Appendix A. Supplementary Explanation

We analyzed why orthogonal constraints cannot solve catastrophic forgetting from the perspective of parameter space in the introduction section. In this section, we demonstrate this further mathematically. First we show in Section A.1 that restricting orthogonality between LoRA matrices still produces forgetting. In Section A.2, we further prove in terms of the loss function that orthogonal gradient descent cannot fully solve catastrophic forgetting.

Appendix A.1. Limiting orthogonality between LoRA matrices

Considering the scenario of sequential learning for two tasks, we assume that the model after learning the first task is represented as $f(Ax)$. If the second task is learned on the basis of $f(Ax)$, the model can be represented as $f((A + B)x)$. Where A is the weight of the first task, B is the weight of the second task. According to the distributive law, it is easy to derive the following equation $f((A + B)x) = f(Ax + Bx)$. Even if there is $A^T B = 0$, it does not mean that Bx is equal to the zero matrix, because B is not necessarily a zero matrix. Therefore, the condition of $A^T B = 0$ is not enough to guarantee $f(Ax) = f((A + B)x)$ without additional constraints. Because of the existence and uncertainty of Bx, the form of $f(x)$ may cause the two functions to produce different outputs. In addition, we also give more specific examples in this section. Furthermore, we conducted an empirical study in our experiment. In summary, learning the parameters of a new task in an orthogonal space is insufficient to solve the issue of catastrophic forgetting in LLM scenarios. The following is explained through specific examples.

Assume that in the classification task, the parameter matrices of tasks A and B are orthogonal to each other. We need to show that the predicted labels of the model after training tasks A and B sequentially will be biased compared to the predicted labels of training only task A.

Appendix A.1.1. One Dimensional Space

Consider the one-dimensional case, where $f(x) = \sin x$, $A = (1, 0)$, $B = (0, -1)$, satisfying $A^T B = 0$. When $x = \left(\frac{\pi}{2}, \pi\right)$, $\sin(Ax) = 1$, $\sin((A + B)x) = -1$. It can be seen that even if matrix B is orthogonal to A, the predicted label after adding matrix B will produce deviations.

Appendix A.1.2. Two-dimensional space

Let A be the matrix given by

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{A.1}$$

and let B be the matrix

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{A.2}$$

These matrices satisfy $A^T B = 0$. Now, consider the vector x given by

$$x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{A.3}$$

We find that Ax is

$$Ax = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{A.4}$$

and $(A + B)x$ is

$$(A + B)x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{A.5}$$

Define a function $f(x)$ as

$$f(x) = \begin{bmatrix} x_1 + x_2 \\ -x_2 \end{bmatrix}. \tag{A.6}$$

This function can also be represented as the product of matrices:

$$f(x) = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{A.7}$$

Now, let's evaluate $f(Ax)$:

$$f(Ax) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{A.8}$$

and $f((A + B)x)$ is

$$f((A + B)x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{A.9}$$

So in the two-dimensional case, there is also $f(Ax) \neq f((A + B)x)$.

Appendix A.1.3. N-dimensional space

Finally, it is extended to n dimensions, let A be the matrix given by

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{n \times n}, \tag{A.10}$$

and let B be the matrix given by

$$B = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}_{n \times n}. \tag{A.11}$$

If $A^T B = 0$, and we define x as

$$x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}_{n \times 1}, \tag{A.12}$$

then we have $Ax = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n \times 1}$ and $(A + B)x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}_{n \times 1}$.

Now, let's define a function $f(x)$ as

$$f(x) = \begin{bmatrix} x_1 + x_n \\ 0 \\ \vdots \\ 0 \\ -x_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & 0 & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}. \tag{A.13}$$

Therefore, we get $f(Ax) = \begin{bmatrix} 1 \\ \vdots \\ 0 \end{bmatrix}_{n\times 1}$ and $f((A+B)x) = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$.

So in the case of n dimensions, there is also $f(Ax) \neq f((A+B)x)$.

We can also find many similar examples. From the above, we can conclude that adding a new parameter matrix that is orthogonal to the previous parameter matrix will bias the prediction. In other words, using multiple mutually orthogonal LoRA matrices to learn sequence tasks in a continuous learning scenario cannot avoid catastrophic forgetting.

Appendix A.2. Analyzing orthogonal gradient descent from the perspective of the loss function

The essence of the forgetting problem lies in whether the loss of previous tasks will change during parameter updates. Therefore, we need to prove that when learning new tasks under orthogonal constraints, there is still the possibility of changes in the loss of previous tasks.

Appendix A.2.1. Assumptions and Settings

We assume that after the model learns the first task, the parameters converge to the optimal solution of the first task, and then it learns the second task.

