SWIFT: A Scalable Lightweight Infrastructure for Fine-Tuning

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, Yingda Chen

ModelScope Team, Alibaba Group

{yuze.zyz, huangjintao.hjt, hujinghan.hjh, xingjun.wxj, maoyunlin.myl, zhangdaoze.zdz, zeyinzi.jzyz, wuzhikai.wzk, baole.abl, wangang.wa, wenmeng.zwm, yingda.chen}@alibaba-inc.com,

Abstract

Recent development in Large Language Models (LLMs) and Multi-modal Large Language Models (MLLMs) have achieved superior performance and generalization capabilities, covered extensive areas of traditional tasks. However, existing large model training frameworks support only a limited number of models and techniques, particularly lacking in support for new models, which makes fine-tuning LLMs challenging for most developers. Therefore, we develop SWIFT, a customizable one-stop infrastructure for large models. With support of over 350+ LLMs and 80+ MLLMs, SWIFT stands as the open-source framework that provide the *most comprehensive support* for fine-tuning large models. In particular, it is the first training framework that provides systematic support for MLLMs. Moreover, SWIFT integrates post-training processes such as inference, evaluation, and quantization, to facilitate fast adoptions of large models in various application scenarios, offering helpful utilities like benchmark comparisons among different training techniques.

Code — https://github.com/modelscope/ms-swift

Introduction

As the number of parameters and memory consumption of large models continue to rise, researchers have proposed numerous resource-efficient training techniques, such as LoRA (Hu et al. 2021), rsLoRA (Kalajdzievski 2023), DoRA (Liu et al. 2024b), and LLaMA-Pro (Wu et al. 2024). Additionally, quantization stands as another solution for reducing memory usage during training. Due to the vast differences and poor adaptability to different models of these techniques, efforts begin to emerge to unify these training interfaces. As a notable example, Hugging Face[1] introduced the PEFT[2], aiming at standardizing interfaces for efficient fine-tuning algorithms; and for quantization, it also developed the Optimum[3] library as a unified framework for various quantization methods. However, fine-tuning large models still remains formidable for most developers, mainly because these aforementioned solutions support only a limited number of models and techniques, and especially lack support for new models. Furthermore, to effectively ensure the model deployment, the post-training, such as inference and evaluation, are also steps in utilization of large models.

To address these issues, we present SWIFT, an open-source framework to facilitate lightweight training and post-training of large models, with the following properties:

- **Comprehensiveness.** SWIFT supports the pre-training, fine-tuning, and human alignment for 350+ LLM models and 80+ MLLM models, including all major open-source models and 160+ pure text and multi-modal datasets. Besides Transformer-based models, other model structures such as Mamba (Gu and Dao 2024) are also supported.
- **Flexible Usage.** To enhance lightweight training, we implement or plant several SOTA tuners in SWIFT, which are designed to be used independent of our SWIFT training loop to allow more flexible usage.
- **Full-chain Capability.** SWIFT also integrate numerous post-training processes, including quantization, LoRA merging, evaluation (140+ pure text and multi-modal evaluation sets), as well as inference and deployment capabilities against most LLMs and MLLMs.

To our knowledge, SWIFT is the most comprehensive training framework and end-to-end solution for large models so far, with the widest support for models and datasets (especially for MLLMs) and the fullest-chain capabilities.

Related Works

As the most popular instance of versatile training frameworks for large models, LLaMA-Factory (Zheng et al. 2024b) supports the training and human alignment of 100+ text LLMs. However, LLaMA-Factory support only a few multi-modal models, including LLaVA (Liu et al. 2024a), PaliGemma (Beyer et al. 2024), YI-VL (AI et al. 2024), and Qwen2-VL (Yang et al. 2024), which presents a notable limitation compared to SWIFT's support for 80+ MLLMs. The comparisons between other existing works, including Firefly[4], FastChat (Zheng et al. 2024a), Axolotl[5], and LMFlow (Diao et al. 2023), and SWIFT are given in Tab. 1.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]https://huggingface.co and https://github.com/huggingface

