Scaling Intelligence: From Pretrained Models to Agentic Systems

Yaowei Zheng

Beihang University

January 31, 2026

Table of Contents

Table of Contents

Transformer Architecture

- **"Attention is All You Need"** (Vaswani et al., 2017)
- Self-attention mechanism: captures long-range dependencies
- Key components:
  - Multi-head attention
  - Positional encoding
  - Feed-forward networks
  - Layer normalization
- Technical evolution of transformer architectures:
  - From dense to sparse attention: MHA $\rightarrow$ GQA $\rightarrow$ MLA $\rightarrow$ DSA
  - From dense to sparse MLP: Dense FFN $\rightarrow$ MoE $\rightarrow$ fine-grained MoE

Probabilistic Modeling

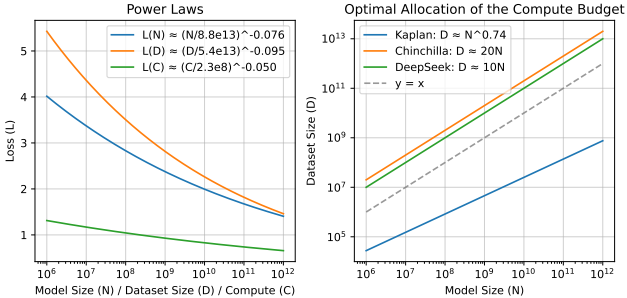
- Word-level probability prediction: $P(w_1, w_2, ..., w_n)$
- Current training paradigms:
  - Autoregressive language modeling
  - Diffusion models
- Training methods: Pre-training, SFT, PPO, DPO, GRPO
- Trends:
  - From expert knowledge to autonomous exploration: RLHF $\rightarrow$ RLVR
  - From off-policy to on-policy learning: DPO $\rightarrow$ GRPO

Scaling Laws for LLMs

- Power law relationship:

$$L(N, D, C) \approx AN^{-\alpha} + BD^{-\beta} + E$$

- N: Model size (parameters)
- D: Dataset size (tokens)
- C: Compute budget (FLOPs)
- Optimal compute allocation strategies



Emergent Abilities of LLMs

- Capabilities that appear at scale, absent in smaller models
- Key emergent behaviors:
  - Few-shot learning (Brown et al., 2020)
  - Zero-shot learning (Kojima et al., 2022)
  - Chain-of-thought reasoning (Wei et al., 2022)
  - Reasoning and acting (Yao et al., 2022)
  - Interleaved thinking (Unknown, 2025)

