



The Hebrew University of Jerusalem
The Rachel and Selim Benin School of Computer Science and Engineering

Augmentation Curriculum Learning

Rom Maltser

Thesis submitted in partial fulfillment of the requirements
for the Master of Sciences degree
in Computer Science

Under the supervision of **Prof. Daphna Weinshall**

January 2025



האוניברסיטה העברית בירושלים
בית הספר להנדסה ולמדעי המחשב על שם רחל וסלים בנין
החוג למדעי המחשב

תכנון סדר אוגמנטציות לאימון רשתות נוירונים

מוגש על ידי
רום מלצר

עבודת גמר לתואר מוסמך במדעי המחשב

עבודה זו הונחתה על ידי
פרופ' דפנה ויינשל

טבת תשפ"ה

תקציר

תכנון תהליך האימון של רשתות נוירונים ובפרט, בחירת סדר על הדאטא עבורו עליו תאומן הרשת, הראו שיפורים משמעותיים באחוזי הדיוק הסופי של רשתות נוירונים. בעקבות הצלחה של ניסויים אלו התפתח התכנון של בחירת האוגמנטציות לאימון הרשת והניהול שלהן במהלך תהליך האימון. אף על פי שיש לא מעט מאמרים העוסקים בנושא לא קיים פתרון המתבסס על העיקרון אשר על פיו יש לאמן מודל בצורה הדרגתית כלומר, לאמן את המודל על אוגמנטציות באופן שמתחשב בקושי שלהן כך שהמודל יחשף לאוגמנטציות קשות רק לאחר שנחשף לקלות. בניגוד לשיטות הקיימות, השיטה המוצגת במאמר מציעה לעדכן את אוסף האוגמנטציות עליו מאמנים את הרשת בצורה הדרגתית הנפרסת על גביי תהליך האימון. גישה זו מתבססת על שימוש ברשת מאמונת מראש על המידע אשר באמצעותה אנו מגדירים את דרגת הקושי של האוגמנטציות השונות. באמצעות ניסויים רבים, אנו מציגים את היכולת של האלגוריתם שלנו לתאר את דרגת הקושי בצורה מהימנה, להיות אינווריאנטי לבחירת מורה ולהגיע לביצועים טובים.

Abstract

Training neural networks traditionally involves supplying a sequence of mini-batches sampled from the training data. Additionally, to encourage better generalization, augmentations are uniformly applied to these mini-batches. In this work, we propose a novel scoring function that quantifies the difficulty of each augmentation and analyze the effect of augmentation curriculum learning. Although previous research has focused on identifying optimal augmentation policies or finding adversarial augmentations for robust training, **our work investigates how to manage the dynamics of augmentations**, and specifically how to adjust the distribution of augmentations throughout the training process. To employ the augmentation curriculum learning, the training algorithm must resolve 2 problems : (1) sort the augmentation by their difficulty (2) update the augmentation distribution through the training process. The empirical analysis, conducted on various network architectures with images from CIFAR-10, CIFAR-100, and subsets of ImageNet, demonstrates a distinct advantage of our method.

Acknowledgements

I am deeply grateful to my supervisor, Prof. Daphna Weishall, for her guidance, support and tolerance throughout this thesis.

Additionally, I am grateful to Dr. Guy Hacoheh, whose expertise greatly enriched the quality and depth of this research.

I also thank the Hebrew University for its resources and excellent staff.

I would also like to express my gratitude to the Ullman Scholarship for providing the opportunity to fully dedicate myself to this research.

I also thank the Negishut staff for consistently trying to help me balance my injury with the academy demands.

To Daphna's lab members, thank you for your valuable input.

I would like to thank my family, friends, and colleagues for their encouragement, assistance, and moral support.

To all those mentioned above and to anyone else who has contributed in any way, however small, thank you.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Curriculum learning	2
1.3	Augmentations technique	3
2	Related Work	6
2.1	Curriculum learning	6
2.2	Data augmentation strategies	7
3	Our Method-ACL	9
3.0.1	Notations and Definitions	9
3.0.2	Augmentation Curriculum Learning Method	10
3.0.3	Scoring Functions	12
4	Empirical Results	15
4.1	Experiment and results	15
4.1.1	Large augmentation set	16

<i>CONTENTS</i>	v
4.1.2 Self supervision	18
4.1.3 Robustness improvement	19
4.1.4 Baseline augmentation set Results	21
4.2 Neural net properties	22
4.2.1 Single augmentation arrangement	22
4.2.2 Baseline augmentation sort	23
4.2.3 Neural net agreement	24
4.2.4 Smart scoring vs Simple scoring	25
4.3 Scoring function optimization	29
5 Summary and Conclusion	31

List of Figures

4.1	CIFAR100 dataset, ResNet18 architecture, trained on large augmentation set, each applied at 3 different intensities, and fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.	17
4.2	Augmentation Curriculum Learning applied to various architectures and datasets with large set of augmentations, each applied at 3 different intensities. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.	18
4.3	Self-supervised Augmentation Curriculum Learning applied to various architectures. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.	19
4.4	Augmentation Curriculum Learning with various architecture and datasets, using a "baseline" augmentation set consisting of augmentations that are better suited to natural images. Bars indicate the average final accuracy, and error bars indicate the STE over 5 iterations.	21

- 4.5 CIFAR-100 dataset, Inception-based non-normalized simple score for crop augmentations at different intensity levels. The X-axis indicates crop size, and the Y-axis represents the augmentation score. 22
- 4.6 CIFAR-100 dataset, Inception-based, non-normalized simple score for: crop (green), rotation (blue), erase (red) and flip (purple) augmentations at different intensity levels. The X-axis indicates crop size, and the Y-axis represents the augmentation score. 23
- 4.7 This matrix shows the cosine similarity between score vectors calculated by different models on the CIFAR-100 dataset with the baseline augmentation set. The value at position (i, j) is the cosine similarity between the scores computed by model i and model j 24
- 4.8 Comparison of Non-Normalized Scores on CIFAR-100 Using the Inception-Based Model: Left – Smart Scoring Function, Right – Simple Scoring Function. 25
- 4.9 CIFAR100 dataset, ResNet18 architecture, "contrastive" augmentation set with fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations. 27
- 4.10 CIFAR100 dataset, ResNet18 architecture, large augmentation set with fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations. 28

List of Tables

4.1	Comparison of ResNet-18 accuracy on various types of augmented training and testing sets.	20
4.2	Final accuracy comparison on the CIFAR-100 dataset using the ResNet18 architecture. smart scoring function vs. optimized smart scoring function.	30

1 Introduction

1.1 Background and Motivation

To effectively teach complex tasks, teachers often opt to design a curriculum that organizes concepts integral to the final objective. This curriculum establishes a logical sequence, typically reflecting the increasing complexity of concepts. By introducing students to examples of gradually increased difficulty, it enables them to build upon prior knowledge, thereby facilitating the understanding and assimilation of more abstract concepts.

We adopt the basic machine learning framework, where the relevant material is the set of training examples. The first challenge requires the selection of an appropriate difficulty function in this domain. Previous work focused on measuring the difficulty of images, thus gradually increasing the set of images being learned by including more difficult data in the training set [21]. In our work, we choose a different path: instead of simply using data augmentation during training, as is customary with image data, we propose sorting the difficulty levels of the augmentations used in the training process. Learning proceeds as usual, with data randomly sampled at each step. Curriculum learning is achieved by progressively increasing the difficulty of the augmentations applied to the training data.

