Teaching Multiple Concepts to a Forgetful Learner

Anette Hunziker[†]   **Yuxin Chen**[¶]   **Oisin Mac Aodha**[§]   **Manuel Gomez Rodriguez**[*]
Andreas Krause[‡]   **Pietro Perona**[⋆]   **Yisong Yue**[⋆]   **Adish Singla**[*]

[†]University of Zurich, `anette.hunziker@gmail.com`,
[¶]University of Chicago, `chenyuxin@uchicago.edu`,
[§]University of Edinburgh, `oisin.macaodha@ed.ac.uk`,
[‡]ETH Zurich, `krausea@ethz.ch`,
[⋆]Caltech, `{perona, yyue}@caltech.edu`,
[*]MPI-SWS, `{manuelgr, adishs}@mpi-sws.org`

Abstract

How can we help a forgetful learner learn multiple concepts within a limited time frame? While there have been extensive studies in designing optimal schedules for teaching a *single* concept given a learner's memory model, existing approaches for teaching *multiple* concepts are typically based on heuristic scheduling techniques without theoretical guarantees. In this paper, we look at the problem from the perspective of discrete optimization and introduce a novel algorithmic framework for teaching multiple concepts with strong performance guarantees. Our framework is both generic, allowing the design of teaching schedules for different memory models, and also interactive, allowing the teacher to adapt the schedule to the underlying forgetting mechanisms of the learner. Furthermore, for a well-known memory model, we are able to identify a regime of model parameters where our framework is guaranteed to achieve high performance. We perform extensive evaluations using simulations along with real user studies in two concrete applications: (i) an educational app for online vocabulary teaching; and (ii) an app for teaching novices how to recognize animal species from images. Our results demonstrate the effectiveness of our algorithm compared to popular heuristic approaches.

1 Introduction

In many real-world educational applications, human learners often intend to learn more than one concept. For example, in a language learning scenario, a learner aims to memorize many vocabulary words from a foreign language. In citizen science projects such as eBird [34] and iNaturalist [38], the goal of a learner is to recognize multiple animal species from a given geographic region. As the number of concepts increases, the learning problem can become very challenging due to the learner's limited memory and propensity to forget. It has been well established in the psychology literature that in the context of *human learning*, the knowledge of a learner decays rapidly without reconsolidation [7]. Somewhat analogously, in the sequential *machine learning* setting, modern machine learning methods, such as artificial neural networks, can be drastically disrupted when presented with new information from different domains, which leads to catastrophic interference and forgetting [19, 14]. Therefore, to retain long-term memory (for both human and machine learners), it is crucial to devise teaching strategies that adapt to the underlying forgetting mechanisms of the learner.

Teaching forgetful learners requires repetition. Properly scheduled repetitions and reconsolidations of previous knowledge have proven effective for a wide variety of real-world learning tasks, including piano [30], surgery [39, 33], video games [29], and vocabulary learning [4], among others. For many of the above applications, it has been shown that by carefully designing the scheduling policy, one can
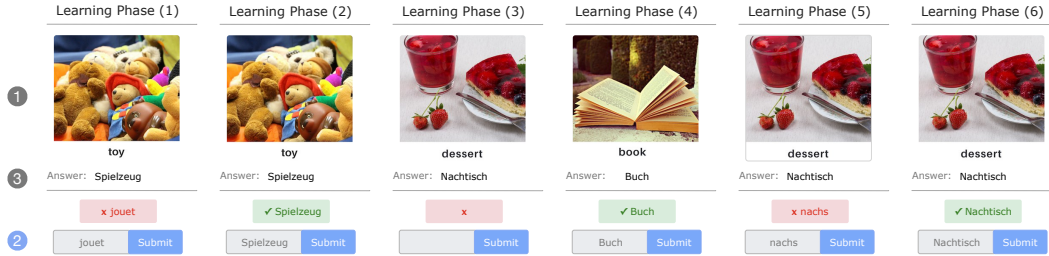
Figure 1: Illustration of our adaptive teaching framework applied to German vocabulary learning, shown here for six time steps in the learning phase. Each time step proceeds in three stages: (1) the system displays a flashcard with an image and its English description, (2) the learner inputs the German translation, and (3) the system provides feedback in the form of the correct answer if the input is incorrect.

achieve substantial gains over simple heuristics (such as spaced repetition at fixed time intervals, or a simple round robin schedule) [3]. Unfortunately, while there have been extensive (theoretical) results in teaching a single concept using spaced repetition algorithms, existing approaches for teaching multiple concepts are typically based on heuristics without theoretical guarantees.

In this paper, we explore the following research question: *Given limited time, can we help a forgetful learner efficiently learn multiple concepts in a principled manner?* More concretely, we consider an adaptive setting where at each time step, the teacher needs to pick a concept from a finite set based on the learner's previous responses, and the process iterates until the learner's time budget is exhausted. Given a memory model of the learner, what is an optimal teaching curriculum? How should this sequence be adapted based on the learner's performance history?

1.1 Overview of our approach

For a high-level overview of our approach, consider the example in Fig. 1, which illustrates one of our applications on German vocabulary learning [2]. Here, our goal is to teach the learner three German words in six time steps. One trivial approach could be to show the flashcards in a round robin fashion. However, the round robin sequence is deterministic and thus not capable of adapting to the learner's performance. In contrast, our algorithm outputs an adaptive teaching sequence based on the learner's performance.

Our algorithm is based on a novel formulation of the adaptive teaching problem. In §2, we propose a novel discrete optimization problem, where we seek to maximize a natural surrogate objective function that characterizes the learner's expected performance throughout the teaching session. Note that constructing the optimal teaching policy boils down to solving a stochastic sequence optimization problem, which is NP-hard in general. In §3, we introduce our greedy algorithm, and derive performance guarantees based on two intuitive data-dependent properties. While it can be challenging to compute these performance bounds, we show that for certain learner memory models, these bounds can be estimated efficiently. Furthermore, we identify parameter settings of the memory models where the greedy algorithm is guaranteed to achieve high performance. Finally, we demonstrate that our algorithm achieves significant improvements over baselines for both simulated learners (cf. §4) and human learners (cf. §5).

2 The Teaching Model

We now formalize the problem addressed in this paper.

2.1 Problem setup

Suppose that the teacher aims to teach the learner n concepts in a finite time horizon T. We highlight the notion of a concept via two concrete examples: (i) when teaching the vocabulary of a foreign language, each concept corresponds to a word, and (ii) when teaching to recognize different animal species, each concept corresponds to an animal name. We consider flashcard-based teaching, where each concept is associated with a flashcard (cf. Fig. 1).

We study the following interactive teaching protocol: At time step t, the teacher picks a concept from the set $\{1, \ldots, n\}$ and presents its corresponding flashcard to the learner without revealing its correct answer. The learner then tries to recall the concept. Let us use $y_t \in \{0, 1\}$ to denote the learner's recall at time step t. Here, $y_t = 1$ means that the learner successfully recalls the concept (e.g., the learner correctly recognizes the animal species), and $y_t = 0$ otherwise. After the learner makes an attempt, the teacher observes the outcome y_t and reveals the correct answer.

2.2 Learner's memory model

Let us use (σ, y) to denote any sequence of concepts and observations. In particular, we use $\sigma_{1:t}$ to denote the sequence of concepts picked by the teacher up to time t. Similarly, we use $y_{1:t}$ to denote the sequence of observations up to time t. Given the history $(\sigma_{1:t}, y_{1:t})$, we are interested in modeling the learner's probability to recall concept i at a future time $\tau \in [t+1, T]$. In general, the learner's probability to recall concept i could depend on the history of teaching concept i or related concepts.[1] Formally, we capture the learner's recall probability for concept i by a memory model $g_i(\tau, (\sigma_{1:t}, y_{1:t}))$ that depends on the entire history (σ, y). In §3.2, we study an instance of the learner model captured by exponential forgetting curve (see Eq. (9)).

2.3 The teaching objective

There are several objectives of interest to the teacher, for instance, maximizing the learner's performance in recalling all concepts measured at the end of the teaching session. However, given that the learning phase might stretch over a long time duration for language learning, another natural objective is to measure learner's performance across the entire teaching session. For any given sequence of concepts and observations $(\sigma_{1:T}, y_{1:T})$ of length T, we consider the following objective:

$$f(\sigma_{1:T}, y_{1:T}) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} g_i(\tau + 1, (\sigma_{1:\tau}, y_{1:\tau})). \tag{1}$$

Here, $g_i(\cdot)$ denotes the recall probability of concept i at $\tau + 1$, given the history up to time step τ. Concretely, for a given concept $i \in [n]$, our objective function can be interpreted as the (discrete) area under the learner's recall curve for concept i across the teaching session.

The teacher's teaching strategy can be represented as a *policy* $\pi : (\sigma, y) \rightarrow \{1, \ldots, n\}$, which maps any history (i.e., sequence of concepts selected σ and observations y) to the next concept to be taught. For a given policy π, we use $(\sigma_{1:T}^{\pi}, y_{1:T}^{\pi})$ to denote a random trajectory from the policy until time T. The average utility of a policy π is defined as:

$$F(\pi) = \mathbb{E}_{\sigma^{\pi}, y^{\pi}} [f(\sigma_{1:T}^{\pi}, y_{1:T}^{\pi})]. \tag{2}$$

Given the learner's memory model for each concept i and the time horizon T, we seek the optimal teaching policy that achieves the maximal average utility:

$$\pi^* \in \max_{\pi} F(\pi). \tag{3}$$

It can be shown that finding the optimal solution for Eq. (3) is NP-hard (proof is provided in the supplemental materials).

Theorem 1. *Problem* (3) *is NP-hard, even when the objective function does not depend on the learner's responses.*

3 Teaching Algorithm and Analysis

We now present a simple, greedy approach for constructing teaching policies. To measure the teaching progress at time $t < T$, we introduce the following generalization of objective defined in Eq. (1):

---

[1]As an example, for German vocabulary learning, the recall probability for the concept "Apfelsaft" (apple juice) could depend on the flashcards shown for "Apfelsaft" and "Apfel" (apple).

$$f(\sigma_{1:t}, y_{1:t}) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} g_i \left(\tau + 1, \left(\sigma_{1:\min(\tau,t)}, y_{1:\min(\tau,t)} \right) \right). \tag{4}$$

Note that this is equivalent to extending $(\sigma_{1:t}, y_{1:t})$ to length T by filling in the remaining sequence from $t+1$ to T with empty concepts and observations. Given the history $(\sigma_{1:t-1}, y_{1:t-1})$, we define the conditional marginal gain of teaching a concept i at time t as:

$$\Delta \left(i \mid \sigma_{1:t-1}, y_{1:t-1} \right) = \mathbb{E}_{y_t} \left[f(\sigma_{1:t-1} \oplus i, y_{1:t-1} \oplus y_t) - f(\sigma_{1:t-1}, y_{1:t-1}) \right], \tag{5}$$

where $\oplus$ denotes the concatenation operation, and the expectation is taken over the randomness of learner's recall y_t, conditioned on the history $(\sigma_{1:t-1}, y_{1:t-1})$. The greedy algorithm, as described in Algorithm 1, iteratively selects the concept that maximizes this conditional marginal gain.

