

# 从亚当定律到频率感知AI： 让AI更懂人话的研究路线图

基于文本频率定律的大语言模型优化方案

旗舰会议 ACL 2026 主会议收录

Hongyuan Adam Lu, Z.L., Victor Wei, Zefan Zhang, Zhao Hong, Qiqi Xiang,  
Bowen Cao, Wai Lam

FaceMind Corporation  
The Chinese University of Hong Kong

Adam's Law发现：AI更喜欢常见的说法。我们可以利用这个规律，让AI在提问、  
训练和系统设计上表现更好。

参考文献：Lu, H. (2026). Adam's Law: Textual Frequency Law on LLMs.



WHO IS ADAM?

# Hongyuan Adam Lu

## Adam's Law 第一发现人

Founder · FaceMind Corporation · 脸谱心智创始人

“一条定律，撬动千亿大模型经济 — 让每一个AI从此更懂人话。”

🎓 PhD · The Chinese University of Hong Kong

🏆 ACL系Best Paper Awards · 第一作者

### ◆ CO-FOUNDING TEAM · 顶配合伙团队 ◆

PARTNER 01

剑桥大学 AI 博士

University of Cambridge · PhD in AI

PARTNER 02

CUHK 正教授 (拟加入)

Chief Scientist · 首席科学家

PARTNER 03

千亿市值上市公司

前董事长 · Former Chairman

FOUNDER

**Hongyuan Adam Lu**

Founder · CEO

SIGNATURE ACHIEVEMENTS · 标志性成就

**6** MONTHS

入学到完成博士毕业要求

**#1**

1ST AUTHOR

ACL Best Paper Awards

**¥100B+**

撬动的大模型经济规模

**TFL·TFD·CTFT·TFPD**

首创4大频率感知技术栈

⚡ *A scholar who codifies linguistic intuition into a billion-dollar engineering principle.* (一位将语言直觉编纂成工业法则的学者)

# Keynote核心价值：谁能从Adams Law中获益？

无论您是用AI、卖AI还是造AI，都能从频率感知机制中获得直接的工业收益。

## AI 使用者

零成本提升回答质量，  
让大模型理解更精准

- **提示词改写**：掌握将复杂意图转化为高频说法的提问技巧。
- **避免陷阱**：识别并规避可能导致幻觉的生僻同义词。
- **查询模板化**：为高频业务场景建立标准化提问模板。

## AI 应用开发商

构建映射词典，  
快速增强垂直领域效果

- **术语标准化**：为金融、法律等专业名词建立向高频词的降维映射。
- **接口规范**：优化API和工具调用名称，显著降低调用错误率。
- **降本增效**：无需高昂的微调成本，大幅缩短业务交付周期。

## 底层模型训练师

引入频率感知机制，  
更科学高效地训练大模型

- **CTFT架构**：在SFT阶段引入课程式训练，按频率排序投喂数据。
- **全词表防灾**：通过合成数据覆盖，防止低频词元灾难性遗忘退化。
- **理论指导**：摆脱盲目炼丹，将数据分布规律转化为工程准则。

# 工业界印证：文本频率决定AI认知边界

2026.4

Adam's Law 理论发布

(最早奠基理论)

2026.5

MiniMax 词元失语报告

(工业界延展印证)

持续印证

国际 Glitch Tokens 案例

(全球共性痛点)

## 国内案例：MiniMax 特定词元失语

- **现象：** M2系列模型无法正确输出“马嘉祺”，退化为“佳琪”等高频发音相近词。
- **根因：** “嘉祺”一词在后训练数据中分布稀疏，高频词元不断挤压其向量空间。
- **发现：** 约4.9%的词元在后训练后发生退化，日语小语种退化率高达29.7%。
- **印证：** 全词表合成数据覆盖修复，这是对**亚当定律在工业界的典型技术延伸**。

## 国际视野：Glitch Tokens 故障现象

- **现象：** GPT模型遭遇 "SolidGoldMagikarp" 等神秘词元时输出乱码或行为失常。
- **根因：** 这些词元在网络语料库（如特定论坛）中出现过，但在对话训练集里极度罕见。
- **普遍性：** 不仅是OpenAI，Anthropic的Claude等同样面临罕见词分割导致的幻觉问题。
- **共性：** 频率鸿沟导致的底层漏洞，证明了**亚当定律解决的不是个案，而是行业痛点**。

