Bandit-Guided Submodular Curriculum for Adaptive Subset Selection

Prateek Chanda, Prayas Agrawal, Saral Sureka, Lokesh Reddy Polu, Atharv Kshirsagar, Ganesh Ramakrishnan

Visit our Paper (Scan this QR)

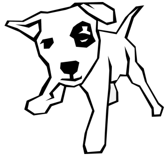
NEURAL INFORMATION PROCESSING SYSTEMS

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

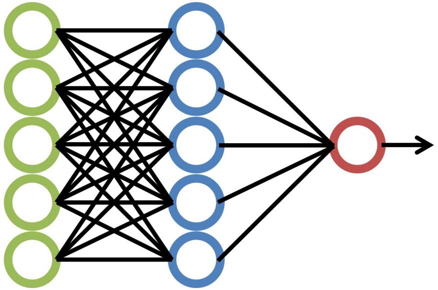
Curriculum Learning

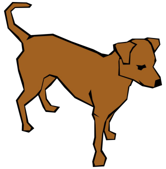
Level of Difficulty across Training Stages

Simple Instances



Early Stage





Moderate Difficulty Instances

Middle Stages

Complex Instances

Later Stage

NEURAL INFORMATION PROCESSING SYSTEMS

Curriculum Learning (A more formal Definition)

Definition 1. (Curriculum Learning) *Given a dataset* $\mathcal{D} = \bigcup_{i=1}^{k} \mathcal{B}_i$ *partitioned into disjoint batches* $\mathcal{B}_i$*, and a batch difficulty score function* $\mathbf{d} : \{\mathcal{B}_i\}_{i=1}^{k} \to \mathbb{R}_{\geq 0}$ *assigning non-negative difficulty scores, a* ***batch-wise curriculum*** *can be represented as a permutation* $\pi : [k] \mapsto [k]$ *over the ordered indices such that the ordered sequence*

$$\mathcal{C} = (\mathcal{B}_{\pi(1)}, \mathcal{B}_{\pi(2)}, \ldots, \mathcal{B}_{\pi(k)}),$$

satisfies the ***monotonic difficulty score condition****:* $\mathbf{d}(\mathcal{B}_{\pi(t)}) \leq \mathbf{d}(\mathcal{B}_{\pi(t+1)}) \quad \forall t \in \{1, \ldots, k-1\}.$

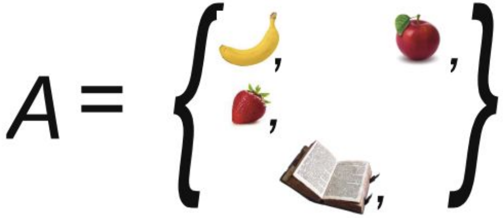
$\boldsymbol{d}$ → How do we define **difficulty?**

Annotators? Dataset Labellers ?

CHALLENGE

General Set Functions

$$f : 2^V \to \mathbb{R}$$



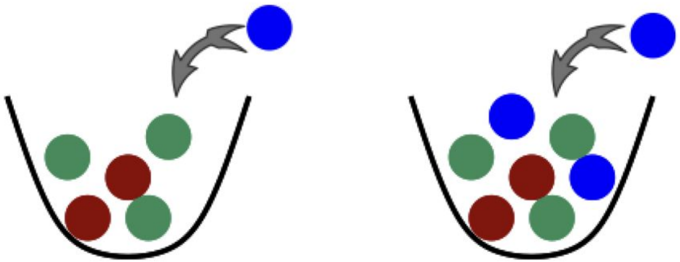
Choose Subset $A \subseteq V$

$$f(A) = 22$$

Submodular Functions

Submodularity (Diminishing Returns)

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B), \text{ if } A \subseteq B$$



$f = \#$ of distinct colors of balls in the urn.

