Lottery Ticket Hypothesis: Evaluating Fidelity and Sample Faithfulness

CSI6900 - Yuanzheng Hu-8100173

Supervisor: Marina Sokolova

Abstract

Deep learning models have stepped into a large model-based era, the large number of parameters empowered deep learning models with intelligent performance on both vision and language tasks. Lottery Ticket Hypothesis is a pruning technique on deep learning models that downsize large models without affecting its performance. To evaluate the effectiveness of this pruning technique, the accuracy is often used as a standard way to evaluate the success of the pruning. In our project, we proposed four additional evaluation metrics to re-consider the evaluation on the performance of pruning. Our four evaluation metrics include fidelity, the accuracy over faithful and unfaithful samples, model invariance and model confidence. The fidelity is used to measure the percentage of prediction results remain the same after the pruning with respect to the original model. The accuracy over faithful and unfaithful samples are used to measure the accuracy of the predictions that stay unchanged and changed after the pruning. The model invariance is similar to the fidelity, and it is used as an intermediate metric for the model confidence. The model confidence is the ratio of the accuracy over the faithful samples to the accuracy of all the samples after the pruning. We have used three different models (VisualBERT, LXMERT, UNITER) and two different datasets (VSR, OPEN-I) to evaluate the fidelity and sample faithfulness. We also found positive correlation between the accuracy and fidelity, a decreasing trend in faithful samples and an increasing trend in unfaithful samples tested on VisualBERT model with VSR dataset. An additional learning rate experiment to identify the winning ticket is also included.

Introduction

Lottery Ticket Hypothesis (LTH) coined by Jonathan Frankle (2018) brought an efficient way to downsize the parameter of deep learning models. It can be considered as an iterative pruning technique, that iteratively zero-out parameters in deep learning models based on certain criterion in each loop. Different from the other popular technique in deep learning pruning, the knowledge distillation (KD) technique by Geoffery Hinton (2015) which creates a teacher-student architecture and distill the parameter into student model to downsize, Lottery ticket hypothesis prunes the model in-place. Other techniques like quantization in deep learning models, alters the lower-level floating point numbers in weights, biases and activations to approximate the model. For example, quantization technique replaces 32-bit floats with 8-bit integers, so that the memory cost can be significantly reduced. Comparing with pruning methods, quantization is more like a hacking technique instead of researching on the redundant parameters in the models.

The Lottery Ticket Hypothesis (LTH) was initially tested on conventional neural networks like convolutional neural network (CNN), ResNet and LetNet by Jonathan Frankle (2018). However, the success of lottery ticket hypothesis can be further applied to other architectures such as BERT, which is done by Tianlong Chen and Jonathen Frankle et.al (2020), and vison-language pre-trained (VLP) models like UNITR, LXMERT, ViLT, which are done by Zhe Gan et.al (2022). These experiments prove the that LTH are universally applicable to major deep learning architectures, and not constraint to the field of computer vision (CV).

On the other hand, throughout experiments and research have done on the concept of Lottery Ticket Hypothesis (LTH), especially its pruning technique Iterative Magnitude Pruning (IMP), an iterative version of the pruning technique based on LTH. In the paper by Jonathan Frankle (2019), he stabilized the IMP algorithm with the rewinding technique, which is a technique that reset the network parameters to iteration *k* of the training procedure. This technique increased the pruning stability and data order stability and gave a possible solution when IMP fails to find a winning ticket. There are also other experiments done on LTH. For example, Sanity checks on LTH by Xiaolong Ma et.al (2021) has done a series of ablation studies on finding the winning ticket, they found four patterns that help to find the winning ticket. First, with residual connections on the network, small learning rate and Iterative Magnitude Pruning (IMP) tend to find the winning ticket. Second, without residual connections on the network, small learning rate is not preferred, and Iterative Magnitude Pruning (IMP) has not advantage over One-shot Magnitude-based Pruning (OMP). Third, when a large network and a small dataset are in use, a winning ticket of high sparsity can be found. Lastly, when the pruned model prefers a large learning rate during pruning, different initialization of weights on the network tends to yield the same result.

During the research on lottery ticket hypothesis, we found that most of the papers evaluate the pruning based on the accuracy, which compares the source model (unpruned model) and target model (pruned model) based on its performance on test set in terms of the accuracy. We assume this evaluation can be further optimized using the concept of fidelity and sample faithfulness. The idea of fidelity comes from a measurement of trustworthiness of the deep learning model, where one can reconstruct the model inputs based on its output. We adapt this idea and treat the unpruned model as the input from the LTH algorithm and pruned model as the output.

Thus, our idea in this project is to use the concept of fidelity, treat it as another way of evaluating the pruning model's performance. We will measure the fidelity of source model (unpruned) model and target model (pruned) model and compare how the distribution of these two models' predictions are close. We will also conduct an ablation study by tuning the learning rate to capture the fidelity changes between pruned and unpruned model.

In addition to fidelity, we adapted the concept of forgetting statistic by Toneva, Mariya et al. (2018) and proposed a similar idea called sample faithfulness; the sample faithfulness is an idea that forgettable samples are hard to learn while the unforgettable samples are easy to learn. Pruned models should easily recognize the unforgettable samples, whereas the forgettable samples can be difficult to recognize. We refer those easy to learn samples as faithful sample and those hard to learn samples as unfaithful sample. Based on the sample faithfulness, we also proposed the idea of invariability and confidence score of the model, which are measurements of how the quality of models' prediction varies between original and pruned model.

Bidirectional Encoder Representations from Transformers (BERT) architecture done by Devlin (2018) not only brought the prosperity to field of Nature Language Processing (NLP) but also helps the evolving of multi-modal field in deep learning. Especially for vision-language (VL) research, majority of the state-of-the-arts models in VL fields adapted the architecture of BERT, such as ViLBERT, LXMERT, UNITER, etc.,. According to a survey done by Yifan Du (2022) in vision-language models, these models either use a two-stream architecture that encodes images and text separately by using standard transformer process or use a single-stream architecture that combine images and text embeddings together with a transformer module. Prior to our research, Zhe Gan et.al (2021) has explored the feasibility on vision-language models UNITER and LXMERT and found a subnetworks that matches 99% of the full accuracy with 50%-70% sparsity. Our project will experiment the model using medical dataset and use VisualBERT as an additional model for our reference.

In this project, we pick three vision-language pre-trained (VLP) models and two datasets. For models, we will be using UNITER by Yen-Chun Chen et.al (2019), LXMERT by Hao Tan et.al (2019), VisualBERT by Liunian Harold Li et.al (2019). For UNITER model, we will be using the pre-trained base model publicly available on their GitHub repository, the pre-trained model has been trained on four tasks, (i) Masked Language Modeling (MLM) conditioned on image; (ii) Masked Region Modeling (MRM) conditioned on text; (iii) Image-Text Matching (ITM); and (iv) Word-Region Alignment (WRA) over four image-text dataset (COCO,Visual Genome, Conceptual Captions, and SBU Captions). For LXMERT model, we will also use the pre-trained model publicly available on their GitHub repository, it is pre-trained on four tasks, masked language modeling, masked object prediction (feature regression and label classification), cross-modality matching, and image question answering over a large aggregated vision-and-language dataset comes from MS COCO (Lin et al., 2014) and Visual Genome (Krishna et al., 2017), VQA v2.0 (Antol et al.,2015), GQA balanced version (Hudson and Manning, 2019), and VG-QA (Zhu et al., 2016). Lastly, pre-trained VisualBERT model publicly available on their GitHub repository is used, two pre-trained tasks (MLM, ITM) over image caption dataset is done (Not mentioned in their paper).

Related work

Lottery Ticket Hypothesis

The lottery ticket hypothesis was first coined by Jonathan Frankle et.al (2018), where it states that for a randomly initialized, dense network, there exists a subnetwork, when trained in isolation it can achieve a comparable test accuracy of the original network. The technique that is used in LTH is called One-shot Magnitude Pruning (OMP) or Iterative Magnitude Pruning (IMP), a randomly initialized network is trained for a certain iteration and chooses a percentage of parameters to be pruned based on their magnitude. The rest of the parameters are reset to their value at the beginning of the training. The resulting network is called the “winning ticket” based on the OMP technique and will be further trained to be tested on the test set. IMP is an iterative version of the previous procedure. The method was tested on both a fully connected network (LeNet)and a convolutional neural network (VGG, ResNet) over two datasets (CIFAR10, MNIST), and the pruned network achieved a comparable result to the unpruned network, the parameters is downsized to ~30% of the original network. Jonathan Frankle et.al (2019) further stabilized their technique by using the concept of rewinding, instead of reinitializing the parameters of the network to the beginning of the training, IMP with rewinding can take the parameters back to any steps after the training.

LTH was extended by other researchers and used in many other architectures. For example, Tianlong Chen et.al (2020) tries to apply the Lottery Ticket Hypothesis in the pre-trained BERT model, they found a trainable and transferable subnetwork exists in the pre-trained BERT model. Also, the subnetworks trained on Masked Language Modeling (MLM) tasks are transferable across the downstream tasks. However, subnetworks found on downstream tasks are not transferable to other downstream tasks, and iterative magnitude pruning (IMP) performs better in some tasks (STS-B, WNLI, RTE, SST-2) and worse in other tasks (QQP, QNLI, MRPC, MLM), whereas rewinding technique does not help to improve the performance for any downstream tasks. The paper by Zhe Gan et.al (2022) aims to prune Vision Language (VL) pre-training models since these models are large in terms of parameters and the lottery ticket hypothesis (LTH) can achieve on par or even better performance with a dense network. The author wants to find both task agnostic tickets and task-specific tickets by performing LTH on pre-training tasks and downstream tasks respectively. They used the UNITER model and found that rewinding does not have a notable effect, with only minor improvement found in very high sparsity (90%) in task-specific pruning. Also, task agonistic pruning is found to be transferable universally across the downstream tasks, only when sparsity is high (80%,90 %) the found subnetwork is worse than the task-specific subnetworks.

The above papers give preliminary research on the feasibility of the lottery ticket hypothesis on complex neural network architectures and motivate us to apply LTH on vision-language pre-trained (VLP) models. Especially for the paper by Zhe Gan et.al (2022), we will primarily compare our reimplemented results with theirs.

There are also some other adaptions to the LTH to prune the network more efficiently, In the paper by Zheng, Rui, et al (2022), instead of using the rewinding technique to prune the network, the author proposed a technique called knowledge distillation (KD) ticket. Contrary to the rewinding, which inherits the weights from the early training phase. KD ticket derives the knowledge from the late training phase. KD ticket can also be used along with rewinding to provide state-of-the-art result for large-scale lottery ticket Inspired by the forgetting statistics by Toneva, Mariya et al (2019), Zhang, Zhenyu (Allen) et al (2021) aims to find an efficient way of finding the lottery ticket. The author coined a concept called Pruning-Aware Critical set (PrAC set), the dataset is selected based on the following criterion and trained along with the lottery ticket pruning to efficiently find the lottery ticket.

For the above variations of the LTH, we found it a bit time-consuming to apply their technique, and since we are using fidelity as our measurement of the models. The feasibility of applying fidelity along with the forgetting statistic has no prior experiments to support, thus we will not experiment on these LTH variation techniques.

At last, the sanity check from Ma, Xiaolong, et al. (2021) gives a throughout ablation study on Lottery Ticket Hypothesis. They pointed out that small learning rate and insufficient training epochs are the main reasons cause controversies in the lottery ticket studies, and found when residual connections exist in the network, a small learning rate is likely to find the winning ticket, and when the small learning rate is not favorable, initialization of weights is likely to make no difference in finding winning ticket.

Fidelity

The idea of fidelity can be inferred from the field of adversarial attacks on deep learning models. According to the paper by Ziqi Yang (2019), models trained on a training set cannot represent the true distribution of the data. Thus, such models can be attacked by adversarial examples, where the adversarial examples can be reconstrued by using the output from the deep learning model, and this can be considered as a security issue or trustworthiness issue of the models. Experiments have been done on such attacks, for example in the paper by Xuejun Zhao et.al (2021), they have reconstrued the image using the output Convolutional Neural Network (CNN) and saliency map. Another paper by Smitha Milli et.al (2018) also demonstrated that using gradient-based explanation from the model to reconstruct the model is viable. However, the formal definition of fidelity is not mentioned in the paper above, but the procedure of reconstructing the inputs from outputs mimic the idea of fidelity, which is that the output distribution is trying to match with the input distribution as much as possible.

The formal definition of fidelity is mentioned in the paper by Ulrich Aivodji et.al (2019), where they developed a systematic algorithm called "LaundryML" to rationalize the black-box explanation from machine learning models. To bring fairness to the model being used, the author enumerates models to find the surrogate model with high fidelity and low unfairness to the input model and then rationalize it, which can be done either by rationalizing the model itself or the outcome.

$$\mathsf{fidelity}(c) = \frac{1}{|X|} \sum_{x \in X} \mathbb{I}(c(x) = b(x)).$$

Formula 1. Formula for fidelity used by Ulrich Aivodji et.al(2019)

The paper authored by Ziqi Yang (2019) uses the following definition of fidelity. In their paper, they consider fidelity as the trustworthiness or confidence of the output from the model to the true population of the underlying task. To estimate the fidelity, the author propose a method called "jury" and uses models from a predefined model pool to estimate the true population of input sample space.

$$F(x, y; H) = 1 - |P_{model}(y|x) - P_{pop}(y|x)|$$

Formula 2. Formula of fidelity definition – Ziqi Yang (2019)

Forgetting statistic

Forgetting statistic is an idea proposed by Toneva, Mariya et al. (2018). Their paper states that during a course of training, when a training sample is misclassified and consequently leads to a decrease in accuracy, this phenomenon is called a "forgetting event" and such a sample is called "forgettable sample". Removing forgettable samples from the training dataset, the model can still achieve a state-of-the-art performance. Also, a dataset's (un)forgettable samples generalize across different neural architecture. They also extract those forgettable samples from the dataset, and from a visual inspection, the forgettable samples are either blurred or clipped while as the unforgettable samples' objects are clear and obvious.

Vision-language model

UNITER (UNiversal Image-TExt Representation Learning) is a single stream vision-language model trained by Yen-Chun Chen et.al (2019) that learned from concatenated text embedding from BERT and image embeddings from Faster R-CNN. The model was trained over four pre-training tasks Masked Language Modeling (MLM), Masked Region Modeling (MRM, with three variants), Image-Text Matching (ITM), and Word-Region Alignment (WRA) over four image-text datasets (COCO, Visual Genome, Conceptual Captions, and SBU Captions) and achieved state of the art across six vision language tasks (over nine datasets), including Visual Question Answering, Image-Text Retrieval, Referring Expression Comprehension, Visual Commonsense Reasoning, Visual Entailment, and NLVR.

