ICARC

International Conference on Advanced Research in Computing

Curriculum Learning
An Efficient Learning Paradigm

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

Utrecht University

SLIIT Research and International

BRAIN LABS
Brain-Inspired AI & Neuroinformatics Research Group

Organizing Committee

Dr. Dharshana Kasthurirathna
Sri Lanka Institute of Information Technology (SLIIT)

Dr. Mahima Weerasinghe
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Menan Velayuthan
Utrecht University

Mr. Asiri Gawesha
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Sanka Mohottala
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Dulara Madhusanka
Sri Lanka Institute of Information Technology (SLIIT)

Ms. Savini Kommalage
Sri Lanka Institute of Information Technology (SLIIT)

Organized by BrAINLabs Research Group, SLIIT
Funded by a SLIIT Research & International (Grant No. PVC(R&I)/RG/2025/12)

Resource Personal

Ms. Savini Kommalage
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Dulara Madhusanka
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Menan Velayuthan
Utrecht University

Mr. Asiri Gawesha
Sri Lanka Institute of Information Technology (SLIIT)

Mr. Sanka Mohottala
Sri Lanka Institute of Information Technology (SLIIT)

International Conference on Advanced Research in Computing

Curriculum Learning
An Efficient Learning Paradigm

Session 1 : Introduction to Curriculum Learning

Savini Kommalage
Sri Lanka Institute of Information Technology (SLIIT)

Learning in Deep Neural Networks

- Learning hierarchical representations from data
- Using multi-layer parameterized models
- Trained via gradient-based optimization

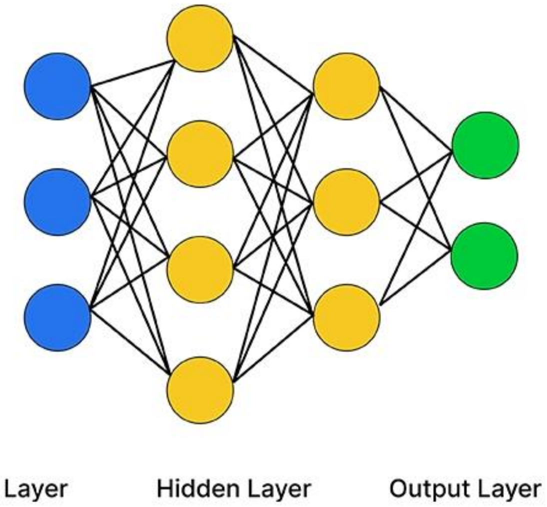


Figure : neural network

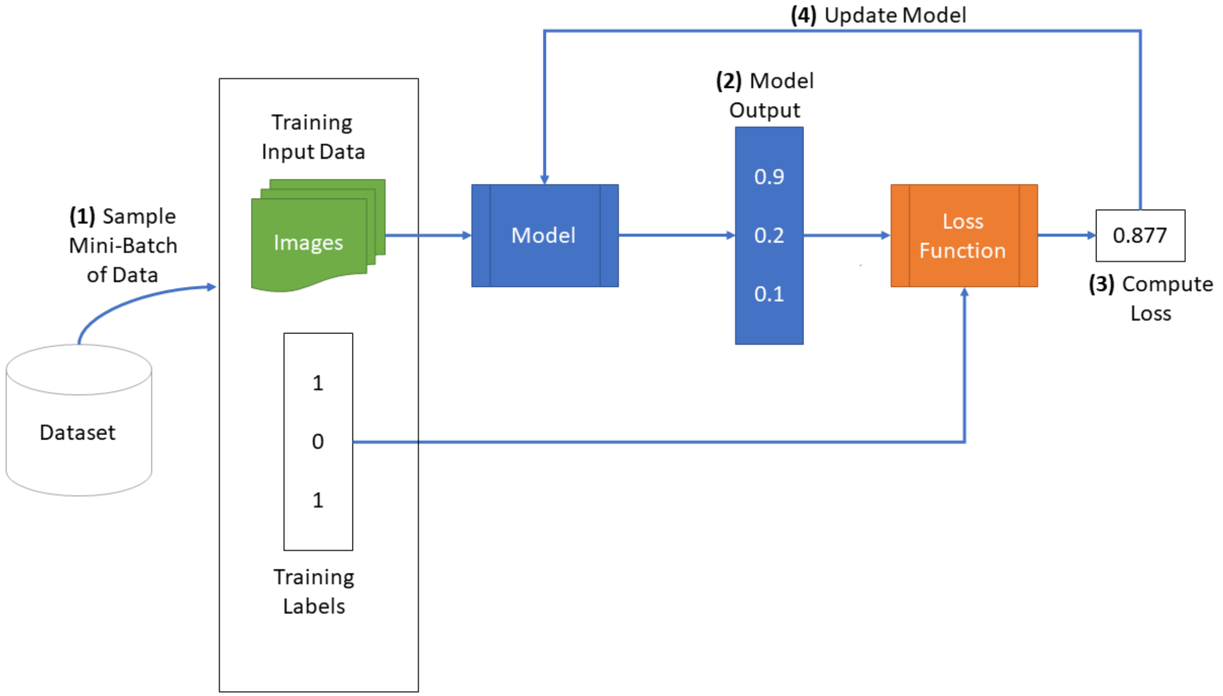


Figure : General supervised deep learning pipeline

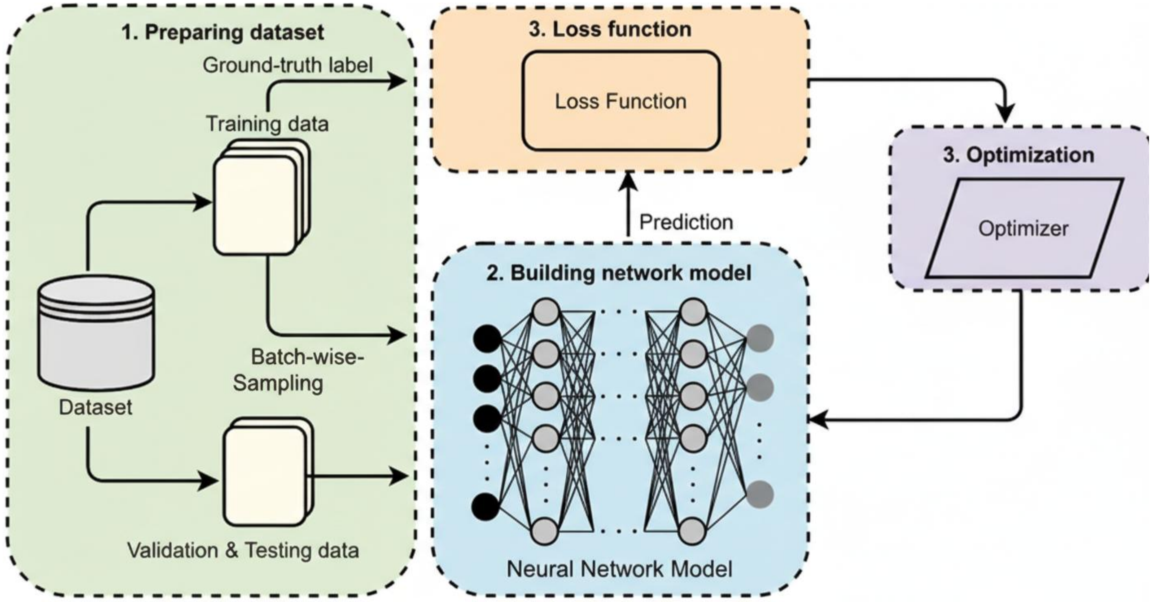


Figure : general deep learning pipeline

Standard Deep Learning

- Most deep models see **all data as equal**
- Start with the full dataset
- Shuffle the dataset
- Mini-batches are sampled uniformly at random
- Iterative optimization

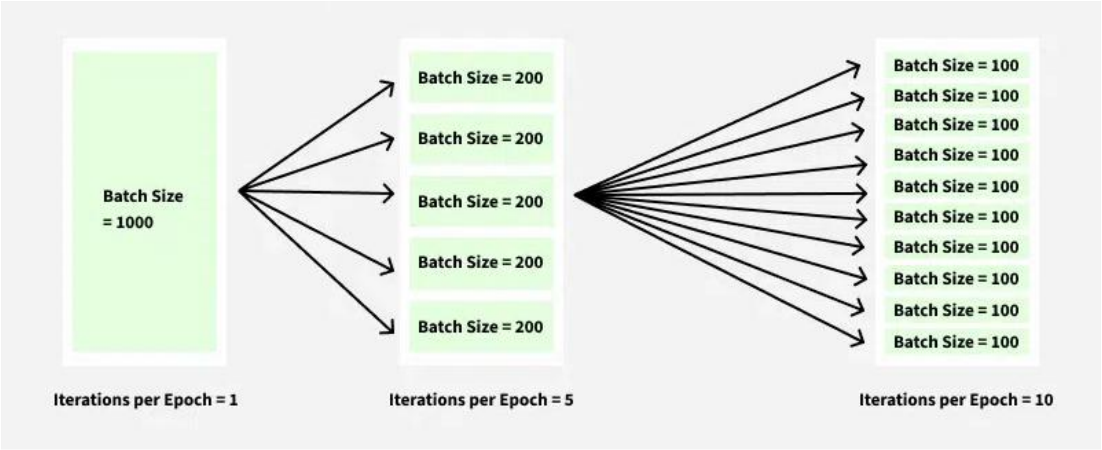


Figure : mini batches/ how different batch sizes affect the number of iterations per epoch.