Evolution and Future Directions

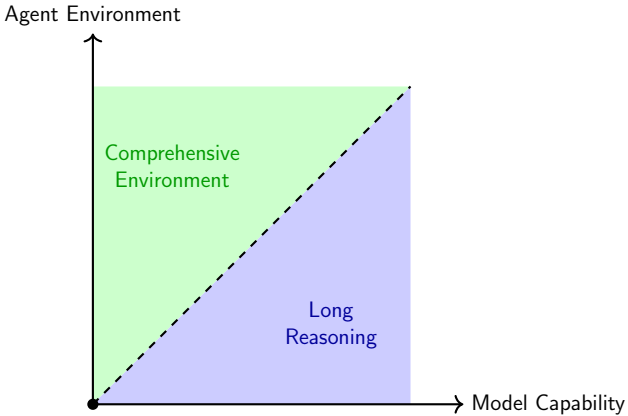


Table of Contents

Pre-training on Large-Scale Corpora

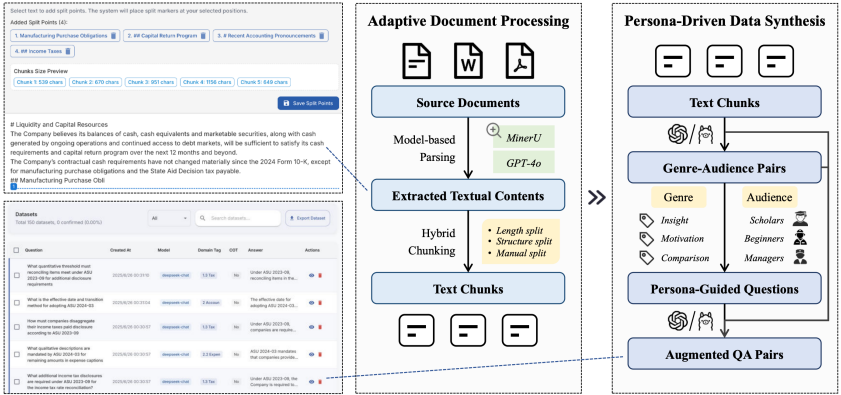
- Autoregressive pre-training objective:

$$\mathcal{L}_{\text{AR}} = -\sum_{t=1}^{T} \log P(x_t | x_{<t}; \theta)$$

- **High-quality data is the core of pre-training**
- Weak-to-strong generalization in data:
  - Natural web corpora $\rightarrow$ Large-scale synthetic data

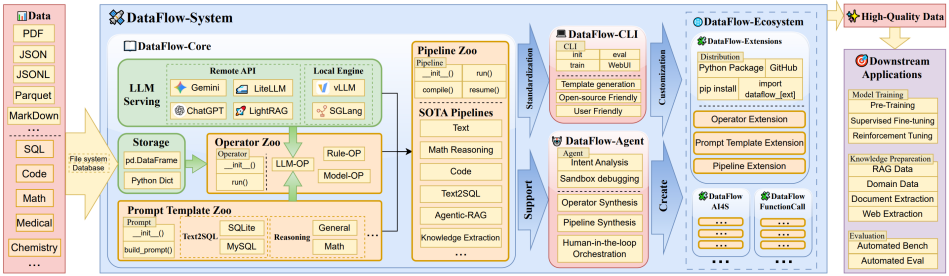
Our work: Easy Dataset (EMNLP'25, 13K GitHub Stars)

A software that uses LLMs to extract synthetic data from unstructured documents.



Our work: DataFlow (arXiv'25, 2.8K GitHub Stars)

A collection of reusable operators and pipelines for automated data processing.

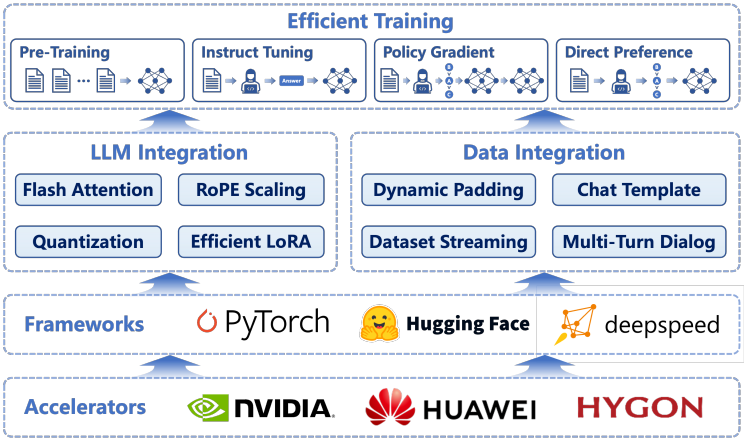


Post-training for Human Preference Alignment

- Traditional three-stage pipeline:
  - SFT $\rightarrow$ RM $\rightarrow$ RLHF
- Modern approaches:
  - Cold-start with knowledge distillation
  - RLVR (Reinforcement Learning with Verifiable Rewards)

Our work: LlamaFactory (ACL'24, 66K GitHub Stars)

Unified efficient training framework supporting over 500 LLMs.

