CURRICULUM DISCOVERY THROUGH AN ENCOMPASSING CURRICULUM LEARNING FRAMEWORK

Anonymous authors
Paper under double-blind review

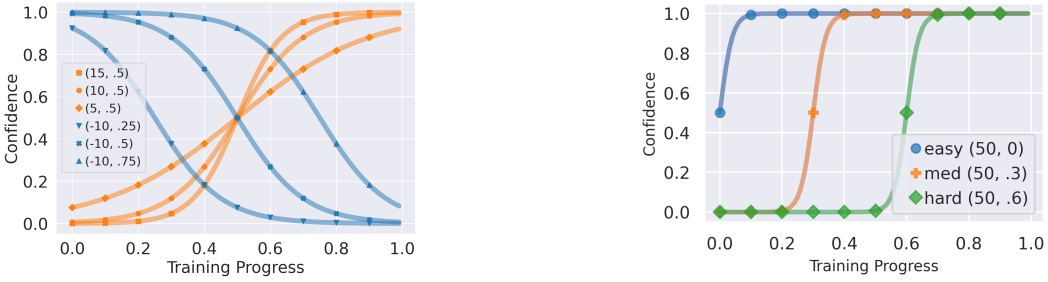
ABSTRACT

We describe a curriculum learning framework capable of discovering optimal curricula in addition to performing standard curriculum learning. We show that this framework encompasses existing curriculum learning approaches such as difficulty-based data sub-sampling, data pruning, and loss re-weighting. We employ the proposed framework to address the following key questions in curriculum learning research: (a) What is the best curriculum to train a given model on a given dataset? (b) What are the characteristics of optimal curricula for different datasets and different difficulty scoring functions? We show that our framework outperforms competing state-of-the-art curriculum learning approaches in natural language inference and other text classification tasks. In addition, exhaustive experiments illustrate the generalizability of the discovered curricula across the three datasets and two difficulty scoring functions.

1 INTRODUCTION

Curriculum Learning (CL) is a technique in Machine Learning that mimics human education systems. To learn a complex subject, students must first learn the foundational and basic materials before learning more complex ones. Without a curriculum, learning may be intractable, inefficient and learners may never reach a full understanding of a topic due to lack of the required background knowledge. Machine learning optimization through stochastic gradient descent teaches models through experience by observing example data instances. Some data instances are harder than others and require background knowledge. That knowledge could be acquired by observing easier examples before harder ones. CL techniques seek to order examples according to their difficulty for training to generate better models. CL has been shown to improve performance in solving harder tasks, and in cases of limited data availability or noisy data (Wu et al., 2021) CL research has made significant progress in the last decade through the work of Bengio et al. (2009), although the principle of ordering training samples from easier to harder was introduced in the 1990s (Elman, 1993; Sanger, 1994; Rohde & Plaut, 1999).

A curriculum can be defined by providing a difficulty measurement for each sample, or by dividing the data into groups of increasing difficulty. Given a measure of difficulty, there are different types of CL approaches for ordering the data. For example, sub-sampling the easiest samples at every iteration and only using those to train (Zhou et al., 2020; Bengio et al., 2009; Xu et al., 2020; Guo et al., 2018; Platanios et al., 2019), using the complete data and weighting samples differently according to their difficulty (Castells et al., 2020; Kumar et al., 2010; Jiang et al., 2015; 2018; Zhou et al., 2020; Yang et al., 2019), and pruning the hard or noisy samples from the dataset prior to training (Northcutt et al., 2021; Guo et al., 2018). Sub-sampling methods can be cumulative, or exclusive, or a combination of both; at every stage of increasing the level of difficulty, we can choose to keep using the examples of the previous stage, or to use only the new subset. A curriculum can also introduce samples from easy to hard (Bengio et al., 2009; Kumar et al., 2010), or hard to easy, which can be an effective learning strategy in some specific tasks (Kocmi & Bojar, 2017; Zhang et al., 2018; 2019). CL methods may impose a curriculum by adjusting model's capacity according to the difficulty of their inputs (Karras et al., 2018; Sinha et al., 2020; Morerio et al., 2017) or schedule the order of tasks in the context of multi-task learning (Caubrière et al., 2019; Sarafianos et al., 2017; Florensa et al., 2017). Other CL approaches such as (Graves et al., 2017; Zhou et al., 2020; Saxena et al., 2019) use O(n), where n is the number of training samples, extra parameters for learning curricula. In this work, we focus on data-oriented CL, which aims to rank and order



(a) Effect of varying the parameters.

(b) Easy to Hard Curriculum.

Figure 1: The effect of the rate of growth and shift parameters (r, s). (a) illustrates different weighting strategies that can be obtained by varying the rate and shift parameters. (b) a specific parameter configuration for a curriculum that first introduces easier training samples to a model, and then gradually introduces medium and hard samples at 30% and 60% of the training epochs.

instances for effective training. Our approach can encode prior knowledge about sample difficulty to CL, and can be considered as a student-instructor collaborative CL.

Current CL approaches calculate difficulty scores for training samples based on the loss of a trained model (Xu et al., 2020; Wu et al., 2021), trainable parameters to weight samples (Kumar et al., 2010; Jiang et al., 2015; Castells et al., 2020), loss value during training (Wu et al., 2021), moving average of loss during training (Zhou et al., 2020), transformations of the loss during training (Jiang et al., 2018; Castells et al., 2020), and consistency in the correct classification of samples (Amiri et al., 2017; Xu et al., 2020). The resulting difficulty scores will then be used to order samples for training. Alternatively, examples can be ordered based on prior knowledge about their difficulty, which is often estimated through intrinsic sample features and with respect to the task, e.g. object shape or orientation in image classification (Bengio et al., 2009).

We will demonstrate a CL framework that encompasses existing CL approaches through an effective and flexible data partitioning and weighting scheme. Our framework provides a new paradigm for selecting an ordering strategy. Instead of a pre-determined strategy, the framework allows searching over the curriculum space to identify the best curriculum for a particular dataset and model. It partitions training data into three groups of training samples, i.e. $\{$*easy*, *medium*, *hard*$\}$ samples, according to a difficulty scoring function. Three parameterized weighting functions will then be defined, which specify the weight of each group at every point in training. Each weight function is controlled by two parameters, which can be set empirically or adjusted using Bayesian optimization. In addition, the framework discovers optimal curricula by optimizing the parameters of the weight functions (only 2 parameters per function) using a hyper-parameter optimization algorithm. Furthermore, the curricula identified through this search provide useful insight about the dataset, such as the relative importance of different samples or knowledge dependency between samples, e.g. which samples should be learned first.

We begin by explaining our framework and showing how it is capable of approximating existing CL approaches. Then, in the context of the proposed framework, we investigate curriculum discovery, characteristics of discovered curricula, and generalizability of curricula with respect to their input datasets.

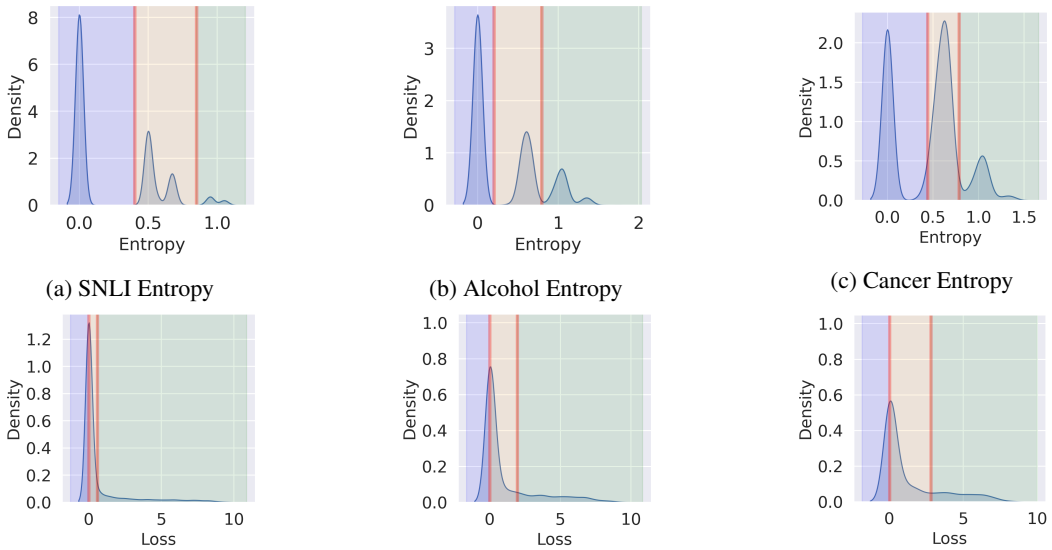
2 Curriculum Learning Framework

2.1 Weighting Functions

We define the curriculum using generalized logistic functions (Richards, 1959) of the form:

$$w(t; r, s) = \frac{1}{1 + \exp(-r * (t - s))}, \tag{1}$$

where $r \in \mathbb{R}$ is the rate of change that can be a positive or negative value, specifying how fast the weight can increase or decrease, $t \in [0, 1]$ is the training progress (iteration number divided



(a) SNLI Entropy (b) Alcohol Entropy (c) Cancer Entropy

(d) SNLI Loss (e) Alcohol Loss (f) Cancer Loss

Figure 2: Distributions of entropy and loss of the three datasets used in the experiments. Figures (a) - (c) show entropy, and (d) - (f) show loss. Samples of the *easy* class are to the left of the first vertical line, those of the *medium* class are between the two vertical lines, and samples of the *hard* class are to the right of the second line.

by max iterations), and $s \in \mathbb{R}$ shifts the pivot weight ($w(.) = .5$) of the logistics function to the left or right such that at $t = s$ the weight is 0.5. Figure 1a illustrates the effect of varying these parameters. Greater absolute values for the rate parameter enforce greater slope and faster rate of changes in weights, while greater values of the shift parameter enforce longer delays in reaching the pivot weight of 0.5. These parameters provide flexibility in controlling sample weights during training, which is critical for deriving effective curricula. The data is split into three partitions of increasing difficulty, and for each partition, we define a weight function by a pair of parameters (r, s).

The proposed sample weighting framework provides flexibility in sample-ordering according to difficulty. It can be learned to approximate existing predetermind curricula, for example, it is possible to begin with only easy instances or only difficult instances (as an *anti-curriculum*) or a combination of both. Figure 1b shows a specific configuration for the logistic functions based on standard CL (Bengio et al., 2009; Kumar et al., 2010), where training starts with easier samples and gradually proceeds with harder ones. In addition, our approach enables discovering new data-driven curricula from data.