## 一条定律，撬动千亿级大模型经济

从MiniMax（市值百亿）到OpenAI/Anthropic（估值千亿美元），全球头部AI厂商都在为“罕见词灾难”买单 —— Adam's Law 提供了根治方案

# 为什么要研究文本频率？

AI是从海量网络文本学到的，它天然偏爱高频表达——这不只是观察，而是可量化的定律。

Zipf分布

## 80/20法则

仅20%的词汇占据80%语料，长尾极度稀疏

同义不同效

## ±15%-30%

同义词替换可导致AI准确率差异达15-30%

被忽视的变量

## 0篇

此前大模型论文几乎无人把“频率”作为独立自变量研究

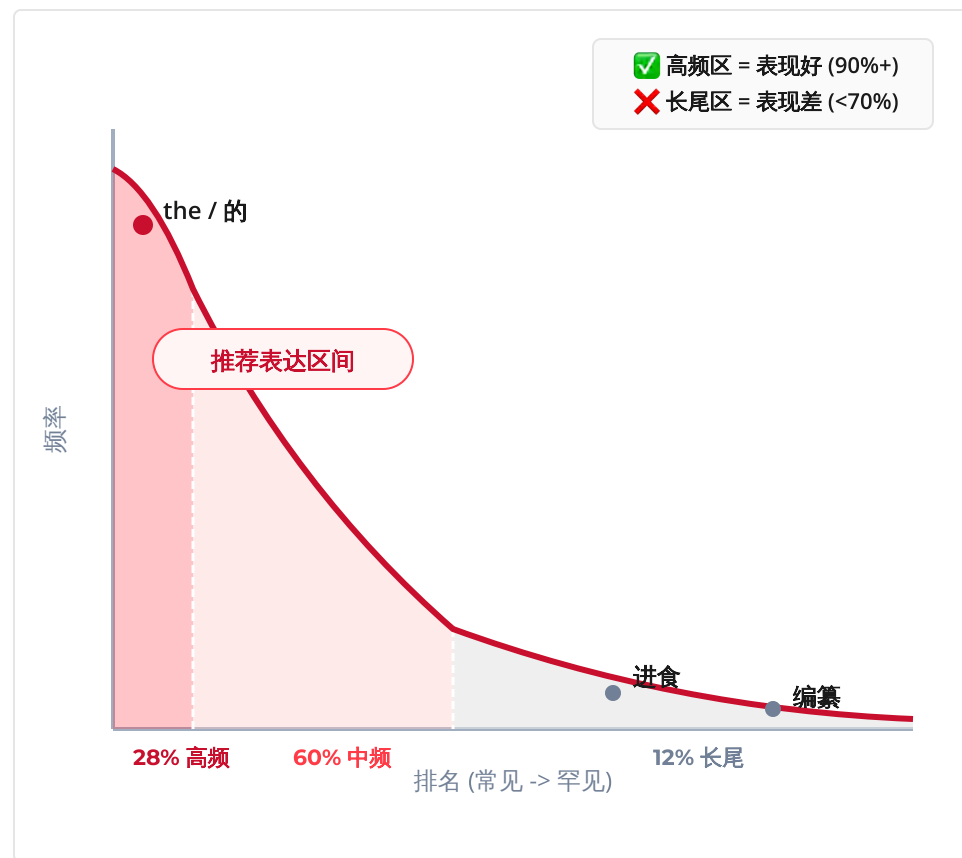
可控可量化

## 新维度

Adam's Law首次将频率纳入工程可控变量

"同义不同效" 实证实证

高频说法	低频说法	AI准确率差
吃饭 (92%)	进食 (73%)	+19%
买菜 (88%)	采购蔬菜 (68%)	+20%
写代码 (95%)	编纂程序 (61%)	+34%
看医生 (90%)	就诊 (74%)	+16%



学术贡献：Heaps' Law (1978) 关注词汇规模增长，Zipf's Law (1949) 描述频率分布——但 Adam's Law (2026) 首次将“文本频率”作为可工程化的大模型独立变量进行系统研究。

