PIXEL-LEVEL TASK HELPS PRUNED NETWORK TRANSFER TO DOWNSTREAM TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pruning well-trained neural networks is effective to achieve a promising accuracy-efficiency trade-off in computer vision regimes. However, most of existing pruning algorithms only focus on the classification task defined on the source domain. Different from the strong transferability of the original model, a pruned network is hard to transfer to complicated downstream tasks such as object detection Girish et al. (2021). In this paper, we show that the image-level pretrain task is not capable of pruning models for diverse downstream tasks. To mitigate this problem, we introduce image reconstruction, a pixel-level task, into the traditional pruning framework. Concretely, an autoencoder is trained based on the original model, and then the pruning process is optimized with both autoencoder and classification losses. The empirical study on benchmark downstream tasks shows that the proposed method can outperform state-of-the-art results explicitly.

1 INTRODUCTION

Fine-tuning a pre-trained model, which can leverage the knowledge from a large-scale pre-training dataset, becomes prevalent for downstream tasks. This strategy avoids overfitting on small datasets leading to better performance on target tasks. Benefits from the pretrain-finetune strategy, scaling up model capacity is a trend in recent research Touvron et al. (2021); Dosovitskiy et al. (2020); Liu et al. (2021). However, large-scale models consume a lot of computational and memory resources, limiting their applications on edge devices. Many efforts are devoted to reducing the computational requirements of neural networks Hubara et al. (2017); Hinton et al. (2015); Tai et al. (2016). Among them, pruning Han et al. (2015) aims to remove unimportant parameters from the original model and can reduce the size of the model significantly. Most pruning methods rely on a well-trained network and can achieve extraordinary compression rates with negligible accuracy drop on the same task. Han et al. (2015); Yang et al. (2019); Sanh et al. (2020)

Although pruning methods demonstrate an excellent accuracy vs. sparsity trade-off, only a few works evaluate the pruned model's transferability, i.e., the performance on different downstream tasks. Given multiple downstream tasks, a pruning algorithm can be applied to the individual task. However, the cost that linearly depends on the number of tasks will become intractable. To mitigate this problem, we try to find a pruned model, called universal winning tickets, that can transfer to diverse downstream tasks.

The lottery tickets hypothesis, proposed by Frankle & Carbin (2018), claims that each over-parameterized neural network has a sparse subnetwork called a winning ticket, which can achieve the same performance as the entire network. The transferability of winning tickets has been investigated in Morcos et al. (2019). They show that an ImageNet ticket can transfer to different downstream classification tasks. In Chen et al. (2021), the authors suggest that a pretraining procedure can be regarded as a special initialized method. This kind of initialization is directly amenable to sparsification. Based on this insight, the authors use task agnostic pretraining to help find the universal winning tickets. Their results show that a universal winning ticket exists across different classification downstream tasks.

However, in some more complicated downstream tasks, such as object detection, tickets found by Chen et al. (2021) can result in a degenerated performance. In Chen et al. (2021), the authors reveal that tickets found by the target object detection task surpass tickets found by image classification with a non-negligible margin. In Girish et al. (2021), the authors check tickets found by supervised learning

on an object detection dataset Lin et al. (2014). The result confirms that ImageNet tickets only transfer to a limited extent to downstream tasks, such as object detection or instance segmentation. These observations illustrate that the pruning method's transferability is highly related to the task type.

In this paper, we aim to find the universal tickets for diverse downstream tasks. Most of the existing methods rely on an image-level task to prune pre-trained models. Although the pruning pipeline has shown an extraordinary performance on downstream tasks He et al. (2020), it does not treat details and global features in the same status. The implicit tendency of image-level loss causes the neural network to forget pixel-level information during the pruning process. After pruning, pixel-level information becomes untraceable while it is essential for complicated downstream tasks, e.g., detection, segmentation, etc. The intuition is theoretically analyzed in Section 3. Therefore, a pixel-level task is necessary for pruning pre-trained models to preserve sufficient information and can help the model transfer to generic downstream tasks.

Unlike image-level tasks, designing appropriate pixel-level tasks is still challenging. Inspired by the recent progress in self-supervised learning He et al. (2021), we introduce the image reconstruction task to find the universal tickets, and a two-stage training paradigm is proposed to obtain the desired ticket. First, an autoencoder structure is introduced for the existing model. The encoder structure inherits the original model structure and weights. Unlike an end-to-end unsupervised pre-training in He et al. (2021), which requires an extremely large model and high mask rate to avoid cheating model, a much smaller decoder is trained in our method for a specific encoder. We use the feature map hint method to accelerate the convergence of decoder. After the first stage of training for the decoder, we have a classification task with a classification head for the second stage of training. Concretely, we freeze the decoder and apply a modified LTH algorithm to get a universal ticket. Finally, the performance is evaluated by transferring the obtained tickets to different downstream tasks.

Our contributions can be summarized as follows.

- We propose a new framework for pruning pre-trained neural networks. Different from directly pruning on the classification tasks, we first train a decoder for the pruned network and then introduce the reconstruction loss. The pruned model is applicable for different downstream tasks, especially object detection and instance segmentation.
- Our result suggests that pixel-level tasks are better than traditional image-level tasks for pruning pre-trained neural networks. Although contrastive learning and classification tasks have been proved to be useful pretraining tasks for large models, the pruning method relying on those tasks may degenerate the transferability. By introducing an appropriate pixel-level task, a pruned model generalizes better on downstream tasks.
- The proposed method is evaluated on benchmark downstream tasks. It achieves 32.7%AP on the COCO dataset when only keeping about 20% parameters of the original model. The superior performance over state-of-the-art result Girish et al. (2021) confirms the effectiveness of our method.

2 Related Work

Pruning and Lottery Tickets Hypothesis Pruning aims to remove the unimportant weights of a neural network to reduce computation costs. It was first proposed in LeCun et al. (1990) where the authors use the Hessian matrix to estimate the importance of parameters. In Han et al. (2015), the authors propose iterative magnitude pruning to achieve a better compression rate. A lot of works follow Han et al. (2015) setting and achieve promising results. Different from those methods, the lottery tickets hypothesis, proposed in Frankle & Carbin (2018), suggests that a sparse trainable subnetwork exists in a given over-parameterized model. This sparse network can achieve similar performance as the entirety. To verify this assumption, Frankle & Carbin (2018) follow the iterative pruning paradigm but set the model parameter to initial values at each pruning round. Some works Lee et al. (2018); Wang et al. (2019); Tanaka et al. (2020) attempt to find the winning tickets in an initialized network. Those methods can find winning tickets in small datasets. However, as stated in Liu et al. (2018), the original LTH method fails with more complicated datasets and large learning rates. In Renda et al. (2019), the authors find that rewinding the parameter to the early training stage of the neural network, rather than the initial value, can bring profits to winning tickets in complicated datasets.

Large Scale Pretrain ImageNet pretraining is widely used in nowadays computer vision training pipeline. It is common sense that using a pretrained network on a large dataset can benefit downstream tasks, both in accuracy and training epochs. Nowadays, self-supervised pretrain methods have become more popular because they can utilize unlabeled data. Image-level self-supervised pretraining has been developed for years He et al. (2020); Chen et al. (2020); Grill et al. (2020); Qian et al. (2021). In those methods, an image is encoded into a single representation vector. The classifier should distinguish a strongly augmented image from irrelevant ones by using their representation vector. Recently, pixel-level or patch-level pretraining has attracted more attention. These methods focus on the recovery of corrupted images Ramesh et al. (2021); Touvron et al. (2021); He et al. (2021). Most of them require an extremely large model such as ViT Dosovitskiy et al. (2020) to achieve better performance.

Autoencoder is a famous traditional machine learning structure. It is widely used in image denoising Vincent et al. (2008) and generative model Kingma & Welling (2013). Recently, the image reconstruction task is also introduced in self-supervised pretraining He et al. (2021). To get abundant semantic information of neural networks and avoid cheating models, the autoencoder usually adopts a strong regularizer or data augmentation and should take a lot of time to train. In this paper, we focus on pruning to downstream tasks, rather than getting a better autoencoder. In this way, we separately train the decoder and encoder of our neural network. Thus, the decoder only needs to be trained for fewer epochs than in previous works.

3 Image-level Tasks are not Sufficient Criterion for Pruning Neural Network

In this section, we show the insufficiency of image-level tasks for finding universal tickets. It is common sense that traditional image classification tasks can produce an abundant feature map so that their backbone can transfer to every downstream task. Thus, we use the difference between the original feature map and the pruned version to imply the transferability of a pruned model. We mainly study two simple but important variants of CNN: linear convolutional neural network (LCNN) and one-layer ReLU convolutional neural network (ORCNN). We should point out that our proof methods can not directly generalize to the deep neural network so the proof of the deep neural network is still an open problem. We focus on pruning in LCNN and finetuning in ORCNN. Our results suggest that image-level tasks cannot produce a transferable pruned neural network. We mainly talk about the 1D convolution, but the 2D case is easy to extend.

3.1 Preliminary

Notations about Tensor A k-th order tensor $(a_{i_1 \dots i_k})$ is a k-dimensional array of real numbers $a_{i_1 \dots i_k}$. We use $||\cdot||$ and $\langle \cdot, \cdot \rangle$ to denote the standard l_2 norm and inner product of tensors. And we define the normalized l_2 distance between tensor $\mathbf{A}$ and $\mathbf{B}$ as

$$dist_{l_2}^N(\mathbf{A}, \mathbf{B}) = \frac{||\mathbf{A} - \mathbf{B}||}{\sqrt{||\mathbf{A}||||\mathbf{B}||}} \tag{1}$$

Convolution Operator is a linear operator represented by $*$. Let $\mathbf{x} = (x_{i,j}) \in \mathbb{R}^{c \times D}$ be the input, where D is the length of input sequence and c is the channel number of input. Let $\mathbf{W} \in \mathbb{R}^{c' \times c \times (2s+1)}$ be the convolution tensor. Then the convolution between $\mathbf{W}$ and $\mathbf{x}$ is defined as:

$$(\mathbf{W} * \mathbf{x})_{i,j} = \sum_{k=1}^{c} \sum_{l=-s}^{s} w_{ik,l} x_{k,j+l} \tag{2}$$

where we use circular padding method, i.e. $x_{i,j+D} := x_{i,j}$.