1.2 Curriculum learning

Training machine learning models in a meaningful order, from easy samples to hard ones, using curriculum learning can provide performance improvements over the standard training approach based on random data shuffling, without incurring additional computational costs. Curriculum learning strategies have been successfully employed across all areas of machine learning and in a wide range of tasks. Although curriculum learning approaches can be divided into different types based on the components involved, the most common way to categorize these methods is as follows:

- **Vanilla CL:** Introduced by Bengio et al. [5], improves model performance by feeding increasingly difficult samples during training, using predefined rules to distinguish between easy and hard examples, such as progressing from simple geometric shapes to complex ones or from shorter to longer sequences.
- **Self-paced learning (SPL):** Differs from Vanilla CL by dynamically determining the order of training samples based on the model's performance, continuously adjusting the difficulty during training [65].
- **Balanced CL (BCL):** This approach extends traditional Curriculum Learning (CL) by introducing multiple ordering criteria, ensuring that training samples are not only presented from easy to hard but also balanced and diverse across image regions or classes [54].
- **Self-paced CL (SPCL):** Self-paced curriculum learning (SPCL) combines predefined criteria with learning-based metrics to determine the training order of samples and has been applied to tasks like ma-

trix factorization [29], multimedia event detection, weakly-supervised object segmentation [64], and person re-identification [38].

- **Teacher-student CL:** This approach splits training into two tasks: a student model learning the main task and a teacher model determining the optimal learning parameters for the student, with the curriculum applied through a policy network guiding the student [30].
- **Implicit CL (ICL):** This approach applies a curriculum-like strategy without explicitly designing one, where the training process naturally progresses from easier to harder tasks, such as gradually increasing the complexity of convolutional [51] activation maps or reducing the number of labeled samples for classification [1].
- **Progressive CL (PCL):** Progressive Curriculum Learning (PCL) focuses on gradually increasing the model's capacity or task difficulty during training, rather than ordering samples by difficulty, with examples like Curriculum Dropout [32] and progressive growth of Generative Adversarial Networks [67].

1.3 Augmentations technique

Data augmentation is a set of techniques that generate high-quality artificial data by manipulating existing samples. By leveraging these techniques, AI models can significantly improve their performance in tasks involving scarce or imbalanced datasets, thereby substantially enhancing their generalization capabilities. The most common way to categorize the augmentation techniques is as follows:

- **Model-free:** Methods that do not rely on a predefined model architecture but rather apply general techniques to augment the data.
 - **Single-image**
 - * Geometrical transformation: translation, rotation, flip, scale [60].
 - * Intensity transformation: blurring and adding noise, Cutout [9], Random Erasing [68].
 - * Intensity transformation: Jittering [60].
 - **Multiple-image**
 - * **Non-instance-level:** SamplePairing [27], Mixup [66], BC Learning [56], CutMix [62], AugMix [24], PuzzleMix [34], Co-Mixup [33].
 - * **Instance-level:** Scale and Blend [18], Context DA [11], Simple CutPas [19], Continuous CutPas [61].
- **Model-based :** Methods that use specific models or architectures, where the augmentation process is based on the characteristics of the model itself.
 - **Unconditional:** DCGAN [16], [15].
 - **Label-conditional:** BDA [57], ImbCGAN [10], BAGAN [39], DAGAN [4].
 - **Image-conditional:** AugGAN [26], Plant-CGAN [71], StyleAug [28], Shape bias [17], EmoGAN [69], δ -encoder [48], StyleMix [25].
- **Optimizing policy-based:** Approaches that focus on optimizing a policy (either through reinforcement learning or adversarial learn-

ing) to determine how data augmentation should be performed during training.

- **Reinforcement learning-based:** AutoAugment [7], RandAugment [22], MADAO [23].
- **Adversarial learning-based:** ADA [13], CDST-DA [45], AdaTransform [55], Adversarial AA [67], IF-DA [37].

2 Related Work

2.1 Curriculum learning

Incorporating a curriculum to accelerate learning is a common strategy in both human education and animal training [36, 42, 53]. In various fields, it is standard practice to introduce concepts in increasing order of difficulty, determined by the human instructor or based on problem-specific factors [2, 41, 63]. With the resurgence of deep learning as a powerful tool across numerous applications, the use of curriculum learning (CL) to manage the sequence of examples presented to neural networks during training has gained increasing attention [14, 20].

In related work, some approaches train teacher and student networks simultaneously, where the teacher dynamically samples mini-batches for the student based on the student’s current outputs. Unlike our approach, these methods design the curriculum around the student’s evolving hypothesis, showing improvements for noisy datasets [31] or smaller datasets [12]. However, they do not demonstrate an enhanced generalization on the original dataset.

2.2 Data augmentation strategies

Data augmentation has played a central role in the training of deep vision models. The most effective augmentation strategies are dataset-specific. First, manually designed strategies were dominant. For instance, on MNIST, many of the top-performing models employ elastic distortions, scale, translation, and rotation [6, 47, 50, 58]. On natural image datasets, such as CIFAR-10 and ImageNet, random cropping, image mirroring and color shifting / whitening are more common.

Later, automated approaches for discovering data augmentation strategies emerged. These include data augmentation policies learned directly from data, such as Autoaugment [7], and random selection of augmentation operations from a predefined set, with a fixed number of transformations and magnitudes, as seen in RandAugment[8]. Later developments introduced adversarial augmentations, like AugMax[59], which focus on more challenging and distorted versions of the input data to maximize training loss.

Additionally, generative adversarial networks (GANs) have been utilized to identify optimal sequences of data augmentation operations [44]. GANs have also been employed to generate training data directly [3, 40, 43, 52, 70], though this approach has not proven to be as effective as learning predefined sequences of augmentation operations [46].

In this paper, we propose a method for discovering the difficulty of augmentations directly from the data. Our approach is inspired by recent advancements in curriculum learning, where varying input difficulty is introduced, emphasizing the importance of learning in an ordered man-

ner. In our case, we treat the entire dataset as it has equal difficulty but argue that the difficulty of the augmentations varies, requiring a different approach.

3 Our Method-ACL

The curriculum learning approach explored in this work addresses how to leverage prior knowledge about the difficulty of augmentations to sample them non-uniformly and thus boost the rate of learning and the accuracy of the final classifier. The paradigm of CL is based on the intuition that the learning process benefits when the learner is exposed to examples that progressively become more challenging.

3.0.1 Notations and Definitions

Let $X = \{X_i\}_{i=1}^N = \{(x_i, y_i)\}_{i=1}^N$ denote the data, where $x_i \in \mathbb{R}^d$ denotes a single data point and $y_i \in [K]$ its corresponding label. Let $A = \{A_i\}_{i=1}^M$ denote the set of augmentations where A_i denotes the i 'th augmentation. Let $h_\theta : \mathbb{R}^d \rightarrow [K]$ denote a target classifier, and mini-batch $B \subseteq X$ denote a subset of X . In the most common training procedure, which is a robust variant of Stochastic Gradient Descent (SGD), h_θ is trained sequentially when given as input a sequence of mini-batches $[B_1, \dots, B_I]$ Shalev-Shwartz and Ben-David [49], and for each data point, an augmentation is sampled from a uniform distribution and then applied.

