

# Z3 Theorem Prover

从逻辑推理到大模型时代的神经符号计算

# | 目录

01 Z3 简介与基础

02 编译与安装方法

03 具体应用场景

04 大模型方向拓展

PART 01

# Z3 简介与基础

---



# Z3 简介

## 什么是 SMT 求解器？

Z3 是 Microsoft Research 开发的高性能 **SMT (Satisfiability Modulo Theories)** 求解器。简单来说，它是一个通用的逻辑引擎，用于判断一组数学约束是否存在解。

## 核心特性

- **多理论支持**：算术、数组、位向量、浮点数。
- **自动化**：无需人工干预即可推导结果。
- **开源**：代码托管于 GitHub，拥有庞大的社区和绑定 (Python, C++, .NET, Java)。

PART 02

# 编译与安装方法

---



# 编译与安装

## 1. 快速安装 (Python)

```
$ pip install z3-solver  
# 验证安装  
$ python -c "import z3;  
print(z3.get_version_string())"
```

## 2. 源码编译 (Linux/macOS)

```
$ git clone https://github.com/Z3Prover/z3  
$ python scripts/mk_make.py  
$ cd build && make  
$ sudo make install
```



A terminal window with a dark background and light-colored text. The window title bar shows three colored dots (red, yellow, green) on the left and the text 'joshu@Josh-MacBook-Pro-188:~' on the right. The terminal content shows two lines: the first line is '~ rm -rf stuff' and the second line is '~' followed by a pink cursor block.

```
joshu@Josh-MacBook-Pro-188:~  
→ ~ rm -rf stuff  
→ ~
```



