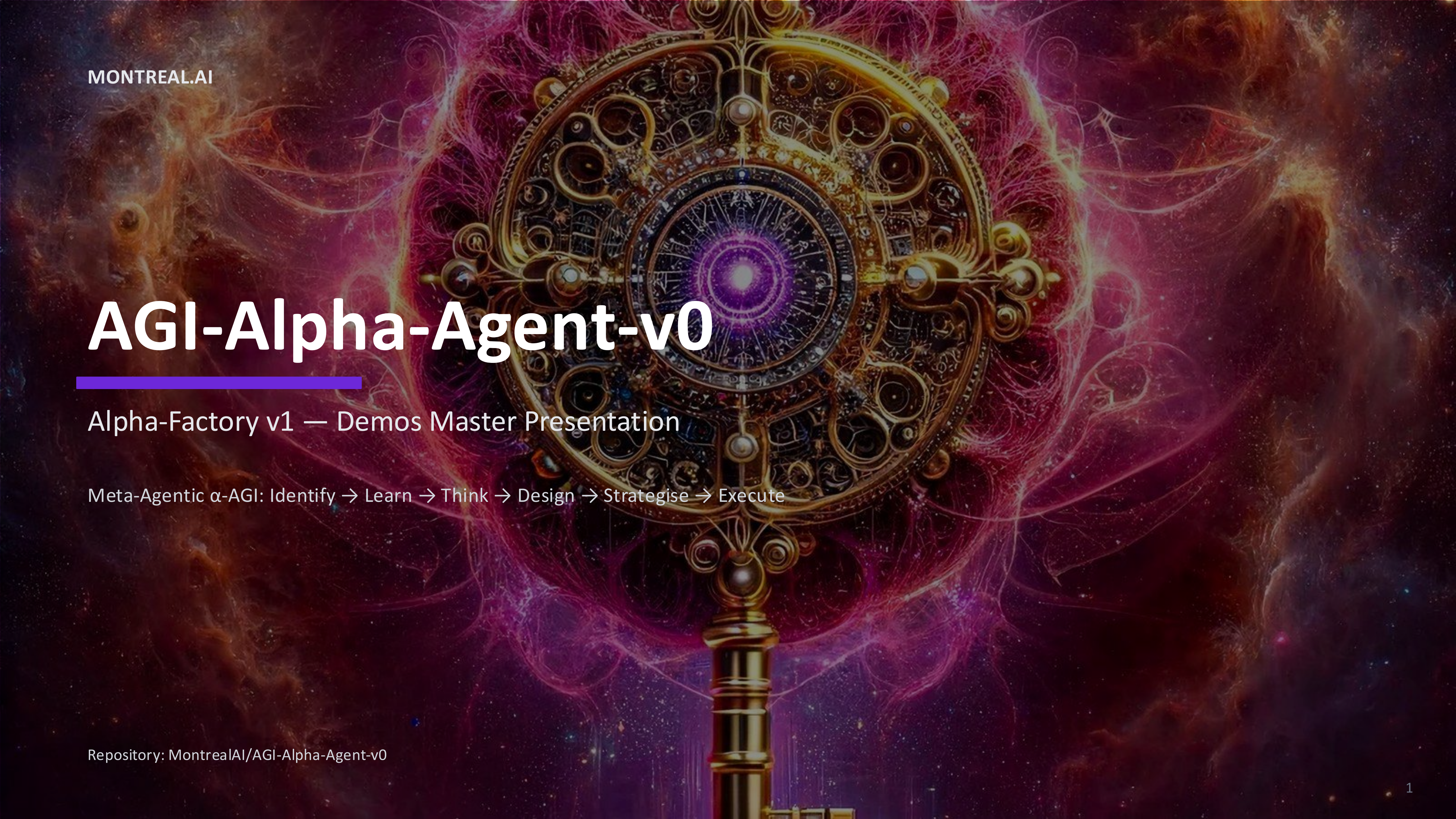


MONTREAL.AI

AGI-Alpha-Agent-v0

Alpha-Factory v1 — Demos Master Presentation

Meta-Agentic α-AGI: Identify → Learn → Think → Design → Strategise → Execute

Repository: MontrealAI/AGI-Alpha-Agent-v0

One orchestrator. Many domains.