We define a scoring function to be function $f : A \rightarrow \mathbb{R}$ that quantifies

the difficulty of an augmentation A_i by counting the number of times the augmentation exceeds the minimum cross-entropy loss on the training set, with respect to the set of augmentations and the set of optimal hypotheses considered, as outlined in Algorithm 2. This approach yields an order of difficulty, such that augmentation A_i is more difficult than augmentation A_j if $f(A_i) < f(A_j)$. The main challenge of Curriculum Learning (CL) is choosing f , as it encodes the prior knowledge of the teacher.

We define a pacing function $g_\theta : [I] \rightarrow [M]$, where I is the number of training iterations and M the number of augmentations. The pacing function is used to determine the subsets of augmentations $A'_1, \dots, A'_I \subseteq A$ to sample from, i.e., from what set of augmentations to uniformly sample at each iteration during the training, such that $|A'_i| = g_\theta(i)$. In CL, the i -th subset A'_i includes the first $g_\theta(i)$ elements of the augmentations, sorted by the scoring function f in ascending order of difficulty. Although the choice of the augmentation can be encoded in the distribution each B_i is sampled from, adding a pacing function simplifies the exposition and analysis.

3.0.2 Augmentation Curriculum Learning Method

Together, each scoring function f and pacing function g_θ define a curriculum. Pseudo-code for the Augmentation Curriculum Learning (ACL) algorithm is given in Algorithm 1. In order to narrow down the specific effects of using a scoring function based on ascending difficulty level, we examine two control conditions. Specifically, we define 3 additional scoring functions and corresponding algorithms: (i) The anti-curriculum algorithm uses the scoring function $f' = -f$, where the augmentations are sorted in descending order of difficulty; thus, harder augmentations are

sampled before easier ones. (ii) The random-curriculum algorithm uses a scoring function where the augmentations are randomly scored. (iii) The manually designed curriculum algorithm uses scoring function based solely on augmentation intensity, where lower intensity results in a higher score, without considering augmentation type of teacher's score.

Algorithm 1 Augmentation Curriculum learning method

Input: pacing function g_θ , scoring function f , data X , number of iterations I , augmentation set A .

Output: sequence of augmented mini-batches $\leftarrow \{B_1, \dots, B_I\}$.

sort A according to f , in ascending difficulty order

$result \leftarrow []$

for all $i = 1, \dots, I$ **do**

$B'_i \leftarrow$ sample mini-batch from X

$B_i \leftarrow []$

$size \leftarrow g_\theta(i)$

$A'_i \leftarrow A[1, \dots, size]$

for each (x_j, y_j) in B'_i **do**

 uniformly sample A_i from A'_i

 append $(A_i(x_j), y_j)$ to B_i

end for

 append B_i to $result$

end for

return $result$

3.0.3 Scoring Functions

The primary challenge in curriculum learning is determining the function that accurately represents the difficulty of each augmentation. Previous methods of CL essentially measured the cross entropy loss of example. Our approach to defining the hardness function is by counting how many times each augmentation achieves the lowest cross entropy loss on the train-set among all augmentations when applied to a model trained without augmentations, denoted as $o_\theta : \mathbb{R}^d \rightarrow [K]$, to avoid biasing the model toward any specific augmentation. In mathematical terms, for augmentation A_i , we define the score function $f(A_i)$ as follows:

$$f(A_i) = \mathbb{E}_{x_j, y_j \sim X} \left[\mathbf{1}_{\{\text{CE}(o_\theta(A_i(x_j)), y_j) \leq \min_{k \neq i} \text{CE}(o_\theta(A_k(x_j)), y_j)\}} \right]$$

where CE denotes the cross-entropy loss between the predicted output $o_\theta(A_i(x_j))$ and the true label y_j . The indicator function takes the value of 1 if the cross-entropy loss for augmentation A_i is less than or equal to the minimum cross-entropy loss for all other augmentations A_k (where $k \neq i$), and 0 otherwise.

Using this type of scoring allows us to separate the inherent difficulty of a data point from the augmentation, ensuring that augmentation difficulty remains independent of the varying difficulty of the data.

A higher value of $f(A_i)$ indicates that augmentation A_i is easier, as it leads to lower cross-entropy loss compared to other augmentations.

Later, we developed a more refined method that also tracks the number of

times each augmentation yields the highest loss.

$$f(A_i) = \mathbb{E}_{x_j, y_j \sim X} \left[\mathbf{1}_{\{\text{CE}(o_\theta(A_i(x_j)), y_j) \leq \min_{k \neq i} \text{CE}(o_\theta(A_k(x_j)), y_j)\}} \right] \\ - \mathbb{E}_{x_j, y_j \sim X} \left[\mathbf{1}_{\{\text{CE}(o_\theta(A_i(x_j)), y_j) \geq \max_{k \neq i} \text{CE}(o_\theta(A_k(x_j)), y_j)\}} \right]$$

The score we assign to each augmentation is the difference between the number of times it was the easiest and the number of times it was the most difficult. The advantage of this approach lies not only in leveraging additional information but also in resolving tie cases, where some augmentations did not accumulate enough scores to make a distinct difference between them. The additional scoring helps establish a complete ranking that has no ties in most cases.

Algorithm 2 Smart scoring function

Input: pretrained classifier h_θ , data X set of augmentations $A = [A_1, \dots, A_N]$

Output: Difficulty-ordered augmentations $[A_{i1}, \dots, A_{iN}]$

augments $\leftarrow \{A_1 : 0, \dots, A_N : 0\}$

easy_augments $\leftarrow \{A_1 : 0, \dots, A_N : 0\}$

hard_augments $\leftarrow \{A_1 : 0, \dots, A_N : 0\}$

for all (x_i, y_i) in X **do**

 minimal_loss $\leftarrow \infty$

 maximal_loss $\leftarrow -\infty$

 minimal_augmentation \leftarrow none

 maximal_augmentation \leftarrow none

for each $A_j \in A$ **do**

 cls_loss \leftarrow Cross_Entropy($h_\theta(A_j(x_i)), y_i$)

if cls_loss < minimal_loss **then**

 minimal_loss \leftarrow cls_loss

 minimal_augmentation $\leftarrow A_j$

end if

if cls_loss > maximal_loss **then**

 maximal_loss \leftarrow cls_loss

 maximal_augmentation $\leftarrow A_j$

end if

end for

 easy_augments[minimal_augmentation] += 1

 hard_augments[maximal_augmentation] += 1

end for

for each aug in augmentation_dictionary **do**

 augments[aug] $\leftarrow \frac{1}{\text{dataset.size}} \times (\text{easy_augments[aug]} - \text{hard_augments[aug]})$

end for

$A' \leftarrow$ augmentation array sorted by augments values in descending order

return A'

4 Empirical Results

4.1 Experiment and results

In this section, we empirically evaluate the performance of Augmented Curriculum Learning (ACL) across four use cases: (i) applying ACL with **large set of augmentations** across various dataset-classifier pairs, including CIFAR-10, CIFAR-100, and ImageNet datasets, using ResNet-18 and Wide-ResNet architectures. The large set of augmentations contains 13 different types of augmentations where some more suited for natural image classification tasks, while others may be less applicable in this context.; (ii) applying ACL with **self supervision** i.e using the trained neural net to order the augmentations and then apply ACL; (iii) applying ACL with different sets of augmentations on CIFAR-100 and **measuring accuracy on augmented test set**, such that the same set of augmentations is applied to both the training and test data; (iv) applying ACL with **baseline set of augmentations** across various dataset-classifier pairs, including CIFAR-10, CIFAR-100, and ImageNet datasets, using ResNet-18 and Wide-ResNet architectures. The baseline set of augmentations contains augmentations that are more suited for natural image classification tasks recommended by Cubuk et al. [7].