# 核心理论能力



## 算术理论

处理线性与非线性整数/实数算术。非常适合解决数学规划问题和资源分配约束。



## 位向量 (Bit-Vectors)

精确模拟 CPU 寄存器和内存操作（位移、溢出、异或）。是硬件验证和逆向工程的基础。



## 数组与函数

支持未解释函数 (EUF) 和数组理论，用于抽象复杂的系统状态和内存模型。

PART 03

# 具体应用场景

---



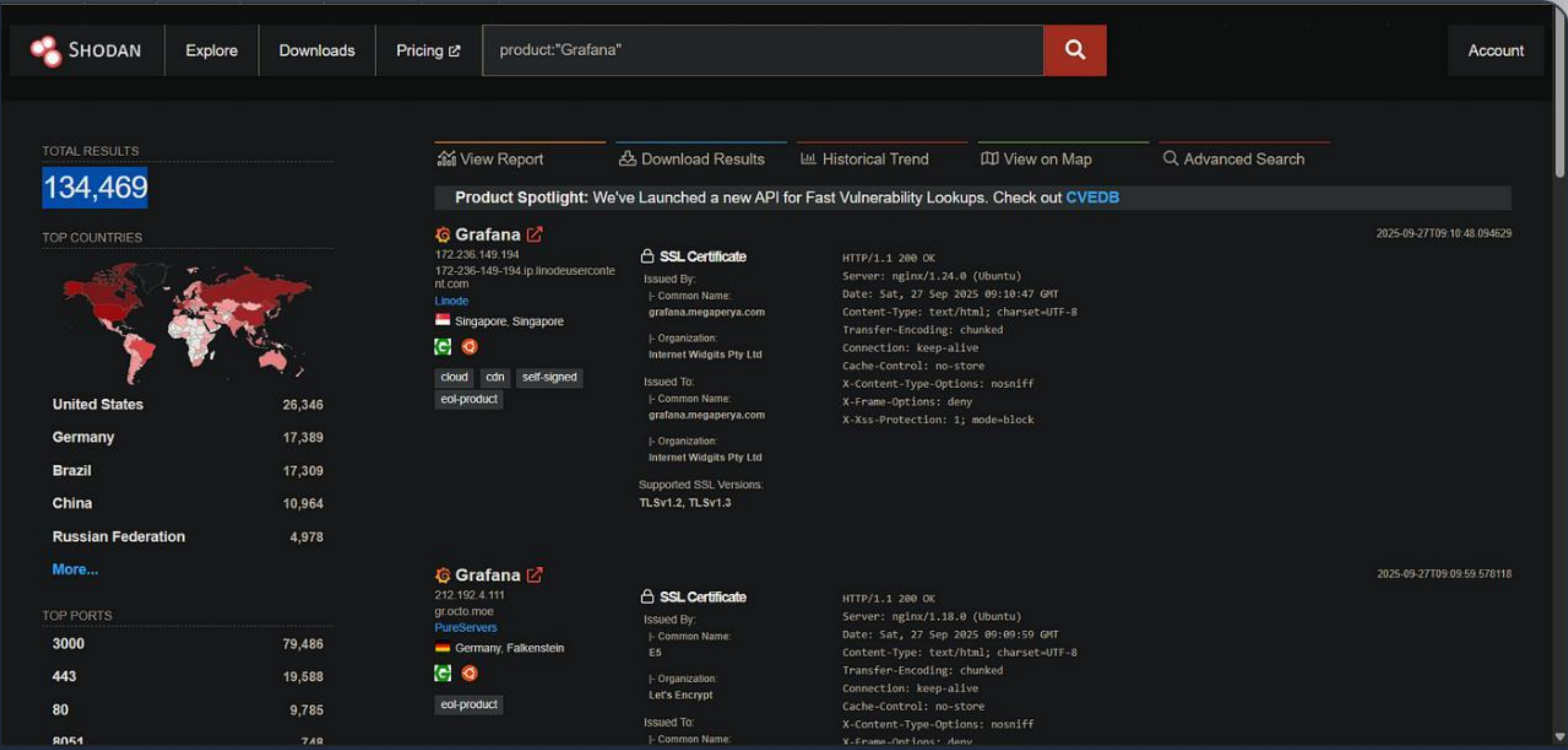
# 应用 I：软件验证

寻找“不可能”的 Bug：

通过符号执行 (Symbolic Execution)，工具如 KLEE 或 Angr 将程序代码转换为 Z3 约束。Z3 可以计算出导致程序崩溃（如除以零、数组越界）的精确输入值。

