On the Diminishing Returns of Width for Continual Learning

Anonymous Author
Anonymous Institution

Abstract

While deep neural networks have demonstrated groundbreaking performance in various settings, these models often suffer from *catastrophic forgetting* when trained on new tasks in sequence. Several works have empirically demonstrated that increasing the width of a neural network leads to a decrease in catastrophic forgetting but have yet to characterize the exact relationship between width and continual learning. We prove that width is directly related to forgetting in Feed-Forward Networks (FFN), demonstrating the diminishing returns of increasing widths to reduce forgetting. We empirically verify our claims at widths hitherto unexplored in prior studies where the diminishing returns are clearly observed as predicted by our theory.

1 Introduction

Deep Neural Networks (DNNs) have achieved breakthrough performance in numerous challenging computational tasks that serve as a proxy for intelligence (LeCun et al., 2015). An essential and practical question is whether the same neural network can continuously learn over a series of tasks and perform well over all of them. Practically, using a well-trained neural network to achieve similar performance over a series of tasks is essential for reducing expensive retraining and computational costs (Diethe et al., 2019) and for mimicking the human-like ability to continually update its knowledge (Hadsell et al., 2020; Kudithipudi et al., 2022; Parisi et al., 2019). In practice, DNNs exhibit *catastrophic forgetting* when trained over a series of tasks, experiencing a sharp drop in performance on the previously learned tasks.

Preliminary work. Under review by AISTATS 2024. Do not distribute.

However, preventing catastrophic forgetting is theoretically and empirically tricky in many situations (Knoblauch et al., 2020; Kim et al., 2022). Many empirical studies of Continual Learning (CL) have observed that a model's hidden dimension or width is positively correlated with the ability to continually learn (Mirzadeh et al., 2022b,a; Ramasesh et al., 2021). Moreover, catastrophic forgetting is more easily mitigated in the infinite-width or Neural Tangent Kernel regime (Bennani and Sugiyama, 2020; Doan et al., 2021; Chizat et al., 2019; Geiger et al., 2020). However, the exact relationship between width and continual learning still needs to be clarified in the more practical finite-width setting. Theoretically, explaining this relationship requires understanding the effects of training and retraining a model on different datasets, which is a complex and difficult-to-analyze process. Through pure experimentation, most works observe a roughly linear relationship between width and continual learning, such as in small FFNs (Mirzadeh et al., 2022a) or in large CNNs and ResNets (Ramasesh et al., 2021; Mirzadeh et al., 2022b), suggesting that increasing the scale of the models is a simple method for improving continual learning.

In this paper, we investigate the relationship between width and continual learning more explicitly. We have observed empirically and theoretically that *simply increasing a model's width suffers diminishing returns in improving continual learning.* In fact, we provably demonstrate this connection between width and continual learning in Feed-Forward Networks of arbitrary depth and nonlinear activations. To circumvent the typical analytical difficulties of continual learning, we use the well-observed empirical observation that wider models move less from initialization during training. For completeness, we empirically verify this relationship on Feed-Forward Networks trained with either Stochastic Gradient Descent (SGD) or Adam (Kingma and Ba, 2015) optimizers. This lazy training phenomenon has often been observed empirically in the literature (Zou et al., 2020; Nagarajan and Kolter, 2019; Li and Liang, 2018; Neyshabur et al., 2018b; Chizat et al., 2019; Ghorbani et al., 2019). Using this observation alongside traditional perturbation analy-

sis, we demonstrate that width acts as a functional regularizer, preventing models trained on subsequent tasks from being too functionally different from previous models. Specifically, our new guarantees formalize this relationship between width and continual learning for finite-width models with nonlinear activations and variable depth, which do not exist in the literature for wide models.

Moreover, we empirically observe these diminishing returns. Many existing works have tested hidden dimensions up to 2048 where the diminishing returns are not immediately obvious. However, using the new training software BOLT (Meisburger et al., 2023), we can train Feed Forward Networks where width in isolation is increased far past this 2048 width. In particular, we measure the continual learning capabilities of FFNs as the hidden dimension is increased to 2^{20} where models have approximately a billion parameters, much larger than previously explored in the literature to our knowledge. With these new expansive experiments, we see this relationship between width and continual learning predicted by our theory. These validate these findings over standard continual learning datasets, including Rotated MNIST and Split CIFAR100.

Our results contribute to the extensive literature examining the relationship between neural network architectures and continual learning performance. In particular, we demonstrate both theoretically and empirically that scaling width alone is insufficient for mitigating the effects of catastrophic forgetting, providing a more nuanced understanding of finite-width forgetting dynamics than results achieved in prior studies (Mirzadeh et al., 2022a; Ramasesh et al., 2021).

As side effects of our analysis, our theoretical analysis suggests that employing row-wise sparsity can improve continual learning. We corroborate this finding empirically and demonstrate that using this sparsity in tandem with width can alleviate these diminishing returns. This result formalizes the intuition of employing task-based sparsity over the rows often employed in the literature (Serra et al., 2018). Moreover, our analysis predicts that the continual learning error scales roughly linearly over the number of tasks trained on. Our experiments again corroborate this linear increase in continual learning error over the tasks. To our knowledge, this is the first work to provably demonstrate the effects of number of tasks and row-wise sparsity on continual learning error.

Contributions In summary, we make the following contributions in our work.

1. Under this observation, we provably demonstrate that the training of nonlinear, variable depth feed-forward networks incurs continual learning error on the order of $\mathcal{O}\left(tW^{-\beta}\alpha^{\frac{1-2\beta}{2}}\right)$ where t is the number of tasks the model has been trained on, W is the width, α is the sparsity percentage, and β is a data-dependent real value. To our knowledge, this is the first work formalizing the connection between width and continual learning for nonlinear models of variable depth.

2. By testing at hidden dimensions not seen previously, we empirically see the diminishing returns of continual learning when increasing width. These experiments hold across many different width Feed Forward Networks on datasets such as Rotated MNIST and Split CIFAR100

3. We formalize the effects of row-wise sparsity and number of tasks on the continual learning error in our theoretical analysis. Our experiments corroborate this analysis.

2 Related Works

2.1 Continual Learning

We review several relevant works in the Continual Learning literature. The original works discussing Continual Learning and Catastrophic Forgetting phenomenon are Ring (1997), McCloskey and Cohen (1989) and Thrun and Mitchell (1995). Perhaps most relevant to this work are Mirzadeh et al. (2022b) and Mirzadeh et al. (2022a), which note the positive correlation between the width of models and continual learning. However, their experiments are limited to small widths, at a maximum of 2048 hidden dimension, which is small relative to the current Deep Learning state of the art. Moreover, their analysis is limited, only proving the continual learning in the setting of two-layer linear networks without nonlinear activation. Ramasesh et al. (2021) and Yoon et al. (2018) discuss how scale broadly empirically affects continual learning but does not provide any theoretical analysis nor focus specifically on width.

Bennani and Sugiyama (2020) and Doan et al. (2021) discuss theoretical frameworks for continually learning models in the infinitely wide or NTK regime. Several works have used explicit functional regularization as a way to mitigate catastrophic forgetting (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2018; Farajtabar et al., 2020; Khan and Swaroop, 2021; Dhawan et al., 2023). Peng et al. (2023) provides a robust theoretical framework, their Ideal Continual Learner framework, which attempts to develop a solid theoretical framework for understanding different continual learning methods. Mirzadeh et al. (2020) discuss the geo-

metric similarities between the different minima found in a continual learning regime

2.2 Wide Networks

Here, we review works relevant to understanding wide networks in the literature and different architectures. Nguyen et al. (2020) discusses the effects of width and depth on the learned representation of the model. Arora et al. (2019) discusses the empirical benefits of using the infinite-width Neural Tangent Kernel on different classification tasks. Lee et al. (2019) discuss the training dynamics of wide neural networks under gradient descent. Lu et al. (2017) discuss the expressiveness of wide neural networks and how wide neural networks can express certain functions better than deep ones. Jacot et al. (2018), Allen-Zhu et al. (2019), and Du et al. (2019) capture the training dynamics of infinitely-wide neural networks under the Neural Tangent Kernel Regime

3 Preliminary

3.1 Notation

Let our model $\mathbf{M}_t$ yielded after training on the tth task be denoted as

$$\mathbf{M}_t(x) = \mathbf{A}_{t,L}\phi_{L-1}(\mathbf{A}_{t,L-1}\phi_{L-2}(\dots \mathbf{A}_{t,2}\phi_1(\mathbf{A}_{t,1}x))).$$

Here, x is an input of dimensionality d_t. ϕ_i is the activation function for the ith layer of L_i Lipschitz-Smoothness. Moreover let W be the width of $\mathbf{M}$ such that the input layer $\mathbf{A}_{t,1} \in \mathbb{R}^{d_t \times W}$, last layer $\mathbf{A}_{t,L} \in \mathbb{R}^{W \times K_t}$, and all the middle layers $\mathbf{A}_{t,l} \in \mathbb{R}^{W \times W}$. Here, K_t is the dimensionality of the output of the tth task. We will often index a matrix by a set of rows. For example, if $\mathcal{S}$ is a set of row indeces, $\mathbf{A}_{t,l}[\mathcal{S}]$ denotes a matrix in $\mathbb{R}^{|\mathcal{S}| \times W}$ that contains the ith row from $\mathbf{A}_{t,l}$ if $i \in \mathcal{S}$.

3.2 Problem Setup

Here, we will formalize the problem setup of Continual Learning. Formally, say we have T training datasets $\mathcal{D}_1, \dots, \mathcal{D}_T$. The goal of continual learning is to design a model $\mathbf{M}$ such that it performs well on all datasets $\mathcal{D}_t$ for $t \in [T]$. We will describe the tth task as a supervised learning classification task where the dataset for the tth task is $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$. Here, $\mathcal{X}_t$ contains n_t datapoints of dimension d_t and $\mathcal{Y}_t$ contains n_t labels of dimension K_t. We wish to form $\mathbf{M}$ by sequentially training it on each dataset in increasing order from $\mathcal{D}_1$ to $\mathcal{D}_t$. We will call $\mathbf{M}_t$ the model outputted after training on the tth dataset. After retraining a model on the new dataset, we want the new model to remember its behavior on the previous dataset. Namely, we wish to reduce the continual learning error $\epsilon_{t,t'}$ where $t \leq t'$ and

$$\max_{x \in \mathcal{D}_t} \|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2 \leq \epsilon_{t,t'}.$$

Here, after training a model for $t' - t$ new datasets, we hope to ensure that the outputs between the original model on the tth dataset and the new model on the t'th dataset are similar. Given $\mathbf{M}_t$ is trained to completion on dataset $\mathcal{D}_t$ and achieves low error, if our new model $\mathbf{M}_{t'}$ is close to $\mathbf{M}_t$ on all inputs from $\mathcal{D}_t$, it will also perform well on $\mathcal{D}_t$. Therefore, the main question is the correlation between W, the width of the model $\mathbf{M}$, and the continual learning error $\epsilon_{t,t'}$.

