

A HYPERPARAMETERS

In Table 1, we provide the hyperparameters used for the GLUE benchmark in the main paper. Note that due to our academic compute we were not able to run full grid searches on any hyperparameters. We only evaluated different learning rates and even relied on existing configurations of LoRA. Table 2 shows the hyperparameter of cross-modal image-text retrieval task, Table 3 shows the hyperparameter of instruction tuning. Table 4 shows the hyperparameter of image classification. In this task, for SeRA, the specific learning rates are set to 4e-3, 4e-3, 2e-3 and 1e-3, respectively, corresponding to the sizes [8, 8], [128, 128], [32, 256] and [8, 256] in the “full” mode. Additionally, the learning rate is set to 4e-3 in “medium” mode and 6e-3 in “easy” mode.

Table 1: Hyperparameter configurations for different model sizes on GLUE benchmark. Optimizer, Warmup Ratio, Epochs, Batch Size, Max Sequence Length and LR Schedule are taken from Hu et al. (2022).

| Model | Hyperparameter | SST-2 | MRPC | CoLA | QNLI | RTE | STS-B |
|-------|-----------------------|-----------------|-------|-------|-------|-------|-------|
| Base | Optimizer | AdamW | | | | | |
| | Warmup Ratio | 0.06 | | | | | |
| | LR Schedule | Linear | | | | | |
| | Init of A, C matrices | Kaiming Uniform | | | | | |
| | Batch Size | 16 | 16 | 32 | 32 | 32 | 16 |
| | Epochs | 60 | 30 | 80 | 25 | 80 | 40 |
| Large | Learning Rate | 5E-04 | 4E-04 | 4E-03 | 4E-04 | 5E-03 | 4E-04 |
| | SeRA Config | r=split=8 | | | | | |
| | α | 16 | | | | | |
| | Max Seq. Len. | 512 | | | | | |
| | Batch Size | 4 | 4 | 4 | 4 | 8 | 8 |
| | Epochs | 10 | 20 | 20 | 10 | 20 | 30 |
| Large | Learning Rate | 4E-04 | 3E-04 | 2E-04 | 2E-04 | 4E-04 | 2E-04 |
| | SeRA Config | r=split=8 | | | | | |
| | α | 16 | | | | | |
| | Max Seq. Len. | 128 | 512 | 128 | 512 | 512 | 512 |
| | Batch Size | 4 | 4 | 4 | 4 | 8 | 8 |
| | Epochs | 10 | 20 | 20 | 10 | 20 | 30 |
| Large | Learning Rate | 4E-04 | 3E-04 | 2E-04 | 2E-04 | 4E-04 | 2E-04 |
| | SeRA Config | r=split=16 | | | | | |
| | α | 32 | | | | | |
| | Max Seq. Len. | 128 | 512 | 128 | 512 | 512 | 512 |
| | Batch Size | 4 | 4 | 4 | 4 | 8 | 8 |
| | Epochs | 10 | 20 | 20 | 10 | 20 | 30 |

B GPU MEMORY CONSUMPTION

We report the GPU memory and the number of fine-tuned parameters for SeRA and VeRA fine-tuned on the ViT model in Table 5, and the results show that VeRA requires additional GPU memory.

SeRA has significant advantages over VeRA in terms of GPU memory efficiency, computational efficiency and initialisation sensitivity. We present more experimental details about SeRA and VeRA in Table 6 which shows various experimental metrics fine-tuned for the ViT model in “full” mode on the RSCD.

Firstly, in VeRA, although the A and B matrices are frozen, these randomly initialised matrices still need to be stored in the GPU. The experimental results show that VeRA requires about 32GB of memory for every one million fine-tuned parameters, while SeRA requires only 0.008GB of memory, which is 4000 times more than SeRA, and this difference is particularly significant in the context of large-scale models. In addition, the frozen A and B matrices will be involved in the computation of the forward and back propagation, which will increase the training time and make the computation less efficient than SeRA.

Table 2: Hyperparameter configurations for all methods on the MSCOCO dataset, for CLIP Base and Large models.

| | Hyperparameter | FT | LoRA | SeRA&MELoRA | VeRA |
|-------|----------------|------|------|----------------------|------|
| BASE | Optimizer | | | SGD | |
| | Momentum | | | 0.9 | |
| | Weight_decay | | | 0.1 | |
| | LR Schedule | | | Linear | |
| | Target_module | | | ['q-proj', 'v-proj'] | |
| | Batch Size | | | 128 | |
| | Epochs | | | 1 | |
| | Warmup Step | | | 400 | |
| | Learning Rate | 2e-5 | 1e-3 | 1e-3 | 1e-2 |
| | Dropout | - | 0.15 | 0.15 | - |
| LARGE | r[text] | - | 8 | 8 | 256 |
| | r[vision] | - | 8 | 8 | 512 |
| | α | - | 16 | 16 | - |
| | Batch Size | | | 32 | |
| | Epochs | | | 1 | |
| | Warmup Step | | | 6000 | |
| | Learning Rate | 1e-5 | 5e-4 | 5e-4 | 5e-3 |
| | Dropout | - | 0.15 | 0.15 | - |
| | r[text] | - | 8 | 8 | 256 |
| | r[vision] | - | 8 | 8 | 512 |
| | α | - | 16 | 16 | - |