Average Pooling Operator is also a linear operator P_a. For any matrix $\mathbf{v} \in \mathbb{R}^{c \times D}$, we have:

$$P_a \mathbf{v} = \frac{1}{D} \sum_{i=1}^{D} \mathbf{v}_{:,i} \in \mathbb{R}^c \tag{3}$$

Linear Convolutional Neural Networks (LCNN): LCNN is a linear mapping $\mathcal{C} : \mathbb{R}^{c\times D} \to \mathbb{R}^{m_{L+1}\times D}$, which can be defined as

$$\mathcal{C}(x) \coloneqq W^L * W^{L-1} * ...W^0 * x \tag{4}$$

where $x \in \mathbb{R}^{c\times D}$,$W^l \in \mathbb{R}^{m_{l+1}\times m_l \times (2s+1)}$.

One-hidden-layer ReLU Convolutional Neural Networks (ORCNN): Let $\mathbf{x} = (x_{i,j}) \in \mathbb{R}^{c\times D}$ be the input. $\mathbf{W} \in \mathbb{R}^{m\times c\times (2s+1)}$ is the convolution tensor, where m is the channel number of feature maps. ORCNN is defined as the following:

$$\begin{aligned} \mathbf{x}^{conv} &= \frac{1}{\sqrt{m}}\sigma(\mathbf{W} * \mathbf{x}) \\ \mathbf{x}^{pool} &= P_a\mathbf{x}^{conv} \\ f(\mathbf{x}) &= \left\langle \mathbf{a}, \mathbf{x}^{pool} \right\rangle \end{aligned} \tag{5}$$

where $\mathbf{x}^{conv}$, $\mathbf{x}^{pool}$ are hidden-layer outputs and $f(\mathbf{x})$ is the predicted label. We use $f_{\mathbf{a},\mathbf{W}}(\mathbf{x})$ to denote the whole newtork. $\sigma(\cdot)$ denotes ReLU activation function which is defined as $\sigma(\cdot) := \max(\cdot, 0)$. $\mathbf{a} = (a_1, a_2, ..., a_m)^{\mathbf{T}} \in \mathbb{R}^m$ are the fully connected weights.

3.2 Pruning in LCNN

In this section, we investigate the pruning step in LCNN. We claim:

Claim Let $\mathcal{C}$ be an LCNN, there must exists another LCNN C', such that

$$P_a\mathcal{C}(x) = P_a\mathcal{C}'(x), \mathcal{C}(x) \neq \mathcal{C}'(x) \tag{6}$$

This is a direct result by considering the translation symmetry in LCNN. Now, the question becomes "Can this LCNN $\mathcal{C}'$ be found by pruning algorithm?". At least, can we find an LCNN $\mathcal{C}'$ by pruning, such that the change of $||P_a\mathcal{C}(x) - P_a\mathcal{C}'(x)||$ is small while $||\mathcal{C}(x) - \mathcal{C}'(x)||$ is large? To answer this question, we come out the following theorem:

Theorem 3.1. *For any random initialized LCNN, where parameter is initialized as i.i.d $\mathcal{N}(0, \Delta)$. Then, for any $p < 0.11$, we can prune p proportion of weights and get a new LCNN $\mathcal{C}'$ with high probability, such that:*

$$\frac{||P_a\mathcal{C}(x) - P_a\mathcal{C}'(x)||}{||P_a\mathcal{C}(x)||} < C_1 p^{3/2}$$

$$\frac{||\mathcal{C}(x) - \mathcal{C}'(x)||}{||\mathcal{C}(x)||} > C_2 p^{1/2}$$

Here C_1 and C_2 are constants related to the kernel size s and the depth L

This theorem means that if we initialize the LCNN properly, we can find some neurons such that removing those neurons does not change the image-level feature vector a lot but destroys the feature map structure. Thus, using a pruning criterion based on image-level loss can not preserve the feature map of LCNN. The detailed proof is in Appendix.

3.3 Finetuning in ORCNN

In this section, we focus on finetuning in ORCNN. Let $f_{\mathbf{a},\mathbf{W}}(\mathbf{x})$ denote the ORCNN parameterized by fully connected weight $\mathbf{a}$ and convolution tensor $\mathbf{W}$. The pruning pipeline can be formalized as the following three phases:

- **Pre-trained Phase:** We randomly initialize the parameters $\mathbf{a}$ and $\mathbf{W}$ to $\mathbf{a}_0$ and $\mathbf{W}_0$. Then, we train the model via image-level tasks on the given labeled dataset S and derive a pre-trained model $f_{\mathbf{a}_{pre},\mathbf{W}_{pre}}(\mathbf{x})$.
- **Pruning Phase:** We apply the structured pruning method to ORCNN with the pruning rate p.

- **Finetuning Phase:** We first reset unpruned parameters to initial values $\mathbf{a}_0$ and $\mathbf{W}_0$. Next, we finetune the network parameters on the same labeled dataset S via the gradient descent algorithm. Finally, we derive the finetuned model $f_{\mathbf{a}_{fin},\mathbf{W}_{fin}}(\mathbf{x})$.

We mainly consider the training process in the finetuning phase. In the finetuning phase, we use the same dataset $S = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_n}, y_n)\}$ as in the pretraining phase and use loss function $L(f) := \frac{1}{2}\sum_{i=1}^{n}(f(\mathbf{x_i}) - y_i)^2$. We use the gradient descent method to update the convolutional tensor $\mathbf{W}$ and freeze the fully connected weights $\mathbf{a}$:

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta \frac{\partial L}{\partial \mathbf{W}(t)} \tag{7}$$

where η is the learning rate, and t denotes t^{th}-iter.

As the original pre-trained model has a strong transferability to diverse downstream tasks, we believe the pre-trained model can learn a good representation of the data. Specifically, the pre-trained ORCNN can derive the feature maps from the image data by the pre-trained convolution tensor $\mathbf{W}_{pre}$. Thus, the 'difference' between convolution tensors $\mathbf{W}_{fin}$ and $\mathbf{W}_{pre}$ implies the transferability of pruned model.

In order to measure the difference between convolution tensors, we multiply an arbitrary rotation operator $\mathbf{Q}$ on the $\mathbf{W}_{fin}$ to recover its density. Then, we calculate the minimal normalized l_2 distance between $\mathbf{QW}_{fin}$ and $\mathbf{W}_{pre}$. The following Theorem 3.2 characterizes the lower bound of the distance under the over-parameterized setting.

Theorem 3.2. *Assume that we set the channel number of feature maps $m = \Omega(\frac{1}{\delta^2}poly(n))$, and the finetuning learning rate η is sufficiently small. After the finetuning phase, the finetuned convolution tensor is $\mathbf{W}_{fin}$. Then with probability at least $1 - \delta$ over the random initialization in the pre-trained phase, we have*

$$\min_{\mathbf{Q}\in\mathbb{O}}\{dist_{l_2}^{N}(\mathbf{QW}_{fin}, \mathbf{W}_{pre})\} \geq \frac{p}{2} \tag{8}$$

where $\mathbb{O}$ is rotation operator space and p is the pruning rate.

The main idea of the proof is to analyze the dynamics of the model Gram matrix in the gradient descent process. The detailed proof can be found in Appendix.

Theorem 3.2 suggests the lower bound of normalized l_2 distance between them is growing linearly with respect to the pruning rate p. It reveals the pruned model's ability to extract features is less than the original although it may have the same good performance as the original model in image-level tasks. Therefore, we demonstrate the insufficiency of image-level tasks for finding universal tickets.

4 METHOD

As we discussed in Section 3, only focusing on the image-level task during the pruning procedure will lead to a degenerated feature map. In general, using image-level loss as a pruning criterion tends to remove image details and destroy the structure of the feature map. That untraceable information will cause a significant accuracy drop on the detection or segmentation tasks. To mitigate this problem, we attempt to introduce the pixel-level task to the traditional pruning framework. Our framework can be formalized into three stages:

i) Train an autoencoder. We modify the pretrained model to the encoder of an autoencoder structure. The last feature map of the original model becomes the compressed code of the autoencoder; then, we freeze the encoder and start training. We use the feature map hint method (illustrated in Section 4.1) to accelerate the training process and improve performance.

ii) Prune the encoder. After decoder training, we prune the encoder part with reconstruction loss and classification loss simultaneously. During this pruning step, the decoder is frozen to keep the information gathered from the encoder. We follow a modified LTH pruning pipeline to get better performance.

iii) Adapt the encoder to the downstream tasks according to the standard transfer learning setting.

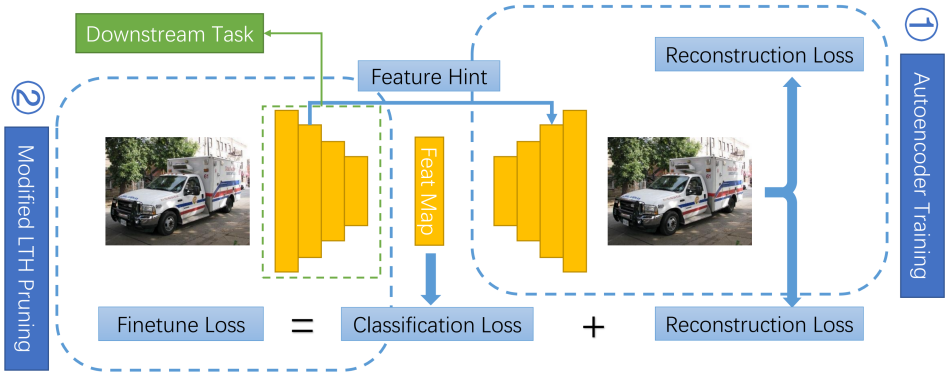

Figure 1: **Overview of our framework.** First, we use the reconstruction loss and the feature map hint method to train an autoencoder structure. Next, we use the modified LTH algorithm for pruning the neural network. We combine classification loss and image reconstruction loss for finetuning procedure. Then, we transfer the encoder part to the downstream tasks. The encoder part is frozen during the autoencoder training process, while the decoder is frozen in the Modified LTH pruning process.