Compounding loop: discover → decide → deploy → measure → improve.

Discovery

Turns weak signals into structured hypotheses

Ranks opportunities under uncertainty

Converts inefficiencies into missions with measurable objectives

Planning

World-model + search for counterfactual evaluation

Schedules specialists (agents/tools) for maximum leverage

Chooses interventions with quantified risk budgets

Execution

Tool-integrated swarm: research, code, market ops

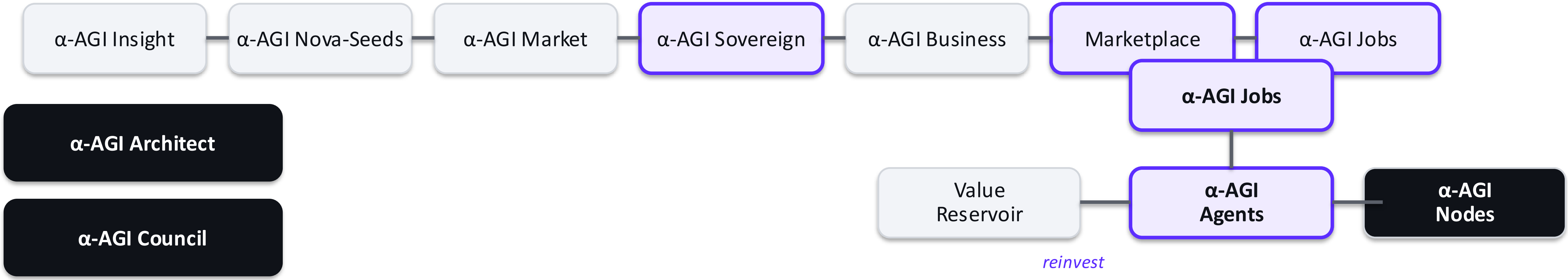
Reproducible deployment: local-first & Docker-native

Self-healing loops and meta-evolution for continuous improvement

Source: MontrealAI/AGI-Alpha-Agent-v0 README (main).

Meta-agentic systems

Second-order agents that create, select, evaluate, and re-configure other agents — and the rules governing their interactions.