[2]https://github.com/huggingface/peft

[3]https://github.com/huggingface/optimum

[4]https://github.com/yangjianxin1/Firefly

[5]https://github.com/axolotl-ai-cloud/axolotl

	LLaMA-Factory	FireFly	FastChat	Axolotl	LMFlow	**SWIFT (ours)**
LoRA, QLoRA	✓	✓	✓	✓	✓	✓
LLaMA-Pro	✓					✓
LongLoRA	✓	✓		✓		✓
GaLore	✓			✓		✓
Q-GaLore				✓		✓
FourierFt						✓
LoRA+	✓			✓		✓
LISA				✓	✓	✓
DoRA, rsLoRA	✓			✓		✓
UnSloth	✓	✓		✓		✓
LLM-Pretrain	✓	✓	✓	✓	✓	✓
LLM-Pretrain (Megatron)						✓
LLM-SFT	✓	✓	✓	✓	✓	✓
LLM-DPO	✓	✓		✓	✓	✓
LLM-CPO				✓		✓
LLM-ORPO	✓			✓		✓
LLM-KTO	✓			✓		✓
LLM-SimPO	✓			✓		✓
MLLM-Pretrain	4					50+
MLLM-SFT	4					50+
MLLM-RLHF	4					50+
vLLM	✓		✓		✓	✓
LMDeploy						✓
LLM Evaluation	4		✓		✓	50
MLLM Evaluation						95
WEB-UI	✓		✓			✓

Table 1: The comparison of supported training capabilities.

Design and Implementations

In this section, we introduce the architecture design of our SWIFT framework (shown in Fig. 1).

Models. The model module includes a basic model loader that allows for flexible customization of model configurations. The patcher module is used to address various compatibility issues, ensuring smooth operation in different scenarios like single-GPU, multi-GPU, full-parameter, or LoRA training. The template module ensures that various models can correctly convert the actual coordinate values of the data into the coordinate values required by models.

Datasets. The dataset module can load data by three ways: from ModelScope[6], Hugging Face, and user-customized datasets. It can also pre-process the input data, including converting different datasets into a standard format.

Tuners. Tuners can incorporate various tuning techniques in PEFT library to ensure compatibility and seamless operation. Additionally, SWIFT supports a much wider range of tuners than those in PEFT, including SCEdit (Jiang et al. 2023b), ResTuning (Jiang et al. 2023a), LLaMA-Pro (Wu et al. 2024), LongLoRA (Chen et al. 2024), and LISA (Pan et al. 2024). These tuners can be used in combination and support offloading of deactivated tuners to CPU or meta devices. This integration allows tuners to be applied to not only models supported within SWIFT, but also external models.

[6]https://modelscope.cn and https://github.com/modelscope

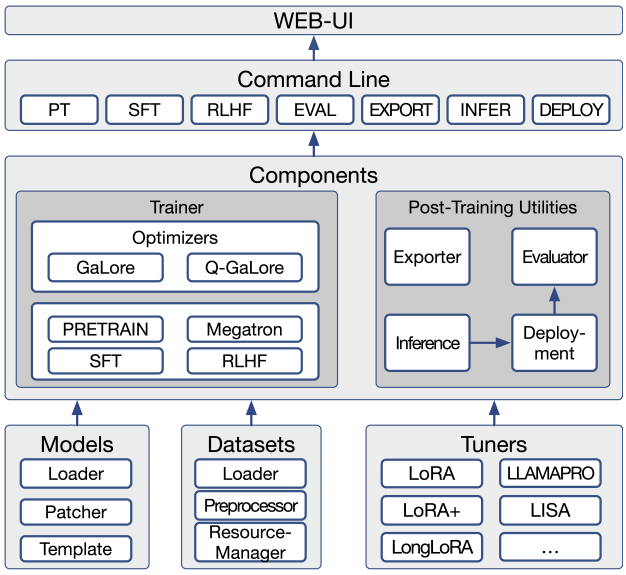

Figure 1: The framework of SWIFT.

Trainer. Trainer module includes both the SFT/PT trainer and the human alignment trainer. The former is used for predicting and training the cross-entropy of the next token; and the latter is used for training various RLHF algorithms. For RLHF of multi-modal models, we made additional modifications and adaptations to ensure that all models can use any compliant alignment dataset. For pre-training, SWIFT supports Megatron architecture models.

Post-training. The inference and deployment are built on three backends: PyTorch Native (PT), vLLM, and LMDeploy, sharing identical arguments. For evaluation, SWIFT rely on the EvalScope[7] framework from ModelScope Community, supporting 140+ text and multi-modal evaluation sets, as well as custom evaluation datasets, covering most of the popular evaluation datasets. Various quantization methods and exporting to Ollama are also supported.

