Teaching How to Learn to Learn

Teacher-Student Curriculum Learning for Efficient Meta-Learning

Bertold B. Kovács[1]

Supervisors: Matthijs Spaan[1]**, Joery de Vries**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Bertold B. Kovács
Final project course: CSE3000 Research Project
Thesis committee: Matthijs Spaan, Joery de Vries, Pradeep Murukannaiah

Abstract

We investigate whether a teacher-student curriculum learning approach using a teacher network with a simpler structure than the student network can achieve better results at meta-learning. The goal of meta-learning is to learn from a set of tasks, and then perform well on a new, structurally similar but unseen task with minimal retraining. Instead of sampling uniformly from all data to create the training batches, the curriculum-learning approach aims to create a sequence of mini-batches that enhances the training process, also known as a curriculum. During teacher-student curriculum learning a ”teacher” network is trained in the standard manner, and then its outputs are used to order the training samples by difficulty and categorise them into mini-batches. This curriculum is then used to train the ”student” network. Previous teacher-student models either had pre-trained more complex teachers, or teachers with the same structure as the student network. We investigate whether a teacher network with a simpler structure can also increase accuracy, while preserving computational resources. We find that using such a curriculum worsens performance compared to not using any curriculum at all.

1 Introduction

When encountering a new task, which is structurally similar to one that has been solved before - such as writing down a word n times, once you already know how to write it once or twice - human intelligence performs well with minimal extra learning (Finn et al., 2017a). The formalized version of this problem is known as ”meta-learning” - learning the structure across several tasks, and after minimal retraining, performing well at a new task that has not been encountered before. This is a problem that artificial agents struggle with to this day.

A promising strategy for speeding up meta-learning is known as curriculum-learning. Instead of sampling uniformly from all tasks, a model using curriculum-learning learns from tasks in a specific order - also known as a curriculum - similar to how a human child might start with learning easier concepts, before understanding more intricate ones (Wu, 1999). Curriculum-learning has been used to achieve better results across diverse domains in machine learning from object detection (Wang et al., 2018) and language translation (Platanios et al., 2019) to reinforcement learning (Mehta et al., 2020).

There have been several attempts at incorporating curriculum-learning in meta-learning in recent years. CubMeta (Zhang et al., 2021) and SepMeta (Zhang et al., 2022) are both meta-learning architectures that use curriculum-learning. CHAML is a curriculum-learning enhanced meta-learning architecture, which was specifically built for solving the task of ”next Point of Interest” recommendation (Chen et al., 2021). Google DeepMind also successfully used curriculum learning for meta-reinforcement learning with their model AdA (Bauer et al., 2023) While these architectures show that using curriculum-learning can make meta-learning more efficient in specific scenarios, it still is not clear which concrete curriculum-learning technique might provide the best performance or if there is even a significant difference.

There are several families of approaches towards curriculum-learning. Most prominently curriculum-learning techniques can be classified as predefined curriculum, self-paced learning, teacher-student curriculum learning, and RL teacher methods. (Wang et al., 2021) As Wang et al. (2021) note, predefined curriculum is not well fitting for a more general-theoretical investigation, as it aims to exploit expert knowledge for a specific domain. A noted drawback of RL teacher methods is that they are computationally far more expensive than the alternatives. Self-paced learning aims to solve the problem without expert knowledge, and uses the structure of the model during training to score the difficulty of data points. Two significant weaknesses arise from this: first, as the structure constantly changes during training, the scores also have to be recomputed, the curriculum reordered, which has a significant computational cost. Second, the initial curriculum can be far from accurate, as the initial model is also far from correct.

To overcome these weaknesses, we use a teacher-student approach, illustrated in figure 1. This technique first trains a learner without curriculum-learning, then utilizes the results of this learner to better understand which task is better to learn first - just like how a human teacher might aid a human pupil in designing a curriculum.