形式化证明：

在关键系统（航空航天、区块链）中，使用 Dafny 等语言编写规范，Z3 作为后端引擎自动证明代码在数学上是绝对正确的。





## 应用 II：安全与逆向



### CTF 神器

在网络安全竞赛 (CTF) 的逆向题中，经常遇到复杂的校验算法。与其手动逆推，不如将校验逻辑写成 Z3 约束，让求解器自动算出 Flag。

### 漏洞挖掘

通过污点分析 (Taint Analysis) 追踪用户输入，结合 Z3 求解路径条件，可以自动化生成 Exploit 载荷，验证缓冲区溢出等严重漏洞。

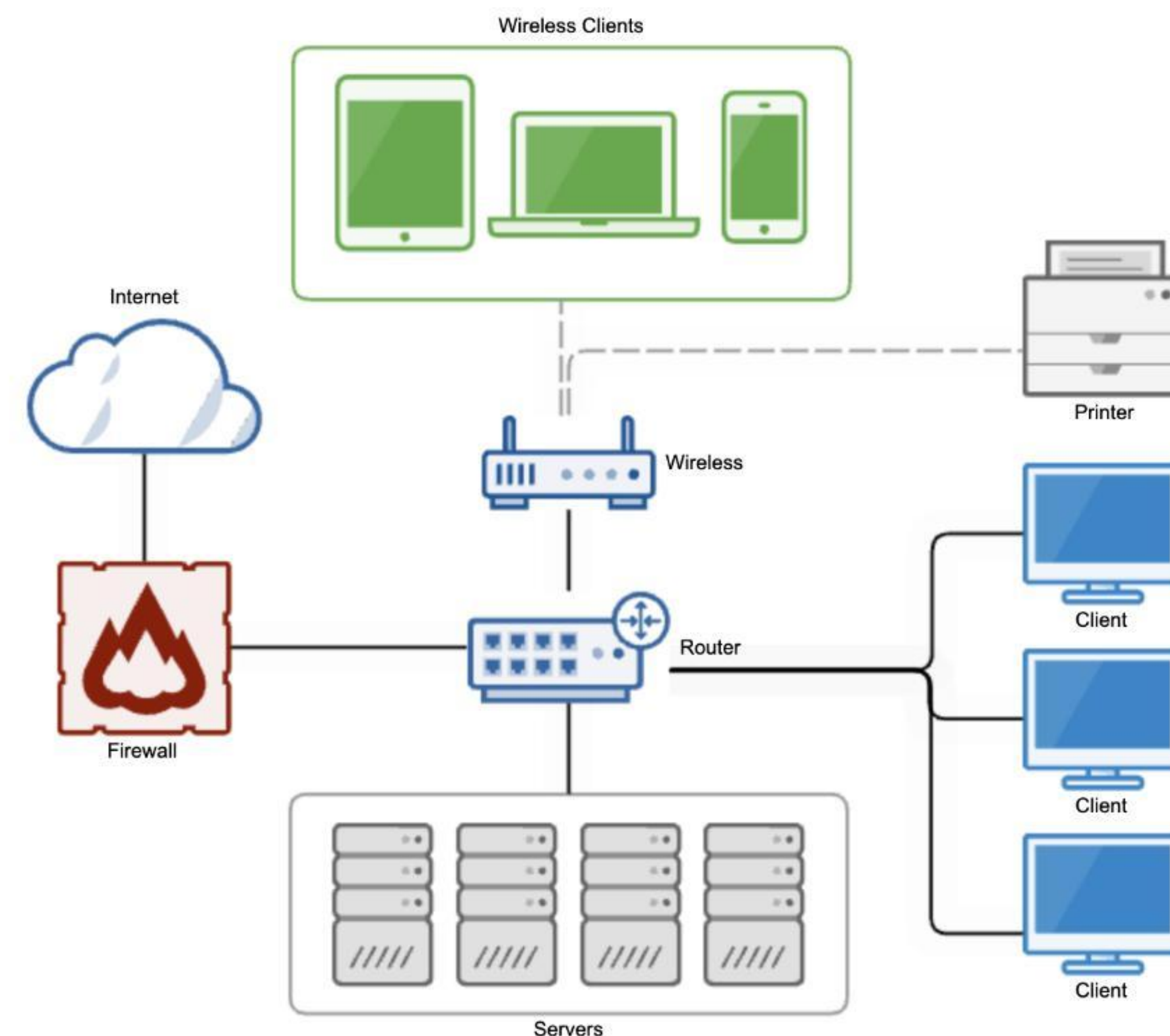


# 应用 III：云网络验证

## Azure Firewall Checker

微软 Azure 云平台使用 Z3 来实时验证防火墙规则。

- **冲突检测**：自动发现相互矛盾的安全规则。
- **可达性分析**：数学上证明 "VM-A" 是否能访问 "Database-B"。
- **规模**：能够在毫秒级处理数千条复杂的网络规则。





