Chemical Language Model

Jatin Kumar

Alexandru Andrița

Sambit Basu

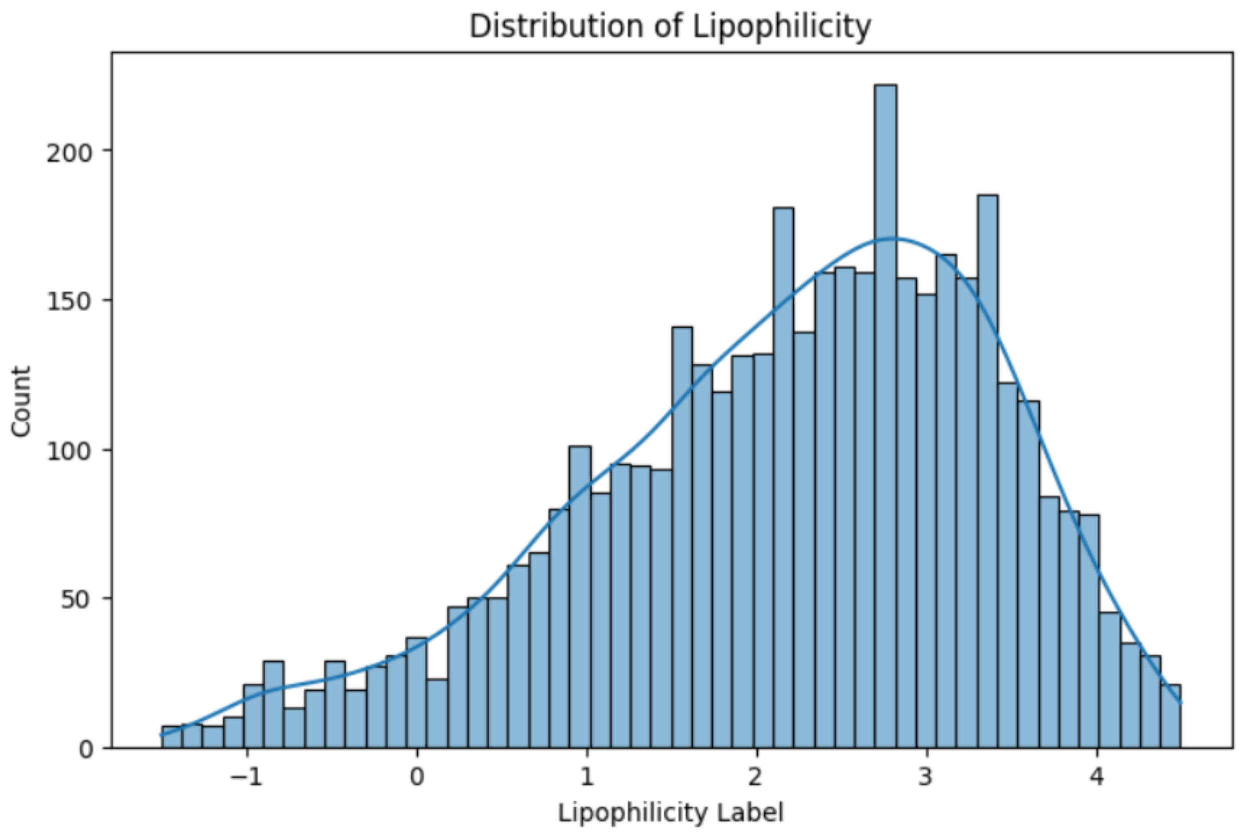
Table of Contents

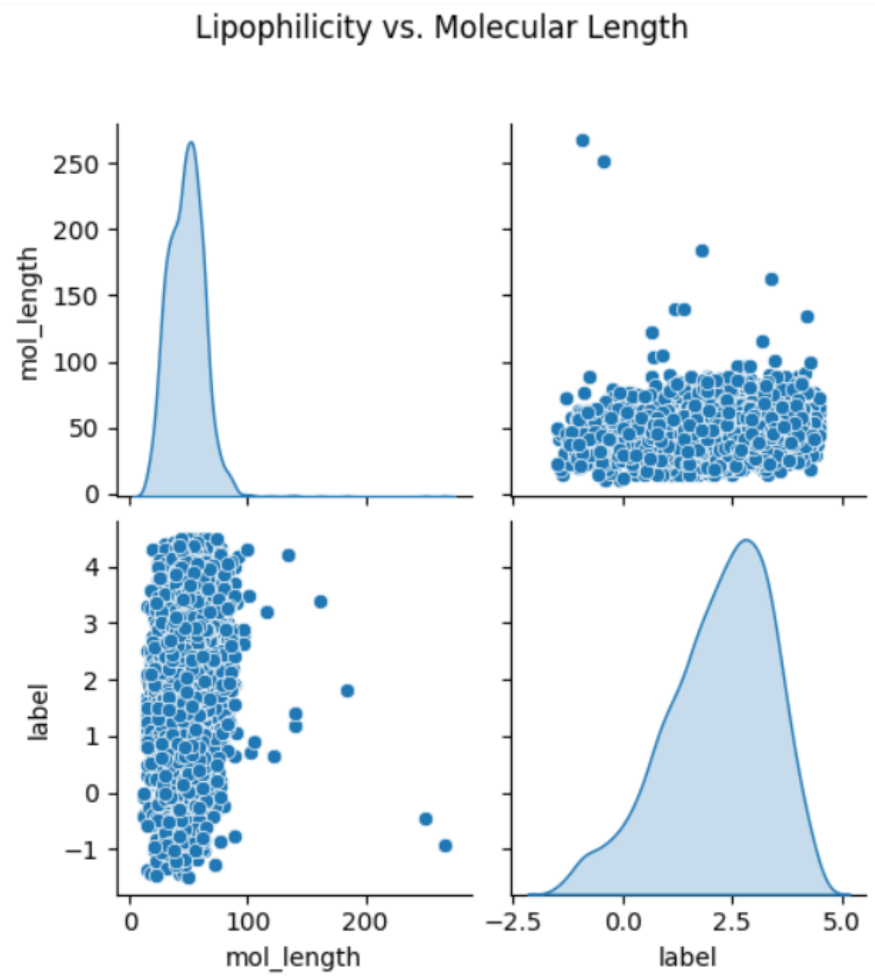
UNIVERSITÄT DES SAARLANDES

Fine-tuning Chemical Models

Data exploration

UNIVERSITÄT DES SAARLANDES





Preprocessing & Model Configuration

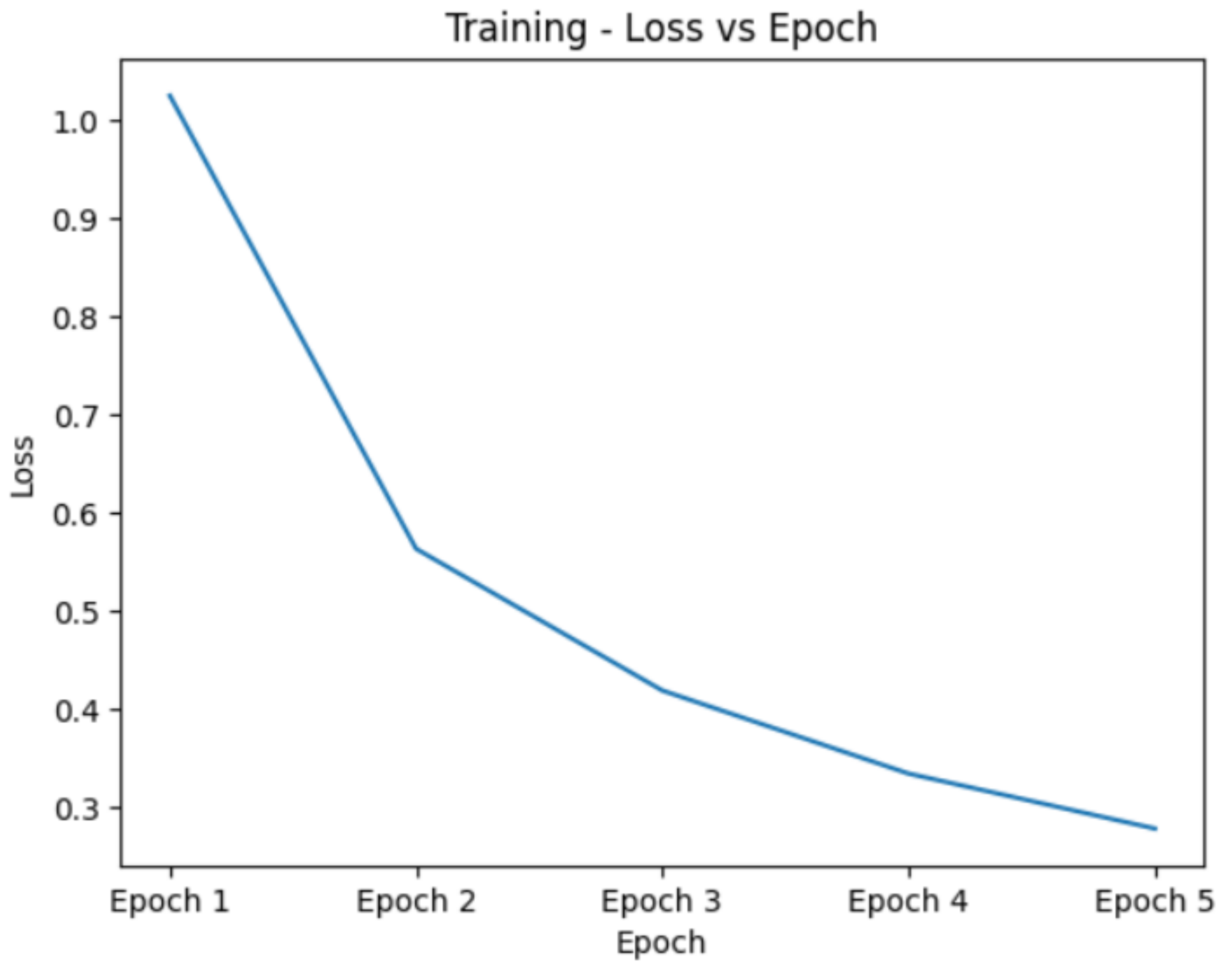
- **SMILESDataset Class** – handles SMILES strings, the attention mask and their corresponding values
- **Pre-trained model** – using as base a pre-trained model (*MoLFormer-XL*) on the Lipophilicity dataset
- **Tokenization** – tokenizes SMILES strings with padding of a maximum length of 300
- **Data loaders** – used in batch-wise data loading for training and testing
- **Regression head** – defines regularization techniques and passes input to the model
- **AdamW optimizer** – adjusts model parameters during training

Evaluation metrics (Initial training)

- Mean Squared Error
- R2 score

Epoch	Training MSE
Epoch 1	1.024
Epoch 2	0.563
Epoch 3	0.419
Epoch 4	0.334
Epoch 5	0.279

