# Training Dynamics for Curriculum Learning: A Study on Monolingual and Cross-lingual NLU

Anonymous ACL submission


## Abstract

Curriculum Learning (CL) is a technique of training models via ranking examples in a typically increasing difficulty trend with the aim of accelerating convergence and improving generalisability. However, current approaches for Natural Language Understanding (NLU) tasks use CL to improve in-domain model performance often via metrics that are detached from the model one aims to improve. In this work, instead, we employ CL for NLU by taking advantage of training dynamics as difficulty metrics, i.e. statistics that measure the behavior of the model at hand on data instances during training. In addition, we propose two modifications of existing CL schedulers based on these statistics. Differently from existing works, we focus on evaluating models on out-of-distribution data as well as languages other than English via zero-shot cross-lingual transfer. We show across four XNLU tasks that CL with training dynamics in both monolingual and cross-lingual settings can achieve significant speedups up to 58%. We also find that performance can be improved on challenging tasks, with OOD generalisation up by 8% and zero-shot cross-lingual transfer up by 1%. Overall, experiments indicate that training dynamics can lead to better performing models and smoother training compared to other difficulty metrics.


## 1 Introduction

Transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019, LMs) have recently achieved great success in a variety of NLP tasks (Wang et al., 2018, 2019). However, generalisation to out-of-distribution (OOD) data and zero-shot cross-lingual natural language understanding (XNLU) tasks still remains a challenge (Linzen, 2020; Hu et al., 2020). Among existing techniques, improving OOD performance has been addressed by training with adversarial data (Yi et al., 2021), while better transfer across languages has mostly focused on selecting appropriate languages to transfer from (Lin et al., 2019; Turc et al., 2021) or employing meta-learing with auxiliary language data (Nooralahzadeh et al., 2020).

Contrastive to such approaches that take advantage of additional training data is Curriculum Learning (Bengio et al., 2009, CL), a technique that aims to train models using a specific ordering of the original training examples. This ordering typically follows an increasing difficulty trend where easy examples are fed to the model first, moving towards harder instances. The intuition behind CL stems from human learning, as humans focus on simpler concepts before learning more complex ones, a procedure that is called shaping (Krueger and Dayan, 2009). Although curricula have been primarily used for Computer Vision (Hacohen and Weinshall, 2019; Wu et al., 2021) and Machine Translation (Zhang et al., 2019a; Platanios et al., 2019), there are only a handful of approaches that incorporate CL into Natural Language Understanding tasks (Sachan and Xing, 2016; Tay et al., 2019; Lalor and Yu, 2020; Xu et al., 2020a).

Typically, CL requires a measure of difficulty for each example in the training set. Existing methods using CL in NLU tasks vastly rely on heuristics such as sentence length, word rarity, depth of the dependency tree (Platanios et al., 2019; Tay et al., 2019) or external model metrics such as perplexity (Zhou et al., 2020), performance (Xu et al., 2020a) or information theory (Lalor and Yu, 2020). Although such metrics do make sense for Machine Translation (e.g. longer sentences are indeed harder to be translated), in language abstraction tasks such as Natural Language Inference or Commonsense Reasoning this is not always the case.

In this study instead, we propose to adopt Training dynamics (TD) (Swayamdipta et al., 2020) as difficulty measures for CL and fine-tune models with curricula on downstream tasks. TD were recently proposed as a set of statistics collected during the course of a model's training to automatically evaluate dataset quality, by identifying annotation artifacts. These statistics, offer a 3-dimensional view of a model's uncertainty towards each training example classifying them into distinct areas–*easy*, *ambiguous* and *hard* examples for a model to learn.

In this work, we test a series of easy-to-hard curricula using TD with existing schedulers as well as novel modifications of those. We evaluate both monolingual and multilingual models on four XNLU tasks: Natural Language Inference, Paraphrase Identification, Commonsense Causal Reasoning and Document Classification, focusing on zero-shot cross-lingual transfer and OOD data performance. To the best of our knowledge, no prior work on NLU considers the impact of CL on such instances. Our findings suggest that CL provides increased zero-shot cross-lingual transfer up to 1% over standard random training, especially on large datasets in addition to gaining speedups up to 58%. In OOD settings, monolingual models trained with curriculum learning incorporating TD can boost performance up to 8% and compared to other metrics provide more stable training.

## 2 Related Work

Curriculum Learning was initially mentioned in the work of Elman (1993) who demonstrated the importance of feeding neural networks with small/easy inputs at the early stages of training. The concept was later formalised by Bengio et al. (2009) where training in an easy-to-hard ordering was shown to result in faster convergence and improved performance. In general, Curriculum Learning requires a *difficulty metric* (also known as the scoring function) used to rank training instances, and a *scheduler* (known as the pacing function) that decides when and how new examples–of different difficulty–should be introduced to the model.

**Example Difficulty** was initially expressed via model loss, in self-paced learning (Kumar et al., 2010; Jiang et al., 2015), increasing the contribution of harder training instances over time. This setting posed a challenge due to the fast-changing pace of the loss during training, thus later approaches used human-intuitive difficulty metrics, such as sentence length or the existence of rare words (Platanios et al., 2019) to pre-compute difficulties of training instances. However, as such metrics often express superficial difficulty, automatic metrics have been proposed over the years, such as measuring the loss difference between two checkpoints (Xu et al., 2020b). In our curricula we use training dynamics to measure example difficulty, i.e. metrics that consider difficulty from the perspective of a model. Example difficulty can be also estimated either in a static or dynamic manner, where in the latter training instances are evaluated and re-ordered at certain times during training, while in the former the difficulty of each example remains the same throughout. In our experiments we adopt the first setting and consider static example difficulties.