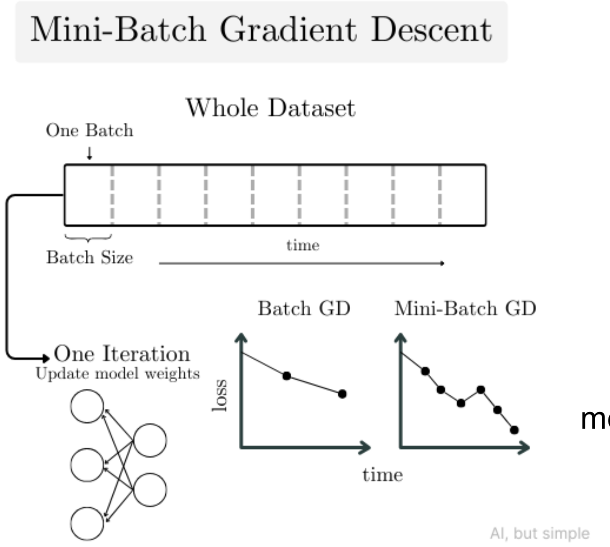
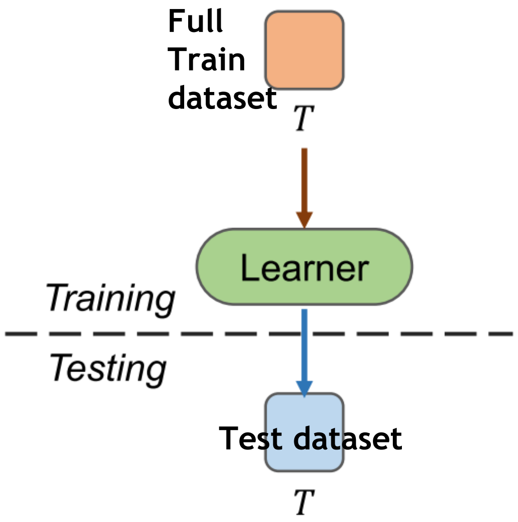


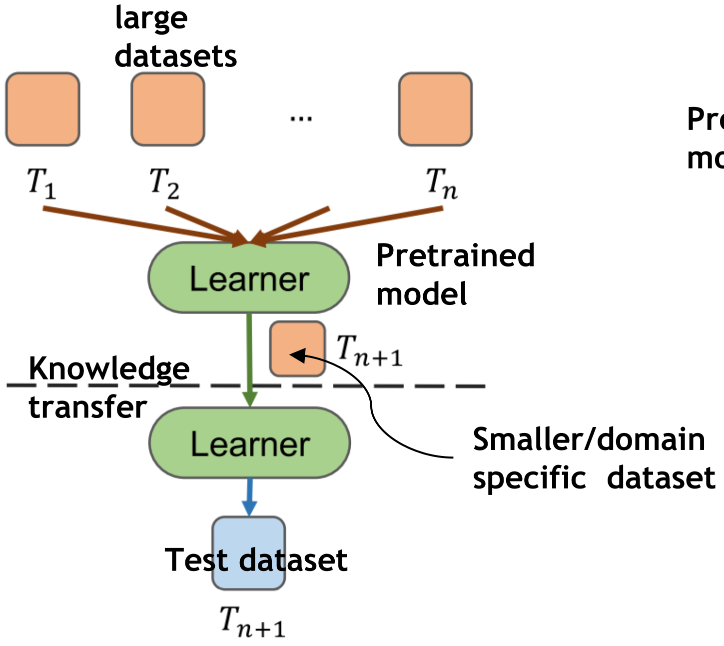
Figure : Mini-batch gradient descent mechanism for iterative model optimization

Learning paradigms

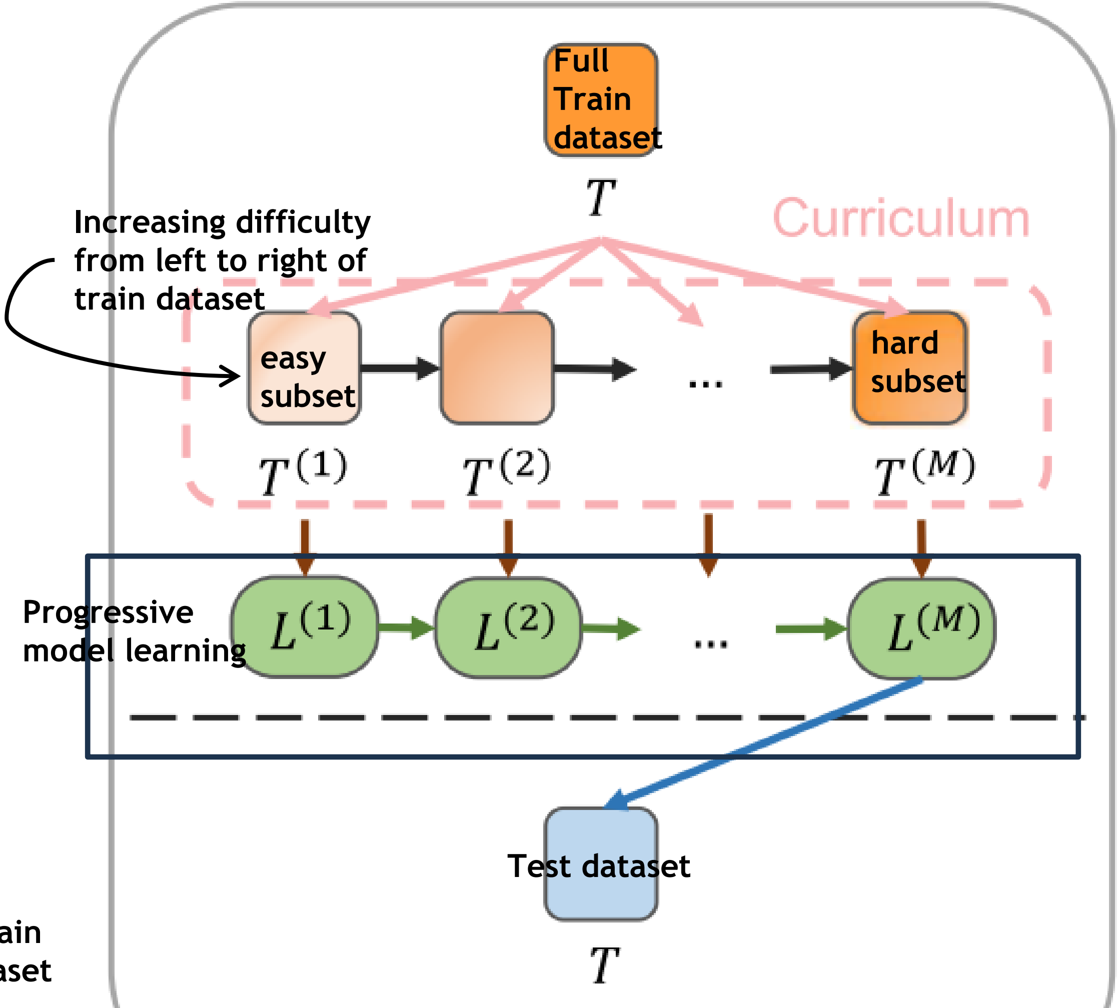
- ❑ Transfer Learning
- ❑ **Curriculum Learning**
- ❑ Self-paced Learning
- ❑ Meta-Learning
- ❑ Continual Learning
- ❑ Active Learning