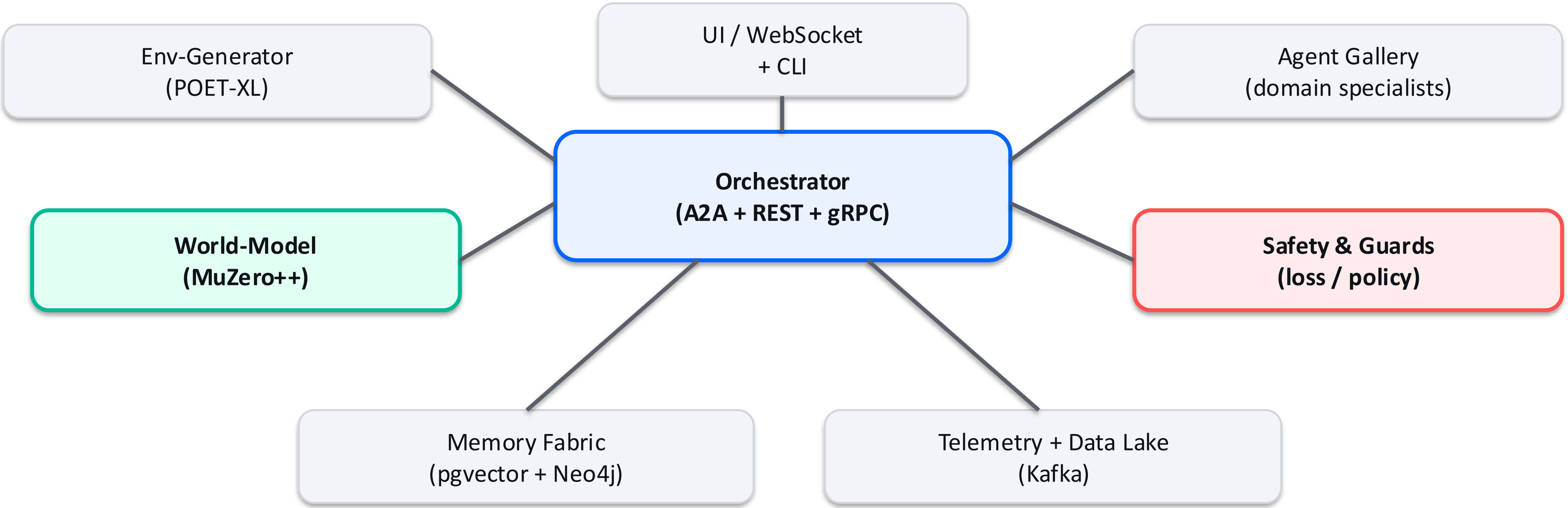


This structure turns agent swarms into an adaptive institution: not a single model, but a self-updating ecosystem.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (demo showcase).

Alpha-Factory v1 architecture

Modular, tool-integrated multi-agent stack with memory, telemetry, and deployable interfaces.



Notes: Orchestrator auto-discovers agents and exports a unified façade. World-model enables counterfactual planning; memory fabric supports dense + causal recall.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (architecture + demos).

Domain agents (archetypes)

Specialists composed into an orchestrated swarm. Roles are modular: add, swap, or evolve as the mission changes.

Planner

agent

Decompose objectives into executable plans with milestones, budgets, and stop conditions.

Research

agent

Ingest and synthesize open sources; produce citations and uncertainty bounds.

Strategy

agent

Game-theoretic reasoning: opponents, incentives, equilibria, commitment moves.

Market

agent

Detect regime shifts; translate signals into tradeable hypotheses and tests.

Builder

agent

Generate code, pipelines, and interfaces; integrate tools and APIs safely.

Operator

agent

Run deployments; monitor health, latency, and outcomes; trigger rollbacks.

Safety

agent

Guardrails: policy checks, sandboxing, red-team probes, anomaly halts.

Memory

agent

Long-horizon recall with provenance; retrieval for planning and audits.

Evaluator

agent

Metrics, ablations, and scoring; converts results into decisions.

Governance

agent

Rules and incentives; proposals, permissions, and accountable execution.

Simulator

agent

Synthetic worlds for open-ended training; curriculum and stress-tests.

Liaison

agent

Human interface: reporting, narratives, and decision support for leaders.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (agent gallery).

Demos: a complete lattice of capabilities

Fourteen runnable demos spanning world-models, autonomous enterprises, meta-evolution, governance, and operational integrity.

