

SKILLS RADAR / LEARNING FIELD GUIDE

A field guide to AI competencies.

*What to learn.
How far to go.*

An evidence-led reading companion to the learning map.

01	Models & learning	10 CONCEPTS
02	Knowledge & retrieval	09 CONCEPTS
03	Agents & tools	08 CONCEPTS
04	Quality & evaluation	06 CONCEPTS
05	Safety & security	05 CONCEPTS
06	Platform & performance	09 CONCEPTS

47 concepts. 94 profile assessments. 56 classification axes.

First editorial assessment edition / 1 October 2026. Recommendations interpret the cited research for two fixed work profiles. The underlying research corpus has not been comprehensively refreshed.

READ THIS FIRST

A guide, not a verdict.

A useful learning map should tell you what a concept means, why it matters and where its usefulness stops. It should also make the evidence easy to inspect.

01 / BROWSE THE LANDSCAPE

Start with the index. The six areas are provisional navigation groups, not mutually exclusive or exhaustive categories. A concept can cross several boundaries.

02 / READ ONE CONCEPT AT A TIME

Each concept has a research dossier and an assessment page. The dossier preserves the case for inclusion, limitations, alternatives, taxonomies and source locators. The assessment gives a bounded learning plan for each profile.

03 / TREAT ADVICE AS AN EDITORIAL INFERENCE

All 47 concepts have an assessment for both profiles. Sources support the technical reasoning; they do not directly prescribe a curriculum. Each assessment explains its evidence basis and confidence.

04 / KEEP THE COMPARISON HONEST

This is the first real editorial baseline. Future editions can compare learning outcomes while holding the profile and task fixed. No historical movement or prior recommendation is claimed here.

05 / RETURN TO THE SOURCES

Reference numbers link to the bibliography. Source titles are clickable. A taxonomy is a useful slice of a subject, not necessarily a complete or universal classification.

Editorial boundaries: 40 concepts belong to the working research selection, five are reserves and two remain under research. These evidence-selection labels are separate from learning advice. This edition adds profile-based editorial assessments to the existing English atlas. It does not comprehensively refresh or independently revalidate the research corpus. Evidence methods and limitations remain in the web reading view.

CONCEPT DIRECTORY / 1 OF 2

Find your next question.

Click a title to open its research dossier. Its profile assessments follow on the next page. Page numbers are not rankings.

01 / MODELS & LEARNING

Direct Preference Optimization	7
In-Context Learning	9
Large Language Models	11
Long-Context Modeling	13
LoRA / QLoRA	15
Reasoning Models	17
Structured LLM Outputs	19
Supervised Fine-Tuning (SFT)	21
Test-Time Compute Scaling	23
Training Data Curation	25

02 / KNOWLEDGE & RETRIEVAL

Agentic RAG	27
Dense Retrieval	29
Embedding Models	31
Hybrid Search	33
Multimodal RAG	35
Retrieval-Augmented Generation	37
Search Re-Ranking	39
Vector Databases	41
Vector Indexing	43

03 / AGENTS & TOOLS

Agent Memory Systems	45	Code Execution Agents	53
Agentic Planning & Task Decomposition	47	FastAPI	55
AI Agent Design	49	Model Context Protocol	57
AI-Assisted Development	51	Multi-Agent Coordination Patterns	59

Reading the framework: p. 5 / Profiles: p. 6
First assessment baseline: 1 October 2026 / Reference library: p. 101

CONCEPT DIRECTORY / 2 OF 2

Find your next question.

Click a title to open its research dossier. Its profile assessments follow on the next page. Page numbers are not rankings.

04 / QUALITY & EVALUATION

Agent Evaluation	61
Evaluation Data Engineering	63
Hallucination Detection	65
LLM Testing	67
LLM-as-Judge	69
RAG Evaluation	71

05 / SAFETY & SECURITY

Agent Threat Modeling (MAESTRO)	73
AI Data Security	75
AI Guardrails	77
AI Red Teaming	79
Secure RAG	81

06 / PLATFORM & PERFORMANCE

Amazon Bedrock	83	LLM Inference Serving	93
Docker	85	Model Quantization	95
Edge AI	87	Speculative Decoding	97
GPU Acceleration	89	vLLM	99
Inference Optimization	91		

Reading the framework: p. 5 / Profiles: p. 6

First assessment baseline: 1 October 2026 / Reference library: p. 101

THE LEARNING-SCOPE LENS

A next step, not a league table.

A recommendation belongs to a profile, a task and a bounded learning outcome. It is not a rating of a whole technology.

Cover the essentials

Required for this profile's tasks. Start here if you do not yet have this skill.

Extend

Keep the foundation and add a specific capability.

Run a learning trial

Use a bounded exercise to decide whether, and what, to study further.

Maintain

The foundation remains necessary; no additional learning scope is recommended now.

Defer with a trigger

Active study is not justified now. A specific event will trigger a reassessment.

OUTSIDE THE PROFILE IS NOT LOW VALUE

This scope belongs to different responsibilities. It is not a judgment of its value.

READ CONFIDENCE WITH THE RECOMMENDATION

Strong, moderate and limited describe how securely the available evidence supports this editorial assessment. Confidence is not a score for the concept or the reader. A limited assessment can still justify a small, reversible learning trial.

FIRST BASELINE / NO HISTORICAL MOVEMENT

The assessment date records an editorial decision. It does not mean every source was checked again that day. Future changes must name the affected learning outcome and the evidence or task change that caused the revision.

TWO FIXED WORK PROFILES / THREE-MONTH HORIZON

Start with the work.

The same concept can call for different learning decisions when responsibilities change. Keep the profile fixed before comparing editions.

01 / AI APPLICATION BUILDER

A document-based application using managed model APIs and limited tool actions. No model training or self-hosted GPU server.

STARTING POINT

Python, HTTP, basic testing and simple RAG. These are curriculum assumptions, not an assessment of your skills.

SCENARIO

Documents have a freshness requirement. The application uses OCI and two direct API integrations; MCP is not required. An isolated embedding comparison is possible.

02 / MODEL PLATFORM OPERATOR

Model hosting, service requirements, costs, change evaluation and OCI environments. Not application-level RAG design or tool integration.

STARTING POINT

Python, HTTP, basic testing and infrastructure operations. Familiarity with vLLM is not assumed.

SCENARIO

An isolated engine-comparison environment and representative workload are available. No advantage for vLLM has been established. Large-image distribution is not the current bottleneck.

These are explicit curriculum assumptions, not measured personal proficiency. The same evidence can justify different scopes when responsibilities differ.

01 / 47 WORKING RESEARCH SELECTION

Direct Preference Optimization

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Choosing the better of two answers can be easier than writing an ideal answer. DPO uses preference pairs without a separate reward model and a full reinforcement-learning loop.

WHERE TO BE CAREFUL

Results depend on preference quality and task alignment. Studies describe overfitting and regressions; this packet lacks an independent production deployment account.

SIMPLER ALTERNATIVES

Compare supervised examples, better instructions and context. A broader preference-tuning entry could situate DPO among alternatives.

Feedback format

What is being judged?

- | | |
|---|--|
| <p>01 One response (point-wise)
A response receives its own rating, such as good or bad.</p> <p>03 A list of responses (list-wise)
Several responses are ordered by preference.</p> | <p>02 A pair of responses (pair-wise)
The evaluator chooses the better of two responses: the starting point for classic DPO.</p> |
|---|--|

Data acquisition

When and how are new judgments obtained?

- | | |
|--|---|
| <p>01 Existing dataset (offline)
Training uses previously collected examples.</p> <p>03 Targeted evaluation requests (active)
Select which cases deserve additional judgments.</p> | <p>02 Fresh generations (online)
Training is updated using newly generated responses.</p> |
|--|---|

Two independent axes selected from Liu et al., sections III-A2 and III-B1. This is part of a broader taxonomy, not a complete inventory of DPO variants.

SOURCE NOTES / TAXONOMY

[001] A Survey of Direct Preference Optimization

III-A2 Preference Feedback; III-B1 Learning Paradigm.

Assessment evidence (abbreviated): 002, 003, 004, 005. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Direct Preference Optimization

CONFIDENCE / MODERATE

The existing DPO paper establishes the preference objective; the IPO analysis and BPO experiments document bounded failure modes and alternatives. Exclusion from these two learning scopes follows their lack of model-training responsibility, not a claim that preference optimization is unhelpful.

Evidence: [002] [ev-direct-preference-optimization-1](#), [003] [ev-direct-preference-optimization-2](#), [005] [ev-direct-preference-optimization-4](#)

AI application builder

Outside this profile

Preference training is outside the managed-API role

No active learning outcome in this profile.

WHY THIS SCOPE

This builder does not update model weights. Task evaluation and clear application behavior are useful now; implementing a preference-training objective has no corresponding delivery responsibility.

STOP AT THIS BOUNDARY

Comparing preferred answers for evaluation is not the same as owning DPO training.

REVISIT WHEN

Reassess if the role gains model adaptation, a trustworthy preference dataset and an evaluation baseline.

Read alongside: [Supervised Fine-Tuning \(SFT\)](#), [Training Data Curation](#)

Model platform operator

Outside this profile

Serving a model does not require training its preferences

No active learning outcome in this profile.

WHY THIS SCOPE

The operator owns hosting, capacity and release evaluation rather than alignment training. Training teams should supply behavior and regression evidence for any DPO-derived artifact entering the service.

STOP AT THIS BOUNDARY

Require release evidence without expanding the role into preference-data curation or objective design.

REVISIT WHEN

Reassess if training and alignment become explicit platform responsibilities.

In-Context Learning

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Selected examples can improve task performance without updating model weights. Research highlights both task-specific benefits and the importance of matching demonstrations to the problem.

WHERE TO BE CAREFUL

Models may exploit format or similarity rather than a general rule. Ordering and model choice matter, and the topic overlaps ordinary prompt design.

SIMPLER ALTERNATIVES

Compare with a strong zero-shot instruction. Consider query-specific example retrieval or fine-tuning for a lasting behavior change.

Demonstration organization

Which part of the example set is changed?

01 Selection (Demonstration Selection)

Choose examples useful for the new input rather than adding arbitrary ones.

02 Reformatting (Demonstration Reformatting)

Change example presentation; the source also includes generation and representation transformations.

03 Ordering (Demonstration Ordering)

Choose the order in which the model sees demonstrations.

Three demonstration-organization categories from A Survey on In-context Learning, table 1 and section 4.1. They are coexisting decisions, not mutually exclusive model types.

SOURCE NOTES / TAXONOMY

[006] A Survey on In-context Learning

Table 1; 4.1 Demonstration Organization, subsections 4.1.1-4.1.3.

Assessment evidence (abbreviated): 007, 008, 009, 010, 011. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

In-Context Learning

CONFIDENCE / MODERATE

Existing few-shot experiments and demonstration studies show adaptation without weight updates and sensitivity to example selection and format. The recommended application exercise and platform deferral are editorial judgments, without claiming that examples will improve this application's results.

Evidence: [007] [evidence-001](#), [008] [evidence-002](#), [009] [evidence-003](#)

AI application builder

Extend

Add evaluated examples to the prompting toolkit

LEARNING OUTCOMES / CHECKS

01 / Compare a few-shot prompt with the current prompt on held-out cases.

Check: Keep evaluation examples out of the demonstrations, vary selection or order, and report task quality and token cost against the same zero-example baseline.

WHY THIS SCOPE

Examples offer a practical way to express document extraction or classification behavior without training. The builder should learn how to test whether they help rather than assuming that more demonstrations improve the task.

STOP AT THIS BOUNDARY

Keep demonstration and evaluation data separate; do not confuse matching a format with learning a general rule.

REVISIT WHEN

Reassess when task labels, input distribution or the model change, or demonstrations consume an excessive context budget.

Read alongside: [Large Language Models, Long-Context Modeling](#)

Model platform operator

Defer with a trigger

Wait until the platform owns prompt-example selection

No active learning outcome in this profile.

WHY THIS SCOPE

Application teams choose task demonstrations in the stated division of responsibility. The operator needs representative prompt lengths for capacity tests, but that does not require studying demonstration selection in depth.

STOP AT THIS BOUNDARY

Include realistic few-shot requests in workload traces without taking ownership of application prompt design.

REVISIT WHEN

Reassess if the service begins providing shared prompt templates or an example-selection feature.

Large Language Models

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Application builders need to understand how model responses depend on context and why they fail. This supports informed model selection and evaluation; the collected evidence does not identify a universal best model.

WHERE TO BE CAREFUL

This is the broadest title in the set, spanning orientation through architecture research. Without a bounded task, it is difficult to assess as a radar signal.

SIMPLER ALTERNATIVES

Use a more actionable scope such as language-model selection and evaluation, or move introductory foundations into an accompanying guide.

Architecture

How does the model process inputs and produce outputs?

01 Encoder only (Encoder-only)

Builds representations with bidirectional context, useful for classification and text labeling.

02 Decoder only (Decoder-only)

Generates successive tokens from preceding ones, as in many generative LLMs.

03 Encoder and decoder (Encoder-decoder / sequence-to-sequence)

Separates input processing from output generation, for example in translation or summarization.

Three Transformer architecture families from the Hugging Face course. This does not cover every possible language-model architecture or define a quality ranking.

SOURCE NOTES / TAXONOMY

[012] Transformer Architectures

Encoder models; Decoder models; Sequence-to-sequence models; table in Choosing the right architecture.

Assessment evidence (abbreviated): 013, 007, 014, 015, 016. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Large Language Models

CONFIDENCE / MODERATE

The existing few-shot research explains context-conditioned behavior, and the Airbnb and HELM material supports task-specific evaluation rather than a single model ranking. The essential learning scope is an editorial prerequisite decision, not a claim about current frontier-model rankings.

Evidence: [013] [ev-large-language-models-1](#), [007] [ev-large-language-models-2](#), [015] [ev-large-language-models-4](#)

AI application builder

Cover the essentials

Understand model behavior well enough to choose and test it

LEARNING OUTCOMES / CHECKS

01 / Explain the model's task contract and demonstrate its limits.

Check: Use a small set of real application cases to show context dependence, unsupported answers and a task-specific model comparison; explain why fluent text is not evidence of correctness.

WHY THIS SCOPE

Every document answer depends on model behavior, so the builder needs a working account of context, uncertainty and task fit. Keep the broad topic tied to choosing and validating a model for this application.

STOP AT THIS BOUNDARY

No pretraining implementation or exhaustive architecture survey is required for this profile.

REVISIT WHEN

Extend the foundation when a new modality or unexplained failure changes how the application must be evaluated.

Read alongside: [LLM Testing](#), [Structured LLM Outputs](#)

Model platform operator

Cover the essentials

Connect model behavior to the service contract

LEARNING OUTCOMES / CHECKS

01 / Define what a model release must preserve for its consumers.

Check: Document input and output limits, generation settings and version identity, then compare a candidate release on representative quality and request-shape checks supplied by consumers.

WHY THIS SCOPE

A service can remain available while a model change breaks the behavior its consumers rely on. The operator needs model literacy to coordinate quality checks with capacity and compatibility checks.

STOP AT THIS BOUNDARY

Consume task-specific acceptance criteria from application owners; do not infer product quality from infrastructure metrics.

REVISIT WHEN

Reassess whenever model family, context behavior or generation semantics change.

Long-Context Modeling

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Document, conversation and code tasks require checking whether models use relevant information across long inputs. Finding one sentence differs from understanding relationships across the document.

WHERE TO BE CAREFUL

Advertised context capacity may exceed effective use. Results depend on position, task and model; matched-cost business evidence remains limited.

SIMPLER ALTERNATIVES

Compare retrieval, summaries and task decomposition rather than treating a larger context window as the goal.

Long-input task

What should the model do with the extended input?

- | | |
|---|---|
| 01 Single-document question answering
(Single-Document QA)
Find an answer within a long text. | 02 Multi-document question answering
(Multi-Document QA)
Combine information distributed across texts. |
| 03 Summarization
Select and condense relevant information. | 04 Few-shot learning
Use numerous task demonstrations in the input. |
| 05 Synthetic tasks (Synthetic Task)
Perform controlled operations such as counting
or locating passages. | 06 Code completion
Use wider file or repository context. |

All six LongBench task categories from table 1 and section 3.2, without the numerical table data. These are evaluation tasks, not methods for extending context windows.

SOURCE NOTES / TAXONOMY

[017] LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding

Table 1; sections 3.1 and 3.2 Dataset Construction.

Assessment evidence (abbreviated): 018, 019, 020, 021, 022. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Long-Context Modeling

CONFIDENCE / MODERATE

Existing Lost in the Middle, LongBench and LongCodeU experiments show task- and position-dependent use of long inputs. The profile recommendations are editorial applications of those limitations, not claims that a current model has the same failure threshold.

Evidence: [018] [ev-long-context-modeling-1](#), [019] [ev-long-context-modeling-2](#), [020] [ev-long-context-modeling-3](#)

AI application builder

Run a learning trial

Compare long-input reading with the current retrieval baseline

LEARNING OUTCOMES / CHECKS

01 / Test whether whole-document context helps the actual questions.

Check: Use the same document questions for retrieval and longer context; vary the location of key evidence and compare grounded correctness, latency and token usage.

WHY THIS SCOPE

Some document questions require relationships that a short retrieved passage may omit. A bounded comparison can establish whether longer context helps this corpus without equating an advertised window with reliable comprehension.

STOP AT THIS BOUNDARY

Do not replace retrieval or freshness handling solely because documents fit in the context window.

REVISIT WHEN

Extend scope if cross-document questions repeatedly fail because relevant context is omitted and the trial improves those cases.

Model platform operator

Extend

Add long-input quality checks to context-capacity testing

LEARNING OUTCOMES / CHECKS

01 / Validate the usable context contract with application owners.

Check: Test representative request lengths and evidence positions, record quality regressions alongside memory and latency, and document a supported limit with the tested model and configuration.

WHY THIS SCOPE

The endpoint's maximum accepted input length does not by itself establish usable task performance. Combining consumer quality cases with capacity tests prevents a service limit from being presented as a comprehension guarantee.

STOP AT THIS BOUNDARY

Own the service limit and regression checks; application teams still own document assembly and task-specific interpretation.

REVISIT WHEN

Reassess when model, precision, context configuration or customer request lengths change.

Read alongside: [Retrieval-Augmented Generation](#), [Inference Optimization](#)

05 / 47 WORKING RESEARCH SELECTION

LoRA / QLoRA

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

LoRA trains a smaller set of added parameters; QLoRA combines adapters with more compact base-model storage. The collected research and Airbnb account cover training and adapter deployment.

WHERE TO BE CAREFUL

Small adapters may not support every behavior change. Training savings can add serving complexity, and transfer across tasks and environments is not guaranteed.

SIMPLER ALTERNATIVES

Compare full fine-tuning, but first check prompting, examples or retrieval. LoRA may fit inside a broader fine-tuning competency.

What is updated and stored?

How does the fine-tuning approach change memory and trainable parameters?

- | | |
|---|--|
| 01 Full fine-tuning
Update base-model parameters and store their gradients and optimizer state. | 02 LoRA
Freeze the base model and train small low-rank matrices. |
| 03 QLoRA
Train adapters with a quantized base model; low-precision storage does not mean every computation is four-bit. | |

Editorial comparison based on LoRA section 4 and QLoRA sections 3-4, not an author-defined taxonomy from a single table.

SOURCE NOTES / TAXONOMY

[023] LoRA: Low-Rank Adaptation of Large Language Models

4.1 Low-Rank-Parametrized Update Matrices; 4.2 Applying LoRA to Transformer.

[024] QLoRA: Efficient Finetuning of Quantized LLMs

3 QLoRA Finetuning; 4 QLoRA vs. Standard Finetuning.

Assessment evidence (abbreviated): [025](#), [026](#), [027](#), [028](#), [029](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

LoRA / QLoRA

CONFIDENCE / MODERATE

Existing LoRA and QLoRA experiments establish adapter training mechanisms; the Airbnb account describes an adapter release lifecycle in one organization. The current profile exclusions and deferral are editorial responsibility decisions, and do not assume that serving operators train models.

Evidence: [025] [ev-lora-qlora-1](#), [026] [ev-lora-qlora-2](#), [027] [ev-lora-qlora-3](#)

AI application builder

Outside this profile

Adapter training is outside the managed-API application scope

No active learning outcome in this profile.

WHY THIS SCOPE

The builder has no model-training responsibility or self-hosted model server. Prompt, retrieval and evaluation work address the stated application without introducing adapter training and deployment.

STOP AT THIS BOUNDARY

A smaller training artifact still requires data, evaluation and lifecycle ownership.

REVISIT WHEN

Reassess when a persistent behavior gap justifies adaptation and the role gains access to a supported training workflow.

Read alongside: [Supervised Fine-Tuning \(SFT\)](#), [Model Quantization](#)

Model platform operator

Defer with a trigger

Wait for an actual adapter-serving requirement

No active learning outcome in this profile.

WHY THIS SCOPE

Self-hosting alone does not imply that the backend serves adapters. The operator can defer adapter-specific compatibility, switching and rollback study until a training team supplies a concrete artifact and service requirement.

STOP AT THIS BOUNDARY

Future scope should begin with serving and lifecycle validation, not with training LoRA or QLoRA.

REVISIT WHEN

Reassess when an adapter must be loaded, combined, updated or served for multiple tenants.

Reasoning Models

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Models trained for multi-step problem solving can help in some mathematical, coding and verifiable tasks. The competency includes deciding when extra computation is useful.

WHERE TO BE CAREFUL

Longer reasoning does not guarantee correctness and can amplify errors. Strong results on verifiable tasks may not transfer to ambiguous business work; the packet is mainly experimental.

SIMPLER ALTERNATIVES

Compare simpler models, computation tools or explicit algorithms. This topic can be paired with test-time compute budgeting.

How reasoning behavior is learned

Which training route produces the behavior?