Evaluation MSE	R2 score
0.512	0.654



Unsupervised Finetuning

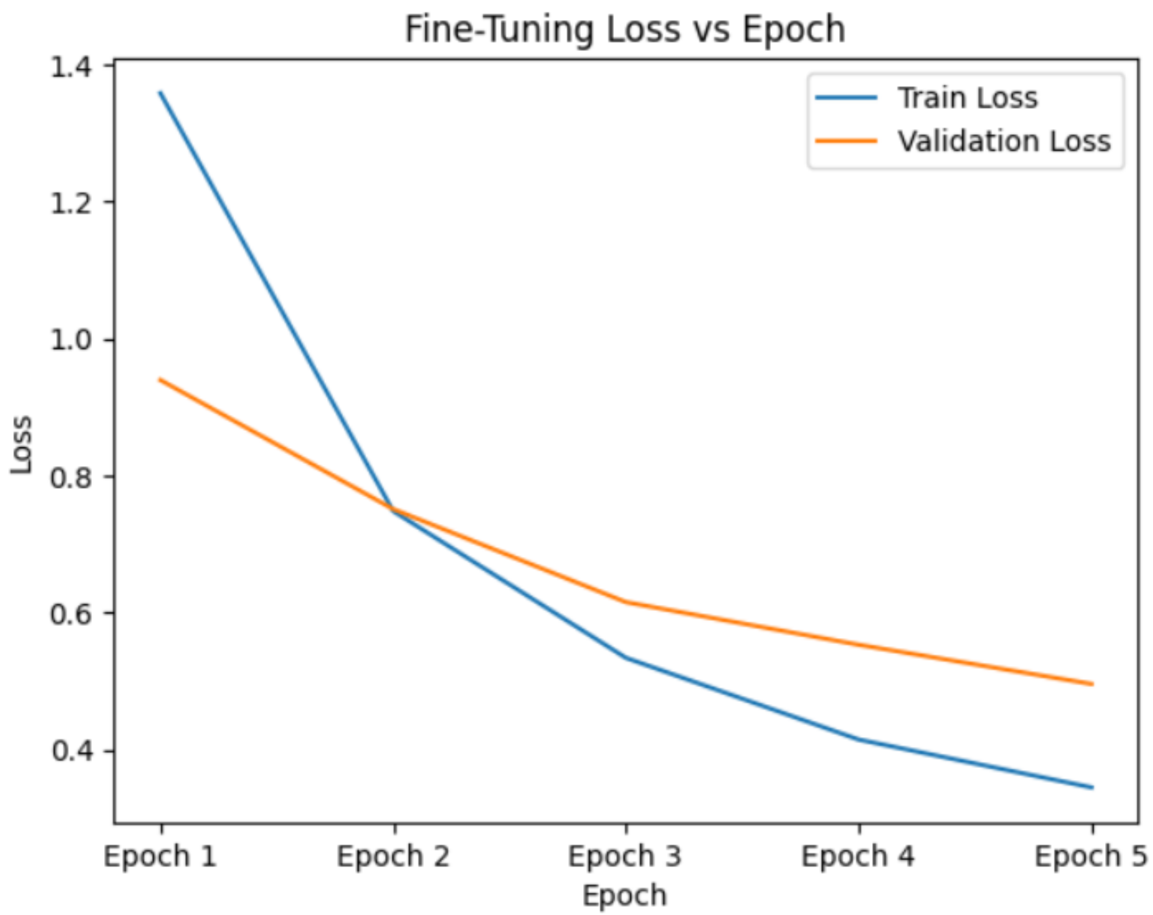
UNIVERSITÄT DES SAARLANDES

- AutoModelFromMaskedLM – pre-trained transformer designed for Masked Language Modelling
- AdamW optimizer
- Purpose of unsupervised fine-tuning is for the model to learn to predict the missing tokens in a sequence given as input.
- $R2 = 0.675$

Fine-Tuning on the regression task

- Regression model
- AdamW optimizer

Epoch	Training MSE	Evaluation MSE
Epoch 1	1.358	0.936
Epoch 2	0.748	0.751
Epoch 3	0.534	0.615
Epoch 4	0.414	0.553
Epoch 5	0.344	0.495



Influence function-based Data Selection

Influence Function-Based Data Selection

UNIVERSITÄT DES SAARLANDES

- *Objective* : Improving model performance by selecting external data
- *Problem* : Not all external data points contribute positively for our model
- *Methodology* : Using "**Influence Functions**[1]" to identify the most impactful data points

1. [Koh & Liang (2017)](https://arxiv.org/abs/1703.04730)

Influence Function-Based Data Selection

Steps:

1. Preprocess the external data
2. Compute test gradients for each external data points
3. Using **LiSSA** approximation[2] to calculate the inverse Hessian Vector product

2. [(Agarwal et al., 2016)](https://arxiv.org/abs/1602.03943)

