[f(v_j|\zeta(v_j, S_{1:t-1}))] \geq \Pr[R_j \cap (S_t \setminus S_{t,j}) \neq \emptyset]\frac{1}{|S_t \setminus S_{t,j}|}\sum_{i \in S_t} f(i|\zeta(i, S_{1:t-1})) \tag{51}$$

$$= \frac{1-\epsilon}{k}\sum_{i \in S_t} f(i|\zeta(i, S_{1:t-1})) \tag{52}$$

Again, we get the argument that we are reducing the gap to the optimal solution by $(1-\epsilon)/k$ for every selected item v_j on expectation.

$$\sum_{j=1:k} \mathbb{E}[f(v_j|\zeta(v_j, S_{1:t-1}))] \geq (1 - (1 - \frac{1-\epsilon}{k})^k)\sum_{i \in S_t} \mathbb{E}[f(i|\zeta(i, S_{1:t-1}))] \tag{53}$$

We can then apply Eq. 53 in the Eq. 15 of Lemma 1, and get

$$\mathbb{E}[g(S_{1:T})] + c_{f,m} \geq (1 - (1 - \frac{1-\epsilon}{k})^k)\mathbb{E}[g(\hat{S}_{1:T}|S_{1:T-1})] \tag{54}$$

Combine with Lemma 2 we get the bound in Theorem 2.

□

Remarks. When n is large and $n \gg k$, we can approximate the sample without replacement using sample with replacement, and we can independently sample k subsets each of size $|R|$ at every time step to generate S_k. In such a case, the bound becomes $\mathbb{E}[g(S_{1:T})] + c_{f,m} \geq \frac{1-e^{-1}-\epsilon}{k}g(S^*_{1:T})$.

Similarly, we can also get the expectation bound on f:

Corollary 2. $\mathbb{E}[f(S_{1:T})] + \frac{1}{k}c_{f,m} \geq (1 - (1 - \frac{1-\epsilon}{k})^k)\frac{1-e^{-1}}{k^2}f(S^*_{1:T})$

Proof.

$$\mathbb{E}[f(S_{1:T})] + \frac{1}{k}c_{f,m} \geq \frac{1}{k}(\mathbb{E}[g(S_{1:T})] + c_{f,m}) \tag{55}$$

$$\geq (1 - (1 - \frac{1-\epsilon}{k})^k)\frac{1-e^{-1}}{k^2}g(S^*_{1:T}) \tag{56}$$

$$\geq (1 - (1 - \frac{1-\epsilon}{k})^k)\frac{1-e^{-1}}{k^2}f(S^*_{1:T}) \tag{57}$$

□

We can extend the setting so that we get noisy feedback from the gains of function f: $f(i|S_{1:t-1}) + \alpha_t$, and the problem becomes a multi-armed bandit problem. Specifically if we assume the noise a_t form a martingale difference sequence, i.e. $\mathbb{E}[\alpha_t|\alpha_1, \alpha_2, \ldots, \alpha_{t-1}] = 0$ and all α_t are bounded $\alpha_t \leq \sigma$ and if we make further assumption about the smoothness of the f and g function (assume the gains of f and g have RKHS-norm bounded by value B with some kernel $\mathbf{k}$), we can utilize the contextual bandit UCB algorithm proposed in [22] to get a $\sqrt{T}$ dependent regret. Also, under the noise setting, the contextual information becomes crucial, as the function has DR-property, and without an estimate of how much the gain decreases, we cannot have a better estimate of the upper bound on the noise

term. However, we note that utilizing the contextual information involves calculating large kernel matrices, which is not feasible for our purpose of efficient curriculum learning. We include the following result for completeness.

Theorem 3. *For $f : \mathbb{Z}_{\geq 0}^{V} \to \mathbb{R}_{\geq 0}$ on ground set V with DR property, suppose the gain of function g has RKHS-norm bounded by value B with some kernel $\mathbf{k}$), and the noise α_t's from a martingale difference sequence: $\mathbb{E}[\alpha_t|\alpha_1, \alpha_2, \ldots, \alpha_{t-1}] = 0$ and all α_t are bounded $|\alpha_t| \leq \sigma$. We define the maximum information gain if we have the perfect information about f, $\rho_T = \max_{A_{1:T}} H(y_{A_{1:T}}) - H(y_{A_{1:T}}|f)$, where H is the Shannon entropy, and $y_{A_{1:T}} = \{f(i|A_{1:t-1}) + a_t | i \in A_t, t = 1 : T\}$ denotes the collection of gain values we get from the sequence of $A_{1:T}$. We get the following regret bound:*

$$\Pr[\frac{1 - e^{-1}}{k} g(S^*_{1:T}) - g(S_{1:T}) \leq \sqrt{CT\beta_T\rho_T} + c_{f,m} + 2] \geq 1 - \delta, \tag{58}$$

Where, $C = 8/\log(1 + \sigma^{-2})$, $\beta_T = 2B^2 + 300\rho_T \ln^3(T/\delta)$.

Proof. The proof directly utilizes the third case of Theorem 1 in [22], using the history sequence $(S_1, S_2, \ldots, S_t)$ as the context:

$$\Pr[\frac{1 - e^{-1}}{k} g(S^*_{1:T}) - g(\hat{S}_{1:T}|S_{1:T-1}) \leq \sqrt{CT\beta_T\rho_T} + 2] \geq 1 - \delta \tag{59}$$

Combine with Lemma 1, we have:

$$\Pr[\frac{1 - e^{-1}}{k} g(S^*_{1:T}) - g(S_{1:T}) \leq \sqrt{CT\beta_T\rho_T} + c_{f,m} + 2] \geq 1 - \delta. \tag{60}$$

□