01 Reward learning from a base model
(DeepSeek-R1-Zero)

DeepSeek-R1-Zero studies reinforcement learning without preceding supervised fine-tuning.

02 Cold-start examples and multi-stage training
(DeepSeek-R1)

DeepSeek-R1 combines initial examples, reinforcement learning and tuning on selected responses.

03 Distillation to a smaller model (Distillation)

A smaller model learns from responses prepared with a stronger model.

Three routes organized from DeepSeek-R1 sections 2.2-2.4. They belong to one research family, not a complete taxonomy of reasoning models.

SOURCE NOTES / TAXONOMY

[030] DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

2.2 DeepSeek-R1-Zero; 2.3 DeepSeek-R1; 2.4 Distillation.

Assessment evidence (abbreviated): **031**, **032**, **033**, **034**. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Reasoning Models

CONFIDENCE / MODERATE

Existing DeepSeek-R1, complexity and token-efficiency experiments show both capability and task-dependent limits. The trials below are editorial recommendations to measure a relevant task or service workload; they do not update model rankings or forecast gains from current offerings.

Evidence: [031] [ev-reasoning-models-1](#), [033] [ev-reasoning-models-3](#), [034] [ev-reasoning-models-4](#)

AI application builder

Run a learning trial

Test reasoning on a verifiable subset of document tasks

LEARNING OUTCOMES / CHECKS

01 / Compare a reasoning configuration with the current model on difficult cases.

Check: Choose cases with checkable answers, keep inputs and scoring fixed, and compare correctness, source grounding, latency and token usage; include simple cases that may not benefit.

WHY THIS SCOPE

Some application questions may require multi-step synthesis, but that need has not established a benefit from a reasoning model. A bounded comparison can distinguish useful extra work from additional cost and unsupported reasoning.

STOP AT THIS BOUNDARY

Judge the answer and evidence, not the length or apparent sophistication of a reasoning trace.

REVISIT WHEN

Extend scope if a repeatable gain on important cases survives quality, latency and budget limits.

Read alongside: [Test-Time Compute Scaling](#), [LLM Testing](#)

Model platform operator

Run a learning trial

Characterize a candidate reasoning workload before offering it

LEARNING OUTCOMES / CHECKS

01 / Measure a supported reasoning model or configuration in the comparison environment.

Check: Use consumer-supplied quality cases and representative request lengths, compare service latency and generated-token demand with the baseline, and record cancellation and output-limit behavior.

WHY THIS SCOPE

A reasoning workload may change token generation and service capacity requirements. The isolated environment supports learning those consequences without assuming that the model is better or that it should be trained in-house.

STOP AT THIS BOUNDARY

Limit the exercise to a compatible candidate and service evaluation; reinforcement-learning training is outside this profile.

REVISIT WHEN

Expand when an application demonstrates task value and requests a supported production service.

07 / 47 RESEARCH RESERVE

Structured LLM Outputs

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Programmatic consumers need predictable fields and types. The competency defines output contracts, validates results and handles failures; the collected research and LinkedIn accounts provide examples.

WHERE TO BE CAREFUL

Correct structure does not establish correct values. Providers support different constraints. Reserve status reflects set composition, not demonstrated low usefulness.

SIMPLER ALTERNATIVES

For simple cases, use code validation and controlled retries. A broader application-interface competency can emphasize the durable data contract.

Failure location

Why is the output unusable?

- | | | | |
|----|---|----|--|
| 01 | Unsupported schema
The engine cannot prepare the constraints required by the schema. | 02 | Over-constrained
The engine blocks outputs the schema permits. |
| 03 | Under-constrained
The engine allows outputs that violate the schema. | 04 | Valid structure, incorrect content
Values pass structural validation but fail the task. |

Editorial failure map: three engine issues from JSONSchemaBench section 5.3, with content quality treated separately in section 6. This is not a product ranking.

SOURCE NOTES / TAXONOMY

[035] JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models

5.3 JSON Schema Test Suite: Complementary Evaluation; 6 Quality.

Assessment evidence (abbreviated): 036, 037, 038, 039, 040. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Structured LLM Outputs

CONFIDENCE / MODERATE

Existing JSONSchemaBench and structured-output studies distinguish syntax, structure and meaning, while LinkedIn documents machine-consumed model signals. Essential application learning is an editorial task-fit judgment despite research-reserve status; no universal schema conformance is claimed.

Evidence: [036] ev-02, [038] ev-04, [040] ev-08

AI application builder

Cover the essentials

Define and validate every machine-consumed model response

LEARNING OUTCOMES / CHECKS

01 / Implement one response schema with semantic checks and a failure path.

Check: Test valid, malformed, incomplete and structurally valid but incorrect responses; verify that invalid values cannot silently drive a downstream action and that refusal or truncation is handled explicitly.

WHY THIS SCOPE

The application connects model responses to program logic and two integrations. Predictable fields and types are useful only when the application also verifies their meaning and handles responses that cannot satisfy the contract.

STOP AT THIS BOUNDARY

Schema validity is not factual correctness or authorization; check the provider's supported constraints for the actual integration.

REVISIT WHEN

Reassess when a schema, provider, model or downstream action changes.

Read alongside: [LLM Testing](#), [AI Guardrails](#)

Model platform operator

Defer with a trigger

Wait for a structured-generation requirement from consumers

No active learning outcome in this profile.

WHY THIS SCOPE

The backend's current service contract does not specify constrained generation. Engine-specific schema support becomes worth studying when a consumer needs it and supplies representative schemas and acceptance cases.

STOP AT THIS BOUNDARY

Application validation remains necessary even if the engine constrains generation.

REVISIT WHEN

Run a compatibility and performance trial when a consumer requests schema-constrained output.

Supervised Fine-Tuning (SFT)

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Instruction-response examples can tune persistent model behavior. The competency helps decide when to change weights rather than the application; results depend strongly on data and training settings.

WHERE TO BE CAREFUL

Tuning can regress earlier capabilities or safeguards. More data need not help, and the packet is dominated by research experiments rather than independent deployment accounts.

SIMPLER ALTERNATIVES

Start with instructions and in-context examples for small behavior changes, or retrieval for changing facts. LoRA changes how tuning is performed, not its objective.

Training-example objective

What behavior do examples demonstrate?

- 01 General instruction following (General Tasks)
Diverse instructions and conversations without a single dominant specialization.
- 03 Mathematics (Math Tasks)
Solutions requiring calculation and mathematical reasoning.

- 02 Coding (Coding Tasks)
Examples of solving programming tasks.
- 04 Classic NLP tasks
In this study, FLAN datasets covering knowledge, reasoning and text understanding.

Four dataset groups from Massive Supervised Fine-tuning Experiments, section 3.2. These are groups in that study, not a universal SFT taxonomy.

SOURCE NOTES / TAXONOMY

[041] Massive Supervised Fine-tuning Experiments Reveal How Data, Layer, and Training Factors Shape LLM Alignment Quality

3.2 Training Datasets; 3.3 Training Settings.

Assessment evidence (abbreviated): 042, 043, 044, 045, 046. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Supervised Fine-Tuning (SFT)

CONFIDENCE / MODERATE

Existing FLAN and LIMA experiments support instruction-based adaptation, while the fine-tuning safety study documents possible regressions in its tested systems. Both outside decisions are editorial role boundaries; the serving operator is not assigned a training role by implication.

Evidence: [042] [ev-supervised-fine-tuning-sft-1](#), [043] [ev-supervised-fine-tuning-sft-2](#), [044] [ev-supervised-fine-tuning-sft-3](#)

AI application builder

Outside this profile

Model training is outside this application profile

No active learning outcome in this profile.

WHY THIS SCOPE

The stated application uses managed APIs and explicitly excludes model training. Its current learning investment should address task definition, context and evaluation before a different role takes ownership of weight adaptation.

STOP AT THIS BOUNDARY

Fresh documents are not by themselves a reason to train model weights; keep knowledge freshness in the retrieval design.

REVISIT WHEN

Reassess if a persistent behavior gap remains after simpler changes and training ownership, data and evaluation are explicitly added.

Read alongside: [LoRA](#) / [QLoRA](#), [Training Data Curation](#)

Model platform operator

Outside this profile

Require training evidence without becoming the training team

No active learning outcome in this profile.

WHY THIS SCOPE

Hosting and evaluating model releases does not include optimizing training datasets or losses. The operator should accept or reject supplied artifacts against service and regression criteria rather than undertake SFT experiments.

STOP AT THIS BOUNDARY

Model provenance and regression checks belong to release operations; training method development does not.

REVISIT WHEN

Reassess only when the platform explicitly gains responsibility for training or managed adaptation services.

Test-Time Compute Scaling

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Hard tasks may benefit from multiple attempts, checking or extra computation. The competency decides when the added work is worthwhile and when to stop.

WHERE TO BE CAREFUL

More steps can add errors, and many tasks lack a reliable improvement signal. Production evidence combining quality, total cost and latency remains limited.

SIMPLER ALTERNATIVES

Consider a better-matched model, a verification tool or human escalation. A combined entry with reasoning models should retain the budget-allocation question.

How additional work is used

What do extra calls or tokens buy?

- 01 One step-by-step solution (CoT)
Develop a solution path before the final answer.
- 02 Independent parallel attempts (parallel scaling)
Generate candidates, then aggregate results or select an answer.
- 03 Sequential revisions (sequential scaling)
Use critique of an earlier response to produce the next version.

Three approaches studied in Inference-Time Scaling for Complex Tasks, section 2. This is the comparison's scope, not every test-time scaling technique.

SOURCE NOTES / TAXONOMY

[047] Inference-Time Scaling for Complex Tasks: Where We Stand and What Lies Ahead

2 Benchmarks and Methods: Inference-time scaling approaches and aggregators.

Assessment evidence (abbreviated): 048, 049, 050, 032, 051. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Test-Time Compute Scaling

CONFIDENCE / MODERATE

Existing adaptive-compute research supports task-dependent inference allocation, while cross-task and inductive-reasoning studies show diminishing gains and possible regressions. The trial and deferral are editorial decisions; the evidence does not justify more computation for every document task.

Evidence: [048] evidence-001, [050] evidence-003, [032] evidence-004

AI application builder

Run a learning trial

Test a small extra-compute budget on checkable failures

LEARNING OUTCOMES / CHECKS

01 / Compare one bounded extra-attempt or verification policy with the baseline.

Check: Use cases with an independent correctness check, cap attempts or tokens, and report additional correct answers, newly introduced errors, latency and total usage; define a stop rule before the run.

WHY THIS SCOPE

Difficult cases may benefit from another attempt or verification, but the document task does not guarantee a reliable improvement signal. A small controlled exercise can establish whether extra computation has useful value at an acceptable cost.

STOP AT THIS BOUNDARY

Do not use a model's self-confidence as the sole verifier or add unlimited reflection loops.

REVISIT WHEN

Extend scope when a reliable verifier and repeatable gains justify allocating different budgets to different task classes.

Read alongside: Reasoning Models, Inference Optimization

Model platform operator

Defer with a trigger

Wait until the service owns adaptive inference policies

No active learning outcome in this profile.

WHY THIS SCOPE

The operator currently hosts model requests rather than deciding how many attempts solve an application task. Capacity tests should include realistic token budgets, but designing adaptive search or verification policies is not yet an assigned responsibility.

STOP AT THIS BOUNDARY

Distinguish enforcing a service budget from deciding which additional reasoning step improves an answer.

REVISIT WHEN

Reassess when the platform offers shared inference orchestration or adaptive budgets with consumer-provided quality checks.

Training Data Curation

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Training data shapes model behavior alongside algorithms. The competency covers deduplication, labeling and provenance; the collected research and Google Ads account illustrate selection rather than volume alone.

WHERE TO BE CAREFUL

A useful filter in one domain may remove valuable examples in another. Better model scores do not settle provenance, consent or usage rights, and no universal recipe follows.

SIMPLER ALTERNATIVES

Start with a smaller expert-reviewed dataset. Use retrieval rather than training on documents when the need is current knowledge.

What is improved?

What should be removed or deliberately selected?

- | | |
|---|---|
| <p>01 Text extraction</p> <p>Separate article content from navigation, templates and page residue.</p> | <p>02 Quality filtering (Filtering)</p> <p>Reject material based on language, quality and repetition rules.</p> |
| <p>03 Deduplication</p> <p>Detect exact or approximate copies; comparison scope also matters.</p> | <p>04 Objective-specific selection (FineWeb-Edu)</p> <p>Favor a chosen kind of content, such as educational material in FineWeb-Edu.</p> |

Editorial map of FineWeb stages in sections 3.2-3.7 and 4. These are decisions for a particular English-language web corpus, not a universal data recipe.

SOURCE NOTES / TAXONOMY

[052] The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale

3.2 Text extraction; 3.3 Base filtering; 3.4 Deduplication; 3.5-3.7; 4 FineWeb-Edu.

Assessment evidence (abbreviated): 053, 054, 055, 056. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Training Data Curation

CONFIDENCE / MODERATE

Existing FineWeb and Google Ads material links particular curation choices to measured training outcomes, while the provenance audit identifies separate traceability concerns. Excluding training-corpus work follows the two stated roles and is an editorial scope judgment, not a universal recipe for dataset quality.

Evidence: [053] [ev-training-data-curation-1](#), [054] [ev-training-data-curation-2](#), [056] [ev-training-data-curation-5](#)

AI application builder

Outside this profile

Training-corpus curation is outside this application's work

No active learning outcome in this profile.

WHY THIS SCOPE

This builder prepares documents and evaluation cases but does not train models. Those datasets need their own quality controls, yet that does not make a pretraining or fine-tuning curation curriculum necessary.

STOP AT THIS BOUNDARY

Keep retrieval-corpus maintenance and evaluation data distinct from model-training data.

REVISIT WHEN

Reassess when an approved adaptation project assigns the builder responsibility for training examples.

Model platform operator

Outside this profile

Corpus selection belongs to the model-development role

No active learning outcome in this profile.

WHY THIS SCOPE

The operator consumes model artifacts and manages their service behavior. It can require provenance and evaluation documentation without owning corpus filtering, labeling or training-data selection.

STOP AT THIS BOUNDARY

Retain artifact traceability in release operations; do not assume it requires reproducing the training corpus.

REVISIT WHEN

Reassess if training pipelines or dataset stewardship become explicit platform services.

Read alongside: [Supervised Fine-Tuning \(SFT\)](#), [Evaluation Data Engineering](#)

Agentic RAG

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Complex questions can require several sources and follow-up searches. Agentic RAG may choose to retrieve again, reformulate a query or check an answer; the collected studies show benefits in some harder information tasks.

WHERE TO BE CAREFUL

The evidence does not yet establish a sufficient comparison against simple RAG at comparable quality, latency and operating cost. Extra steps add planning and recovery failures; selection remains under research.

SIMPLER ALTERNATIVES

Compare against single-pass RAG and a predefined multi-step workflow. Autonomous next-step selection needs a benefit that the simpler baseline cannot provide.

Retrieval depth

How many retrieval steps are used to answer?

- 01

No retrieval (Non Retrieval)

The model answers without retrieving new documents.
- 02

Single-step retrieval (Single-step Approach)

One set of documents is supplied before answer generation.
- 03

Multi-step retrieval (Multi-step Approach)

Later searches use earlier findings until an answer can be produced.

Three strategies from Adaptive-RAG section 3.1; section 3.2 describes choosing between them. This is one retrieval-control method, not a complete agentic RAG taxonomy or proof that every such system is an agent.

SOURCE NOTES / TAXONOMY

[057] Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity

Section 3.1: Non Retrieval, Single-step and Multi-step; section 3.2: strategy selection.

Assessment evidence (abbreviated): 058, 059, 060, 061. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Agentic RAG

CONFIDENCE / LIMITED

Existing packet evidence: Adaptive-RAG routes between retrieval strategies and RAGChecker separates retrieval from generation failures. The packet lacks a matched end-to-end comparison for this application; deferral is an editorial workload decision and does not resolve the separate research-selection gate.

Evidence: [\[059\] evidence-002](#), [\[060\] evidence-003](#)

AI application builder

Defer with a trigger

Establish the simple retrieval baseline first

No active learning outcome in this profile.

WHY THIS SCOPE

Document freshness requires reliable ingestion and source selection, not necessarily autonomous repeated searches. There is no established failure in the stated scenario that requires a model to choose the next retrieval step.

STOP AT THIS BOUNDARY

Continue basic RAG and its evaluation; this defers dynamic retrieval control, not the underlying retrieval competency.

REVISIT WHEN

Reassess after a labeled set shows multi-source questions that single-pass RAG and a fixed multi-step workflow cannot answer within agreed quality and cost limits.

Read alongside: [Retrieval-Augmented Generation](#), [RAG Evaluation](#)

Model platform operator

Outside this profile

Retrieval control belongs to the application layer

No active learning outcome in this profile.

WHY THIS SCOPE

The operator owns the model backend and has no mandate to choose documents or retrieval sequences. Variable agent workloads matter for capacity tests, but designing their retrieval policy is a separate responsibility.

STOP AT THIS BOUNDARY

Accept representative request traces from application teams without taking ownership of the RAG architecture.

REVISIT WHEN

Reassess if a managed retrieval-and-generation service becomes part of the platform contract.

12 / 47 WORKING RESEARCH SELECTION

Dense Retrieval

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Users and documents often express the same idea differently. Dense retrieval can bridge this gap, and the collected studies compare it with strong lexical baselines.

WHERE TO BE CAREFUL

Meaning similarity is not always the priority for identifiers, names or exact terms. Performance can fall across domains; vector search should not automatically replace lexical search.

SIMPLER ALTERNATIVES

Compare with tuned BM25 and hybrid retrieval. A broader retrieval-selection competency could cover all three approaches.

Where negative passages come from

Which misleading examples train the retriever?

- | | |
|--|---|
| <p>01 Random documents (Random)
Passages sampled randomly from the collection.</p> | <p>02 Lexical matches without the answer (BM25)
High-ranking BM25 passages that do not contain the answer.</p> |
| <p>03 Correct answers to other questions (Gold)
A passage correct for another question becomes a negative for this one, including in-batch negatives.</p> | |

Three negative-example types from DPR section 3.2, compared in table 3. This is one training problem, not all dense retrieval; the radar's broader focus remains retriever selection and evaluation.

SOURCE NOTES / TAXONOMY

[062] Dense Passage Retrieval for Open-Domain Question Answering

3.1 Overview; 3.2 Training: Positive and negative passages; table 3.

Assessment evidence (abbreviated): 063, 064, 060, 065, 066. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Dense Retrieval

CONFIDENCE / MODERATE

Existing packet evidence: DPR demonstrates dual-encoder retrieval, while BEIR documents variation across datasets and retrieval methods. Requiring a task-specific retrieval check is an editorial learning recommendation; no dense-retrieval advantage on this corpus has been measured.

Evidence: [063] evidence-001, [064] evidence-002

AI application builder

Cover the essentials

Know when semantic retrieval misses an exact fact

LEARNING OUTCOMES / CHECKS

01 / Compare semantic and lexical retrieval

Check: Label a small query set containing paraphrases, identifiers and exact names. Compare a dense retriever with a lexical baseline on the same documents and inspect missed relevant passages before selecting a default.

WHY THIS SCOPE

The application depends on retrieving the right documents even when users phrase questions differently. Basic RAG familiarity does not establish that a dense retriever handles exact identifiers or this document domain reliably.

STOP AT THIS BOUNDARY

Use existing embeddings and indexes; retriever training and large-scale search infrastructure are not required.

REVISIT WHEN

Expand to domain adaptation only after representative misses persist across reasonable baseline retrieval choices.

Read alongside: [Embedding Models](#), [Hybrid Search](#)

Model platform operator

Outside this profile

Retrieval relevance is outside this backend role

No active learning outcome in this profile.

WHY THIS SCOPE

The profile hosts models but does not own document collections or search relevance. Dense retrieval requires task-specific queries and relevance judgments from an application owner.

STOP AT THIS BOUNDARY

Serving an embedding endpoint, if separately requested, does not transfer responsibility for retrieval strategy.

REVISIT WHEN

Reassess if the platform is assigned a managed search service with explicit retrieval-quality objectives.

13 / 47 STILL UNDER RESEARCH

Embedding Models

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Embedding choice affects retrieval, clustering and related similarity tasks. The collected research shows differences across languages and applications, supporting task-specific evaluation rather than leaderboard-only selection.

WHERE TO BE CAREFUL

A high overall ranking does not prove the best result on our documents. The packet still lacks a matched quality, representation-size, hardware, index and cost comparison; this concerns selection readiness, not the importance of embeddings.

SIMPLER ALTERNATIVES

Lexical search can be the starting point for exact terminology. Model selection could be included within dense retrieval until a dedicated entry has a strong matched comparison.

Embedding task

What is vector similarity used for?

- | | |
|--|---|
| 01 Bitext mining
Match corresponding sentences across languages. | 02 Classification
Assign text to a predefined category. |
| 03 Clustering
Group texts with related topics. | 04 Pair classification
Decide whether two texts have a relationship, such as being duplicates. |
| 05 Reranking
Reorder an existing candidate list. | 06 Retrieval
Find relevant documents in a collection. |
| 07 Semantic textual similarity (STS)
Estimate how similar the meanings of two sentences are. | 08 Summarization assessment
Compare a summary against human references; this is not summary generation. |

Eight task types from the original MTEB, section 3.2. This maps embedding applications and evaluation, not model architectures or a current leaderboard.

SOURCE NOTES / TAXONOMY

[067] MTEB: Massive Text Embedding Benchmark

3.2 Tasks and Evaluation; figure 1.

Assessment evidence (abbreviated): 068, 069, 060, 070, 071. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Embedding Models

CONFIDENCE / MODERATE

Existing packet evidence: Sentence-BERT explains sentence representations and MTEB compares embeddings across tasks. An isolated comparison is justified by the profile's available environment, but a preferred model, net gain or research-selection readiness is not established by these sources.

