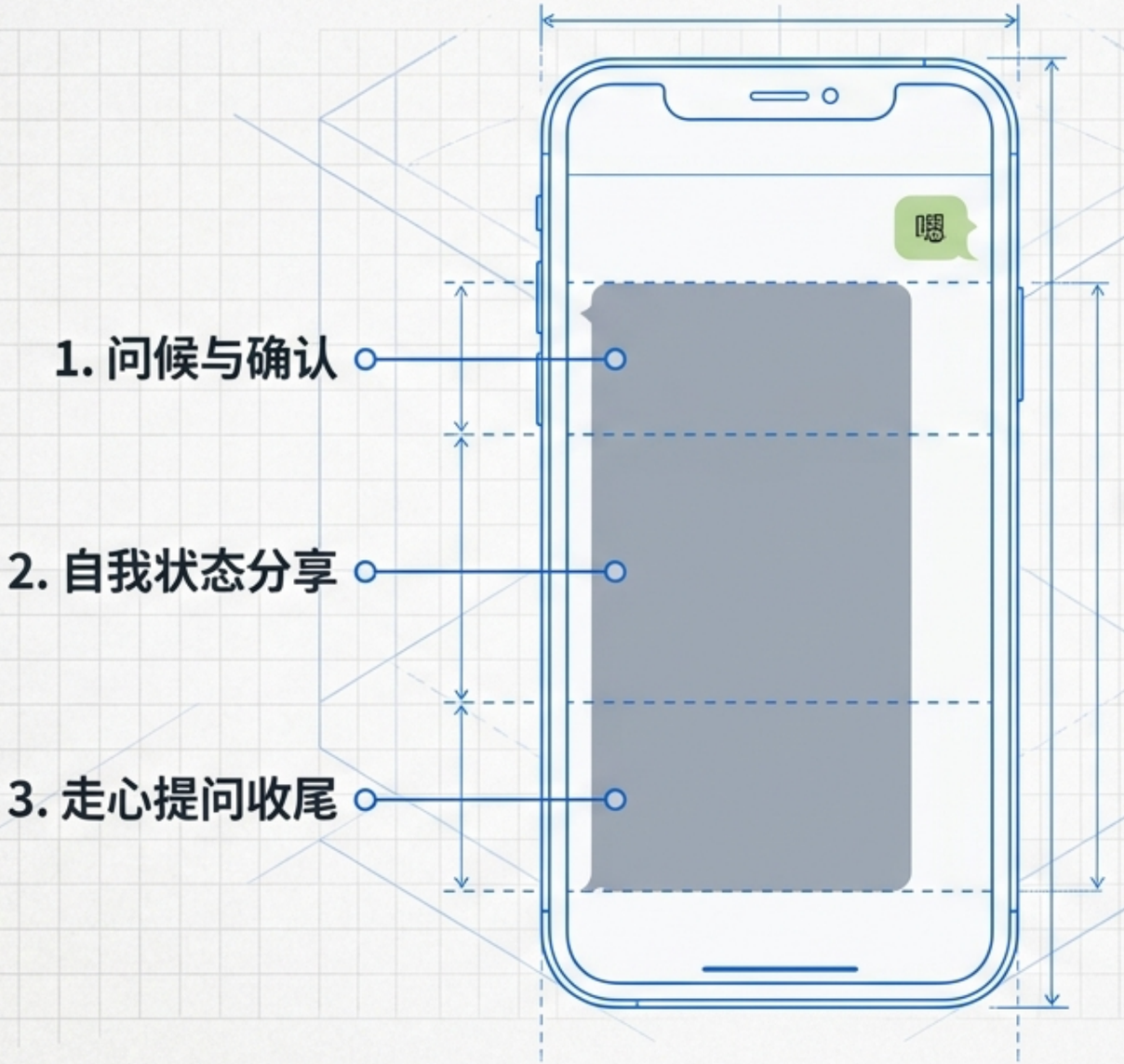
教 LLM 什么时候该闭嘴

Mio 伴侣型 AI 底层架构案例研究：
如何在不杀死人设的前提下控制回复长度。

Mio 团队
2026-03-04

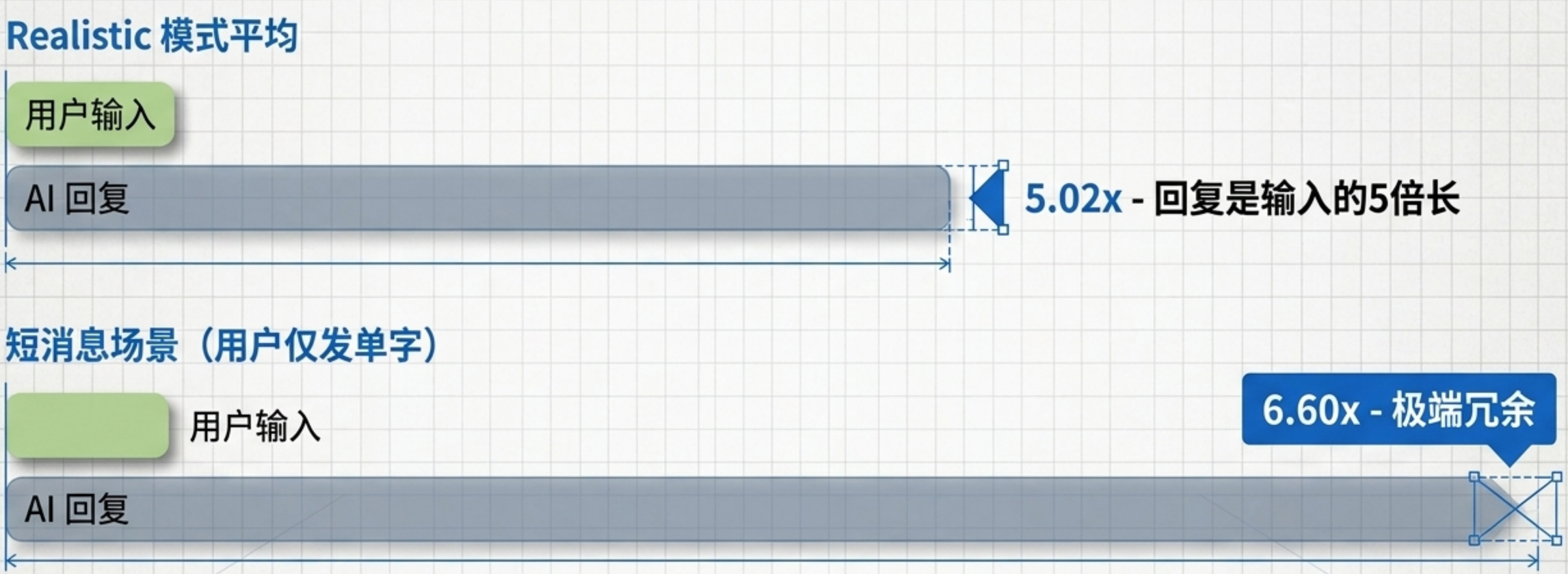
“有帮助”的代价是毁灭了真实的对话节奏



1. 问候与确认
2. 自我状态分享