The overall structure design is described in Figure 1. We will describe the first two steps in our framework in the following sections.

4.1 Autoencoder Training

Image reconstruction is a conventional computer vision task but was introduced as a pretraining method recentlyHe et al. (2021). Autoencoder is the basic architecture of image reconstruction. As the first step of our framework, the original model will be embedded in the autoencoder structure, which will be trained until the decoder captures the pixel-level information.

In this paper, we focus on ResNet structure, but our method can easily generalize to other kinds of structures. We remove the last pooling layer and fully connection layer of the original model as the encoder part. In this way, the final feature map is regarded as the compressed code of the autoencoder. Different from unsupervised pretraining, the decoder part in our method is an inversed ResNet. The training purpose of an autoencoder is to minimize

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{i=1}^{N} ||\mathcal{D}(\mathcal{F}(x_i)) - x_i||^2 \tag{9}$$

where N is the number of training samples, $\mathcal{F}$ represents the encoder part, $\mathcal{D}$ represents the decoder part, x_i is the input image. Obviously, without any constraint on $\mathcal{F}$ or $\mathcal{D}$, the loss function will lead to a trivial solution. Therefore, we freeze the parameters in F during the autoencoder training process.

4.2 Feature Map Hint

We use the previous stage's feature map as a hint to the inversed ResNet decoder for better image reconstruction results. Directly transporting the feature map to the decoder part must lead to a fault model. The decoder tends to rely on low-level features while ignoring the high-stage information. Therefore, we mix the original feature map and the following decoder's feature map with a specific proportion t. Formally, let f_i be the feature map in the encoder stage-i, g_i is the output feature in the decoder stage-i, and the input feature map of the decoder $i+1$ stage is:

$$g_i' = (1-t)g_i + tf_i \tag{10}$$

This mixing method can stabilize the finetuning process and avoid fault models. In practice, the proportion t will be a small positive number. In Section 6.2, we conduct an ablation study on the choice of different feature maps. The final result shows that using f_3 and f_4 as feature map hints for the decoder can achieve the best performance.

4.3 RECONSTRUCTION LOSS IN PRUNING STEP

With the autoencoder structure, we can add the reconstruction loss to the pruning process. The loss function during the pruning process is composed of two parts: L_{class} refers to the traditional classification loss, and L_{rec} refers to the reconstruction loss. We use a hyperparameter λ to balance $\mathcal{L}_{class}$ and $\mathcal{L}_{rec}$

$$\mathcal{L} = \mathcal{L}_{class}(\mathcal{F}) + \lambda \mathcal{L}_{rec}(\mathcal{F}, \mathcal{D}) \tag{11}$$

Notice that the decoder $\mathcal{D}$ is related to $\mathcal{L}_{rec}$ in the above function, which means the decoder will be trained during the finetuning process. However, we hope the decoder part guides the finetuning process and transfers pixel-level information to the encoder. The decoder change will perturb the information remaining in the decoder and may lead to an unexpected result. It also slows down the training. Therefore, we freeze the parameter in the decoder part during the pruning process.

4.4 MODIFIED LTH PRUNING

There are two widely used pruning pipelines, IMP and LTH, where LTH resets the parameter to the initial value, but IMP does not. In this section, We argue that neither setting is the most capable method to find universal tickets. We introduce a modified LTH pipeline, which is showed more powerful to find universal tickets.

In previous works, Chen et al. (2021) used an IMP method to produce universal tickets. Although the IMP method usually provides better accuracy on the upstream tasks, it may cause the neural network to fall in the minima of the upstream task while it is hard to finetune on the downstream task. In Girish et al. (2021), the authors examine whether the ImageNet tickets produced by LTH can work for detection tasks. We should point out that their method does not utilize the pretrained network power and limits the performance of their method. We try to combine the strengths of those two methods.

Inspired by Chen et al. (2021), we replace the initial values in the original LTH method with the pretrained values. Let $f(x;\theta)$ represent the neural network, $\theta \in \mathbb{R}^n$. Pruning some parameters means we permanently set some dimensions of θ to zero. In this way, we can use mask $m \in \{0,1\}^n$ to describe the pruned parameter. Thus, a pruned network can be described by $f(x; m \odot \theta)$, where $m_i = 0$ means we prune this parameter from the whole neural network. Let θ_{pre} represent the pretrained value of the original neural network. We train the network for T epochs and prune the smallest p proportion of unmasked parameters for every training round. After the prune step, the parameters θ reset to pretrained values θ_{pre}. Then another training round begins.

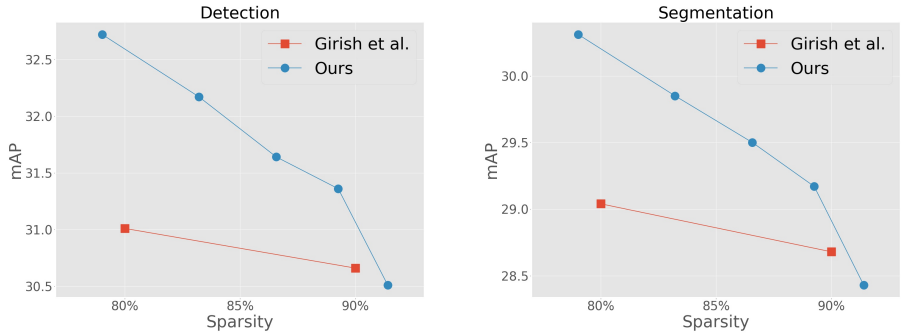

Figure 2: **Performance on the COCO detection and segmentation dataset**. It is noticeable that our method outperforms the reported result in Girish et al. (2021). We can produce transferable tickets at sparsity around 80%.

5 EXPERIMENT

We only post our experiment results in this section, the detailed experiment setting can be found in Appendix

5.1 Detection and Segmentation Results

We evaluate our method on the COCO dataset. In Figure 2, we compare our method with the reported ImageNet tickets results in Girish et al. (2021). Notice that our results surpass the baseline results at every point of the accuracy-sparsity curve. Our method can achieve 32.7 mAP on the detection task and 30.3 mAP on the segmentation task at sparsity 79.03%. To make a fair comparison with the reported result in Girish et al. (2021), we interpolate our accuracy-sparsity curve at 80% and 90% sparsity. The result is listed in Table 1. At high-level sparsity, e.g., 92% sparsity, our method's performance goes down due to the limitation of ResNet50 itself. As stated in Girish et al. (2021), it is hard for ResNet50 to find winning tickets at 90% sparsity, even in the setting of directly applying LTH on the downstream task.

5.2 Classification Results

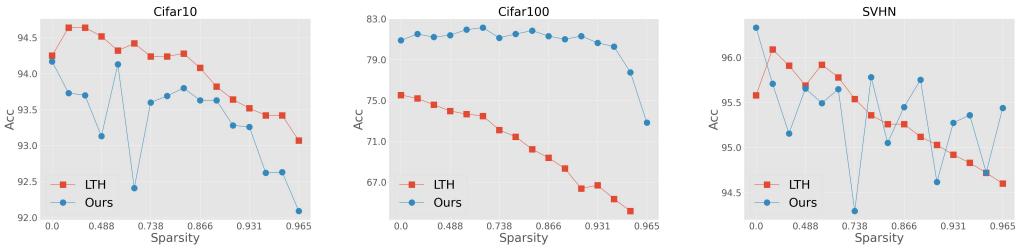

Figure 3: **Classification transfer results.** From left to right are the results of Cifar10, Cifar100, SVHN. In Cifar10 and SVHN datasets, our method is comparable to the directly applying LTH method. At Cifar100, our method outperforms with LTH method, showing the transferability of our selected tickets.

Baseline setting We compare our results with the baseline that directly performs LTH on the target dataset in classification task transfer. We perform 15 pruning rounds. Each pruning round has 182 epochs and will cut 20% parameters. The learning rate starts from 0.1 in each round, $\times 0.1$ at 91, 136. The weight decay is 2e-4. This setting is the same as the task setting in Chen et al. (2021).

We evaluate our tickets on Cifar10, Cifar100, SVHN. The results are presented in Figure 3. Our method is comparable or even better than directly applying the LTH method on the target dataset, especially in Cifar100 datasets. In Cifar100, our method achieves over 80% accuracy, while the LTH method only achieves 76% at the beginning of training.

Although we are mainly concerned about the pruned network's transferability, our method still achieves comparable ImageNet unstructured pruning results. As shown in Figure 5, our method has a similar performance with the iterative magnitude unstructured pruning result, which implies that our tickets are also winning tickets on the ImageNet.