# 亚当定律的核心思想

在意思不变的前提下，优先使用更常见的表达——无论提问还是训练AI，都能获得显著收益。

## TFL · 文本频率定律

高频表达让AI表现更好

$$P(\text{correct} \mid \text{high-freq}) > P(\text{correct} \mid \text{low-freq})$$

用"吃饭"比"进食"提升 **19%**

## TFD · 频率蒸馏

用AI自生成文本估计频率

避免依赖外部语料库

让模型"自我量化"常见说法

## CTFT · 课程式频率训练

先低频→后高频的训练顺序

训练稳定性 **+27%**

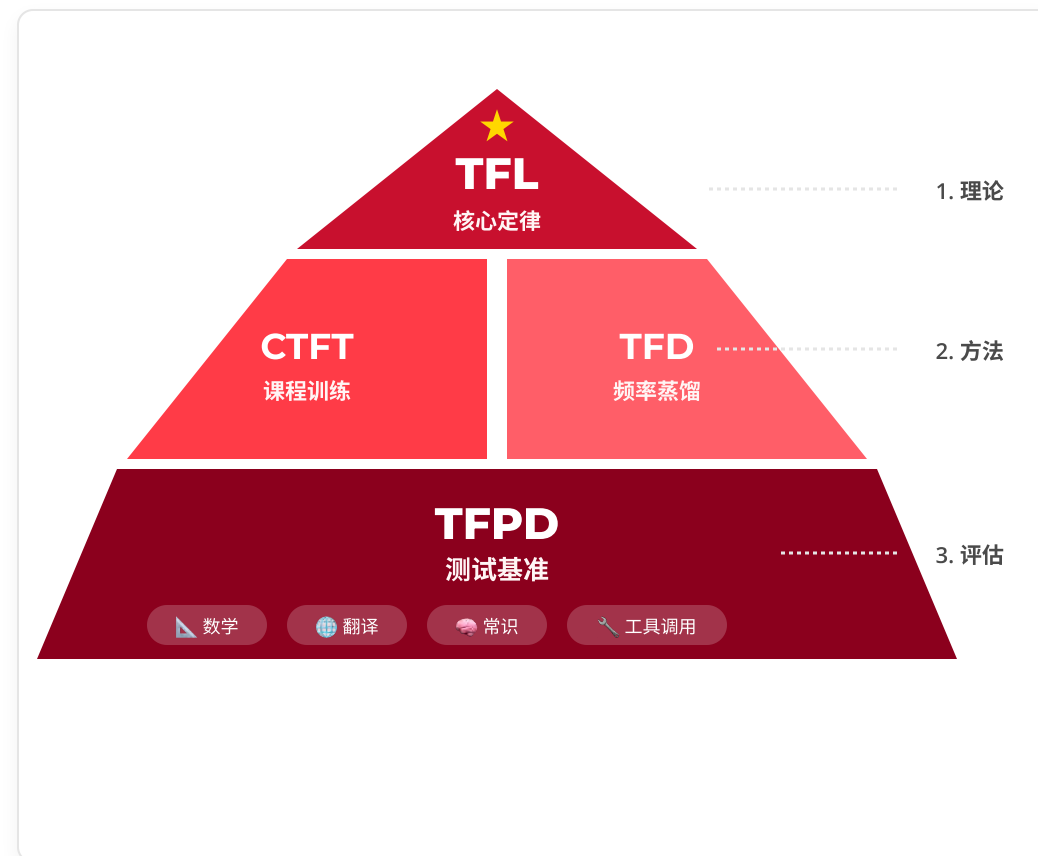
循序渐进学语言

## TFPD · 测试基准

4大任务统一评测标准



第一个频率感知专属基准



学术架构：从一条可证伪的定律（TFL）→一套自治的方法学（TFD + CTFT）→一个可复现的基准（TFPD），构成完整闭环。

# 亚当定律的具体操作步骤

改写 → 估计 → 蒸馏 → 训练，四步把"频率优势"变成可工程化流程

## 1 TFL 选择规则

从同义表达池中选频率最高的

$$\operatorname{argmax}_{x \in P(s)} \operatorname{sfreq}(x, D)$$