**Transfer Teacher CL** is a particular family of such approaches that use an external model (namely the teacher) to measure the difficulty of training examples. Notable works incorporate a simpler model as the teacher (Zhang et al., 2018) or a larger-sized model (Hacohen and Weinshall, 2019), as well as using similar-sized learners trained on different subsets of the training data. These methods have considered as example difficulty, either the teacher model perplexity (Zhou et al., 2020), the norm of a teacher model word embeddings (Liu et al., 2020), the teacher's performance on a certain task (Xu et al., 2020a) or simply regard difficulty as a latent variable in a teacher model (Lalor and Yu, 2020). In the same vein, we also incorporate Transfer Teacher CL via teacher and student models of the same size and type. However, differently, we take into account the behavior of the teacher *during the course of its training* to measure example difficulty instead of considering its performance at the end of training or analysing internal embeddings.

Moving on to **Schedulers**, these can be divided into discrete and continuous. Discrete schedulers, often referred to as *bucketing*, group training instances that share similar difficulties into distinct sets. Different configurations include accumulating buckets over time (Cirik et al., 2016), sampling a subset of data from each bucket (Xu et al., 2020a; Kocmi and Bojar, 2017) or more sophisticated sampling strategies (Zhang et al., 2018). In cases where the number of buckets is not obtained in a straightforward manner, methods either heuristically split examples (Zhang et al., 2018), adopt uniform splits (Xu et al., 2020a) or employ schedulers that are based on a continuous function. A characteristic approach is that of Platanios et al. (2019) where at each training step a monotonically increasing function chooses the amount of training data the model has access to, sorted by increasing difficulty. As we will describe later on, we experiment with two established schedulers and propose modifications of those based on training dynamics.

Other tasks where CL has been employed include Question Answering (Sachan and Xing, 2016), Reading comprehension (Tay et al., 2019) and other general NLU classification tasks (Lalor and Yu, 2020; Xu et al., 2020a). Others have developed curricula in order to train models for code-switching (Choudhury et al., 2017), anaphora resolution (Stojanovski and Fraser, 2019), relation extraction (Huang and Du, 2019), dialogue (Saito, 2018; Shen and Feng, 2020) and self-supervised NMT (Ruiter et al., 2020), while more advanced approaches combine it with Reinforcement Learning in a collaborative teacher-student transfer curriculum (Kumar et al., 2019).

## 3 Methodology

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be a set of training data instances. A curriculum is comprised of three main elements: the *difficulty metric*, responsible for associating a training example to a score that represents a notion of difficulty, the *scheduler* that determines the type and number of available instances at each training step $t$ and the *curriculum order*, i.e. sorting examples in increasing, decreasing or random order of difficulty. In this study, we experiment with 3 difficulty metrics we introduce by training dynamics, 2 orderings (easy-to-hard and random) and 4 schedulers: 2 existing ones and 2 variations of those that we also introduce.

### 3.1 Difficulty Metrics

As aforementioned, we use training dynamics (Swayamdipta et al., 2020), i.e. statistics originally introduced to analyse dataset quality, as difficulty metrics. The suitability of such statistics to serve as difficulty measures for CL is encapsulated in three core aspects. Firstly, TD are straightforward. They can be easily obtained by training a single model on the target dataset and keeping statistics about its predictions on the training set. Secondly, TD correlate well with model uncertainty and follow a similar trend to human (dis)agreement in terms of data annotation, essentially combining the view of both worlds. Finally, TD manifest a clear pattern of separating instances into distinct areas–*easy*, *ambiguous* and *hard* examples for a model to learn–something that aligns well with the ideas behind Curriculum Learning.

The difficulty of an example $(\mathbf{x}_i, y_i)$ can be determined by a function $f$, where an example $i$ is considered more difficult than example $j$ if $f(\mathbf{x}_i, y_i) > f(\mathbf{x}_j, y_j)$. We list three difficulty metrics that use statistics during the course of a model's training, as follows:

**CONFIDENCE** of an example $\mathbf{x}_i$ is the average probability assigned to the gold label $y_i$ by a model with parameters $\theta$ across a number of epochs $E$. This is a continuous metric with higher values corresponding to easier examples.

$$f_{\text{CONF}}(x_i, y_i) = \mu_i = \frac{1}{E} \sum_{e=1}^{E} p_{\theta^{(e)}}(y_i|x_i) \quad (1)$$

**VARIABILITY** of an example $\mathbf{x}_i$ is the standard deviation of the probabilities assigned to the gold label $y_i$ across $E$ epochs. It is a continuous metric with higher values indicating greater uncertainty for a training example and as such higher difficulty.

$$f_{\text{VAR}}(x_i, y_i) = \sqrt{\frac{\sum_{e=1}^{E} \left(p_{\theta^{(e)}}(y_i|x_i) - \mu_i\right)^2}{E}} \quad (2)$$

**CORRECTNESS** is the number of times a model classifies example $\mathbf{x}_i$ correctly across its training. It takes values between 0 and $E$. Higher correctness indicates easier examples for a model to learn.

$$f_{\text{CORR}}(x_i, y_i) = \sum_{e=1}^{E} o_i^{(e)},$$

$$o_i^{(e)} = \begin{cases} 1 & \text{if } \arg\max p_{\theta^{(e)}}(x_i) = y_i \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Confidence and correctness are the primary metrics that we use in the curricula that we test since low and high values correspond to hard and easy examples respectively. On the other hand, variability is used as an auxiliary metric since only high variability scores clearly represent ambiguous examples while low scores offer no important information on their own.

### 3.2 Schedulers

In our experiments, we consider both discrete and continuous schedulers $g$, described below.
The **ANNEALING** ($\text{Anneal}_{\text{TD}}$) scheduler proposed by Xu et al. (2020a), assumes that training data are split into buckets $\{d_1 \subset D, \ldots, d_K \subset D\}$ with possibly different sizes $|d_i|$. In particular, we group examples into the same bucket if they have the same *correctness* score (see Equation (3)). In

total, this results in $E+1$ buckets, which are sorted in order of increasing difficulty. Training starts with the easiest bucket. We then move on to the next bucket by also randomly selecting $1/(E+1)$ examples from each previous bucket. This provides a smooth transition between buckets. Following prior work, we train on each bucket for one epoch. The COMPETENCE ($\text{Comp}_{\text{TD}}$) scheduler was originally proposed by Platanios et al. (2019). Here, we sort examples based on the *confidence* metric (see Equation (1)), and use a monotonically increasing function to obtain the percentage of available training data at each step. The model can use only the top $K$ most confident examples as instructed by this function. A mini-batch is then sampled uniformly from the available examples[1].