Table 1: Object Detection and Segmentation Results. (mAP)

Task	Detection		Segmentation	
Sparsity	80%	90%	80%	90%
Girish et al. (2021)	31.0	30.7	29.0	28.6
ours (interpolated)	32.6	31.1	30.2	28.9

6 Ablation Study

6.1 Comparison vs IMP Method

As we argued in Section 4.4, the IMP method is not the best way to create universal tickets. In this section, we compare the transferability of tickets produced by IMP and our method. We evaluate those tickets on detection downstream tasks at different sparsity. It is easy for the IMP method to get a more sparse network in complicated datasets. However, we find out that in the transferability

Table 2: Comparison with IMP Method. Object Detection and Segmentation Results. (mAP)

Sparsity (backbone)	Detection		Segmentation	
	IMP	Ours	IMP	Ours
79.02%	32.1	32.7	29.9	30.3
83.22%	31.8	32.3	29.6	29.8
86.57%	31.3	31.6	29.1	29.5
89.26%	30.8	31.4	28.7	29.2
91.41%	30.1	30.5	28.0	28.4

tasks, our method brings significant AP improvement in the downstream detection and segmentation datasets. The results are shown in Table 2

6.2 Feature Map Hint Selection

In this section, we investigate the influence of feature map hints. As Figure 4 shows, different feature map hints lead to a different result. We attribute this phenomenon to the different feature maps' detailed and semantic information levels. If we only use a high-level feature for autoencoder, it takes a lot of network capacity and needs more epochs to converge; if we use a low-level feature map, those features are so close to the original information that the decoder part does not get useful information. We find that using f_3, f_4 as feature map hints can lead to the best performance compared to other settings. So we use f_3, f_4 as the feature map hints choice on other settings.

7 Conclusion

In this paper, we propose a new framework to find the universal tickets that can transfer to diverse downstream datasets. First, we theoretically show that the image-level tasks may result in a degenerated feature map with high probability. This analysis implies that image-level tasks are not sufficient for neural network pruning. To address the problem, we introduce a new pruning framework that includes the image reconstruction task to guide the pruning process. Our framework has three steps: i) Train an autoencoder, ii) Prune the encoder with an improved LTH method. iii) Transfer the encoder to downstream tasks. Besides, the feature map hint method is developed to accelerate the autoencoder training stage. The obtained tickets are evaluated on diverse downstream tasks at different sparsity ratios. The empirical study demonstrates that the proposed method can outperform state-of-the-art method Chen et al. (2021); Girish et al. (2021) on benchmark datasets.

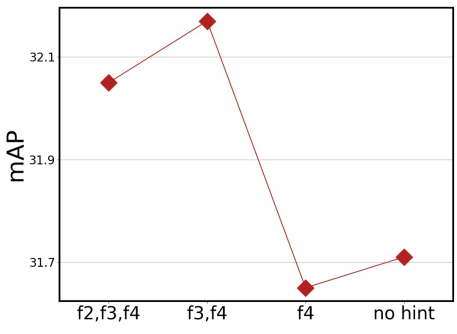

Figure 4: **The ablation study of different feature map hints**. We use the detection transfer result at pruning round 8 (sparsity 83.22%) as the selection criterion. We observe that using f_3 and f_4 as feature map hints can induce the best result on detection datasets.

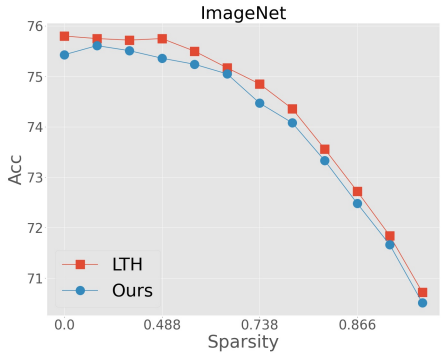

Figure 5: **The comparison between our method and IMP on ImageNet**. We find that our method has a similar performance as the IMP result. Considering that our method only fine-tunes 10 epochs for each ticket, we believe that our ticket is also winning tickets on ImageNet.

REFERENCES

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16306–16316, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685. PMLR, 09–15 Jun 2019.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

Sharath Girish, Shishira R Maiya, Kamal Gupta, Hao Chen, Larry S Davis, and Abhinav Shrivastava. The lottery ticket hypothesis for object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 762–771, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*, 28, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pp. 598–605, 1990.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2018.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

Ari Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *Advances in Neural Information Processing Systems*, 32:4932–4942, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Qi Qian, Yuanhong Xu, Juhua Hu, Hao Li, and Rong Jin. Unsupervised visual representation learning by online constrained k-means. *CoRR*, abs/2105.11527, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2019.

Victor Sanh, Thomas Wolf, and Alexander M Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *NeurIPS*, 2020.

Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, and E Weinan. Convolutional neural networks with low-rank regularization. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2019.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Yang Yang, Yaxiong Yuan, Avraam Chatzimichailidis, Ruud JG van Sloun, Lei Lei, and Symeon Chatzinotas. Proxsgd: Training structured neural networks under regularization and constraints. In *International Conference on Learning Representations*, 2019.

A EXPERIMENT SETTING

Dataset For the autoencoder training step and modified LTH pruning step, we conduct all experiments on ImageNet Deng et al. (2009). For the image-level transfer task, we still focus on image classification. We evaluate the tickets gathered from the ImageNet dataset on Cifar10 Krizhevsky et al. (2009), Cifar100Krizhevsky et al. (2009), and SVHN Netzer et al. (2011). We also show the accuracy of ImageNet. For pixel/patch-level task transfer, we investigate object detection and instance segmentation. As stated in Girish et al. (2021), ImageNet tickets transfer to the COCO Lin et al. (2014) dataset is harder than transfer to small detection dataset such as VOC datasets. Therefore, We use the COCO dataset as a standard benchmark.

Model We evaluate our method with ResNet50, a standard CNN model on the ImageNet and COCO object detection/instance segmentation task backbone. We use official PyTorch pretrained weights as our pretrained values. The decoder is an inverse ResNet architecture. This decoder part is removed in the downstream tasks, and only the encoder part transfer. For classification transfer tasks, we adjust the first kernel size of ResNet50 to 3×3 and remove the first max-pooling layer. We use a famous structure mask RCNNHe et al. (2017) as the segmentation and detection head for object detection and instance segmentation tasks.

Training and Pruning Setting In the autoencoder training step, we train the decoder with the AdamW optimizer. We use a multi-step learning rate schedule with an initial learning rate 1e-4 and $\times 0.1$ at the 10, 30 epoch. The total training epoch is 50; the batch size is 512; weight decay is 2e-4. In the modified LTH pruning step, We follow the pruning setting in Chen et al. (2021), where for each pruning round, we prune 20% parameters. Each pruning round has 10 epochs. We use the SGD optimizer, and the learning rate is kept 3e-4, the batch size is 512, momentum is 0.9. The reconstruction penalty λ defined in (9) is 10, the feature map hint proportion is 0.1. For classification task transfer setting, we follow the setting in Chen et al. (2021). We draw the accuracy-compression rate curve for different classification tasks. For segmentation and detection tasks, we use a standard Detectron2Wu et al. (2019) FPN $1\times$ training config. We mainly evaluate the performance at 7,8,9,10,11 pruning round (79.03%,83.22%,86.58%,89.26%,91.41%) to make a comparison with result in Girish et al. (2021). The numbers in the brackets represent the sparsity at each pruning round. All the experiment are conducted on 8 V100 GPUs.

B PROOF OF THEOREM 3.1

We use Discrete Fourier Transformation to proof Theorem 3.1

Discrete Fourier Transformation (DFT) of a vector sequence $x \in \mathbb{R}^{c\times N}$is defined as:

$$\mathcal{F}(x_{i,:})(k) = \tilde{x}_i^k = \frac{1}{N}\sum_s x_{i,s} e^{\frac{2\pi J}{N} sk} \tag{12}$$

We use J represent imaginary unit to distinguish from footnote i. Here we assume $x_{i,:}$ follow the Periodic Boundary Conditions. DFT has two important propositions:

Proposition B.1. *Let $\tilde{x}_i^k = \mathcal{F}(x_{i,:})(k)$, then we have:*
(1) $x_{i,s} = \sum_k \tilde{x}_i^k e^{\frac{2\pi J}{N} sk}$
(2) $[W * x]_{i,j} = \sum_{t,k} \tilde{W}(k)_{it}\tilde{x}_t^k e^{\frac{2\pi J}{N} jk}$

We define $[\tilde{W}(k)]_{ij} = \sum_s W_{ij,s} e^{\frac{2\pi J}{N} sk}$ for simplicity. By those propositions, $\mathcal{C}(x)$ can be written as a simple version:

$$\mathcal{C}(x)_{:,p} = \sum_k W^L(k)W^{L-1}(k)...W^0(k)\tilde{x}_:^k e^{\frac{2\pi J}{N} pk} \tag{13}$$

Because the avgpool operation only associate with the $k = 0$ component, while other part is independent with it. It's easy to find another network such the zero component unchanged will other component are different. Thus we can proof the claim we stated in Section 3.2

claim Let $\mathcal{C}$ be a LCNN, there must exists another LCNN C', such that

$$P_a\mathcal{C}(x) = P_a\mathcal{C}'(x), \mathcal{C}(x) \neq \mathcal{C}'(x) \tag{14}$$

To prove Theorem 3.1 we should notice that in such a LCNN $\mathcal{C}$, we can find some weights (around $\mathcal{O}(\frac{\epsilon}{\Delta})$) such that $|\sum_s W_{ij,s}| < \epsilon$. By the assumption that each weights is initialized independent, the high frequency component of W is in $\mathcal{O}(1)$ while the zero frequency component is $\mathcal{O}(\epsilon)$. Therefore, the total influence of $||P_a(\mathcal{C}(x))||^2$ is $\mathcal{O}(\frac{\epsilon^3}{\Delta^3})$ but the influence of $||\mathcal{C}(x)||^2$ is $\mathcal{O}(\frac{\epsilon}{\Delta})$

Lemma B.2. *For a randomly initialized convolutional tensor, $W_{ij,s} \sim i.i.d\mathcal{N}(0, \Delta)$, we have:*

(1) $\mathbb{E}(\sum_i \sum_{k=-s}^{s} W_{ij,k} \sum_{n=-s}^{s} W_{im,n}) = (2s+1)n\Delta^2\delta_{jm}$
(2) $p = \mathbb{P}(\sum_{k=-s}^{s} W_{ij,k} < \epsilon) < \sqrt{\frac{1}{2\pi(2s+1)}}\frac{\epsilon}{\Delta}$