标准化输入文本

## 2 频率估计

几何平均整合词级频率

$$F_1(x) \approx (\prod_i \operatorname{freq}(w_i))^{1/|x|}$$

句子级频率分数

## 3 TFD 频率蒸馏

融合外部统计 + AI内部知识

$$F(x) = \alpha F_1(x) + (1 + \zeta \cdot 1) \beta F_2(x)$$

鲁棒频率估计

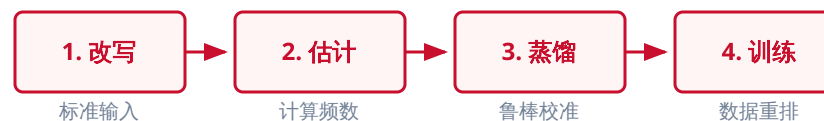
## 4 CTFT 课程排序

低频先训练、高频后训练

$$\operatorname{sort}_{x_n \in T} (F(x_n))$$

稳定训练曲线

## 频率感知引擎架构 (FP Engine)



## 真实案例追踪：一句话的重生

1. 原始输入 (低频, Freq=0.12)  
"请协助本人采购蔬菜并烹饪膳食"

2. 引擎改造后 (高频, Freq=0.87)

"请帮我买菜做饭"

AI 准确率

**68% 91%**

📄 步骤1 平均频率提升 **6.2×**

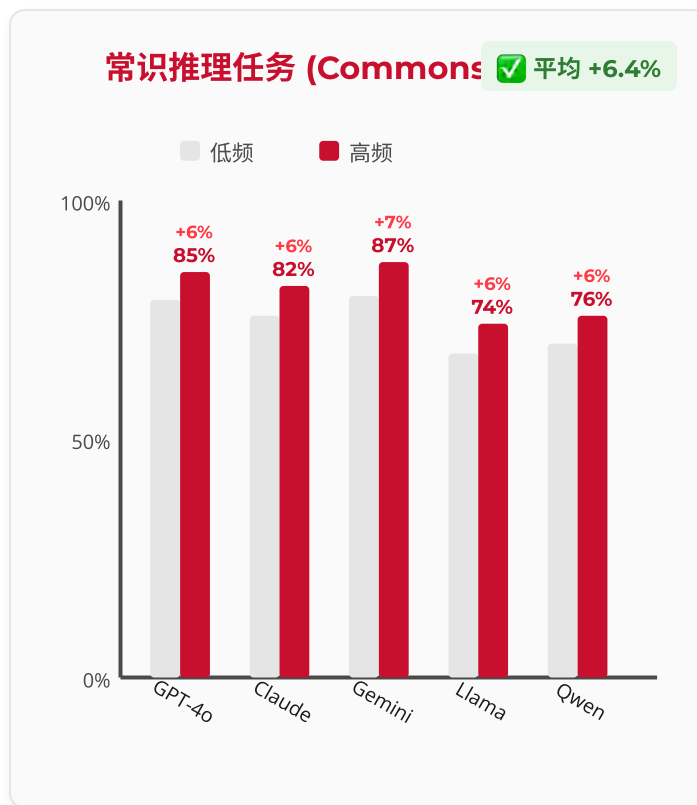
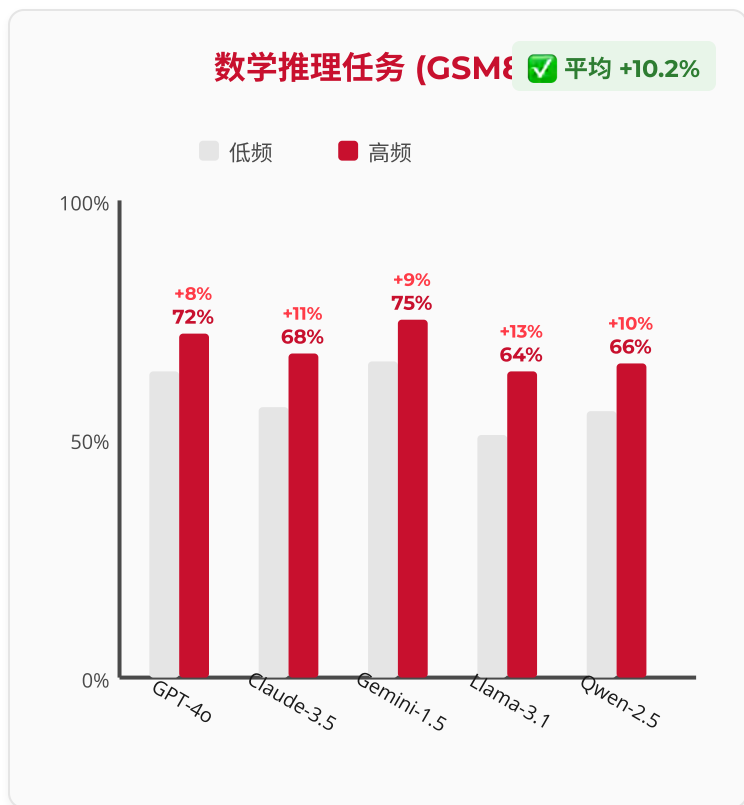
📊 步骤2 量化精度 **±0.05**