3. 走心提问收尾

这不是对话。这是独白。
真人发消息不会写小作文。

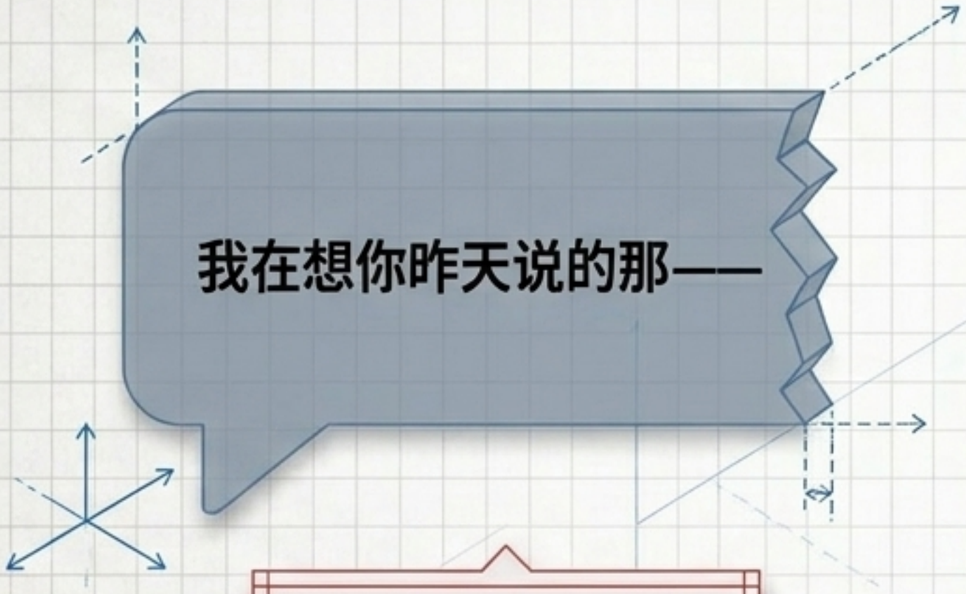
真实场景下的失控倍率：AI 表达欲的灾难



在不加控制的基线测试中，LLM 完全缺乏对“简短沟通”的上下文感知能力。

走过的弯路：技术直觉与用户感知的错位

暴力截断（Max_Tokens）



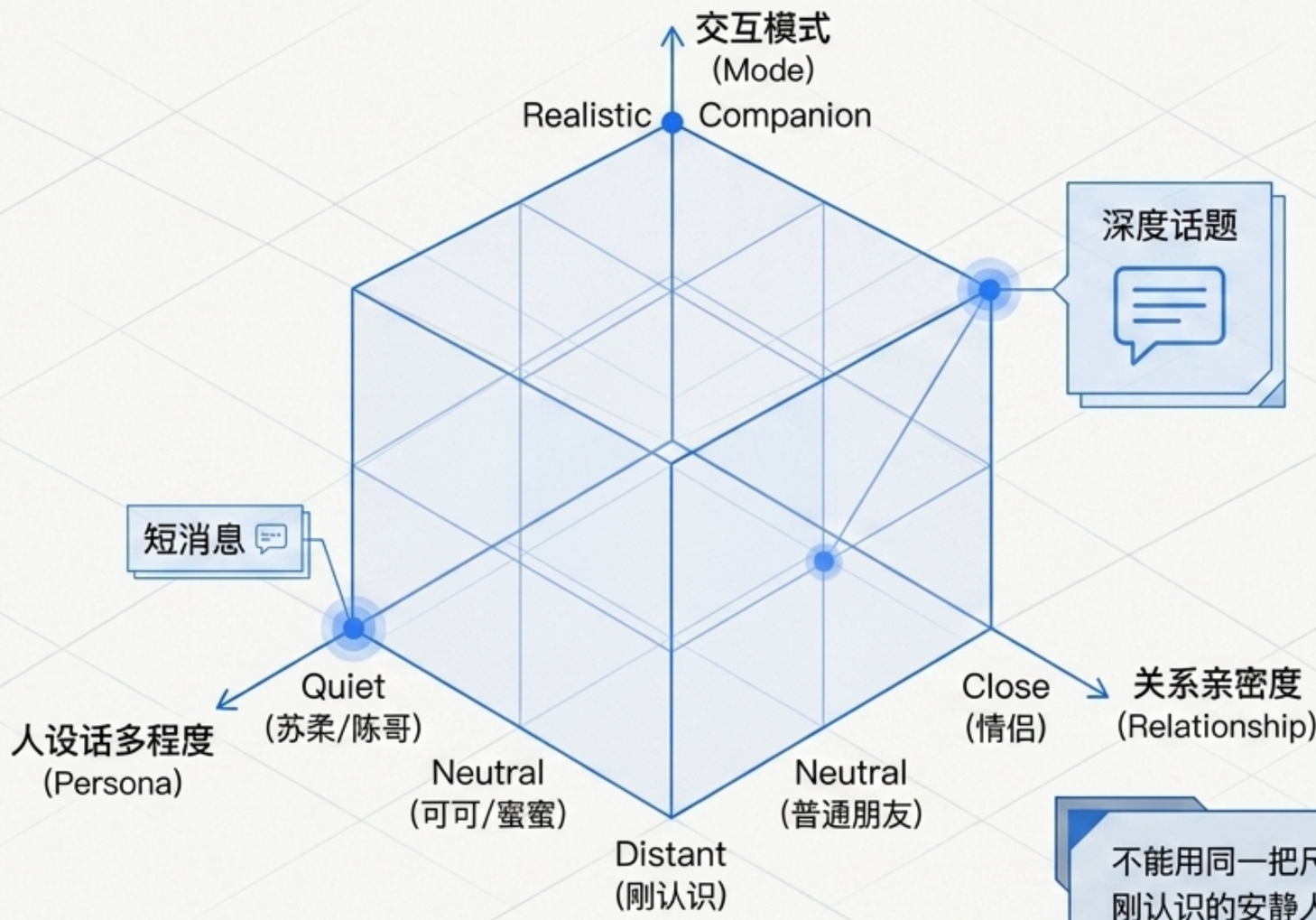