LXMERT (Learning Cross-Modality Encoder Representations from Transformers) is a dual stream vision-language model, which learns textual embeddings from BERT and image embeddings from Faster R-CNN separately and then learns the contextualize from both embeddings in a cross-modality encoder. Different from the UNITER model, LXMERT uses object-level image embeddings, which extract objects from images and output multiple object-level embeddings as image embeddings. LXMERT pre-trained on five pre-training tasks, masked language modeling (MLM), masked object prediction (feature regression and label classification), cross-modality matching, and image question answering over five datasets (COCO, Visual Genome, VQA v2.0, GQA balanced version, VG-QA)and achieved the state of the art result on two visual question answering datasets (i.e., VQA and GQA).

VisualBERT, similar to UNITER is a single stream vision-language model that is designed as a baseline model for VL tasks. It uses the visual feature from Faster R-CNN and textual feature from BERT and pre-trained on two tasks (MLM, ITM) over the COCO dataset and then applied on four VL tasks, VQA, VCR, NLVR, and region-to-phrase grounding.

Identifying the winning ticket

In the paper by Xiaolong ma et.al (2021), they have done a comprehensive sanity check on the Lottery Ticket Hypothesis, where they have a more rigorous definition on the Lottery Ticket Hypothesis. They found that the original Lottery Ticket Hypothesis experiments done by Jonathan Frankle et.al (2018) was using the models that have not been fully trained, which means that the accuracy still fluctuated and not yet stabilized. Thus, they have defined a concept called “jackpot”, which refer to the winning tickets that their original network has been fully trained, and they call the winning tickets which the original network was not fully trained “secondary prize”. Aside from a more rigorous definition, they also found that Jackpot rarely exists, but secondary prize exists most of the time. For large datasets, One-time Magnitude Pruning (OMP) is preferred over the Iterative Magnitude Pruning (IMP). The rewinding of weights in IMP setting only helps under the condition that the weights before and after the pruning have a positive correlation. Lastly, they discovered an empirical tip for finding the winning ticket that smaller learning rate tends to find secondary prize more often than the large learning rate. To verify the last point they found, we will also conduct experiments on different learning rate, to see if we are able to find the winning ticket more often than the large learning rate.

Feature Engineering Steps

Visual Features

Across our research on vision and language models, as well as the datasets, there are generally two methods that have been applied to those papers. The first one is Faster R-CNN method, the second one is Bottom-Up and Top-Down Attention (BUTD).

For the models we use, In pretrained LXMERT model, the authors used 101-layer Faster R-CNN method proposed by Shaoqing Ren et.al (2015) to extract object level features and the coordinates of those features on the image, the Faster R-CNN model is trained on Visual Genome dataset (Jiang et.al 2018) . For pretrained UNITER model, it also uses Faster R-CNN model which is trained on MS COCO (Lin et.al 2014) and Visual Genome to extract features for each region along with the coordinates of objects’ bounding boxes. As for VisualBERT, it uses a ResNetXt-based Faster R-CNN pretrained on Visual Genome, different from LXMERT and UNITER, it does not consider the position of the objects.

For the dataset we use, the authors of these dataset conducted baseline experiments for us to reference to, the visual feature engineering technique they used, consists of Faster R-CNN, BUTD, VGGNet(Karen Simonyan et.al 2014), ResNet (Kaiming He et.al 2015).

For PathVQA dataset, the author extracts features using Faster R-CNN, VGGNet and ResNet, with the Faster R-CNN pretrained on Visual Genome and later two pretrained on ImageNet (Jia Deng et.al 2009). For OPEN-I dataset experiments, the author applied a special version of BUTD visual encoder to get the visual features. In SLAKE dataset, the author uses VGG16 to extract the visual features.

Faster R-CNN

Faster R-CNN (Shaoqing Ren et.al 2015) is an optimized version of the Fast R-CNN (Ross Girshick, 2015), and composed of two modules. The first module is a CNN network called Region Proposal Network (RPN) that used to propose regions, and the second module is a Fast R-CNN detector that used to predict the class for proposed regions. In RPN, to generate bounding boxes for proposed objects, a sliding window is skim over the feature map of images, for each sliding window location, the RPN predicts a maximum number of possible proposals called anchors. These anchors will be further filtered to generated translation-invariant anchors based on multi-scale design technique. After generating proposed regions from the RPN, these regions are then feed into the Fast R-CNN network for further classifications.

Button-Up and Top-Down Attention (BUTD)

BUTD (Peter Anderson et.al, 2018) is a combination of bottom-up and top-down visual attention mechanism, where the bottom-up mechanism refers to propose salient images regions, and top-down mechanism refers to determine the feature weights based on the task given. For bottom-up model, the author uses Faster R-CNN in conjunction with ResNet-101 CNN, and the use non-maximum suppression (NMS) for each object class with an IoU (Intersection over Union) threshold (Fig 1). For the top-down model, the author uses two LSTM layers to weight each feature, since the top-down model is task specific, which means for each kind of task the weights on feature will be different. In the paper, the author tested the top-down model on VQA task and uses the input question to generate weights on input features.

Our visual feature extraction

For our visual extraction method, we use the bottom-up model from BUTD. Because the top-down model can be considered as a language model, and we only want the visual features at this step, so we use the bottom-up model only. We make change to the IoU threshold mentioned above for us to generate enough objects to use.

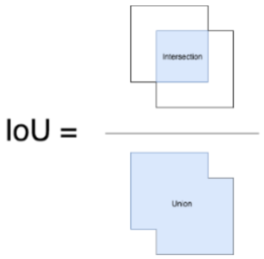
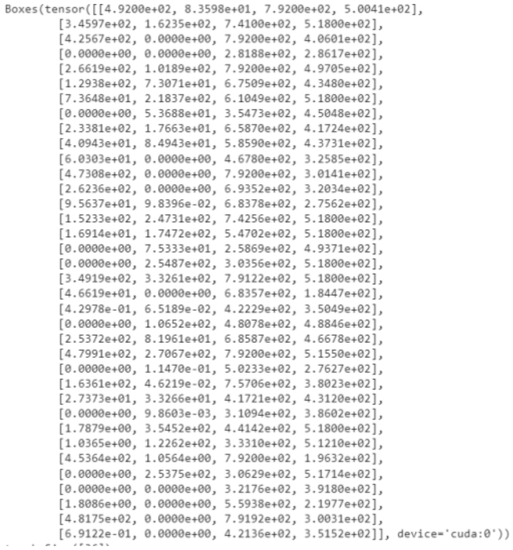


Figure1. Intersection of Union

The last step of bottom-up model from BUTD is use non-maximum suppression (NMS) algorithm, the input of the algorithm is a list of bounding boxes of for a single object, the then for each pair of bounding boxes, we calculate the Intersection of Overlap (IoU) threshold, if the calculated value exceeds the threshold then two bounding boxes will be merged, the algorithm runs for all bounding boxes to find a stable bounding box for a single object. In the BUTD paper, they've used a threshold of 0.5. However, in our project, such a threshold will not be able to generate enough number of objects, thus we lower the threshold to 0.04. our final visual feature is having a dimension of 36. That means 36 objects are identified in each image and each object is represented by a vector. Together 36 vectors form the visual feature for a single image. As shown in the Figure 2.



```
Boxes(tensor([[4.9200e+02, 8.3598e+01, 7.9200e+02, 5.0041e+02],
        [3.4597e+02, 1.6235e+02, 7.4100e+02, 5.1800e+02],
        [4.2567e+02, 0.0000e+00, 7.9200e+02, 4.0601e+02],
        [0.0000e+00, 0.0000e+00, 2.8188e+02, 2.8617e+02],
        [2.6619e+02, 1.0189e+02, 7.9200e+02, 4.9705e+02],
        [1.2938e+02, 7.3071e+01, 6.7509e+02, 4.3480e+02],
        [7.3648e+01, 2.1837e+02, 6.1049e+02, 5.1800e+02],
        [0.0000e+00, 5.3688e+01, 3.5473e+02, 4.5048e+02],
        [2.3381e+02, 1.7663e+01, 6.5870e+02, 4.1724e+02],
        [4.0943e+01, 8.4943e+01, 5.8590e+02, 4.3731e+02],
        [6.0303e+01, 0.0000e+00, 4.6780e+02, 3.2585e+02],
        [4.7308e+02, 0.0000e+00, 7.9200e+02, 3.0141e+02],
        [2.6236e+02, 0.0000e+00, 6.9352e+02, 3.2034e+02],
        [9.5637e+01, 9.8396e-02, 6.8378e+02, 2.7562e+02],
        [1.5233e+02, 2.4731e+02, 7.4256e+02, 5.1800e+02],
        [1.6914e+01, 1.7472e+02, 5.4702e+02, 5.1800e+02],
        [0.0000e+00, 7.5333e+01, 2.5869e+02, 4.9371e+02],
        [0.0000e+00, 2.5487e+02, 3.0356e+02, 5.1800e+02],
        [3.4919e+02, 3.3261e+02, 7.9122e+02, 5.1800e+02],
        [4.6619e+01, 0.0000e+00, 6.8357e+02, 1.8447e+02],
        [4.2978e-01, 6.5189e-02, 4.2229e+02, 3.5049e+02],
        [0.0000e+00, 1.0652e+02, 4.8078e+02, 4.8846e+02],
        [2.5372e+02, 8.1961e+01, 6.8587e+02, 4.6678e+02],
        [4.7991e+02, 2.7067e+02, 7.9200e+02, 5.1550e+02],
        [0.0000e+00, 1.1470e-01, 5.0233e+02, 2.7627e+02],
        [1.6361e+02, 4.6219e-02, 7.5706e+02, 3.8023e+02],
        [2.7373e+01, 3.3266e+01, 4.1721e+02, 4.3120e+02],
        [0.0000e+00, 9.8603e-03, 3.1094e+02, 3.8602e+02],
        [1.7879e+00, 3.5452e+02, 4.4142e+02, 5.1800e+02],
        [1.0365e+00, 1.2262e+02, 3.3310e+02, 5.1210e+02],
        [4.5364e+02, 1.0564e+00, 7.9200e+02, 1.9632e+02],
        [0.0000e+00, 2.5375e+02, 3.0629e+02, 5.1714e+02],
        [0.0000e+00, 0.0000e+00, 3.2176e+02, 3.9180e+02],
        [1.8086e+00, 0.0000e+00, 5.5938e+02, 2.1977e+02],
        [4.8175e+02, 0.0000e+00, 7.9192e+02, 3.0031e+02],
        [6.9122e-01, 0.0000e+00, 4.2136e+02, 3.5152e+02]], device='cuda:0'))
```

Figure2. Visual feature of shape 4x36

In this project, the aforementioned technique is only applied to the visual spatial reasoning (VSR) dataset. For OPEN-I dataset, we have used the predefined visual features mentioned in the original paper by Li, Yikuan et.al (2020). However, the way they achieved the visual feature is not applicable, they did not mention how they obtain the visual feature in the paper.

Datasets

For datasets, we will use the Open-I dataset, collected by Indiana University (2016), the dataset is a publicly available chest X-ray dataset and labeled by medical professionals. The dataset consists of 15 different types of chest/lung diseases and each of the sample is associated with one text sentence and an image. Each sample can have an answer in one of the 15 different types. Thus, this can be considered as a multi-class classification problem.

The other dataset is Visual Question Answering (VQA), it is a probing benchmark for spatial understanding of vision and language model. The dataset consists of 66 types of spatial relations in English (such as: under, in front of, facing). Similar to the OPEN-I dataset, each of the sample is associated with on text sentence and an image. Different from the OPEN-I dataset, each sample has only true or false response, which can be considered as a binary classification problem.

The main conclusion of this project comes from Visual Question Answering (VQA) dataset, for which the task is to let the model selects the correct answer from an answer pool based on the text and image given. We will train the aforementioned models and prune them using the Iterative Magnitude Pruning (IMP) technique then further evaluate them using fidelity.

Approach(es)

In this section, we will talk about the idea of fidelity and sample faithfulness, and how they are used in the context of model pruning.

Fidelity

As we mentioned before, fidelity has been used in the field of trustworthiness and fairness in AI. For trustworthiness, fidelity is used for identifying adversarial attacks for the deep learning model. It is used to describe the gap between what a model learns, and the ground truth learnt by the humans, if the value of fidelity is lower than a threshold, then it means there's a huge gap between the model's prediction and the ground truth, therefore it can be considered as an adversarial attack. Likewise, if the value of fidelity is higher than a threshold, it means the deep learning model's prediction shares a lot in common with the ground truth distribution, then the input is considered to be within the training distribution and therefore it's not an adversarial attack. The author Yang ziqi (2019) used the formula in Figure 2 to describe the fidelity, if the model's prediction distribution is close to what the population prediction distribution, then fidelity is high, and if there's a large discrepancy between the ground truth and model's prediction distribution then the fidelity is low.

In the other work we cited, Aivodji U et.al (2019) uses fidelity as a metric to evaluate the similarity between different models and pick the one with the highest fidelity and the lowest unfairness of the original model. In their work, fidelity is defined in Figure 1, where the fidelity is considered as the accuracy discrepancy of two models' prediction on a dataset.

In the two above cases, the authors use accuracy or the commonality in models' prediction to calculate the value of fidelity. Similarly, we adopted the same idea, where the fidelity in our project is defined as follows (Fig 3).

$$F = \frac{1}{|X|} \sum_{x \in X} (f_{\theta_0}(x) = f_{m \odot \theta_0}(x)))$$

Formula 3. Fidelity of pruned model

Here, Θ_0 is defined as the original network and $m \odot \Theta_0$ represents the pruned network. The value of the fidelity is calculated as the normalization of common predictions they share on a dataset X. We hypothesize that the model's fidelity decreases after pruning. A). Assuming a winning ticket is found after pruning, then accuracy increases, mistaken predictions are fixed, as the number of true positives and false negatives must be corrected, thus the fidelity drops. B). If no winning ticket is found, then the fidelity should also drop, as the number of true positives and false negatives must drop. If the accuracy stays the same, by definition of the winning ticket, it has a comparable accuracy with the original model, so it belongs to case A. Thus, we want to see if there's statistical correlation between the accuracy and the fidelity, also how fidelity behaves across different pruning percentages.