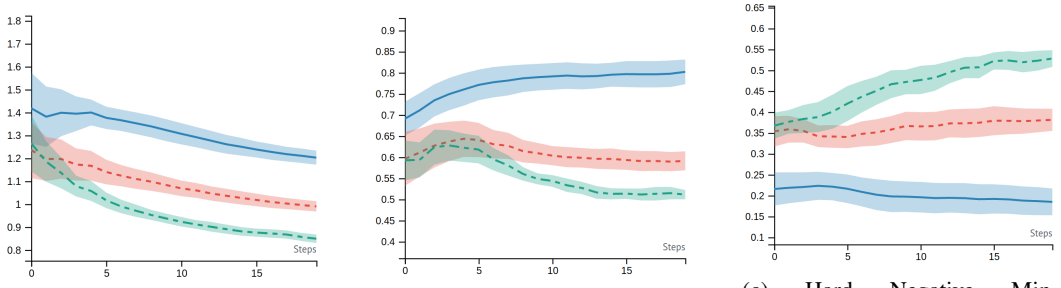
Prior to training, the difficulty scores of samples are computed and each sample is assigned to a difficulty class $c \in \{$*easy, medium, hard*$\}$, see §2.2. In addition, the hyper-parameters of our three weight functions are optimized prior to training and kept fixed throughout training $\{(r_e, s_e), (r_m, s_m), (r_h, s_h)\}$. During training, the weighted loss is computed as follows.

$$\hat{l}_i = w(t; r_c, s_c) * l_i \tag{2}$$

Where l_i is the unweighted and instantaneous loss of instance i, $\hat{l}_i$ is the weighted loss, t is the current training iteration divided by the maximum number of iterations, c is the difficulty class of instance i, and (r_c, s_c) are the corresponding weighting parameters for the difficulty class c.

2.2 Scoring Functions

Ground-truth labels for many datasets are often obtained through human annotation and crowdsourcing. This is achieved by collecting multiple annotations per data sample and aggregating the result, typically by majority voting. For such datasets, we define a difficulty score for each sample



(a) The SuperLoss (Castells et al., 2020)

(b) Self-paced Learning (Kumar et al., 2010)

(c) Hard Negative Mining (Felzenszwalb et al., 2009)

Figure 3: Confidence assignment to samples in our datasets by three curriculum learning approaches that re-weight their loss functions by computing confidence scores for samples. The x-axis is the epoch number, and y-axis is the average weight assigned to instances of different difficulty. Blue (solid) is *easy*, orange (dashed) is *medium*, and green (dash-dot) is *hard*. The shaded area is the 95% confidence interval (CI) over three datasets with five random seeds each. The curves are monotonic for most parts, and can be approximated by the monotonic curves generated by our framework.

using Shannon entropy (Shannon, 2001), where higher disagreement among annotators corresponds to higher sample difficulty. Entropy is a natural measure of difficulty (for the human population) and may serve as a reliable criterion for partitioning data by difficulty. Nevertheless, training loss contains valuable information about the difficulty to the learner, which may be different among architectures and tasks, and indicative of the model's specific needs. We will use these two difficulty scores in our analysis.

Entropy of each sample x_i is calculated as $H(x_i) = -\sum_l p_c \log p_c$ where c is a class category and p_c is the fraction of annotators who chose label c for the sample. The use of entropy is supported by experiments in Nie et al. (2020), who also study the correlation between human agreement and model performance, and find that there is a consistent positive correlation between model accuracy and level of human agreement, showing that model performs better on samples with a high level of agreement. Loss at a particular step (e.g., final loss) is dictated by the stochastic gradient descent and mini-batching dynamics and therefore is not a good indicator of difficulty (Zhou et al., 2020; Wu et al., 2021). Using a baseline model, trained with no curriculum and with default hyper-parameters, we collect the loss values of all training instances at intervals of 0.5 epochs and use the average loss to estimate sample difficulty. In our experiments, we obtain twenty observations of the loss and compute the average for each instance. Such an estimation is supported by Zhou et al. (2020) who showed that the moving average of a sample's instantaneous loss is a good metric for difficulty. Partitioning the data into three groups of increasing difficulty can be done using difficulty score percentiles, or 1-dimensional k-means clustering of the scores. Examples of data partitions using entropy and loss are shown in Figure 2.

2.3 Encompassing Framework

Curriculum learning approaches can be divided into three categories depending on how they process their input data: approaches that identify and prune noisy data that may hurt performance (Northcutt et al., 2021; Guo et al., 2018; Rooyen et al., 2015; Patrini et al., 2016; Chen et al., 2019), approaches that use different sub-samples of data during different stages of training (Bengio et al., 2009; Zhou et al., 2020; Xu et al., 2020; Platanios et al., 2019; Zhou & Bilmes, 2018), and approaches that re-weight loss according to sample difficulty, choosing to emphasize either easy or hard samples (Castells et al., 2020; Jiang et al., 2015; 2018; Yang et al., 2019; Saxena et al., 2019). The framework presented in this paper is capable of representing all of the three approaches.

First, data pruning can be done by assigning negative values to the rate change and shift parameters in our framework, r and s in Eq. 2. A negative r causes the weight to approach zero, and a negative s shifts the curve to the left, so the curve reaches zero before training begins. The framework also allows flexibility in pruning: by setting a small positive s, the noisy data can be seen by the model

for a short amount of time before reaching zero weight, or by setting a positive r and a large positive s the noisy data will only be seen at the end of training after training has stabilized.

Second, data sub-sampling can be represented by the weight going to zero or increasing from zero at different stages of training. For instance, Figure 1b illustrates a curriculum where the easy samples are sub-sampled in the beginning, and harder samples are introduced at later stages.

Third, we display in Figure 3 the confidence scores assigned to our data by three loss re-weighting approaches. The results are generated by our implementations of the three approaches, evaluated on the three datasets introduced in §3.1, where each model runs with five random seeds. The partitioning of *easy*, *medium*, and *hard* is according to the entropy-based difficulty classes, as described in §2.2. We record the average weight (confidence) assigned to each class. The result is averaged over all runs, and the shaded area is the 95% confidence interval. The approaches that estimate sample confidence based on loss (Zhou et al., 2020; Castells et al., 2020; Kumar et al., 2010; Jiang et al., 2015; Felzenszwalb et al., 2009) tend to generate monotonic curves over the course of training because training loss tends to be non-increasing at every step. Therefore, the confidence scores assigned by these re-weighting approaches follow a monotonic curve that can be approximated by our weighting functions (§2.1, Figure 1). We note that although the weight scale of SuperLoss (Castells et al., 2020) in Figure 3a is larger than one, this model can still be represented by our CL framework because the increased scale corresponds to a scaling of the learning rate, as shown below:

$$\theta_t = \theta_{t-1} - \eta \nabla \frac{1}{n} \sum_i \sigma_i l_i = \theta_{t-1} - (\eta \cdot \sigma_{max}) \nabla \frac{1}{n} \sum_i \frac{\sigma_i}{\sigma_{max}} l_i, \tag{3}$$

where l_i and σ_i are the loss and confidence of sample i, respectively. Therefore, our framework can also represent CL approaches with a confidence scale larger then one.

2.4 Curriculum Discovery

We employ a hyperparameter optimization framework (Akiba et al., 2019) to find the optimal value of the curriculum parameters (r, s) for each difficulty-class. Using this method, we can learn a data-driven curriculum beyond what we could manually design through empirical settings or a choice among the limited ordering strategies. To optimize the three pairs of parameters, we use the Tree-structured Parzen Estimator (TPE) sampling algorithm (Bergstra et al., 2011). Unlike grid or random search (Bergstra & Bengio, 2012), TPE traverses the parameter space by estimating the parameters that are most probable to perform better on a trial, based on the previous trials. TPE defines two Gaussian Mixture Models, $l(x)$ and $g(x)$ which are formed using the best and remaining observed parameters, respectively. TPE selects the parameter x with a high probability under $l(x)$ and low probability under $g(x)$; $\max_x l(x)/g(x)$. This choice of sampling algorithm greatly speeds up the search of our six curriculum parameters. The search space that we define in our experiments consists of 11 possible values for r and 9 for s, for a total of 970k combinations. Using TPE, we find the optimal set of parameters within 100 trials, as compared to over 1000 trials by random search.

We note that the discovered curricula are optimal within this framework, constrained by the method of data partitioning and the class of weight functions. We argue that the proposed framework is able to approximate curricula defined by existing CL approaches, and outperform existing CL approaches across several datasets.

3 Experiments

3.1 Datasets

We evaluate our approach on three datasets that contain multiple annotations for each sample. First, the Stanford Natural Language Inference (SNLI) benchmark (Bowman et al., 2015), which contains 550k training samples, 10k development samples, and 10k test samples. Within the training samples, there are 36.7k samples annotated by 5 workers and 2.6k annotated by 4 workers, which we use for our experiments and refer to as SNLI "full." Furthermore, in order to control for variance due to imbalanced difficulty classes, we create a balanced subset of the data. As shown in Figure 2a (notice the y-axis scale), SNLI is highly imbalanced in entropy classes. The data is downsampled by selecting an equal number of samples from each entropy-class. The balanced subset contains a total

of 2.3k samples, i.e. 774 samples in each entropy class. On the other hand, loss classes are fairly distributed, see Figure 2d. The downsampled subset contains both entropy and loss classes which will be used in experiments.

The Alcohol dataset (Weitzman et al., 2020) has been developed to obtain population-level statistics of alcohol use reports through social media. The dataset consists of more than 9k tweets. Given an alcohol relevant tweet, annotators are asked to determine if it reports first-person alcohol use, and if yes, the intensity of the drinking (light vs. heavy), the context of drinking (social vs. individual), and the time of drinking (past, present, or future). All samples in the dataset are labeled by at least three workers, including over 1.3k samples labeled by five or more workers. We define a multi-class classification task for this dataset based on alcohol relevant, intensity and context of drinking. The categories and their data distributions are reported in Appendix C. We randomly split the data into 5.4k training samples, 1.8k development samples, and 1.8k test samples. The balanced version of the training set contains a total of 2.5k training samples, i.e. 863 samples in each entropy class.

The Cancer[1] dataset has been developed to obtain population-level statistics of cancer patients; it contains 3.8k Reddit posts. Annotators are asked to determine if the post describes the experience of a cancer patient, the type of cancer, and the relation of the author of the post to the patient. We define a multi-class classification task based on post relevance and caner type. The categories and their data distribution are reported in Appendix C. All samples are labeled by at least three workers, including about 1k labeled by at least five. We randomly split the data into around 2.2k training samples, 765 development samples, and 765 test samples. The balanced version of the training set contains a total number of 1.7k sample, i.e. 578 samples in each entropy class.

