# Conversational AI. Dialogsysteme, Chatbots, Assistenten

Veranstalter: Christoph Ringlstetter

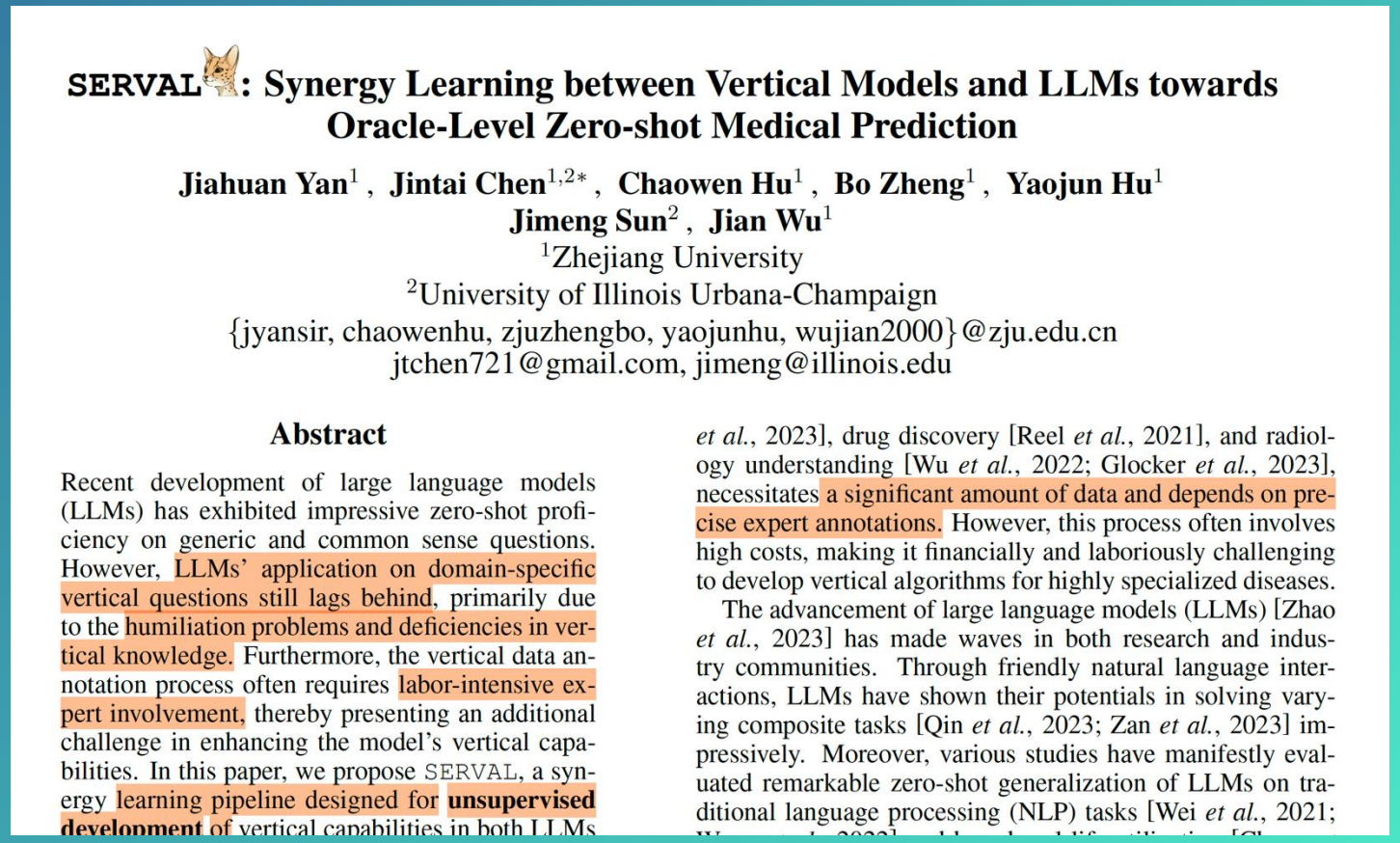
Sitzung 7: Vertical Models

**Was machen wir denn heute.**

- Orga. Referate, Zulassung, Termine.
- News of the Week
- Referat Vertical Models
- Besprechung Paper Serval Synergy Learning between V Models and LLMs
- Besprechung Paper LLAMA Factory