3.3 Training Setup

We will examine the setup where our model $\mathbf{M}$ is a Feed Forward Network with some nonlinear activation. We randomly initialize every layer in the entire model to train the model on the first task $\mathbf{M}_1$. Before training, as is often done in Continual Learning, we will choose a subset of rows to be active. For every row with probability α, we will select that row to be active. Otherwise, it will be inactive. Only the active rows will be used for computation during training and inference. Inactive rows will not change during training. We will denote $\mathcal{A}_{t,l}$ as the set of rows active for task t at layer l. We use this setup to capture the connections between row-wise sparsity and continual learning empirically observed in continual learning literature (Serra et al., 2018). Setting α to 1 will recover fully dense training. We then train using Adam or SGD till convergence. To train on a subsequent task, we replace the input and output layers to match the dimensionality of the new tasks. Moreover, we choose which rows will be active for this new task for the intermediate layers. We then retrain till convergence with Adam or SGD. We repeat this training procedure iteratively for every task. For inference on the tth task, we take the intermediate layers learned at the final task and replace the input and output layers with the input and output layers trained for the tth task to match the dimensionality of the data from the tth task. We only use the active rows during training for the tth task.

4 Theoretical Analysis

We develop the theoretical connection between the width of the intermediate layers W and the error of continual learning $\epsilon_{t,t'}$ for different task indices t and t'. We begin by stating our final theorem and then provide a brief proof sketch. We mention a complete proof in Appendix C.1.

4.1 Main Theorem

Here, we present our full theorem, Theorem 4.1

Theorem 4.1. *(Informal) Say we generate a series of models $\mathbf{M}_1, \ldots, \mathbf{M}_T$ by training sequentially on datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$ according to Section 3.3. Let $\lambda^l_{i,j} = \frac{\|\mathbf{A}_{l,j}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}$ denote the ratio of the spectral norms of the weights of different row indices for different tasks. Moreover, let $\bar{\lambda} = \max_{l\in[L],i,j\in[T]} \lambda^l_{i,j}$. For all input vectors from the tth dataset $\forall x \in \mathcal{D}_t$, the ℓ_2 norm of the difference of the outputs from models $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ such that $t' \geq t$ are upper bounded [1] by*

$$\mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\right] = \mathcal{O}\left((t'-t)L2^L\bar{\lambda}\chi\left(\prod_{l=1}^{L} L_l\|\mathbf{A}_{t,l}\|_2\right)\gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}\right).$$

Here, χ denotes the maximum norm of the input in $\mathcal{D}_t$, i.e. $\chi = \max_{x\in\mathcal{D}_t}\|x\|_2$. Here, γ, β are data-dependent positive real values.

As the width of the layers increases, we see that the error $\epsilon_{t,t'}$ is decreased on the order of $W^{-\beta}$. This formalization furthermore formalizes the connection between width and the continual learning error $\epsilon_{t,t'}$. This theorem demonstrates that the continual learning error will decrease slowly as the width increases. As side effects of our analysis, we also make explicit how this error scales over tasks. As the number of datasets trained between $\mathcal{D}_{t'}$ and $\mathcal{D}_t$ increases, the continual learning error increases linearly. This linear increase in error over tasks is, to our knowledge, unique to this analysis. Moreover, we have an exponential dependence on L, denoting that increasing depth decreases continual learning ability. This analysis corroborates some empirical evidence, but our analysis is likely loose in its dependence on depth (Mirzadeh et al., 2022b). Moreover, as the sparsity coefficient is decreased, meaning the chance of a row being active is decreased, the continual learning ability is increased as long as $\frac{1-2\beta}{2}$ is positive, which is often the case in practice. Thus, this suggests a tradeoff between continual learning ability and accuracy, as increasing sparsity will increase continual learning ability but decrease the capacity of the model to learn on a given task. We note that there is some dependence on the layers' matrix norms, which may correlate with the layers' width, but this is usually initialization dependent (Nagarajan and Kolter, 2019). We can change the dependence on the weight norms using the noise stability property of neural networks as observed in Arora et al. (2018) (for more details, please see Section 4.3). Overall, this theorem makes the relationship between several model parameters such as width and continual learning, concrete and makes the diminishing returns more explicit.

[1] We can reduce the dependence on weight norms by using noise stability properties. For more details, please see Section 4.3.

4.2 Proof Sketch

Here, we present a brief proof sketch of Theorem 4.1 and the intuition behind it. We will begin this proof by finding the continual learning error between subsequentially trained models $\mathbf{M}_t$ and $\mathbf{M}_{t+1}$, i.e. $\epsilon_{t,t+1}$, and then scale the analysis to work over more tasks $t' \geq t+1$. To analyze $\epsilon_{t,t+1}$, we split the proof into three parts: (1) finding how many active rows are shared between layers in the first and second model at the same position, (2) finding how far these active rows can change during training, and (3) combining the two parts using perturbation analysis.

4.2.1 Number of Shared Active Rows

From Section 3.3, each row in a layer $\mathbf{A}_{t,l}$ is active with probability α and inactive with probability $1-\alpha$. Moreover, the activity of the rows are independent and identical random variables. Therefore, over the randomness of row selection, the expected number of shared active rows between two consecutive models is shown in Lemma 4.1.

Lemma 4.1. *For any two sequential task indices t and $t+1$ and layer l, the expected size of the intersection between the sets of active rows $\mathcal{A}_{t,l}$ and $\mathcal{A}_{t+1,l}$ is*

$$\mathbb{E}(|\mathcal{A}_{t,l} \cap \mathcal{A}_{t+1,l}|) = \alpha^2 W.$$

4.2.2 Distance between active rows after training

For rows that are active for both datasets $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$, we need to bound their distance. Here, we use a critical empirical observation. As observed in Zou et al. (2020); Nagarajan and Kolter (2019); Li and Liang (2018); Neyshabur et al. (2018b); Chizat et al. (2019), as the width of the neural network was increased, the distance from initialization was observed to decrease[2]. Intuitively, as the width of the neural network increases, the implicit regularization of standard gradient-based learning methods finds models closer to the initialization. While it is challenging to provide theoretical analyses of the connection between width, implicit regularization, and the distance from initialization save for restricted settings (Li and Liang, 2018), empirically, this is a well-observed phenomenon.

[2] Similar intuition on a "lazy training regime" was presented in Mirzadeh et al. (2022a). Still, they did not offer any formal analysis based on this observation.

For the sake of our analysis, we take this as an assumption in Assumption 4.1 and build our analysis on top of it.

Assumption 4.1. *Let t be any task index $t \in [T]$. After training on dataset $\mathcal{D}_{t+1}$ with initialization $\mathbf{M}_t$ via gradient-based learning methods to generate $\mathbf{M}_{t+1}$, for all layers l, we have that*

$$\frac{\|\mathbf{A}_{l,t+1}[\mathcal{A}_{l,t+1}] - \mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_F}{\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2} \leq \gamma |\mathcal{A}_{l,t+1}|^{-\beta}$$

where W is the width of the layers of $\mathbf{M}$ where $\gamma, \beta > 0$. Here, $\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]$ and $\mathbf{A}_{l,t+1}[\mathcal{A}_{l,t+1}]$ denote matrices that only contain the active rows from the weights at the lth layer for the tasks t and $t+1$ respectively.

This assumption states that after training when indexed by the active rows, the matrix norm of the difference between the lth layer of $\mathbf{M}_t$ and the lth layer of $\mathbf{M}_{i+1}$ normalized by the initial matrix norm is bounded by some function which is inversely proportional to the width. This relative distance from initialization is well analyzed in the literature. For more details on these values in practice, see Section 5.4. We reiterate that this phenomenon has been observed across the literature, and we reproduce this property empirically. Using this assumption, we can demonstrate an upper bound in the matrix norm of the difference of the weights at a layer for two sequentially trained models.

Lemma 4.2. *Let $\lambda_{i,j}^l = \frac{\|\mathbf{A}_{l,j}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}$ denote the ratio of the spectral norms of the weights of different row indices for different tasks. For any task t and layer l, we have*

$$\mathbb{E}\left[\frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}\right] \leq \lambda_{t,t+1}^l \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}.$$

The matrix norm of the difference between two layers measures the functional distance of the two layers. Given this upper bound on the matrix norm, we can use simple perturbation analysis to calculate how much the difference between the two outputs of the two sequentially trained models accumulates throughout the layers.

4.2.3 Width as a Functional Reguralizer

We now use simple perturbation analysis to see how far $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ will be on input from $\mathcal{D}_t$. Given the Lipschitz-smoothness of the activations at each layer and a bound on the matrix norm of the weights of different layers from Lemma 4.2, we can use simple perturbation analysis, i.e., Lemma 2 from Neyshabur et al. (2018b), to bound the continual learning error. This analysis gives us a simple guarantee on the error $\epsilon_{t,t+1}$. Moreover, using simple triangle inequality, we expand this claim to general $\epsilon_{t,t'}$ where $t' > t$.

We state the continual learning error between subsequent tasks using this analysis technique.

Lemma 4.3. *For any input vector from the dataset for the tth task $x \in \mathcal{D}_t$, the ℓ_2 norm of the difference of the outputs from models $\mathbf{M}_t$ and $\mathbf{M}_{t+1}$ are upper bounded by $\forall x \in \mathcal{D}_t$,*

$$\mathbb{E}\Big[\|\mathbf{M}_t(x) - \mathbf{M}_{t+1}(x)\|_2\Big] \leq L 2^L \chi \bar{\lambda} \left(\prod_{l=1}^{L} L_l \|\mathbf{A}_{t,l}\|_2\right) \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}.$$

Here, χ denotes the maximum norm of the input in $\mathcal{D}_t$, i.e. $\chi = \max_{x \in \mathcal{D}_t} \|x\|_2$.