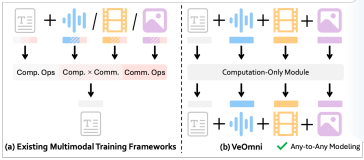


Multimodal Alignment

- **Multimodal understanding**
  - Image-to-text (I2T)
  - Video-to-text (V2T)
  - Audio-to-text (A2T)
- **Multimodal generation**
  - Text-to-image (T2I)
  - Text-to-video (T2V)
  - Text-to-speech (TTS)
- **Unified multimodal understanding and generation**
  - Any-to-Any (X2X)

Our work: VeOmni (AAAI'26, 1.6K GitHub Stars)

A torch-native distributed training framework with decoupled 3D parallelism for omni-modal models.



- **single modality**
- **complicated**
- **only NVIDIA**

- **omni modality**
- **light-weight & decoupled**
- **support Ascend NPU**

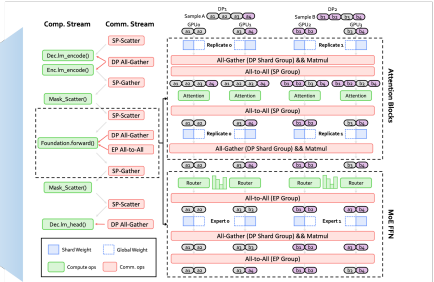
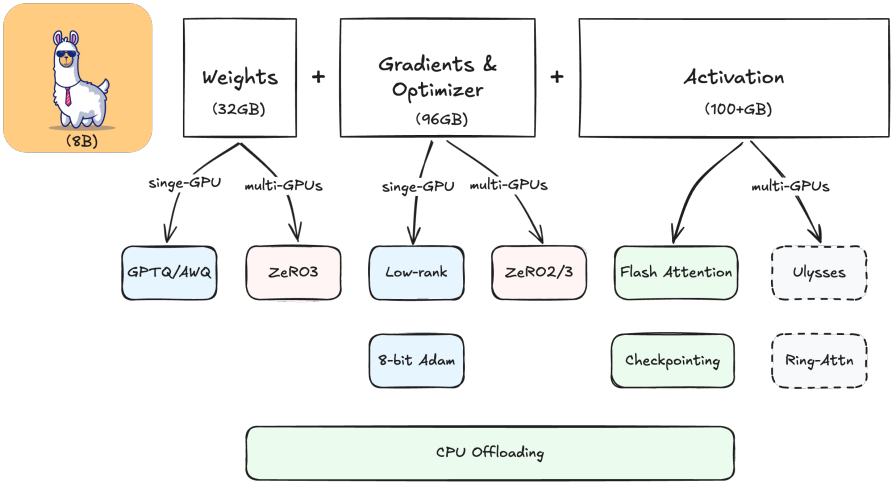


Figure 3 Overview of VeOmni's n-D parallelism. The figure on the left illustrates the data flow of VeOmni during the training of omni-modal LLMs. The encoder for each modality processes its respective inputs and then scatters the features to the corresponding ranks through an all-to-all communication operation. The right figure shows the 3D parallelism of VeOmni, where attention blocks apply HSDP with a data sharded size of 2 and a data replicate size of 2, inputs apply sequence parallelism with a size of 2, and mixture-of-experts blocks apply expert parallelism and FSDP with an EP size of 2 and a data sharded size of 2.

Training Infrastructure

- **Distributed training:**:
  - Megatron: DP + TP + PP
  - Torch-native: FSDP + SP + EP
- **Optimization techniques**:
  - Mixed precision: FP16, BF16, FP8, FP4
  - Gradient accumulation and activation recomputation
  - GPU Kernels: FlashAttention, Triton, DeepEP, LinearAttention
  - Distributed optimizers: DeepSpeed ZeRO

