HUAWEI | EUROPEAN RESEARCH CENTER

Efficient training of machine learning algorithms

— Optimisation of results at reduced costs

Heiko Joerg Schick
Chief Architect & Industry Expert | Advanced Computing, Artificial Intelligence & Semiconductor

Presenting the work of many people at Huawei

2024-03-19 v2 | Generative AI DACH 2024, Berlin, Germany

HUAWEI TECHNOLOGIES DÜSSELDORF GmbH

HUAWEI

Agenda

Huawei Pangu-Weather

- Data and settings
- Computational costs

Implementation of end-to-end lifecycle in AI projects

Fine-tuning

- GPT assistant training pipeline | LLMs model sizes over time | Full-parameter fine-tuning
- How is Low-Rank Adaption (LoRA) different?
- Number of trainable parameters | Percent of total parameters
- QLoRA

Retrieval-augmented generation (RAG) system

- Overcoming challenges of LLMs
- General Purpose AI (GPAI) classification and key requirements for providers
- Fine Tuning vs. Retrieval Augmented Generation
- Basic chatbot architecture | Example
- Retrieval-augmented generation (RAG) architecture | Example | Many questions ?!?

Closing remarks

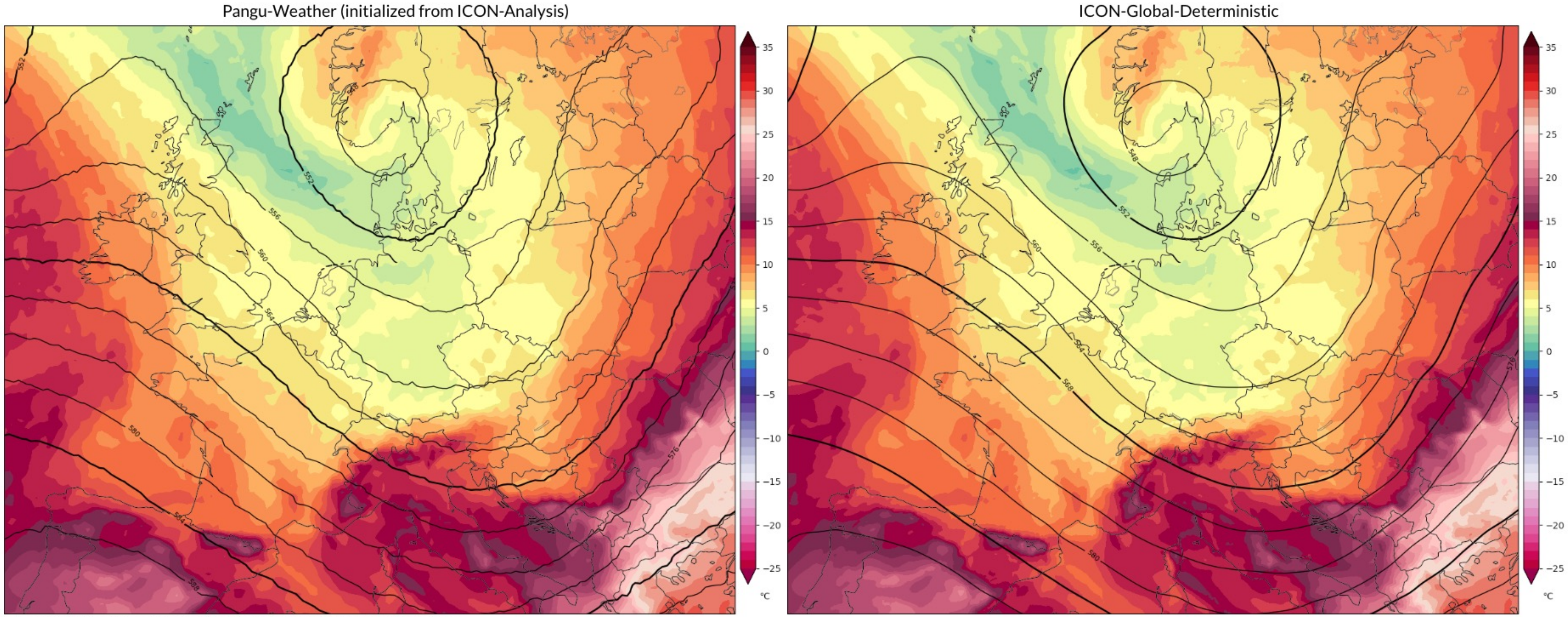
Heiko Joerg Schick
Chief Architect & Industry Expert
Advanced Computing, Artificial Intelligence & Semiconductor
Munich Research Center

HUAWEI TECHNOLOGIES
Duesseldorf GmbH
Riesstrasse 25, D0
80992 Munich

Mobile +49-151-54682218
E-mail heiko.schick@huawei.com

LinkedIn https://www.linkedin.com/in/heikojoergschick/

Huawei Pangu-Weather – ICON Comparison Maps (Bi et al., 2023)



Shaded: Temperature at 850hPa, Lines: Geopotential Height at 500hPa in gpdm
Run: Wed, 26 Jul 2023, 12UTC, Valid Date: Wed, 26 Jul 2023, 12UTC (+0h)

With the current setup, a single 7-day forecast with Huawei Pangu-Weather consumes 14 Wh of energy. For a 7-day forecast with the ICON model, the energy consumption amounts to approximately 30000 Wh. This simple calculation of course does not include the energy consumption required to generate the training data and to train the model.

Huawei Pangu-Weather (Bi et al., 2023)

— Data and settings

- The dataset includes the **5th generation of ECMWF reanalysis (ERA5) data**, which is publicly available.
- It comprises hourly reanalysis data from the year 1940 onwards.
- For our study, we used data from **1979 to 2017 for training** purposes, **2019 data for validation**, and **2018, 2020, and 2021 data for testing** to ensure a fair comparison with WeatherBench.
- The dataset contains a variety of surface and upper-air variables across **37 pressure levels**.
- Specifically, we selected **four surface variables** (2m temperature, u- and v-components of 10m wind speed, mean sea-level pressure) and **five upper-air variables** (geopotential, specific humidity, temperature, u- and v-components of wind speed) at **13 selected pressure levels** (ranging from 50hPa to 1000hPa).
- Although the **full dataset exceeds 2000 TB** in size, our analysis used approximately **60 TB** of data.

Huawei Pangu-Weather (Bi et al., 2023)

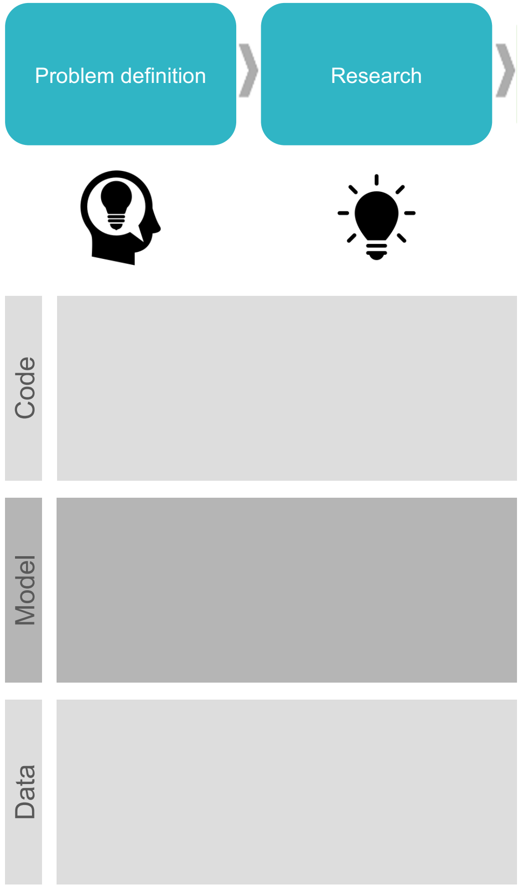
— Computational costs

- The training phase involves each forecast model having approximately **64 million parameters**.
- Each model is trained for **100 epochs** over **16 days** using 192 NVIDIA Tesla V100 GPUs, indicating that the models have not yet converged.
- During inference, each forecast takes about **1.4 seconds** on a single V100 GPU.
- Inference can also be carried out on a CPU, albeit with a longer processing time.
- Executing a 7-day global forecast involves running the 24-hour model seven times, totalling **less than 10 seconds**.
- Faster inference facilitates easier ensemble forecasting.

Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements



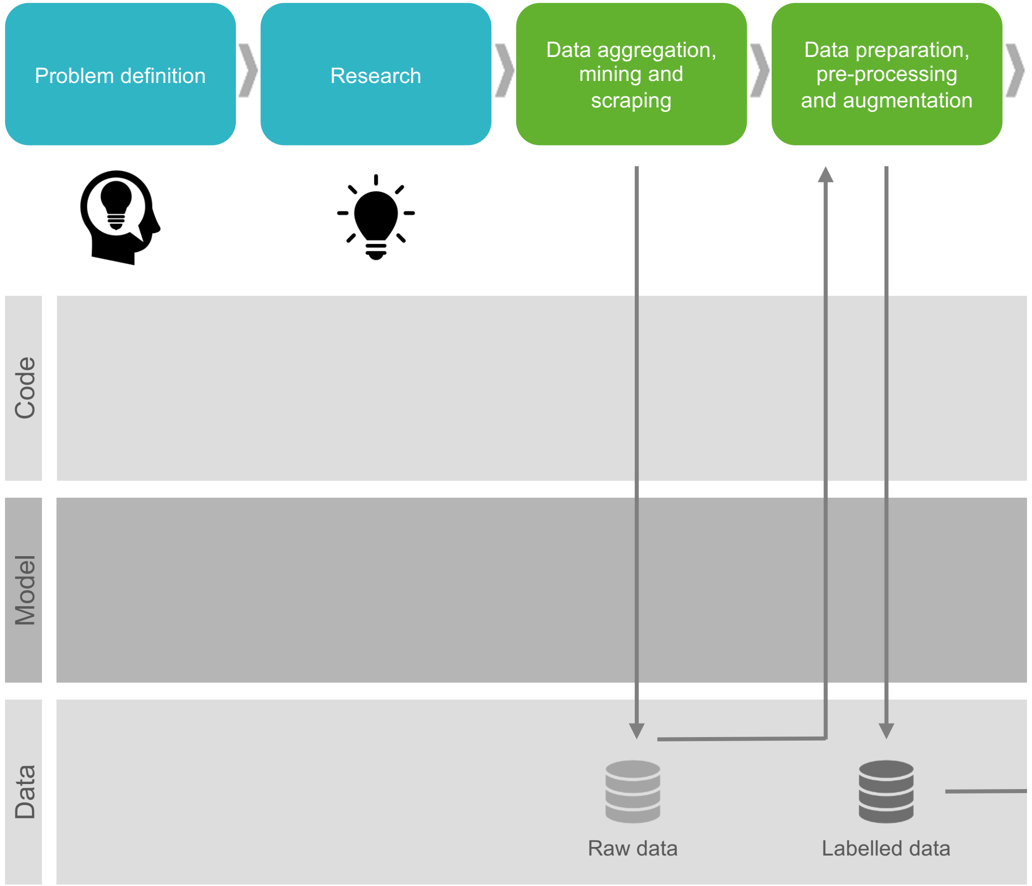
Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

