AI BENCH LAB

AiBenchLab User Manual

Complete offline reference guide for AiBenchLab — professional AI model benchmarking.

Version 1.0 | March 2026

aibenchlab.com

Complete Reference · Professional AI Model Benchmarking for Windows

Contents

Complete offline reference guide for AiBenchLab — professional AI model benchmarking.

1. Introduction

AiBenchLab is a professional desktop application that benchmarks AI language models for intelligence, quality, and safety. Unlike public leaderboards that test models on cloud servers under ideal conditions, AiBenchLab tests models on your hardware, with your configuration, using tests that models haven't been trained on.

The application runs 254 tests across 11 evaluation domains and 998 scoring dimensions. It works with both local models (through the built-in llama.cpp server, Ollama, or LM Studio) and cloud APIs (OpenAI, Anthropic, Google Gemini, xAI, Groq, and any OpenAI-compatible endpoint). Results are exportable as professional PDF reports, machine-readable JSON, CSV, or the tamper-proof MBX format.

AiBenchLab collects zero telemetry. Your benchmarks stay on your machine.

Who This Manual Is For

This manual covers all tiers of AiBenchLab: Trial, Pro, Consultant, and Enterprise. Features that require a specific tier are marked accordingly throughout the guide. If you are evaluating AiBenchLab with the 14-day free trial, everything in this manual applies — the trial provides full access to all features.

How This Manual Is Organized

Chapter	What You'll Learn
Getting Started	Download, install, and run your first benchmark
The Benchmark Wizard	How to describe your AI workload and build a benchmark plan
Understanding Your Score	What the composite Score means and how to read results
Local Providers	Set up Ollama, LM Studio, or the built-in inference server
Cloud Providers	Connect OpenAI, Anthropic, Google Gemini, and others
Cost Estimator	Estimate API and electricity costs before running benchmarks

Chapter	What You'll Learn
The Interface	Dashboard, sidebar, navigation, and keyboard shortcuts
Settings and Preferences	Configure the application to fit your workflow
Export and Reports	Generate PDF reports, comparison reports, and data exports
Licensing and Tiers	What each tier includes and how licensing works
CLI Reference	Run benchmarks from the command line
REST API Reference	Automate benchmarks with HTTP endpoints
MCP Server Reference	Let AI tools run benchmarks through the Model Context Protocol
Troubleshooting	Common issues and how to resolve them

2. Getting Started

System Requirements

Component	Minimum	Recommended
OS	Windows 10+, macOS 10.15+, Linux (Ubuntu 20.04+)	Windows 11, macOS 13+
RAM	8 GB	16 GB+
GPU	None (CPU inference supported)	NVIDIA GPU with 6 GB+ VRAM
Storage	3 GB (app + required components)	Additional space for local AI models

Download and Install

Windows: Download `AiBenchLab-Setup.exe` from the Download page at aibenchlab.com/download. Run the installer, accept the EULA, choose per-user or system-wide installation, and launch from the Start Menu.

macOS: Download `AiBenchLab.dmg`, open the disk image, and drag AiBenchLab to Applications. Right-click and select Open on first launch to pass the Gatekeeper prompt.

Linux: Download `AiBenchLab.AppImage`, make it executable with `chmod +x AiBenchLab.AppImage`, and run it. Debian/Ubuntu users can use the `.deb` package. Fedora/RHEL users can use the `.rpm` package.

First Launch Setup

When you launch AiBenchLab for the first time, the Setup screen downloads three required components:

Component	What It Is	Download Size
llama.cpp runtime	Built-in inference server for running local models	~141 MB (CUDA) or ~55 MB (Vulkan)
Judge model	Gemma 2 2B IT — scores model responses during benchmarks	~1.5 GB
Quick-start model	Llama 3.2 1B Instruct — a bundled model so you can benchmark immediately	~1.3 GB

GPU detection runs automatically. NVIDIA GPUs use the CUDA backend; AMD and Intel GPUs use Vulkan. Components are stored locally and never uploaded anywhere.

Run Your First Benchmark

› Click **New Run** in the sidebar to open the Benchmark Wizard.

› Type a short description of your AI workload — for example, "Answer customer support questions about our product documentation."

› The wizard recommends relevant test disciplines. Select the ones that match your use case.

› Choose the bundled quick-start model (already available with no download).

› Click **Create & Enqueue** to start.

› Watch the live execution screen: progress bar, pass/fail counters, GPU telemetry.