Influence Function-Based Data Selection

Steps (continued):

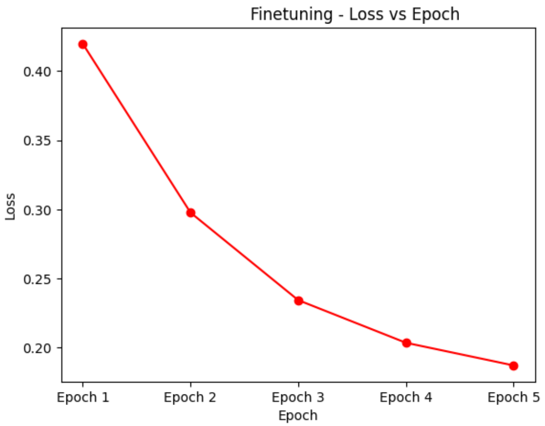
4. Compute Influence Scores by calculating the dot product of the test gradients and iHVP
5. Select top-k samples from the external dataset
6. Retrain the model from task 1 (unsupervised fine training) on the augmented dataset
7. Compare the results with task 1

Evaluation and finding the best k value

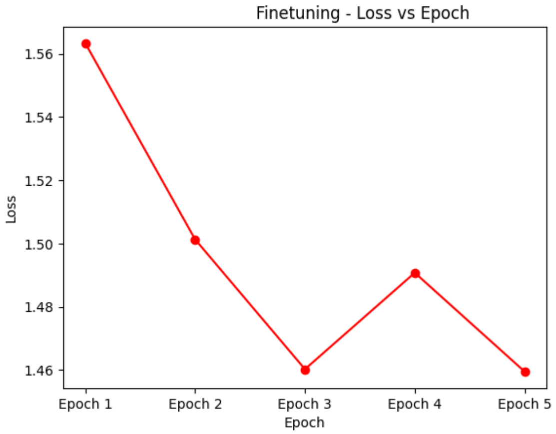
K value	Evaluation MSE	R2 Score
K = 64	**0.443**	**0.699**
K = 100	0.461	0.688
K = 278	1.488	0.001