Table 3: Hyperparameter configurations for all methods on the Alpaca dataset, for LLaMA3 model.

| Hyperparameter | LoRA & SeRA &MELoRA &MoSLORA |
|--------------------|------------------------------|
| GPUs | 1 |
| Optimizer | AdamW |
| Warmup Ratio | 0.1 |
| LR Schedule | Cosine |
| Dropout | 0 |
| Target_module | Q, K, V, O, Up, Down, Gate |
| Batch Size | 8 |
| Accumulation Steps | 2 |
| Epochs | 1 |
| Learning Rate | 4e-4 |

Table 4: Hyperparameter configurations for all methods on the RSCD with different modes, for ViT model.

| Hyperparameter | Full | Medium | Easy |
|----------------|----------------|--------|------|
| Optimizer | AdamW | | |
| Warmup Ratio | 0.06 | | |
| LR Schedule | Linear | | |
| Dropout | 0.1 | | |
| Target_module | [query, value] | | |
| Weight_decay | 0.01 | | |
| Epochs | 10 | | |
| Batch Size | 32 | | |
| LR_VeRA | 4e-2 | | |
| LR_VeRA_Head | 2e-3 | | |
| LR_LoRA | 4e-4 | | |
| LR_LoRA_Head | 4e-3 | | |
| LR_FT | 4e-5 | | |
| LR_FT_Head | 4e-3 | | |
| LR_Head | 4e-4 | | |
| LR_SeRA_Head | 4e-3 | | |
| LR_SeRA | 1e-3 | | |
| LR_MELoRA_Head | 4e-3 | | |
| LR_MELoRA | 4e-4 | | |

Table 5: GPU memory requirement of SeRA and VeRA methods.

| Method | Rank | GPU Memory | Trainable Parameters |
|--------|------|------------|----------------------|
| SeRA | 8 | 10797Mb | 0.1M |
| VeRA | 1024 | 12217Mb | 0.1M |

Table 6: Computational consumption and performance of each method in “full” mode of RSCD dataset

| Method | Rank | GPU Memory | Trainable Parameters | Train time | Accuracy |
|--------|--------------|------------|----------------------|------------|----------|
| FT | - | 16G | 303M | 255min | 84.9 |
| SeRA | 256(split32) | 11.53G | 3.9M | 244min | 83.6 |
| SeRA | 256(split8) | 11.55G | 6.3M | 210min | 83.7 |
| VeRA | 1024 | 12.2G | 0.1M | 263min | 81.6 |
| VeRA | 4096 | 17G | 0.25M | 320min | 82.3 |

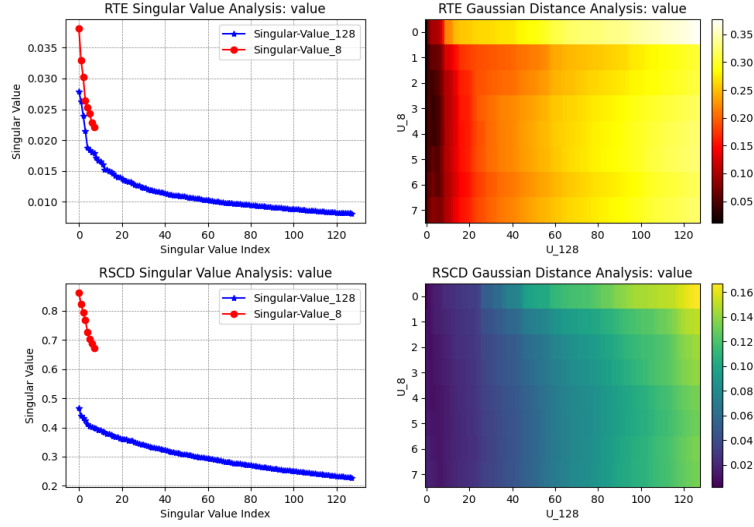


Figure 1: The left side of the image shows the singular values in descending order of size, and the right one shows the singular vector subspace similarity between $U_{r=8}$ and $U_{r=128}$.

C SVD ANALYSIS

In the section on singular value decomposition for matrices, we show only the average results for the query layer, Figure 1 shows the results of the analysis for the value layer. Results lead to the same conclusions as the main text.

D A MORE DETAILED RESEARCH OF THE IMAGE CLASSIFICATION TASK

We selected one image from each category in the RSCD validation set to show them together in Figure 2. Looking closely at the images for each category, we see that these images are not common. Since the dataset used for pre-training the model rarely contains such images, the ViT model, which has been pre-trained using the imagenet-21k dataset, gives poor results on RSCD when only the classification header is trained. However, when we perform full parameter fine-tuning, the performance improves significantly and outperforms even all PEFT methods on “full” mode. The reason is that the training process is more about memorising new features and learning the details of new images, rather than just fine-tuning the pre-trained model’s existing knowledge. Therefore, we need more trainable parameters to capture the nuances between different categories.