Sample Faithfulness

As we mentioned before, sample faithfulness is an idea inspired from the forgetting statistic. In conventional neural network training, catastrophic forgetting is a phenomenon that the neural network forgets the previously learnt training samples upon learning the new samples. Based on this, Mariya Toneva et.al (2018) proposed forgettable and unforgettable samples, also forgetting and learning events. The algorithm they provide indicates that if there's an increase between two training steps, then it is considered as a learning event. Conversely, if there's a decrease between two consecutive training updates, then it is considered as a forgetting event. For the definition of unforgettable samples, if a training sample experience a learning event at some point of the whole training process and experience no forgetting event after that, then it is classified as unforgettable samples. At last, the samples that have been forgotten at least once are called forgettable samples.

We adapted the idea of forgettable and unforgettable samples and believe that if the prediction of a test sample varies between original and pruned model then it is considered as unfaithful sample while as if a test sample's prediction stays unchanged between original and pruned model then it is considered as faithful sample. One critical difference between the forgettable/unforgettable samples and faithful/unfaithful samples comes from the difference between training data (the former) and test data (the later). Forgettable/unforgettable samples are stem from the training test, they reflect if the sample is easy or hard to learn with respect to the model. On the other hand, the faithful/unfaithful samples come from the test samples, they show if the test sample is considered as easy/hard samples to recognize with respect to the original and pruned models.

The idea of the unfaithful sample is that, assuming there's a winning ticket, pruning prunes away the unnecessary parameters which means that these samples are hard to learn, hard to learn samples are similar to the forgettable samples in Toneva's et.al (2018) paper, where the forgettable samples refer to the samples that decrease the mode's accuracy during the training phase. Assuming there's no winning ticket, pruning prunes away the intelligent parameters and therefore the model is not confident in the prediction of these samples as they're hard to learn.

On the other hand, faithful samples are, assuming there's a winning ticket, the prediction of those samples remain unchanged, they are easy to learn. The easy to learn samples are similar to the unforgettable samples where they refer to the samples experience no forgetting event during the whole course of training. If there's no winning ticket, the test samples are easy to learn (unforgettable samples) or the model is confident in its prediction.

Table 1.1 below shows the difference between faithful and unfaithful samples, and visualized our analysis above in two different settings, the winning ticket setting and the non-winning ticket.

	Winning ticket	Non-winning ticket
Faithful samples	Test samples: easy to learn	Test samples: easy to learn
Unfaithful samples	Test samples: hard to learn	Test samples: hard to learn

Table1.1. Analysis on test samples

We define the faithful samples as follows, where the Θ_0 is defined as the original network and $m \odot \Theta_0$ represents the pruned network,

$$X_{fatith} \doteq f_{\theta_0}(x) = f_{m\odot\theta_0}(x)$$

Formula 4. Faithful samples

On the other hand, the unfaithful samples are defined below,

$$X_{unfatith} \doteq f_{\theta_0}(x) \neq f_{m\odot\theta_0}(x)$$

Formula 5. Unfaithful samples

Model Invariance and Confidence

To explain the behaviour of faithful and unfaithful samples, which the prediction stays unchanged and changed after pruning. We believe that the model's confidence might also be a factor that influences this behaviour. The faithfulness contributes the prediction invariance to the testing samples, because the idea of faithfulness considers the testing samples themselves are easy/hard to learn. However, the model's performance on these data is also a factor that could affect the prediction invariance.

As we can see from the above assumptions, there're two dimensions are taken into the consideration during the course of pruning. One is test samples' faithfulness, the other one is the model's confidence. For the model's confidence, we believe there are two magnitudes that associate with the confidence, one is the invariance between original and pruned model's prediction, the other one is the accuracy on its invariant predictions, because we think that the confidence score should also reflect accuracy over the faithful and unfaithful samples. For example, the author Sampurana Mandal et.al (2021) states that the concept of confidence score should show the probability of an image being detected correctly by the model as a percentage. Thus, we first define the invariability score of the model as follows, where it is simply the number of faithful samples over all the test samples.

$$Invariability(f_{\theta_0}, f_{m\odot\theta_0}) \doteq \frac{X_{faith}}{X_{faith} + X_{unfaith}}$$

Formula6. Model invariability score

Then, to include the concept of correctness, we use the following definition of the confidence score, where the confidence score is considered as the ratio between the accuracy of the faithful samples and the accuracy of all test samples.

$$confidence(f_{\theta_0}, f_{m\odot\theta_0}) = \frac{Acc(X_{faith})}{Acc(X_{faith} + X_{unfaith})}$$

Formula 7. Model Confidence score

As we can see from the analysis above. Model's confidence might also be one of the factors that will affect predictions' result. Here, we distinguish the confidence and the accuracy of the model's prediction. Most of the definition of confidence of the model is associate with the accuracy of the prediction. However, in a pruning experiment since we have two different models, an original model, and a pruned model. Thus, we need an additional evaluation metric to measure to what extent the model's prediction stay unchanged and accurate between the original and pruned state. Thus, the confident of the model is not only used to measure the accuracy of the prediction, but also to measure the stability and the accuracy of the model between original and pruned model.

Considering the following table, the model is confident in faithful samples in both winning ticket and non-winning ticket settings, because those samples' prediction stay unchanged. The mode is not confident on those unfaithful samples in both settings because the model's decision swinging before and after the pruning.

	Winning ticket	Non-winning ticket
Faithful samples	Model: confident	Model: confident
Unfaithful samples	Model: not confident	Model: not confident

Table1.2. Analysis on Model confidence.

In this project, due to the scope of the project, we will not cover the comparison of the confidence score of the model. We will report the comparison between the number of faithful and unfaithful samples in this project's experiments as we mentioned in Table 1.1.

Datasets

Open-I

Open-I dataset is another publicly available chest X-ray dataset provided by Dina Demner-Fushman et.al (2016) from Indiana University. From their paper, the dataset consists of 3996 radiology reports with 8121 images, and labels are labeled by medical professionals using Medical Subject Heading (MeSH) indexing. We use the pre-processing version from Yikuan Li et.al (2020) paper. Where each image is associate one text description statement. We summarize the information we have in Figure 10 (Appendix). In total we have 3391 training samples and 293 testing samples.

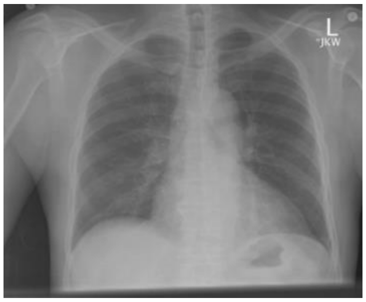
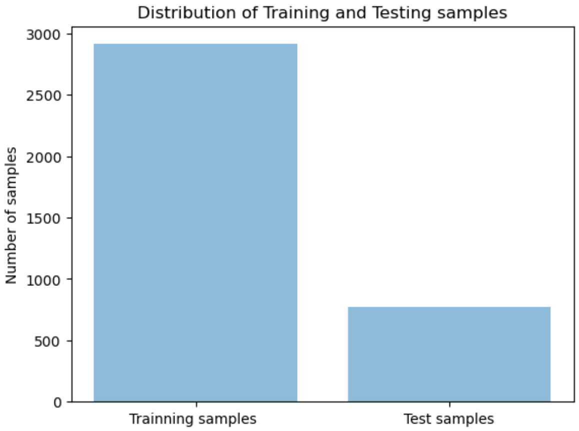


Figure3. Q: Type of symptom?

A: Normal



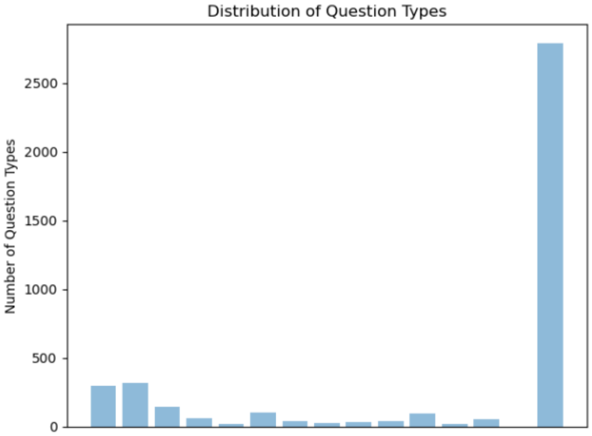


Figure 4. Distribution of training and testing samples and question types

Figure 4. The labels on question type figure are 'Atelectasis', 'Cardiomegaly', 'Effusion', 'Infiltration', 'Mass', 'Nodule', 'Pneumonia', 'Pneumothorax', 'Consolidation', 'Edema', 'Emphysema', 'Fibrosis', 'Pleural_Thickening', 'Hernia', 'Normal'

Feature Extraction

The feature extraction for OPEN-I dataset is a special edition of BUTD visual encoder for UNITER, LXMERT and VisualBERT, it consistently extract 36 features for each image. The detail implementation of the feature extraction is not mentioned in the original paper.

Baseline Result

Learning rate: 0.01

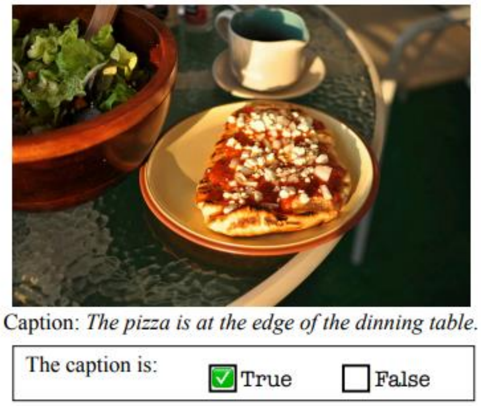
Batch size: 16

Model	Accuracy	Epoch
VisualBERT	98.7	12
LXMERT	98.4	12
UNITER	98.6	12

Table 2. OPEN-I original result

Visual Spatial Reasoning (VSR)

The visual spatial VQA dataset, Visual Spatial Reasoning (VSR) is derived and created by Fangyu Liu et.al (2022), it is extracted from MS COCO 2017. The dataset contains 10,119 data images and each of them is associate with a spatial question. E.g., if the pizza is on the edge of the table.



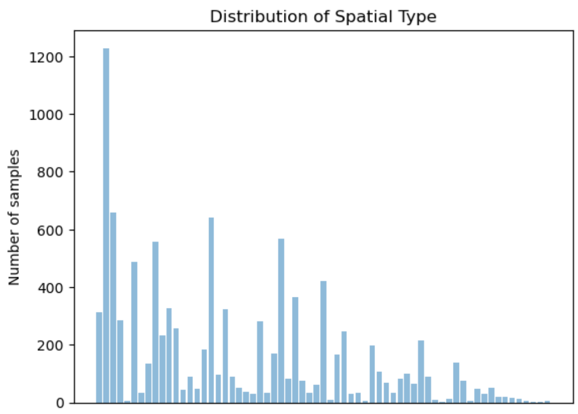
Caption: *The pizza is at the edge of the dinning table.*

The caption is: ☑ True □ False

Figure 5. Example of a spatial question

Source: Fangyu Liu et.al (2022)

The distributions of the dataset are shown in the following figures, we can see that the “Touching” type has the most images. On right hand side is the distribution of the response from the caption question, we can see that the dataset is almost balanced, with 4882 false labels and 5237 true labels.



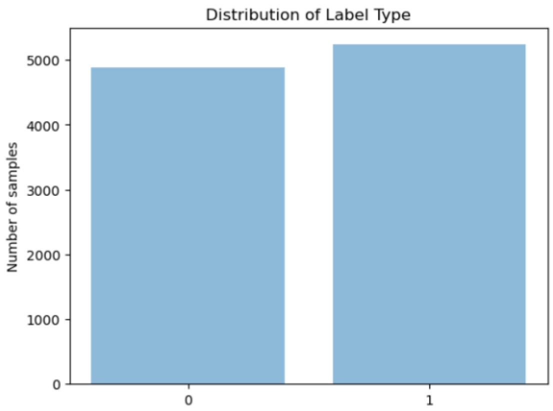


Figure6, Distribution of the spatial type

Figure7, Distribution of the label type

Note: the spatial type from left to right are *'above', 'touching', 'behind', 'next to', 'at', 'on top of', 'consists of', 'far from', 'under', 'inside', 'contains', 'in', 'against', 'close to', 'at the side of', 'left of', 'in front of', 'right of', 'beneath', 'across from', 'attached to', 'ahead of', 'into', 'facing', 'outside', 'beside', 'on', 'at the back of', 'at the left side of', 'over', 'connected to', 'perpendicular to', 'at the right side of', 'detached from', 'facing away from', 'below', 'opposite to', 'toward', 'past', 'at the edge of', 'part of', 'adjacent to', 'within', 'parallel to', 'near', 'off', 'far away from', 'surrounding', 'out of', 'down', 'across', 'away from', 'in the middle of', 'along', 'alongside', 'has as a part', 'by', 'with', 'enclosed by', 'beyond', 'down from', 'around', 'among', 'congruent', 'between'*.

Feature extraction

Feature extraction for this dataset uses Faster R-CNN method by (Shaoqing Ren et.al 2015), it uses ResNet + Feature Pyramid Network (FPN) by Tsung-Yi Lin et.al (2016) to generated features of the images and then pass into the Region Proposal Network (RPN) to generate 1000 region proposals, then uses NMS with a threshold 0.5 to give a class score for the boxes.

Baseline result

The following results are from Fangyu Liu’s paper.

