



Hands-on with Language Models and AI Agents:

Small Language Models, Fine-Tuning, Alignment &
Explainability.

Meet the Team



Theekshana Samaradiwakara
Machine Learning Engineer



Varatharajah Vaseekaran
Senior Machine Learning
Engineer



Purusanth Shanmukanathan
Associate Technical Lead
Machine Learning

Content

- 1. Small Language Models**
- 2. Supervised Fine-Tuning**
- 3. Preference Alignment**
- 4. AI Agents**
- 5. Explainability**



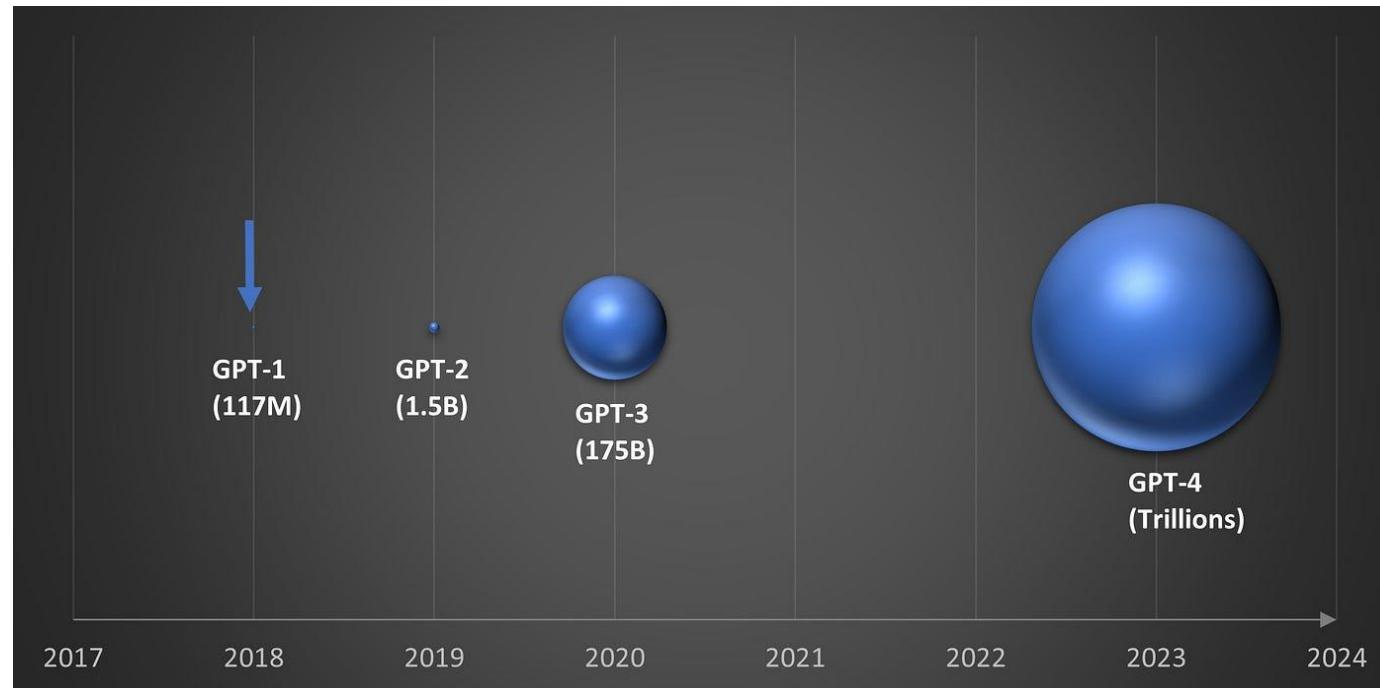
Language Models

What are Large Language Models (LLM)?

A large model trained on large corpora of text data making them capable of Natural Language Processing (NLP).

GPT-4

- 1.76 trillion parameters (6800 GB)
- trained using 10 trillion words (570 GB)



<https://towardsai.net/p/artificial-intelligence/size-matters-how-big-is-too-big-for-an-lm>

(NLP) Natural Language Processing .

NLP is understanding and generating human language.

- (NLU) Natural Language Understanding - Sentiment classification
- (NLG) Natural Language Generation - Translation, Essay writing

Language Generation

I _____

https://lena-voita.github.io/nlp_course/language_modeling.html

Drawbacks of LLMs

- LLMs have revolutionized the AI domain with their remarkable capabilities.
- However, the **need of high computational resources** and their **complexity** make them impractical for real-world use cases.

The answer: **Small Language Models.**



Small Language Models (SLM)



Understanding SLMs

What are Small Language Models (SLMs)?

- Lightweight AI models.
- Optimized for low-computational intensive devices - smartphones and embedded systems
- Significantly smaller than LLMs (around 1 million to 10 billion parameters)
- Retain important NLP capabilities - summarization, question-answering, translation, etc.

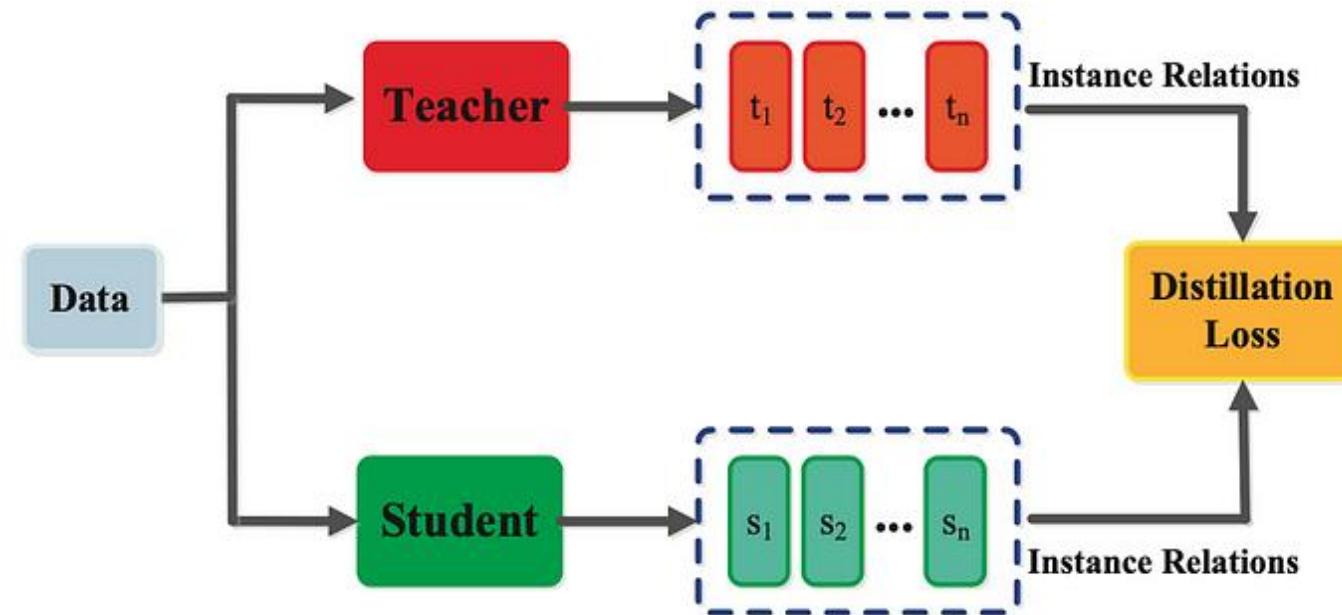
Understanding SLMs

How Are They Made Small?

1. Knowledge Distillation
2. Pruning
3. Quantization

How Are They Made Small - Knowledge Distillation

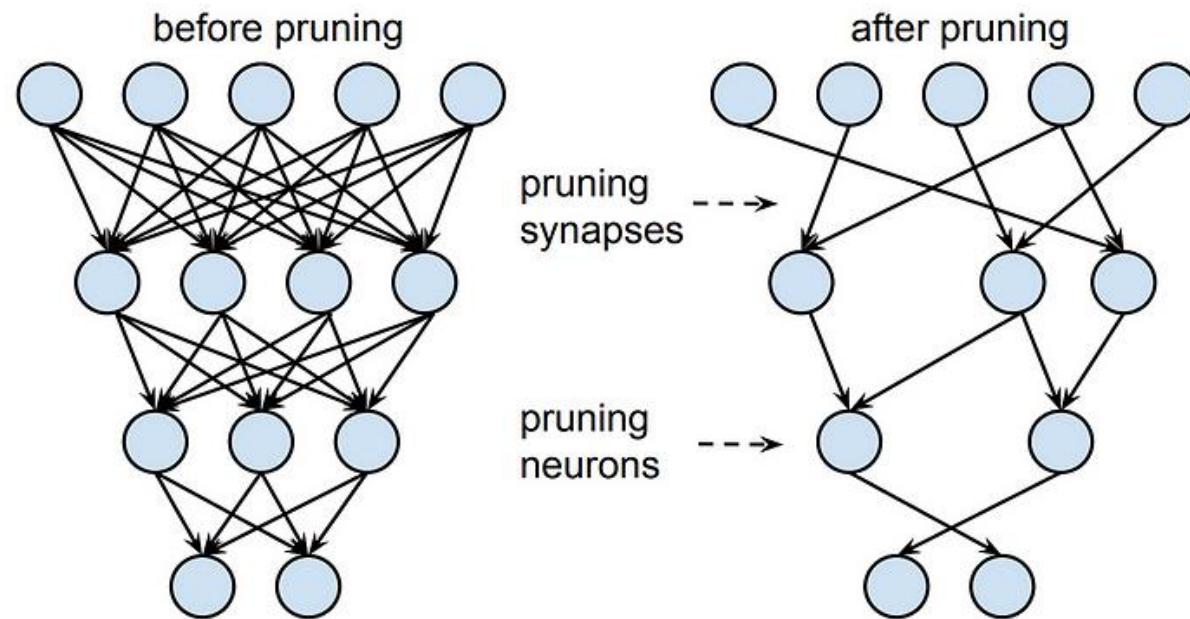
Training a smaller, simpler "student" model using knowledge transferred from a larger complex "teacher" model.



https://medium.com/@VK_Venkatkumar

How Are They Made Small - Pruning

Removing redundant or less important parameters within the neural network architecture.

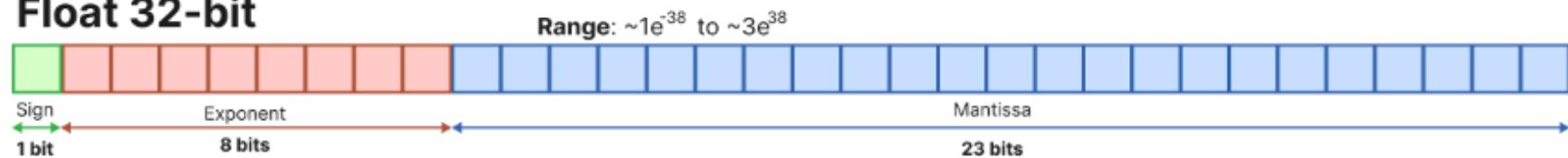


https://medium.com/@VK_Venkatkumar

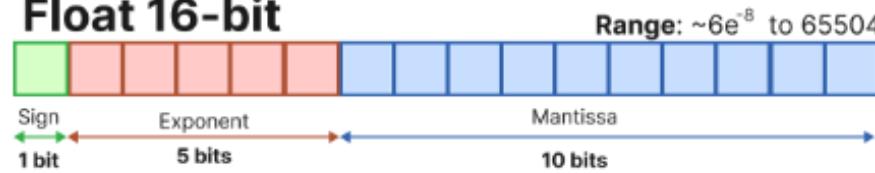
How Are They Made Small – Quantization

Reducing the precision of numerical values used in calculations (e.g. converting floating-point numbers to integers).