› When complete, click **View Full Results** to see your Score and domain breakdowns.

How the Built-In Server Works

AiBenchLab ships with its own llama.cpp inference server. You do not need Ollama, LM Studio, or any external tool to run benchmarks. The built-in server launches automatically when you select a local model, runs on auto-assigned ports (8900–9999), and shuts down when done. A separate Judge Server runs the scoring model on CPU (port 8899) so it doesn't compete with the model under test for GPU memory.

3. The Benchmark Wizard

The Benchmark Wizard transforms a description of your AI workload into a reproducible benchmark plan. It operates in two modes: Simple (natural language input, minimal decisions) and Advanced (structured questionnaire, full control). Toggle between them with the Simple / Advanced switch at the top.

Simple Mode

Step 1 — Describe Your Task. Type what the AI model needs to do in plain language. The wizard runs your description through a two-stage classifier that identifies relevant evaluation disciplines. Results appear as selectable discipline tags below your input. Each tag shows the discipline name, test count, and estimated run time.

The wizard is transparent about its confidence: high-confidence matches get a clear explanation of why the discipline fits; low-confidence results ask you for clarification.

Step 2 — Select Models. Models from all configured providers appear in a unified list. Each model card shows the name, provider, parameter count, context window, match percentage for your selected disciplines, and whether it fits your GPU's VRAM. Select one or more models.

Step 3 — Review and Queue. Review your selections — disciplines, test counts, selected models, estimated run time — and click **Create & Enqueue** to submit.

Advanced Mode

Advanced mode expands the wizard into 8 steps with full control over every parameter.

Step 1 — Structured Questionnaire. Fill out a detailed form covering primary intent type, output requirements, context bucket, deployment constraints, latency priority, and target hardware. Eight Quick Picks are available for common use cases.

Step 2 — Model Selection. Same model list as Simple mode, but models are scored against your full questionnaire answers.

Step 3 — Suite and Discipline Selection. Browse all 22 pre-built test suites. Select a suite or let the wizard auto-assemble a plan from your questionnaire responses.

Step 4 — Review and Customize Tests. See every test that will run, organized by domain. Toggle individual tests on or off. Save custom selections as a reusable suite (Consultant tier and above).

Step 5 — Session Configuration. Set session name, contact name, deterministic mode, seed, temperature override, and runtime target.

Step 6 — Final Review. Summary of all choices before submission.

Step 7 — Cost Estimator. For cloud models, see projected token counts and API costs before you run. A confirmation dialog appears if the estimate exceeds your warning threshold.

Step 8 — Benchmark Execution. Live progress with test counters, GPU telemetry, thermal protection, and activity log.

The Intent Block System

Behind the scenes, the wizard assembles benchmark plans using intent blocks — pre-built modules that map capabilities to specific tests. Each block defines what it proves, which tests to run (with weights), exclusions, and score multipliers. When multiple blocks combine, the engine deduplicates tests, merges weights, resolves conflicts, and generates an audit trail.

---

4. Understanding Your Score

The Composite Score

Every benchmark session produces a single composite Score on a 0–100 scale. This number combines three factors:

Factor	What It Measures	Range
Capability Score (CS)	Performance across tested domains	0–1.0
Deployment Index (DI)	Safety penalty for critical failures	0.1–1.0
Consistency Factor (CF)	Variance penalty for inconsistent results	0.6–1.0

Formula: Score = 100 × CS × DI × CF

A model that performs well, has no safety failures, and produces consistent results scores near 100. Safety failures apply exponential penalties — a model scoring 85 on capability but 0.5 on the Deployment Index would produce a final Score around 42.

The 11 Domains

Capability Domains (6): Reasoning (20% weight), Coding (20%), Agentic (20%), Multimodal (15%), Tool Calling (15%), Chat (10%).

Safety Domains (2): Deployment Risk and Adversarial Safety. These don't contribute to the Capability Score directly — failures apply penalties through the Deployment Index.

Specialized Domains (3): Multi-Turn Adversarial, Agentic Email, and Context Retention.

How Scores Are Calculated

Each test is scored across multiple dimensions — specific aspects like correctness, efficiency, safety, and style. Dimensions come in two types: differentiating (no cap, measures real quality separation) and saturating (capped at 70–80%, measures baseline compliance that most good models meet). This prevents "table stakes" behaviors from inflating scores.

Individual test scores combine into domain scores using a difficulty-weighted approach: harder tests carry more weight than easy ones. A single failure on a hard test pulls the domain score down significantly.