Evaluation MSE from task 1	R2 score
0.512	0.654

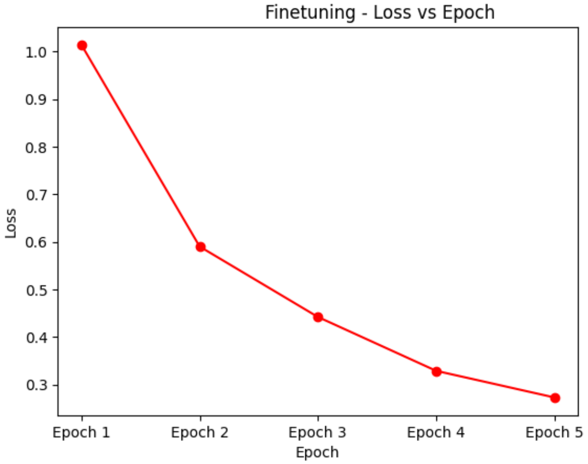
Training Loss Curves



k = 64



k = 278



k = 100

Data Selection Methods

Curriculum Learning

Active learning via Uncertainty Sampling

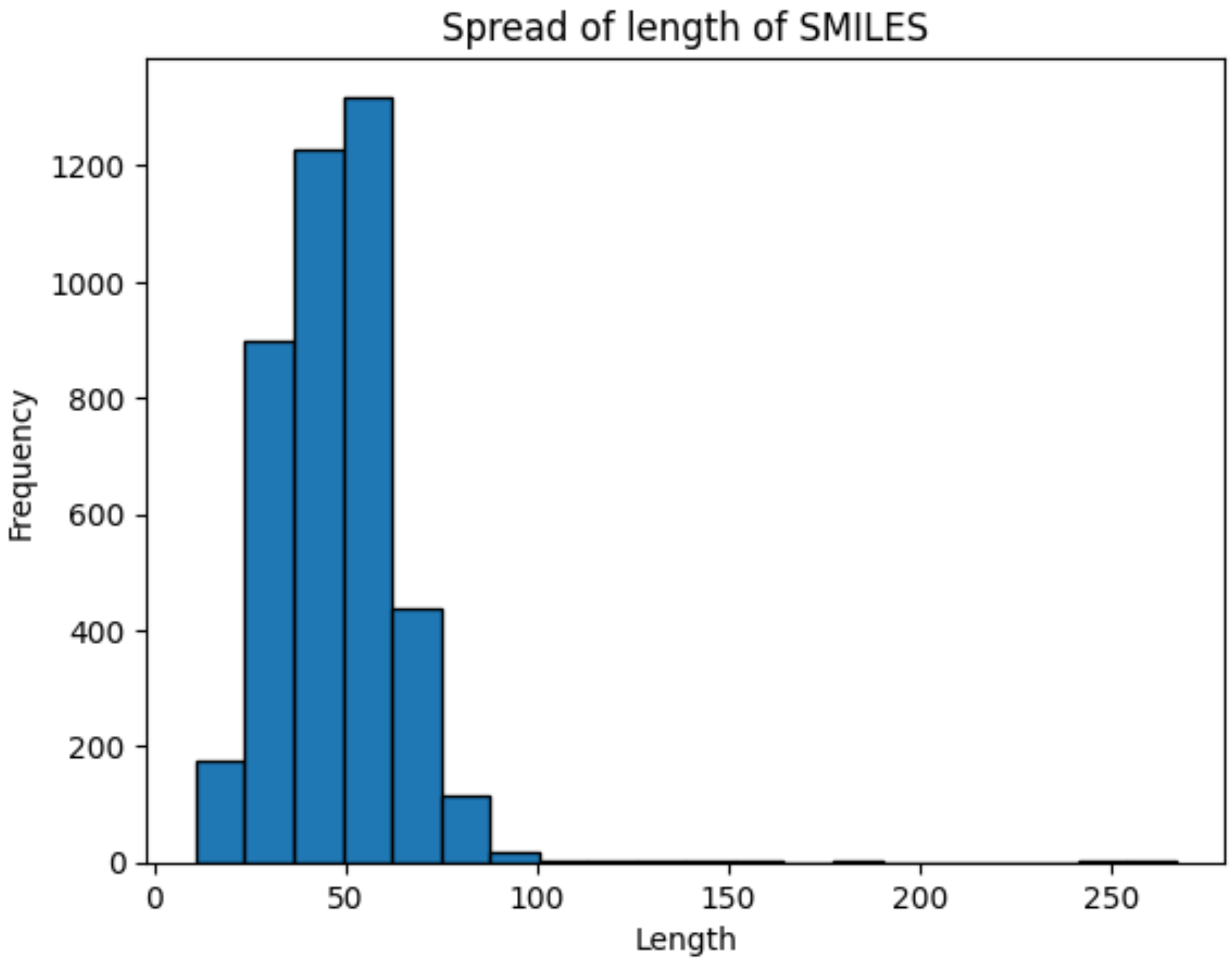
Curriculum Learning

- Inspired by human learning
- Easier topics, then harder topics
- SMILES string represents molecules
  - Each letter is an atom
- Shorter string → Simpler molecule → Easier problem
- Longer string → Complex molecule → Harder problem

Yoshua Bengio, Jérôme Louradour, Ronan Collobert,and Jason Weston. 2009. Curriculum learning. InProceedings of the 26th Annual International Conference on Machine Learning, ICML ’09, page 41–48,New York, NY, USA. Association for Computing Machinery.

