CurBench: Curriculum Learning Benchmark

Yuwei Zhou [1] **Zirui Pan** [1] **Xin Wang** [1] **Hong Chen** [1] **Haoyang Li** [1] **Yanwen Huang** [1] **Zhixiao Xiong** [1]
Fangzhou Xiong [1] **Peiyang Xu** [1] **Shengnan Liu** [1] **Wenwu Zhu** [1]

Abstract

Curriculum learning is a training paradigm where machine learning models are trained in a meaningful order, inspired by the way humans learn curricula. Due to its capability to improve model generalization and convergence, curriculum learning has gained considerable attention and has been widely applied to various research domains. Nevertheless, as new curriculum learning methods continue to emerge, it remains an open issue to benchmark them fairly. Therefore, we develop CurBench, the first benchmark that supports systematic evaluations for curriculum learning. Specifically, it consists of 15 datasets spanning 3 research domains: computer vision, natural language processing, and graph machine learning, along with 3 settings: standard, noise, and imbalance. To facilitate a comprehensive comparison, we establish the evaluation from 2 dimensions: performance and complexity. CurBench also provides a unified toolkit that plugs automatic curricula into general machine learning processes, enabling the implementation of 15 core curriculum learning methods. On the basis of this benchmark, we conduct comparative experiments and make empirical analyses of existing methods. CurBench is open-source and publicly available at https://github.com/THUMNLab/CurBench.

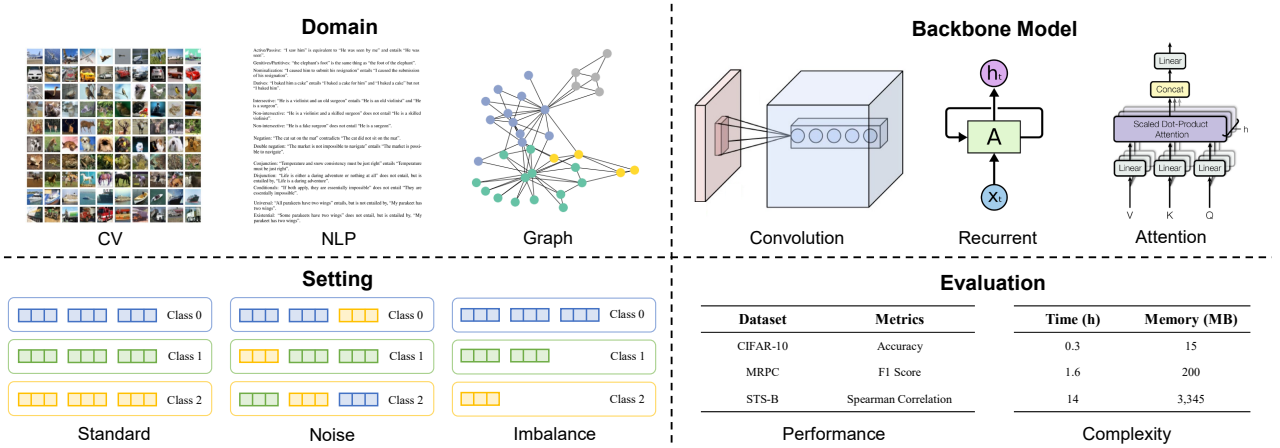
1. Introduction

Throughout the development of machine learning, a large number of works have been greatly influenced by human learning. Curriculum learning is such a research topic within machine learning that draws inspiration from a remarkable aspect of human learning: curriculum, i.e., learning in a purposeful and meaningful order (Wang et al., 2021a; Soviany et al., 2022). In contrast to conventional machine learning methods where training examples are randomly input, curriculum learning aims to facilitate learning by gradually increasing the difficulty of data or tasks experienced by the model (Bengio et al., 2009). Since this easy-to-hard training paradigm is verified to bring the advantage of enhancing model generalization and accelerating convergence speed (Gong et al., 2016; Weinshall et al., 2018), it has aroused widespread interest among researchers in harnessing its potential across diverse application domains, such as computer vision (CV) (Guo et al., 2018; Soviany et al., 2020; Gui et al., 2017), natural language processing (NLP) (Platanios et al., 2019; Tay et al., 2019; Liu et al., 2018), graph machine learning (Li et al., 2023; Wang et al., 2021b; Wei et al., 2023; Qin et al., 2024; Yao et al., 2024), multimodal learning (Lan et al., 2023; Chen et al., 2023; Zhou et al., 2023), recommender systems (Chen et al., 2021b;a; Wu et al., 2023; Wang et al., 2023a), reinforcement learning (RL) (Florensa et al., 2017; Narvekar et al., 2017; Ren et al., 2018b), and others (Zhang et al., 2022; Zhou et al., 2022b).

Despite the significant progress and the wide application of curriculum learning, the increasing number of works has posed challenges in terms of their comparison and evaluation, mainly due to the differences in their experimental setups including datasets, backbone models, and settings. For instance, DCL (Saxena et al., 2019) and DDS (Wang et al., 2020) use the same WideResNet-28-10 model (Zagoruyko & Komodakis, 2016), but perform experiments on different datasets: CIFAR-100 and CIFAR-10 (Krizhevsky et al., 2009) respectively. Similarly, DI-HCL (Zhou et al., 2020) and CBS (Sinha et al., 2020) leverage the same ImageNet (Deng et al., 2009) dataset, but employ distinct models: ResNet-50 and ResNet-18 (He et al., 2016) respectively. Furthermore, while MCL (Zhou & Bilmes, 2018) and LRE (Ren et al., 2018a) utilize the same MNIST dataset and LeNet model (LeCun et al., 1998), they adopt different settings: standard and imbalanced labels respectively. Consequently, their experimental results cannot be compared directly, which makes it challenging to conduct a fair evaluation. The absence of a standardized evaluation not only hinders researchers from accurately assessing their own contributions when they propose a new

[1]Department of Computer Science and Technology, BNRIST, Tsinghua University. Correspondence to: Xin Wang <xin_wang@tsinghua.edu.cn>, Wenwu Zhu <wwzhu@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



Dataset	Metrics	Time (h)	Memory (MB)
CIFAR-10	Accuracy	0.3	15
MRPC	F1 Score	1.6	200
STS-B	Spearman Correlation	14	3,345

Figure 1. CurBench includes 15 datasets spanning 3 research domains, 9 backbone models, 3 training settings, and 2 evaluation dimensions, providing a comprehensive benchmark for existing curriculum learning methods.

method but also poses barriers for users when they seek a suitable method for their specific tasks.

To deal with this issue, researchers have made notable efforts to evaluate and summarize existing works. From a theoretical perspective, there have been surveys covering general curriculum learning (Wang et al., 2021a; Soviany et al., 2022) as well as specific ones for graph (Li et al., 2023) and RL (Narvekar et al., 2020; Portelas et al., 2020), all of which manage to formulate and categorize relevant methods comprehensively. Although they offer valuable theoretical insights, current surveys do not incorporate any practical implementation or experimental results. From an empirical perspective, there has been an open-source library on curriculum learning (Zhou et al., 2022a), which reproduces multiple related methods through a unified framework. Although it provides empirical results of the implemented methods, this library only supports the classification task on CIFAR-10, limited in experimental setups. In conclusion, the related works fail to address the open issue of evaluating and comparing curriculum learning methods completely.

In order to address the absence of benchmarks in this field, we propose CurBench, the first benchmark for systematic evaluations of curriculum learning, as shown in Figure 1. Concretely, it encompasses 15 prevalent datasets, spanning 3 research domains including CV, NLP, and graph to ensure the reliability of evaluation. These datasets are further preprocessed into 3 settings including standard, noise, and imbalance to reveal the capability of methods to enhance model generalization and robustness. Without loss of generality, a total of 9 prevalent backbone models of varying types and scales adapted to the above datasets are employed in an appropriate manner, incorporating corresponding hyperparameters, optimizers, and so on. Most of the datasets, settings, and models are commonly used in previous related works, while the rest are supplemented in this work to investigate how these methods can adapt to the tasks in other domains. For ease of use, this benchmark also provides a unified toolkit that plugs automatic curricula into general machine learning processes and reproduces a collection of 15 core curriculum learning approaches. Based on these implementations in CurBench, we further perform a comprehensive evaluation from 2 dimensions including performance and complexity, presenting the improvements the methods bring and the additional resources they consume.

Furthermore, we delve into our benchmark, organize experimental outcomes, conduct in-depth analyses, and obtain some intriguing findings. First, there has been no such method that outperforms others all the time, and the effectiveness depends on specific scenarios. Second, curriculum learning brings more significant improvements in noise settings than in standard and imbalance ones. Third, methods by teacher transferring have edges in noise settings, while methods by reweighting perform relatively well in imbalance settings. Lastly, methods involving gradient calculation and extra learnable networks generally have higher time and space complexity.

Our contributions are summarized as follows:

- We propose CurBench, the first benchmark on curriculum learning to the best of our knowledge.
- We conduct extensive experiments to impartially evaluate and compare the performance and complexity of existing curriculum learning methods under various experimental setups.
- We make in-depth analyses and demonstrate intriguing observations on curriculum learning based on empirical results derived from CurBench.

Domain	Dataset	Setting	Training	Validation	Test	Class	Metrics
CV	CIFAR-10	Standard / Noise-0.4	45,000	5,000	10,000	10	Accuracy
		Imbalance-50	12,536	5,000	10,000	10	Accuracy
	CIFAR-100	Standard / Noise-0.4	45,000	5,000	10,000	100	Accuracy
		Imbalance-50	12,536	5,000	10,000	100	Accuracy
	Tiny-ImageNet	Standard / Noise-0.4	90,000	10,000	10,000	200	Accuracy
		Imbalance-50	22,700	10,000	10,000	200	Accuracy
NLP	RTE	Standard / Noise-0.4	2,490	277	-	2	Accuracy
	MRPC	Standard / Noise-0.4	3,668	408	-	2	F1 Score
	STS-B	Standard / Noise-0.4	5,749	1,500	-	6	Spearman
	CoLA	Standard / Noise-0.4	8,551	1,043	-	2	Matthews
	SST-2	Standard / Noise-0.4	67,349	872	-	2	Accuracy
	QNLI	Standard / Noise-0.4	104,743	5,463	-	2	Accuracy
	QQP	Standard / Noise-0.4	363,846	40,430	-	2	F1 Score
	MNLI-(m/mm)	Standard / Noise-0.4	392,702	9,815/9,832	-	3	Accuracy
Graph	MUTAG	Standard / Noise-0.4	150	19	19	2	Accuracy
	PROTEINS	Standard / Noise-0.4	890	111	112	2	Accuracy
	NCI1	Standard / Noise-0.4	3,288	411	411	2	Accuracy
	ogbg-molhiv	Standard / Noise-0.4	32,901	4,113	4,113	2	ROC-AUC

Table 1. The statistics of 15 datasets adopted in CurBench, which covers a wide range of scales across 3 research domains in 3 settings. "Spearman" and "Matthews" refers to the correlation coefficient. "Noise-0.4" means 40% data samples are independently attached with random incorrect labels. "Imbalance-50" means a ratio of 50 between the number of samples in the largest class and that in the smallest class in a long-tailed dataset where the number of samples for each class follows a geometric sequence. The imbalance setting is not applied to NLP and graph datasets, which are imbalanced originally.

2. Related Work

2.1. Curriculum Learning

Curriculum learning, much like many other topics in machine learning, draws inspiration from human learning. It refers to a training strategy where models learn from input data in a meaningful order, imitating the way humans learn from curricula. The emergence of this idea could at least be traced back to Elman's work (Elman, 1993) in 1993, which advocated the importance of starting small. In 2009, Bengio et al. (Bengio et al., 2009) first introduced a formal definition of curriculum learning and explored when, why, and how a curriculum could benefit machine learning. In the early stages, curricula for models were entirely predefined by humans, and the most typical method was named Baby Step (Spitkovsky et al., 2010). However, this type of predefined approach is not flexible and general enough for widespread applications. In 2010, Kumar et al. (Kumar et al., 2010) proposed self-paced learning (SPL), enabling automatic curriculum scheduling by ordering data according to their training loss. Subsequently, a variety of automatic curriculum learning methods have continued to emerge. For example, transfer learning methods (Weinshall et al., 2018; Hacohen & Weinshall, 2019) employ teacher models to offer student models curricula. Reinforcement learning methods (Graves et al., 2017; Matiisen et al., 2019; Zhao et al., 2020) allow teacher models to adapt curriculum based on the feedback from student models. In addition, there are other ones based on Bayesian optimization (Tsvetkov et al., 2016), meta-learning (Ren et al., 2018a; Shu et al., 2019), and adversarial learning (Zhang et al., 2020) for implementing automatic curriculum learning.