# Jinhuan Yan et al. Synergy Learning between V Models and LLMs – Überblick

**Ausgangsthese: Zero Shot gut für allgemeine Fragen. Domänenspezisch, Vertical: Halluzination, Wissensdefizite. Verstärkung des Domänenwissens: Expertenbottleneck Labeling Problem – Supervised Finetuning. Lösung: LLM Supervised Finetuning, Bootstrapping.**



**SERVAL: Synergy Learning between Vertical Models and LLMs towards Oracle-Level Zero-shot Medical Prediction**


**Jiahuan Yan**[1] , **Jintai Chen**[1,2*] , **Chaowen Hu**[1] , **Bo Zheng**[1] , **Yaojun Hu**[1]
**Jimeng Sun**[2] , **Jian Wu**[1]
[1]Zhejiang University
[2]University of Illinois Urbana-Champaign
{jyansir, chaowenhu, zjuzhengbo, yaojunhu, wujian2000}@zju.edu.cn
jtchen721@gmail.com, jimeng@illinois.edu



**Abstract**

Recent development of large language models (LLMs) has exhibited impressive zero-shot proficiency on generic and common sense questions. However, LLMs' application on domain-specific vertical questions still lags behind, primarily due to the humiliation problems and deficiencies in vertical knowledge. Furthermore, the vertical data annotation process often requires labor-intensive expert involvement, thereby presenting an additional challenge in enhancing the model's vertical capabilities. In this paper, we propose SERVAL, a synergy learning pipeline designed for **unsupervised development** of vertical capabilities in both LLMs


*et al.*, 2023], drug discovery [Reel *et al.*, 2021], and radiology understanding [Wu *et al.*, 2022; Glocker *et al.*, 2023], necessitates a significant amount of data and depends on precise expert annotations. However, this process often involves high costs, making it financially and laboriously challenging to develop vertical algorithms for highly specialized diseases.

The advancement of large language models (LLMs) [Zhao *et al.*, 2023] has made waves in both research and industry communities. Through friendly natural language interactions, LLMs have shown their potentials in solving varying composite tasks [Qin *et al.*, 2023; Zan *et al.*, 2023] impressively. Moreover, various studies have manifestly evaluated remarkable zero-shot generalization of LLMs on traditional language processing (NLP) tasks [Wei *et al.*, 2021;

J. Yan et al. Synergy Learning – Überblick Paper

Lösungsansatz:

Lernpipeline für „unsupervised“ Tuning.

1) LLM Zeroshot für Domänen-Annotationen

2) Trainieren des Vertical LMs auf diesen Annotationen

3) Verbessern Zeroshot des General LLM durch Tuning über das Vertical LM

4) Repeat

=> Experimente zu Medizindomäne Vergleich mit Full supervised

Testergebnisse auf den medizinischen Diagnose Datensets superior oder par mit supervised – klar abhängig vom Detail bei voller Korpusstärke supervised upper bound?

Zeroshot wird mit Iterationen noch besser. Hier auch Experimente begrenzt aussagekräftig aber erster Hinweis.

J. Yan et al. Synergy Learning – Related Work.

Noisy Label Learning (interessantes Feld – weiter lesen)
Mature Research Field CV Trainieren von DNNs mit noisy labels [Song et al 2022]
(i) loss correction based methods. Korrektur der loss functions während des Trainings durch relabeling [Hendricks 2018].
(ii) sample-selection based methods. semi supervised, ensemble learning, curriculum learning.

→Nochmal in Lit nachlesen. Nur aus dem Paper nicht nachvollziehbar.

J. Yan et al. Synergy Learning – Methoden

1) Fine grained LLM Annotation. Im Unterschied zum existierenden zero-shot Prediction Process: Queries führen direkt zu hartem Output e.g. für NLP Klassifikationsaufgaben.

SERVAL: braucht die “SOFT OUTPUT VERTEILUNG“ LLM Unsicherheit kann über den Prompt in die Ausgabe überführt werden.

=> GPT: „Please predict the probablility that this patient suffers from X disease: 0.0 means very unlikely 1.0 means very likely“

Detailierte Prompts im Paper Anhang.

?Kommen wir so an die interne Verteilung im Netz. Hat das Netz das im Instruction Tuning gelernt? Interessante Fragen für weitere Recherchen.

## J. Yan et al. Synergy Learning – Methoden

**2) LLM Supervised Learning Vertikales Modell**

Trainiere ein Vertikales Modell (e.g. tabular FT Transformer: medizinische Tabellen, oder Bert Finetuning für medizinische Fragen).

Das kann als Lernen von Noisy Labels begriffen werden.