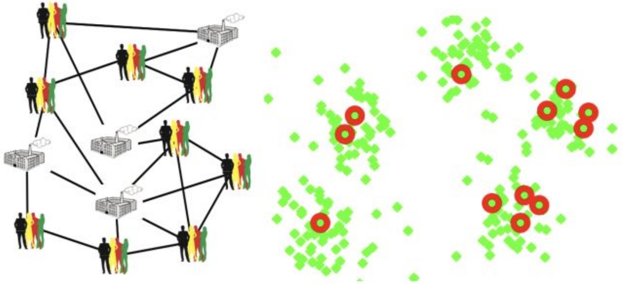
Function	$\boldsymbol{f}(X)$
Representative	
Facility Location	$\sum_{i \in \mathcal{V}} \max_{j \in X} s_{ij}$
Graph Cut	$\sum_{i \in \mathcal{V}, j \in X} s_{ij} - \rho \sum_{i,j \in X} s_{ij}$
Diversity	
Log Determinant	$\log\det(\mathcal{S}_X)$
Disparity-Min	$\min_{i \neq j \in X}(1 - s_{ij})$
Disparity-Sum	$\sum_{i \neq j \in X}(1 - s_{ij})$

Table 1: *Submodular functions used in arm definitions. $\mathcal{V}$ is the ground set, $X \subseteq \mathcal{V}$, s_{ij} denotes pairwise similarity, and $\mathcal{S}_X$ is the similarity submatrix. ρ indicates the balancing factor between representative and diversity nature. We also utilise mutual information variants (Details in Appendix)*

2 Broad Categories

NEURAL INFORMATION PROCESSING SYSTEMS

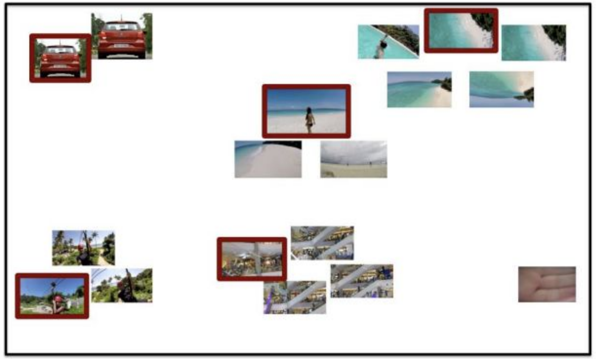
Representative Based Submodular Functions



Facility Location	$\sum_{i \in V} \max_{k \in X} s_{ik}$
Saturated Coverage	$\sum_{i \in V} \min\{\sum_{j \in X} s_{ij}, \alpha_i\}$
Graph Cut	$\lambda \sum_{i \in V} \sum_{j \in X} s_{ij} - \sum_{i,j \in X} s_{ij}$

↑

Similarity Kernel



Representation Functions

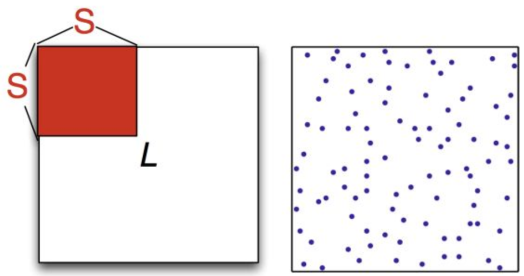
Picks Centroids

Iyer 2015, Kaushal et al 2019, Tschiatchek et al 2014, …

Facility Location (k-Mediods)

