2025 AI Technology Landscape Review: From the End of the "Compute Iron Law" to the Dawn of the "Cognitive Core" Architecture

If we ask which year saw the fastest AI development since the explosion of the large model wave in late 2022, it must be 2025. The speed of development this year has been breathtaking.

A Look Back at the Year:

- **Early Year**: **DeepSeek** completely detonated the training algorithms and paradigms of **GRPO** and **RLVR** (Reinforcement Learning with Verifiable Rewards). This triggered a wave of local deployment of large models by major domestic companies, as well as an industrial and academic craze for **Agentic RL** that lasted all year and will continue into 2026. This spawned a series of methods like **DAPO** and **GSPO**, guiding the industry—following OpenAI o1—toward the route of "slow-thinking inference models with CoT (Chain of Thought)."
- **March**: A massive shift in the Application Layer. The **MCP** (Model Context Protocol) exploded in popularity; OpenAI open-sourced its Agent SDK, diving into the Agent application layer; Google released the **Gemini 2.5** series, marking a qualitative change in capabilities and sounding the counterattack horn in the AI era; **Manus**'s marketing success intensified Agent competition; **Claude 3.7 Sonnet**'s outstanding coding ability brought AI editors like **Cursor**, **Trae**, and **Augment** into the mainstream spotlight.
- **April**: Google proposed the **A2A** (Agent-to-Agent) protocol, attempting to seize the right to set the protocol benchmarks for Agent development.
- **Late April - Early May**: The **Qwen3** model, with its full-modality, full-parameter open-source strategy and high-quality performance, further seized the open-source ecosystem and market. By continuously open-sourcing various models, it increasingly became the **GOAT** of deployment in open-source, academia, and industry, solidifying this position.
- **May**: Claude released the **4-series** models, further consolidating its position as the top choice in the programming domain.
- **June**: Meta CEO Mark Zuckerberg spent $100 million poaching talent; **Claude Code** launched, a revolutionary command-line AI Coding Agent.
- **August - September**: **GPT-5** released; **Claude 4.1** series released; Google released the **Genie 3** World Model. In September, the **Claude 4.5** series was released. The US capital market intensified the "All in AI" trend: Nvidia, Oracle, and OpenAI signed circular investment agreements involving hundreds of billions of dollars; OpenAI announced a $1.5 trillion data center construction plan for the coming years; **Sora 2** AI video began to explode.
- **November**: **GPT-5.1** released; **Google Gemini 3** released, demonstrating the potential of its proprietary **TPU**. Giants like Google, playing the game with strength in both software and hardware, are terrifying.
- **December**: **DeepSeek** released the **v3.2** official version, using its proprietary new algorithm **DSA** (Dynamic/Sparse Attention); **Gemini 3 Flash** released—while incredibly fast, its performance in certain areas surprisingly surpassed the 2.5 Pro.

Based on these observations, we can summarize the following trends:

1. Large Models have shifted towards Inference Optimization and Fine-tuning.

This is not hard to understand: pure large model training will only be a game for a few giants in the future. Even if DeepSeek invents more new algorithms, it cannot break this fact. The reason is simple; two laws are immutable:

- **First, the limitation of the Autoregressive Mechanism.** The mechanism of LLMs is autoregressive causal logical prediction; probability prediction accuracy can never reach 100%. This seemingly insignificant error, after continuous autoregressive prediction, leads to exponentially amplified errors. To improve overall performance, probability accuracy must be optimized to the extreme, which requires massive, diverse, and high-quality datasets.
- **Second, the cost of Implicit Modeling.** As a representative of the **Generation School**, LLMs model the world completely **implicitly**. The laws of the world, knowledge, and various noises are mixed together. The model is not just compressing knowledge; to fit laws, it needs more parameters. For example, the physical law of gravitational acceleration is just a law in reality—it simply exists. But for a large model to fit these laws one by one, to fit the understanding of gravity, it needs more parameters. **Complete implicit modeling inevitably leads to an explosion in parameter count.**

To summarize the logic chain:

① Transformer autoregressive prediction accumulates and amplifies errors -> Giants are forced to pursue 99.99% accuracy -> Requires massive high-quality data -> **"Data Iron Law" turns AI into a heavy-asset game.**

② Because the Generation School uses fully implicit modeling -> Fitting explicit laws requires more parameters -> Model scale expands -> **"Parameter Iron Law" turns AI into a heavy-asset game.**

The "Data Iron Law" and "Parameter Iron Law" are essentially the **Compute Iron Law**.