Algorithm 1 Adaptive Teaching Algorithm

Sequence $\sigma \leftarrow \emptyset$; observation history $y \leftarrow \emptyset$
for $t = \{1, \dots, T\}$ **do**
    Select $i_t \leftarrow \arg\max_i \Delta \left(i \mid \sigma, y \right)$
    Show i_t to the learner; Observe y_t
    Update $\sigma \leftarrow \sigma \oplus i_t, y \leftarrow y \oplus y_t$

3.1 Theoretical guarantees

We now present a general theoretical framework for analyzing the performance of the adaptive teaching algorithm (Algorithm 1). Our bound depends on two natural properties of the objective function f, both related to a notion of *diminishing returns* of a sequence function. Intuitively, the following two properties reflect how much a greedy choice can affect the optimality of the solution.

Definition 1 (Online stepwise submodular coefficient). *Consider policy π for time T. The online submodular coefficient of function f with respect to policy π at step t is defined as*

$$\gamma_t^{\pi} = \min_{\sigma_{1:t}^{\pi}, y_{1:t}^{\pi}} \gamma(\sigma_{1:t}^{\pi}, y_{1:t}^{\pi}) \tag{6}$$

where $\gamma(\sigma, y) = \min_{i,(\sigma',y'):|\sigma|+|\sigma'|<T} \frac{\Delta(i|\sigma,y)}{\Delta(i|\sigma\oplus\sigma', y\oplus y')}$ denotes the minimal ratio between the gain of any concept i given the current history (σ, y) and the gain of i in any future steps.

Definition 2 (Online stepwise backward curvature). *Consider policy π for time T. The online backward curvature of function f with respect to policy π at step t is defined as*

$$\omega_t^{\pi} = \max_{\sigma_{1:t}^{\pi}, y_{1:t}^{\pi}} \omega(\sigma_{1:t}^{\pi}, y_{1:t}^{\pi}) \tag{7}$$

where $\omega(\sigma, y) = \max_{\sigma^{\pi'}, y^{\pi'}} \left[1 - \frac{f(\sigma \oplus \sigma^{\pi'}, y \oplus y^{\pi'}) - f(\sigma^{\pi'}, y^{\pi'})}{f(\sigma, y) - f(\emptyset)} \right]$ denotes the normalized maximal second-order difference when considering the current history (σ, y).

Here, $\gamma(\sigma, y)$ and $\omega(\sigma, y)$ generalizes the notion of *string submodularity* and *total backward curvature* for sequence functions [43] to the stochastic setting. Intuitively, $\gamma(\sigma, y)$ measures the degree of diminishing returns of a sequence function in terms of the *ratio* between the conditional marginal gains. If $\forall(\sigma, y), \gamma(\sigma, y) = 1$, then the conditional marginal gain of adding any concept to any subsequent observation history is non-decreasing. In contrast, $\omega(\sigma, y)$ measures the degree of diminishing returns in terms of the *difference* between the marginal gains. As our first main theoretical result, we provide a data-dependent bound on the average utility of the greedy policy against the optimal policy.

Theorem 2. *Let π^g be the online greedy policy induced by Algorithm 1, and F be the objective function as defined in Eq. (2). Then for all policies π^*,*

$$F(\pi^g) \geq F(\pi^*) \sum_{t=1}^{T} \left(\frac{\gamma_{T-t}^g}{T} \prod_{\tau=0}^{t-1} \left(1 - \frac{\omega_\tau^g \gamma_\tau^g}{T} \right) \right), \tag{8}$$

where γ_t^g and ω_t^g denote the online stepwise submodular coefficient and online stepwise backward curvature of f with respect to the policy π^g at time step t.

The summand on the R.H.S. of Eq. (8) is in fact a lower bound on the expected one-step gain of the greedy policy. We can further relax the bound by considering the worst-case online stepwise submodularity ratio and curvature across all time steps, given by the following corollary.

Corollary 3. *Let* $\gamma^g = \min_t \gamma_t^g$ *and* $\omega^g = \max_t \omega_t^g$. *For all* π^*, $F(\pi^g) \geq \frac{1}{\omega^g}\left(1 - e^{-\gamma^g \omega^g}\right) F(\pi^*)$.

Note that Corollary 3 generalizes the string submodular optimization result from [43] to the stochastic setting. In particular, for the special case where $\gamma^g = \omega^g = 1$ and $f(\sigma_{1:t}, y_{1:t})$ is independent of $y_{1:t}$, Corollary 3 reduces to $f(\sigma^g, \cdot) \geq \left(1 - e^{-1}\right) f(\sigma^*, \cdot)$ where σ^g, σ^* denote the sequences selected by the greedy and the optimal algorithm. However, constructing the bounds in Theorem 2 and Corollary 3 requires us to compute γ_t^g, ω_t^g, which is as expensive as computing $F(\pi^*)$. In the following subsection, we investigate a specific learning setting, and provide a polynomial time approximation algorithm for computing the theoretical lower bound in Theorem 2.

3.2 Analysis for HLR memory model

Here, we consider the setting where the learner's memory for each concept $i \in [n]$ is captured by an independent HLR memory model. Concepts being independent means that the memory model $g_i(\tau, (\sigma_{1:t}, y_{1:t}))$ for concept i only depends on the history when flashcards for concept i was shown.[2]

More specifically, we consider the case of an HLR memory model with the following exponential forgetting curve [28]:

$$g_i(\tau, (\sigma_{1:t}, y_{1:t})) = 2^{-\frac{\tau - \ell_i}{h_i}}, \tag{9}$$

where ℓ_i is the last time concept i was taught, and $h_i = 2^{\langle \theta_i, n_i \rangle}$ denotes the half life of the learner's recall probability of concept i. Here, $\theta_i = (a_i, b_i, c_i)$ parameterizes the learner's retention rate, and $n_i = (n_+^i, n_-^i, 1)$, where $n_+^i := |\{\tau' \in [t] : \sigma_{\tau'} = i \wedge y_{\tau'} = 1\}|$ and $n_-^i := |\{\tau' \in [t] : \sigma_{\tau'} = i \wedge y_{\tau'} = 0\}|$ denote the number of correct and incorrect recalls of concept i in $(\sigma_{1:t}, y_{1:t})$, respectively. Intuitively, a_i scales n_+^i, b_i scales n_-^i, and c_i is an offset that can be considered as scaling time.

We would like to bound the performance of Algorithm 1. While computing γ_t^g, ω_t^g is intractable in general, we show that one can efficiently approximate γ_t^g, ω_t^g for the HLR model with $a_i = b_i$.

Theorem 4. *Assume that the learner is characterized by the HLR model (Eq. (9)) where* $\forall i, a_i = b_i$. *We can compute empirical bounds on* γ_t, ω_t *in polynomial time.*

Theorem 4 shows that it is feasible to compute explicit lower bounds on the utility of Algorithm 1 against the maximal achievable utility. The following theorem shows that for certain types of learners, the algorithm is guaranteed to achieve a high utility.

Theorem 5. *Consider the task of teaching n concepts where each concept is following an independent HLR memory model sharing the same parameters, i.e.,* $\forall i, \theta_i = (a, a, 0)$. *A sufficient condition for the algorithm to achieve $1 - \epsilon$ utility is* $a \geq \max\left\{\log T, \log(3n), \log\left(\frac{2n^2}{\epsilon T}\right)\right\}$, *where the parameter a essentially captures the learner's memory strength.*

Note that Theorem 5 provides a sufficient condition for our algorithm to achieve a high utility. One interesting open question is to establish an upper bound for the greedy (or the optimal) algorithm under particular model configurations, e.g., to provide a necessary condition for achieving a certain target utility under the HLR model.

4 Simulations

In this section, we experimentally evaluate our algorithm by simulating learner responses based on a known memory model. This allows us to inspect the behavior of our algorithm and compare it with several baseline algorithms in a controlled setting.

---

[2]We note that the hardness result of Theorem 1 doesn't directly apply to this setting with independent concepts. Nevertheless, the problem setting is still computationally challenging. If we express the optimal solution using Bellman equations and apply dynamic programming, the number of states will be exponential in the number of concepts n and polynomial w.r.t. time horizon T.

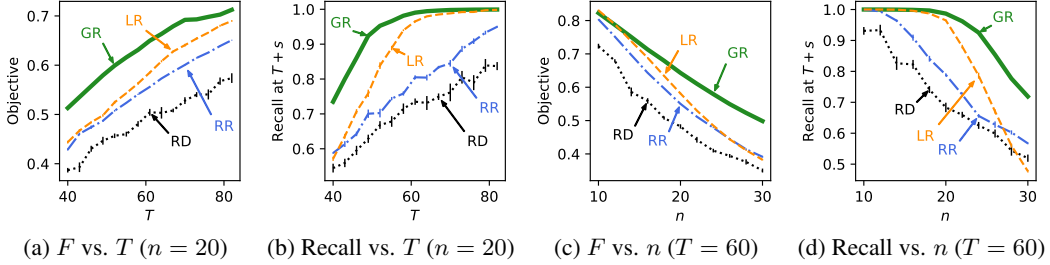| (a) F vs. T $(n = 20)$ | (b) Recall vs. T $(n = 20)$ | (c) F vs. n $(T = 60)$ | (d) Recall vs. n $(T = 60)$ |

Figure 2: Simulation results comparing our algorithm (GR) and three baseline algorithms (RD, RR, and LR). Two performance metrics are considered: (i) the objective value in Eq. (4) and (ii) recall at the end of the teaching session denoted at 'Recall at $T + s$" with $s = 10$.

4.1 Experimental setup

Dataset   We simulated concepts of two different types: "easy" and "difficult". The learner's memory for each concept is captured by an independent HLR model. Concepts of the same type share the same parameter configurations. Specifically, for "easy" concepts the parameters are $\theta_1 = (a_1 = 10, b_1 = 5, c_1 = 0)$, and for "difficult" concepts the parameters are $\theta_2 = (a_2 = 3, b_2 = 1.5, c_2 = 0)$, with the following interpretation in terms of recall probabilities. For "easy" concepts, the recall probability of a concept i drops to $g_i\left(\tau = 2, (\sigma_1 = i, y_1 = 1)\right) = 2^{-1/(2^{a_1+c_1})} = 0.99$ and $g_i\left(\tau = 2, (\sigma_1 = i, y_1 = 0)\right) = 2^{-1/(2^{b_1+c_1})} = 0.98$ in the immediate next step after showing concept i. For "difficult" concepts these probabilities are $(0.92, 0.78)$.

Evaluation metric   We consider two different criteria when assessing the performance. Our first evaluation metric is the objective value as defined in Eq. (4), which measures the learner's average cumulative recall probability *across* the entire teaching session. The second evaluation metric is the learner's average recall probability at the *end* of the teaching session. We call this second objective "Recall at $T + s$", where $s > 0$ denotes how far in the future we choose to evaluate the learner's recall.