Traditional Machine Learning



Transfer Learning



Curriculum Learning

Learning Is Not Random : Human Curriculum

- Start with simple concepts
- Gradually build complexity

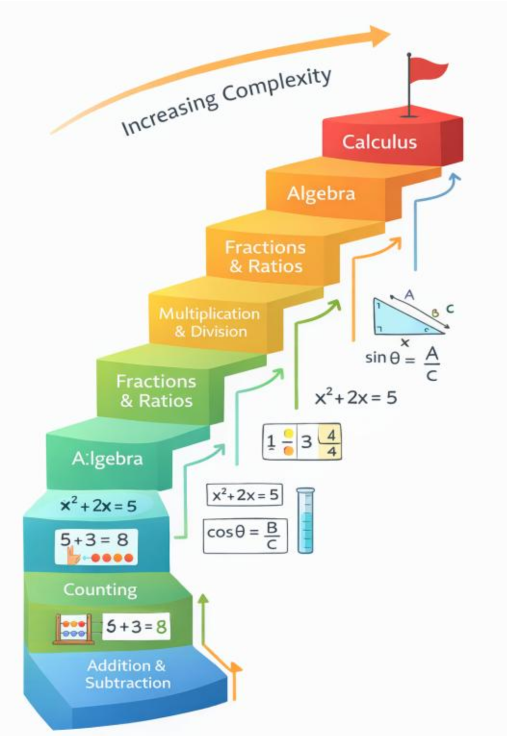


Figure : Example of a human-designed curriculum: mathematics learning progression

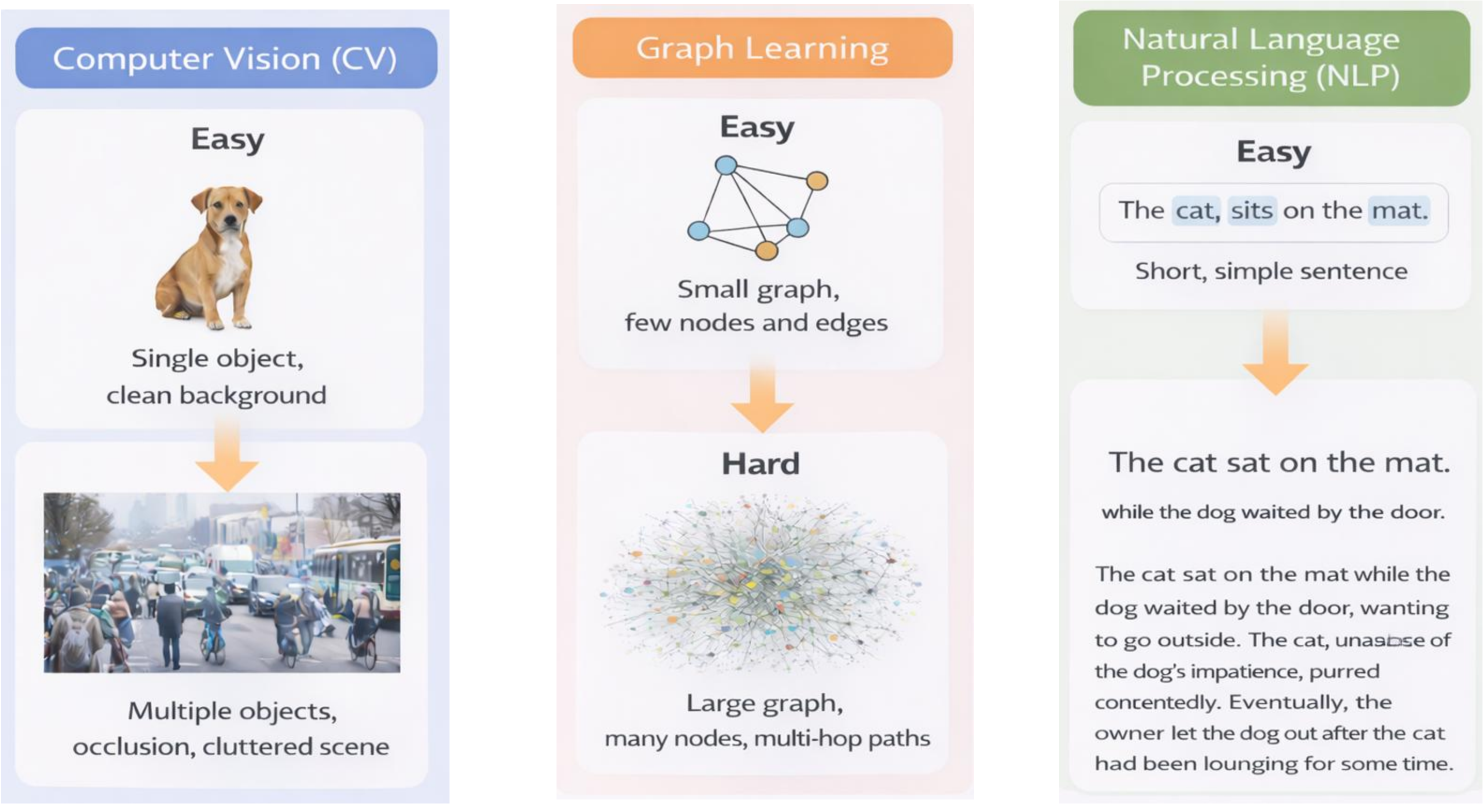


Figure : Conceptual illustration of of progression of sample difficulty in CV, Graphs, NLP

Curriculum learning

- Curriculum Learning (CL) is a training strategy where a model learns from simple examples first and gradually progresses to more complex ones.

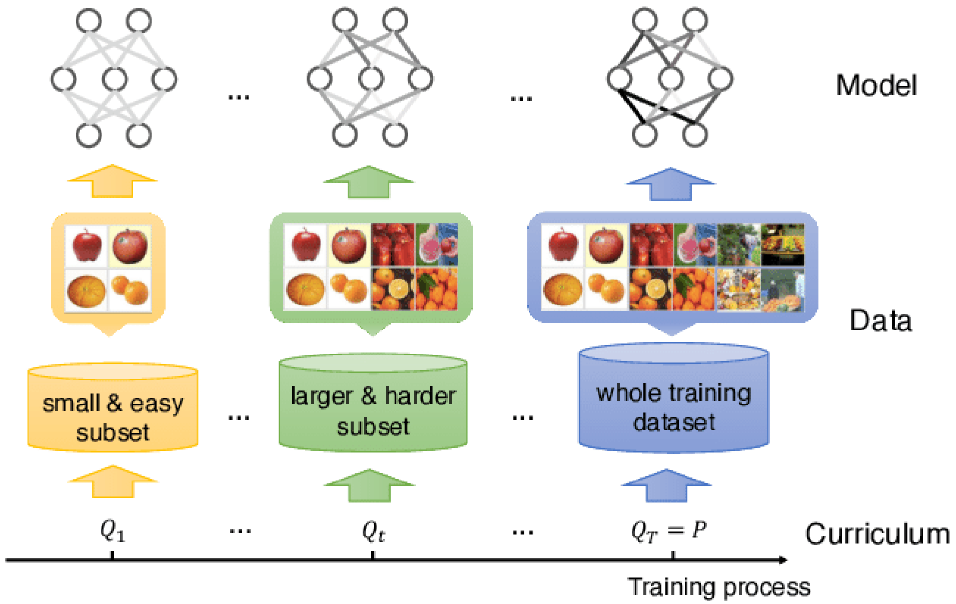


Figure : illustration of pre- defined data level curriculum training strategy [2]

Origin: [Bengio et al. (2009), *Curriculum Learning*,](#) ICML

Formalized curriculum learning in machine learning.

Shape Classification Using Curriculum Learning

- Task : 3-class image classification (rectangle, ellipse, triangle)

- Input : 32 × 32 grayscale images
Basic Shapes – low variability (easy)
Geometric Shapes – high variability (hard)

- Model : Neural network architecture, Stochastic Gradient Descent (SGD)