2.2. Summative Work on Curriculum Learning

To the best of our knowledge, CurBench is the first benchmark on curriculum learning. Despite no related benchmarks, there have been numerous efforts to investigate and summarize the curriculum learning methods from different perspectives. For example, Wang et al. (Wang et al., 2021a) survey curriculum learning and propose a general framework to cover the related methods by abstracting them into two key components, i.e., a difficulty measurer to tell what data or task is easy or hard to learn and a learning scheduler to decide when to learn the easier or harder part, and further categorize the methods according to the implementation of these two components. Soviany et al. (Soviany et al., 2022) also survey curriculum learning and propose a generic algorithm for it based on the definition of machine learning, i.e., data, modal, and task, and organize the methods according to their application domains and tasks. Narvekar et al. (Narvekar et al., 2020) survey the relevant methods applied to RL and abstract them into three steps, i.e., task generation, sequencing, and transfer learning. Portelas et

al. (Portelas et al., 2020) also focus on curriculum learning for RL, and classify the methods based on three questions, i.e., why, what control, and what optimize. Li et al. (Li et al., 2023) review the tailored methods for graph, and group them according to the tasks, i.e., node-level, link-level, and graph-level. However, these works only summarize and analyze the methods from the theoretical aspect. On the other hand, Zhou et al. (Zhou et al., 2022a) develop CurML, a code library for curriculum learning, which designs a unified framework for the reproduction and comparison of existing methods from the empirical aspect. Nevertheless, it can only conduct experiments on a single task within a specific domain, significantly limiting its generality and reliability. Therefore, it is necessary to develop a benchmark across diverse experimental setups for a fair, reliable, and systematic study on curriculum learning.

3. Curriculum Learning Benchmark

In this section, we describe our design for the benchmark in detail. First, we clarify the scope of this benchmark in Section 3.1. Then, we introduce the adopted datasets in Section 3.2, followed by the corresponding settings in Section 3.3 and the backbone models in Section 3.4. Lastly, we elaborate on the evaluation dimensions in Section 3.5.

3.1. Benchmark Scope

CurBench focuses on benchmarking existing prevalent curriculum learning methods for supervised tasks in CV, NLP, and graph domains. This is because CV and NLP are representative research domains in machine learning, with datasets in these areas frequently used to validate the performance of curriculum learning methods, as shown in Table 6. Graph data, being structured, differs from the unstructured data of images and text, contributing to the diversity of CurBench, and curriculum learning in the graph domain has gained significant attention recently. Besides, the main challenge of the tasks included in CurBench lies in designing appropriate curricula at the data level so that the models can be guided to better cope with standard, noisy, and imbalanced datasets. In contrast, the methods designed at the task level and specifically targeting the RL domain are not within the scope of this work. We plan to expand the scope of CurBench in a future version, as stated in Section 6.

3.2. Dataset

Table 1 outlines the datasets included in CurBench, all of which are publicly available and widely used in their respective domains. Besides, they vary in scale from hundreds of samples to hundreds of thousands. A brief introduction to the datasets and our preprocessing is listed as follows.

CV Domain: CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) consist of $32 \times 32 \times 3$ color images in 10 and 100 classes respectively. Tiny-ImageNet (Le & Yang, 2015) is a subset of the ILSVRC2012 version of ImageNet (Deng et al., 2009) and consists of $64 \times 64 \times 3$ down-sampled images. Since the test set of Tiny-ImageNet is not released with labels, we use the validation set as the test set. For these 3 datasets, we split the original training set into a new training set and a validation set with a 9:1 ratio.

NLP Domain: All 8 datasets are sourced from GLUE (Wang et al., 2018), which is a collection of tools for evaluating models across diverse natural language understanding tasks. GLUE originally contains 9 datasets, and we follow BERT (Devlin et al., 2018), excluding the problematic WNLI set and using the remaining 8 datasets. Since the test sets are not released with labels, we report the results on the validation sets.

Graph Domain: The ogbg-molhiv dataset belongs to Open Graph Benchmark (OGB) (Hu et al., 2020), a collection of realistic, large-scale, and diverse benchmark datasets for graphs. We strictly follow its origin split scheme, split ratios, and metrics. The other 3 datasets come from TU-Dataset (Morris et al., 2020), a collection that consists of over 120 graph datasets of varying sizes from a wide range of applications. Since there are no established training and test set split, we randomly divide the original datasets into training, validation, and test sets with an 8:1:1 ratio.

3.3. Setting

To robustly evaluate the curriculum learning methods, we establish the 3 settings as follows.

Standard: After dividing the datasets into training, validation, and test sets as mentioned above, we do not perform any further data processing.

Noise-p: We follow previous works (Zhang et al., 2016; Ren et al., 2018a; Shu et al., 2019) and apply uniform noise by independently changing the label of each sample in the training set to a random one with a probability of $p \in (0.0, 1.0]$. When $p = 0$, it degenerates to the standard setting.

Imbalance-r: We follow previous works (Cui et al., 2019; Shu et al., 2019) to form a long-tailed dataset by reducing the number of samples per class in the training set. Let $c \in \{0, 1, 2, ..., C-1\}$ be the class index, C be the number of classes, n_c be the number of samples in the c^{th} class, and then an originally balanced dataset satisfies $n_0 \approx n_1 \approx ... \approx n_{C-1}$. We implement the imbalance setting by requiring n_c to follow the exponential function $n_c = n_0 d^c$ where $d \in (0, 1)$ and define the imbalance factor $r = n_0 : n_{C-1}$ as the ratio between the number of samples in the largest class and that in the smallest class. When $r = 1$, it degenerates to the standard setting.

Domain	Model	Mechanism	Parameters
CV	LeNet	Convolution	$\sim$ 0.07M
	ResNet-18	Convolution	$\sim$ 11.2M
	ViT	Attention	$\sim$ 9.6M
NLP	LSTM	Recurrent	$\sim$ 10.4M
	BERT	Attention	$\sim$ 109M
	GPT2	Attention	$\sim$ 124M
Graph	GCN	Convolution	$\sim$ 0.01M
	GAT	Attention	$\sim$ 0.14M
	GIN	Isomorphism	$\sim$ 0.01M

Table 2. The statistics of 9 backbone models adopted in CurBench, which covers various mechanisms and scales. "$\sim$" signifies an approximation, and "M" represents million.

3.4. Backbone Model

Table 2 overviews the backbone models that we employ in CurBench. All the values in the last column are approximations because the number of parameters varies depending on the input sizes and output classes. All of the models are commonly applied to the aforementioned datasets, and they are distinct from each other in mechanism and model size.

CV Domain: LeNet (LeCun et al., 1998) is one of the earliest convolutional neural networks (CNN), which is composed of 3 convolution layers, two pooling layers, and some fully-connected layers. ResNet (He et al., 2016) is a classic CNN with residual connection designed for easier training of deeper networks, and ResNet-18 refers to the 18-layer version. ViT (Dosovitskiy et al., 2020) is the standard Transformer directly applied to images by treating image patches as word tokens. ViT in CurBench is not pretrained because its pretrained weights are derived from ImageNet (Deng et al., 2009), which leads to the risk of data leakage when evaluating its performance on Tiny-ImageNet (Le & Yang, 2015), a subset of ImageNet.

NLP Domain: LSTM (Hochreiter & Schmidhuber, 1997) is a typical recurrent neural network (RNN), which introduces gate functions to control what to remember and what to forget in the face of long sequences. BERT (Devlin et al., 2018) is a deep bidirectional Transformer pretrained by masked language model task and it excels at semantic representation due to its encoder-based architecture. GPT2 (Radford et al., 2019) is a decoder-based Transformer pretrained through left-to-right language modeling objectives, and as a result, works well on text generation. BERT and GPT2 in CurBench are pretrained because training them from scratch would result in poor performance, making it difficult to maintain consistency with their suggested performance.

Graph Domain: GCN (Kipf & Welling, 2016) is a variant of CNN, designed to operate directly on graphs. Its insight lies in the choice of convolutional architecture via a localized first-order approximation of spectral graph convolutions. GAT (Veličković et al., 2017) introduces masked self-attentional layers based on GCN to enable implicitly specifying different weights to different nodes in a neighborhood. GIN (Xu et al., 2018) is developed based on Weisfeiler-Lehman test theory and emphasizes the importance of summation as the readout function.

3.5. Evaluation

To ensure a comprehensive analysis of existing methods, we consider the following 2 evaluation dimensions.

Performance: We adopt the widely accepted metrics on each dataset, such as accuracy on image, F1 score, Spearman Correlation, and Matthews Correlation on the GLUE benchmark, AUC (Yang et al., 2021) on graph. To display the results clearly, we report the average and standard deviation of the metric over 5 runs for each dataset.

Complexity: It is essential to examine the time and space complexity of each method because they always cost extra computational time and sources to assess model competence and data difficulty for appropriate curricula design. We record the training time and maximum memory consumption on the same GPU device as the indicators of the complexity.

4. CurBench Toolkit

4.1. Modules

To facilitate the use of our CurBench, we develop a companion toolkit based on CurML (Zhou et al., 2022a) for the entire pipeline of applying curriculum learning to various machine learning tasks, reproducing 15 core methods. Compared to CurML, this toolkit extends the methods to accommodate inputs in various data formats and diverse output evaluation metrics and provides searched hyperparameters for each method. As illustrated in Figure 2, we summarize and abstract the whole toolkit into 5 modules: data processing, model loading, objective fitting, curriculum learning, and evaluation.

Data Processing: This module aims to prepare data according to the specified dataset and setting. Given a data name in a format like "cifar10", "cifar100-noise-p" or "tinyimagenet-imbalance-r", this module can automatically parse it, split the dataset into training, validation, and test set, and process the training set by adding noise with probability p or forming imbalance with factor r.

Model Loading: This module is used to initialize the model based on the model name and the target dataset. For instance, CV models need to modify their input layer to accommodate input images and patch sizes. Similarly, graph models require node features and edge relationships when construct-

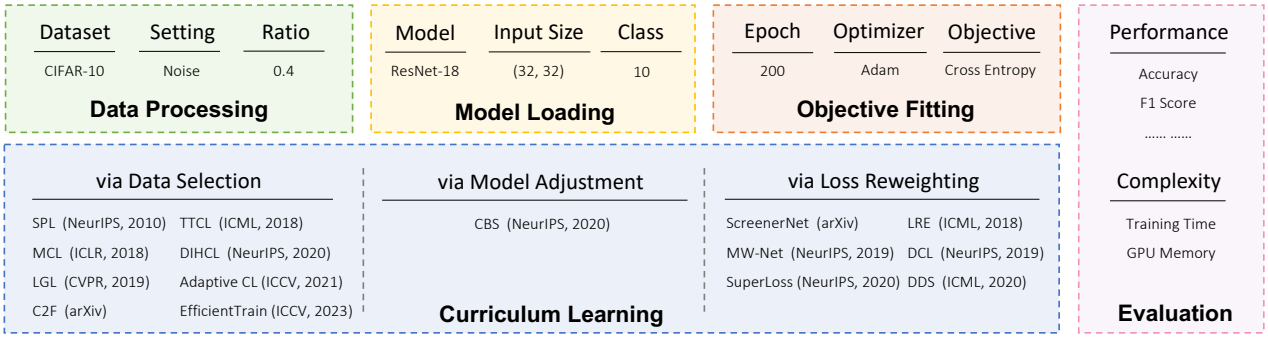


Figure 2. Our CurBench toolkit, which is composed of 5 modules, offers a unified and complete pipeline from initiation to evaluation, aiming for easy implementation and reproduction of curriculum learning methods. This figure showcases an example of noisy CIFAR-10.

ing graph convolutional layers. Besides, the class number of the dataset determines the models' output layer.

Objective Fitting: This module handles the process where models learn and fit datasets to accomplish target tasks. For different research domains, we select tailored hyperparameters, optimizers, loss functions, and so on. Unlike common machine learning, the training procedure in this module is guided by the curriculum learning module.

Curriculum Learning: This module integrates 15 core curriculum learning methods, all of which are abstracted as a class for easy plug-in into the objective fitting module. This design of abstracting methods as classes ensures that the module is extensible for new methods. Currently, we divide the existing methods into the following 3 categories. It is worth noting that this categorization is intended to facilitate the implementation and extension of various methods within a unified framework, but it does not imply that methods within the same category necessarily share similar properties or performance.

- **via Data Selection**: The primary approach to implementing curriculum is through data selection so that models can progressively learn from a subset to the entire dataset in a meaningful order. The methods belong to this category are vanilla SPL (Kumar et al., 2010), DIHCL (Zhou et al., 2020), and so on (Weinshall et al., 2018; Zhou & Bilmes, 2018; Cheng et al., 2019; Kong et al., 2021; Wang et al., 2023b). Some methods select data subsets based on sample difficulty, while others select data based on sample class.
- **via Model Adjustment**: An innovative idea for designing curricula is to regulate the amount of data information the model receives by modifying its architecture. CBS (Sinha et al., 2020), which employs a Gaussian filter to manage information intake, is a typical one.
- **via Loss Reweighting**: Loss reweighting can be regarded as a "soft" version of data selection. Intuitively, assigning a low weight to a data sample is almost equivalent to disregarding it. A common practice to reweight loss is through meta-learning (Finn et al., 2017), such as LRE (Ren et al., 2018a), MW-Net (Shu et al., 2019), and DDS (Wang et al., 2020), all of which employ a meta-network to assess the weights of losses and optimize the meta-network with the validation set. Additionally, there are other approaches, such as variants of SPL (Fan et al., 2017; Castells et al., 2020), DCL (Saxena et al., 2019), ScreenerNet (Kim & Choi, 2018), and SuperLoss (Castells et al., 2020).