Baselines   To demonstrate the performance of our adaptive greedy policy (referred to as GR), we consider three baseline algorithms. The first baseline, denoted by RD, is a random teacher that presents a random concept at each time step. The second baseline, denoted by RR, is a round robin teaching policy that picks concepts according to a fixed round robin schedule, i.e., iterating through concepts at each time step. Our third baseline is a variant of the teaching strategy employed by [28], which can be considered as a generalization of the popular Leitner and Pimsleur systems [16, 25]. At each time step, the teacher chooses to display the concept with the lowest recall probability according to the HLR memory model of the learner. We refer to this algorithm as LR.

4.2 Simulation results

We first evaluate the performance as a function of the teaching horizon T. In Fig. 2a and Fig. 2b, we plot the objective value and average recall at $T + s$ for all algorithms over 10 random trials, where we set $s = 10$, $n = 20$ with half easy and half difficult concepts, and vary $T \in [40, 80]$. As we can see from both plots, GR consistently outperforms baselines in all scenarios. The gap between the performances of GR and the baselines is more significant for smaller T. As we increase the time budget, the performance of all algorithms improves—this behavior is expected, as it corresponds to the scenario where all concepts get a fair chance of repetition with abundant time budget. In Fig. 2c and Fig. 2d, we show the performance plot for a fixed teaching horizon of $T = 60$ when we vary the number of concepts $n \in [10, 30]$. Here we observe a similar behavior as before—GR is consistently better; as n increases, the gap between the performances of GR and the baselines becomes more significant. Our results suggest that the advantage of GR is most pronounced for more challenging settings, i.e., when we have a tight time budget (small T) or a large number of concepts (large n).

	German				Biodiversity			
	GR	LR	RR	RD	GR	LR	RR	RD
avg gain	0.572	0.487	0.462	0.467	0.475	0.411	0.390	0.251
p-value	–	0.0652	0.0197	0.0151	–	0.0017	<0.0001	<0.0001

	Biodiversity (common)				Biodiversity (rare)			
	GR	LR	RR	RD	GR	LR	RR	RD
avg gain	0.143	0.118	0.150	0.086	0.766	0.668	0.601	0.396
p-value	–	0.3111	0.8478	0.0047	–	0.0001	<0.0001	<0.0001

Table 1: Summary of the user study results. Here, the performance is measured as the gain in learner's performance from prequiz phase to postquiz phase (see main text for details). We have $n = 15, T = 40$, and ran algorithms with a total of 80 participants for German app and 320 participants for Biodiversity app.

5   User Study

We have developed online apps for two concrete real-world applications: (i) German vocabulary teaching [2], and (ii) teaching novices to recognize animal species from images, motivated by citizen science projects for biodiversity monitoring [1]. Next, we briefly introduce the datasets used for these two apps and then present the user study results of teaching human learners.

5.1   Experimental setup

Dataset   For the German vocabulary teaching app, we collected 100 English-German word pairs in the form of flashcards, each associated with a descriptive image. These word pairs were provided by a language expert (see [8]) and consist of popular vocabulary words taught in an entry-level German language course. For the biodiversity teaching app, we collected images of 50 animal species. To extract a fine-grained signal for our user study, we further categorize the Biodiversity dataset into two difficulty levels, namely "common" and "rare", based on the prevalence of these species. Examples from both datasets are provided in the supplemental materials.

For real-world experiments, we do not know the learner's memory model. While it is possible to fit the HLR model through an extensive pre-study as in [28], we instead simply choose a fixed set of parameters. For the Biodiversity dataset, we set the parameters of each concept based on their difficulty level. Namely, we set $\theta_1 = (10, 5, 0)$ for "common" (i.e., easy) species and $\theta_2 = (3, 1.5, 0)$ for "rare" (i.e., difficult) species, as also used in our simulation. For the German dataset, since the parameters associated with a concept (i.e., vocabulary word) depend heavily on learner's prior knowledge, we chose a more robust set of parameters for each of the concepts given by $\theta = (6, 2, 0)$. We defer the details of our sensitivity study of the HLR parameters to the supplemental materials.

Online teaching interface   Our apps provide an online teaching interface where a user (i.e., human learner) can participate in a "teaching session". As in the simulations, here each session corresponds to teaching n concepts (sampled randomly from our dataset) via flashcards over T time steps. We demonstrate the teaching interface and present the detailed design ideas in the supplemental materials.

5.2   User study results

Results for German   We now present the user study results for our German vocabulary teaching app [2]. We run our candidate algorithms with $n = 15, T = 40$ on a total of 80 participants (i.e., 20 per algorithm) recruited from Amazon Mechanical Turk. Results are shown in Table 1. where we computed the average gain of each algorithm, and performed statistical analysis on the collected results. The first row (*avg gain*) is obtained by treating the performance for each (participant, word) pair as a separate sample (e.g., we get 20×15 samples per algorithm for the German app). The second row (*p-value*) indicates the statistical significance of the results measured by the χ^2 tests [6] (with contingency tables where rows are algorithms and columns are observed outcomes), when comparing GR with the baselines. Overall, GR achieved higher gains compared to the baselines.

Results for Biodiversity   Next, we present the user study results on our Biodiversity teaching app [1]. We recruited a total of 320 participants (i.e., 80 per algorithm). Here, we used different parameters for the learner's memory as described in §5.1; all other conditions (i.e., $n = 15$, $T = 40$, and interface) were kept the same as for the German app. In Table 1, in addition to the overall performance of the algorithms across all concepts, we also provide separate statistics on teaching the "common" and "rare" concepts. Note that, while the performance of GR is close to the baselines when teaching the "common" species (given the high prequiz score due to learner's prior knowledge about these species), GR is significantly more effective in teaching the "rare" species.

Remarks   This user study provides a proof-of-concept that the performance of our algorithm GR demonstrated on simulated learners is consistent with the performance observed on human learners. While teaching sessions in our current user study were limited to a span of 25 mins with participants recruited from Mechanical Turk, we expect that the teaching applications we have developed could be adapted to real-life educational scenarios for conducting long-term studies.

6   Related Work

Spaced repetition and memory models   Numerous studies in neurobiology and psychology have emphasized the importance of the *spacing* effects in human learning. The spacing effect is the observation that spaced repetition (i.e., introducing appropriate time gaps when learning a concept) produces greater improvements in learning compared to massed repetition (i.e., "cramming") [37]. These findings have inspired many computational models of human memory, including the Adaptive Character of Thought–Rational model (ACT-R) [24], the Multiscale Context model (MCM) [22], and the Half-life Regression model (HLR) [28]. In particular, HLR is a trainable spaced repetition model, which can be viewed as a generalization of the popular Leitner [16] and Pimsleur [25] systems. In this paper, we adopt a variant of HLR to model the learner. One of the key characteristics of these memory models is the function used to model the forgetting curve. Power-law and exponential functions are two popular ways of modeling the forgetting curve (for detailed discussion, see [27, 41, 24, 40]).

Optimal scheduling with spaced repetition models   [13] and [17] studied the ACT-R model and the MCM model respectively for the optimal review scheduling problem where the goal is to maximize a learner's retention through an intelligent review scheduler. One of the key differences between their setting and ours is that, they consider a fixed curriculum of new concepts to teach, and the scheduler additionally chooses which previous concept(s) to review at each step; whereas our goal is to design a complete teaching curriculum. Even though the problem settings are somewhat different, we would like to note that our theoretical framework can be adapted to their setting.

Recently, [26] presented a queuing model for flashcard learning based on the Leitner system and consider a "mean-recall approximation" heuristic to tractably optimize the review schedule. One limitation is that their approach does not adapt to the learner's performance over time. Furthermore, the authors leave the problem of obtaining guarantees for the original review scheduling problem as a question for future work. [35] considered optimizing learning schedules in continuous time for a single concept, and use control theory to derive optimal scheduling to minimize a penalized recall probability area-under-the-curve loss function. In addition to being discrete time, the key difference of our setting is that we aim to teach multiple concepts.

Sequence optimization   Our theoretical framework is inspired by recent results on sequence submodular function maximization [43, 36] and adaptive submodular optimization [10]. In particular, [43] introduced the notion of string submodular functions, which, analogous to the classical notion of submodular set functions [15], enjoy similar performance guarantees for maximization of deterministic sequence functions. Our setting has two key differences in that we focus on the stochastic setting with potentially non-submodular objective functions. In fact, our theoretical framework (in particular Corollary 3) generalizes string submodular function maximization to the adaptive setting.

Forgetful learners in machine learning   Here, we highlight the differences with some recent work in the machine learning literature involving forgetful learners. In particular, [44] aimed to teach the learner a binary classifier by sequentially providing training examples, where the learner has an exponential decaying memory of the training examples. In contrast, we study a different problem, where we focus on teaching multiple concepts, and assume that the learner's memory of each concept

decays over time. [14] explored the problem of how to construct a neural network for learning multiple concepts. Instead of designing the optimal training schedule, their goal is to design a good learner that suffers less from the forgetting behavior.

Machine teaching Our work is also closely related to machine/algorithmic teaching literature (e.g., [46, 45, 32, 9]). Most of these works in machine teaching consider a non-adaptive setting where the teacher provides a batch of teaching examples at once without any adaptation. In this paper, we focus primarily on designing interactive teaching algorithms that adaptively select teaching examples for a learner based on their responses. The problem of adaptive teaching has been studied recently (e.g., [12, 42, 11, 5, 18, 31]). However, these works in machine teaching have not considered the phenomena of forgetting. [23, 21] have studied the problem of concept learning and machine teaching when learner has "limited-capacity" in terms of retrieving exemplars in memory during the decision-making process. They model the learner via the Generalized Context Model [20] and investigated the problem of choosing the optimal exemplars for teaching a *classification* task. In our setting, the exemplars for each class are already given (in other words, we have only one exemplar per class), and we aim at optimally teaching the learner to *memorize* the (label of) exemplars.

7   Conclusions

We presented an algorithmic framework for teaching multiple concepts to a forgetful learner. We proposed a novel discrete formulation of teaching based on stochastic sequence function optimization, and provided a general theoretical framework for deriving performance bounds. We have implemented teaching apps for two real-world applications. We believe our results have made an important step towards bringing the theoretical understanding of algorithmic teaching closer to real-world applications where the forgetting phenomenon is an intrinsic factor.

Acknowledgements

References

[1] App-Biodiversity. Website for teaching animal species. https://www.teaching-biodiversity.cc, 2018.

[2] App-German. Website for teaching German vocabulary. https://www.teaching-german.cc, 2018.

[3] David A Balota, Janet M Duchek, and Jessica M Logan. *Is expanded retrieval practice a superior form of spaced retrieval? A critical review of the extant literature*. Psychology Press New York, NY, 2007.

[4] Kristine C Bloom and Thomas J Shuell. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. *The Journal of Educational Research*, 74(4):245–248, 1981.

