

# AI Context Engineering 领域若干重要论文

📅 2025年10月12日 ⌚ 1 分钟阅读

#Context Engineering

#Agent

#AI

这里将收集Context Engineering相关的重要文献，具体解读将在其他博客展开。

这里将收集Context Engineering相关的重要文献，具体解读将在其他博客展开，这里也会提供链接方便跳转。

## 1. A Survey of Context Engineering for Large Language Models

作者: L. Mei 等人

发表时间: 2025年7月

核心贡献:

该综述论文系统性地建立了上下文工程(Context Engineering)的理论框架，分析了1400多篇相关研究论文，首次提出了上下文工程的正式定义与分类法。论文确立了上下文工程作为超越简单提示词设计的系统性学科，涵盖了对大语言模型信息负载的系统优化([9+])。

创新点:

提出"上下文工程"正式定义：系统性优化大语言模型所接收信息的技术与方法

建立了包含上下文检索、处理和管理的完整技术框架

揭示了大语言模型理解能力与生成能力的不对称性关键问题

为该领域未来研究指明了方向，建立了评估挑战与标准化方法

更具体的解读: TBD

### 目录

### 文章信息

字数

阅读时间

发布时间

更新时间

### 标签

#Context Engineering

#Agent

## 2. Context Engineering: Enhancing Large Language Model Performance Through Comprehensive Contextual Management

**作者:** ResearchGate团队

**发表时间:** 2025年

**核心贡献:**

该论文将上下文工程定义为通过全面上下文管理提高大语言模型性能的系统化框架。它超越了传统的提示词工程，关注如何战略性地设计和优化提供给模型的上下文信息 ([28†])。

**创新点:**

- 提出综合上下文管理框架，作为提升LLM性能的系统化方法
- 超越简单提示词优化，关注上下文的全面设计与管理
- 为上下文工程提供了实践指导和理论基础
- 展示了上下文工程在实际应用中的性能提升效果

更具体的解读: TBD

## 3. Core Context Aware Transformers for Long

**作者:** ICML 2025入选研究团队

**发表时间:** 2025年

**核心贡献:**

该论文提出了核心上下文感知(Core Context Aware, CCA)注意力机制，用于大语言模型的高效长上下文建模。这是一个插拔式模块，专门针对处理和管理长上下文信息的设计 ([31†])。

**创新点:**

- 设计了高效的长上下文建模机制，解决了标准Transformer架构中的上下文长度限制
- 提出的CCA注意力机制可作为插拔组件，便于集成到现有模型

为处理超长文本序列提供了计算复杂度更低的解决方案  
在长文档理解任务中展现了优于传统注意力机制的性能

更具体的解读： TBD

## 上下文工程领域概述

上下文工程(Context Engineering)是人工智能领域，特别是大语言模型(LLM)应用中的前沿技术方向。它关注如何优化提供给AI模型的上下文信息，以最大化模型性能。

### 定义与发展:

上下文工程是对提示词工程的扩展，关注系统性优化信息负载  
发展历程从早期的简单提示词优化，到检索增强生成(RAG)，再到多智能体系统  
解决了从上下文窗口限制到信息相关性优化等一系列挑战

### 主要技术组件:

上下文检索与生成：包括链式思考(CoT)、树式思考(ToT)等方法  
上下文处理：长序列处理、自修正与适应、结构化信息整合  
上下文管理：记忆分层架构、上下文压缩技术

### 应用领域:

AI智能体开发：提升多轮对话一致性和任务执行效率  
检索增强生成：结合外部知识源增强模型输出  
长期对话系统：解决上下文窗口限制和信息衰减问题  
多智能体协作：协调多个专门智能体之间的上下文交换

## 未来研究方向

**理论框架完善：**建立更完善的形式化理论基础，发展任务特定的评估指标  
**技术突破：**探索更高效的上下文压缩与管理机制，改进长上下文处理算法  
**跨领域应用：**将上下文工程应用于多模态模型、强化学习等领域  
**社会责任：**研究上下文工程对模型安全、隐私和伦理影响

上下文工程作为AI领域的新兴重要方向，正在快速演进并重塑我们与  
大语言模型交互的方式。以上三篇论文代表了该领域从理论建立  
到技术实现的全链条创新，为后续研究和应用提供了重要参考。

分  
享  
这  
篇  
文  
章



## 相关文章推荐

Agent 相关  
课程收集

### AI Agents for Beginners - From MS

微软发布，11 节  
课，教授开始构  
建人工智能代理  
所需了解的一切  
知识

home page:

[https://microsoft.github.io/ai-agents-for-](https://microsoft.github.io/ai-agents-for-beginners/)

[beginners/](https://microsoft.github.io/ai-agents-for-beginners/)      github:

<https://github.com/microsoft/ai-agents-for-beginners>

old one:

<https://learn.microsoft.com/en-us/shows/ai-agents-for-beginners/>

<https://learn.microsoft.com/en-us/shows/ai-agents-for-beginners/what-are-ai-agents>

## Agent Lightning

### 介绍

微软开源的 **Agent Lightning** 项目，它的核心价值在于为开发者和研究者提供了一个强大的工具，用于**训练和优化 AI Agent（智能代理）**，特别是**几乎不需要修改现有 Agent 代码**就能实现显著的性能提升。

这个项目有以下重要作用：

**零代码/低代码训练 AI Agent (核心价值):**

**最大亮点:** 它允许你使用**强化学习 (Reinforcement Learning, RL)** 等高级优化算法来训练你现有的 AI Agent, 而**几乎不需要修改你的 Agent 业务逻辑代码**。这意味着你可以保留你用 LangChain, AutoGen, CrewAI, OpenAI SDK 等框架 (甚至裸 Python) 编写的 Agent 逻辑, 然后让 Agent Lightning 负责优化它的决策过程。

**解决痛点:** 传统上, 将 RL 等技术应用到现有 Agent 框架中需要大量的工程改造和集成工作。Agent Lightning 极大地简化了这个过程。

**强大的优化能力:**

**算法支持:** 内置支持**强化学习 (VERL)** 作为核心优化算法, 并明确提到支持**自动提示优化 (Automatic**

提供训练基础设施：

## Context Engineering

Context Engineering 是...

在调用 LLM、XX 等组件时，其可靠性可以得到显著提升。

**广泛的兼容性和灵活性：**

**框架无关：** 明确支持所有主流 Agent 框架 (LangChain, OpenAI Agent SDK, AutoGen, CrewAI) 以及纯 Python 实现的 Agent。你可以“即插即用”。

**多 Agent 系统优化：** 可以在包含多个 Agent 的复杂系统中，**选择性地优化其中一个或几个特定的 Agent**，而不是整个系统，提供了更精细的控制。