We note that the datasets are significantly different in terms of document length, the average document length in SNLI, Alcohol and Cancer datasets are 10, 15 and 174 words. This variation can induce significant variance in the created models.

3.2 BASELINES

We compare the performance of our CL approach with the following state-of-the-art approaches. SuperLoss (SL) (Castells et al., 2020) is a CL approach that defines a task-agnostic confidence-aware loss function. It infers the confidences of instances from the instance loss with minimal cost, providing a closed-form solution function for estimating confidence. It up-weights samples with smaller loss values (easier instances), while down-weights those with larger loss values (harder ones) (Figure 3a). MentorNet (Jiang et al., 2018) uses an auxiliary network to generate a weight for training samples at every training iteration. It incorporates additional signals such as epoch number and instance loss history to learn data-driven curricula and is particularly strong against noisy data. Difficulty Prediction (DP) (Yang et al., 2019) defines a difficulty score based on multi-annotator data, and weight samples according to the following formula $w = 1 - \alpha(d_i - \tau_{DP})/(1 - \tau_{DP})$, where d_i is the sample difficulty and τ is a pre-defined threshold.

As discussed before, our approach employs two scoring functions (§ 2.2) and two curriculum configurations for each dataset. A curriculum configuration refers to a particular setting of the six parameters controlling the three weight curves (§ 2.1). The two scoring functions are labeled as *Loss* and *Ent* (entropy). In addition, the first curriculum configuration is a gradually increasing approach in Figure 1b named *inc.*, this configuration is applied identically to all models. The second configuration is the specialized configuration (*sp.*) that is obtained through hyper-parameters search for each triple of dataset, balance, and scoring function (illustrated in Figure 5).

3.3 SETTINGS

We tune the parameters λ of SL and α and τ_{DP} of DP using development data. The optimal values found are $\lambda = 1.2$, $\alpha = 0.9$ and τ_{DP} is set dynamically upon loading the dataset to the 50-percentile difficulty value. Following Castells et al. (2020), we set τ_{SL} (instances with $l_i > \tau_{SL}$ are considered hard) to the moving average of the loss in all experiments.

We use the transformers python package (Wolf et al., 2020), using the `bert-base-uncased` model for SNLI and Cancer, and `twitter-roberta-base` for Alcohol. We set learning_rate

[1]Cancer dataset will be released along with the paper.

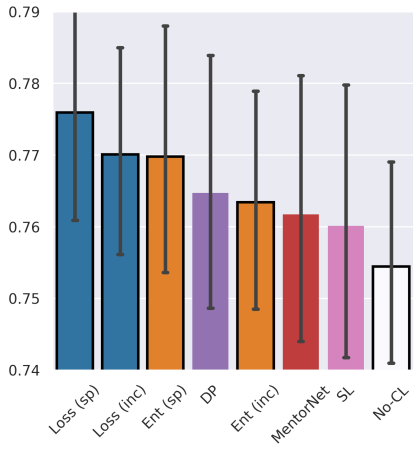


Figure 4: Overall accuracy averaged over three datasets with five random seeds. *Loss* is a curriculum that partitions the data based on loss-classes, and *Ent* partitions the data based on entropy-classes (§2.2). *inc* is the easy to hard curriculum shown in Figure 1b, *sp* is the specialized curriculum obtained by curriculum discovery (§2.4), which is different for each dataset.

= $1e-5$, batch_size = 16, epochs = 10 (we confirm that this number of iterations is sufficient for all models to converge), and the optimizer to Adam (Kingma & Ba, 2017). For each experiment, we train five models using five random seeds applied to both `pytorch` and `numpy`. Additionally, during all data pre-processing, splitting, and sub-sampling, the random seed is set to 0, and a single NVIDIA A100 40GB GPU is used for training. The development set is used to determine the best training step which is used for the final evaluation.

In addition, we conduct a hyper-parameter search over the (r, s) for each weight function, for the three difficulty classes. We set the search space of r to be from -10 to 10 with a step of 2, and of s to be -0.5 to 1.5 with a step of 0.25. We observe that changes smaller than this step size have little effect on performance. The search is run for at least 100 trials using the method described in §2.4. Each trial is run with 5 random seeds and the result is averaged. The search objective is to maximize development-set accuracy.

3.4 Performance Gain from Curricula Discovery

Results are shown in Figure 4. Accuracy is averaged over the six datasets (full and balanced version of each dataset). The gradually increasing curriculum (*inc*) achieves a significant improvement over *No-CL* using either of the scoring functions while being a static, off-the-shelf curriculum configuration. This shows that the effectiveness of our approach of partitioning the data using generalized logistic functions (§2.1) as weighting functions. Moreover, the specialized (*sp*) curricula obtained through curriculum discovery perform significantly better than state-of-the-art CL approaches. In our three datasets, loss as a scoring function performs better on average than entropy. This is expected as loss values can capture sample difficulty with respect to the downstream learner (model), as apposed to entropy values which do not take into account the model.

Appendix B includes further breakdown of the results by dataset and accuracy across samples of different difficulty.

3.5 Characteristics of Discovered Curricula

Figure 5 shows the mean and 95% CI of the top 25 performing configurations on our datasets and scoring functions. We observe several insightful patterns: The resulting curricula are non-trivial and greatly differ from the known strategies reported in current literature, such as gradually increasing difficulty or anti-curriculum. In addition, the weights of hard samples tend to approach zero, supporting the hypothesis that either these instances are too difficult for the models to learn or they are noisy. The results support the principle of pruning techniques because noisy samples induce more

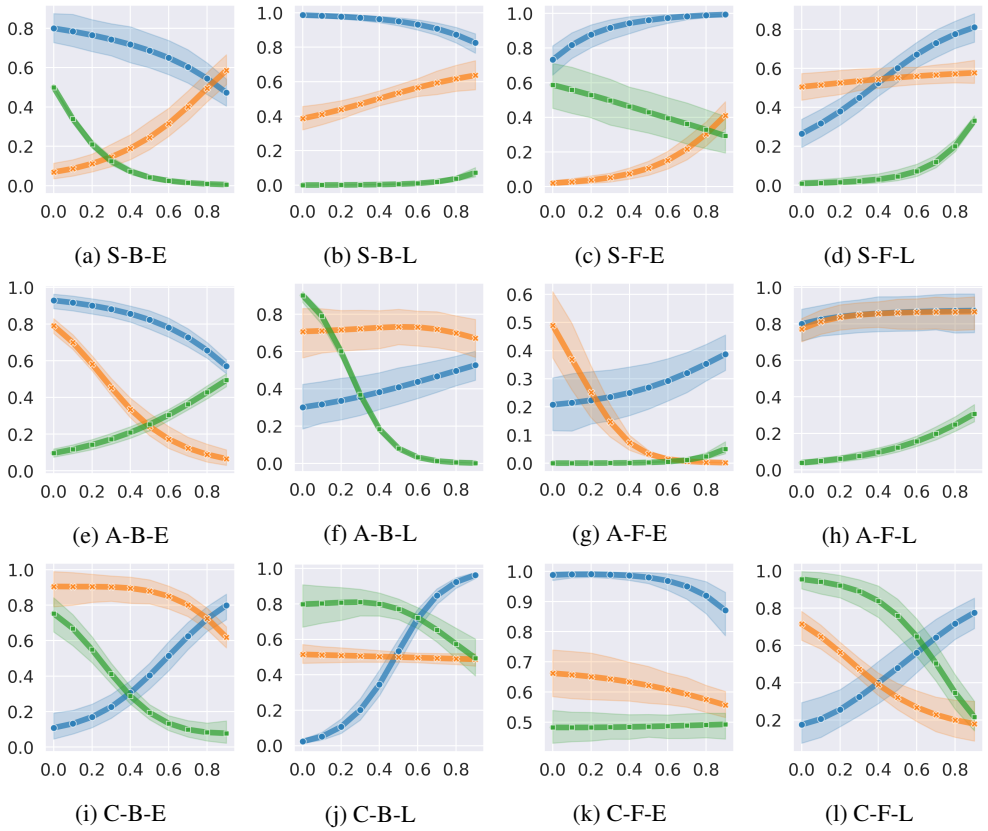


Figure 5: The notation is composed of the first character of the dataset name: {**S**NLI, **A**lcohol, **C**ancer}, {**B**alanced or **F**ull} dataset, and the difficulty score used {**E**ntropy, **L**oss}. The x-axis is the training progress and y-axis is the confidence assigned to samples of a difficulty-class. The blue line (circle marker) is *easy*, orange line (x marker) is *medium*, and green line (diamond marker) is *hard*. The solid line is the mean of the top 25 performing configurations for each dataset and scoring function pair, and the shaded area represents the 95% CI.

noise than useful signal; in several plots, hard samples increase in weight during the last portion of training after the model has stabilized. In addition, in SNLI and Alcohol *easy* samples carry the most significant weight, unlike Cancer, where *easy* samples are down-weighted early during the training. These weighting patterns reveal the relative importance of samples in each dataset. Finally, the full SNLI dataset with entropy-class partitions provides useful information. Figure 2a shows that entropy classes are highly imbalanced, with *hard* samples being much fewer than *easy* ones. In the optimal curriculum shown in Figure 5c, *hard* samples are assigned weights around 0.5, unlike the three other cases of SNLI. We attribute this result to the reduced presence/effect of *hard* samples.