Model Training Anatomy



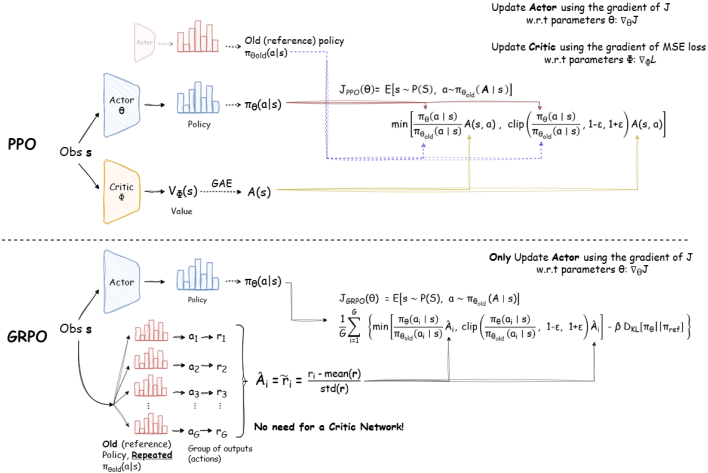
Inference Infrastructure

- **Popular frameworks**: vLLM, SGLang
- **Core technologies**:
  - PagedAttention
  - Continuous batching
  - Prefill-Decode disaggregation
  - Attention-FFN disaggregation
  - Model parallelism
  - Post Training Quantization (PTQ)
  - Speculative decoding
  - Multi-Token Prediction (MTP)

Reinforcement Learning for LLMs

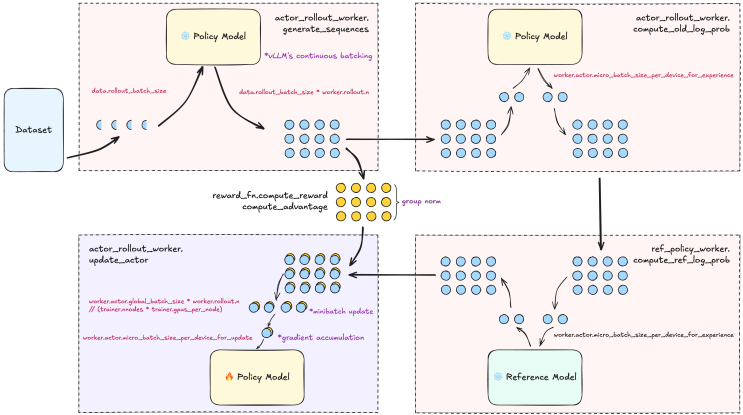
- **Example:** Playing card game
- **Trajectory sampling:** Play one complete game
- **Environment feedback:** Win or lose
- **Advantage function:** Game review and analysis
- **Policy update:** Adjust strategy for next game

RL Algorithms: PPO vs GRPO



Our work: EasyR1 (4.5K GitHub Stars)

An efficient, scalable and multimodality RL training framework for LLMs.



