The Importance of Being Parameters: An Intra-Distillation Method for Serious Gains
Final Report

Brennan R. Dury, Yash Manne, Bhanu Sharma
University of Washington
`{dury, ymanne, bhanuuw}@uw.edu`

Reproducibility Summary

Scope of Reproducibility

The goal of this report is to reproduce “The Importance of Being Parameters: An Intra-Distillation Method for Serious Gains” (Xu et al, 2022) and evaluate the central finding that fine-tuning language models with intra-distillation loss leads to improved performance, the secondary finding that self-distillation and intra-distillation lead to greater parameter contribution balance, and the hypothesis that intra-distillation improves generalization of neural networks in domains outside natural language processing.

Methodology

We trained the machine translation and cross-lingual transfer models using the authors’ code. For Machine Translation, we used an NVIDIA V100 GPU, 8 vCPUs, and 30 GB RAM. The machine translation models required 3 GPU hours to train with intra-distillation and 1 GPU hour without. Model evaluation with SacreBLEU took approximately 1 minute on the GPU. For Zero-Shot Cross-Lingual Transfer, we used an NVIDIA T4 GPU, 2 vCPUs, and 13 GB RAM. Evaluation time on 10000 samples was 74 sec. The code for image classification was written from scratch, other than the intra-distillation loss functions provided by the authors. For image classification, we trained eight models on an Nvidia T4 Tensor Core GPU on Google Colab, all models took 4GB of RAM and 2GB of GPU RAM and the total runtime was 18 hours.

Results

For Machine Translation, intra-distillation increased SacreBLEU score by 6.72%. The NER and TDQIA tasks saw 1.28% and 5.214% increase after intra-distillation from their baseline model results. In addition, for an image classification task, intra-distillation models averaged 0.00564 and self-distillation models averaged 0.005635 smaller standard deviation of normalized parameter sensitivity. Intra-distillation loss also lead to a 0.098 improvement in test negative log-likelihood across all models trained.

What was Easy

It was easy to access and preprocess data for the designated tasks. Additionally, training and evaluating the default model with intra-distillation was very straightforward.

What was Difficult

Without sufficient knowledge of the Fairseq codebase, it was difficult to follow the inheritance of classes which made it more difficult than expected to change any parameters or modify small subsections. Specifically, arguments were passed using a special configuration object that lacked sufficient documentation to understand.

Communication with Original Authors

We had no correspondence with the original authors.

1 Introduction

In “The Importance of Being Parameters: An Intra-Distillation Method for Serious Gains,” Haoran Xu, Philipp Koehn, and Kenton Murray posit that a more balanced contribution of parameters to the output of the model often leads to better generalization. Then they propose a term that can be added to the loss function of arbitrary parameterized models to encourage the optimization process to choose models with a balanced contribution of parameters.

First, they define the sensitivity of a model to a parameter set S as the absolute difference in loss between the model and the model with S set to zero. From this, they define the degree of contribution balance as the standard deviation of the sensitivity to each parameter.

Next, they show as an empirical case study that iterations of knowledge distillation decrease the standard deviation of the sensitivity to each parameter.

Next, they define intra-distillation loss, a practical term to add to the loss function of arbitrary classification tasks for parameterized models to measure parameter sensitivity balance. K forward passes are run for dropping K sets of randomly sampled parameters. Then the KL divergences between each output distribution and the mean of their output distributions are averaged.

Finally, they evaluate test loss on three tasks after fine-tuning pre-trained models with and without intra-distillation loss, showing that using this method leads to decreased test loss.

2 Scope of Reproducibility

The central claim of the paper is that balancing the contribution of model parameters leads to significantly better generalization performance. Thus, we want to validate that balancing the contribution of parameters does in fact lead to better performance. In particular, we will evaluate the following claims:

1. For the small transformer finetuned on IWSLT’14 De→En, the SacreBLEU of the model trained with intra-distillation is greater than the SacreBLEU of the model trained without intra-distillation.