- Data reformatting
- Data cleaning
- Data normalisation
- Data augmentation



Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

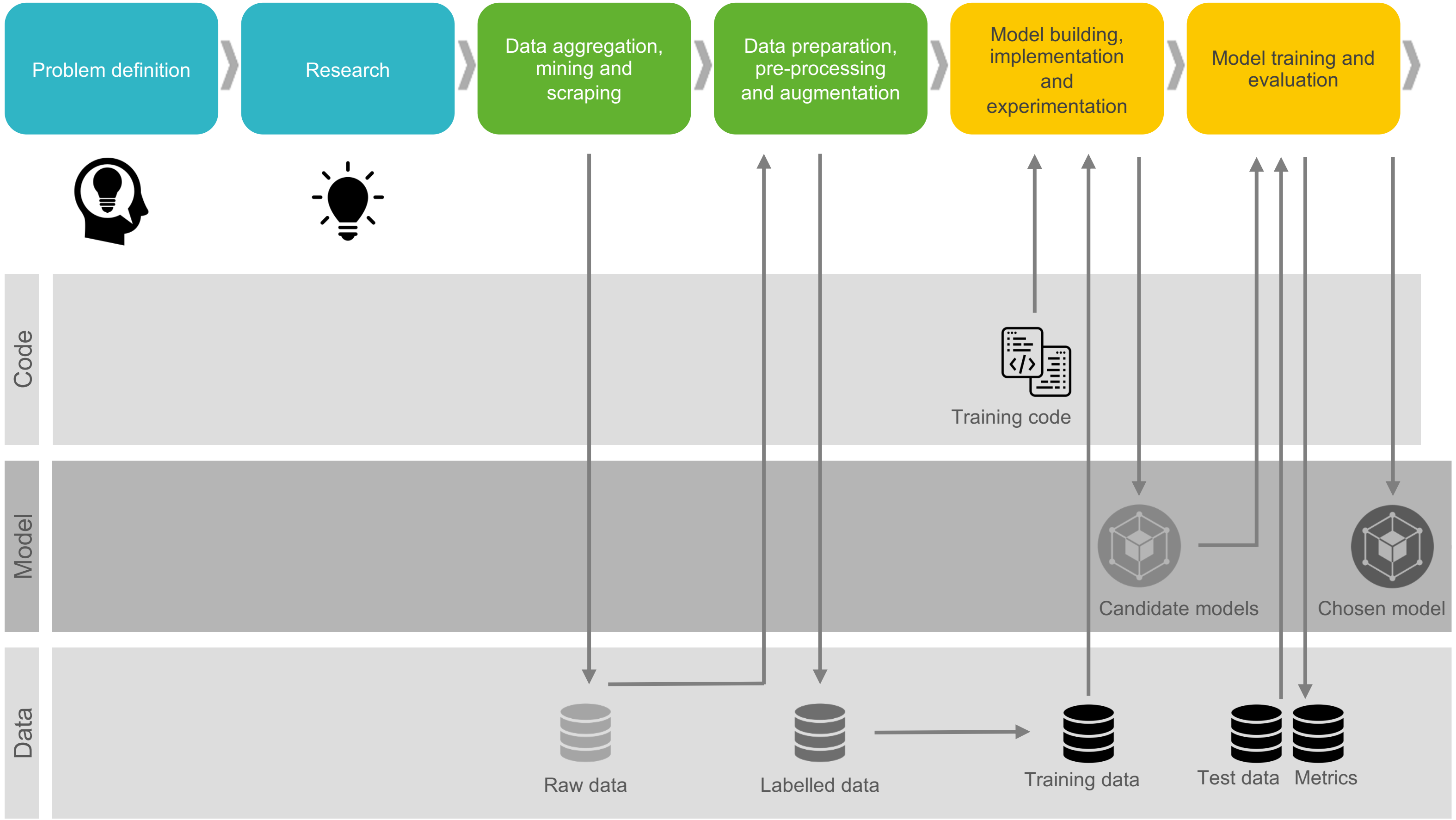
- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

- Data reformatting
- Data cleaning
- Data normalisation
- Data augmentation

- Usage of pre-trained models?
- Fine-tuning pre-trained models

- Training accuracy
- Validation accuracy
- Training loss
- Validation loss
- Underfitting or overfitting?



Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

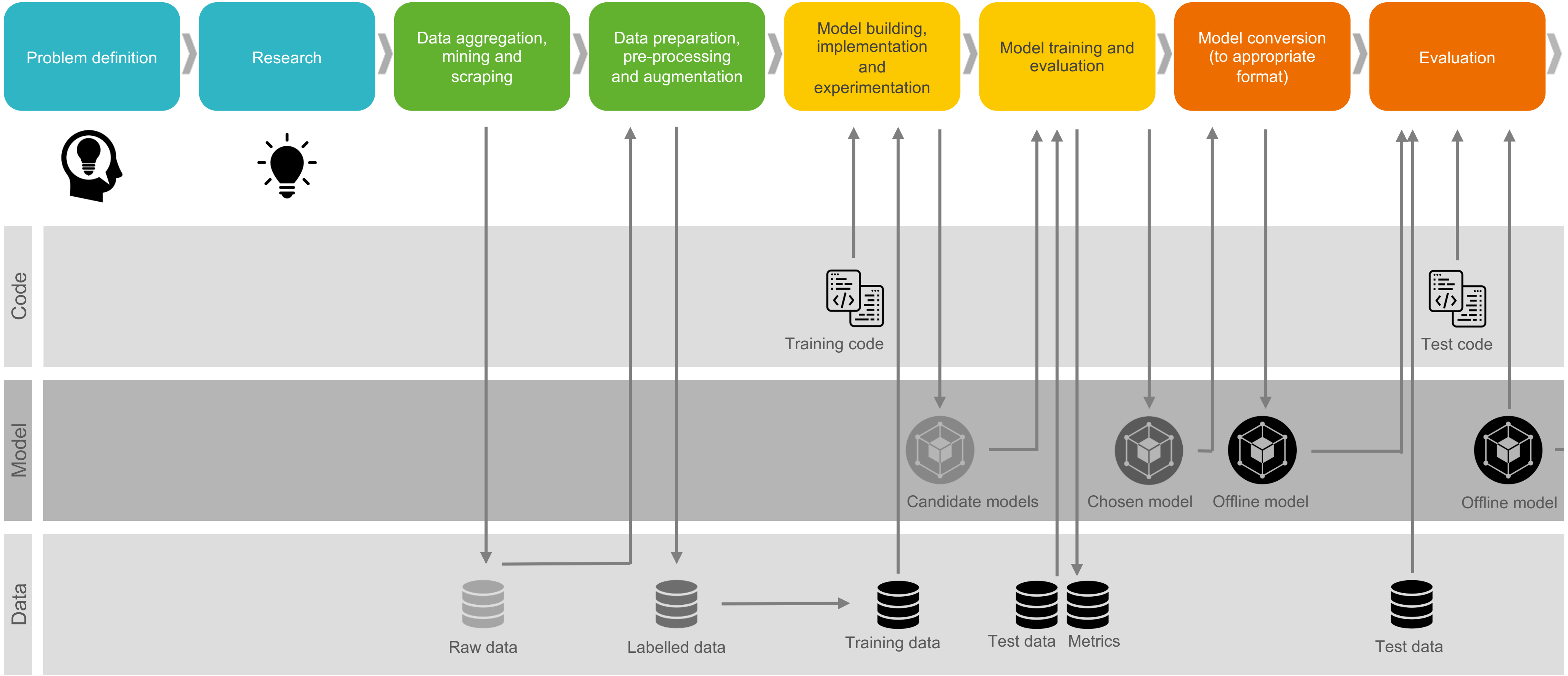
- Data reformatting
- Data cleaning
- Data normalisation
- Data augmentation

- Usage of pre-trained models?
- Fine-tuning pre-trained models

- Training accuracy
- Validation accuracy
- Training loss
- Validation loss
- Underfitting or overfitting?