If we prune every neuron with $|\sum_s W_{ij,s}| < \epsilon$ and let the new tensor is $W'_{ij,s}$, we have $W'_{ij,s} = m_{ij}W_{ij,s}$
(3) $\mathbb{P}(m_{ij} = 0) = p$
(4) $\mathbb{E}(\sum_t m_{ij}W_{ij,t} \sum_k W_{ij,k}) > \Delta^2(1 - \frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3})$
(5) $\mathbb{E}(\sum_t m_{ij}W_{ij,t} \sum_k m_{ij}W_{ij,k}) = \mathbb{E}(\sum_t m_{ij}W_{ij,t}, \sum_k W_{ij,k})$
(6) $\mathbb{E}(\sum_t m_{ij}W_{ij,t} \sin tk\theta \sum_m W_{ij,m} \sin tk\theta) = (1-p)\sum_t \sin^2 tk\theta$
(7) $\mathbb{E}(\sum_t m_{ij}W_{ij,t} \cos tk\theta \sum_m W_{ij,m} \cos tk\theta) < (1-p)\sum_t \cos^2 tk\theta$
in (6)(7) $\theta = \frac{2\pi}{D}$*, and* $k \in \{1, 2, ..., D-1\}$

(1)(2)(3)(5) are easy to understand, for(4), we have:

$$\mathbb{E}(\sum_t m_{ij}W_{ij,t}, \sum_k W_{ij,k}) \tag{15}$$

$$=\mathbb{E}(\sum_t W_{ij,t} \sum_k W_{ij,k} \big| m_{ij} = 1)(1-p) + \mathbb{E}(m \sum_t W_{ij,t} \sum_k W_{ij,k} \big| m_{ij} = 0)p \tag{16}$$

$$=\Delta^2 - \int_{-\epsilon}^{\epsilon} \frac{1}{\sqrt{2\pi}\Delta} x^2 e^{-\frac{x^2}{2\pi\Delta}} dx > \Delta^2(1 - \frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3}) \tag{17}$$

for (6) this result is a direcly result by considering the conditional expectation:

$$\mathbb{E}(\sum_t W_{ij,t} \sin tk\theta \sum_k W_{ij,k} \sin tk\theta \big| |\sum_t W_{ij,t}| > \epsilon) \tag{18}$$

Due to $\sum_t 1 \cdot \sin tk\theta = 0$, this expectation is independent of conditional variable. Thus we have:

$$\mathbb{E}(\sum_t m_{ij}W_{ij,t} \sin tk\theta \sum_t W_{ij,t} \sin tk\theta) \tag{19}$$

$$=\mathbb{E}(\sum_t m_{ij}W_{ij,t} \sin tk\theta \sum_k W_{ij,k} \sin tk\theta \big| m_{ij} = 1)(1-p) \tag{20}$$

$$+ \mathbb{E}(\sum_t m_{ij}W_{ij,t} \sin tk\theta \sum_t W_{ij,t} \sin tk\theta \big| m_{ij} = 0)p \tag{21}$$

$$=(1-p)\sum_t (\sin tk\theta)^2 \tag{22}$$

For (7), although the expectation term is dependent on the conditional term, we knew that this condition will enlarge the expectation, so we have:

$$\mathbb{E}(\sum_t m_{ij}W_{ij,t} \cos tk\theta \sum_t W_{ij,t} \cos tk\theta) \tag{23}$$

$$=\mathbb{E}(\sum_t m_{ij}W_{ij,t} \cos tk\theta \sum_k W_{ij,k} \cos tk\theta \big| m_{ij} = 1)(1-p) \tag{24}$$

$$+ \mathbb{E}(\sum_t m_{ij}W_{ij,t} \cos tk\theta \sum_t W_{ij,t} \sin tk\theta \big| m_{ij} = 0)p \tag{25}$$

$$<(1-p)\sum_t (\cos tk\theta)^2 \tag{26}$$

Now, we can calculate the difference caused by prune the weights:

Lemma B.3. *For any random initialized LCNN, where parameter is i.i.d initialized as $\mathcal{N}(0, \Delta)$. If we prune every neuron with $|\sum_s W^l_{ij,s}| < \epsilon$ and get a pruned LCNN $\mathcal{C}'$, then we have*

$$\frac{||P_a\mathcal{C}(x) - P_a\mathcal{C}(x)||^2}{||P_a\mathcal{C}(x)||^2} \sim \mathcal{O}(\frac{\epsilon^3}{\Delta^3})$$

proof According to (13), we can rewrite $P_a\mathcal{C}(x)$ as:

$$P_a\mathcal{C} = W^L(0)W^{L-1}(0)...W^0(0)P_a(x) \tag{27}$$

for simplicity, we write $F^l(k) = W^l(k)W^{l-1}(k)...W^0(k)\mathcal{F}(x)(k)$. Thus,

$$||P_a\mathcal{C}(x) - P_a\mathcal{C}'(x)||^2 = ||F^L(0) - F'^L(0)||^2 \tag{28}$$

We define $\mathbb{E}_l$ as the expectation about the l-th layer weights, then we have:

$$\begin{aligned}
&\mathbb{E}||P_a\mathcal{C}(x) - P_a\mathcal{C}'(x)||^2 \\
=&\mathbb{E}\mathbb{E}_L||F^L(0) - F'^L(0)||^2 \\
=&\mathbb{E}\mathbb{E}_L \left\langle F^L(0), F^L(0)\right\rangle + \mathbb{E}\mathbb{E}_L \left\langle F'^L(0), F'^L(0)\right\rangle - 2\mathbb{E}\mathbb{E}_L \left\langle F^L(0), F'^L(0)\right\rangle \\
=&\mathbb{E}\mathbb{E}_L \left\langle W^L(0)F^{L-1}(0), W^L(0)F^{L-1}(0)\right\rangle \\
&+ \mathbb{E}\mathbb{E}_L \left\langle W'^L(0)F'^{L-1}(0), W'^L(0)F'^{L-1}(0)\right\rangle - \\
&2\mathbb{E}\mathbb{E}_L \left\langle W^L(0)F^{L-1}(0), W'^L(0)F'^{L-1}(0)\right\rangle \\
<&m_L(2s+1)\Delta^2 \left\langle F^{L-1}(0), F^{L-1}(0)\right\rangle \\
&+ m_L(2s+1)\Delta^2(1 - \frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3}) \left\langle F'^{L-1}(0), F'^{L-1}(0)\right\rangle \\
&- 2m_L(2s+1)\Delta^2(1 - \frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3}) \left\langle F^{L-1}(0), F'^{L-1}(0)\right\rangle \\
<&m_Lm_{L-1}...m_1(2s+1)^L\Delta^{2L}[1 - (1 - \frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3})^L] \left\langle P_a(x), P_a(x)\right\rangle \\
<&m_Lm_{L-1}...m_1(2s+1)^L\Delta^{2L}L\frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3} \left\langle P_a(x), P_a(x)\right\rangle \\
=&L\frac{2\epsilon^3}{3\sqrt{2\pi}\Delta^3}||P_a\mathcal{C}(x)||^2
\end{aligned} \tag{29}$$

By using the concentretion inequality, we proof Lemma B.3

Lemma B.4. *For any random initialized LCNN, where parameter is i.i.d initialized as $\mathcal{N}(0, \Delta)$. If we prune every neuron with $|\sum_s W^l_{ij,s}| < \epsilon$ and get a pruned LCNN $\mathcal{C}'$, then we have*

$$\frac{||\mathcal{C}(x) - \mathcal{C}(x)||^2}{||\mathcal{C}(x)||^2} \sim \mathcal{O}(\frac{\epsilon}{\Delta})$$

proof First we have

$$||\mathcal{C}(x) - \mathcal{C}(x)||^2 = \sum_k ||\mathcal{F}(\mathcal{C}(x))(k) - \mathcal{F}(\mathcal{C}'(x))(k)||^2$$

We caculate $k \neq 0$ term, due to $k = 0$ is calculated in B.3 and it is $\mathcal{O}(\frac{\epsilon}{\Delta})$

$$
\begin{aligned}
&\sum_{k\neq 0} \mathbb{E}||\mathcal{F}(\mathcal{C}(x))(k) - \mathcal{F}(\mathcal{C}'(x))(k)||^2 \\
=&\sum_{k\neq 0} \mathbb{E}\mathbb{E}_L||F^L(k) - F'^L(k)||^2 \\
=&\sum_{k\neq 0} \mathbb{E}\mathbb{E}_L \left\langle W^L(k)F^{L-1}(k), W^L(k)F^{L-1}(k)\right\rangle - \mathbb{E}\mathbb{E}_L \left\langle W^L(k)F^{L-1}(k), W'^L(k)F'^{L-1}(0)\right\rangle \\
>&\sum_{k\neq 0} \mathbb{E}(2s+1)m_L\Delta^2\mathbb{E}\left\langle F^{L-1}(k), F^{L-1}(k)\right\rangle - \\
&\mathbb{E}m_L\Delta^2\mathbb{E}\left\langle F^{L-1}(k), F^{L-1}(k)\right\rangle (1-p)(\sum_t \cos^2 tk\theta + \sum_t \sin^2 tk\theta) \\
=&\sum_{k\neq 0} m_L m_{L-1}....m_1(2s+1)^L\Delta^{2L}\mathbb{E}\left\langle \mathcal{F}(x)(k), \mathcal{F}(x)(k)\right\rangle (1-(1-p)^L) \\
=&(1-(1-p)^L)\sum_{k\neq 0} \mathbb{E}||\mathcal{F}(\mathcal{C}(x))(k)||^2
\end{aligned}
\tag{30}
$$

Thus we have:

$$
\begin{aligned}
&\sum_{k} \mathbb{E}||\mathcal{F}(\mathcal{C}(x))(k) - \mathcal{F}(\mathcal{C}'(x))(k)||^2 \\
=&(1-(1-p)^L)\sum_{k} \mathbb{E}||\mathcal{F}(\mathcal{C}(x))(k)||^2 = \mathcal{O}(\frac{\epsilon}{\Delta})||\mathcal{C}(x)||^2
\end{aligned}
\tag{31}
$$

By combining Lemma B.3 and Lemma B.4, we proof Theorem 3.1

C Proof of Theorem 3.2

In order to prove Theorem 3.2, we need more notations to represent the problem setup.