Many people ask, why haven't DeepSeek's new algorithms broken these iron laws? Because they still use pure **Model-free RL**, relying entirely on brute-force search. It seems they don't need to prepare data in advance, but they are actually "manufacturing" data through "Brute-force Search + RLVR." Performance has improved, but efficiency has actually lowered. Secondly, this method lacks scalability; RLVR can only be used in verifiable domains like mathematics and code, but has limitations in open domains like humanities.

Others say, what about the **MoE (Mixture of Experts)** architecture? MoE only selectively activates weights during inference, but to implicitly model the entire world, it still requires hundreds of billions (B) of total parameters. Their contribution lies in lowering training costs algorithmically, but due to the restrictions of the Iron Laws, the democratization of model training remains out of reach. Moreover, due to the error accumulation effect, global giants are constantly pushing towards the 99.99% goal. Without data, compute, and massive parameters to carry the implicit world, how can ordinary vendors compete?

2. The Second Half of the Application Layer Explodes. Based on the analysis of point 1, this has become inevitable.

3. World Model Theory is the Future. The upper limit of the pure Generation School is OK, but that implies immense financial power. To roll prediction accuracy up to 99.99%, even the resources of giants might not be enough. Therefore, in the future, we must **separate Laws and Knowledge**. Current large models waste a lot of parameters on implicit modeling to balance both.

My Hypothesis and Architectural Conjecture

The future should be as **Andrej Karpathy** mentioned in his podcast: there will be a small-parameter **Cognitive Core**.


有观点
RØDE
将其精简到只保留认知成分
refine even the pre-training set to just narrow


但我们真正需要的是认知部分
But what we really want is the cognitive

This **Core** directly models laws and **First Principles**. Then, we compress the knowledge content itself and train a separate model. The Cognitive Core absorbs user input, transforms it into output derived from first principles, and then uses this output as input for the "Knowledge Content Compression Model." After retrieving relevant information, we can directly train the compression model to integrate laws and knowledge, or train a **third model**, letting the second model act purely as a RAG (Retrieval-Augmented Generation) role, while the third model handles the integrated output of laws and knowledge.

This approach, by separating laws and knowledge, greatly reduces the parameters required for the model to implicitly model laws within data. I will further practice and research this hypothesis in the future.

Note: How do we train the Cognitive Core? My current idea is to take many problems as input and convert the answers into **pure Python blocks or pseudocode blocks** to train a small-parameter model.

Here is an example of the full flow:

User: Analyze why Lin Daiyu died in *Dream of the Red Chamber*.

Cognitive Core (Outputting Logic Code):

```
def analyze_death(person="Lin Daiyu"):
    reason_1 = knowledge_db.search("Lin Daiyu health condition") # Physical cause
    reason_2 = knowledge_db.search("Lin Daiyu psychological state") # Psychological cause
    reason_3 = relation_graph.query("Lin Daiyu", "Jia Baoyu", "marriage_failure") # External trigger
    final_conclusion = synthesize(reason_1, reason_2, reason_3)
    return final_conclusion
```

Compression Model (Stuffed with the original text of *Dream of the Red Chamber*, medical common sense, psychological knowledge):

- Sees `search("health")` -> Its implicit memory immediately recalls "tuberculosis, frail and sickly."
- Sees `synthesize` -> It immediately comes up with relevant vocabulary like "comparative analysis method," "difference analysis," etc.

Integration Model (Input is the output of the Cognitive Core and Compression Model):

- **Output**: A well-reasoned, eloquent essay.

World Model theory, by separating **Simulation**, **Generation**, and **Cognition**, offers hope to solve the problems of massive parameter counts and massive data requirements faced by the Generation School which merges all three into implicit modeling.

Finally, here is a unified modeling formula I envision for how World Models can build AGI (this formula may change with the development of AI and changes in my cognition during practice):

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{A_{t+k}\sim\pi(\cdot|\mathbf{G}_{t+k})}\left[\sum_{k=0}^{\infty}\gamma^k \cdot R\left(\mathbf{G}_{t+k+1} = F_{\text{gen}}\left(O_{t+k} = \mathcal{H}_{\text{cog}}\left(\mathbf{G}_{t+k}, \mathcal{S}_{\text{sim}}(A_{t+k})\right),\ A_{t+k}\right)\right)\right]$$

while G is the Generative World Model,H is the Cognitive World Model and S is the Physics-based Simulation.