Configuration:

Learning rate: 0.0001

Batch size: 32

Model	**Accuracy**	**Epoch**
VisualBERT	56.225	100

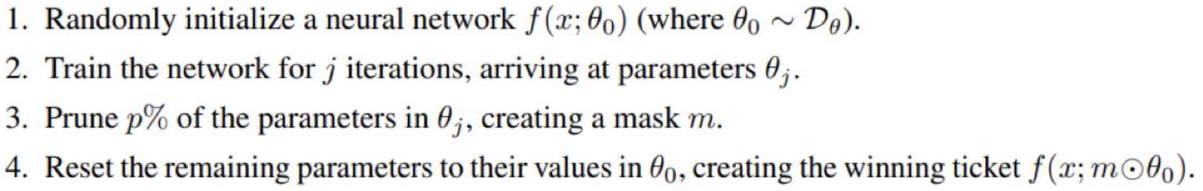
LXMERT	72.233	100

Table 3. VSR original result

Experiments settings

We will conduct our experiments on two different dataset (OPEN-I) and three different models (VisualBert, LXMERT and UNITER). Three evaluation metrics will be used for the experiment, the accuracy, fidelity and faithfulness/unfaithfulness accuracy. For all the evaluation metric, we performed two set of the experiments across 12 pruned percentages (25% 50%, 70%, 85%, 90%, 99%) and (10% 17%, 25%, 30%, 40%)with 10 runs. Assuming we are using (25% 50%, 70%, 85%, 90%, 99%) as our pruning percentages, where each run means for each pruned percentage, we run each pruning percentage starting from 0% (Original) to 99% 10 times. The prediction's data will be recorded at the end of entire training (3 epochs) for each percentage. Such training and recording happen to both of the two set of pruning percentages mentioned before. Also, for each run, we will train the models with 3 epochs due to the computation constraint. To evaluate the significant changes in accuracy, fidelity and faithfulness, we will be using t-test to indicate the significance. Additionally, we will also be using Wilcoxon test on accuracy as an additional significance test.

For the pruning procedure, we use the exact same pruning steps mentioned in the original lottery ticket hypothesis paper, which is shown below.



1. Randomly initialize a neural network $f(x; \theta_0)$ (where $\theta_0 \sim \mathcal{D}_\theta$).
2. Train the network for j iterations, arriving at parameters θ_j.
3. Prune $p\%$ of the parameters in θ_j, creating a mask m.
4. Reset the remaining parameters to their values in θ_0, creating the winning ticket $f(x; m \odot \theta_0)$.

Figure 8, Steps for identifying the wining ticket

In our experiment setting, we train the models for 3 epochs and for each of the epoch, we prune (10% 17%, 25%, 30%, 40%, 50%) and (25% 50%, 70%, 85%, 90%, 99%) of the parameters from the model. Thus, by following the steps mentioned in the above figure, we train the network for 1 iteration, and prune (10% 17%, 25%, 30%, 40%, 50%) and (25% 50%, 70%, 85%, 90%, 99%) of the parameters, then reset the remaining parameter to its initial values. This procedure will be repeated for 3 times, therefore we are doing an iterative magnitude pruning experiment.

Due to the limitation of PyTorch library, by specifying 10% of the parameter will not prune exact 10% of the parameters. Thus, we collect the number of parameters remaining in the final round and graph the following plot.

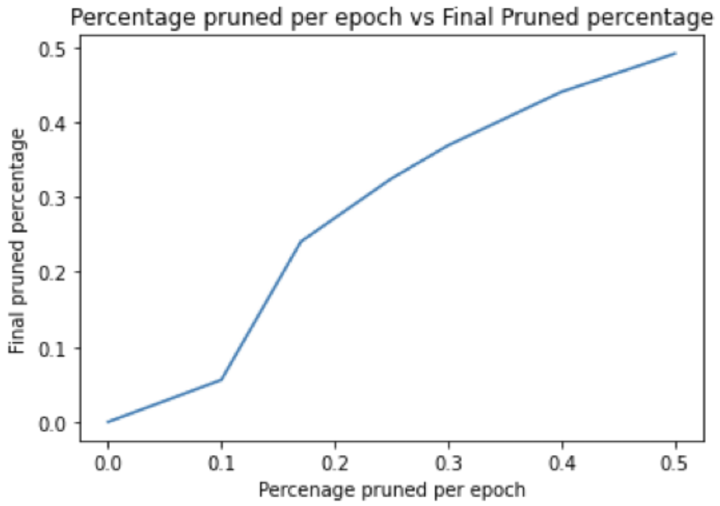


Figure9, final percentage of parameters

The figure above shows the final pruned percentage versus the percentage of parameters pruned per epoch; we can see that by pruning 10% of the parameters each epoch will lead to 7% of the parameters pruned in the final round.

For the technical aspect, we are using pytorch-lightning as our pruning library, the backbone of our code is from Fengyu Liu et.al (2022). For VisualBERT and UNITER model, we use *bert-base-uncased* as the tokenizer. For LXMERT, we used *lxmert-base-uncased* as the tokenizer. For pretrained models, *visualbert-nlvr2-coco-pre, lxmert-base-uncased, uniter-base* are used. For visual features, both VBERT, LXMERT and UNITER are generated by pretrained RCNN (region-based CNN) model, which is same as indicated in Fengyu Liu's paper.

Pruning Experiments

In this section, we will show the result obtained by the pruning percentages 25% 50%, 70%, 85%, 90%, 99%, which we refer to as secondary prune. For the pruning percentages 10% 17%, 25%, 30%, 40%, 50%, we refer to them as concluding prune. This naming convention is because we were working on the secondary pruning percentages and then we found a new set of pruning percentages show better result than the old ones. Thus, we name it concluding pruning.

Throughout the section, we mainly focus on three sets of evaluation metric, accuracy, the fidelity and sample faithfulness. For the accuracy, we will report the average accuracy for 10 runs across different pruning percentages, and we will also report the p value for the increasing or decreasing trend between different pruning percentages. E.g., the row (O, 0.25) in table 5 means the p value between original model and 25% pruned (per epoch) model is insignificant.

For the fidelity, we will report the p value table between different compared models, and you can find the average fidelity over 10 runs in the appendix. E.g. the cell ((O, 0.25) , (O, 0.5)) of table 7 means, p value between the fidelity value of the original model and 25% pruned (per epoch) model versus the fidelity value of the original model and 50% pruned (per epoch) model is insignificant.

For the sample faithfulness, we will report the average (un)faithful sample accuracy over 10 runs for each pruning percentage (per epoch) with respect to the original model. You can find the t-test result to analyze the accuracy significance between each compared model in the appendix.

The usage of three evaluation metrics, accuracy, fidelity and sample faithfulness are used to find the winning tickets, comparing the prediction distribution by using the prediction result and to test the recognition ability of the models.

For the secondary pruning experiments, as the trend is not obvious and result shows that we over-pruned the model, thus the conventional evaluation metrics (precision, recall and F1) are not included. For the concluding pruning experiments, we include the convention evaluation metrics, as well as the confusion matrix in the appendix. In addition, we also provide a summary of the experiments for each dataset at the end of the experiment result.

Accuracy

Spatial dataset – VBERT model (3 epochs, secondary prune)

From the table below we can see that the average accuracy has a clear decreasing trend, where the original model has the highest accuracy over all the models. The table shows the average accuracy over 10 runs.

Model	**Average accuracy**
Original (O)	0.52252
0.25	0.50464
0.5	0.48498
0.7	0.47134
0.85	0.47134
0.9	0.47134
0.99	0.47134

Table 4, average accuracy for VBERT under VSR

We will first evaluate the accuracy across 7 pruned percentages and evaluate the significance based on the t-test, the goal of this step is to find the wining ticket, we define the wining ticket to be the model with the pruned percentage that has an insignificant change compared to the original model.

We first compare all the original model (O) with all other pruned models using the t-test with 10 runs. From the table below we found that, by using p value 0.05 as the threshold for the significance indication. 25% pruned model shows a comparable performance in accuracy compared to the original model.

Comparing with the performance from the original paper by Fangyu Liu et.al (2022), they have achieved 57.4% accuracy by using 100 epochs. We averaged the performance over 10 runs and trained by 3 epochs, achieved 52.25% of the accuracy. For the wining tickets, we achieved an average of 50.46% of accuracy with 25% pruned model.

Model compared	**p-value**
O vs 0.25	0.08928
O vs 0.5	0.00079
O vs 0.75	9.18439e-8
O vs 0.85	9.18439e-8
O vs 0.9	9.18439e-8
O vs 0.99	9.18439e-8

Table 5, p value table for accuracy fluctuation

Wilcoxon test on the wining ticket

We will also perform Wilcoxon test on the rankings of pruned models based on their ranks to determine if a pruned model is a winning ticket or not. We will collect the ranks of pruned models based on their accuracy, and evaluate if they have significant difference in terms of the rank. For example, original model will have a set of rank A in terms of accuracy and will be compared with the other set of rank B from other pruned models.

Model compared	**p-value**
O vs 0.25	0.13028
O vs 0.5	0.00221
O vs 0.75	0.00782
O vs 0.85	0.00421
O vs 0.9	0.00443
O vs 0.99	0.00443

Table6, p value table for Wilcoxon test

From the table above, we can observe that if we take p value 0.05 as the significant threshold, then 25% pruned model shows no significant different with the original model, which has the same result from the t-test.

Fidelity

	(O, 0.5)	(O, 0.75)	(O, 0.85)	(O, 0.9)	(O, 0.99)
(O, 0.25)	0.81926	0.12078	0.12078	0.12078	0.12078
(O,0.5)		0.25560	0.25560	0.25560	0.25560
(O, 0.75)			1	1	1
(O, 0.85)				1	1
(O, 0.9)					1

Table 7, p value table for Fidelity of VBERT under VSR

The table above shows p value of the fidelity values' increasing or decreasing trend between different pruned percentages using the t-test. We can see that there's a clear decreasing between (O,0.25) and (O,0.75) where the O here represents the original model. However, such trend becomes not obvious if we see the rest of the table. So, this is also the reason that we did another set of analysis using pruning percentages 10% 17%, 25%, 30% and 40%. The average fidelity value for VisualBERT model across different pruning percentage can be found in the appendix

Faithfulness

Faithful and unfaithful sample accuracy:

Compared model	**Faithful sample accuracy**	**Unfaithful sample accuracy**
Origin vs 25% pruned	0.52080	0.51129
Origin vs 50% pruned	0.48106	0.52884
Origin vs 75% pruned	0.41883	0.53625
Origin vs 85% pruned	0.41883	0.53625
Origin vs 90% pruned	0.41883	0.53625
Origin vs 99% pruned	0.41883	0.53625

Table8, VBERT sample faithfulness

In this section we compared the faithful and unfaithful samples' accuracy across different pruning percentages. The table above represents the faithful/unfaithful sample average accuracy for 10 runs using the original model as the target model. The target model here refers to all other models will compare with this model. One obvious observation from the table is that the faithful sample accuracy is decreasing, and the unfaithful sample accuracy is increasing. However, such trend is not obvious, as we mentioned before we will use a new set of pruning percentages to see a more obvious trend. The t-test for faithful and unfaithful increasing and decreasing trend can be found in the appendix.

LXMERT- secondary pruning

Model	Average accuracy
Original (O)	0.61216
0.25	0.52266
0.5	0.51900
0.7	0.52744
0.85	0.52887
0.9	0.51556
0.99	0.51173

Table 9, LXMERT average accuracy under VSR

For the case of LXMERT, we can observe that the trend for average accuracy over 10 runs is also obvious, when there's no winning ticket, the average accuracy consistently decreasing. By using the t-test to compare pruned model's accuracy with the original model, we can see that there's a significant of decrease if we select p value using 0.05 as the threshold.

Model compared	p-value
O vs 0.25	7.70772e-16
O vs 0.5	1.16069 e-16
O vs 0.75	2.56710 e-16
O vs 0.85	5.99909 e-16
O vs 0.9	4.67587 e-16
O vs 0.99	8.59584 e-16

Table 10, p value table for accuracy fluctuation

	(O, 0.5)	(O, 0.75)	(O, 0.85)	(O, 0.9)	(O, 0.99)

(O, 0.25)	0.00370	0.00088	0.06115	0.34744	0.00228
(O,0.5)		0.36898	0.55669	0.05393	0.84804
(O, 0.75)			0.21819	0.01327	0.24091
(O, 0.85)				0.29001	0.61698
(O, 0.9)					0.04916

Table 11, p value table for fidelity change of LXMERT under VSR

Similar to the VBERT case, the table above shows the p value of fidelity's decreasing/increasing trend across different pruning percentage by using the t-test. If we use the p value of 0.05 as the threshold, we can see that the only significant difference happens between (O,0.25) and most of the pruned model. The trend is not obvious here. The average fidelity value for LXMERT model across different pruning percentage can be found in the appendix

Faithful and unfaithful sample accuracy:

Compared model	**Faithful sample accuracy**	**Unfaithful sample accuracy**
Origin vs 25% pruned	0.48967	0.44778
Origin vs 50% pruned	0.48749	0.45360
Origin vs 75% pruned	0.49715	0.43429
Origin vs 85% pruned	0.49845	0.44079
Origin vs 90% pruned	0.47570	0.45432
Origin vs 99% pruned	0.47355	0.45548

Table 12, LXMERT sample faithfulness under VSR

The table above shows the faithful and unfaithful sample accuracy across different pruning percentages. As we can see, there's no increasing nor deceasing trend happen here. The t-test for faithful and unfaithful increasing and decreasing trend can be found in the appendix.

UNITER- secondary pruning

Model	**Average accuracy**
Original (O)	0.62094
0.25	0.49624

0.5	0.50810
0.7	0.47134
0.85	0.47134
0.9	0.47134
0.99	0.47134

Table 13, average accuracy for UNITER under VSR

The above table shows the result obtained by UNITER model, as we can see from the table, the accuracy in general is decreasing, but the accuracy stay stable after 70% of the pruning. From the t-test table below we can see that, the drop in accuracy is significant. However, it's hard to conclude any trend here, as the decrease in accuracy is not monotonic , accuracy raises slightly between 25% and 50% of pruning.

Model compared	**p-value**
O vs 0.25	1.02443e-8
O vs 0.5	3.77065e-8
O vs 0.75	1.81917e-12
O vs 0.85	1.81917e-12
O vs 0.9	1.81917e-12
O vs 0.99	1.81917e-12

Table 14, p value table for accuracy change

	(O, 0.5)	(O, 0.75)	(O, 0.85)	(O, 0.9)	(O, 0.99)
(O, 0.25)	0.77848	0.88572	0.88572	0.88572	0.88572
(O,0.5)		0.67345	0.67345	0.67345	0.67345
(O, 0.75)			1	1	1
(O, 0.85)				1	1
(O, 0.9)					1

Table 15, p value table for Fidelity of UNITER under VSR

Similar to aforementioned evaluation, the table above shows the p values of fidelity's increase and decrease trend from the t-test. As we can observe, there's no significant changes everywhere in the table.

