

ITEM #249 - Structural Tensions Between RAG, Continual Learning, and Backprop-Based Generalization:
Why These Conflicts Are Fundamental — and How to Engineer Around Them

Conversation: LLM DBM 统一模型

20260126

Authors: Sizhe Tan & GPT-Obot

DBM-COT ITEM #249 (EN) Structural Tensions Between RAG, Continual Learning, and Backprop-Based Generalization

Why These Conflicts Are Fundamental — and How to Engineer Around Them

Abstract

Retrieval-Augmented Generation (RAG) and Continual Learning are widely regarded as necessary extensions to Large Language Models.

However, both mechanisms expose fundamental structural tensions with backpropagation-based unified generalization.

This document argues that these tensions are not implementation defects, but arise from irreconcilable objectives:

- RAG introduces *local, conditional, time-sensitive truth* at inference time.
- Continual learning introduces *temporal partitioning of knowledge* across learning phases.

- Backprop-based generalization seeks to *eliminate conditionality* by compressing experience into a single unified function Φ .

We further provide engineering guidance for deploying RAG and Continual Learning responsibly, without attempting to force them back into a unified Φ .

1. Problem Statement

Modern LLM systems increasingly rely on:

1. RAG to correct hallucinations and inject up-to-date or domain-specific facts.
2. Continual Learning to incorporate new knowledge over time.

Yet both approaches exhibit persistent issues:

- RAG behaves like an external patch rather than a stable fix.
- Continual learning leads to catastrophic forgetting or uncontrolled drift.

We claim these issues arise because both mechanisms conflict structurally with the goals of backprop-based unified generalization.

2. RAG vs Backprop-Based Generalization

2.1 The Role of RAG

RAG functions as:

- A runtime, conditional correction operator.
- Injecting *context-specific truth* during Phase-2 decision making.
- Operating outside the learned Φ .

In DBM terms, RAG is a local $\Phi_{lo}c_{al}$ adjudicator, not part of Φ itself.

2.2 The Role of Backpropagation

Backprop-based learning seeks to:

- Compress diverse experiences into a single, stable, reusable Φ .
- Average away contextual variance.
- Maximize global generalization.

Its implicit objective is:

To eliminate conditionality in favor of statistical regularity.

2.3 Structural Conflict

RAG Requires	Backprop Requires
Local truth	Global averaging
Contextual validity	Context erasure
Replaceable knowledge	Frozen weights
Runtime adjudication	Training-time consolidation

Attempting to absorb RAG outputs back into training creates an information-theoretic loss loop, where conditional truth is progressively flattened.

This conflict is structural, not accidental.

3. Continual Learning vs Backprop-Based Generalization

3.1 What Continual Learning Demands

Continual learning requires that:

- New knowledge does not overwrite old knowledge.

- Temporal order matters.
- Multiple “versions of truth” may coexist.

This implies Φ must be partitionable in time.

3.2 Why Unified Φ Cannot Support This

Backpropagation assumes:

- Shared parameters.
- Gradient superposition.
- A single optimization objective.

Thus:

New learning inevitably reweights old knowledge, rather than coexisting with it.

“Catastrophic forgetting” is not a bug — it is the expected outcome of a temporally unified Φ .

4. Unifying the Two Conflicts

RAG and Continual Learning expose the same structural limitation: Any mechanism that introduces conditional, local, or time-dependent truth is incompatible with a single unified Φ .

- RAG = spatial conditionality.

- Continual learning = temporal conditionality.

Backprop-based generalization eliminates conditionality across all dimensions.

5. Why DBM-Style Architectures Avoid This by Design

DBM systems:

- Maintain multiple Φ operators.
- Allow Φ to be:
 - local,
 - frozen,
 - replaced,
 - versioned.

Thus:

- RAG becomes a legitimate Φ plug-in.
- Continual learning becomes Φ accretion, not Φ rewriting.

No re-burning into a single weight space is required.

6. Engineering Recommendations (If RAG Must Be Used)

Recommendation 1: Treat RAG as a Runtime Authority, Not Training Data

- Do not indiscriminately feed RAG-corrected outputs back into training.
- Preserve RAG sources as external adjudicators.

Recommendation 2: Version and Scope RAG Explicitly

- Attach validity windows, domains, and confidence levels.
- Prevent silent global absorption.

Recommendation 3: Prefer “RAG-as- Φ ” Over “RAG-asData”

- Model RAG as a decision operator, not a data augmenter.