NEURAL INFORMATION PROCESSING SYSTEMS

Diversity Based Submodular Functions

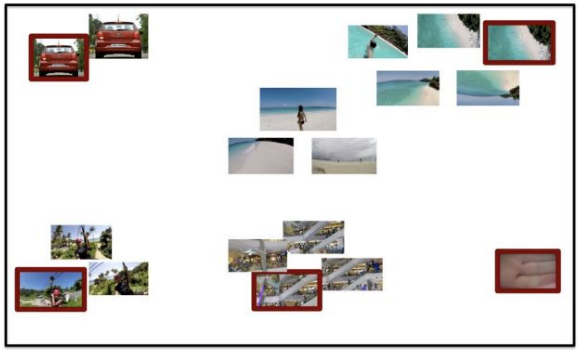


Determinantal Point Processes

$$F(S) = \log\det(L_S)$$

↑

Similarity Kernel



Diversity Functions

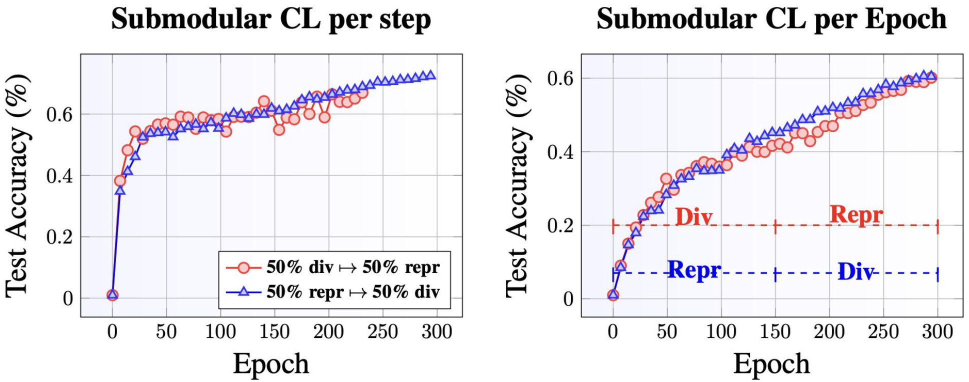
Picks items as different as possible!

Kulesza-Taskar 2012, ...

DisparityMin,
DisparitySum,
LogDeterminant

NEURAL INFORMATION PROCESSING SYSTEMS

Key Observation: Representative First, Diversity Later



[1] observed that beginning training with samples chosen by **representative** functions, and gradually transitioning to **diversity** function selections in later stages, led to improved overall performance.

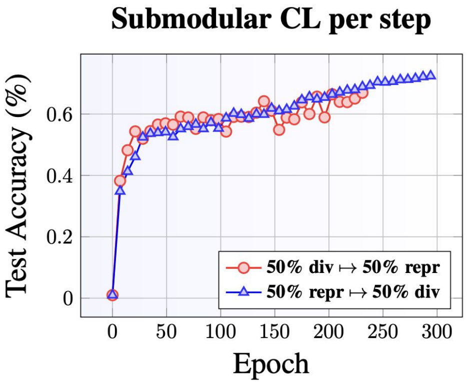
Sequential Ordering of Submodular Functions on CIFAR100: Training first with representation-based subsets, then diversity-based ones, yields better performance than the reverse. (**a**) First 50% of steps per epoch (**micro**). (**b**) First 50% of epochs (**macro**).

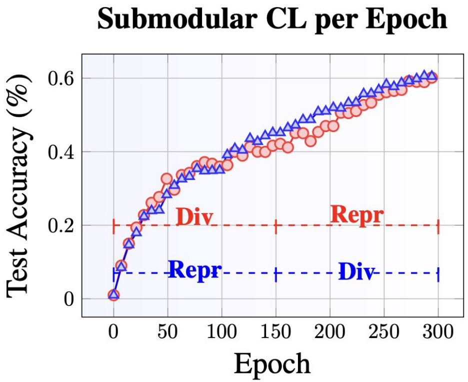
We observe the same phenomenon (both macro and micro level)

NEURAL INFORMATION PROCESSING SYSTEMS

[1] Milo: Model-agnostic subset selection framework for efficient model training and tuning

Key Observation: Representative First, Diversity Later





[1] observed that beginning training with samples chosen by **representative** functions, and gradually transitioning to **diversity** function selections in later stages, led to improved overall performance.

Sequential Ordering of Submodular Functions on CIFAR100: Training first with representation-based subsets, then diversity-based ones, yields better performance than the reverse. (**a**) First 50% of steps per epoch (**micro**). (**b**) First 50% of epochs (**macro**).

Representative => Easy
Diversity => Difficult

Correct Assumption?

NEURAL INFORMATION PROCESSING SYSTEMS

[1] Milo: Model-agnostic subset selection framework for efficient model training and tuning

Is there a more principled approach?

$$\mathscr{A} := \left\{ \boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, \ldots, \boldsymbol{f}^{(\mathcal{K})} \right\}$$

Dynamic/ Adaptive

Can we learn a Principled ordering of these submodular functions ?