3.6 Generalizable Curricula

Figure 6 shows the accuracy obtained when training using different curriculum configurations from Figure 5. Each cell in the figure is the average result of 5 seeds. We observe common characteristics among datasets that cause the curriculum to be transferable between them. First, the three top configurations are all products of down-sampled, balanced datasets. Second, the curricula obtained using the small balanced datasets generalize and achieve high performance on the large datasets. This is useful as it allows performing the hyper-parameter search much faster and cheaper on smaller datasets, providing evidence that the framework can be applied to large datasets by searching for a curriculum on a small subset of the data. Third, the *inc* curriculum is available off-the-shelf and performs consistently well across datasets and scoring functions. Fourth, as noted previously, instances of the Cancer dataset consist of long paragraphs, causing high variance in models trained

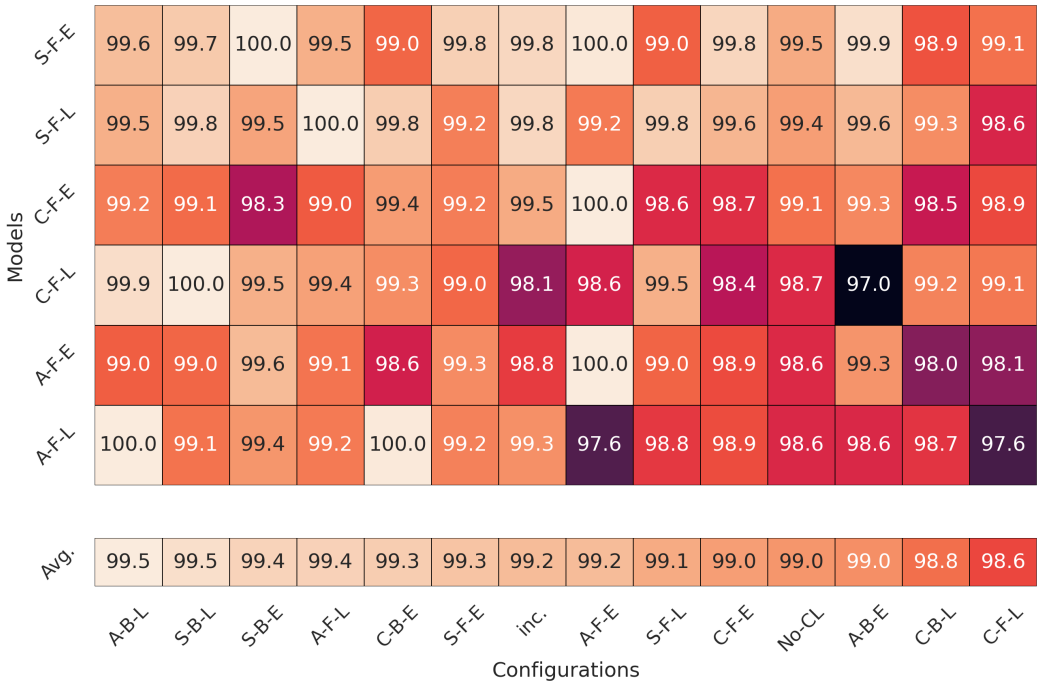


Figure 6: Using the same notation as Figure 5, the x-axis lists curriculum configurations discovered using a particular dataset and scoring function, in addition to the *inc* curriculum (Figure 1b), and *No-CL*. The y-axis lists models that are trained using each configuration. For example, the cell at the intersection of row "S-F-L" and column "A-F-E" represents a model trained using the SNLI full dataset partitioned by loss classes, and trained using the curriculum defined by the curves discovered using the full Alcohol dataset partitioned by entropy (Figure 5g). Each row of the table is normalized to match the scales of the different models.

using the dataset. Consequently, the curriculum obtained using the Cancer loss-class partitions is of low quality and performs the worst.

An extended version of Figure 6 is included in Appendix A with results of models trained with balanced versions of datasets.

4 Conclusion and Future Work

We introduce a curriculum learning (CL) framework that partitions the data into three groups of increasing difficulty, defines three parameterized weight functions to weight the losses of three groups, and provides the capability of tuning the weight function parameters to discover new curricula performing better than existing baselines. We demonstrated that this framework is capable of representing the major CL approaches. The proposed research opens a new paradigm in CL by removing the limitations imposed by selecting a single CL strategy, and instead, using a flexible framework to discover optimal curricula for given datasets and models, and extract valuable insights about the data. Our work has the limitation of using monotonic logistic functions as the weight curves can not take arbitrary forms. This challenge can be addressed in future work by adding an extra set of logistic functions and having every weight function be a linear combination of two or more logistic functions, which makes it possible to represent non-monotonic functions. Nevertheless, we have demonstrated that major CL approaches that estimate difficulty as a function of training loss result in monotonic curves, since gradient descent cause loss to be non-increasing.

Promising areas for future work include learning new difficulty indicators (centered on data, e.g. text difficulty), prioritizing medium level instances and those with greatest progress in training, and, developing challenge datasets that contain diverse data samples with different level of difficulty.

5 ETHICS STATEMENT

This investigation included publicly available yet sensitive data from social media, developed for an important task: obtaining population-level statistics about different public health issues. Our work does not access, use, or release any personal data, and only textual data is used for experiments.

6 REPRODUCIBILITY STATEMENT

The source code is included as supplementary material. It provides the required details for data processing, hyper-parameter setting, random seed setting, model evaluation, etc.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Hadi Amiri, Timothy Miller, and Guergana Savova. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2401–2410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1255. URL `https://aclanthology.org/D17-1255`.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ACM International Conference Proceeding Series*, volume 382, pp. 1–8, New York, New York, USA, jul 2009. ACM Press. ISBN 9781605585161. doi: 10.1145/1553374.1553380.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 13(2), 2012.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems (NIPS)*, 24, 2011.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp. 632–642. Association for Computational Linguistics (ACL), 2015.

Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Antoine Caubrière, Natalia Tomashenko, Antoine Laurent, Emmanuel Morin, Nathalie Camelin, and Yannick Estève. Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. In *20th Annual Conference of the International Speech Communication Association (InterSpeech)*, pp. 1198–1202, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070. PMLR, 2019.

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pp. 482–495. PMLR, 2017.

Alex Graves, Marc G Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *34th International Conference on Machine Learning, ICML 2017*, volume 3, pp. 2120–2129, 2017. ISBN 9781510855144.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 135–150, 2018.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pp. 107–112. Association for Computational Linguistics (ACL), 2018.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*, pp. 2304–2313. PMLR, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Tom Kocmi and Ondřej Bojar. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pp. 379–386, 2017.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems (NIPS)*, 23:1189–1197, 2010.

Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, Rene Vidal, and Vittorio Murino. Curriculum dropout. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3564–3572. IEEE Computer Society, 2017.

Yixin Nie, Xiang Zhou, and Mohit Bansal. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9131–9143, 2020.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pp. 708–717, 2016.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pp. 1162–1172, 2019.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. *NAACL HLT 2018*, pp. 180, 2018.

FJ Richards. A flexible growth function for empirical use. *Journal of experimental Botany (JXB)*, 10(2):290–301, 1959.

Douglas LT Rohde and David C Plaut. Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109, 1999.

Brendan van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: the importance of being unhinged. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pp. 10–18, 2015.

Terence D Sanger. Neural network learning control of robot manipulators using gradually increasing task difficulty. *IEEE transactions on Robotics and Automation*, 10(3):323–333, 1994.

Nikolaos Sarafianos, Theodore Giannakopoulos, Christophoros Nikou, and Ioannis A Kakadiaris. Curriculum learning for multi-task classification of visual attributes. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2608–2615, 2017.

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. *Advances in Neural Information Processing Systems*, 32: 11095–11105, 2019.

Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

Samarth Sinha, Animesh Garg, and Hugo Larochelle. Curriculum by smoothing. *Advances in Neural Information Processing Systems*, 33, 2020.

Elissa R Weitzman, Kara M Magane, Po-Hua Chen, Hadi Amiri, Timothy S Naimi, and Lauren E Wisk. Online searching and social media to detect alcohol use risk at population scale. *American journal of preventive medicine (ACPM)*, 58(1):79–88, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? In *International Conference on Learning Representations (ICLR)*, 2021. URL `https://openreview.net/forum?id=tW4QEInpni`.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 6095–6104, 2020.

Yinfei Yang, Oshin Agarwal, Chris Tar, Byron C Wallace, and Ani Nenkova. Predicting annotation difficulty to improve task routing and model performance for biomedical information extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pp. 1471–1480, 2019.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*, 2018.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1903–1915, 2019.

Tianyi Zhou and Jeff Bilmes. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *International Conference on Learning Representations*, 2018.

Tianyi Zhou, Shengjie Wang, and Jeff A Bilmes. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

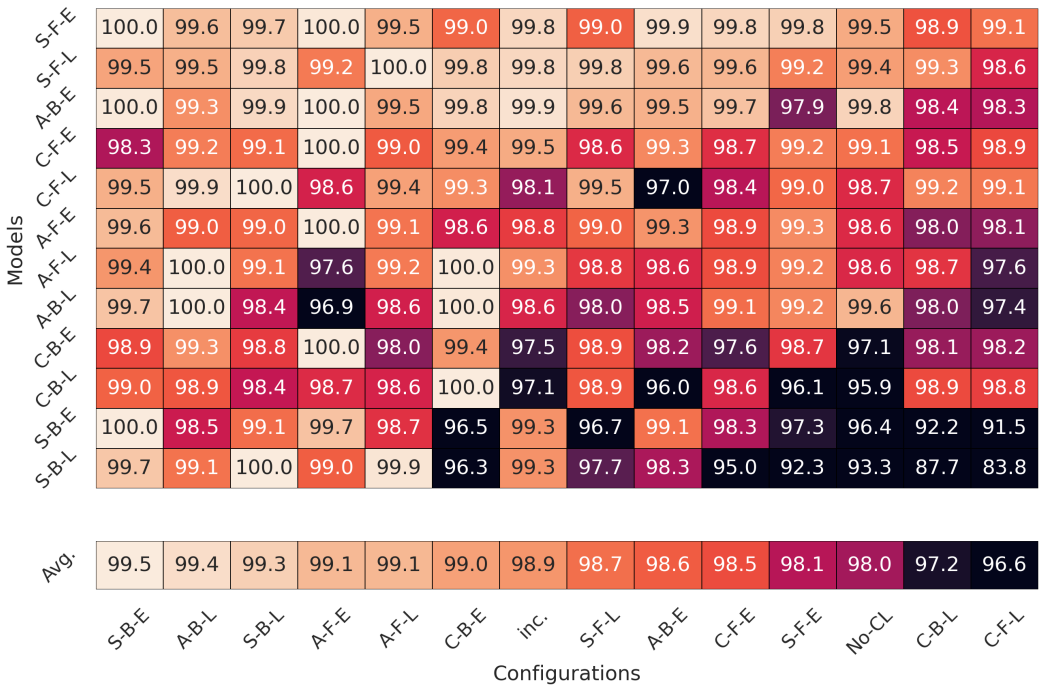


Figure 7: An extended version of Figure 6 including experiments on balanced versions of the datasets.

