



Morphik Core

Note: For our hosted service: <https://www.morphik.ai> (<https://www.morphik.ai>). We also deploy our Morphik on prem or VPC, happy to chat: <https://cal.com/adityavardhan-agrawal-x6jyhq/30min> (<https://cal.com/adityavardhan-agrawal-x6jyhq/30min>).



[_\(https://github.com/morphik-org/morphik-core/tree/main?tab=License-1-ov-file#readme\)](https://github.com/morphik-org/morphik-core/tree/main?tab=License-1-ov-file#readme)



[_\(https://pypi.org/project/morphik/\)](https://pypi.org/project/morphik/)



[_\(https://discord.gg/BwMtv3Zaju\)](https://discord.gg/BwMtv3Zaju)

What is Morphik?

Morphik is an open-source database designed for AI applications that simplifies working with unstructured data. It provides advanced RAG (Retrieval Augmented Generation) capabilities with multi-modal support, knowledge graphs, and intuitive APIs.

Built for scale and performance, Morphik can handle millions of documents while maintaining fast retrieval times. Whether you're prototyping a new AI application or deploying production-grade systems, Morphik provides the infrastructure you need.

Features

- ☐ **First-class Support for Unstructured Data**
 - Ingest ANY file format (PDFs, videos, text) with intelligent parsing
 - Advanced retrieval with ColPali multi-modal embeddings
 - Automatic document chunking and embedding
- ☐ **Knowledge Graph Integration**
 - Extract entities and relationships automatically
 - Graph-enhanced retrieval for more relevant results
 - Explore document connections visually
- ☐ **Advanced RAG Capabilities**
 - Multi-stage retrieval with vector search and reranking
 - Fine-tuned similarity thresholds
 - Detailed metadata filtering
- ☐ **Natural Language Rules Engine**
 - Define schema-like rules for unstructured data
 - Extract structured metadata during ingestion
 - Transform documents with natural language instructions
- ☐ **Persistent KV-caching**
 - Pre-process and "freeze" document states
 - Reduce compute costs and response times
 - Cache selective document subsets
- ☐ **MCP Support**
 - Model Context Protocol integration
 - Easy knowledge sharing with AI systems
- ☐ **Extensible Architecture**
 - Support for custom parsers and embedding models
 - Multiple storage backends (S3, local)
 - Vector store integration with PostgreSQL/pgvector

Quick Start

Installation

```
# Clone the repository
git clone https://github.com/morphik-org/morphik-core.git
cd morphik-core

# Create a virtual environment
python3.12 -m venv .venv
source .venv/bin/activate # Linux/macOS

# Install dependencies
pip install -r requirements.txt

# Configure and start the server
python quick_setup.py
python start_server.py
```

Using the Python SDK

```

from morphik import Morphik

# Connect to Morphik server
db = Morphik("morphik://localhost:8000")

# Ingest a document
doc = db.ingest_text("This is a sample document about AI technology.",
                    metadata={"category": "tech", "author": "Morphik"})

# Ingest a file (PDF, DOCX, video, etc.)
doc = db.ingest_file("path/to/document.pdf",
                    metadata={"category": "research"})

# Use ColPali for multi-modal documents (PDFs with images, charts, etc.)
doc = db.ingest_file("path/to/report_with_charts.pdf", use_colpali=True)

# Apply natural language rules during ingestion
rules = [
    {"type": "metadata_extraction", "schema": {"title": "string", "author": "string"}},
    {"type": "natural_language", "prompt": "Remove all personally identifiable information"}
]
doc = db.ingest_file("path/to/document.pdf", rules=rules)

# Retrieve relevant document chunks
chunks = db.retrieve_chunks("What are the latest AI advancements?",
                            filters={"category": "tech"},
                            k=5)

# Generate a completion with context
response = db.query("Explain the benefits of knowledge graphs in AI applications",
                    filters={"category": "research"})
print(response.completion)

# Create and use a knowledge graph
db.create_graph("tech_graph", filters={"category": "tech"})
response = db.query("How does AI relate to cloud computing?",
                    graph_name="tech_graph",
                    hop_depth=2)