Evidence: [068] evidence-001, [069] evidence-002

AI application builder

Run a learning trial

Compare embeddings on the documents you actually use

LEARNING OUTCOMES / CHECKS

01 / Run a controlled embedding comparison

Check: Compare the existing representation with one alternative using identical document chunks, queries and search settings. Report retrieval quality, dimensions, indexing effort and request cost, including failed query types and migration implications.

WHY THIS SCOPE

The scenario explicitly permits an isolated embedding comparison, and document retrieval makes the result relevant. Benchmark rankings alone cannot choose the best representation for the application's language and content.

STOP AT THIS BOUNDARY

One controlled comparison is enough for this horizon; no custom encoder training or automatic production reindexing is recommended.

REVISIT WHEN

Adopt a larger learning scope only if the trial identifies a meaningful quality or resource trade-off on the application's acceptance criteria.

Read alongside: [Dense Retrieval](#), [RAG Evaluation](#)

Model platform operator

Outside this profile

Embedding selection needs a retrieval workload owner

No active learning outcome in this profile.

WHY THIS SCOPE

This profile has no assigned embedding service or application retrieval objective. Choosing a representation for semantic relevance would require responsibilities beyond the specified model backend.

STOP AT THIS BOUNDARY

Do not infer an embedding-hosting requirement from general model hosting.

REVISIT WHEN

Reassess when embedding inference becomes an explicit service, with a defined workload and an owner for semantic-quality evaluation.

14 / 47 RESEARCH RESERVE

Hybrid Search

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Lexical search helps with exact terms, while semantic retrieval handles alternative wording. Hybrid search is the skill of combining and evaluating these signals.

WHERE TO BE CAREFUL

Two methods need not beat one; a weaker branch can damage ranking while adding cost and tuning. Reserve status reflects overlap in the set, not demonstrated lower value.

SIMPLER ALTERNATIVES

Compare the combination against the strongest standalone lexical and semantic methods. A common retrieval-strategy entry may avoid excessive fragmentation.

Relevance signal

Why does a document rank highly?

01 Lexical match (BM25)

The query matches terms in the document.

02 Vector similarity (kNN)

The query and document representations are similar.

Result fusion

How are different rankings combined?

01 Rank-based fusion (RRF)

Combine list positions without directly comparing score scales.

02 Score-based fusion (linear retriever)

Combine weighted scores; normalization may align their scales.

Two editorial axes based on Elastic's explanations of BM25, kNN, RRF and linear fusion. The scope is the common lexical-vector hybrid.

SOURCE NOTES / TAXONOMY

[072] Hybrid search revisited: introducing the linear retriever in Elasticsearch!

Meet the linear retriever!; Scaling the scores: kNN vs BM25; Normalization to the rescue.

Assessment evidence (abbreviated): 073, 060, 074, 072. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Hybrid Search

CONFIDENCE / MODERATE

Existing packet evidence: Elastic explains rank and score fusion; Balancing the Blend examines component-quality and fusion trade-offs. A controlled comparison is an editorial learning choice, not an assumption that combining retrievers will help; reserve status concerns catalog composition separately.

Evidence: [072] [elastic-linear-hybrid](#), [074] [evidence-003](#)

AI application builder

Run a learning trial

Test fusion only against both standalone retrievers

LEARNING OUTCOMES / CHECKS

01 / Compare lexical, dense and fused retrieval

Check: On the same labeled queries, compare lexical retrieval, dense retrieval and one fusion method. Separate identifier queries from paraphrases and report retrieval quality and latency for every branch, including cases where fusion loses.

WHY THIS SCOPE

Document questions may mix exact terms with alternative wording, making complementary signals worth a bounded experiment. The result depends on the corpus and each component's quality, so fusion should earn its additional complexity.

STOP AT THIS BOUNDARY

One fusion trial is sufficient; it does not require changing the database or adding a reranker at the same time.

REVISIT WHEN

Extend study if the trial shows recurring complementary successes and the fused system meets the application's latency budget.

Read alongside: [Dense Retrieval](#), [Search Re-Ranking](#)

Model platform operator

Outside this profile

Search fusion belongs to retrieval owners

No active learning outcome in this profile.

WHY THIS SCOPE

The backend role has no responsibility for lexical indexes, vector candidate lists or query relevance. Hosting the model alone does not create a need to learn rank fusion implementation.

STOP AT THIS BOUNDARY

Application-provided request mixes may inform capacity tests without bringing search-strategy ownership into this profile.

REVISIT WHEN

Reassess if a shared hybrid-search service and its relevance targets become platform responsibilities.

Multimodal RAG

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Tables, charts and page layout can carry information lost by plain-text extraction. The collected studies examine retrieval and interpretation that preserve this visual structure.

WHERE TO BE CAREFUL

Errors can arise from page reading through number-label alignment. Higher cost needs useful visual information to justify it; evidence is mostly benchmarks rather than independent deployments.

SIMPLER ALTERNATIVES

Start with strong text and table extraction, escalating difficult pages to a vision-capable model. A separate entry needs a distinct visual-document problem.

Retrieval granularity

What is one retrieval result?

- 01

Whole page (page-level)
Retrieve a page; whether its layout is preserved depends on its representation.
- 02

Page element (layout-level)
Retrieve a table, figure, equation or text region.

Document representation

What does the retriever process?

- 01

Extracted text (text-driven)
OCR or a vision model converts content to text before retrieval.
- 02

Visual-language representation (visual-driven)
Process an image or a hybrid image-text input.

Two independent MMDocIR axes: tasks in section 2 and retriever types in section 5.2. These concern document retrieval, not every RAG modality.

SOURCE NOTES / TAXONOMY

[075] MMDocIR: Benchmarking Multimodal Retrieval for Long Documents

2 Dual-Task Retrieval Definition; 5.2 Baseline Models and Setting; tables 5-6.

Assessment evidence (abbreviated): 076, 060, 077, 078, 079. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Multimodal RAG

CONFIDENCE / MODERATE

Existing packet evidence: ColPali evaluates page-image retrieval, and MMDocIR tests multimodal retrieval over long documents. The sources support visual-information use cases; the fixed scenario has no established visual-retrieval requirement, so deferral is an editorial scope judgment.

Evidence: [076] evidence-001, [077] evidence-003

AI application builder

Defer with a trigger

Wait for evidence that text extraction loses required answers

No active learning outcome in this profile.

WHY THIS SCOPE

Document-based work alone does not establish that page images or visual layouts are needed for the application's questions. First identify answers that depend on charts, tables or spatial relationships and cannot be recovered by adequate text extraction.

STOP AT THIS BOUNDARY

This defers multimodal retrieval architecture, not inspecting source pages or preserving tables during ingestion.

REVISIT WHEN

Reassess when a labeled failure set depends on visual evidence; compare a page-image path with improved extraction on answer quality and cost.

Read alongside: [Retrieval-Augmented Generation, RAG Evaluation](#)

Model platform operator

Outside this profile

Document interpretation belongs to the retrieval application

No active learning outcome in this profile.

WHY THIS SCOPE

The operator has no ownership of page representations, document indexes or answer grounding. Serving a vision-language model, if later required, would still be distinct from designing multimodal RAG.

STOP AT THIS BOUNDARY

No document extraction or visual-retrieval pipeline is included in this profile.

REVISIT WHEN

Reassess if the platform takes responsibility for an end-to-end multimodal document service.

Retrieval-Augmented Generation

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

RAG supplies external knowledge that can be updated without retraining model parameters. The collected material includes operating systems and outcomes reported by their operators.

WHERE TO BE CAREFUL

Retrieved documents can be stale, irrelevant or misused. A citation alone does not establish support, and particular deployment accounts do not prove value for every application.

SIMPLER ALTERNATIVES

Consider ordinary document search, passing a short source directly into context, or controlled queries over structured data.

System organization

How are retrieval and generation connected?

- 01

Simple pipeline (Naive RAG)
Prepare documents, retrieve passages and supply them to a model.
- 02

Enhanced pipeline (Advanced RAG)
Improve pre- and post-retrieval steps, such as query reformulation or passage selection.
- 03

Modular system (Modular RAG)
Combine modules and vary the workflow, for example through source selection or repeated retrieval.

Three paradigms identified in the RAG survey, section II and figure 3. They describe architecture, not guaranteed quality or mandatory maturity stages.

SOURCE NOTES / TAXONOMY

[080] Retrieval-Augmented Generation for Large Language Models: A Survey

II-A, II-B and II-C; figure 3, Comparison between the three paradigms of RAG.

Assessment evidence (abbreviated): 081, 082, 083, 084, 085. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Retrieval-Augmented Generation

CONFIDENCE / STRONG

Existing packet evidence: the RAG survey covers retrieval, augmentation, generation and limitations; the RGB benchmark probes noise, rejection and conflicting inputs. Extending basic RAG to document freshness is an editorial response to the explicit scenario, not a measured outcome or a claim that citations guarantee support.

Evidence: [083] evidence-003, [082] evidence-002

AI application builder

Extend

Extend simple RAG with a document freshness contract

LEARNING OUTCOMES / CHECKS

01 / Test updates, removal and source support

Check: Replace and delete a document in a small corpus, then rerun questions whose answers depend on it. Verify which source version was retrieved, measure update visibility against the stated freshness requirement, and confirm unsupported questions do not receive invented support.

WHY THIS SCOPE

Simple RAG is already an entry assumption, while the scenario explicitly requires fresh documents. The next useful learning scope is governing source versions and proving that updates reach answers correctly.

STOP AT THIS BOUNDARY

Preserve the simple retrieval baseline; this does not require agentic retrieval, custom model training or a new vector database.

REVISIT WHEN

Expand further when access boundaries, conflicting sources or observed retrieval failures introduce requirements beyond freshness.

Read alongside: [RAG Evaluation](#), [Vector Databases](#)

Model platform operator

Outside this profile

Application-level RAG is outside this hosting profile

No active learning outcome in this profile.

WHY THIS SCOPE

The profile explicitly excludes application-level RAG design. Hosting the generator provides capacity for RAG requests but does not establish ownership of document ingestion, retrieval or evidence support.

STOP AT THIS BOUNDARY

Use representative RAG-shaped requests for serving tests without taking over the application's knowledge pipeline.

REVISIT WHEN

Reassess if the platform is assigned an end-to-end managed knowledge service.

Search Re-Ranking

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

A fast first-stage search may find useful candidates but order them poorly. Reranking applies a more detailed comparison; the collected research and Uber account provide practical examples.

WHERE TO BE CAREFUL

Additional comparisons cost time and compute. Reranking cannot recover a document missed by retrieval, and a whole-system improvement does not isolate its contribution.

SIMPLER ALTERNATIVES

First examine retrieval, filters and document chunking. Measure incremental quality against the added second-stage cost.

Scoring unit

How does the model assess relevance to the query?

- | | |
|--|---|
| <p>01 One document (pointwise)
Score each document individually and build a ranking from the scores.</p> <p>03 A document list (listwise)
Order candidates presented together.</p> | <p>02 Two documents (pairwise)
Choose which document in a pair better answers the query.</p> |
|--|---|

Scoring taxonomy used in the RankLLM research tool's Overview. It describes assessment methods, not a leaderboard of models.

SOURCE NOTES / TAXONOMY

[086] RankLLM

README: Overview; MonoT5 Suite - Pointwise Rerankers.

Assessment evidence (abbreviated): **087**, **064**, **060**, **070**, **088**. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Search Re-Ranking

CONFIDENCE / MODERATE

Existing packet evidence: Passage Re-ranking with BERT evaluates second-stage ranking, while BEIR shows cross-domain variation. A local trial can test incremental value; the sources do not establish that reranking fixes missing candidates or improves this application's final answers.

Evidence: [087] evidence-001, [064] evidence-002

AI application builder

Run a learning trial

Test a reranker on a fixed candidate set

LEARNING OUTCOMES / CHECKS

01 / Measure the added ranking stage

Check: Freeze retrieved candidate sets for labeled questions, then compare their original order with one reranker. Measure top-result relevance, downstream answer support and added latency, and count questions whose evidence was absent before reranking.

WHY THIS SCOPE

A bounded reranking trial can reveal whether useful evidence is retrieved but placed too low for the answer context. It should follow retrieval diagnosis because no reranker can recover an absent candidate.

STOP AT THIS BOUNDARY

Keep retrieval and generation settings fixed during the comparison; no production adoption is implied by the learning trial.

REVISIT WHEN

Continue when ranking errors are a recurring bottleneck and the added stage meets the application's quality and latency criteria.

Read alongside: [Dense Retrieval](#), [RAG Evaluation](#)

Model platform operator

Outside this profile

Ranking policy needs a search workload owner

No active learning outcome in this profile.

WHY THIS SCOPE

The operator's model backend has no assigned candidate-ranking task or relevance labels. Reranker hosting can be an operational service, but choosing and evaluating ranking policy belongs to the retrieval owner.

STOP AT THIS BOUNDARY

No reranker model or endpoint is assumed by the stated hosting scenario.

REVISIT WHEN

Reassess if a reranking service with explicit workload, quality ownership and latency targets is added to the platform.

18 / 47 WORKING RESEARCH SELECTION

Vector Databases

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Operational vector retrieval involves updates, filters, replication and availability as well as similarity search. The collected material treats this as data-system architecture.

WHERE TO BE CAREFUL

A separate vector database is not always needed. An existing database or local index may suffice; matched total-cost comparisons are limited.

SIMPLER ALTERNATIVES

Consider vector support in an existing relational or graph database, or a library for small collections. Do not assume another product is required.

Data management

Which operational problem is being addressed?

- | | |
|--|---|
| 01 Sharding
Distribute data and work across nodes. | 02 Partitioning
Organize datasets to restrict the scope of reads. |
| 03 Caching
Speed up reuse of frequently needed data. | 04 Replication
Maintain copies for availability and read service. |

Search accuracy

Is approximate neighbor selection acceptable?

- | | |
|--|--|
| 01 Exact search
Find the nearest vectors under the chosen measure. | 02 Approximate search
Trade some accuracy for lower search cost. |
|--|--|

Part of Ma et al.'s taxonomy: section 2 and figure 2, plus the search distinction from section 3. Functions can coexist. Product ratings from the comparison table are not reproduced.

SOURCE NOTES / TAXONOMY

[089] A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge

Figure 2; 2 Storage (2.1-2.4); 3 Search.

Assessment evidence (abbreviated): 090, 091, 092, 088. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Vector Databases

CONFIDENCE / MODERATE

Existing packet evidence: the vector-database survey distinguishes storage and search responsibilities; NaviX provides a native vector-index alternative inside a graph DBMS. Learning lifecycle guarantees is an editorial response to the freshness requirement and does not endorse a separate database product.

Evidence: [090] evidence-001, [092] evidence-003

AI application builder

Cover the essentials

Understand the retrieval store's update and deletion behavior

LEARNING OUTCOMES / CHECKS

01 / Verify the chosen store's retrieval contract

Check: Using the existing retrieval store, test inserting, updating and deleting documents together with metadata filtering. Record when each change is visible to queries and explain whether an existing database or local index meets the required lifecycle.

WHY THIS SCOPE

A freshness requirement depends on the data store's behavior as well as the embedding model. The reader needs enough database understanding to verify updates and filtering without assuming a dedicated vector product is necessary.

STOP AT THIS BOUNDARY

Learn the contract of the chosen store; cluster design, replication internals and a vendor migration are not prescribed.

REVISIT WHEN

Expand operational study when measured collection size, update load or availability requirements exceed the current store's capabilities.

Read alongside: [Retrieval-Augmented Generation](#), [Vector Indexing](#)

Model platform operator

Outside this profile

Vector storage is a separate data-service responsibility

No active learning outcome in this profile.

WHY THIS SCOPE

The fixed operator profile owns model serving rather than application retrieval storage. A vector database would introduce persistence and search responsibilities that are not implied by hosting a model.

STOP AT THIS BOUNDARY

Model weights and inference caches are not vector-database workloads.

REVISIT WHEN

Reassess when operating a shared vector store becomes an explicit platform service with data-lifecycle and availability targets.

Vector Indexing

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

At scale, comparing every vector may be too expensive. Index selection trades retrieval accuracy against latency and memory, with settings and data characteristics affecting results.

WHERE TO BE CAREFUL

The same algorithm can behave differently with insertion order and data structure. Managed-service users may need less depth, and a separate entry can fragment the map.

SIMPLER ALTERNATIVES

For small collections, compare with exact search. Combine with vector databases unless large-scale retrieval tuning needs distinct treatment.

Candidate selection

Which vectors are compared with the query?

- 01

Full scan (Flat)

Search the whole collection.

02

Selected clusters (IVF)

Visit only selected lists of similar vectors.
- 03

Graph navigation (HNSW)

Follow connections between vectors to find candidates.

Vector representation

How is storage reduced?

- 01

Uncompressed

Keep the full representation.

02

Scalar quantization (SQ)

Store each coordinate with lower precision.
- 03

Product quantization (PQ)

Replace vector parts with codebook entries.

Two editorial axes from the official Faiss methods table and encoding sections. These are representative options, not a complete index catalog.

SOURCE NOTES / TAXONOMY

[093] Faiss indexes

Summary of methods; Vectors encodings; Cell-probe methods; IndexHNSW variants.

Assessment evidence (abbreviated): 094, 095, 091, 088. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Vector Indexing

CONFIDENCE / MODERATE

Existing packet evidence: HNSW describes approximate graph search, while the insertion-order and dimensionality study demonstrates recall sensitivity. Deferring index tuning follows the absence of a measured index bottleneck in the scenario; it does not imply that default index settings are reliable.

Evidence: [094] evidence-001, [091] evidence-003

AI application builder

Defer with a trigger

Tune an index after measuring a retrieval bottleneck

No active learning outcome in this profile.

WHY THIS SCOPE

The scenario provides no collection size, recall failure or latency limit that calls for index-specific optimization. Retrieval evaluation and the store's update contract should establish the problem before learning several approximate-index families.

STOP AT THIS BOUNDARY

Continue measuring retrieval quality; this defers algorithm and parameter tuning rather than accepting unexplained misses.

REVISIT WHEN

Reassess when comparison with exact search reveals material approximate-search losses or measured search latency or memory exceeds the application budget.

Read alongside: [Vector Databases](#), [Dense Retrieval](#)

Model platform operator

Outside this profile

Search indexes are outside the inference backend

No active learning outcome in this profile.

WHY THIS SCOPE

This profile does not operate a vector-retrieval service or own its recall target. GPU serving expertise does not by itself create a need to design approximate-nearest-neighbor indexes.

STOP AT THIS BOUNDARY

Model memory management and KV-cache tuning remain separate from vector search indexing.

REVISIT WHEN

Reassess if a shared retrieval index with defined recall, memory and latency requirements is assigned to the platform.

Agent Memory Systems

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Long-running assistants need to retain earlier decisions and recognize outdated ones. This involves writing, retrieving, updating and deleting memories. The collected research compares memory designs, accuracy and cost.

WHERE TO BE CAREFUL

More stored information can introduce stale or conflicting instructions. The material contains more experiments than long-term operational evidence, particularly for deletion, privacy and memory repair.

SIMPLER ALTERNATIVES

Conversation history or a summary may suffice for short tasks. Store authoritative facts in explicit application state when model interpretation is unnecessary.

Four memory-system families

Who turns interaction history into memory, and how?

- 01

History in context (Long-context memory)
The model receives earlier conversation text without an external memory store.
- 02

Retrieval over stored history (Flat RAG memory)
History is indexed without LLM interpretation; examples include BM25 and embedRAG.
- 03

Structured facts and relations
(Structure-augmented RAG memory)
The model extracts information at predefined points. Appending differs from merging and updating existing records.
- 04

Agent-controlled memory operations (Agentic control flow)
The model chooses memory operations rather than following a fixed read/write workflow.

Four families from Omri et al., section 2.2 and table 1 (PDF p. 4), compared along four additional independent dimensions. The original atlas visually checked the table. These are summaries, not a reproduction; the source is CC BY 4.0.

SOURCE NOTES / TAXONOMY

[096] Agent Memory: Characterization and System Implications of Stateful Long-Horizon Workloads

v1; sections 2.1-2.2; table 1, PDF p. 4; figure 1, PDF p. 1.

Assessment evidence (abbreviated): 097, 019, 098, 099, 100. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Agent Memory Systems

CONFIDENCE / MODERATE

Existing packet evidence: the memory-system study separates construction, retrieval and generation costs; the evolving-state benchmark tests superseded facts and constraints. These support lifecycle questions, while deferring a memory subsystem follows the profiles' stated responsibilities rather than a demonstrated lack of value.

Evidence: [099] ev-04, [100] ev-05

AI application builder

Defer with a trigger

Wait for a real cross-session memory requirement

No active learning outcome in this profile.

WHY THIS SCOPE

The scenario requires fresh documents and limited tool actions, but does not require an assistant to retain personal facts across sessions. Explicit application state and retrieved source documents cover the stated needs without an additional memory lifecycle.

STOP AT THIS BOUNDARY

This defers agent-managed memory, not ordinary conversation context, durable business records or document updates.

REVISIT WHEN

Reassess when a task must recall and revise user-specific facts across sessions; include deletion and supersession cases in the trial.

Read alongside: [Long-Context Modeling](#), [Retrieval-Augmented Generation](#)

Model platform operator

Outside this profile

Keep agent memory separate from serving caches

No active learning outcome in this profile.

WHY THIS SCOPE

Durable agent memories encode application meaning and retention rules that this backend profile does not own. KV-cache operations are a different competency and do not supply memory governance.

STOP AT THIS BOUNDARY

No memory-store selection, personalization or conversation-history policy is included in this profile.

REVISIT WHEN

Reassess if operating a managed agent-memory service becomes an explicit platform responsibility.

Agentic Planning & Task Decomposition

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Complex tasks may benefit from smaller steps and checks along the way. The competency covers plans, dependencies and recovery. Research also reveals how longer plans fail.

