第三部分：CUDA 工程实现 (The Metal)

🛠️ 核心代码展示 (Sanitized)

我们展示一段核心的 CUDA Kernel 代码，证明我们对底层硬件的掌控力：

```
// HLPO Core Kernel: Sparse Attention Gate (Snippet)
__global__ void sparse_gate_kernel(...) {
    // 2. Register Loading (Optimization)
    // Load Query vector into registers to maximize reuse
    float q_reg[BLOCK_D];

    // 3. Dynamic Sparse Traversal (The Logic)
    // Only iterate through valid blocks defined by the Index Map
    // Skipping empty regions (Hypersparsity)
    int num_valid = *cnt_ptr;

    for (int k_idx = 0; k_idx < num_valid; ++k_idx) {
         int target_block = lut_ptr[k_idx];

         // 4. Tensor Core Operation Simulation
         // Perform MatMul and Accumulation in high precision
    }
}
```

(完整代码见 `Code_Snippet.cu`)

💡 核心洞察

“我的框架不仅能写 Python，还能写 Tensor Core 原生代码。”

HLPO 不仅仅是一个算法，它是一套穿透软件栈的哲学：

- **软硬同构**: 我们在 Python 层定义的 “稀疏逻辑”，在 C++ 层被无缝翻译为 Tensor Core 的指令。我们不依赖黑盒库，我们直接对话硅片。
- **一气呵成**: 从高层的意图定义，到底层的寄存器分配，整个链路是透明且通过数学严密推导的。