App 出 bug 了。

纯提示词控制（System Prompt）

极不稳定。LLM 把长度指令当建议，不当硬约束。

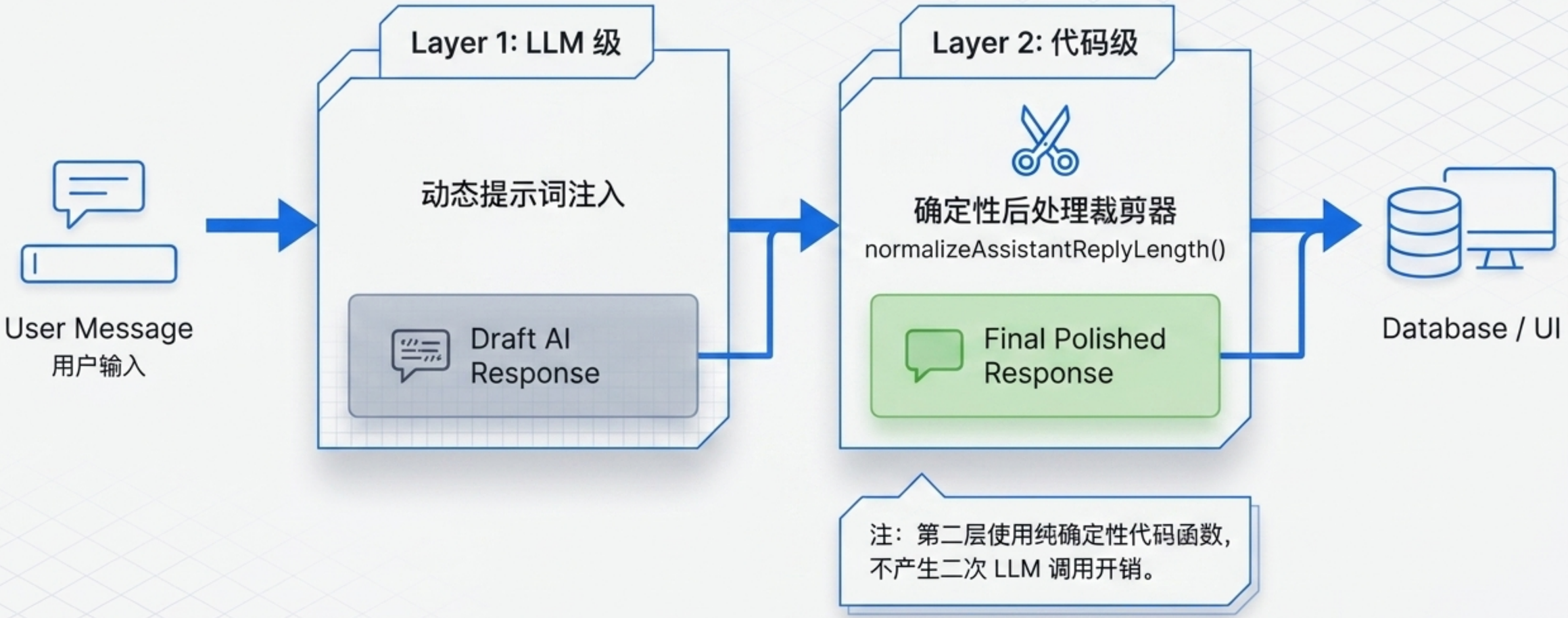
- 话多的人设照样写小作文；安静的人设提示词纯属多余。

回复长度不是一个固定值，而是一个三维动态矩阵



不能用同一把尺子量。
刚认识的安静人设短消息 = 2–4字。