2. For XLM-RBase finetuned on TyDiQA, the test loss with intra-distillation is less than the test loss without intra-distillation.

Additionally, the paper claims that the balanced parameter contribution explains ‘dark knowledge’ transfer in knowledge distillation (Hinton et al., 2015). Specifically, in the case of self-distillation, a subcase of knowledge distillation in which both the teacher and student models have the same architecture, the authors of this paper show empirical evidence that a model’s parameter sensitivities become more balanced after successive rounds of self-distillation. Then they hypothesize that balanced parameter contribution explains the effectiveness of self-distillation. However, they do not show that intra-distillation fully explains the effectiveness of self-distillation. We will test this hypothesis by performing rounds of self-distillation with a model trained with intra-distillation. In particular, we intended to evaluate the following claims:

1. For the small transformer finetuned on IWSLT’14 De→En with intra-distillation, rounds of self-distillation do not decrease test loss.

2. For XLM-RBase finetuned on TyDiQA with intra-distillation, rounds of self-distillation do not decrease test loss.

We were unable to reproduce these particular findings, for reasons explained in the difficulties section. We find a substitute below.

The authors hypothesize that intra-distillation could improve neural network generalization in fields other than natural language processing, but did not investigate this claim. We investigate this claim for the task of image classification on Oxford-IIIT Pet (Parkhi et al., 2012) with ResNet50 (He et al, 2015). The authors additionally claim that the benefits of self-distillation are explained by parameter sensitivity, but they do not demonstrate that intra-distillation fully captures the benefits of self-distillation. So, we additionally investigate the combined effects of intra-distillation and self-distillation, as well as label smoothing, which is equivalent to a weighted KL divergence from a uniform model, and thus a form of knowledge distillation from a uniform model. (Meister et al., 2020)

2.1 Addressed Claims from the Original Paper

1. More-balanced model results in better performance across sub-tasks of machine translation, and zero-shot cross-lingual transfer as evaluated by sacreBLEU and average F1 score compared to baseline models of TransformerBase and XLM-RSmall, respectively.
2. Intra-distillation and self-distillation both lead to more balanced parameter contributions. Speculating that balanced parameter contribution explains the 'dark knowledge' transfer in knowledge distillation, intra-distillation captures the benefits of self-distillation.
3. Untested hypothesis: Intra-distillation improves the generalization of neural networks trained for tasks outside of natural language processing.

3 Methodology

To reproduce the result that intra-distillation leads to decreased test loss, we plan to use the author's code from their publicly-available repository. The code is present as separate bash scripts for each of the tasks of Machine Translation and Zero-Shot Cross-Lingual Transfer. Each of these bash scripts references tens of other bash and Python scripts, each with its own documentation. PyTorch (Paszke et al., 2019) is used for model training and data is available on Google Drive. We will be training models using Google Cloud.

To evaluate the claim that balanced parameter contribution explains the effectiveness of self-distillation, we will implement self-distillation in PyTorch (Paszke et al., 2019). That is, after training each model, we will use their outputs on the training data to relabel the target training data probability distribution and rerun training. We will perform 2 rounds of self-distillation.

For image classification, we train 8 models in total for each combination of the binary decision of whether or not to perform intra-distillation, self-distillation, and label smoothing. The architecture for all models is ResNet50 (He et al, 2015), with hyperparameters described in the Models section of this report. For intra-distillation, we set K=3, N=4000, q=5, p=10, alpha=5, with parameters names consistent with the original paper. For self-distillation, we weight the cross entropy from the teacher equally with the log-likelihood.

Training each model for 80 epochs, we select the iteration with the lowest validation loss for evaluation. We measure the negative log-likelihood of the test data and measure the standard deviation of parameter sensitivity. For measuring the parameter sensitivities, we average the parameter sensitivities over all batches in the training set.