[5] Yuxin Chen, Adish Singla, Oisin Mac Aodha, Pietro Perona, and Yisong Yue. Understanding the role of adaptivity in machine teaching: The case of version space learners. In *NeurIPS*, 2018.

[6] William G Cochran. The $\chi 2$ test of goodness of fit. *The Annals of Mathematical Statistics*, pages 315–345, 1952.

[7] Hermann Ebbinghaus. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot, 1885.

[8] germanwordoftheday. German Word of the Day: Website for learning German vocabulary. https://germanwordoftheday.de, 2018.

[9] Sally A Goldman and Michael J Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

[10] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

[11] Luis Haug, Sebastian Tschiatschek, and Adish Singla. Teaching inverse reinforcement learners via features and demonstrations. In *NeurIPS*, 2018.

[12] Parameswaran Kamalaruban, Rati Devidze, Volkan Cevher, and Adish Singla. Interactive teaching algorithms for inverse reinforcement learning. In *IJCAI*, pages 2692–2700, 2019.

[13] Mohammad M Khajah, Robert V Lindsey, and Michael C Mozer. Maximizing students' retention via spaced review: Practical guidance from computational models of memory. *Topics in cognitive science*, 2014.

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017.

[15] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

[16] S. Leitner and R. Totter. *So lernt man lernen*. Angewandte Lernpsychologie ein Weg zum Erfolg. Herder, 1972.

[17] Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[18] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative machine teaching. In *ICML*, pages 2149–2158, 2017.

[19] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. 1989.

[20] Robert M Nosofsky. Attention, similarity, and the identification–categorization relationship. *Journal of experimental psychology: General*, 115(1):39, 1986.

[21] Robert M Nosofsky, Craig A Sanders, Xiaojin Zhu, and Mark A McDaniel. Model-guided search for optimal natural-science-category training exemplars: A work in progress. *Psychonomic bulletin & review*, 26(1):48–76, 2019.

[22] Harold Pashler, Nicholas Cepeda, Robert V Lindsey, Ed Vul, and Michael C Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. In *NIPS*, pages 1321–1329, 2009.

[23] Kaustubh R Patil, Jerry Zhu, Łukasz Kopeć, and Bradley C Love. Optimal teaching for limited-capacity human learners. In *NIPS*, pages 2465–2473, 2014.

[24] Philip I Pavlik Jr and John R Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive Science*, 29(4):559–586, 2005.

[25] Paul Pimsleur. A memory schedule. *The Modern Language Journal*, 51(2):73–75, 1967.

[26] Siddharth Reddy, Igor Labutov, Siddhartha Banerjee, and Thorsten Joachims. Unbounded human learning: Optimal scheduling for spaced repetition. In *KDD*, pages 1815–1824, 2016.

[27] David C Rubin and Amy E Wenzel. One hundred years of forgetting: A quantitative description of retention. *Psychological review*, 1996.

[28] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *ACL*, volume 1, pages 1848–1858, 2016.

[29] Wayne L Shebilske, Barry P Goettl, Kip Corrington, and Eric Anthony Day. Interlesson spacing and task-related processing during complex skill acquisition. *Journal of Experimental Psychology: Applied*, 5(4):413, 1999.

[30] Amy L Simmons. Distributed practice and procedural memory consolidation in musicians' skill learning. *Journal of Research in Music Education*, 59(4):357–368, 2012.

[31] Adish Singla, Ilija Bogunovic, G Bartók, A Karbasi, and A Krause. On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*, 2013.

[32] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, pages 154–162, 2014.

[33] Edward N Spruit, Guido PH Band, and Jaap F Hamming. Increasing efficiency of surgical training: effects of spacing practice on skill acquisition and retention in laparoscopy training. *Surgical endoscopy*, 29(8):2235–2243, 2015.

[34] Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.

[35] Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Enhancing human learning via spaced repetition optimization. *PNAS*, 116(10):3988–3993, 2019.

[36] Sebastian Tschiatschek, Adish Singla, and Andreas Krause. Selecting sequences of items via submodular maximization. In *AAAI*, 2017.

[37] Ovid J Tzeng. Stimulus meaningfulness, encoding variability, and the spacing effect. *Journal of Experimental Psychology*, 99(2):162–166, 1973.

[38] Grant Van Horn, Oisin Mac Aodha, Yang Song, et al. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

[39] EGG Verdaasdonk, LPS Stassen, RPJ Van Wijk, and J Dankelman. The influence of different training schedules on the learning of psychomotor skills for endoscopic surgery. *Surgical endoscopy*, 21(2):214–219, 2007.

[40] Matthew M Walsh, Kevin A Gluck, Glenn Gunzelmann, Tiffany Jastrzembski, Michael Krusmark, Jay I Myung, Mark A Pitt, and Ran Zhou. Mechanisms underlying the spacing effect in learning: A comparison of three computational models. *Journal of Experimental Psychology: General*, 147(9):1325, 2018.

[41] Thomas D Wickens. Measuring the time course of retention. 1999.

[42] Teresa Yeo, Parameswaran Kamalaruban, Adish Singla, Arpit Merchant, Thibault Asselborn, Louis Faucon, Pierre Dillenbourg, and Volkan Cevher. Iterative classroom teaching. In *AAAI*, pages 5684–5692, 2019.

[43] Zhenliang Zhang, Edwin KP Chong, Ali Pezeshki, and William Moran. String submodular functions with curvature constraints. *IEEE Transactions on Automatic Control*, 61(3):601–616, 2016.

[44] Yao Zhou, Arun Reddy Nelakurthi, and Jingrui He. Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners. In *KDD*, pages 2817–2826, 2018.

[45] Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087, 2015.

[46] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N. Rafferty. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.

A List of Appendices

In this section we provide a brief description of the content provided in the appendices of the paper.

- Appendix B provides details of the sensitivity analysis of our model.
- Appendix C provides details of the user study.
- Appendix D provides further performance analysis for the greedy algorithm when teaching an HLR learner.
- Appendix E provides proofs of the theoretical results.

B Robustness and Sensitivity Analysis

We conducted a sensitivity study on simulated learners before choosing the HLR parameters for our user study. These detailed results are demonstrated in Fig. 3.
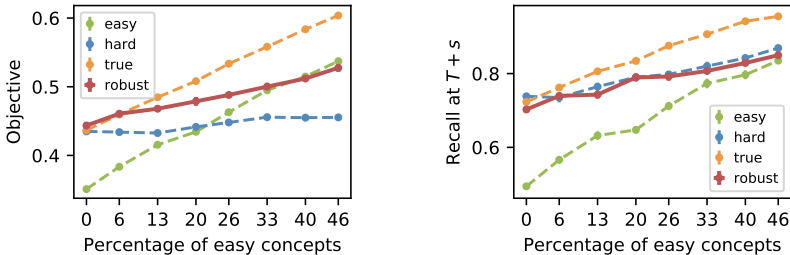

Figure 3: Sensitivity analysis of our teaching algorithm

In this experiment, we consider two groups of concepts: "easy/common" concepts with $\theta = (10, 5, 0)$, and "hard/rare" concepts with $\theta = (3, 1.5, 0)$. Other configurations are kept the same as our user study, with $T = 40$, $n = 15$ and $s = 10$.

We vary the number of "easy" concepts from $\{0, 1, \ldots, 8\}$ (i.e., up to 50% of the concepts being easy), and consider four types of teachers: (i) "easy": $\theta = (10, 5, 0)$ for all concepts; (ii) "hard": $\theta = (3, 1.5, 0)$ for all concepts; (iii) "true": using the true parameters for each concept; (iv) "robust": $\theta = (6, 2, 0)$ for all concepts. We plot the performances of these different teachers measured by the two metrics considered in simulations (i.e., the objective value and future recall). As shown in the figures, the "robust" teacher performs well on both metrics, and hence is used for our user study on the German dataset.

C User Study

C.1 Online teaching interface

We set up a simple and adaptive interface to keep the learners engaged in our user study (see Fig. 4). To establish a setup that accurately reflects our modeling assumptions, we integrate the following design ideas.

An important component of the user evaluation is to understand the learner's bias (or prior knowledge), which we cannot easily assess purely based on the learner's inputs during the learning phase. To resolve this issue, we introduce a prequiz phase (before the learning phase starts) where we test the learner's knowledge of concepts in the session by asking them to provide answers for all n concepts. After the learning phase, the learner will enter a postquiz (i.e., testing) phase. By recording the change in the learner's performance from prequiz to postquiz phase, we can estimate the gain of the teaching session.

In order to align the online teaching session with our discrete-time problem formulation, we impose a minimum and maximum time window for each flashcard presentation during the learning phase. In particular, a participant has a maximum time of 20 seconds to provide input, and has 10 seconds to review the correct answer provided by the teacher.
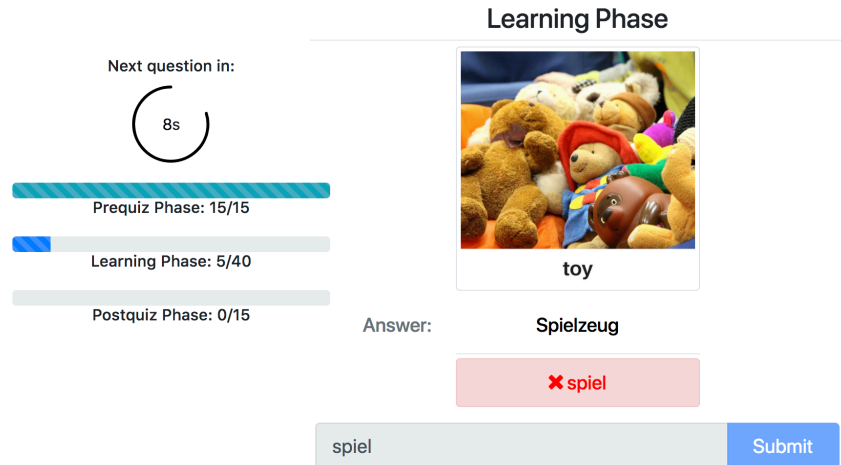
Figure 4: The teaching interface of our online apps (German and Biodiversity).

Another important aspect is the short-term memory effect. In general, it is non-trivial to carry out large scale user studies that span over weeks/months (even though it better fits the HLR model of the learner). Given the physical constraints of our real-world experiments, we consider shorter teaching sessions of 25 mins in duration, involving the teaching of $n = 15$ concepts for a total number of $T = 40$ time steps. To mitigate the short-term memory effect present in our experiments, we impose an additional constraint for the user study[3], such that the algorithms do not pick the same concept for two consecutive time steps (otherwise, a learner will simply "copy" the answer she sees on the previous screen).

C.2 User study results

Fig. 5a and Fig. 5b illustrates the distribution of learners' performances. Even though some learners failed to achieve good performance, GR managed to teach a larger fraction of learners to achieve better performance compared to the baselines—this suggests that our algorithm is an effective strategy for teaching vocabulary.

(a) German   (b) Biodiversity

Figure 5: User study results

---

[3]An alternative way to mitigate the short-term memory effect is to introduce a small break between two teaching iterations.