In addition to those schedulers, we introduce the following modifications that take advantage of the variability metric. ANNEALING VARIABILITY ($\text{AnnealVar}_{\text{TD}}$) is a modification of the Annealing scheduler and COMPETENCE VARIABILITY ($\text{CompVar}_{\text{TD}}$) is a modification of the Competence scheduler. In both variations, instead of sampling uniformly across available examples, we give higher probability to instances with high *variability* scores (Equation (2)). We assume that since the model is more uncertain about such examples further training on them can be beneficial. For all curricula, after the model has finished the curriculum stage, we resume training as normal, i.e. by random sampling of training instances.

### 3.3 Transfer Teacher Curriculum Learning

In a transfer teacher CL setting a teacher model is used to obtain the difficulty of training examples (Matiisen et al., 2019). As such, the previously presented difficulty metrics are suitable to be used in this setting, due to their nature, where we first need to fine-tune a model for a few epochs on a given dataset to get training dynamics for each training example. Then, a student model can be trained with the curriculum defined by the teacher.

The two-step procedure that we follow in this study is depicted in Figure 1. Initially a model (the *teacher*) is fine-tuned normally on a target dataset and training dynamics are collected during the course of training. The collected dynamics are then converted into difficulty metrics, following Equations (1)-(3). In the second stage, the difficulty metrics and the original training data are fed into a scheduler that re-orders the examples according to their difficulty (in our case from easy-to-hard) and feeds them into another model (the *student*) that is the same in size as the teacher.

[1]The competence curriculum that we test is slightly different from that proposed in prior work. Here, we simply use the competence function to select a *portion* of data at each step ordered by increasing difficulty, instead of selecting examples with scores less than the output of the competence function.

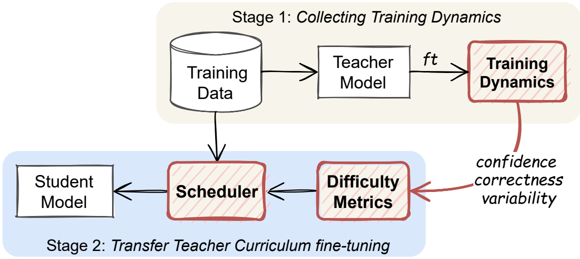



Figure 1: Transfer Teacher Curriculum Learning used in our study. A teacher model determines the difficulty of training examples by collecting training dynamics during fine-tuning (Stage 1). The collected dynamics are converted into difficulty metrics and into a student model via a scheduler (Stage 2).

| | | PAWS-X | XNLI | XCOPA | MLDoc |
|---|---|---|---|---|---|
| # Languages | | 7 | 15 | 12 | 8 |
| Training set | | PAWS | MultiNLI | SIQA | Reuters |
| ID | # Train | 49,401 | 392,702 | 33,410 | 10,000 |
| | # Dev. | 2,000 | 2,490 | 100 | 1,000 |
| | # Test | 2,000 | 5,010 | 500 | 4,000 |
| OOD | | TwitterPPBD | NLI Diag. | CSQA | - |
| | # Test | 9,324 | 1,105 | 1,221 | - |

Table 1: Datasets statistics. ID and OOD denote in-distribution and out-of-distribution, respectively. ID Development and Test statistics are per language.

## 4 Experimental Setup

### 4.1 Datasets

In this work we focus on four XNLU tasks: Natural Language Inference, Paraphrase Identification, Commonsense Causal Reasoning and Document Classification. The datasets that we use include XNLI (Conneau et al., 2018), PAWS-X (Yang et al., 2019), XCOPA (Ponti et al., 2020) and MLDoc (Schwenk and Li, 2018) that combined cover 25 languages. We also use OOD test sets, including NLI Diagnostics (Wang et al., 2018), TwitterPPBD (Lan et al., 2017) and CommonSenseQA (Talmor et al., 2019) for each dataset respectively, except for MLDoc. The corresponding statistics are shown in Table 1 and more details can be found in Appendix A.

### 4.2 Curriculum Parameters

In order to collect TD we first fine-tune either a RoBERTa or an XLM-R model on the English training set of each dataset. TD for each example are collected over 10 epochs on XNLI, PAWS-X and SIQA, while for MLDoc we train for 5 epochs. The COMPETENCE and COMPETENCE VARIABILITY schedulers require to set in advance the number of steps, i.e. total duration of the curriculum phase. We employ the same parameters as in Platanios et al. (2019) and set this value to 90% of steps that the baseline model requires to achieve its best performance on the development set. The initial competence is set to 0.01 for all datasets. We evaluate each model at the end of each epoch and at regular intervals (Dodge et al., 2020), every 500 updates for XNLI (corresponding to 24 times per epoch) and 10 times per epoch for the rest of the datasets. Performance is reported over three random seeds.

### 4.3 Evaluation Settings

For all datasets, we report accuracy as the main evaluation metric on the following settings.

**ZERO-SHOT**: Constitutes the zero-shot cross-lingual transfer setting, where a multilingual model (e.g. XLM-R) is trained on English data only and tested on languages other than English (Hu et al., 2020). **OOD**: Monolingual models (e.g. RoBERTa) are evaluated on out-of-distribution datasets with and without curriculum learning.

In all experiments, we select the best checkpoint based on the *English development set* performance. We use the pre-trained versions of RoBERTa (Liu et al., 2019) and XLM-R (Conneau et al., 2020) from the HuggingFace library[2] (Wolf et al., 2020).