Evaluation: This module is utilized to report results from 2 aspects, i.e., performance and complexity, in order to respectively demonstrate the effectiveness and efficiency of different methods. The performance metrics depend on the target datasets and tasks, and the complexity metrics include training time and maximum GPU memory consumption.

4.2. Example Usage

Figure 3 illustrates the python-like sample code of our CurBench toolkit, where an object of the SPLTrainer class is instantiated given the essential parameters, including a CIFAR-10 dataset name with the noise setting for data processing and a ResNet-18 net name for model loading. All of the above are put together to fit and evaluate the final result. With only a few lines of code, a dozen curriculum learning methods can be easily implemented and reproduced. On the basis of this tool, we conduct a multitude of experiments, and we will report the experimental setups and results in the next section.

5. Experiments and Analyses

5.1. Experimental Setup

To ensure a fair and reproducible evaluation, we fix all possible confounding factors and report the average and

```
from curbench.algorithms import SPLTrainer

# Instantiate curriculum learning class
trainer = SPLTrainer(
    # CIFAR-10 with 40% wrong labels
    data_name='cifar10-noise-0.4',
    # ResNet-18 with 32×32 input size
    net_name='resnet18',
    # Self-Paced Learning in a linear way
    start_rate=0.0,
    grow_epochs=100,
    grow_fn='linear',
    weight_fn='hard',
)
# Automatic, no need to specify:
# trainer._init_dataloader()
# trainer._init_model()

# Fitting and evaluating
trainer.fit()
trainer.evaluation()
```

Figure 3. Python-like sample code for an example of Self-Paced Learning applied to image classification with CurBench Toolkit.

standard deviation results of 5 runs with different fixed random seeds for each combination of datasets, backbone models, and settings. The detailed hyperparameters for both training processes and curriculum learning methods are presented in the Appendix.

5.2. Performance

5.2.1. Main Results

Table 3 presents the overall performances with and without curriculum learning under different combinations of backbone models, datasets, and settings. The detailed results of each specific curriculum learning method are attached in the Appendix, and we report the best ones among them in this table. The imbalance setting is not applied to NLP and graph datasets, where the number of samples in each class is imbalanced originally.

It is observed that curriculum learning can bring consistent improvement across domains. Compared to standard and imbalance settings, curriculum learning benefits much more in noise settings. This phenomenon is consistent with existing theoretical analysis, where curriculum learning is able to denoise and guide machine learning by discarding the difficult and possibly noisy data in the early stages of training. Besides, there is no such method that can outperform the others all the time, and the effectiveness of curriculum learning methods still depends on the target scenarios. For example, ScreenerNet (Kim & Choi, 2018) exhibits superior performance on CV datasets compared to graph datasets, and TTCL (Weinshall et al., 2018) performs better in noise settings than in standard and imbalance ones. Therefore, it is essential to explore more general methods while also researching methods tailored to specific environments.

5.2.2. Results in Noise Settings

Figure 4 demonstrates the performances of curriculum learning methods on datasets with different noise ratios $p \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. Without loss of generality, we select a backbone model and a dataset from each research domain. Some methods such as CBS, LGL, C2F, and EfficientTrain are only applied to CV datasets and not to NLP and graph datasets due to the following reasons. CBS (Sinha et al., 2020) requires convolutional layers in backbone models, and such models in CurBench are only within the CV domain. LGL (Cheng et al., 2019) and C2F (Stretcu et al., 2021) require multiple classes for clustering, but most NLP and graph datasets in CurBench have only two classes. EfficientTrain (Wang et al., 2023b) is based on data augmentation techniques on images.

We can observe that TTCL (Weinshall et al., 2018), the method by teacher transferring, obtains competitive performances regardless of the noise ratio, thanks to the guidance from the teacher model pretrained on the clean dataset. In contrast, SPL (Kumar et al., 2010), which is similar to TTCL but guides the learning by itself, performs relatively poorly. It is because a model not fully trained is not that competent to accurately distinguish noisy or hard data.

5.2.3. Results in Imbalance Settings

Figure 5 depicts the performances on CIFAR-10 with varying imbalance factor $r \in \{1, 10, 20, 50, 100, 200\}$.

It is observed that all methods achieve similar performances under different imbalance ratios. When the imbalance factor r increases, the differences between the methods become evident. Relatively speaking, the methods by data reweighting, such as DCL (Saxena et al., 2019) and SuperLoss (Castells et al., 2020), perform well because they can mitigate the impact of imbalanced classes by reassigning the weight of data or even class.

Compared with noise settings, curriculum learning brings less significant improvements and shows less variation between methods in imbalance settings. This is primarily because most curriculum learning methods focus on the difficulty of samples instead of classes, leading to overall better performances in noise settings than in imbalance settings. Additionally, the differences in judging difficult or noisy samples result in larger performance disparities among methods in noise settings.

	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50
LeNet	$69.95_{1.00}$	$65.02_{1.12}$	$44.93_{0.56}$	$35.46_{0.70}$	$29.59_{0.40}$	$19.57_{0.64}$	$22.08_{0.61}$	$18.63_{0.43}$	$11.65_{0.30}$
LeNet + CL	$\mathbf{70.43}_{0.41}$	$\mathbf{65.93}_{0.57}$	$\mathbf{45.28}_{0.56}$	$\mathbf{35.63}_{0.78}$	$\mathbf{30.87}_{0.48}$	$\mathbf{19.74}_{0.17}$	$\mathbf{22.83}_{0.44}$	$\mathbf{19.91}_{0.26}$	$\mathbf{12.36}_{0.47}$
ResNet-18	$92.33_{0.16}$	$82.75_{2.06}$	$75.49_{0.87}$	$69.97_{0.27}$	$52.14_{0.39}$	$42.57_{0.68}$	$51.41_{1.74}$	$39.42_{0.21}$	$28.83_{0.38}$
ResNet-18 + CL	$\mathbf{92.88}_{0.23}$	$\mathbf{86.92}_{0.20}$	$\mathbf{76.43}_{0.96}$	$\mathbf{71.31}_{0.14}$	$\mathbf{58.56}_{0.60}$	$\mathbf{43.47}_{0.43}$	$\mathbf{53.61}_{0.48}$	$\mathbf{43.64}_{0.72}$	$\mathbf{30.82}_{0.36}$
ViT	$79.90_{0.38}$	$64.19_{0.51}$	$52.12_{0.81}$	$51.05_{0.62}$	$35.25_{0.24}$	$26.05_{0.52}$	$38.16_{0.53}$	$24.90_{0.26}$	$17.15_{0.31}$
ViT + CL	$\mathbf{80.66}_{0.27}$	$\mathbf{69.83}_{0.53}$	$\mathbf{52.85}_{0.81}$	$\mathbf{51.93}_{0.64}$	$\mathbf{39.15}_{0.30}$	$\mathbf{26.40}_{0.34}$	$\mathbf{38.92}_{0.53}$	$\mathbf{29.76}_{0.34}$	$\mathbf{17.47}_{0.14}$

	RTE		MRPC		STS-B		CoLA	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
LSTM	$52.95_{1.34}$	$53.43_{1.77}$	$81.43_{0.14}$	$81.22_{0.00}$	$12.73_{0.72}$	$10.90_{1.19}$	$11.29_{1.27}$	$3.27_{1.68}$
LSTM + CL	$\mathbf{53.07}_{1.29}$	$\mathbf{54.22}_{1.77}$	$\mathbf{81.54}_{0.18}$	$\mathbf{81.24}_{0.05}$	$\mathbf{14.11}_{2.21}$	$\mathbf{11.75}_{1.61}$	$\mathbf{12.65}_{1.21}$	$\mathbf{8.55}_{2.10}$
BERT	$64.62_{3.33}$	$54.22_{3.14}$	$88.54_{0.45}$	$81.89_{0.83}$	$85.26_{0.22}$	$80.71_{1.01}$	$57.39_{1.30}$	$32.35_{0.79}$
BERT + CL	$\mathbf{66.35}_{1.76}$	$\mathbf{56.32}_{5.04}$	$\mathbf{88.69}_{1.24}$	$\mathbf{81.94}_{0.55}$	$\mathbf{85.42}_{0.22}$	$\mathbf{81.31}_{0.25}$	$\mathbf{57.80}_{1.96}$	$\mathbf{45.79}_{1.64}$
GPT2	$65.34_{1.95}$	$52.92_{4.49}$	$85.49_{0.86}$	$78.23_{1.72}$	$76.44_{1.20}$	$69.65_{1.85}$	$37.00_{3.72}$	$5.86_{1.69}$
GPT2 + CL	$\mathbf{66.35}_{2.10}$	$\mathbf{57.40}_{3.39}$	$\mathbf{86.29}_{0.36}$	$\mathbf{82.55}_{0.88}$	$\mathbf{80.82}_{1.39}$	$\mathbf{71.57}_{1.74}$	$\mathbf{39.95}_{3.16}$	$\mathbf{12.54}_{2.75}$

	SST-2		QNLI		QQP		MNLI-(m/mm)	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
LSTM	$81.67_{0.85}$	$64.36_{1.12}$	$50.54_{0.00}$	$50.62_{0.16}$	$75.69_{0.27}$	$60.72_{0.79}$	$61.38_{0.30}$ / $61.21_{0.45}$	$44.41_{0.51}$ / $44.83_{0.90}$
LSTM + CL	$\mathbf{82.87}_{0.88}$	$\mathbf{78.58}_{1.64}$	$\mathbf{51.02}_{0.46}$	$\mathbf{50.83}_{0.45}$	$\mathbf{75.73}_{0.21}$	$\mathbf{66.47}_{0.72}$	$\mathbf{62.47}_{0.36}$ / $\mathbf{62.33}_{0.42}$	$\mathbf{58.59}_{0.54}$ / $\mathbf{58.50}_{0.64}$
BERT	$92.66_{0.28}$	$87.22_{0.82}$	$91.21_{0.24}$	$81.21_{0.76}$	$88.05_{0.12}$	$76.23_{0.48}$	$83.89_{0.31}$ / $84.38_{0.29}$	$78.65_{0.70}$ / $79.21_{0.62}$
BERT + CL	$\mathbf{92.82}_{0.16}$	$\mathbf{91.25}_{0.59}$	$\mathbf{91.49}_{0.13}$	$\mathbf{89.45}_{0.44}$	$\mathbf{88.16}_{0.13}$	$\mathbf{84.50}_{0.25}$	$\mathbf{84.27}_{0.07}$ / $\mathbf{84.40}_{0.42}$	$\mathbf{81.73}_{0.31}$ / $\mathbf{82.25}_{0.40}$
GPT2	$91.95_{0.49}$	$85.83_{0.57}$	$87.92_{0.31}$	$78.72_{0.37}$	$86.00_{0.23}$	$75.40_{0.84}$	$81.53_{0.21}$ / $82.40_{0.21}$	$76.56_{0.15}$ / $77.69_{0.15}$
GPT2 + CL	$\mathbf{92.25}_{0.42}$	$\mathbf{90.34}_{0.53}$	$\mathbf{88.17}_{0.67}$	$\mathbf{84.00}_{0.70}$	$\mathbf{86.68}_{0.16}$	$\mathbf{82.16}_{0.35}$	$\mathbf{81.90}_{0.23}$ / $\mathbf{82.59}_{0.35}$	$\mathbf{78.36}_{0.19}$ / $\mathbf{79.62}_{0.44}$

	MUTAG		PROTEINS		NCI1		ogbg-molhiv	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
GCN	$73.68_{2.11}$	$66.31_{7.14}$	$70.71_{4.20}$	$63.57_{6.45}$	$69.59_{1.23}$	$55.23_{3.21}$	$75.84_{1.02}$	$64.29_{4.55}$
GCN + CL	$\mathbf{74.74}_{3.94}$	$\mathbf{71.58}_{5.37}$	$\mathbf{73.21}_{4.41}$	$\mathbf{71.61}_{6.62}$	$\mathbf{71.39}_{1.29}$	$\mathbf{67.98}_{2.01}$	$\mathbf{77.41}_{1.15}$	$\mathbf{72.81}_{1.14}$
GAT	$69.47_{6.14}$	$65.26_{5.37}$	$64.46_{2.96}$	$65.71_{9.13}$	$56.74_{2.86}$	$53.77_{2.12}$	$68.07_{2.34}$	$65.37_{2.66}$
GAT + CL	$\mathbf{72.63}_{8.42}$	$\mathbf{69.47}_{10.21}$	$\mathbf{69.82}_{7.13}$	$\mathbf{69.11}_{3.77}$	$\mathbf{59.37}_{1.59}$	$\mathbf{55.67}_{4.70}$	$\mathbf{72.64}_{1.16}$	$\mathbf{66.73}_{1.84}$
GIN	$86.84_{7.90}$	$78.95_{3.72}$	$74.11_{4.24}$	$69.82_{1.73}$	$79.32_{1.40}$	$60.24_{3.92}$	$74.72_{1.36}$	$63.07_{3.73}$
GIN + CL	$\mathbf{88.42}_{2.10}$	$\mathbf{81.58}_{4.56}$	$\mathbf{77.14}_{4.88}$	$\mathbf{73.93}_{1.82}$	$\mathbf{82.04}_{1.90}$	$\mathbf{62.14}_{6.47}$	$\mathbf{76.53}_{1.97}$	$\mathbf{65.53}_{1.61}$

Table 3. The empirical performances of 9 backbone models over 15 datasets in 3 settings with and without curriculum learning methods. The rows with "+ CL" present the best performances achieved among the methods involved in this benchmark. The bold font highlights the superior performances brought by curriculum learning. The imbalance setting is not applied to NLP and graph datasets, which are imbalanced originally. **Note:** The detailed performances of each method are reported in Table 9-11 in the Appendix.