Guided by Model Performance

Choose an action (submodular function)

Observe reward and Iterate

NEURAL INFORMATION PROCESSING SYSTEMS

Preliminaries

$\mathcal{D}_{\texttt{train}}$	Entire Training Set consisting of n instances
$\mathcal{D}_{\texttt{val}}$	Entire Validation Set consisting of m instances
$\mathbf{z}_i$	i-th training instance in a batch
$\mathcal{B}_t$	Denotes the full sized t-th step train minibatch : $\{\boldsymbol{x}_p\}_{p=1}^{\lvert\mathcal{B}_t\rvert}$
$\mathcal{B}_t^{\texttt{val}}$	Denotes the full sized t-th step validation minibatch
$\mathcal{B}_t^{(<i)}$	Denotes the t-th step minibatch being constructed uptil $\boldsymbol{x}_{i-1}$ i.e. $\{\boldsymbol{x}_p\}_{p=1}^{i-1}$
$\mathcal{S}_{(a_t)}^{\texttt{opt}}$	Optimal subset obtained when submodular function $f^{(a_t)}$ is applied

Submodular Maximization

$$\mathcal{B}^{opt} = \underset{\mathcal{B}\subseteq\mathcal{V};\lvert\mathcal{B}\rvert\leq\beta}{\operatorname{argmax}} \ \boldsymbol{f}(\mathcal{B}).$$

NEURAL INFORMATION PROCESSING SYSTEMS

Validation Performance as a Reward Utility Metric

$$\mathcal{U}_t(\mathcal{B}_t, \mathbf{z}_{\text{val}}) = \ell(\mathbf{z}_{\text{val}}, \boldsymbol{\theta}_t) - \ell(\mathbf{z}_{\text{val}}, \tilde{\boldsymbol{\theta}}_{t+1}(\mathcal{B}_t)),$$

Utility Function over entire batch for **single validation point**

Difference in Validation Loss b/w two consecutive iterates

$$\tilde{\boldsymbol{\theta}}_{t+1}(\mathcal{B}_t) = \boldsymbol{\theta}_t - \eta_t \nabla_{\boldsymbol{\theta}} \left(\frac{1}{|\mathcal{B}_t|} \sum_{\mathbf{z} \in \mathcal{B}_t} \ell(\mathbf{z}, \boldsymbol{\theta}_t) \right)$$

NEURAL INFORMATION PROCESSING SYSTEMS

Instance wise Marginal Utility Gain

$$\Delta\mathcal{U}_t(\mathbf{z}_i \mid \mathcal{B}_t^{(<i)}, \mathbf{z}_{\text{val}}) = \mathcal{U}_t(\mathcal{B}_t^{(<i)} \cup \{\mathbf{z}_i\}; \mathbf{z}_{\text{val}}) - \mathcal{U}_t(\mathcal{B}_t^{(<i)}; \mathbf{z}_{\text{val}})$$

Marginal Gain w.r.t $\mathbf{z}_i$

$$\mathcal{B}_t^{(<i)} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{i-1}\}$$

Partially constructed mini batch at time step t

NEURAL INFORMATION PROCESSING SYSTEMS

First order Approximation - Marginal Utility Gain

$$
\begin{aligned}
\Delta \mathcal{U}_t(\mathbf{z}_i \mid \mathcal{B}_t^{(<i)}, \mathbf{z}_{\text{val}}) &= \mathcal{U}_t(\mathcal{B}_t^{(<i)} \cup \{\mathbf{z}_i\}; \mathbf{z}_{\text{val}}) - \mathcal{U}_t(\mathcal{B}_t^{(<i)}; \mathbf{z}_{\text{val}}) \\
&\approx \eta_t \nabla_{\boldsymbol{\theta}} \ell(\mathbf{z}_i, \boldsymbol{\theta}_t) \cdot \nabla_{\boldsymbol{\theta}} \ell(\mathbf{z}_{\text{val}}, \boldsymbol{\theta}_{t+1}(\mathcal{B}_t^{(<i)}))
\end{aligned}
$$