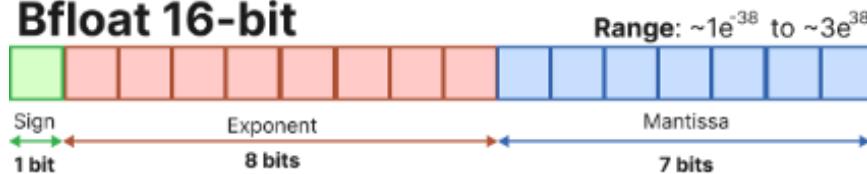
Float 32-bit



Float 16-bit



Bfloat 16-bit



Examples of Small Language Models

- **Llama3.2-1B** – By Meta
- **Qwen2.5-1.5B** – By Alibaba
- **DeepSeek-R1-1.5B** - DeepSeek's model distilled from Qwen2.5
- **SmoLM2-1.7B** – By HuggingFaceTB
- **Phi-3.5-Mini-3.8B** – Microsoft's
- **Gemma3-4B** - Google DeepMind

Calculate Memory Requirements

How large a model will fit on your local GPU for loading ?

$$\text{Memory} = \text{Number of Parameters} \times \text{Byte Precision}$$

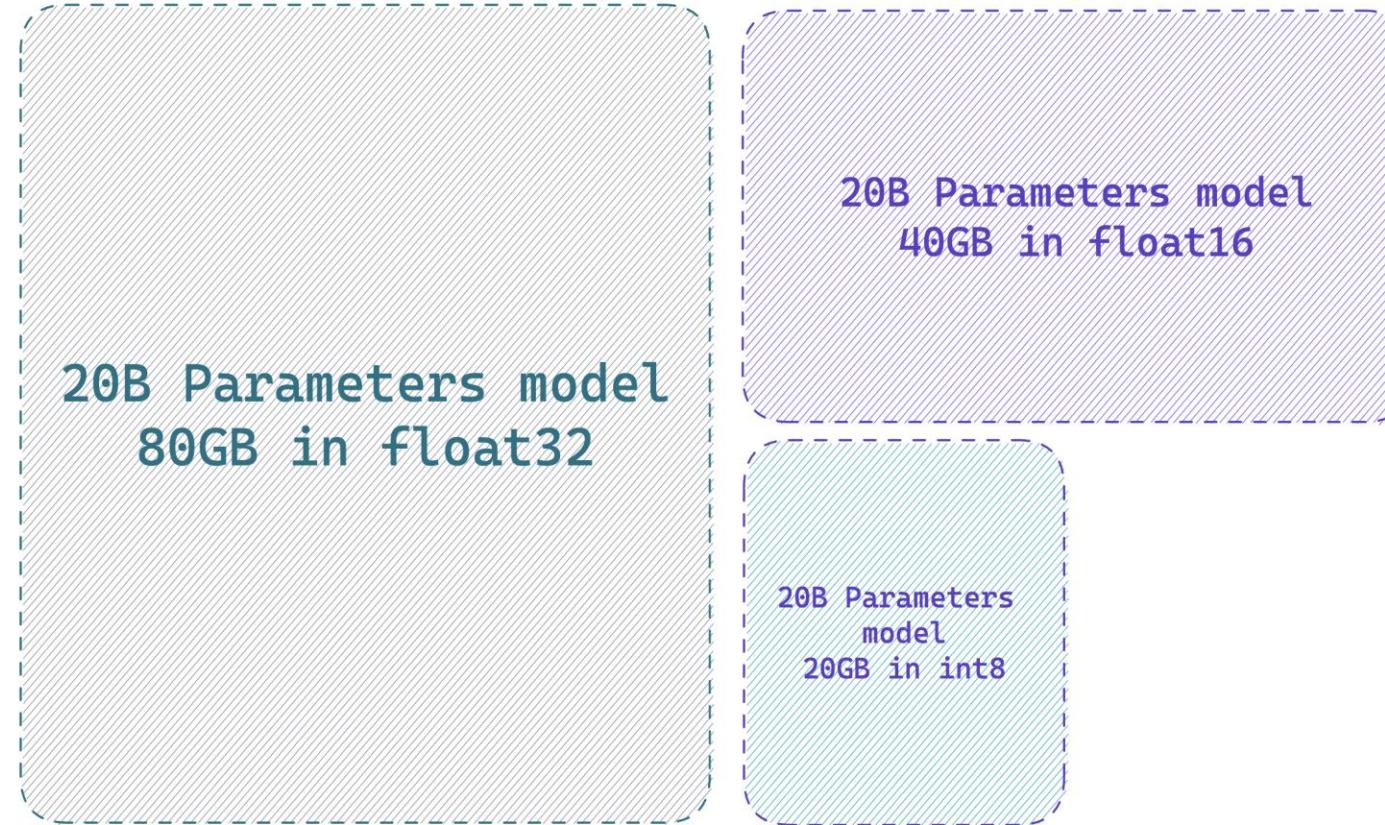
eg.. Model with 20 Billion parameters

$$\text{in 32 bit precision} = 20 \times (32/8) = 20 \times 4 = 80 \text{ GB}$$

$$\text{in 16 bit precision} = 20 \times (16/8) = 20 \times 2 = 40 \text{ GB}$$

$$\text{in 8 bit precision} = 20 \times (8/8) = 20 \times 1 = 20 \text{ GB}$$

Calculate Memory Requirements



<https://huggingface.co/blog/trl-peft>

Calculate Memory Requirements

How large a model will fit on your local GPU for inference ?

Total Inference Memory = Model Size + Forward Pass

- Forward Pass includes KV Cache and Activations
- Its usually about 20% of model size.

Memory = Number of Parameters × Byte Precision × 1.2

Benefits of Small Language Models

- **Low Compute Requirements** – Can run on consumer laptops, edge devices, and mobile phones.
- **Lower Energy Consumption** – Efficient models reduce power usage, making them environmentally friendly.
- **Faster Inference** – Smaller models generate responses quickly, ideal for real-time applications.
- **On-Device AI** – No need for an internet connection or cloud services, enhancing privacy and security.
- **Cheaper Deployment** – Lower hardware and cloud costs make AI more accessible to startups and developers.
- **Customizability** – Easily fine-tuned for domain-specific tasks (e.g., legal document analysis).

Real-World Applications of Small Language Models

1. Chatbots & Virtual Assistants
2. Code Generation
3. Language Translation
4. Summarization & Content Generation
5. Healthcare Applications
6. IoT & Edge Computing
7. Educational Tools
8. AI Agents

Demo - Run SLMs Locally



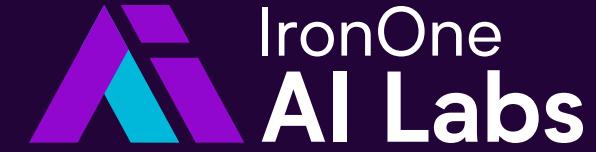
<https://bit.ly/3FUImvp>

Understanding SLMs

Limitations of Small Language Models

While SLMs offer numerous advantages, they also come with certain trade-offs:

- Narrow Scope
- Bias Risks
- Reduced Complexity
- Less Robustness



Supervised Fine-Tuning (SFT)



Supervised Fine-Tuning

Pre-training phase - Model learns general language knowledge, understanding about language modeling.

(e.g., GPT4)

Fine-tuning - Specialize the model's capabilities and optimize its performance on a narrower, task-specific dataset.

(e.g., chat GPT)

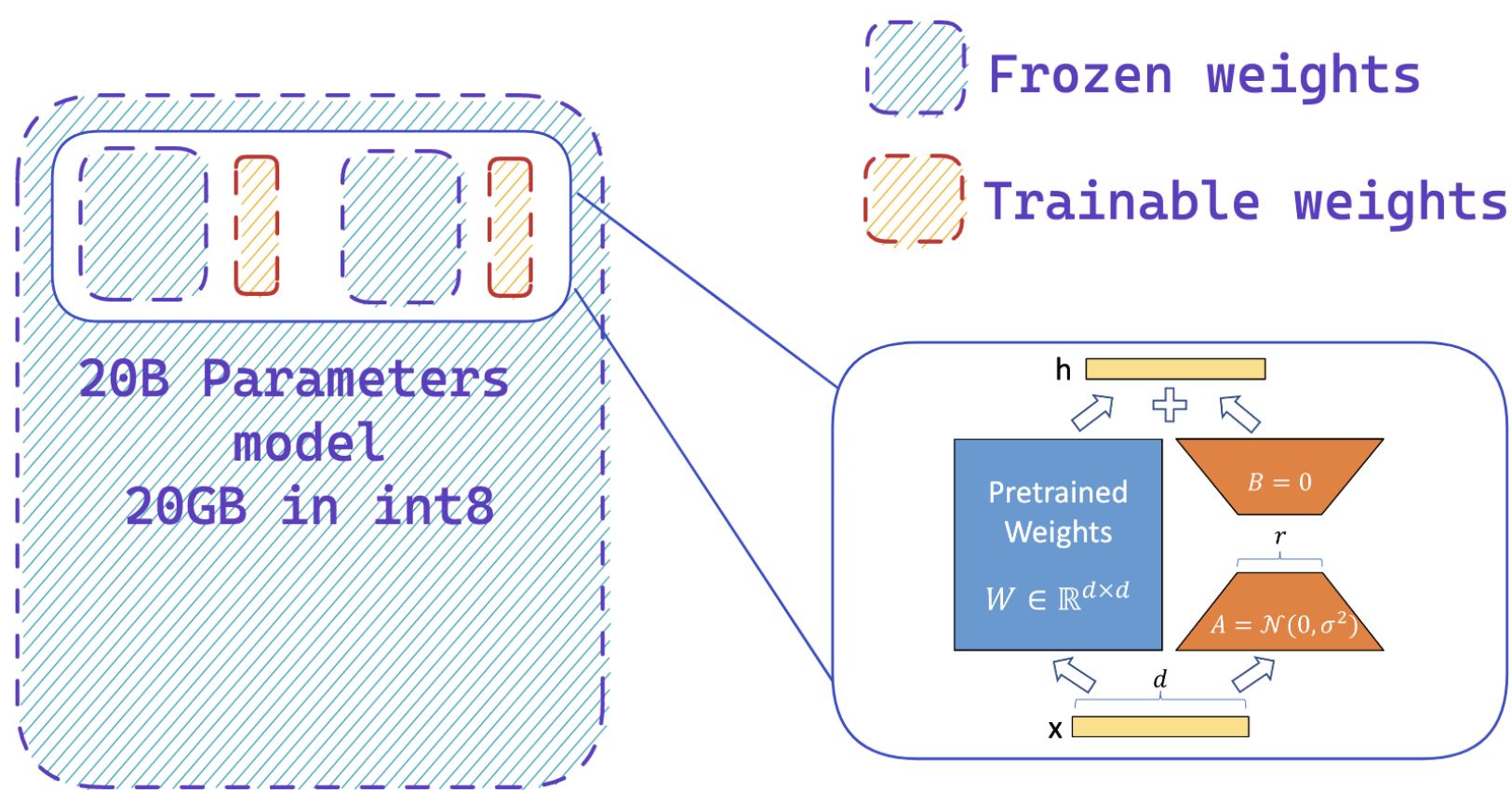
Supervised Fine-Tuning

Fine-tuning an LLM is computationally expensive.

Training requires more resources than inference due to the optimizer and gradient states.

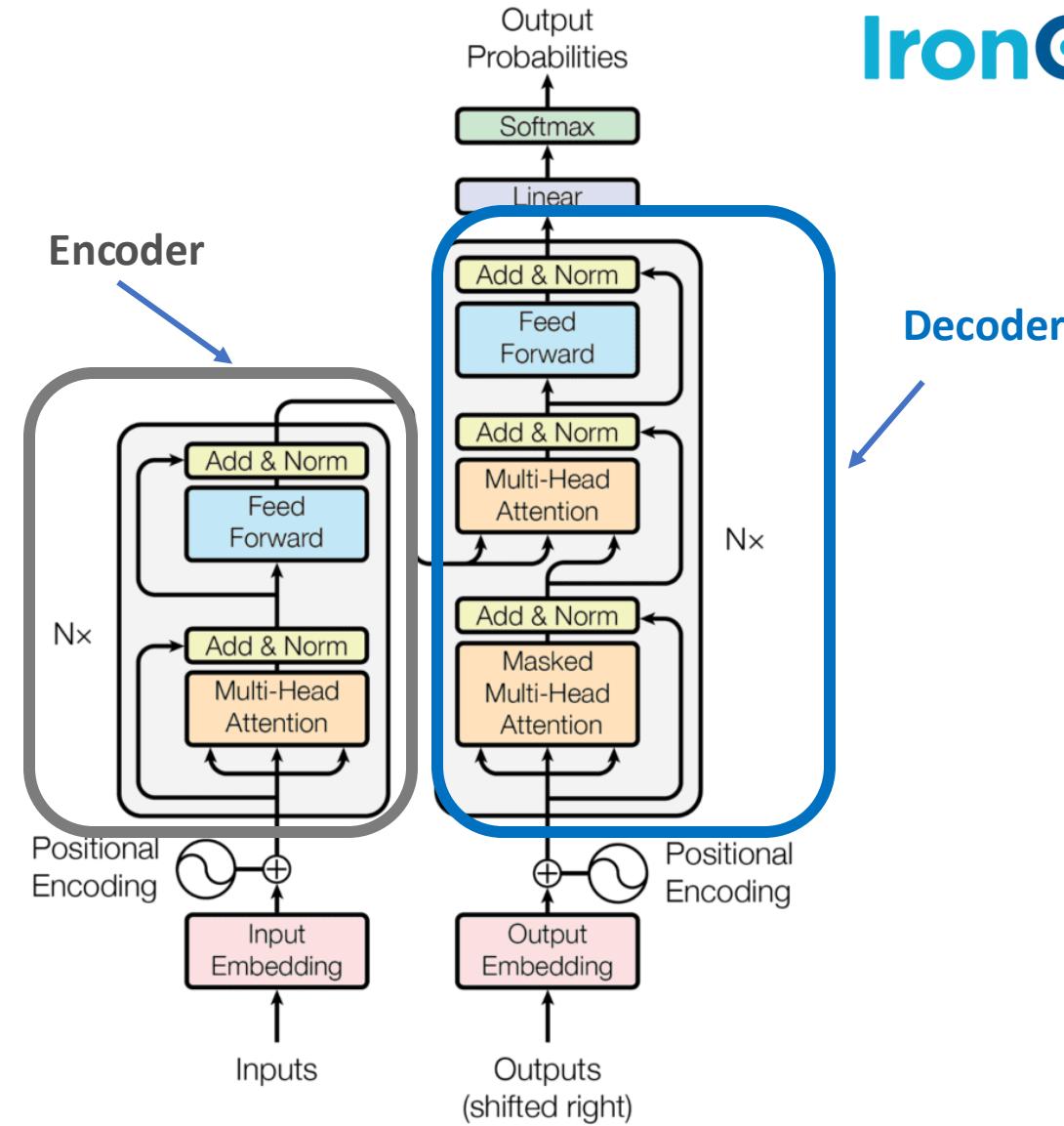
Parameter-Efficient Fine-Tuning (PEFT)