- The loss function of task T_1 is $L_1(\theta)$, and the loss function of task T_2 is $L_2(\theta)$.
- Assume that on task T_1, the parameter θ^* is the optimal solution, that is:

$$\nabla L_1(\theta^*) = 0 \tag{A.14}$$

- The gradient of task T_2 is $g_2(\theta)$.
- When using orthogonal gradient descent, the parameter update rule is:

$$\theta_{t+1} = \theta_t - \eta \cdot g_2^{\perp}(\theta_t) \tag{A.15}$$

  Here, $g_2^{\perp}(\theta_t)$ is the component of the gradient $g_2(\theta_t)$ in the direction orthogonal to $g_1(\theta_t)$.

To simplify, the orthogonal gradient component is defined as follows:

$$g_2^{\perp}(\theta) = g_2(\theta) - \frac{g_2(\theta) \cdot g_1(\theta)}{\|g_1(\theta)\|^2} g_1(\theta) \tag{A.16}$$

Here, $\cdot$ represents the dot product.

Appendix A.2.2. Proof

Starting from the gradient update rule:

$$\theta_{t+1} = \theta_t - \eta \cdot g_2^{\perp}(\theta_t) \tag{A.17}$$

First, calculate the first-order Taylor expansion of $L_1(\theta_{t+1})$:

$$L_1(\theta_{t+1}) \approx L_1(\theta_t) + \nabla L_1(\theta_t)^T(\theta_{t+1} - \theta_t) \tag{A.18}$$

Since $\nabla L_1(\theta_t) = 0$ at θ_t (the optimal solution on task T_1), then:

$$L_1(\theta_{t+1}) \approx L_1(\theta_t) + 0 = L_1(\theta_t) \tag{A.19}$$

This indicates that the first-order Taylor expansion of $\nabla L_1(\theta_t)$ is zero.

To capture more detailed changes, the second-order effect needs to be considered. Calculate the second derivative (Hessian matrix) of the loss function L_1 at θ_{t+1}:

$$\frac{\partial^2 L_1(\theta)}{\partial \theta^2} = H_1(\theta) \tag{A.20}$$

$$L_1(\theta_{t+1}) \approx L_1(\theta_t) + \frac{1}{2}(\theta_{t+1} - \theta_t)^T H_1(\theta_t)(\theta_{t+1} - \theta_t) \tag{A.21}$$

Substituting $\theta_{t+1} - \theta_t = -\eta \cdot g_2^{\perp}(\theta_t)$, we get:

$$L_1(\theta_{t+1}) \approx L_1(\theta_t) + \frac{1}{2}\eta^2 (g_2^{\perp}(\theta_t))^T H_1(\theta_t)(g_2^{\perp}(\theta_t)) \tag{A.22}$$

Here, $(g_2^{\perp}(\theta_t))^T H_1(\theta_t)(g_2^{\perp}(\theta_t))$ may not be zero, even if $g_2^{\perp}(\theta_t)$ is updated in the direction orthogonal to $g_1(\theta_t)$. The Hessian matrix $H_1(\theta_t)$ may have nonzero components in the direction of $g_2^{\perp}(\theta_t)$, so the loss function L_1 will increase.

In conclusion, even when using orthogonal gradient descent (OGD) for gradient updates, forgetting may still be encountered on task T_1. Because during the gradient update, although the update is in the orthogonal direction, the second-order effect of the loss function (through the Hessian matrix) may cause the loss of the previous task to rise. This is because orthogonal gradient descent cannot completely control all directions that may cause the loss of the previous task to rise, especially in higher-dimensional feature spaces, such as LLMs with larger parameter spaces.

Appendix B. Other settings of the experiments

Appendix B.1. Details of The Training Tasks

In this part, we present the specifics of the tasks as well as the datasets within our experiments.

Table B.1 shows the details of all 15 datasets in the standard CL benchmark and the Large number of tasks. In general, we adopted 5 datasets from CL benchmark [5], 4 datasets from GLUE [3] and 6 datasets from SuperGLUE [2] benchmarks, following [1].

Following [4], we present the task sequences arranged in our experiments involving T5-Large and LLaMA2-7B models in Table B.2. Furthermore, we use the instruction tuning method, so we have corresponding instructions for each task. Prompts for different tasks are shown in Table B.3. Natural language inference (NLI), which encompasses MNLI, RTE and CB, is one of the task categories. Sentiment analysis (SC), covering Amazon, Yelp, SST-2 and IMDB, is another task category. And topic classification (TC), involving AG News, DBpedia and Yahoo, is also among the task categories.

Appendix B.2. Additional computation of AM-LoRA

Similar to O-LoRA, AM-LoRA increase the number of LoRA's parameters by one for each new task trained. In addition, AM-LoRA adds an additional Attentional Selector of size 4096*1. For example, in LLaMA2-7B, the AM-LoRA method adds an additional 4096*2*32 number of trainable senators, which is 0.00003 of the total number of senators, and this amount can be considered as minimal.