- Refine and optimise the model
- Model conversion
- Mobile-optimised model

- Confusion matrix (error matrix)
- Precision-recall



Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

- Data reformatting
- Data cleaning
- Data normalisation
- Data augmentation

- Usage of pre-trained models?
- Fine-tuning pre-trained models

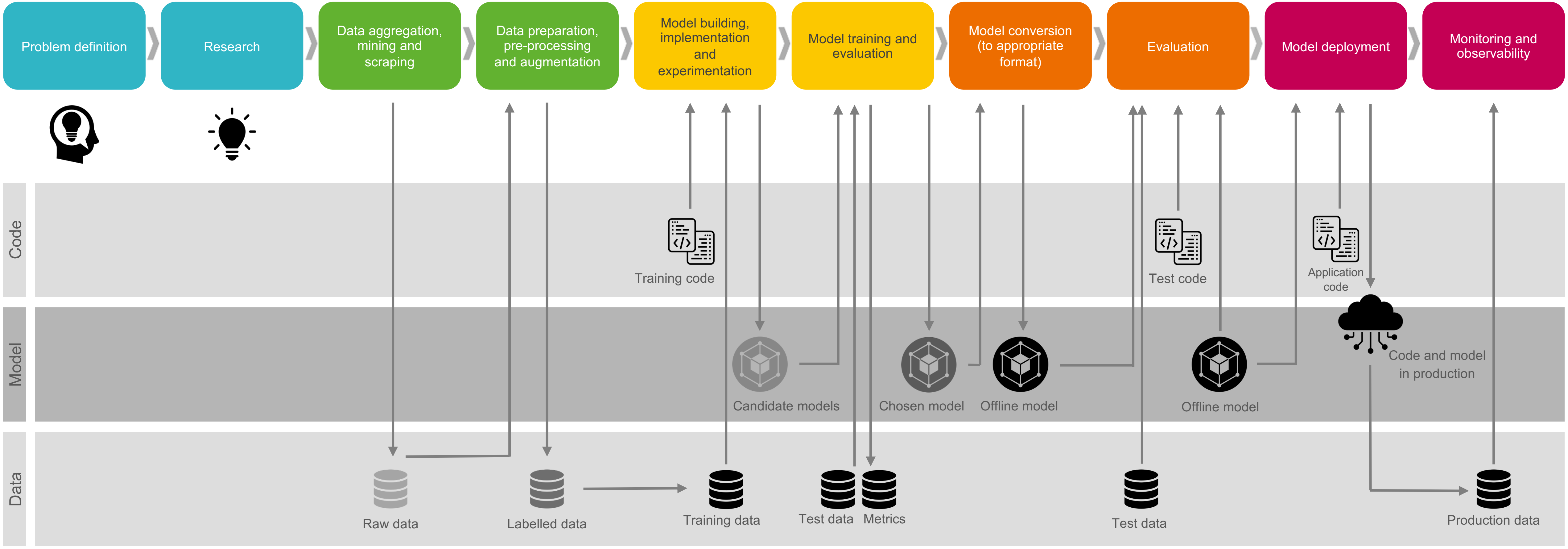
- Training accuracy
- Validation accuracy
- Training loss
- Validation loss
- Underfitting or overfitting?

- Refine and optimise the model
- Model conversion
- Mobile-optimised model

- Confusion matrix (error matrix)
- Precision-recall

- UI interface to access model functionalities
- Continuous integration pipeline that enables model redeployment

- Model performance monitoring system



Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)

- Problem statement
- Ideal problem solution
- Understanding and insight into the problem
- Technical requirements

- Data structure and source
- Solution form
- Model architecture
- Algorithm research
- Hardware requirements

- Data gathering (diverse, unbiased and abundant)

- Data reformatting
- Data cleaning
- Data normalisation
- Data augmentation

- Usage of pre-trained models?
- Fine-tuning pre-trained models

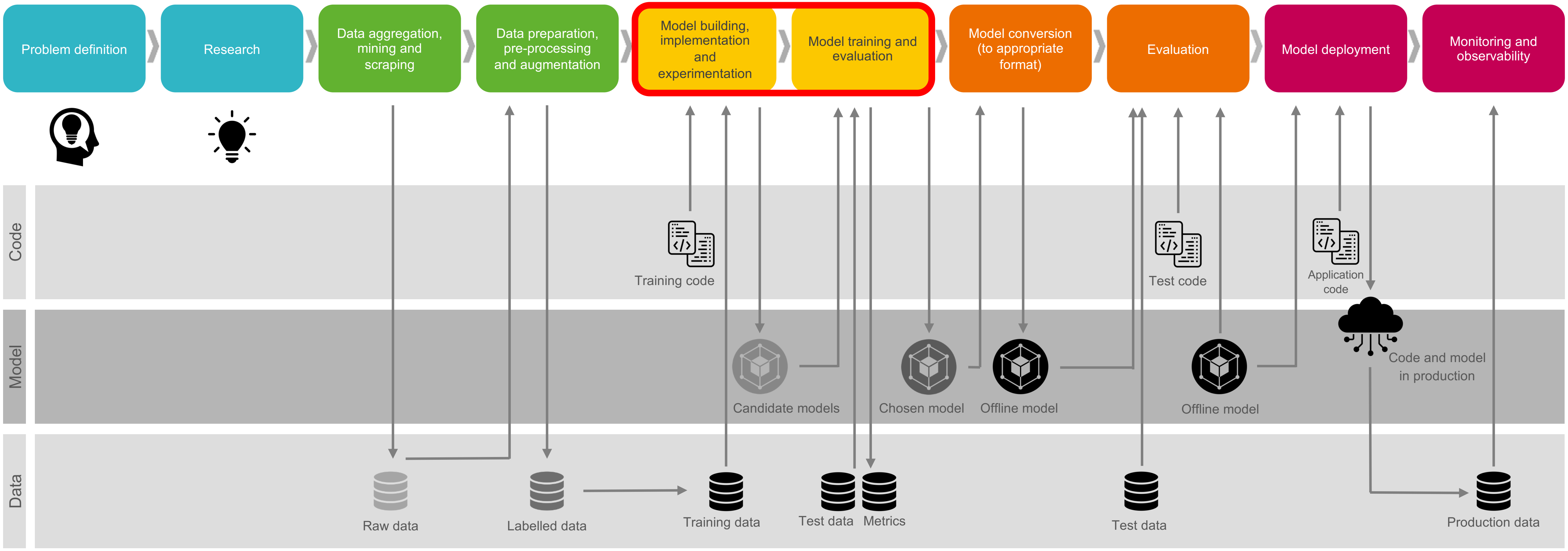
- Training accuracy
- Validation accuracy
- Training loss
- Validation loss
- Underfitting or overfitting?

- Refine and optimise the model
- Model conversion
- Mobile-optimised model

- Confusion matrix (error matrix)
- Precision-recall

- UI interface to access model functionalities
- Continuous integration pipeline that enables model redeployment

- Model performance monitoring system



GPT assistant training pipeline (Karpathy, 2023)

Pretraining
- 1000s of GPUs
- Months of training
- Example: GPT, LLaMA, PaLM

→ We can deploy this model.

Supervised Finetuning
- 1-100 GPUs
- Days of training
- Example: Vicuna-13B

→ We can deploy this model.

Reward Modelling
- 1-100 GPUs
- Days of training

Reinforcement Learning
- 1-100 GPUs
- Days of training
- Example: ChatGPT, Claude

→ We can deploy this model.

Model conversion (to appropriate format)
- Refine and optimise the model
- Model conversion
- Mobile-optimised model

Evaluation
- Confusion matrix (error matrix)
- Precision-recall

Code
Language modelling
- Predict the next token

Language modelling
- Predict the next token

Binary classification
- Predict rewards consistent with preferences

Reinforcement Learning
- Generate tokens that maximise the reward

Test code

Model
Base model

Initialised from BM

SFT model

Initialised from SFT

RM model

Initialised from SFT and use RM

RL model

Chosen model

Offline model

Offline model

Data
Raw Internet
- Text trillions of words
- Low-quality and large quantity

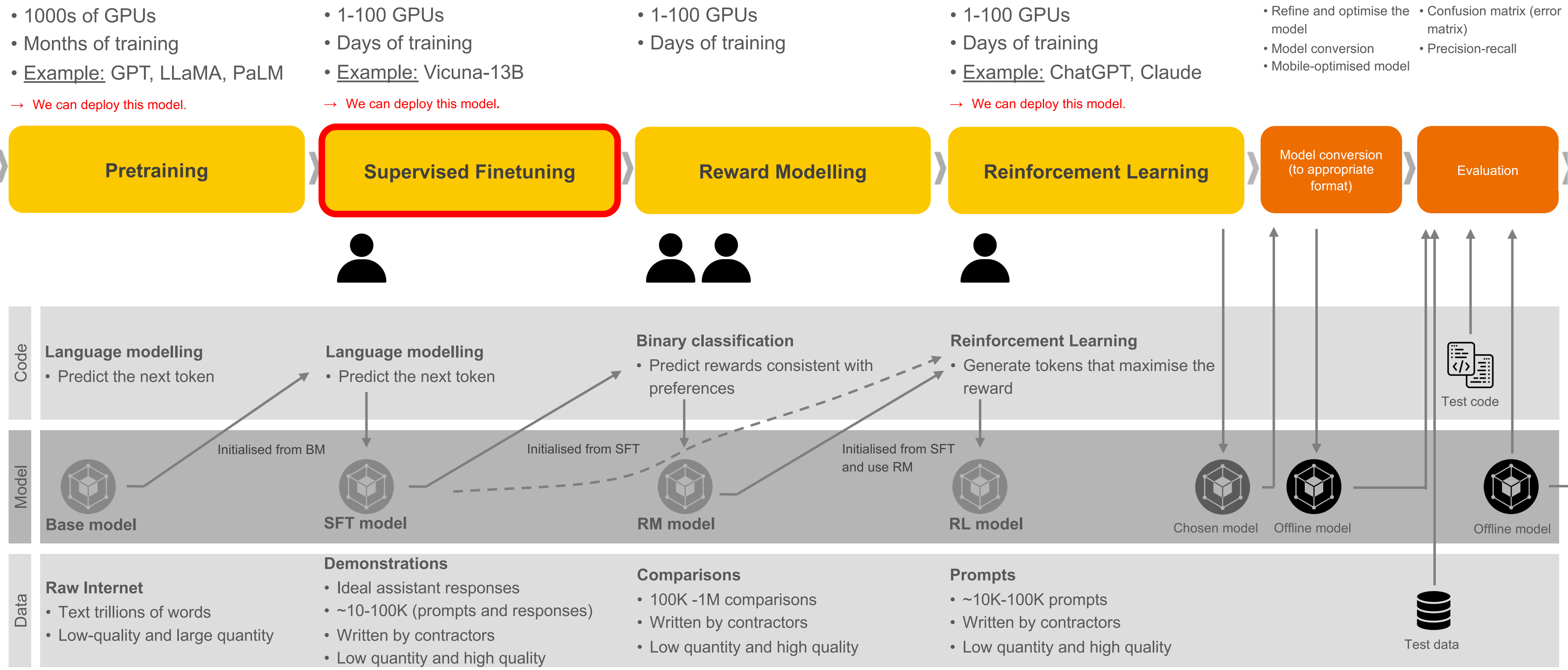
Demonstrations
- Ideal assistant responses
- ~10K-100K (prompts and responses)
- Written by contractors
- Low quantity and high quality

Comparisons
- 100K -1M comparisons
- Written by contractors
- Low quantity and high quality