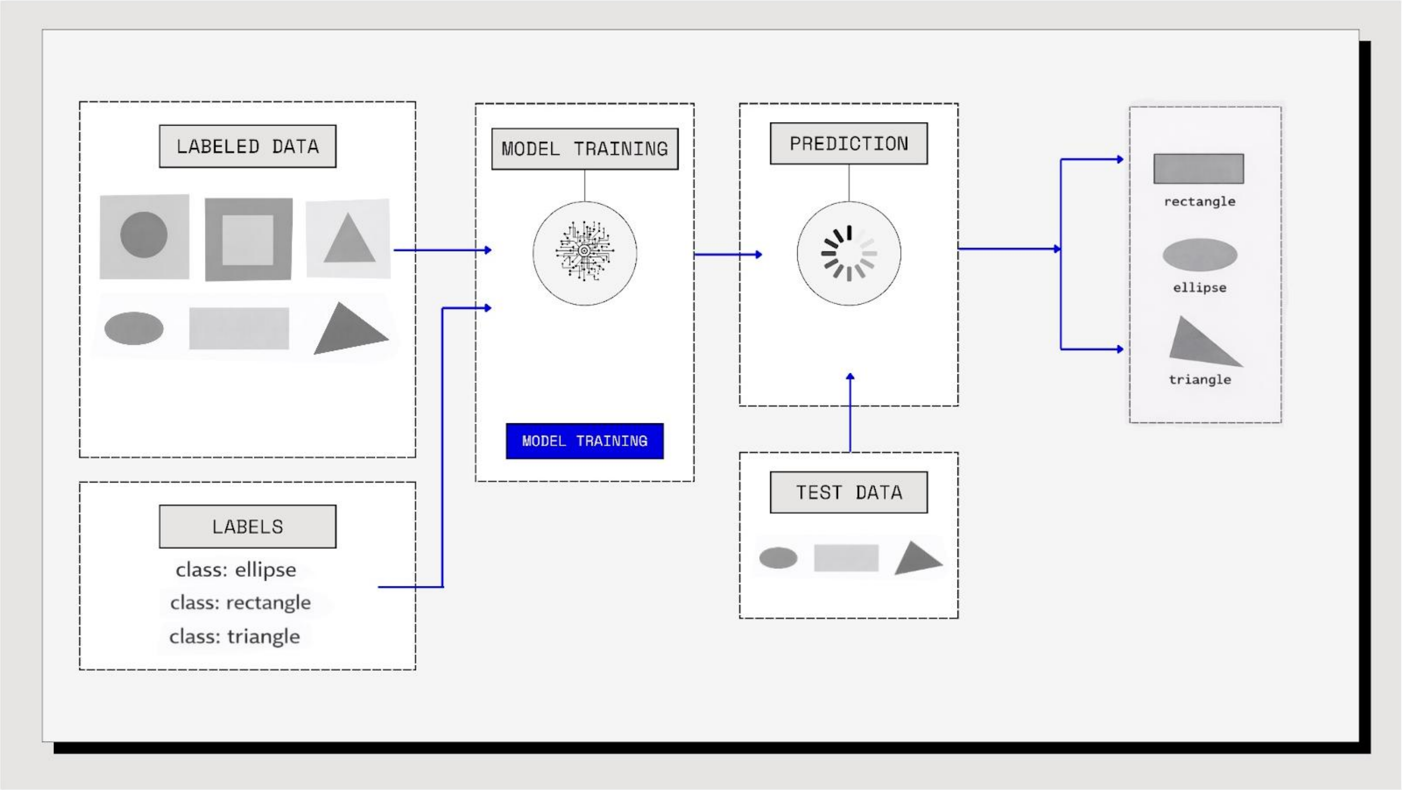
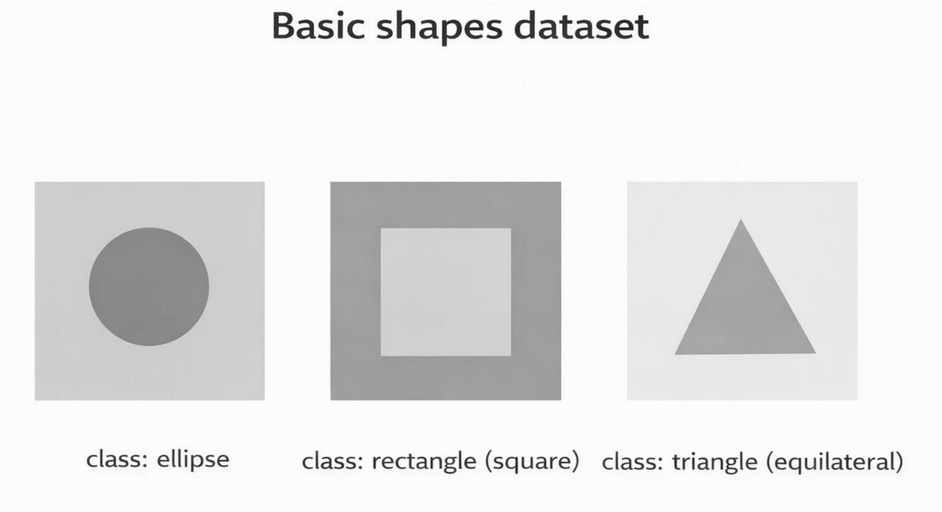
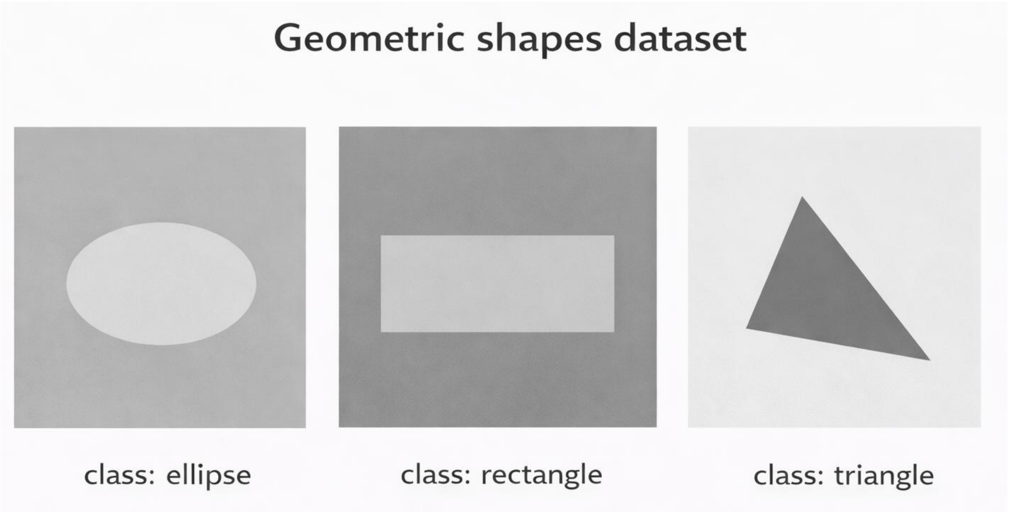


Figure : Experimental set up :Two-stage curriculum design for synthetic shape classification





- Ellipse class : (uniform radius)
- Rectangle class : (width = length)
- triangle (equilateral triangle)

- Ellipse : any ellipse (varying major/minor axes)
- Rectangle : any rectangle (arbitrary width, height)
- Triangle → any triangle (scalene, isosceles, etc.)

"Less variability in shape" acts as a **heuristic measure of sample difficulty.**
In curriculum learning, this notion of *how easy or hard a sample* is what we call a **scoring function.**

- Stage 1 (Easy distribution)
Train only on BasicShapes

- Stage 2 (Difficult/Target distribution)
Switch to GeomShapes

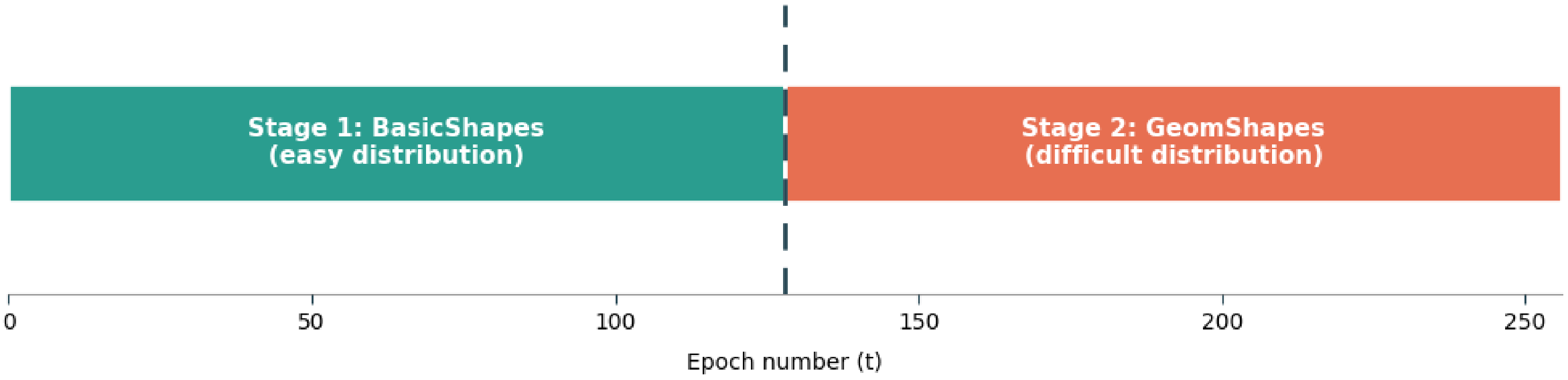
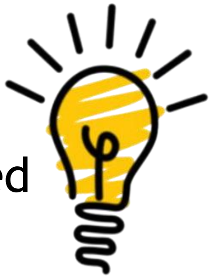