WHERE TO BE CAREFUL

A flawed initial plan can amplify errors. Better results may reflect more attempts or model calls rather than better planning; maintenance costs are less well understood.

SIMPLER ALTERNATIVES

Use a fixed workflow with explicit conditions for repeatable work. Dynamic planning needs a reason: the next step genuinely depends on information discovered during execution.

Planning horizon

Is the plan created before execution or after observing earlier results?

- 01 Holistic planning

The entire path and tool sequence are planned before execution.
- 02 Step-wise planning

The next action or short sequence uses the existing trace and tool responses.

Planning failure modes

Several failure modes can occur together.

- 01 Goal misunderstanding

The agent solves a different task from the intended one.
- 02 Incomplete plan or early termination

Part of the goal is omitted or execution stops before required steps.
- 03 Constraint violation

The plan violates an explicit task requirement.
- 04 Logical error

Steps are ordered incorrectly or necessary premises are missing.
- 05 Tool-use error

The agent misunderstands an available tool.
- 06 Fabricated information or tools

The plan relies on invented facts, unobserved results or nonexistent tools.

Planning horizons and failure causes from Agent Planning Benchmark. Figure 2 was visually checked in the original atlas. This is a benchmark diagnostic framework, not a complete inventory of planning algorithms.

SOURCE NOTES / TAXONOMY

[101] Agent Planning Benchmark: A Diagnostic Framework for Planning Capabilities in LLM Agents

v1; section 3.1, PDF p. 3; section 3.3, pp. 4-5; figure 2, p. 2; appendix C.1, pp. 24 and 26.

Assessment evidence (abbreviated): 102, 103, 032, 104, 105. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Agentic Planning & Task Decomposition

CONFIDENCE / MODERATE

Existing packet evidence: Agent Planning Benchmark separates plan quality from execution; Tree of Thoughts studies search and backtracking on a small task set with additional model calls. The learning decision is an editorial fit judgment for the fixed scenario, not a claim that dynamic planning generally fails.

Evidence: [105] ev-05, [103] ev-02

AI application builder

Defer with a trigger

Use the known workflow until dynamic planning is needed

No active learning outcome in this profile.

WHY THIS SCOPE

Two direct integrations and document retrieval do not by themselves require a model to invent a plan. Dynamic decomposition adds branches and recovery decisions whose value has not been established for this scenario.

STOP AT THIS BOUNDARY

Keep ordinary task decomposition and explicit workflow design; defer autonomous planner implementation and search over plans.

REVISIT WHEN

Reassess when representative tasks require discovering their next step during execution and a fixed workflow has documented failures.

Read alongside: [AI Agent Design](#), [Agent Evaluation](#)

Model platform operator

Outside this profile

Application planning is outside model hosting

No active learning outcome in this profile.

WHY THIS SCOPE

The platform provides inference capacity and service guarantees, not task-level goals and dependencies. Learning an autonomous planning architecture would expand this profile beyond its stated responsibilities.

STOP AT THIS BOUNDARY

Capacity planning and deployment orchestration are not agentic task planning.

REVISIT WHEN

Reassess if the platform is asked to provide and support a shared planning runtime.

AI Agent Design

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Before choosing a library, decide the agent's autonomy, tools and human checkpoints. The collected architecture accounts and comparisons illustrate these design choices.

WHERE TO BE CAREFUL

The title can overlap most other agent competencies. No architecture is best everywhere. Its reserve status reflects the composition of the set, not a finding of low value.

SIMPLER ALTERNATIVES

Keep planning, memory and coordination separate, or use architecture as an umbrella if a shorter map better serves readers.

Answering, reasoning and acting

What does the model do before finishing a task?

- 01

Direct answer (Standard)
The model answers without an explicit intermediate action loop.
- 02

Reasoning only (Chain-of-thought)
The model generates reasoning steps without obtaining observations through tools.
- 03

Acting without explicit reasoning steps (Act-only)
Actions and observations occur without interleaved reasoning narration.
- 04

Interleaved reasoning and acting (ReAct)
Observations can change the plan and the next action.

Editorial map of four variants compared in ReAct figure 1; section 2 explains the loop. These are historical prompting and action patterns, not a complete taxonomy of modern agent architectures.

SOURCE NOTES / TAXONOMY

[106] ReAct: Synergizing Reasoning and Acting in Language Models

Figure 1: four approaches to answering a question; section 2, ReAct: Synergizing Reasoning + Acting.

Assessment evidence (abbreviated): 107, 108, 109, 110, 111. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

AI Agent Design

CONFIDENCE / MODERATE

Existing packet evidence: ReAct describes interleaved reasoning and actions; AgentArch compares bounded architecture configurations and shows task-dependent results. The essentials recommendation concerns deciding autonomy and control, not endorsing a winning architecture; reserve status remains a separate catalog-selection decision.

Evidence: [107] ev-01, [109] ev-03

AI application builder

Cover the essentials

Define where the model may decide and act

LEARNING OUTCOMES / CHECKS

01 / Specify a bounded tool workflow

Check: Draw the two-integration workflow with model choices, deterministic steps, approvals and stop conditions. Walk through an ambiguous request and a failed tool call, and explain why each model-controlled step is necessary.

WHY THIS SCOPE

Limited tool actions still require a deliberate choice about what the model may decide. A small design exercise makes human checkpoints and failure handling explicit before a framework makes those choices implicitly.

STOP AT THIS BOUNDARY

Cover one bounded workflow; no multi-agent architecture, framework migration or general-purpose autonomy is required.

REVISIT WHEN

Revisit the design when tool permissions, action consequences or workflow uncertainty increase.

Read alongside: [Agent Evaluation](#), [Agentic Planning & Task Decomposition](#)

Model platform operator

Outside this profile

Keep task autonomy with the application team

No active learning outcome in this profile.

WHY THIS SCOPE

The model-hosting contract does not include business workflows or authority to perform their actions. Serving models used by agents does not make the operator responsible for designing those agents.

STOP AT THIS BOUNDARY

Model API capabilities and operational limits remain in scope; application autonomy and approvals do not.

REVISIT WHEN

Reassess if the platform begins supplying a shared agent product rather than an inference endpoint.

AI-Assisted Development

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Coding assistants change how software is drafted, reviewed and revised. The competency is verified human-tool collaboration, including task division and judging proposed changes.

WHERE TO BE CAREFUL

Results are mixed: the collected METR experiment found slower completion for some experienced developers on the studied tasks. Code volume is not value, and long-term review and maintenance costs remain uncertain.

SIMPLER ALTERNATIVES

Start with drafts, documentation or tests, and compare against the existing workflow. Keep the competency independent of a particular assistant.

Collaboration mode

Which part of the work is delegated to AI?

- | | |
|--|---|
| <div>01</div> <div>Code completion</div> <div>The person leads editing while the model suggests fragments.</div> | <div>02</div> <div>Conversation and explanation</div> <div>The model helps explain code or plan a change.</div> |
| <div>03</div> <div>Task execution</div> <div>An agent selects files, makes changes and iterates on the task.</div> | <div>04</div> <div>Change review</div> <div>AI comments on code that has already been written.</div> |

Editorial map based on GitHub Copilot feature descriptions, not a universal standard. The METR study limits productivity claims. The broader feature map does not extend this radar competency to autonomous issue-to-PR operation.

SOURCE NOTES / TAXONOMY

[112] GitHub Copilot features

Assistive features; Agentic features.

[113] Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity

Methodology; limitations table following Core Result.

Assessment evidence (abbreviated): 114, 113, 115, 116, 117. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

AI-Assisted Development

CONFIDENCE / MODERATE

Existing packet evidence: the Google engineering trial and METR study report different productivity outcomes in different settings. They justify measuring an assisted workflow locally; neither establishes a speed or quality gain for these readers, and the trial recommendation is editorial.

Evidence: [114] ev-01, [113] ev-02

AI application builder

Run a learning trial

Measure an assistant on a small application change

LEARNING OUTCOMES / CHECKS

01 / Evaluate the complete assisted change cycle

Check: Use an assistant for one bounded application change and compare with a similar unassisted task. Record implementation, review and repair time, run the same acceptance checks, and list limitations before deciding whether to continue.

WHY THIS SCOPE

The profile already includes Python and testing, so a supervised coding exercise can use existing skills. Mixed study results make local review effort and correctness more useful than an assumed productivity gain.

STOP AT THIS BOUNDARY

Keep a human responsible for scope, review and merging; do not infer causal productivity gains from one pair of tasks.

REVISIT WHEN

Expand use only after repeated tasks show acceptable review effort and no unresolved correctness regressions.

Read alongside: [Code Execution Agents](#), [LLM Testing](#)

Model platform operator

Run a learning trial

Try assistance on a reversible operational script

LEARNING OUTCOMES / CHECKS

01 / Review and test assisted operational code

Check: Generate a small workload-analysis or configuration-validation script in an isolated environment. Check every command and dependency, compare its outputs with known cases, and record review time as part of the trial.

WHY THIS SCOPE

Operational Python work offers a bounded place to test coding assistance without redesigning the platform. Infrastructure consequences make verification and human review part of the learning target.

STOP AT THIS BOUNDARY

No autonomous production changes or credentials in prompts; the exercise is supervised code assistance, not an autonomous infrastructure agent.

REVISIT WHEN

Continue when repeated low-risk scripts pass independent checks and the review burden is worthwhile.

24 / 47 WORKING RESEARCH SELECTION

Code Execution Agents

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Executing a test, calculation or script lets an agent check the effects of its proposals. The collected Spotify account and research examine code changes and repeated development cycles.

WHERE TO BE CAREFUL

Accepted changes do not reveal total review time or later defects. Benchmark contamination can inflate results, and code execution increases the potential impact of errors.

SIMPLER ALTERNATIVES

Let a human run proposed code, or restrict the agent to a closed set of vetted tools rather than arbitrary execution.

Responsibilities around a code change

Which work belongs to the agent and which to its surrounding process?

01 Creating a change

An agent interprets the task and edits code instead of following a prewritten migration script.

02 Checking a change

Tools format and analyze code; an additional assessment may inspect the diff.

03 Delivering a change

The surrounding process selects repositories, opens pull requests, and organizes review and merging.

Editorial synthesis of the Honk account, particularly Enter AI coding agents and the text around figures 3-4. Spotify does not present these as a universal taxonomy of code-execution agents.

SOURCE NOTES / TAXONOMY

[118] 1,500+ PRs Later: Spotify's Journey with Our Background Coding Agent (Honk, Part 1)

Automating toil using Fleet Management; Enter AI coding agents; text around figures 3 and 4.

Assessment evidence (abbreviated): 119, 120, 121, 118. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Code Execution Agents

CONFIDENCE / MODERATE

Existing packet evidence: SWE-bench evaluates repository patches through tests; SWE-CI adds repeated maintenance cycles and the contamination study challenges benchmark interpretation. Deferral follows the absence of an autonomous-code requirement in these profiles, not a claim that execution agents are ineffective.

Evidence: [119] ev-01, [121] ev-03, [120] ev-02

AI application builder

Defer with a trigger

Keep autonomous code execution out of the current scope

No active learning outcome in this profile.

WHY THIS SCOPE

A document application with two API integrations has no stated need to generate and execute arbitrary programs. A coding assistant can be tried separately before adding an autonomous execution and review workflow.

STOP AT THIS BOUNDARY

This defers code-executing agents; it does not defer ordinary tests, scripts or supervised development tools.

REVISIT WHEN

Reassess when a concrete calculation, transformation or repeated repository task requires generated code and a sandbox with explicit acceptance tests is available.

Read alongside: [AI-Assisted Development, Agent Evaluation](#)

Model platform operator

Defer with a trigger

Wait for a bounded automation task and execution controls

No active learning outcome in this profile.

WHY THIS SCOPE

The hosting scenario provides an engine-comparison environment but no requirement for an autonomous coding workflow. Adding generated execution would require separate isolation, permission and review decisions.

STOP AT THIS BOUNDARY

Continue deterministic operational automation and supervised coding assistance; do not grant an agent production execution authority.

REVISIT WHEN

Reassess for a repeatable maintenance task with an isolated clone, restricted tools, rollback and human-reviewed acceptance criteria.

25 / 47 WORKING RESEARCH SELECTION

FastAPI

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

A model-powered service must accept requests, validate data and handle failures. FastAPI provides a concrete Python example for examining validation, concurrency and application boundaries.

WHERE TO BE CAREFUL

It is a general framework rather than an AI-specific competency. Its inclusion does not establish superiority, and it does not replace a specialized inference engine.

SIMPLER ALTERNATIVES

Use a framework the team knows and separate the application interface from model serving. A broader AI-service competency can use FastAPI as one example.

Endpoint workload

Is the endpoint waiting, blocking or computing?

- | | |
|---|--|
| <p>01 Waiting with an asynchronous library</p> <p>With an await-compatible library, an async def endpoint can yield to other work.</p> | <p>02 Blocking input/output</p> <p>FastAPI runs a regular def endpoint in a thread pool.</p> |
| <p>03 CPU-heavy work (CPU-bound)</p> <p>Asynchrony alone does not make computation faster; separate parallel execution may be needed.</p> | |

Decision map based on the official Concurrency and async / await guide. It combines endpoint declaration advice with concurrency versus parallelism; it is not a taxonomy of frameworks.

SOURCE NOTES / TAXONOMY

[122] Concurrency and async / await

In a hurry?; Concurrency + Parallelism: Web + Machine Learning; Very Technical Details: Path operation functions and Other utility functions.

Assessment evidence (abbreviated): [123](#), [124](#), [125](#), [126](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

FastAPI

CONFIDENCE / MODERATE

Existing packet evidence: FastAPI documentation explains concurrency, and the captured production postmortem shows that handler and thread-pool choices matter for a particular backend. These establish a useful learning example, not framework superiority; the bounded trial and deferral are editorial scope choices.

Evidence: [122] [fastapi-concurrency](#), [124] [ev-02](#)

AI application builder

Run a learning trial

Use one endpoint to learn the service boundary

LEARNING OUTCOMES / CHECKS

01 / Build and test a small API adapter

Check: Implement one local endpoint that validates input and calls a mocked model API. Test invalid payloads, upstream failure and concurrent slow calls, then explain whether its sync or async handlers fit the libraries used.

WHY THIS SCOPE

Python and HTTP are already assumed, so a small adapter is a manageable next step for an API-based application. The framework is one concrete way to study validation and concurrency, but the scenario does not require a migration to it.

STOP AT THIS BOUNDARY

Keep the exercise local and framework-specific; skip adoption if the existing application framework already meets the service requirements.

REVISIT WHEN

Extend study when this adapter is selected for a real service or profiling exposes request-handling failures.

Read alongside: [Structured LLM Outputs](#), [LLM Inference Serving](#)

Model platform operator

Defer with a trigger

Add a custom API layer only when the service contract needs it

No active learning outcome in this profile.

WHY THIS SCOPE

The operator's immediate task is evaluating and hosting an inference engine. A separate Python web layer is not required by the scenario and could add a second concurrency and failure boundary.

STOP AT THIS BOUNDARY

This defers FastAPI-specific implementation; request validation, API compatibility and service reliability remain necessary.

REVISIT WHEN

Reassess when a required control-plane or adapter endpoint cannot be met by the selected serving interface.

Model Context Protocol

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

MCP provides a common way to expose tools and data to model applications. The collected Uber account describes an MCP gateway with permission checks and caller attribution.

WHERE TO BE CAREFUL

A common protocol does not guarantee security or client interoperability. Malicious tool descriptions and excessive permissions remain risks; independent failure data is limited.

SIMPLER ALTERNATIVES

A few stable integrations may need only direct APIs. MCP can be a specific case within tool integration rather than a separate protocol-level competency.

Server-provided primitive

What does the server expose to the application?

- 01

Prompts

Reusable interaction templates selected by the user in the described control model.
- 02

Resources

Contextual content and data whose inclusion is managed by the client application.
- 03

Tools

Functions the model can choose to retrieve information or perform actions.

From the Server Features / Overview table in MCP specification 2025-06-18. This covers that version's server primitives, not every protocol feature or an authorization guarantee.

SOURCE NOTES / TAXONOMY

[127] Model Context Protocol - Server Features / Overview

Primitive / Control / Description / Example table; pinned specification version 2025-06-18.

Assessment evidence (abbreviated): 128, 129, 130, 131. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Model Context Protocol

CONFIDENCE / MODERATE

Existing packet evidence: the pinned 2025-06-18 specification defines protocol primitives, and MCP-AgentBench evaluates tasks through selected operational servers. Deferral follows the explicit two-direct-API scenario; it is not a claim about the latest protocol version, security guarantees or general adoption.

Evidence: [130] ev-03, [128] ev-01

AI application builder

Defer with a trigger

Revisit MCP when an integration requires it

No active learning outcome in this profile.

WHY THIS SCOPE

The scenario explicitly uses two direct API integrations and does not require MCP. Learning and operating an additional protocol layer has no demonstrated task benefit within that boundary.

STOP AT THIS BOUNDARY

Defer MCP implementation, not general tool contracts, authorization or error handling.

REVISIT WHEN

Reassess when a required partner exposes MCP or repeated integrations justify a comparison with direct APIs; check the relevant specification version then.

Read alongside: [AI Agent Design](#), [AI Data Security](#)

Model platform operator

Outside this profile

Tool interoperability is outside model hosting

No active learning outcome in this profile.

WHY THIS SCOPE

This operator profile does not integrate tools or host an agent gateway. Model inference endpoints and MCP tool servers serve different responsibilities in the defined scenario.

STOP AT THIS BOUNDARY

No MCP client, server or gateway implementation is part of the three-month backend learning scope.

REVISIT WHEN

Reassess if hosting a shared MCP gateway with explicit identity, authorization and support obligations is assigned to the platform.

Multi-Agent Coordination Patterns

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Dividing retrieval, analysis, execution and review creates decisions about handoffs, shared state and recovery. The collected material treats coordination as engineering rather than simply multiple conversations.

WHERE TO BE CAREFUL

More agents add communication, latency and context-loss risks, and can share the same error. Matched-budget single-agent comparisons remain important.

SIMPLER ALTERNATIVES

Compare with one agent or a fixed workflow. Teach criteria for splitting work, not multiplying agents by default.

Coordination topology

How are work and information distributed?

- 01

Star

One coordinator assigns work and collects results.
- 02

Tree

Coordinators delegate through a hierarchy of sub-coordinators.
- 03

Graph or mesh (Graph-Mesh)

Agents communicate directly and can plan in parallel.
- 04

Chain

Agents pass results or decisions along a defined sequence.

Four MultiAgentBench configurations from section 3.1.1 and figure 3. The authors classify star/tree as centralized and graph/chain as decentralized within their framework.

SOURCE NOTES / TAXONOMY

[132] MultiAgentBench: Evaluating the Collaboration and Competition of LLM agents

Section 3.1.1 Coordination Engine; figure 3(a); full HTML text of v1.

Assessment evidence (abbreviated): 133, 134, 135, 136. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Multi-Agent Coordination Patterns

CONFIDENCE / MODERATE

Existing packet evidence: MultiAgentBench compares coordination topologies; DPBench exposes failures under simultaneous resource decisions. These establish coordination as a distinct problem, while the decision to defer follows the scenario's lack of a demonstrated need for multiple agents.

Evidence: [134] ev-02, [135] ev-03

AI application builder

Defer with a trigger

Wait until one workflow has a proven coordination limit

No active learning outcome in this profile.

WHY THIS SCOPE

The current document application and two integrations do not establish a need for multiple autonomous participants. Additional handoffs and shared state create failure modes before any workload-specific benefit has been demonstrated.

STOP AT THIS BOUNDARY

Ordinary parallel API calls and separate deterministic workers do not require a multi-agent curriculum.

REVISIT WHEN

Reassess when distinct task responsibilities cannot be handled adequately by one agent or fixed workflow and a matched-budget comparison can be run.

Model platform operator

Outside this profile

Agent coordination is not request scheduling

No active learning outcome in this profile.

WHY THIS SCOPE

The operator schedules inference resources and observes service behavior, but does not assign application tasks to agents. Multi-agent coordination therefore sits outside the stated backend role.

STOP AT THIS BOUNDARY

Inference batching, concurrency and distributed serving remain separate platform competencies.

REVISIT WHEN

Reassess if the platform starts operating a shared agent orchestration service with owned handoff and state semantics.

Read alongside: [AI Agent Design](#), [Agentic Planning & Task Decomposition](#)

28 / 47 WORKING RESEARCH SELECTION

Agent Evaluation

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

A correct final answer can hide an incorrect intermediate action. Evaluate tool choices, data changes, policy compliance and outcomes. The collected LinkedIn account combines pre-release checks with human reviews and usage signals.

WHERE TO BE CAREFUL

An aggregate score can hide rare, costly failures. Public benchmarks cover limited situations, and company accounts do not expose all evaluation data.

SIMPLER ALTERNATIVES

For a fixed workflow, integration tests and manual review may suffice. A separate competency becomes useful when the agent chooses its own actions.

Success across repeated attempts

Are we measuring success once, at least once, or every time?

- | | |
|--|--|
| <p>01 Single-attempt success (pass^1)
The probability that one run completes the task.</p> | <p>02 At least one success ($\text{pass}@k$)
At least one of k independent attempts succeeds.</p> |
| <p>03 Success every time (pass^k)
All k attempts must succeed: a stricter reliability requirement.</p> | |

Metrics from tau-bench, section 3. They compare k independent repetitions per task and average over tasks. $\text{pass}^1 = \text{pass}@1$ is the shared single-attempt case, not a third independent metric. This is one slice of agent evaluation.

SOURCE NOTES / TAXONOMY

[137] tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains

Section 3: Reward and Pass^k metric; the user-confirmation limitation immediately before the metric definitions.