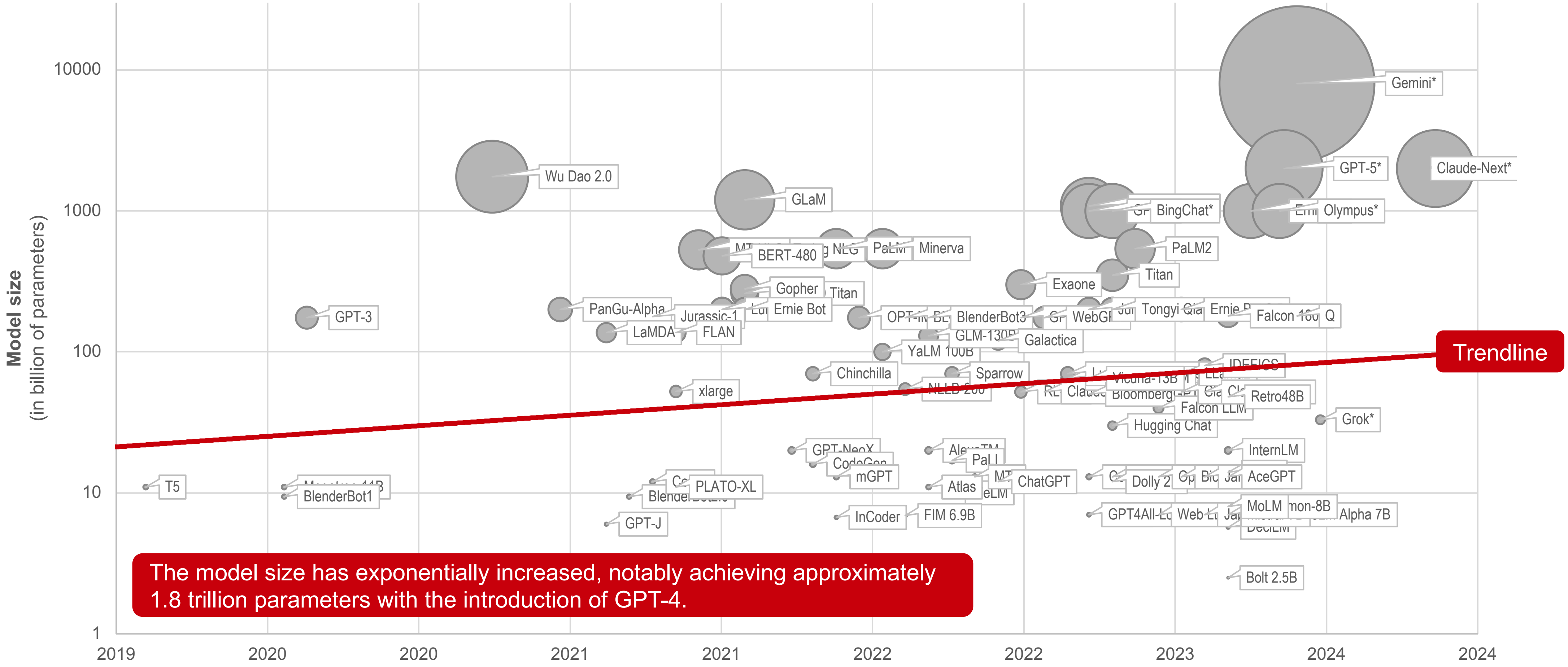
Prompts
- ~10K-100K prompts
- Written by contractors
- Low quantity and high quality

Test data

GPT assistant training pipeline (Karpathy, 2023)



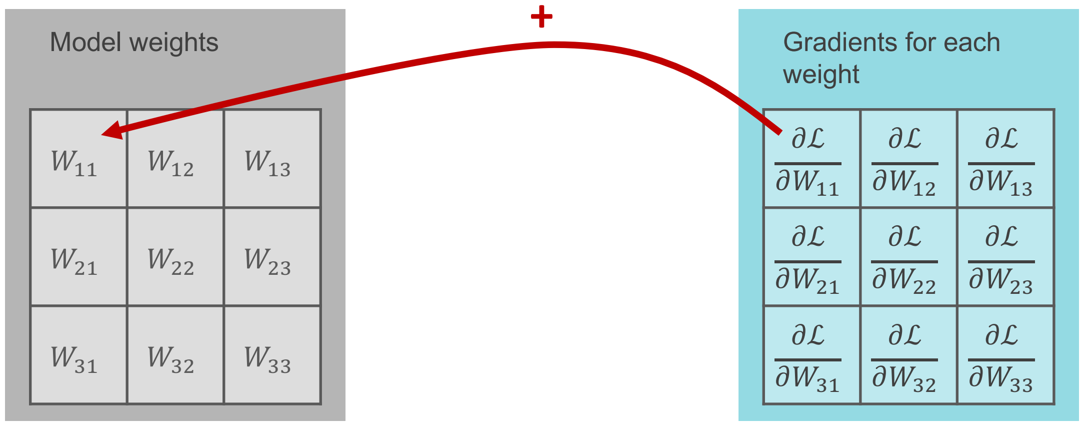
LLMs model sizes over time (Information is Beautiful, 2024)



The model size has exponentially increased, notably achieving approximately 1.8 trillion parameters with the introduction of GPT-4.

Full-parameter fine-tuning

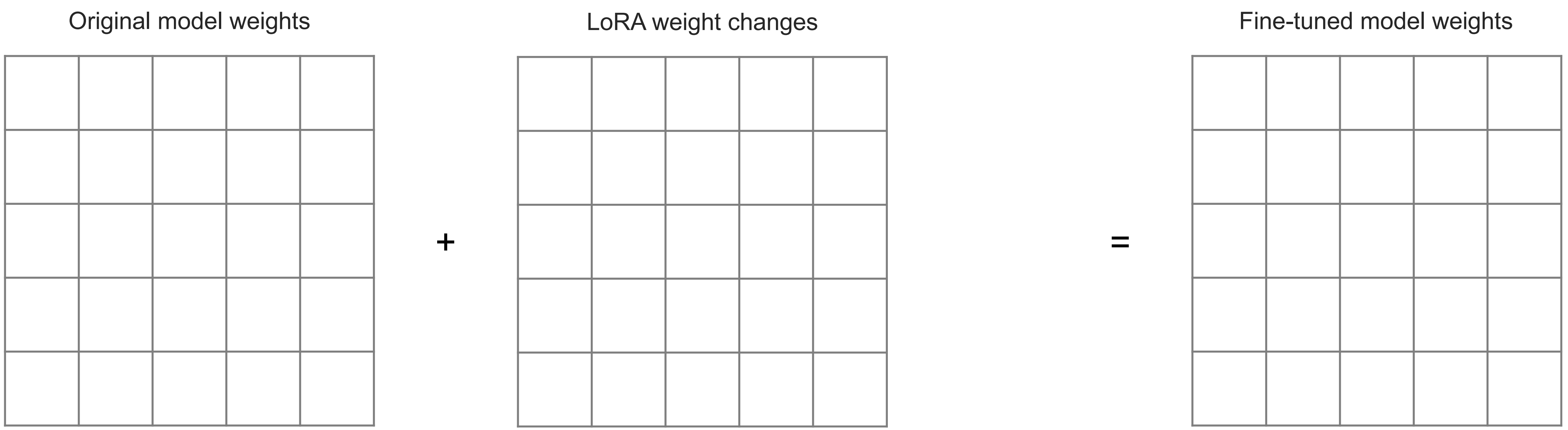
- Updates are applied to all model weights.
- Models feature large weight matrices, e.g., 7 billion weights for a 7B model and 13 billion for a 13B model.
- Weight updates occur over multiple epochs.
- Extensive memory is required to store and update weights.
- Fine-tuning is restricted to high-capacity GPUs or GPU clusters due to these memory demands.



Suppose hardware constraints limit our ability to test diverse strategies for enhancing the base model. In that case, Low-Rank Adaption (LoRA) offers two principal methods for solving this problem and can fine-tune LLMs at only a fraction of the cost.

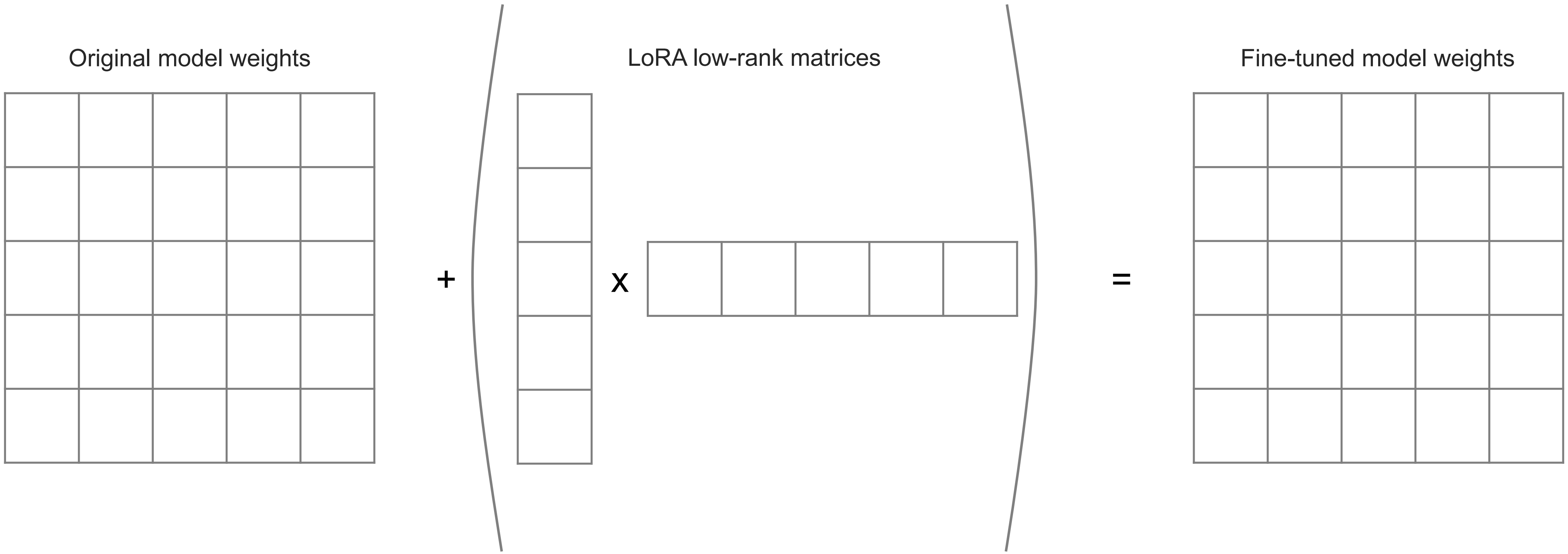
How is Low-Rank Adaption (LoRA) different? (Hu et al., 2021)

1. We **monitor weight changes** instead of directly updating them.
2. These weight changes are tracked in **two distinct and smaller matrices**, which are multiplied to create a product identical in size to the model's weight matrix.