Die Intuition hinter dem Vorgehen: Labeling durch fortgeschrittene LLMs wird nicht komplett falsch sein (vgl [Song et al. 2022 Studie zu Computer Vision Lernen auf Noisy Data])

Aber schon Unterschiede zum traditionellen NLL. Vision Datensets sind gut verfügbar, günstig verteilt.

Im Paper Setting: Daten ev. unpassend, distorted Patterns → Zielverteilung entsprechendes Validation Set benötigt um Probleme zu erkennen

J. Yan et al. Synergy Learning – Methoden

LLM Supervised Learning Vertikales Modell

Inspiriert von Sample Selection basierten NLL Methoden [Li, Zhou] → teilen der Daten in labeled und unlabeled um einen semisupervised Training Prozess zu starten → Details hierzu in der im Paper angebenen Literatur.

Im unterschied zu NLL haben wir keine harten Labels sondern Soft labels.

Benutzen dann die Labels mit hoher Konfidenz für Early Stopping im Training Prozess. Hier > 0.9

Sample Methode: Divide Mix [Li et al 2019]

Methoden können über externes Validation Set getunt/kontrolliert werden. Das ist ohnehin immer ratsam.

J. Yan et al. Synergy Learning – Methoden

LLM Backtraining: Reverse Tuning Schritt um die zero-shot domänenspezifische Performanz des LLM zu verstärken – hier Medizindomäne.

=> ohne Einsatz spezieller Techniken ein Retraining durchführen z.B. über die APIs des LLMs für Finetuning.

=> Günstig hier: „Memorization Effekt“. Hilft dem LLM einfache, generelle Pattern zu lernen. Outlier, falsch annotierte Beispiele werden weniger gelernt.

Siehe angegebene Literatur. Backtuning: Hen et al 2018 aus der NLL Community.