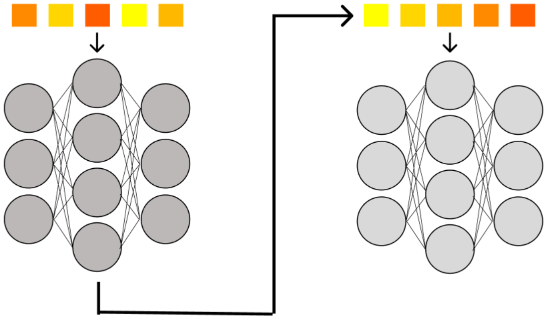

Figure 1: An illustration of the teacher-student strategy. The teacher network on the left is trained without curriculum, and then the ordering of the data points (the curriculum) is created using its outputs. The student network on the right is trained using this ordering.

The success of a model using teacher-student curriculum learning depends on the details of the teacher. Previous teacher-student curricula either used more complex networks pretrained on a more extensive dataset, also known as *transfer teachers*, or networks with the same structure as the student trained on the very same dataset, known as *bootstrapping CL* (Hacohen and Weinshall, 2019). Our hypothesis is that similar results can be achieved by using smaller networks as teachers, which also speeds up training. Thus the research question that we will investigate is: *Can a teacher-student* *curriculum using a less complex teacher structure than the student network improve the accuracy of meta-learning?*

Our contributions are the following: we propose a teacher-student curriculum learning approach for achieving better results in meta-learning. We implement this approach for the neural processes meta-learning algorithm, and evaluate its performance against different metrics.

2 Background

We aim to tackle the problem of meta-learning by using the neural processes algorithm (Garnelo et al., 2018b), and enhancing its performance with a teacher-student curriculum strategy (Hacohen and Weinshall, 2019). In this section, we first explain the abstract problem setup of meta-learning, then describe the neural process model, and finally look at the general abstract setup of a curriculum strategy.

2.1 Meta-Learning Problem Setup

Meta-learning aims to train a model, that is able to - using minimal additional training samples and with minimal re-training - perform well on unseen tasks. Consider a model $M : \mathcal{X} \to \mathcal{A}$, which maps observations in the set $\mathcal{X}$ to predictions in the set $\mathcal{A}$. This model is meta-trained over a set of tasks S, with the goal of being able to adapt well to unseen tasks in this set. In the current paper, we only consider meta-learning for supervised learning, and not for reinforcement learning, each task can be formalized in a simple manner: a task $T = \langle L, q \rangle$ where $L : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the loss function, $q \in \mathbb{P}(\mathcal{X})$ is the distribution for observations.

The model is trained so that it can learn a new task T_i sampled from a distribution $p(T)$ from seeing K samples drawn from q_i and seeing the corresponding feedback got from L_{T_i}. Following the meta-training phase, the meta-testing is done on new tasks sampled from $p(T)$.

2.2 Neural Processes for Meta-Learning

The meta-learning algorithm we use is a Neural Process. Neural Processes combine neural networks and Gaussian processes - most importantly, they are able to estimate uncertainty in predictions (like Gaussian processes) but are computationally efficient (like neural networks) (Garnelo et al., 2018b). Neural Processes are able to meta-learn a wide variety of tasks, amongst them 1-D regression, which is relevant for our paper (Garnelo et al., 2018a). Neural Processes are conditioned on context points, and then generate predictions for target points.

The architecture of the Neural Process model consists of three core components:

1. An **encoder** h, parameterised as a neural network. It takes in pairs of context values $(x,\ y)_i$ and produces a representation r_i for each of them.
2. An **aggregator** a which summarizes the inputs encoded by h. It does so by taking their mean: $\sum_{i=1}^{n} r_i$ This is used as parameterisation of the latent distribution z.
3. A **conditional decoder**, which takes as input the latent distribution z as well as the target points, and outputs the predictions.