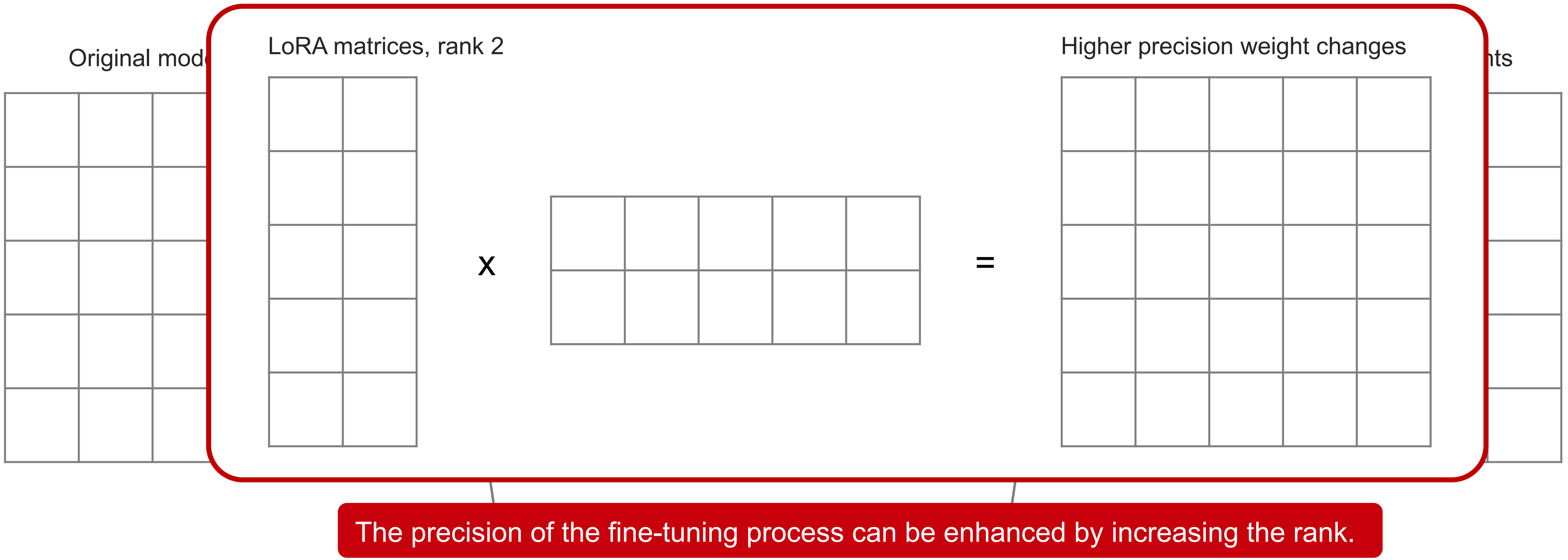
How is Low-Rank Adaption (LoRA) different? (Hu et al., 2021)

1. We **monitor weight changes** instead of directly updating them.
2. These weight changes are tracked in **two distinct and smaller matrices**, which are multiplied to create a product identical in size to the model's weight matrix.



How is Low-Rank Adaption (LoRA) different? (Hu et al., 2021)

1. We **monitor weight changes** instead of directly updating them.
2. These weight changes are tracked in **two distinct and smaller matrices**, which are multiplied to create a product identical in size to the model's weight matrix.



Number of trainable parameters

	Model size (in billion of parameters)			
Rank	**7B**	**13B**	**70B**	**180B**
1	167k	228k	529k	849k
2	334k	456k	1M	2M
8	1M	2M	4M	7M
16	3M	4M	8M	14M
512	86M	117M	270M	434M
1,024	171M	233M	542M	869M
8,192	1.4B	1.8B	4.3B	7.0B

In reality, LLMs consist of multiple layers of varying sizes, contrary to the simplification of being a single layer.

Percent of total parameters

	Model size (in billion of parameters)			
Rank	**7B**	**13B**	**70B**	**180B**
1	0.00%	0.00%	0.00%	0.00%
2	0.01%	0.00%	0.00%	0.00%
8	0.02%	0.01%	0.01%	0.00%
16	0.04%	0.03%	0.01%	0.01%
512	1.22%	0.90%	0.39%	0.24%
1,024	2.45%	1.80%	0.77%	0.48%
8,192	19.58%	14.37%	6.19%	3.86%

Involvement from higher-ranked individuals is particularly beneficial in teaching complex behaviours and addressing behaviours that contradict or fall outside the range of initial training.

Percentages may be understated due to the multi-layered structure of models, but the core concept remains clear.

QLoRA (Dettmers et al., 2023)

— Efficient fine-tuning of quantised LLMs

- This is basically LoRA 2.0 with “recoverable” quantisation for reduced memory usage.

- The scientific paper has two critical findings:
  - Training all network layers is crucial for matching the performance of full-parameter fine-tuning.
  - The rank values between 8 and 256 are observed to have minimal impact on performance.

arXiv:2305.14314v1 [cs.LG] 23 May 2023

QLoRA: Efficient Finetuning of Quantized LLMs

Tim Dettmers* **Artidoro Pagnoni*** **Ari Holtzman**

Luke Zettlemoyer

University of Washington
{dettmers,artidoro,ahai,lsz}@cs.washington.edu

Abstract

We present QLoRA, an efficient finetuning approach that reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance. QLoRA backpropagates gradients through a frozen, 4-bit quantized pretrained language model into Low Rank Adapters (LoRA). Our best model family, which we name **Guanaco**, outperforms all previous openly released models on the Vicuna benchmark, reaching 99.3% of the performance level of ChatGPT while only requiring 24 hours of finetuning on a single GPU. QLoRA introduces a number of innovations to save memory without sacrificing performance: (a) 4-bit NormalFloat (NF4), a new data type that is information theoretically optimal for normally distributed weights (b) Double Quantization to reduce the average memory footprint by quantizing the quantization constants, and (c) Paged Optimizers to manage memory spikes. We use QLoRA to finetune more than 1,000 models, providing a detailed analysis of instruction following and chatbot performance across 8 instruction datasets, multiple model types (LLaMA, T5), and model scales that would be infeasible to run with regular finetuning (e.g. 33B and 65B parameter models). Our results show that QLoRA finetuning on a small high-quality dataset leads to state-of-the-art results, even when using smaller models than the previous SoTA. We provide a detailed analysis of chatbot performance based on both human and GPT-4 evaluations showing that GPT-4 evaluations are a cheap and reasonable alternative to human evaluation. Furthermore, we find that current chatbot benchmarks are not trustworthy to accurately evaluate the performance levels of chatbots. A lemon-picked analysis demonstrates where **Guanaco** fails compared to ChatGPT. We release all of our models and code, including CUDA kernels for 4-bit training.[2]

1 Introduction

Finetuning large language models (LLMs) is a highly effective way to improve their performance, [40, 62, 43, 61, 59, 37] and to add desirable or remove undesirable behaviors [43, 2, 4]. However, finetuning very large models is prohibitively expensive; regular 16-bit finetuning of a LLaMA 65B parameter model [57] requires more than 780 GB of GPU memory. While recent quantization methods can reduce the memory footprint of LLMs [14, 13, 18, 66], such techniques only work for inference and break down during training [65].

We demonstrate for the first time that it is possible to finetune a quantized 4-bit model without any performance degradation. Our method, QLoRA, uses a novel high-precision technique to quantize a pretrained model to 4-bit, then adds a small set of learnable Low-rank Adapter weights [28]

*Equal contribution.
[2] https://github.com/artidoro/qlora and https://github.com/TimDettmers/bitsandbytes

Preprint. Under review.

Overcoming challenges of LLMs

- **Hallucination problem:**
  - Models frequently generate false information with high confidence, presenting a significant risk in scenarios that require strict accuracy. This represents the most prominent challenge in Generative AI.
- **Attribution problem:**
  - More clarity is needed regarding why models produce specific outputs, which makes it difficult to trust or validate their responses.
- **Staleness:**
  - Language models quickly become outdated, needing more information on recent events, diminishing their relevance and utility over time.
- **Revisions challenge:**
  - Models must comply with regulations (e.g., GDPR), which entails the ability to delete or revise data, a functionality that remains underdeveloped. The AI Act commits to monitoring across various dimensions of risk: fairness, autonomy, transparency, security, reliability, and data protection.
- **Customisation issue:**
  - Adapting models to specific use cases or datasets is an on-going challenge, necessitating innovative solutions for effective personalisation and application in diverse environments. This includes strategies for integrating models with unique corporate data or adjusting outputs to align with the specific needs of different user groups.

A prevalent strategy currently being adopted involves integrating existing LLMs with external memory resources. Retrieval-Augmented Generation (RAG) systems, which dynamically retrieve and incorporate external data into the decision-making process, are solving these challenges.

General Purpose AI (GPAI) classification and key requirements for providers (European Parliament, 2024), (Pinto, 2024)

ALL GPAI MODELS

Large models and systems capable of competently performing a wide range of distinctive tasks, such as generating video, text, images, computer code, or conversing.

- Transparency obligations before market placement, including:
  - Drawing up technical documentation for downstream providers
  - Complying with EU copyright law and disseminating detailed summaries about the content used in training
  - Watermarking AI generated or manipulated content

SYSTEMIC RISK GPAI MODELS

Foundation models trained with a large amount of data and with advanced complexity, capabilities, and performance well above the average can disseminate systemic risks along the value chain.