Our findings show that the direct application of ACL improves neural network performance in both scenarios: when using a specifically chosen augmentation set and when employing a broader group of augmentations. Notably, ACL not only improves performance on a simple test set but also shows improvement on augmented test sets, which present a more challenging problem and shows that the model is more robust in this way. Furthermore, in the self-supervision experiment, we demonstrate that the model can independently calculate its own scoring function while still benefiting from the advantages of ACL. In next sections, we will present significant findings that suggest different teachers (VGG, Resnet, Wide Resnet) yield nearly identical scores, further supporting the efficacy of self-supervision. Later, we empirically show that the arrangement of augmentations aligns with human intuition and is likely invariant to the choice of the teacher network.

4.1.1 Large augmentation set

The first experiment (Figure 4.1) compares various augmentation scheduling techniques using a pretrained Inception model. The augmentation set used for training is derived from Cubuk et al. [7] and includes the following transformations: ShearX/Y, TranslateX/Y, Rotate, Invert, Crop, Equalize, Contrast, Color, Brightness, Sharpness, and Cutout, each applied with different intensity variations. The results demonstrate that ACL provides a clear and significant advantage, with faster initial learning and convergence to a better solution. Additionally, we observe that the performance of ACL with a random scoring function is similar to vanilla approach, indicating that the main reason for the improvement achieved by ACL is due

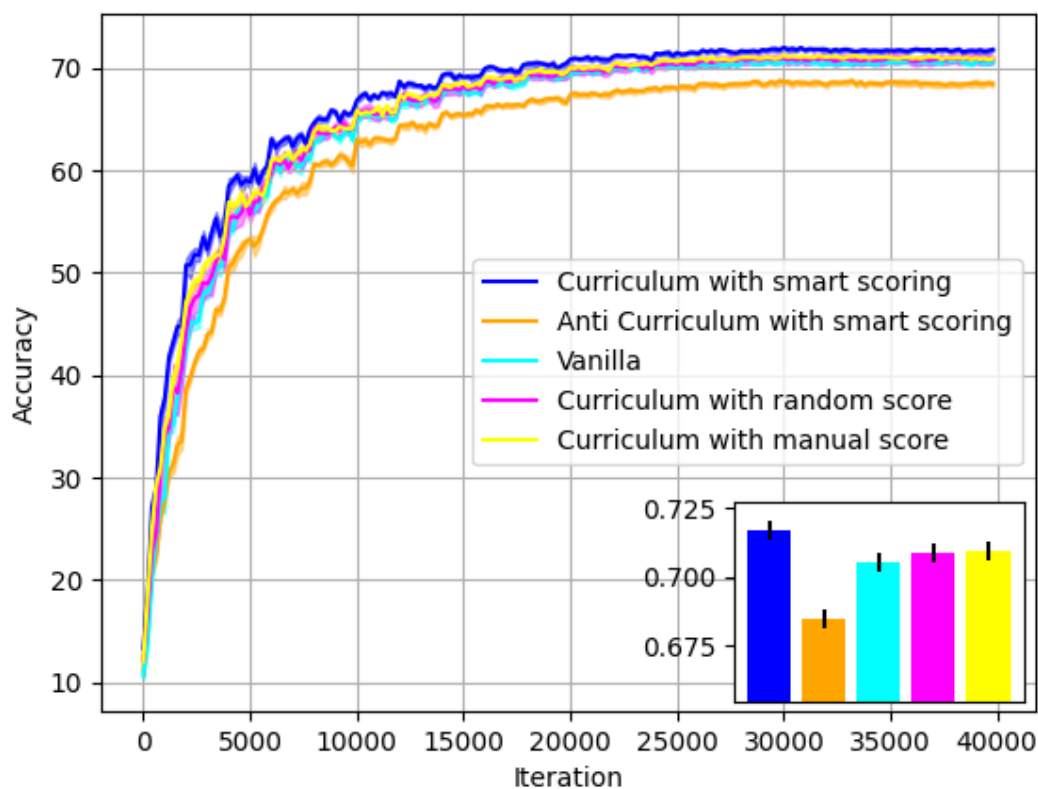


Figure 4.1: CIFAR100 dataset, ResNet18 architecture, trained on large augmentation set, each applied at 3 different intensities, and fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.

to its beneficial augmentation scoring function. It is also evident that gradually increasing the intensity of augmentations alone is insufficient, and that the ACL scoring function goes beyond simply arranging augmentations in increasing order of intensity, instead creating a more sophisticated ordering. Notably, despite being optimized independently, the learning rate hyperparameters for all methods are the same, confirming that the improved performance is due to the use of an effective scoring function.

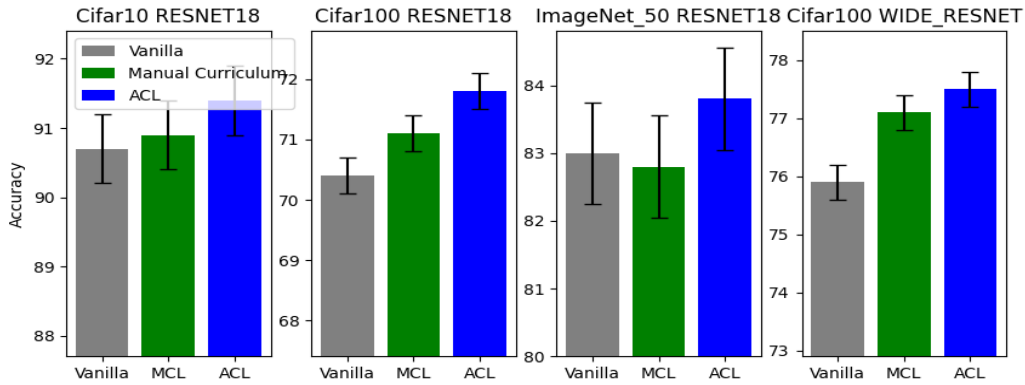


Figure 4.2: Augmentation Curriculum Learning applied to various architectures and datasets with large set of augmentations, each applied at 3 different intensities. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.

4.1.2 Self supervision

In the second experiment (Figure 4.3), we show that the algorithm can be applied in an unsupervised manner. Instead of selecting a teacher (pre-trained network), the neural network is first trained without augmentations for several epochs to estimate the scoring function, after which the model is trained on the augmented data using ACL. The results indicate that the self-supervised version of ACL also offers a distinct advantage, with a better convergence compared to both the vanilla approach and random scoring.

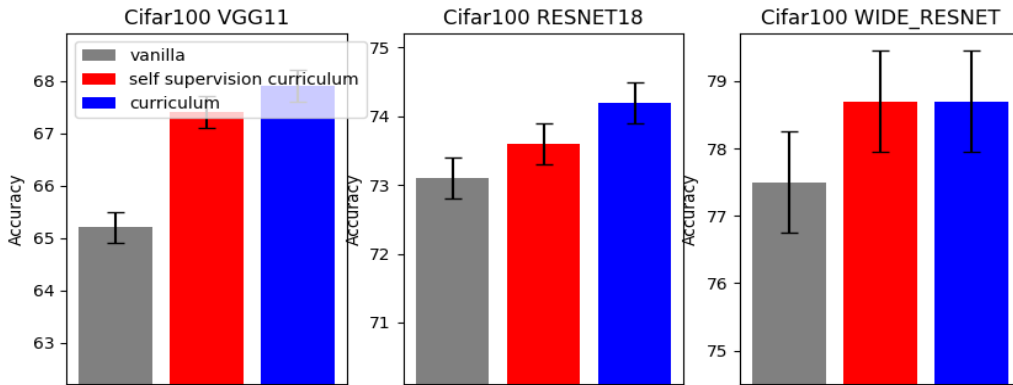


Figure 4.3: Self-supervised Augmentation Curriculum Learning applied to various architectures. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.