# J. Yan et al. Synergy Learning – Methoden

**Überblick zum iterativen Vorgehen SERVAL.**

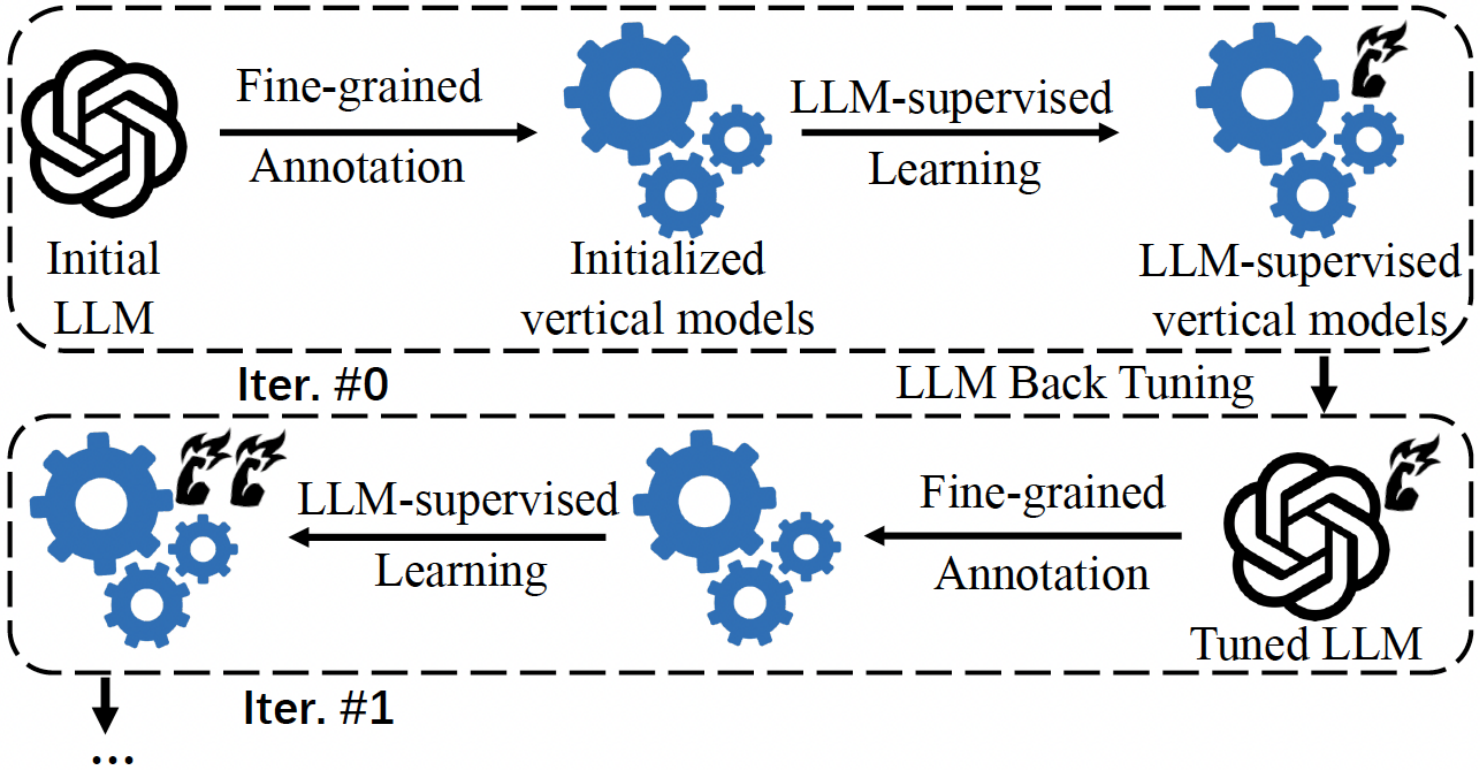



Figure 2: Iterative illustration of SERVAL. LLM Back Tuning step bridges two iterations by reversely updating the LLM with well-trained vertical models, forming a co-teaching manner.

Yan et al. p 4

# J. Yan et al. Synergy Learning – Experimente

Reichhaltige und durchaus viele Datensets in der medizinischen Domäne vorhanden die im Paper zitiert und benutzt werden.

Als vertikale Modelle ältere, kleine Modelle. FT Transformer, Bert

Setup: Pytorch, Python 3.8, NVIDIA RTX3090.

Resultate: teilweise über 10% besser,sehr gut auch im Vergleich zu full supervised.

Details bitte selbst nachlesen.

# Yaowei Zheng et al. Llama Factory – Introduction

Finetuning must be efficient

Downstream Tasks profit

Factory Framework for efficient Training/Methods

Released on Github

Wide Usage/Community

Extremly good Lit.

## LLAMAFACTORY: Unified Efficient Fine-Tuning of 100+ Language Models

**Yaowei Zheng[1], Richong Zhang[1*], Junhao Zhang[1], Yanhan Ye[1], Zheyan Luo[1], Zhangchi Feng[1], Yongqiang Ma[2]**

[1]School of Computer Science and Engineering, Beihang University, China

[2]School of Software and Microelectronics, Peking University, China

{hiyouga,zhang.jh,yeyanhan,akamya,zcmuller}@buaa.edu.cn, zhangrc@act.buaa.edu.cn, codingma@pku.edu.cn

Demonstration video: **https://youtu.be/W29FgeZEpus**

### Abstract

Efficient fine-tuning is vital for adapting large language models (LLMs) to downstream tasks. However, it requires non-trivial efforts to implement these methods on different models. We present LLAMAFACTORY, a unified framework that integrates a suite of cutting-edge efficient training methods. It provides a solution for flexibly customizing the fine-tuning of 100+

To address the above problems, we develop LLAMAFACTORY, a framework that democratizes the fine-tuning of LLMs. It unifies a variety of efficient fine-tuning methods through scalable modules, enabling the fine-tuning of hundreds of LLMs with minimal resources and high throughput. In addition, it streamlines commonly used training approaches, including generative pre-training (Rad-

## Yaowei Zheng et al. Llama Factory – Introduction

Finetuning von Modellen mit sehr großen Parameterzahlen: effizienter Zugang auch für Research und kleinere Gruppen

Zahlreiche Methoden für effizientes Finetuning implementiert, see also Survey Papers on LLM für Details dazu.

Ziel: Adaptiere Metoden, modular schaltbar, Interface für Customisierung von LM

1) generative pre-training
2) supervised fine-tuning
3) RLHF
4) Directed Preference Optimization