- Complying with all requirements applicable to all GPAI models and systems
- Conducting model evaluations
- Assessing and mitigating systemic risks
- Conducting adversarial testing
- Reporting of serious incidents to the EU Commissions
- Ensuring sufficient cybersecurity protection
- Reporting on energy efficiency

Fine Tuning vs. Retrieval-Augmented Generation (RAG) (Soudani et al., 2024)

- Researchers have studied the comparison between Retrieval Augmented Generation (RAG) and fine-tuning methods on synthetic data.
  - Their investigations reveal that both strategies significantly enhance the capability of AI to handle specialised information during question-answering tasks.
  - **RAG emerges as the leading methodology, outperforming fine-tuning in improving model responses to obscure queries.**
  - This does not eliminate fine-tuning's relevance but suggests **RAG as a more efficient option for bolstering AI against niche topics**.
  - Fine-tuning is acknowledged for its depth in embedding knowledge, but it shares similar limitations with pre-training, particularly in learning about infrequent concepts.

arXiv:2403.01432v2 [cs.CL] 7 Mar 2024

Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge

Heydar Soudani
Radboud University
Nijmegen
The Netherlands
heydar.soudani@ru.nl

Evangelos Kanoulas
University of Amsterdam
Amsterdam
The Netherlands
e.kanoulas@uva.nl

Faegheh Hasibi
Radboud University
Nijmegen
The Netherlands
faegheh.hasibi@ru.nl

Abstract

Large language models (LLMs) memorize a vast amount of factual knowledge, exhibiting strong performance across diverse tasks and domains. However, it has been observed that the performance diminishes when dealing with less-popular or low-frequency concepts and entities, for example in domain specific applications. The two prominent approaches to enhance the performance of LLMs on low-frequent topics are: Retrieval Augmented Generation (RAG) and fine-tuning (FT) over synthetic data. This paper explores and evaluates the impact of RAG and FT on customizing LLMs in handling low-frequency entities on question answering task. Our findings indicate that FT significantly boosts the performance across entities of varying popularity, especially in the most and least popular groups, while RAG surpasses other methods. Additionally, the success of both RAG and FT approaches is amplified by advancements in retrieval and data augmentation techniques. The code and data is available at https://github.com/informagi/RAGvsFT.

Figure 1: Correlation between subject entity popularity in a question and the effects of RAG and FT on FlanT5-small performance in open-domain question answering. FT markedly improves accuracy in the initial and final buckets relative to others (indicated by the pink line).

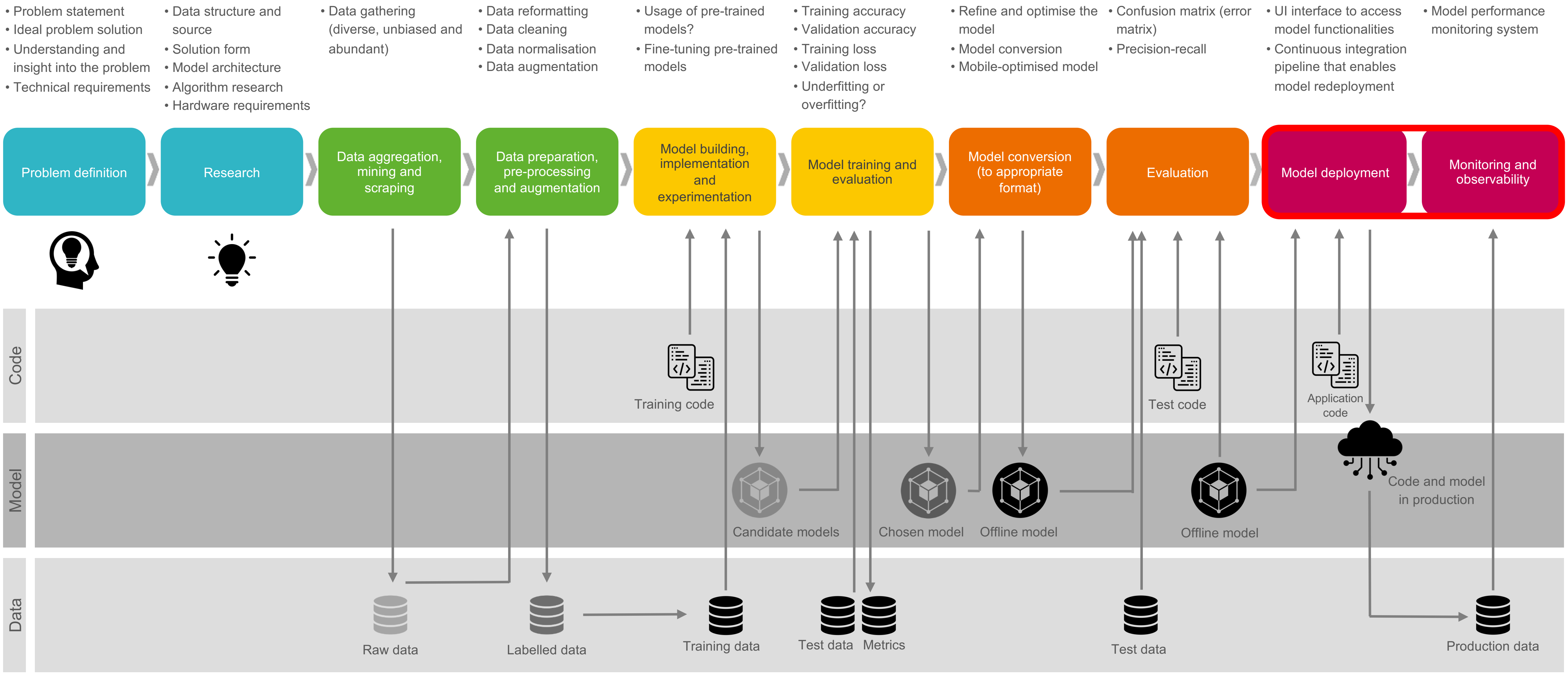
1 Introduction

Large Language Models (LLMs) exhibit outstanding capabilities in executing tasks that demand extensive memorization of factual data (Chowdhery et al., 2023). However, their memorization capabilities are constrained when dealing with less frequent entities (Mallen et al., 2023; Kandpal et al., 2023; Sun et al., 2023), and even the largest models may encounter the well-known "hallucination" problem (Shuster et al., 2021) and temporal degradation (Kasai et al., 2022). Consequently, when LLMs are intended for deployment in less resourced domains, customization becomes imperative to ensure optimal performance. A common example is within the industrial setup, where chatbots or Question Answering (QA) systems need to accurately answer users' questions about a proprietary knowledge graph or intra-company terminology with limited textual description.

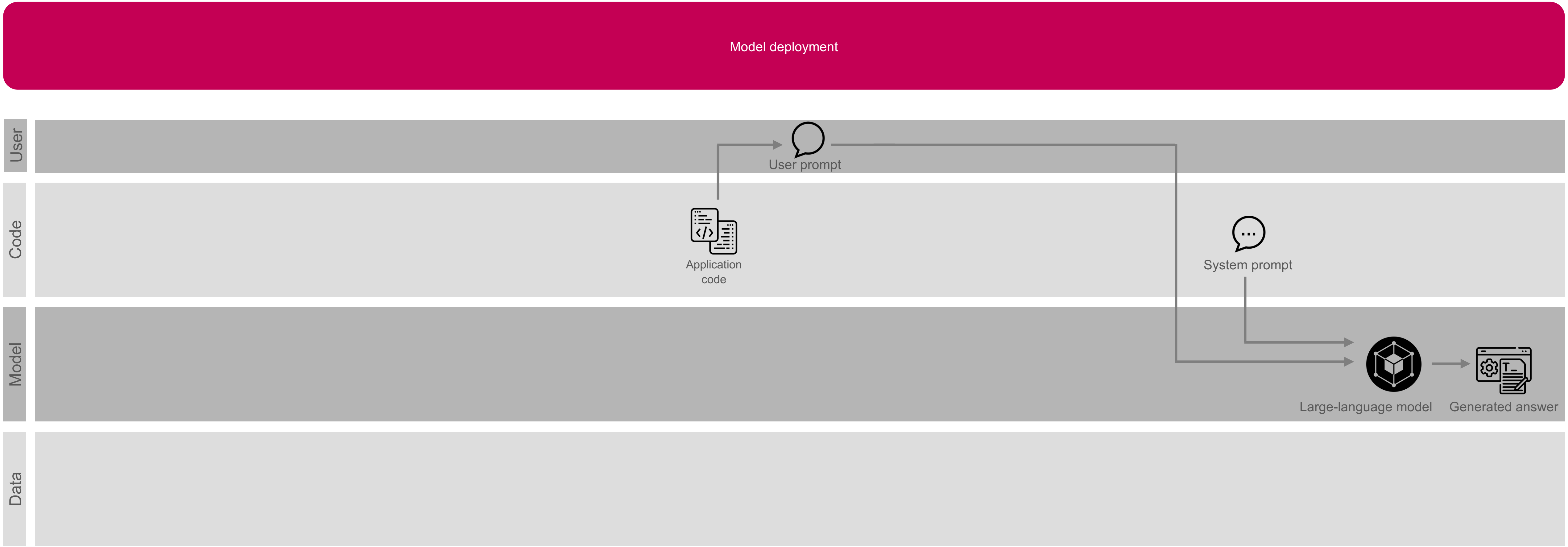
Retrieval-Augmented Generation (RAG) and Fine-Tuning (FT) stand out as two prominent approaches for adapting LLMs to specific domains. RAG retrieves relevant information from a document corpus and enhances LLM's response generation through the implementation of in-context learning (ICL). Conversely, the fine-tuning approach updates model weights to become adept at recalling specific information and enhance its memorization capabilities during inference. In the context of less popular knowledge, where limited data is available, data augmentation methods are utilized to generate synthetic training data, serving as an initial step towards fine tuning.