### 4.4 Model Comparisons

We primarily compare all curricula that use training dynamics against each other and against a baseline (*Random*) that does not employ any curriculum and is using standard random order training.

We also compare with another teacher-transfer curriculum proposed by Xu et al. (2020a), namely *Cross-Review* (indicated as Anneal$_{\text{CR}}$ in the next sections). This curriculum uses the annealing scheduler, but does not employ training dynamics as difficulty scores. Instead, the method splits the training set into subsets and a model is trained on each subset containing $1/N$ of the training set. The resulting models are then used to evaluate all examples belonging in different subsets and the difficulty score of an example is considered the sum of its correct classifications across teachers.

The difference between this metric and the *correctness* metric is that Cross-Review uses $N$ fully trained teacher models on subsets of data, while the latter uses $E$ epochs of a single model trained on the entire training set to obtain the number of correct classifications for each training example. We split each training set into 10 subsets for all datasets, except MLDoc where we split into 5 due to its smaller size, following prior work.

We denote curricula that employ Training Dynamics as difficulty metrics with the TD subscript and curricula employing the Cross Review metric with CR. Finally, when comparing models on the same dataset we make sure that all of them are trained for the same number of total steps, i.e. after the end of the entire curriculum phase, training continues as normal for the remaining steps.

## 5 Experiments

### 5.1 Training Time

Since CL can typically achieve faster convergence, we first report the training time required by each model to achieve its best performance on the English development set. Results on Table 2 show the training time required for multilingual (Table 2a) and monolingual models (Table 2b). In particular, the reported numbers are calculated as the ratio $N_{\text{curric}}/N_{\text{random}}$, i.e. the number of steps the curriculum needs to reach best performance ($N_{\text{curric}}$) divided by the number of steps the random training needs to reach its best performance ($N_{\text{random}}$). By default, random training has a ratio of 1.0 and a lower score indicates a larger speedup. In addition, we report in parentheses the minimum time obtained across 3 random seeds.

Looking across the board in the majority of datasets AnnealVar$_{\text{TD}}$ (our proposed Annealing scheduler modification with sampling examples based on variability) is the curriculum that offers the most speedup in XLM-R models, with 24% in PAWS-X, 22% in XNLI and 20% in MLDoc on average and 49% in PAWS-X, 57% in XNLI and 58% in MLDoc in the best case. Other curricula require a few more training steps compared to random on average. Compared to Anneal$_{\text{CR}}$ our proposed variability sampling achieves higher speedups both on average and in the best scenario.

[2] https://huggingface.co/roberta-base, https://huggingface.co/xlm-roberta-base

| Train / Test | PAWS-X | XNLI | SIQA / XCOPA | MLDoc |
|---|---|---|---|---|
| Random | 1.00 | 1.00 | 1.00 | 1.00 |
| Anneal$_{\text{TD}}$ | 1.04 (0.70) | 1.12 (0.94) | 0.80 (0.38) | 0.91 (0.81) |
| AnnealVar$_{\text{TD}}$ | **0.76** (**0.51**) | **0.78** (**0.43**) | 1.14 (0.38) | **0.81** (**0.42**) |
| Comp$_{\text{TD}}$ | 1.43 (1.03) | 1.15 (0.46) | 0.49 (0.32) | 1.12 (1.03) |
| CompVar$_{\text{TD}}$ | 1.47 (0.94) | 1.18 (0.93) | 0.56 (**0.13**) | 0.99 (0.71) |
| Anneal$_{\text{CR}}$ | 1.08 (0.65) | 1.02 (0.86) | **0.39** (0.22) | 0.82 (0.74) |

(a) Zero-shot cross-lingual training time across 4 datasets using XLM-R models with and without CL.

| | PAWS-X / TwitterPPDB | XNLI / NLI Diag. | SIQA / CSQA |
|---|---|---|---|
| Random | 1.00 | 1.00 | 1.00 |
| Anneal$_{\text{TD}}$ | **0.79** (**0.63**) | **0.87** (**0.51**) | 0.85 (0.68) |
| AnnealVar$_{\text{TD}}$ | 0.97 (0.64) | 1.61 (1.34) | **0.44** (**0.23**) |
| Comp$_{\text{TD}}$ | 1.71 (0.58) | 1.32 (1.11) | 0.79 (0.31) |
| CompVar$_{\text{TD}}$ | 1.64 (1.51) | 1.47 (1.33) | 0.92 (0.61) |
| Anneal$_{\text{CR}}$ | 1.56 (0.89) | 1.31 (0.63) | 0.69 (0.55) |

(b) OOD training time across 3 datasets using RoBERTa models with and without CL.

Table 2: Numbers correspond to the ratio $N_{\text{curric}}/N_{\text{random}}$, where the numerator is the number steps a curriculum needs to reach the reported performance and the denominator is the number of steps the Random training baseline requires to reach its performance. Results are reported as mean over 3 random seeds, with the minimum shown in parentheses.

An exception is the case of XCOPA where cross-review appears to be much faster. We speculate that maybe the examples sampled for this particular task could not offer meaningful information for better performance earlier. However, looking at the best performance achieved by this scheduler (shown later on in Table 3), we see that despite the speedup Anneal$_{\text{CR}}$ offers, it results in lower performance than the random baseline. In the case of OOD data with RoBERTa models, we find that in CSQA all curricula offer significant speedup, while the Anneal$_{\text{TD}}$ curriculum achieves the highest speedup, 21%, 13% on average and 37%, 49% in the base case, on TwitterPPDB and NLI Diagnostics, respectively.

### 5.1.1 Learning Curves

In order to examine the behavior of the curricula during the course of training, we further plot the average language development performance as a function of the number of training steps when using XLM-R models. In Figure 2 we draw vertical lines to show the exact step that training with CL achieves higher performance to that of random training for the first time.