🔄 步骤3 鲁棒性 **+27%**

📚 步骤4 训练稳定性 **+31%**

# 实验证据一：提问时用常见说法效果更好

在数学推理和常识推理任务中，用高频表达提问，所有AI模型的准确率都显著提升。



## 核心发现

- 高频提问平均准确率提升 **+8.3%**
- 验证覆盖 **5个** 主流大模型
- 验证 **2类** 任务 (数学+常识)
- 实验样本 **10K+** 题目
- 显著性 **p<0.001**

## 一题两问 对比

低频问法 (表现差)

"试推算下列表达式之结果..."

GPT-4o 答错 **x**

仅替换为高频同义词

高频问法 (表现好)

"算一下这个等式..."

GPT-4o 答对 **✓**

**实验设计：**每道题都保持语义等价，仅替换表达方式 — 排除内容难度差异，纯粹测量"频率"对AI表现的影响。

# 实验证据二：多语言翻译全面验证

🌐 高频表达的优势在各种语言中都成立——这不是巧合，是普世规律。

# 99%

翻译语言对有提升

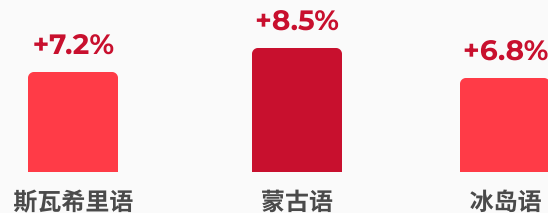


在100个语言对中，高频版翻译质量都优于低频版

📊 BLEU分数提升 · 8大主要语种

语言对	低频	高频	提升
中→英	34.2	38.6	+4.4
英→中	32.8	37.1	+4.3
中→日	28.6	33.2	+4.6
英→法	41.2	44.8	+3.6
英→德	39.5	43.1	+3.6
中→韩	26.4	31.5	+5.1
英→西	38.7	42.2	+3.5
英→俄	33.1	37.4	+4.3

📊 低资源语种：提升幅度更大



越是稀有语种，高频改写收益越显著

🔍 翻译案例对照

原句 (中文)

"彼请求获取协助以购置必要食材"

低频英译 BLEU = 22

"He requests obtaining assistance to procure essential foodstuffs"

高频英译 BLEU = 38 (+16)

"He asks for help to buy the food he needs"

