# Few-Shot Causal Distillation

Stanford CS224N Custom Project

**Tom Starshak**
Department of Computer Science
Stanford University
starshak@stanford.edu

## Abstract

Model distillation is a technique that replicates the performance of a large model (teacher) in a small model (student) by adding an objective to match the output of the teacher when training the student. It has been shown that adding a third objective that encourages the student to match causal dynamics of the teacher by using a distillation interchange intervention training objective (DIITO) can further increase the performance of the student. DIITO swaps the internal representations of both the student and teacher models with a counter-factual representation and attempts to match the change in output between the student and teacher. However, the best way to perform DIITO has been under explored in the small-data regime. In this paper we explore two different parts of DIITO: the mapping of the teacher layers to the student layers, and strategies to generate effective counter-factual representations. We explore scheduling the alignment of the student and teacher models top-down and bottom-up. Additionally we explore using synonyms, antonyms, meronyms, holonyms, and random noise as counterfactual inputs. Finally, we explore doing away with the counter-factual examples entirely, instead directly modifying the intervention internal representations. When testing on portions of the GLUE benchmark suite, the main findings of this paper are: random alignment of all layers works best, the counterfactual example should use the same context as the base network, and the counterfactual example can be eliminated entirely which saves time and memory costs.

## 1 Key Information to include

- Mentor: Isabel Papadimitriou
- External Mentor: Zhengxuan Wu

## 2 Introduction

Large language models (LLMs) have become extremely powerful with applications in a variety of fields. This extreme power has come at a cost though; modern LLMs can have hundreds of billions of parameters and are exorbitantly expensive to train, store, and perform inference on.

Distillation [1] is a method where a smaller, more economical model, can by taught to have similar performance to a larger model. The standard approach to distillation is taking a teacher model (an LLM) and a student model (a smaller model) and training the student with two objectives: 1) the standard task objective (token prediction, sentiment classification, etc) and 2) an imitation objective to match the output of the teacher model. Causal Distillation [2] is an extension to vanilla distillation where the student also tries to align internal representations with the teacher model. While causal distillation has been shown to be effective in improving the performance of smaller neural networks, how well the method works when there isn't much training data hasn't been deeply explored.

In this paper we explore strategies for how best to train a student model using DIITO when there is not much training data. As a baseline we replicate the work from the original paper[2] for the Stanford Sentiment Treebank (SST-2)[3] dataset and then modify three parts of the training loop:

1. Modify the alignment mapping from the student model to the teacher model so that the student is taught top-down or bottom-up rather than all layers at once.
2. Modify the context in which the counterfactual interventions are taken from to match the context of the base model.
3. Instead of randomly swapping input tokens for counterfactual representations, use related tokens.

After exploring these strategies on SST-2 we expanded testing to other GLUE tasks, The Corpus of Linguistic Acceptability (CoLA), Microsoft Research Paraphrase Corpus (MRPC), and Question NLI (QNLI). The four main findings of this paper are:

1. Scheduling the intervention alignment is not helpful.
2. Restricting the counterfactual interventions by using related tokens is counter-productive.
3. Using the same context for the counterfactual example as the main example boosts performance.
4. It is possible to perform an activation intervention without a counterfactual example at all, eliminating the need to store two extra networks or compute a forward pass on them.

## 3 Related Work

Knowledge distillation is a popular technique for transferring knowledge from a large, complex model to a smaller, simpler one. Hinton et al.[1] proposed the technique of distillation where the output probabilities of a larger teacher network are used as soft targets for training a smaller student network. This technique has been widely adopted to train efficient models for various applications.

Another related technique is interchange intervention training (IIT) [4], where the changes in a program are transferred to a base model. The base model is trained to match the output of a program when aligned representations in both models are set to counterfactual representations. This guarantees that the target causal model is a causal abstraction of the source model, when the IIT loss is zero.

The Causal Distillation for Language Models [2] paper introduces the DIITO objective which combines distillation and interventions. Four networks are used during training a base model and source model for both student and teacher. The source model generates counter-factual representations for the base model, and the layers of the student model are aligned with layers of the teacher model. By ensuring that interchanging representations produce similar changes in output for both the student and teacher models, DIITO showed improvements on the GLUE [3] benchmant as well as SQuAD [5] and CoNLL [6].