For all datasets, there are curricula that always achieve similar performance earlier than the random training, i.e. AnnealVar$_{\text{CR}}$ and Anneal$_{\text{CR}}$. However, for Anneal$_{\text{CR}}$ we observe a performance drop around 3K steps in PAWS-X and a much more evident one around 20K steps in XNLI. Further investigation revealed that during these steps the curriculum is going through the examples of the last bucket–which is the hardest one. This drop in performance possibly indicates that buckets created by cross-review do not necessarily contain examples that help the model prepare for the hardest exam-

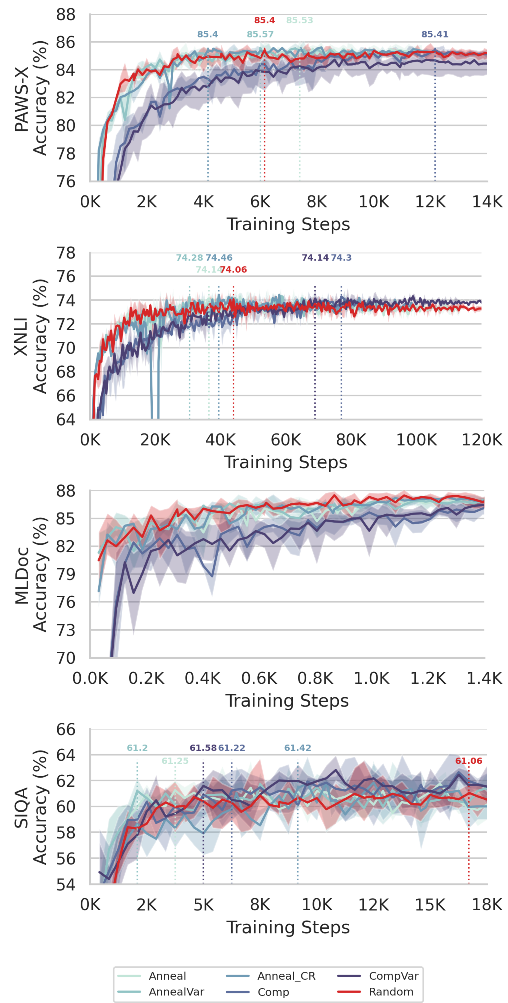



Figure 2: Average development set accuracy across languages as a function of learning steps (in thousands) with XLM-R models as student and teacher. The vertical lines indicate the best performance of random training order (red) and the next closest (higher) performance of one of the tested curricula (color shown is based on best curriculum). Results are reported over 3 random seeds.

| TRAIN<br>TEST | PAWS-X | XNLI | SIQA<br>XCOPA | MLDOC |
|---|---|---|---|---|
| Prior Work | 84.90* | 75.00* | 60.72 | 77.66 |
| Random | 84.49 ±0.08 | 73.93 ±0.18 | 60.62 ±0.54 | **86.74** ±0.46 |
| $\text{Anneal}_{\text{TD}}$ | **84.70** ±0.15 | 73.92 ±0.11 | 60.95 ±0.40 | 86.47 ±0.64 |
| $\text{AnnealVar}_{\text{TD}}$ | 84.52 ±0.27 | **74.66** ±0.06 | **61.68** ±0.51 | 86.14 ±0.23 |
| $\text{Comp}_{\text{TD}}$ | 84.51 ±0.45 | 74.32 ±0.41 | 61.09 ±0.28 | 86.30 ±0.70 |
| $\text{CompVar}_{\text{TD}}$ | 84.03 ±0.65 | 74.43 ±0.18 | 61.04 ±0.31 | 85.78 ±0.74 |
| $\text{Anneal}_{\text{CR}}$ | 84.35 ±0.46 | 74.57 ±0.40 | 60.44 ±0.39 | 86.59 ±0.29 |

(a) Zero-shot cross-lingual transfer performance of XLM-R models between curricula as the average accuracy across languages.

| TRAIN<br>TEST | PAWS-X<br>TWITTERPPDB | XNLI<br>NLI DIAG. | SIQA<br>CSQA |
|---|---|---|---|
| Prior Work | - | - | - |
| Random | 72.80 ±5.45 | 61.87 ±1.36 | 44.61 ±0.96 |
| $\text{Anneal}_{\text{TD}}$ | 71.97 ±2.69 | 62.15 ±0.94 | 45.81 ±1.40 |
| $\text{AnnealVar}_{\text{TD}}$ | 72.62 ±1.17 | **62.57** ±1.32 | 44.31 ±0.88 |
| $\text{Comp}_{\text{TD}}$ | 75.18 ±6.71 | 61.31 ±1.00 | 43.93 ±1.59 |
| $\text{CompVar}_{\text{TD}}$ | **81.33** ±2.10 | 61.82 ±0.98 | **45.84** ±0.67 |
| $\text{Anneal}_{\text{CR}}$ | 72.83 ±6.65 | 61.78 ±0.27 | 44.85 ±0.72 |

(b) Zero-shot accuracy results of RoBERTa models on out-of-distribution (OOD) data.

Table 3: Test set accuracies on cross-lingual and monolingual settings with and without CL. Mean and standard deviation across 3 random seeds. We also report prior work results for reference as follows: PAWS-X (Chi et al., 2021), XNLI (Chi et al., 2021), XCOPA (Ponti et al., 2020), MLDoc (Keung et al., 2020) (mBERT). *Note that Chi et al. (2021) tune on the target languages validation sets.

ples adequately, compared to training dynamics that instead result in smooth training.

Regarding the continuous schedulers ($\text{Comp}_{\text{TD}}$ and $\text{CompVar}_{\text{TD}}$) we observe that in the largest dataset (XNLI) after a certain point $\text{CompVar}_{\text{TD}}$ is able to surpass random training (steps 70K-120K), despite having an initial performance much lower than the other schedulers. In addition, on SIQA it is superior to other schedulers by consistently improving performance for almost half of training (from step 8K and after) as well as obtaining higher performance faster compared to $\text{Comp}_{\text{TD}}$ that does not employ variability sampling.