The Pass Threshold

A test passes if its score is 0.5 or higher. This threshold is uniform across all tests and domains. The pass rate — the percentage of tests meeting this bar — appears alongside the composite Score.

Score Ranges

Score Range	Interpretation
90–100	Exceptional. High reliability and quality for this workload.
75–89	Strong. Suitable for production in most scenarios.
60–74	Adequate. Works for many tasks but may struggle with edge cases.
40–59	Below average. Significant capability or consistency gaps.
Below 40	Poor fit. Consider a different model or configuration.

---

5. Local Providers

AiBenchLab supports three local inference options: the built-in llama.cpp server, Ollama, and LM Studio. You can use any combination simultaneously.

Built-In llama.cpp Server

Ships with the app. No setup, no terminal, no external software. When you select a local model, the app launches a server instance for it, runs the benchmark, and shuts it down. GPU acceleration is automatic: CUDA for NVIDIA, Vulkan for AMD/Intel, CPU fallback if neither works.

The app detects common failure modes and provides guidance: chat template errors, CUDA issues, out-of-memory conditions, and corrupted model files. Run the system check in Settings > Components to verify your setup.

Ollama

Install Ollama from ollama.ai, pull a model (`ollama pull llama3.2`), and start the server (`ollama serve`). AiBenchLab auto-detects Ollama at localhost:11434 on every launch. Vision-capable models (LLaVA, Gemma 3) are automatically detected for multimodal tests.

LM Studio

Install LM Studio from lmstudio.ai, download a model, load it, and start the local server. AiBenchLab detects LM Studio at localhost:1234. LM Studio may take a moment to become available after starting — the app confirms the connection is stable before listing it as ready.

The Model Catalog

AiBenchLab includes a searchable catalog of over 51,000 HuggingFace models. Browse by size using filter buttons (Tiny, Small, Medium, Large, XL). The GPU Fit column shows whether each model fits your hardware. Presets include Production Ready, Commercial Safe, Air-Gapped, and Quick Explore.

---

6. Cloud Providers

Supported Providers

Provider	Default Endpoint	Key Source
OpenAI	api.openai.com/v1	platform.openai.com/api-keys
Anthropic	api.anthropic.com/v1	console.anthropic.com/settings/keys
Google Gemini	generativelanguage.googleapis.com	aistudio.google.com/apikey
xAI (Grok)	api.x.ai/v1	console.x.ai
Groq	api.groq.com/openai/v1	console.groq.com/keys
Together AI	api.together.xyz/v1	—
Custom	User-configured	—

Adding a Provider

Go to Settings, click + Add Provider, select a template, enter your API key, save, and test the connection. Keys are stored locally and never sent to AiBenchLab servers.

Performance Metrics

AiBenchLab captures timing data for every cloud API call: TTFT (Time to First Token), TPOT (Time Per Output Token), TPS (Tokens Per Second), and E2E Latency (total request time). These appear alongside quality scores in your results.

---

7. Cost Estimator

The Cost Estimator calculates projected costs before you run a benchmark.

Cloud models: Estimates input and output token counts, looks up per-token pricing, and calculates total API cost. Pricing is cached and updated automatically. A confidence level (High, Medium, Low) indicates how reliable the estimate is.

Local models: Estimates electricity costs based on your GPU's power draw, system overhead, electricity rate, and expected tokens per second.

Operational projections: Extends the single-benchmark estimate into daily, monthly, and annual scenarios for different usage levels (Light, Moderate, Production, Heavy).

Cost warning: If estimated cloud costs exceed your threshold (default $5), a confirmation dialog appears before execution. Adjust the threshold in Preferences > Cost Settings.

---

8. The Interface

Dashboard

The Dashboard is your home screen. It shows recent benchmark sessions, summary statistics, and quick actions. From here you can start a new run, view previous results, or access settings.

Sidebar Navigation

The sidebar provides access to all major sections:

Section	What It Does
Dashboard	Home screen with recent runs and stats
Wizard	Start a new benchmark
Test Suites	Browse and manage the 22 pre-built suites
Create Suite	Build a custom test suite (Consultant+)
Model Queue	View and manage the benchmark queue

Section	What It Does
Scheduled Runs	Set up recurring benchmarks
Results	Browse all completed benchmark sessions
History	Full session history with search and filter
Model Catalog	Browse 51,000+ models with GPU Fit detection
Local Models	Manage downloaded local models
Cloud Models	View and configure cloud provider models

Keyboard Shortcuts

Press **Ctrl+K** (or **Cmd+K** on macOS) to open the Command Palette — a quick-access search for all 23 available commands. Start typing to filter.