Second order Approximation - Marginal Utility Gain

$$\eta_t \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_i) \cdot \nabla_{\boldsymbol{\theta}} \ell(\mathbf{z}_{\text{val}}, \boldsymbol{\theta}_{t+1}(\mathcal{B}_t^{(<i)})) \approx \eta_t \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_i) \cdot \nabla_{\boldsymbol{\theta}} \ell(\mathbf{z}_{\text{val}}, \boldsymbol{\theta}_t - \eta_t \frac{1}{|\mathcal{B}_t^{(<i)}|} \sum_{\mathbf{z} \in \mathcal{B}_t^{(<i)}} \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}))$$

$$\approx \eta_t \underbrace{\boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_i) \cdot \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_{\text{val}})}_{\text{Gradient Influence Function}(\textbf{Term I})} \underbrace{- \eta_t^2 \, \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_i)^\top \boldsymbol{\mathcal{H}}_{\mathbf{z}_{\text{val}}}(\boldsymbol{\theta}_t) (\frac{1}{|\mathcal{B}_t^{(<i)}|} \sum_{\mathbf{z} \in \mathcal{B}_t^{(<i)}} \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}))}_{\text{Hessian Weighted Relative Similarity}(\textbf{Term II})}$$

Abbrv. $\boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_i) = \nabla_{\boldsymbol{\theta}} \ell(\mathbf{z}_i, \boldsymbol{\theta}_t)$

NEURAL INFORMATION PROCESSING SYSTEMS

Expected Marginal Utility Gain

$$\mathbb{E}_{\mathbf{z}_i \in \mathcal{B}_t^{(<i)}} \left[\boldsymbol{\Delta}\mathcal{U}_t(\mathbf{z}_i \mid \mathcal{B}_t^{(<i)}, \mathbf{z}_t^{\mathsf{val}}) \right] \triangleq \eta_t \bar{\mathbf{g}}_{\boldsymbol{\theta}_t}^{(b)} \cdot \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z}_t^{\mathsf{val}}) - \eta_t^2 \bar{\mathbf{g}}_{\boldsymbol{\theta}_t}^{(b)\top} \left(\mathbf{I}_d - \frac{1}{|\mathcal{B}_t|} \mathbf{1}_{d \times |\mathcal{B}_t|} \mathbf{G}_{\boldsymbol{\theta}_t}^{\top} \right) \boldsymbol{\mathcal{H}}_{\mathbf{z}_t^{\mathsf{val}}}(\boldsymbol{\theta}_t) \bar{\mathbf{g}}_{\boldsymbol{\theta}_t}^{(b)}.$$

Computed over entire training batch w.r.t sample validation point

NEURAL INFORMATION PROCESSING SYSTEMS

Reward Utility w.r.t Submodular Function

$a_t \in \mathscr{A}$, we compute an approximately optimal subset $\mathcal{S}_{a_t}^{\mathtt{opt}} \subseteq \mathcal{B}_t$ of size at most β, chosen to maximize the submodular objective $f^{(a_t)}(\mathcal{S})$.

$$\vartheta(a_t \mid \mathcal{B}_t) = \mathbb{E}_{\mathbf{z}_t^{\mathsf{val}} \in \mathcal{B}_t^{\mathsf{val}}, \mathbf{z}_i \in \mathcal{S}_{a_t}^{\mathtt{opt}}} \left[\Delta \mathcal{U}_t(\mathbf{z}_i \mid \mathcal{S}_{a_t}^{\mathtt{opt}\,(<i)}, \mathbf{z}_t^{\mathsf{val}}) \right]$$

$$\hat{a}_t = \arg\max_{a_t \in \mathscr{A}} (\vartheta(a_t \mid \mathcal{B}_t))$$

Highest utility among all current actions

NEURAL INFORMATION PROCESSING SYSTEMS

Optimality Gap

$$\boldsymbol{\Delta}_{(a_t)}(\mathcal{B}_t) := \max\{0,\ \boldsymbol{\vartheta}(a_t^* \mid \mathcal{B}_t) - \boldsymbol{\vartheta}(a_t \mid \mathcal{B}_t)\}$$

$$\sum_{t=1}^{T} \boldsymbol{\Delta}_{a_t}(\mathcal{B}_t)$$

Cumulative Regret

NEURAL INFORMATION PROCESSING SYSTEMS

Back to Curriculum Learning

Recap: We learned that **specific ordering** of submodular functions **leads to better convergence**