- LoRA - Low-Rank Adaptation
- QLoRA - Quantized Low-Rank Adaptation

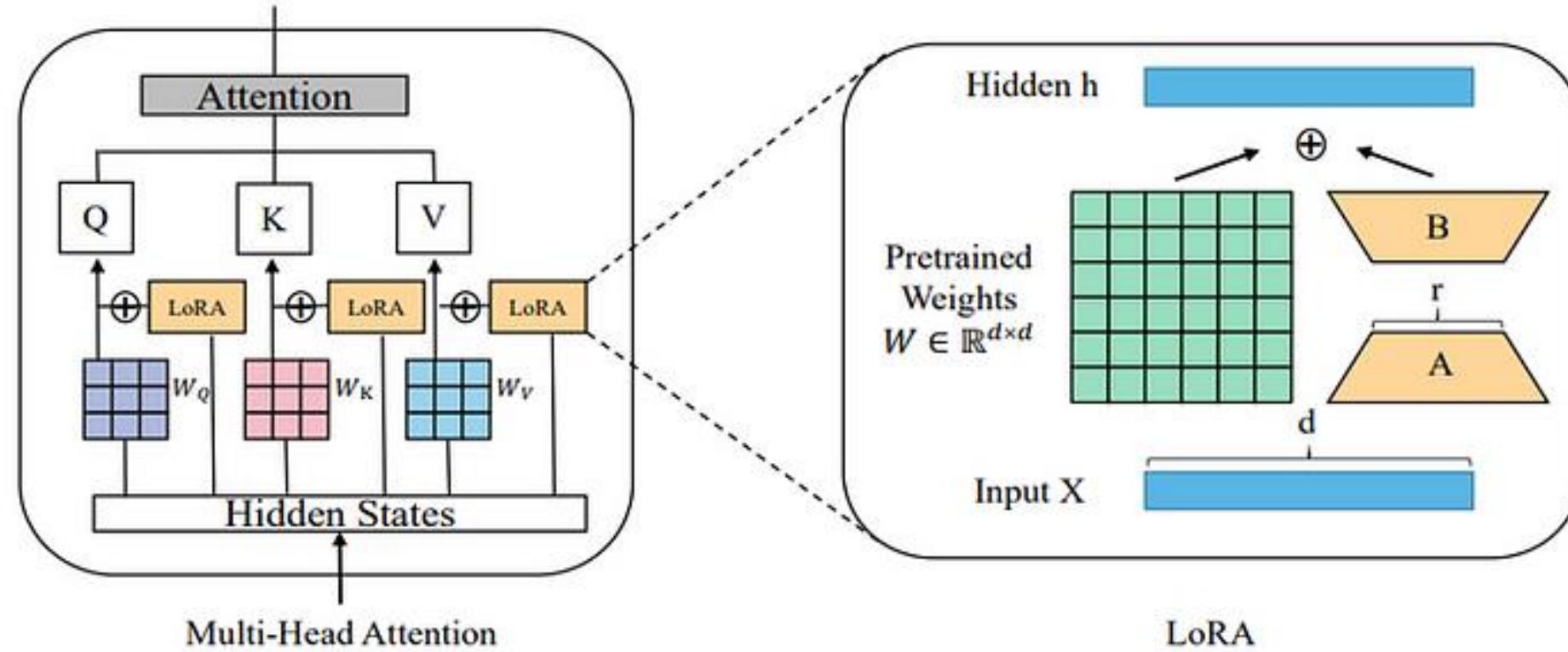


<https://huggingface.co/blog/trl-peft>

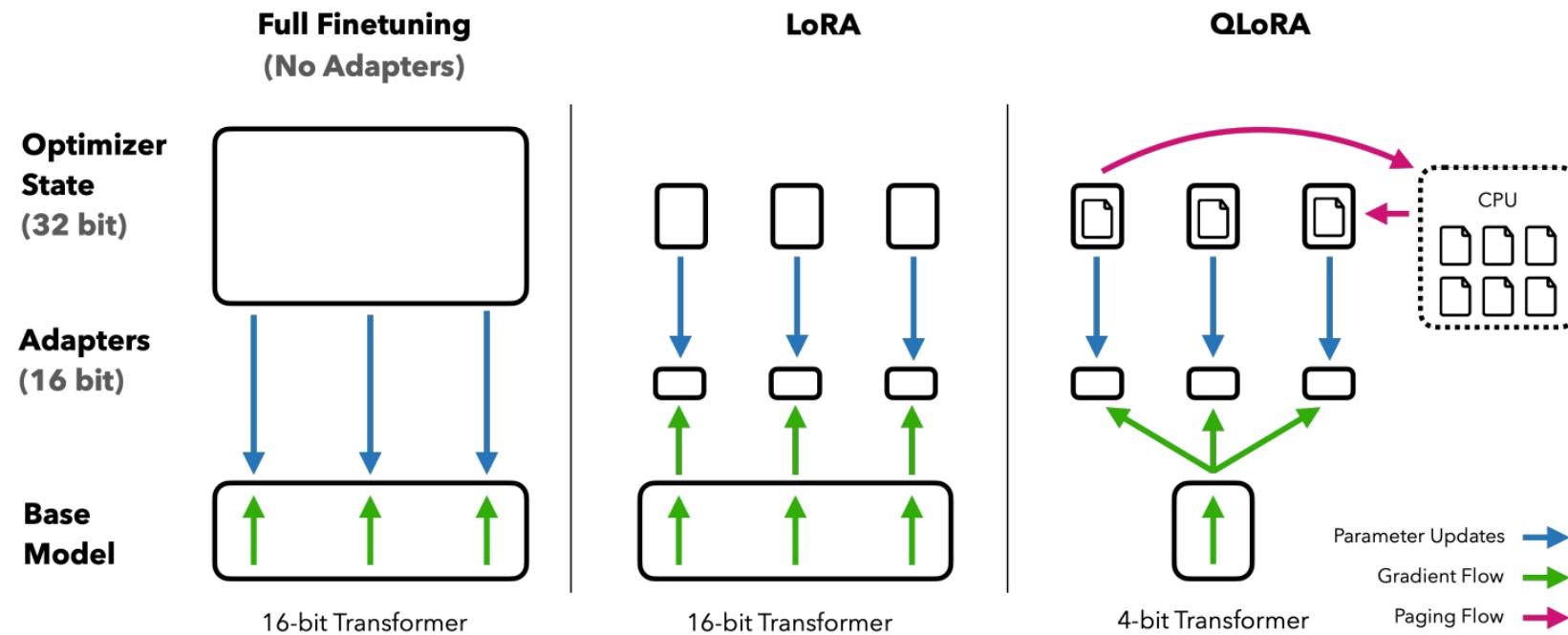
The Transformer Architecture



Low-Rank Adaptation (LoRA)



Quantization + LoRA



<https://arxiv.org/pdf/2305.14314>

Demo – Supervised Fine-Tuning



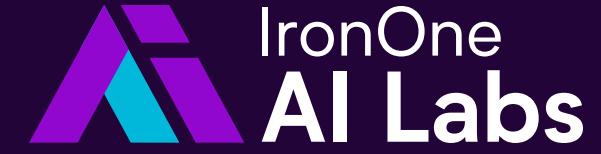
<https://bit.ly/3R6oOq4>

Supervised Fine-Tuning

Achieving precise control of their behavior is difficult.

Ensure that the language model's outputs are aligned with human expectations.

eg .. chattiness , bias, safety features.



Preference Alignment



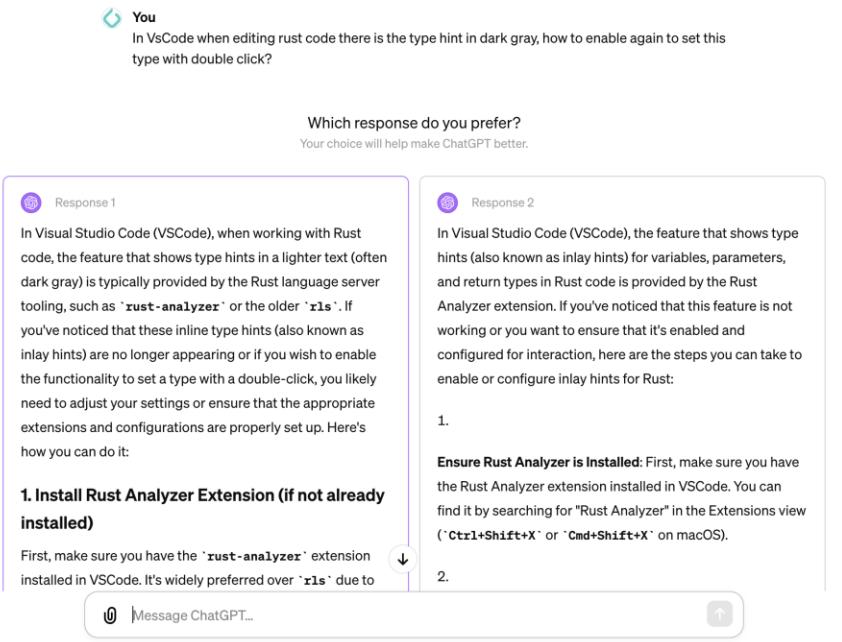
Preference Alignment

GPT 3 to GPT 3.5 ?

The base model is exactly the same.

Reinforcement Learning from Human Feedback (RLHF)

Letting the model learn the kind of answers
that are preferred by humans. (eg moderation)



You
In VsCode when editing rust code there is the type hint in dark gray, how to enable again to set this type with double click?

Which response do you prefer?
Your choice will help make ChatGPT better.

Response 1

In Visual Studio Code (VSCode), when working with Rust code, the feature that shows type hints in a lighter text (often dark gray) is typically provided by the Rust language server tooling, such as '`rust-analyzer`' or the older '`rls`'. If you've noticed that these inline type hints (also known as inlay hints) are no longer appearing or if you wish to enable the functionality to set a type with a double-click, you likely need to adjust your settings or ensure that the appropriate extensions and configurations are properly set up. Here's how you can do it:

1. Install Rust Analyzer Extension (if not already installed)

First, make sure you have the '`rust-analyzer`' extension installed in VSCode. It's widely preferred over '`rls`' due to

Response 2

In Visual Studio Code (VSCode), the feature that shows type hints (also known as inlay hints) for variables, parameters, and return types in Rust code is provided by the Rust Analyzer extension. If you've noticed that this feature is not working or you want to ensure that it's enabled and configured for interaction, here are the steps you can take to enable or configure inlay hints for Rust:

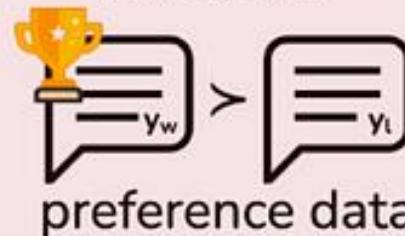
1. Ensure Rust Analyzer is Installed: First, make sure you have the Rust Analyzer extension installed in VSCode. You can find it by searching for "Rust Analyzer" in the Extensions view ('`Ctrl+Shift+X`' or '`Cmd+Shift+X`' on macOS).
- 2.

Message ChatGPT...