The network is trained over functions sampled from S. For each function, our dataset has a set of $(x,\ y)_i$ tuples, which during the training are separated into context and target points. The loss function that we aim to minimize during training is the negative evidence lower bound, or negative ELBO (in other words, we are trying to maximize ELBO). The ELBO provides a lower bound to the log likelihood, and is defined the following way:

$$\log p(y_{1:n} \mid x_{1:n}) \geq \mathbb{E}_{q(z|x_{1:n},y_{1:n})} \Bigg[\sum_{i=1}^{n} \log p(y_i \mid z, x_i) + \log \frac{p(z)}{q(z \mid x_{1:n}, y_{1:n})} \Bigg] \tag{1}$$

2.3 Curriculum Learning

Curriculum strategy is an approach for enhancing learning, where instead of uniformly sampling from all training samples, the training samples are used for training in an order that aims to maximize learning. Usually this means using "simpler" training samples first, and leaving more complicated ones for later, by some measure of difficulty.

For the intuition behind curriculum strategy, we can imagine a human pupil learning about a complicated subject. We would first teach them simple examples, and only later on delve into "edge cases", or complicated outliers. We can expect the same principle to be useful in training a machine learning model, which can initially learn more on the simple examples, and later refine its weights on the outliers.

The general curriculum learning algorithm takes as input a *pacing function*, a *scoring function* and the training data $\mathbb{X}$, as shown in the figure Algorithm 1. The scoring function is some function $f : \mathbb{X} \to \mathbb{R}$, where the assigned real number denotes the difficulty, the greater the number, the more difficult we find the given example. The pacing function is some function $g : [M] \to [N]$ which determines a sequence of subsets of the data $\mathbb{X}$, from which we uniformly sample the data to form the training batches. After taking these two functions as input, we use f to order $\mathbb{X}$, and then, using g we sample M batches. These are the training batches that our learning algorithm will receive during the training.

Based on what the pacing and scoring functions look like, several main approaches have been identified for curriculum learning: predefined curriculum, self-paced learning, teacher-student, and RL teacher (Wang et al., 2021). In this paper, we use the teacher-student approach to enhance meta-learning.

Curriculum design for human education is studied by psychologists and other social scientists. The classical method of designing a curriculum for human learners involves asking an expert - such as academic faculty - who is familiar with the domain to plan the order of the educational materials in a way that they most facilitate the learning process. (O'Neill, 2010) This is also the intuition behind the teacher-student approach: to understand how difficult a given data point is, we should look at how well a trained network performs on it. This gives us a good curriculum right from the moment when we start training the student network (as opposed to self-paced learning, where the initial curriculum might be less accurate) with-

Algorithm 1 General Curriculum Learning Algorithm (Hacohen and Weinshall, 2019)

```
Input: pacing function $g$, scoring function $f$, data $X$
Output: sequence of mini-batches $[B'_1, \dots, B'_M]$
Sort $X$ according to $f$, in ascending order
Initialize $result \leftarrow []$
for $i = 1$ to $M$ do
    $size \leftarrow g(i)$
    $X'_i \leftarrow X[1, \dots, size]$
    Uniformly sample $B'_i$ from $X'_i$
    Append $B'_i$ to $result$
end for
return $result$
```

out requiring any human domain knowledge (as opposed to predefined curriculum).

A weakness of this approach is that it requires the training of an additional network, the teacher network. Two ways can be considered to mitigate this burden of extra training of the teacher: applying pretrained teachers for transfer learning, or speeding up the training of the teacher by making the network smaller. The first approach is also called *transfer teacher*, and has been used by Weinshall et al. (2018). If we have access to an already trained network, which has been trained in the past for a different purpose, we do not burden ourselves with extra training. While this might be a practical approach for real-life applications, it also has the trivial downside that it is dependent on an already existing, relevant network to exist. It would be great if we could come up with an approach that can be used for an arbitrary dataset. Hacohen and Weinshall (2019) use an approach called *bootstrap CL* where the teacher has the same structure as the student. We investigate whether this can be continued - could we also find a useful curriculum with a smaller teacher network than the student?

3 Teacher-Student Curriculum Learning

We use curriculum learning with the teacher-student approach to enhance our training for the meta-learning problem. This approach is based on using two neural networks during the training: the *teacher* and the *student*. The *teacher* network is trained without any curriculum, and used to calculate the difficulties for the training sample. The *student* network is trained using a curriculum based on these difficulties.