📊 100个语言对

🏆 99% 胜出

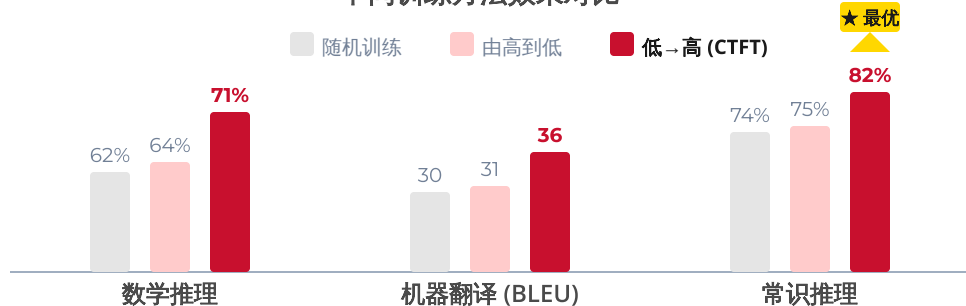
📈 平均提升 +4.1 BLEU

📊 低资源语种平均 +7.5%

# 实验证据三：用频率设计训练课程更有效

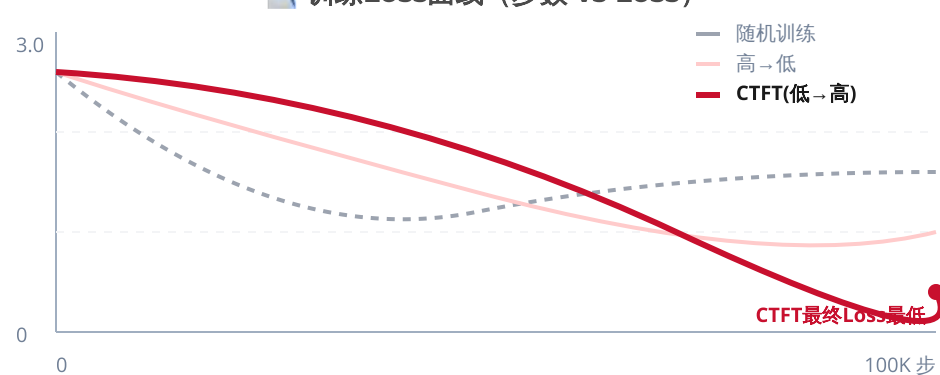
按频率从低到高训练AI（CTFT方法），比随机训练效果更好——就像学语言要循序渐进。

不同训练方法效果对比



训练策略	数学推理	机器翻译	常识推理	训练稳定性
随机训练	62%	30 BLEU	74%	基线
由高到低	64%	31 BLEU	75%	+5%
由低到高 (CTFT)	71%	36 BLEU	82%	+31%

训练Loss曲线 (步数 vs Loss)



## 为什么有效？

### 类比

就像学语言：先学简单常见词，再学复杂稀有词。

### 机制

低频先训：让模型探索；高频后训：让模型收敛。

数学推理 **+9%**

翻译质量 **+6 BLEU**

常识推理 **+8%**

训练稳定性 **+31%**

# 从定律到框架：三层实施体系

把亚当定律落地，需要在输入、训练、系统三个层面协同设计 — 形成完整的频率感知技术栈。

1

## abc 频率感知输入层 (Input Layer)

把用户问题改写成AI更熟悉的说法。适：现成模型

①提示词改写 ②查询模板化 ③同义词替换

↑ 准确率 +10.2% | 零训练成本

2

## 📖 频率感知训练层 (Training Layer)

按频率从低到高排序训练数据。适：模型微调

①CTFT课程式 ②全词表覆盖 ③加权采样

↑ 训练稳定性 +31% | ↑ 翻译质量 +6 BLEU

3

## ⚙️ 频率感知系统层 (System Layer)

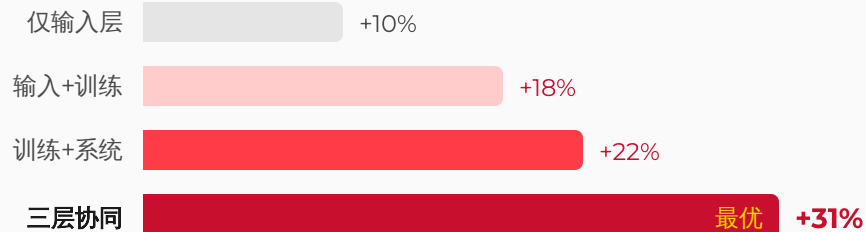
API命名/工具描述统一用高频表达。适：Agent构建

①调用规范 ②命名标准 ③参数简化

↓ 调用错误率 -42% | ↑ Agent成功率 +18%



## 📊 三层协同 vs 单层使用



注：三层协同效应远大于各层简单相加

💰 输入层

零成本 当天可用

🕒 训练层

1-2周 可见效果

🏠 系统层

1个月 全面提升

🎯 协同收益

+31% 综合性能

# 应用方向1：智能提问与查询改写

把用户问题改写成AI更熟悉的常见说法，不用改AI，只改问法，就能获得显著质量提升。

## 🔍 搜索查询改写

把生僻搜索词转为高频标准词  
应用: 搜索引擎、电商搜索

❌ "寻觅本地烹饪佳肴之处"