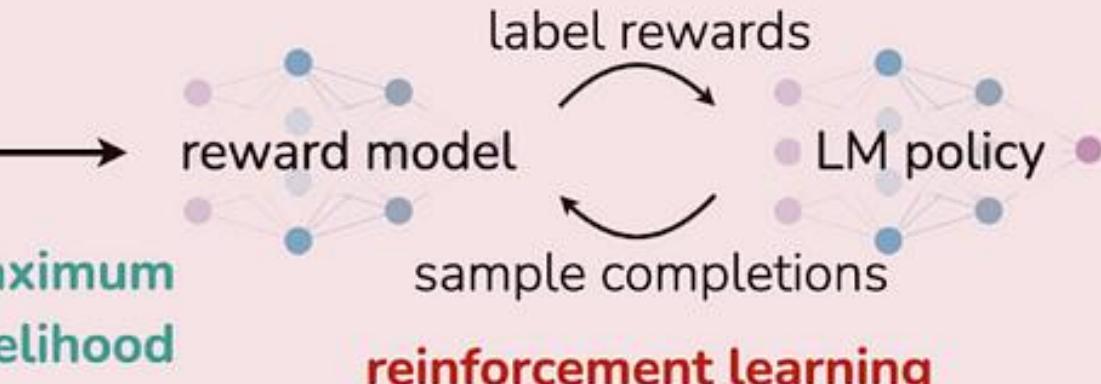
RLHF

Reinforcement Learning from Human Feedback (RLHF)

x: "write me a poem about
the history of jazz"



maximum
likelihood



<https://arxiv.org/pdf/2305.18290>

Proximal Policy Optimization (PPO)

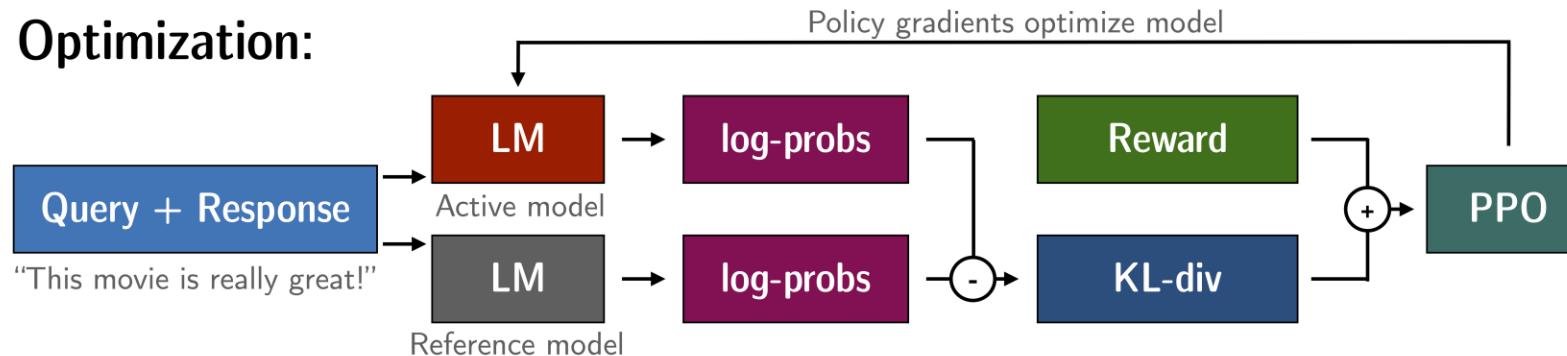
Rollout:



Evaluation:



Optimization:



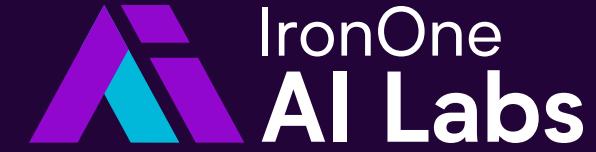
<https://huggingface.co/docs/trl/quickstart>

RLHF PPO

Reinforcement Learning from Human Feedback (RLHF)

But, training a reward model from human feedback was,

- Complex.
- often unstable.
- high computational requirement.

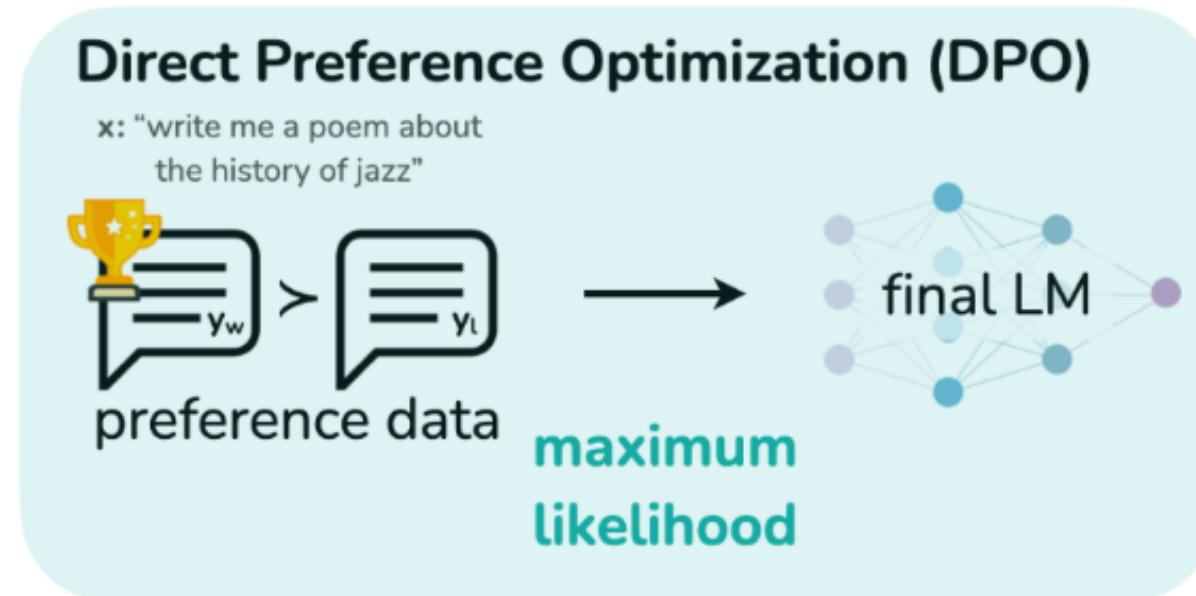


Direct Preference Optimization (DPO)



DPO

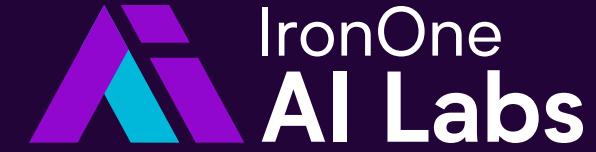
Direct Preference Optimization (DPO)



<https://arxiv.org/pdf/2305.18290>

Direct Preference Optimization (DPO)

- Skip the reward model and directly fine-tune an LLM with the preference dataset .
- The core idea of the Stanford team was that, internally, LLM itself worked as a reward model and just giving it preference data would do the trick.
- Stable, performant, and computationally lightweight.
- Eliminates the need for sampling from the LM during fine-tuning.



Preference Alignment Other Methods



Preference Alignment

- Proximal policy optimization (PPO)
- Direct Preference Optimization (DPO)
- **Identity Preference Optimization (IPO)**
- **Kahneman-Tversky Optimization (KTO)**

Demo – Preference Alignment



<https://bit.ly/3DZbuB5>



AI Agents



What is an AI Agent?

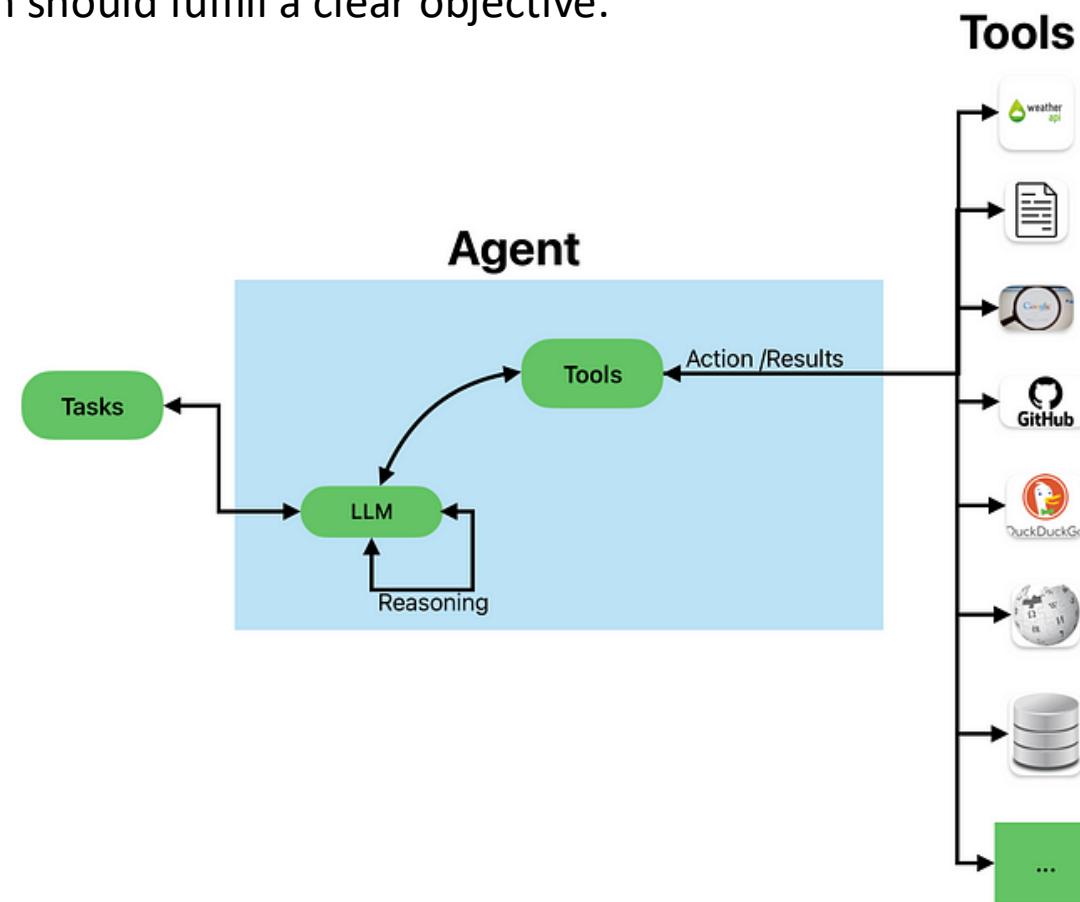
AI system capable of **reasoning, planning, and interacting** with its environment to achieve a user-defined objective using an AI model.

- **Understand natural language:** Interpret and respond to human instructions in a meaningful way.
- **Reason and plan:** Analyze information, make decisions, and devise strategies to solve problems.
- **Interact with its environment:** Gather information, take actions, and observe the results of those actions.

AI Tools

A Tool is a function given to the LLM. This function should fulfill a clear objective.

- Web Search
- Image Generation
- Retrieval
- API Interface
- SQL connections

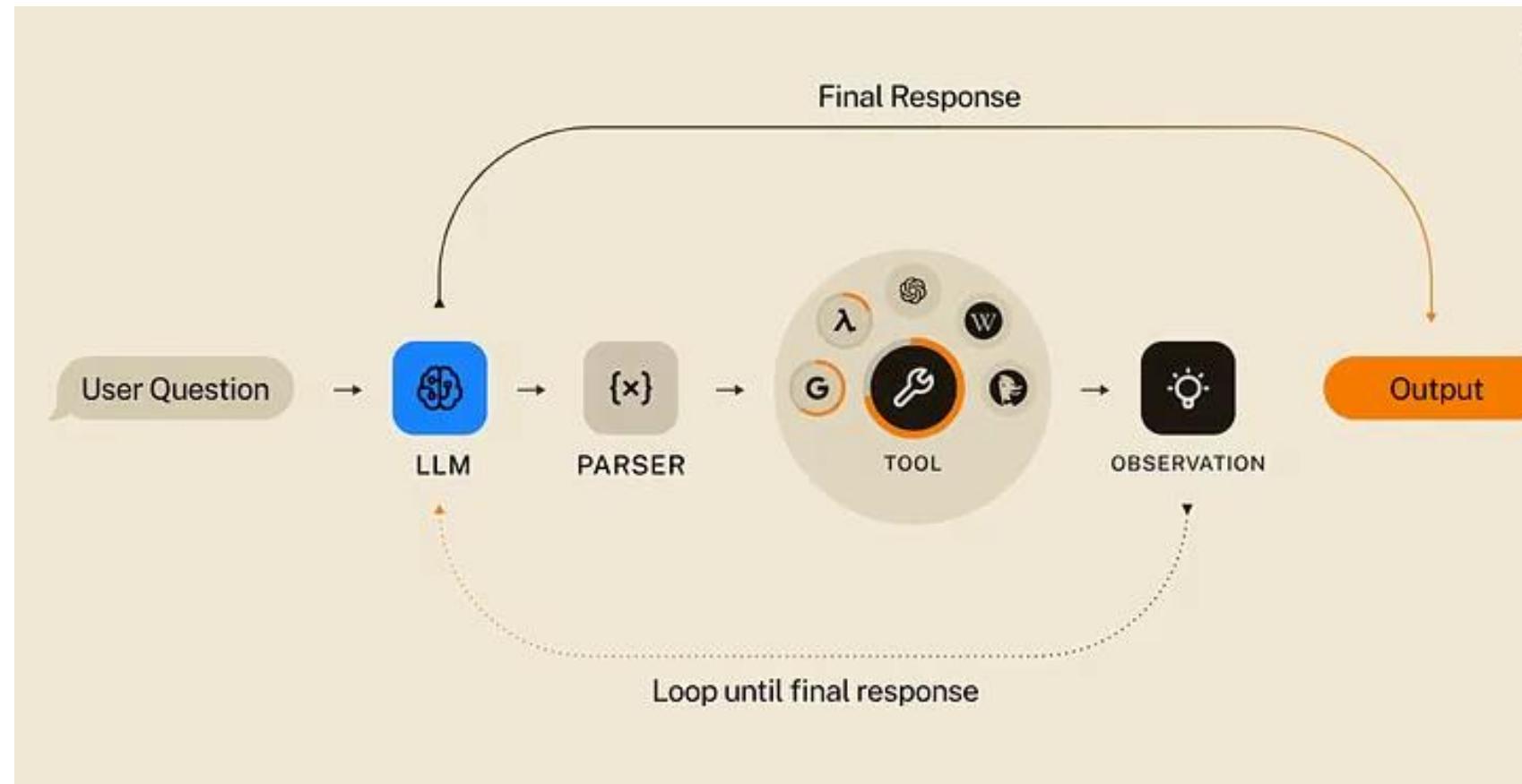


<https://medium.com/@Shamimw/understanding-langchain-tools-and-agents-a-guide-to-building-smart-ai-applications-e81d200b3c12>

How do we give tools to an LLM?