As mentioned in subsection 2.3 the curriculum learning takes as input a *pacing function* and a *scoring function* in addition to the data. In the case of the teacher-student approach, to get the scoring function we take the trained teacher network, and use it to compute a difficulty score for each input.

We use a smaller, simpler version of the same network as the teacher network. Previous work only investigated approaches where the teacher is at least as large as the student. Larger teachers are called *transfer teacher*, such as the one used by Weinshall et al. (2018), where a more complex network pretrained on a more extensive dataset is used as the teacher. Teachers with the same structure as the student are called *bootstrap CL*, such as in Hacohen and Weinshall (2019) which is pretrained on the same dataset. These approaches have some clear drawbacks: we need to have these pretrained networks, and in the case of the first, the larger dataset. We hypothesize that a smaller teacher will already be able to increase accuracy, while using less computational resources - a sort of *mini bootstrap CL*. More precisely, we have two hypotheses:

Hypothesis 1 (H1): *We can improve the accuracy of our model using a teacher-student curriculum, where the teacher has the same structure as our student model, and is trained on the same dataset.*

Hypothesis 2 (H2): *We can improve the accuracy of our model using a teacher-student curriculum, where the teacher has a less complex structure than our student model and is trained on the same dataset. This accuracy increase should be similar in degree to the accuracy increase in the case of the teacher and the student sharing the same structure.*

As the scoring function, we simply take the ELBO of the teacher network on the samples. This is the same as the loss function of the main model, which is the dominant difficulty measure used in the teacher-student curriculum learning literature (Wang et al., 2021).

Hacohen and Weinshall (2019) propose *varied exponential pacing* as a general form of pacing functions, and investigate *single step pacing*, *fixed exponential pacing* as special cases of it. As the success of general varied exponential pacing highly differs based on its hyperparameters, we only look at a representative from the two specific cases. In this paper, we will call them *single step pacing* and *multi step pacing*, and we aim to evaluate which one works better for our method.

To make it more precise, the single step pacing function is defined as: $g(i) = \begin{cases} p & \text{if } i < \textit{step\_length} \\ 1 & \text{if } i \geq \textit{step\_length} \end{cases}$ where p and *step_length* are hyperparameters. p is the starting percentage of data that we sample from, and *step_length* is the step size. The multi-step pacing function has a further hyperparameter: the growth of the percentage of data that we sample from, as we only achieve sampling from the full training dataset over several steps. Both of these pacing functions are illustrated in figure 2.

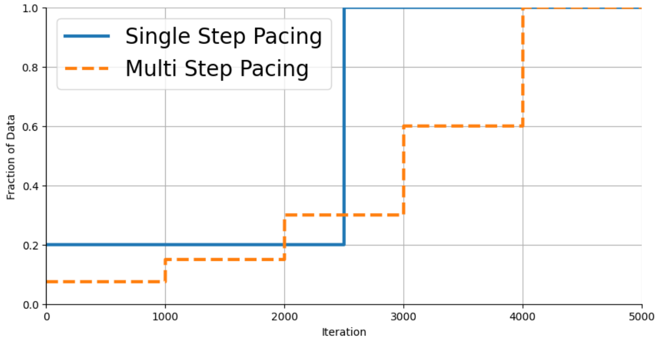

Figure 2: The pacing functions investigated in this paper.

4 Experimental Setup and Results

4.1 1-D function regression

The specific problem that we use for experiments is 1-D function regression on sinusodial functions. Each task of the multitask meta-learning problem is one function sampled from a distribution of structurally similar functions.

During training, the training losses are saved. Furthermore, the model-in-training is evaluated on additional sets of data points using the root mean squared error (RMSE) metric. These RMSE values are not used for training, but indicate how well the model is able to perform and generalize on unseen data. RMSE is defined as $\sqrt{\sum_{i=1}^{m} e_i^2}$, where e_i is the difference between the i-th prediction and the i-th target value (Hyndman and Koehler, 2006).