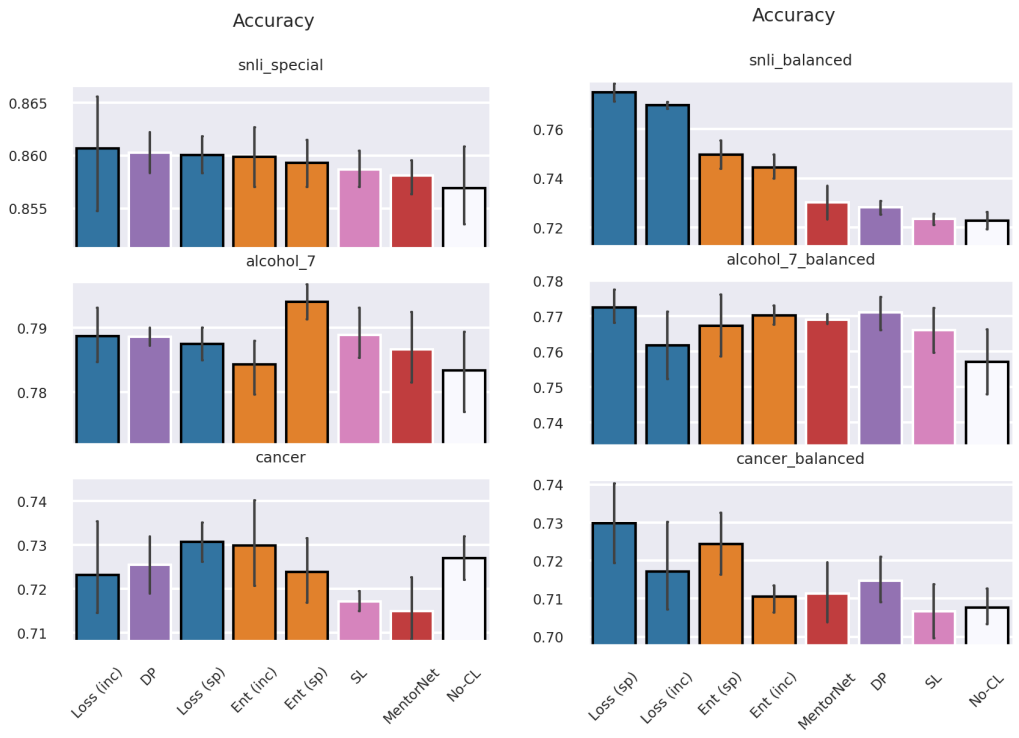
A Extended Configuration Generalizablity Experiments

Figure 7 shows the full results including evaluation on the balanced datasets.

B ACCURACY BREAKDOWN

The accuracy achieved on each dataset using each approach is shown in Figure 8.

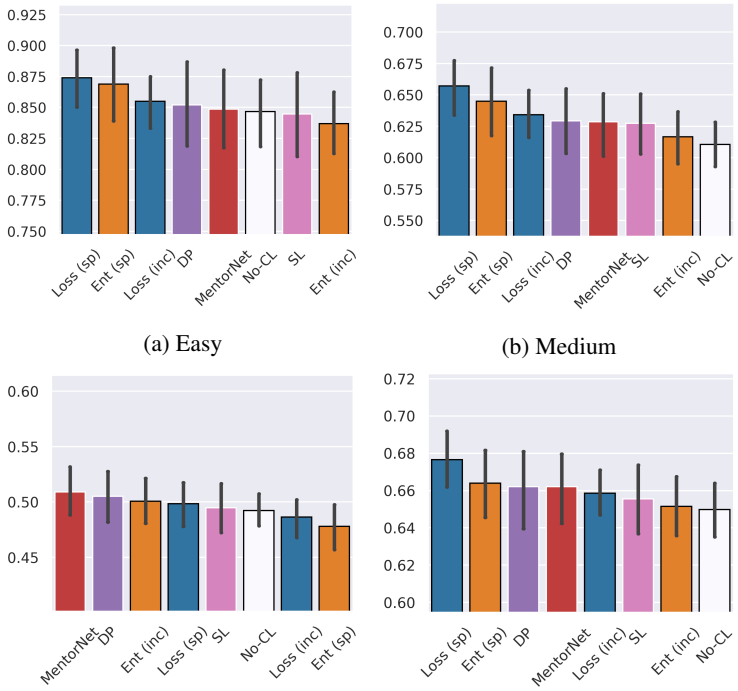
Models that achieve a high accuracy may be generating correct predictions for a high percentage of easy samples while failing to correctly predict the output of the *medium* and *hard* samples. Standard evaluation benchmarks often contains artifacts that make it east to correctly predict the label of some samples (Gururangan et al., 2018; Poliak et al., 2018). Therefore, it is important to closely analyze the model's performance on harder instances. We break down the accuracy of each difficulty class (with entropy-class partitioning) for balanced, full, and all datasets in Figures 9, 10, and 11, respectively. Our approach is exceptionally powerful in predicting samples of *medium* difficulty. Furthermore, the accuracy achieved on *hard* samples is almost equal by all approaches, including No-CL, supporting the hypothesis that those samples tend to be noisy or inaccurately labeled.



(a) Full datasets.

(b) Balanced datasets.

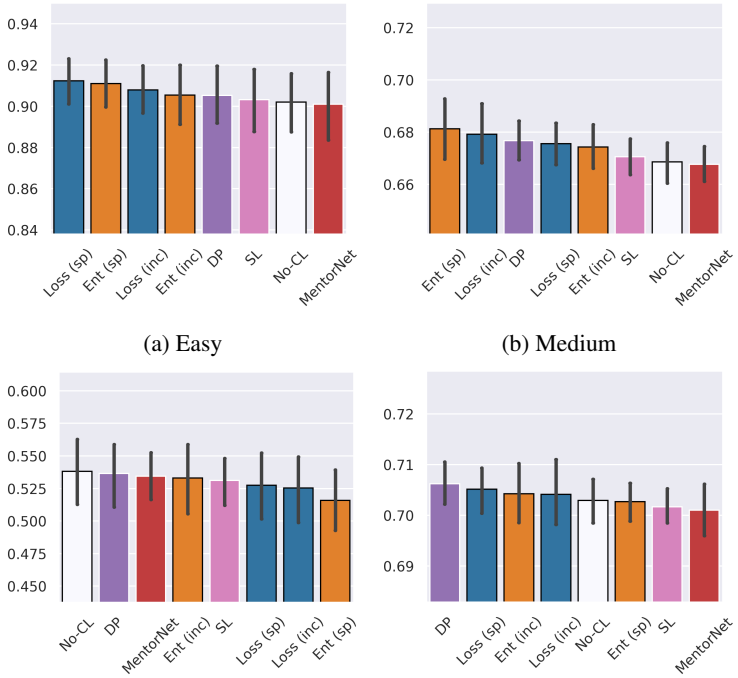
Figure 8: Accuracy of different CL approaches on each dataset.



(a) Easy (b) Medium

(c) Hard (d) Balanced Accuracy

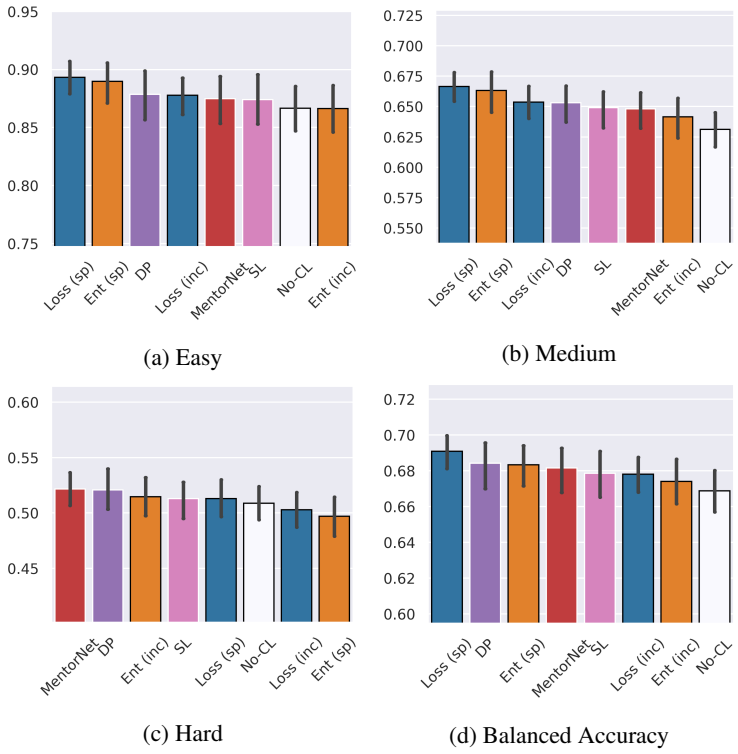
Figure 9: Average over balanced datasets.



(a) Easy (b) Medium

(c) Hard (d) Balanced Accuracy

Figure 10: Average over full datasets.



(a) Easy

(b) Medium

(c) Hard

(d) Balanced Accuracy

Figure 11: Average over all datasets.

C DATA CATEGORIES DISTRIBUTION

Class	**Count**
(no)	5,325
(yes, light use, individual)	1,464
(yes, heavy use, individual)	964
(yes, not sure, individual)	457
(yes, heavy use, other)	423
(yes, heavy use, group)	284
(yes, light use, group)	161
Total	9,078

(a) Alcohol

Class	**Count**
(irrelevant, no patient experience)	1,996
(relevant, breast cancer)	617
(relevant, colon cancer)	444
(relevant, brain cancer)	284
(irrelevant, none of the above)	251
(irrelevant, other cancer types)	162
(irrelevant, news related to cancer)	70
Total	3,824

(b) Cancer

Table 1: Statistics of the Alcohol and Cancer datasets.

D FULL CURRICULUM SEARCH RESULTS

This section shows the top 25 performing configurations on each dataset scoring function pair. Blue lines (circle marker) is *easy*, orange (x marker) is *medium*, and green (diamond marker) is *hard*. Above each plot is the development set accuracy. The plot legend contains the parameters (r, s) for each class.