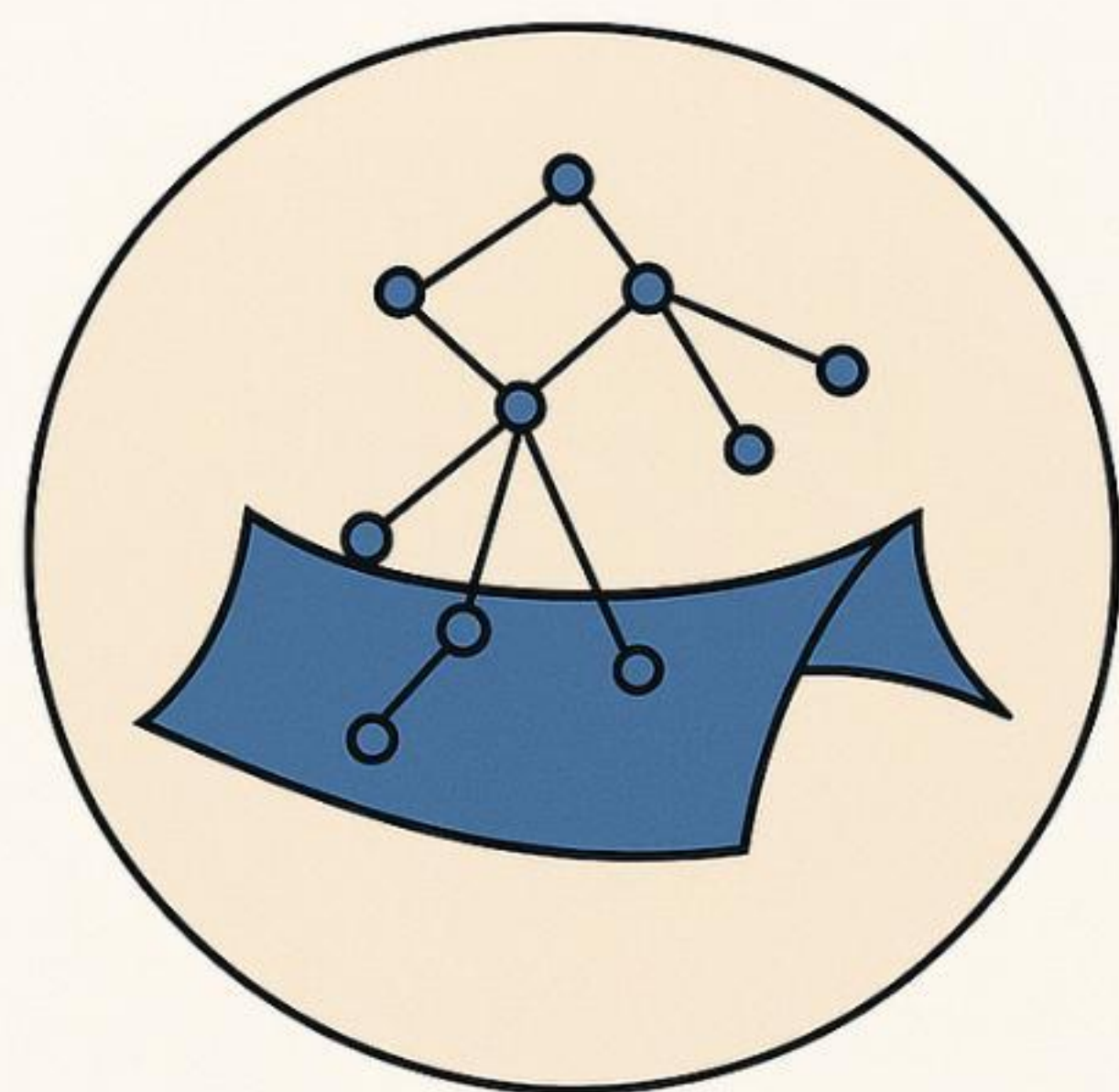
PART 04

# 大模型方向拓展

---

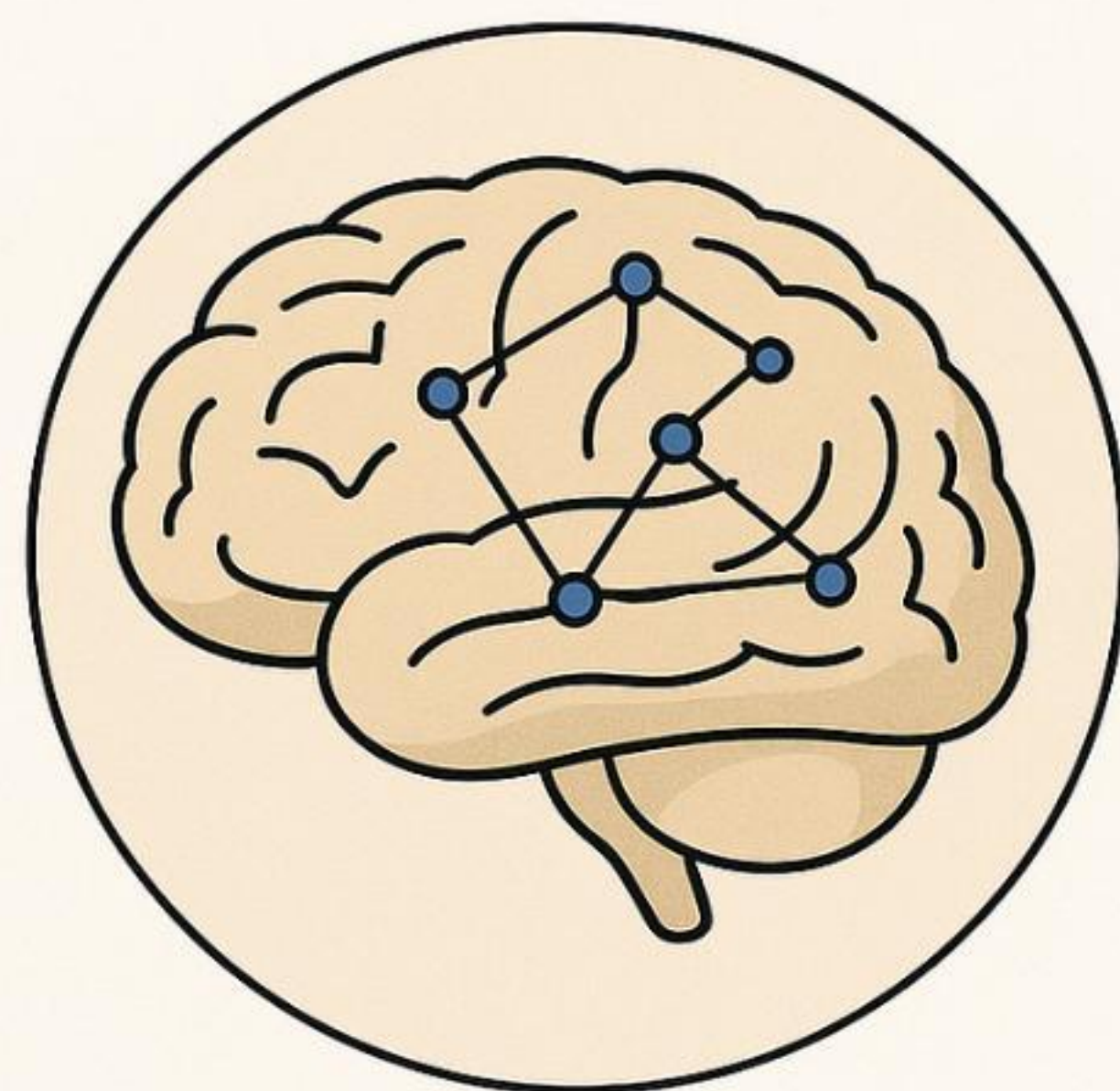


# How Hyperbolic Latent Spaces Make AI More Brain-Like



Hyperbolic Latent Space in AI

and



Brain-Inspired Geometry

## 大模型方向拓展

### 神经符号计算 (Neuro-Symbolic AI)

**痛点：** LLM 擅长生成文本，但在这逻辑推理和数学计算上容易“产生幻觉”。

**解决方案：** 将 LLM 作为“翻译器”，将自然语言问题转化为 Z3 代码；将 Z3 作为“验证器”，执行代码并返回绝对正确的结果。

这种组合完美互补：LLM 提供灵活性，Z3 提供严谨性。



# workflow : Program of Thought

1

**用户提问**

"鸡兔同笼，头35，脚94..."

2

**LLM 生成**

生成 Z3 Python 脚本建模

3

**Z3 求解**

执行脚本，进行数学推导

4

**最终答案**

返回 Proven 结果



# Q & A

Thank you for listening