Use the system prompt to provide textual descriptions of available tools to the model:

```
● ● ●  
  
system_message="""You are an AI assistant designed to help users efficiently and accurately. Your primary goal is to provide helpful, precise, and clear responses.  
  
You have access to the following tools:  
{tools_description}  
"""
```



The spectrum of “Agency”

Agency Level	Description	How that's called	Example Pattern
☆☆☆	LLM output has no impact on program flow	Simple Processor	<code>process_llm_output(llm_response)</code>
★☆☆	LLM output determines an if/else switch	Router	<code>if llm_decision(): path_a() else: path_b()</code>
★★☆	LLM output determines function execution	Tool Caller	<code>run_function(llm_chosen_tool, llm_chosen_args)</code>
★★★	LLM output controls iteration and program continuation	Multi-step Agent	<code>while llm_should_continue(): execute_next_step()</code>
★★★	One agentic workflow can start another agentic workflow	Multi-Agent	<code>if llm_trigger(): execute_agent()</code>

https://huggingface.co/docs/smolagents/conceptual_guides/intro_agents

When to Use an Agentic System?

When you need an LLM to determine the workflow of an app.

If a fixed workflow fails too often → Need more flexibility

When to Use Predefined Workflows?

LLMs can determine workflows, but they're often unnecessary.

Ask: Do I need workflow flexibility?

AI Agent Frameworks

- LangGraph
- smolagents
- LlamaIndex



LangGraph



smolagents



LlamaIndex

When to Use an Agentic Framework?

Simple tasks may not require the use of agentic frameworks.

- Mostly, predefined workflows are sufficient to handle user requests.
- Simple, straightforward chain-of-thought prompts using plain code are enough.

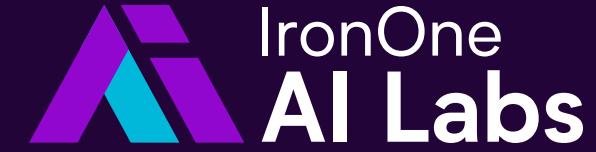
However, agentic frameworks are suitable for complex workflows.

- Helps when the LMs use multiple agents.

Demo – Create Agent System using smolagents



<https://bit.ly/3RDMVwx>



Explainability of Language Models



Techniques that provide information on how a machine learning model makes predictions.

Why it's important?

Transparency	Opens up the 'black-box' complex models.
Trust	Enables users to understand and trust the model's decisions (especially in regulated domains)
Compliance	Assists in meeting regulatory requirements (GDPR, FDA).
Debugging	Possible to identify and fix errors and biases, and improve models

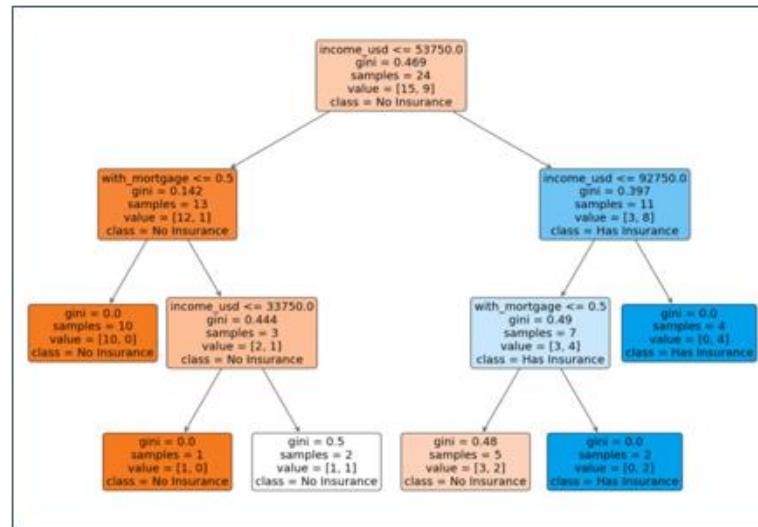
Traditional ML Explainability Approaches

Following are some techniques that are used to interpret traditional ML models.

Simple and Interpretable Models

Simple models, such as Logistic Regression, Linear Regression, and Decision Trees are easy to interpret.

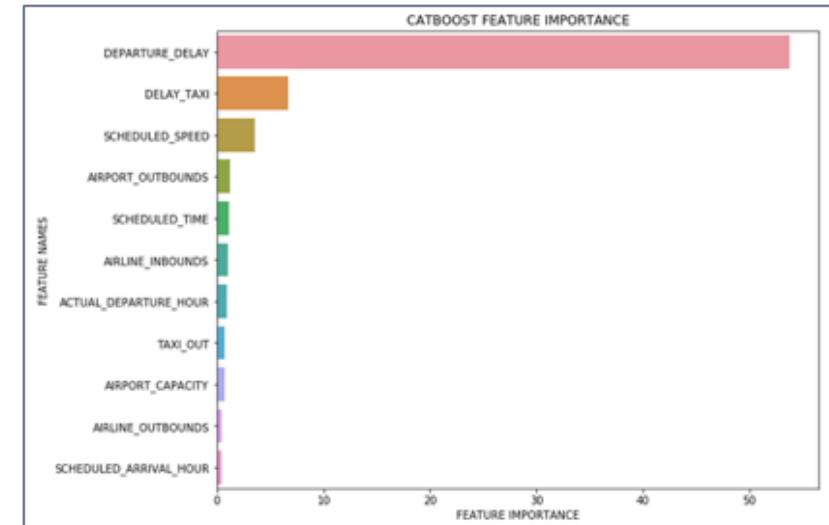
However, simple models may fail when handling complex data.



Feature Importance

Feature importance provides an overall ranking of the features that affect all of the predictions.

However, this doesn't capture feature interaction or how features affect specific predictions.

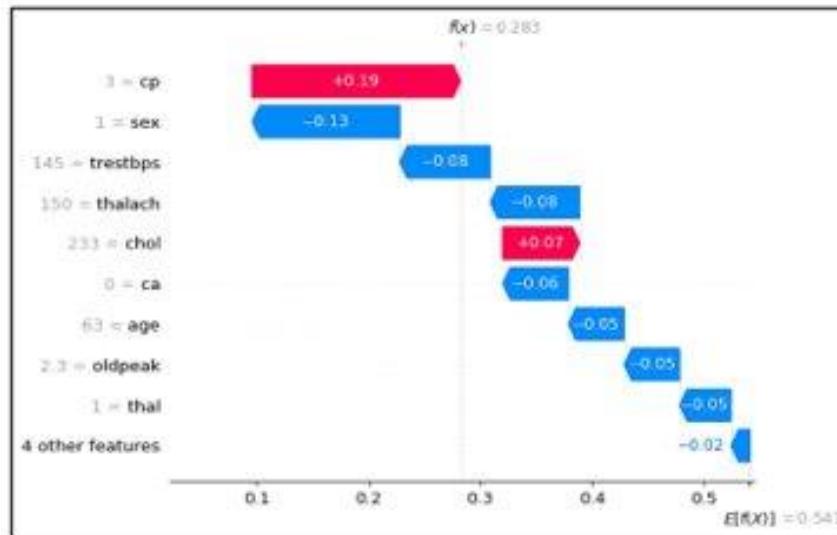


Traditional ML Explainability Approaches

Local Explanations

These techniques (SHAP, GradCAM, LIME) can explain how features affect individual predictions.

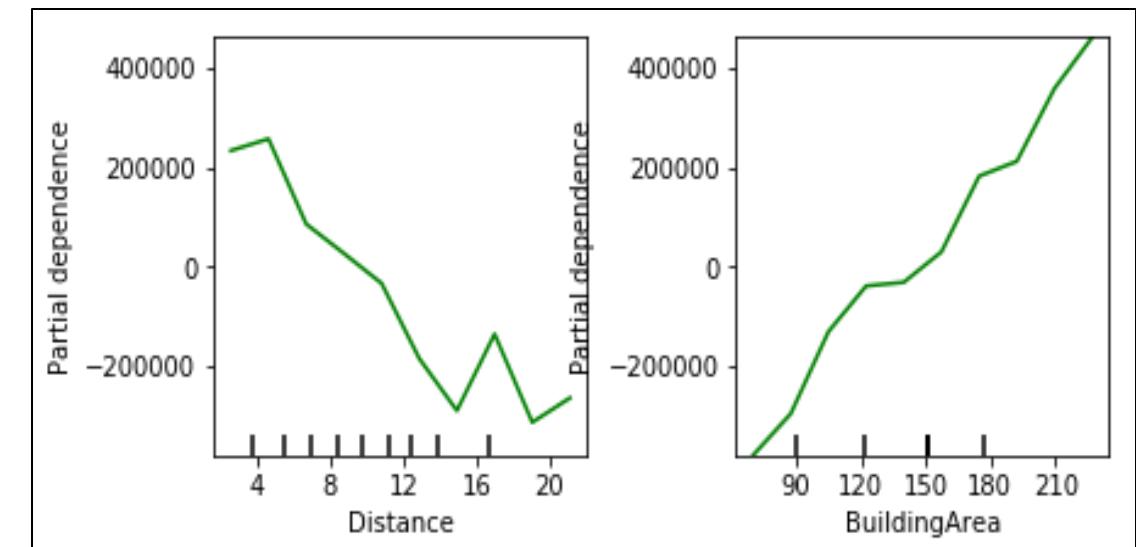
However, these techniques can be computationally expensive and explanations may vary with similar inputs.



Partial Dependence Plots

PDP plots can visually display the relationship between a feature and the target.

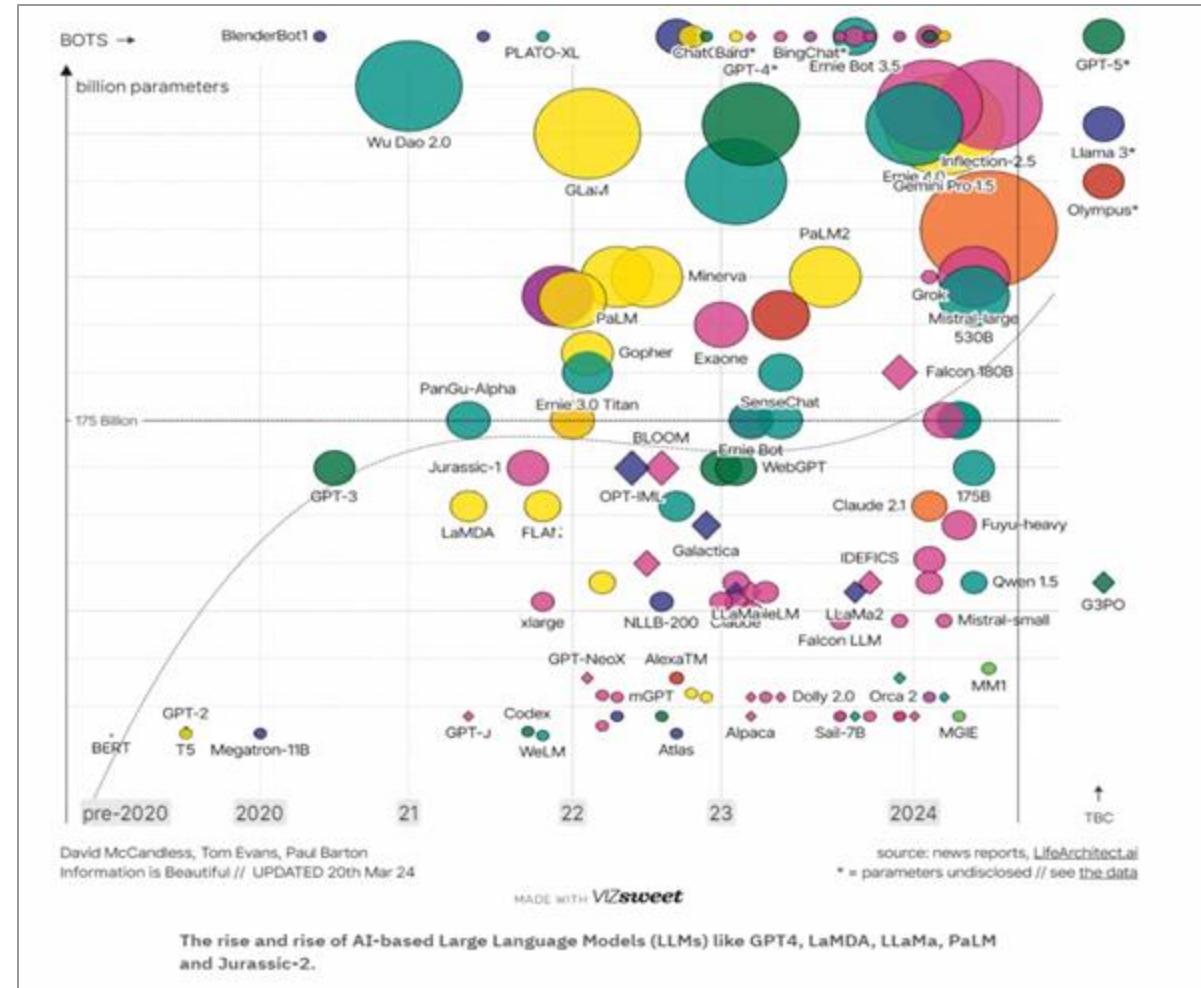
However, each feature is considered independently.