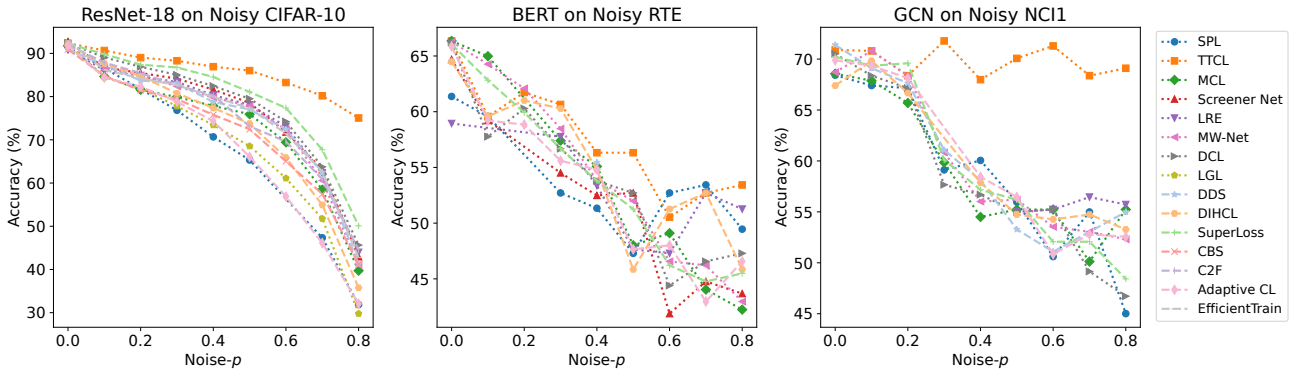


Figure 4. The performances as a function of noise ratio p for different curriculum learning methods on datasets from 3 research domains.

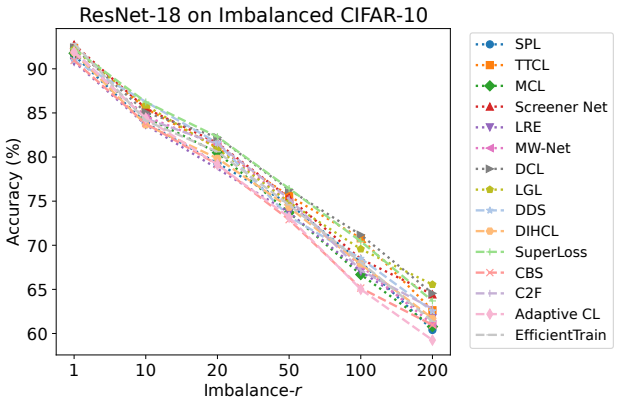


Figure 5. The performances as a function of imbalance factor r.

5.3. Complexity

Figure 6 shows the time and space complexity of each method in the case of ResNet-18 and CIFAR-10, measured by GPU training time (Hour) and maximum GPU memory consumption (GB).

The whole figure can be divided into 3 parts. The first is the upper right corner, which contains the methods requiring gradient calculation and meta-network training, resulting in high time and space complexity. The second is the middle part with the point of ScreenerNet, which also introduces an extra network but only requires once backward, leading to less complexity. The third is the lower left corner, which includes most of the methods consuming similarly small amounts of training time and GPU memory because they measure data difficulty and schedule curriculum in a relatively intuitive way and do not demand a learnable network with a large number of parameters.

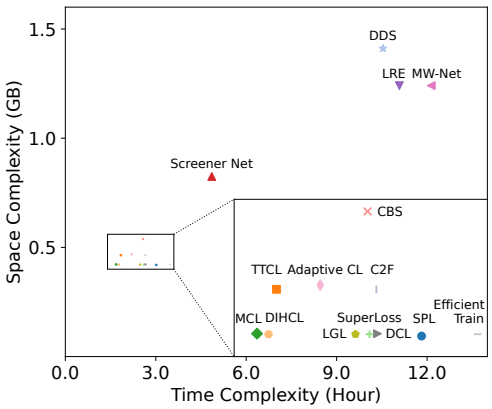


Figure 6. Time and space complexity of different methods in the case of ResNet-18 and CIFAR-10. **Note:** The numerical results of 3 different cases are reported in Table 8 in the Appendix.

6. Conclusion

In this paper, we propose CurBench, the first benchmark for curriculum learning. It covers a broad range of research domains, datasets, backbone models, settings, and evaluation dimensions, ensuring a fair, reliable, and systematic evaluation of existing curriculum learning methods. For convenient utilization, it is complemented by a toolkit that implements essential related works in a unified pipeline and applies them to various machine learning tasks. Through empirical results and theoretical analyses, we provide valuable findings on curriculum learning. In conclusion, CurBench holds the potential to benefit future research and suggest promising directions.

Limitations: Despite the benefits of our CurBench, we also recognize the following limitations in this version and intend to refine them in future expansions.

- CurBench mainly covers supervised learning in CV, NLP, and graph domains, but has not incorporated the datasets, backbone models, and tasks related to other domains such as audio processing, multimodal learning, recommender systems, and robotics. Additionally, CurBench has not involved unsupervised, semi-supervised, and reinforcement learning. Given the importance of these topics in the context of curriculum learning applications, they will be integrated as a significant part of future versions.

- CurBench currently employs publicly available datasets that are commonly used in their respective domains. However, CurBench has not yet introduced any new datasets. Designing specialized datasets for curriculum learning is essential because these datasets can better align with the unique requirements and objectives of curriculum learning methodologies. We recognize the importance of this task and intend to undertake it in the future.

- CurBench has not evaluated the performance of curriculum learning on large models, which deserves in-depth exploration in this era of large models. Considering that large models often encounter vast amounts of data with varying quality when learning, it is suitable to utilize curriculum learning for guidance and denoising. We plan to include the prevalent large-scale language and multimodal models in our future work.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work is supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China (No. 62222209, 62250008, 62102222), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023RC01003, BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Castells, T., Weinzaepfel, P., and Revaud, J. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems*, 33:4308–4319, 2020.

Chen, H., Chen, Y., Wang, X., Xie, R., Wang, R., Xia, F., and Zhu, W. Curriculum disentangled recommendation with noisy multi-feedback. *Advances in Neural Information Processing Systems*, 34:26924–26936, 2021a.

Chen, H., Wang, X., Lan, X., Chen, H., Duan, X., Jia, J., and Zhu, W. Curriculum-listener: Consistency-and complementarity-aware audio-enhanced temporal sentence grounding. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3117–3128, 2023.

Chen, Y., Wang, X., Fan, M., Huang, J., Yang, S., and Zhu, W. Curriculum meta-learning for next poi recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2692–2702, 2021b.

Cheng, H., Lian, D., Deng, B., Gao, S., Tan, T., and Geng, Y. Local to global learning: Gradually adding classes for training deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4748–4756, 2019.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Fan, Y., He, R., Liang, J., and Hu, B. Self-paced learning: An implicit regularization perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pp. 482–495. PMLR, 2017.

Gong, T., Zhao, Q., Meng, D., and Xu, Z. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data & Information Analytics*, 1(1):111, 2016.

Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. Automated curriculum learning for neural networks. In *international conference on machine learning*, pp. 1311–1320. PMLR, 2017.

Gui, L., Baltrušaitis, T., and Morency, L.-P. Curriculum learning for facial expression recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 505–511. IEEE, 2017.

Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M. R., and Huang, D. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 135–150, 2018.

Hacohen, G. and Weinshall, D. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pp. 2535–2544. PMLR, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Kim, T.-H. and Choi, J. Screenernet: Learning self-paced curriculum for deep neural networks. *arXiv preprint arXiv:1801.00904*, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kong, Y., Liu, L., Wang, J., and Tao, D. Adaptive curriculum learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5067–5076, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kumar, M., Packer, B., and Koller, D. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Lan, X., Yuan, Y., Chen, H., Wang, X., Jie, Z., Ma, L., Wang, Z., and Zhu, W. Curriculum multi-negative augmentation for debiased video grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1213–1221, 2023.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Wang, X., and Zhu, W. Curriculum graph machine learning: A survey. *arXiv preprint arXiv:2302.02926*, 2023.

Liu, C., He, S., Liu, K., Zhao, J., et al. Curriculum learning for natural answer generation. In *IJCAI*, pp. 4223–4229, 2018.

Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740, 2019.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Narvekar, S., Sinapov, J., and Stone, P. Autonomous task sequencing for customized curriculum design in reinforcement learning. In *IJCAI*, pp. 2536–2542, 2017.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. Curriculum learning for reinforcement learning domains: A framework and survey. *The Journal of Machine Learning Research*, 21(1):7382–7431, 2020.

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., and Mitchell, T. M. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019.

Portelas, R., Colas, C., Weng, L., Hofmann, K., and Oudeyer, P.-Y. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

Qin, Y., Wang, X., Zhang, Z., Chen, H., and Zhu, W. Multi-task graph neural architecture search with task-aware collaboration and curriculum. *Advances in neural information processing systems*, 36, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018a.

Ren, Z., Dong, D., Li, H., and Chen, C. Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning. *IEEE transactions on neural networks and learning systems*, 29(6):2216–2226, 2018b.

Saxena, S., Tuzel, O., and DeCoste, D. Data parameters: A new family of parameters for learning a differentiable curriculum. *Advances in Neural Information Processing Systems*, 32, 2019.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Sinha, S., Garg, A., and Larochelle, H. Curriculum by smoothing. *Advances in Neural Information Processing Systems*, 33:21653–21664, 2020.

Soviany, P., Ardei, C., Ionescu, R. T., and Leordeanu, M. Image difficulty curriculum for generative adversarial networks (cugan). In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3463–3472, 2020.

Soviany, P., Ionescu, R. T., Rota, P., and Sebe, N. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565, 2022.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 751–759, 2010.

Stretcu, O., Platanios, E. A., Mitchell, T. M., and Póczos, B. Coarse-to-fine curriculum learning. *arXiv preprint arXiv:2106.04072*, 2021.

Tay, Y., Wang, S., Tuan, L. A., Fu, J., Phan, M. C., Yuan, X., Rao, J., Hui, S. C., and Zhang, A. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. *arXiv preprint arXiv:1905.10847*, 2019.

Tsvetkov, Y., Faruqui, M., Ling, W., MacWhinney, B., and Dyer, C. Learning the curriculum with bayesian optimization for task-specific word representation learning. *arXiv preprint arXiv:1605.03852*, 2016.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Wang, X., Pham, H., Michel, P., Anastasopoulos, A., Carbonell, J., and Neubig, G. Optimizing data usage via differentiable rewards. In *International Conference on Machine Learning*, pp. 9983–9995. PMLR, 2020.

Wang, X., Chen, Y., and Zhu, W. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021a.

Wang, X., Pan, Z., Zhou, Y., Chen, H., Ge, C., and Zhu, W. Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *International Conference on Machine Learning*, pp. 36174–36192. PMLR, 2023a.

Wang, Y., Wang, W., Liang, Y., Cai, Y., and Hooi, B. Curgraph: Curriculum learning for graph classification. In *Proceedings of the Web Conference 2021*, pp. 1238–1248, 2021b.

Wang, Y., Yue, Y., Lu, R., Liu, T., Zhong, Z., Song, S., and Huang, G. Efficienttrain: Exploring generalized curriculum learning for training visual backbones. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5852–5864, 2023b.

Wei, X., Gong, X., Zhan, Y., Du, B., Luo, Y., and Hu, W. Clnode: Curriculum learning for node classification. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 670–678, 2023.

Weinshall, D., Cohen, G., and Amir, D. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*, pp. 5238–5246. PMLR, 2018.

Wu, Z., Wang, X., Chen, H., Li, K., Han, Y., Sun, L., and Zhu, W. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 9329–9335, 2023.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Yang, Z., Xu, Q., Bao, S., Cao, X., and Huang, Q. Learning with multiclass auc: Theory and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (11):7747–7763, 2021.

Yao, Y., Wang, X., Qin, Y., Zhang, Z., Zhu, W., and Mei, H. Data-augmented curriculum graph neural architecture search under distribution shifts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16433–16441, Mar. 2024.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ArXiv*, abs/1611.03530, 2016. URL https://api.semanticscholar.org/CorpusID:6212000.

Zhang, D., Tian, H., and Han, J. Few-cost salient object detection with adversarial-paced learning. *Advances in Neural Information Processing Systems*, 33:12236–12247, 2020.