First, we introduce the convolution expansion operator $\phi_.(\cdot)$ that can simplify convolution operator to inner product of tensors.

Convolution Tensor In convolutional neural network(CNN), the convolution tensor is described as follows. Let $\mathbf{w}_{ij} = (w_{ij,-s}, w_{ij,-s+1}, ..., w_{ij,s})^{\mathbf{T}} \in \mathbb{R}^{2s+1}(1 \leq i \leq c', 1 \leq j \leq c)$ be convolution kernel, where c' is the number of filters and $2s+1$ is the size of convolution kernel. And $\mathbf{W}_i = (\mathbf{w}_{i1}, \mathbf{w}_{i2}, ..., \mathbf{w}_{ic})^{\mathbf{T}} \in \mathbb{R}^{c\times(2s+1)}$ denotes i_{th} filter. Then the convolution tensor is defined as a 3-dimensional tensor $\mathbf{W} = (\mathbf{W}_i)_{1\leq i\leq c'} \in \mathbb{R}^{c'\times c\times(2s+1)}$.

Convolution Expansion Operator Let $\mathbf{x} = (x_{i,j}) \in \mathbb{R}^{c\times D}$ be the 1-dimensional input, where D is the length of input sequence and c is the channel number of input. Let $\mathbf{W} \in \mathbb{R}^{c'\times c\times(2s+1)}$ be the convolution tensor. Then we define $\phi_k(\mathbf{x})$ as

$$
\phi_k(\mathbf{x}) = \begin{pmatrix} x_{1,k-s}, & \dots & , x_{1,k+s} \\ \dots, & \dots, & \dots \\ x_{c,k-s}, & \dots, & x_{c,k+s} \end{pmatrix} \tag{32}
$$

Recall the definition of convolution operator, we have

$$
(\mathbf{W} * \mathbf{x})_{r,k} = \langle \mathbf{W}_r, \phi_k(\mathbf{x})\rangle \tag{33}
$$

Because we use structured pruning method to prune model, we can assume the new channel number of feature maps is $M = mq = m(1-p)$, where p is our pruning rate. And we assume mq is an integer in order to simplify the proof. Now, we can represent the pruned ORCNN as

$$
f(\mathbf{x}) = \frac{\sqrt{q}}{\sqrt{M}D}\sum_{r=1}^{M} a_r \sum_{k=1}^{D} \sigma(\langle \mathbf{W}_r, \phi_k(\mathbf{x})\rangle) \tag{34}
$$

To analyze the gradient descent process, inspired by Du et al. (2019), we focus on the Gram matrix $\mathbf{G} = (\mathbf{G}_{i,j})$ of ORCNN, which is defined as

$$\mathbf{G}_{i,j} = \frac{q}{MD^2} \sum_{r=1}^{M} \sum_{k=1}^{D} \sum_{l=1}^{D} \langle \phi_k(\mathbf{x}_i), \phi_l(\mathbf{x}_j) \rangle \mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i) \rangle \geq 0, \langle \mathbf{W}_r, \phi_l(\mathbf{x}_j) \rangle \geq 0\} \tag{35}$$

And if we consider the case that M goes to infinity, we can represent the expectation of $\mathbf{G}$ and we know $\mathbf{G}$ converges to it by central limit theorem and independence of $\mathbf{W}_r (1 \leq r \leq M)$. We use $\mathbf{G}^{\infty}$ to denote the expectation, which can be formally defined as

$$\mathbf{G}^{\infty}_{i,j} = \mathbb{E}_{\mathbf{W} \sim \mathbf{N}(\mathbf{0}, \mathbf{I})} \left[\frac{q}{D^2} \sum_{k=1}^{D} \sum_{l=1}^{D} \langle \phi_k(\mathbf{x}_i), \phi_l(\mathbf{x}_j) \rangle \mathbb{I}\{\langle \mathbf{W}, \phi_k(\mathbf{x}_i) \rangle \geq 0, \langle \mathbf{W}, \phi_l(\mathbf{x}_j) \rangle \geq 0\} \right] \tag{36}$$

Proposition C.1. *We use $\lambda_{min}(\cdot)$ to denote the least eigenvalue of a matrix, then we assume that $\mathbf{G}^{\infty}$ has the positive least eigenvalue $\lambda_0 = \lambda_{min}(\mathbf{G}^{\infty})$. In fact, we know $\mathbf{G}^{\infty}$ is positive definite. Thus, the assumption is weak and reasonable.*

We use $\mathbf{G}_0$ to denote the initial Gram matrix of pruned model. By the standard concentration analysis of independent variables, we can derive the following Lemma C.2, which means the initial Gram matrix of model also has the bounded positive least eigenvalue with high probability.

Lemma C.2. *When the channel number of feature maps in pruned model $M = \Omega\left(\frac{n^2}{\lambda_0^2} \log\left(\frac{n}{\delta}\right)\right)$ and initialization in pre-trained phase satisfies the standard Guassian distribution, with probability at least $1 - \delta$, we have*

$$\lambda_{min}(\mathbf{G}_0) \geq \frac{3}{4}\lambda_0 \tag{37}$$

Before analyzing the finetuning process, we assume some settings as follows to simplify the process in order to focus on the change of convolution tensor.

Proposition C.3. *In finetuning phase, we randomly initialize the fully connected weight $\mathbf{a} \sim Unif(\{-1, 1\}^M)$. And we normalize the input such that $||\phi_k(\mathbf{x}_i)|| = 1$ for any k, i. In fact, without the normalization assumption we can also proof the similar conclusion, but it will dependent with $\frac{\max_{k,i}\{||\phi_k(\mathbf{x}_i)||\}}{\min_{k,i}\{||\phi_k(\mathbf{x}_i)||\}}$, which is also an constant.*

Because the fully connected weight $\mathbf{a}$ is fixed, we can represent the model output w.r.t. the given dataset S as

$$F_i(\mathbf{W}(t)) = \frac{\sqrt{q}}{\sqrt{M}D} \sum_{r=1}^{M} a_r \sum_{k=1}^{D} \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i) \rangle) \tag{38}$$

which is output of i^{th} sample in dataset. And we use $\mathbf{F}(\mathbf{W}(t)) = (F_1(\mathbf{W}(t)), F_2(\mathbf{W}(t)), \cdots, F_n(\mathbf{W}(t)))^{\mathbf{T}}$ to denote the output vector, where t means it is t^{th}-round.

Proposition C.4. *Recall the initialization of the pruned convolution tensor is the same as that in pre-trained phase, we know each unpruned entry of the pruned convolution tensor satisfies indenpendent standard Guassian distribution $\mathbf{N}(0, 1)$. With the knowledge of sub-exponential distribution, we can prove that w.h.p. the l_2 norm of the pruned convolution tensor$\mathbf{W}(0)$ is close to $\sqrt{qmD(2s+1)}$ and the l_2 norm of the original convolution tensor before pre-trained phase is close to $\sqrt{mD(2s+1)}$.*

Next, we introduce Lemma C.5 that describes the dynamic process in finetuning phase.

Lemma C.5. *Under the above assumptions, if the channel number of feature maps in pruned model M is $\Omega\left(\frac{n^6}{\lambda_0^4 \delta^3}\right)$ and learning rate η is $O\left(\frac{\lambda_0}{n^2}\right)$, with probability at least $1 - \delta$, we have*

$$||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 \leq \left(1 - \frac{\eta\lambda_0}{2}\right)^t ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||^2 \tag{39}$$

where $\mathbf{y} = (y_1, y_2, \cdots, y_n)^{\mathbf{T}}$ is the label vector of dataset S.

By Lemma C.5, with the standard analysis of gradient descent, we can prove the following Lemma C.6

Lemma C.6. *Under the above assumptions, if the channel number of feature maps in pruned model M is $\Omega\left(\frac{n^6}{\lambda_0^4\delta^3}\right)$ and learning rate η is $O\left(\frac{\lambda_0}{n^2}\right)$, with probability at least $1-\delta$, for some constant C, we have*

$$||\mathbf{W}_{fin} - \mathbf{W}(0)|| \leq \frac{C\sqrt{q}n}{\lambda_0} \tag{40}$$

Remark of Lemmma C.6 In fact, we have similar conclusion of Lemmma C.6 in pre-trained phase, which implies $||\mathbf{W}_{pre}|| = \sqrt{m}(1+o(1))$ and can be proven by the same method as follows.

Proof of Theorem 3.2 By $m = \Omega(\frac{n^6}{\lambda_0^4})$, Proposition C.4 and Lemma C.6, with probability $1-\delta$, we have $||\mathbf{W}_{fin} - \mathbf{W}(0)|| = o(||\mathbf{W}(0)||)$ and $||\mathbf{W}_{fin}|| = \sqrt{qm}(1+o(1))$. Combined with the property of rotation operator, for any rotation operator $\mathbf{Q}$, we have $dist_{l_2}^N(\mathbf{Q}\mathbf{W}_{fin}, \mathbf{W}_{pre}) \geq \frac{||\mathbf{W}_{pre}||-||\mathbf{Q}\mathbf{W}_{fin}||}{\sqrt{||\mathbf{W}_{pre}||||\mathbf{Q}\mathbf{W}_{fin}||}} = (1-p)^{-\frac{1}{4}} - (1-p)^{\frac{1}{4}} \geq \frac{p}{2}$. Thus, we prove Theorem 3.2 .