Assessment evidence (abbreviated): 108, 138, 139, 140, 141, 110. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Agent Evaluation

CONFIDENCE / MODERATE

Existing packet evidence: tau-bench tests policy-following tool interactions and repeated-attempt reliability; the benchmark audit identifies weaknesses in reward and task design. Requiring action-level checks for this application's limited tools is an editorial inference, not a measured benefit in this application.

Evidence: [139] ev-03, [141] ev-05

AI application builder

Cover the essentials

Check what the tool interaction actually changed

LEARNING OUTCOMES / CHECKS

01 / Evaluate outcomes and intermediate actions

Check: Replay a small set of valid, ambiguous and unauthorized requests. Inspect tool arguments and resulting state, count policy violations separately from task success, and repeat cases to expose inconsistent behavior.

WHY THIS SCOPE

The application can perform limited actions through two integrations, so a plausible final answer is insufficient evidence of success. Learn to check the requested action, permission boundary and actual outcome together.

STOP AT THIS BOUNDARY

Limit the exercise to the application's existing tools; it does not require adopting an agent framework or a public benchmark leaderboard.

REVISIT WHEN

Expand coverage when a new write action, approval boundary or multi-step workflow is introduced.

Read alongside: [AI Agent Design](#), [LLM Testing](#)

Model platform operator

Outside this profile

Agent trajectories belong to the application owner

No active learning outcome in this profile.

WHY THIS SCOPE

This profile operates model endpoints without owning an application's tools or business state. Backend change testing is necessary, but end-to-end agent policy evaluation needs an application-specific harness and owner.

STOP AT THIS BOUNDARY

Provide model version and request diagnostics to evaluation teams; do not infer agent safety from a serving benchmark.

REVISIT WHEN

Reassess if the platform takes responsibility for a shared agent runtime or tool-execution service.

29 / 47 WORKING RESEARCH SELECTION

Evaluation Data Engineering

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Model comparisons need representative examples, explicit criteria, dataset versions and reproducible execution. The collected Airbnb account describes this infrastructure in product development.

WHERE TO BE CAREFUL

A reproducible test can consistently measure the wrong thing. Contamination can inflate results, and the topic overlaps application testing.

SIMPLER ALTERNATIVES

Start with a small, well-described versioned set. Keep data quality and maintenance distinct from using tests to make release decisions.

Purpose of an example

Which decision should these data support?

- | | |
|---|--|
| <p>01 Failure discovery</p> <p>Inspect responses and traces to identify what is actually failing.</p> | <p>02 Judge calibration (Golden dataset)</p> <p>Expert-rated examples test agreement with human criteria; bad answers are needed too.</p> |
| <p>03 Regression and emerging-failure checks</p> <p>Repeat evaluation after changes and add cases based on product behavior.</p> | |

Editorially extracted data roles from Airbnb sections 1.1, 2.2.1 and 4. The authors do not present three mutually exclusive dataset types.

SOURCE NOTES / TAXONOMY

[013] Eval-driven development: Lessons from evaluating GenAI at scale

1.1 The one rule; 2.2.1 Calibration; 4 A practical walkthrough.

Assessment evidence (abbreviated): **013**, **025**, **142**, **143**. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Evaluation Data Engineering

CONFIDENCE / STRONG

Existing packet evidence: HELM structures scenarios, adaptations and multiple metrics; Airbnb describes task-specific criteria and human-calibrated evaluation infrastructure. The foundational learning scope is an editorial inference from both profiles' change-evaluation duties, not independent validation of Airbnb's outcome claims.

Evidence: [142] [ev-evaluation-data-engineering-3](#), [013] [ev-evaluation-data-engineering-1](#)

AI application builder

Cover the essentials

Build a versioned set of answerable and failing cases

LEARNING OUTCOMES / CHECKS

01 / Make application evaluations reproducible

Check: Create a small versioned set of representative questions, source snapshots and acceptance criteria, including stale evidence, missing answers and tool failures. Keep development examples separate and rerun one comparison from its recorded inputs.

WHY THIS SCOPE

The application must distinguish a real improvement from a changed question set or changed documents. Basic unit tests do not supply representative language examples or stable human judgments.

STOP AT THIS BOUNDARY

Start with a maintainable application set; no organization-wide evaluation platform or claimed statistically representative sample is required.

REVISIT WHEN

Expand coverage when new users, document types, languages or observed failures fall outside the recorded cases.

Read alongside: [LLM Testing](#), [LLM-as-Judge](#)

Model platform operator

Cover the essentials

Preserve the workload behind every engine comparison

LEARNING OUTCOMES / CHECKS

01 / Version the backend comparison inputs

Check: Record a representative workload's provenance, prompt and output lengths, concurrency, decoding settings and quality checks. Run two configurations against the same version and make a second person able to reproduce the comparison.

WHY THIS SCOPE

The scenario includes an engine-comparison environment and a representative workload, so reproducible input management is immediately useful. Different request mixes or scoring rules can otherwise be mistaken for a serving improvement.

STOP AT THIS BOUNDARY

Manage backend evaluation data and application-supplied acceptance cases; do not invent domain labels on behalf of application owners.

REVISIT WHEN

Revise the workload when traffic shape, model family, service requirements or client task mix materially changes.

Hallucination Detection

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Confident answers can contain unsupported claims. The competency breaks answers into checkable statements and examines their support; the collected research compares several signals and human judgments.

WHERE TO BE CAREFUL

Detectors also fail, especially across domains or with incomplete sources. Agreement between models does not establish truth; important-error recall and false positives need measurement.

SIMPLER ALTERNATIVES

Constrain answers to verifiable data, require source support or escalate difficult cases to a person. This may fit within RAG evaluation.

Response-comparison signal

How are inconsistencies between generated samples detected?

- | | |
|--|---|
| 01 Semantic similarity (BERTScore)
Compare sentence representations rather than requiring identical words. | 02 Question answering (Question-Answering)
Check whether different versions support similar answers to questions about a claim. |
| 03 Repeated language fragments (n-gram)
Use statistical occurrence of words or word sequences. | 04 Contradiction detection (NLI)
A separate model evaluates the relationship between a sentence and comparison text. |
| 05 Language-model assessment (LLM-Prompting)
Ask a model whether a sentence is supported by the comparison sample. | |

Five variants documented in the SelfCheckGPT authors' repository. These are self-consistency methods, not a complete hallucination-detection taxonomy or independent verification of all world knowledge.

SOURCE NOTES / TAXONOMY

[144] SelfCheckGPT - authors' repository

README: five variants; SelfCheckGPT Usage: BERTScore, QA, n-gram, NLI and LLM Prompt.

Assessment evidence (abbreviated): [145](#), [146](#), [147](#), [148](#), [149](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Hallucination Detection

CONFIDENCE / MODERATE

Existing packet evidence: FactScore evaluates atomic factual support and RAGTruth supplies human-annotated grounded outputs; SelfCheckGPT uses consistency signals with domain limits. The recommendation distinguishes claim verification from a detector guarantee and is an editorial application of those methods.

Evidence: [146] ev-02, [147] ev-03, [145] ev-01

AI application builder

Cover the essentials

Check factual claims against their source passages

LEARNING OUTCOMES / CHECKS

01 / Label supported and unsupported claims

Check: Break a small set of document answers into checkable claims and label each against the cited passages. Compare one automatic signal with those labels and inspect both missed errors and false alarms before setting an escalation rule.

WHY THIS SCOPE

A document application can produce a fluent answer whose claims are not supported by its retrieved evidence. Learning to verify support is required even when an automatic detector will not be deployed.

STOP AT THIS BOUNDARY

Judge support from the available source, not all-world truth; neither repeated agreement nor a citation guarantees correctness.

REVISIT WHEN

Expand detector calibration when document domains, languages or the consequences of an unsupported claim change.

Read alongside: [RAG Evaluation](#), [LLM-as-Judge](#)

Model platform operator

Run a learning trial

Try a grounded factuality check in model-change evaluation

LEARNING OUTCOMES / CHECKS

01 / Test a detector against human-labeled backend cases

Check: Use a small set of supplied passages and human-labeled answers to compare two backend configurations. Check an automatic detector's missed unsupported claims and false alarms, and keep uncertain cases for human review.

WHY THIS SCOPE

Changing a model or its serving configuration should not be judged only by latency and throughput. A small grounded check can inform change review, while domain-wide factual assurance remains with the application owner.

STOP AT THIS BOUNDARY

This is an offline comparison with supplied references, not a universal runtime truth filter or ownership of document retrieval.

REVISIT WHEN

Expand only when the supported service contract includes factual-quality checks and task owners can maintain the labels.

LLM Testing

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Changing a model or prompt can improve some outputs and regress others. Application tests should detect these changes before release; the collected Airbnb account provides a product-development example.

WHERE TO BE CAREFUL

Large test suites can measure the wrong behavior. Many language tasks lack a single correct answer, and the scope overlaps agent evaluation and evaluation data.

SIMPLER ALTERNATIVES

Start with representative cases, automatic format checks and human review. Keep this entry focused on product-change testing rather than autonomous action traces.

Test expectation

What must remain true after running the model?

01 Minimum functionality (Minimum Functionality Test)

A simple case checks a specific capability against a label or condition.

02 Invariance (Invariance Test)

A task-irrelevant input change should preserve the assessed behavior within tolerance.

03 Directional expectation (Directional Expectation Test)

A meaningful input change should move the result in an expected direction.

Three CheckList test types, checked against the authors' README and MFT, INV and DIR class definitions in the original atlas. Applying this NLP framework to LLM applications requires task-specific criteria.

SOURCE NOTES / TAXONOMY

[150] CheckList - authors' repository

README: Creating and running tests; MFT, INV and DIR examples.

[151] CheckList - test_types.py

Definitions and docstrings of MFT, INV and DIR; the original review read the full file.

Assessment evidence (abbreviated): 013, 025, 152, 153, 154. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

LLM Testing

CONFIDENCE / STRONG

Existing packet evidence: CheckList defines behavioral expectations and Airbnb describes task-specific regression controls and repeatable evaluation runs. Extending basic testing to model changes is an editorial recommendation; the sources do not guarantee defect-free releases or validate this exact test suite.

Evidence: [152] ev-03, [013] ev-01, [025] ev-02

AI application builder

Extend

Extend basic tests to prompt and model regressions

LEARNING OUTCOMES / CHECKS

01 / Test behavior across a controlled application change

Check: Add format checks and task-specific behavioral cases to an existing test suite. Change one prompt or model, repeat variable cases, and explain which failures block release using recorded expected behavior rather than exact prose matching.

WHY THIS SCOPE

Basic testing is an entry assumption, but language behavior can change without a code exception. The next scope is regression detection across prompts, models and document-dependent responses.

STOP AT THIS BOUNDARY

Reuse the existing test workflow; a large evaluation platform or an exact-output oracle for every answer is unnecessary.

REVISIT WHEN

Add cases when a user-visible failure escapes the suite or a new behavior becomes part of the application contract.

Read alongside: [Evaluation Data Engineering](#), [Agent Evaluation](#)

Model platform operator

Extend

Add output-contract checks to serving changes

LEARNING OUTCOMES / CHECKS

01 / Gate a backend change on behavior and service checks

Check: Compare two backend configurations with recorded model, tokenizer, template and decoding settings. Check API errors, output constraints and application-supplied behavioral cases, then demonstrate a rollback decision when a required check regresses.

WHY THIS SCOPE

Infrastructure operations and basic testing are assumed, but backend changes can affect response behavior as well as service health. Extend the existing release process with a reproducible model-output contract.

STOP AT THIS BOUNDARY

Use client-owned cases for domain correctness; the operator owns repeatability, configuration identity and service-level acceptance.

REVISIT WHEN

Repeat the relevant suite when the model, engine, tokenizer, prompt template or numerical representation changes.

LLM-as-Judge

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Models can help compare large volumes of responses against explicit criteria and identify cases for human review. Studies report useful human agreement in some tasks.

WHERE TO BE CAREFUL

Judges can favor styles, positions or answers resembling their own. Agreement may not transfer across domains and cannot replace defining what a good result means.

SIMPLER ALTERNATIVES

Use executable tests or deterministic rules when possible. For qualitative judgments, combine expert samples with ongoing model calibration.

Judgment format

What is the model asked to judge?

01 Single-answer grading (Single answer grading)
Rate one answer against specified criteria.

02 Pairwise comparison
Choose the better response or declare a tie.

Reference availability

Does the judge receive a reference solution?

01 With a reference (Reference-guided grading)
A supplied solution, such as a math answer, supports the judgment.

02 Without a reference
Judge from the instruction, criteria and response without a supplied solution.

MT-Bench section 3.1 presents three combinable judge variants. This editorial two-axis view avoids suggesting that reference-guided grading excludes pairwise comparison.

SOURCE NOTES / TAXONOMY

[155] Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena

Section 3.1 Types of LLM-as-a-Judge; section 3.3 Position bias; appendix A, figures 4-9: judgment templates.

Assessment evidence (abbreviated): 156, 157, 158, 159, 160. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

LLM-as-Judge

CONFIDENCE / MODERATE

Existing packet evidence: MT-Bench compares model judgments with human preferences, and controlled order experiments expose positional bias. These support calibration and order checks; an editorial trial does not establish factual or safety validity for the reader's task.

Evidence: [160] ev-05, [159] ev-04

AI application builder

Run a learning trial

Calibrate one judge before relying on its scores

LEARNING OUTCOMES / CHECKS

01 / Compare a narrow rubric with human judgments

Check: Write one explicit answer-quality rubric and human-label a small sample. Compare model judgments, reverse pairwise answer order, and inspect disagreements by error type rather than reporting agreement alone.

WHY THIS SCOPE

Open-ended document answers can be expensive to review repeatedly. A calibrated judge may help select cases for review, but its usefulness must be checked against the application's criteria.

STOP AT THIS BOUNDARY

Use deterministic checks for executable properties and retain human review for consequential or disputed judgments.

REVISIT WHEN

Expand automation only when task-level disagreement and important-error rates are acceptable and can be monitored after judge changes.

Read alongside: [Evaluation Data Engineering](#), [Hallucination Detection](#)

Model platform operator

Run a learning trial

Use a calibrated judge as one model-comparison signal

LEARNING OUTCOMES / CHECKS

01 / Audit judging in a backend comparison

Check: Compare outputs from two configurations on an application-supplied rubric and human-labeled sample. Blind configuration names, swap answer order and record disagreements alongside deterministic checks and service metrics.

WHY THIS SCOPE

Model changes may alter response usefulness even when operational metrics improve. A bounded judge experiment can add a quality signal, provided task owners supply criteria and human reference judgments.

STOP AT THIS BOUNDARY

The operator validates the comparison process, not a universal definition of answer quality or an automatic release veto based solely on a judge score.

REVISIT WHEN

Continue when clients provide stable rubrics and enough reviewed disagreements to detect judge drift.

RAG Evaluation

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Evaluate both retrieval and how generated answers use the retrieved information. This distinguishes source, passage-selection and generation failures; the collected Uber account uses domain-expert questions and answers.

WHERE TO BE CAREFUL

Automated metrics can disagree with people and depend on documents and question types. One deployment does not establish transfer, and the topic overlaps broader application testing.

SIMPLER ALTERNATIVES

Begin with manual analysis of important questions and failures. Keep a separate entry focused on end-to-end diagnosis rather than listing metric libraries.

Three quality questions

Which relationship between question, context and answer fails?

01 Faithfulness

Are answer claims supported by the supplied context?

02 Answer relevance

Does the answer address the question rather than a side topic?

03 Context relevance

Are retrieved passages useful rather than irrelevant context overhead?

Three independent aspects from section 3 of the original RAGAS paper, v1. This is the historical paper framework, not the full metric set of the current Ragas library.

SOURCE NOTES / TAXONOMY

[161] RAGAS: Automated Evaluation of Retrieval Augmented Generation

Section 3 Evaluation Strategies; Faithfulness, Answer relevance and Context relevance subsections.

Assessment evidence (abbreviated): 162, 163, 164, 060, 165, 166. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

RAG Evaluation

CONFIDENCE / STRONG

Existing packet evidence: RAGChecker separates retrieval and generation diagnostics, while ARES calibrates automated judgments with human-labeled data. These support component-level measurement; making it foundational for a document application is an editorial inference, not a guarantee that any metric matches its users.

Evidence: [060] ev-04, [164] ev-03

AI application builder

Cover the essentials

Separate a missed source from a misused source

LEARNING OUTCOMES / CHECKS

01 / Diagnose retrieval and generation separately

Check: Use a human-reviewed question set with expected supporting passages. Record whether retrieval found support, whether the answer used it faithfully and whether unsupported questions were handled correctly; inspect disagreements with automatic scores.

WHY THIS SCOPE

A document application cannot be improved reliably if every incorrect answer is treated as a model problem. The reader needs to distinguish missing evidence, poor passage selection and unsupported generation before changing components.

STOP AT THIS BOUNDARY

Start with a small representative set and human review; no single RAG score should stand in for all quality dimensions.

REVISIT WHEN

Revisit the evaluation set when ingestion, retrieval, prompt design, model choice or document freshness rules change.

Read alongside: [Retrieval-Augmented Generation](#), [Hallucination Detection](#)

Model platform operator

Outside this profile

The application owns retrieval and grounding evaluation

No active learning outcome in this profile.

WHY THIS SCOPE

This backend role does not select sources or construct answer context. It can run supplied acceptance cases during a model change, but cannot diagnose the full retrieval pipeline from inference metrics alone.

STOP AT THIS BOUNDARY

Backend output tests remain in scope; ownership of passage labels, retrieval failures and end-to-end RAG metrics does not.

REVISIT WHEN

Reassess if the platform contract expands to a managed RAG service with retrieval-quality responsibilities.

Agent Threat Modeling (MAESTRO)

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Threat modeling identifies what an agent can access and where control can be lost before safeguards are designed. MAESTRO structures this analysis across system layers; the material includes a network-monitoring prototype.

WHERE TO BE CAREFUL

The collected evidence does not establish that MAESTRO finds more important issues than alternatives at comparable effort. Results depend on reviewers, and the competency is broader than one framework.

SIMPLER ALTERNATIVES

Consider STRIDE adapted to agent-specific risks or scenarios based on observed attacks. MAESTRO can remain an example within general agent threat modeling.

System layer

Where does a threat originate, and where can it propagate?

- 01

1. Foundation models (Foundation Models)
The model, its behavior and its vulnerabilities.
- 02

2. Data operations (Data Operations)
Preparing, storing and supplying data to the agent.
- 03

3. Agent frameworks (Agent Frameworks)
Libraries and mechanisms organizing agent behavior.
- 04

4. Deployment and infrastructure (Deployment and Infrastructure)
The environment running the system.
- 05

5. Evaluation and observability (Evaluation and Observability)
Tests, metrics and tools for inspecting behavior.
- 06

6. Security and compliance (Security and Compliance)
Cross-cutting controls, including agents used for protection.
- 07

7. Agent ecosystem (Agent Ecosystem)
Relationships with applications, other agents and users.

Seven layers from Ken Huang's architecture as described by CSA, section 3.2. Security is cross-cutting; numbering indicates neither execution order nor importance.

SOURCE NOTES / TAXONOMY

[167] Agentic AI Threat Modeling Framework: MAESTRO

Section 3.2, MAESTRO's Elements; figure 1; A. Layer-Specific Threat Modeling and B. Cross-Layer Threats.

Assessment evidence (abbreviated): 168, 169, 170, 167, 171. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Agent Threat Modeling (MAESTRO)

CONFIDENCE / LIMITED

The existing CSA framework and network-monitoring prototype support a structured review of agent threats. They do not establish MAESTRO's superiority or reviewer consistency; the learning trial is an editorial inference for the stated application profile, not a new empirical result.

Evidence: [170] ev-03, [167] ev-04, [168] ev-01

AI application builder

Run a learning trial

Try one structured threat review of the two integrations

LEARNING OUTCOMES / CHECKS

01 / Trace one tool action across data, model and execution boundaries.

Check: Produce an asset-and-permission map, three plausible abuse scenarios and a testable control for each; compare omissions with a basic asset-and-threat review.

WHY THIS SCOPE

The application combines untrusted documents with limited tool actions, so threats can cross component boundaries. A short MAESTRO exercise can reveal useful omissions, but available evidence does not justify mastering this framework before ordinary threat modeling.

STOP AT THIS BOUNDARY

Use MAESTRO as an optional review structure; it neither certifies security nor replaces authorization tests.

REVISIT WHEN

Reassess when tool permissions, persistent memory or connections to other agents expand.

Read alongside: [AI Guardrails](#), [AI Red Teaming](#)

Model platform operator

Outside this profile

Agent-specific modeling belongs to the application owner

No active learning outcome in this profile.

WHY THIS SCOPE

This profile runs a model backend and does not own agent planning or tool integrations. Keep infrastructure threat modeling in the platform remit and provide the application team with clear service boundaries.

STOP AT THIS BOUNDARY

This excludes agent-specific MAESTRO study, not ordinary platform security review.

REVISIT WHEN

Reassess if the platform begins hosting agent runtimes, tools or shared agent memory.

35 / 47 WORKING RESEARCH SELECTION

AI Data Security

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

AI applications can expose data through answers, retrieval, memory or tools. The competency traces these paths and limits access, including sensitive attributes inferred rather than copied verbatim.

WHERE TO BE CAREFUL

The title overlaps established data-security practices. The collected material covers threats and experiments better than complete operational protection, deletion and incident handling.

SIMPLER ALTERNATIVES

Start with data minimization, ordinary access controls and auditing. A narrower map could combine this with Secure RAG while retaining model-memory and inference risks.

What can be disclosed?

Which asset is exposed?

- | | |
|---|--|
| <p>01 Personal information (PII Leakage)
An answer exposes information identifying an individual.</p> | <p>02 Proprietary technical information (Proprietary Algorithm Exposure)
Proprietary algorithms or model-related data are revealed.</p> |
| <p>03 Confidential business information (Sensitive Business Data Disclosure)
A recipient receives business information they should not access.</p> | |

Three disclosure examples from OWASP LLM02:2025, Common Examples of Vulnerability. This covers confidentiality, not a complete data-security taxonomy including integrity and availability.