## 4 Approach

We adapt procedures outlined in the DIITO paper.1 A student model is finetuned on the given task and given several auxiliary objectives. The total loss for the student is:

$$\mathcal{L} = \mathcal{L}_{task} + \mathcal{L}_{CE} + \mathcal{L}_{Cos} + \mathcal{L}_{DIITO}$$

Where the terms in the loss are: standard task loss (binary cross-entropy for the tasks in this paper), the cross-entropy loss between the student and teacher logits, the cosine embedding loss between the representations of the last hidden layers of the student and teacher models, and the DIITO loss which is the cross-entropy loss for the logits of the student and teacher during intervention-interchange.

The teacher models are 12-layer BERT [7] models with 12 attention heads, GELU activations, and embedding size of 768. Fine-tuned teacher models were downloaded from Hugging Face [8] for each of the four tasks. The student models are 6-layer BERT models. As in DistilBERT [9], the student models are initialized with every the weights of every other layer of the teacher models.

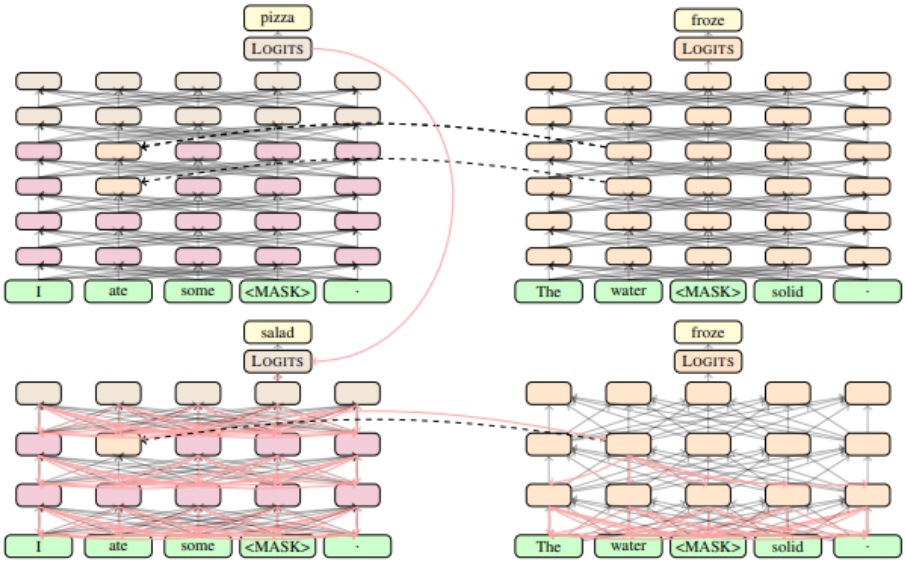


Figure 1: Causal distillation mechanism. Activations are swapped in both the student and teacher networks. The student network is trained such that the change in its output from this swap, should be similar to how the output of the teacher changes.

### 4.1 Layer Alignment

The alignment of an intervention is defined as the mapping of a student layer to one or more teacher layers. In Figure 1 the second layer of the student is aligned with the 3rd and 4th layers of the teacher model. During an intervention, the corresponding activations from the source networks are copied to the base networks. The standard strategy for alignment is to randomly select a student layer every iteration and then interchange 30% of the embeddings for that layer. In this paper we explored two alternate alignment strategies: top-down and bottom-up. In top-down, the last layer of the network is chosen as the intervention layer for the first $\frac{num\ iters}{num\ layers}$ iterations, followed by the second to last layer, etc. Bottom-up is the same, except starting with the first layer.

### 4.2 Intervention Strategies

In the DIITO paper, the source networks drew separate training examples than the base networks. This ensures that the token representations at the intervention layer are different between the base and source networks. In this paper, we used the same input for both the base and source networks, but then randomly swapped 30% of the input tokens according to a specific strategy 2. Word embeddings are extremely context sensitive, and giving the counterfactual representations the same context allows the networks to learn how a specific token is causal rather than a completely different input. We call this "in context DIITO."