3.1 Model Descriptions

The authors define an intra-distillation term to add to the loss function of arbitrary classification tasks for parameterized models to encourage parameter sensitivity balance. K forward passes are run for dropping K sets of randomly sampled parameters. Then the KL divergences between each output distribution and the mean of their output distributions are averaged

Multiple standard models are used for benchmarking:

1. Machine Translation
    (a) Small Transformer Architecture (Vaswani et al. 2017):
        i. 49.98M parameters
        ii. FFN dimension: 1024
        iii. Attention dimension: 512
        iv. 4 attention heads
        v. Batch size of 4096 tokens
        vi. The maximum learning rate is 0.0005. The optimizer is Adam (Kingma and Ba, 2014) with inverse_sqrt learning rate scheduler and weight decay of 0.0001. The maximum training update is 50K with 8K warm-up steps. At inference time, they used beam search with a width of 5 and a length penalty of 1.
        vii. Jointly trains a 12K bilingual vocabulary by using SentencePiece (Kudo and Richardson, 2018) for each language pair after filtering out the training pairs whose length ratio is larger than 1.5 or one of length is longer than 175 tokens.
    (b) Dropout rates:
        i. 0.1 for attention layers

        ii. 0.3 for feed-forward (FFN) layers

2. Zero-Shot Cross-Lingual Transfer:
    (a) NER:
        i. 48 languages
        ii. XLM-RLarge: 550M parameters
        iii. The max length is 128. We train the model for 10 epochs with learning rate 2e-5, batch size 8 and gradient accumulation 4.
        iv. BERTBase (L=12, H=768, A=12, Total Parameters=110M):
        i. Maximum sentence length of 128.
        ii. Batch Size: 32 sentences
        iii. The optimizer is Adamax (Kingma and Ba, 2014) with a 2e-4 learning rate. We run 20 epochs for each task. The result for STS-B is the Pearson correlation. Matthew's correlation is used for CoLA. F1 is used for QQP and MRPC. Other tasks are measured by Accuracy.
    (b) TyDiQA:
        i. 9 languages
        ii. XLM-RBase: 270M parameters
        iii. The max length is 384. We train the model for 15 epochs with learning rate 3e-5, batch size 8 and gradient accumulation 4.

3.2 Datasets

The GitHub code provided has bash scripts to download these datasets.

1. For the Machine Translation task, IWSLT'14 De→En was downloaded and processed using a script provided by the authors.
2. For Zero-Shot Cross-Lingual Transfer: Wikiann Named-Entity Recognition (NER) task (Pan et al., 2017) and Typologically Di- verse Question Answering-Gold Passage (TyDiQA) (Artetxe et al., 2020) was downloaded and processed using a script provided by the authors.
3. For Image Classification: Oxford-IIIT Pet (Parkhi et al., 2012) was downloaded from PyTorch (Paszke et al., 2019). As 3680 images are available for training and 3669 images are available for testing, 3000 images were randomly chosen as the fixed training set for all models, the remaining 680 images were used as the validation set, and 3669 images were used as the test set.

3.3 Hyperparameters

Machine Translation: Small transformer model: maximum lr=0.0005. The optimizer is Adamwith inverse_sqrt learning rate scheduler and weight decay of 0.0001. The maximum training update is 50K with 8K warm-up steps. At inference time, use beam search with width 5 and a length penalty of 1. 0.1 for attention layers 0.3 for feed-forward (FFN) layers. Trained for 46 epochs.

Zero-Shot Cross-Lingual Transfer: For NER task: bert-base-multilingual-cased was used and was run for 10 epochs with learning rate 2e-5, batch size 8 and gradient accumulation 4 and with 0.1 dropout and 1 alpha. For TDQIA task: xlm-mlm-tlm-xnli15-1024 was used and was run for 15 epochs with learning rate 3e-5, batch size 8, gradient accumulation 4, and with 0.1 dropout and 1 alpha.