SOURCE NOTES / TAXONOMY

[172] OWASP LLM02:2025 Sensitive Information Disclosure

Common Examples of Vulnerability, items 1-3; Prevention and Mitigation Strategies: Access Controls; User Education and Transparency.

Assessment evidence (abbreviated): [173](#), [174](#), [172](#), [175](#), [176](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

AI Data Security

CONFIDENCE / MODERATE

Existing OWASP guidance identifies disclosure paths and control categories, while privacy-inference experiments show that removing explicit identifiers may not remove all disclosure risk. The profile priorities are editorial judgments; these sources do not prove an end-to-end protection level.

Evidence: [173] ev-01, [172] ev-03, [175] ev-04

AI application builder

Cover the essentials

Control which data can reach context, logs and tools

LEARNING OUTCOMES / CHECKS

01 / Map and test the application's sensitive-data paths.

Check: Trace a restricted document through retrieval, prompts, output, logs and tool arguments; show where access is enforced and verify that a different user cannot retrieve its content.

WHY THIS SCOPE

Document processing can disclose information through more than the final answer. The builder needs a concrete account of permitted data flows before adding retrieval or tool capabilities.

STOP AT THIS BOUNDARY

Prioritize minimization, access checks and retention decisions; a redaction prompt is not an access-control mechanism.

REVISIT WHEN

Repeat the review for a new data class, integration, logging destination or retention policy.

Read alongside: [Secure RAG](#), [AI Guardrails](#)

Model platform operator

Cover the essentials

Protect prompts, outputs and operational traces at the service boundary

LEARNING OUTCOMES / CHECKS

01 / Define and verify the model service's data-handling contract.

Check: Document who can inspect requests and logs, verify the intended user or tenant isolation model, and verify the configured retention and deletion behavior on test data.

WHY THIS SCOPE

A self-hosted endpoint and its telemetry can expose customer inputs even without application-level retrieval. Platform ownership therefore includes access boundaries and explicit handling of sensitive operational data.

STOP AT THIS BOUNDARY

Application document permissions remain with the application owner; infrastructure controls do not establish answer-level confidentiality.

REVISIT WHEN

Reassess when tenancy, logging, support access or storage changes.

AI Guardrails

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Applications that read data or invoke tools need controls on allowed behavior: input/output checks, permissions and human approval. The collected Grab account describes multiple layers in a working product.

WHERE TO BE CAREFUL

Adding a filter does not make a system safe. Controls can be bypassed, and the account does not isolate each layer's effectiveness, false-positive rate or latency.

SIMPLER ALTERNATIVES

Use ordinary authorization, code-based validation and human approval for consequential operations. Preserve the distinction between runtime controls and broader security design.

Intervention point

At which stage is a check or block applied?

- 01

Input (Input rails)
Inspect a message before it reaches the model, for example to remove sensitive data.
- 02

Retrieval (Retrieval rails)
Check documents and passages before they enter model context.
- 03

Dialogue (Dialog rails)
Enforce conversation rules and sequencing, including across turns.
- 04

Execution (Execution rails)
Check tool calls, arguments and results.
- 05

Output (Output rails)
Inspect or transform a response before it reaches the user.

Five types from the NeMo Guardrails Guardrail Types table. This groups one tool's functions by intervention point, not all AI security. Types can be combined.

SOURCE NOTES / TAXONOMY

[177] NeMo Guardrails - Guardrail Types

Stage / Rail Type / Common Use Cases table; Use Cases and Applicable Rails. Original full-text review: 2026-09-07.

Assessment evidence (abbreviated): 178, 179, 180, 181, 182, 183. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

AI Guardrails

CONFIDENCE / MODERATE

The collected NeMo work explains intervention points, Grab documents layered controls in one operating application, and JailbreakBench demonstrates the need for adversarial validation. The recommendations are editorial scope decisions; no source isolates a universal guardrail effectiveness rate.

Evidence: [179] ev-02, [180] ev-03, [183] ev-06

AI application builder

Cover the essentials

Enforce tool permissions and validate consequential outputs

LEARNING OUTCOMES / CHECKS

01 / Build a testable control path around one tool action.

Check: Show that an unauthorized action is blocked in code, malformed arguments fail validation, and the required human approval cannot be bypassed by model output; also test legitimate requests.

WHY THIS SCOPE

The model may propose actions that the application must not execute. Runtime controls need to enforce permissions and contracts while tests reveal both missed violations and unnecessary blocks.

STOP AT THIS BOUNDARY

Start with authorization and deterministic validation; add learned filters only for a defined policy and measured trade-off.

REVISIT WHEN

Review the controls when a tool gains write access or policy, model or input sources change.

Read alongside: [AI Data Security](#), [AI Red Teaming](#)

Model platform operator

Run a learning trial

Test where shared model-safety checks should live

LEARNING OUTCOMES / CHECKS

01 / Compare one candidate endpoint safety check with application-side enforcement.

Check: Use an agreed set of prohibited and legitimate requests, record missed cases, unnecessary blocks and added latency, and document the responsibilities that remain with applications.

WHY THIS SCOPE

A shared endpoint can centralize some model-level policies, but applications have different contexts and permissions. A bounded trial can establish whether a common check adds value without assuming that a platform filter secures every downstream application.

STOP AT THIS BOUNDARY

Do not centralize application-specific tool authorization or treat refusal behavior as complete security.

REVISIT WHEN

Reassess when multiple tenants request the same policy or a model change alters refusal and false-positive behavior.

37 / 47 WORKING RESEARCH SELECTION

AI Red Teaming

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Red teaming deliberately seeks unwanted answers or actions rather than only testing intended behavior. Company accounts and research provide concrete diagnostic methods.

WHERE TO BE CAREFUL

A striking bypass does not establish its frequency in normal use. Teams test different threats, and evidence connecting exercises to fewer later incidents is limited.

SIMPLER ALTERNATIVES

Retain architecture reviews and conventional security testing. Include remediation and retesting, not only a collection of successful attacks.

Scenario description

What is needed to understand and reproduce an issue?

- | | |
|--|---|
| 01 System
The model or application and its connections. | 02 Actor
The person whose behavior is modeled, with malicious or benign intent. |
| 03 Tactics, techniques and procedures (TTPs)
The objective, method and reproducible sequence of steps. | 04 Weakness
The property enabling the unwanted result. |
| 05 Impact
Consequences for a user or system, not just an unusual answer. | |

Five components from Microsoft's AIRT ontology in Lessons From Red Teaming 100 Generative AI Products, section 1.2 and figure 1. They describe one scenario, not mutually exclusive attack classes.

SOURCE NOTES / TAXONOMY

[184] Lessons From Red Teaming 100 Generative AI Products

Section 1.2, AI threat model ontology; figure 1; Lesson 3 distinguishes red teaming from safety benchmarking.

Assessment evidence (abbreviated): 185, 186, 187, 188, 189. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

AI Red Teaming

CONFIDENCE / MODERATE

The existing Microsoft retrospective and NIST ARIA pilot support scoped adversarial exercises alongside other testing. The research critique limits claims about coverage and incident prevention; the learning priorities below are editorial, not measured reductions in deployment risk.

Evidence: [185] ev-01, [186] ev-02, [189] ev-05

AI application builder

Extend

Add adversarial scenarios to the basic test practice

LEARNING OUTCOMES / CHECKS

01 / Run and close a small adversarial review of documents and tool actions.

Check: Define the attacker permissions, reproduce one plausible unwanted action or disclosure attempt, implement a mitigation, and retain a regression test plus a legitimate-use counterexample.

WHY THIS SCOPE

Ordinary happy-path tests do not explore how document content could redirect a tool-enabled application. Add a scoped exercise that ends with remediation and retesting rather than only a list of successful attacks.

STOP AT THIS BOUNDARY

An exercise reveals weaknesses under stated conditions; it does not estimate incident frequency or certify safety.

REVISIT WHEN

Repeat for a materially new tool, data source, trust boundary or observed abuse pattern.

Read alongside: [AI Guardrails](#), [Agent Threat Modeling \(MAESTRO\)](#)

Model platform operator

Run a learning trial

Pilot a model-release adversarial check with service owners

LEARNING OUTCOMES / CHECKS

01 / Define a bounded adversarial evaluation for a candidate model release.

Check: Agree a permitted test scope, compare the candidate and current model on relevant misuse scenarios, assign each finding to a service or application owner, and retest one resolved finding.

WHY THIS SCOPE

Changing a hosted model can change unwanted behavior even if the serving API stays stable. A limited joint exercise teaches useful release checks without making the operator responsible for every application's threat model.

STOP AT THIS BOUNDARY

Stay within the authorized test environment and the platform's service responsibilities.

REVISIT WHEN

Expand the scope when model behavior, tenant exposure or the platform's own action permissions change.

Secure RAG

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Document-based answers must respect user permissions even when paraphrased. Source integrity also matters; the collected Uber account describes access controls, query checks, controlled export and change tests.

WHERE TO BE CAREFUL

Adding documents does not prevent attacks. Poisoned content can propagate through a pipeline; comprehensive comparisons of permissions, updates, leakage and incident recovery remain limited.

SIMPLER ALTERNATIVES

For sensitive data, consider permission-aware search and excerpts without free-form generation. A combined data-security entry must retain retrieval-specific requirements.

Content manipulation

How does apparently useful material corrupt an answer?

- | | |
|---|---|
| 01 Relevant but incomplete noise (Silver Noise)
Many partial sources limit the evidence diversity needed for a complete answer. | 02 Conflicting context (Inter-Context Conflict)
A planted version of facts plausibly competes with correct context. |
| 03 Embedded promotion (Soft Ad)
Promotional content is woven into apparently credible material. | 04 Induced refusal (White Denial-of-Service)
A false warning about unreliable sources prompts the system to refuse an answer. |

Four SafeRAG classes from sections 3.3.1-3.3.4. This is a content-attack benchmark taxonomy, not a complete account of permissions, leakage or infrastructure security.

SOURCE NOTES / TAXONOMY

[190] SafeRAG: Benchmarking Security in Retrieval-Augmented Generation of Large Language Model

Section 3.3 Selection of Attack-Targeted Texts and Generation of Attacking Texts; four subsections; figure 2: construction process.

Assessment evidence (abbreviated): 191, 192, 193, 188, 194. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Secure RAG

CONFIDENCE / MODERATE

Existing poisoning benchmarks establish plausible retrieval attacks, and Uber's Finch account documents governed sources and role checks in an operating system. Essential application scope is an editorial judgment; the sources do not prove end-to-end tenant isolation or that any filter prevents all poisoning.

Evidence: [191] ev-01, [192] ev-02, [194] ev-05

AI application builder

Cover the essentials

Preserve document permissions through retrieval and answers

LEARNING OUTCOMES / CHECKS

01 / Test retrieval authorization and an untrusted-document case.

Check: Use two test users with different permissions; verify restricted passages never enter the unauthorized context, then inject a misleading or instruction-bearing document and demonstrate the expected containment and review path.

WHY THIS SCOPE

A document application must preserve who may read each source, even when the final answer combines several passages. The builder also needs to distinguish source content from instructions and recognize that a retrieved document may be malicious or incorrect.

STOP AT THIS BOUNDARY

Enforce permissions before content reaches the model; an output refusal or relevance filter does not replace access checks.

REVISIT WHEN

Repeat the review when document permissions, ingestion sources, sharing rules or freshness and deletion behavior change.

Read alongside: [AI Data Security](#), [Retrieval-Augmented Generation](#)

Model platform operator

Outside this profile

Retrieval authorization belongs to the application layer

No active learning outcome in this profile.

WHY THIS SCOPE

The profile explicitly excludes application-level RAG design and does not operate a retrieval corpus. Secure request handling at the model endpoint remains necessary, but it cannot determine which documents a particular user may retrieve.

STOP AT THIS BOUNDARY

Support service-level data isolation without claiming to enforce document permissions that the endpoint does not know.

REVISIT WHEN

Reassess if the platform starts offering a shared retrieval, indexing or document-ingestion service.

39 / 47 WORKING RESEARCH SELECTION

Amazon Bedrock

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Managed AI platforms combine model access, permissions and service operations. Bedrock provides a concrete AWS example; the collected Salesforce account describes a particular custom-model deployment workflow.

WHERE TO BE CAREFUL

One workflow does not establish superiority of the entire platform. Costs, limits, outages and security assessment remain relevant, and a brand-specific entry can age quickly.

SIMPLER ALTERNATIVES

Compare a managed platform with direct provider APIs and self-hosting. A broader managed-platform competency could include multiple providers without declaring a winner.

Response delivery mode

Do you need one complete response, a stream, or a batch?

01 Single response (InvokeModel)

Submit an input and receive a result in the API response.

02 Streamed response

(InvokeModelWithResponseStream)

Receive output fragments during generation when the model supports it.

03 Batch inference

Submit multiple inputs as files and collect results later from S3.

Editorial grouping based on Invoke API and Batch inference documentation. These are result-delivery modes, not a taxonomy of the whole platform or compute-reservation options.

SOURCE NOTES / TAXONOMY

[195] Inference using Invoke API

Submit a single prompt; InvokeModel and InvokeModelWithResponseStream operations.

[196] Process multiple prompts with batch inference

S3 input/output workflow and documented limitations.

Assessment evidence (abbreviated): 197, 198, 199, 200, 201, 202. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Amazon Bedrock

CONFIDENCE / LIMITED

The existing Salesforce and Pushpay accounts establish specific Bedrock workflows, while the cloud-incident study describes risks in the managed-service class without identifying Bedrock as the affected provider. Deferral is an editorial fit judgment because neither profile specifies an AWS requirement; this is not a fresh product or pricing comparison.

Evidence: [202] ev-08, [200] ev-04, [199] ev-03

AI application builder

Defer with a trigger

Wait for an AWS-specific requirement

No active learning outcome in this profile.

WHY THIS SCOPE

The application needs managed model access but has no stated AWS constraint. Learning a particular platform now adds vendor-specific work without resolving an identified capability or integration gap.

STOP AT THIS BOUNDARY

Retain transferable API, evaluation and data-handling skills; do not infer platform superiority from a customer case.

REVISIT WHEN

Reassess when AWS identity, regional controls, an available model or an existing cloud estate makes Bedrock a concrete candidate.

Read alongside: [LLM Inference Serving](#), [AI Guardrails](#)

Model platform operator

Defer with a trigger

Revisit Bedrock when evaluating a managed-service alternative

No active learning outcome in this profile.

WHY THIS SCOPE

The current responsibility is a self-hosted model backend. A vendor-specific managed-service curriculum becomes useful when there is an actual build-versus-buy or overflow-serving decision.

STOP AT THIS BOUNDARY

A future comparison must include service constraints, data handling, failure modes and matched workload costs.

REVISIT WHEN

Reassess if a migration, managed fallback or AWS model-import requirement enters the roadmap.

40 / 47 WORKING RESEARCH SELECTION

Docker

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

AI applications need reproducible environments and dependencies. Containers help move them between development and deployment; the material also covers image size, distribution and model-file storage.

WHERE TO BE CAREFUL

This is an established general software competency, not necessarily a new AI signal. Containers are not automatically sufficient isolation for untrusted code, and the packet does not establish adoption rates.

SIMPLER ALTERNATIVES

Use a managed service or stronger isolation such as a separate VM when appropriate. Reproducible environments may work better as a prerequisite than a brand-specific signal.

Data location and persistence

Who manages files, and should they outlive a container?

- | | |
|---|---|
| <p>01 Writable container layer</p> <p>Writes belong to that container and disappear when it is removed.</p> | <p>02 Volume</p> <p>Docker manages storage that can outlive container removal.</p> |
| <p>03 Bind mount</p> <p>A host path is visible in the container; host processes may also change its files.</p> | <p>04 Temporary memory (tmpfs)</p> <p>Files live in memory and do not persist after the container stops.</p> |

Selected storage decisions from Docker Docs. The source distinguishes the container layer and mount types; specialized image mounts and named pipes are omitted. This is not a taxonomy of all Docker features.

SOURCE NOTES / TAXONOMY

[203] Storage

Container layer basics; Storage mount options: Volume mounts, Bind mounts and tmpfs mounts.

Assessment evidence (abbreviated): [204](#), [205](#), [206](#), [207](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Docker

CONFIDENCE / MODERATE

The existing NIST container guide supports security boundaries, and the startup and OCI-distribution studies show environment-dependent behavior. The essential-versus-maintain distinction is an editorial reading of the application and infrastructure entry assumptions; no deployment-speed or adoption claim is made.

Evidence: [205] [ev-docker-2](#), [204] [ev-docker-1](#), [206] [ev-docker-5](#)

AI application builder

Cover the essentials

Make the OCI application environment reproducible

LEARNING OUTCOMES / CHECKS

01 / Package and run the application with explicit dependencies and secrets handling.

Check: Build from a clean checkout, start with documented configuration, verify the health check, and demonstrate that credentials are supplied at runtime rather than stored in the image.

WHY THIS SCOPE

The scenario already uses OCI, while container competence is not part of the builder's stated entry skills. A small reproducible packaging exercise closes a concrete delivery gap.

STOP AT THIS BOUNDARY

Do not extend this into Kubernetes or treat a normal container as sufficient isolation for hostile code.

REVISIT WHEN

Add scope when native dependencies, image size or deployment failures create a demonstrated problem.

Read alongside: [LLM Inference Serving](#), [AI Data Security](#)

Model platform operator

Maintain

Maintain a reproducible and reviewable container baseline

LEARNING OUTCOMES / CHECKS

01 / Keep the serving image and runtime contract verifiable.

Check: Rebuild and deploy a pinned image in the comparison environment; confirm expected model mounts, permissions, health behavior and rollback to the prior image.

WHY THIS SCOPE

OCI environments are already within this infrastructure role. The scenario identifies no large-image distribution bottleneck, so advanced image-delivery study has no present justification.

STOP AT THIS BOUNDARY

Maintain the foundation; diagnose before studying specialized filesystem or distribution techniques.

REVISIT WHEN

Reassess when measured startup, transfer or cache failures prevent the service meeting its requirements.

41 / 47 WORKING RESEARCH SELECTION

Edge AI

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Some applications need offline operation or must meet device memory, energy and latency limits. Model selection then requires measurement on the target device.

WHERE TO BE CAREFUL

Results from one device may not transfer. Local execution does not remove update, privacy or observability challenges; the collected evidence is mainly controlled testing.

SIMPLER ALTERNATIVES

Compare cloud execution and a split device/server design. A local deployment needs an explicit reason.

Operational responsibility

What must be managed after moving inference onto devices?

- | | |
|--|---|
| 01 Variants and hardware compatibility
Devices require different models, numeric formats and supported operations. | 02 Observability
Measure quality and resource use without automatically collecting all user data. |
| 03 Updates and personalization
Manage versions and, where appropriate, local or federated learning. | 04 Usage metering (Pay-per-query)
Counting calls is harder when a device operates offline. |

Editorial grouping of TinyMLOps features from figure 1 and sections III-IV, not a complete Edge AI taxonomy. Figure 1 was visually checked in the original atlas. The source packet had contained an incorrect paper title for this URL.

SOURCE NOTES / TAXONOMY

[208] TinyMLOps: Operational Challenges for Widespread Edge AI Adoption

Figure 1, PDF p. 2; sections III.A-III.D and IV, PDF pp. 2-4.

Assessment evidence (abbreviated): 209, 210, 211, 212. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Edge AI

CONFIDENCE / MODERATE

Existing edge studies and MLPerf Tiny describe device-specific constraints and measurement. The outside classifications are editorial consequences of the cloud/API and server-hosting scenarios, not a conclusion about the value of edge deployment generally.

Evidence: [\[209\] ev-edge-ai-2](#), [\[210\] ev-edge-ai-3](#), [\[212\] ev-edge-ai-5](#)

AI application builder

Outside this profile

On-device inference is outside this application's scope

No active learning outcome in this profile.

WHY THIS SCOPE

The application uses managed APIs and has no offline or device-execution requirement. Studying thermal, battery and embedded-memory constraints would not support the stated three-month work.

STOP AT THIS BOUNDARY

Local execution is a separate architecture choice and does not automatically provide complete privacy.

REVISIT WHEN

Reassess if offline operation, device latency or a data-locality requirement makes remote inference unsuitable.

Read alongside: [Model Quantization](#), [GPU Acceleration](#)

Model platform operator

Outside this profile

Device fleets are outside this server-hosting role

No active learning outcome in this profile.

WHY THIS SCOPE

The profile operates a self-hosted backend with a representative server workload. Edge distribution, device energy and fleet updates involve a different operating environment.

STOP AT THIS BOUNDARY

Do not transfer datacenter or Raspberry Pi results directly to an unspecified target device.

REVISIT WHEN

Reassess when the platform becomes responsible for models running on end-user or embedded hardware.

42 / 47 WORKING RESEARCH SELECTION

GPU Acceleration

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Faster hardware does not help when a program waits for data or uses memory poorly. This competency diagnoses bottlenecks and matches computation to hardware.

WHERE TO BE CAREFUL

It can be too specialized for API-only teams. Results depend on workloads and hardware, and overlap broader inference optimization.

SIMPLER ALTERNATIVES

First consider a smaller model or less work. A separate entry best serves people profiling or programming accelerators.

Dominant bottleneck

What is the computation actually waiting for?

- | | |
|---|---|
| <p>01 Data movement (Memory-bound)
Memory supplies data too slowly relative to arithmetic capacity.</p> | <p>02 Arithmetic throughput (Compute-bound)
Available compute throughput limits performance at the given arithmetic intensity.</p> |
| <p>03 Implementation underutilization
Poor locality, cache misses or insufficient parallelism keep performance below the attainable bound.</p> | |

Diagnostic map based on Berkeley Lab's Roofline explanation. The first two limits come from the model; the third summarizes below-roofline causes. Categories are not mutually exclusive.