C.3 Datasets

In this subsection, we show a few samples from both the German dataset (for the German vocabulary teaching app) in Fig. 6, and the Biodiversity dataset (for the biodiversity teaching app) in Fig. 7.

German dataset

Figure 6: Samples from the German dataset

Biodiversity dataset

(a) Common: Owl, Cat, Horse, Elephant, Lion, Tiger, Bear

(b) Rare: Angwantibo, Olinguito, Axolotl, Ptarmigan, Patrijshond, Coelacanth, Pyrrhuloxia

Figure 7: Samples from the Biodiversity dataset

D Teaching Algorithm and Analysis

D.1 Analysis for HLR memory model

Figure 8: Performance analysis for the greedy algorithm when teaching an HLR learner with $T = 15$ and $n = 3$. Each colored marker from Fig. 8a–8c represents a different concept, with $\theta_1 = (2.50, 2.50, 1.26)$ for blue, $(\theta_2 = 1.00, 1.00, 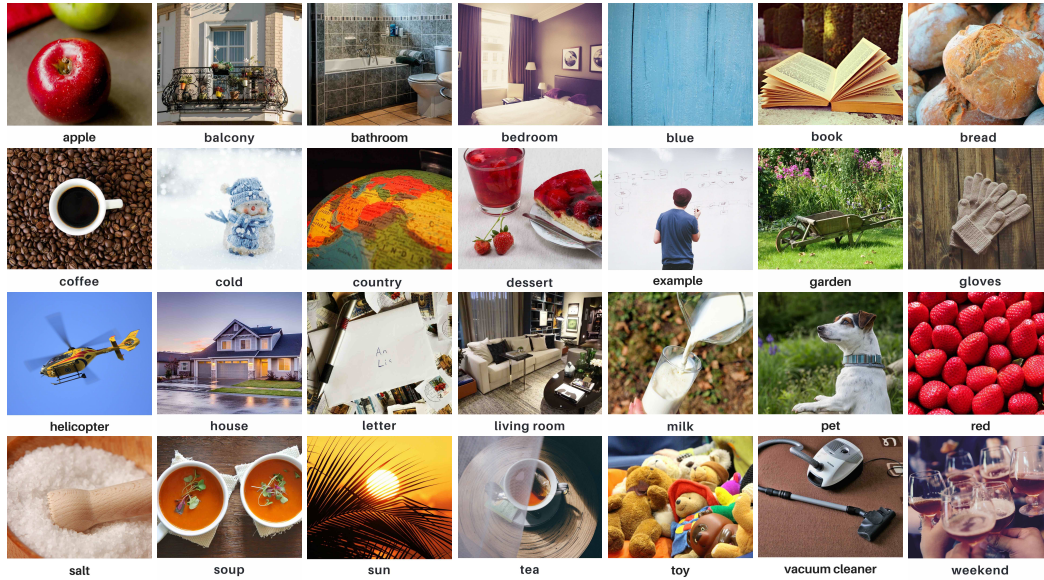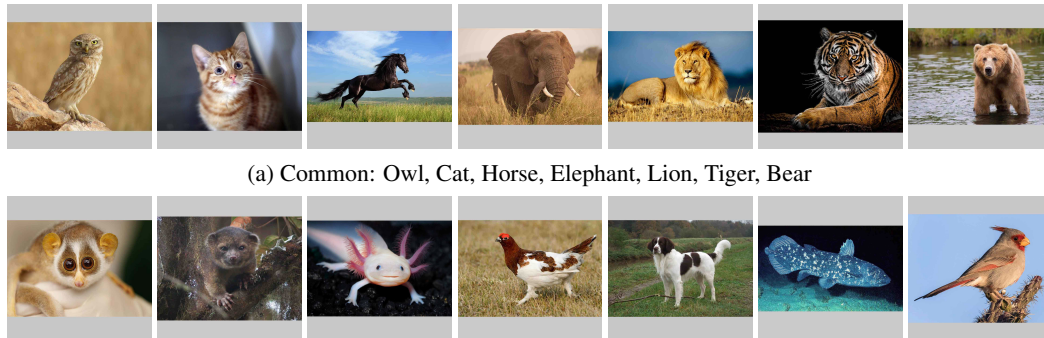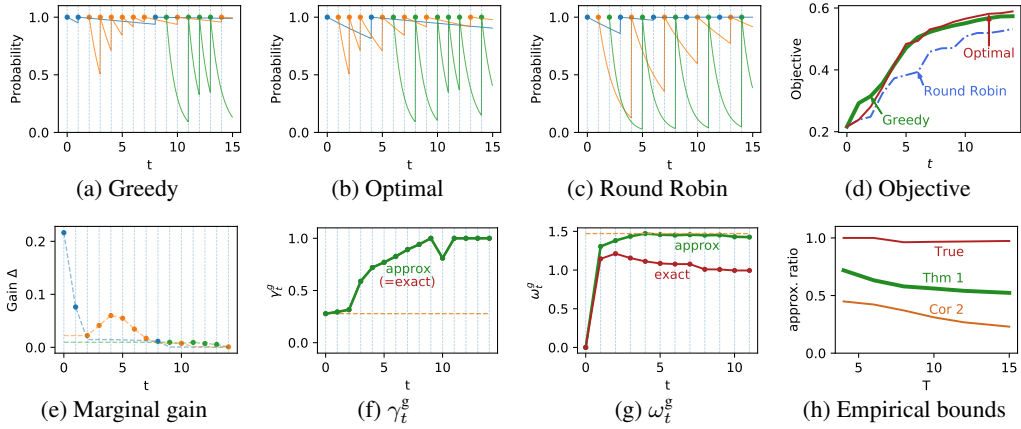-1.00)$ for orange, and $\theta_3 = (0.08, 0.08, -0.88)$ for the green concept. Intuitively, concepts with higher θ_i values are easier to teach.

In Fig. 8, we demonstrate the behavior of three teaching algorithms on a toy problem with $T = 15, n = 3$. Fig. 8a-8c show the learner's forgetting curve (i.e., recall probabilities) and the sequences selected by three algorithms: Greedy (Algorithm 1), Optimal (the optimal solution for Problem (3)), and Round Robin (a fixed round robin teaching schedule for all concepts). Observe that Greedy starts with easy concepts (i.e., concepts with higher memory retention rates), moves on to teaching new concepts when the learner has "enough" retention for the current concept, and repeats previously shown concepts towards the end of the teaching session. This behavior is similar to the optimal teaching sequence, and achieves higher utility in comparison to the fixed round robin scheduling (Fig. 8d).

In Fig. 8e, we see that the marginal gain of the orange item is increasing in the early stages (as opposed to many classical discrete optimization problems that exhibit the diminishing returns property), which makes the analysis of the greedy algorithm non-trivial. In Fig. 8f and Fig. 8g, we show the empirical bounds on $\gamma_t^{\mathrm{g}}, \omega_t^{\mathrm{g}}$, as well as the exact values of $\gamma_t^{\mathrm{g}}, \omega_t^{\mathrm{g}}$ when running the greedy algorithm. Note that our procedure for computing γ_t^{g} actually outputs the *exact* value of γ_t^{g} (a näive approach to computing γ_t^{g} is via extensive enumeration of all possible teaching sequences).

In Fig. 8h, we plug in the empirical bounds on γ_t^{g} and ω_t^{g} to Theorem 2 and Corollary 3, and plot the empirical approximation bounds on $F(\pi^{\mathrm{g}})/F(\pi^*)$ as a function of the teaching horizon T. For problem instances with a large teaching horizon T, it is infeasible to compute the true approximation bound. However, one can still efficiently compute the empirical approximation bound as a useful indicator of the performance of our algorithm.

E Proofs

E.1 Proof of Theorem 1

In this section, we provide the proof of Theorem 1. We first show that any non-negative string submodular function can be represented as a constant factor of the objective function f as defined in Eq. (4). We then prove the NP-hardness of the optimization problem (Problem (3)) by the NP-hardness result of string submodular optimization [43].

Proof. Recall from Eq. (4) that

$$f(\sigma_{1:t}, y_{1:t}) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} g_i \left(\tau + 1, \left(\sigma_{1:\min(\tau,t)}, y_{1:\min(\tau,t)} \right) \right).$$

15

In the following, we show how one can represent an arbitraty non-negative string submodular function in the form of the RHS of the above equation (i.e., Eq. (4)). Let $\mu : \Sigma \rightarrow \mathbb{R}_{\geq 0}$ be a (non-negative) string submodular function, where Σ denote the set of possible sequences defined over n items (i.e., concepts). For a fixed budget T, the string submodular optimization problem can be stated as follows [43]:

$$\sigma^* = \arg\max_{\sigma \in \Sigma, |\sigma|=T} \mu(\sigma). \tag{10}$$

For any sequence $\sigma \in \Sigma$, define $t_i(\sigma)$ to be the (time) index of item i in the sequence[4]. That is,

$$t_i(\sigma) = \begin{cases} \text{the index of item } i \text{ in } \sigma & \text{if } i \in \sigma \\ 0 & \text{o.w.} \end{cases}$$

For simplicity, we drop the dependency of σ in $t_i(\sigma)$ when it is clear from the context. Define

$$g_i\left(\tau+1, \left(\sigma_{1:\min(\tau,t)}, \cdot\right)\right) = \begin{cases} \frac{n}{c} \cdot \mu(i) & \text{if } t_i = 1 \\ \frac{nT}{c \cdot (T-t_i+1)} \cdot \mu(i \mid \sigma_{1:t_i-1}) & \text{if } 1 < t_i \leq \min(\tau,t) \\ 0 & \text{if } t_i > \min(\tau,t) \text{ or } t_i = 0 \end{cases} \tag{11}$$

Here, $\mu(i \mid \sigma_{1:t_i-1}) = \mu(\sigma_{1:t_i-1} \oplus i) - \mu(\sigma_{1:t_i-1})$ denotes the marginal gain of item i. Since $\mu_i(\cdot)$ is (string) submodular, we set $c = nT \cdot \max_i \mu(i)$ as a normalizing constant so that $g_i\left(\tau+1, \left(\sigma_{1:\min(\tau,t)}, \cdot\right)\right) \leq 1$.

Substituting g_i on the RHS of Eq.(4) by Eq.(11), we get

$$\begin{aligned} f(\sigma_{1:t}, \cdot) &= \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} g_i\left(\tau+1, \left(\sigma_{1:\min(\tau,t)}, \cdot\right)\right) \\ &\stackrel{(a)}{=} \frac{1}{nT} \sum_{i \in \sigma_{1:t}} \sum_{\tau=1}^{T} g_i\left(\tau+1, \left(\sigma_{1:\min(\tau,t)}, \cdot\right)\right) \\ &\stackrel{(b)}{=} \frac{1}{nT} \sum_{i \in \sigma_{1:t}} (T - t_i + 1) \cdot \frac{nT}{c \cdot (T - t_i + 1)} \cdot \mu(i \mid \sigma_{1:t_i-1}) \\ &= \sum_{i \in \sigma_{1:t}} \frac{1}{c} \cdot \mu(i \mid \sigma_{1:t_i-1}) \\ &= \frac{1}{c} \cdot \mu(\sigma_{1:t}) \end{aligned}$$

Here, step (a) and (b) are by the definition of g_i in Eq. (11). Therefore, for any sequence $\sigma \in \Sigma$, one can represent $\mu(\sigma)$ in terms of $cf(\sigma, \cdot)$. By the NP-hardness result of string submodular optimization [43], we conclude that the general optimization problem (Problem (3)) is NP-hard. □

E.2 Proof of Theorem 2 and Corollary 3

E.2.1 Notations and definitions

For simplicity, we first introduce the notation which will be used in the proof.