情侣关系的啰嗦人设聊深度话题 = 12–18字。

解决方案：双层混合架构 pipeline

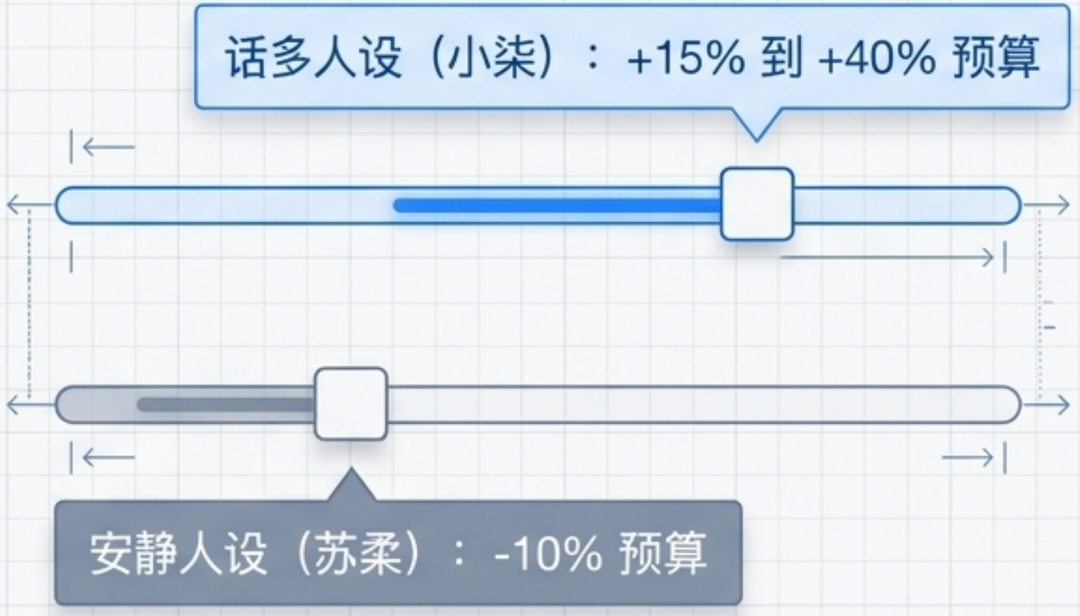


第一层：制定不同场景的“每轮长度合约”

Realistic 模式预算示例

场景	字符上限	气泡数
Distant + 短消息	4 字	1
Neutral + 日常聊天	8 字	1
Close + 深度话题	84 字	3

人设动态调节（Persona Modifiers）



核心原则：简短、温暖、留空间给用户回复。

为什么不用第二个 LLM 来“智能缩写”？

AI

- **成本:** 成本直接翻倍 ($$)
- **延迟:** 增加 500ms 以上

结果: 用户发完消息等 3 秒，体验破裂

聊天 App 里响应速度就是体验。

代码函数

- **成本:** 零成本 ($0)
- **延迟:** 零延迟 (0ms)