Discovery, world-models, evolution

alpha_agi_insight_v1 — zero-data sector ranking for AGI disruption

meta_agentic_tree_search_v0 — recursive agent rewrites via best-first search

aiga_meta_evolution — agents evolve new agents; genetic tests score fitness

alpha_asi_world_model — MuZero-style world-model in an open-ended grid-world

muzero_planning — MuZero in 60 seconds; online world-model + MCTS

era_of_experience — lifelong RL mixing real + synthetic experience streams

cross_industry_alpha_factory — ingest → plan → act across multiple verticals

Autonomous enterprise & operations

alpha_agi_business_v1 — end-to-end digital-first company formation

alpha_agi_business_2_v1 — business iteration with live market RAG

alpha_agi_business_3_v1 — forecasting & fundraising swarm; Ω-grade framing

alpha_agi_marketplace_v1 — peer-to-peer agent marketplace; price discovery

finance_alpha — momentum + risk-parity bot; stress-tests against tail risk

macro_sentinel — news scanner that proposes hedges for macro shocks

self_healing_repo — CI fails → agent patch → PR green again

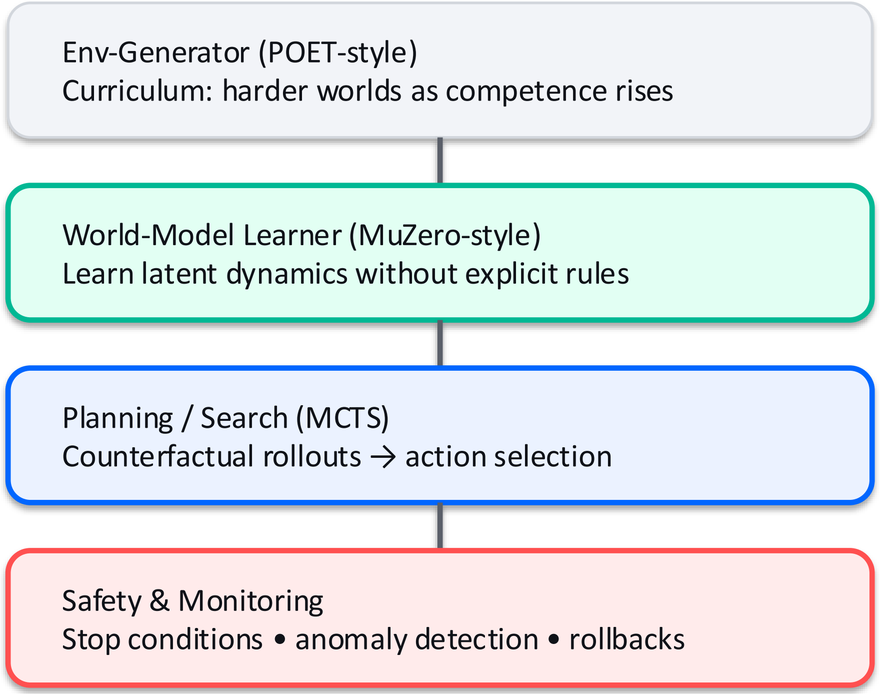
Source: MontrealAI/AGI-Alpha-Agent-v0 README (demo table).

Open-ended learning and world-models

From first principles: minimize surprise, search counterfactuals, and accumulate experience at scale.

α-ASI World-Model: open-ended learning loop

A runnable prototype: generate worlds → learn dynamics → plan actions → measure outcomes → iterate.



Env-Generator (POET-style)
Curriculum: harder worlds as competence rises

World-Model Learner (MuZero-style)
Learn latent dynamics without explicit rules

Planning / Search (MCTS)
Counterfactual rollouts → action selection

Safety & Monitoring
Stop conditions • anomaly detection • rollbacks

Key properties

Open-endedness: the curriculum expands as agents succeed
World-model learning: latent dynamics for prediction and control
Counterfactual planning: search in imagination before acting
Antifragility: detect instability (NaNs, divergences) and recover safely
Local-first: offline mode, reproducible installs, deterministic runs

First-principles framing

Minimize expected surprise (free energy) subject to safety constraints.
Treat planning as a search over low-entropy trajectories with bounded risk.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (alpha_asi_world_model, muzero_planning).

MuZero planning — distilled

A minimal, runnable reference: learn latent dynamics, search in imagination, then act.

1. Represent

Encode observations into a latent state that captures what matters for prediction.

2. Imagine & search

Roll forward the latent dynamics and evaluate actions via MCTS (counterfactual rollouts).

3. Act & learn

Execute the best action, observe outcomes, and refine the model for the next decision.

```
./alpha_factory_v1/demos/muzero_planning/run_muzero_demo.sh
```

Source: MontrealAI/AGI-Alpha-Agent-v0 README (muzero_planning).

Era of Experience + AIGA meta-evolution

Lifelong learning at scale: experience streams + evolutionary search over agent designs.

Era of Experience

Blend real + simulated experience into a single learning ledger
Ground rewards in measurable outcomes (health, cost, time, risk)
Support tool use and non-human reasoning via structured interfaces

AIGA meta-evolution

Generate agent variants (mutate / recombine / re-prompt)
Score fitness with automated tests and task suites
Select and reintegrate winners into the agent gallery

Autopoiesis framing
The system maintains itself by continually producing better components (agents) and tighter constraints (tests, safety gates, incentives).

Source: MontrealAI/AGI-Alpha-Agent-v0 README (era_of_experience, aiga_meta_evolution).

Autonomous enterprises and compounding loops

Identify alpha → allocate agents → execute → verify → reinvest.
Built for reproducible evaluation, not demos that only work once.

α-AGI Business: v1 → v2 → v3 (Ω-grade)

Progressive demos that move from formation → iteration → capital strategy, with a shared orchestrator backbone.

Business v1

Auto-incorporates a digital-first company end-to-end.
Turns an opportunity into an executable entity.

Business v2

Iterates the business model with live market data RAG.
Adapts hypotheses into durable advantage.