Next, We will use induction method to prove Lemma C.5 and Lemma C.6 . Our inductive hypothesis is the inequality is true for $0, 1, \cdots, t$ and we want to prove the inequality is also true for $t+1$. First, we have the following decomposition of l_2 loss.

$$\begin{aligned}
||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{y}||^2 &= ||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t)) + \mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 \\
&= ||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))||^2 \\
&\quad + 2\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}}\left(\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))\right) + ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2
\end{aligned} \tag{41}$$

The third term can be bounded by the inductive hypothesis. Thus, we only need to bound the first and the second terms.

The following Lemma C.7 can help us to bound the gradient norm in finetuing phase.

Lemma C.7. *In finetuing phase, we can control the upper bound of gradient norm as*

$$\left|\left|\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}_r}\right|\right| \leq \frac{\sqrt{qn}}{\sqrt{M}}||\mathbf{F}(\mathbf{W}) - \mathbf{y}|| \tag{42}$$

for any $r \in [M]$.

Proof of Lemma C.7 The proof is very standard to analysis gradient descent.

$$\begin{aligned}
\left|\left|\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}_r}\right|\right| &= \left|\left|\frac{\sqrt{q}}{\sqrt{M}D}\sum_{i=1}^n (F_i(\mathbf{W}) - y_i)\sum_{k=1}^D \phi_k(\mathbf{x}_i)\mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0\}\right|\right| \\
&\leq \frac{\sqrt{q}}{\sqrt{M}D}||\mathbf{F}(\mathbf{W}) - \mathbf{y}||\left|\left|\left(\sum_{k=1}^D \phi_k(\mathbf{x}_i)\mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0\}\right)_{1\leq i\leq n}\right|\right| \\
&= \frac{\sqrt{q}}{\sqrt{M}D}||\mathbf{F}(\mathbf{W}) - \mathbf{y}||\sqrt{\sum_{i=1}^n ||\sum_{k=1}^D \phi_k(\mathbf{x}_i)\mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0\}||^2} \\
&\leq \frac{\sqrt{q}}{\sqrt{M}D}||\mathbf{F}(\mathbf{W}) - \mathbf{y}||\sqrt{\sum_{i=1}^n \left(\sum_{k=1}^D ||\phi_k(\mathbf{x}_i)\mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0\}||\right)^2} \\
&\leq \frac{\sqrt{q}}{\sqrt{M}D}||\mathbf{F}(\mathbf{W}) - \mathbf{y}||\sqrt{nD^2} \\
&= \frac{\sqrt{qn}}{\sqrt{M}}||\mathbf{F}(\mathbf{W}) - \mathbf{y}||
\end{aligned} \tag{43}$$

Now, we can derive an upper bound of the first term by Lemma C.7 and 1-Lipschitz property of ReLU funciton.

$$
\begin{aligned}
&||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))||^2 \\
&= \sum_{i=1}^{n} |F_i(\mathbf{W}(t+1)) - F_i(\mathbf{W}(t))|^2 \\
&= \sum_{i=1}^{n} \left| \frac{\sqrt{q}}{\sqrt{M}D} \sum_{r=1}^{M} a_r \sum_{k=1}^{D} (\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)) \right|^2 \\
&= \sum_{i=1}^{n} \left| \frac{\sqrt{q}}{\sqrt{M}D} \sum_{r=1}^{M} a_r \sum_{k=1}^{D} \left(\sigma\left(\left\langle \mathbf{W}_r(t) - \eta \frac{\partial L(\mathbf{W}(\mathbf{t}))}{\partial \mathbf{W}(\mathbf{t})_r}, \phi_k(\mathbf{x}_i) \right\rangle \right) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle) \right) \right|^2 \\
&\leq \frac{q}{D} \sum_{i=1}^{n} \sum_{r=1}^{M} \sum_{k=1}^{D} \left| \left\langle \eta \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)}, \phi_k(\mathbf{x}_i) \right\rangle \right|^2 \\
&\leq \frac{q\eta^2}{D} \sum_{i=1}^{n} \sum_{r=1}^{M} \sum_{k=1}^{D} \left|\left| \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)} \right|\right|^2 \leq \frac{q\eta^2}{D} nMD \frac{qn}{M} ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 \\
&= q^2\eta^2 n^2 ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2
\end{aligned}
\tag{44}
$$

To bound the second term, we need a more refined analysis of its form. The following Lemma C.8 will inspire us how to deal with the term.

Lemma C.8. *Under the inductive hypothesis, for any $T \in [t+1]$ and $rin[M]$, we have*

$$
||\mathbf{W}_r(T) - \mathbf{W}_r(0)|| \leq \frac{4\sqrt{qn}}{\sqrt{M}\lambda_0} ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}|| \tag{45}
$$

Proof of Lemma C.8 By the gradient descent and inductive hypothesis, we have

$$
\begin{aligned}
||\mathbf{W}_r(T) - \mathbf{W}_r(0)|| &\leq \sum_{j=0}^{T-1} ||\mathbf{W}_r(j+1) - \mathbf{W}_r(j)|| = \eta \sum_{j=0}^{T-1} \left|\left| \frac{\partial L(\mathbf{W}(j))}{\partial \mathbf{W}_r(j)} \right|\right| \\
&\leq \eta \sum_{j=0}^{T-1} \frac{\sqrt{qn}}{\sqrt{M}} ||\mathbf{F}(\mathbf{W}(j)) - \mathbf{y}|| \\
&\leq \eta \sum_{j=0}^{T-1} \frac{\sqrt{qn}}{\sqrt{M}} \left(1 - \frac{\eta\lambda_0}{2}\right)^{\frac{j}{2}} ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}|| \\
&\leq \eta \frac{\sqrt{qn}}{\sqrt{M}} ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}|| \sum_{j=0}^{\infty} \left(1 - \frac{\eta\lambda_0}{2}\right)^{\frac{j}{2}} \\
&\leq \eta \frac{\sqrt{qn}}{\sqrt{M}} ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}|| \sum_{j=0}^{\infty} \left(1 - \frac{\eta\lambda_0}{4}\right)^{j} \\
&\leq \frac{4\sqrt{qn}}{\sqrt{M}\lambda_0} ||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||
\end{aligned}
\tag{46}
$$

Notice we have

$$
F_i(\mathbf{W}(t+1)) - F_i(\mathbf{W}(t)) = \frac{\sqrt{q}}{\sqrt{M}D} \sum_{r=1}^{M} a_r \sum_{k=1}^{D} (\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)) \tag{47}
$$

And Lemma C.8 implies the distance between $\mathbf{W}_r(T)$ and $\mathbf{W}_r(0)$ is bounded, we can use truncation estimation method to deal with $F_i(\mathbf{W}(t+1)) - F_i(\mathbf{W}(t))$. Specifically, we use $U_{r,k,i}(R)$ to denote the event $\langle \mathbf{W}_r(0), \phi_k(\mathbf{x}_i)\rangle < R$.

Because $\phi_k(\mathbf{x}_i)$ is normalized, we know $\langle \mathbf{W}_r(0), \phi_k(\mathbf{x}_i)\rangle$ is a standard Guassian Variable. We have the following Lemma C.9 .

Lemma C.9. *For any $r\ in[M], k \in [D]$ and $i \in [n]$, we have the probability of the $U_{r,k,i}(R)$ satisfies*

$$\mathbb{P}\{U_{r,k,i}(R)\} \leq \frac{2R}{\sqrt{2\pi}} \tag{48}$$

If the event $U_{r,k,i}(R)$ is false and $\frac{4\sqrt{qn}}{\sqrt{M}\lambda_0}||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}|| < R$, then we know $\mathbb{I}\{\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle \geq 0\} = \mathbb{I}\{\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle \geq 0\} = \mathbb{I}\{\langle \mathbf{W}_r(0), \phi_k(\mathbf{x}_i)\rangle \geq 0\}$ by Lemma C.8 . So we use $J_{true}(R)$ to denote the set $\{(r,k,i)|U_{r,k,i}(R) \text{ is true}\}$ and $J_{false}(R)$ to denote the set $\{(r,k,i)|U_{r,k,i}(R) \text{ is false}\}$.

By Markov's inequality, we can derive the following Lemma C.10

Lemma C.10. *With probability at least $1-\delta$, we have*

$$|J_{true}(R)| \leq \frac{nMDR}{\sqrt{2\pi}\delta} \tag{49}$$

If we use $\mathbf{e}_i (1 \leq i \leq n)$ to denote the standard basis of $\mathbb{R}^n$, then we can decompose $\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))$ as

$$\begin{aligned}
&\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t)) \\
&= \sum_{i=1}^{n} \mathbf{e}_i \frac{\sqrt{q}}{\sqrt{M}D} \sum_{r=1}^{M} a_r \sum_{k=1}^{D} \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \\
&= \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{true}(R)} a_r \mathbf{e}_i \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \\
&\quad + \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{false}(R)} a_r \mathbf{e}_i \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right)
\end{aligned} \tag{50}$$

We use I_1, I_2 to denote two terms indexed by $J_{true}(R), J_{false}(R)$. And the term indexed by $J_{true}(R)$ can be bounded by the cardinal of $J_{true}(R)$ as

$$\begin{aligned}
||I_1|| &= \left|\left| \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{true}(R)} a_r \mathbf{e}_i \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \right|\right| \\
&= \frac{\sqrt{q}}{\sqrt{M}D} \sqrt{\sum_{i=1}^{n} \left| \sum_{r,k|(r,k,i)\in J_{true}(R)} a_r \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \right|^2} \\
&= \frac{\sqrt{q}}{\sqrt{M}D} \sqrt{\sum_{i=1}^{n} \left| \sum_{r,k|(r,k,i)\in J_{true}(R)} a_r \left(\sigma\left(\left\langle \mathbf{W}_r(t) - \eta \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)}, \phi_k(\mathbf{x}_i)\right\rangle\right) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \right|^2} \\
&\leq \frac{\sqrt{q}}{\sqrt{M}D} \sqrt{\sum_{i=1}^{n} \left(\sum_{r,k|(r,k,i)\in J_{true}(R)} \eta \left|\left| \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)} \right|\right| \right)^2} \\
&\leq \frac{\sqrt{q}|J_{true}(R)|}{\sqrt{M}D} \eta \left|\left| \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)} \right|\right| \\
&\leq \frac{\sqrt{q}|J_{true}(R)|}{\sqrt{M}D} \eta \frac{\sqrt{qn}}{\sqrt{M}} ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}|| \\
&= \frac{\eta q\sqrt{n}|J_{true}(R)|}{MD} ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||
\end{aligned} \tag{51}$$