4.1.3 Robustness improvement

In the third experiment, we investigated the extent to which the use of Augmentation Curriculum Learning (ACL) enhances the robustness of neural networks. A common approach to improving network robustness for certain augmentation is to apply the augmentation to the training set and train the network on this augmented data. In this experiment, **we applied the same set of augmentations to both the training and test datasets.**

Table 4.1: Comparison of ResNet-18 accuracy on various types of augmented training and testing sets.

AUGMENTATION	CURRICULUM	VANILLA	ANTI CURRICULUM
COLOR DISTORTION	57.13 ± 0.21	56.04 ± 0.19	53.91 ± 0.24
ROTATION	61.31 ± 0.14	59.45 ± 0.11	60.17 ± 0.09
ERASE	43.21 ± 0.11	39.5 ± 0.16	41.22 ± 0.13
COLOR&ERASE &ROTATE	59.2 ± 0.09	58.1 ± 0.11	54.1 ± 0.12

The table summarizes the results of four experiments utilizing different augmentations such that each one compares between ACL, anti ACL and vanilla . As can be seen in Table 4.1 , the use of ACL consistently yields improvements across all cases.

4.1.4 Baseline augmentation set Results

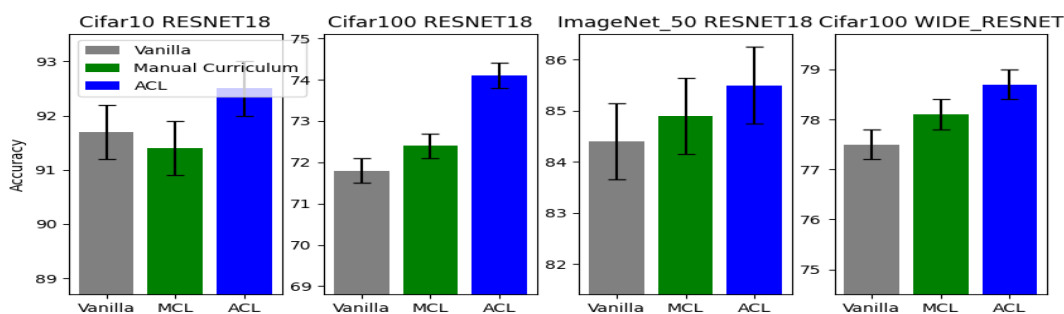


Figure 4.4: Augmentation Curriculum Learning with various architecture and datasets, using a “baseline” augmentation set consisting of augmentations that are better suited to natural images. Bars indicate the average final accuracy, and error bars indicate the STE over 5 iterations.

In the fourth experiment, we leveraged insights from [7] regarding natural images and applied augmentations specifically suited for natural image datasets, including rotation, cropping, translation, and erasure, each with several intensity variations. For instance, the crop augmentation was implemented with 10 different variations, each varying in crop size, similar approach was used for other augmentations. The results of this experiment (Figure 4.4) are consistent with those presented in (Figure 4.1), demonstrating that Augmentation Curriculum Learning (ACL) offers a clear and significant advantage. Furthermore, as previously noted, the goal of ACL is to control the dynamics of the selected set of augmentations, without relying on how well each augmentation type is suited to the dataset. The results are consistent with existing literature on augmentations, demonstrating that the method achieves higher accuracy when applied to augmentation sets that are more suited for natural images, as op-

posed to a general augmentation set (Figure 4.1).

4.2 Neural net properties

4.2.1 Single augmentation arrangement

In the previous section, We defined the score for augmentation i as the expected value of the indicator function for the event that the i -th augmentation achieved the smallest loss, compared to all other augmentations, on a given image.

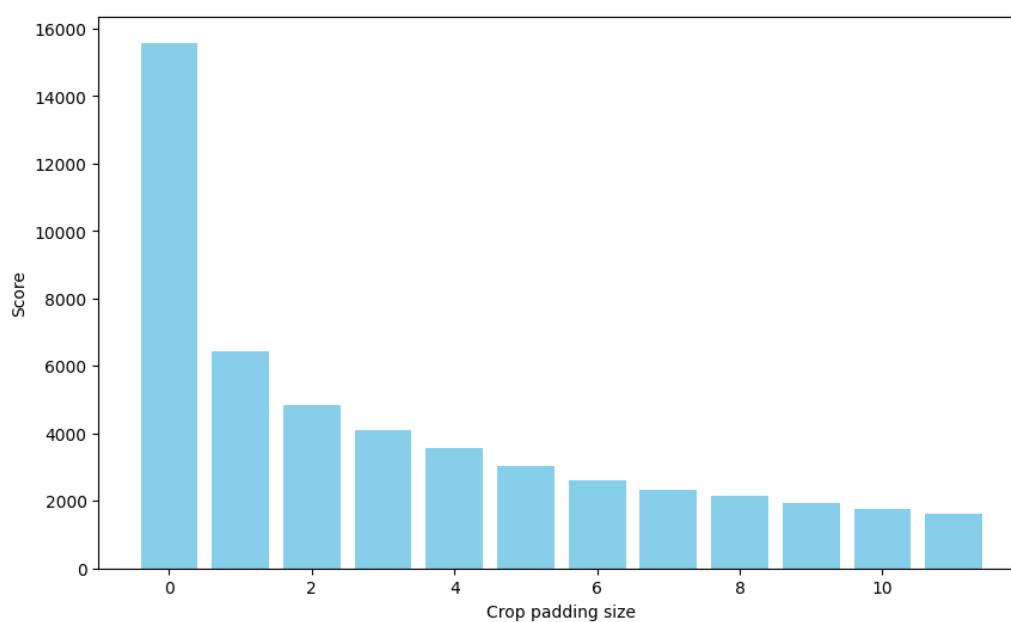


Figure 4.5: CIFAR-100 dataset, Inception-based non-normalized simple score for crop augmentations at different intensity levels. The X-axis indicates crop size, and the Y-axis represents the augmentation score.

The experiment shows that the arrangement of augmentations aligns with

human intuition which is: the higher the crop size the harder the augmentation is. The results indicate that for a single augmentation type, such as crop, the augmentation difficulty increases as the crop size grows.

4.2.2 Baseline augmentation sort

In this part, We present the scoring of the baseline augmentation set, which consists of four types of augmentations, which are more suited to natural images Krizhevsky et al. [35], each applied at ten different intensity levels, except for the flip augmentation, which can only be performed in one way.