This bound states that the difference between outputs in two models can be tracked as the input flows through the models. If the two models have close weights at each layer, they will also have close outputs. Here, we see how width and sparsity cause the layers to be similar, resulting in low continual learning error between two sequential models. Using such a bound, we can prove that the difference between the original and the new model differs only so much. Moreover, we can extend the above to compare models more than one task apart, i.e., comparing $\mathbf{M}_t$ and $\mathbf{M}_{t'}$. Doing so yields our final theorem. The implicit regularization of training large-width networks to find minima close to initialization acts as a functional regularization in the continual learning setting on the order of $W^{-\beta}$.

4.3 Extension to Noise Stability

The dependence on the weight norm is pessimistic in our bounds since neural networks tend to be stable to noise in the input Arora et al. (2018). To remove such dependence, we can rely on two terms: layer cushion and activation contraction.

Definition 4.1. *Let $x_{t,l-1}$ be defined as the output of the first $l-1$ layers of $\mathbf{M}_t$ for some input $x \in \mathcal{D}_t$. The layer cushion of layer l of the model $\mathbf{M}_t$ on the task t is defined to be the smallest number $\mu_{t,l}$ such that for all inputs in $x \in \mathcal{D}_t$, we have that*

$$\|\mathbf{A}_{t,l}\|_2 \|\phi_l(x_{t,l-1})\|_2 \leq \mu_{t,l} \|\mathbf{A}_{t,l}\phi_l(x_{t,l-1})\|_2.$$

Intuitively, the layer cushion constant $\mu_{t,l}$ is a data-dependent constant that tightens the pessimistic analysis of using the weight norms. Moreover, we will define the activation contraction similarly as the following.

Definition 4.2. *Let $x_{t,l-1}$ be defined as the output of the first $l-1$ layers of $\mathbf{M}_t$ for some input $x \in \mathcal{D}_t$. The activation contraction c_t for layer l and model $\mathbf{M}_t$ is defined as the smallest number such that for any layer l and any $x \in \mathcal{D}_t$ such that*

$$\|x_{t,l-1}\|_2 \leq c_t \|\phi_l(x_{t,l-1})\|_2.$$

This connection tightens the pessimistic analysis surrounding the activation layers. Combining these two definitions, we can remove the dependence on the weight norm in our bounds.

Theorem 4.2. *Denote $\Gamma_t = \max_{x \in \mathcal{D}_t} \|\mathbf{M}_t(x)\|_2$. Then, we can characterize the continual learning error between two subsequently trained models as*

$$\mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\right] \leq \Gamma_t (t'-t)\gamma\bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}} \eta,$$

where $\eta = \left(\prod_{i=1}^{l} \kappa_i + \kappa_i (t'-t)\gamma\bar{\lambda}\mu_{t,i}\right)\left(\sum_{i=1}^{l} \kappa_i\right)$ *and* $\kappa_i = L_i c_i \mu_{i,t}$.

Empirically, such constants can improve upon the pessimistic analysis using weight norms. For more details on the values of these constants in practice, please see Arora et al. (2018).

5 Experiments

To empirically validate the theoretical results in this work, we conducted a series of experiments across a pair of established continual learning benchmarks, namely Rotated MNIST and Split CIFAR 100, following prior work Mirzadeh et al. (2022a,b).

5.1 Datasets

Following prior studies in the continual learning literature (Mirzadeh et al., 2022a,b; Ramasesh et al., 2021), we train models on two standard benchmark tasks: Rotated MNIST and Split CIFAR100. In the Rotated MNIST setting, we construct five tasks comprising the original MNIST images rotated by 0, 22.5, 45, 67.5, and 90 degrees, respectively. We then train on each of these tasks in sequence. For Split CIFAR100, we follow the same procedure as in (Mirzadeh et al., 2022a) and randomly partition the original 100 classes into 20 groups, each representing a task in our continual learning process. We do not perform additional preprocessing or normalization on the original data before training.

5.2 Metrics

Following the work of (Mirzadeh et al., 2022a) we evaluate the efficacy of a continual learning model via three metrics: *Average Accuracy* (AA), *Average Forgetting* (AF), *Learning Accuracy* (LA), and *Joint Accuracy* (JA). Average accuracy is defined as the mean test accuracy of the final model over all the tasks after training on all tasks in sequence. Average forgetting is calculated as the mean difference over tasks between the accuracies obtained by the intermediate model trained on that task and the final model. The learning accuracy of a given task measures the accuracy that the model achieves immediately after training on that task. We also report learning accuracy as an average over all tasks. Finally, we report Joint Accuracy, which is the accuracy of training a model on all of the combined datasets.

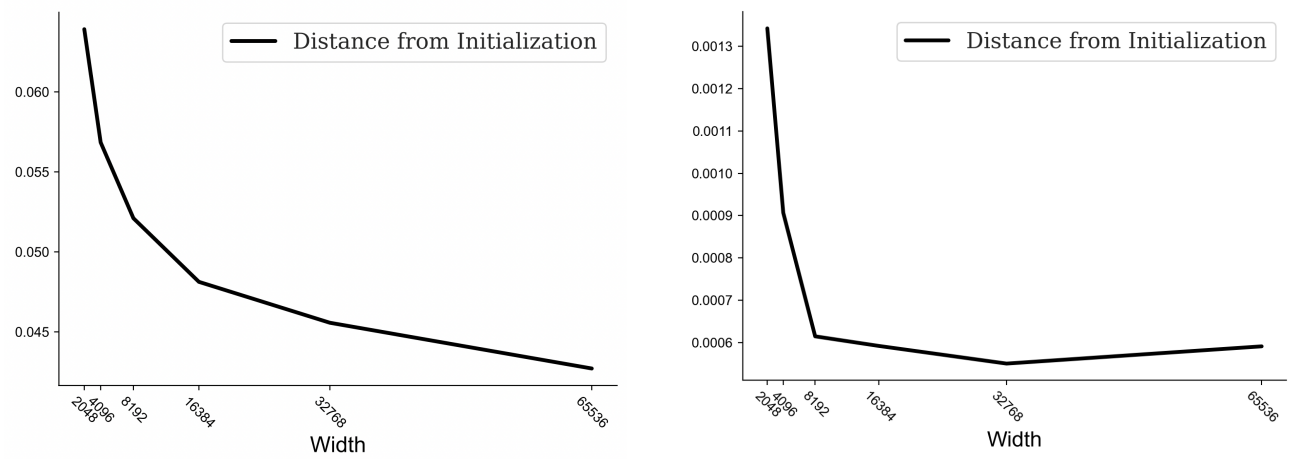
5.3 Modeling Setup

We utilize the same model architecture for all of our experiments: a Feed Forward Network consisting of an input, hidden, and output layer. Between each layer, we use ReLU activations. We vary the width of the hidden layer in all experiments while keeping all other hyperparameters fixed. We train these models using the Adam optimizer (Kingma and Ba, 2015) and Stochastic Gradient Descent, which we include in the appendix. We train these models as classifiers on each task using CrossEntropy Loss for 5 epochs. This training paradigm is a standard modeling choice on Rotated MNIST benchmark and on Split CIFAR100 in the Continual Learning literature (Mirzadeh et al., 2020) [3]. To note, while swapping out the input and output layers for each task is standard, we found that doing so did not impact the results significantly since each task has the same input and output dimensionality, so we do not swap the input and output layers for each task to better align with the theoretical analysis.

5.4 Distance From Initialization

First, in Figure 1a and Figure 1b, we note that a model's distance from initialization tends to decrease as a function of width, which provides empirical evidence of Assumption 4.1 holding. Our relative distance metric has in the numerator the Frobenius norm of the difference between the hidden layers of the intermediate models trained on the first task and second tasks. In the denominator of our metric, we have the ℓ_2 norm of the hidden layer of the model trained on the first task. We see a slowly decreasing distance as the width increases exponentially. To better understand the con-

[3] These MLP models perform well short of state-of-the-art convolutional neural networks (LeCun et al., 1995) and vision transformer (Dosovitskiy et al., 2020) architectures. Nevertheless, our work focuses on developing a rigorous and principled understanding of the effect of width on the training dynamics of continual learning.



(a) MNIST

(b) Split CIFAR100

Figure 1: We plot the distance from initialization for both Rotated MNIST and Split CIFAR 100 experiments. We see that distance from initialization decreases slowly as the width is increased for both datasets. For the constants discussed in Assumption 4.1, the best fitting constants are $\gamma = 2.5, \beta = 0.12$ for Rotated MNIST and $\gamma = 0.013, \beta = 0.311$ for Split CIFAR100.

stants in Assumption 4.1, we find what values of γ and β best explain the curves seen in Figure 1a and Figure 1a. We see values of $\gamma = 2.5, \beta = 0.12$ for MNIST and $\gamma = 0.013, \beta = 0.311$ for Split CIFAR100. This finding also validates several prior conclusions in the literature on the relationship between width and distance from initialization (Nagarajan and Kolter, 2019; Mirzadeh et al., 2020).

5.5 Dense Model Results

Furthermore, in Table 1, we see that, while learning accuracy tends to increase with width, the average continual learning accuracy and amount of forgetting plateaus as predicted by our theory. For example, in Rotated MNIST, we see that forgetting is around 39.4 with models of width 2^5 and decreases to 30.2 at widths of 2^{10}. Still, the forgetting does not significantly decrease for models of increasing width. In Split CIFAR, we see a similar trend. We note that the average forgetting is less at the beginning since small models need to be more accurate, achieving learning accuracy of only 1.0. We see a trend in Average Accuracy where the model's accuracy over all tasks slowly increases for small widths but plateaus at accuracies of around 2.2. We note that due to the large number of tasks, the Average Forgetting is high when the learning accuracy is high since the models largely forgot the initial tasks. However, the Average Accuracy increases slowly as the width increases, demonstrating diminishing returns in continual learning ability. We note that the overall learning accuracy of these models is small since FFNs are notoriously poor on CIFAR100. These findings validate our core result that width, on its own, provides diminishing returns in the continual learning setting. This demonstrates that while width positively correlated with average accuracy over all the tasks and negatively correlated with average forgetting, increasing width received diminishing returns in both metrics.

5.6 Forgetting over Time

We briefly explore the error over tasks: how much the final model forgets as the number of tasks seen increases. We plot the error over the task indices in Figure 3 for a few FFNs of differing widths on Rotated MNIST. We see a roughly linear increase in error as the number of tasks increases. We note that a task index of 1 means the final model trained on that data 1 task ago. This relationship is seemingly independent of width. This linear relationship independent of width corroborates the theoretical analysis in Theorem 4.1, which predicts a linear relationship. We relegate more details to the appendix.