Research Question: Can we learn this ordering? 💡

NEURAL INFORMATION PROCESSING SYSTEMS

Algorithm 1: ONLINESUBMOD

Input: $T \in \mathbb{N}$: Total training steps
$\{\boldsymbol{f}^{(a)}\}_{a=1}^{\mathcal{K}}$: Candidate submodular arms
$\lambda(\cdot),\ \pi(\cdot)$: Time-varying exploration parameters

Output: $\boldsymbol{\theta}_{T+1}$: Final model parameter

for $t = 1$ **to** T **do**

- **Receive** batch $\mathcal{B}_t$
- **Sample** $\zeta \sim \mathcal{U}(0, 1)$
- **Threshold**: $\Xi_t \leftarrow \frac{t}{(t+\lambda(t))^{\pi(t)}}$
- $\hat{a}_t \leftarrow \begin{cases} \arg\max_{a_t \in \mathscr{A}} \boldsymbol{\vartheta}(a_t \mid \mathcal{B}_t) & \text{if } \zeta > \Xi_t \\ \text{Uniform}(\mathscr{A}) & \text{otherwise} \end{cases}$
- $\mathcal{S}_{(\hat{a}_t)} \leftarrow \arg\max_{|\mathcal{S}| \leq \beta, \mathcal{S} \subseteq \mathcal{B}_t} \boldsymbol{f}^{(\hat{a}_t)}(\mathcal{S})$
- $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \frac{\eta_t}{|\mathcal{S}_{(\hat{a}_t)}|} \sum_{\mathbf{z} \in \mathcal{S}_{(\hat{a}_t)}} \boldsymbol{g}_{\boldsymbol{\theta}_t}(\mathbf{z})$

return $\boldsymbol{\theta}_{T+1}$

Exploration

Choose at random any arm

Exploitation

$$\hat{a}_t = \arg\max_{a_t \in \mathscr{A}} (\boldsymbol{\vartheta}(a_t \mid \mathcal{B}_t))$$

Highest utility among all current actions

NEURAL INFORMATION PROCESSING SYSTEMS

Large Language Models - OnlineSubmod outperforms other baselines

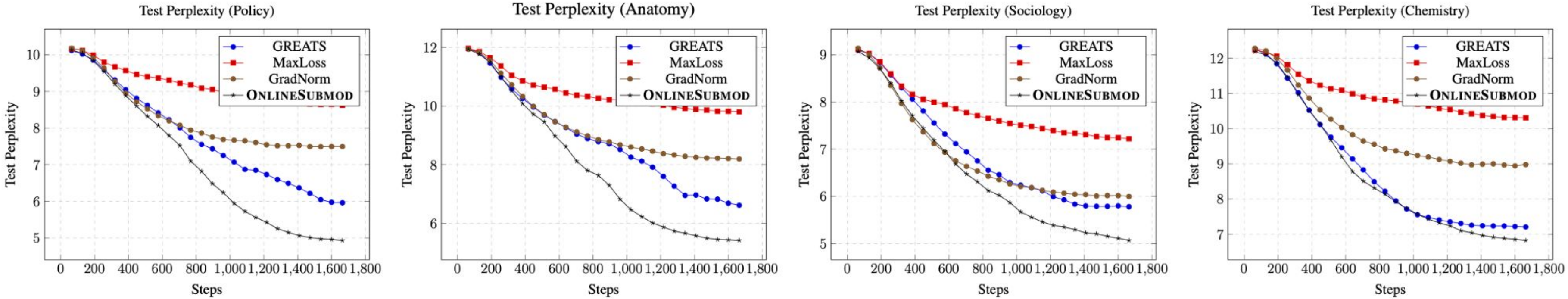


Figure 2: **Test perplexity dynamics** on `LLAMA-2-7B` during training with various **online batch selection strategies** on `MMLU`. We evaluate on `US Foreign Policy`, `Anatomy`, `Sociology`, and `Chemistry`. ONLINESUBMOD significantly outperforms baselines.