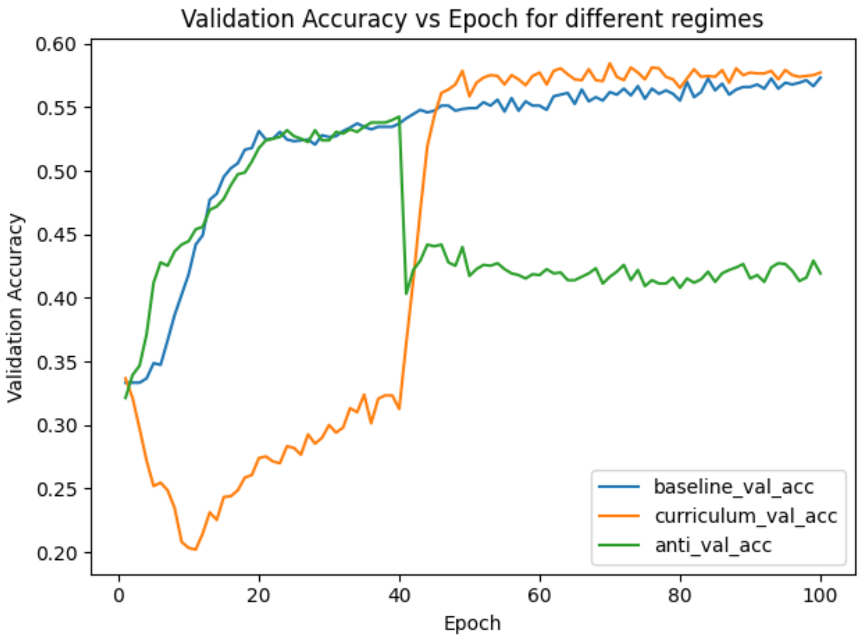
Figure : switch epoch pacing function

This two-step curriculum defines how training samples are paced over time.
In curriculum learning, this notion of *when* easy and hard samples are presented to the model is called a **pacing function**, or **training scheduler**.



Results of using Curriculum Learning over Standard Training

- Curriculum Learning improves generalization.
- The improvement is consistent across random seeds
- Curriculum guides optimization to better minima



Plot : Validation accuracy comparison between Baseline, Curriculum Learning (CL), and Anti-Curriculum (Anti-CL) regimes.

Questions You Might Have About The Shape Classification Experiment

- Why choose the switch epoch (e.g., 128)?
- How is evaluation done? What is the test set?
- What are the hyperparameters?
- How is the dataset prepared?

Click Link to check out the full implementation here

Curriculum Learning: One Idea, Many Implementations

- Curriculum Learning encompasses diverse training strategies in the literature.
- General framework for curriculum design

Difficulty Measurer + Training Scheduler

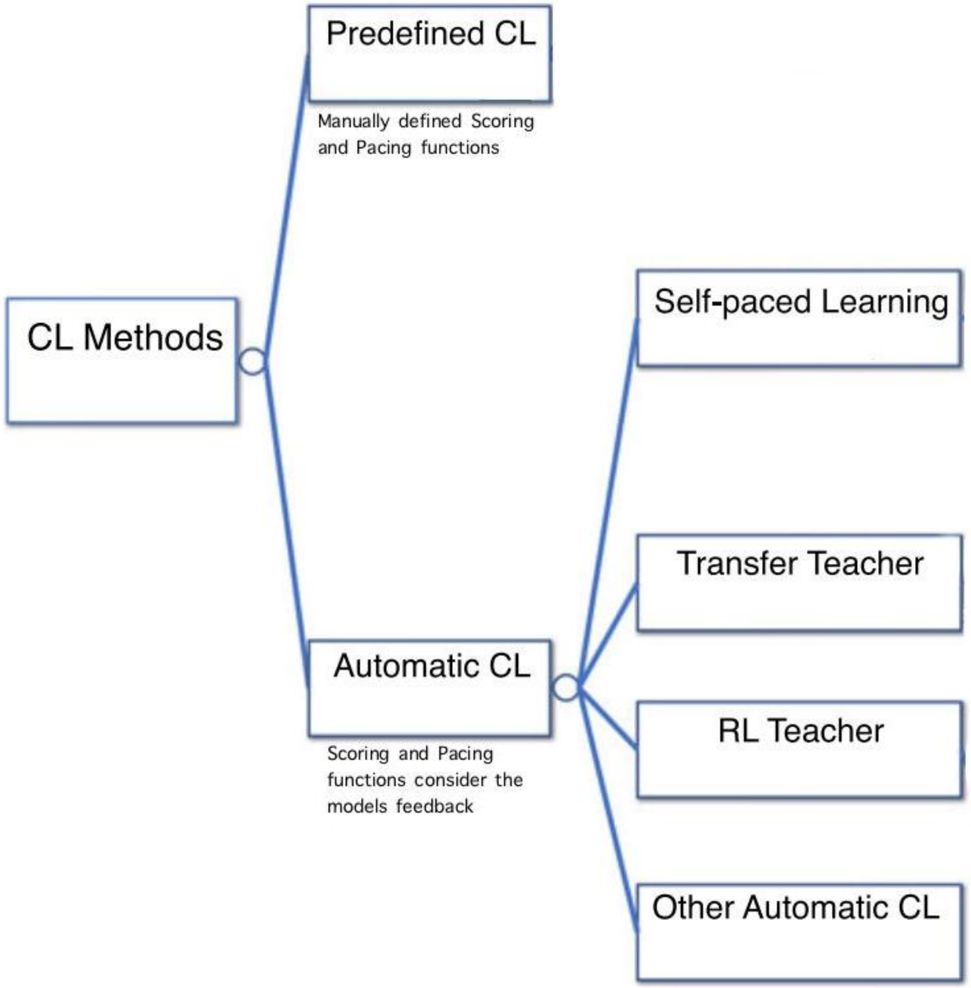


Figure : Taxonomy of Curriculum Learning methods [2]

Predefined Curriculum Learning

Instances where scoring and pacing functions are either human defined or fixed,

- ❑ Need expert domain knowledge
- ❑ Human defined
- ❑ fixed
- ❑ Ignore model feedback

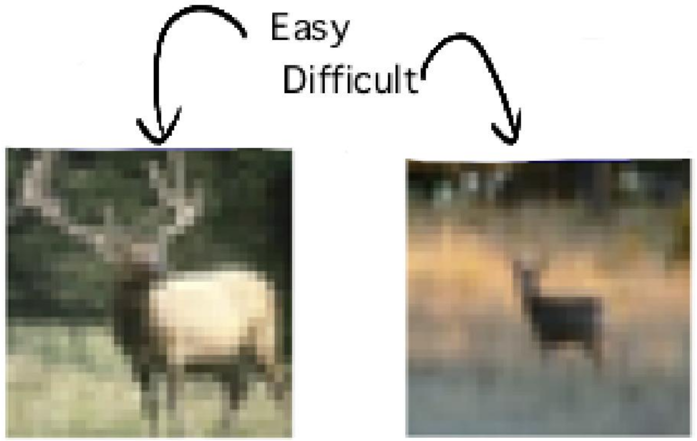


Figure : Deer class for cifar10 dataset

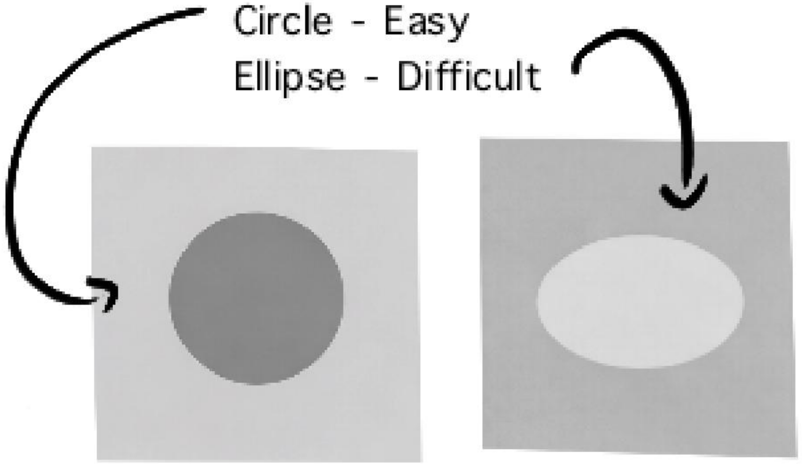


Figure : Ellipse class from : Bengio et al. (2009), *Curriculum Learning* [1]

Predefined Scoring functions in CV (Computer Vision)

Facial expressions intensity as a difficulty measure [3]