5.7 Experiments in the Appendix

Here, we provide a brief overview of the experiments in the appendix. While the experiments above were conducted with Adam, we repeated the experiments with Stochastic Gradient Descent. We see similar diminishing returns in the Continual Learning ability, demonstrating that the choice of optimizer does not cause this effect. Moreover, we test how inducing rowwise sparsity affects the relationship between Continual Learning ability. Increasing the sparsity can signif-

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	63.7	39.4	93.5	93.8
2^6 $(50K)$	67.2	36.6	94.9	95.8
2^7 $(100K)$	69.1	35.1	96.1	96.7
2^8 $(200K)$	72.6	31.3	96.9	97.3
2^9 $(400K)$	72.6	31.8	97.6	97.7
2^{10} $(814K)$	74.0	30.2	98.0	97.9
2^{11} $(1.6M)$	73.5	31.0	98.1	98.1
2^{12} $(3.2M)$	72.6	32.1	96.2	98.0
2^{13} $(6.5M)$	73.6	31.0	96.9	98.1
2^{14} $(13M)$	73.1	31.5	97.1	98.0
2^{15} $(26M)$	73.2	31.5	97.2	98.2
2^{16} $(52M)$	72.7	32.1	97.1	98.0
2^{17} $(100M)$	73.4	31.1	97.2	98.1
2^{18} $(200M)$	73.6	30.9	98.1	98.4
2^{19} $(400M)$	73.3	31.2	98.0	98.2
2^{20} $(800M)$	72.9	31.9	98.1	98.3

(a) Rotated MNIST

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	1.0	19.3	19.4	5.9
2^6 $(50K)$	1.0	25.7	25.6	5.6
2^7 $(100K)$	1.1	30.0	29.7	5.5
2^8 $(200K)$	1.6	35.2	35.0	5.4
2^9 $(400K)$	1.9	38.7	38.6	6.0
2^{10} $(814K)$	2.0	39.5	39.6	5.7
2^{11} $(1.6M)$	2.1	40.9	40.9	5.5
2^{12} $(3.2M)$	2.1	40.6	40.7	5.5
2^{13} $(6.5M)$	2.2	41.2	41.3	5.5
2^{14} $(13M)$	2.1	40.5	40.5	5.3
2^{15} $(26M)$	1.9	40.6	40.5	5.2
2^{16} $(52M)$	2.2	41.2	41.3	5.5
2^{17} $(100M)$	2.1	41.0	41.0	5.5
2^{18} $(200M)$	2.1	41.0	41.1	5.2
2^{19} $(400M)$	2.1	40.8	40.7	5.2
2^{20} $(800M)$	2.2	40.3	40.4	5.2

(b) Split CIFAR 100

Table 1: Our Continual Learning experiments on varying width FFNs on Rotated MNIST and Split CIFAR100. We see that the Average Forgetting slowly stops decreasing after a width of 2^{10}.

(a) Width 16384 (b) Width 32768 (c) Width 65535

Figure 3: We see that the error over tasks is roughly linear on MNIST. This linear relationship between the number of tasks and errors corroborates our theoretical analysis.

icantly decrease the average forgetting, as predicted by our theoretical analysis. Furthermore, we study the relationship between forgetting and the number of tasks trained. We find a roughly linear relationship, as indicated by our theory. Moreover, we test whether these diminishing returns can be seen when wide models are used alongside existing Continual Learning methods.

6 Discussion

We examine both empirically and theoretically the connection between continual learning ability and the width of a Feed Forward Network. Increasing width receives diminishing returns at some point. We empirically see these diminishing returns on Rotated MNIST and Split CIFAR100 when training FFNs of larger hidden dimensions than tested in previous literature. As possible extensions, it would be interesting to see if similar increases in scale receive diminishing returns such as depth, number of channels in CNNs, hidden dimension in LLMs, etc. Moreover, an important work could be to examine if width, in conjunction with other functional regularization methods, can reduce the effect of diminishing returns.

Limitations Our analysis relies on an assumption about the distance from initialization seen during training, which we have yet to prove rigorously. Moreover, our analysis is restricted to Feed Forward Networks and has yet to be extended to more complex architectures such as Residual Networks or Attention networks. Furthermore, our analysis is likely loose concerning parameters such as the dependence on depth.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/arora18b.html.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Mehdi Abbana Bennani and Masashi Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. In *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Nikita Dhawan, Sicong Huang, Juhan Bae, and Roger Baker Grosse. Efficient parametric approximations of neural network function space distance. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7795–7812. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dhawan23a.html.

Tom Diethe, Tom Borchert, Eno Thereska, Borja Balle, and Neil Lawrence. Continual learning in practice. *arXiv preprint arXiv:1903.05202*, 2019.

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR, 2021.

A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, and T Unterthiner. 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, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1675–1685. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/du19c.html.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020.

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11): 113301, 2020.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Limitations of lazy training of two-layers neural network. *Advances in Neural Information Processing Systems*, 32, 2019.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Mohammad Emtiyaz E Khan and Siddharth Swaroop. Knowledge-adaptation priors. *Advances in Neural Information Processing Systems*, 34:19757–19770, 2021.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. *Advances in Neural Information Processing Systems*, 35:5065–5079, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR (Poster)*, 2015. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pages 5327–5337. PMLR, 2020.

Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj

Chakravarthi Raja, Nick Cheney, Jeff Clune, et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31, 2018.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Nicholas Meisburger, Vihan Lakshman, Benito Geordie, Joshua Engels, David Torres Ramos, Pratik Pranav, Benjamin Coleman, Benjamin Meisburger, Shubh Gupta, Yashwanth Adunukota, et al. Bolt: An automated deep learning framework for training and deploying large-scale neural networks on commodity cpu hardware. *arXiv preprint arXiv:2303.17727*, 2023.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*, 2020.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International Conference on Machine Learning*, pages 15699–15717. PMLR, 2022a.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning. *arXiv preprint arXiv:2202.00275*, 2022b.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018a.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018b.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2020.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Liangzu Peng, Paris Giampouras, and René Vidal. The ideal continual learner: An agent that never forgets. In *International Conference on Machine Learning*, pages 27585–27610. PMLR, 2023.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2021.

Mark B Ring. Child: A first step towards continual learning. *Machine Learning*, 28:77–104, 1997.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109:467–492, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Infrastructure

We conducted all model training experiments on a single Intel(R) Xeon(R) CPU E5-2680 server with 250GB of RAM. Since extremely wide feed-forward networks consume a considerably large amount of memory (the largest models we train require approximately 150GB of RAM), we conduct all of our experiments on a commodity CPU server, which has significantly more memory than commercial GPU machines. To train these models in a feasible amount of time on a CPU machine, we utilize the BOLT deep learning library Meisburger et al. (2023), which is optimized for CPU neural network training, particularly wide FFNs [4].

B Additional Continual Learning Experiments

Here, we report several additional experiments to investigate the relationship between width and the ability to learn continually. We demonstrate the relationship between SGD and this diminishing returns phenomenon by repeating the experiments with SGD in Appendix B.1. Moreover, we test how inducing row-wise sparsity affects the relationship between Continual Learning ability. Increasing the sparsity can significantly decrease the average forgetting, as predicted by our theoretical analysis. We demonstrate this in Appendix B.2.

Furthermore, we study the relationship between forgetting and the number of tasks trained. We report on this in Appendix B.3. We also note that the original Split CIFAR100 experiment was done over 20 tasks as is traditional in the continual learning experiment. However, since forgetting is very large in this setting for FFNs, we repeat the Split CIFAR100 experiments with 5 tasks and 10 tasks. Here, forgetting should be less. We report this in Appendix B.4.

B.1 SGD and Diminishing Returns

To further investigate the connection between width and continual learning, we wish to see whether the phenomenon of diminishing returns is visible in other optimization algorithms. In our theoretical analysis, we argue that the implicit regularization of gradient-based optimizers finds minima closer to previous tasks, improving continual learning ability. In our experiments in the main section, we conduct all experiments with Adam and empirically observe these diminishing returns. We wish to examine if this implicit regularization and diminishing returns are unique to Adam or generalizes to other optimization algorithms. To do so, we repeat the continual learning experiments in the main text with SGD as the optimizer. We report our results in Table 2. Indeed, we see very similar results to Adam as in Table 2. We again observe the diminishing returns for both datasets with SGD as the optimizer. This observation suggests that our analysis and this phenomenon are not unique to Adam and extend to different optimizers.

B.2 Sparsity and Width

In Table 3, we report the results of our investigation into the power of row-wise sparsity to mitigate the effects of catastrophic forgetting, as predicted by our theory. To summarize our earlier discussion, we define row-wise sparsity as activated only a subset of the hidden layer neurons in our feedforward model architecture. We select a fraction α of the hidden layer neurons for each task to be active uniformly at random. Note that some neurons may be shared across multiple tasks, which leads to some amount of forgetting, but at a significantly reduced rate to what we observed in the dense case. Moreover, the overall learning accuracy is not decreased despite fewer neurons being used during training. In these experiments, we also apply sparsity to the output layer such that each task has a distinct set of output classes. As predicted by our theory, we observe that sparsity can counterweight the diminishing returns on reducing forgetting purely through width alone. However, some effects of diminishing returns can still be seen in both datasets.

B.3 Accuracy over number of tasks

We measure the accuracies of the final learned model over all tasks and plot how the accuracy decreases. We do this over both datasets, Split CIFAR100 and Rotated MNIST. We conducted this experiment over several

[4]We note that our ability to conduct all of our empirical studies on a low-cost CPU machine may be of independent interest to the machine learning community in demonstrating a viable research agenda that does not require the need for expensive and scarce GPU servers.