We conducted additional experiments, choosing more common datasets, and obtained different results compared to the above. We chose the datasets CIFAR100 (Krizhevsky, 2009), Food101 (Bossard et al., 2014), and RESISC45 (Cheng et al., 2017). The large version of Vision Transformer, which was pre-trained on 21K imagenet, was chosen as the pre-trained model. Each task was trained using the full training set of images, and all experiments were tested on the corresponding test set after 10 epochs of training. For all tuning methods, the classification head was fully adjusted and excluded when calculating the number of parameters, only the query and value layers are fine-tuned. The rank is 8 for LoRA and SeRA, 256 for VeRA. Full parameter fine-tuning and training only the classification head were chosen as baseline.

Table 7 shows results. All the fine-tuning methods have achieved similar performance to the full-parameter fine-tuning. It is worth noting that fine-tuning the header of the classification alone can yield good results, indicating that the pre-trained model already has some ability to recognise these images. And these results cannot reflect the performance difference between different methods. Thus, we performed additional experiment through freezing the randomly initialized classification

Table 7: ViT finetuned with VeRA, LoRA and SeRA on different image classification datasets.

| Method | #Trainable Parameters | CIFAR100 | Food101 | RESISC45 |
|--------|--------------------------|----------|---------|----------|
| Head | – | 88.7 | 85.7 | 90.1 |
| FT | 303.4M | 93.0 | 90.0 | 96.8 |
| LoRA | 0.79M | 93.0 | 89.5 | 95.7 |
| VeRA | 0.06M | 93.0 | 89.4 | 95.7 |
| SeRA | 0.1M | 92.8 | 88.9 | 95.4 |

head weights to exclude the effect of the classification head, which aims to conduct a pure comparison for different fine-tuning methods. Freezing the classification head means that the classification boundary will be randomly fixed and cannot be adjusted, which requires the fine-tuning method to have a stronger ability to adjust the model output.

For LoRA we use the rank of [4, 8], for VeRA we use the rank of [256, 512, 1024], and for SeRA we use the rank of [8, 16, 32, 64, 128]. The experimental results of the image classification task in Figure 3 show that increasing the number of training parameters can improve the model performance more significantly. We obtained similar conclusions to the RSCD experiments. Further proof of SeRA’s scalability and efficiency. And we also revealed the reason for the difference performance of the fine-tuning method compared to full-parameter fine-tuning in specific task contexts.



Figure 2: RSCD validation set of different categories with category names above the subimages.

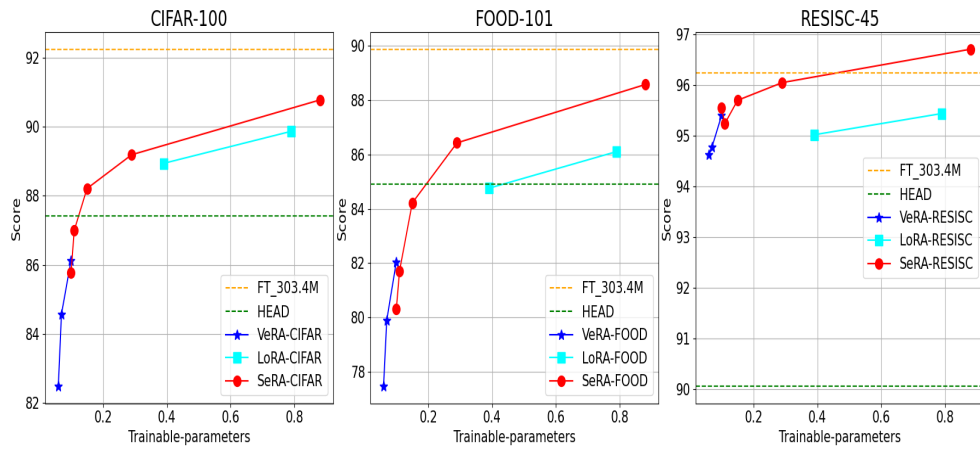


Figure 3: The left image is the result of fine-tuning on the CIFAR-100 dataset, the middle image is the result of fine-tuning on the FOOD-101 dataset, and the right image is the result of fine-tuning on the RESISC-45 dataset, and SeRA performs better in each dataset with the same number of parameters.

E EXPERIMENTAL DETAILS ON INSTRUCTION TUNING

The cleaned Alpaca dataset is used as train set. We used all data and trained only one epoch for all methods. Following INSTRUCTEVAL (Chia et al., 2023), we use 5-shot direct prompting for MMLU, 3-shot direct prompting for BBH, 3-shot direct prompting for DROP (dev), and 0-shot direct prompting for HEval. During training, we use AdamW as the optimizer.

REFERENCES

- Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.
- Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, Oct 2017. ISSN 1558-2256. doi: 10.1109/jproc.2017.2675998. URL <http://dx.doi.org/10.1109/JPROC.2017.2675998>.
- Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. Instructeval: Towards holistic evaluation of instruction-tuned large language models, 2023. URL <https://arxiv.org/abs/2306.04757>.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, no, 2009.