## Yaowei Zheng et al. Llama Factory – Introduction

**Module:**

**Model Loader: Adapter an vortrainierte Modelle anschließen**

**Data Worker: Datensets laden und aufbereiten**

**Trainer: State of the Art Trainingmethoden bereitstellen**

**Code Basis: Pytorch, Transformers, PEFT, TRL**

**UI Bord: Gradio**

# Yaowei Zheng et al. Llama Factory – Introduction

**Feature Vergleich zu anderen Frameworks**

**LLAMA Factory deutlich umfassender**

| | LLAMAFACTORY | FastChat | LitGPT | LMFlow | Open-Instruct |
|---|---|---|---|---|---|
| LoRA | ✓ | ✓ | ✓ | ✓ | ✓ |
| QLoRA | ✓ | ✓ | ✓ | ✓ | ✓ |
| DoRA | ✓ | | | | |
| LoRA+ | ✓ | | | | |
| PiSSA | ✓ | | | | |
| GaLore | ✓ | ✓ | | ✓ | ✓ |
| BAdam | ✓ | | | | |
| Flash attention | ✓ | ✓ | ✓ | ✓ | ✓ |
| $S^2$ attention | ✓ | | | | |
| Unsloth | ✓ | | ✓ | | |
| DeepSpeed | ✓ | ✓ | ✓ | ✓ | ✓ |
| SFT | ✓ | ✓ | ✓ | ✓ | ✓ |
| RLHF | ✓ | | | ✓ | |
| DPO | ✓ | | | | ✓ |
| KTO | ✓ | | | | |
| ORPO | ✓ | | | | |

Table 1: Comparison of features in LLAMAFACTORY with popular frameworks of fine-tuning LLMs.

Yaowei Zheng et al S.2

## Yaowei Zheng et al. Llama Factory – Related Work

Viele Frameworks nach dem ersten Hype um Finetuning. Oft entlang spezifischer Modelle implementiert: z.B. Llama Adapter. Fast Chat – Chat Completion; Open Instruct – Instruction Tuning. LitGPT -- generelle Modelle
GPT4All – Consumer Devices. Integration der Bibliotheken in LLAMA Factory

Sehr guter Überblick mit exzellenten Literaturangaben

## Yaowei Zheng et al. Llama Factory – Effiziente Finetuning Techniken

Zwei Haupt Kategorien: Optimierungsfokussiert, Berechnungsfokussiert

Optimierung: Finetuning der NN Parameter bei minimalen Kosten: Speicher, Durchläufe z.B. durch Begrenzung der geänderten Parameter.

Berechnung: Zeit/Speicheraufwand für Berechnung minimieren.

Effizienzoptimierung:

Freeze-Tune: kleine Teilmenge von Decoderlayern. Gradient Low Rank Projection; Gradienten auf niedrig dimensionalen Raum berechnen: Speicher Effizient auch bei vollem Parameterlearning.

**Yaowei Zheng et al. Llama Factory – Finetuning**

**Optimierung:**

**LORA alle Parameter einfrieren und ein Paar trainierbare LowRank Matrizen von gleicher Dimension hinzufügen.**

**QLORA mit Quantisierung**

**DORA: vortrainierte Gewichte nach Richtung und Beitrag updaten**

**etc pp.**

## Yaowei Zheng et al. Llama Factory – Finetuning

Effizientere Berechnung: Frühe Techniken mixed precision training, activation checkpointing.
Flash Attention: Hardware Optimierung
$S^2$ Attention: ausgedehnter Kontext wird effizient berechnet – weniger Speicherbedarf durch low prec Repräsentation Datentypen.
Schnellere Berechnungen durch effiziente Funktionen.

Durch die Kombinationen von Techniken von 18 Bytes per Param auf 0.6 Bytes

# Yaowei Zheng et al. Llama Factory – Framework Überblick

Loader, Worker, Trainer

LLMs + VLMs

Single Turn, Multi Turn

verschiedene Training Vorgehensweisen

LLAMA Board: Grafisches Interface

um auf die Module Zuzugreifen.