### 5.2 Cross-lingual & OOD Performance

In addition to the speedup offered by CL and the observations from the learning curves, we test for potential improvements in test set performance. Table 3 shows accuracies for both multilingual and monolingual models when tested for zero-shot cross-lingual transfer or OOD data.

Initially we observe that CL with XLM-R seems to have a larger impact in terms of performance primarily on XNLI and XCOPA, gaining $0.73$ and $1.06$ points respectively with the $\text{AnnealVar}_{\text{TD}}$ curriculum. As for the remaining datasets, CL is unable to achieve any performance improvement on MLDoc (as also shown in Figure 2) while on PAWS-X it has incremental improvement of $0.2$ points with the cost of no speedup[3]. Other schedulers can offer smaller performance improvement but higher speedup, e.g. in the case of XCOPA with $+0.42$ points and $87\%$ speedup in the base case with $\text{CompVar}_{\text{TD}}$. Finally, comparing with the *Cross-Review* method, we observe that performance is on par with other curricula, however it cannot surpass our proposed variability sampling. As another drawback, it is more resource demanding since it needs $N$ teacher models instead of 1.

To evaluate OOD generalisation we test a RoBERTa model with and without CL on OOD data. Table 3b shows zero-shot accuracies on different OOD datasets. The behavior of CL in these cases is not as consistent as in zero-shot cross-lingual transfer, where $\text{CompVar}_{\text{TD}}$ achieves the best performance on TwitterPPDB ($+8.5$ points) and CommonSenseQA ($+1.23$ points) while $\text{AnnealVar}_{\text{TD}}$ performs best for NLI Diagnostics ($+0.7$ points). We speculate that $\text{CompVar}_{\text{TD}}$ achieves higher OOD performance thanks to its slow pacing learning that trains models adequately on easy and ambiguous examples before moving on to harder ones, something that is crucial for OOD generalisation as also noted by Swayamdipta et al. (2020). This though comes at the cost of speedup by requiring another $50\%$ of training steps.

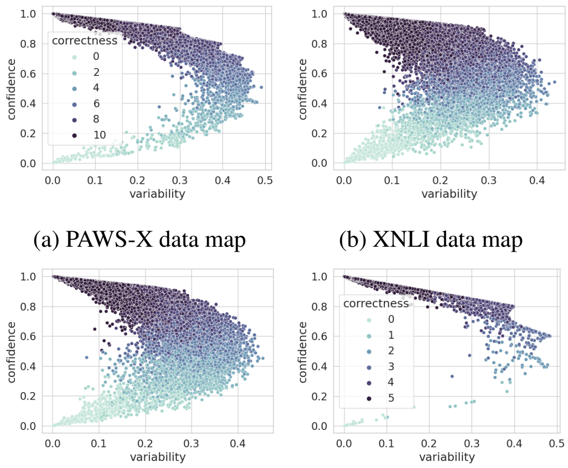
### 5.3 Training with limited budget

Since training a teacher model can add overhead to the general training process (training a teacher model plus a similar-sized student), we further conduct a minimal experiment on PAWS-X, where we collect training dynamics for a teacher $\text{XLM-R}_{\text{base}}$ model for different number of epochs (stopping training early) and then train a student $\text{XLM-R}_{\text{base}}$ model for 10 epochs. Results are reported in Table 4 for standard random training as well as for our best overall curriculum $\text{AnnealVar}_{\text{TD}}$ as the aver-

[3]We report complete tables with one-to-one association between performance and speedup in Appendix C.

| Teacher Epochs | Random | AnnealVar$_{TD}$ | Time ↓ |
|---|---|---|---|
| 3 | 85.28 ± 0.18 | 85.20 ± 0.17 | 0.88 (0.51) |
| 4 | | 85.46 ± 0.25 | 0.98 (0.64) |
| 5 | | 84.94 ± 0.30 | 0.90 (0.70) |
| 10 | | 85.34 ± 0.19 | 0.76 (0.52) |

Table 4: Development set performance (average across languages) on PAWS-X with XLM-R teacher and student. Student is trained for 10 epochs, while training dynamics are collected from the teacher for different number of epochs. Time for the Random setting is 1.0.




(a) PAWS-X data map (b) XNLI data map

(c) SIQA data map (d) MLDoc data map

Figure 3: Data map for the training set of each dataset using an XLM-R$_{base}$ model. We plot maximum 25K examples for clarity. For the first 3 datasets (3a)-(3c) correctness obtains values in [0, 10].

age of the development set languages performance.

We observe that it is not actually necessary to collect training dynamics for a long period of training (e.g. 10 epochs) as even with much less training, for instance just 3 epochs, we can still get close performance to the random order baseline for 12% speedup on average and almost 50% in the best case. This adds minimal overhead to training, suitable when one wants to train with a limited budget. Compared to Cross-Review, that essentially requires full training of $N$ teacher models plus the student model, TD offer a much more efficient solution. Ultimately, even having less accurate dynamics (by training the teacher for less epochs) we can achieve a small speedup on the student model and result in overall less training time for both models. Longer teacher training might be proven beneficial for future training of different student versions.

### 5.4 Analysing Data Maps

Finally, to better understand the reason for the reported CL benefits we plot data maps that result from training an XLM-R model on each dataset in Figure 3, with confidence in the y-axis, variability in the x-axis and correctness in the legend. As observed, the easiest overall datasets, i.e. PAWS-X (3a) and MLDoc (3d) result in quite crisp maps with very few hard-to-learn examples, while in XNLI (3b) and SIQA (3c) the data maps are very dense and the number of difficult examples is high. This can potentially explain why CL with XLM-R models was more beneficial on those datasets in terms of performance, confirming that CL can be used to better prepare a model for harder instances.