Let us use function $\phi(i, t)$ to represent a learner's recall of item i at t, where $\phi(i, t) = 1$ indicates that the learner recalls item i correctly at time t, and $\phi(i, t) = 0$ otherwise. We call the function ϕ a *realization*, and use Φ to denote a random realization. A realization ϕ is consistent with the observation history $(\sigma_{1:t}, y_{1:t})$, if $\phi(\sigma_\tau, \tau) = y_\tau$ for all $\tau \in \{1, \ldots, t\}$. We denote such a case by $\phi \sim (\sigma_{1:t}, y_{1:t})$.

We further use $(\sigma^\pi(\phi), y^\pi(\phi))$ to denote the sequence of items and observations obtained by running policy π under realization ϕ. Here, $\sigma^\pi(\phi)$ denotes the sequence of items selected by π if the learner is responding according to ϕ.

---

[4]We consider that an item cannot appear twice in a sequence $\sigma \in \Sigma$

Similarly with the conditional marginal gain of an item (Eq. (5)), we define the conditional marginal gain of a sequence of items as follows.

Definition 3 (Conditional marginal gain of a sequence). *Given observation history $(\sigma_{1:t}, y_{1:t})$, the conditional marginal gain of a sequence of items σ is defined as*

$$\Delta\left(\sigma \mid \sigma_{1:t}, y_{1:t}\right) = \mathbb{E}[f(\sigma_{1:t} \oplus \sigma, y_{1:t} \oplus y) - f(\sigma_{1:t}, y_{1:t}) \mid (\sigma_{1:t}, y_{1:t})]. \tag{12}$$

We also define the conditional marginal gain of a policy.

Definition 4 (Conditional marginal gain of a policy). *Given observation history $(\sigma_{1:t}, y_{1:t})$, the conditional marginal gain of a policy π is defined as*

$$\Delta\left(\pi \mid \sigma_{1:t}, y_{1:t}\right) = \mathbb{E}[f(\sigma_{1:t} \oplus \sigma^\pi(\Phi), y_{1:t} \oplus y^\pi(\Phi)) - f(\sigma_{1:t}, y_{1:t}) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})]. \tag{13}$$

By $\sigma_{1:t} \oplus \sigma^\pi(\Phi)$, we mean concatenating the sequence chosen by π under realization Φ (i.e., $\sigma^\pi(\Phi)$) with some existing history $\sigma_{1:t}$ (note that the first t elements of $\sigma^\pi(\Phi)$ could be completely different from $\sigma_{1:t}$).

E.2.2 Proof of Theorem 2

To prove Theorem 2, we first establish a lower bound on the one-step gain of the greedy algorithm. The following lemma provides a lower bound of the one-step conditional marginal gain of the greedy policy π^g against the conditional marginal gain of any policy (of length T).

Lemma 6. *Suppose we have selected sequence $\sigma_{1:t}$ and observed $y_{1:t}$. Then, for any policy π of length T,*

$$\max_i \Delta\left(i \mid \sigma_{1:t}, y_{1:t}\right) \geq \frac{\gamma_t^\pi}{T} \Delta\left(\pi \mid \sigma_{1:t}, y_{1:t}\right) \tag{14}$$

Proof. By Definition 4 we know that for all π it holds that

$$\Delta\left(\pi \mid \sigma_{1:t}, y_{1:t}\right) = \mathbb{E}[f(\sigma_{1:t} \oplus \sigma_{1:T}^\pi(\Phi), y_{1:t} \oplus y_{1:T}^\pi(\Phi)) - f(\sigma_{1:t}, y_{1:t}) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})]$$

$$\overset{(a)}{=} \mathbb{E}\left[\sum_{\tau=1}^{T} (f(\sigma_{1:t} \oplus \sigma_{1:\tau}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau}^\pi(\Phi)) - \right.$$
$$\left. f(\sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi))) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})\right]$$

$$= \sum_{\tau=1}^{T} \mathbb{E}\left[f(\sigma_{1:t} \oplus \sigma_{1:\tau}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau}^\pi(\Phi)) - \right.$$
$$\left. f(\sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi)) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})\right]$$

$$\overset{(b)}{=} \sum_{\tau=1}^{T} \mathbb{E}\left[\mathbb{E}\left[f(\sigma_{1:t} \oplus \sigma_{1:\tau}^\pi(\Phi'), y_{1:t} \oplus y_{1:\tau}^\pi(\Phi')) - \right.\right.$$
$$f(\sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi'), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi'))$$
$$\left.\left. \mid \Phi' \sim (\sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi))\right] \mid \Phi \sim (\sigma_{1:t}, y_{1:t})\right]$$

$$\overset{\text{Eq. (5)}}{=} \sum_{\tau=1}^{T} \mathbb{E}\left[\Delta\left(\sigma_\tau^\pi(\Phi') \mid \Phi' \sim \sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi)\right)\right.$$
$$\left. \mid \Phi \sim (\sigma_{1:t}, y_{1:t})\right] \tag{15}$$

Here, step (a) is a telescoping sum, and step (b) is by the law of total expectation.

Further, by the definition of γ_t (Definition 1) we know that for all π and ϕ it holds that

$$\max_i \Delta\left(i \mid \sigma_{1:t}, y_{1:t}\right) \geq \gamma_t^\pi \Delta\left(\sigma_\tau^\pi(\Phi') \mid \Phi' \sim \sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\phi)\right) \tag{16}$$

Combining Eq. (15) with Eq. (16) to get

$$\Delta\left(\pi \mid \sigma_{1:t}, y_{1:t}\right) \overset{\text{Eq. (15)}}{=} \sum_{\tau=1}^{T} \mathbb{E}\left[\Delta\left(\sigma_\tau^\pi(\Phi') \mid \Phi' \sim \sigma_{1:t} \oplus \sigma_{1:\tau-1}^\pi(\Phi), y_{1:t} \oplus y_{1:\tau-1}^\pi(\Phi)\right)\right.$$

$$\Big| \; \Phi \sim (\sigma_{1:t}, y_{1:t}) \Big]$$

$$\overset{\text{Eq. (16)}}{\leq} \sum_{\tau=1}^{T} \mathbb{E}\left[\frac{1}{\gamma_t^{\pi}} \max_i \Delta\left(i \mid \sigma_{1:t}, y_{1:t}\right) \;\Big|\; \Phi \sim (\sigma_{1:t}, y_{1:t})\right]$$

$$= \frac{T}{\gamma_t^{\pi}} \max_i \Delta\left(i \mid \sigma_{1:t}, y_{1:t}\right)$$

which completes the proof. □

In the following we provide the proof of Theorem 2.

Proof of Theorem 2. By the definition of ω_t (Definition 2, Eq. (7)) we know that for all π it holds that

$$\omega_t \geq 1 - \frac{\mathbb{E}[f(\sigma_{1:t} \oplus \sigma^{\pi}(\Phi), y_{1:t} \oplus y^{\pi}(\Phi)) - f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi)) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})]}{f(\sigma_{1:t}, y_{1:t})}$$

Therefore, we get

$$\Delta\left(\pi \mid \sigma_{1:t}, y_{1:t}\right) = \mathbb{E}[f(\sigma_{1:t} \oplus \sigma^{\pi}(\Phi), y_{1:t} \oplus y^{\pi}(\Phi)) - f(\sigma_{1:t}, y_{1:t}) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})]$$
$$\geq \mathbb{E}[f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi)) - \omega_t f(\sigma_{1:t}, y_{1:t}) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})] \tag{17}$$

Now suppose that we have run greedy policy π^{g} up to time step t and have observed sequence $(\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}})$. Combining Lemma 6 (Eq. (14)) with Eq. (17), we get

$$\max_i \Delta\left(i \mid \sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}}\right) = \mathbb{E}\left[f(\sigma_{1:t+1}^{\text{g}}(\Phi), y_{1:t+1}^{\text{g}}(\Phi)) - f(\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}}) \mid \Phi \sim (\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}})\right]$$

$$\geq \frac{\gamma_t}{T} \cdot \mathbb{E}\left[f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi)) - \omega_t f(\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}}) \mid \Phi \sim (\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}})\right]$$

which implies

$$\mathbb{E}\left[f(\sigma_{1:t+1}^{\text{g}}(\Phi), y_{1:t+1}^{\text{g}}(\Phi)) \mid \Phi \sim (\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}})\right]$$
$$\geq \frac{\gamma_t}{T} \cdot \mathbb{E}\left[f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi)) \mid \Phi \sim (\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}})\right] + \left(1 - \frac{\gamma_t \omega_t}{T}\right) f(\sigma_{1:t}^{\text{g}}, y_{1:t}^{\text{g}}) \tag{18}$$

Therefore, we get

$$F\left(\pi^{\text{g}}\right) = \mathbb{E}\left[f(\sigma_{1:T}^{\text{g}}(\Phi), y_{1:T}^{\text{g}}(\Phi))\right]$$

$$\overset{(a)}{=} \mathbb{E}\left[\mathbb{E}\left[f(\sigma_{1:T}^{\text{g}}(\Phi), y_{1:T}^{\text{g}}(\Phi)) \mid \Phi \sim (\sigma_{1:T-1}^{\text{g}}(\Phi'), y_{1:T-1}^{\text{g}}(\Phi'))\right]\right]$$

$$\overset{\text{Eq. (18)}}{\geq} \mathbb{E}\left[\frac{\gamma_{T-1}}{T} \cdot \mathbb{E}\left[f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi)) \mid \Phi \sim (\sigma_{1:T-1}^{\text{g}}(\Phi'), y_{1:T-1}^{\text{g}}(\Phi'))\right]\right] +$$

$$\mathbb{E}\left[\left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) f(\sigma_{1:T-1}^{\text{g}}(\Phi'), y_{1:T-1}^{\text{g}}(\Phi'))\right]$$

$$\overset{(b)}{=} \frac{\gamma_{T-1}}{T} \cdot \mathbb{E}[f(\sigma^{\pi}(\Phi), y^{\pi}(\Phi))] + \left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) \cdot \mathbb{E}\left[f(\sigma_{1:T-1}^{\text{g}}(\Phi'), y_{1:T-1}^{\text{g}}(\Phi'))\right]$$