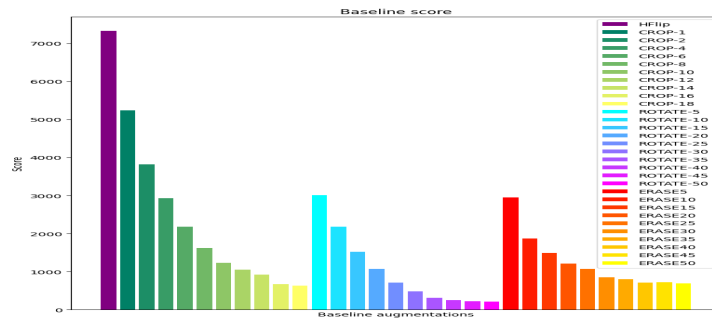


Figure 4.6: CIFAR-100 dataset, Inception-based, non-normalized simple score for: crop (green), rotation (blue), erase (red) and flip (purple) augmentations at different intensity levels. The X-axis indicates crop size, and the Y-axis represents the augmentation score.

The results indicate that for each augmentation the order aligns with their intensity, reflecting human intuition. Additionally, the arrangement of augmentations from easy to hard is not continuous but exhibits an internal complex structure. For instance, a 5-degree rotation is found to be easier than a crop with a padding size of 4. The non-trivial order of augmentations suggests that the difficulty of each augmentation is shaped by factors

beyond intensity and rely on the inherent nature of each augmentation.

4.2.3 Neural net agreement

One of the key challenges in methods where a teacher guides a student (in our case, a trained network assisting in the training of another network) is the selection of the teacher, as different teachers may yield varying results when instructing the student. To demonstrate that our method is being invariant to changes in the teacher, we calculated the similarity of scores across different teachers by calculating the cosine distance between the scores for each pair of teachers .

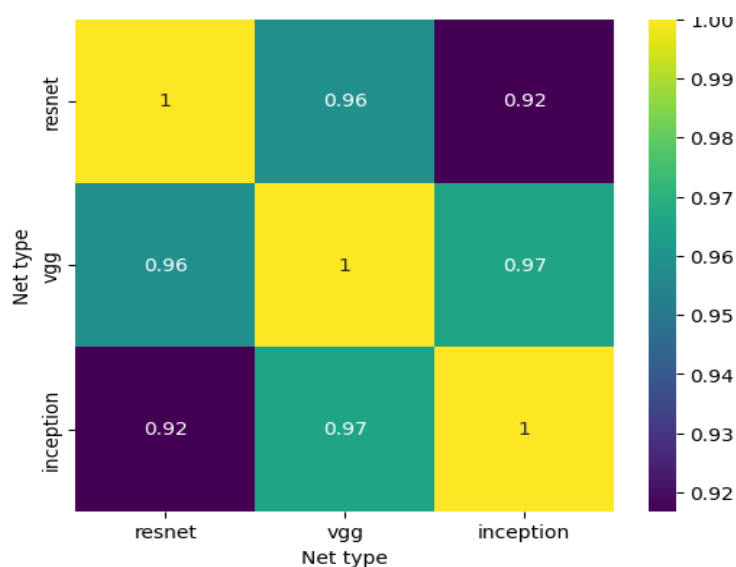


Figure 4.7: This matrix shows the cosine similarity between score vectors calculated by different models on the CIFAR-100 dataset with the baseline augmentation set. The value at position (i, j) is the cosine similarity between the scores computed by model i and model j .

As can be observed in (Figure 4.7), the different teachers produce nearly

identical scores, suggesting that the proposed method is invariant to the choice of convolution neural network teacher.

4.2.4 Smart scoring vs Simple scoring

As noted in an earlier section, there are two methods to calculate the score of each augmentation: a "simple" method and "smart" method. The distinction between these methods lies in how different groups of augmentations influence the model's training. More specifically, for each dataset, augmentations can generally be divided into two groups: one that is well-suited to the dataset and another that is less suitable [35].

When the augmentation set includes both suitable and less suitable augmentations, the simple scoring approach tends to bias the scores significantly toward the more compatible (easier) augmentations.

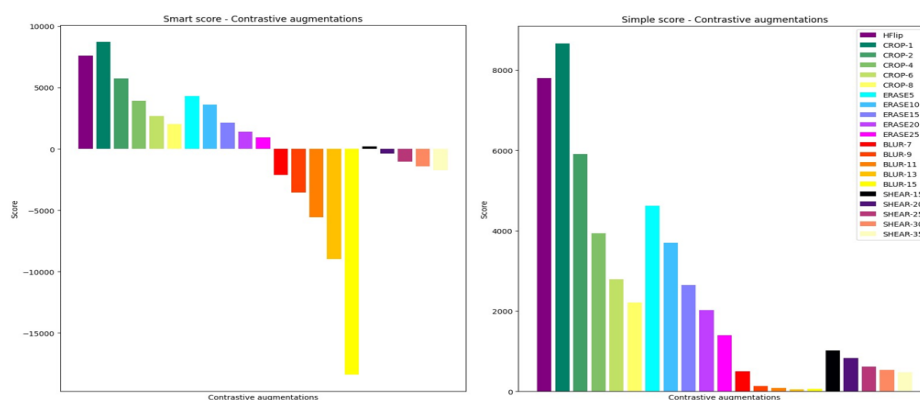


Figure 4.8: Comparison of Non-Normalized Scores on CIFAR-100 Using the Inception-Based Model: Left – Smart Scoring Function, Right – Simple Scoring Function.

The "contrastive" augmentation set comprises horizontal flip and four

types of augmentations: two that are well-suited to the dataset (easy augmentations: erase (blue) and crop (green) and two that are less suited (hard augmentations: blur (red) and shear&translate (black)). Each augmentation is applied across five different intensity levels.

As can be seen in (Figure 4.8) when using "simple" scoring method over 85% of the score concentrated in horizontal flip and the two easier augmentations (first 11 bars), with the remaining augmentations receiving less attention, which results in a less distinct ranking for the more challenging augmentations that leading to less effectively learning in the part where the harder augmentations start to appear (iteration 2000) which cause lower final accuracy compared to smart loss (Figure 4.9). In contrast, when using the "smart" scoring method, the easier augmentation score is around 50% of the total score , and thus the more difficult augmentations receive higher "recognition" in the scoring process, producing a clearer ranking among them.

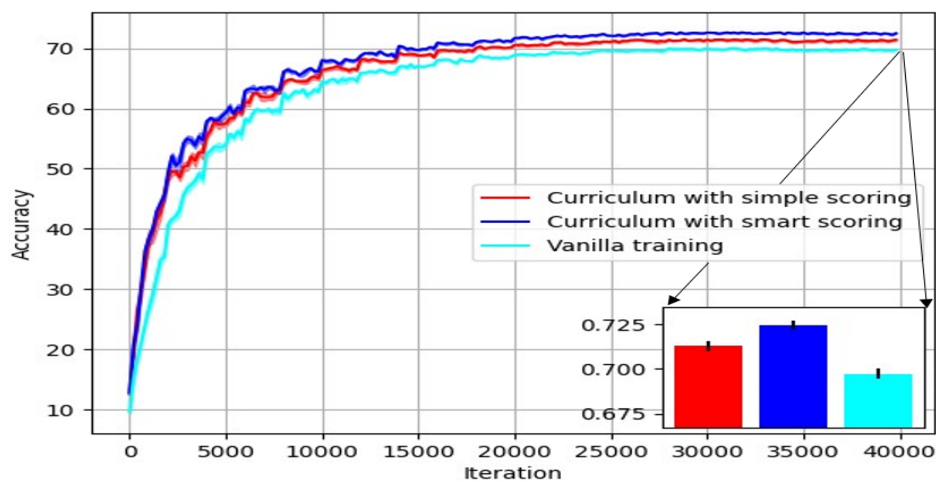


Figure 4.9: CIFAR100 dataset, ResNet18 architecture, “contrastive” augmentation set with fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.