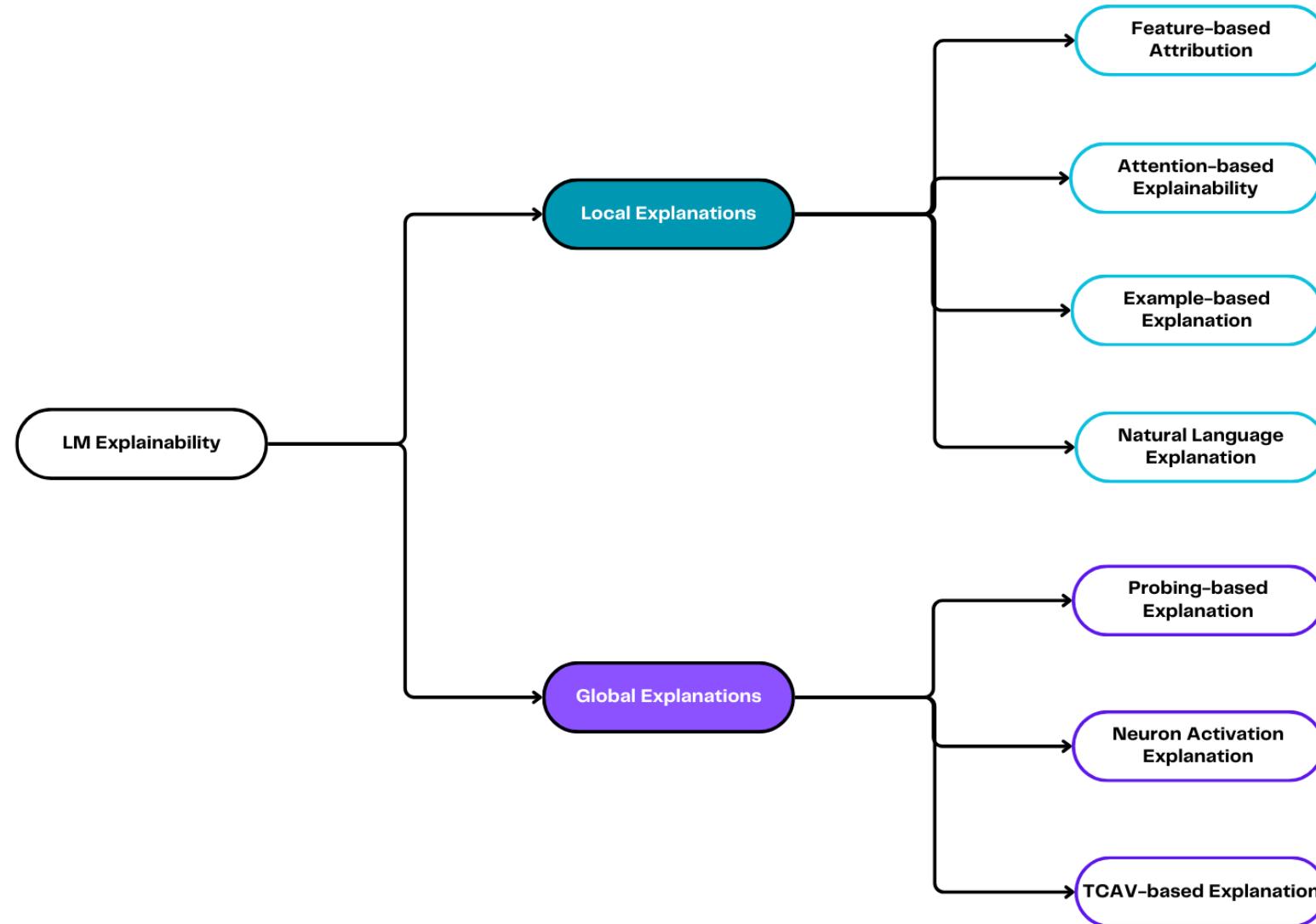


However, for LMs?

Explainability for Language Models, especially LLMs, can be challenging due to:

- Having **millions and billions of parameters**, with complex attention layers and non-linear activations.
- The **long range dependencies** between the tokens, and wide context windows (even past prompts).





Local explanations provide an understanding of how a language model makes a prediction for an instance.

Feature Attribution Explanation

Attempts to determine the relevance of each input feature (tokens) to the model's prediction.

Perturbation-based methods (such as SHAP) and gradient-based methods.

Context: In 1899, John Jacob Astor IV invested \$100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his **Colorado Springs experiments**.

Question: What did Tesla spend Astor's money on?

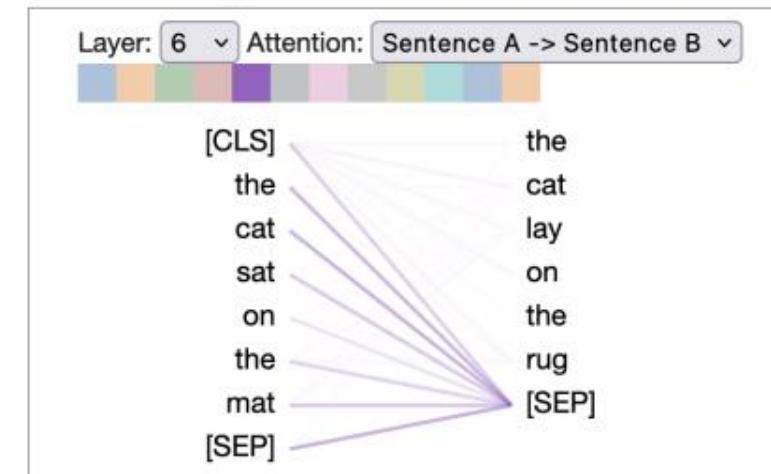
Confidence: 0.78 → 0.91

Explainability for Large Language Models: A Survey:
<https://arxiv.org/pdf/2309.01029>

Attention-based Explanation

Analyzes the attention weights to understand and highlight the important parts of an input sequence.

Heatmaps and bipartite graphs to visualize attention heads.



Explainability for Large Language Models: A Survey:
<https://arxiv.org/pdf/2309.01029>

Example-based Explanation

Aims to understand the model's behaviour by examining how the output changes relative to specific input instances.

Adversarial examples and counterfactual

Original text: The characters, cast in impossibly contrived situations, are totally estranged from reality (**Negative**).

Perturbed text: The characters, cast in impossibly engineered circumstances, are fully estranged from reality (**Positive**)

Original text: It is great for kids (**positive**).

Negation examples: It is not great for kids (**negative**)

Explainability for Large Language Models: A Survey:
<https://arxiv.org/pdf/2309.01029>

Natural Language Explanation

Generating text to explain how the model makes a decision for an input sequence.

Chain-of-thought prompting and reasoning.

Question: While eating a **hamburger with friends**, what are people trying to do?

Choices: have fun, tasty, or indigestion

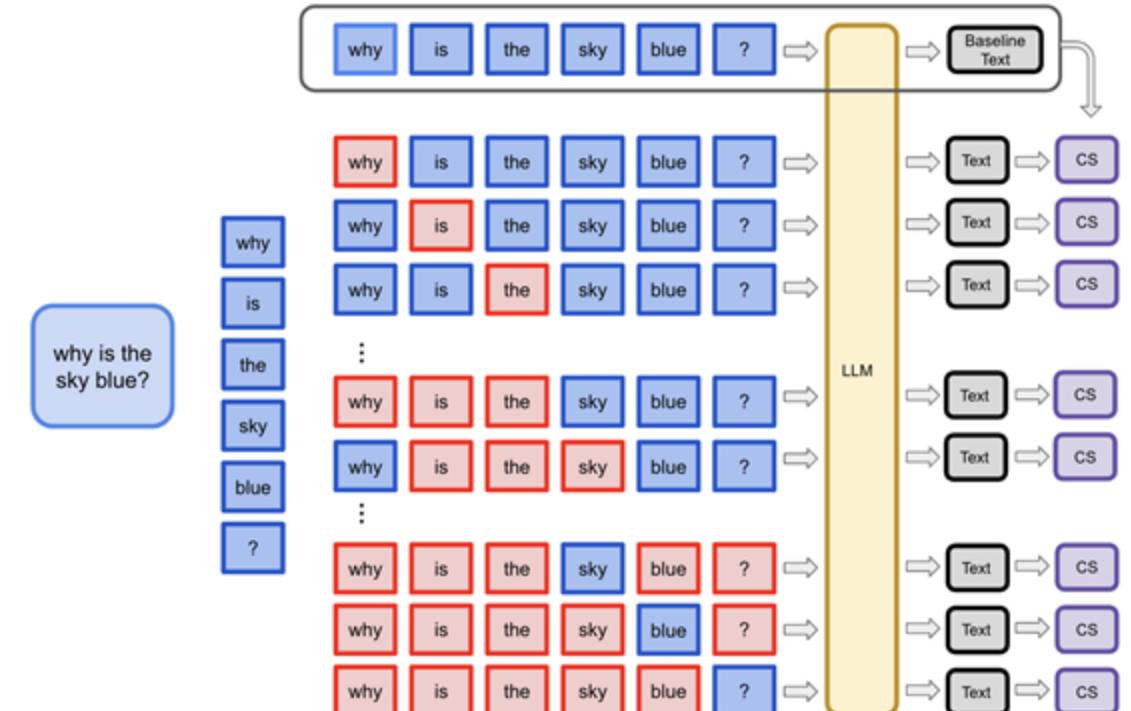
Explanation: Usually a hamburger with friends indicates a good time.

Explainability for Large Language Models: A Survey:
<https://arxiv.org/pdf/2309.01029>

Feature Attribution Explanation

TokenSHAP

- Adapts SHAP to compute each token's contribution to model output.
- TokenSHAP measures marginal contribution of each token to output probabilities.
- Captures interdependence between tokens.
- Attribution scores are generated, which shows each token's positive or negative influence on the output.



TokenSHAP: Interpreting Large Language Models with Monte Carlo Shapley
Value Estimation: <https://arxiv.org/abs/2407.10114>

Demo – TokenSHAP



<https://bit.ly/43IClq3>

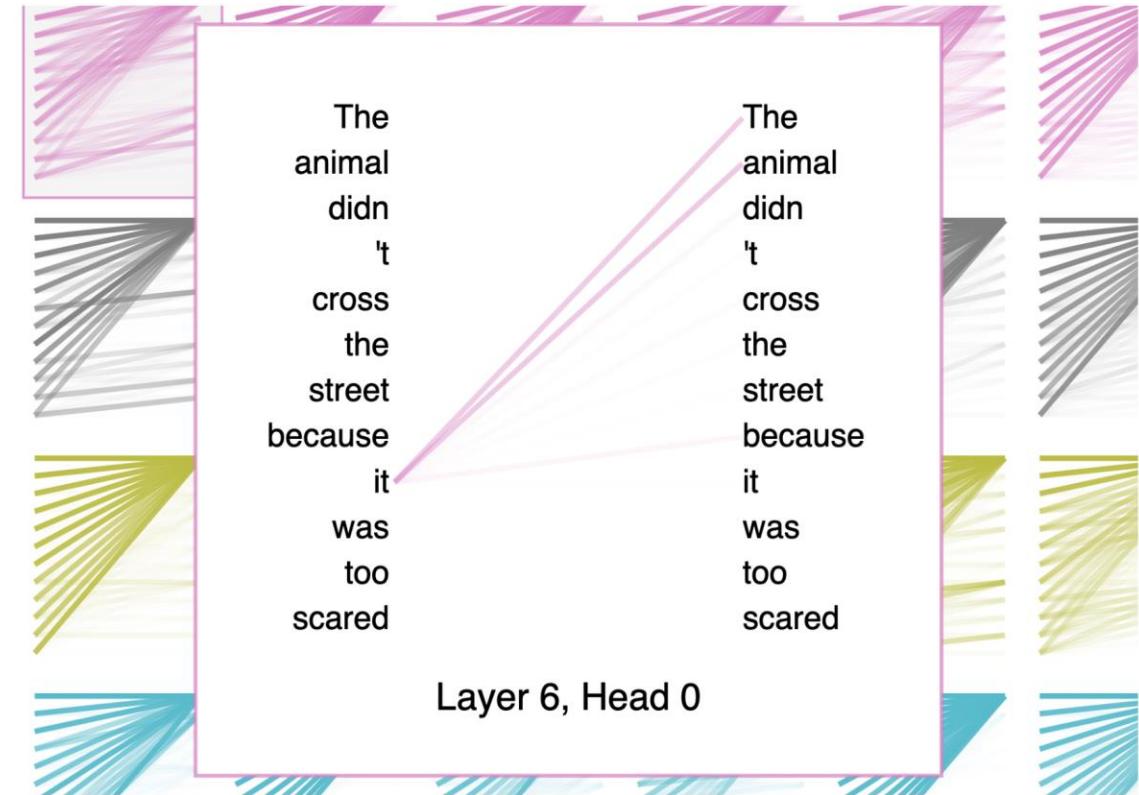
Attention-based Explanation

Attention Visualization

This shows what the model looks at when generating each output; the attention weights between tokens can be visualized.