The average fidelity value for UNITER model across different pruning percentage can be found in the appendix

Faithful and unfaithful sample accuracy:

Compared model	Faithful sample accuracy	Unfaithful sample accuracy
Origin vs 25% pruned	0.49576	0.51398
Origin vs 50% pruned	0.52427	0.49777
Origin vs 75% pruned	0.45651	0.57021
Origin vs 85% pruned	0.45651	0.57021
Origin vs 90% pruned	0.45651	0.57021
Origin vs 99% pruned	0.45651	0.57021

Table 16, UNITER sample faithfulness

The table above shows the faithful and unfaithful sample accuracy for the UNITER model, as we can observe, there's no increasing or decreasing pattern happens here. The accuracy for faithful and unfaithful samples stay unchanged after 75% pruning percentage, this might indicate that the model might be over-pruned, as the prediction seems not working anymore. The t-test for faithful and unfaithful increasing and decreasing trend can be found in the appendix.

VisualBERT- Concluding Pruning

In this section, we use a new set of pruning percentages to redo our experiments. As shown from the previous experiments, both VisualBERT and UNITER are over pruned, and causes fidelity to be one for most of the cases. Thus, in this section, we choose to use a different set of pruning percentage (10% 17%, 25%, 30%, 40%) which is much less than before.

Spatial dataset – VBERT model (3 epochs, Concluding prune)

Accuracy

Model	Average accuracy
Original (O)	0.53369
0.1	0.51057
0.17	0.50760

0.25	0.51027
0.3	0.50029
0.4	0.50029
0.5	0.48003

Table 17, average accuracy for VBERT under VSR

The pruning above shows the results obtained from the new set of pruning percentages. As we can observe from the above table, the average accuracy has a clear decreasing trend. To verify our findings, we did the t-test analysis, and the p values are shown below.

Model compared	**p-value**
O vs 0.10	0.02255
O vs 0.17	0.00412
O vs 0.25	0.01862
O vs 0.3	0.00337
O vs 0.4	0.00387
O vs 0.5	2.20176e-07

Table 18, p value table for accuracy change

As we can observe from the table above, if we set the p value as 0.05, the decreasing trend is significant as all p values are below 0.05.

Fidelity

	(O, 0.17)	(O, 0.25)	(O, 0.3)	(O, 0.4)	(O, 0.5)
(O, 0.1)	0.50101	0.45173	0.15746	0.09224	0.00045
(O,0.17)		0.91263	0.45270	0.32891	0.00990
(O, 0.25)			0.53723	0.40861	0.02060
(O, 0.3)				0.85343	0.10452
(O, 0.4)					0.12839

Table 19, p value table for Fidelity of VBERT under VSR

The table above shows the p values for the fidelity's increasing and decreasing trend, the average fidelity shows a decreasing trend from low pruning percentage to high pruning percentage (see appendix), and the p values verified such a trend. As we can see the p values has a decreasing trend from left to right for each row, which indicates that the prediction distribution become more and more irrelevant with the original model.

Faithfulness

Faithful and unfaithful sample accuracy:

Compared model	Faithful sample accuracy	Unfaithful sample accuracy
Origin vs 10% pruned	0.53133	0.47527
Origin vs 17% pruned	0.53533	0.47219
Origin vs 25% pruned	0.48557	0.46709
Origin vs 30% pruned	0.47471	0.57866
Origin vs 40% pruned	0.48162	0.51363
Origin vs 50% pruned	0.42005	0.54072

Table 20, VBERT sample faithfulness

From the above table we can see that, for faithful sample accuracy, it shows a decreasing trend. However, this trend is not significant if we perform a t-test between the compared model (see appendix), but the trend of the p value also has a decreasing trend which indicates that the as the pruning percentage grows higher, its faithful sample accuracy become more and more significant to the original compared model. For the case of unfaithful sample accuracy, the table above shows an increasing trend, similar to the faithful sample accuracy, then trend is not obvious when observing the p values from the appendix, but the p values shows similar decreasing trend that indicates the unfaithful sample accuracy become more and more significant. Noted, the accuracy shown in the above table is an average accuracy over 10 runs.

Spatial dataset – LXMERT model (3 epochs, Concluding prune)

Accuracy

Model	Average accuracy
Original (O)	0.67207
0.1	0.55660
0.17	0.46979

0.25	0.52569
0.3	0.51623
0.4	0.47134
0.5	0.48023

Table 21, average accuracy for LXMERT under VSR

The pruning above shows the results obtained from the new set of pruning percentages for LXMERT. As we can see from the above table, unlike VsiualBERT, the trend for LXMERT is inconsistent, accuracy fluctuated between different pruning percentages.

Model compared	**p-value**
O vs 0.10	1.15400e-08
O vs 0.17	8.98571e-13
O vs 0.25	1.87861e-09
O vs 0.3	8.68895e-11
O vs 0.4	8.61188e-13
O vs 0.5	7.04395e-10

Table 22, p value table for accuracy change

As we can observe from the table above, if we set the p value as 0.05, the fluctuation is significant between different pruning percentages.

Fidelity.

	(O, 0.17)	(O, 0.25)	(O, 0.3)	(O, 0.4)	(O, 0.5)
(O, 0.1)	0.00546	0.00089	5.22767e-5	0.00671	0.00136
(O,0.17)		0.54912	0.28031	0.98270	0.64417
(O, 0.25)			0.65501	0.54086	0.88903
(O, 0.3)				0.27912	0.55049
(O, 0.4)					0.63367

Table 23, p value table for Fidelity of LXMERT under VSR

The table above shows the p values for the fidelity's increasing and decreasing trend, the average fidelity shows a fluctuation between different pruning percentages. Same as the accuracy, there's no obvious trend observed here.

Faithfulness

Faithful and unfaithful sample accuracy:

Compared model	Faithful sample accuracy	Unfaithful sample accuracy
Origin vs 10% pruned	0.57935	0.48823
Origin vs 17% pruned	0.47672	0.57804
Origin vs 25% pruned	0.55846	0.50493
Origin vs 30% pruned	0.56018	0.52288
Origin vs 40% pruned	0.47662	0.57504
Origin vs 50% pruned	0.49337	0.56286

Table 24, LXMERT sample faithfulness

From the above table we can see that, for both faithful sample accuracy and unfaithful sample accuracy there's no trend observed like the one we have for VisualBERT. Noted, the accuracy shown in the above table is an average accuracy over 10 runs.

UNITER – Concluding Pruning

Spatial dataset – UNITER model (3 epochs, Concluding prune)

Accuracy

Model	Average accuracy
Original (O)	0.60820
0.1	0.60103
0.17	0.52050
0.25	0.5
0.3	0.52149
0.4	0.47727
0.5	0.5

Table 25, average accuracy for UNITER under VSR

The pruning above shows the results obtained from the new set of pruning percentages for UNITER. As we can see from the above table, similar to LXMERT, the trend for UNITER is also inconsistent, accuracy fluctuated between different pruning percentages.

Model compared	**p-value**
O vs 0.10	0.75083
O vs 0.17	0.03498
O vs 0.25	0.00300
O vs 0.3	0.00216
O vs 0.4	0.00022
O vs 0.5	0.00300

Table 26, p value table for accuracy change

As we can observe from the table above, if we set the p value as 0.05, the fluctuation is significant between different pruning percentages except for the 10% pruning.

Fidelity

	(O, 0.17)	(O, 0.25)	(O, 0.3)	(O, 0.4)	(O, 0.5)
(O, 0.1)	0.66346	0.48791	0.11369	0.35730	0.10481
(O,0.17)		0.19817	0.01082	0.12014	0.01485
(O, 0.25)			0.35449	0.80740	0.30625
(O, 0.3)				0.50235	0.81571
(O, 0.4)					0.42778

Table 27, p value table for Fidelity of UNITER under VSR

The table above shows the p values for the fidelity's increasing and decreasing trend, the average fidelity shows a fluctuation between different pruning percentages. Same as the accuracy, there's no obvious trend observed here, most of the p values shows insignificant change.

Faithfulness

Faithful and unfaithful sample accuracy:

Compared model	Faithful sample accuracy	Unfaithful sample accuracy
Origin vs 10% pruned	0.57720	0.32609
Origin vs 17% pruned	0.48118	0.50095
Origin vs 25% pruned	0.48517	0.51771
Origin vs 30% pruned	0.53734	0.43297
Origin vs 40% pruned	0.46562	0.53328
Origin vs 50% pruned	0.52666	0.47622

Table 28, UNITER sample faithfulness

From the above table we can see that, for both faithful sample accuracy and unfaithful sample accuracy there's no trend observed like the one we have for VisualBERT. Noted, the accuracy shown in the above table is an average accuracy over 10 runs.

Summary for Spatial Dataset

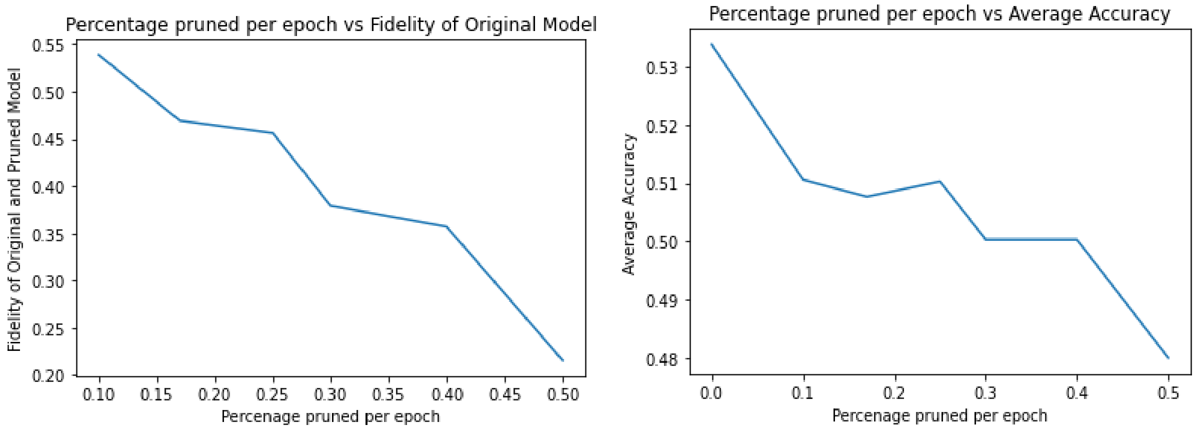
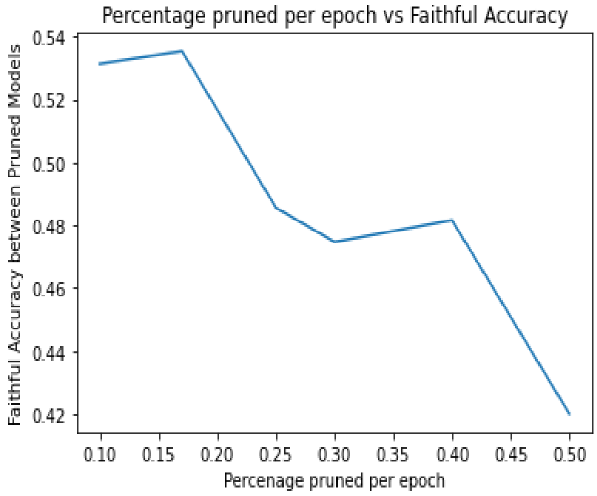


Figure 10. accuracy and fidelity of VBERT under VSR

In summary, for the Visual Spatial Reasoning dataset, we observed that the accuracy and fidelity of the VisualBERT model shares the same trend. We performed a Pearson correlation on these two sets of data and found that the p-value of their correlation is 0.00245, which indicate that they have a positive correlation.



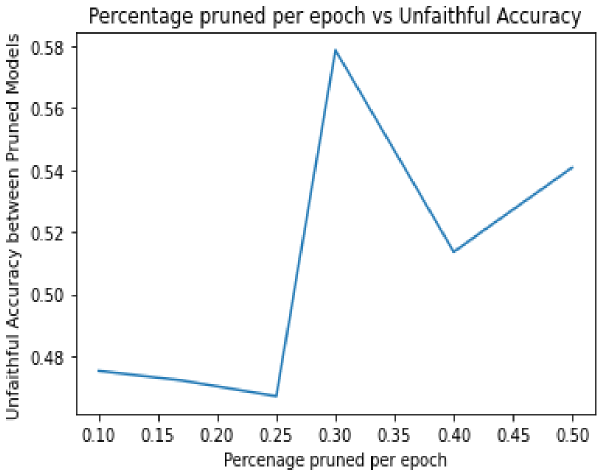
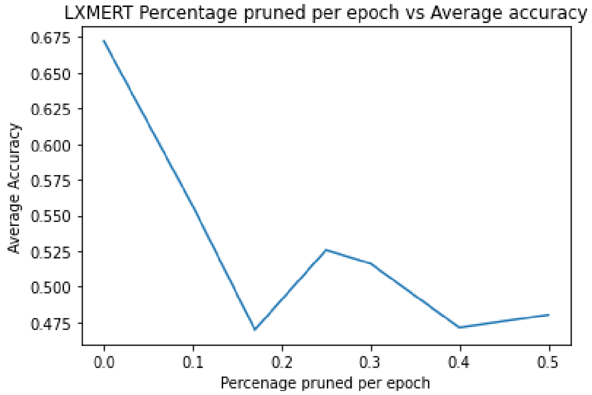


Figure 11. Faithful sample accuracy

Figure 12, Unfaithful sample accuracy

Figure 11 and 12 shows the faithful sample accuracy and unfaithful sample accuracy. As we can see, faithful sample accuracy has a clear decreasing trend. For unfaithful sample accuracy, the trend is not obvious, but it has an increasing tendency. Additionally, the Pearson correlation shows a p-value of 0.04786 on the faithful samples accuracy and the average sample accuracy.

All other models (LXMERT, UNITER) shows no trend in faithful and unfaithful samples' accuracy. However, their accuracy and fidelity curve are similar to the VisualBERT.



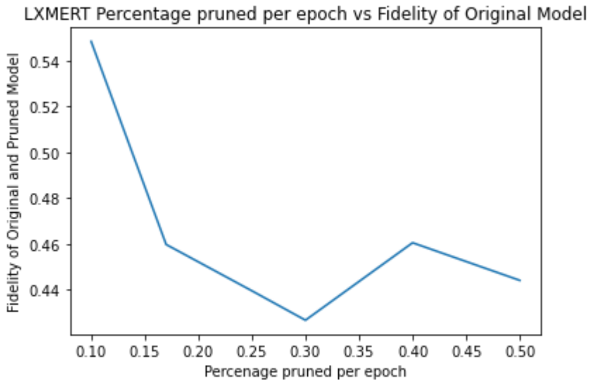


Figure 13, average accuracy for LXMERT under VSR Figure 14, Fidelity for LXMERT under VSR

For LXMERT, as we can observe from the above figures, both accuracy and fidelity show a decreasing trend. However, the p value for the Pearson's correlation between the fidelity and accuracy is 0.26400, if we use 0.05 as the significant threshold, then there's no correlation between the accuracy and fidelity for LXMERT.