The specific strategies for swapping input tokens are based on WordNet relationships between tokens in the input vocabulary. The specific strategies are: synonyms, antonyms, meronyms, holonyms, and random. Meronyms and holonyms are relationships for a part of something and the whole of it, e.g. *finger* is a meronym of *hand*. For random intervention, selected input tokens were swapped with any other random token in the vocabulary. For the other strategies, we created a dictionary mapping vocabulary tokens to all other tokens in the vocabulary that have the specified relationship according to the NLTK WordNet library. At each intervention location a corresponding token was selected randomly if it existed, otherwise a random token was selected.

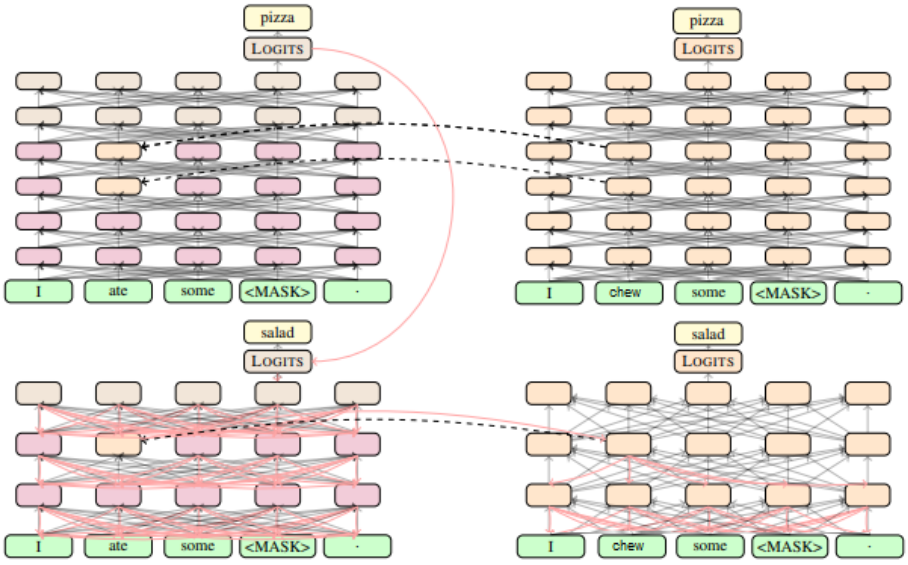


Figure 2: Using the same input for base and source networks ensures that the intervention representations have a similar context in both networks. In this example, the second token was swapped according to the synonym strategy.

| Corpus | \|Train\| | \|Test\| | Task | Metrics | Domain |
|---|---|---|---|---|---|
| | | | Single-sentence tasks | | |
| CoLA | 8.5k | 1k | acceptability | Matthews corr. | misc. |
| SST-2 | 67k | 1.8k | sentiment | acc. | movie reviews |
| | | | Similarity Tasks | | |
| MRPC | 3.7k | 1.7k | paraphrase | acc. | news |
| | | | Inference Tasks | | |
| QNLI | 105k | 5.4k | NLI | acc. | Wikipedia |

Table 1: Selected GLUE tasks.

The last intervention strategy does away with the source networks altogether 3 For the mixup intervention, at every iteration, draw a random proportion and a random vector with the same dimension as the embedding dimension. At a given interchange intervention location updated the embedding as follows:

$$X = \lambda * X + (1 - \lambda) * Y$$

Where lambda $\in [0, 1]$ and $Y \sim \mathcal{N}(0, 1)$.

All intervention strategies were first explored on SST-2, and then the most successful strategy, random intervention in context, was tested on all the datasets along with baselines.

## 5 Experiments

### 5.1 Data

The datasets we used in this study are four datasets that are part of the GLUE [3] benchmark. All of these tasks are binary classification tasks. For testing for few-shot performance, all models were trained on 10% of the full train set and tested on the full test set. We chose these specific tasks due to time and compute constraints, but still to have a variety of dataset sizes and at least one from each task category 1.