Image Classification: Trained 8 models in total for each combination of the binary decision of whether or not to perform intra-distillation, self-distillation, and label smoothing. Epochs: 80, Batch size: 60, Learning rate: increases linearly from 0.00002 to 0.001 over 50 batches, then decreases as inverse square root. Weight decay: 0.001. The optimizer is AdamW (Loshchilov et al., 2017) with beta1 = 0.9, beta2 = 0.999. For intra-distillation, we set K=3, N=4000, q=5, p=10, alpha=5, with parameter names consistent with the original paper. For self-distillation, we weigh the cross entropy from the teacher equally with the log-likelihood,

3.4 Implementation

We will be using existing code from the authors' GitHub repository found here: https://github.com/fe1ixxu/Intra-Distillation/

Our results will be published at this repository: https://github.com/yashmanne/intra-distillation/

3.5 Experimental Setup

Machine Translation task: NVIDIA V100 GPU, 8vCPU and 30 GB of RAM.

Cross-Lingual Transfer: 2vCPU, 13 GB RAM, NVIDIA T4 GPU (16GB)

Image Classification: Nvidia T4 Tensor Core GPU on Google Colab, all models took 4GB of RAM and 2GB of GPU RAM.

Our codebase: https://github.com/yashmanne/intra-distillation/

3.6 Computational Requirements

Estimate: We aim to use Google Cloud to run our experiments. We expect each model to require 16 GB of memory and 12 hours to train. We plan to test this by calculating the memory usage and training time for a single epoch locally and scaling up in order to reserve appropriate memory and compute time on Google Cloud.

Actual

Machine Translation: We used a Google Cloud instance with an NVIDIA V100 GPU, 8 vCPU, and 30 GB of RAM. Training the small transformer architecture for the German-to-English translation on the IWSLT' 14 data set took 3 hours and 2 minutes for the X-Divergence Intra-Distillation model while the baseline small transformer model took 1 hour and 1 minute. It used an average of 23.5 GB (max 24 GB) and 16.5 GB (max 17 GB) of GPU memory for each model, respectively. The model evaluation took approximately 61s for both models with a translation time of 34s.

Zero-Shot Cross-Lingual Transfer: There were two levels of tasks that were run under this experiment. Each was run on the same hardware. Both tasks were run on a google cloud VM instance.

...	NER Task	TQDIA
Type of hardware	2vCPU, 13 GB RAM, NVIDIA T4 GPU	2vCPU, 13 GB RAM, NVIDIA T4 GPU
Avg runtime for each epoch	49.7 minutes/epoch	66.33 minutes/epoch
Total number of trial	1	1
GPU hrs used	8	16
training epochs	10	15

The training times for both levels of tasks were different. It took 8 hr 16 minutes and 59 seconds for the Wikiann Named-Entity Recognition task when trained on a bert-base-multilingual-cased and 16 hr 34 minutes and 25 seconds for the Typologically Diverse Question Answering-Gold Passage task which was trained on xlm-mlm-tlm-xnli15-1024. The time here is inclusive of baseline and intra-distillation training. Both the tasks took 15 GB of GPU memory.

Training ResNet50 (He et al, 2015) for 80 epochs on 3000 training images and 680 validation images from the Oxford-IIIT Pet (Parkhi, et al., 2012) dataset took approximately 3 hours and 20 minutes on one Nvidia T4 Tensor Core GPU on Google Colab for the Intra-Distillation models and approximately 1 hour and 10 minutes for the baseline models. All models used 4 GB of system RAM and 2 GB of GPU RAM. This experiment was added after the proposal, so we don't have an initial estimate to compare to.

On the whole, the actual computational requirements were less than our original estimate and varied greatly from task to task.

4 Results

The authors took an opposite view of pruning and went on with the hypothesis that intra-distillation leads to significant improvement in model performance in tasks of machine translation, natural language understanding and zero-shot cross-lingual transfer tasks and leads to balance the parameter contribution effectively. We verified this by running their experiments on the tasks as defined in their paper. We obtained positive results for all the experiments, i.e we saw increased performance after intra-distillation. In addition, for an image classification task, on average across all models trained, intra-distillation models averaged 0.00564 smaller standard deviation of normalized parameter sensitivity and self-distillation models averaged 0.005635 smaller standard deviation of normalized parameter sensitivity. We also found that intra-distillation loss leads to a 0.098 improvement in test negative log-likelihood across all models trained.