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	62.5	41.0	93.5	93.8
2^6 $(50K)$	67.6	36.0	94.8	95.8
2^7 $(100K)$	70.0	36.0	94.7	96.7
2^8 $(200K)$	71.4	33.0	97.0	97.3
2^9 $(400K)$	72.8	31.4	97.6	97.7
2^{10} $(814K)$	74.2	30.0	97.9	97.9
2^{11} $(1.6M)$	73.5	31.1	98.1	98.1
2^{12} $(3.2M)$	73.0	32.1	98.1	98.0
2^{13} $(6.5M)$	73.2	31.5	98.1	98.1
2^{14} $(13M)$	72.7	31.6	98.1	98.0
2^{15} $(26M)$	73.4	31.2	98.0	98.0
2^{16} $(52M)$	72.7	31.0	98.0	98.0
2^{17} $(100M)$	73.6	30.9	97.8	98.1
2^{18} $(200M)$	73.9	30.6	98.0	98.4
2^{19} $(400M)$	73.5	31.1	98.0	98.2
2^{20} $(800M)$	73.4	31.3	98.0	98.3

(a) Rotated MNISTT

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	1.0	13.4	12.8	5.9
2^6 $(50K)$	1.4	24.6	24.7	5.6
2^7 $(100K)$	1.8	29.0	29.3	5.5
2^8 $(200K)$	2.0	36.0	36.2	5.4
2^9 $(400K)$	1.9	35.3	35.4	6.0
2^{10} $(814K)$	2.2	39.8	40.0	5.7
2^{11} $(1.6M)$	2.2	40.5	40.6	5.5
2^{12} $(3.2M)$	2.3	40.7	40.9	5.5
2^{13} $(6.5M)$	2.3	41.2	41.4	5.5
2^{14} $(13M)$	2.1	40.5	40.5	5.3
2^{15} $(26M)$	2.1	40.1	40.2	5.2
2^{16} $(52M)$	2.2	40.5	40.6	5.5
2^{17} $(100M)$	2.2	40.4	40.7	5.5
2^{18} $(200M)$	2.3	40.8	41.0	5.2
2^{19} $(400M)$	2.2	40.8	41.0	5.2
2^{20} $(800M)$	2.2	40.9	41.1	5.2

(b) Split CIFAR100

Table 2: Continual Learning Results with Stochastic Gradient Descent Training. We see that despite changing the optimizer to SGD, we see very similar results, including the diminishing returns trend. This suggests that the diminishing returns phenomenon is not optimizer-dependent.

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	68.5	5.4	73.9	93.8
2^6 $(50K)$	75.6	13.3	88.9	95.8
2^7 $(100K)$	82.0	9.8	91.8	96.7
2^8 $(200K)$	86.2	6.8	83.1	97.3
2^9 $(400K)$	91.2	3.3	94.5	97.7
2^{10} $(814K)$	94.6	1.3	95.9	97.9
2^{11} $(1.6M)$	95.5	1.3	96.7	98.1
2^{12} $(3.2M)$	96.4	1.0	97.3	98.0
2^{13} $(6.5M)$	97.1	0.5	97.6	98.1
2^{14} $(13M)$	97.7	0.2	97.9	98.0
2^{15} $(26M)$	97.6	0.3	98.0	98.2

(a) Rotated MNIST

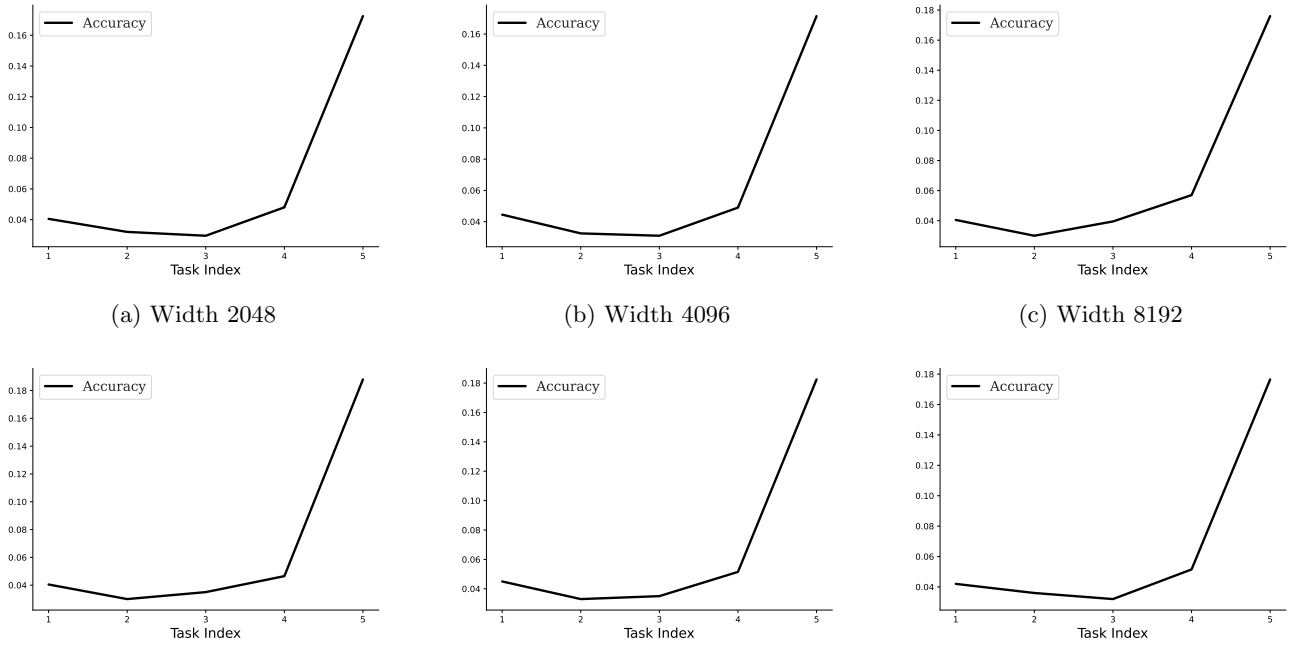
Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	23.9	2.4	26.3	5.9
2^6 $(50K)$	27.4	7.7	35.1	5.6
2^7 $(100K)$	28.6	8.6	37.3	5.5
2^8 $(200K)$	29.0	9.8	38.9	5.4
2^9 $(400K)$	30.4	9.1	39.6	6.0
2^{10} $(814K)$	32.2	8.6	40.8	5.7
2^{11} $(1.6M)$	31.1	11.0	42.2	5.5
2^{12} $(3.2M)$	33.2	9.4	42.6	5.5
2^{13} $(6.5M)$	32.6	9.8	42.4	5.5
2^{14} $(13M)$	35.5	6.2	41.7	5.3
2^{15} $(26M)$	34.4	7.4	41.5	5.2

(b) Split CIFAR 100

Table 3: Continual Learning experiments with Row-Wise Sparsity with $\alpha = 0.1$. We see that increasing row-wise sparsity can significantly decrease forgetting while not decreasing overall learning accuracy. This corroborates our theoretical results. We still do see diminishing returns in terms of increasing width and continual learning.

different width networks. Our analysis predicts that there should be a roughly linear relationship between the accuracy and the number of tasks. With more intermediate tasks between an initial task and the final task, the model will decrease roughly linearly in accuracy.

Moreover, our analysis predicts that this error should be independent of the width. We report the results of our experiments in Figure 6 and Figure 7. We see in our Rotated MNIST experiments in Figure 7 that the accuracy roughly decreases as the number of tasks increases. Moreover, this relationship holds across all the widths tested. This observation corroborates our theory. However, on Split CIFAR100 in Figure 6, the accuracy decreases roughly exponentially as the number of tasks increases. This connection may be because the accuracy is already very low even after the 1 task has passed, so there is no room for another linear decrease in accuracy. However, we still see that this relationship holds across all widths tested.



(a) Width 2048 (b) Width 4096 (c) Width 8192

(d) Width 16384 (e) Width 32768 (f) Width 65535

Figure 6: We report the accuracy over tasks on Split CIFAR100. We see that the model is the most accurate on the most recent task, and the accuracy decreases roughly exponentially. This does not corroborate our theory, which predicts a linear relationship.

B.4 Split CIFAR100 Experiments with fewer tasks

Here, we repeat the Split CIFAR 100 experiments with fewer tasks. In the main experiments, we split the CIFAR100 into 20 tasks. We empirically found that doing so led to very high forgetting of FFNs. To investigate the relationship between width and continual learning on the Split CIFAR100 dataset more clearly, we repeat the main experiment over 10 and 5 tasks and report the results in Table 4. In both cases, reducing the number of tasks increases overall accuracy. Moreover, the average accuracy increases slowly but plateaus as the width increases. This trend is similar to that of Table 1 with 20 tasks.

C Theoretical Analysis

We will first prove the main claim connecting width and continual learning Theorem 4.1 in Appendix C.1. We then extend this analysis to noise stability in Appendix C.2.

C.1 Proof of Theorem 4.1

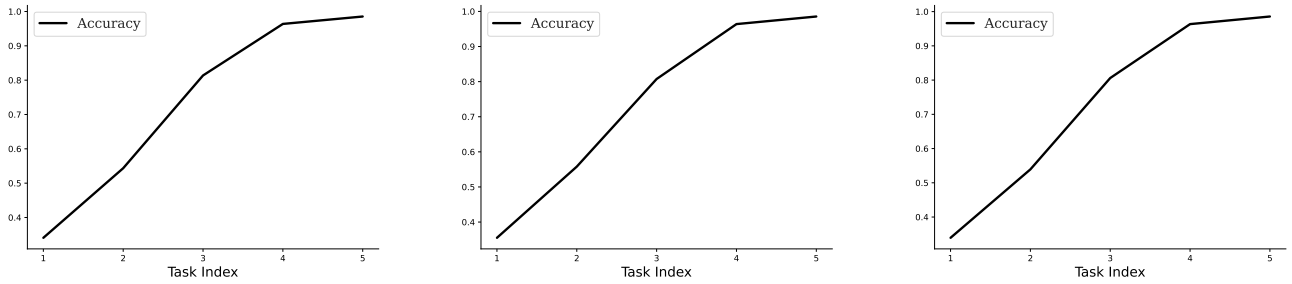
We formalize the proof of Theorem 4.1 given the intuition from proof sketch of from the main body. To do the analysis for $\epsilon_{t,t+1}$, we split the proof into three parts: (1) finding how many active rows are shared between anything two layers, (2) finding how far these active rows can change during training, and (3) combining the two parts together using perturbation analysis.

