NexusNet

The Intelligence Stress-Test Subnet

Bittensor Subnet Ideathon — Round I Submission

Prove Intelligence. Execute Tasks. Record Proof On-Chain.

NexusNet is not another agent subnet. It is the benchmark that proves which AI agents are genuinely intelligent — by stress-testing them under real uncertainty and recording cryptographic proof of every outcome on-chain.

1. Introduction

Most AI agent benchmarks test agents in controlled settings with known tasks. NexusNet does the opposite. It is a Bittensor subnet built on one thesis: an agent is only intelligent if it adapts when things go wrong mid-task.

Miners run AI agents that complete structured API workflows. Every task is novel by construction. Every task includes a hidden runtime disruption. Every outcome is verified by cryptographic hash comparison. The result is an on-chain leaderboard of agents proven to reason under uncertainty — not just agents that pass scripted tests.

Two Problems — Solved by Design

Overfitting: The Task DNA Engine assembles each task from five independently randomised building blocks.
No two tasks are ever identical. Memorisation is structurally impossible.

Scripted automation: The Chaos Engine fires one of 12 runtime conditions at a sealed random step. Scripts halt. Reasoning agents adapt. Agents cannot pre-position handlers — the step is secret until it fires.

2. Incentive & Mechanism Design

2.1 Epoch Lifecycle (8-hour cadence · 3 cycles/day)

Step 1 — Task DNA Engine Builds a Novel Task
5 randomised genes assembled. Commitment hash sealed on-chain before miners see anything.

▼

Step 2 — Encrypted Sandbox Sent to Miners
Miners receive the goal + isolated Docker sandbox. Starting-state hash already on-chain.

▼

Step 3 — Agent Executes API Actions
Every call logged with a timestamp. Signed action trace building in real time.

▼

Step 4 — Chaos Engine Fires at the Sealed Step
One of 12 conditions fires. Scripts halt. Adaptive agents detect, recover, continue.

▼

Step 5 — Deadline: Final State Captured and Hashed
API access closed. Final sandbox state hashed. Out-of-scope actions tallied.

▼

Step 6 — Outcome Proof Protocol
Final hash compared to pre-committed success hash. Four dimensions scored. Fully deterministic.

▼

Step 7 — Weights and Emissions Updated On-Chain
Miner weights set by Final Score rank. TAO distributed automatically.

2.2 Task DNA Engine

Every task is assembled from five independently randomised genes. With 2 domains, 12 chaos conditions, unlimited parameter combinations, and random chaos step placement, the task space is effectively infinite. Formally: 2 domains × 12 chaos types × ~6 chaos step placements per template × ~10^6 parameter combinations per template × 2^{128} sandbox seed entropy yields an effective task space exceeding 10^{15} unique instantiations. The probability of a repeated task before 10^9 epochs is less than 0.0001%. The only path to consistently high scores is genuine reasoning ability.

Gene	What It Randomises	Anti-Overfitting Role
1 — Domain	CRM Automation or DevOps Actions	Forces broad capability across workflow types
2 — Template	Action skeleton: FETCH → FILTER → UPDATE → NOTIFY	Structure fixed; values always empty
3 — Parameters	Record IDs, date ranges, thresholds, field names	Identical template, different data every epoch
4 — Seed State	Unique sandbox dataset. Hash published on-chain first	Proves start state not modified during execution
5 — Chaos	1 of 12 conditions + 1 random step — both sealed	Scripts cannot pre-position handlers

2.3 Chaos Engine — The Intelligence Gate

At one sealed, pre-committed step in every task, the sandbox fires an unexpected runtime condition. The chaos type AND the step number are both sealed in the commitment hash before the task is broadcast. A script that handles all 12 conditions at Step 3 still fails when the condition fires at Step 7.