SOURCE NOTES / TAXONOMY

[213] The Roofline Model: Visualizing and Optimizing Performance

Introduction to the Roofline Model; Why Your Performance Might Be Below the Roofline.

Assessment evidence (abbreviated): [213](#), [214](#), [215](#), [216](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

GPU Acceleration

CONFIDENCE / MODERATE

The existing Roofline explanation, Meta profiling account and MLPerf inference specification support bottleneck diagnosis and constrained comparison. Prioritizing profiling over kernel development is an editorial scope choice; the sources establish no fastest device for this workload.

Evidence: [213] [ev-gpu-acceleration-1](#), [214] [ev-gpu-acceleration-3](#), [216] [ev-gpu-acceleration-5](#)

AI application builder

Outside this profile

GPU programming is outside the managed-API role

No active learning outcome in this profile.

WHY THIS SCOPE

The builder cannot choose kernels or control the provider's GPU execution. Application latency and usage can still be measured through the API without a GPU-optimization curriculum.

STOP AT THIS BOUNDARY

Do not equate slow API responses with a diagnosed GPU bottleneck.

REVISIT WHEN

Reassess if the application begins running its own inference or other GPU computation.

Model platform operator

Cover the essentials

Learn to locate compute and memory bottlenecks

LEARNING OUTCOMES / CHECKS

01 / Interpret a representative inference profile before proposing hardware changes.

Check: Collect request latency and device traces, distinguish memory pressure, compute use and host or queue delays, and justify one next experiment from the observations.

WHY THIS SCOPE

Operating a model backend requires understanding why available accelerator capacity does or does not translate into useful service capacity. Basic profiling supports engine and precision decisions without assuming that more hardware solves the problem.

STOP AT THIS BOUNDARY

Stop at measurement and diagnosis unless a demonstrated kernel bottleneck justifies low-level programming.

REVISIT WHEN

Extend the scope when profiling identifies an unresolved kernel, memory-transfer or utilization bottleneck.

Read alongside: [Inference Optimization](#), [LLM Inference Serving](#)

43 / 47 WORKING RESEARCH SELECTION

Inference Optimization

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Useful applications must meet latency and budget requirements. Measure time to first output, generation speed and concurrent-user behavior; the collected material examines resource trade-offs.

WHERE TO BE CAREFUL

Improving one metric can worsen another, including tail latency. One model and workload do not establish a universal choice, and the topic overlaps serving, GPUs and quantization.

SIMPLER ALTERNATIVES

Reduce calls, shorten inputs or choose a smaller model before deeper optimization. This competency can also serve as an umbrella for narrower techniques.

Workload pattern

How does work arrive?

- | | |
|---|--|
| <p>01 Single stream (Single-stream)
The next query starts after the previous one completes; response latency is central.</p> | <p>02 Multiple streams (Multistream)
Groups of inputs arrive periodically, such as images from several cameras.</p> |
| <p>03 Server traffic (Server)
Requests arrive independently; throughput must respect latency constraints.</p> | <p>04 Offline batch (Offline)
Inputs are available in advance; measure processing throughput for the batch.</p> |

Four MLPerf Inference scenarios from section III-C and table II. Used here to frame workload questions, not as current benchmark-certification instructions.

SOURCE NOTES / TAXONOMY

[217] MLPerf Inference Benchmark

III-C Realistic End-User Scenarios; table II; IV Inference Submission System.

Assessment evidence (abbreviated): [218](#), [219](#), [220](#), [221](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Inference Optimization

CONFIDENCE / MODERATE

Existing MLPerf methodology and serving-system studies support measurements under explicit quality and service constraints. Applying that discipline to API request design and operator experiments is an editorial inference; the cited system gains are not forecasts for these profiles.

Evidence: [\[218\] ev-inference-optimization-1](#), [\[219\] ev-inference-optimization-2](#), [\[221\] ev-inference-optimization-5](#)

AI application builder

Extend

Add quality-constrained latency and usage experiments

LEARNING OUTCOMES / CHECKS

01 / Measure one application-controlled inference trade-off.

Check: Compare a bounded prompt or output-budget change on the same task cases; report answer quality, latency, token usage and failures, including slow requests rather than only averages.

WHY THIS SCOPE

Managed APIs leave request shape and application behavior under the builder's control. A measured comparison can reveal unnecessary work while preserving the task's quality requirements.

STOP AT THIS BOUNDARY

Do not tune provider kernels or infer serving-engine behavior from API timing alone.

REVISIT WHEN

Extend study when a measured latency or usage limit remains unmet after simple request-level changes.

Read alongside: [LLM Inference Serving](#), [Model Quantization](#)

Model platform operator

Extend

Optimize one measured serving constraint at a time

LEARNING OUTCOMES / CHECKS

01 / Run a controlled serving configuration experiment.

Check: Replay the same workload before and after one batching, placement or capacity change; compare quality, time to first token, inter-token latency, tail latency and resource use against explicit limits.

WHY THIS SCOPE

Hosting responsibility creates a direct need to balance concurrency, latency and resource use. The available comparison environment makes a focused experiment feasible, while research shows that optimizing one metric can worsen another.

STOP AT THIS BOUNDARY

Diagnose first and keep the workload and quality checks fixed; a higher token rate alone is not success.

REVISIT WHEN

Revisit after a workload, model, hardware or service-target change exposes a new limiting factor.

44 / 47 WORKING RESEARCH SELECTION

LLM Inference Serving

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

A multi-user model service needs routing, capacity planning, overload handling and scaling. Architecture affects latency, memory and hardware utilization.

WHERE TO BE CAREFUL

The collected systems and deployment accounts cover limited environments. Advantages depend on traffic, models and interconnects, and serving must be distinguished from optimization.

SIMPLER ALTERNATIVES

Small teams may prefer a managed API. Keep service architecture separate from improving specific measured service parameters.

Response phase

Are you waiting for the first token or the rest of the answer?

01 Prefill

Process the prompt and prepare the first token.

02 Decoding

Generate subsequent tokens using earlier computation stored in the KV cache.

Phase placement

Do both phases share a GPU pool?

01 Colocation

Both phases share resources and may interfere with each other.

02 Disaggregation

Phases scale separately, requiring state transfer between them.

Two editorial axes from DistServe sections 2.1-2.3 and 3.3. Computation phase and placement are separate dimensions, not levels of one hierarchy.

SOURCE NOTES / TAXONOMY

[222] DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving

2.1 LLM Inference; 2.2 LLM Serving Optimization; 2.3 Problems and Opportunities; 3.3 Practical Problems.

Assessment evidence (abbreviated): 219, 223, 220, 221. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

LLM Inference Serving

CONFIDENCE / MODERATE

Existing DistServe, comparative engine and autoscaling studies establish workload-dependent serving trade-offs. Assigning essentials to the operator and excluding engine operation from the API builder is an editorial responsibility mapping; no studied architecture is assumed best here.

Evidence: [219] [ev-llm-inference-serving-1](#), [223] [ev-llm-inference-serving-2](#), [221] [ev-llm-inference-serving-6](#)

AI application builder

Outside this profile

Serving infrastructure is owned by the managed provider

No active learning outcome in this profile.

WHY THIS SCOPE

The application consumes managed model APIs and does not run an inference backend. Reliable client integration remains important, but engine scheduling, model placement and GPU capacity are outside this role.

STOP AT THIS BOUNDARY

Handle API limits and failures in the application without adding a self-hosted serving curriculum.

REVISIT WHEN

Reassess if self-hosting or operating a dedicated model endpoint becomes part of the product.

Read alongside: [vLLM](#), [Inference Optimization](#)

Model platform operator

Cover the essentials

Establish a measurable model-serving baseline

LEARNING OUTCOMES / CHECKS

01 / Operate one reproducible endpoint under representative load.

Check: Measure time to first token, inter-token and tail latency across concurrency levels; demonstrate overload behavior, health checks and rollback while retaining agreed quality checks.

WHY THIS SCOPE

Running a multi-user model backend is the central responsibility of this profile. A measured baseline is needed before choosing engine-specific features or advanced optimization.

STOP AT THIS BOUNDARY

Begin with the simplest service that meets the workload; disaggregation and large-scale scheduling require a demonstrated need.

REVISIT WHEN

Extend the architecture when measured capacity, loading or latency constraints cannot be met by the baseline.

Model Quantization

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

Lower-precision representations can reduce model memory requirements and enable deployment on available hardware. The collected studies examine quality, speed and energy trade-offs.

WHERE TO BE CAREFUL

Regressions can depend on language, task or context length, and smaller storage does not guarantee faster execution. Reserve status reflects overlap and limited production evidence in this packet.

SIMPLER ALTERNATIVES

Consider a smaller model or a different serving setup. Quantization can be covered under optimization or Edge AI for a less hardware-specialized audience.

Weight and activation formats

Which values are quantized, and to what representation?

- 01

8-bit floating-point weights and activations (W8A8-FP)
Both groups use an eight-bit floating-point format in the study.
- 02

8-bit integer weights and activations (W8A8-INT)
The bit count is the same, but representation and calibration differ from floating point.
- 03

4-bit weights, 16-bit activations (W4A16-INT)
Reduce weight storage more aggressively while retaining higher-precision activations.

Three configurations compared by Kurtic et al., section 3.2, printed p. 26875 / PDF p. 4. The original atlas visually checked the page. This is the study's comparison, not all quantization methods.

SOURCE NOTES / TAXONOMY

[224] "Give Me BF16 or Give Me Death"? Accuracy-Performance Trade-Offs in LLM Quantization

3.2 Models, Formats, and Algorithms, printed p. 26875 / PDF p. 4; 5 Performance Evaluation.

Assessment evidence (abbreviated): 225, 226, 210, 227. Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Model Quantization

CONFIDENCE / MODERATE

Existing precision-comparison and long-context studies support measuring quality, memory and speed jointly; GPTQ establishes one quantization mechanism. A bounded operator trial is an editorial learning decision despite research-reserve status, not a prediction of cost or latency gains.

Evidence: [225] [ev-model-quantization-1](#), [226] [ev-model-quantization-2](#), [227] [ev-model-quantization-4](#)

AI application builder

Outside this profile

Weight precision is outside this managed-API responsibility

No active learning outcome in this profile.

WHY THIS SCOPE

The builder does not control the provider's weight representation or execution kernels. Model comparisons should still use application quality tests, but implementing quantization has no direct role in the stated scenario.

STOP AT THIS BOUNDARY

Do not infer a provider's quality or speed from an assumed internal bit width.

REVISIT WHEN

Reassess if the application begins selecting and running local model artifacts.

Model platform operator

Run a learning trial

Compare one lower-precision configuration with the baseline

LEARNING OUTCOMES / CHECKS

01 / Measure the precision trade-off on the available hardware.

Check: Compare a supported lower-precision configuration with a higher-precision baseline on the same model and workload; record memory, latency, throughput and quality, including long inputs and relevant languages.

WHY THIS SCOPE

The operator controls artifacts and hardware and has an isolated comparison environment. A short trial can teach whether precision changes meet the service requirements, while research warns that smaller storage does not ensure faster or equally accurate execution.

STOP AT THIS BOUNDARY

This is a learning trial, not a production switch; retain the baseline and distinguish inference quantization from QLoRA training.

REVISIT WHEN

Extend scope only when a measured capacity or cost constraint and acceptable quality results justify a particular method.

Read alongside: [Inference Optimization](#), [Long-Context Modeling](#)

46 / 47 RESEARCH RESERVE

Speculative Decoding

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

A cheaper mechanism proposes tokens for a target model to verify. Appropriate methods can reduce generation time while preserving the target distribution; research compares different drafting approaches.

WHERE TO BE CAREFUL

Drafting and rejected proposals also cost resources. Benefits depend on workload, memory and concurrency. Reserve status reflects specialization, overlap and missing matched production evidence.

SIMPLER ALTERNATIVES

Compare batching, memory management and a smaller model. For broader audiences, treat this as an inference-optimization technique.

Draft source

Where do proposed tokens come from?

- | | |
|--|---|
| <p>01 Independent drafting</p> <p>Proposals are produced outside the target model, usually by a separate cheaper model.</p> | <p>02 Self-drafting</p> <p>Use a simplified internal path or additional lightweight heads.</p> |
|--|---|

Verification criterion

How are proposed tokens checked?

- | | |
|---|--|
| <p>01 Greedy verification (Greedy decoding)</p> <p>In the basic variant, an accepted token matches the target model's best choice.</p> | <p>02 Probabilistic verification (Speculative sampling)</p> <p>Acceptance and correction account for probability distributions to preserve target-model sampling.</p> |
|---|--|

Two axes selected from Xia et al., sections 5.1-5.2 and 6.1-6.2, table 2. Draft shape, including token trees, is a separate omitted axis and is not mixed with acceptance criteria.

SOURCE NOTES / TAXONOMY

[228] [Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding](#)

5.1 Independent Drafting; 5.2 Self-Drafting; 6.1 Greedy Decoding; 6.2 Speculative Sampling; table 2.

Assessment evidence (abbreviated): [229](#), [230](#), [231](#), [232](#), [233](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

Speculative Decoding

CONFIDENCE / LIMITED

Existing speculative-decoding research establishes draft-and-verify methods, while the survey and vLLM benchmark describe acceptance, overhead and workload limitations. Deferral is an editorial sequencing decision because no decoding bottleneck is established; research-reserve status alone does not determine the action.

Evidence: [229] [ev-speculative-decoding-1](#), [230] [ev-speculative-decoding-2](#), [232] [ev-speculative-decoding-4](#)

AI application builder

Outside this profile

Token drafting is outside the managed-API role

No active learning outcome in this profile.

WHY THIS SCOPE

The application does not operate the generation engine or select its drafting method. Its useful responsibility is to measure end-to-end response behavior rather than optimize inaccessible decoding internals.

STOP AT THIS BOUNDARY

Speculation is an execution technique; it is not a method for improving the model's answer quality.

REVISIT WHEN

Reassess if the application takes ownership of a compatible self-hosted engine.

Read alongside: [Inference Optimization](#), [vLLM](#)

Model platform operator

Defer with a trigger

Wait for a measured decoding-latency bottleneck

No active learning outcome in this profile.

WHY THIS SCOPE

The current scenario supplies a comparison environment but no evidence that decoding is the limiting factor. Establish the serving baseline and simpler optimization choices before adding draft-model and verification complexity.

STOP AT THIS BOUNDARY

Do not assume theoretical token savings become workload-level speedups; batching and draft overhead must be measured.

REVISIT WHEN

Run a bounded trial when decoding dominates an unmet latency target and the chosen engine supports an appropriate draft method.

47 / 47 WORKING RESEARCH SELECTION

vLLM

RESEARCH DOSSIER / PROFILE ASSESSMENTS ON NEXT PAGE

WHY IT BELONGS IN THE RESEARCH

vLLM offers a concrete environment for learning model serving, memory management and concurrent requests. The collected LinkedIn account complements research and performance comparisons.

WHERE TO BE CAREFUL

Advantages depend on version, model and hardware. The evidence does not identify one best engine for every workload, and the entry overlaps general serving skills.

SIMPLER ALTERNATIVES

Compare alternative engines or managed services on the actual workload. vLLM can remain an example within a broader serving competency.

Size of one model replica

Where do the model and request memory fit?

- | | |
|--|--|
| <p>01 One GPU (Single GPU)
The simplest option when the model need not be split across GPUs.</p> | <p>02 Multiple GPUs within one node (Tensor parallel inference)
Tensor-parallel inference distributes model operations across connected GPUs.</p> |
| <p>03 Multiple nodes (Tensor and pipeline parallel inference)
Tensor parallelism can be combined with pipeline partitioning of layers across nodes.</p> | |

Simplified decision map from the official Parallelism and Scaling documentation. It concerns one model replica and omits special MoE configurations and uneven GPU splits.

SOURCE NOTES / TAXONOMY

[234] Parallelism and Scaling

Distributed inference strategies for a single-model replica; GPU KV cache size and Maximum concurrency explanations.

Assessment evidence (abbreviated): [235](#), [219](#), [223](#), [236](#). Full methods and caveats are retained in the web view.

EDITORIAL ASSESSMENT / 1 OCTOBER 2026

vLLM

CONFIDENCE / MODERATE

Existing PagedAttention research, an independent vLLM/TGI comparison and LinkedIn's account establish mechanisms and bounded operating examples. The operator trial is an editorial judgment enabled by the stated environment; no current version advantage or migration benefit has been established.

Evidence: [235] [ev-vllm-1](#), [223] [ev-vllm-3](#), [236] [ev-vllm-5](#)

AI application builder

Outside this profile

A serving-engine curriculum is outside the managed-API role

No active learning outcome in this profile.

WHY THIS SCOPE

The application relies on a managed provider and does not deploy or tune a model engine. Learning the client contract has direct value; specializing in vLLM does not address the stated delivery responsibilities.

STOP AT THIS BOUNDARY

An API that resembles a vLLM endpoint does not create engine-operating responsibility.

REVISIT WHEN

Reassess if self-hosting or direct ownership of engine configuration enters the application architecture.

Model platform operator

Run a learning trial

Compare vLLM on the representative workload

LEARNING OUTCOMES / CHECKS

01 / Run an isolated comparison with a supported baseline engine or configuration.

Check: Pin model and engine versions, replay the same request distribution, and compare quality, memory, time to first token, tail latency and throughput under the same service limits; document reasons to continue or stop.

WHY THIS SCOPE

The operator has both a representative workload and an isolated comparison environment, and vLLM familiarity is not assumed. One fair comparison teaches a concrete serving implementation without assuming it is the best engine for this service.

STOP AT THIS BOUNDARY

Do not study every tuning option or migrate production before the measured comparison justifies a specific next step.

REVISIT WHEN

Extend scope when the trial identifies a useful advantage, a diagnosable limitation or a changed model, workload or hardware requirement.

Read alongside: [LLM Inference Serving](#), [Inference Optimization](#)

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

Distinct URLs, not independent studies. Titles link to the original pages. Taxonomy locators sit beside each concept. The full methodology and limitations of assessment evidence are retained in the web view and repository.

- 001 [A Survey of Direct Preference Optimization](#)
arxiv.org
- 002 [Direct Preference Optimization: Your Language Model is Secretly a Reward Model](#)
arxiv.org / Stanford University
- 003 [A General Theoretical Paradigm to Understand Learning from Human Preferences](#)
arxiv.org / IPO authors
- 004 [A Survey of Direct Preference Optimization](#)
arxiv.org / arXiv / DPO survey authors
- 005 [BPO: Revisiting Preference Modeling in Direct Preference Optimization](#)
arxiv.org / arXiv / BPO authors
- 006 [A Survey on In-context Learning](#)
arxiv.org
- 007 [Language Models are Few-Shot Learners](#)
arxiv.org / OpenAI
- 008 [Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?](#)
arxiv.org / University of Washington and collaborators
- 009 [A Survey on In-context Learning](#)
arxiv.org / ICL survey authors
- 010 [In-Context Learning Boosts Speech Recognition via Human-like Adaptation to Speakers and Language Varieties](#)
arxiv.org / Stanford University
- 011 [Rethinking Example Selection in the Era of Million-Token Models](#)
deepmind.google / Google DeepMind
- 012 [Transformer Architectures](#)
huggingface.co
- 013 [Eval-driven development: Lessons from evaluating GenAI at scale](#)
airbnb.tech / Airbnb Engineering

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 014 **On the Opportunities and Risks of Foundation Models**
arxiv.org / Stanford CRFM
- 015 **Introducing HELM Capabilities**
crfm.stanford.edu / Stanford CRFM
- 016 **Large computer language models carry environmental, social risks**
washington.edu / University of Washington
- 017 **LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding**
arxiv.org
- 018 **Benchmarking Long-Context Language Models on Long Code Understanding**
aclanthology.org / ACL 2025 / LongCodeU authors
- 019 **Lost in the Middle: How Language Models Use Long Contexts**
arxiv.org / arXiv / Stanford University et al.
- 020 **LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding**
arxiv.org / LongBench authors
- 021 **Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context**
arxiv.org / Google DeepMind
- 022 **AcademicEval: Live Long-Context LLM Benchmark**
arxiv.org / arXiv / AcademicEval authors
- 023 **LoRA: Low-Rank Adaptation of Large Language Models**
arxiv.org
- 024 **QLoRA: Efficient Finetuning of Quantized LLMs**
arxiv.org
- 025 **From weeks to a day: how we made LLM evaluation fast enough to iterate on**
airbnb.tech / Airbnb Engineering
- 026 **LoRA: Low-Rank Adaptation of Large Language Models**
arxiv.org / Microsoft Research

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 027 **QLoRA: Efficient Finetuning of Quantized LLMs**
arxiv.org / University of Washington
- 028 **LoRA Learns Less and Forgets Less**
arxiv.org / LoRA analysis authors
- 029 **zFLoRA: Zero-Latency Fused Low-Rank Adapters**
arxiv.org / arXiv / zFLoRA authors
- 030 **DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning**
arxiv.org
- 031 **DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning**
arxiv.org / DeepSeek AI
- 032 **Reasoning Can Hurt the Inductive Abilities of Large Language Models**
arxiv.org / arXiv
- 033 **The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity**
arxiv.org / arXiv / NeurIPS research authors
- 034 **OckBench: Measuring the Efficiency of LLM Reasoning**
arxiv.org / arXiv research collaboration
- 035 **JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models**
arxiv.org
- 036 **JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models**
arxiv.org / JSONSchemaBench authors
- 037 **StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs**
arxiv.org / StructEval authors
- 038 **Empirical Study for Structured Output Control in LLMs for Software Engineering**
arxiv.org / arXiv / study authors
- 039 **Musings on Building a Generative AI Product**
linkedin.com / LinkedIn Engineering