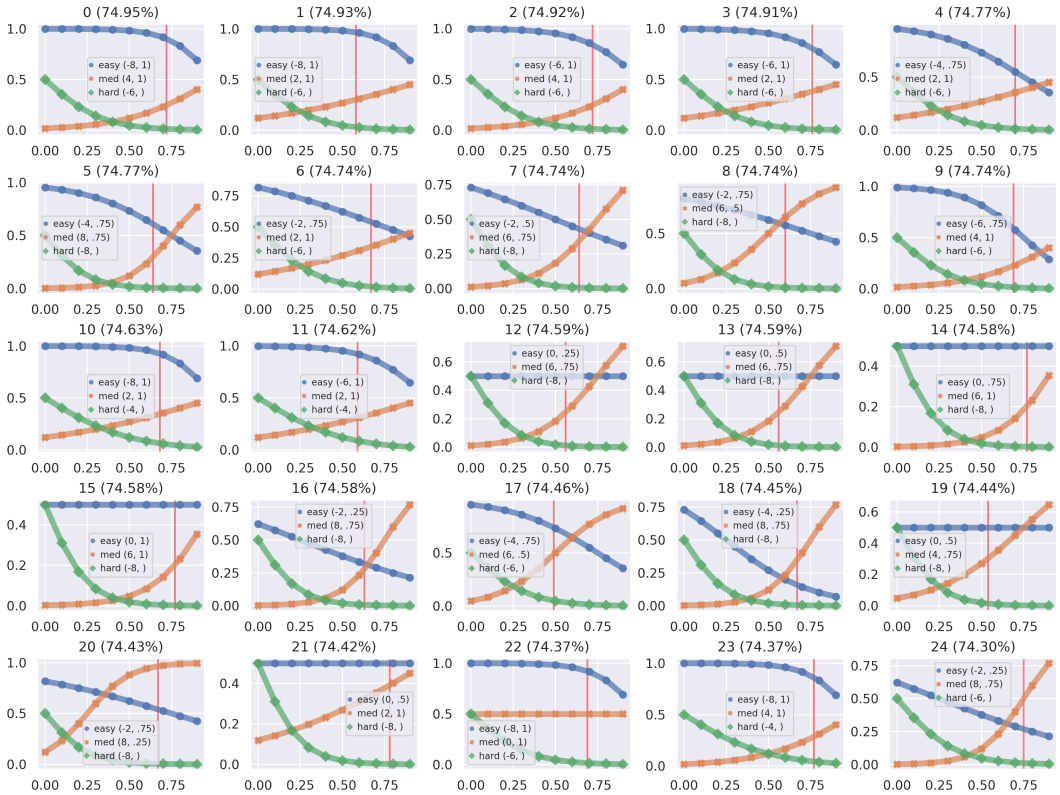


Figure 12: S-B-E

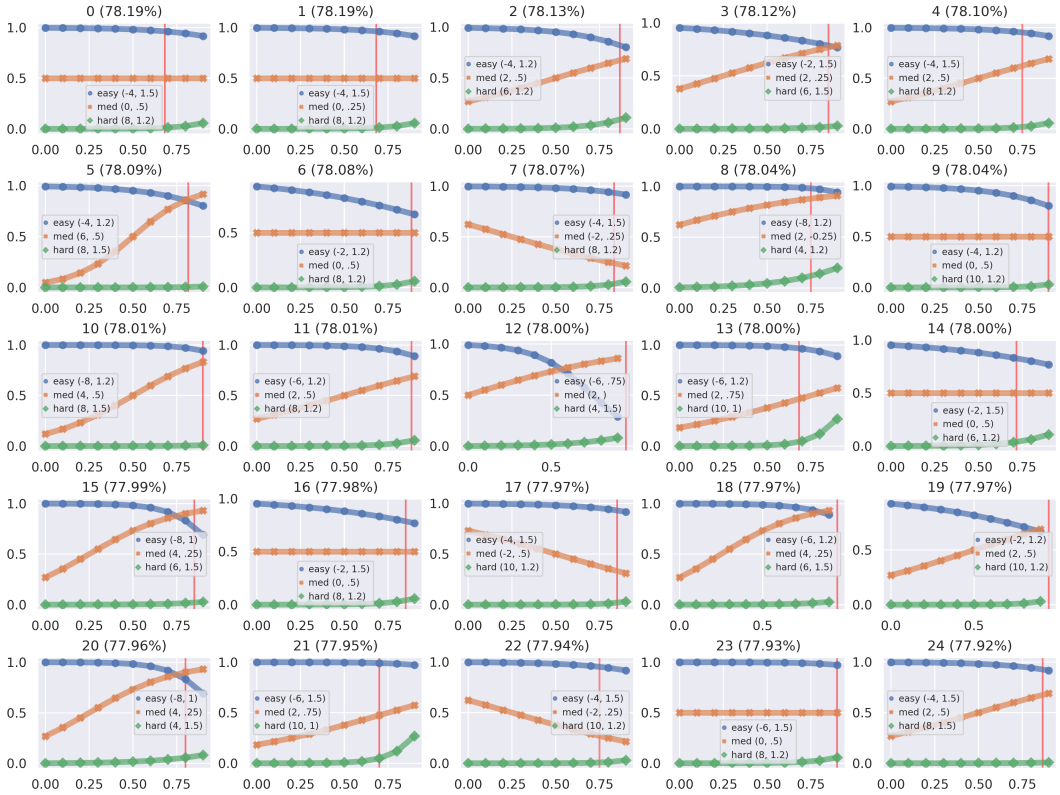


Figure 13: S-B-L

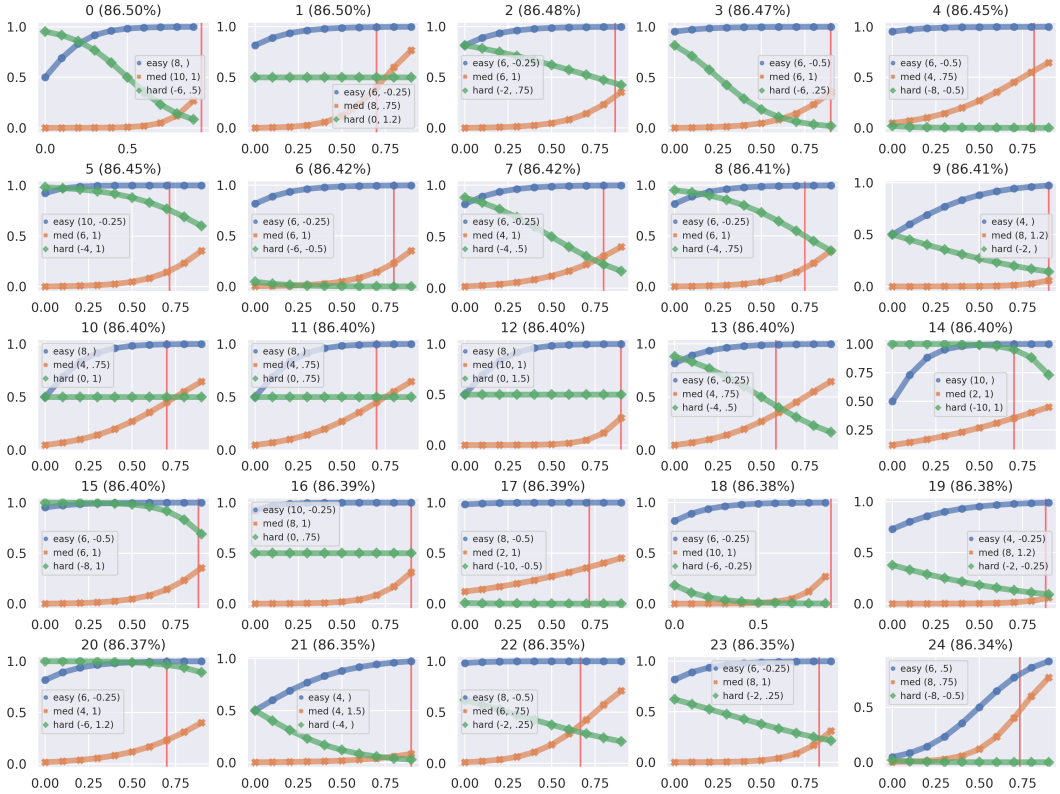


Figure 14: S-F-E

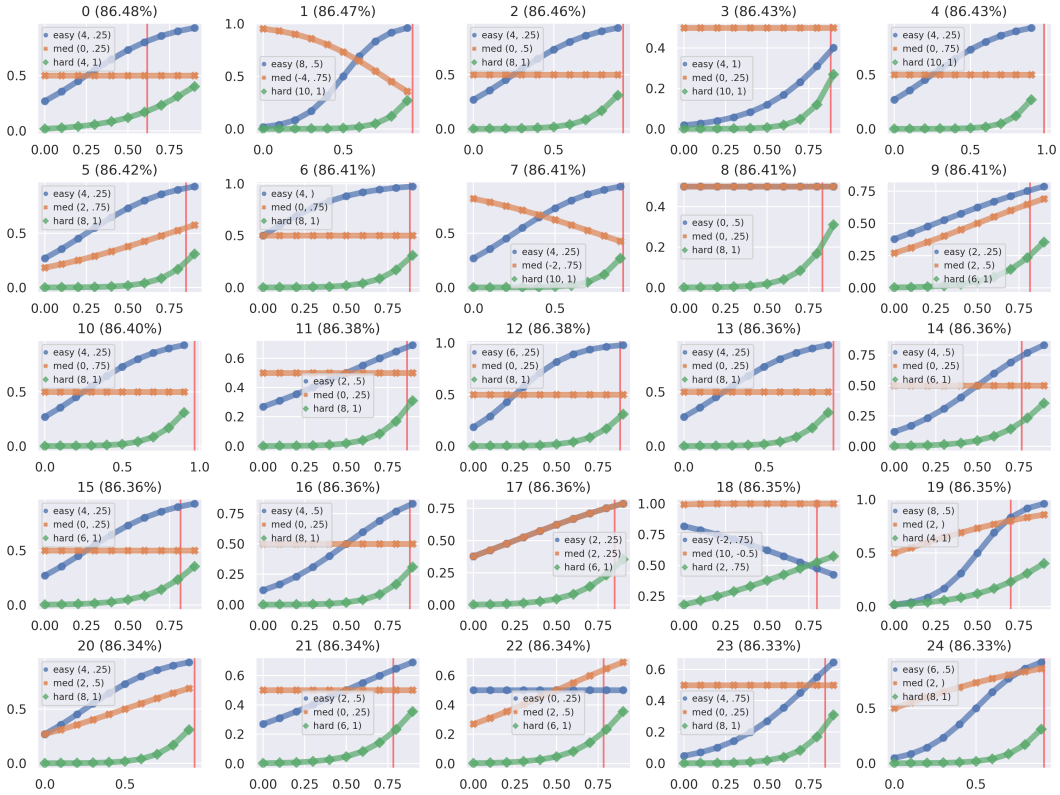


Figure 15: S-F-L