Zhang, Z., Zhang, Z., Wang, X., and Zhu, W. Learning to solve travelling salesman problem with hardness-adaptive curriculum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9136–9144, 2022.

Zhao, M., Wu, H., Niu, D., and Wang, X. Reinforced curriculum learning on pre-trained neural machine translation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 9652–9659, 2020.

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

Zhou, T., Wang, S., and Bilmes, J. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613, 2020.

Zhou, Y., Chen, H., Pan, Z., Yan, C., Lin, F., Wang, X., and Zhu, W. Curml: A curriculum machine learning library. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 7359–7363, 2022a.

Zhou, Y., Wang, X., Chen, H., Duan, X., Guan, C., and Zhu, W. Curriculum-nas: Curriculum weight-sharing neural architecture search. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 6792–6801, 2022b.

Zhou, Y., Wang, X., Chen, H., Duan, X., and Zhu, W. Intra- and inter-modal curriculum for multimodal learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3724–3735, 2023.

A. Appendix Abstract

In this appendix, we first list the essential information of the datasets in Section B and backbone models in Section C. Then we summarize the curriculum methods implemented in this work in Section D to present how these methods were evaluated when they were proposed. After providing the training hyperparameters in Section E and method hyperparameters in Section F, we report the detailed performance and complexity of each method in various experimental setups in Section G.

B. Datasets

All the datasets included in CurBench are publicly available for research. To eliminate the risk of ethical or license issues, we list the essential information of the datasets, such as their home pages, common download links, and licenses.

Domain	Home Page	Download Link	License
CV	CIFAR	PyTorch	MIT
	Tiny-ImageNet	CS231n	MIT
NLP	GLUE	Hugging Face	Various
Graph	TUDataset	PyTorch Geometric	Various
	OGB	OGB Dataset	MIT

Table 4. The home pages, download links, and licenses of datasets.

Concretely, in this work, we download CIFAR via PyTorch API, GLUE via Hugging Face API, TUDataset via PyTorch Geometric (PyG) API, and OGB dataset via OGB API. For Tiny-ImageNet, we download the zip file from CS231n, and adjust its file structure to the same form as CIFAR for easier loading with the help of the tool code from Github: lromor/tinyimagenet.py.

C. Backbone Models

For the standardization and reliability of CurBench, we implement all backbone models by referencing highly recognized code repositories as shown in Table 5.

Domain	Model	Reference
CV	LeNet, ResNet-18	pytorch-cifar
	ViT	vit-pytorch
NLP	LSTM	lstm-gru-pytorch
	BERT, GPT2	Hugging Face
Graph	GCN, GAT, GIN	PyTorch Geometric

Table 5. The implementation references of backbone models.

Among these models, BERT and GPT2 are initiated with the pretrained parameters from Hugging Face and finetuned in this work, while others are trained from scratch.

D. Curriculum Learning Methods

When designing CurBench, we are inclined to the datasets and models used in previous works for evaluation. Therefore, we have surveyed what datasets and models are commonly employed and completed the Table 6.

It can be obviously found that when researchers propose a curriculum learning method, they always conduct experiments on image classification tasks for performance evaluation. Only a few authors will try to apply their methods to the datasets for object detection or neural machine translation. Besides, not all works take different settings, such as noise or imbalance, into consideration.

Therefore, as stated in the main text, we not only select the datasets and models in the CV domain, which are commonly used in previous related works, but also supplement those in the NLP and graph domains to investigate how the methods can adapt to various scenarios.

E. Training Hyperparameters

To ensure a fair evaluation, we run 5 times with fixed different random seeds $s \in \{42, 666, 777, 888, 999\}$, and report the average and standard deviation results. Besides, we strictly set the training hyperparameters as follows:

LeNet, ResNet-18, ViT: We choose a batch size of 50, and use an Adam optimizer to train the model with a constant learning rate of 0.0001 for 200 epochs.

LSTM: We choose a batch size of 50, and use a SGD optimizer to train the model with a cosine annealing learning rate of 0.00001$\sim$1 for 10 epochs.

BERT, GPT2: We choose a batch size of 50, and use an AdamW optimizer to train the model with a constant learning rate of 0.00002 for 3 epochs.

GCN, GAT, GIN: We choose a batch size of 50, and use an Adam optimizer to train the model for 200 epochs with learning rates of 0.01 for TUDataset and 0.001 for OGB.

F. Method Hyperparameters

For a reproducible evaluation, we demonstrate the hyperparameters that we select for curriculum learning methods in Table 7. It should be noted that this table includes the hyperparameters for the experiments with 200 epochs. For text domain tasks trained for 3 or 10 epochs, we sightly adjust some epoch-related hyperparameters to adapt the tasks, such as *grow_epochs*, *warm_epochs*, and *schedule_epochs*.

G. Detailed Complexity and Performance

Tables from 8 to 11 report complexity and performance.

Method	Conference	Datasets	Models	Settings		
				Std	Noi	Imb
SPL (Kumar et al., 2010)	NIPS, 2010	MUC6, UniProbe, MNIST, Mammals	SSVM	✓		
TTCL (Weinshall et al., 2018)	ICML, 2018	CIFAR-100, STL-10	CNN	✓		
MCL (Zhou & Bilmes, 2018)	ICLR, 2018	News-20, MNIST, CIFAR-10, STL-10, SVHN, Fashion	LeNet5, CNN	✓		
ScreenerNet (Kim & Choi, 2018)	ArXiv, 2018	Cart-pole-v0, CIFAR-10, MNIST, Pascal VOC	DDQN, CNN	✓		
LRE (Ren et al., 2018a)	ICML, 2018	MNIST, CIFAR-10, CIFAR-100	LeNet, ResNet-32, WideResNet-28-10		✓	✓
MW-Net (Shu et al., 2019)	NIPS, 2019	CIFAR-10, CIFAR-100, Clothing1M	ResNet-32, ResNet-50, WideResNet-28-10	✓	✓	✓
DCL (Saxena et al., 2019)	NIPS, 2019	CIFAR-10, CIFAR-100, ImageNet, WebVision, KITTI	VGG-16, SSDNet, ResNet-18, WideResNet-28-10	✓	✓	
LGL (Cheng et al., 2019)	CVPR, 2019	CIFAR-10, CIFAR-100, ImageNet	VGG-16, ResNet-50	✓		
DDS (Wang et al., 2020)	ICML, 2020	CIFAR-10, ImageNet, TED	LSTM, ResNet-50, WideResNet-28-10	✓		✓
DIHCL (Zhou et al., 2020)	NIPS, 2020	CIFAR-10, CIFAR-100, ImageNet, Food-101, FGVC Aircraft, Stanford Cars, Birdsnap, FMNIST, KMNIST, STL10, SVHN	ResNet-50, WideResNet-16-8, WideResNet-28-10, ResNeXt50-32x4d, PreActResNet34	✓		
SuperLoss (Castells et al., 2020)	NIPS, 2020	MNIST, UTKFace, CIFAR-10, CIFAR-100, WebVision, Pascal VOC, Revisited Oxford and Paris	ResNet-18, ResNet-50, ResNet-101, WideResNet-28-10, Faster R-CNN, RetinaNet	✓	✓	
CBS (Sinha et al., 2020)	NIPS, 2020	CIFAR-10, CIFAR-100, ImageNet, SVHN, CelebA, Pascal VOC, MNIST, USPS	VGG-16, ResNet-18, Wide-ResNet-50, ResNeXt-50, VAE, β-VAE	✓		
C2F (Stretcu et al., 2021)	ArXiv, 2021	CIFAR-10, CIFAR-100, Shapes, Tiny-ImageNet	Resnet-18, Resnet-50, WideResnet-28-10	✓		
Adaptive CL (Kong et al., 2021)	ICCV, 2021	CIFAR-10, CIFAR-100, Subset of ImageNet	MLP, HNN, VGG-16, ResNet-18 ResNet-v1-14	✓		
EfficientTrain (Wang et al., 2023b)	ICCV, 2023	ImageNet-1K/22K, MS COCO, Flowers-102, CIFAR, Stanford Dogs	ResNet, ConvNeXt, DeiT, PVT, Swin, CSWin	✓		

Table 6. Summary of the methods reproduced in CurBench, where we overview the datasets and models involved in the related works. "Std" stands for the standard setting, "Noi" for noise, and "Imb" for imbalance.

Method	Hyerparameter	Value
SPL	start_ratio	0.0
	grow_epochs	100
	grow_fn	linear
	weight_fn	hard
TTCL	start_ratio	0.0
	grow_epochs	100
	grow_fn	linear
	weight_fn	hard
MCL	schedule_epochs	20
	warm_epochs	5
	lam	1
	minlam	0.2
	gamma	0.1
	fe_alpha	2
	fe_beta	0.75
	fe_gamma	0.9
	fe_lambda	0.9
ScreenerNet	M	1.0
LRE	meta_split	0.1
MW-Net	meta_split	0.1
	VNet	[1, 100, 1]
DCL	init_class_param	0.0
	lr_class_param	0.1
	wd_class_param	0.0
	init_data_param	1.0
	lr_data_param	0.1
	wd_data_param	0.0
LGL	start_ratio	0.1
	grow_ratio	0.3
	grow_interval	20
	strategy	random
DDS	meta_split	0.1
	eps	0.001
DIHCL	warm_epochs	50
	discount_factor	0.9
	decay_rate	0.9
	bottom_size	0.5
	type	loss
	sample_type	random
SuperLoss	tau	0.0
	lam	1.0
	fac	0.9
CBS	kernel_size	3
	start_std	1.0
	grow_factor	0.9
	grow_interval	5
C2F	cluster_K	3
	pace_p	0.1
	pace_q	2.5
Adaptive CL	pace_r	15
	inv	20
	alpha	0.7
	gamma	0.1
	bottom_gamma	0.1
EfficientTrain	epochs	{120, 160, 200}
	crop_size	{160, 192, 224}
	rand_aug	0→9

Table 7. The default hyperparameters we set for each method when the number of training epochs is 200.

	Training Time (Minute)	GPU Memory (MB)
SPL	175	420
TTCL	111	464
MCL	106	422
ScreenerNet	291	825
LRE	665	1241
MW-Net	728	1241
DCL	158	422
LGL	149	421
DDS	632	1411
DIHCL	107	421
SuperLoss	156	421
CBS	155	538
C2F	159	464
Adaptive CL	132	468
EfficientTrain	214	421

(a) ResNet-18 on CIFAR-10

	Training Time (Minute)	GPU Memory (MB)
SPL	1.28	6615
TTCL	1.12	7036
MCL	2.12	6615
ScreenerNet	2.05	13114
LRE	3.27	22989
MW-Net	4.23	22989
DCL	1.18	6615
DDS	4.02	23997
DIHCL	1.05	6615
SuperLoss	1.10	6615
Adaptive CL	0.68	7036

(b) BERT on RTE

	Training Time (Minute)	GPU Memory (MB)
SPL	4.75	6.12
TTCL	3.50	5.76
MCL	3.03	5.82
ScreenerNet	6.25	7.46
LRE	7.77	105.41
MW-Net	8.65	24.79
DCL	3.87	5.79
DDS	11.62	20.50
DIHCL	2.12	5.71
SuperLoss	3.90	5.76
AdaptiveCL	3.53	5.46

(c) GCN on NCI1

Table 8. Time and space complexity, measured by training time and GPU memory usage on NVIDIA V100 GPU.