$$= \frac{\gamma_{T-1}}{T} \cdot F\left(\pi\right) + \left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) \cdot \mathbb{E}\left[f(\sigma_{1:T-1}^{\text{g}}(\Phi), y_{1:T-1}^{\text{g}}(\Phi))\right] \tag{19}$$

where step (a) and step (b) are by the law of total expectation. Recursively applying Eq. (19) gives us

$$F\left(\pi^{\text{g}}\right) \geq \frac{\gamma_{T-1}}{T} \cdot F\left(\pi\right) + \left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) \cdot \mathbb{E}\left[f(\sigma_{1:T-1}^{\text{g}}(\Phi), y_{1:T-1}^{\text{g}}(\Phi))\right]$$

$$\geq \left(\frac{\gamma_{T-1}}{T} + \left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) \frac{\gamma_{T-2}}{T}\right) F\left(\pi\right) +$$
$$\left(1 - \frac{\gamma_{T-1}\omega_{T-1}}{T}\right) \left(1 - \frac{\gamma_{T-2}\omega_{T-2}}{T}\right) \mathbb{E}\left[f(\sigma_{1:T-2}^{\text{g}}(\Phi), y_{1:T-2}^{\text{g}}(\Phi))\right]$$

$$\geq \cdots$$

$$\geq F\left(\pi\right) \sum_{t=1}^{T-1} \frac{\gamma_{T-t}}{T} \prod_{\tau=1}^{t-1} \left(1 - \frac{\gamma_{\tau}\omega_{\tau}}{T}\right)$$

which completes the proof. □

E.2.3 Proof of Corollary 3

Proof of Corollary 3. Since $\gamma^{\mathrm{g}} = \min_t \gamma_t$ and $\omega^{\mathrm{g}} = \max_t \omega_t$, by Theorem 2 we obtain

$$
F\left(\pi^{\mathrm{g}}\right) \geq F\left(\pi\right) \sum_{t=1}^{T} \frac{\gamma_{T-t}}{T} \prod_{\tau=0}^{t-1} \left(1 - \frac{\gamma_{\tau}\omega_{\tau}}{T}\right)
$$

$$
\geq F\left(\pi\right) \frac{\gamma^{\mathrm{g}}}{T} \sum_{t=1}^{T} \left(1 - \frac{\gamma^{\mathrm{g}}\omega^{\mathrm{g}}}{T}\right)^{t}
$$

$$
= F\left(\pi\right) \frac{1}{\omega^{\mathrm{g}}} \left(1 - \left(1 - \frac{\gamma^{\mathrm{g}}\omega^{\mathrm{g}}}{T}\right)^{T}\right)
$$

which completes the proof. $\qquad\square$

E.3 Proof of Theorem 4

In this section, we provide the proof for Theorem 4. In particular, we divide the proof into two parts. In §E.3.1, we propose a polynomial time algorithm which outputs a lower bound on γ_t^g; in §E.3.2, we provide an upper bound on ω_t^{g} which can be computed in linear time.

E.3.1 Empirical lower bound on γ_t for the case $a = b$

Let us use count (σ, i) to denote the function that returns the number of times item i appears in sequence σ. We first show the following lemma.

Lemma 7. *Fix $s \leq t$. For any $\sigma' \in \{\sigma : |\sigma| = t, count\,(\sigma, i) = s\}$, we have*

$$
\Delta\left(i \mid \sigma_{t,1:s}^i, \cdot\right) \geq \Delta\left(i \mid \sigma', \cdot\right)
$$

where $\sigma_{t,1:s}^i := \underbrace{i \oplus i \oplus \cdots \oplus i}_{s \ times} \oplus \underbrace{\_ \oplus \_ \oplus \cdots \oplus \_}_{t - s \ times}$ denotes the sequence of items of length t, where the first s items are item i and the remaining $t - s$ items are empty.

Proof. By definition of the marginal gain (Eq. (5))

$$
\Delta\left(i \mid \sigma, y\right) = \mathbb{E}[f(\sigma_{1:t} \oplus i, y_{1:t} \oplus \Phi(i, t+1)) - f(\sigma_{1:t}, y_{1:t}) \mid \Phi \sim (\sigma_{1:t}, y_{1:t})]
$$

For the case $a = b$, the objective function f is independent of the observed outcomes of the learner's recall. That is,

$$
\Delta\left(i \mid \sigma_{1:t}, \cdot\right) = f(\sigma_{1:t} \oplus i, \cdot) - f(\sigma_{1:t}, \cdot)
$$

$$
= \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} \{g_i\left(\tau + 1, \sigma_{1:t} \oplus i, \cdot\right) - g_i\left(\tau + 1, \sigma_{1:t}, \cdot\right)\}
$$

$$
= \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=t+1}^{T} \{g_i\left(\tau + 1, \sigma_{1:t} \oplus i, \cdot\right) - g_i\left(\tau + 1, \sigma_{1:t}, \cdot\right)\}
$$

Denote $\Sigma_{t,s}^i = \{\sigma : |\sigma| = t, count\,(\sigma, i) = s\}$. For any $\sigma, \sigma' \in \Sigma_{t,s}^i$, we know that

$$
\sum_{\tau=t+1}^{T} g_i\left(\tau + 1, \sigma_{1:t} \oplus i, \cdot\right) = \sum_{\tau=t+1}^{T} g_i\left(\tau + 1, \sigma'_{1:t} \oplus i, \cdot\right)
$$

Therefore,

$$
\max_{\sigma_{1:t} \in \Sigma_{t,s}^i} \Delta\left(i \mid \sigma_{1:t}, \cdot\right) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=t+1}^{T} \left\{g_i\left(\tau + 1, \sigma_{1:t} \oplus i, \cdot\right) - \min_{\sigma_{1:t} \in \Sigma_{t,s}^i} g_i\left(\tau + 1, \sigma_{1:t}, \cdot\right)\right\}
$$

$$
\overset{(a)}{=} \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=t+1}^{T} \left\{g_i\left(\tau + 1, \sigma_{1:t} \oplus i, \cdot\right) - g_i\left(\tau + 1, \sigma_{t,1:s}^i, \cdot\right)\right\}
$$

Here, step (a) is due to the fact that the learner's recall of an item is monotonously decreasing (therefore showing item i earlier leads to lower recall in the future). Therefore, it completes the proof. $\qquad\square$

Algorithm 2 Computing the empirical lower bound on the greedy online stepwise submodular coefficient

---

Require: $\sigma_{1:t}; y_{1:t}$
  for $i = \{1, \ldots, n\}$ **do**
    $\texttt{CurrentGain}_i \leftarrow \Delta\left(i \mid \sigma_{1:t}, y_{1:t}\right)$
    for $\tau = \{1, \ldots, T - t\}$ **do**
      for $s \in \{1, \ldots, \tau\}$ **do**
        $\sigma' \leftarrow \underbrace{i \oplus i \oplus \cdots \oplus i}_{s \text{ times}} \oplus \underbrace{\_ \oplus \_ \oplus \cdots \oplus \_}_{\tau - s \text{ times}}$ $\triangleright$ Only consider insertions in the front
        $v_{\tau,s} \leftarrow \Delta\left(i \mid \sigma_{1:t} \oplus \sigma', \cdot\right)$ $\triangleright$ Gain of item i at $t + \tau$, with s insertions
    $\texttt{FutureGain}_i \leftarrow \max_{\tau,s} v_{\tau,s}$ $\triangleright$ Maximal gain of item i at future time steps
  $\gamma_t \leftarrow \min_i \frac{\texttt{CurrentGain}_i}{\texttt{FutureGain}_i}$ $\triangleright$ Choosing the minimal ratio among all items
  return γ_t

An approximation algorithm for γ_t is provided in Algorithm 2.

E.3.2 Empirical upper bound on ω_t for the case $a = b$

In this section, we derive an upper bound on ω_t which can be computed in polynomial time.

By definition of the online greedy stepwise backward curvature ω_t, we know

$$\omega_t := \omega(\sigma^{\mathrm{g}}_{1:t}, y^{\mathrm{g}}_{1:t}) = \max_{\sigma^\pi, y^\pi} \left\{ 1 - \frac{f(\sigma^{\mathrm{g}}_{1:t} \oplus \sigma^\pi, y^{\mathrm{g}}_{1:t} \oplus y^\pi) - f(\sigma^\pi, y^\pi)}{f(\sigma^{\mathrm{g}}_{1:t}, y^{\mathrm{g}}_{1:t})} \right\}$$

For the case $a = b$, the objective function f is independent of the observed outcomes of the learner's recall (i.e., f is a deterministic function of the input teaching sequence). Therefore,

$$\omega_t = \max_\pi \left\{ 1 - \frac{f(\sigma^{\mathrm{g}}_{1:t} \oplus \sigma^\pi, \cdot) - f(\sigma^\pi, \cdot)}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \right\}$$

$$= 1 + \max_\pi \left\{ \frac{f(\sigma^\pi, \cdot) - f(\sigma^{\mathrm{g}}_{1:t} \oplus \sigma^\pi, \cdot)}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \right\}$$

For simplicity let us use $\sigma^{\mathrm{g}+\pi} := \sigma^{\mathrm{g}}_{1:t} \oplus \sigma^\pi$ to denote the concatenated sequence, and w.l.o.g, assume that π represent the one which maximizes the RHS of the above equation (i.e., π is the optimal policy). Substituting the objective function f in the above equation with its definition (Eq. (4)), we get

$$\omega_t = 1 + \frac{1}{nT} \frac{1}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \sum_{i=1}^{n} \sum_{\tau=1}^{T} \left\{ g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) - g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right) \right\}$$

$$= 1 + \frac{1}{nT} \frac{1}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \sum_{i=1}^{n} \left\{ \sum_{\tau=1}^{T-t} g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) + \sum_{\tau=T-t+1}^{T} g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) \right.$$
$$\left. - \sum_{\tau=t+1}^{T} g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right) \right\}$$

$$= 1 + \frac{1}{nT} \frac{1}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \sum_{i=1}^{n} \left\{ \sum_{\tau=T-t+1}^{T} g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right) \right.$$
$$\left. + \underbrace{\sum_{\tau=1}^{T-t} g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) - \sum_{\tau=t+1}^{T} g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right)}_{\leq 0} \right\}$$

$$\leq 1 + \frac{1}{nT} \frac{1}{f(\sigma^{\mathrm{g}}_{1:t}, \cdot)} \sum_{i=1}^{n} \left\{ \sum_{\tau=T-t+1}^{T} g_i\left(\tau + 1, \sigma^\pi_{1:\tau}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau + 1, \sigma^{\mathrm{g}+\pi}_{1:\tau}, \cdot\right) \right\} \qquad (20)$$

Let $\sigma_{1:t}^i := \underbrace{i \oplus i \oplus \cdots \oplus i}_{t \text{ times}}$ denote the sequence of items of length t that consists of all i's. Then, clearly

$$\sum_{\tau=T-t+1}^{T} g_i\left(\tau+1, \sigma_{1:\tau}^{\pi}, \cdot\right) \leq \sum_{\tau=T-t+1}^{T} g_i\left(\tau+1, \sigma_{1:\tau}^{i}, \cdot\right) \tag{21}$$