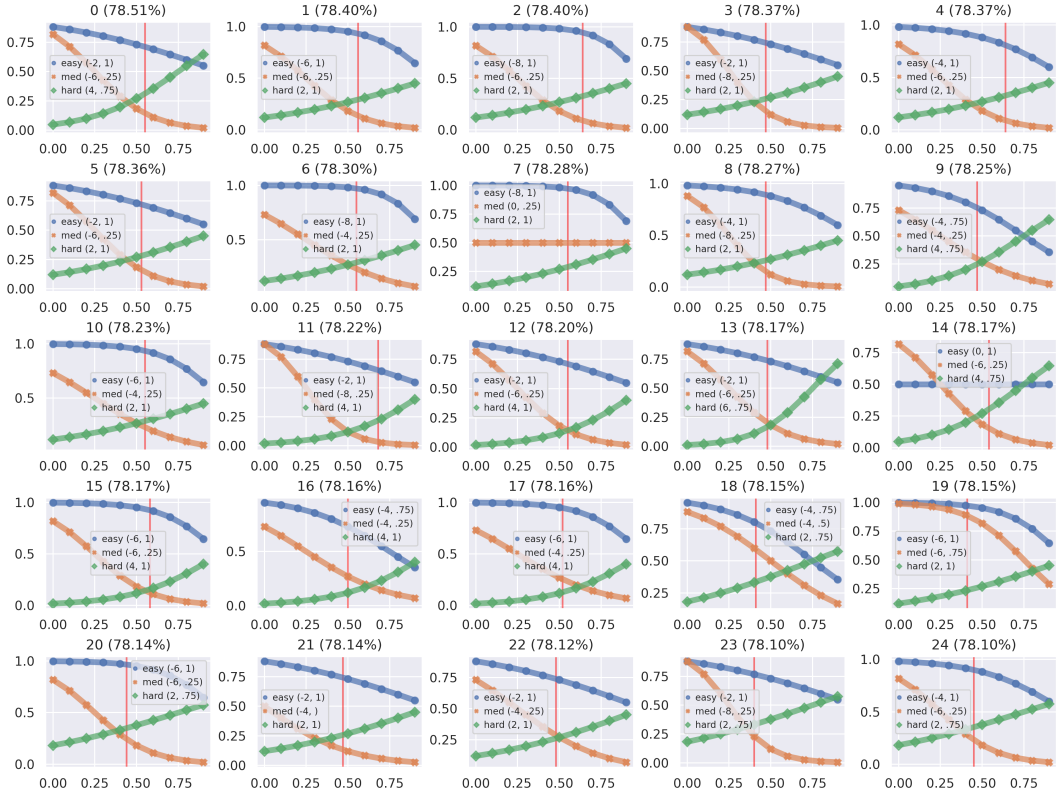


Figure 16: A-B-E

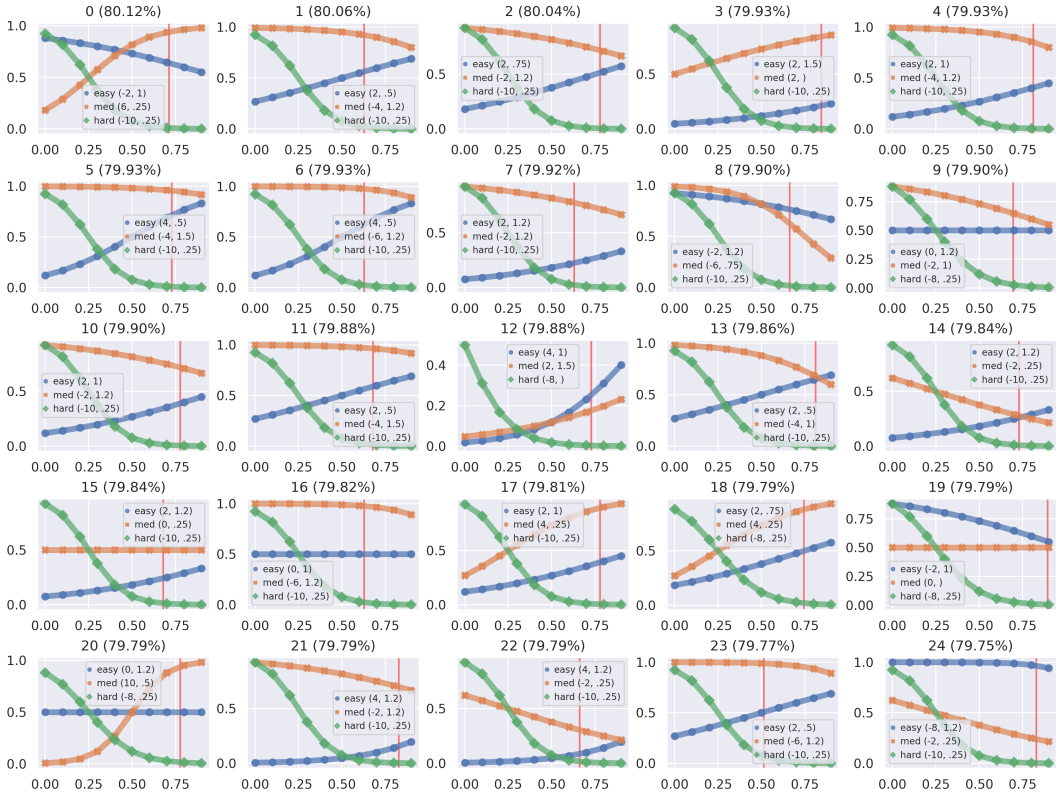


Figure 17: A-B-L

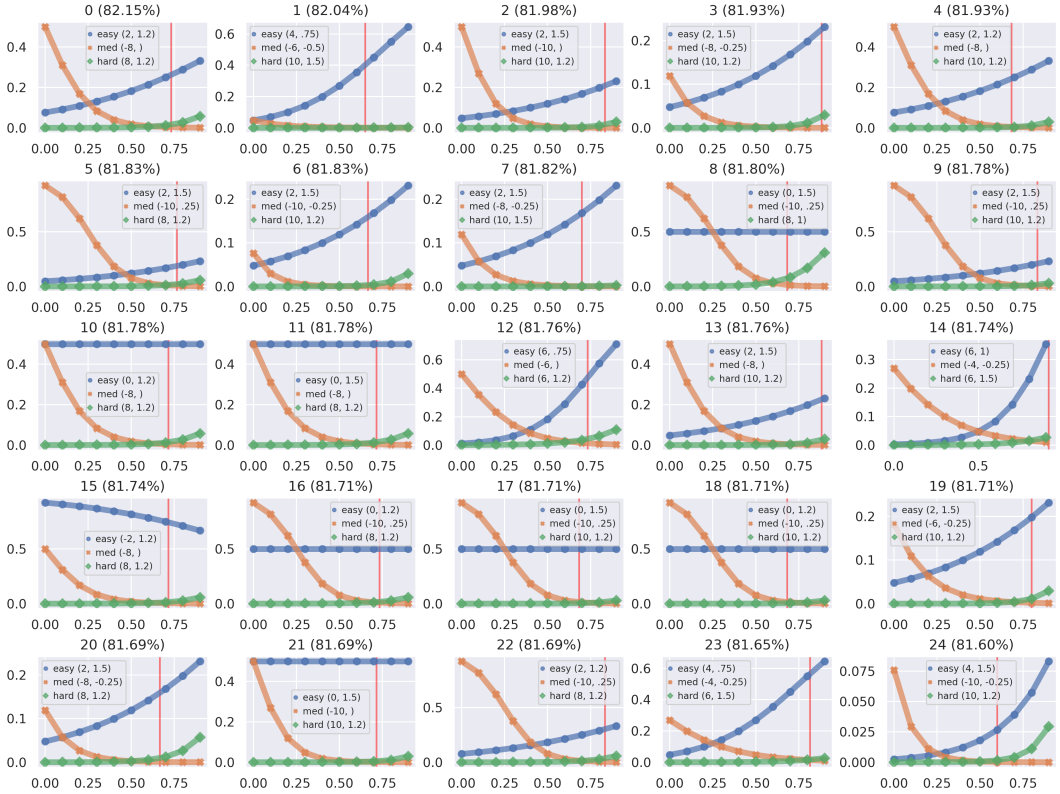


Figure 18: A-F-E

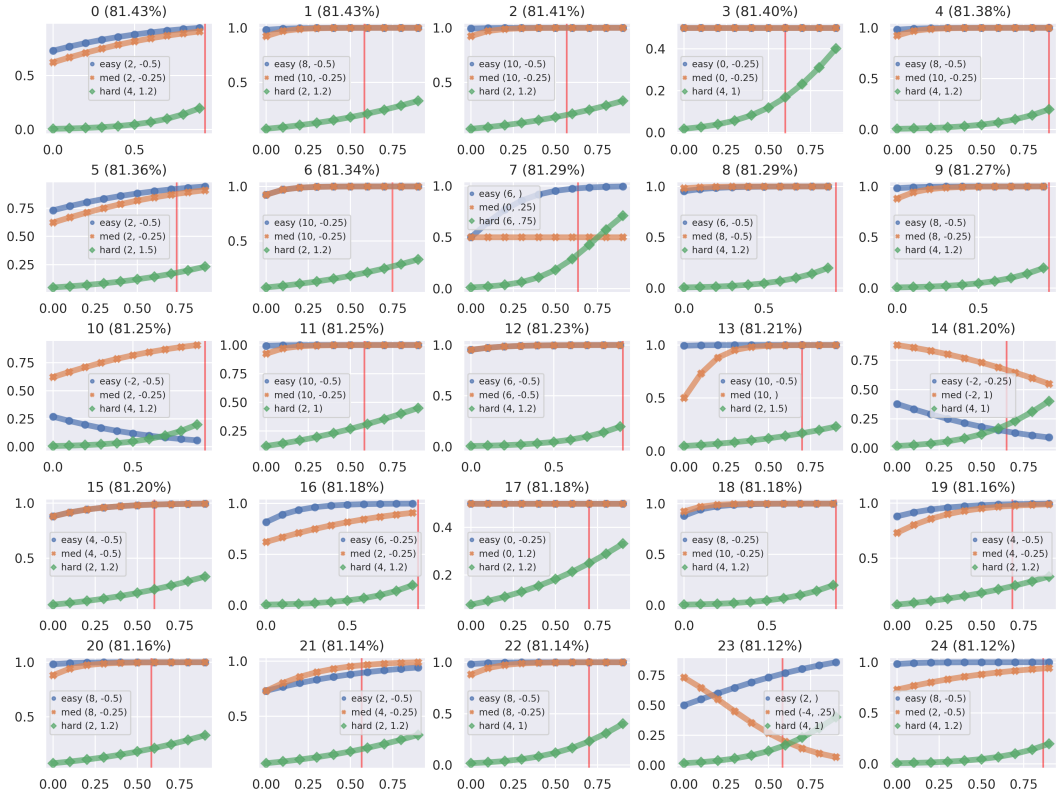


Figure 19: A-F-L