	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50
SPL	$69.08_{0.78}$	$63.68_{1.01}$	$42.34_{0.90}$	$34.70_{0.72}$	$26.09_{0.69}$	$18.15_{0.68}$	$21.53_{0.25}$	$15.59_{0.63}$	$10.17_{0.12}$
TTCL	$68.87_{0.69}$	$64.63_{1.00}$	$44.03_{0.54}$	$34.19_{0.88}$	$28.83_{0.96}$	$18.39_{0.42}$	$22.08_{0.48}$	$18.84_{0.18}$	$11.17_{0.34}$
MCL	$65.86_{0.31}$	$62.50_{1.01}$	$34.59_{0.90}$	$32.60_{0.75}$	$27.09_{0.34}$	$15.90_{0.31}$	$20.99_{0.37}$	$17.06_{0.37}$	$9.82_{0.35}$
ScreenerNet	$\mathbf{70.43}_{0.41}$	$65.45_{0.92}$	$\mathbf{45.28}_{0.56}$	$\mathbf{35.63}_{0.78}$	$29.72_{0.69}$	$\mathbf{19.74}_{0.17}$	$\mathbf{22.83}_{0.44}$	$18.54_{0.29}$	$11.77_{0.19}$
LRE	$64.52_{0.86}$	$59.88_{0.49}$	$36.24_{2.17}$	$29.29_{0.73}$	$23.37_{0.34}$	$14.52_{0.19}$	$18.86_{0.66}$	$14.97_{0.21}$	$8.23_{0.13}$
MW-Net	$69.13_{0.44}$	$63.92_{0.98}$	$45.17_{0.82}$	$35.40_{0.54}$	$28.09_{0.66}$	$18.95_{0.32}$	$22.16_{0.36}$	$17.88_{0.25}$	$10.97_{0.30}$
DCL	$67.23_{0.49}$	$64.77_{0.59}$	$39.16_{0.87}$	$34.09_{0.51}$	$30.02_{0.82}$	$18.13_{0.42}$	$22.01_{0.55}$	$19.65_{0.20}$	$10.95_{0.20}$
LGL	$69.87_{0.71}$	$65.09_{0.78}$	$44.94_{1.25}$	$35.04_{0.84}$	$29.56_{0.54}$	$19.28_{0.64}$	$22.55_{0.30}$	$18.40_{0.05}$	$11.25_{0.43}$
DDS	$65.65_{2.84}$	$63.45_{1.84}$	$41.51_{4.52}$	$35.11_{1.04}$	$28.49_{0.47}$	$19.05_{0.40}$	$22.29_{0.41}$	$17.03_{1.19}$	$10.30_{1.3}$
DIHCL	$66.46_{0.83}$	$58.42_{0.73}$	$40.89_{1.21}$	$28.49_{0.59}$	$27.87_{0.23}$	$15.77_{0.62}$	$17.72_{0.50}$	$14.74_{0.43}$	$8.16_{0.33}$
SuperLoss	$70.29_{0.68}$	$\mathbf{65.93}_{0.57}$	$43.13_{0.51}$	$34.91_{0.68}$	$\mathbf{30.87}_{0.48}$	$18.57_{0.17}$	$22.27_{0.29}$	$\mathbf{19.91}_{0.26}$	$11.23_{0.31}$
CBS	$69.79_{0.36}$	$63.47_{0.96}$	$44.60_{1.77}$	$35.17_{0.63}$	$28.14_{0.74}$	$18.87_{0.60}$	$21.87_{0.58}$	$17.78_{0.58}$	$11.10_{0.43}$
C2F	$69.49_{0.37}$	$64.35_{0.79}$	$43.74_{0.86}$	$35.51_{0.40}$	$29.92_{0.58}$	$19.24_{0.52}$	$22.44_{0.22}$	$18.78_{0.23}$	$11.69_{0.32}$
Adaptive CL	$69.25_{0.43}$	$63.93_{0.97}$	$42.87_{0.47}$	$34.58_{0.51}$	$28.36_{0.43}$	$18.59_{0.26}$	$22.62_{0.30}$	$18.09_{0.37}$	$10.98_{0.20}$
EfficientTrain	$70.34_{0.44}$	$62.96_{0.84}$	$43.92_{1.01}$	$35.59_{0.66}$	$28.04_{0.71}$	$18.78_{0.62}$	$22.31_{0.42}$	$18.05_{0.17}$	$\mathbf{12.36}_{0.47}$

(a) LeNet

	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50
SPL	$91.54_{0.26}$	$70.68_{2.25}$	$74.71_{0.74}$	$68.13_{0.47}$	$34.09_{0.39}$	$39.80_{0.93}$	$48.99_{0.41}$	$22.49_{0.41}$	$26.04_{0.93}$
TTCL	$92.35_{0.13}$	$\mathbf{86.92}_{0.20}$	$75.59_{0.56}$	$67.52_{0.46}$	$\mathbf{58.56}_{0.60}$	$38.40_{0.97}$	$48.50_{0.34}$	$41.81_{0.67}$	$25.32_{0.46}$
MCL	$91.76_{0.15}$	$77.84_{0.33}$	$73.71_{0.85}$	$68.68_{0.37}$	$45.95_{0.58}$	$40.49_{0.67}$	$51.46_{0.16}$	$34.39_{0.66}$	$28.08_{0.28}$
ScreenerNet	$92.74_{0.20}$	$81.63_{0.70}$	$75.37_{0.56}$	$\mathbf{71.31}_{0.14}$	$51.96_{0.56}$	$\mathbf{43.47}_{0.43}$	$\mathbf{53.61}_{0.48}$	$39.22_{0.57}$	$\mathbf{30.82}_{0.36}$
LRE	$90.80_{0.22}$	$80.35_{0.50}$	$73.71_{0.36}$	$66.99_{0.24}$	$50.31_{0.88}$	$40.69_{0.69}$	$49.86_{0.37}$	$36.40_{0.33}$	$27.49_{0.41}$
MW-Net	$91.79_{0.26}$	$79.77_{0.44}$	$74.86_{0.59}$	$69.09_{0.25}$	$49.87_{0.32}$	$40.99_{0.48}$	$50.93_{0.36}$	$37.79_{0.43}$	$27.96_{0.55}$
DCL	$92.41_{0.25}$	$82.44_{0.66}$	$76.30_{0.88}$	$69.80_{0.47}$	$54.01_{0.57}$	$42.31_{0.41}$	$52.25_{0.43}$	$40.67_{0.42}$	$28.83_{0.63}$
LGL	$92.19_{0.20}$	$73.42_{0.41}$	$74.87_{0.40}$	$69.08_{0.15}$	$39.93_{0.58}$	$41.07_{0.31}$	$50.32_{0.38}$	$27.35_{0.32}$	$27.21_{0.21}$
DDS	$90.94_{2.26}$	$78.74_{3.07}$	$70.24_{8.53}$	$68.87_{0.17}$	$46.87_{1.72}$	$37.93_{2.71}$	$50.84_{0.30}$	$37.54_{0.37}$	$26.96_{1.29}$
DIHCL	$91.87_{0.21}$	$77.38_{0.42}$	$74.31_{0.60}$	$67.36_{0.33}$	$44.19_{0.37}$	$39.51_{0.75}$	$50.59_{0.32}$	$32.70_{0.38}$	$26.36_{0.34}$
SuperLoss	$92.27_{0.22}$	$84.54_{0.40}$	$\mathbf{76.43}_{0.96}$	$69.53_{0.43}$	$57.51_{0.45}$	$42.43_{1.01}$	$52.38_{0.53}$	$\mathbf{43.64}_{0.72}$	$28.85_{0.38}$
CBS	$90.94_{0.27}$	$75.79_{0.79}$	$72.90_{0.66}$	$63.67_{0.37}$	$41.14_{0.39}$	$36.19_{0.91}$	$45.67_{0.25}$	$30.42_{0.53}$	$24.19_{0.34}$
C2F	$91.98_{0.17}$	$80.27_{0.52}$	$75.26_{1.16}$	$69.86_{0.17}$	$50.48_{1.32}$	$42.47_{0.79}$	$51.96_{0.45}$	$38.04_{0.43}$	$28.90_{0.39}$
Adaptive CL	$91.91_{0.08}$	$74.30_{0.79}$	$73.18_{1.37}$	$66.04_{0.41}$	$38.13_{0.88}$	$36.30_{0.49}$	$46.47_{0.24}$	$27.75_{0.34}$	$23.31_{0.51}$
EfficientTrain	$\mathbf{92.88}_{0.23}$	$79.91_{0.23}$	$74.58_{1.84}$	$69.40_{0.20}$	$50.52_{0.32}$	$39.91_{0.62}$	$51.76_{0.42}$	$38.33_{0.24}$	$28.15_{0.52}$

(b) ResNet-18

	CIFAR-10			CIFAR-100			Tiny-ImageNet		
	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50	Standard	Noise-0.4	Imbalance-50
SPL	$78.10_{1.29}$	$60.82_{0.92}$	$49.81_{1.29}$	$47.66_{0.40}$	$28.42_{1.21}$	$24.39_{0.78}$	$33.71_{0.63}$	$17.30_{0.87}$	$15.09_{0.39}$
TTCL	$77.36_{0.34}$	$\mathbf{69.83}_{0.53}$	$50.82_{0.79}$	$45.35_{0.59}$	$\mathbf{39.15}_{0.30}$	$24.15_{0.25}$	$35.61_{0.15}$	$\mathbf{29.76}_{0.34}$	$15.83_{0.38}$
MCL	$77.85_{0.55}$	$61.61_{0.65}$	$49.68_{0.79}$	$49.90_{0.53}$	$31.46_{0.59}$	$25.02_{0.62}$	$36.66_{0.75}$	$21.50_{0.58}$	$16.30_{0.43}$
ScreenerNet	$80.45_{0.53}$	$64.20_{0.50}$	$51.34_{1.08}$	$\mathbf{51.93}_{0.64}$	$34.77_{0.21}$	$26.32_{0.23}$	$38.14_{0.80}$	$24.90_{0.49}$	$\mathbf{17.47}_{0.14}$
LRE	$75.81_{0.52}$	$61.11_{2.95}$	$46.13_{2.60}$	$45.59_{0.64}$	$30.91_{0.29}$	$24.00_{0.51}$	$34.10_{0.71}$	$21.42_{0.28}$	$13.98_{0.44}$
MW-Net	$77.39_{2.30}$	$63.01_{0.60}$	$51.19_{1.25}$	$49.46_{0.44}$	$33.99_{0.38}$	$24.86_{0.47}$	$37.16_{0.29}$	$23.49_{0.33}$	$16.13_{0.40}$
DCL	$\mathbf{80.66}_{0.27}$	$66.00_{0.07}$	$51.73_{1.25}$	$51.23_{0.62}$	$37.01_{0.35}$	$\mathbf{26.40}_{0.34}$	$\mathbf{38.92}_{0.53}$	$26.17_{0.37}$	$17.20_{0.34}$
LGL	$79.52_{0.38}$	$63.19_{0.91}$	$52.14_{1.18}$	$50.39_{0.63}$	$31.34_{1.16}$	$26.09_{0.51}$	$36.25_{0.47}$	$20.22_{0.52}$	$16.43_{0.43}$
DDS	$77.54_{2.13}$	$63.46_{0.22}$	$51.12_{0.73}$	$49.67_{0.66}$	$33.79_{0.45}$	$24.81_{0.38}$	$36.60_{0.46}$	$23.47_{0.42}$	$16.18_{0.61}$
DIHCL	$78.09_{0.73}$	$63.39_{0.41}$	$50.78_{0.72}$	$49.80_{0.34}$	$33.64_{0.22}$	$25.49_{0.32}$	$37.89_{0.48}$	$22.36_{0.57}$	$16.29_{0.32}$
SuperLoss	$79.42_{0.25}$	$66.13_{0.49}$	$51.86_{0.60}$	$49.25_{0.37}$	$37.84_{0.39}$	$25.72_{0.27}$	$38.25_{0.42}$	$28.04_{0.39}$	$16.93_{0.26}$
CBS	$79.85_{0.37}$	$64.07_{0.65}$	$\mathbf{52.85}_{0.81}$	$51.05_{0.62}$	$35.25_{0.24}$	$26.05_{0.52}$	$38.28_{0.71}$	$24.88_{0.27}$	$17.15_{0.31}$
C2F	$79.63_{0.65}$	$61.97_{1.38}$	$52.00_{1.14}$	$50.16_{0.74}$	$32.58_{0.67}$	$25.28_{0.32}$	$38.51_{0.21}$	$25.22_{0.77}$	$17.02_{0.68}$
Adaptive CL	$78.85_{0.60}$	$62.55_{0.78}$	$51.60_{1.49}$	$48.30_{0.68}$	$31.73_{0.58}$	$24.81_{0.56}$	$33.94_{0.45}$	$20.12_{0.43}$	$15.27_{0.40}$
EfficientTrain	$79.67_{0.47}$	$62.62_{0.37}$	$50.71_{1.53}$	$50.98_{0.50}$	$34.56_{0.22}$	$25.47_{0.62}$	$38.21_{1.06}$	$25.08_{0.33}$	$16.20_{0.17}$

(c) ViT

Table 9. The performances of each curriculum learning method in the CV research domain.