SMILES dataset

UNIVERSITÄT DES SAARLANDES



Bin Range	Frequency	Cum. Frequency
(10.744, 23.8]	174	174
(23.8, 36.6]	897	1071
(36.6, 49.4]	1228	2299
(49.4, 62.2]	1319	3618
(62.2, 75.0]	452	4070
(75.0, 87.8]	102	4172
(87.8, 100.6]	16	4188
(100.6, 113.4]	3	4191
(113.4, 126.2]	2	4193
(126.2, 139.0]	1	4194
(139.0, 151.8]	2	4196
(151.8, 164.6]	1	4197
(177.4, 190.2]	1	4198
(241.4, 254.2]	1	4199
(254.2, 267.0]	1	4200

https://huggingface.co/datasets/scikit-fingerprints/MoleculeNet_Lipophilicity

Curriculum Learning

We created a DataLoader, so that during early epochs, easier problems will be fed for training. Slowly, with increasing epochs, harder problems will be introduced.

$$Threshold = d_{min} + \left(d_{max} - d_{min}\right) \times \frac{e_{current}}{e_{total}}$$

d_{min}=minimum difficulty (267)

d_{min}=maximum difficulty (11)

$e_{current}$=current epoch

e_{total}=total number of epochs

Kumar, M., Packer, B., & Koller, D. (2010). "Self-Paced Learning for Latent Variable Models." In Advances in Neural Information Processing Systems 23, 1189–1197

Training Error

0.0939

Evaluation MSE

0.3752

Evaluation R^2

0.746

Active Learning via Uncertainty Sampling

- Identifies most uncertain data points
- Uncertainty introduced by stochasticity of dropout layer
- Uncertainty computed by calculating Standard Deviation of predictions across multiple forward passes

Yarin Gal and Zoubin Ghahramani. 2016. Dropoutas a bayesian approximation: Representing model uncertainty in deep learning.
Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017.Deep bayesian active learning with image data.

Active Learning via Uncertainty Sampling

- Number of selected indices for sampling is a hyper-parameter
- Indices selected in training mode
  - To activate dropout layer
- DataLoader created to feed the selected indices for training.

Yarin Gal and Zoubin Ghahramani. 2016. Dropoutas a bayesian approximation: Representing model uncertainty in deep learning.
Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017.Deep bayesian active learning with image data.

No. of samples	Training MSE	Evaluation MSE	R^2
25	0.0941	0.5184	0.65
64	0.084	0.5053	0.6597
128	0.1048	0.4545	0.6923
256	0.0965	**0.4295**	**0.7092**
512	0.1197	0.5012	0.6607
1000	0.1013	0.4537	0.6928

Fine tuning strategies

BitFit

LoRA

(IA)³

BitFit

- Full fine-tuning is effective… but
  - Each pre-trained task has large model
  - Deployment difficulty $\propto$ No. of tasks
- **B**i**a**s-**t**erms **Fi**ne-**t**uning
- Just updates the bias terms
- Freezes all other weight updates
- Reduces trainable parameters to **0.1%** !!

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg.2022. Bitfit: Simple parameter-efficient fine-tuningfor transformer-based masked language-models.