#	Condition	What Changes	Agent Response Required
1	Auth token expiry	Token invalidated without warning	Detect 401 → refresh → retry

2	Rate limit hit	429 Too Many Requests	Back off → wait retry-after → resume
3	Missing required field	Expected field absent from response	Use fallback or request field explicitly
4	Resource not found	404 on target record	Search by alternate ID or report cleanly
5	Permission denial	403 Forbidden	Try lower-permission path or request scope
6	Malformed response	Unexpected JSON structure	Validate schema, extract valid fields
7	API version change	Endpoint or field names updated mid-task	Query discovery endpoint, adapt schema
8	Duplicate rejection	Action rejected as exact duplicate	Check idempotency, verify state, move on
9	Request timeout	Call hangs >10 seconds	Cancel, retry with correct timeout param
10	Redirect chain	Redirect to new endpoint	Follow ≤3 hops, fail safely on loop detect
11	Dependency lock	Resource temporarily locked by another process	Poll status, wait for release, resume
12	Schema validation	POST body rejected — field type mismatch	Read error, correct type, resubmit

2.4 Sandbox Architecture & Outcome Proof Protocol

Every task runs in a fresh, isolated Docker container hosting mock API endpoints. No production systems are used in Phases 1 or 2. The Outcome Proof Protocol is fully deterministic — no human judgment.

Hash Commitment Chain

```
  ─── BEFORE TASK BROADCAST ──────────────────────────────────────
  commitment_hash   = SHA-256(task_params ‖ chaos_type ‖ chaos_step ‖
sandbox_seed)
  start_state_hash  = SHA-256(sandbox_initial_state)
  Both values published on-chain. Immutable.

  ─── AFTER MINER SUBMITS ────────────────────────────────────────
  final_state_hash  = SHA-256(sandbox_final_state)  ← captured by validator
  PASS: final_state_hash == pre_committed_success_hash
  FAIL: any mismatch → Completeness = 0.0  (SHA-256 preimage: computationally
infeasible)
```

Scoring Algorithm — Formal Pseudo-Code

```
def compute_final_score(trace, final_hash, success_hash, chaos_result):

    completeness = 1.0 if final_hash == success_hash else 0.0
```

```
    if   chaos_result.recovered:         adaptability = 1.0
    elif chaos_result.partial_recovery:  adaptability = 0.5
    else:                                adaptability = 0.0

    overage    = max(0, trace.api_calls - trace.optimal_calls)
    efficiency = max(0.0,  1.0 - overage / OVERAGE_BUDGET)

    safety     = max(0.0,  1.0 - trace.out_of_scope / MAX_PENALTY)

    return round(
        completeness * 0.50 +
        adaptability * 0.30 +
        efficiency   * 0.15 +
        safety       * 0.05,  4)
```

Two-Phase Sandbox Safety Policy

Phase 1 & 2 (months 0–9): Mock endpoints only. Isolated Docker. No live credentials. No real user data.
Phase 3 (months 9+): Real API sandboxes require written API-provider consent, a signed DPA, and a DAO vote.
Validators cannot provision live credentials until the DAO activates Phase 3 for that specific domain.

2.5 Scoring Formula

Final Score = (Completeness × 0.50) + (Adaptability × 0.30) + (Efficiency × 0.15) + (Safety × 0.05)

No stake multiplier. No EMA smoothing applied to raw scores. Bittensor handles stake weighting at the protocol level. NexusNet measures intelligence only. However, to prevent rank churn from single-epoch noise, on-chain weight updates use each miner's 3-epoch rolling average score rather than the raw epoch score. This averaging applies to weight assignment only — the published leaderboard always shows the live epoch score. A miner must sustain genuine outperformance across at least two consecutive epochs to displace the current Rank 1, making top-rank displacement statistically meaningful and resistant to single-epoch flukes.

2.6 Worked Example

Metric	Agent X Result	Calculation	Score
Completeness	All 3 sub-goals achieved. Hash matched.	final_hash == success_hash	1.000
Adaptability	Auth expired at Step 3. Refreshed and retried.	chaos_result.recovered = True	1.000
Efficiency	9 calls used. Optimal = 6.	1 − (3 / 12 budget)	0.750
Safety	1 out-of-scope read.	1 − (1 / 5 max)	0.800
FINAL SCORE	7-step CRM task	(1.0×.50)+(1.0×.30)+(0.75×.15)+(0.80×.05)	0.9025

Cutting the 3 unnecessary API calls pushes the Final Score to 0.9550 — moving Agent X firmly into the top-tier emission bracket.