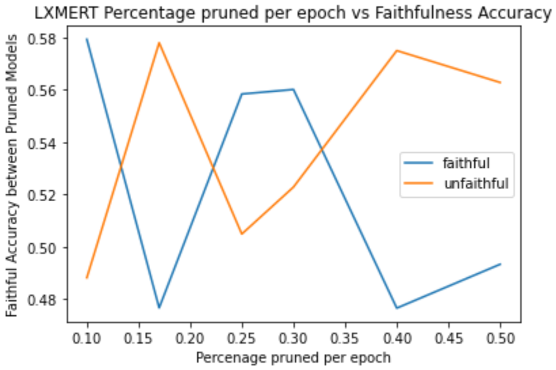
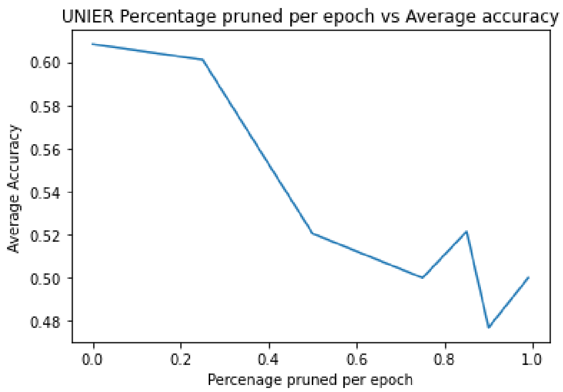


Figure 15, sample faithfulness for LXMERT under VSR

The above figure shows the faithful and unfaithful curves for LXMERT. As we can see, unlike the curves for VisualBERT where the faithful curve shows a constant decreasing trend, the faithful curve for LXMERT shows a vacillating trend. The same trend happens to the unfaithful curve, we observed no clear trend like the one we had in VisualBERT.



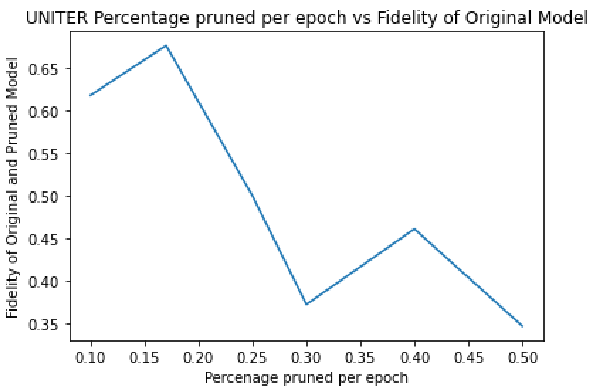


Figure 16, average accuracy for UNITER under VSR Figure 17, Fidelity for UNITER under VSR

The above figures shows the accuracy and fidelity curves for the UNITER model. Though it shares the similar trend as the VisualBERT, but the Pearson's correlation shows a p-value of 0.80794, if we pick the p-value 0.05 as the threshold, then there's no correlation between the decrease in fidelity and the decrease in accuracy for the model UNITER.

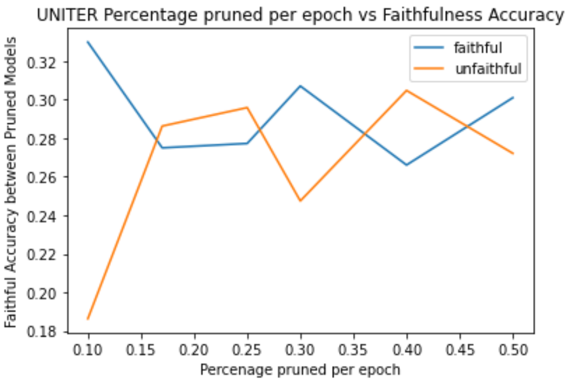
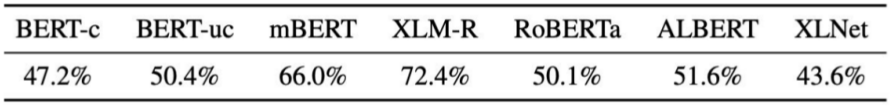


Figure 18, sample faithfulness for UNITER under VSR

The figure above shows the curves for unfaithful sample and faithful sample. Unlike the curves for VisualBERT and LXMERT, where faithful and unfaithful curves show a reverse trend, UNITER model shows an entangled relation between faithful and unfaithful curve. This also indicates that we cannot conclude a pattern like the one we had for the VisualBERT model.

Discussion on summary

In summary, we believe in this experiment pruning did not find the winning ticket for all models at all pruning percentages. This indicates that the pruning procedure has damaged the performance of the model. By using the concept of knowledge manifolds by Hassan Sajjad et.al (2022) where the author has used an unsupervised algorithm to visualize the knowledge manifold. Knowledge manifold is defined as the words that grouped together in high dimensional space, and it shows how model interpretate the latent space in terms of human language. They have also used an evaluation metric called latent concept, which is an alignment score that reflects the amount of linguistic hierarchy learnt in a model. They compared the latent concept learnt by the model with the human-defined concepts and found multilingual models like mBERT and XLM-R have higher alignment with human-defined concepts than monolingual models.



BERT-c	BERT-uc	mBERT	XLM-R	RoBERTa	ALBERT	XLNet
47.2%	50.4%	66.0%	72.4%	50.1%	51.6%	43.6%

Figure 13. Latent concept alignement score comparaison

By using the concept of knowledge manifold, a failed pruning like the one in our case, might have pruned away the knowledge manifolds and decrease the size of latent concepts. Thus, when we align with the test samples, the alignment score which is accuracy is our case will decrease. However, in our project, we are using a vision-language based dataset, different from the language dataset been used in Hassan Sajjad's paper, vision-language dataset is used to describe a specific kind of scenario. For example, the pizza is on the edge of a table (fig 5).

In this case, we believe the knowledge manifold might consider this scenario in two different dimensions, the vision and the language. The language part will be similar to what Hassa Sajjad described in their paper, where each word is associated with a group of words that has close Euclidean distance in latent space. For the vision part, similar to what language part has described, will also has latent concepts for each image, and those images can also be encoded into the latent space and can also use Euclidean distance to find the close image cluster associated with the image we want to target.

Also, for the architecture of LXMERT, where it is considered as a dual-stream architecture followed by a fusion encoder to combine the embeddings from vision and language, we believe that knowledge manifolds can be generated by the outputs from the fusion encoder instead of been generated by language encoder and vision encoder separately. Single-stream architecture models like VisualBERT and UNITER in our case, use transformer architecture to intake both vision and language features, can also use the knowledge manifold concept to generate latent concept by using the output from transformer directly.

The alignment score is another evaluation metric they proposed, it is used to measure the commonality between the model generated latent concepts and the human-defined latent concepts. By observing the formula they use, it shares a lot in common we use in the formula in fidelity. Where they compared the word cluster between the model generated cluster and human-defined cluster and compute how many of them have in common and normalized it, then it becomes the alignment score of the model with the human-defined latent concept.

In our case, we use the concept of fidelity to compute the alignment between original model and pruned model, instead of using the latent concept to compute how much commonalities they share, we compute how much commonality of the original and pruned model share by using their predictions' result. However, the alignment score they computed compare directly with the ground truth, which is the human-defined latent concepts, which means that by solely comparing the commonality is not enough, we have to also consider the accuracy with respect to the ground truth.

Thus, we proposed the concept of confidence, where it is a combination of commonality and accuracy evaluation. The model confidence in our case, first compute how many invariant predictions result from the original model to the pruned model, and then evaluate the accuracy over those invariant samples.

OPEN-I Experiments

From the table below, we can see that most of the accuracy are the same, so it less likely that these models will have similar pattern as the VBERT in Spatial dataset. In this case, we ceased to analyze the fidelity and sample faithfulness on OPEN-I dataset. Noted, visualization of OPEN-I experiments are available in the appendix. However, we can observe that the smaller learning rate tends to find the winning ticket in OPEN-I experiment settings. For both VisualBERT and UNITER model, the winning ticket appears at small learning rate. Here, we label the increased accuracy with red color and decreased accuracy with blue color. For OPEN-I dataset, we have used the secondary pruning percentages only.

VisualBERT

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	97.56	98.15	97.02	96.86	96.86	39.07	3.137
0.000075	96.86	96.86	96.86	96.86	96.86	70.16	3.137
0.0001	96.86	96.86	96.86	96.86	96.86	76.49	3.137
0.000125	96.86	96.86	96.86	96.86	96.86	48.99	3.137
0.00015	96.86	96.86	96.86	96.86	96.86	62.5	3.137

Table 29, VBERT accuracy under OPEN-I

UNITER

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	98.1	98.41	97.4	96.86	96.86	76.05	3.137
0.000075	96.86	96.86	96.86	96.86	96.86	69.96	3.137
0.0001	96.86	96.86	96.86	96.86	96.86	76.69	3.137
0.000125	96.86	96.86	96.86	96.86	96.86	83.63	3.137
0.00015	96.86	96.86	96.86	96.86	96.86	82.84	3.137

Table 30, UNITER accuracy under OPEN-I

LXMERT

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	96.86	96.86	97.91	97.67	96.86	96.86	3.137
0.000075	96.86	96.86	96.86	97.41	96.86	96.86	3.137
0.0001	96.86	96.86	96.86	97.05	96.86	96.86	3.137
0.000125	96.86	96.86	96.86	97.41	96.86	96.86	3.137
0.00015	96.86	96.86	96.86	96.86	96.86	96.86	3.137

Table 31, LXMERT accuracy under OPEN-I

Summary for OPEN-I dataset

For the OPNE-I dataset, we found the winning ticket for both of the three models. For the VisualBERT we found the winning ticket at 75% sparsity level with an accuracy of 98.15%, after 75% of sparsity, the accuracy has a decreasing trend for all learning rate level until 1% sparsity level with 3.137% of accuracy. If we compare the accuracy vertically by the learning rate, we found that for the sparsity from 100% to 25%, learning rate greater than 0.0005 has no effect on the accuracy. For sparsity lower than 25%, the accuracy has an obvious drop in accuracy. Especially for the 10% of the sparsity, the accuracy has a clear linear decrease when the learning rate raises. Thus, in the case of VisualBERT, we can conclude that the winning ticket indeed found that lower learning rate.

For the case of UNITER, it has similar trend with the VisualBERT, the winning ticket is found at 75% sparsity level with an accuracy of 98.41% and the accuracy shows an decreasing trend after that for all learning rate level, the accuracy keeps dropping until 1% of sparsity level with an accuracy of 3.137%. The performance at other learning rate level is similar to VisualBERT, where the accuracy stays the same for the sparsity level from 100% to 15% for all learning rate level above 0.00005. Same situation as the VisualBERT, the winning ticket is found at the lower leaning rate.

For LXMERT, different from the VisualBERT and UNITER, the winning ticket is found at 50% and 75% sparsity level with learning rate 0.00005, 0.000075, 0.0001, 0.000125 respectively. We have found the winning ticket for all learning rate level except for the 0.00015, which is the greatest learning rate we used. The decreasing trend is also different from VisualBERT and UNITER, the decrease only happens at 1% sparsity level for all learning rate and most of the accuracy stays the same from 100% to 50% sparsity level. Also, the winning tickets are found at learning rate level from 0/0005 to 0.000125, but not with the greatest learning rate, this also verifies the conclusion from the sanity checks of the lottery ticket hypothesis from XiaoLong Ma et.al (2021).

Finally, we found that most of the accuracy stays the same for most of the learning rate and most of the pruning percentages. We believe that there are two reasons for this, the first is that, as we can see from the dataset section, the OPEN-I dataset is highly imbalanced, most of the result contributed to “Normal” case. So, we believe that the imbalance causes model not to learn other types of diseases very well. The second reason is that, the feature engineering steps are not open-sourced, so we aren’t able to know what are the features have been used for this experiment. As we are using the open-sourced visual features from their repository, the visual features are pre-computed without detail description, so we assume that they have used a special way to compute the visual features so that the accuracy stays stable for most of the cases.

Additional Learning Rate Experiment

In addition to the learning rate experiment done on the OPEN-I dataset, we also did a minor learning rate experiment on the Visual Spatial Reasoning dataset. This experiment is prior to our experiments above. Thus, the UNITER model is not included in this experiment. Also, Different from the previous experiment, where we used a learning rate of 2e-6 on the VSR dataset, we use the secondary pruning percentages with learning rate (0.00005, 0.000075, 0.0001, 0.000125, 0.00015) to observe if smaller learning rate will result in a winning ticket. As mentioned before, since this experiment is prior to our experiments above, so the performance of the model might inferior and observed no obvious pattern compared to the aforementioned experiments.

VisualBERT

Accuracy

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0.5287	0.5287	0.5287	0.5287	0.5287	0.5287	0.5287
0.000075	0. 5287	0.5287	0.5287	0.5287	0.5287	0.4713	0.5287
0.0001	0. 5287	0.5287	0.5287	0.5287	0.5287	0.4713	0.5287
0.000125	0. 5287	0.5287	0.5287	0.4713	0.5287	0.4713	0.5287
0.00015	0. 5287	0.5287	0.5287	0.5287	0.5287	0.4713	0.5287

The table above shows the model's performance with secondary pruning percentage under different learning rate. As we can observe, most of the accuracy stays the same, except for the 10% pruning percentage, where the accuracy has a clear drop when the learning rate raises. If we observe the other metics (precision, recall ,F1-score) in the appendix, we can see that most of the values are invariant to the pruning percentages. However, in this case, we did find the winning ticket, since by the definition of winning ticket the accuracy of the pruned model has a comparable accuracy with the original model. Additionally, we also attached fidelity changed between the learning rate in the appendix, but we observed no obvious pattern for the fidelity change.