Common shortcuts:

Shortcut	Action
Ctrl+K	Open Command Palette
Ctrl+N	New Benchmark Run
Ctrl+E	Export Current Session
Ctrl+,	Open Settings
Ctrl+D	Go to Dashboard

---

9. Settings and Preferences

Settings

Access from the sidebar or Ctrl+,. Settings include:

Providers — Add, edit, remove, and test AI providers (local and cloud). Each provider shows connection status, model count, and response time.

Components — View installed components (llama.cpp runtime, judge model, quick-start model). Run system checks to verify GPU backends and dependencies.

Storage Paths — Configure where exports, components, models, user content, and plugins are stored. Five user-configurable paths, all under %LOCALAPPDATA%\AiBenchLab\ by default.

Preferences

Cost Settings — Electricity rate ($/kWh), system overhead (watts), and cost warning threshold.

Export Defaults — Default export format, output directory, and whether to remember the last export location.

Display — Theme (dark/light), language, and UI density.

---

10. Export and Reports

Export Formats

Format	Description	Trial	Pro	Consultant
PDF	Professional benchmark report	Watermarked	Yes	Yes (white-label)
MBX	Tamper-proof signed package	Yes	Yes	Yes
JSON	Machine-readable results	No	Yes	Yes
CSV	Spreadsheet-compatible	No	Yes	Yes

PDF Reports

PDF reports include an executive summary, composite Score, per-domain breakdowns, individual test results, and session metadata. Consultant tier and above can remove AiBenchLab branding (white-label) and customize report sections.

Comparison Reports

Run the same suite against multiple models, then select sessions to compare. The comparison report shows side-by-side domain breakdowns, pass rates, and individual test comparisons.

---

11. Licensing and Tiers

Tier Overview

Feature	Trial	Pro	Consultant	Enterprise
Price	Free (14 days)	$999 lifetime	$4,999 lifetime	Contact sales
All 254 tests	Yes (14 days)	Yes	Yes	Yes
Models per session	3	10	15	Unlimited
Pre-built suites	2	All 22	All 22	All 22 + custom
PDF export	Watermarked	Yes	White-label	White-label
MBX export	Yes	Yes	Yes	Yes
JSON/CSV export	No	Yes	Yes	Yes
Custom suites	No	No	Yes	Yes
CLI access	No	Yes	Yes	Yes
REST API	No	No	Yes	Yes
MCP Server	No	No	Yes	Yes
Plugin management	No	No	Yes	Yes
Seats	1	1	3	Site license
Support	Community	Priority email	Priority + direct	Dedicated
Annual updates	—	$399/yr	$1,999/yr	Included

How Licensing Works

AiBenchLab uses a lifetime license model — pay once, own forever. The license key is tied to your machine. Annual update subscriptions are optional — your software continues to work without them, but you won't receive new tests, features, or domain updates.

Trial

The 14-day trial provides full access to all features. After expiration, the app reverts to permanent free limits (2 suites, 3 models per session, watermarked PDF, MBX only). No credit card required.

30-Day Satisfaction Promise

Talk to us within the first 30 days, and if we can't solve it, you get a full refund. Fair enough?

12. CLI Reference

The `aibenchlab-cli` binary provides headless benchmark execution for automation and CI/CD. Requires Pro tier or higher.

Commands

run — Run a benchmark session. Key options: `--model` (repeatable), `--domain` (filter), `--tier` (quick/standard/comprehensive), `--suite_id`, `--deterministic`, `--seed`, `--export_format`, `--export_path`, `--output_json`.

export — Export a completed session: `--session_id`, `--format` (pdf/mbx/json/csv), `--output`.

list-models — List all available models across configured providers.

estimate-cost — Estimate benchmark cost without executing: `--model`, `--provider_type`, `--suite_id`.

CI/CD Integration

The CLI integrates with GitHub Actions, GitLab CI, and scripted workflows. Use `--deterministic --seed 42` for reproducible results. Use `--output_json` for machine-readable output suitable for parsing in scripts.

Output Behavior

User-facing output goes to stdout. Progress, logs, and errors go to stderr. Control verbosity with the `RUST_LOG` environment variable.

---

13. REST API Reference

The `aibenchlab-api` binary starts an HTTP server for benchmark automation. Requires Consultant tier or higher. Default address: http://127.0.0.1:8787.