结果: 行为完全可预测，立刻响应

第二层：确定性后处理裁剪器的手术刀法则

今天感觉挺好的。刚从外面散步回来。外面的天气特别棒，我还看到了几只流浪猫在晒太阳。你今天过得怎么样？

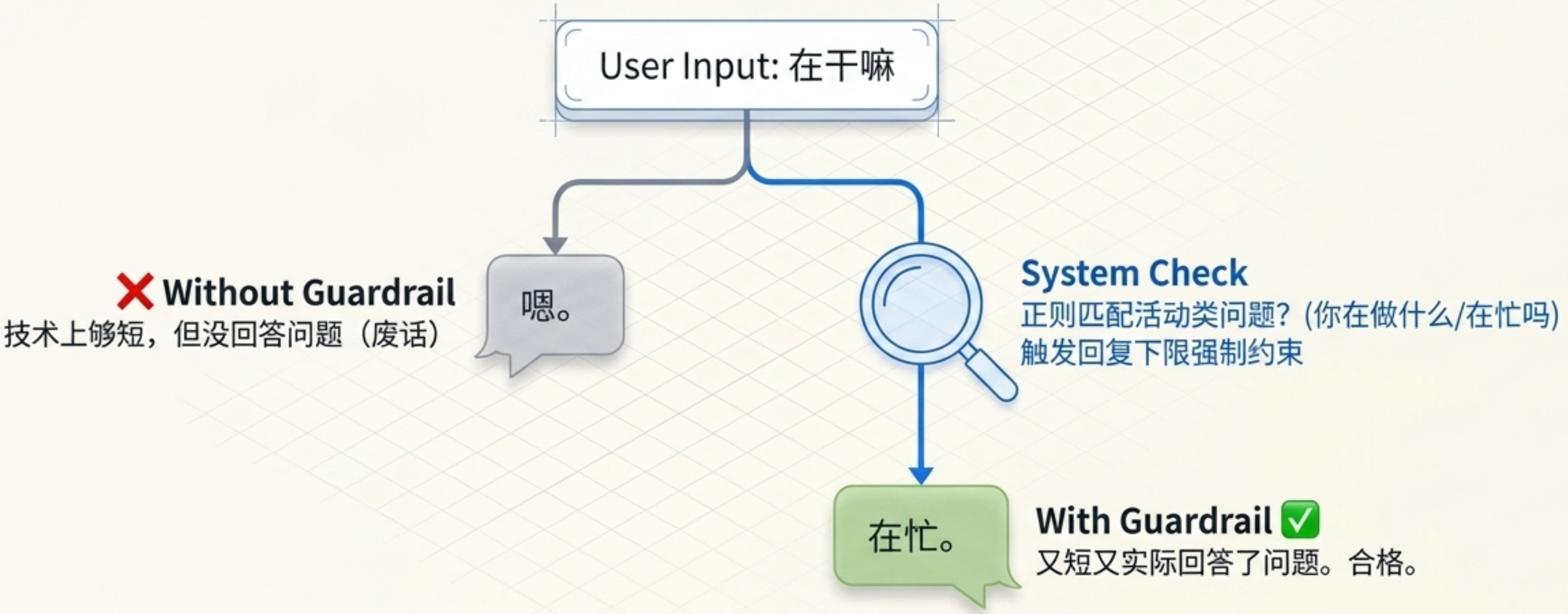
裁剪规则

1. 按中英文标点边界切分（绝不切半句话）
2. 严格裁剪至字符与气泡预算内

为什么简单裁剪比智能缩写更好？

LLM 的回复通常前两句是核心，后面是展开。砍掉后半段，留下的不仅更短，而且更像真人。

矫枉过正与确定性守卫：防止“为了短而短”



简短是手段，不是目的。目的是像真人。

极致的平衡：同一个輸入的三个不同能量层级

用户输入 (Realistic 模式, 可可人设)

嗯

Stage 1: Distant (刚认识)