	RTE		MRPC		STS-B		CoLA	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$52.42_{0.84}$	$53.36_{0.53}$	$80.64_{0.87}$	$80.46_{1.17}$	$11.04_{1.13}$	$8.76_{2.64}$	$9.96_{2.17}$	$3.69_{2.58}$
TTCL	$52.78_{0.14}$	$53.79_{1.81}$	$\mathbf{81.54}_{0.18}$	$81.22_{0.00}$	$\mathbf{14.11}_{2.21}$	$11.10_{2.25}$	$12.44_{2.22}$	$\mathbf{8.55}_{2.10}$
MCL	$52.85_{0.29}$	$52.64_{0.58}$	$81.22_{0.00}$	$80.95_{0.54}$	$12.95_{1.23}$	$10.55_{1.32}$	$10.13_{1.36}$	$4.16_{1.92}$
ScreenerNet	$52.85_{0.18}$	$53.72_{0.86}$	$81.40_{0.11}$	$\mathbf{81.24}_{0.05}$	$13.22_{0.96}$	$10.99_{1.41}$	$12.33_{1.01}$	$3.51_{2.16}$
DCL	$\mathbf{53.07}_{1.29}$	$\mathbf{54.22}_{1.77}$	$81.46_{0.18}$	$81.22_{0.00}$	$12.67_{0.79}$	$11.62_{1.10}$	$11.06_{1.68}$	$2.50_{1.89}$
DDS	$52.71_{0.00}$	$53.14_{0.42}$	$81.37_{0.08}$	$81.23_{0.03}$	$12.54_{1.28}$	$11.27_{2.73}$	$\mathbf{12.65}_{1.21}$	$3.51_{2.26}$
DIHCL	$52.71_{0.00}$	$53.72_{0.77}$	$81.37_{0.14}$	$81.22_{0.00}$	$13.99_{1.26}$	$9.89_{0.80}$	$11.69_{2.90}$	$3.41_{2.69}$
SuperLoss	$52.71_{0.00}$	$53.43_{1.10}$	$81.39_{0.14}$	$81.22_{0.00}$	$12.36_{1.65}$	$\mathbf{11.75}_{1.61}$	$10.82_{1.93}$	$3.59_{1.65}$
Adaptive CL	$52.06_{1.30}$	$53.00_{0.27}$	$81.39_{0.17}$	$81.22_{0.00}$	$12.91_{1.16}$	$10.32_{0.91}$	$9.82_{0.68}$	$4.38_{2.36}$

	SST-2		QNLI		QQP		MNLI-(m)		MNLI-(mm)	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$81.90_{0.62}$	$63.23_{0.76}$	$\mathbf{51.02}_{0.46}$	$50.74_{0.19}$	$74.39_{0.35}$	$59.63_{0.79}$	$60.62_{0.30}$	$36.58_{0.97}$	$60.45_{0.36}$	$36.36_{0.99}$
TTCL	$82.13_{0.91}$	$\mathbf{78.58}_{1.64}$	$50.65_{0.22}$	$50.73_{0.18}$	$75.14_{0.16}$	$\mathbf{66.47}_{0.72}$	$\mathbf{62.47}_{0.36}$	$\mathbf{58.59}_{0.54}$	$\mathbf{62.33}_{0.42}$	$\mathbf{58.50}_{0.64}$
MCL	$82.52_{0.99}$	$63.10_{2.08}$	$50.54_{0.00}$	$50.72_{0.29}$	$75.10_{0.15}$	$59.29_{0.39}$	$60.92_{0.42}$	$45.55_{1.91}$	$60.82_{0.24}$	$46.32_{2.08}$
ScreenerNet	$82.07_{0.43}$	$64.42_{0.85}$	$50.55_{0.02}$	$50.72_{0.23}$	$74.27_{0.19}$	$61.33_{0.30}$	$61.38_{0.37}$	$42.36_{1.49}$	$60.71_{0.25}$	$43.03_{1.60}$
DCL	$82.02_{0.76}$	$64.36_{1.08}$	$50.54_{0.00}$	$50.62_{0.15}$	$75.58_{0.31}$	$60.77_{0.70}$	$61.61_{0.34}$	$44.13_{0.74}$	$61.21_{0.41}$	$45.04_{0.77}$
DDS	$82.48_{0.68}$	$62.16_{1.36}$	$50.54_{0.00}$	$50.77_{0.27}$	$74.92_{0.14}$	$60.95_{0.42}$	$60.75_{0.42}$	$42.46_{0.89}$	$60.43_{0.19}$	$42.85_{0.98}$
DIHCL	$82.09_{0.88}$	$62.43_{0.92}$	$50.54_{0.00}$	$\mathbf{50.83}_{0.45}$	$74.09_{0.10}$	$59.71_{1.05}$	$58.84_{0.39}$	$37.17_{0.60}$	$58.84_{0.74}$	$36.65_{0.81}$
SuperLoss	$\mathbf{82.87}_{0.88}$	$65.48_{0.62}$	$50.59_{0.10}$	$50.76_{0.18}$	$\mathbf{75.73}_{0.21}$	$59.83_{0.19}$	$60.64_{0.33}$	$47.08_{1.68}$	$60.91_{0.58}$	$47.63_{1.52}$
Adaptive CL	$82.74_{0.75}$	$64.22_{2.23}$	$50.54_{0.00}$	$50.70_{0.23}$	$74.85_{0.45}$	$60.05_{1.30}$	$61.39_{0.34}$	$41.43_{1.69}$	$60.65_{0.45}$	$42.10_{1.82}$

(a) LSTM

	RTE		MRPC		STS-B		CoLA	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$61.37_{3.63}$	$51.34_{3.48}$	$87.21_{2.02}$	$80.57_{1.61}$	$85.07_{0.49}$	$80.91_{0.63}$	$56.07_{4.94}$	$15.08_{6.41}$
TTCL	$\mathbf{66.35}_{1.76}$	$\mathbf{56.32}_{5.04}$	$88.63_{1.88}$	$81.79_{0.57}$	$84.91_{0.68}$	$80.74_{1.66}$	$57.26_{0.87}$	$\mathbf{45.79}_{1.64}$
MCL	$\mathbf{66.35}_{2.02}$	$55.09_{2.22}$	$\mathbf{88.69}_{1.24}$	$78.94_{2.59}$	$\mathbf{85.42}_{0.22}$	$79.21_{0.65}$	$56.24_{2.37}$	$30.20_{5.94}$
ScreenerNet	$64.69_{1.62}$	$52.49_{5.06}$	$87.78_{0.99}$	$79.04_{4.22}$	$84.91_{0.45}$	$80.69_{0.97}$	$56.37_{1.62}$	$33.25_{3.26}$
LRE	$58.94_{1.34}$	$53.36_{1.24}$	$81.73_{0.34}$	$80.90_{0.64}$	$81.08_{1.76}$	$75.52_{2.07}$	$51.56_{2.12}$	$26.92_{3.88}$
MW-Net	$66.28_{0.81}$	$53.86_{2.73}$	$88.09_{0.61}$	$80.89_{0.67}$	$84.99_{0.92}$	$79.16_{1.19}$	$56.34_{2.19}$	$30.80_{1.89}$
DCL	$66.21_{2.58}$	$53.79_{4.20}$	$88.53_{1.13}$	$\mathbf{81.94}_{0.55}$	$85.09_{0.51}$	$80.99_{1.22}$	$57.47_{1.91}$	$32.66_{3.66}$
DDS	$64.55_{1.03}$	$55.45_{3.89}$	$87.32_{1.11}$	$79.41_{2.43}$	$84.38_{0.88}$	$78.00_{1.97}$	$56.12_{1.23}$	$27.49_{1.52}$
DIHCL	$64.48_{1.22}$	$54.80_{2.44}$	$86.85_{1.12}$	$81.47_{0.39}$	$85.05_{0.27}$	$\mathbf{81.31}_{0.25}$	$52.34_{1.49}$	$30.49_{4.98}$
SuperLoss	$66.06_{1.98}$	$53.79_{4.85}$	$88.05_{0.95}$	$81.82_{0.65}$	$84.58_{0.68}$	$79.78_{1.35}$	$57.35_{2.10}$	$31.81_{2.97}$
Adaptive CL	$65.85_{1.18}$	$54.80_{4.51}$	$87.54_{0.61}$	$81.64_{0.64}$	$85.27_{0.35}$	$79.72_{0.64}$	$\mathbf{57.80}_{1.96}$	$31.58_{3.18}$

	SST-2		QNLI		QQP		MNLI-(m)		MNLI-(mm)	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$91.49_{1.78}$	$85.13_{2.62}$	$90.28_{0.62}$	$80.98_{1.29}$	$87.30_{0.34}$	$76.17_{1.30}$	$83.87_{0.61}$	$77.63_{0.63}$	$84.25_{0.61}$	$78.59_{0.71}$
TTCL	$92.48_{0.41}$	$\mathbf{91.25}_{0.59}$	$91.37_{0.16}$	$\mathbf{89.45}_{0.44}$	$87.45_{0.46}$	$\mathbf{84.50}_{0.25}$	$83.99_{0.31}$	$\mathbf{81.73}_{0.31}$	$84.34_{0.45}$	$\mathbf{82.25}_{0.40}$
MCL	$92.41_{0.20}$	$84.33_{0.91}$	$91.24_{0.23}$	$80.71_{1.08}$	$\mathbf{88.16}_{0.13}$	$74.19_{1.02}$	$83.86_{0.42}$	$76.85_{0.79}$	$84.11_{0.29}$	$77.92_{0.79}$
ScreenerNet	$92.48_{0.27}$	$87.75_{0.96}$	$91.18_{0.11}$	$81.87_{1.40}$	$87.53_{0.22}$	$75.85_{1.26}$	$83.83_{0.42}$	$78.59_{0.52}$	$84.13_{0.44}$	$79.16_{0.61}$
LRE	$92.18_{0.38}$	$86.61_{1.54}$	$89.32_{0.47}$	$80.37_{0.83}$	$84.56_{0.32}$	$72.30_{0.02}$	$82.21_{0.29}$	$75.63_{0.51}$	$82.58_{0.29}$	$76.40_{0.53}$
MW-Net	$92.62_{0.41}$	$87.06_{0.96}$	$91.28_{0.20}$	$81.27_{1.40}$	$87.44_{0.19}$	$75.48_{0.61}$	$84.01_{0.23}$	$78.35_{0.68}$	$84.39_{0.38}$	$78.96_{0.62}$
DCL	$\mathbf{92.82}_{0.16}$	$86.67_{2.23}$	$\mathbf{91.49}_{0.13}$	$81.41_{1.98}$	$88.03_{0.21}$	$75.26_{0.95}$	$84.24_{0.27}$	$78.55_{0.46}$	$\mathbf{84.40}_{0.42}$	$79.39_{0.89}$
DDS	$92.41_{0.28}$	$86.19_{0.56}$	$91.14_{0.14}$	$81.88_{0.71}$	$87.50_{0.25}$	$76.04_{0.57}$	$83.89_{0.12}$	$78.51_{0.37}$	$84.38_{0.18}$	$78.85_{0.25}$
DIHCL	$92.52_{0.31}$	$87.75_{0.81}$	$91.23_{0.11}$	$83.03_{1.09}$	$86.74_{0.35}$	$76.75_{0.43}$	$83.28_{0.32}$	$78.51_{0.73}$	$83.57_{0.32}$	$79.42_{0.86}$
SuperLoss	$92.69_{0.41}$	$87.57_{1.45}$	$91.18_{0.14}$	$82.33_{0.51}$	$87.79_{0.20}$	$75.90_{0.55}$	$\mathbf{84.27}_{0.07}$	$77.68_{0.65}$	$84.36_{0.22}$	$78.71_{0.67}$
Adaptive CL	$92.32_{0.32}$	$85.89_{1.43}$	$91.24_{0.27}$	$80.58_{1.91}$	$87.60_{0.40}$	$76.27_{1.02}$	$84.11_{0.50}$	$78.79_{0.35}$	$84.39_{0.45}$	$79.35_{0.59}$

(b) BERT

	RTE		MRPC		STS-B		CoLA	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$59.42_{4.22}$	$54.73_{2.38}$	$84.32_{1.09}$	$79.47_{2.26}$	$76.66_{2.48}$	$63.16_{4.95}$	$30.72_{8.65}$	$2.95_{0.70}$
TTCL	$64.55_{0.62}$	$\mathbf{57.40}_{3.39}$	$84.79_{0.81}$	$\mathbf{82.55}_{0.88}$	$73.06_{4.74}$	$68.35_{4.64}$	$33.83_{3.10}$	$\mathbf{12.54}_{2.75}$
MCL	$66.21_{0.87}$	$54.95_{1.97}$	$\mathbf{86.29}_{0.36}$	$80.26_{1.29}$	$\mathbf{80.82}_{1.39}$	$\mathbf{71.57}_{1.74}$	$\mathbf{39.95}_{3.16}$	$8.40_{1.86}$
ScreenerNet	$65.13_{1.61}$	$53.36_{3.78}$	$84.97_{0.54}$	$78.58_{3.03}$	$74.77_{4.11}$	$69.49_{2.69}$	$35.89_{7.49}$	$6.27_{2.33}$
LRE	$60.22_{2.11}$	$52.85_{2.54}$	$81.98_{0.08}$	$75.27_{3.39}$	$56.41_{1.41}$	$65.02_{4.00}$	$35.00_{1.98}$	$3.31_{2.50}$
MW-Net	$64.33_{3.46}$	$54.94_{4.57}$	$84.06_{0.82}$	$77.33_{5.87}$	$77.11_{2.14}$	$65.77_{3.44}$	$35.24_{5.04}$	$3.47_{1.68}$
DCL	$\mathbf{66.35}_{2.10}$	$55.52_{2.75}$	$85.39_{0.89}$	$77.80_{3.50}$	$77.63_{1.76}$	$68.68_{2.96}$	$36.59_{3.57}$	$6.95_{3.88}$
DDS	$61.23_{3.39}$	$53.79_{3.00}$	$82.63_{0.69}$	$74.59_{3.52}$	$72.41_{6.45}$	$60.72_{3.34}$	$31.87_{1.82}$	$4.11_{2.82}$
DIHCL	$63.83_{2.48}$	$55.45_{2.38}$	$83.26_{0.53}$	$78.61_{2.44}$	$73.10_{3.53}$	$63.71_{1.27}$	$33.58_{1.92}$	$3.66_{1.96}$
SuperLoss	$66.21_{0.96}$	$53.72_{1.70}$	$85.12_{0.62}$	$79.18_{3.14}$	$73.65_{4.55}$	$66.13_{3.65}$	$37.60_{2.98}$	$8.90_{5.55}$
Adaptive CL	$65.49_{1.38}$	$53.86_{0.86}$	$84.82_{0.98}$	$78.05_{3.47}$	$76.58_{3.05}$	$66.30_{2.02}$	$33.61_{3.90}$	$6.50_{1.55}$