A similar phenomenon, where hard augmentations do not receive sufficient scores, can be observed when training ResNet-18 using a large augmentation set comprising 13 different augmentations (Figure 4.10).

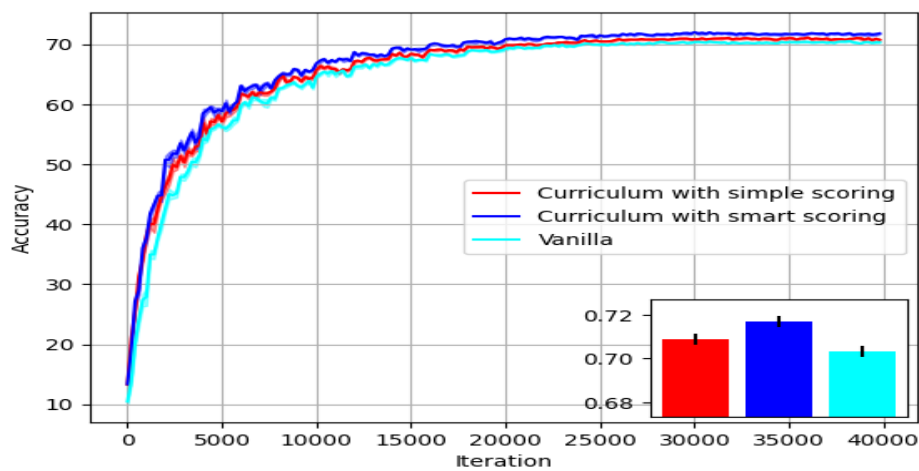


Figure 4.10: CIFAR100 dataset, ResNet18 architecture, large augmentation set with fixed pacing function. Bars represent the average final accuracy, error bars indicate the STE over 5 iterations.

Based on these findings, in the general case it is recommend to use the smart scoring method. However, in cases where it is known that the augmentations are highly compatible with the dataset, the simple scoring method may also be appropriate.

4.3 Scoring function optimization

Calculating the score can become computationally expensive as the size of the dataset (m) and the number of augmentations (n) increase. Using algorithms 2 and 3, the scoring computational complexity is $O(mn)$.

Let $X_{i,j}$ be an indicator variable that denotes which augmentation achieved the lowest loss on the image. Specifically, $X_{i,j} = 1$ if for the i -th image, the j -th augmentation results in the lowest loss, and $X_{i,j} = 0$ otherwise. We denote $\hat{\mu}_m = \frac{1}{m} \sum_{i=1}^m X_i$ and $\mu = \mathbb{E}[\mu_m]$.

Proposition Let $\hat{\mu}_m$ be the estimated parameter vector and μ be its expectation. To reduce the number of points required to calculate the scoring function, such that the difference between $\hat{\mu}_m$ and μ is within ϵ with probability at least $1 - \delta$, we choose $m = \frac{\ln(\frac{2n}{\delta})}{2\epsilon^2}$.

For the i 'th augmentation $\hat{\mu}_{m,i}$, Hoeffding's inequality provides:

$$\mathbb{P}(|\hat{\mu}_{m,i} - \mu_i| \geq \epsilon) \leq 2 \exp(-2m\epsilon^2)$$

Thus, for the general case we get that:

$$\begin{aligned} \mathbb{P}\left(\max_{1 \leq i \leq n} |\hat{\mu}_i - \mu_i| \geq \epsilon\right) &\leq \mathbb{P}\left(\bigcup_{i=1}^n |\hat{\mu}_i - \mu_i| \geq \epsilon\right) \\ &\leq \sum_{i=1}^n \mathbb{P}(|\hat{\mu}_i - \mu_i| \geq \epsilon) \leq 2n \exp(-2m\epsilon^2) \\ &= 2n \exp\left(-2 \left(\frac{\ln(\frac{2n}{\delta})}{2\epsilon^2}\right) \epsilon^2\right) = \delta \end{aligned}$$

NEURAL NET TYPE	REGULAR SCORING	OPTIMIZED SCORING	VANILLA
RESNET18	74.13 ± 0.07	73.62 ± 0.13	71.81 ± 0.06
WIDERESNET	85.51 ± 0.076	85.12 ± 0.093	84.46 ± 0.071
VGG11	67.92 ± 0.041	67.35 ± 0.082	65.62 ± 0.048

Table 4.2: Final accuracy comparison on the CIFAR-100 dataset using the ResNet18 architecture. smart scoring function vs. optimized smart scoring function.

As shown in Table 4.2, by optimizing the scoring function calculation, we achieve results nearly identical to the standard method, while still preserving the advantage of ACL over vanilla training. This optimization allows for a 60% reduction in scoring computation time for CIFAR-100 (from 537 seconds to 214 seconds) and an 85% reduction for ImageNet-50 (from 1,351 seconds to 217 seconds).

5 Summary and Conclusion

We investigate the challenge of Curriculum Learning (CL) in environments with diverse data augmentations. Our approach is motivated by the principle that the learning process should progressively expose the neural network to training examples, ordered by increasing difficulty. By leveraging a teacher model to quantify the complexity of each augmentation, we design a training process that gradually introduces more challenging samples over time. Through extensive empirical evaluation across multiple datasets, we demonstrate that different teacher models consistently rank the augmentation difficulty similarly, minimizing dependency on specific teacher choices. Our findings show that Augmentation Curriculum Learning not only outperforms alternative methods on both standard and augmented test sets, but also introduces a novel and effective approach to quantify the complexity of augmentations.

Bibliography

- [1] Almeida, J., Saltori, C., Rota, P., and Sebe, N. (2020). Low-budget unsupervised label query through domain alignment enforcement. *arXiv preprint arXiv:2001.00238*.
- [2] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., and Chen, G. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182.
- [3] Antoniou, A., Storkey, A., and Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint*, arXiv:1711.04340.
- [4] Antoniou, A., Storkey, A., and Edwards, H. (2018). Augmenting image classifiers using data augmentation generative adversarial networks. *University of Edinburgh and Open AI*.
- [5] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.
- [6] Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649. IEEE.

- [7] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019a). Autoaugment: Learning augmentation strategies from data. *Google Brain*.
- [8] Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2019b). Randaugment: Practical automated data augmentation with a reduced search space. *Google Research, Brain Team*.
- [9] DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.
- [10] Douzas, G. and Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. *NOVA Information Management School, Universidade Nova de Lisboa*.
- [11] Dvornik, N., Mairal, J., and Schmid, C. (2018). Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 760–776. Springer.
- [12] Fan, Y., Tian, F., Qin, T., Li, X.-Y., and Liu, T.-Y. (2018). Learning to teach. In *International Conference on Learning Representations*.
- [13] Fawzi, A., Samulowitz, H., Turaga, D., and Frossard, P. (2016). Adaptive data augmentation for image classification. *EPFL, Switzerland & IBM Watson Research Center, USA*.
- [14] Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. (2017). Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pages 482–495.