Task : Facial expression recognition

Facial expressions vary in intensity
High-intensity expressions are easier to recognize than subtle ones

(big smile)
easy to recognize

c) High-intensity smiles to low-intensity smiles displayed by an Eastern Asian female

Figure : Facial expression intensity

subtle smile
ambiguous, noisy

Facial expression sequences progress from **neutral** → **peak emotion**

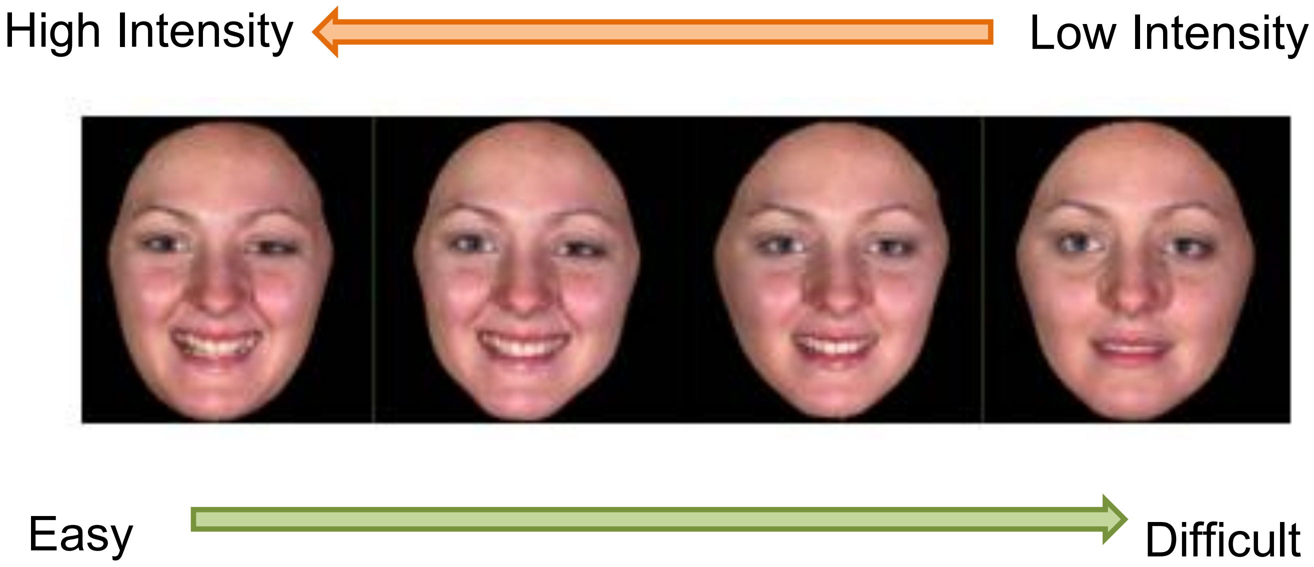


Figure : Facial expression intensity[3]

Results

- ❑ Improved **recognition accuracy** compared to random training
- ❑ Better **generalization** across subjects and datasets
- ❑ More reliable recognition of **subtle, low-intensity expressions**

Human response time as a difficulty measure [4]

Human response time = difficulty signal

❑ Goal
To quantify how difficult an image, using human behavior rather than model heuristics.

❑ Intuition
Instead of defining difficulty by, number of objects, clutter, occlusion.

Quantify visual difficulty by recording the time required for a human to identify a target object.

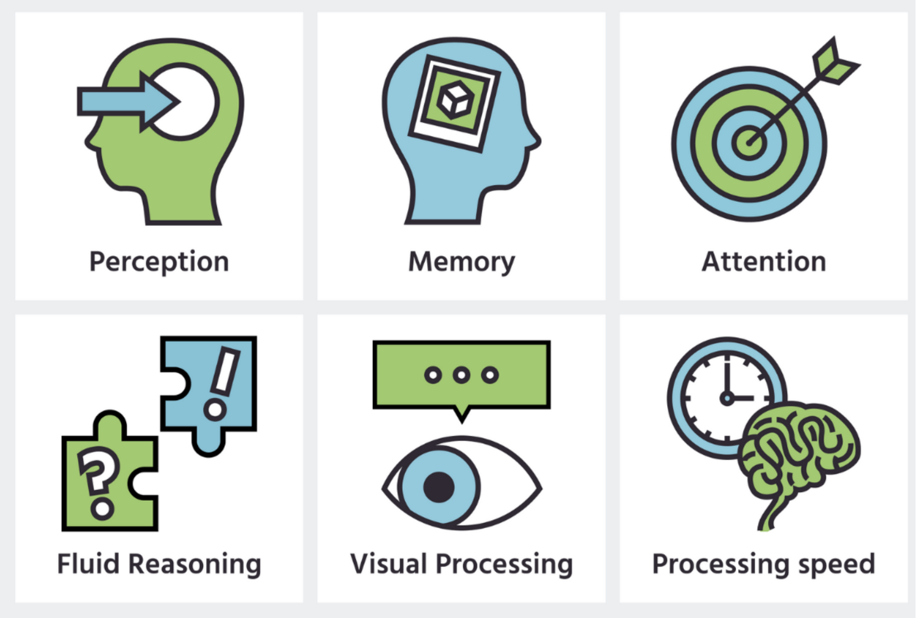


Figure : Cognitive Processes associated with the Visual Search Task

Figure : Human response time as a grounded difficulty signal [4]

- ❑ Human Visual Search

Output : human reaction times per image.

- ❑ Constructing the Difficulty Scoring Function

Response times are:
normalized across users
averaged per image

- ❑ Scoring Function/Difficulty

A continuous difficulty score (≈ 2.7 → 3.8)

Validating Human Response Time as a Grounded Difficulty Signal

The difficulty scores that are derived from human response times are compared against model mAP (mean average precision) of a class.

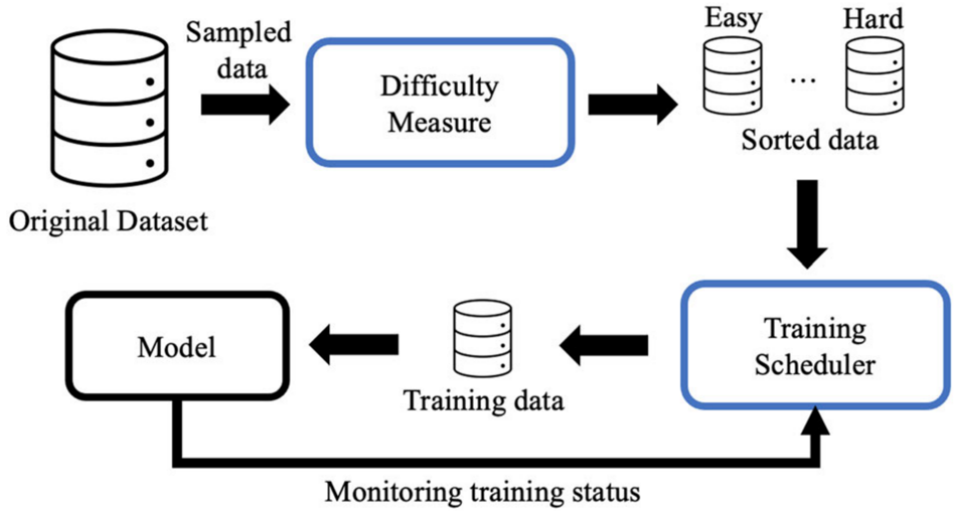
humans find *easy* = high mAP

Class	Score	mAP	Class	Score	mAP
bird	3.081	92.5%	bicycle	3.414	90.4%
cat	3.133	91.9%	boat	3.441	89.6%
aeroplane	3.155	95.3%	car	3.463	91.5%
dog	3.208	89.7%	bus	3.504	81.9%
horse	3.244	92.2%	sofa	3.542	68.0%
sheep	3.245	82.9%	bottle	3.550	54.4%
cow	3.282	76.3%	tv monitor	3.570	74.4%
motorbike	3.355	86.9%	dining table	3.571	74.9%
train	3.360	95.5%	chair	3.583	64.1%
person	3.398	95.2%	potted plant	3.641	60.7%

humans find *hard* = low mAP

Predefined Scoring functions in NLP (Natural language processing)

Sentence	Length
Thank you very much!	4
Barack Obama loves ...	13
My name is ...	6
What did she say ...	123



Sentence Length as Difficulty

Sentence length alone ≠ difficulty

❑ Intuition
Longer sentences require modeling longer dependencies

❑ Goal
Map each sentence to a scalar difficulty score ∈ [0,1]