Reasoning Models Timeline

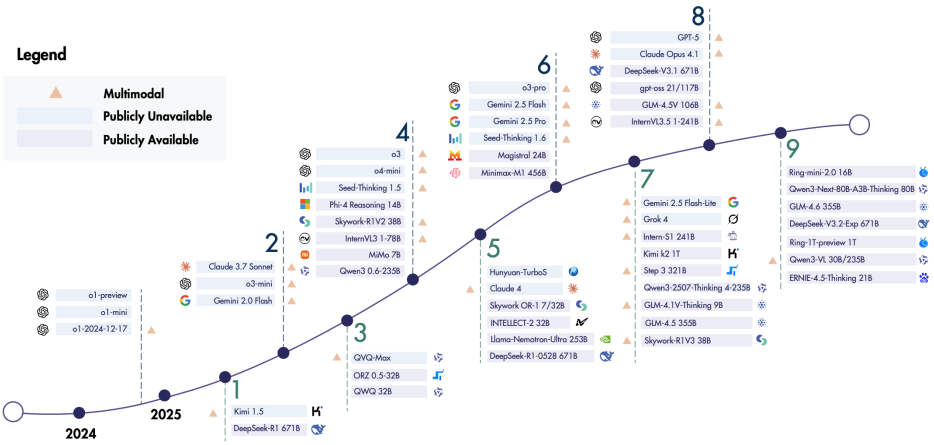


Table of Contents

From Models to Agents

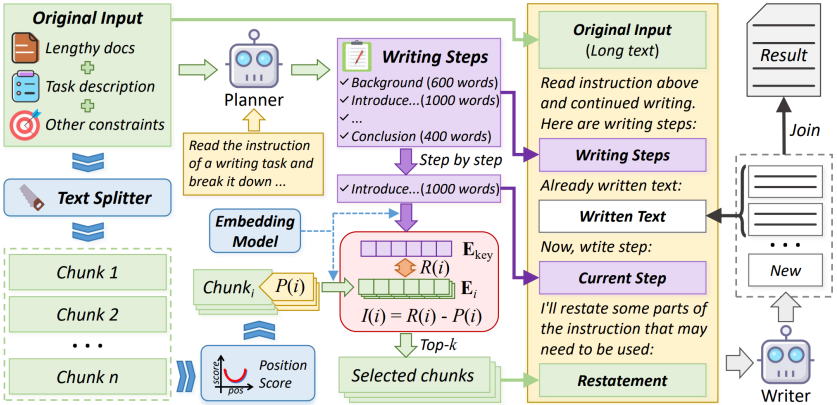
- **Traditional LLMs**: Passive text generators
- **LLM Agents**: Active problem solvers with:
  - Perception (understanding environment)
  - Planning (decomposing tasks)
  - Action (executing operations)
  - Memory (maintaining state)
- Evolution: Prompting $\rightarrow$ RAG $\rightarrow$ Web Search $\rightarrow$ Tools & MCP
- Goal: Autonomous task completion

Stage 1: Prompting Strategies

- **Few-shot prompting**: Few-shot exemplars
- **Zero-shot prompting**: Task instructions
- **Chain-of-Thought (CoT)**:
  - "Let's think step by step"
  - Improved complex problem solving
- **Advanced prompting techniques**:
  - Reflection: Self-evaluation and refinement
  - Planning: Task decomposition and strategy
  - Role playing: Adopting specific personas
  - Debating: Multi-perspective reasoning

Our work: RAL-Writer (arXiv'25)

Proposes a retrieval-augmented long-text writer, combines plan-and-write strategy and context retrieval for controllable long-text generation.

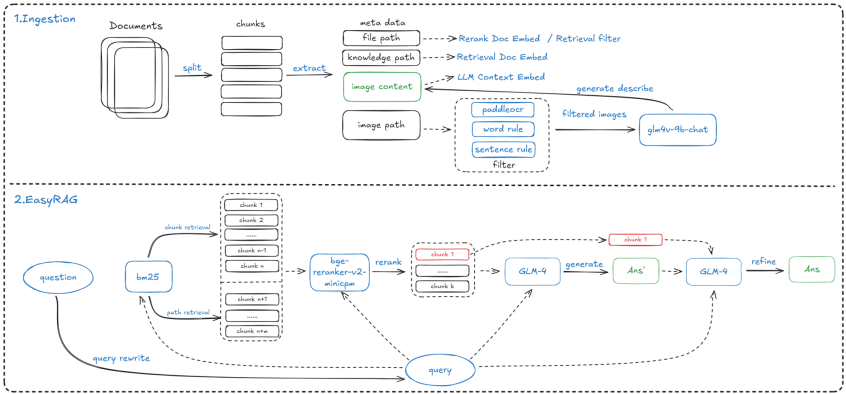


Stage 2: Retrieval-Augmented Generation (RAG)

- **Traditional RAG workflow**:
  1. Document chunking
  2. BM25 or vector-based retrieval
  3. Prompt augmentation
  4. LLM generates response
- **From traditional RAG to agentic RAG**:
  - RAG as a tool that agents can invoke
  - Active decision-making on when to retrieve
  - Dynamic retrieval strategies

Our work: EasyRAG (KDD'25 Workshop, 600 GitHub Stars)

Enhanced Q&A accuracy through query expansion, dual-route retrieval, LLM re-ranking, and LLM answer refinement.