	SST-2		QNLI		QQP		MNLI-(m)		MNLI-(mm)	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$91.93_{0.45}$	$85.44_{1.40}$	$87.79_{0.35}$	$76.29_{2.81}$	$85.29_{0.31}$	$73.76_{1.51}$	$81.05_{0.27}$	$76.47_{0.27}$	$81.93_{0.52}$	$77.48_{0.40}$
TTCL	$92.18_{0.66}$	$\mathbf{90.34}_{0.53}$	$88.10_{0.22}$	$\mathbf{84.00}_{0.70}$	$85.50_{0.28}$	$\mathbf{82.16}_{0.35}$	$81.55_{0.27}$	$\mathbf{78.36}_{0.19}$	$82.18_{0.23}$	$\mathbf{79.62}_{0.44}$
MCL	$92.18_{0.44}$	$84.13_{1.83}$	$\mathbf{88.17}_{0.67}$	$77.80_{1.75}$	$\mathbf{86.68}_{0.16}$	$74.27_{2.28}$	$\mathbf{81.90}_{0.23}$	$75.44_{0.86}$	$\mathbf{82.59}_{0.35}$	$76.92_{0.92}$
ScreenerNet	$91.77_{0.63}$	$86.74_{1.35}$	$87.88_{0.50}$	$77.93_{1.86}$	$85.87_{0.05}$	$73.43_{2.11}$	$81.78_{0.22}$	$76.29_{0.30}$	$82.40_{0.11}$	$77.54_{0.44}$
LRE	$91.24_{0.20}$	$84.44_{1.20}$	$84.83_{0.58}$	$63.25_{3.95}$	$83.11_{0.73}$	$70.22_{1.22}$	$78.93_{0.47}$	$72.35_{0.71}$	$80.06_{0.51}$	$74.05_{0.66}$
MW-Net	$91.56_{0.28}$	$86.40_{1.58}$	$88.00_{0.38}$	$75.53_{3.18}$	$85.70_{0.27}$	$74.64_{0.64}$	$81.58_{0.36}$	$75.89_{0.28}$	$82.42_{0.30}$	$76.81_{0.18}$
DCL	$92.06_{0.49}$	$86.05_{1.25}$	$87.98_{0.19}$	$78.82_{0.64}$	$85.99_{0.21}$	$75.44_{0.72}$	$81.53_{0.27}$	$76.60_{0.47}$	$82.41_{0.20}$	$77.53_{0.30}$
DDS	$91.97_{0.23}$	$87.73_{1.61}$	$84.59_{2.24}$	$79.88_{0.02}$	$85.73_{0.05}$	$72.56_{2.21}$	$81.41_{0.31}$	$75.49_{0.21}$	$82.14_{0.40}$	$76.86_{0.08}$
DIHCL	$91.88_{0.41}$	$87.02_{1.14}$	$86.85_{0.34}$	$78.97_{0.88}$	$83.92_{0.41}$	$75.07_{0.57}$	$80.30_{0.23}$	$76.41_{0.14}$	$81.69_{0.12}$	$77.68_{0.12}$
SuperLoss	$\mathbf{92.25}_{0.42}$	$87.55_{0.72}$	$87.99_{0.52}$	$79.70_{0.65}$	$86.13_{0.18}$	$75.83_{0.70}$	$81.33_{0.18}$	$75.90_{0.29}$	$82.14_{0.27}$	$77.05_{0.25}$
Adaptive CL	$92.11_{0.24}$	$85.78_{1.42}$	$87.79_{0.18}$	$78.14_{1.66}$	$85.72_{0.21}$	$75.72_{0.57}$	$81.38_{0.11}$	$76.04_{0.41}$	$82.38_{0.34}$	$77.44_{0.37}$

(c) GPT2

Table 10. The performances of each curriculum learning method in the NLP research domain.

	MUTAG		PROTEINS		NCI1		ogbg-molhiv	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$71.58_{7.14}$	$62.10_{3.94}$	$69.46_{5.91}$	$65.54_{6.48}$	$68.42_{1.90}$	$60.05_{2.38}$	$\mathbf{77.41}_{1.15}$	$60.87_{3.09}$
TTCL	$70.52_{7.14}$	$\mathbf{71.58}_{5.37}$	$72.68_{7.63}$	$\mathbf{71.61}_{6.62}$	$70.90_{2.21}$	$\mathbf{67.98}_{2.01}$	$75.89_{0.81}$	$\mathbf{72.81}_{1.14}$
MCL	$71.58_{7.14}$	$\mathbf{71.58}_{8.55}$	$70.54_{5.15}$	$65.00_{4.71}$	$68.56_{1.04}$	$54.50_{2.85}$	$74.10_{1.40}$	$64.26_{4.17}$
ScreenerNet	$72.63_{3.94}$	$64.21_{5.16}$	$71.96_{5.61}$	$67.14_{4.05}$	$69.78_{2.22}$	$56.06_{5.14}$	$73.71_{0.45}$	$61.00_{7.79}$
LRE	$70.52_{9.18}$	$61.40_{4.96}$	$68.03_{6.17}$	$66.61_{5.25}$	$58.23_{1.60}$	$51.22_{2.38}$	$73.74_{1.48}$	$57.92_{7.98}$
MW-Net	$74.73_{2.11}$	$63.16_{4.71}$	$70.54_{4.55}$	$66.79_{4.13}$	$68.71_{1.78}$	$56.01_{1.37}$	$75.57_{1.03}$	$62.81_{6.19}$
DCL	$74.73_{2.11}$	$61.05_{13.56}$	$71.96_{3.46}$	$63.57_{6.32}$	$70.51_{0.66}$	$56.69_{1.58}$	$75.78_{1.39}$	$61.26_{3.57}$
DDS	$\mathbf{74.74}_{3.94}$	$64.21_{5.16}$	$\mathbf{73.21}_{4.41}$	$64.11_{6.50}$	$\mathbf{71.39}_{1.29}$	$58.10_{3.28}$	$70.48_{3.02}$	$57.09_{4.80}$
DIHCL	$71.58_{5.37}$	$68.42_{7.44}$	$73.03_{3.59}$	$63.22_{7.02}$	$67.40_{1.71}$	$57.86_{2.04}$	$70.47_{2.10}$	$61.20_{4.67}$
SuperLoss	$71.58_{5.37}$	$69.47_{6.14}$	$72.32_{3.44}$	$65.89_{3.84}$	$70.22_{2.00}$	$57.17_{3.38}$	$75.97_{1.03}$	$61.21_{5.12}$
Adaptive CL	$73.68_{3.33}$	$66.31_{7.88}$	$72.68_{4.84}$	$65.71_{4.51}$	$69.88_{2.17}$	$58.44_{5.29}$	$75.49_{1.13}$	$60.95_{7.96}$

(a) GCN

	MUTAG		PROTEINS		NCI1		ogbg-molhiv	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$64.21_{11.72}$	$65.26_{5.37}$	$69.29_{6.93}$	$67.14_{3.85}$	$56.49_{2.61}$	$54.74_{4.10}$	$69.69_{2.38}$	$64.88_{2.73}$
TTCL	$69.47_{6.98}$	$65.26_{10.84}$	$\mathbf{69.82}_{7.13}$	$64.46_{1.31}$	$56.79_{1.40}$	$55.47_{2.73}$	$68.27_{2.04}$	$\mathbf{66.73}_{1.84}$
MCL	$64.21_{11.24}$	$68.42_{8.81}$	$69.64_{6.29}$	$66.96_{6.89}$	$57.56_{2.63}$	$55.23_{4.73}$	$69.25_{3.06}$	$63.20_{3.03}$
ScreenerNet	$64.21_{8.42}$	$65.26_{7.88}$	$65.71_{5.25}$	$\mathbf{69.11}_{3.77}$	$54.55_{2.64}$	$55.28_{1.53}$	$71.13_{2.07}$	$65.94_{2.61}$
LRE	$66.31_{11.34}$	$63.16_{4.71}$	$66.43_{1.84}$	$66.07_{3.19}$	$54.11_{2.32}$	$52.94_{2.36}$	$66.59_{2.45}$	$63.74_{2.61}$
MW-Net	$61.84_{9.40}$	$65.26_{7.14}$	$66.78_{3.01}$	$67.14_{4.42}$	$57.56_{2.29}$	$55.33_{1.18}$	$68.54_{3.76}$	$62.39_{2.60}$
DCL	$67.37_{14.28}$	$\mathbf{69.47}_{10.21}$	$68.03_{7.39}$	$64.28_{3.84}$	$\mathbf{59.37}_{1.59}$	$55.33_{1.72}$	$\mathbf{72.64}_{1.16}$	$62.22_{3.98}$
DDS	$66.31_{7.14}$	$67.37_{6.14}$	$67.14_{3.31}$	$66.78_{9.69}$	$53.24_{1.81}$	$54.45_{3.35}$	$68.50_{2.05}$	$62.22_{5.88}$
DIHCL	$\mathbf{72.63}_{8.42}$	$66.32_{8.55}$	$65.00_{6.81}$	$68.57_{6.93}$	$57.18_{1.73}$	$\mathbf{55.67}_{4.70}$	$69.07_{2.79}$	$66.38_{2.78}$
SuperLoss	$67.37_{13.06}$	$68.42_{7.44}$	$63.93_{2.63}$	$66.07_{7.23}$	$57.08_{2.27}$	$55.13_{2.39}$	$70.58_{1.52}$	$60.92_{2.13}$
Adaptive CL	$67.37_{10.21}$	$66.32_{7.14}$	$68.39_{3.07}$	$64.47_{6.37}$	$57.61_{2.22}$	$55.08_{2.02}$	$69.71_{1.84}$	$62.98_{2.53}$

(b) GAT

	MUTAG		PROTEINS		NCI1		ogbg-molhiv	
	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4	Standard	Noise-0.4
SPL	$82.10_{5.37}$	$72.37_{6.84}$	$72.86_{5.13}$	$72.86_{2.37}$	$77.54_{1.69}$	$56.87_{4.93}$	$\mathbf{76.53}_{1.97}$	$63.35_{2.34}$
TTCL	$84.21_{7.44}$	$\mathbf{81.58}_{4.56}$	$75.71_{2.36}$	$\mathbf{73.93}_{1.82}$	$80.24_{1.67}$	$56.27_{4.27}$	$75.13_{1.55}$	$62.16_{3.07}$
MCL	$84.21_{6.45}$	$73.69_{5.27}$	$75.72_{3.93}$	$70.00_{2.68}$	$75.67_{1.00}$	$57.73_{4.11}$	$74.20_{0.48}$	$63.82_{3.85}$
ScreenerNet	$82.10_{7.14}$	$75.00_{5.74}$	$75.71_{1.82}$	$68.39_{4.88}$	$79.61_{1.09}$	$55.57_{5.11}$	$74.39_{1.24}$	$61.07_{2.33}$
LRE	$78.95_{3.72}$	$80.27_{2.28}$	$72.68_{5.46}$	$66.43_{6.48}$	$71.41_{1.71}$	$54.08_{1.72}$	$73.49_{2.36}$	$63.30_{4.44}$
MW-Net	$\mathbf{88.42}_{2.10}$	$75.00_{4.37}$	$73.75_{4.10}$	$66.61_{8.54}$	$79.22_{1.21}$	$55.52_{4.78}$	$75.22_{0.80}$	$65.43_{2.70}$
DCL	$85.26_{8.42}$	$76.32_{4.56}$	$74.11_{3.14}$	$64.46_{4.39}$	$79.66_{1.39}$	$56.06_{3.79}$	$75.23_{2.22}$	$61.65_{3.38}$
DDS	$85.26_{3.94}$	$80.26_{5.73}$	$70.31_{2.78}$	$65.89_{6.69}$	$77.62_{3.58}$	$54.89_{4.85}$	$72.85_{2.67}$	$63.38_{3.91}$
DIHCL	$85.53_{4.36}$	$73.68_{3.72}$	$73.75_{4.54}$	$71.61_{4.28}$	$76.55_{1.70}$	$53.33_{1.28}$	$72.43_{1.80}$	$62.23_{5.86}$
SuperLoss	$\mathbf{88.42}_{5.16}$	$77.63_{4.37}$	$\mathbf{77.14}_{4.88}$	$71.25_{5.69}$	$\mathbf{82.04}_{1.90}$	$\mathbf{62.14}_{6.47}$	$74.51_{1.47}$	$\mathbf{65.53}_{1.61}$
Adaptive CL	$86.31_{4.21}$	$80.26_{9.40}$	$75.89_{3.79}$	$70.36_{3.97}$	$79.32_{1.90}$	$62.05_{1.67}$	$76.17_{1.46}$	$61.81_{4.81}$

(c) GIN

Table 11. The performances of each curriculum learning method in the graph research domain.