Authentication

Set `AIBL_API_TOKEN` as an environment variable. All endpoints except /health require the token in the `x-aibl-token` header. If not set, requests are accepted without authentication.

Endpoints

GET /health — Health check. No auth required.

GET /models — List all available models.

POST /benchmark — Start a benchmark run. Returns a session ID immediately. Parameters: model_ids (required), session_name, domains, tier, suite_id, deterministic, seed, temperature.

POST /estimate-cost — Estimate cost without executing. Parameters: model_name, provider_name, provider_type, suite_id.

POST /export — Export a completed session. Parameters: session_id, format, output_path.

POST /export/batch — Export multiple sessions. Parameters: session_ids, format, output_path (directory).

---

14. MCP Server Reference

The `aibenchlab-mcp` binary exposes benchmarking capabilities through the Model Context Protocol. Requires Consultant tier or higher. AI tools, agents, and coding assistants can discover models, trigger benchmarks, monitor progress, and export results.

Configuration

Add to your AI tool's MCP settings:

JSON

```
{
  "mcpServers": {
    "aibenchlab": {
      "command": "aibenchlab-mcp",
      "args": []
    }
  }
}
```

Tools

The MCP server exposes 8 tools: list_models, list_providers, list_sessions, run_benchmark, get_session, get_queue, cancel_queue_item, export_report, and estimate_cost.

Resources

Read-only resources at aibl://sessions, aibl://sessions/{id}, aibl://models, and aibl://providers.

Example Workflows

Evaluate before deployment: List models → estimate cost → run benchmark → poll results → export report.

Regression testing: Run deterministic benchmark → poll → compare Score against baseline.

Cost-aware selection: Estimate costs for multiple models → select cheapest above quality threshold → verify with benchmark.

15. Troubleshooting

Installation Issues

Setup downloads fail. Check your internet connection. The setup screen downloads from GitHub and HuggingFace. If behind a corporate firewall, ensure these domains are accessible.

GPU not detected. Run the system check in Settings > Components. For NVIDIA GPUs, verify your driver is up to date. The CUDA backend requires CUDA 13.1+.

Provider Issues

Ollama not detected. Verify Ollama is running: `ollama list` should return your models. Check the port: `curl http://localhost:11434/api/tags`.

LM Studio not detected. Ensure the local server is running and a model is loaded. LM Studio won't respond to API calls without a loaded model.

Cloud provider authentication failed (401/403). Verify your API key is correct and active. Check that the key has the required permissions.

No models found. The provider may require a paid plan. For custom endpoints, verify the /models endpoint returns data.

Benchmark Issues

Built-in server fails to start. Run the system check. If CUDA errors appear, try the Vulkan backend. For out-of-memory errors, try a smaller model or reduce GPU layers.

Benchmark pauses or aborts. Thermal protection is active: benchmarks pause at 80°C and abort at 95°C. Ensure adequate cooling for long benchmark sessions.

Rate limiting on cloud providers. AiBenchLab does not retry rate-limited requests during benchmarks. Use a provider with higher rate limits or run fewer tests.

Export Issues

PDF export fails. Ensure the export directory exists and you have write permissions. Check available disk space.

MBX export during trial. MBX export is allowed during the trial period. All other export formats are blocked after trial expiration except watermarked PDF.

16. Appendix

Valid Domain Names

Use these in the CLI, REST API, and MCP Server:

reasoning, code, chat, tool_calling, adversarial_safety, deployment_risk, agentic, multimodal, multi_turn_adversarial, agentic_email, context_retention

Export Format Summary

Format	Extension	Machine-Readable	Importable	Trial
PDF	.pdf	No	No	Watermarked
MBX	.mbx	Yes	Yes	Yes
JSON	.json	Yes	No	No
CSV	.csv	Yes	No	No

Getting Help

- Documentation: aibenchlab.com/docs
- Support: aibenchlab.com/contact
- YouTube walkthroughs: youtube.com/@aibenchlab
- Email: support@aibenchlab.com

Resources

- Documentation: **aibenchlab.com/docs**
- Support: **aibenchlab.com/contact**
- YouTube walkthroughs: **youtube.com/@aibenchlab**
- Email: **support@aibenchlab.com**

AiBenchLab is built by a solo founder with 40+ years of software engineering experience. Every feature is designed with the conviction that you deserve honest, transparent, and reproducible AI model evaluations. No hype. Just truth.