Business v3

Forecasting + fundraising agent swarm.
Optimises capital stack under risk constraints.

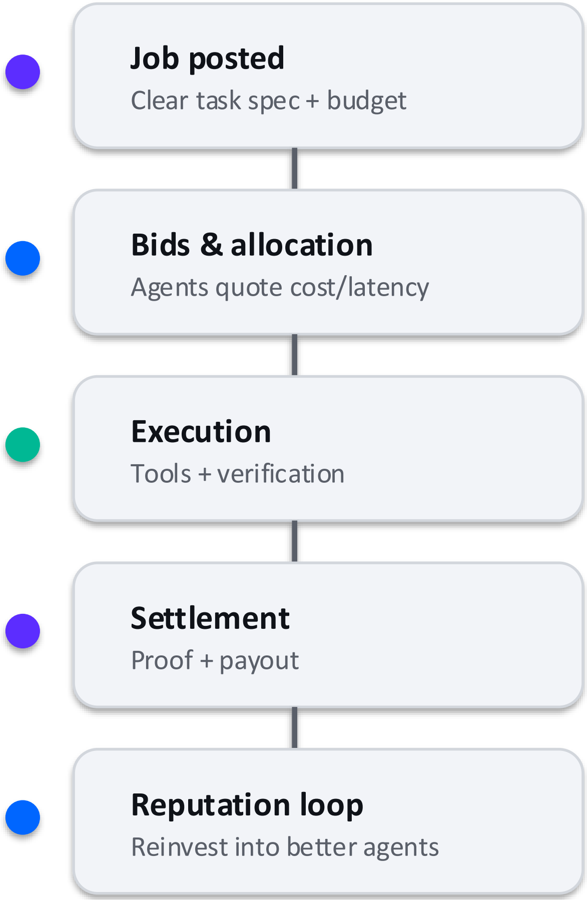
Ω-Lattice v0 framing

Model enterprise decisions as an energy landscape: allocate compute, capital, and time toward trajectories that maximize long-horizon value while minimizing downside.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (alpha_agi_business_v1/2/3, Ω-Lattice note).

Marketplace + jobs: incentives and price discovery

Mechanism design under uncertainty: agents compete, coordinate, and settle work under explicit rules.



Job posted
Clear task spec + budget

Bids & allocation
Agents quote cost/latency

Execution
Tools + verification

Settlement
Proof + payout

Reputation loop
Reinvest into better agents

Why it matters

Mechanism design: align incentives so truthful reporting is the best strategy
Game theory: handle adversarial agents and coalitions via rules + audits
Price discovery: translate uncertainty into market-clearing allocations
Verifiability: settlement depends on proofs, not promises

```
docker compose -f demos/docker-compose.marketplace.yml up
```

Source: MontrealAI/AGI-Alpha-Agent-v0 README (alpha_agi_marketplace_v1).

Finance Alpha + Macro Sentinel

Two demos for live signal ingestion, risk control, and action under uncertainty.

finance_alpha

Momentum + risk-parity bot (test-net)

Measures real P&L in controlled conditions
Stress-tests allocation against tail risk (CVaR framing)
Logs decisions for post-mortem and auditability

```
./alpha_factory_v1/demos/finance_alpha/
deploy_alpha_factory_demo.sh
```

macro_sentinel

News-RAG scanner + hedge proposals

Monitors macro narratives and regime shifts
Proposes hedge actions (CTA futures) under constraints
Designed for fast iteration and clear attribution

```
docker compose -f demos/docker-compose.macro.yml
up
```

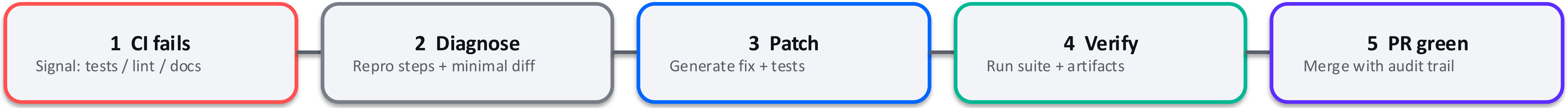
Source: MontrealAI/AGI-Alpha-Agent-v0 README (finance_alpha, macro_sentinel).