Different types of attention visualizations:

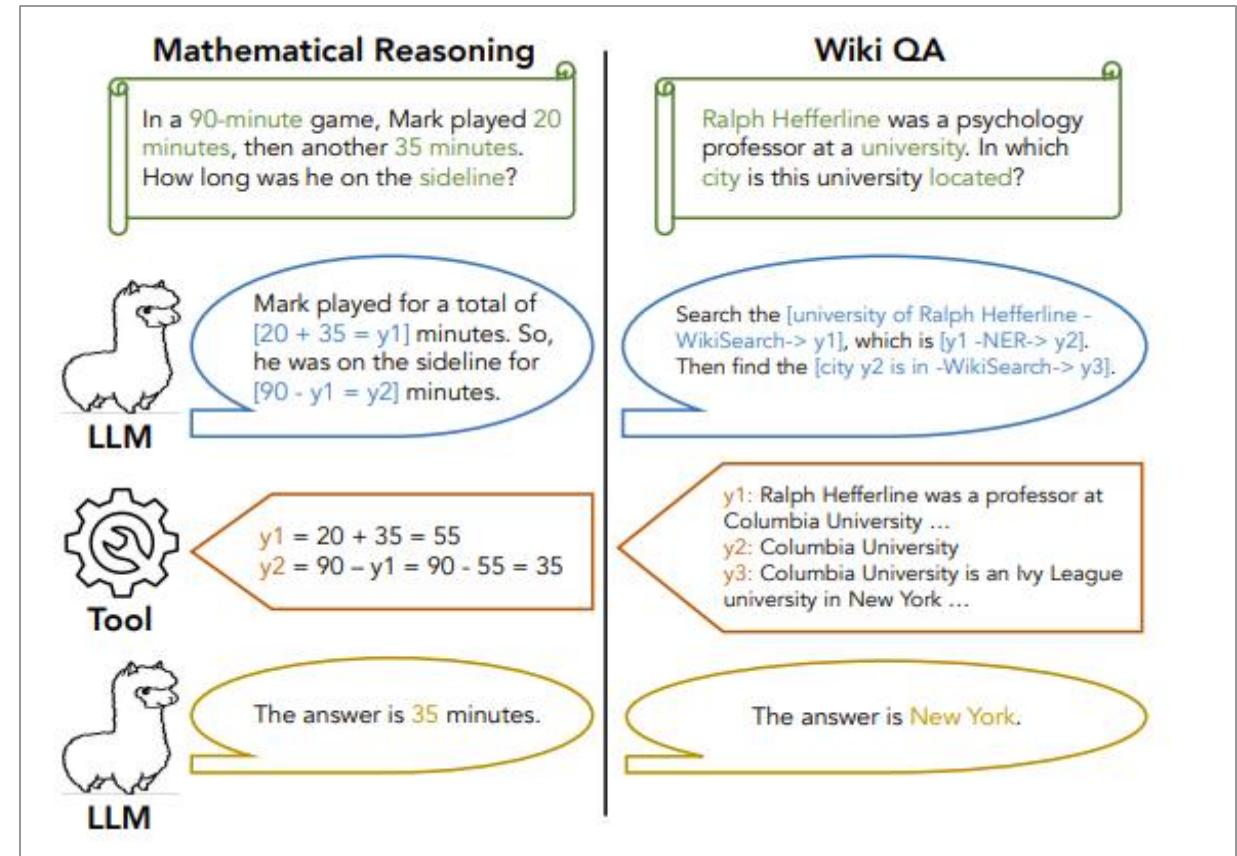
- Self-attention
- Cross-attention
- Layer-by-layer analysis



Reasoning models enhance explainability by **showing their decision making process step-by-step** which can be easily followed.

This allows the developers to identify where the errors occur and also to verify if the reasoning is acceptable.

**R1 by Deepseek, o1 and o3-mini by OpenAI,
Claude's 3.7 Sonnet**



Efficient Tool Use with Chain-of-Abstraction Reasoning:
<https://arxiv.org/pdf/2401.17464>

Global explanation provides a high-level understanding of what the language models learned and how they operate.

Probing-based Explanation

Training a shallow classifier on frozen pre-trained language models, and/or also using specific datasets and/or prompts to understand the linguistic knowledge within the model's parameters.

- **Classifier-based probing**
- **Probing attention heads**
- **Probing with prompt search**

Neuron Activation Explanation

Examining individual neurons to determine their importance for model's performance and linguistic properties.

- **Ablation experiments**
- **Using a LLM (such as GPT-4) to generate explanations for neural activations in another language model (such as GPT-2) [Study by OpenAI]**

TCAV-based Explanation

Testing with Concept Activation Vectors (TCAV) uses a set of abstract pre-defined concept vectors to understand the inner workings of the model by checking if it recognizes specific concepts.

Challenges in LMs explainability

Scale and Complexity

Models with hundreds of billions of parameters might lack detailed explanations and explanations might oversimplify model performance.

Reliability Issues

Explanations may be misleading or incomplete, and similar inputs might provide different explanations.

Implementation Tradeoffs

Large models might cost computational resources and time; and API access models might have issues in implementing explainability.

Thank you

USA

9600 Great Hills Trail, Suite 150W
Austin, Texas 78759

Sri Lanka

Level 20, Parkland Building,
No.33, Park Street,
Colombo 02, Sri Lanka

Germany

The Drivery,
Mariendorfer Damm 1
12099 Berlin

Malaysia

Level 31, Menara Prestige,
No. 1 Jalan Pinang,
Kuala Lumpur, 50450,
Malaysia

India

Vaswani Chambers,
1st floor, 264-265,
Dr. Annie Besant Road,
Worli Mumbai,
MH 400030

IronOne



Recap

- 1. Small Language Models**
- 2. Supervised Fine-Tuning**
- 3. Preference Alignment**
- 4. AI Agents**
- 5. Explainability**

Questions?

github.com/IronOne-AI-Labs/Hands-on-with-Language-Models-and-AI-Agents



Theekshana

theekshanas@irononetech.com

<https://www.linkedin.com/in/theekshana-samaradiwakara-047261177/>

<https://huggingface.co/theekshana>

Vaseekaran

vaseekarav@irononetech.com

<https://www.linkedin.com/in/vaseekaran/>

Purusanth

purusanths@irononetech.com

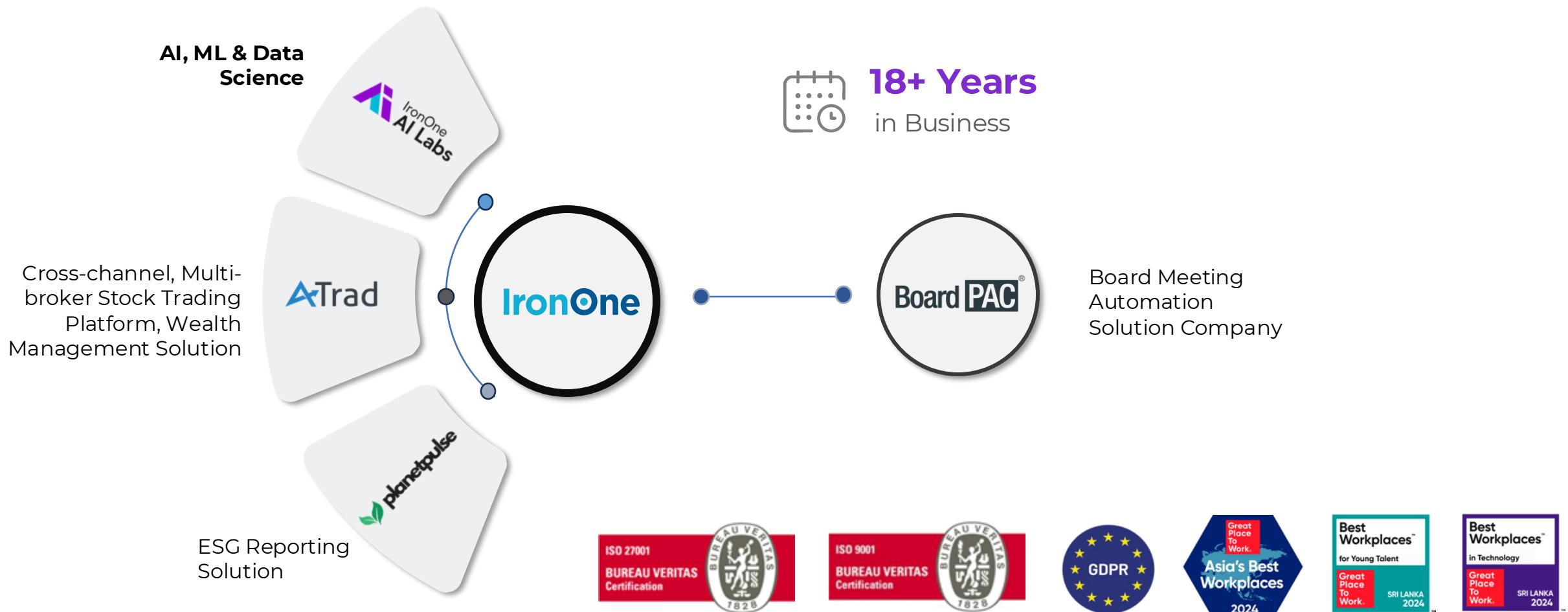
www.linkedin.com/in/purusanth-shanmukanathan-4a976615b



**AI Solutions
For Digital
Transformation**

IronOne

IronOne Group



Global Footprint



-  **500+** Clients
-  **50,000+** Global Users
-  **40+** Countries
-  **18+** Years in Business
-  **8** Global Offices

Value Chain

AI Strategy & Consulting

We work closely with you to understand your unique business goals and develop a tailored AI strategy that aligns with your overall vision.



Custom AI Development

Our team of experienced AI engineers will design and build cutting-edge AI solutions specific to your needs.



Ongoing Support

We provide comprehensive support throughout the entire AI life-cycle, from ideation to implementation and beyond.

Seamless Integration

We ensure smooth integration of your AI solution with your existing systems and infrastructure.

Research & Development

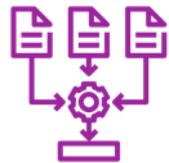
Strength in extensive R&D drives unparalleled innovation in crafting tailored AI solutions.

Our Expertise



AI Agents Automating Knowledge work

AI Agents help you to make decisions and take actions using tools to **complete a task more autonomously**, often with the ability to learn and adapt over time



Leverage LLMs on local data

Retrieval Augmented Generation (RAG) is a compound AI system that helps to improve accuracy of responses with **external knowledge**



Building Small Language Models

Building optimized and efficient models **specialized for specific tasks and domains**. It offers faster processing, lower costs, and easier deployment.

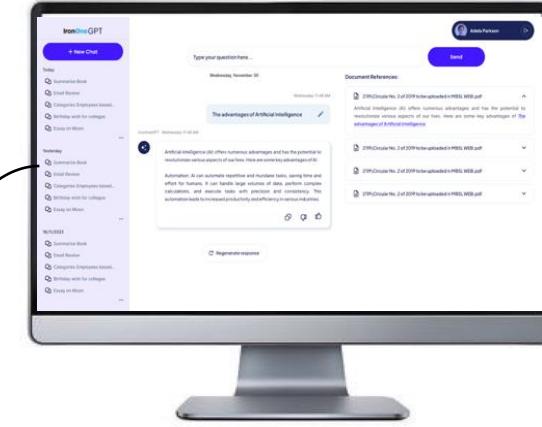
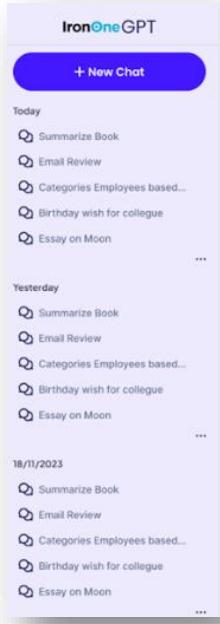


Risk Modeling and behavioral analysis

Building **Explainable & Fairness risk models and agents** for decision making by capturing hidden behavioral patterns in the financial, insurance and medical domains.

Customized AI Agents for your Business

Single Agent Platforms



Real-time data insights, and scenario analysis, all without compromising the privacy or data security of the board.

Report Writing Agent

Regulations GPT

An AI-powered engine that consolidates and analyzes regulatory documents, providing compliance teams with instant access to updated information on industry specific regulations.