## 6 Conclusion

We presented a set of experiments using training dynamics (Swayamdipta et al., 2020) as difficulty metrics for CL on (X)NLU tasks. Differently from existing works, we focus our evaluation on zero-shot cross-lingual transfer and OOD data–testing existing discrete and continuous schedulers as well as modifications of those in a transfer-teacher curriculum setting.

Our findings on four cross-lingual datasets offer evidence that simply reordering the training examples in a meaningful way can have an impact on both zero-shot cross-lingual transfer and OOD data. In particular, we found that datasets without a clear distinction between training instances in data maps are mostly benefited from CL, with speedup improvements up to 58%, while others have incremental improvements in zero-shot cross-lingual transfer. Our proposed Continuous scheduler with variability sampling provided a boost up to 8% on a challenging OOD dataset potentially thanks to its slow pacing learning. Comparing our proposed application of training dynamics to other transfer-teacher curriculum methods that are using more than 1 teacher model, we observed greater speedups, effeciency and more stable training.

Overall, our experiments suggest there is no curriculum outperforming others by a large margin which is consistent with findings in Zhang et al. (2018). However we show that training dynamics are potentially better difficulty metrics for CL in both monolingual and multilingual models, easily obtained by fine-tuning a single teacher model for a minimal number of epochs.

## References


Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. 2021. XLM-E: Cross-lingual language model pre-training via electra. *arXiv preprint arXiv:2106.16138*.

Monojit Choudhury, Kalika Bali, Sunayana Sitaram, and Ashutosh Baheti. 2017. Curriculum design for code-switching: Experiments with language identification and language modeling with deep neural networks. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 65–74, Kolkata, India. NLP Association of India.

Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

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

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 389–398, Hong Kong, China. Association for Computational Linguistics.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2694–2700. AAAI Press.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. Don't use English dev: On the zero-shot cross-lingual evaluation of contextual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554, Online. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

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

Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

John P. Lalor and Hong Yu. 2020. Dynamic data selection for curriculum learning via ability estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 545–555, Online. Association for Computational Linguistics.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-shot cross-lingual transfer with meta learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4547–4562, Online. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Dana Ruiter, Josef van Genabith, and Cristina España-Bonet. 2020. Self-induced curriculum learning in self-supervised neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2560–2571, Online. Association for Computational Linguistics.

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 453–463, Berlin, Germany. Association for Computational Linguistics.

Atsushi Saito. 2018. Curriculum learning based on reward sparseness for deep reinforcement learning of task completion dialogue management. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 46–51, Brussels, Belgium. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Lei Shen and Yang Feng. 2020. CDL: Curriculum dual learning for emotion-controllable response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 556–566, Online. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2019. Improving anaphora resolution in neural machine translation using curriculum learning. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 140–150, Dublin, Ireland. European Association for Machine Translation.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When do curricula work? In *International Conference on Learning Representations*.

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

Chen Xu, Bojie Hu, Yufan Jiang, Kai Feng, Zeyang Wang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2020b. Dynamic curriculum learning for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3977–3989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Mingyang Yi, Lu Hou, Jiacheng Sun, Lifeng Shang, Xin Jiang, Qun Liu, and Zhi-Ming Ma. 2021. Improved ood generalization via adversarial training and pretraining. *arXiv preprint arXiv:2105.11144*.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat.

2018. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019a. 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, Volume 1 (Long and Short Papers)*, pages 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

## A Datasets

In this study, we use the following datasets:

**PAWS-X** (Yang et al., 2019) is the cross-lingual version of the English Paraphrase Adversaries from Word Scrambling dataset (Zhang et al., 2019b) containing paraphrase identification pairs from Wikipedia. It consists of human translated pairs in six topologically distinct languages. The training set contains only English examples taken from the original PAWS dataset. As OOD we use the TwitterPPDB dataset (Lan et al., 2017).

**XNLI** is the cross-lingual NLI dataset (Conneau et al., 2018), an evaluation set created by extending the development and test sets of the MultiNLI dataset (Williams et al., 2018) and translating it into 14 languages. Training data constitutes the original MultiNLI English training set. A OOD we use NLI Diagnostics (Wang et al., 2018), a set of human-annotated examples that reveal model behavior on particular semantic phenomena.

**XCOPA** is the Cross-lingual Choice of Plausible Alternatives (Ponti et al., 2020), a typologically diverse multilingual dataset for causal common sense reasoning in 11 languages. The dataset consists of development and test examples for each language, which are translations from the English COPA (Roemmele et al., 2011) validation and test sets. Following Ponti et al. (2020) we use the Social IQA dataset (Sap et al., 2019) as training data (containing 3 possible choices), and the English COPA development set as validation data (containing 2 possible choices). For OOD, we consider the CommonSenseQA (CSQA) dataset (Talmor et al., 2019) that contains 5 possible choices.

| | RoBERTa$_{\text{base}}$ | XLM-R$_{base}$ |
|---|---|---|
| XNLI | 7.5 h | 11.5 h |
| PAWS-X | 1.0 h | 1.8 h |
| SIQA | 1.0 h | 1.3 h |
| MLDoc | - | 1.0 h |

Table 5: Training time required for a full model training.

**MLDoc** is a document classification dataset with 4 target categories: corporate/industrial, economics, government/social, and markets (Schwenk and Li, 2018). The dataset is an improved version of the Reuters benchmark (Klementiev et al., 2012) consisting of 7 languages and comes with 4 different sets of English training data (1k, 2k, 5k, 10k). Here, we use the 10k following prior work (Keung et al., 2020).

## B Training Details

**Hyper-parameter Settings**: For all the reported experiments we used the HuggingFace Transformers library with PyTorch[4]. We use base models, XLM-R and RoBERTa with 470M and 340M parameters respectively. We fix sentence length to 128 for all datasets except MLDoc where we use 256. We did minimal learning rate tuning on each dataset's English validation set, searching among [7e-6, 1e-5, 2e-5, 3e-5] and choosing the best performing one (1e-5 for PAWS-X, 7e-6 for SIQA and XNLI, 3e-5 for MLDoc). We clip gradients to 1.0 after each update, use AdamW optimizer (Loshchilov and Hutter, 2017) without any warmup and a batch size of 32 for PAWS-X, XNLI and MLDoc and 8 for SIQA/XCOPA. All reported experiments use the same 3 random seeds and all models were trained on a single Nvidia V100 16GB GPU. In terms of training time, Table 5 shows the training time required for each dataset with the above parameters.