4.1 Result 1: Machine Translation

Training the small transformer architecture with intra-distillation (X-divergence) over 46 epochs for 50,000 updates took a total of 10,891.7 seconds (236.8 s/epoch). The resulting model had a validation cross-entropy loss of 3.727 and a SacreBLEU score of 35.11. This is different from the reported value of 36.10 in Table 2. Training the small transformer architecture without intra-distillation over 46 epochs for 50,000 updates took a total of 3,615.9 seconds (78.6 s/epoch). The resulting model had a validation cross-entropy loss of 3.962 and a SacreBLEU score of 32.90. This is different from the reported value of 33.07 in Table 2. While the individual SacreBLEU values differ, the results support that of the paper in that intra-distillation does in fact lead to better model performance, with the SacreBLEU scoring increasing by 2.21 points (6.72%) and validation loss decreasing by 0.24. However, it should be noted that the increase in SacreBLEU is significantly less than the reported increase of 3.03 points (9.16%). Nevertheless, the increase is almost double the variation between corresponding SacreBLEU scores and could be considered significant.

4.2 Results 2: Zero-Shot Cross-Lingual Transfer

As described, the zero-shot cross-lingual transfer task was divided into two levels of tasks. A low level task i.e Wikiann Named-Entity Recognition and a high level task i.e Typologically Diverse Question Answering-Gold Passage. Lighter models were used to replicate the experiment in the paper. The results exhibited in the paper are from XLM-large model, but in constraint of resources, lesser parameter models were chosen for these tasks. Upon reproducing and training both the tasks, we got results concurring the claim in 2.1 that intra-distillation led to better performance in both the tasks. Wikiann NER data was trained using bert-base-multilingual-cased. It was trained for 10 epochs with 0.1 dropout and 1 alpha (as cited in the paper) The highest baseline f1 score obtained was 82.353 for "english" and with intra-distillation the highest f1 score obtained for "english" was 83.412 (1.28% increase). In contrast to from 84.50 to 85.40 (a 1.06% increase) mentioned in the paper for XLM-R. For TQDIA task the highest baseline f1 score obtained for "english" was 63.54 and with intra-distillation it was 66.853(5.214% increase). In contrast to from 65.51 to 68.20 (a 4.10%) The scores aren't exact as the model results displayed in the paper are from XLM-large but we had to choose a lighter models to run the experiment.

4.3 Additional Experiments: Image Classification

	s=0, teacher	s=0, student	s=0.1, teacher	s=0.1, student
Intra-distillation	0.480	0.537	0.495	0.561
Baseline	0.565	0.657	0.659	0.584

Test negative log likelihood. s refers to amount of label smoothing.

All models trained with intra-distillation outperformed all models trained without intra-distillation, by an average of 0.098. This indicates support for the authors hypothesis that intra-distillation improves generalization for neural networks outside of natural language processing. Considering just the models with label smoothing, the student with intra-distillation underperformed its teacher, by 0.066, while the baseline student outperformed its teacher by 0.075. This indicates that intra-distillation captures the benefit of self-distillation. Further, the intra-distillation teacher outperformed the baseline student by 0.098. This indicates that while intra-distillation captures the benefit of self-distillation, intra-distillation is the more effective choice. On the other hand, considering just the models without label smoothing, students underperformed their teachers by 0.057 for intra-distillation and 0.092 for baseline. Because students underperformed regardless of whether they used intra-distillation or not, we find this is uninformative with respect to the effectiveness of intra-distillation, and rather indicates that label smoothing is important for self distillation on this dataset and model.

	s=0, teacher	s=0, student	s=0.1, teacher	s=0.1, student
Intra-distillation	1.61 / 3.13	1.75 / 2.80	2.40 / 2.92	1.70 / 2.45
Baseline	0.414 / 15.8	8.32 / 2.74	1.10 / 12.0	1.30 / 3.32

Standard deviation of parameter sensitivity, unnormalized / normalized by loss.
Values are scaled by a factor of 1000.