NEURAL INFORMATION PROCESSING SYSTEMS

Image Models - OnlineSubmod outperforms other baselines

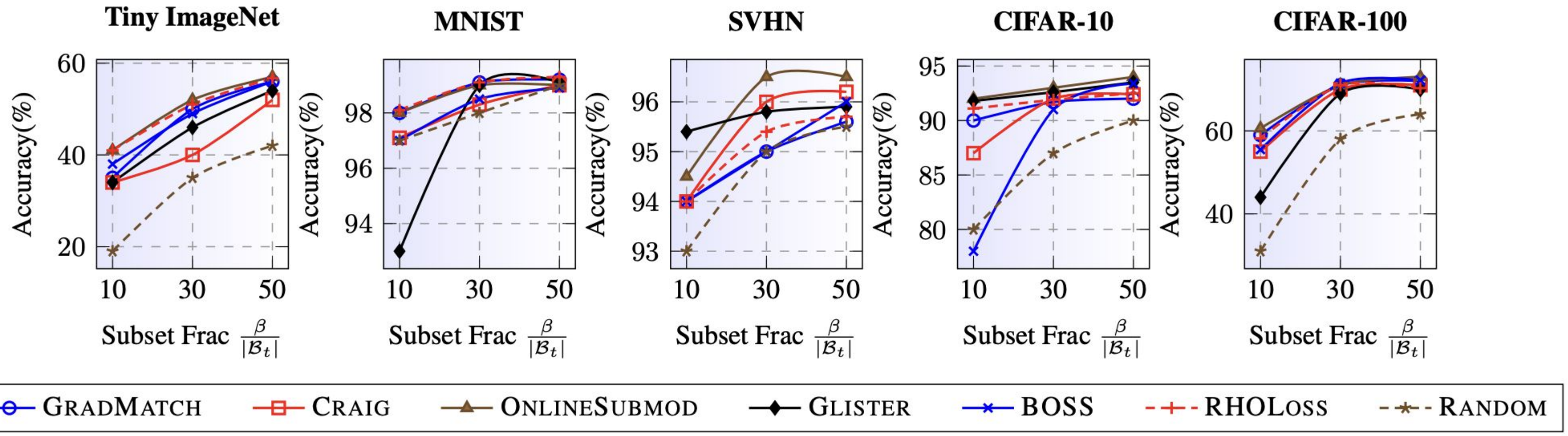


Figure 3: **Samplewise Submodular Curriculum:** ONLINESUBMOD consistently achieves top-1 accuracy across all subset sizes on TINYIMAGENET, SVHN, CIFAR-10, and CIFAR-100, and remains competitive on MNIST. Notably, it matches or outperforms all baselines at early subset fractions (10%, 30%) on all datasets except MNIST.

NEURAL INFORMATION PROCESSING SYSTEMS

Theoretical Results

Theorem 1 (Regret Guarantees). *Under Assumptions **a - d**, for all $t > t_0$, with probability at least*

$$1 - \mathcal{K} \exp\left(-\frac{3(t-2)(1+(1-\pi)\epsilon)}{28\mathcal{K}(2-\pi)}\right),$$

the expected instantaneous regret incurred by the arm selection policy satisfies

$$\begin{aligned}\mathbb{E}[\text{Regret}_t] &:= \mathbb{E}_{\mathcal{B}_t}\mathbb{E}_{\hat{a}_t \in \mathscr{A}}\mathbb{E}_{\boldsymbol{\vartheta}}\left[\boldsymbol{\vartheta}(a_t^* \mid \mathcal{B}_t) - \boldsymbol{\vartheta}(\hat{a}_t \mid \mathcal{B}_t)\right] \\ &= O\left(\frac{1}{t}\right) + O\left(\frac{\mathcal{K}^{3/2}(\max_a \mathfrak{C}_{(a)} + \mathfrak{C}_*)}{\boldsymbol{\varrho}}\sqrt{\frac{\log t}{t}}\right),\end{aligned} \tag{8}$$

where $\mathfrak{C}_*$ is the approximation constant corresponding to the optimal arm a^*.

NEURAL INFORMATION PROCESSING SYSTEMS

For more details : Scan

Thank You

NEURAL INFORMATION PROCESSING SYSTEMS