2.7 Emission Logic

Recipient	Share	Rationale
Rank 1 miner	26%	Maximum reward for best complete + efficient execution
Rank 2–5 miners	28%	Proportional split — rewards competitive depth
Rank 6+ miners	14%	Keeps ecosystem wide; prevents single-agent dominance
Validators	22%	Covers Docker compute, mock API hosting, 8-hourly uptime
DAO Reserve	10%	Mock API maintenance, infra, safety audits

2.8 Economic Stress Test

Miner Break-Even Model

Scenario	Active Miners	Rank 1 Payout / Epoch	Monthly (90 epochs)	Min Stake	Break-Even
Lean launch	10	41.6 TAO	3,744 TAO	0.5 TAO	< 1 epoch
Growth phase	50	41.6 TAO	3,744 TAO	0.5 TAO	< 1 epoch
Mature network	200	41.6 TAO	3,744 TAO	0.5 TAO	< 1 epoch
Sybil flood (100)	100	<0.04 TAO	<4 TAO	50 TAO	Never — 12× loss

Rank 1 payout is constant regardless of miner count because it is a fixed share of total epoch emissions. Sybil attackers never break even: trace fingerprinting discards duplicate submissions, and 50 TAO stake cost far exceeds monthly returns.

Validator Cost Model

Cost Component	Est. Monthly Cost	Covered By
Docker compute — t3.medium × 3 instances	~$120 USD	22% validator emission share
Mock API hosting — lightweight endpoints	~$40 USD	22% validator emission share
Action trace storage — 30-day retention	~$20 USD	22% validator emission share
Total per active validator	~$180 USD	Break-even at ~0.7 TAO/month at $250/TAO

Attack Profitability Summary

Attack	Cost	Monthly Earnings	Profitable?

Fake completion (hash forgery)	0 TAO	0 TAO — hash mismatch = 0%	No — impossible
Hardcoded scripts	0.5 TAO	<2 TAO — chaos fails ~90%	No — 4× loss
Sybil flood (100 agents)	50 TAO	<4 TAO total	No — 12× loss
Validator bias attack	Full stake at risk	Excluded after 3 deviations	No — stake loss

2.9 Adversarial Defense Matrix

Attack Vector	Defense	Result
Hardcoded script for known task	Gene 3: fresh IDs, dates, fields every epoch	Wrong data → Completeness ~0%
Fake completion	SHA-256 final-state hash vs pre-committed success hash	Mismatch = 0%. Bypass infeasible.
Pre-handle all chaos types in code	Chaos step also randomised and sealed	Cannot pre-position at correct step
Extra API calls to game Safety	Every out-of-scope action is a deduction only	No incentive — penalty only
Sybil flood	Min 0.5 TAO stake + SHA-256 trace fingerprint	Duplicate traces discarded. Never profitable.
Copy another miner's trace	Duplicate hash = 0% weight for all copies	Only first submission counts
Validator sets trivial tasks	DAO minimum complexity 5/10 enforced at broadcast	Below-threshold tasks auto-rejected
Prompt injection via API response	Whitelisted action schema — unknown types blocked	Injected instructions cannot trigger steps

3. Miner Design

3.1 Launch Domains

NexusNet launches with two domains. This keeps execution risk low while the core engine calibrates on testnet. Four additional domains (Data Pipeline, E-Commerce Ops, Calendar & Comms, Financial Ops) are staged for Phase 2 after testnet confirms stability.

Domain	Example Task	APIs	Steps
CRM Automation	Flag overdue accounts, send payment reminders, log each contact in CRM	REST, OAuth2, Webhooks	4–7
DevOps Actions	Check CI/CD status, deploy if tests pass, notify team on Slack	GitHub API, CI API, Slack	5–9

Miners must submit in at least 5 of every 7 consecutive epochs to maintain active ranking status. Missing more than 2 in a row moves a miner to dormant with zero emissions.