Lemma 4.1. *For any two sequential task indices t and $t+1$ and layer l, the expected size of the intersection between the sets of active rows $\mathcal{A}_{t,l}$ and $\mathcal{A}_{t+1,l}$ is*

$$\mathbb{E}(|\mathcal{A}_{t,l} \cap \mathcal{A}_{t+1,l}|) = \alpha^2 W.$$

Proof. Let $\mathbb{I}_i$ be the indicator random variable of whether a row i is active in both models $\mathbf{M}_t$ and $\mathbf{M}_{t+1}$, i.e. $\mathbb{I}_i$ if $i \in \mathcal{A}_{t,l} \wedge i \in \mathcal{A}_{t',l}$ and 0 otherwise. Therefore, the expected value of the size of the intersection of the two sets $\mathcal{A}_{t,l}$ and $\mathcal{A}_{t'}$ is

(a) Width 2048 (b) Width 4096 (c) Width 8192



(d) Width 16384 (e) Width 32768 (f) Width 65535

Figure 7: We report the accuracy over tasks on Rotated MNIST. We wee that the model is the most accurate on the most recent task and the accuracy decreases roughly linearly as predicted by our theory.

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	11.5	13.0	23.2	5.9
2^6 $(50K)$	11.7	12.7	23.2	5.6
2^7 $(100K)$	12.0	13.5	24.0	5.5
2^8 $(200K)$	11.8	13.5	24.0	5.4
2^9 $(400K)$	12.2	13.8	24.6	6.0
2^{10} $(814K)$	12.0	13.9	24.5	5.7
2^{11} $(1.6M)$	12.0	13.9	24.6	5.5
2^{12} $(3.2M)$	12.1	14.0	24.6	5.5
2^{13} $(6.5M)$	12.2	14.4	24.9	5.5
2^{14} $(13M)$	12.3	14.2	25.1	5.3
2^{15} $(26M)$	12.2	13.6	24.5	5.2
2^{16} $(52M)$	11.7	14.5	24.8	5.5
2^{17} $(100M)$	11.7	14.1	24.4	5.5
2^{18} $(200M)$	12.2	13.9	24.7	5.2
2^{19} $(400M)$	12.2	13.5	24.4	5.2
2^{20} $(800M)$	12.3	13.9	24.8	5.2

(a) Split CIFAR 100 (10 tasks)

Width (Parameters)	AA	AF	LA	JA
2^5 $(25K)$	6.6	12.6	16.6	5.9
2^6 $(50K)$	6.7	13.0	17.1	5.6
2^7 $(100K)$	6.7	13.4	17.1	5.5
2^8 $(200K)$	6.8	13.4	17.4	5.4
2^9 $(400K)$	6.8	13.8	17.8	6.0
2^{10} $(814K)$	6.6	13.8	17.8	5.7
2^{11} $(1.6M)$	6.6	13.4	17.4	5.5
2^{12} $(3.2M)$	7.2	13.2	17.8	5.5
2^{13} $(6.5M)$	6.5	14.4	17.4	5.5
2^{14} $(13M)$	6.5	14.2	17.9	5.3
2^{15} $(26M)$	6.6	13.6	17.5	5.2
2^{16} $(52M)$	6.6	13.6	13.6	5.5
2^{17} $(100M)$	6.5	13.6	17.4	5.5
2^{18} $(200M)$	6.5	14.3	17.9	5.2
2^{19} $(400M)$	6.7	14.1	18.0	5.2
2^{20} $(800M)$	7.0	13.4	17.5	5.2

(b) Split CIFAR 100 (5 tasks)

Table 4: Additional Split CIFAR100 Results with 10 and 5 tasks. We see a similar trend that the average accuracy increases slowly and plateaus. Moreover, the forgetting does not significantly decrease as the width is increased. We do see that the overall accuracies are larger since having fewer tasks makes continual learning easier.

$$\begin{aligned}\mathbb{E}(|\mathcal{A}_{t,l} \cap \mathcal{A}_{t',l}|) &= \mathbb{E}\left[\sum_{i\in[W]} \mathbb{I}_i\right] \\ &= \sum_{i\in[W]} \mathbb{E}\left[\mathbb{I}_i\right]\end{aligned}$$

Now, given that the probability that each row i is in a given active set with probability α, the probability a row is randomly in both active sets is α^2, i.e. $\mathbb{E}(\mathbb{I}_i) = \alpha^2$. Given there are W total rows this yields $\mathbb{E}(|\mathcal{A}_{t,l} \cap \mathcal{A}_{t',l}|) = \alpha^2 W$. □

Lemma 4.2. *Let $\lambda_{i,j}^l = \frac{\|\mathbf{A}_{l,j}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}$ denote the ratio of the spectral norms of the weights of different row indices for different tasks. For any task t and layer l, we have*

$$\mathbb{E}\left[\frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}\right] \leq \lambda_{t,t+1}^l \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}.$$

Proof. Now, in expectation, the size of $\mathcal{A}_{t,l}$ is αW, i.e.

$$\mathbb{E}(|\mathcal{A}_{t,l}|) = \alpha W.$$

Only $\alpha^2 W$ of the αW active rows $\mathcal{A}_{t,l}$ will intersect $\mathcal{A}_{t+1,l}$ in expectation, from Lemma 4.1. Therefore, when training $\mathbf{M}_{t+1}$ only $\alpha^2 W$ of the αW rows in expectation will change from its initialization. The rest of the rows will stay unchanged during training for task $t+1$. Now, $\mathbf{A}_{t,l}[\mathcal{A}_{t,l}](x) - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]$ is a matrix in $\mathbb{R}^{\mathcal{A}_{t,l} \times W}$. Moreover, this matrix will have $\alpha(1-\alpha)W$ rows of all 0's in expectation. For the other rows, we know from Assumption 4.1, the difference of the two layers indexed by $\mathcal{A}_{t+1,l}$ is bounded by

$$\frac{\|\mathbf{A}_{l,t+1}[\mathcal{A}_{l,t+1}] - \mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_F}{\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2} \leq \gamma[\alpha W]^{-\beta}$$

By the definition of Frobenius Norm

$$\begin{aligned}\sum_{i \in \mathcal{A}_{l,t+1}} \|\mathbf{A}_{l,t+1}[i] - \mathbf{A}_{l,t}[i]\|_2^2 &= \|\mathbf{A}_{l,t+1}[\mathcal{A}_{l,t+1}] - \mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_F^2 \\ &\leq \gamma^2[\alpha W]^{-2\beta}\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2^2\end{aligned}$$

Since the expectation of sum is the sum of expectation, we have that

$$\mathbb{E}\left(\sum_{i \in \mathcal{A}_{l,t+1}} \|\mathbf{A}_{l,t+1}[i] - \mathbf{A}_{l,t}[i]\|_2^2\right) = \sum_{i \in \mathcal{A}_{l,t+1}} \mathbb{E}\left(\|\mathbf{A}_{l,t+1}[i] - \mathbf{A}_{l,t}[i]\|_2^2\right).$$

Since all rows are exchangeable under training, we have that the expected ℓ_2 norm of a row is upper bounded by

$$\mathbb{E}\left[\|\mathbf{A}_{l,t+1}[i] - \mathbf{A}_{l,t}[i]\|_2^2\right] \leq \gamma^2[\alpha W]^{-2\beta-1}\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2^2.$$

Therefore, for rows in both active sets $\mathcal{A}_{l,t+1} \cup \mathcal{A}_{l,t}$, the expected ℓ_2 norm of the difference of the rows in $\mathbf{M}_t$ and $\mathbf{M}_{t+1}$ is $\gamma^2[\alpha W]^{-2\beta-1}$. For notational ease, let $\mathcal{I} = \mathcal{A}_{t,l} \cap \mathcal{A}_{t+1,l}$ denote the intersection of the active rows while $\mathcal{O} = \mathcal{A}_{t,l} \cap \mathcal{A}_{t+1,l}^C$ denote the active rows not in the active task. Therefore,

$$\begin{aligned}&\mathbb{E}\left[\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]\|_2\right] \\ &\quad\leq \mathbb{E}\left[\|\mathbf{A}_{t,l}[\mathcal{I}] - \mathbf{A}_{t+1,l}[\mathcal{I}]\|_2 + \|\mathbf{A}_{t,l}[\mathcal{O}] - \mathbf{A}_{t+1,l}[\mathcal{O}]\|_2\right] &(1)\\ &\quad= \mathbb{E}\left[\|\mathbf{A}_{t,l}[\mathcal{I}] - \mathbf{A}_{t+1,l}[\mathcal{I}]\|_2\right] &(2)\\ &\quad\leq \mathbb{E}\left[\|\mathbf{A}_{t,l}[\mathcal{I}] - \mathbf{A}_{t+1,l}[\mathcal{I}]\|_F\right] \\ &\quad\leq \sqrt{\sum_{i \in \mathcal{I}} \mathbb{E}\left[\|\mathbf{A}_{t,l}[i] - \mathbf{A}_{t+1,l}[i]\|_2^2\right]} &(3)\\ &\quad\leq \sqrt{\alpha^2 W \cdot \gamma^2[\alpha W]^{-2\beta-1}\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2^2} \\ &\quad\leq \gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t+1}]\|_2 \\ &\quad= \gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t}]\|_2\lambda_{t,t+1}^l\end{aligned}$$

Here, Equation (1) comes from splitting the matrix-multiply by different rows, Equation (2) comes from seeing that all rows in $\mathcal{O}$ will remain unchanged after training, and Equation (3) comes from $\mathbb{E}(\sqrt{X}) \leq \sqrt{\mathbb{E}(X)}$ by Jensen's Inequality for a random variable X.