Combining Eq. (20) with Eq. (21) we get

$$\begin{aligned}
\omega_t &\leq 1 + \frac{1}{nT}\frac{1}{f(\sigma_{1:t}^{\text{g}}, \cdot)}\sum_{i=1}^{n}\left\{\sum_{\tau=T-t+1}^{T} g_i\left(\tau+1, \sigma_{1:\tau}^{\pi}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau+1, \sigma_{1:\tau}^{\text{g}+\pi}, \cdot\right)\right\} \\
&\leq 1 + \frac{1}{nT}\frac{1}{f(\sigma_{1:t}^{\text{g}}, \cdot)}\sum_{i=1}^{n}\left\{\sum_{\tau=T-t+1}^{T} g_i\left(\tau+1, \sigma_{1:\tau}^{i}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau+1, \sigma_{1:\tau}^{\text{g}+\pi}, \cdot\right)\right\} \\
&= 1 + \frac{1}{nT}\frac{1}{f(\sigma_{1:t}^{\text{g}}, \cdot)}\sum_{i=1}^{n}\left\{\sum_{\tau=T-t+1}^{T} g_i\left(\tau+1, \sigma_{1:\tau}^{i}, \cdot\right) - \sum_{\tau=1}^{t} g_i\left(\tau+1, \sigma_{1:\tau}^{\text{g}}, \cdot\right)\right\} \tag{22}
\end{aligned}$$

Proof of Theorem 4. Clearly, both the empirical bounds on γ_t^{g} (Algorithm 2) and ω_t^{g} (RHS of Eq. (22)) can be computed in polynomial time. Plugging the values into Theorem 2 and Corollary 2 we get a polynomial time approximation of the empirical bound. $\square$

E.4 Proof of Theorem 5

In this section, we provide the proof of Theorem 5.

Suppose there are n items, and T is a multiple of n. Fix a, and assume that $a_i = b_i = a$ and $c_i = 0$ for all $i \in \{1, \ldots, n\}$. We first show a sufficient condition on a under which the greedy policy reduces to the round robin policy.

Recall from Eq. (9) that the recall probability of an item is

$$g_i(\tau, \cdot) = 2^{\frac{\tau-\ell}{h_i}} \tag{23}$$

where $h_i = 2^{an_i}$ denotes the half life of item i, and n_i denotes the number of times item i is presented so far.

Now assume that the greedy algorithm picks item i at $t = 1$. Then, in order for the greedy algorithm not to pick the same item at $t = 2$, we need to make sure that at $t = 2$, the gain of item i is smaller than the gain of the best item. To achieve that, there must exist some other item j, such that

$$\Delta\left(j \mid \sigma_1 = i\right) > \Delta\left(i \mid \sigma_1 = i\right)$$

That is,

$$\sum_{t=2}^{T}\left(g_j\left(t, \sigma_1 = i, \sigma_2 = j\right) - g_j\left(t, \sigma_1 = i\right)\right) > \sum_{t=2}^{T}\left(g_i\left(t, \sigma_1 = i, \sigma_2 = i\right) - g_i\left(t, \sigma_1 = i\right)\right)$$

A sufficient condition for the above inequality to hold is

$$\begin{aligned}
g_j\left(T, \sigma_1 = i, \sigma_2 = j\right) - g_j\left(T, \sigma_1 = i\right) &= g_j\left(T, \sigma_1 = i, \sigma_2 = j\right) \\
&> g_i\left(T, \sigma_1 = i, \sigma_2 = i\right) - g_i\left(T, \sigma_1 = i\right)
\end{aligned}$$

Plugging in the definition of g_i, g_j, we get

$$2^{-\frac{T-1}{2^a}} > 2^{-\frac{T-1}{2^{2a}}} - 2^{-\frac{T}{2^a}} \tag{24}$$

It is easy to verify numerically that a sufficient condition for Eq. (24) to hold is

$$a \geq \log T \tag{25}$$

Next, we provide a lower bound on the cost of the round robin algorithm. Let $\sigma_{1:T}$ be the round robin teaching sequence. W.l.o.g., assume that the order of items shown in each round is $1, 2, \ldots, n$. Therefore,

$$
\begin{aligned}
f(\sigma_{1:T}) &= \frac{1}{nT} \sum_{i=1}^{n} \sum_{\tau=1}^{T} g_i \left(\tau + 1, \sigma_{1:\tau} \right) \\
&= \frac{1}{nT} \sum_{i=1}^{n} \sum_{r=1}^{T/n} \sum_{\tau=1}^{n} g_i \left((r-1)n + \tau + 1, \sigma_{1:(r-1)n+\tau} \right) \\
&\geq \frac{1}{nT} \sum_{i=1}^{n} \sum_{r=1}^{T/n} n g_i \left(rn + 1, \sigma_{1:(r-1)n+\tau} \right) \\
&= \frac{1}{T} \sum_{i=1}^{n} \sum_{r=1}^{T/n} g_i \left(rn + i, \sigma_{1:(r-1)n+\tau} \right)
\end{aligned}
$$

For simplicity, define $p_{i,r} = g_i \left(rn + i, \sigma_{1:(r-1)n+\tau} \right)$. We thus have

$$
f(\sigma_{1:T}) = \frac{1}{T} \sum_{i=1}^{n} \sum_{r=1}^{T/n} p_{i,r} \tag{26}
$$

Observe that for $r \in \{1, \ldots, T/n\}$, it holds that

$$
\frac{1 - p_{i,r+1}}{1 - p_{i,r}} \geq \frac{1 - p_{i,r+2}}{1 - p_{i,r+1}}, \text{ and } 1 - p_{i,r} \geq 1 - p_{i,r+1} \tag{27}
$$

From the above inequalities we get

$$
\begin{aligned}
1 - p_{i,r+1} &= (1 - p_{i,r}) \frac{1 - p_{i,r+1}}{1 - p_{i,r}} \\
&\leq (1 - p_{i,r}) \frac{1 - p_{i,r}}{1 - p_{i,r-1}} \\
&\leq (1 - p_{i,r-1}) \frac{1 - p_{i,r-1}}{1 - p_{i,r-2}} \cdot \frac{1 - p_{i,r}}{1 - p_{i,r-1}} \\
&\leq (1 - p_{i,r-1}) \left(\frac{1 - p_{i,r-1}}{1 - p_{i,r-2}} \right)^2 \\
&\leq (1 - p_{i,1}) \left(\frac{1 - p_{i,2}}{1 - p_{i,1}} \right)^r
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\sum_{r=1}^{T/n} (1 - p_{i,r}) &\leq (1 - p_{i,1}) + (1 - p_{i,1}) \frac{1 - p_{i,2}}{1 - p_{i,1}} + \cdots + (1 - p_{i,1}) \left(\frac{1 - p_{i,2}}{1 - p_{i,1}} \right)^{T/n-1} \\
&= \sum_{r=1}^{T/n} (1 - p_{i,1}) \left(\frac{1 - p_{i,2}}{1 - p_{i,1}} \right)^{r-1} \\
&= \frac{(1 - p_{i,1}) \left(1 - \left(\frac{1 - p_{i,2}}{1 - p_{i,1}} \right)^{T/n} \right)}{1 - \left(\frac{1 - p_{i,2}}{1 - p_{i,1}} \right)} \\
&\leq \frac{(1 - p_{i,1})^2}{p_{i,2} - p_{i,1}} \tag{28}
\end{aligned}
$$

Combining Eq. (26) with Eq. (28) we get

$$
f(\sigma_{1:T}) = \frac{1}{T} \sum_{i=1}^{n} \sum_{r=1}^{T/n} p_{i,r}
$$

$$= 1 - \frac{1}{T} \sum_{i=1}^{n} \sum_{r=1}^{T/n} (1 - p_{i,r})$$

$$\geq 1 - \frac{1}{T} \sum_{i=1}^{n} \frac{(1 - p_{i,1})^2}{p_{i,2} - p_{i,1}}$$

$$\overset{(a)}{=} 1 - \frac{n}{T} \frac{(1 - p_{i,1})^2}{p_{i,2} - p_{i,1}}$$

where step (a) is due to the fact that $p_{i,1} = 2^{-n/2^a}$, and $p_{i,2} = 2^{-n/2^{2a}}$ for all i.

Now suppose that we would like to lower bound the utility $f(\sigma_{1:T})$ by $1 - \epsilon$. Therefore,

$$\frac{n}{T} \frac{(1 - p_{i,1})^2}{p_{i,2} - p_{i,1}} \leq \epsilon \tag{29}$$

While it is challenging to solve Eq. (29) in an analytical form, we consider a stronger condition to simplify the calculation. Consider a configuration of a which also satisfies the following inequality

$$1 - p_{i,2} \leq \frac{1 - p_{i,1}}{2} \tag{30}$$

Therefore, a sufficient condition for Inequality (29) to hold is

$$\frac{(1 - p_{i,1})^2}{p_{i,2} - p_{i,1}} = \frac{(1 - p_{i,1})^2}{(1 - p_{i,1}) - (1 - p_{i,2})} \overset{\text{Eq. (30)}}{\leq} \frac{(1 - p_{i,1})^2}{(1 - p_{i,1}) - \frac{1 - p_{i,1}}{2}} = 2(1 - p_{i,1}) \leq \frac{\epsilon T}{n}$$

Plugging in $p_{i,1} = 2^{-n/2^a}$ into the above inequality, we get

$$2^{-n/2^a} \geq 1 - \frac{\epsilon T}{2n} \tag{31}$$

Now, let us consider the following two cases:

C1 $1 - \frac{\epsilon T}{2n} > 0$ (that is, $\epsilon < 2n/T$). In this case, we get

$$a \geq \log \left(\frac{n}{\log \left(\frac{1}{1 - \epsilon T/(2n)} \right)} \right)$$

$$= \log n - \log \log \left(\frac{1}{1 - \epsilon T/(2n)} \right)$$

$$\overset{(a)}{\geq} \log n - \log \left(\left(\frac{1}{1 - \epsilon T/(2n)} \right) - 1 \right)$$

$$= \log \left(\left(\frac{2n^2}{\epsilon T} \right) - n \right)$$

where step (a) is by the inequality $\log(x) \leq x - 1$ for $x > 0$. A feasible configuration of a satisfying the above inequality is

$$a \geq \log \left(\frac{2n^2}{\epsilon T} \right) \tag{32}$$

It is easy to verify that Condition Eq. (32) also satisfies our additional constraint Eq. (30).

C2 A second case is $\epsilon \geq 2n/T$. In this case, Eq. (31) holds for all a, and we only need to find a feasible configuration of a that satisfies Eq. (30). A suitable choice of such a constraint is

$$a \geq \log(3n) \tag{33}$$

Combining Eq. (25) Eq. (32) and Eq. (33) we obtain

$$a \geq \max \left\{ \log T, \log(3n), \log \left(\frac{2n^2}{\epsilon T} \right) \right\}$$

which finishes the proof.