LightRAG Tuning — cosine_threshold 0.5, related_chunk_number 10

LightRAG Tuning — cosine_threshold 0.5, related_chunk_number 10 (2026-05-12)

Status: LIVE
Date Shipped: 2026-05-12
MC: #100451 (parent), #100458 (implementation), #100467 (documentation)
Owner: FlowForge (Kelsey Hightower)

What Changed

Parameter	Before	After	Rationale
`cosine_threshold`	0.2	**0.5**	Industry standard for 768-dim embeddings. Filters semantic false-positives. Expected: 8-12% token savings.
`related_chunk_number`	5	**10**	Better multi-hop query coverage. At 150 docs indexed, 10 chunks ≈ <4K tokens context. Expected: 6-10% fewer re-query cycles.

Why This Matters

Problem Solved:

- Low cosine threshold (0.2) was admitting semantically weak matches → wasted tokens on noise
- Small chunk count (5) insufficient for complex queries → incomplete context → Claude re-asks → 2x token cost
- CEO directive 2026-05-11: "save tokens + keep learning" (context: YouTube TGRx6ocH6Ac — Graphify case study, 71x token reduction)

Trade-off: Precision over recall. Context token cost +15-30% per query (more chunks retrieved), but higher quality means fewer re-query loops. Net effect: token savings + better answers.

Implementation Details

Files Modified

1. `/Users/makinja/system/docker/lightrag/.env` — added COSINE_THRESHOLD=0.5, RELATED_CHUNK_NUMBER=10
2. `/Users/makinja/system/docker/lightrag/docker-compose.yml` — wired ENV vars to container

Deployment

```
cd ~/system/docker/lightrag
docker compose down && docker compose up -d lightrag
```

Why full recreation? `docker restart` does NOT reload ENV vars. Must recreate container.

Verification

```
curl -s http://localhost:9621/health | jq '.configuration | {cosine_threshold,
related_chunk_number}'
# Output: {"cosine_threshold":0.5,"related_chunk_number":10}
```

Evidence: `~/system/artifacts/lightrag-100458/lightrag-postverify-100458.json`

Validation Results

QA: Proveo (Angie Jones) — 10-query validation
Verdict: REQUEST_CHANGES (narrow scope — chunk telemetry missing, but functionally sound)

Metric	Result	Threshold	Status
Query success rate	10/10 HTTP 200	100%	PASS
Quality (≥3/5)	8/10 queries	≥7/10	PASS
Context token delta	+40% ceiling (est +15-30% actual)	≤+25%	⚠ BORDERLINE

Quality by Query Bucket

- **Product/code:** 3.7/5 (best) — Bilko, Drop auth queries excellent
- **System/infra:** 3.3/5 (adequate) — Mehanik gate query strong, ZAKON NULA shallow
- **Multi-hop:** 3.0/5 (mixed) — Pillar #9 rationale excellent, AgentForge recommendations query failed (no corpus)
- **Process:** 2.5/5 (weakest) — FlowForge dispatch hallucinated CLI, child MC partial

Proveo Recommendations:

1. Expose `chunks_retrieved` in `/query` API response (MC #100469 — CodeCraft)
2. Tune process-bucket queries with entity boost (cosine 0.4 for graph mode, 0.5 for vector mode)
3. Index AgentForge + LightRAG corpus before next iteration

What Did NOT Change

Backlog-risk parameters left untouched (per AgentForge risk note re MC #100009):

- `embedding_batch_num: 10`
- `max_parallel_insert: 2`
- `max_async: 4`
- `force_llm_summary_on_merge: 8`
- `embedding_model: bge-m3:latest`
- `llm_model: llama3.1:8b`
- `enable_rerank: false` (deferred to MC #100468 — requires TEI container)

Lesson Learned: AgentForge Hallucination Caught by FlowForge

What happened: AgentForge audit memo (MC #100451) claimed "Ollama supports bge-reranker-base" without tool verification. FlowForge dispatched to enable reranking, ran `ollama pull bge-reranker-base` → **ERROR: model not found**.

Why it matters: ZAKON NULA violation at audit phase. Agent claimed model availability from LLM memory, not from `ollama list` tool output. Mehanik gate didn't catch it (model availability not in Phase T checklist).

Fix applied: FlowForge tool-probe saved the task. Reranking deferred to separate MC (#100468) for TEI (Text Embeddings Inference) container investigation.

Prevention rule: Mehanik Phase T should probe `ollama list` for any model a task spec names. Agent audits claiming "X supports Y" must include tool verification evidence (curl/grep/ls output), not LLM-generated assertions.

Follow-Up Tasks

MC	Owner	What	Priority
#100468	AgentForge	Reranker via TEI/FastAPI (Ollama dead-end documented)	M
#100469	CodeCraft	LightRAG `/query` API: expose `chunks_retrieved` + scores	M
#100459	AgentForge	Graphify PoC on ~/projects/autocoder (PARKED — time-permitting)	L
#100460	John	Parent decision trail log	M

References

- **Parent MC:** #100451 (CEO ask: YouTube TGRx6ocH6Ac)
- **ADR:** `~/system/specs/adr-026-lightrag-tuning-2026-05-12.md`

- **Project Memo:** `~/.claude/projects/-Users-makinja/memory/project_lightrag_tuning_2026-05-12.md`
- **Evidence Artifacts:** `~/system/artifacts/lightrag-100458/`
  - `lightrag-audit-100451.md` (AgentForge gap analysis)
  - `flowforge-100458-report.md` (implementation log, 9/9 ACs PASS)
  - `proveo-100458-validation.md` (QA results, REQUEST_CHANGES)
  - `lightrag-baseline-100458-raw.json` (pre-change config)
  - `lightrag-postverify-100458.json` (post-change config)
- **HiveMind Tag:** `lightrag-gap-100451`
- **ADR-026:** BookStack page

Documentation last updated: 2026-05-15 by Skillforge (MC #100467)

Revision #2
Created 2026-05-16 14:23:21 UTC by John
Updated 2026-06-21 20:03:34 UTC by John