4.2 Results

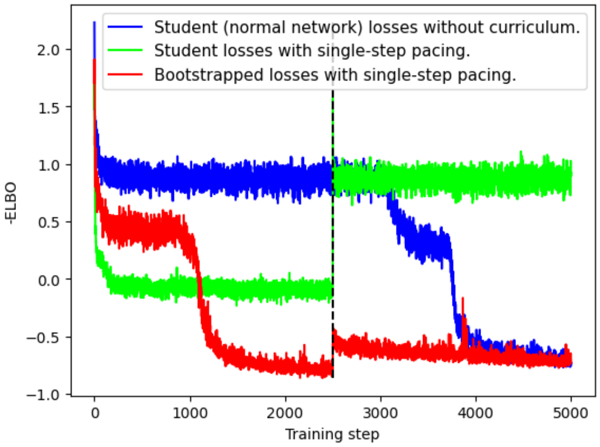

Figure 3: Negative ELBO losses during training between different curriculum setups. The dotted line indicates when the curriculum changes to include all training samples.

Figure 3 shows the negative ELBO losses during training. The curriculum-using models both show drastically better performance during the first 2500 training steps than the no-curriculum network. However, this is skewed by the fact that the set of possible samples is smaller here, because of the curriculum setup, and as shown on figure 4, this does not translate well to the full dataset for the model with the simpler teacher network. This is also demonstrated on figure 3 after the curriculum ”unlocks” the full training dataset at training step 2500. While the bootstrapped network still performs significantly better than the no-curriculum model, the model with the simpler teacher performs significantly worse.

The curriculum designed by the more complex teacher (the bootstrap) unsurprisingly performs better than the one designed by the simpler teacher. More importantly, we see that the curriculum designed using our method performs worse, while using more computational resources, than the one without any curriculum. This is somewhat in accordance with the findings of Hacohen and Weinshall (2019), who found that the *anti-curriculum*, which first teaches the difficult training samples and then progressively the easier ones decreases performance compared to no curriculum. The idea that a bad curriculum is worse than no curriculum is reinforced - and our less complex teacher network seems to not be able to learn useful insights, instead misclassifying which training samples are actually useful to learn.

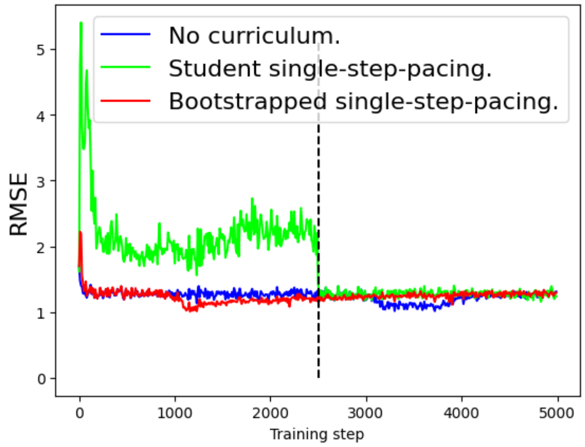

Figure 4: In-task RMSE during training. The teacher-student CL with a simple teacher performs worse than the no-curriculum and the bootstrap models.

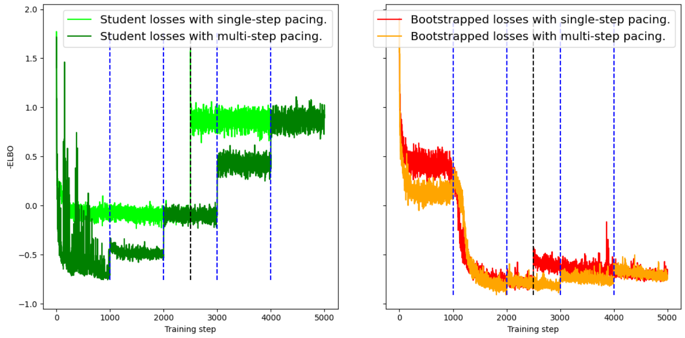

Figure 5: Comparison of pacing functions on different teachers. Vertical lines indicate when the curriculum changes. The different pacing functions lead to similar performance on both models.