Appendix B.3. Implementation Details

With respect to all orders of task streams, we carried out the training of the models for one epoch, setting a learning rate at 0.0001, a batch size of 1 per GPU in LLaMA2-7B, a batch size of 8 per GPU in T5-Large, a weight decay rate of 0, and a dropout rate of 0.1.

The values of λ are not the same across orders 1 to 6. For order 1,2,3 in T5-Large and LLaMA2-7B, we set λ = 0.00001, 0.00001, 0.00001, 0.00001. For every task in order 4(SST-2, Agnews, IMDB, Yelp, Yahoo, MultiRC, DBpedia, COPA, WiC, RTE, MNLI, QQP, BoolQA, CB), we define λ = 0, 0.01, 0.01, 0.01, 0.01, 0.01, 0.3, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001. For order 5(IMDB, SST-2, Yahoo, DBpedia, Agnews, QQP, MNLI, RTE, Yelp, Amazon, CB, COPA, BoolQA, MultiRC, WiC), we set λ = 0.0001, 0.0001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.01, 0.01, 0.01, 0.01, 0.01. For order 6(COPA, IMDB, SST-2, Agnews, Yahoo, MultiRC, DBpedia, QQP, WiC, RTE, MNLI, Yelp, Amazon, CB, BoolQA), we define λ = 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001 respectively.

Dataset	Benchmark	Task	Domain	Metric
Yelp	CL Benchmark	Sentiment Analysis	Yelp Reviews	Accuracy
Amazon	CL Benchmark	Sentiment Analysis	Amazon Reviews	Accuracy
DBpedia	CL Benchmark	Topic Classification	Wikipedia	Accuracy
Yahoo	CL Benchmark	Topic Classification	Yahoo Q&A	Accuracy
AG News	CL Benchmark	Topic Classification	News	Accuracy
MNLI	GLUE	NLI	Various	Accuracy
QQP	GLUE	Paragraph Detection	Quora	Accuracy
RTE	GLUE	NLI	News, Wikipedia	Accuracy
SST-2	GLUE	Sentiment Analysis	Movie Reviews	Accuracy
WiC	SuperGLUE	Word Sense Disambiguation	Lexical Databases	Accuracy
CB	SuperGLUE	NLI	Various	Accuracy
COPA	SuperGLUE	QA	Blogs,Encyclopedia	Accuracy
BoolQA	SuperGLUE	Boolean QA	Wikipedia	Accuracy
MultiRC	SuperGLUE	QA	Various	Accuracy
IMDB	SuperGLUE	Sentiment Analysis	Movie Reviews	Accuracy

Table B.1: The details of 15 datasets utilized in our CL experiments. Natural language inference is denoted as NLI, and questions and answers task is denoted as QA. Among these tasks, the first five are in line with the standard CL benchmark, and the remaining ones are employed in long-sequence experiments.

Order	Model	Task Sequence
1	T5-Large, LLaMA2-7B	dbpedia → amazon → yahoo → ag
2	T5-Large, LLaMA2-7B	dbpedia → amazon → ag → yahoo
3	T5-Large, LLaMA2-7B	yahoo → amazon → ag → dbpedia
4	T5-Large	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
5	T5-Large	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
6	T5-Large	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table B.2: Six different orders of task sequences utilized in the continual learning experiments. Orders 1,2,3 correspond to the standard CL benchmark adopted by prior works. Orders 4,5,6 act as long-sequence orders that span 15 tasks and follow [23].

Task	Prompts
MultiRC	According to the following passage and question, is the candidate answer true or false? Choose one from the option.
QQP	Whether the "first sentence" and the "second sentence" have the same meaning? Choose one from the option.
WiC	Given a word and two sentences, whether the word is used with the same sense in both sentences? Choose one from the option.
TC	What is the topic of the following paragraph? Choose one from the option.
BoolQA	According to the following passage, is the question true or false? Choose one from the option.
SC	What is the sentiment of the following paragraph? Choose one from the option.
NLI	What is the logical relationship between the "sentence 1" and the "sentence 2"? Choose one from the option.

Table B.3: Prompts for all tasks.

References

[1] Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., Almahairi, A.: Progressive prompts: Continual learning for language models. arXiv preprint arXiv:2301.12314 (2023)

[2] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems **32** (2019)

[3] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)

[4] Wang, X., Chen, T., Ge, Q., Xia, H., Bao, R., Zheng, R., Zhang, Q., Gui, T., Huang, X.: Orthogonal subspace learning for language model continual learning. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 10658–10671. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.findings-emnlp.715, `https://aclanthology.org/2023.findings-emnlp.715`

[5] Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Advances in neural information processing systems **28** (2015)