LXMERT

Accuracy

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0.5287	0.5287	0.5287	0.5287	0.5287	0. 4713	0. 4713
0.000075	0. 5287	0. 5287	0.5287	0.5287	0. 4713	0.4713	0. 4713
0.0001	0. 5287	0.5287	0.5287	0.5287	0.5287	0.4713	0. 4713
0.000125	0. 5287	0.5287	0.5287	0. 5287	0.5287	0.4713	0.4713
0.00015	0. 5287	0.5287	0.5287	0.5287	0.5287	0.4713	0.5287

The table above shows the learning rate experiment done by LXMERT, similar to the VisualBERT model, most of the values stay unchanged before and after the pruning. Except for the 10% and 1% case, where the accuracy has a clear drop compared to the previous pruning percentages. We also attached the other metrics evaluation in the appendix.

Limitations

There are also some limitations for our project. Throughout the project we have two main findings. Firstly, the positive correlation between fidelity and the accuracy only observed on Visual Spatial Dataset using the VisualBERT model, for the OPEN-I dataset, as we can see from the previous section, the accuracy remains the same up until 1% of the sparsity level, so the positive correlation is unlikely happens there. Secondly, the accuracy for faithful test samples drops as the pruning percentage raises, the accuracy for unfaithful test samples raise as the pruning percentage raises. However, these two behaviors are observed only with VisualBERT on visual spatial reasoning dataset, the other two models, LXMERT and UNITER do not have such a trend. Also, the OPEN-I dataset does not have such a trend either as its accuracy stays same for most of the pruning percentages and learning rate.

The previous mentioned findings cannot be generalized because of several reasons. Firstly, we did not test on enough datasets, if we can verify our findings on enough datasets, then we might conclude a stronger statement. Visual spatial reasoning dataset is a relatively new dataset and benchmark, there aren’t many papers and experiments on such a dataset yet. Many models were trained on tasks like visual questions answering and natural language inferences, no specialized task associated with spatial reasoning. So, training on such a dataset cannot guarantee the optimal performance on the models we tried.

Secondly, we did not train on enough number of epochs and wait enough number of training epochs to prune. In our project, we have only trained 3 epochs for each pruning percentage. This is because of our computation and time limit for this project. For example, training VisualBERT on spatial dataset take 3.5 hours for one run of experiments, and 3 hours for LXMERT, around 1 hour for UNITER model. The entire 10 runs takes a very long time, since we were training on Google Colab, consecutive training on the cloud for a very long time are not permitted because of Colab’s usage constraint. This computation constraint burdened our progress in experiments.

Also, as we mentioned in the introduction section, the steps for identifying the winning ticket need to train for n epoch before pruning. In our case, we pick the n to be one, so that we train every epoch and prune. Though the steps of identifying the winning tickets did not mention what are the influences for us to pick the n to be one, we assume that wait the model to train on more epochs have better performance than just wait the model to train on one epoch and prune, since as we can see from the sanity checks by Xiaolong Ma et.al (2021), one of the drawbacks in the original lottery ticket hypothesis is that they did not wait enough training epochs to prune.

Lastly, we believe that we should train on more models, in our project we only trained on 3 models. We believe the conclusion with more models will be persuasive. Especially for models with dual-architecture, we can see that from our experiment, dual-stream models tends to perform better in spatial dataset. We should include more dual-stream models like ViLBERT.

Future work

There are several future works can be done as an extension for this project. Firstly, the relation between forgettable/unforgettable training samples and unfaithful/faithful testing sample can be further investigated. As mentioned in the forgetting statistic paper, forgettable and unforgettable samples are visually different, which means the forgettable samples are blurred and clipped, they are usually hard to recognize compare to the unforgettable samples even from a human perspective. However, in our project, we did not extract faithful and unfaithful samples from the testing samples and visually inspect them, we believe this can be done in our further research. Also, in the original paper of forgetting statistic, they removed the forgettable samples from the training set and showed that they have no influence on the model's performance in terms of the accuracy. However, it does not make sense for us to remove unfaithful samples out of the testing samples, as we have to resolve all samples in the testing set. One thing that could be investigated is that the if the model been trained on the forgettable samples, if this will boost up the accuracy on the unfaithful samples' accuracy. In the other words, if the model is been specialized to train on the hard to learn samples, will this boots up the accuracy for the hard to learn samples in the testing set based on the assumption that the unfaithful samples are hard to learn.

The other future work that can be done is the behavior of fidelity and faithfulness in a winning ticket setting. In our project, for the visual spatial dataset, we observed the positive correlation in fidelity and accuracy in a non-winning ticket setting for the VisualBERT model. For the OPEN-I dataset, as the accuracy remain exact same, the fidelity has not changed in the experiment. We want to see how fidelity and faithfulness behaves in more datasets with a winning ticket setting.

Also, as we mentioned in the Approach section, we have defined the formula for the confidence score and the invariance of the model. In a further research, we want to compute the invariance score and the confidence score of the pruned model, and see if this can be used for measure model's performance in a pruning setting.

In the sanity checks paper by Xiaolong Ma et.al (2021), they also mentioned that resetting the weights help to find the winning ticket if there's a positive correlation between the original models' weight distribution and the pruned models' weight distribution. Our further research could also be extended by using the fidelity on weights distribution between the original model and the pruned model.

As we mentioned before, analysis through the knowledge manifold could also be applied to pruning. Analyzing the latent concepts between original model and pruned model could be a more explainable way to compute the fidelity instead of using the prediction results. As the latent concepts reflect the model's definition on a word in its latent space, it will be more trivial to see the impact of pruning on the model.

Conclusion

In conclusion, from a theoretical point of view, in this project, we proposed four evaluation metrics that are used to describe the performance between original and the pruned model. Firstly, we proposed the concept of fidelity, it is used to how much common the model share between and after the pruning. Secondly, we proposed the concept of sample faithfulness, it is the number of same predictions the model share between and after the pruning. Thirdly, we proposed the invariance score, which is used to describe how many testing samples' prediction remain the same before and after the pruning. Lastly, we proposed the confidence score, it is used to describe within those testing samples remains the same before and after the pruning, how many of them are correctly classified.

From an experimental perspective, in this project we tried to use the concept of fidelity and sample faithfulness on the pruning models based on the Lottery Ticket Hypothesis. We evaluated the fidelity and sample faithfulness on three vision-language models (VisualBERT, LXMERT and UNITER) over two datasets (VSR, OPEN-I). we picked two set of pruning percentages, secondary percentages (25% 50%, 70%, 85%, 90%, 99%) and concluding percentages (10% 17%, 25%, 30%, 40%, 50%). We use the both secondary percentages and concluding percentages on visual spatial reasoning dataset and use secondary percentages on OPEN-I dataset. For each dataset, each model and each pruning percentage, we tested our experiment 10 times to get the statistical significance.

For the visual spatial dataset, we found positive correlation between the fidelity and accuracy for VisualBERT when there's no winning ticket. We also found that there's a decreasing trend on the accuracy of number of faithful samples while as an increasing trend on the accuracy of number of unfaithful samples.

The decreasing trend on the accuracy over the faithful samples indicates that, when there's no winning ticket, the accuracy of faithful samples drops as the pruning percentages raises. The correlation test also shows there's a negative correlation on the overall accuracy and the accuracy over the faithful samples. On the other hand, the increasing trend on the accuracy over the unfaithful samples indicates that there might be a winning ticket over the unfaithful samples. However, as the total accuracy shows a decreasing trend, this means faithful samples have more influence over the unfaithful samples in this experiment.

Comparing by solely rely on the accuracy to show performance of pruning, fidelity and faithfulness shows more details about the pruning. These two metrics indicates how the prediction of test samples changes after the pruning. Also, the fidelity and the accuracy over faithful samples shows the correlation with the accuracy to make the metric more reliable. Thus, the fidelity and sample faithfulness can be alternative evaluation metrics to show and explain the performance of the pruning.

For the OPEN-I dataset, we found the winning ticket at 75% sparsity level for VisualBERT and UNITER model. For the LXMERT, we found the winning ticket at both 75% and 50% sparsity level. However, most of the accuracy from the experiment stays exact the same and indicate they the fidelity and sample faithfulness experience no change during the pruning. The other findings we found is that most of the winning ticket, like VisualBERT and UNITER, their winning tickets are indeed found at lower learning rate, this verifies the conclusion from the sanity checks on lottery ticket hypothesis done by XiaoLong ma et.al (2021).

For the future work and analysis, we reviewed Hassan Sajjad's work, which they used knowledge manifold to describe the latent space of each word in a sentence from the model's perspective. We think this idea will be benefit to our work, especially it can be merged with the concept of fidelity we propose.

References

Ma, X., Yuan, G., Shen, X., Chen, T., Chen, X., Chen, X., Liu, N., Qin, M., Liu, S., Wang, Z., & Wang, Y. (2021). Sanity Checks for Lottery Tickets: Does Your Winning Ticket Really Win the Jackpot? *NeurIPS*.

Toneva, M., Sordoni, A., Combes, R.T., Trischler, A., Bengio, Y., & Gordon, G.J. (2019). An Empirical Study of Example Forgetting during Deep Neural Network Learning. *ArXiv, abs/1812.05159*.

Milli, S., Schmidt, L., Dragan, A.D., & Hardt, M. (2019). Model Reconstruction from Model Explanations. *Proceedings of the Conference on Fairness, Accountability, and Transparency*.

Zhao, X., Zhang, W., Xiao, X., & Lim, B.Y. (2021). Exploiting Explanations for Model Inversion Attacks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 662-672.

Zhang, Z., Chen, X., Chen, T., & Wang, Z. (2021). Efficient Lottery Ticket Finding: Less Data is More. *ArXiv, abs/2106.03225*.

Gan, Z., Chen, Y., Li, L., Chen, T., Cheng, Y., Wang, S., & Liu, J. (2022). Playing Lottery Tickets with Vision and Language. *ArXiv, abs/2104.11832*.

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., & Carbin, M. (2020). The Lottery Ticket Hypothesis for Pre-trained BERT Networks. *ArXiv, abs/2007.12223*.

Zheng, R., Rong, B., Zhou, Y., Liang, D., Wang, S., Wu, W., Gui, T., Zhang, Q., & Huang, X. (2022). Robust Lottery Tickets for Pre-trained Language Models. *ACL*.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6077-6086.

Frankle, J., Dziugaite, G.K., Roy, D.M., & Carbin, M. (2019). Stabilizing the Lottery Ticket Hypothesis. *arXiv: Learning*.

Frankle, Jonathan and Michael Carbin. “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks.” *arXiv: Learning* (2019): n. pag.

Li, Y., Wang, H., & Luo, Y. (2020). A Comparison of Pre-trained Vision-and-Language Models for Multimodal Representation Learning across Medical Images and Reports. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1999-2004.

Demner-Fushman, D., Kohli, M.D., Rosenman, M.B., Shooshan, S.E., Rodriguez, L.M., Antani, S.K., Thoma, G.R., & McDonald, C.J. (2016). Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association : JAMIA, 23 2*, 304-10 .

Liu, B., Zhan, L., Xu, L., Ma, L., Yang, Y.F., & Wu, X. (2021). Slake: A Semantically-Labeled Knowledge-Enhanced Dataset For Medical Visual Question Answering. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 1650-1654.

He, X., Zhang, Y., Mou, L., Xing, E.P., & Xie, P. (2020). PathVQA: 30000+ Questions for Medical Visual Question Answering. *ArXiv, abs/2003.10286*.

Chen, Y., Li, L., Yu, L., Kholy, A.E., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2020). UNITER: UNiversal Image-TExt Representation Learning. *ECCV*.

Tan, H.H., & Bansal, M. (2019). LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *ArXiv, abs/1908.07490*.

Li, L.H., Yatskar, M., Yin, D., Hsieh, C., & Chang, K. (2019). VisualBERT: A Simple and Performant Baseline for Vision and Language. *ArXiv, abs/1908.03557*.

Suhr, A., Lewis, M., Yeh, J., & Artzi, Y. (2017). A Corpus of Natural Language for Visual Reasoning. *ACL*.

Du, Y., Liu, Z., Li, J., & Zhao, W.X. (2022). A Survey of Vision-Language Pre-Trained Models. *IJCAI*.

Girshick, R.B. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440-1448.

Ren, S., He, K., Girshick, R.B., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*, 1137-1149.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2017). Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6077-6086.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR, abs/1409.1556*.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248-255.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., & Fei-Fei, L. (2016). Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision, 123*, 32-73.

Dalvi, Fahim et al. “Discovering Latent Concepts Learned in BERT.” *ArXiv* abs/2205.07237 (2022): n. pag.

Mandal, Sampurna et al. “Single shot detection for detecting real-time flying objects for unmanned aerial vehicle.” *Artificial Intelligence for Future Generation Robotics* (2021): n. pag.

