神经网络等价解耦与“三层分治”（MDQ 网络 × 索引泛函 × OOV 内存库）落地方案

- 作者：GaoZheng

- 日期：2025-09-26
- 版本：v1.0.0

摘要

提出 MDQ 机制稳定离散 LLM/策略管道：支持小单元交互与统一版本控制，缓解长序列采样的非平稳与暴露偏差。结合指令设计与记忆扩展策略，给出训练/推理一体化的实现路线与评估指标。

- **目标**：把“产能”与“治理”彻底解耦。上层神经网络只做**微分控制**（MDQ），不直接产文；中层**索引泛函**算分与路由；底层**非神经网络内存库**解决 OOV 与显式知识。
- **等价解耦**两类：
  i. **非NN实现 ↔ NN实现**的**行为等价**（接口不变、KPI 边界一致、可回放）；
  ii. **NN 基于 MDQ 的控制面**与**索引/内存库的数据面**的**架构解耦**（可热插拔、可版本共存）。
- **收益**：训练预算只投在“微分与索引”，推理端**成本可控、可审计、可回滚**，对 OOV 免疫。

---

1. 设计原则（四条红线）

1. **接口等价**：对外统一 Operator API（Lex-KAT 算子族）；实现可换，但输入输出契约不变。
2. **行为可证**：KAT-tests 必过，JSONL 事件可 100% 回放；漂移受 KPI 约束。
3. **热插拔/可回滚**：一切更新以 **MDQ-pkg** 原子落地；双缓冲 + 金丝雀 + 自动回滚。
4. **资源闭环**：L_h, L_p 作为一等公民（Flex-Attn），与吞吐/成本进入同一 ROI 账本。

---

2. 架构总览（控制面/数据面/知识面三层）

- **控制面（NN 可选）**：**MDQ 网络**（Micro-Differential-Quantum Network）输出最小增量 Δ：阈值、长度、权重、门控。
- **数据面（无 NN 也可）**：**索引泛函网络** $\mathcal{I}(\mathrm{seg}) = \sum_k w_k \varphi_k$，特征含 IDF、别名、正则、字符 n-gram、SimHash/MinHash 等。
- **知识面（非 NN）**：**OOV 等价内存数据库**（EKB）：Trie/AC、加权 FST、别名/译名、规则词表；文件→内存→热缓存三级。

> 统一编排由 **Lex-KAT 作用幺半群**实现：左/右乘、投影、tests、闭包（命中即停）等算子序列。

---

3. 等价解耦 #1：非NN实现 ↔ NN实现（功能等价）

目标：同一 API，不同实现；可并行灰度、可替换、可回滚。

3.1 接口契约（Operator API）

- $Apply(operators : [G_i], state) -> state'$；$G_i \in L, R, \Pi, Head, Test, Cl, D...$
- 约束：幂等（投影、tests）、闭包（CI）扩张/单调、长度边界、合规硬闸。

3.2 等价定义（三级）

- **强等价**：逐步事件日志一致（hit 词、长度、tests 结果完全一致）。
- **弱等价**：KPI 等价（BERTScore/ROUGE、word_noncompliance、P95/QPS 在阈内）。
- **统计等价**：分布距离（KL/KS）与漂移监控在阈内。

3.3 验收与灰度

- 影子流量对打（Shadow-Diff Harness）→ 金丝雀 10–20% → 全量；
- 失败即按 `MDQ-pkg.rollback` 自动回滚；
- 合同化阈值：`word_noncompliance` ↓≥30%、术语召回 `+8–15pp`、P95/QPS 达标。

---

4. 等价解耦 #2：MDQ 网络 × 索引泛函 × OOV 内存库（架构分治）

4.1 MDQ 网络（NN）

- **职责**：只输出微分增量 Δ（$\tau, \lambda_{\text{lex}}, L_h, L_p$, 权重等），**不产文**。
- **训练**：策略梯度/占用测度 + 非交换惩罚（$[G_i, G_j]$ 抑制同时上调），小型可蒸馏。
- **输出协议**：序列化为 **MDQ-pkg**，走治理链路（审计/回放/回滚）。

4.2 索引泛函网络（可纯规则）

- **形式**：$\mathcal{I}(\text{seg}) = \sum_k w_k \varphi_k$。
- **特征**：IDF、别名/译名、n-gram、正则、SimHash/MinHash、域词概率、位置特征。
- **更新**：线性积累（MDQ 累加权重），幂等合并；无 NN 亦可运行。

4.3 OOV 等价内存库（非 NN）

- **结构**：反向 Trie/AC、加权 FST、别名图、音译/形近表、拉丁/数字正则。
- **策略**：OOV 命中优先走“等价替换/近似命中”→ 通过 tests → 进入闭包。
- **特性**：文件热更→内存→缓存，秒级生效；可 TTL 与热度淘汰。

> 三层之间只通过 **MDQ-pkg** 与 **事件日志**通讯，彼此可独立版本演进。

---

5. 数据与版本治理（统一标准）

- **包结构（MDQ-pkg）**：`name/semver/scope/atoms/tests/rollback/kpi`。
- **兼容矩阵**：`[Lex-KAT 算子版本 × 索引 schema × Gate 策略] → OK/警告/拒绝`。
- **流程**：双缓冲挂载 → 影子评估（Eval-w/o-Top-p）→ 金丝雀 → 全量 → ledger 记账。
- **回放**：JSONL 事件流 + 变更 ledger，支持“时间旅行”对账。

---

6. 关键指标（SLA/KPI）

- **质量**：术语/要点召回↑；`word_noncompliance` ↓≥30%；Eval-w/o-Top-p 不劣化。
- **成本**：P95 延迟/QPS/显存在阈内；平均 L_p 与 L_h 受控。

- **治理**：回放成功率 100%；回滚成功率 100%；MDQ-pkg 失败率 < 阈。
- **稳态**：训练收敛步数 ↓≥15%，方差 ↓≥20%。

---

7. 风险与补丁

- **长词偏置**：L_p 上限 + 长度成本 + 语义门控 + IDF/二字降权。
- **索引污染/OOV 中毒**：白/黑名单 tests 前置；MDQ-pkg 可逆；EKB 读写隔离。
- **非交换冲突**：MDQ 量化加入 commutator 惩罚；分步上调。
- **API 碎片**：以适配层固化 Operator API；供应商差异收敛到包层。

---

8. 两周落地清单（Minimal Plan）

- **Week 1**：固化 Operator API；上线非 NN 版（索引泛函 + OOV 内存库）；接 KAT-tests 与 JSONL 回放；打通 MDQ-pkg 流程。
- **Week 2**：接入小型 MDQ 网络（只调 $\tau, \lambda_{\text{lex}}, L_h, L_p$）；影子流量对打 → 金丝雀；完成回滚演练与仪表盘。

---

9. 一句话总结

把“神经网络”从“产文黑箱”里解耦出来，只负责**微分控制**；把“知识/OOV/路由”下沉到**索引泛函 + 非NN内存库**；统一用 **Lex-KAT 算子**粘合全链路、用 **MDQ-pkg**治理变更。这样既能**等价替换实现**，又能**架构分治演进**：成本更低、风险更小、上线更快、审计更强。