From here, we have that

$$\mathbb{E}\left[\frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{l,t}[\mathcal{A}_{l,t}]\|_2}\right] \leq \gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}\lambda_{t,t+1}^l$$

□

Lemma C.1. *Say we generate a series of models $\mathbf{M}_1, \ldots, \mathbf{M}_T$ by training sequentially on datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$. Let $\lambda_{i,j}^l = \frac{\|\mathbf{A}_{l,j}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}$ denote the ratio of the spectral norms of the weights of different row indeces for different tasks. Moreover, let $\bar{\lambda} = \max_{l\in[L],i,j\in[T]} \lambda_{i,j}^l$. For any input vector from the ith dataset $x \in \mathcal{D}_i$, the ℓ_2 norm of the difference of the outputs from models $\mathbf{M}_t$ and $\mathbf{M}_{t+1}$ are upper bounded by*

$$\forall x \in \mathcal{D}_t, \quad \mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t+1}(x)\|_2\right] \leq L2^L\bar{\lambda}\chi\left(\prod_{l=1}^{L} L_l\|\mathbf{A}_{t,l}\|_2\right)\gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}.$$

Proof. We will begin by proving the perturbation analysis The first part of this proof mainly follows from Neyshabur et al. (2018a). We restate it here with the differing notation for clarity and completeness. We will prove the induction hypothesis that for any $x \in \mathcal{D}_t$,

$$\|\mathbf{M}_{t,l}(x) - \mathbf{M}_{t+1,l}(x)\|_2 \leq 2^l\|x\|_2\left(\prod_{i=1}^{l} L_i\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right)\sum_{i=1}^{l}\frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}.$$

Here, $\mathbf{M}_{t,l}$ and $\mathbf{M}_{t+1,l}$ denote the models $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ respectively with only the first l layers. The base case of induction trivially holds, given that $\|\mathbf{M}_{t,0}(x) - \mathbf{M}_{t+1,0}(x)\|_2 = 0$ by definition. Now, we prove the induction step. Assume that the induction hypothesis holds for $l-1$. We will prove that it holds for l. For notational ease, denote $\mathbf{U}_{t,l} = \mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t+1,l}[\mathcal{A}_{t,l}]$, $x_{t,l} = \mathbf{M}_{t,l}(x)$, and $x_{t+1,l} = \mathbf{M}_{t+1,l}(x)$. We have that

$$\begin{aligned}
&\|x_{t,l} - x_{t+1,l}\|_2 \\
&\quad\leq \|\left(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] + \mathbf{U}_{t,l}\right)\phi_l(x_{t+1,l-1}) - \mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq \|\left(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] + \mathbf{U}_{t,l}\right)\left(\phi_l(x_{t+1,l-1}) - \phi_l(x_{t,l-1})\right) + \mathbf{U}_{t,l}\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq \left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 + \|\mathbf{U}_{t,l}\|_2\right)\|\phi_l(x_{t+1,l-1}) - \phi_l(x_{t,l-1})\|_2 + \|\mathbf{U}_{t,l}\|_2\|\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq L_l\left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 + \|\mathbf{U}_{t,l}\|_2\right)\|x_{t+1,l-1} - x_{t,l-1}\|_2 + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (4) \\
&\quad\leq 2L_l\left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right)\|x_{t+1,l-1} - x_{t,l-1}\|_2 + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \\
&\quad\leq 2L_l\left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right)\left(1+\frac{1}{L}\right)^{l-1}\|x_{t,0}\|_2\left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right)\sum_{i=1}^{l-1}\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2} \\
&\qquad\qquad + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (5) \\
&\quad\leq 2^l L_l\left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right)\sum_{i=1}^{l-1}\frac{\|\mathbf{U}_{t,i}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2}\|x_{t,0}\|_2 \\
&\qquad\qquad + L_l\|x_{t,0}\|_2\|\mathbf{U}_{t,l}\|_2\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2 \\
&\quad\leq 2^l L_l\left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right)\sum_{i=1}^{l-1}\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2}\|x_{t,0}\|_2 \\
&\qquad\qquad + \|x_{t,0}\|_2\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}\prod_{i=1}^{l} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2 \\
&\quad\leq 2^L\left(\prod_{i=1}^{l} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right)\sum_{i=1}^{l}\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2}\|x_{t,0}\|_2
\end{aligned}$$

Here, Equation (4) comes from the fact that ϕ_l is L_l-Lipschitz smooth and that $\phi_l(0) = 0$. Moreover, Equation (5) comes from applying the induction hypothesis. Therefore, we have proven the induction hypothesis for all layers. We have

$$\forall x \in \mathcal{D}_t, \quad \|\mathbf{M}_t(x) - \mathbf{M}_{t+1}(x)\|_2 \leq 2^L \|x\|_2 \left(\prod_{l=1}^{L} L_l \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \sum_{l=1}^{L} \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}.$$

Denoting $\chi = \max_{x \in \mathcal{D}_t} \|x\|_2$ and using Lemma 4.2

$$\forall x \in \mathcal{D}_t, \quad \mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t+1}(x)\|_2\right] \leq 2^L \chi \left(\prod_{l=1}^{L} L_l \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \sum_{l=1}^{L} \lambda_{t,t+1}^{l} \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}.$$

Here, we reminded the reader that $\lambda_{i,j}^{l} = \frac{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,j}]\|_2}$ denotes the ratio of the spectral norms of the weights of different row indeces for different tasks. Moreover, for any matrix, removing rows cannot increase the matrix norm, we have that $\|\mathbf{A}_{t,l}\|_2 \geq \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}\|_2$. Therefore, we have

$$\prod_{l=1}^{L} \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 \leq \prod_{l=1}^{L} \|\mathbf{A}_{t,l}\|_2.$$

Given $\bar{\lambda} = \max_{l \in [L], i,j \in [T]} \lambda_{i,j}^{l}$,

$$\forall x \in \mathcal{D}_t, \quad \mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t+1}(x)\|_2\right] \leq L 2^L \bar{\lambda} \chi \left(\prod_{l=1}^{L} L_l \|\mathbf{A}_{t,l}\|_2\right) \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}.$$

□

Theorem 4.1. *(Informal) Say we generate a series of models $\mathbf{M}_1, \ldots, \mathbf{M}_T$ by training sequentially on datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$ according to Section 3.3. Let $\lambda_{i,j}^{l} = \frac{\|\mathbf{A}_{l,j}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}$ denote the ratio of the spectral norms of the weights of different row indices for different tasks. Moreover, let $\bar{\lambda} = \max_{l \in [L], i,j \in [T]} \lambda_{i,j}^{l}$. For all input vectors from the tth dataset $\forall x \in \mathcal{D}_t$, the ℓ_2 norm of the difference of the outputs from models $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ such that $t' \geq t$ are upper bounded [5] by*

$$\mathbb{E}\Big[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\Big] = \mathcal{O}\left((t'-t) L 2^L \bar{\lambda} \chi \left(\prod_{l=1}^{L} L_l \|\mathbf{A}_{t,l}\|_2\right) \gamma W^{-\beta} \alpha^{\frac{1-2\beta}{2}}\right).$$

Here, χ denotes the maximum norm of the input in $\mathcal{D}_t$, i.e. $\chi = \max_{x \in \mathcal{D}_t} \|x\|_2$. Here, γ, β are data-dependent positive real values.

Proof. We need only repeat the proof above but with a different perturbation to account for the number of tasks. We repeat it for clarity. We will prove the induction hypothesis that for any $x \in \mathcal{D}_t$,

$$\|\mathbf{M}_{t,l}(x) - \mathbf{M}_{t',l}(x)\|_2 \leq 2^l \|x\|_2 \left(\prod_{i=1}^{l} L_i \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \sum_{i=1}^{l} \frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t',l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}.$$

Here, $\mathbf{M}_{t,l}$ and $\mathbf{M}_{t',l}$ denote the models $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ respectively with only the first l layers. The base case of induction trivially holds, given that $\|\mathbf{M}_{t,0}(x) - \mathbf{M}_{t',0}(x)\|_2 = 0$ by definition. Now, we prove the induction step.

[5] We can reduce the dependence on weight norms by using noise stability properties. For more details, please see Section 4.3.

Assume that the induction hypothesis holds for $l-1$. We will prove that it holds for l. For notational ease, denote $\mathbf{U}_{t,l} = \mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t',l}[\mathcal{A}_{t,l}]$, $x_{t,l} = \mathbf{M}_{t,l}(x)$, and $x_{t',l} = \mathbf{M}_{t',l}(x)$. We have that

$$
\begin{aligned}
&\|x_{t,l} - x_{t',l}\|_2 \\
&\quad\leq \| \left(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] + \mathbf{U}_{t,l}\right) \phi_l(x_{t',l-1}) - \mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq \| \left(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] + \mathbf{U}_{t,l}\right) \left(\phi_l(x_{t',l-1}) - \phi_l(x_{t,l-1})\right) + \mathbf{U}_{t,l}\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq \left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 + \|\mathbf{U}_{t,l}\|_2\right) \|\phi_l(x_{t',l-1}) - \phi_l(x_{t,l-1})\|_2 + \|\mathbf{U}_{t,l}\|_2\|\phi_l(x_{t,l-1})\|_2 \\
&\quad\leq L_l \left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 + \|\mathbf{U}_{t,l}\|_2\right) \|x_{t',l-1} - x_{t,l-1}\|_2 + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (6) \\
&\quad\leq 2L_l \left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \|x_{t',l-1} - x_{t,l-1}\|_2 + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \\
&\quad\leq 2L_l \left(\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \left(1 + \frac{1}{L}\right)^{l-1} \|x_{t,0}\|_2 \left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right) \sum_{i=1}^{l-1} \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2} \\
&\qquad\qquad + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (7) \\
&\quad\leq 2^l L_l \left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right) \sum_{i=1}^{l-1} \frac{\|\mathbf{U}_{t,i}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2} \|x_{t,0}\|_2 \\
&\qquad\qquad + L_l\|x_{t,0}\|_2\|\mathbf{U}_{t,l}\|_2 \prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2 \\
&\quad\leq 2^l L_l \left(\prod_{i=1}^{l-1} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right) \sum_{i=1}^{l-1} \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2} \|x_{t,0}\|_2 \\
&\qquad\qquad + \|x_{t,0}\|_2 \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \prod_{i=1}^{l} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2 \\
&\quad\leq 2^L \left(\prod_{i=1}^{l} L_i\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2\right) \sum_{i=1}^{l} \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,i}[\mathcal{A}_{t,i}]\|_2} \|x_{t,0}\|_2
\end{aligned}
$$

Here, Equation (6) comes from the fact that ϕ_l is L_l-Lipschitz smooth and that $\phi_l(0) = 0$. Moreover, Equation (7) comes from applying the induction hypothesis. Therefore, we have proven the induction hypothesis for all layers. We have

$$
\forall x \in \mathcal{D}_t, \quad \|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2 \leq 2^L\|x\|_2 \left(\prod_{l=1}^{L} L_l\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) \sum_{l=1}^{L} \frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}.
$$

Now,

$$
\begin{aligned}
\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} &= \frac{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{A}_{t',l}[\mathcal{A}_{t,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \\
&= \frac{\left\|\sum_{i=t}^{t'-1} \mathbf{A}_{i,l}[\mathcal{A}_{i,l}] - \mathbf{A}_{i+1,l}[\mathcal{A}_{i+1,l}]\right\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \\
&\leq \frac{\sum_{i=t}^{t'-1} \|\mathbf{A}_{i,l}[\mathcal{A}_{i,l}] - \mathbf{A}_{i+1,l}[\mathcal{A}_{i+1,l}]\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}
\end{aligned}
$$