Operational integrity

Local-first. Observable. Verifiable. Designed to survive contact with reality.

Self-healing repository

When CI fails, agents propose a patch that restores green checks — and surfaces root causes faster.



1 CI fails
Signal: tests / lint / docs

2 Diagnose
Repro steps + minimal diff

3 Patch
Generate fix + tests

4 Verify
Run suite + artifacts

5 PR green
Merge with audit trail

Implementation note

The demo illustrates an operational pattern: fast diagnosis → controlled patching → reproducible verification.
In the current repo version, the underlying MetaRefinementAgent simulates improvement by generating placeholder diffs; replacing this with genuine optimisation is an open contribution path.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (self_healing_repo note).

α-AGI Insight: zero-data sector ranking

Forecast sectors primed for AGI transformation — with offline and online modes.

Method

Runs as a browser demo (Pyodide offline simulation) or with an API key
Uses scenario shocks + Monte-Carlo style runs to score sensitivity
Produces a ranked list with rationale and uncertainty notes
Designed for reproducible evaluation and iteration

```
alpha-agi-insight-v1 --episodes 5
```

Output format

Sector	Shock	Score
Semiconductors	S10	0.92
Energy storage	S10	0.88
Insurance	S10	0.83
Legal services	S10	0.79

Browser demo: montrealai.github.io/AGI-Alpha-Agent-v0/alpha_agi_insight_v1/

Source: MontrealAI/AGI-Alpha-Agent-v0 README (Quick Demo + alpha_agi_insight_v1).

Reproducible deployments

Local-first, Docker-native, and verifiable release practices to support serious evaluation.

Local & offline

Wheelhouse builds for air-gapped hosts
Offline mode flags prevent network access
CSV snapshots ship for offline demos

Docker & one-click

One-click script deploys the stack
Compose services expose /healthz and /status
Pinned versions across Python + Node lockfiles

Verifiable releases

CI runs multi-OS smoke tests and docs builds
Images can be signed and verified
Locked requirements enable deterministic installs

```
./alpha_factory_v1/scripts/one_click_install.sh
--deploy
|
cosign verify
ghcr.io/montrealai/agi-insight-demo:latest
```

Source: MontrealAI/AGI-Alpha-Agent-v0 README (offline wheels, one-click install, cosign verify).

Why these demos matter

Not isolated toys: a compounding capability stack designed for evaluation, iteration, and deployment realism.

Institutional discipline

Reproducible runs (offline mode, pinned deps, deterministic outputs)
Observable systems (telemetry, logs, health endpoints)
Verifiability (signed artifacts, auditable decisions)
Failure-tolerant workflows (self-healing patterns)

Strategic leverage

One orchestrator coordinating many specialist agents
World-model + search enables counterfactual decision-making
Markets + incentives convert uncertainty into allocation decisions
Meta-evolution expands the strategy space over time

North star

A system that can continuously identify opportunities, allocate specialists, execute safely, and reinvest gains into better capabilities.

Source: MontrealAI/AGI-Alpha-Agent-v0 README (demo suite + deployment notes).

Sources & reproducibility notes

Grounded in public documentation. Designed for repeatable evaluation.

Primary sources

GitHub repo: github.com/MontrealAI/AGI-Alpha-Agent-v0
Docs site + demo gallery: montrealai.github.io/AGI-Alpha-Agent-v0/
Explore demos: /alpha_factory_v1/demos/
Quick demo (no setup): /alpha_agi_insight_v1/

Reproducibility primitives

Offline wheelhouse workflow for air-gapped runs
Smoke tests validate a short simulation end-to-end
Lockfiles pin Python + Node dependencies
Health endpoints support automated verification