RAG based Chatbots using LLMs and SLMs

Regulations Monitoring Agent

Multi AI Agent Platforms

AI Agent Orchestration

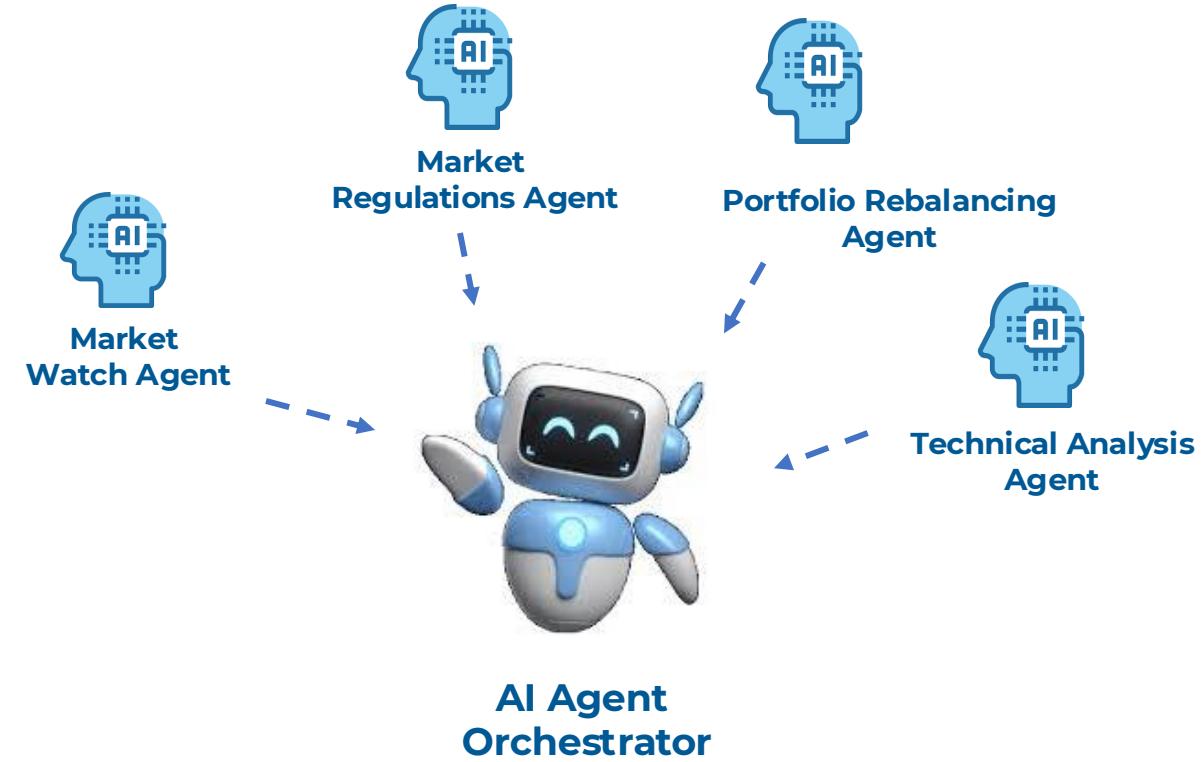


Personal Stock Trading Assistant

ATrad is a comprehensive trading and wealth management platform designed for financial institutions, brokerage firms, and wealth managers. It offers advanced solutions for managing equities, bonds, treasury bills, and unit trusts, providing a seamless experience for investment professionals.

Revolutionize Stock Trading with the power of AI Agents

- Education and awareness for new and existing users
- Realtime analysis of market data
- Fundamental & Technical Analysis of the Market
- Portfolio Rebalancing Agent
- Stock Recommendations and Market Movement Forecasting Integrations (In Progress)



Industry Automation using Agentic AI

Autonomous Intelligence, Unmatched Efficiency



Impact on Industry 4.0

From rule-based automation to **intelligent, autonomous operations**.

Empowering industries with AI-driven **self-learning systems**.

Why it Matters?

- Enhanced Efficiency
- Scalability
- Real-Time Adaptability
- Predictive Intelligence
- Realtime Intelligence



Tailored AI Solutions

Selected Cross-Industry Use Cases

Use Case - Security Risk Monitoring

Situational Awareness Monitoring Platform

Customer

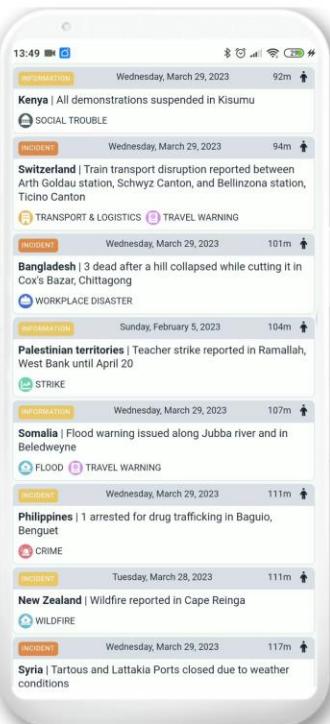
Threat Intelligence Company, established in 2014 and headquartered in Brussels, Belgium, is a leading provider of AI-based threat intelligence solutions. Their platform offers real-time alerts on safety and security risks, utilizing data analytics and geofencing to assist clients in managing travel risks, situational awareness, and business continuity.

Problem Statement

Harnessing AI automation to deliver real-time threat intelligence and risk monitoring by analyzing news and social media, precisely identifying geo-location, risk type, and severity. With their existing language models their location identifier was achieving **45% accuracy** which was very low.

Solution

Leveraging **Small Language Models** and AI-driven threat intelligence agents to automate real-time risk analysis through advanced Natural Language Processing. With state-of-the-art models we achieved in improvement of **91% accuracy**.



Risk Monitoring Agent

Story Creation Agent

Use Case – Financial Risk

Predicting Customer Debt Default

Customer

The client is a major asset and credit portfolio management organization in the USA. The client is a beneficial owner of billions in underlying credit receivable assets.

Problem Statement

The client faced **significant losses from acquired Credit Card debt portfolios due to high default rates**, Inadequate risk assessment and recovery strategies threatened portfolio viability, urgently requiring improved credit evaluation and default mitigation measures.

Total Annual Loss due to customer Default

\$12,600,000

Annual Net Savings

**\$3.5 Mn
(27%)**

Solution

Using power of AI to analyze thousands of data points—**payment history, market trends, macro economic data & borrower behavior**—to spotlight hidden risks and rank portfolios by threat level.

Models adhere to **Explainable AI and Fairness principals**.

Output - AI-Driven Delinquency Risk Score, 6-12 months ahead.

This enabled Fidem to take Soft & Hard Actions to prevent default, well in advance

AI Model Statistics

- 88% Overall Model Accuracy
- 98% Accuracy of Predicting Non-Default
- 65% accuracy of predicting default 6-12 months ahead



JOHN SMITH

Date of Birth
02.06.1996

Address
123 Broadway, New York,
NY 10001

State
New York

Annual Income
USD 150,000

Client ID
647228R

760

Overall ForZee Rating

Use Case - Automotive

Extending Electric Vehicle Battery Efficiency

Customer

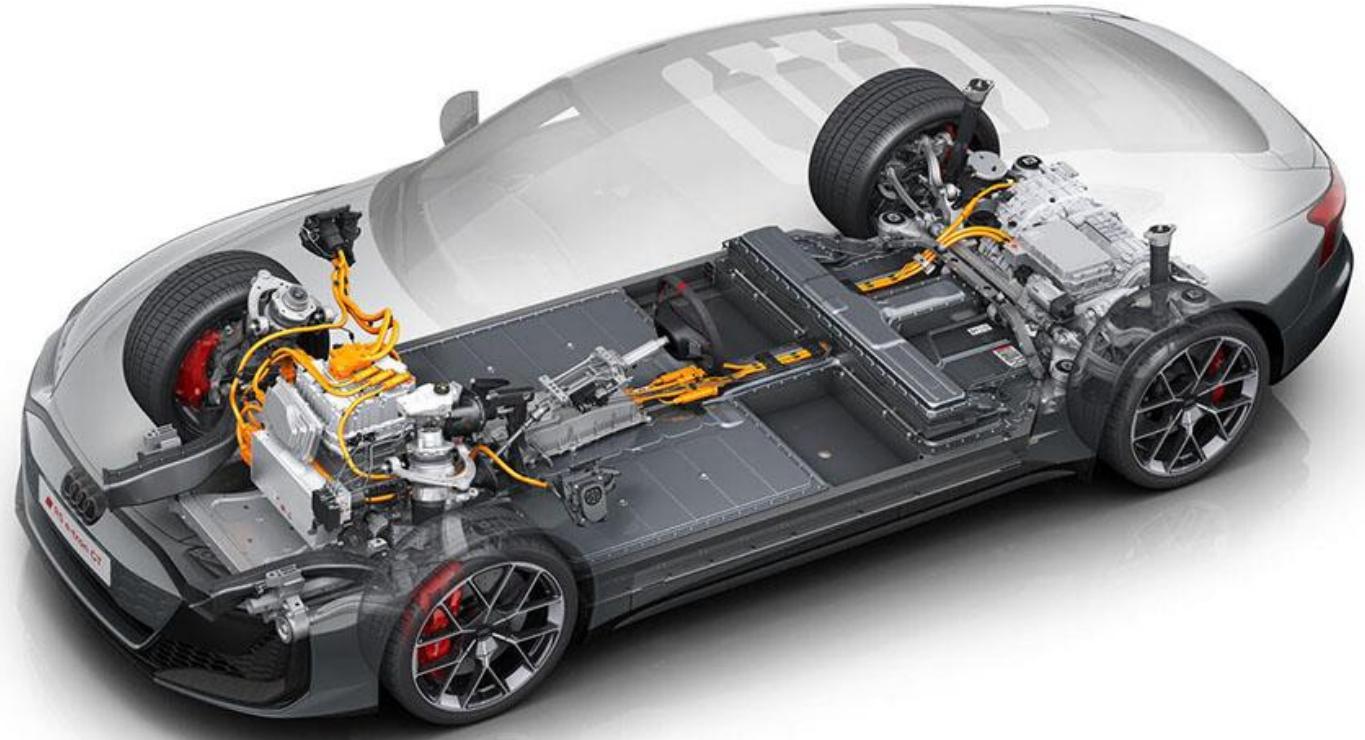
German Automotive component Supplier - With over 65 years of expertise, it operates globally, enhancing vehicle efficiency, sustainability, and comfort while expanding into eco-friendly industrial applications. Specializes in automotive and industrial functional solutions, focusing on noise control, fluid management, thermal management, and polymer-based components.

Problem Statement

Electric vehicle (EV) batteries experience energy loss due to inefficiencies in charging, discharging, and thermal management, reducing overall capacity and lifespan. Enhancing battery efficiency requires optimizing energy retention, minimizing degradation, and improving thermal regulation for prolonged performance.

Solution

Using **Physics-informed Machine Learning** to Monitor sensory data to optimize battery efficiency. Building soft sensors based on Machine Learning to reduce costs. This will help to improve thermal regulation.



Use Case - MedTech

Breast Cancer Risk Score using Genomics

Collaboration Partner

IronOne AI Labs collaboration with Industry Experts and Academics in the Preventive Medicine & Public Health Sector.

Problem Statement

Develop an AI-driven predictive risk score for cancer using genomic data to improve early detection and personalized treatment of **missense variants**. This is generally classified as “**uncertain significance**”

The model will analyze genetic markers to predict individual cancer risk, helping clinicians make informed decisions on missense variants. Key challenges include data complexity, model accuracy, interpretability and generalization.

Solution

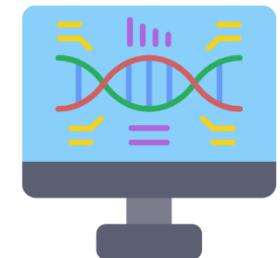
Pioneered a **precision-engineered genomic annotator** for Asian genetic diversity and a transparent AI-powered predictive analytics engine, transforming patient-specific data variants into actionable, personalized Risk Scores for proactive breast cancer prevention



Most common cancer worldwide
2.3 million new cases diagnosed



Leading cause of cancer-related deaths
Claims 670,000 lives globally each year.



AI / ML Engineering Talent

Tech Stack & Skills Set

Machine Learning and Deep Learning



Generative AI / Natural Language Processing



Optimization and MLOps



Other





**Innovation by Collaboration
Fueling Industry
Growth with Campus
Brilliance.**

Research & Development University Collaborations



On-going R&D Projects

- Automated candlestick chart pattern detection using Deep Learning in collaboration with University of Moratuwa
- Machine Learning models for predicting Environmental, Social, and Governance (ESG) scores, developed in collaboration with Informatics Institute of Technology



Paper Title: Deep Learning-Based Personalized Stock Recommender System

Published by:

Students from University of Moratuwa (UOM) & IronOne team

Was given the prestigious opportunity to be presented at ICONIP 2023 conference. This was a joint initiative between IronOne AI Labs & UOM.



ICONIP is the annual flagship conference of Asia Pacific Neural Network Society (APNNS)

Nurturing Young Talent

At IronOne AI Labs, we are committed to fostering the next generation of AI professionals through strategic university collaborations, CSR initiatives, and real-world AI projects.

- **Internship Program:** (15% of Team Comprises Interns): We provide hands-on experience to young talent by integrating them into real AI projects. This allows them to gain industry exposure and contribute meaningfully to innovation.
- **AI Clubs for Schools (National Government Program)** : We actively engage with schools to introduce AI concepts to students through workshops and mentorship, inspiring the next wave of AI talent.





**Innovation by Collaboration
Team, Culture &
Standards**

Corporate Culture

We are a Great Place To Work

- Best Workplaces in Asia 2024 : #50 in S&M Category
- Best Workplaces in Sri Lanka 2024 -#4 in Micro Workplaces - IT/ ITES Industry
- Best Workplaces™ in the Technology Industry 2024
- Best Workplaces for Young Talent 2024
- Best Workplaces for Women in Sri Lanka 2024



CERTIFICATIONS

Standards & Compliance

Strong emphasis on security, compliance and data protection of sensitive information in highly regulated industries

- **ISO 27001 & ISO 9001 Certified** & use military grade **AES-256 Encryption** Algorithm to secure all sensitive information.
- We are **Carbon Neutral** since 2021 with **ISO 14064-1 Certification**
- Systems audited and vulnerabilities assessed regularly by 3rd Party Auditors and Regulators.



AES 256 Encryption



Thank you

IronOne



USA

9600 Great Hills Trail, Suite 150W
Austin, Texas 78759

Sri Lanka

Level 20, Parkland Building,
No.33, Park Street,
Colombo 02, Sri Lanka

Germany

The Drivery,
Mariendorfer Damm 1
12099 Berlin