Both pacing functions achieve similar performance for both the simple-teacher and the bootstrapped models. This is in line with the findings of Hacohen and Weinshall (2019). On the other plots only the models using single-step pacing function are presented for easier understanding.

In accordance with (Setlur et al., 2021), our model is evaluated both on samples from in-task and out-task distributions. However, this is tricky, as the current out-task distribution seems too similar to the in-task, as seen on the similar results on figure 4 and figure 6; but with a less similar one, we risk losing the shared structure that makes meta-learning possible.

Our findings **confirm that a bootstrapped curriculum improves the accuracy of our model (1)** in accordance with Hacohen and Weinshall (2019). However, they directly **contradict our hypothesis that a teacher network with a**

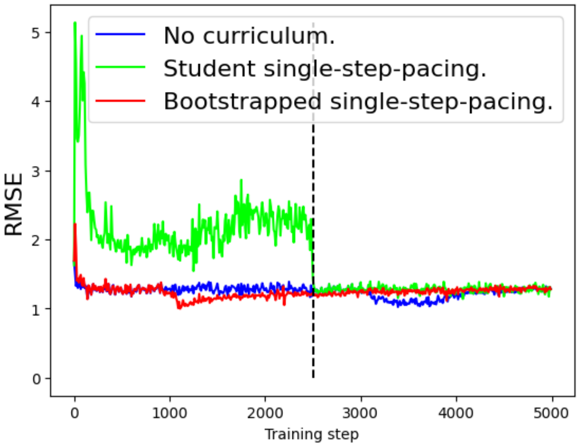

Figure 6: Out-task RMSE during training.

less complex structure can improve accuracy (2), which considering the additional computational resources used for training the teacher and creating the curriculum make this approach infeasible.

4.3 Limitations

The most important finding of our paper is that using curriculum learning with a teacher having a simpler structure has a negative effect on model performance. Two important questions arise related to this: does this also hold for *other model architectures* and *other domains*?

The basic principle of a bad curriculum being worse than no curriculum is not only trivially true on a theoretical level (unless the no curriculum is the worst possible ordering of the data, we can theoretically create a curriculum that will make the model perform worse), it has also been empirically demonstrated previously by Hacohen and Weinshall (2019). Therefore as long as a simpler model is unable to learn enough to create a curriculum that is as good as a random ordering, the finding will hold for other model architectures. However, we do not see any specific reason why the line would be specifically drawn at the complexity of the student - it is possible that for really complex models, even a somewhat simpler teacher will already be able to learn from the data well enough to provide a useful curriculum.

Our results do not use any special feature of 1-D function regression with sinusodial functions that would uniquely prompt this behaviour. Therefore we expect this behaviour to hold generally in other domains as well.

5 Responsible Research

This research is a theoretical investigation, using only synthetic, non-real life data. Consequently, most common integrity concerns that could make research unethical, such as mistreatment of the people participating in experiments, do not arise. However, it is important to consider the possible applications of our results. Even though this research is theoretical, theories are often applied to the real world. While the exact degree of responsibility a researcher has for the practical use of their research is debated, it is clear that to some degree it is present, especially in research related to engineering (Forge, 2004). However, the domain of this paper - making meta-learning more efficient - is not obviously unethical, to the contrary, once used in practical applications it can benefit society. The primary uses of meta-learning lie in robotics (Finn et al., 2017b) (Kaushik et al., 2020) and personalized federated learning (Fallah et al., 2020), which could both provide meaningful improvements in the quality of life of many people.

Furthermore, the work keeps itself to foundational scientific guidelines and frameworks, paying specific attention to the Netherlands Code of Conduct (KNAW, 2018). The Netherlands Code of Conduct lists five principles as the basis of integrity in research: honesty, scrupulousness, transparency, independence and responsibility (KNAW, 2018). This research takes **honesty** seriously, by being critical about our proposed method of improving meta-learning, and also being clear about the limitations of our experiments. **Scrupulousness** and **transparency** are present with the used data, results and used code. This is not particularly challenging in this case, as our data is synthetically generated, not collected from the outer world. The study is also **independent** in the sense that it is not guided by any non-scholarly considerations (such as commercial interests). Finally, it **responsibly** takes into account the needs of individuals and society at large, as detailed above.