The paper defines,

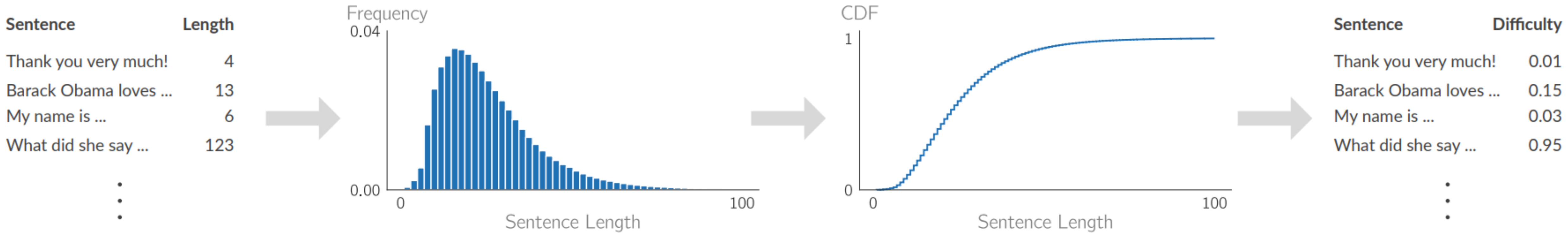
"A sentence is considered difficult if it is longer than most sentences in the dataset, This difficulty score depends on the dataset [5]"

Sentence	Length	Difficulty
Thank you very much!	4	0.01
Barack Obama loves ...	13	0.15
My name is ...	6	0.03
What did she say ...	123	0.95

Table : Sentence length as a normalized difficulty score [5]

Sentence Length as Difficulty

Figure : sentence difficulty pipeline [5]



- ❑ Token count Compute sentence lengths
- ❑ Histogram of sentence lengths' in the dataset
- ❑ Normalize Convert histogram to empirical CDF (**cumulative distribution function)**
- ❑ Difficulty = 0.15 sentence is longer than 15% of the dataset

Word Rarity as Difficulty

Word rarity = difficulty

❑ Intuition
Training examples are harder when they contain rare words, because the model sees them fewer times during training.

❑ Goal
rank words by their rarity

How do we find the rare words ?

Word Rarity as Difficulty for Neural Machine Translation (NMT)

- ❑ Task = Neural Machine Translation (English → Czech)
- ❑ Model = Encoder–Decoder architecture
- ❑ Data = Parallel corpus

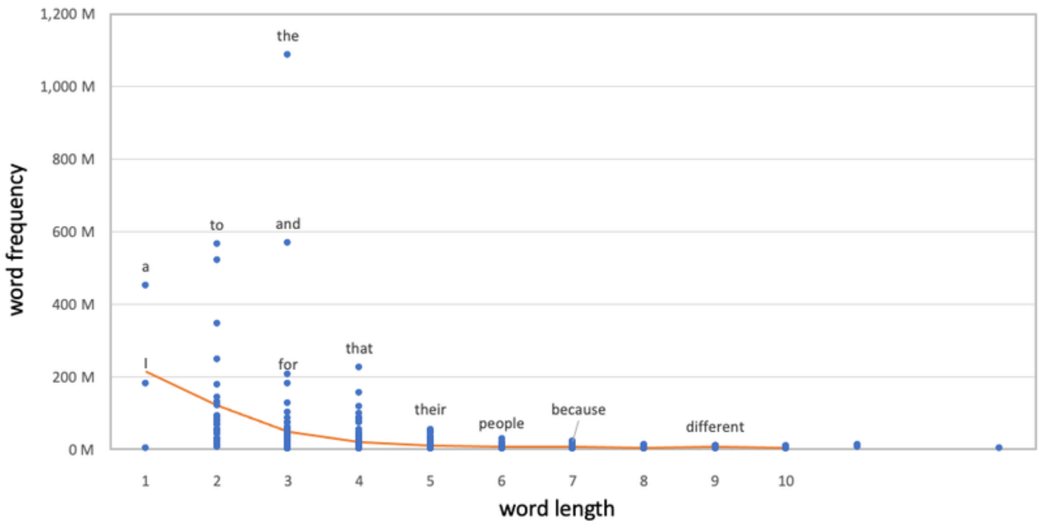
$(x_{\text{eng}}, y_{\text{czech}})$

"Rare words and long sentences are harder for an NMT model, especially early in training." [6]

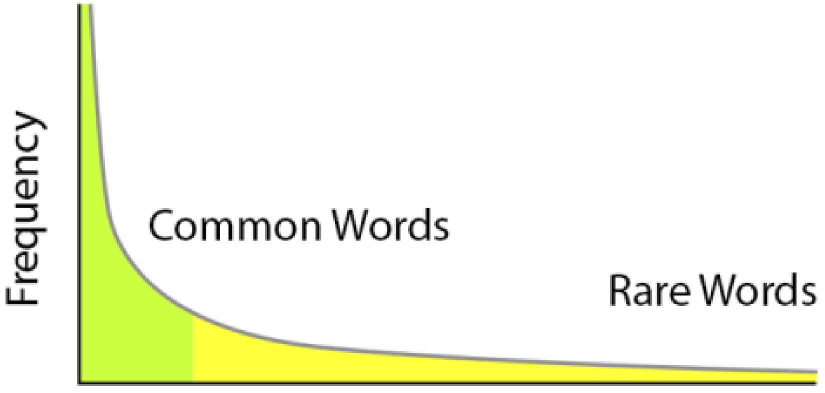
- ❑ Source sentence (English)
- ❑ Target sentence (Czech)

Curriculum set-up

simpler sentence pairs → difficult sentence pairs



Plot : Relationship between word length and corpus frequency.

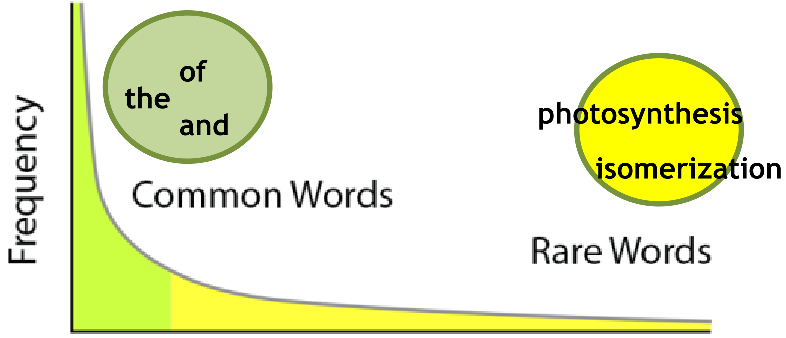


- ❑ Assign word ranks

rank 1 → common word
Rank 1000 → rare word

- ❑ Build word frequency lists
- ❑ Count how often each word appears in the English corpus

Word Rarity



- Compute sentence difficulty function

Method 1
Highest word rank

difficulty(sentence) = $\max(\textit{word ranks})$

One rare word → hard sentence

Method 2
Max word rank

difficulty(sentence) = *target* (Czech) *OR source* (English)

Consider both English and Czech then, a sentence is hard if either side has rare words

Method 3
Combined rank

difficulty(sentence) =
target (Czech) *AND source* (English)

In a joint vocabulary (English + Czech), Use maximum rank over both sides

Predefined Difficulty Measurers : An Overview

Difficulty Measurer	Domain	Difficulty Intuition
Sentence Length	NLP	Shorter sentences have simpler structure and are easier to learn
Word Rarity	NLP	Frequent words are easier than rare or unusual vocabulary
Expression Intensity	CV	Stronger/exaggerated expression are easier to classify
Human Response Time	CV	Longer human reaction times indicate higher visual difficulty

The difficulty measures discussed here represent only a subset of predefined curriculum learning strategies.

Prior work has proposed many additional difficulty measurers across data types,

- ❑ structural complexity
- ❑ distributional diversity
- ❑ noise estimation
- ❑ domain knowledge
- ❑ and human-centered annotations

Difficulty Measurer*	Angle	Data Type
Sentence length [86], [107]	Complexity	Text
Number of objects [122]	Complexity	Images
# conj. [50], #phrases [113]	Complexity	Text
Parse tree depth [113]	Complexity	Text
Nesting of operations [131]	Complexity	Programs
Shape variability [6]	Diversity	Images
Word rarity [50], [86]	Diversity	Text
POS entropy [113]	Diversity	Text
Mahalanobis distance [14]	Diversity	Tabular
Cluster density [11], [31]	Noise	Images
Data source [10]	Noise	Images
SNR/SND [7], [89]	Noise	Audio
Grammaticality [66]	Domain	Text
Prototypicality [113]	Domain	Text
Medical based [44]	Domain	X-ray film
Retrieval based [18], [82]	Domain	Retrieval
Intensity [30]/Severity [111]	Intensity	Images
Image difficulty score [106], [114]	Annotation	Images
Norm of word vector [68]	Multiple	Text

Table : types of pre-defined difficulty measures/ scoring functions [2]

Predefined Pacing functions

A pacing function (also called a training scheduler or competence function) → how the training data exposure changes over time during training.