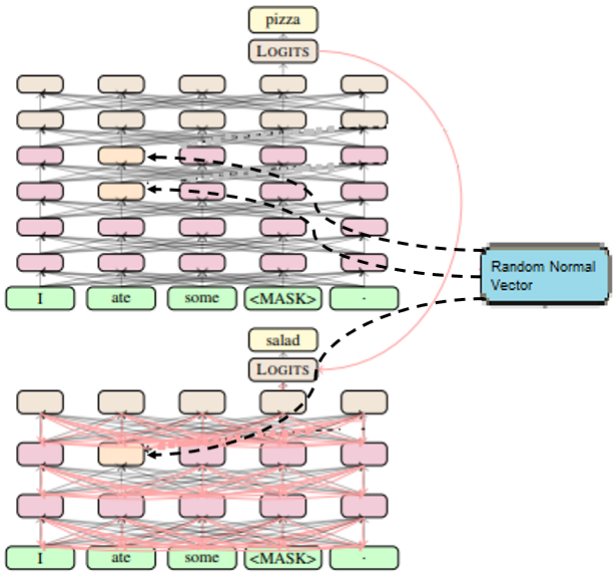


Figure 3: Performing interchange intervention without a source network by generating a random vector for each intervention location at each iteration. The student and teacher networks get the same updates.

| | |
|---|---|
| maximum sequence length | 128 |
| batch size | 16 |
| learning rate | 2e-5 |
| interchange probability | 0.3 |
| weight decay | 0.01 |
| dropout probability | 0.1 |

Table 2: Model Hyper-parameters.

### 5.2 Evaluation method

Three of the evaluation metrics are simply the accuracy of the predictions. The other metric, Matthews correlation coefficient is a valued from -1 to +1 where 1 is perfect accuracy, 0 is random, and -1 is perfectly inaccurate. It is used instead of accuracy due to class imbalance.

### 5.3 Experimental details

All models used the same 12-layer BERT teacher models and 6-layer BERT student models. Finetuned teacher models for each tasks were downloaded from Hugging Face. All models used the hyper-parameters shown in Table 2 and were trained with the BERTAdam optimizer, which is similar to AdamW. The only model that didn't use those hyper-parameters was the mixup intervention interchange, which used weight decay of 0.05 and dropout probability of 0.3. We trained SST-2 models for 50 epochs, MRPC for 150 epochs, and CoLA/QNLI for 30 epochs. The number of epochs were chosen to have the same number of iterations as the results in [2]. All models were trained on a PC with a single Nvidia GTX 1070.

### 5.4 Results

#### 5.4.1 Layer Alignment

As shown in Figure 4, the basic "Full" alignment strategy outperformed both top down and bottom up scheduling.

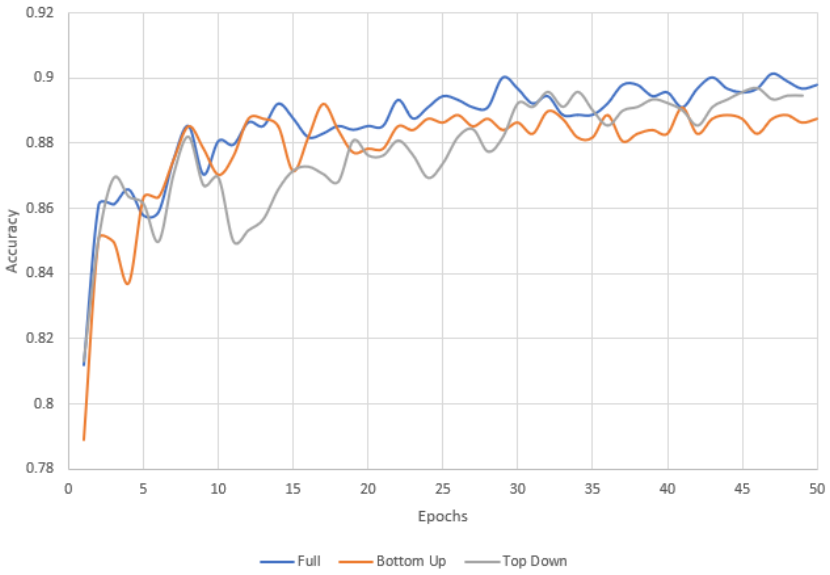


Figure 4: SST-2 accuracy with different alignment strategies.

| Intervention Strategy | Final Accuracy |
|---|---|
| Full Dataset | 90.94 |
| Full Dataset w/ DIITO | 90.94 |
| Baseline | 86.81 |
| Baseline DIITO | 89.79 |
| Synonym | 88.65 |
| Antonym | 88.99 |
| Meronym | 87.38 |
| Holonym | 88.19 |
| Mixup | 89.22 |
| Random (in-context) | 90.60 |

Table 3: Final accuracy on SST-2.