Next, we focus on I_2 and establish relation between it and the Gram matrix.

$$\begin{aligned}
I_2 =& \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{false}(R)} a_r \mathbf{e}_i \left(\sigma(\langle \mathbf{W}_r(t+1), \phi_k(\mathbf{x}_i)\rangle) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \\
=& \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{false}(R)} a_r \mathbf{e}_i \left(\sigma\left(\left\langle \mathbf{W}_r(t) - \eta \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)}, \phi_k(\mathbf{x}_i)\right\rangle\right) - \sigma(\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle)\right) \\
=& \frac{\sqrt{q}}{\sqrt{M}D} \sum_{(r,k,i)\in J_{false}(R)} a_r \mathbf{e}_i \left(-\eta \frac{\partial L(\mathbf{W}(t))}{\partial \mathbf{W}_r(t)}\right) \mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0\} \\
=& -\eta \frac{q}{MD^2} \sum_{(r,k,i)\in J_{false}(R)} \sum_{j=1}^{n} \sum_{l=1}^{D} \mathbf{e}_i (F_j(\mathbf{W}(t)) - y_j) \langle \phi_k(\mathbf{x}_i), \phi_l(\mathbf{x}_j)\rangle \\
& \mathbb{I}\{\langle \mathbf{W}_r, \phi_k(\mathbf{x}_i)\rangle \geq 0, \langle \mathbf{W}_r, \phi_l(\mathbf{x}_j)\rangle \geq 0\} \\
=& -\eta \sum_{i=1}^{n} \mathbf{e}_i \sum_{j=1}^{n} \tilde{\mathbf{G}}_{i,j}(t)(F_j(\mathbf{W}(t)) - y_j) \\
=& -\eta \tilde{\mathbf{G}}(t)(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y})
\end{aligned} \tag{52}$$

where $\tilde{\mathbf{G}}(t)$ is defined as

$$\tilde{\mathbf{G}}_{i,j}(t) = \frac{q}{MD^2} \sum_{r,k,l|(r,k,i)\in J_{false}(R)} \langle \phi_k(\mathbf{x}_i), \phi_l(\mathbf{x}_j)\rangle \mathbb{I}\{\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle \geq 0, \langle \mathbf{W}_r(t), \phi_l(\mathbf{x}_j)\rangle \geq 0\} \tag{53}$$

By matrix perturbation technique, we can prove the following Lemma C.11

Lemma C.11. *If the channel number M is $\Omega\left(\frac{q^3 n^6}{\delta^2 {\lambda_0}^4}\right)$, then with probability at least $1-\delta$, we have*

$$\lambda_{min}(\mathbf{G}(t)) \geq \frac{\lambda_0}{2} \tag{54}$$

Proof of Lemma C.11 First, by $M = \Omega\left(\frac{q^3 n^6}{\delta^2 {\lambda_0}^4}\right)$ and Lemma C.8, if we set R' to $\frac{4\sqrt{qn}}{\sqrt{M}\lambda_0}||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||$, then we have

$$||\mathbf{W}_r(t) - \mathbf{t}_r(0)|| \leq R' \tag{55}$$

for any $r \in [M]$.

Next, we establish relation between $\mathbf{G}(t)$ and $\mathbf{G}(0)$. We consider the difference of each entry of them as

$$\begin{aligned}
& \mathbb{E}|\mathbf{G}_{i,j}(t) - \mathbf{G}_{i,j}(0)| \\
=& \frac{q}{MD^2} \sum_{r=1}^{M} \sum_{k=1}^{D} \sum_{l=1}^{D} |\langle \phi_k(\mathbf{x}_i), \phi_l(\mathbf{x}_j)\rangle (\mathbb{I}\{\langle \mathbf{W}_r(t), \phi_k(\mathbf{x}_i)\rangle \geq 0, \langle \mathbf{W}_r(t), \phi_l(\mathbf{x}_j)\rangle \geq 0\} \\
& - \mathbb{I}\{\langle \mathbf{W}_r(0), \phi_k(\mathbf{x}_i)\rangle \geq 0, \langle \mathbf{W}_r(0), \phi_l(\mathbf{x}_j)\rangle \geq 0\})| \\
\leq& \frac{q}{MD^2} \sum_{r=1}^{M} \sum_{k=1}^{D} \sum_{l=1}^{D} \mathbb{P}\{U_{r,k,i}(R') \cup U_{r,L,i}(R')\} \\
\leq& \frac{4qR'}{\sqrt{2\pi}}
\end{aligned} \tag{56}$$

By the Markov's inequality, with probability at least $1-\delta$, we have $||\mathbf{G}(t) - \mathbf{G}(0)||_{l_1} \leq \frac{4qn^2R'}{\sqrt{2\pi}\delta}$. And by the positive definite of $\mathbf{G}(t), \mathbf{G}(0)$we know

$$\lambda_{max}(\mathbf{G}(t) - \mathbf{G}(0)) \leq ||\mathbf{G}(t) - \mathbf{G}(0)||_F \leq ||\mathbf{G}(t) - \mathbf{G}(0)||_{l_1} \leq \frac{4qn^2R'}{\sqrt{2\pi}\delta} \tag{57}$$

Thus, we have

$$\begin{aligned}\lambda_{min}(\mathbf{G}(t)) &\geq \lambda_{min}(\mathbf{G}(0)) - \lambda_{max}(\mathbf{G}(t) - \mathbf{G}(0)) \\ &\geq \frac{3}{4}\lambda_0 - \frac{4qn^2R'}{\sqrt{2\pi}\delta} \geq \frac{\lambda_0}{2}\end{aligned} \tag{58}$$

Under conclusion of Lemma C.11, by the similar technique used by proof of Lemma C.11 and Lemma C.10, we can prove

Lemma C.12. *If the channel number M is $\Omega\left(\frac{q^3n^6}{\delta^2\lambda_0{}^4}\right)$, then with probability at least $1-\delta$, we have*

$$\lambda_{min}(\tilde{\mathbf{G}}(t)) \geq \frac{\lambda_0}{2} - \frac{qn^2R}{\sqrt{2\pi}\delta} \tag{59}$$

Thus, by Lemma C.12, we have

$$\begin{aligned}\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}} I_2 &= -\eta \left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}} \tilde{\mathbf{G}}(t)(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}) \\ &\leq \left(-\frac{\eta\lambda_0}{2} + \frac{\eta qn^2R}{\sqrt{2\pi}\delta}\right)||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2\end{aligned} \tag{60}$$

With the upper bound of the cardinal of $J_{true}(R)$(Lemma C.10), we also have

$$\begin{aligned}\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}} I_1 &\leq ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||||I_1|| \\ &\leq ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||\frac{\eta q\sqrt{n}|J_{true}(R)|}{MD}||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}|| \\ &\leq \frac{q\eta n^{\frac{3}{2}}R}{\sqrt{2\pi}\delta}||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2\end{aligned} \tag{61}$$

Based on the estimation of upper bound of three terms, we have

$$\begin{aligned}||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{y}||^2 &= ||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))||^2 \\ &\quad + 2\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}}\left(\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))\right) + ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 \\ &= ||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 + ||\mathbf{F}(\mathbf{W}(t+1)) - \mathbf{F}(\mathbf{W}(t))||^2 \\ &\quad + 2\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}} I_1 + 2\left(\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}\right)^{\mathbf{T}} I_2 \\ &\leq \left(1 + q^2\eta^2n^2 - \eta\lambda_0 + \frac{2\eta qn^2R}{\sqrt{2\pi}\delta} + \frac{2q\eta n^{\frac{3}{2}}R}{\sqrt{2\pi}\delta}\right)||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2 \\ &\leq \left(1 - \frac{\eta\lambda_0}{2}\right)||\mathbf{F}(\mathbf{W}(t)) - \mathbf{y}||^2\end{aligned} \tag{62}$$

Finally, we only need to select the value of R. To make all the lemmas are true, we can set R to $\frac{\sqrt{2\pi}\delta\lambda_0}{16qn^2}$. Now, by combining the inductive hypothesis, we prove Lemma C.5.

Next, we prove Lemma C.6 by the expansion of Lemma C.8. In fact, Lemma C.5 is true for any t, so Lemma C.8 is true for any T. We use $\mathbf{W}_{fin,i}(1 \leq i \leq m)$ to denote the finetuned convolution filters and have $\mathbf{W}_{fin} = (\mathbf{W}_{fin,i})_{1\leq i\leq m}$. Notice some filters have been pruned, by Lemma C.8, then we have

$$\begin{aligned}||\mathbf{W}_{fin} - \mathbf{W}(0)||^2 &= \sum_{r=1}^{M}||\mathbf{W}_{fin,r} - \mathbf{W}_r(0)||^2 \\ &\leq \sum_{r=1}^{M}\left(\frac{4\sqrt{qn}}{\sqrt{M}\lambda_0}||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||\right)^2 \\ &= \frac{16qn}{\lambda_0^2}||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||^2\end{aligned} \tag{63}$$

By the concentration of random initialization, with high probability, $||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||$ is $O(\sqrt{n})$. So there exists an constant C such that $||\mathbf{F}(\mathbf{W}(0)) - \mathbf{y}||^2 \leq Cn$, which means we prove Lemma C.6 .