A Appendix

A.1 Proof for Theorem 1

Proof. Let us first consider $\mathbf{\Theta}_0 = 0$, and $\mathbf{\Theta}$ is updated using gradient descent. When the problem is in continuous time (gradient flow) and $\mathbf{\Theta}_0 = 0$, then α is in fact trivial, it doesn't matter to the problem since the PDE is homogeneous.

However in discrete time, the dynamics of the problem is dependent on the learning rate η, then at any given iteration, suppose for two dynamical systems that satisfy $\alpha\mathbf{\Theta} = \alpha'\mathbf{\Theta}'$ in the current iteration, we proof that for the next iteration, there exist η' such that $\alpha\mathbf{\Theta}^+ = \alpha'\mathbf{\Theta}'^+$. Since we have $\mathbf{\Theta}_0 = \mathbf{\Theta}'_0$ at initialization, the theorem is proved by induction.

Suppose for the new scaling factor $\alpha' = k\alpha$, where k is a constant, then,

$$\mathbf{\Theta}' = \mathbf{\Theta}/k, \frac{d}{d\mathbf{\Theta}'}f(\alpha'\mathbf{\Theta}') = k\frac{d}{d\mathbf{\Theta}}f(\alpha\mathbf{\Theta}),$$

since

$$\mathbf{\Theta}'^+ = \mathbf{\Theta}' - \eta'\frac{d}{d\mathbf{\Theta}'}f(\alpha'\mathbf{\Theta}'),$$
$$\mathbf{\Theta}^+ = \mathbf{\Theta} - \eta\frac{d}{d\mathbf{\Theta}}f(\alpha\mathbf{\Theta})$$

Then we can find $\eta' = \eta/k^2$ such that for the next iteration, $\alpha\mathbf{\Theta}^+ = \alpha'\mathbf{\Theta}'^+$.

For this optimization problem, the introduction of α is trivial and can be replaced with appropriate choices of learning rate η.

□

A.2 LLaMA exp examples and details

We sample some generated answers from the log of our GSM8K experiments in Table 6, with LLaMA and LoRA or FFA-LoRA.

A.3 A motivation for further reducing trainable parameters

Our approach named FFA-LoRA in Section 4 exhibits a number of theoretical benefits compared to LoRA, additionally, it also performs better and is more consistent as shown in Section 5. We can conclude that for PEFT with adapters, by freezing randomly initialized parameters and only train on the set of parameters that were initialized at 0 is a valid and practical approach. This guided us to get an even more aggressive construction of adapters, which we refer to as QVP Adapters, formulated as below.

For a weight matrix $\boldsymbol{W} \in \mathbb{R}^{d\times k}$, we consider the model update to be projected to a low-rank matrix such that

$$\boldsymbol{W} = \boldsymbol{W}_0 + \Delta\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{Q}_0\boldsymbol{V}\boldsymbol{P}_0,$$

where $\boldsymbol{Q}_0 \in \mathbb{R}^{d\times r}, \boldsymbol{V} \in \mathbb{R}^{r\times r}, \boldsymbol{P}_0 \in \mathbb{R}^{r\times k}$. Similar to FFA-LoRA, $\boldsymbol{W}_0$ is the pre-trained weight, and $\boldsymbol{P}_0, \boldsymbol{Q}_0$ follows a random Gaussian initialization. We consider $\boldsymbol{V}$ trainable and start with $\boldsymbol{V}_0 = 0$, $\boldsymbol{W}_0, \boldsymbol{Q}_0, \boldsymbol{P}_0$ are kept frozen throughout the training process. We provide the performance of QVP adapters below in Table 7, and compare with LoRA and FFA-LoRA.

For the experiments where these algorithms have the same parameter budget ($r = 64$ for QVP versus $r = 4$ for FFA-LoRA, etc.), QVP do not perform as good as the previously mentioned algorithms. But a unique advantage offered by QVP adapters is that it is possible to even further reduce the number of trainable parameters, and the algorithm is still able to learn meaningful features from data. The same is impossible for LoRA and FFA-LoRA since the rank r can not be smaller than 1 for these methods. Therefore, QVP is potentially useful in the case where the parameter budget is extremely constrained, such as local private training in mobile devices.

Question:	A new program had 60 downloads in the first month. The number of downloads in the second month was three times as many as the downloads in the first month, but then reduced by 30% in the third month. How many downloads did the program have total over the three months?
FFA-LoRA:	The number of downloads in the first month was 60. The number of downloads in the second month was 3 times as many as the first month, so that is 3 * 60 = 180. The number of downloads in the third month was 180 - 30% = 180 - 54 = 126. The total number of downloads over the three months is 60 + 180 + 126 = 366. The answer is 366.
LoRA:	The number of downloads in the first month was 60. The number of downloads in the second month was 60 x 3 = 180. The number of downloads in the third month was 180 - 30 = 150. The answer is 150.
Question:	John decides to get a loan by mortgaging his home. His house is worth $250,000. He gets a loan worth 40% of that. He uses 60% of that to pay off his debts. How much money did he have leftover after paying debt?
FFA-LoRA:	John's house is worth 250,000 dollars. He gets a loan worth 250,000 x 40% = 100,000 dollars. He uses 100,000 x 60% = 60,000 dollars to pay off his debts. 100,000 - 60,000 = 40,000 dollars. The answer is 40,000.
LoRA:	John's house is worth 250,000 dollars. 40% of that is 250,000 x 0.4 = 100,000 dollars. 100,000 dollars is the amount of the loan. 100,000 / 0.6 = 166,666 dollars. 166,666 - 100,000 = 66,666 dollars. 66,666 dollars is the amount of money he has left.
Question:	Octavia wants to make a quilt for her sister. She is going to build it from small, medium, and large t-shirts that her family is done with. A small t-shirt is 3 square feet of fabric. A medium one is 4 square feet. A large one is 6 square feet. If she uses 11 small shirts, 8 medium, and 6 large shirts, how many square feet is the quilt?
FFA-LoRA:	A small shirt is 3 square feet. A medium shirt is 4 square feet. A large shirt is 6 square feet. 11 small shirts is 11 x 3 = 33 square feet. 8 medium shirts is 8 x 4 = 32 square feet. 6 large shirts is 6 x 6 = 36 square feet. 33 + 32 + 36 = 101 square feet. The answer is 101.
LoRA:	Octavia is going to use 11 small shirts, 8 medium, and 6 large shirts. 11 + 8 + 6 = 25. 3 square feet is 12 square feet. 25 * 12 = 300 square feet. The answer is 300 square feet.

Table 6: Example of generated answer for GSM8K datasets

Method	# of params	acc w/o DP	acc@$\epsilon = 6$	acc@$\epsilon = 3$
LoRA (rank 16)	3145728 (0.877%)	92.49%	86.87%	86.23%
LoRA (rank 4)	786432 (0.220%)	91.40%	85.2%	85.35%
LoRA fix A (rank 16)	1572864 (0.440%)	92.49%	87.333 %	86.3628%
LoRA fix A (rank 4)	393216 (0.110%)	92.20%	86.7472%	86.2164%
QVP (rank 128)	1572864 (0.412%)	90.46%	84.23%	83.16%
QVP (rank 64)	393216 (0.107%)	90.17%	86.41%	84.44%
QVP (rank 32)	98304 (0.0272%)	87.31%	85.69%	84.31%
QVP (rank 16)	24576 (0.00685%)	83.40%	84.44%	83.67%

Table 7: Comparison between LoRA, FFA-LoRA and QVP adapters, including number of trainable parameters.