We find that models trained with self-distillation exhibited greater parameter sensitivity in three out of four cases and models trained with intra-distillation exhibited greater parameter sensitivity in three out of four cases. This contradicts the claim of the paper that parameter sensitivity decreases with self-distillation and intra-distillation. However, we additionally present the parameter sensitivity normalized by the loss for each batch $\frac{\Phi_s^T \nabla L_{\Phi_{-s}}}{L_\Phi}$. By this measure, models trained by self-distillation exhibit less parameter sensitivity in all cases, with an average 0.005635 smaller standard deviation of normalized parameter sensitivity, and models trained by intra-distillation exhibit less parameter sensitivity in three out of four cases. In the one counterexample case, comparing the student models without smoothing, the magnitude of difference is just 0.00006. In the other three cases, the intra-distillation models average a magnitude of difference of 0.00754. So on average across the four cases, intra-distillation models average 0.00564 smaller standard deviation of normalized parameter sensitivity.

5 Discussion

The experimental results fully support the central finding that fine-tuning language models with intra-distillation loss leads to improved performance. All models trained with intra-distillation outperformed models trained without intra-distillation.

The results mostly support the secondary finding that self-distillation and intra-distillation lead to less parameter sensitivity. There are however two complications to this claim. First, by the measure of parameter sensitivity suggested in the paper, the claims that self-distillation and intra-distillation lead to less parameter sensitivity is contradicted in three out of four cases. However, we suggest a different measure of parameter sensitivity normalized by the loss for each batch. We consider this a better measure of parameter sensitivity because it is less influenced by training loss. By this measure, self-distillation lead to less parameter sensitivity in all cases and intra-distillation lead to less parameter sensitivity in three out of four cases. The second complication is this counter example case. However, in this case where intra-distillation lead to greater parameter sensitivity, the magnitude of difference is very small compared to the magnitude of difference for the other cases. So we consider the claim that self-distillation and intra-distillation lead to less parameter sensitivity mostly true.

Further, we demonstrate the authors' untested hypothesis that intra-distillation improves generalization of neural networks in domains outside natural language processing by showing that all ResNet50 (He et al, 2015) models trained with intra-distillation on Oxford-IIIT Pet outperformed their baseline counterparts.

Because the codebase was highly abstracted, we were unable to verify that the parameter contribution became more balanced through self-distillation and intra-distillation on the machine translation task. Instead, in the image classification section, we measured the parameter sensitivity balance for all models. In this section we perform self-distillation and intra-distillation, so we can verify on this task that both self-distillation and intra-distillation lead to more balanced parameters.

5.1 What was Easy

Intra-distillation loss is a simple additional term to add to a loss function. This made it easy to transfer to an image classification task. Additionally, the authors provided a code block on the github readme to copy the function.

Machine Translation: Training a small transformer architecture with intra-distillation on the IWSLT'14 DE->En dataset to translate between German and English was very easy. The documentation provided a ready-to-run example that trained without any bugs. Similarly, it was very easy to evaluate this model's SacreBLEU score with a provided demonstration code for inference. However, it was difficult to track what exactly the code was doing due to a great

Zero-Shot Cross-Lingual Transfer: Running of the experiment was straightforward after environment setup. But only for the example given in the readme, any other changes required a lot of time and debugging.

5.2 What was Difficult

Machine Translation: Setting up a working environment was more difficult than anticipated due to the lack of a well-documented requirements.txt file. Specifically, to run the code for the Machine Translation task, multiple Torch versions from 1.13 to 1.7.0 needed to be installed and tested before succeeding with Torch 1.18.1. Adapting the demonstration code for machine translation between German and English to train without intra-distillation was much more difficult than expected. This was because the author's did not provide a list of potential "tasks" possible in the Fairseq library and instead relied on the experience of the user in parsing code and searching for valid arguments. Over the course of training, the debugging information made it easy to pinpoint the root of the issue but difficult to solve.