Therefore, in expectation,

$$
\mathbb{E}\left[\frac{\|\mathbf{U}_{t,l}\|_2}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}\right] \leq (t'-t)\bar{\lambda}\gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}
$$

Denoting $\chi = \max_{x\in\mathcal{D}_t} \|x\|_2$ and using Lemma 4.2

$$
\forall x \in \mathcal{D}_t, \quad \mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\right] \leq L2^L\chi \left(\prod_{l=1}^{L} L_l\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\right) (t'-t)\bar{\lambda}\gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}.
$$

Here, we reminded the reader that $\lambda_{i,j}^l = \frac{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,i}]\|_2}{\|\mathbf{A}_{l,i}[\mathcal{A}_{l,j}]\|_2}$ denotes the ratio of the spectral norms of the weights of different row indeces for different tasks. Moreover, for any matrix, removing rows cannot increase the matrix norm, we have that $\|\mathbf{A}_{t,l}\|_2 \geq \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}\|_2$. Therefore, we have

$$\prod_{l=1}^{L} \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2 \leq \prod_{l=1}^{L} \|\mathbf{A}_{t,l}\|_2.$$

Given $\bar{\lambda} = \max_{l\in[L],i,j\in[T]} \lambda_{i,j}^l$,

$$\forall x \in \mathcal{D}_t, \quad \mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\right] \leq (t'-t)L2^L\bar{\lambda}\chi\left(\prod_{l=1}^{L} L_l\|\mathbf{A}_{t,l}\|_2\right)\gamma W^{-\beta}\alpha^{\frac{1-2\beta}{2}}.$$

□

C.2 Noise Stability

Theorem 4.2. *Denote $\Gamma_t = \max_{x\in\mathcal{D}_t}\|\mathbf{M}_t(x)\|_2$. Then, we can characterize the continual learning error between two subsequently trained models as*

$$\mathbb{E}\left[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2\right] \leq \Gamma_t(t'-t)\gamma\bar{\lambda}W^{-\beta}\alpha^{\frac{1-2\beta}{2}}\eta,$$

where $\eta = \left(\prod_{i=1}^{l}\kappa_i + \kappa_i(t'-t)\gamma\bar{\lambda}\mu_{t,i}\right)\left(\sum_{i=1}^{l}\kappa_i\right)$ and $\kappa_i = L_i c_i \mu_{i,t}$.

Proof. We will prove the induction hypothesis that

$$\|\mathbf{M}_{t,l}(x) - \mathbf{M}_{t',l}(x)\|_2 \leq \epsilon_l\|\mathbf{M}_{t,l}(x)\|_2,$$

where $\epsilon_l = \prod_i^l a_i\left(\sum_{i=1}^{l} b_i\right)$ where $a_i = c_i\mu_{i,t}L_i + \frac{L_i\mu_{i,l}\|\mathbf{U}_{i,l}\|_2 c_l}{\|\mathbf{A}_{i,l}[\mathcal{A}_{i,l}]\|_2}$ and $b_i = \frac{L_i\mu_{i,l}\|\mathbf{U}_{i,l}\|_2 c_l}{\|\mathbf{A}_{i,l}[\mathcal{A}_{i,l}]\|_2}$. The base case trivially holds given $\|\mathbf{M}_{t,0}(x) - \mathbf{M}_{t',0}(x)\|_2 = 0$. Here, $\mathbf{M}_{t,l}$ and $\mathbf{M}_{t',l}$ denote the models $\mathbf{M}_t$ and $\mathbf{M}_{t'}$ respectively with only the first l layers. We now perform our induction.

$$\begin{aligned}
\|\mathbf{M}_{t,l}(x) - \mathbf{M}_{t',l}(x)\|_2 &= \|(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}] - \mathbf{U}_{t,l})\phi_l(x_{t',l-1}) - \mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\phi_l(x_{t,l-1})\|_2 \\
&\leq \|(\mathbf{A}_{t,l}[\mathcal{A}_{t,l}](\phi_l(x_{t',l-1}) - \phi_l(x_{t,l-1})) - \mathbf{U}_{t,l}\phi_l(x_{t',l-1})\|_2 \\
&\leq \|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\|\phi_l(x_{t',l-1}) - \phi_l(x_{t,l-1})\|_2 + \|\mathbf{U}_{t,l}\|_2\|\phi_l(x_{t',l-1})\|_2 \\
&\leq L_l\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\|x_{t',l-1} - x_{t,l-1}\|_2 + L_l\|\mathbf{U}_{t,l}\|_2\|x_{t',l-1}\|_2 \qquad (8) \\
&\leq L_l\epsilon_{l-1}\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\|x_{t,l-1}\|_2 + (1+\epsilon_{l-1})L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (9) \\
&\leq c_l L_l\epsilon_{l-1}\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2\|\phi_l(x_{t,l-1})\|_2 + (1+\epsilon_{l-1})L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (10) \\
&\leq c_l\mu_{t,l}L_l\epsilon_{l-1}\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\phi_l(x_{t,l-1})\|_2 + (1+\epsilon_{l-1})L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2 \qquad (11) \\
&\leq c_l\mu_{t,l}L_l\epsilon_{l-1}\|x_{t,l}\|_2 + (1+\epsilon_{l-1})L_l\|\mathbf{U}_{t,l}\|_2\|x_{t,l-1}\|_2
\end{aligned}$$

Here, Equation (8) comes from the Lipschitz-Smoothness of the activation layers, Equation (9) comes from applying our induction hypothesis from the previous layer, Equation (10) comes from applying the activation contraction from Definition 4.2, Equation (11) comes from applying the layer cushion from Definition 4.1. We also note that the ratio outputs of two subsequential layers is bounded as in the following.

$$\begin{aligned}
\frac{\|x_{t,l-1}\|_2}{\|x_{t,l}\|_2} &\leq \frac{c_l\|\phi_l(x_{t,l-1})\|_2}{\|x_{t,l}\|_2} \qquad (12) \\
&\leq \frac{c_l\mu_{t,l}}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \qquad (13)
\end{aligned}$$

Here, Equation (12) come from applying the activation contraction from Definition 4.2 and Equation (13) comes from applying the layer cushion from Definition 4.1. Therefore, we have that

$$\begin{aligned}
\epsilon_l &\leq \frac{c_l \mu_{t,l} L_l \epsilon_{l-1} \|x_{t,l}\|_2 + (1+\epsilon_{l-1}) L_l \|\mathbf{U}_{t,l}\|_2 \|x_{t,l-1}\|_2}{\|x_{t,l}\|_2} \\
&\leq c_l \mu_{t,l} L_l \epsilon_{l-1} + \frac{(1+\epsilon_{l-1}) L_l \|\mathbf{U}_{t,l}\|_2 \|x_{t,l-1}\|_2}{\|x_{t,l}\|_2} \\
&\leq c_l \mu_{t,l} L_l \epsilon_{l-1} + \frac{(1+\epsilon_{l-1}) \mu_{t,l} L_l \|\mathbf{U}_{t,l}\|_2 c_l}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}
\end{aligned}$$

For mathematical ease, denote $a_l = c_l \mu_{t,l} L_l + \frac{L_l \mu_{t,l} \|\mathbf{U}_{t,l}\|_2 c_l}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}$ and $b_l = \frac{L_l \mu_{t,l} \|\mathbf{U}_{t,l}\|_2 c_l}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2}$. Thus, we have

$$\begin{aligned}
\epsilon_l &\leq a_l \epsilon_{l-1} + b_l \\
&\leq \prod_i^l a_i \sum_{i=1}^{l-1} b_i + b_l \\
&\leq \prod_i^l a_i \left(\sum_{i=1}^{l-1} b_i + b_l \right) \\
&\leq \prod_i^l a_i \left(\sum_{i=1}^{l} b_i \right)
\end{aligned}$$

We have thus proved our hypothesis. We will now simplify the bound. We have that

$$\begin{aligned}
\mathbb{E}[a_l] &= \mathbb{E}\left[c_l \mu_{t,l} L_l + \frac{L_l \mu_{t,l} \|\mathbf{U}_{t,l}\|_2 c_l}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \right] \\
&\leq c_l \mu_{t,l} L_l + L_l \mu_{t,l} (t'-t) \gamma \bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}} \qquad (14) \\
&\leq c_l \mu_{t,l} L_l + L_l c_l \mu_{t,l} (t'-t) \gamma \bar{\lambda}
\end{aligned}$$

Here, Equation (14) comes from Theorem 4.1. Similarly, we can bound b_l such that

$$\begin{aligned}
\mathbb{E}[b_l] &= \mathbb{E}\left[\frac{L_l \mu_{t,l} \|\mathbf{U}_{t,l}\|_2 c_l}{\|\mathbf{A}_{t,l}[\mathcal{A}_{t,l}]\|_2} \right] \\
&\leq L_l c_l \mu_{t,l} (t'-t) \gamma \bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}}
\end{aligned}$$

Therefore, putting it all together yields

$$\begin{aligned}
\mathbb{E}[\epsilon_l] &\leq \left(\prod_{i=1}^l c_i \mu_{t,i} L_i + L_i c_i \mu_{t,i} (t'-t) \gamma \bar{\lambda} \right) \left(\sum_{i=1}^l L_i c_i (t'-t) \gamma \bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}} \mu_{t,i} \right) \\
&\leq (t'-t) \gamma \bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}} \left(\prod_{i=1}^l c_i \mu_{t,i} L_i + L_i (t'-t) \gamma \bar{\lambda} \mu_{t,i} \right) \left(\sum_{i=1}^l L_i \mu_{t,i} \right)
\end{aligned}$$

We now have that $\forall x \in \mathcal{D}_t$,

$$\mathbb{E}[\|\mathbf{M}_t(x) - \mathbf{M}_{t'}(x)\|_2] \leq \max_{x \in \mathcal{D}_t} \|\mathbf{M}_t(x)\|_2 \cdot (t'-t) \gamma \bar{\lambda} W^{-\beta} \alpha^{\frac{1-2\beta}{2}} \left(\prod_{i=1}^l c_i \mu_{t,i} L_i + L_i c_i (t'-t) \gamma \bar{\lambda} \mu_{t,i} \right) \left(\sum_{i=1}^l L_i c_i \mu_{t,i} \right).$$

□