In this paper, we aim to understand which approach and under what conditions is more effective for industry-specific models. Specifically, we seek to answer the following research questions:
(RQ1) What is the effectiveness of RAG and fine-tuning with synthetic data on QA for low-frequency factual knowledge?
(RQ2) Which parameters, including the quality of

Implementation of end-to-end lifecycle in AI projects (Alake, 2020), (Sato et al., 2019)



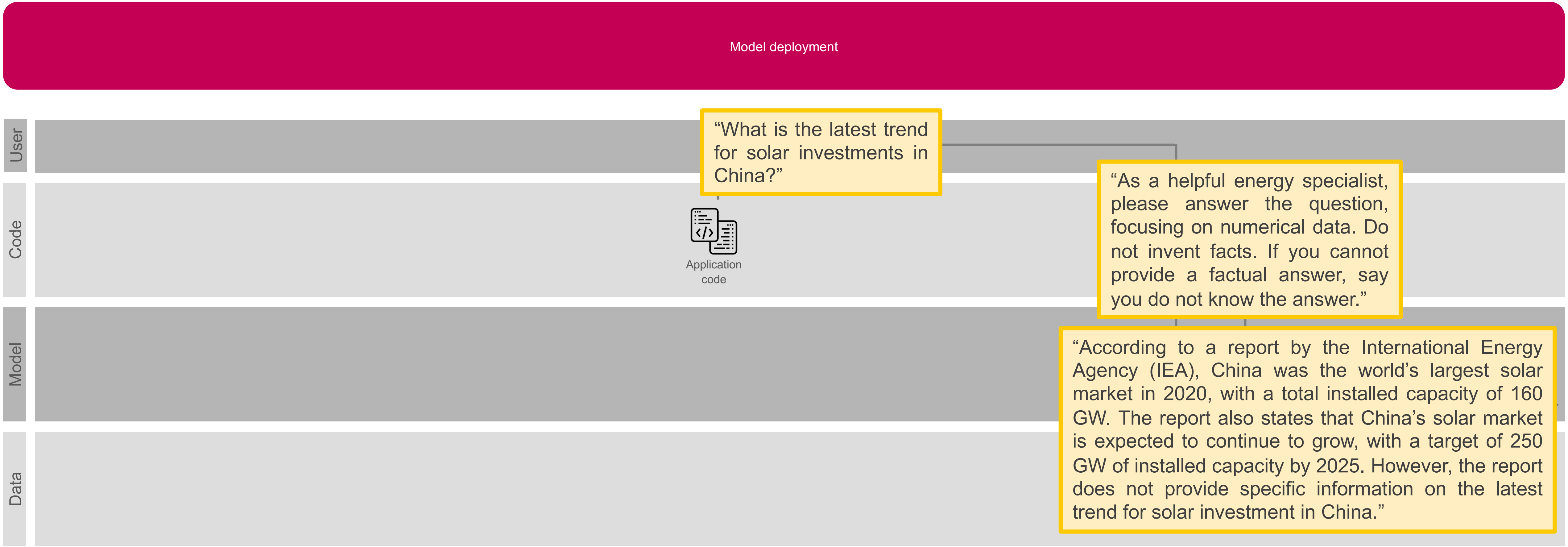
Basic chatbot architecture



Document corpus

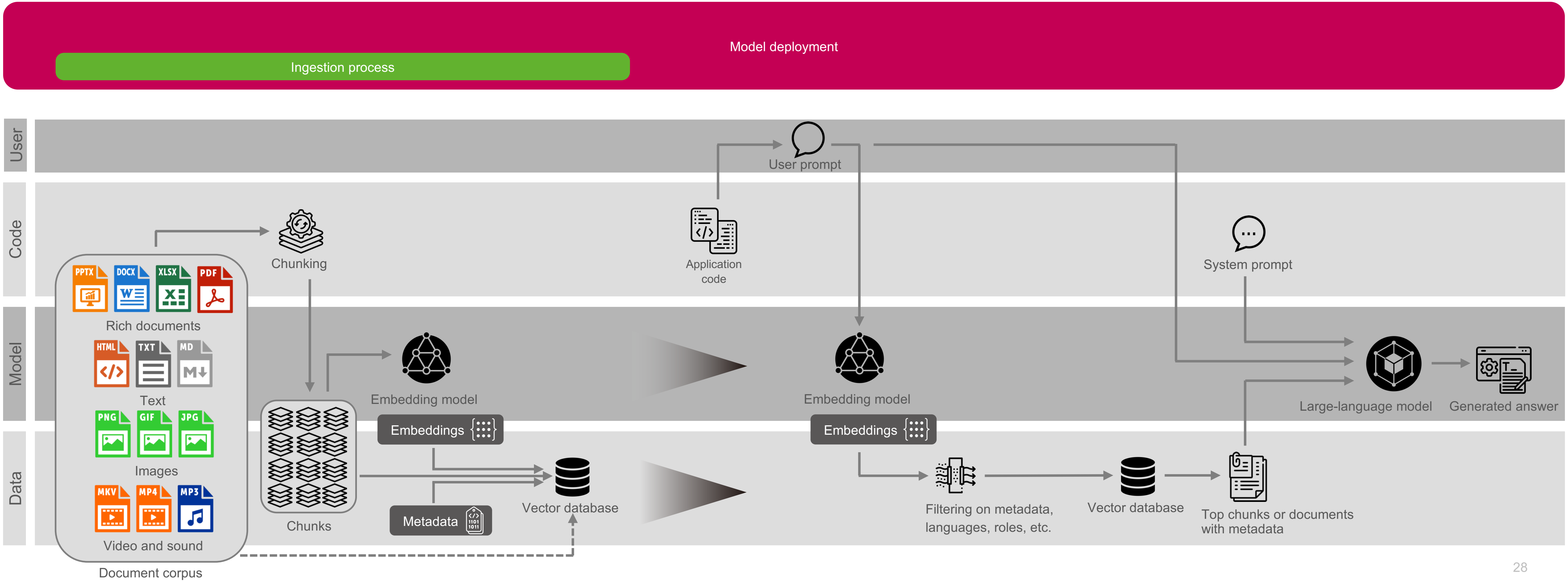
Basic chatbot architecture (Simon, 2023)

— Example



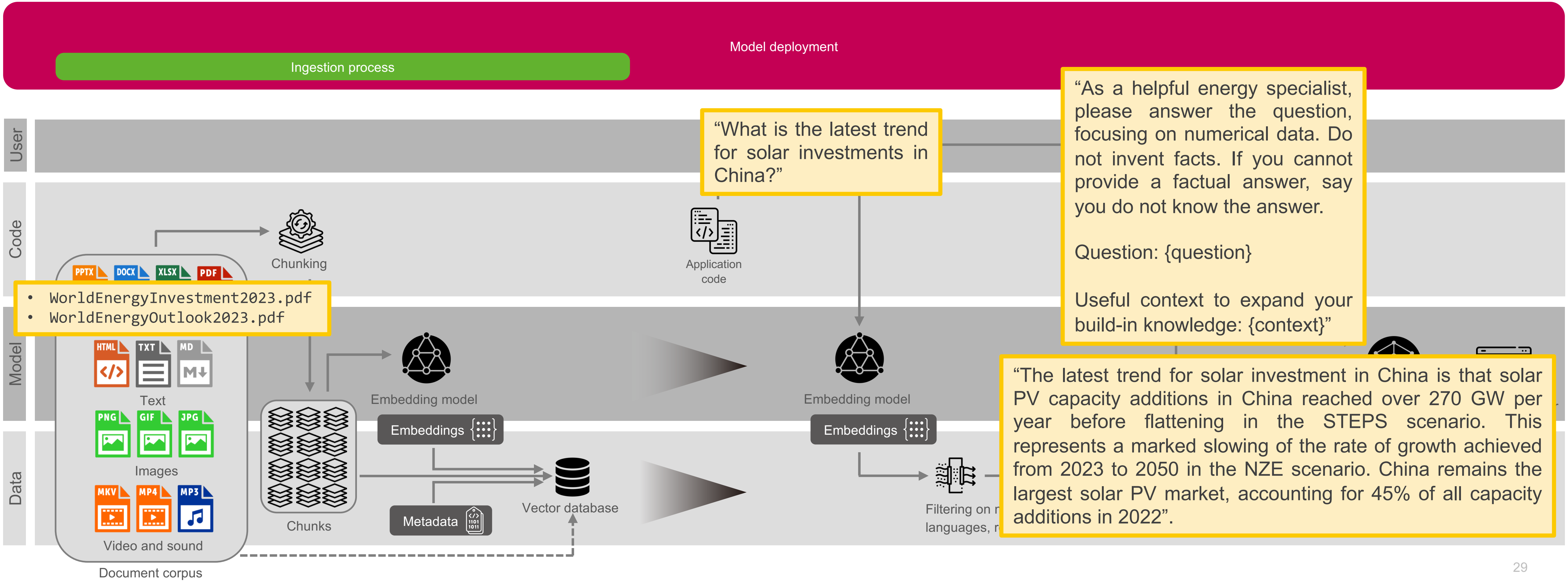
Document corpus

Retrieval-Augmented Generation (RAG) architecture



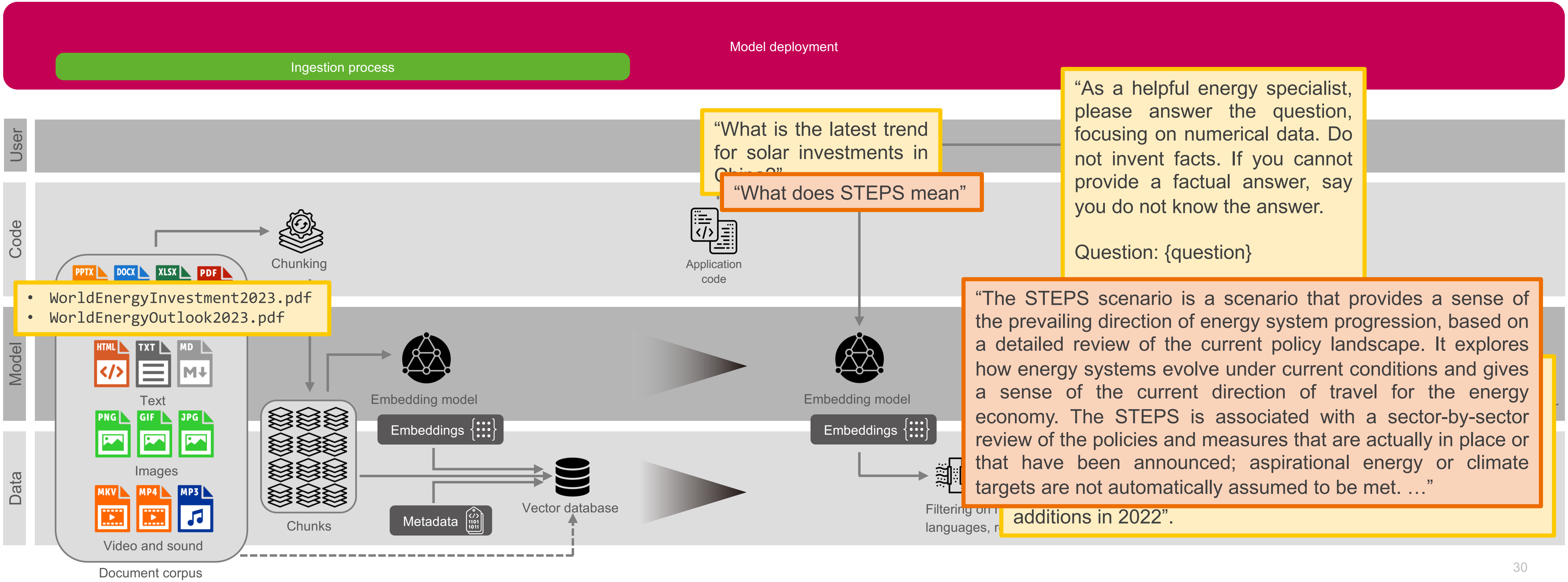
Retrieval-Augmented Generation (RAG) architecture (Simon, 2023)