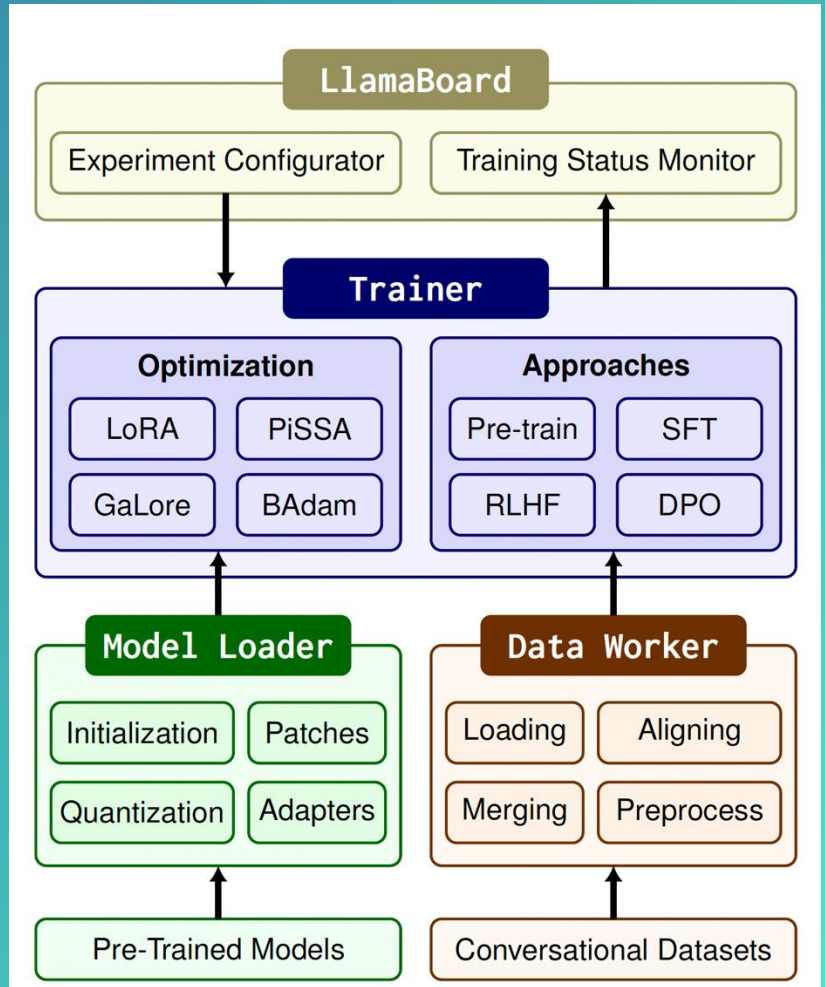


Figure 1: The architecture of LLAMAFACTORY.

Yaowei Zheng et al. S. 3

Yaowei Zheng et al. Llama Factory – Framework

Model Loader: initialization, patching, quantization, adapter attachment

1) Model Initialisierung: Auto Klassen aus dem Transformer Paket um vortrainierte Modelle zu laden und die Parameter zu initialisieren.

AutoModelForVision2Seq

AutoModelForCausalLM

Tokenizer wird mit dem Modell geladen.

Automatisierte Fallbacks e.g. zu Resize bei Kapazitätsüberschreitungen.

Yaowei Zheng et al. Llama Factory – Framework

2) Model Patching $S^2$ Attention

Monkeypatch um die Forward Berechnung der Modelle umzusetzen.

Mixture of Experts Blöcke als Blattmodule um exzessives partitionieren zu verhindern.

Flash Attention ermöglicht.

Deep Speed Bibliothek integriert: KDD 2020 Tutorial www.deepspeed.ai

Yaowei Zheng et al. Llama Factory – Framework

3) Model Quantisierung

Dynamische Herabsetzung auf 8/4 Bit über die LLM.int8 Bibliothek Dettmers et al.

QLORA: Auch für Posttraining Quantisierung: Adaptermethoden.

4) Adapteranschluss: Passende Layer für die Adapter automatisch gefunden. PEFT Bibliothek integriert. Backward Berechnung mit UnSloth Bibliothek implementiert.

RLHF mit einem Value Head Layer implementiert: statt 4 Modellen Eines.

Präzision Adaptierung: Float Implementierungen je nach Hardware Architektur

Yaowei Zheng et al. Llama Factory – Framework

Data Worker: Datenset laden, alignieren, mergen, preprocessing

Hugging Face API

Standardstruktur adaptieren

Merges durch Standardstruktur einfache Konkatenation, auch Streams implementiert.

Preprocessing: Chat Templates für Instruction Tuning je nach Modeltyp

Yaowei Zheng et al. Llama Factory – Framework

Trainer: SOTA Finetuning Methoden integriert, laufende Updates des Frameworks

Transformer: SFT, Pretraining

TRL: RLHF und DPO

Modul Sharing RLHF: macht Training auf Consumer Devices möglich. Demokratisierung von LLM Finetuning. Über Adapter und Valueheads die dynamisch ausgetauscht werden können implementiert:
policy model, value model, reference model, reward model alle über ein pretrained model simuliert.

Yaowei Zheng et al. Llama Factory – Training

Verteiltes Training: Deep Speed Bibliothek

Für Model Inferenz mehrere Tuning Methoden implementiert

Model Evaluation: Multiple Joice, Bleu Rouge etc.

LLAMABoard: Graphische Parameterkonfigurationen Testen. Trainingsmonitoring.

Empirische Studie: selbst lesen. Sehr gute Ergebnisse.