The research is highly reproducible, as all code used is shared in a public GitHub repository [1]. As the data used for the experiments is not gathered from outside sources, but synthetically generated, generating the data can also be repeated by other researchers, and we need not care about possible mistakes commited during data gathering.

6 Conclusions and Future Work

Our findings indicate that a curriculum learning algorithm based on a less complex teacher than the student model does not increase performance in meta-learning, it slightly decreases it, while using more computational resources. This is unlike the behaviour of models trained with a curriculum created by a teacher using the same structure as the student (also known as ”bootstrap CL”), which we also find to achieve better performance than the model with no curriculum. Both of these stay true for both single-step and multi-step pacing functions.

These findings did not completely align with our expectations, as we expected the curriculum designed by the simpler teacher to still improve upon model performance. Providing a theoretical framework that can better predict what properties a teacher needs for the teacher-student curriculum to perform better could improve upon these findings. It is also not clear what exact properties a model architecture needs in order for this phenomenon to be present - investigating different models and their features, such as in an ablation study, would be useful to better understand where teacher-student curriculum learning can provide advantages.

[1] The repository can be accessed at the following URL: https://github.com/bbkovacs/CL_teacher_student

References

Bauer, J., Baumli, K., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., Dasagi, V., Gonzalez, L., et al. (2023). Human-timescale adaptation in an open-ended task space. In *International Conference on Machine Learning*, pages 1887–1935. PMLR.

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

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2020). Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

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

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. (2017b). One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR.

Forge, J. (2004). The morality of weapons research. *Science and Engineering Ethics*, 10:531–542.

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. A. (2018a). Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. (2018b). Neural processes. *arXiv preprint arXiv:1807.01622*.

Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688.

Kaushik, R., Anne, T., and Mouret, J.-B. (2020). Fast online adaptation in robotics through meta-learning embeddings of simulated priors. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5269–5276. IEEE.

KNAW, V. (2018). Netherlands code of conduct for research integrity. Royal Netherlands Academy of Arts and Sciences. Accessed: 2023-06-17.

Mehta, B., Deleu, T., Raparthy, S. C., Pal, C. J., and Paull, L. (2020). Curriculum in gradient-based meta-reinforcement learning. *arXiv preprint arXiv:2002.07956*.

O'Neill, G. (2010). Initiating curriculum revision: exploring the practices of educational developers. *International Journal for Academic Development*, 15(1):61–71.

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

Setlur, A., Li, O., and Smith, V. (2021). Two sides of meta-learning evaluation: In vs. out of distribution. *Advances in neural information processing systems*, 34:3770–3783.

Wang, J., Wang, X., and Liu, W. (2018). Weakly-and semi-supervised faster r-cnn with curriculum learning. In *2018 24th international conference on pattern recognition (ICPR)*, pages 2416–2421. IEEE.

Wang, X., Chen, Y., and Zhu, W. (2021). A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576.

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

Wu, H. (1999). Basic skills versus conceptual understanding. *American Educator*, 23(3):14–19.

Zhang, J., Song, J., Gao, L., Liu, Y., and Shen, H. T. (2022). Progressive meta-learning with curriculum. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):5916–5930.

Zhang, J., Song, J., Yao, Y., and Gao, L. (2021). Curriculum-based meta-learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1838–1846.

A Use of LLMs

OpenAI's ChatGPT 4o model was used as a grammar-checking or "autocorrect" tool during at the final stage. It was prompted the following way:

This is the [chapter name] chapter of a scientific paper. Is it clearly written? Highlight your changes in bold.

This was done for each chapter separately, including the abstract ([chapter name] was always changed accordingly). This was then used to fix minor grammar mistakes, primarily with the use of commas.