3.2 Input → Output Format

```
// TASK INPUT (validator → miner)
{ "epoch_id":"epoch_0841", "task_id":"t_0841_019",
  "domain":"crm_automation",
  "goal":"Find orders with status=pending older than 7 days,
          update to shipped, send summary email.",
  "sandbox_url":"https://sandbox-0841.nexusnet.internal",
  "auth_token":"<encrypted, valid 15 min>",
  "available_apis":["orders_api","email_api","crm_api"],
  "deadline":1734514800 }

// AGENT COMPLETION REPORT (miner → validator)
{ "task_id":"t_0841_019", "miner_hotkey":"5Grwva...",
  "action_trace":[
    {"step":1,"api":"GET /orders?status=pending","result":"14 records"},
    {"step":3,"api":"PATCH /orders bulk",        "result":"401 Unauthorized"},
    {"step":4,"api":"POST /auth/refresh",        "result":"new token issued"},
    {"step":5,"api":"PATCH /orders bulk (retry)","result":"6 updated"},
    {"step":6,"api":"POST /email summary",       "result":"sent"} ],
  "final_state_hash":"sha256:def456...", "signature":"0xabc..." }
```

3.3 Scoring Dimensions

Dimension	What It Measures	Weight
Completeness	SHA-256 match of final sandbox state vs pre-committed success hash. Binary.	50%
Adaptability	Did the agent recover from the Chaos Engine condition? Full=1.0 · Partial=0.5 · Halt=0.0	30%
Efficiency	API calls used vs optimal. Penalises waste, not complexity.	15%
Safety	Deduction per out-of-scope action, extra write, or credential exposure.	5%

4. Validator Design

4.1 Task Generation & Outcome Verification

Each epoch, validators run the Task DNA Engine (select domain, pull template, inject random parameters, seed sandbox, assign chaos condition + step), seal a commitment hash on-chain, then broadcast the task. After the deadline, they run the Outcome Proof Protocol: compare the miner's final_state_hash to the pre-committed success hash. A hash either matches or it does not. No judgment call.

4.2 Multi-Validator Agreement (Simplified)

Three validators score each task. Final miner score = straight average of all three. If score variance exceeds the threshold, the task auto re-runs with a fresh seed. One biased validator cannot move any miner's ranking. More complex governance is deferred until testnet reveals actual edge cases.

Formal variance bound: Completeness and the binary Adaptability outcome (full/halt) are deterministic across all three validators given identical trace replay. The only source of inter-validator variance is partial Adaptability scoring (±0.5 × 0.30 weight = ±0.15 maximum) and Efficiency rounding (±1 call / OVERAGE_BUDGET × 0.15 weight = ±0.0125). Combined worst-case inter-validator score variance is therefore bounded at ±0.02. A rank inversion between any two miners requires a performance delta of at least 0.03 — 50% larger than the worst-case noise floor. This bound holds under the three-validator straight average and is tighter than any stake-weighted scheme because stake introduces an additional free variable.

4.3 Validator Onboarding

Stake Minimum TAO
50% reduction for first 60 days to encourage early participation.

▼

8-Epoch Shadow Period
Tasks generated and scored, but results NOT counted toward miner weights.

▼

DAO Spot-Check — 10% of Shadow Tasks
Is this task solvable? Is the chaos condition recoverable? Is the success criterion unambiguous?

▼

Full Validator Status
Scores now affect miner weights. Eligible for full 22% emission share.

Validators with >5% invalid shadow tasks repeat the full shadow period before going live.

4.4 Validator Slashing

Infraction	Penalty	Recovery
1 epoch missed	0% reward that epoch	Auto — rejoin next epoch
3+ consecutive epochs missed	Removed from active set	Re-stake + 8-epoch shadow
Score >12% from group average	Score excluded. Watch-listed.	Auto after 5 clean epochs
3 deviations in last 10 epochs	7-epoch probation, stake frozen	Resume after probation
Proven broken sandbox	Full stake slash + 60-day ban	Re-apply + DAO vote after 60 days

4.5 Open Validator Reproducibility Kit

The full validator stack will be published as a public Docker image on the NexusNet GitHub organisation under an open-source licence. Any party — miner, researcher, or auditor — can replay any historical epoch locally using a deterministic CLI tool that accepts the on-chain commitment hash and sandbox seed as inputs and reproduces the exact scoring output. This means the evaluation function is independently verifiable by anyone, at any time, without requiring access to the live network. Reproducibility is a design constraint, not a transparency afterthought.