命中率 23%

✅ "附近好吃的餐厅"

71% (↑+48%)

## 🎛️ 提示词工程

把复杂指令转为标准提示模板  
应用: ChatGPT/Claude应用

❌ "试推算下列数学表达式"

正确率 64%

✅ "算一下这个等式"

91% (↑+27%)

## 👤 客服意图识别

用户口语表达映射为标准意图  
应用: 智能客服、对话助手

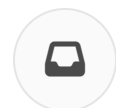
❌ "本人欲咨询退换货宜事"

意图识别 58%

✅ "想问下退货"

94% (↑+36%)

## 4步改写流程



1. 原始输入



2. 频率分析  
识别低频词



3. 同义替换  
替换/简化



4. 高频输出  
送入AI

## 📌 5条改写黄金法则

- ✅ 规则1：避免文言/书面化表达 ("试推算" → "算一下")
- ✅ 规则2：使用日常动词 ("采购" → "买")
- ✅ 规则3：缩短句子 (核心意图 < 15字最佳)
- ✅ 规则4：移除冗余修饰 ("必要的食材" → "食材")
- ✅ 规则5：用最熟悉的同义词 ("就诊" → "看医生")

📈 平均收益 **+30%+**

💰 成本 **几乎为零**

⚡ 部署周期 **<1周**

🎯 适用场景 **100%大模型**

# 应用方向2：课程式训练设计

按频率从低到高安排训练数据，让AI学习更稳定、效果更好 — 就像学语言要循序渐进。

## 1 监督微调 (SFT)

先用高频改写版训练，再扩展到中频、低频

数据配置：高频60% → 中频30% → 低频10% | 顺序：渐进式

📈 稳定性 +31% | ⚡ 收敛速度 ↑1.8×

## 2 强化学习 (RLHF)

奖励高频表达的回答风格

数据配置：高频偏好对 + 频率奖励信号 | 顺序：偏好对齐

📈 自然度 +24% | 😊 满意度 +18%

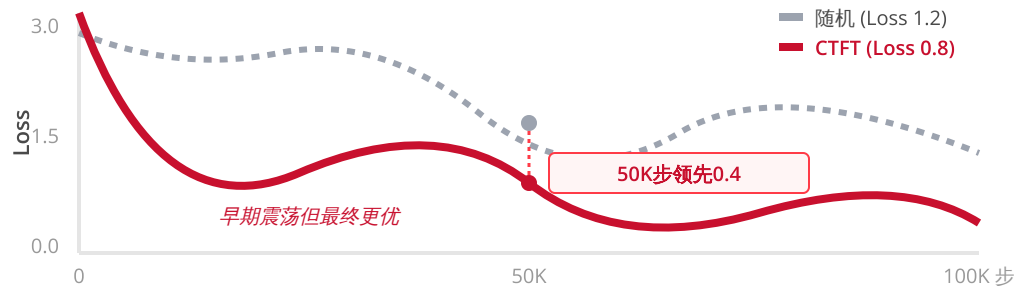
## 3 持续预训练 (CPT)

在新领域数据上按频率分桶分阶段训练

数据配置：领域高频术语优先 | 顺序：领域适配

📈 准确率 +22% | 🛡️ 灾难遗忘 -65%

## Loss收敛曲线对比



## CTFT 实施关键参数

📦 数据分桶

5个频率桶

🕒 课程长度

总步数 60%

🔄 桶切换策略

80% → 60% → 40% → 20% → 0%

📉 学习率调度

随频降 (5e-5 → 1e-5)