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 040 **Reimagining LinkedIn's search tech stack**
linkedin.com / LinkedIn Engineering
- 041 **Massive Supervised Fine-tuning Experiments Reveal How Data, Layer, and Training Factors Shape LLM Alignment Quality**
arxiv.org
- 042 **Finetuned Language Models Are Zero-Shot Learners**
arxiv.org / Google Research
- 043 **LIMA: Less Is More for Alignment**
arxiv.org / Meta AI and collaborators
- 044 **Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To!**
arxiv.org / Princeton University and collaborators
- 045 **Massive Supervised Fine-tuning Experiments Reveal How Data, Layer, and Training Factors Shape LLM Alignment Quality**
arxiv.org / arXiv / EMNLP research authors
- 046 **Analyzing the Effects of Supervised Fine-Tuning on Model Knowledge from Token and Parameter Levels**
arxiv.org / arXiv / EMNLP research authors
- 047 **Inference-Time Scaling for Complex Tasks: Where We Stand and What Lies Ahead**
arxiv.org
- 048 **Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters**
arxiv.org / UC Berkeley and collaborators
- 049 **s1: Simple test-time scaling**
arxiv.org / Stanford University and collaborators
- 050 **Inference-Time Scaling for Complex Tasks: Where We Stand and What Lies Ahead**
arxiv.org / Microsoft Research
- 051 **simple-evals: o3 and o4-mini benchmark results**
github.com / OpenAI
- 052 **The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale**
arxiv.org

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 053 **The Data Provenance Initiative: A Large Scale Audit of Dataset Licensing & Attribution in AI**
arxiv.org / Data Provenance Initiative
- 054 **The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale**
arxiv.org / Hugging Face
- 055 **Ultra-FineWeb: Efficient Data Filtering and Verification for High-Quality LLM Training Data**
arxiv.org / arXiv research collaboration
- 056 **Achieving 10,000x training data reduction with high-fidelity labels**
research.google / Google Research / Google Ads
- 057 **Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity**
arxiv.org
- 058 **Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection**
arxiv.org / University of Washington and collaborators
- 059 **Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity**
arxiv.org / KAIST and collaborators
- 060 **RAGChecker: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation**
arxiv.org / arXiv / Amazon AWS AI Labs et al.
- 061 **InfoDeepSeek: Benchmarking Agentic Information Seeking for Retrieval-Augmented Generation**
arxiv.org / InfoDeepSeek authors
- 062 **Dense Passage Retrieval for Open-Domain Question Answering**
arxiv.org
- 063 **Dense Passage Retrieval for Open-Domain Question Answering**
arxiv.org / Facebook AI Research and collaborators
- 064 **BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models**
arxiv.org / UKP Lab and collaborators
- 065 **ReasonIR: Training Retrievers for Reasoning Tasks**
arxiv.org / MIT, Meta, University of Washington and collaborators

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 066 **RaDeR: Reasoning-aware Dense Retrieval Models**
arxiv.org / RaDeR authors
- 067 **MTEB: Massive Text Embedding Benchmark**
arxiv.org
- 068 **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**
arxiv.org / UKP Lab
- 069 **MTEB: Massive Text Embedding Benchmark**
arxiv.org / MTEB authors
- 070 **Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models**
arxiv.org / Alibaba Qwen team
- 071 **Maintaining MTEB: A Long-Term Evaluation Framework for Multilingual Text Embeddings**
arxiv.org / MTEB maintainers
- 072 **Hybrid search revisited: introducing the linear retriever in Elasticsearch!**
elastic.co
- 073 **SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval**
arxiv.org / Naver Labs Europe and collaborators
- 074 **Balancing the Blend: An Experimental Analysis of Trade-offs in Hybrid Search**
arxiv.org / Zhejiang University, Huazhong University of Science and Technology and collaborators
- 075 **MMDocIR: Benchmarking Multimodal Retrieval for Long Documents**
arxiv.org
- 076 **ColPali: Efficient Document Retrieval with Vision Language Models**
arxiv.org / Illuin Technology and collaborators
- 077 **MMDocIR: Benchmarking Multimodal Retrieval for Long Documents**
arxiv.org / MMDocIR authors
- 078 **UNIDOC-BENCH: A Unified Benchmark for Document-Centric Multimodal RAG**
arxiv.org / UNIDOC-BENCH authors

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 079 **Building a multimodal RAG based application using Amazon Bedrock Data Automation and Amazon Bedrock Knowledge Bases**
aws.amazon.com / Amazon Web Services
- 080 **Retrieval-Augmented Generation for Large Language Models: A Survey**
arxiv.org
- 081 **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**
arxiv.org / Facebook AI Research and collaborators
- 082 **Benchmarking Large Language Models in Retrieval-Augmented Generation**
arxiv.org / RGB benchmark authors
- 083 **Retrieval-Augmented Generation for Large Language Models: A Survey**
arxiv.org / RAG survey authors
- 084 **Retrieval-Augmented Generation to Support Railways Engineering Tasks: A Case Study**
arxiv.org / NIER Ingegneria, University of Bologna and collaborators
- 085 **Genie: Uber's Gen AI On-Call Copilot**
uber.com / Uber Engineering
- 086 **RankLLM**
github.com
- 087 **Passage Re-ranking with BERT**
arxiv.org / University of Waterloo and collaborators
- 088 **Evolution and Scale of Uber's Delivery Search Platform**
uber.com / Uber Engineering
- 089 **A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge**
arxiv.org
- 090 **A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge**
arxiv.org / Vector-database survey authors
- 091 **The Impacts of Data, Ordering, and Intrinsic Dimensionality on Recall in HNSW**
arxiv.org / HNSW evaluation authors

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 092 **NaviX: A Native Vector Index Design for Graph DBMSs With Robust Predicate-Agnostic Search Performance**
arxiv.org / University of Waterloo
- 093 **Faiss indexes**
github.com
- 094 **Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs**
arxiv.org / HNSW authors
- 095 **Billion-scale similarity search with GPUs**
arxiv.org / Facebook AI Research
- 096 **Agent Memory: Characterization and System Implications of Stateful Long-Horizon Workloads**
arxiv.org
- 097 **Augmenting Language Models with Long-Term Memory**
arxiv.org / arXiv / Microsoft Research
- 098 **MemGPT: Towards LLMs as Operating Systems**
arxiv.org / arXiv / UC Berkeley
- 099 **Agent Memory: Characterization and System Implications of Stateful Long-Horizon Workloads**
arxiv.org / arXiv / MIT, Stanford University and KU Leuven
- 100 **Can Agent Memory Systems Track Evolving State?**
arxiv.org / arXiv / University of Illinois Urbana-Champaign
- 101 **Agent Planning Benchmark: A Diagnostic Framework for Planning Capabilities in LLM Agents**
arxiv.org
- 102 **Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models**
arxiv.org / arXiv / Singapore Management University et al.
- 103 **Tree of Thoughts: Deliberate Problem Solving with Large Language Models**
arxiv.org / arXiv / Princeton University and Google DeepMind
- 104 **DeepPlanning: Benchmarking Long-Horizon Agentic Planning with Verifiable Constraints**
arxiv.org / arXiv / Alibaba Group et al.

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 105 **Agent Planning Benchmark: A Diagnostic Framework for Planning Capabilities in LLM Agents**
arxiv.org / arXiv / Shanghai Jiao Tong University et al.
- 106 **ReAct: Synergizing Reasoning and Acting in Language Models**
arxiv.org
- 107 **ReAct: Synergizing Reasoning and Acting in Language Models**
arxiv.org / arXiv / Princeton University and Google Research
- 108 **AgentBench: Evaluating LLMs as Agents**
arxiv.org / arXiv / Tsinghua University et al.
- 109 **AgentArch: A Comprehensive Benchmark to Evaluate Agent Architectures in Enterprise**
arxiv.org / arXiv / ServiceNow authors
- 110 **Building the agentic future of recruiting: how we engineered LinkedIn's Hiring Assistant**
linkedin.com / LinkedIn Engineering
- 111 **The LinkedIn Generative AI Application Tech Stack: Extending to Build AI Agents**
linkedin.com / LinkedIn Engineering
- 112 **GitHub Copilot features**
docs.github.com
- 113 **Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity**
metr.org
- 114 **How much does AI impact development speed? An enterprise-based randomized controlled trial**
arxiv.org / arXiv / Google study authors
- 115 **How generative AI affects highly skilled workers**
mitsloan.mit.edu / MIT Sloan
- 116 **Who is using AI to code? Global diffusion and impact of generative AI**
real.mtak.hu / Science
- 117 **Examining the Use and Impact of an AI Code Assistant on Developer Productivity and Experience in the Enterprise**
research.ibm.com / IBM Research

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

118

1,500+ PRs Later: Spotify's Journey with Our Background Coding Agent (Honk, Part 1)
engineering.atspotify.com

119

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?
arxiv.org / arXiv / Princeton University et al.

120

Does SWE-Bench-Verified Test Agent Ability or Model Memory?
arxiv.org / arXiv / University of Waterloo et al.

121

SWE-CI: Evaluating Agent Capabilities in Maintaining Codebases via Continuous Integration
arxiv.org / arXiv / Chongqing University and collaborators

122

Concurrency and async / await
fastapi.tiangolo.com

123

Scalable and Secure AI Inference in Healthcare: A Comparative Benchmarking of FastAPI and Triton Inference Server on Kubernetes
arxiv.org / arXiv / FastAPI-Triton benchmark author

124

Fixing FastAPI Throughput Without Going Fully Async
dpdzero.com / DPDzero Engineering

125

Netflix Dispatch release v20241220
github.com / Netflix / GitHub

126

Benchmarking the performance of Python web frameworks
ph.pollub.pl / Journal of Computer Sciences Institute / Lublin University of Technology

127

Model Context Protocol - Server Features / Overview
modelcontextprotocol.io

128

MCP-AgentBench: Evaluating Real-World Language Agent Performance with MCP-Mediated Tools
arxiv.org / arXiv / MCP-AgentBench authors

129

MCP Security Bench: Benchmarking Attacks Against Model Context Protocol in LLM Agents
arxiv.org / arXiv

130

Model Context Protocol Specification 2025-06-18
modelcontextprotocol.io / Model Context Protocol project

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 131 **Solving the Identity Crisis for AI Agents**
uber.com / Uber Engineering Security
- 132 **MultiAgentBench: Evaluating the Collaboration and Competition of LLM agents**
arxiv.org
- 133 **AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation**
arxiv.org / arXiv / Microsoft Research
- 134 **MultiAgentBench: Evaluating the Collaboration and Competition of LLM agents**
arxiv.org / arXiv
- 135 **DPBench: Large Language Models Struggle with Simultaneous Coordination**
arxiv.org / arXiv
- 136 **The LinkedIn Generative AI Application Tech Stack: Extending to Build AI agents**
news.linkedin.com / LinkedIn Engineering
- 137 **tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains**
arxiv.org
- 138 **GAIA: a benchmark for General AI Assistants**
arxiv.org / arXiv / Meta-FAIR et al.
- 139 **tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains**
arxiv.org / arXiv
- 140 **tau-squared-bench: Evaluating Conversational Agents in a Dual-Control Environment**
arxiv.org / arXiv / Princeton University and Salesforce Research
- 141 **Establishing Best Practices for Building Rigorous Agentic Benchmarks**
arxiv.org / arXiv / Princeton University, Stanford University and collaborators
- 142 **Holistic Evaluation of Language Models**
arxiv.org / arXiv / Stanford CRFM
- 143 **Benchmark Data Contamination of Large Language Models: A Survey**
arxiv.org / Benchmark-contamination survey authors

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 144 **SelfCheckGPT - authors' repository**
github.com
- 145 **SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models**
arxiv.org / arXiv / University of Cambridge
- 146 **FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation**
arxiv.org / arXiv / University of Washington et al.
- 147 **RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models**
arxiv.org / arXiv
- 148 **Detecting Hallucinations in Authentic LLM-Human Interactions**
arxiv.org / arXiv / University of Hamburg
- 149 **Rethinking Evaluation for LLM Hallucination Detection: A Desiderata, A New RAG-based Benchmark, New Insights**
arxiv.org / arXiv / Pennsylvania State University and collaborators
- 150 **CheckList - authors' repository**
github.com
- 151 **CheckList - test_types.py**
raw.githubusercontent.com
- 152 **Beyond Accuracy: Behavioral Testing of NLP Models with CheckList**
arxiv.org / arXiv / Microsoft Research and UC Irvine
- 153 **MORTAR: Multi-turn Metamorphic Testing for LLM-based Dialogue Systems**
arxiv.org / arXiv
- 154 **Measuring what Matters: Construct Validity in Large Language Model Benchmarks**
arxiv.org / arXiv / University of Oxford et al.
- 155 **Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena**
arxiv.org
- 156 **An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4**
aclanthology.org / Association for Computational Linguistics

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 157 Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges
aclanthology.org / Association for Computational Linguistics
- 158 G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment
arxiv.org / arXiv / Microsoft Research
- 159 Large Language Models are not Fair Evaluators
arxiv.org / arXiv
- 160 Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena
arxiv.org / arXiv / LMSYS Org
- 161 RAGAS: Automated Evaluation of Retrieval Augmented Generation
arxiv.org
- 162 GaRAGE: A Benchmark with Grounding Annotations for RAG Evaluation
aclanthology.org / Association for Computational Linguistics
- 163 RAGAS: Automated Evaluation of Retrieval Augmented Generation
arxiv.org / arXiv / Exploding Gradients
- 164 ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems
arxiv.org / arXiv / Stanford University et al.
- 165 The Viability of Crowdsourcing for RAG Evaluation
arxiv.org / arXiv / University of Waterloo and collaborators
- 166 Enhanced agentic RAG: Improving an on-call copilot at Uber
uber.com / Uber Engineering
- 167 Agentic AI Threat Modeling Framework: MAESTRO
cloudsecurityalliance.org
- 168 AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents
arxiv.org / arXiv / ETH Zurich
- 169 strideSEA: A STRIDE-centric Security Evaluation Approach
arxiv.org / arXiv / Carleton University

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 170 **Securing Agentic AI: Threat Modeling and Risk Analysis for Network Monitoring Agentic AI System**
arxiv.org / arXiv / University of North Texas
- 171 **OWASP Top 10 for Agentic Applications for 2026**
genai.owasp.org / OWASP GenAI Security Project
- 172 **OWASP LLM02:2025 Sensitive Information Disclosure**
genai.owasp.org
- 173 **Beyond Memorization: Violating Privacy Via Inference with Large Language Models**
arxiv.org / arXiv / ETH Zurich
- 174 **Privacy Auditing of Large Language Models**
arxiv.org / arXiv / Princeton University and Google DeepMind
- 175 **OWASP GenAI Security Project - Solutions Reference Guide Q2_Q3'25**
genai.owasp.org / OWASP GenAI Security Project
- 176 **NIST Privacy Framework**
nist.gov / NIST
- 177 **NeMo Guardrails - Guardrail Types**
docs.nvidia.com
- 178 **Do-Not-Answer: A Dataset for Evaluating Safeguards in LLMs**
arxiv.org / arXiv / MBZUAI et al.
- 179 **NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails**
arxiv.org / arXiv / NVIDIA
- 180 **JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models**
arxiv.org / arXiv / University of Pennsylvania and ETH Zurich et al.
- 181 **Adaptive Adversaries: A Multi-Turn, Multi-LLM Benchmark for LLM Agent Security**
arxiv.org / arXiv / Gray Swan AI
- 182 **MLCommons Alluminate Jailbreak Benchmark v0.5**
mlcommons.org / MLCommons

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 183 **From firefighting to building: How AI agents transformed data engineering at Grab**
engineering.grab.com / Grab Tech
- 184 **Lessons From Red Teaming 100 Generative AI Products**
arxiv.org
- 185 **Red-Teaming for Generative AI: Silver Bullet or Security Theater?**
arxiv.org / arXiv / Carnegie Mellon University et al.
- 186 **Lessons From Red Teaming 100 Generative AI Products**
arxiv.org / arXiv / Microsoft
- 187 **Security Challenges in AI Agent Deployment: Insights from a Large Scale Public Competition**
arxiv.org / arXiv / Gray Swan AI and collaborators
- 188 **Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile**
nist.gov / NIST
- 189 **Assessing Risks and Impacts of AI (ARIA): Pilot Evaluation Report**
nist.gov / NIST
- 190 **SafeRAG: Benchmarking Security in Retrieval-Augmented Generation of Large Language Model**
arxiv.org
- 191 **SafeRAG: Benchmarking Security in Retrieval-Augmented Generation of Large Language Model**
aclanthology.org / Association for Computational Linguistics
- 192 **PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models**
arxiv.org / arXiv / University at Buffalo et al.
- 193 **Defending Against Knowledge Poisoning Attacks During Retrieval-Augmented Generation**
arxiv.org / arXiv
- 194 **Unlocking Financial Insights with Finch: Uber's Conversational AI Data Agent**
uber.com / Uber Engineering
- 195 **Inference using Invoke API**
docs.aws.amazon.com

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 196 **Process multiple prompts with batch inference**
docs.aws.amazon.com
- 197 **Cloud Platforms for AI: A Scoping Review**
arxiv.org / Independent cloud-AI review authors
- 198 **No Free Lunch with Guardrails**
arxiv.org / arXiv / Enkrypt AI research authors
- 199 **An Empirical Study of Production Incidents in Generative AI Cloud Services**
arxiv.org / arXiv / GenAI cloud incident study authors
- 200 **Build reliable Agentic AI solution with Amazon Bedrock: Learn from Pushpay's journey on GenAI evaluation**
aws.amazon.com / Amazon Web Services
- 201 **How GoDaddy built a category generation system at scale with batch inference for Amazon Bedrock**
aws.amazon.com / Amazon Web Services
- 202 **How Amazon Bedrock CMI Cut AI Model Onboarding Time by 75%**
engineering.salesforce.com / Salesforce Engineering
- 203 **Storage**
docs.docker.com
- 204 **Decomposing Docker Container Startup Performance: A Three-Tier Measurement Study on Heterogeneous Infrastructure**
arxiv.org / arXiv / Shamsheer Khan
- 205 **NIST SP 800-190: Application Container Security Guide**
csrc.nist.gov / NIST
- 206 **On-Demand Container Partitioning for Distributed ML**
usenix.org / USENIX Association
- 207 **MettEagle: Costs and Benefits of Implementing Containers on Microkernels**
usenix.org / USENIX Association
- 208 **TinyMLOps: Operational Challenges for Widespread Edge AI Adoption**
arxiv.org

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 209 **Deep Learning at the Edge: Challenges and Trends**
arxiv.org / Edge deep-learning survey authors
- 210 **Sustainable LLM Inference for Edge AI: Evaluating Quantized LLMs for Energy Efficiency, Output Accuracy, and Inference Latency**
arxiv.org / arXiv / sustainable edge-LLM inference study authors
- 211 **Energy-Efficient Vision Transformer Inference for Edge-AI Deployment**
arxiv.org / arXiv / edge ViT energy-efficiency study authors
- 212 **MLPerf Tiny benchmark**
mlcommons.org / MLCommons
- 213 **The Roofline Model: Visualizing and Optimizing Performance**
amcr.lbl.gov
- 214 **Zoomer: Powering AI Performance at Meta's Scale Through Intelligent Debugging and Optimization**
engineering.fb.com / Engineering at Meta
- 215 **New MLCommons MLPerf Training v5.0 Benchmark Results Reflect Rapid Growth and Evolution of the Field of AI**
mlcommons.org / MLCommons
- 216 **MLPerf Inference: Datacenter benchmark suite**
mlcommons.org / MLCommons
- 217 **MLPerf Inference Benchmark**
arxiv.org
- 218 **MLPerf Inference Benchmark**
arxiv.org / MLCommons researchers
- 219 **DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving**
arxiv.org / OSDI / Peking University and collaborators
- 220 **Torpor: GPU-Enabled Serverless Computing for Low-Latency, Resource-Efficient Inference**
usenix.org / USENIX ATC 2025 / Torpor authors
- 221 **BlitzScale: Fast and Live Large Model Autoscaling with O(1) Host Caching**
usenix.org / USENIX OSDI 2025 / BlitzScale authors

Follow the evidence.

- 222 **DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving**
arxiv.org
- 223 **Comparative Analysis of Large Language Model Inference Serving Systems: A Performance Study of vLLM and HuggingFace TGI**
arxiv.org / arXiv / vLLM-TGI comparison author
- 224 **"Give Me BF16 or Give Me Death"? Accuracy-Performance Trade-Offs in LLM Quantization**
aclanthology.org
- 225 **"Give Me BF16 or Give Me Death"? Accuracy-Performance Trade-Offs in LLM Quantization**
aclanthology.org / ACL 2025 / LLM quantization study authors
- 226 **GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers**
arxiv.org / IST Austria and collaborators
- 227 **Long-Context Inference with Quantized LLMs**
arxiv.org / Quantized long-context study authors
- 228 **Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding**
arxiv.org
- 229 **Accelerating Large Language Model Decoding with Speculative Sampling**
arxiv.org / DeepMind
- 230 **A Survey on Speculative Decoding**
arxiv.org / Speculative-decoding survey authors
- 231 **Scaling Up, Speeding Up: A Benchmark of Speculative Decoding for Efficient LLM Test-Time Scaling**
arxiv.org / arXiv research collaboration
- 232 **Speculative Decoding: Performance or Illusion?**
arxiv.org / arXiv / production-grade serving benchmark authors
- 233 **Fast Inference from Transformers via Speculative Decoding**
proceedings.mlr.press / ICML / Google Research
- 234 **Parallelism and Scaling**
docs.vllm.ai

REFERENCE LIBRARY / ABBREVIATED RECORDS

Follow the evidence.

- 235 **Efficient Memory Management for Large Language Model Serving with PagedAttention**
arxiv.org / arXiv / SOSP authors
- 236 **How we leveraged vLLM to power our GenAI applications at LinkedIn**
linkedin.com / LinkedIn Engineering