**Multiple Choice QA**: We treat SIQA-XCOPA as a sentence-pair classification task and feed the model

[4]https://pytorch.org/

a (premise-question, choice) tuple converting each *cause* into “What was the cause?" and each *effect* into “What was the effect?" question which is concatenated to the premise. Similar to prior work (Ponti et al., 2020) we use a feed forward linear layer on top of the input's first special token (<s> in the case of RoBERTa and XLM-R) to produce a score for each of the possible choices. In the case of CSQA that does not have a premise, we simply feed the network the question-choice pair.

## C Detailed Results

In Tables 6 and 7 we report detailed results with test set accuracy and time speedup for each curriculum on zero-shot cross-lingual transfer and OOD generalisation, respectively.

| | PAWS-X | | XNLI | | XCOPA | | MLDoc | |
|---|---|---|---|---|---|---|---|---|
| | Test | *Time* ↓ | Test | *Time* ↓ | Test | *Time* ↓ | Test | *Time* ↓ |
| Prior Work | 84.90* | - | 75.00* | - | 60.72 | - | 77.66 | - |
| Random | 84.49 ±0.08 | 1.00 | 73.93 ±0.18 | 1.00 | 60.62 ±0.54 | 1.00 | **86.74** ±0.46 | 1.00 |
| $\text{Anneal}_{\text{TD}}$ | **84.70** ±0.15 | 1.04 (0.70) | 73.92 ±0.11 | 1.12 (0.94) | 60.95 ±0.40 | 0.80 (0.38) | 86.47 ±0.64 | 0.91 (0.81) |
| $\text{AnnealVar}_{\text{TD}}$ | 84.52 ±0.27 | **0.76** (**0.51**) | **74.66** ±0.06 | **0.78** (**0.43**) | **61.68** ±0.51 | 1.14 (0.38) | 86.14 ±0.23 | **0.81** (**0.42**) |
| $\text{Comp}_{\text{TD}}$ | 84.51 ±0.45 | 1.43 (1.03) | 74.32 ±0.41 | 1.15 (0.46) | 61.09 ±0.28 | 0.49 (0.32) | 86.30 ±0.70 | 1.12 (1.03) |
| $\text{CompVar}_{\text{TD}}$ | 84.03 ±0.65 | 1.47 (0.94) | 74.43 ±0.18 | 1.18 (0.93) | 61.04 ±0.31 | 0.56 (**0.13**) | 85.78 ±0.74 | 0.99 (0.71) |
| $\text{Anneal}_{\text{CR}}$ | 84.35 ±0.46 | 1.08 (0.65) | 74.57 ±0.40 | 1.02 (0.86) | 60.44 ±0.39 | **0.39** (0.22) | 86.59 ±0.29 | 0.82 (0.74) |

Table 6: Zero-shot performance between curricula as the average accuracy across languages (mean and standard deviation over 3 random seeds). *Time* corresponds to the ratio $N_{\text{curric}}/N_{\text{random}}$, where the numerator is the number steps a curriculum needs to reach the reported performance and the denominator is the number of steps the Random training baseline requires to reach its performance. The value in parentheses corresponds to the minimum time across seeds (lower is better). All curricula use XLM-$\text{R}_{\text{base}}$ as the underlying model. We also report prior work results for reference as follows: PAWS-X (Chi et al., 2021), XNLI (Chi et al., 2021), XCOPA (Ponti et al., 2020), MLDoc (Keung et al., 2020) (mBERT). *Note that Chi et al. (2021) tune on the target languages validation sets.

| Train (ID)<br>Test (OOD) | PAWS-X<br>TwitterPPDB | *Time* ↓ | XNLI<br>NLI Diag. | *Time* ↓ | SIQA<br>CSQA | *Time* ↓ |
|---|---|---|---|---|---|---|
| Random | 72.80 ±5.45 | 1.00 | 61.87 ±1.36 | 1.00 | 44.61 ±0.96 | 1.00 |
| $\text{Anneal}_{\text{TD}}$ | 71.97 ±2.69 | **0.79** (**0.63**) | 62.15 ±0.94 | **0.87** (**0.51**) | 45.81 ±1.40 | 0.85 (0.68) |
| $\text{AnnealVar}_{\text{TD}}$ | 72.62 ±1.17 | 0.97 (0.64) | **62.57** ±1.32 | 1.61 (1.34) | 44.31 ±0.88 | **0.44** (**0.23**) |
| $\text{Comp}_{\text{TD}}$ | 75.18 ±6.71 | 1.71 (0.58) | 61.31 ±1.00 | 1.32 (1.11) | 43.93 ±1.59 | 0.79 (0.31) |
| $\text{CompVar}_{\text{TD}}$ | **81.33** ±2.10 | 1.64 (1.51) | 61.82 ±0.98 | 1.47 (1.33) | **45.84** ±0.67 | 0.92 (0.61) |
| $\text{Anneal}_{\text{CR}}$ | 72.83 ±6.65 | 1.56 (0.89) | 61.78 ±0.27 | 1.31 (0.63) | 44.85 ±0.72 | 0.69 (0.55) |

Table 7: Zero-shot accuracy results of monolingual models on out-of-distribution (OOD) data. All curricula use $\text{RoBERTa}_{\text{base}}$ as the underlying model. *Time* corresponds to the ratio $N_{\text{curric}}/N_{\text{random}}$ with $N$ being the number of steps a model achieves the reported performance. Results are reported over 3 random seeds and in parenthesis we include the minimum time required across these seeds.