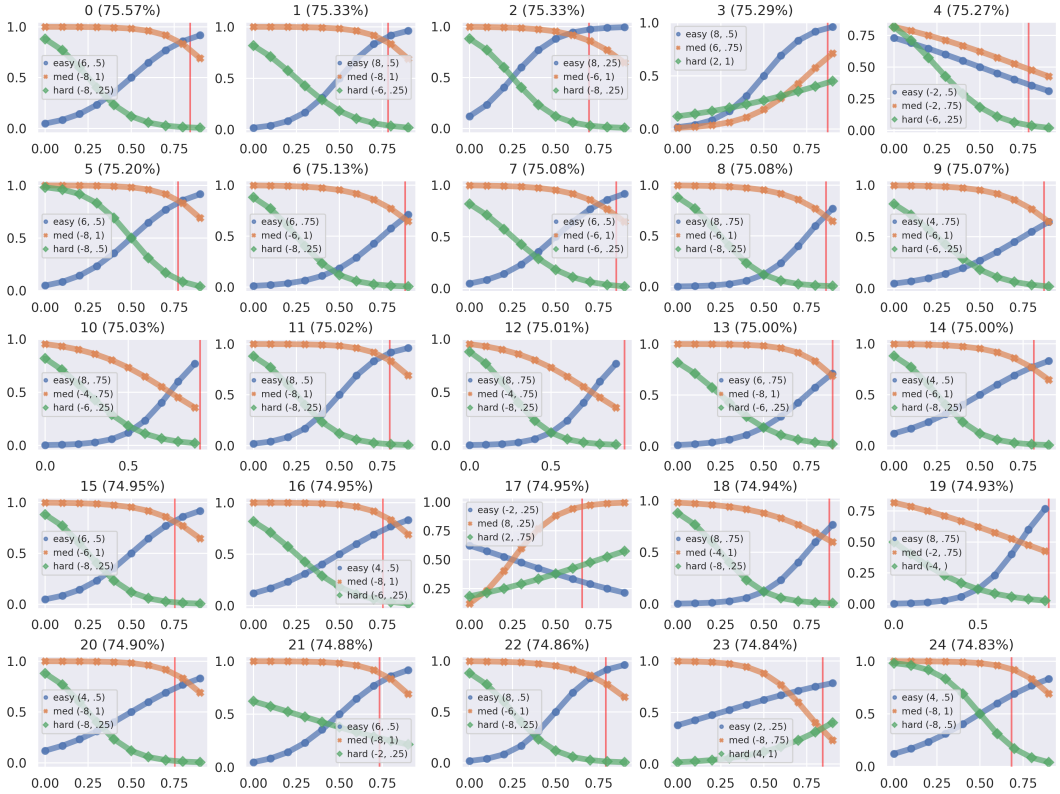


Figure 20: C-B-E

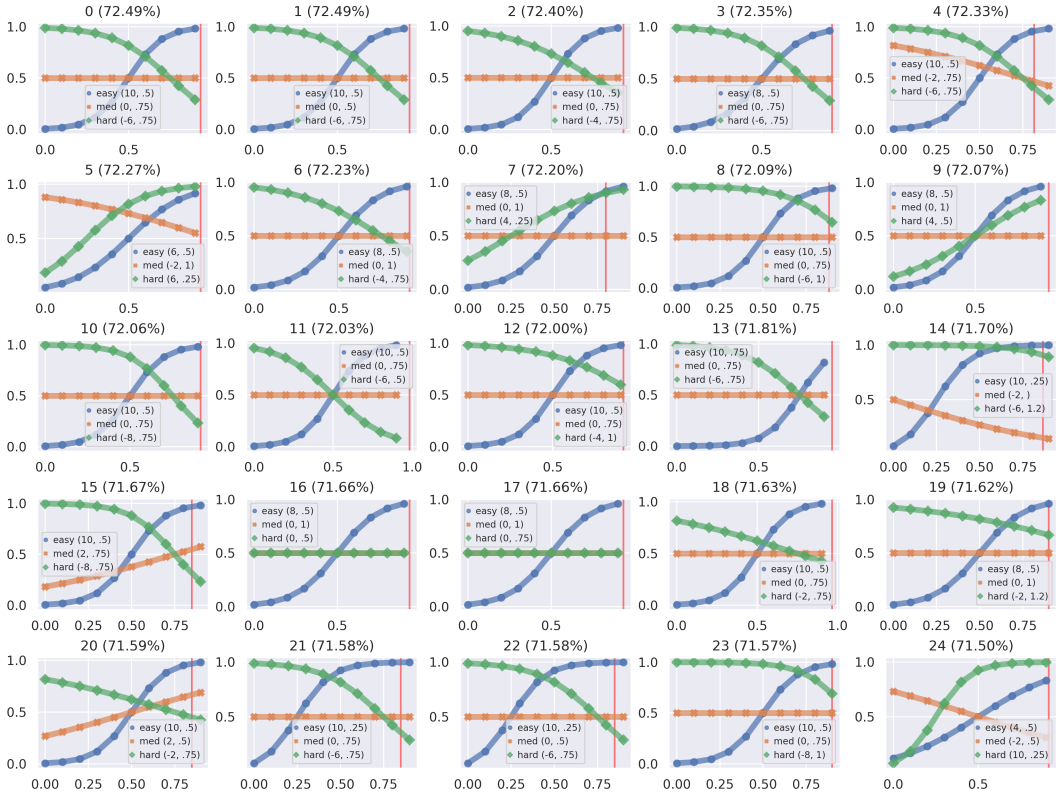


Figure 21: C-B-L

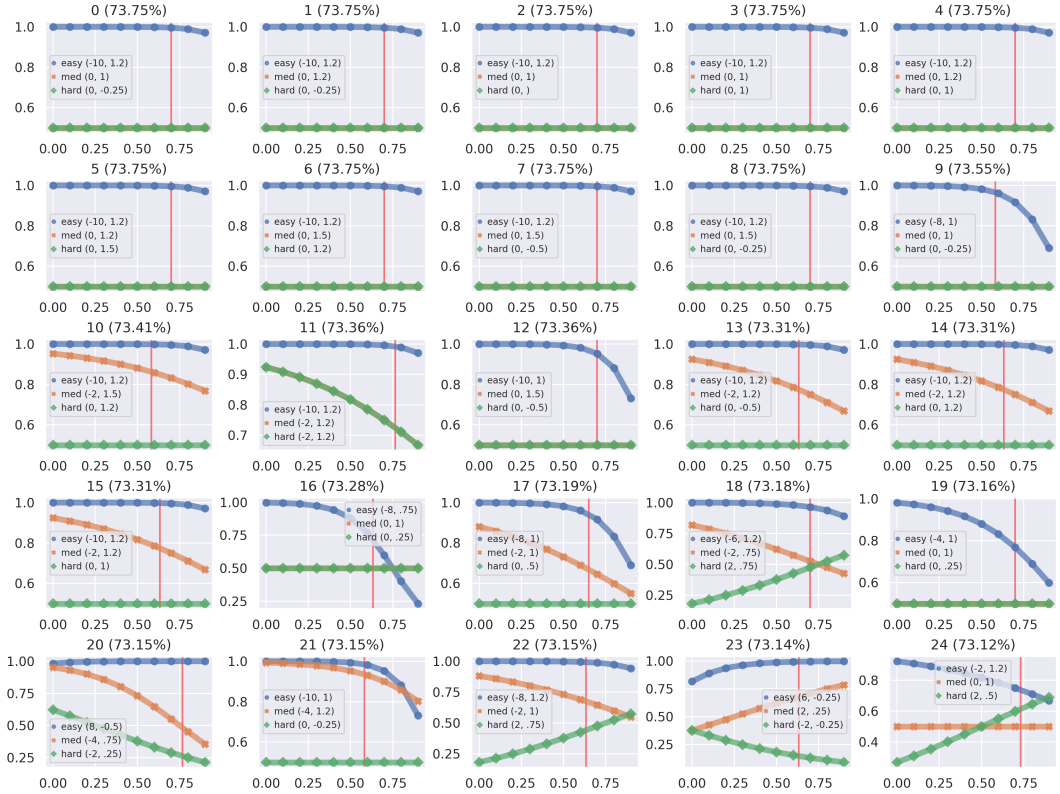


Figure 22: C-F-E

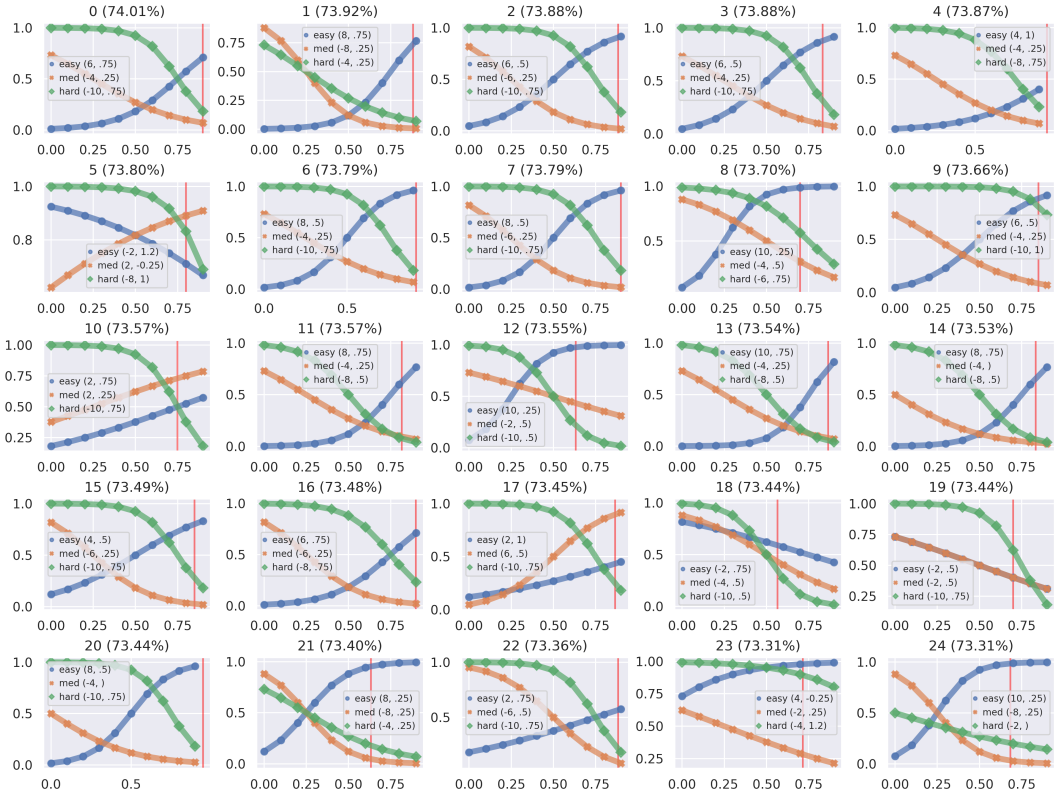


Figure 23: C-F-L