Training Error

0.5896

Evaluation MSE

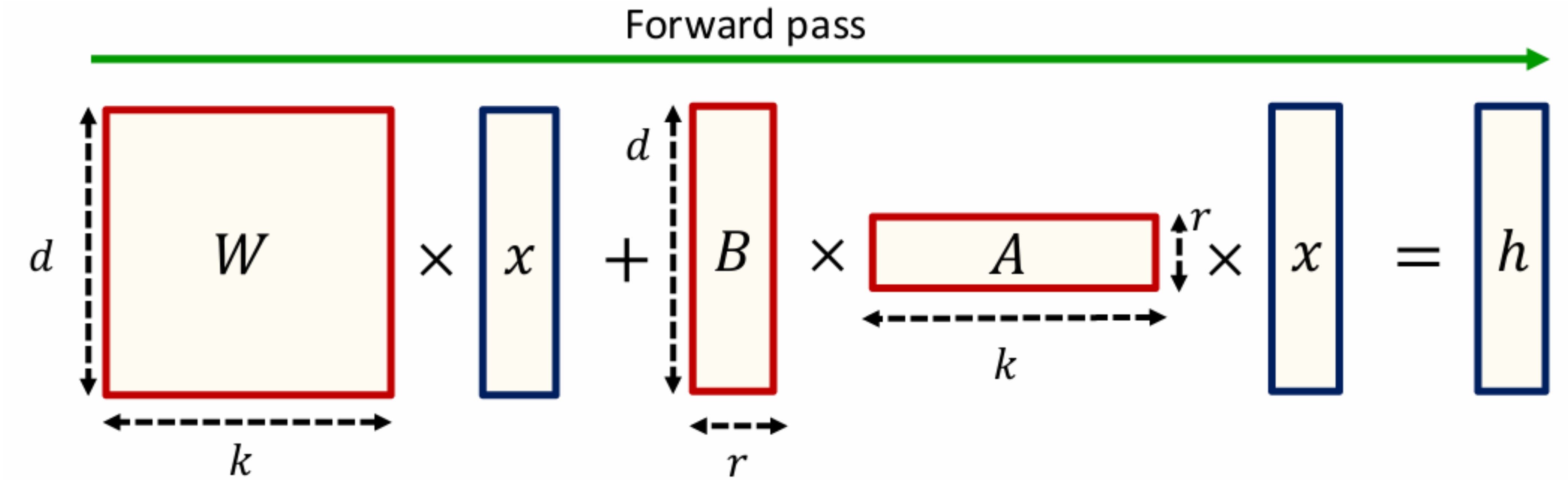
0.7127

Evaluation R^2

0.5176

LoRA

- **Low Rank Adaptation**
- Uses Low Rank decomposition of matrices



LoRA

- Weight matrix to update is decomposed into 2 matrices
- Only those matrices are updated
- r<<d and r<<k
- Instead of $d \times k$, the no. of trainable params become $r \times (d + k)$

Edward J. Hu, Yelong Shen, Phillip Wallis, ZeyuanAllen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, andWeizhu Chen. 2021. Lora: Low-rank adaptation oflarge language models.

r	Training MSE	Evaluation MSE	R^2
2	0.2386	0.5071	0.6567
4	0.2419	0.5421	0.633
8	0.2174	0.5071	0.6567
12	0.2087	**0.4857**	**0.6712**
16	0.2197	0.5028	0.6596

(IA)^3

- Base model is frozen
- Per-dimension scaling vector is learnt during training
- Only 1 parameter per hidden dimension is added
- Instead of updating all weights, only n parameters are trained (n: no. of hidden dimensions)

(IA)^3

- We multiply the pooled output with the learned scaling vector

```python
class IA3Adapter(nn.Module):
    def __init__(self, hidden_size):
        super().__init__()
        self.scale=nn.Parameter(torch.ones(hidden_size))

    def forward(self, hidden_states):
        return hidden_states * self.scale
```


Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mo-hta, Tenghao Huang, Mohit Bansal, and Colin Raffel.2022. Few-shot parameter-efficient fine-tuning isbetter and cheaper than in-context learning

Training Error

0.1614

Evaluation MSE

0.4702

Evaluation R^2

0.6817

Training done only on 5 epochs due to lack of compute

Evaluation Comparison

UNIVERSITÄT DES SAARLANDES

Task	Model/Technique	Evaluation MSE	R2 Score
Task1	MoLFormer with Regression head	0.495	0.673
Task2	Influence based data selection	0.443	0.699
Task3	**Curriculum learning**	**0.375**	**0.740**
	Uncertainty sampling	0.429	0.701
	LoRA	0.486	0.671
	BitFit	0.712	0.517
	IA3	0.470	0.682

Future Work and Improvements

UNIVERSITÄT DES SAARLANDES

- Fine-Tuning alternative pre trained models such as SMILES-BERT[5] , ChemBERTa[6]
- Try out different datasets such as OPERA Lipophilicity Dataset
- Use hybrid data selection techniques (Influence Selection for Active Learning (ISAL)[7])

[5]. [Wang et al., 2019] (https://arxiv.org/abs/1907.03338)
[6]. [Chithrananda et al., 2020] (https://arxiv.org/abs/2010.09885)
[7]. Influence Selection for Active Learning (Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, Conghui He)

Thank You