CLI and Web UI. To facilitate the use of SWIFT in actual production environments, we released SWIFT on PYPI and supported various functionalities via the command line. The highest-level interface is the web UI, where users can select different training stages and adjust various training hyperparameters directly. After training starts, the UI will display training logs and loss or accuracy charts. The inference and deployment are also supported in web UI. This workflow is applicable to both pure text models and multi-modal models.

Conclusion and Future Work

This paper presents SWIFT, the most comprehensive multi-task training framework and end-to-end solution for large models to date. To continually track the advancements in large models, SWIFT is an ongoing project that is continuously maintained and updated. SWIFT fosters unprecedented levels of scalability, efficiency, and adaptability, thereby pushing the boundaries of large model development.

[7]https://github.com/modelscope/evalscope

References

AI, .; :; Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Li, H.; Zhu, J.; Chen, J.; Chang, J.; Yu, K.; Liu, P.; Liu, Q.; Yue, S.; Yang, S.; Yang, S.; Yu, T.; Xie, W.; Huang, W.; Hu, X.; Ren, X.; Niu, X.; Nie, P.; Xu, Y.; Liu, Y.; Wang, Y.; Cai, Y.; Gu, Z.; Liu, Z.; and Dai, Z. 2024. Yi: Open Foundation Models by 01.AI. arXiv:2403.04652.

Beyer, L.; Steiner, A.; Pinto, A. S.; Kolesnikov, A.; Wang, X.; Salz, D.; Neumann, M.; Alabdulmohsin, I.; Tschannen, M.; Bugliarello, E.; Unterthiner, T.; Keysers, D.; Koppula, S.; Liu, F.; Grycner, A.; Gritsenko, A.; Houlsby, N.; Kumar, M.; Rong, K.; Eisenschlos, J.; Kabra, R.; Bauer, M.; Bošnjak, M.; Chen, X.; Minderer, M.; Voigtlaender, P.; Bica, I.; Balazevic, I.; Puigcerver, J.; Papalampidi, P.; Henaff, O.; Xiong, X.; Soricut, R.; Harmsen, J.; and Zhai, X. 2024. PaliGemma: A versatile 3B VLM for transfer. arXiv:2407.07726.

Chen, Y.; Qian, S.; Tang, H.; Lai, X.; Liu, Z.; Han, S.; and Jia, J. 2024. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models. arXiv:2309.12307.

Diao, S.; Pan, R.; Dong, H.; Shum, K. S.; Zhang, J.; Xiong, W.; and Zhang, T. 2023. Lmflow: An extensible toolkit for finetuning and inference of large foundation models. *arXiv preprint arXiv:2306.12420*.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.

Jiang, Z.; Mao, C.; Huang, Z.; Ma, A.; Lv, Y.; Shen, Y.; Zhao, D.; and Zhou, J. 2023a. Res-Tuning: A Flexible and Efficient Tuning Paradigm via Unbinding Tuner from Backbone. arXiv:2310.19859.

Jiang, Z.; Mao, C.; Pan, Y.; Han, Z.; and Zhang, J. 2023b. SCEdit: Efficient and Controllable Image Diffusion Generation via Skip Connection Editing. arXiv:2312.11392.

Kalajdzievski, D. 2023. A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA. arXiv:2312.03732.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved Baselines with Visual Instruction Tuning. arXiv:2310.03744.

Liu, S.-Y.; Wang, C.-Y.; Yin, H.; Molchanov, P.; Wang, Y.-C. F.; Cheng, K.-T.; and Chen, M.-H. 2024b. DoRA: Weight-Decomposed Low-Rank Adaptation. arXiv:2402.09353.

Pan, R.; Liu, X.; Diao, S.; Pi, R.; Zhang, J.; Han, C.; and Zhang, T. 2024. LISA: Layerwise Importance Sampling for Memory-Efficient Large Language Model Fine-Tuning. arXiv:2403.17919.

Wu, C.; Gan, Y.; Ge, Y.; Lu, Z.; Wang, J.; Feng, Y.; Shan, Y.; and Luo, P. 2024. LLaMA Pro: Progressive LLaMA with Block Expansion. arXiv:2401.02415.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024. Qwen2 Technical Report. arXiv:2407.10671.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2024a. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Red Hook, NY, USA: Curran Associates Inc.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; Feng, Z.; and Ma, Y. 2024b. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. arXiv:2403.13372.