— Example



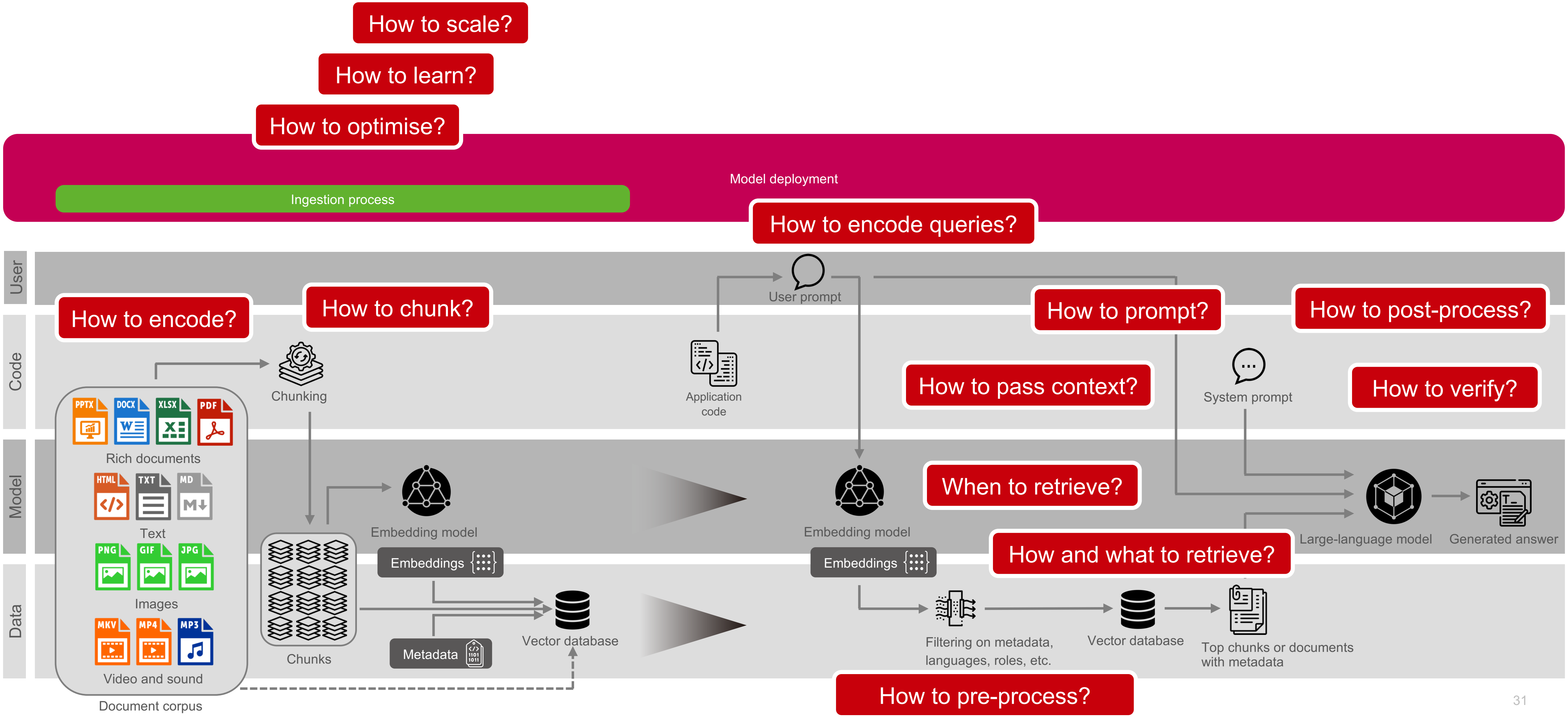
Retrieval-Augmented Generation (RAG) architecture (Simon, 2023)

— Example



Retrieval-Augmented Generation (RAG) architecture

— Many questions ?!?



Retrieval-Augmented Generation (RAG) architecture

Chunking strategy:
- Chunk size optimisation
- Character, recursive character or document specific
- Sliding window chunking

Embedding strategy:
- Which embedding algorithm or model?

(Muennighoff et al., 2023), (Hugging Face, 2024)

Document retriever:
- Metadata attachment
- Mixed retrieval
- Cognitive reviewer

Failure points:
- Missing content
- Missed the top-ranked documents
- Not in context – consolidation strategy limitations
- Not extracted
- Wrong format

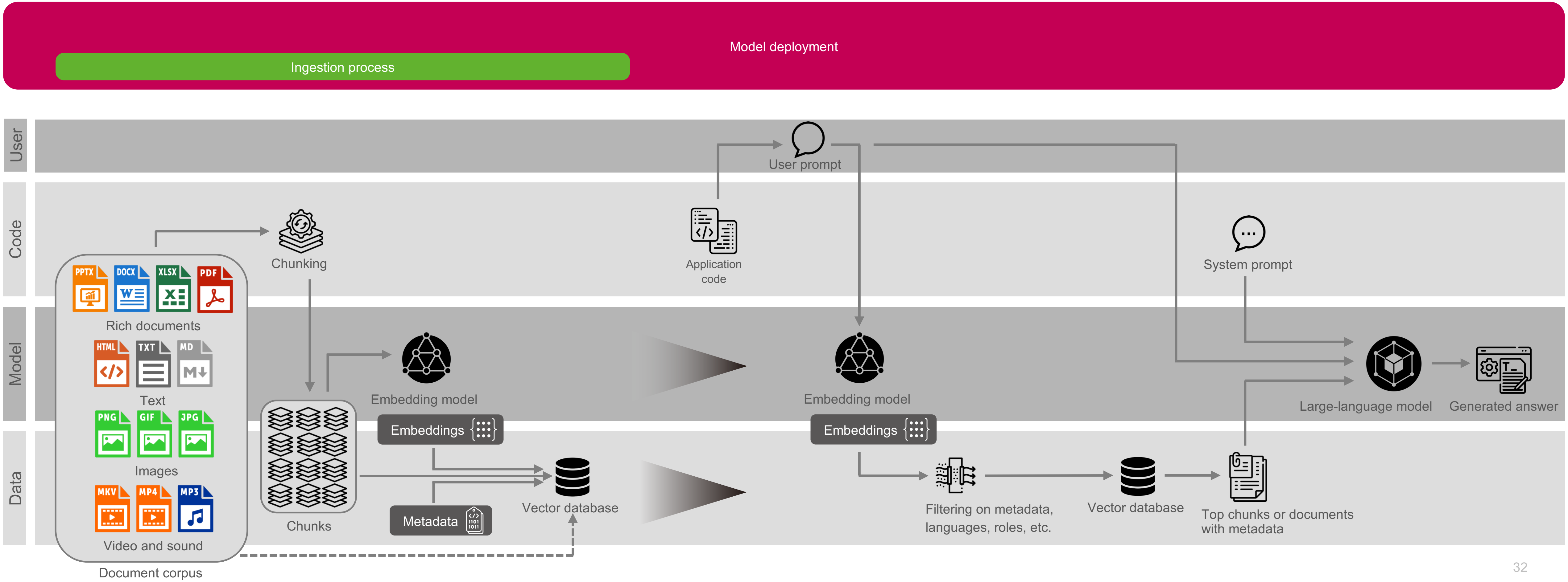
User authentication:
- Access control
- Data security
- User privacy
- Legal compliance
- Accountability

Guardrails:
- Anonymization
- Restrict substrings, topics, code, language
- Detect prompt injection
- Detect toxicity

Query strategy:
- Rewrite based on history
- Create subqueries or similar queries
- Query cost

Choice of LLM:
- Architecture
- Number of parameters
- Access
- Use-case
- Data privacy

Evaluate responses:
- Prompt evaluation
- RAG retrieval evaluation
- Relevance metrics
- Tasks-specific metrics
- Alignment metrics



Closing remarks

- A single 7-day forecast consumes 14 Wh with Huawei Pangu-Weather compared to 30,000 Wh with the ICON model, illustrating a significant difference in energy efficiency.
- LoRA fine-tunes LLMs by monitoring and updating weight changes through smaller matrices, enhancing fine-tuning precision without direct weight modification.
- LLMs can generate false information with high confidence, presenting a significant risk in scenarios that require strict accuracy.
- Integrating existing LLMs with external memories through Retrieval-augmented Generation (RAG) systems is a leading solution to current challenges.
- RAG outshines fine-tuning in AI's handling of niche topics by significantly enhancing response precision to obscure queries.
- Despite RAG's superiority in handling niche topics, ongoing research and numerous open questions highlight the evolving nature of this AI methodology.

HUAWEI

Advancing the Intelligent World

References

Alake, R. (2020). *10 Stages Of A Machine Learning Project In 2020 (And Where You Fit)*. *2020*, 1–17.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs* (arXiv:2305.14314). arXiv. https://doi.org/10.48550/arXiv.2305.14314

European Parliament. (2024). *Provisional Agreement Resulting from Interinstitutional Negotiations*. https://doi.org/10.5040/9781782258674

Information is Beautiful. (2024). *The Rise of Generative AI Large Language Models (LLMs) like ChatGPT*. Information Is Beautiful. https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

Karpathy, A. (2023, May 23). *State of GPT*. Microsoft BUILD. https://karpathy.ai/stateofgpt.pdf

Pinto, T. (2024, January 8). General-purpose AI under the AI Act. *Artificial Intelligence Act*. https://artificialintelligenceact.com/general-purpose-ai-under-the-ai-act/

Sato, D., Wider, A., & Windheuser, C. (2019). *Continuous Delivery for Machine Learning*. https://martinfowler.com/articles/cd4ml.html

Simon, J. (Director). (2023, October 24). *Retrieval-Augmented Generation chatbot, part 1: LangChain, Hugging Face, FAISS, AWS*. https://www.youtube.com/watch?v=7kDaMz3Xnkw