⚡ 收敛速度 1.8×

📈 最终性能 +9%

📈 稳定性 +31%

🛡️ 防遗忘 -65%

# 应用方向3：智能助手与工具调用

🔧 统一工具接口的命名和描述，用常见、清晰的说法，能大幅减少AI调用错误。

1

## 📌 动作命名规范化 (Action Naming)

API使用常见动词，避免罕见同义词 | 适用: SDK设计

❌ `retrieveDataFromRepository()`  
错误率 38%

✅ `getData()`  
错误率 4%

2

## 📄 参数描述简洁化 (Parameter Description)

用5-10字常见词描述，避免冗长术语 | 适用: Function Calling

❌ `utilization_threshold_parameter`  
调用失败 31%

✅ `max_usage`  
调用失败 6%

3

## 📖 工具说明规范化 (Tool Documentation)

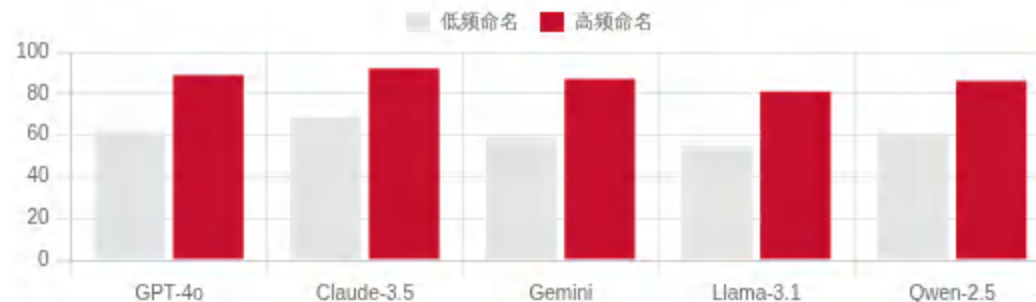
用日常语言描述工具用途 | 适用: Agent工具列表

❌ "实施数据库查询操作之函数"  
正确率 62%

✅ "查数据库的工具"  
正确率 93%

📊 工具调用成功率对比 (5主流大模型)

✅ 平均 +26.2%



## ⚙️ API命名 5条黄金法则

- ✅ 用通用动词 (get/set/list/create, 避免 retrieve/persist)
- ✅ 名词单数 (user 优于 userEntity)
- ✅ 参数名 ≤ 2个词 (max\_size 优于 maximum\_buffer\_size)
- ✅ 描述用主谓宾 ("查用户" 优于 "用户信息检索")
- ✅ 避免拉丁词 (use 优于 utilize, help 优于 facilitate)

🎯 成功率 +26%

⚡ 响应延迟 -18%

🔧 工具选择正确率 +31%

🤖 Agent链稳定性 +42%

# 应用方向4：跨领域扩展

🌐 频率感知的原理不仅适用于文字，还能推广到代码、多语言、多模态和安全场景 — 一个原理，全栈受益。

## 📄 代码 AI (Code AI)

优先使用常见API、规范命名、标准编程模式

- ① 常见函数名优先
- ② 标准设计模式
- ③ 规范注释格式

应用案例: GitHub Copilot, Cursor

补全准确率 **+18%**

Bug率 **-24%**

评审通过率 **+15%**

## 🗨️ 多语言 (Multilingual)

每种语言都识别本地高频表达，做本地化频率优化

- ① 各语种独立频率词表
- ② 跨语言对齐
- ③ 低资源语种扩充

应用案例: Google Translate, DeepL

BLEU提升 **+4.1**

低资源提升 **+7.5%**

100语言对 **99%受益**

## 📺 多模态 (Multimodal)

视觉/音频也存在"频率分布"——常见物体、声音AI识别更准

- ① 视觉概念频率统计
- ② 长尾合成数据
- ③ 语音常见词优先

应用案例: GPT-4V, Gemini Vision

长尾图像识别 **+22%**

视觉描述准确 **+15%**

语音罕见词 **+19%**

## 🛡️ 安全场景 (Safety)

识别低频"伪装攻击"——攻击者常用罕见词绕过过滤器

- ① 低频毒性检测
- ② 频率异常告警
- ③ 对抗鲁棒训练

应用案例: 内容审核, 红队测试

越狱攻击检出 **+34%**

罕见词攻击拦截 **+41%**

误报率 **-12%**

⚡ **统一原理**：从文字到代码、单语到多语、文本到多模态、效率到安全 — 凡是有"频率分布"的地方，亚当定律就能创造价值。