Discrete pacing

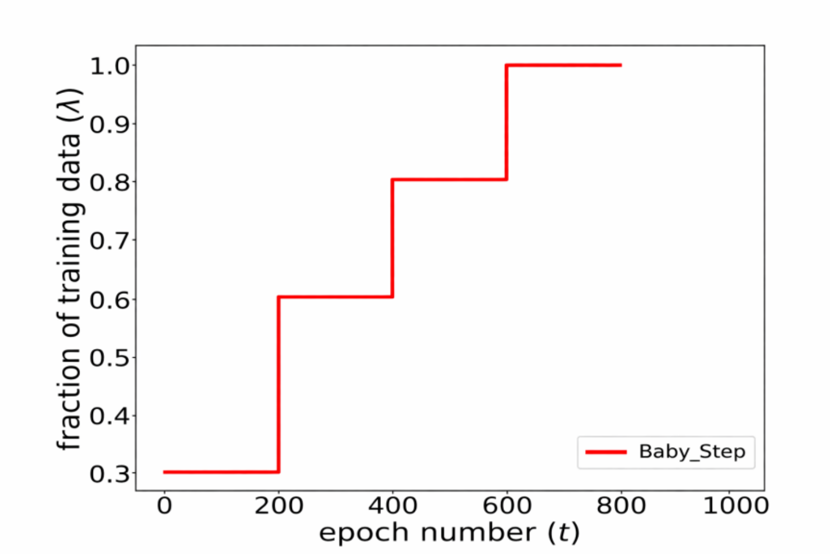
Training data is partitioned into buckets Training starts with the easiest bucket
Harder buckets are merged progressively after fixed epochs or convergence

Continuous pacing

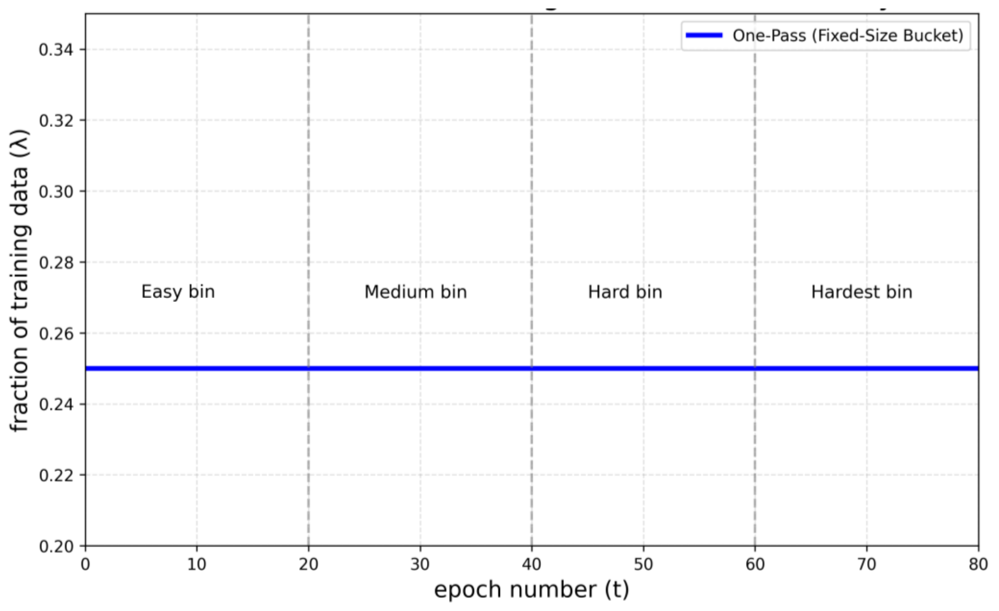
A function that maps training time to the proportion of easiest samples used at each epoch, gradually expanding the training set until all data is included.

Predefined Discreet Training Schedulers

Baby Step Scheduler



- ❑ Sort data from easy to hard
- ❑ Split into difficulty-based buckets
- ❑ Start training with the easiest bucket
- ❑ Progressively merge harder buckets over time

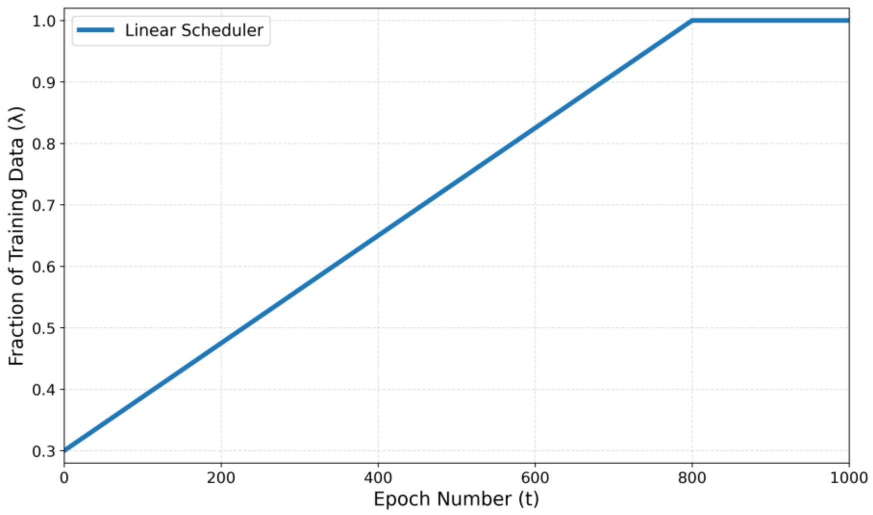


One-Pass Scheduler

- ❑ Data is also bucketed from easy to hard
- ❑ Train on one bucket at a time
- ❑ Discard the current bucket when moving to a harder one

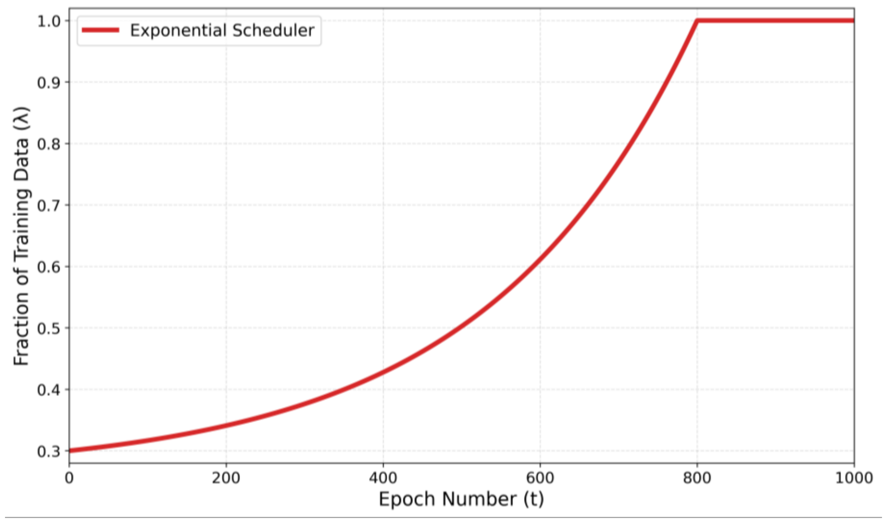
Predefined Continuous Training Schedulers

Linear Scheduler



- ❑ Training data increases linearly over time
- ❑ Equal amount of new data added each epoch
- ❑ Simple and intuitive baseline

Exponential Scheduler



- ❑ Training data increases slowly at first
- ❑ Growth accelerates later in training
- ❑ Gives easier samples more training time

References

[1] Bengio, Yoshua, et al. "Curriculum learning." *Proceedings of the 26th annual international conference on machine learning*. 2009.

[2] Soviany, Petru, et al. "Curriculum learning: A survey." *International Journal of Computer Vision* 130.6 (2022)

[3] Gui, Liangke, Tadas Baltrušaitis, and Louis-Philippe Morency. "Curriculum learning for facial expression recognition." *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017.

[4] Tudor Ionescu, Radu, et al. "How hard can it be? Estimating the difficulty of visual search in an image." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[5] Platanios, Emmanouil Antonios, et al. "Competence-based curriculum learning for neural machine translation." *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)*. 2019.

[6] Kocmi, Tom, and Ondrej Bojar. "Curriculum learning and minibatch bucketing in neural machine translation." arXiv preprint arXiv:1707.09533 (2017).