```

Batch Operations

```

# Ingest multiple files
docs = db.ingest_files(
    ["doc1.pdf", "doc2.pdf"],
    metadata={"category": "research"},
    parallel=True
)

# Ingest all PDFs in a directory
docs = db.ingest_directory(
    "data/documents",
    recursive=True,
    pattern="*.pdf"
)

# Batch retrieve documents
docs = db.batch_get_documents(["doc_id1", "doc_id2"])

```

Multi-modal Retrieval (ColPali)

```

# Ingest a PDF with charts and images
db.ingest_file("report_with_charts.pdf", use_colpali=True)

# Retrieve relevant chunks, including images
chunks = db.retrieve_chunks(
    "Show me the Q2 revenue chart",
    use_colpali=True,
    k=3
)

# Process retrieved images
for chunk in chunks:
    if hasattr(chunk.content, 'show'): # If it's an image
        chunk.content.show()
    else:
        print(chunk.content)

```

Why Choose Morphik?

Feature	Morphik	Traditional Vector DBs	Document DBs	LLM Frameworks
Multi-modal Support	<input type="checkbox"/> Advanced ColPali embedding for text + images	<input type="checkbox"/> or Limited	<input type="checkbox"/>	<input type="checkbox"/>
Knowledge Graphs	<input type="checkbox"/> Automated extraction & enhanced retrieval	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Feature	Morphik	Traditional Vector DBs	Document DBs	LLM Frameworks
Rules Engine	<input type="checkbox"/> Natural language rules & schema definition	<input type="checkbox"/>	<input type="checkbox"/>	Limited
Caching	<input type="checkbox"/> Persistent KV-caching with selective updates	<input type="checkbox"/>	<input type="checkbox"/>	Limited
Scalability	<input type="checkbox"/> Millions of documents with PostgreSQL	<input type="checkbox"/>	<input type="checkbox"/>	Limited
Video Content	<input type="checkbox"/> Native video parsing & transcription	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deployment Options	<input type="checkbox"/> Self-hosted, cloud, or hybrid	Varies	Varies	Limited
Open Source	<input type="checkbox"/> MIT License	Varies	Varies	Varies
API & SDK	<input type="checkbox"/> Clean Python SDK & RESTful API	Varies	Varies	Varies

Key Advantages

- **ColPali Multi-modal Embeddings:** Process and retrieve from documents based on both textual and visual content, maintaining the visual context that other systems miss.
- **Cache Augmented Retrieval:** Pre-process and "freeze" document states to reduce compute costs by up to 80% and drastically improve response times.
- **Schema-like Rules for Unstructured Data:** Define rules to extract consistent metadata from unstructured content, bringing database-like queryability to any document format.
- **Enterprise-grade Scalability:** Built on proven PostgreSQL database technology that can scale to millions of documents while maintaining sub-second retrieval times.

Documentation

For comprehensive documentation:

- [Installation Guide \(https://docs.morphik.ai/getting-started\)](https://docs.morphik.ai/getting-started)
- [Core Concepts \(https://docs.morphik.ai/concepts/naive-rag\)](https://docs.morphik.ai/concepts/naive-rag)
- [Python SDK \(https://docs.morphik.ai/python-sdk/morphik\)](https://docs.morphik.ai/python-sdk/morphik)
- [API Reference \(https://docs.morphik.ai/api-reference/health-check\)](https://docs.morphik.ai/api-reference/health-check)

License

This project is licensed under the MIT License - see the [LICENSE \(LICENSE\)](#) file for details.

Community

- [Discord \(https://discord.gg/BwMtv3Zaju\)](https://discord.gg/BwMtv3Zaju) - Join our community
- [GitHub \(https://github.com/morphik-org/morphik-core\)](https://github.com/morphik-org/morphik-core) - Contribute to development