Appendix

Secondary pruning

VisualBERT

Average fidelity

	O	25	50	75	85	90	99
O		0.37995	0.35635	0.23263	0.23263	0.23263	0.23263
25			0.4035	0.37359	0.37359	0.373591	0.373591
50				0.73277	0.73277	0.73277	0.732777
75					1	1	1
85						1	1
90							1

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.09014936	0.057121296	0.0571212	0.0571212	0.05712129
(O,50)		0.2614304	0.2614304	0.2614304	0.2614304
(O,75)			1	1	1
(O,85)				1	1
(O,90)					1

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)

(O,25)	0.178060	0.05780247	0.0578024	0.05780247	0.057802
(O,50)		0.3307701	0.3307701	0.33077019	0.330770
(O,75)			1	1	1
(O,85)				1	1
(O,90)					1

LXMERT

Average fidelity

	O	25	50	75	85	90	99
O		0.50564	0.47626	0.4683	0.4827511	0.49581	0.47761
25			0.54153	0.54805	0.511522	0.49414	0.508591
50				0.624552	0.59283	0.57617	0.48219
75					0.6300801	0.6031	0.3838045
85						0.67581	0.40209
90							0.307250

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.695256234	0.1011713	0.1196057	0.0332415	0.0045403
(O,50)		0.054347	0.06912955	0.078695	0.01690
(O,75)			0.7797449	0.00083	1.83678e-15
(O,85)				0.0018321	0.00014274
(O,90)					0.719885880

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.4432535	0.0838831	0.40539050	0.50161224	0.2865360
(O,50)		0.0014763	0.0567537	0.928718	0.6886526
(O,75)			0.31144	0.0207142	0.0001919
(O,85)				0.136627	0.020856
(O,90)					0.879786

UNITER

Average fidelity

	O	25	50	75	85	90	99
O		0.4742791	0.45406	0.485180	0.4851808	0.48518	0.48518
25			0.371497	0.570223	0.57022	0.57022	0.5702235
50				0.400331	0.40033	0.40033	0.400331
75					1	1	1
85						1	1
90							1

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.39782517	0.15985944	0.159859	0.159859	0.159859
(O,50)		0.024797787	0.024797	0.024797787	0.0247977871
(O,75)			1	1	1

(O,85)				1	1
(O,90)					1

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.6194865	0.048456	0.0484561	0.04845612	0.048456
(O,50)		0.0214312	0.0214312	0.02143128	0.02143128
(O,75)			1	1	1
(O,85)				1	1
(O,90)					1

Concluding Pruning

VisualBERT

Average fidelity

	O	25	50	75	85	90	99
O		0.538844	0.46906	0.456096	0.379046	0.35695	0.215596
25			0.52169	0.71855	0.33834	0.173356	0.237191
50				0.4430928	0.23530	0.36569	0.4146
75					0.325943	0.27773	0.2848
85						0.58257	0.37381
90							0.62708

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.68440807	0.41573568	0.30784729	0.373367	0.133452

(O,50)		0.37713	0.276096	0.3371969	0.120946
(O,75)			0.88859	0.959551	0.47169
(O,85)				0.92863	0.544604
(O,90)					0.4970370

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.9680209	0.9147714	0.1662838	0.49771	0.24252
(O,50)		0.946476	0.1521740548	0.4595717	0.216908
(O,75)			0.132490	0.40274	0.1818453
(O,85)				0.20146838	0.4359765
(O,90)					0.07375

Other metrics

Model	**Average F1**
Original (O)	0.60889745
0.1	0.48646965
0.17	0.50358009
0.25	0.4534074
0.3	0.34694588
0.4	0.34295185
0.5	0.1074827

Model	**Average Precision**
Original (O)	0.54821783

0.1	0.53476564
0.17	0.54338313
0.25	0.3760512
0.3	0.36432806
0.4	0.26496417
0.5	0.17220373

Model	**Average Recall**
Original (O)	0.71420561
0.1	0.6588785
0.17	0.61158879
0.25	0.6282243
0.3	0.50056075
0.4	0.4871028
0.5	0.12635514

Model	**Average tn, fp, fn, tp (decimal are removed)**
Original (O)	158 319 152 382
0.1	164 312 182 352
0.17	186 290 207 327
0.25	180 296 198 336
0.3	238 238 267 267
0.4	245 231 274 260

0.5	418 58 467 67

LXMERT

Average fidelity

	O	25	50	75	85	90	99
O		0.548393	0.45980	0.43959	0.42677	0.460565	0.44415
25			0.48528	0.42799	0.413985	0.481433	0.469838
50				0.342189	0.328093	0.9445049	0.834662
75					0.51581	0.32187	0.38005
85						0.30699	0.37056
90							0.889226

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	2.1320262e-6	0.0695542	0.2404921	2.3937911e-6	5.90279e-5
(O,50)		2.52172e-5	0.000335	0.99430	0.30964
(O,75)			0.91446214	2.787055e-5	0.0007208
(O,85)				0.00035	0.00348
(O,90)					0.310215

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	8.806261e-5	0.482562	0.1095705	0.00014477	0.007284
(O,50)		0.008508	0.01868	0.856408	0.5280157
(O,75)			0.511025	0.011418	0.0681294
(O,85)				0.0260801	0.15814351

(O,90)					0.6148813

Other metrics

Model	**Average F1**
Original (O)	0.66770696
0.1	0.53959578
0.17	0.05182571
0.25	0.57804142
0.3	0.58612965
0.4	0
0.5	0.09238666

Model	**Average Precision**
Original (O)	0.71972172
0.1	0.60004441
0.17	0.42021919
0.25	0.543982
0.3	0.53509
0.4	0
0.5	0.07704655

Model	Average Recall
Original (O)	0.6293725
0.1	0.49799733
0.17	0.0376502
0.25	0.69933244
0.3	0.69933244
0.4	0
0.5	0.11535381

Model	**Average tn, fp, fn, tp (decimal are removed)**
Original (O)	343 133 198 336
0.1	296 180 268 266
0.17	455 21 514 20
0.25	157 319 160 374
0.3	148 328 160 374
0.4	477 0 535 0
0.5	424 52 473 61

UNITER

Average fidelity

	O	25	50	75	85	90	99
O		0.617958	0.67641	0.499891	0.372250	0.46094	0.34694
25			0.4992367	0.47045	0.618089	0.265858	0.32974
50				0.5992747	0.31761	0.6314117	0.42991

75					0.4615247	0.385024	0.555555
85						0.25492	0.59657
90							0.646080

Faithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.049605	0.04215040	0.1500063	0.0188996	0.216990
(O,50)		0.9406372	0.2492479	0.7696218	0.414850
(O,75)			0.2507454	0.699310	0.433402
(O,85)				0.1260903	0.806961069
(O,90)					0.258296

Unfaithful t-test

	(O,50)	(O,75)	(O,85)	(O,90)	(O,99)
(O,25)	0.0337636633	0.013735	0.05110157	0.0040907	0.0293830
(O,50)		0.787267	0.21147369	0.54553751	0.67795994
(O,75)			0.063020	0.70786427	0.41405488
(O,85)				0.002501	0.234979
(O,90)					0.158771656

Other metrics

Model	**Average F1**
Original (O)	0.58386415

0.1	0.68230857
0.17	0.32532928
0.25	0.34583064
0.3	0.66786877
0.4	0.14792167
0.5	0.34583064

Model	**Average Precision**
Original (O)	0.66918734
0.1	0.59607475
0.17	0.28672469
0.25	0.26432806
0.3	0.52653382
0.4	0.244982
0.5	0.26432806

Model	**Average Recall**
Original (O)	0.57242991
0.1	0.82383178
0.17	0.39205607
0.25	0.5
0.3	0.92476636
0.4	0.15186916
0.5	0.5

Model	**Average tn, fp, fn, tp (decimal are removed)**
Original (O)	309 167 228 306
0.1	167 309 94 440
0.17	317 160 325 209
0.25	238 238 267 267
0.3	33 444 40 494
0.4	401 75 453 81
0.5	238 238 267 267

OPEN-I

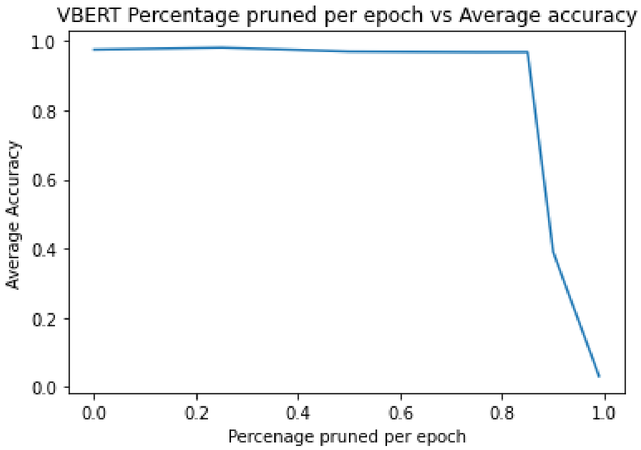


Figure 19, average accuracy for VBERT under OPEN-I

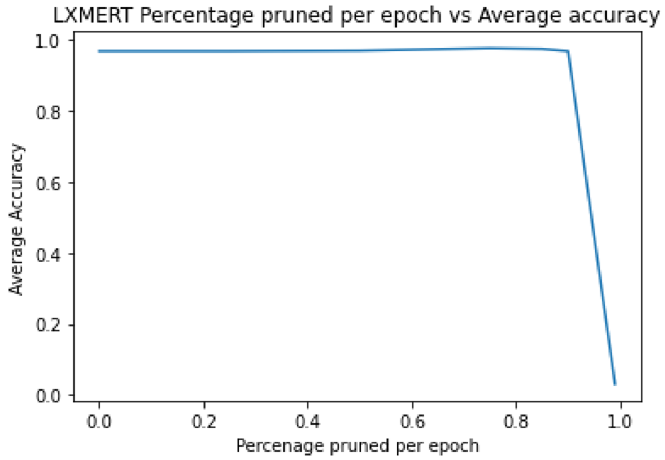


Figure 20, average accuracy for LXMERT under OPEN-I

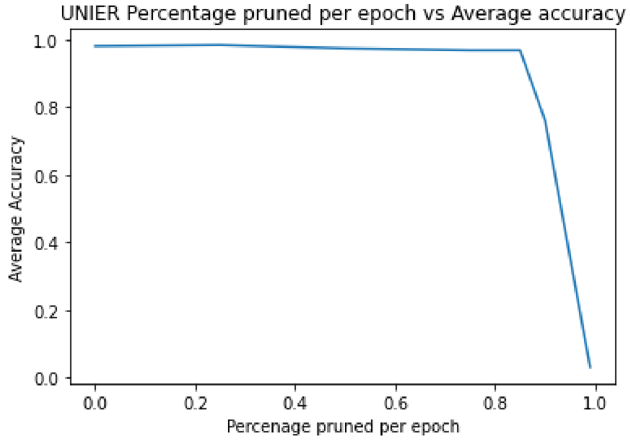


Figure 21, average accuracy for UNITER under OPEN-I

Learning rate experiment

VisualBERT

Other metrics

F1-SCORE

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0. 6879	0.6879	0.6879	0.6879	0.6879	0.6879	0.6879
0.000075	0. 6879	0.6879	0.6879	0.6879	0.6879	0	0.6879

0.0001	0. 6879	0.6879	0.6879	0.6879	0.6879	0	0.6879
0.000125	0. 6879	0.6879	0.6879	0	0.6879	0	0.6879
0.00015	0. 6879	0.6879	0.6879	0.6879	0.6879	0	0.6879

PRECISION

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0. 5287	0.5287	0.5287	0.5287	0.5287	0.5287	0.5287
0.000075	0. 5287	0.5287	0.5287	0.5287	0.5287	0	0.5287
0.0001	0. 5287	0.5287	0.5287	0.5287	0.5287	0	0.5287
0.000125	0. 5287	0.5287	0.5287	0	0.5287	0	0.5287
0.00015	0. 5287	0.5287	0.5287	0.5287	0.5287	0	0.5287

RECALL

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	1	1	1	1	1	1	1
0.000075	1	1	1	1	1	0	1
0.0001	1	1	1	1	1	0	1
0.000125	1	1	1	0	1	0	1
0.00015	1	1	1	1	1	0	1

LXMERT

Other metrics

F1-SCORE

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0. 6879	0. 6538	0. 6538	0. 6538	0. 6538	0	0
0.000075	0. 6879	0. 6538	0. 6538	0. 6538	0	0	0
0.0001	0. 6879	0. 6538	0. 6538	0. 6538	0. 6538	0	0
0.000125	0. 6879	0. 6538	0. 6538	0. 6538	0. 6538	0	0
0.00015	0. 6879	0. 6538	0. 6538	0. 6538	0. 6538	0	0.6538

PRECISION

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%

0.00005	0. 5287	0. 5287	0.5287	0.5287	0.5287	0	0
0.000075	0. 5287	0.5287	0.5287	0.5287	0	0	0
0.0001	0. 5287	0.5287	0.5287	0.5287	0.5287	0	0
0.000125	0. 5287	0.5287	0.5287	0. 5287	0.5287	0	0
0.00015	0. 5287	0.5287	0.5287	0.5287	0.5287	0	0.5287

RECALL

LR/ Parameter remaining	100%	75%	50%	25%	15%	10%	1%
0.00005	0. 9684	0. 9684	0. 9684	0. 9684	0. 9684	0	0
0.000075	0. 9684	0. 9684	0. 9684	0. 9684	0	0	0
0.0001	0. 9684	0. 9684	0. 9684	0. 9684	0. 9684	0	0
0.000125	0. 9684	0. 9684	0. 9684	0. 9684	0. 9684	0	0
0.00015	0. 9684	0. 9684	0. 9684	0. 9684	0.9684	0	0.9684

Fidelity

Visual spatial reasoning - VBERT – LR 0. 000075 - Fidelity

Percentage	0.25	0.5	0.75	0.85	0.9	0.99
0.25	100%	-	-	-	-	-
0.5	43.5783%	100%	-	-	-	-
0.75	50.816%	51.281%	100%	-	-	-
0.85	31.263%	43.721%	51.3771%	100%	-	-
0.9	30.758%	20.585%	37.177%	30.924%	100%	
0.99	45.121%	32.667%	49.193%	41.865%	32.314%	100%

Visual spatial reasoning - VBERT – LR 0. 0001 - Fidelity

Percentage	0.25	0.5	0.75	0.85	0.9	0.99
0.25	100%	-	-	-	-	-
0.5	31.2928%	100%	-	-	-	-
0.75	31.1302%	44.151%	100%	-	-	-
0.85	43.536%	44.020%	44.2422%	100%	-	-
0.9	20.349%	31.292%	43.417%	43.5369%	100%	
0.99	76.771%	43.871%	43.697%	58.607%	31.094%	100%

Visual spatial reasoning - VBERT – LR 0. 000125 - Fidelity

Percentage	0.25	0.5	0.75	0.85	0.9	0.99
0.25	100%	-	-	-	-	-
0.5	43.8793%	100%	-	-	-	-

0.75	31.4657%	44.1518%	100%	-	-	-
0.85	36.9646%	25.961%	26.1404%	100%	-	-
0.9	37.137%	26.106%	50.559%	43.2113%	100%	
0.99	509828%	37.633%	50.835%	43.555%	43.718%	100%

Visual spatial reasoning - VBERT – LR 0. 000125 - Fidelity

Percentage	0.25	0.5	0.75	0.85	0.9	0.99
0.25	100%	-	-	-	-	-
0.5	43.8793%	100%	-	-	-	-
0.75	31.4657%	44.1518%	100%	-	-	-
0.85	36.9646%	25.961%	26.1404%	100%	-	-
0.9	37.137%	26.106%	50.559%	43.2113%	100%	
0.99	50.9828%	37.633%	50.835%	43.555%	43.718%	100%

Visual spatial reasoning - VBERT – LR 0. 00015 - Fidelity

Percentage	0.25	0.5	0.75	0.85	0.9	0.99
0.25	100%	-	-	-	-	-
0.5	36.999645%	100%	-	-	-	-
0.75	36.493%	43.8540%	100%	-	-	-
0.85	36.9646%	44.020%	31.690%	100%	-	-
0.9	30.4131%	36.999%	36.530%	36.964%	100%	
0.99	58.393%	37.6336%	50.559%	51.0092%	20.7077%	100%