- [15] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018a). Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331.
- [16] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018b). Synthetic data augmentation using gan for improved liver lesion classification. *Neurocomputing*, 321:321–331.
- [17] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *University of Tübingen & IMPRS-IS*.
- [18] Georgakis, G., Mousavian, A., Berg, A. C., and Kosecka, J. (2017). Synthesizing training data for object detection in indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1460–1468. IEEE.
- [19] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.-Y., Cubuk, E. D., Le, Q. V., and Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928. IEEE.
- [20] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwinska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., and Agapiou, J. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471.

- [21] Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning (ICML)*.
- [22] Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. (2020a). Faster autoaugment: Learning augmentation strategies using back-propagation. *arXiv preprint arXiv:2001.03971*.
- [23] Hataya, R., Zdenek, J., Yoshizoe, K., and Nakayama, H. (2020b). Meta approach to data augmentation optimization. *The University of Tokyo and RIKEN AIP*.
- [24] Hendrycks, D., Mu, N., Cubuk, E., Zoph, B., Gilmer, J., and Lakshminarayanan, B. (2020). Augmix: A simple method to improve robustness and uncertainty under data shift. In *International Conference on Learning Representations*, volume 1, page 6.
- [25] Hong, M., Choi, J., and Kim, G. (2021). Stylemix: Separating content and style for enhanced data augmentation. *Seoul National University*.
- [26] Huang, S.-W., Lin, C.-T., Chen, S.-P., Wu, Y.-Y., Hsu, P.-H., and Lai, S.-H. (2018). Auggan: Cross domain adaptation with gan-based data augmentation. *Department of Computer Science, National Tsing Hua University, Taiwan*.
- [27] Inoue, H. (2018). Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*.
- [28] Jackson, P. T., Atapour-Abarghouei, A., Bonner, S., Breckon, T., and Obara, B. (2021). Style augmentation: Data augmentation via style randomization. *Department of Computer Science, Durham University*.

- [29] Jiang, L., Meng, D., Zhao, Q., Shan, S., and Hauptmann, A. G. (2015). Self-paced curriculum learning. In *Proceedings of AAAI*, pages 2694–2700.
- [30] Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018a). Mentor-net: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of ICML*, pages 2304–2313. International Conference on Machine Learning (ICML).
- [31] Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018b). Mentor-net: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2309–2318.
- [32] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of ICLR*.
- [33] Kim, J., Choo, W., and Song, H. (2021). Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*.
- [34] Kim, J.-H., Choo, W., and Song, H. (2020). Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning, PMLR*, pages 5275–5285.
- [35] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.
- [36] Krueger, K. A. and Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

- [37] Lee, D., Park, H., Pham, T., and Yoo, C. D. (2020). Learning augmentation network via influence functions. *Korea Advanced Institute of Science and Technology (KAIST)*.
- [38] Ma, F., Meng, D., Xie, Q., Li, Z., and Dong, X. (2017). Self-paced co-training. In *Proceedings of ICML*, pages 2275–2284.
- [39] Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., and Malossi, C. (2018). Bagan: Data augmentation with balancing gan. *IBM Research - Zurich*.
- [40] Mun, S., Park, S., Han, D. K., and Ko, H. (2017). Generative adversarial network based acoustic scene training set augmentation and selection using svm hyperplane. In *Detection and Classification of Acoustic Scenes and Events Workshop*.
- [41] Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). *Search and rescue robotics*, pages 1151–1173. Springer.
- [42] Pavlov, P. I. (2010). Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex. *Annals of Neurosciences*, 17(3):136.
- [43] Perez, L. and Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint*, arXiv:1712.04621.
- [44] Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunmon, J., and Re, C. (2017a). Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, pages 3239–3249.

- [45] Ratner, A. J., Ehrenberg, H. R., Hussain, Z., Dunnmon, J., and Ré, C. (2017b). Learning to compose domain-specific transformations for data augmentation. *Stanford University*.
- [46] Ravuri, S. and Vinyals, O. (2019). Classification accuracy score for conditional generative models. *arXiv preprint*, arXiv:1905.10887.
- [47] Sato, I., Nishimura, H., and Yokoi, K. (2015). Apac: Augmented pattern classification with neural networks. *arXiv preprint*, arXiv:1505.03229.
- [48] Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., and Bronstein, A. M. (2021). Δ -encoder: An effective sample synthesis method for few-shot object recognition. *IBM Research AI*.
- [49] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- [50] Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of International Conference on Document Analysis and Recognition*.
- [51] Sinha, S., Garg, A., and Larochelle, H. (2020). Curriculum by smoothing. In *Proceedings of NeurIPS*, volume 33, pages 21653–21664. NeurIPS.
- [52] Sixt, L., Wild, B., and Landgraf, T. (2016). Rendergan: Generating realistic labeled data. *arXiv preprint*, arXiv:1611.01331.
- [53] Skinner, B. F. (1958). Reinforcement today. *American Psychologist*, 13(3):94.

- [54] Soviany, P. (2020). Curriculum learning with diversity for supervised computer vision tasks. *Unspecified Publisher*.
- [55] Tang, Z., Peng, X., Li, T., Zhu, Y., and Metaxas, D. (2020). Adatransform: Adaptive data transformation. *Rutgers University & University of Delaware*.
- [56] Tokozume, Y., Ushiku, Y., and Harada, T. (2018). Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494.
- [57] Tran, T., Pham, T., Carneiro, G., Palmer, L., and Reid, I. (2025). A bayesian data augmentation approach for learning deep models. *The University of Adelaide*.
- [58] Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066.
- [59] Wang, H., Xiao, C., Kossaiji, J., Yu, Z., Anandkumar, A., and Wang, Z. (2022). Augmax: Adversarial composition of random augmentations for robust training. In *Department of Electrical and Computer Engineering, University of Texas at Austin; NVIDIA; Arizona State University; California Institute of Technology*.
- [60] Xu, M., Yoon, S., Fuentes, A., and Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347.
- [61] Xu, Y., Wang, X., Song, K., Du, J., Liu, J., Miao, Y., and Li, Y. (2021). Bsa-encapsulated cyclometalated iridium complexes as nano-

- photosensitizers for photodynamic therapy of tumor cells. *RSC Advances*, 11:14622–14630.
- [62] Yun, S., Han, D., Oh, S., Chun, S., Choe, J., and Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032.
- [63] Zaremba, W. and Sutskever, I. (2014). Learning to execute. *arXiv preprint*, arXiv:1410.4615.
- [64] Zhang, D., Yang, L., Meng, D., Xu, D., and Han, J. (2017a). Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *Proceedings of CVPR*, pages 4429–4437.
- [65] Zhang, D., Yang, L., Meng, D., Xu, D., and Han, J. (2020). Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. *Northwestern Polytechnical University, Xi'an Jiaotong University, University of Sydney*.
- [66] Zhang, H., Cisse, M., Dauphin, Y., and Lopez-Paz, D. (2017b). Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.
- [67] Zhang, X., Wang, Q., Zhang, J., and Zhong, Z. (2019). Adversarial autoaugment. *Huawei*.
- [68] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008.
- [69] Zhu, X., Liu, Y., Li, J., Wan, T., and Qin, Z. (2018). Emotion classification with data augmentation using generative adversarial networks. In

- Advances in Knowledge Discovery and Data Mining (PAKDD 2018)*, pages 349–360. Springer.
- [70] Zhu, X., Liu, Y., Qin, Z., and Li, J. (2017). Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint*, arXiv:1711.00648.
- [71] Zhu, Y., Aoun, M., Krijn, M., and Vanschoren, J. (2021). Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. *Department of Mathematics and Computer Science, Eindhoven University of Technology*.