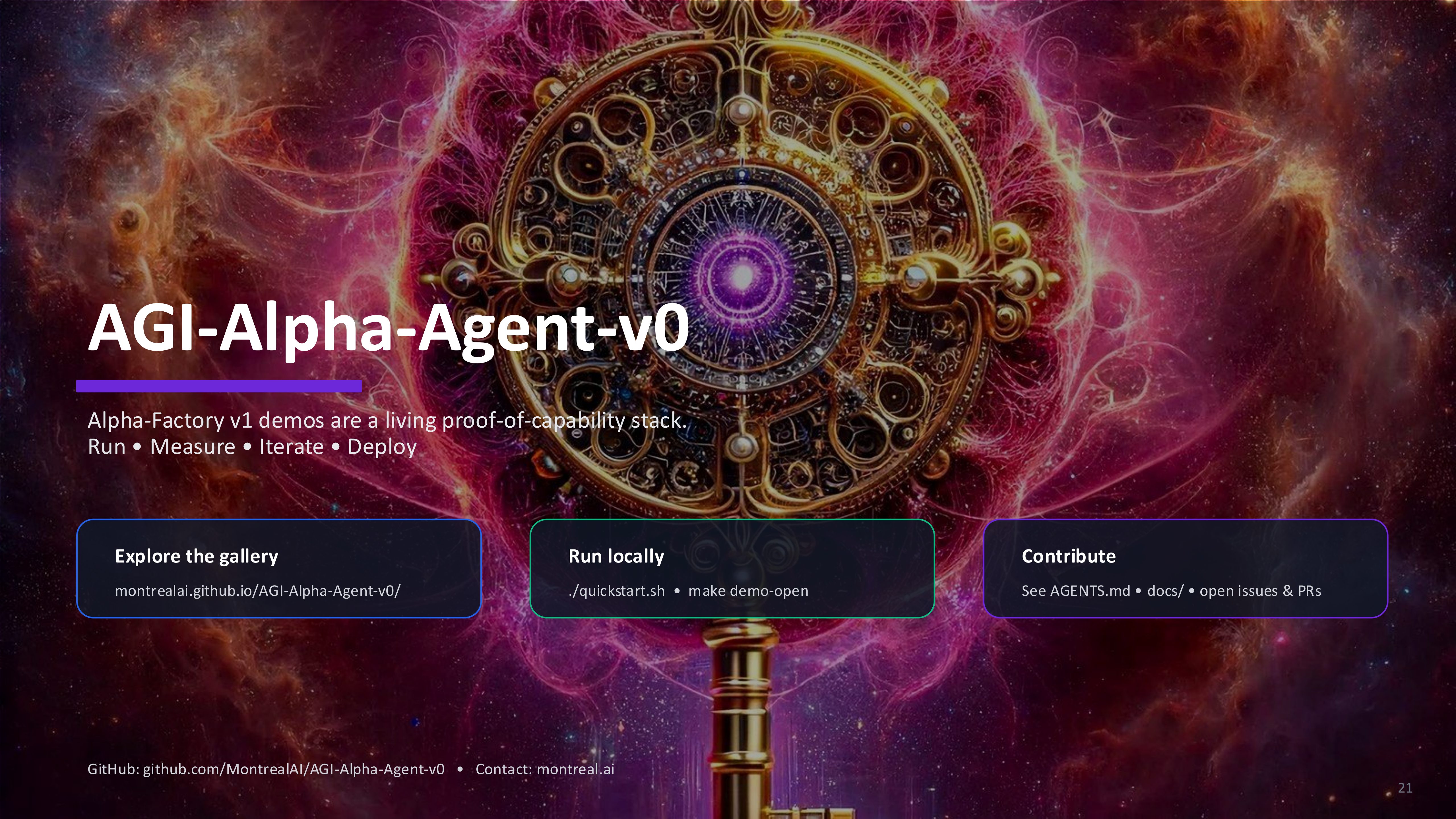
Design reference

Layout discipline, whitespace, and diagram-first storytelling inspired by DeepMind product launches (e.g., AlphaFold).

Caveats

Demos are research prototypes; validate before production use.
Some modules simulate behavior; others integrate real tooling.
Treat high-level framings (energy landscapes, free energy) as decision metaphors unless formally specified.

Source: MontrealAI/AGI-Alpha-Agent-v0 README + docs pages.



AGI-Alpha-Agent-v0

Alpha-Factory v1 demos are a living proof-of-capability stack.
Run • Measure • Iterate • Deploy

Explore the gallery

montrealai.github.io/AGI-Alpha-Agent-v0/

Run locally

./quickstart.sh • make demo-open

Contribute

See AGENTS.md • docs/ • open issues & PRs

GitHub: github.com/MontrealAI/AGI-Alpha-Agent-v0 • Contact: montreal.ai