Loading a model from a given checkpoint was incredibly difficult if it wasn't already part of a preset pipeline coded by the authors. It was similarly difficult to load data for subsequent analyses. The authors did not provide code for determining the sensitivity of parameters in each model. Due to the difficulties of loading the model weights and assigning the proper functionality, it was not possible to determine the sensitivity of parameters. Similarly, no code was provided to verify the claims of self-distillation in Table 1, and multiple attempts over several days to incorporate knowledge distillation proved to be unsuccessful.

Zero-Shot Cross-Lingual Transfer:

Environment setup and running it with older version of libraries (torch 1.3+cudnn10.0) was really difficult. Multiple issues in setup of cuda and the environment were noticed. Additionally, the code could be documented better to make edits and experimentation easier.

5.3 Recommendations for Reproducibility

The first and foremost recommendation is to mention software version requirements explicitly. To run the code for the Machine Translation task without errors, multiple Torch versions from 1.13 to 1.7.0 were installed and tested before succeeding with Torch 1.18.1.

Additionally, we recommend that the authors provide more documentation and structure for how to read their repository. The initial README.md only contained a few lines to run pre-set data preprocessing model training code without any information to actually change parameters or model architectures. The rest of the code was abstracted into multiple layers, which requires extensive knowledge of the Fairseq package and model framework in order to follow or debug. The code base provided some documentation buried within various subfolders but the authors should include an overarching README file with references to documentation for more granular tasks.

For example, a demonstration for loading saved models may prove useful. Additionally, it was hard to follow which functions and modeling tasks accepted which arguments due to the use of abstracted configuration objects for parsing arguments.

Communication with Original Authors

We had no communication with the original authors.

References

[1] CONNEAU, A., KHANDELWAL, K., GOYAL, N., CHAUDHARY, V., WENZEK, G., GUZMÁN, F., GRAVE, E., OTT, M., ZETTLEMOYER, L., AND STOYANOV, V. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), Association for Computational Linguistics, pp. 8440–8451.

[2] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 4171–4186.

[3] FANG, Z., WANG, J., WANG, L., ZHANG, L., YANG, Y., AND LIU, Z. Seed: Self-supervised distillation for visual representation, 2021.

[4] FURLANELLO, T., LIPTON, Z. C., TSCHANNEN, M., ITTI, L., AND ANANDKUMAR, A. Born again neural nethttps://www.crossref.org/labs/citation-formatting-service/works, 2018.

[5] GOTMARE, A., KESKAR, N. S., XIONG, C., AND SOCHER, R. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation, 2018.

[6] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition, 2015.

[7] HINTON, G., VINYALS, O., AND DEAN, J. Distilling the knowledge in a neural network, 2015.

[8] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization, 2014.

[9] KUDO, T., AND RICHARDSON, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Brussels, Belgium, Nov. 2018), Association for Computational Linguistics, pp. 66–71.

[10] LOSHCHILOV, I., AND HUTTER, F. Decoupled weight decay regularization, 2017.

[11] MEISTER, C., SALESKY, E., AND COTTERELL, R. Generalized entropy regularization or: There's nothing special about label smoothing, 2020.

[12] PARKHI, O. M., VEDALDI, A., ZISSERMAN, A., AND JAWAHAR, C. V. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 3498–3505.

[13] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KÖPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library, 2019.

[14] POST, M. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (Brussels, Belgium, Oct. 2018), Association for Computational Linguistics, pp. 186–191.

[15] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. U., AND POLOSUKHIN, I. Attention is all you need. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

[16] XU, H., KOEHN, P., AND MURRAY, K. The importance of being parameters: An intra-distillation method for serious gains, 2022.

[17] ZHANG, L., SONG, J., GAO, A., CHEN, J., BAO, C., AND MA, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation, 2019.

(16) (1) (2) (9) (8) (7) (4) (5) (17) (3) (15) (14) (6) (11) (12) (10) (13)