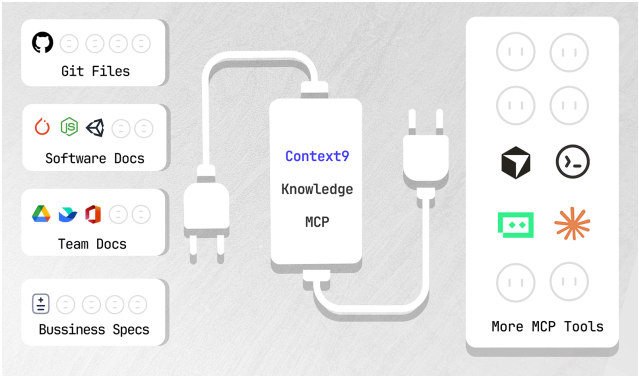


Stage 3: Web Search

- **Comparison with RAG**:
  - Web search: More proactive and dynamic
  - RAG: More passive with pre-indexed knowledge
  - Web search $\approx$ Agentic RAG (both use tool-use capabilities)
- **Key insight**: Web search is essentially tool use
- **Advantages**:
  - Real-time information access
  - Broader knowledge coverage
  - Active query formulation

Our work: Context9 MCP Server

Transforming local knowledge into an LLM-ready second brain through MCP integration.



Stage 4: Tools and MCP

- **Evolution of tool use**:
  - Single tool call → Multi-turn tool calls
  - No reasoning → One-shot reasoning → Interleaved reasoning
- **Expanding tool scope**:
  1. Pre-defined functions
  2. Python interpreter
  3. GUI interactions (embodied AI)
  4. Full system shell access (Claude Code)
- **Model Context Protocol (MCP)**:
  - Standard for connecting LLMs to data sources
  - Unified interface for tools and integrations

Our work: UI-TARS (arXiv'25, 9.1K GitHub Stars)

Breaking GUI agent benchmark records through carefully designed fine-tuning (SFT, DPO, and RL) and diverse agent environments.

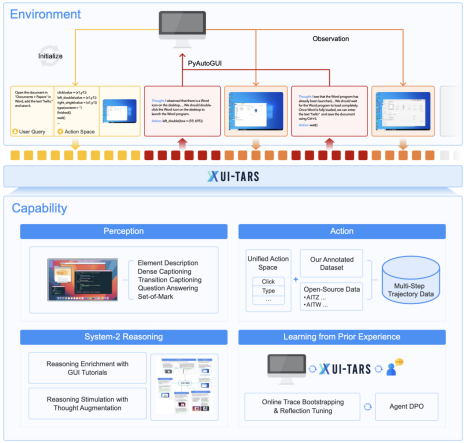


Table of Contents

Long-Horizon Reinforcement Learning

- **Longer reasoning traces**
  - Models will execute deeper multi-step thought processes to solve highly non-trivial problems.
- **Extended interaction rounds**
  - Agents will maintain coherence across much longer sequences of feedback and environment responses.
- **Complex agent environments**
  - Systems will evolve to navigate high-dimensional and unpredictable task spaces with minimal supervision.

Broader Agent Environments

- **Professional softwares**
  - Future agents will seamlessly interface with a vast array of specialized professional software tools.
- **Operating systems**
  - Agents will gain the capacity to operate directly on operating system interfaces and file systems.
- **Edge devices**
  - Intelligence will be deployed locally on hardware to provide low-latency and privacy-preserving agency.

World Models

- **Physical world modeling**
  - Research will focus on teaching models to internalize the causal laws and constraints of the physical universe.
- **Advanced video generation**
  - Generative models will move toward maintaining perfect temporal consistency and object permanence in video.

Continual Learning and Adaptation

- **Nested learning architecture**
  - Google's nested learning framework utilizes varying parameter update frequencies to alleviate catastrophic forgetting.
- **Cross-scenario adaptation**
  - Models will autonomously fine-tune their internal states to excel across rapidly changing task domains.

Thank You!

Questions?