7. Engineering Recommendations (If Continual Learning Must Be Used) Recommendation 4: Freeze Core Φ , Add Peripheral Φ

- Avoid rewriting central weights.
- Add side modules, adapters, or rule layers.

Recommendation 5: Preserve Temporal Boundaries

- Maintain explicit versioning of learned components.
- Allow coexistence rather than forced merging.

Recommendation 6: Accept Partial Non-Generalization

- Not all knowledge should generalize.
- Some knowledge is *meant* to remain local or temporal.

8. Implications for Unified LLM–DBM Models

In a unified framework:

- LLM should handle:
 - high-compression,
 - weakly conditional Φ .
- DBM should handle:
 - strongly conditional,

- time-evolving Φ .

Attempting to collapse both into a single Φ degrades both.

9. Conclusion

RAG and Continual Learning reveal the boundary of backprop-based generalization.

They are not flaws to be engineered away, but signals that Φ must be pluralized.

DBM-style architectures resolve these tensions structurally, not statistically.

DBM-COT ITEM #249 (中文) RAG、连续学习与反向泛化烧结之间的结构性矛盾为何这些矛盾不可消解，以及工程上应如何应对

摘要

RAG (检索增强生成) 与连续学习被视为弥补 LLM 缺陷的必要机制，但二者同时暴露出与基于反向传播的统一泛化之间的根本性结构矛盾。

本文指出：

- RAG 引入的是运行期的、条件化、局部真理；
- 连续学习引入的是时间维度上的知识分区；
- 而反向泛化烧结的目标，是消除一切条件性，压缩为单一 Φ 。

这些冲突不是工程不足，而是目标函数层面的不可调和性。

1. 问题背景

当前 LLM 系统普遍依赖：

1. RAG 纠正幻觉、补充新知识；

2. 连续学习 应对知识更新。

但现实表现是：

- RAG 永远像外挂；

- 连续学习不可避免地产生灾难性遗忘。

原因并不在 “还没优化好” 。

2. RAG 与反向泛化烧结的矛盾

2.1 RAG 的本质角色

RAG 本质上是：

- Phase-2 中的运行期裁决算子；

- 引入特定时间、上下文下成立的真理；

- 并不属于模型内部 Φ 。

2.2 反向泛化烧结的目标

反向传播的目标是：

- 将大量经验压缩为稳定、统一、可复用的 Φ ；

- 抹平条件差异，换取统计泛化。

2.3 不可调和的冲突

RAG 需要	Backprop 需要
条件真理	平均真理
局部有效	全局稳定
可替换	不可扰动
运行期裁决	训练期固化

把 RAG 结果“再训练回去”，只会损耗信息。

3. 连续学习与反向泛化烧结的矛盾

3.1 连续学习的真实诉求

- 新知识不覆盖旧知识；
- 时间顺序必须被尊重；
- 多版本真理并存。

3.2 统一 Φ 的结构限制

反向传播假设：

- 参数共享；
- 梯度叠加；
- 单一优化目标。

因此，新知识只能重加权旧知识，而非并存。

所谓“灾难性遗忘”，本质是：统一 Φ 在时间维度上不可分解。

4. 两类矛盾的统一理解

RAG 与连续学习指向同一个结论：

凡是引入条件化、局部化、时间化真理的机制，都与统一
 Φ 存在根本张力。

5. DBM 为何天然规避这一问题

DBM 的 Φ 具备：

- 多 Φ ；
- 可冻结；
- 可替换；
- 可并存。

因此：

- RAG 是合法 Φ 插件；
- 连续学习是 Φ 的累加，而非重写。

6. 工程建议 (RAG 不得不用时)

建议 1：把 RAG 当作裁决器，而非训练数据源不要轻易把 RAG 纠正结果回灌模型。

建议 2：显式管理 RAG 的作用域与时效避免被“默默吞并”。

建议 3：RAG = Φ 插件，而非数据增强

7. 工程建议 (连续学习不得不用时)

建议 4：冻结核心 Φ ，新增外围 Φ 建议

5：保留时间版本，而非强行合并建议

6：接受“不可泛化”的知识存在

8. 对 Unified LLM-DBM 模型的启示

- LLM 负责高压缩、弱条件 Φ ;
- DBM 承载强条件、可演化 Φ 。

强行统一只会两败俱伤。

9. 结论

RAG 与连续学习揭示的不是工程缺陷，而是统一泛化的边界。

解决之道不是更激进的烧结，而是 Φ 的多元共存。

•