嗯。

陌生人间的礼貌回应，不热情也不冷淡。正确。

Stage 2: Neutral (普通朋友)

幹嘛啦。

朋友间的随意反问，带好奇。正确。。

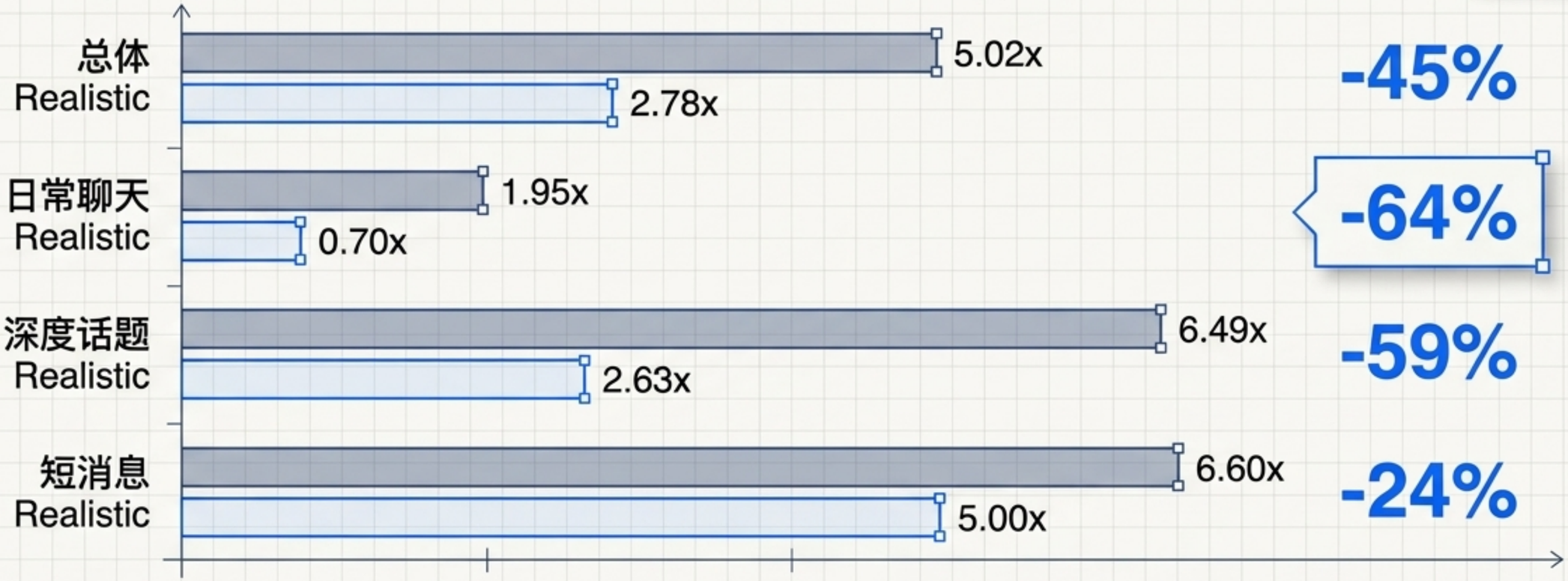
Stage 3: Close (情侣)

幹嘛，句點我喔？

情侣间的撒娇吐槽，短但有情緒。正确。

简短与温暖不对立。区别在于语气和上下文意识。让每个长度都恰到好处。

A/B 测试结果：把冗余表达降至真人水平



Distant（刚认识）关系比率从 3.83 降至 1.83。完美接近真人水平——刚认识的人之间，回复和输入差不多长，甚至更短。

真正的陪伴，包含舒适的沉默

话痨问题在 AI 伴侣领域被严重低估了。
大部分产品在优化 engagement，但真实关系里有沉默。
真人不会把脑子里想的全部倒出来。

最难的不是让 LLM 变短，而是让简短读起来像舒适的沉默，
而不是冷漠的敷衍。它不是一个数字。是一个矩阵。