### 5.4.2 Intervention Strategies

All intervention strategies were first tested on SST-2 while baselines with and without DIITO and the random in-context strategy were tested on all datasets.

The baseline results using the full datasets can be found the Table 2 of the original DIITO paper [2]. The relevant values are replicated here.

| Model | Dataset Size | CoLA | MRPC | QNLI | SST-2 |
|---|---|---|---|---|---|
| DIITO | 100% | 43.43 | 88.17 | 85.57 | 90.01 |
| w/o DIIT0 | 10% | 15.04 | 72.00 | 83.29 | 86.81 |
| DIITO | 10% | 28.52 | 71.01 | 86.73 | 89.79 |
| DIITO in context | 10% | 32.22 | 72.53 | 86.18 | 90.59 |

Table 4: Metric comparisons for full data, low data with and without DIITO, and using in-context interventions.

## 6 Analysis

There are several things of note that emerged from this study, most of which involved the different intervention strategies and two of which I think might be significant for future work.

First, the layer alignment strategies did not improve performance. Randomly selecting layers to perform the interchange interventions was superior to both top-down and bottom-up scheduling. Interventions affect the current layer and every layer between it and the logits. The thought of scheduling the alignment layers was that the student would better gradually learn the causality of the teacher rather than all at once.

Second, in context DIITO outperformed not using DIITO on all tasks and in the case of CoLA more than doubled the score. It also outperformed the original DIITO on 3 of the 4 tasks and only slightly performed worse on QNLI. In context DIITO even outperformed the full-dataset DIITO model on QNLI and SST-2. We also note that the tasks where there is a large gap between few-shot and full-dataset results (CoLA and MPRC) have significantly smaller dataset than the other tasks.

Third, using related input tokens for the interchange interventions instead of random tokens did not improve performance. A hypothesis for why this might be the case is that the size of the set of related tokens is necessarily smaller than all the tokens and training is less diverse as a result.

Finally, we showed that interventions are possible without a counterfactual network to create embeddings. This is a significant memory and compute savings during training. As shown in Table 3, the mixup strategy outperformed the baseline, but was slightly worse than the original DIITO. We believe that it was slightly worse than the original DIITO because the alignment between the student and teacher maps one student layer to two teacher layers which this intervention interacts with both teacher layers in the same way. More realistically the second teacher layer would be updated by $X + f(X)$ where X is the intervention embedding and f is some function of the neural network. In the case of using a counterfactual network, f is just that network. A potential avenue for further investigation would be construction of an f that is smaller than a full network, but still propagates the intervention embedding in a way similar to the full model.

## 7 Conclusion

This paper explored several strategies for improving the performance of student models in the context of distillation interchange intervention training objective (DIITO) when there is not much training data. Through experimentation on the Stanford Sentiment Treebank (SST-2) and other GLUE tasks, we found that randomly aligning all layers of the teacher and student models works best. We also found that restricting the counterfactual interventions by using related tokens is counter-productive, and using the same context for the counterfactual example as the main example boosts performance. Finally, we discovered that it is possible to perform an activation intervention without a counterfactual example at all, eliminating the need to store two extra networks or compute a forward pass on them. These findings can help to optimize the DIITO training process and improve the performance of smaller neural networks, making them more cost-effective and practical for a wider range of applications.

Some limitations of this paper are that we were not able to test these methods on all of the GLUE tasks and that only a small number of tests were run using each strategy.

Further work could be testing on the remaining GLUE tasks and getting a larger sample size of results. In addition, we think that a deeper investigation into intervention without a counterfactual network is warranted. Two potential avenues of investigation are: 1) are there a better intervention embeddings than random vectors and 2) is there an effective way to modify the embedding to take into account the teacher model having multiple intervention layers.

## References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[2] Zhengxuan Wu, Atticus Geiger, Josh Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah D. Goodman. Causal distillation for language models. 2021.

[3] GLUE: A multi-task benchmark and analysis platform for natural language understanding.

[4] Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. 2021.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition, 2003.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.