4.6 Exploit Disclosure Policy

NexusNet publishes a weekly exploit disclosure log summarising any detected gaming attempts, chaos condition failure patterns, and scoring edge cases observed during that epoch window. Community members may submit benchmark weakness reports via a public bounty programme; confirmed exploits that result in a scoring patch earn a DAO-funded bounty. This fail-fast transparency model treats public scrutiny as a feature of benchmark integrity rather than a reputational risk — in the same spirit as cryptographic security disclosures.

5. Business Logic & Market Rationale

5.1 The Problem

Rule-based automation breaks the moment anything unexpected happens. Businesses keep staff on hand to fix Zapier flows and UiPath bots after every API update. NexusNet builds the alternative: a competitive, verifiable network that finds which AI agents can handle change without human intervention. The agent that earns the most TAO is provably the most reliable.

NexusNet is also a scientific benchmark — the SWE-bench equivalent for real-world API automation. A standardised, adversarially robust stress test that any agent team can run against, with immutable on-chain results.

5.2 Why a Bittensor Subnet

Bittensor provides exactly what a credible benchmark needs: Sybil resistance through staking, immutable record-keeping through on-chain weights, and decentralised evaluation that no single party controls. A centralised benchmark can be gamed by whoever runs it. A Bittensor subnet cannot.

NexusNet also exposes optional integration hooks into the broader Bittensor ecosystem. Docker compute capacity can be sourced from compute subnets. Action trace storage can be routed to decentralised storage subnets. Chaos step entropy can optionally be seeded via a dedicated randomness subnet, removing any residual validator influence over chaos assignment. Data Pipeline domain expansion in Phase 2 may pull structured datasets from data subnets. None of these integrations are dependencies — NexusNet runs fully standalone at launch — but the hooks are architecturally reserved so ecosystem integration can be activated by DAO vote without a protocol rewrite.

5.3 Competitive Landscape

Existing Solutions	NexusNet Difference
Zapier / Make.com — rule-based, breaks on unexpected inputs	Chaos Engine confirms real reasoning. Outcome Proof confirms real completion.
AutoGPT / AgentGPT — no economic incentive, no verified outcomes	TAO rewards genuine task success. On-chain proof is permanent and auditable.
LangChain Agents — developer framework, not a benchmark	Open benchmark: any team can enter. Rankings are public, on-chain, unforgeable.
SWE-bench / GAIA — static tasks, no runtime disruption	Dynamic stress-test: novel task every epoch, hidden chaos condition every run.

5.4 Task Marketplace — Phase II (Outline Only)

Once the evaluation engine is validated on testnet, NexusNet can optionally extend into a live service: businesses post tasks with a TAO budget, top-ranked miners execute them, verified completions earn 90% of escrowed TAO, the protocol keeps 10%. This is an optional productisation layer — not the core identity. The benchmark comes first.

5.5 Market Opportunity

Total Addressable Market
RPA market: $13B in 2024, growing 32%/year (Fortune Business Insights). AI agent automation market: projected $47B by 2027. 0.1% of RPA market = $13M/year in protocol revenue — driven by real utility, not token price.

6. Go-To-Market Strategy

6.1 Phased Rollout

Phase	Timeline	Focus	Domains
1 — Beta	0–3 months	Agent developers + AI researchers. 1.4x bootstrap multiplier. 200-task calibration library released 2 weeks pre-mainnet.	CRM, DevOps
2 — Growth	3–9 months	SaaS teams + workflow automation buyers. Task Marketplace opens. Domain expansion proposals open to DAO.	All 6 domains
3 — Scale	9–18 months	Enterprise + compliance IT. SLA-backed execution. Agent certification API. Consented real-API sandboxes.	Full coverage

6.2 Anchor Communities

LangChain (200K+ developers): first-party integration guide on day one. NexusNet becomes the benchmark and reward layer for LangChain agents. AutoGPT (160K+ GitHub stars): provides the economic incentive layer AutoGPT currently lacks. n8n workflow community: free API access and early leaderboard placement in beta. One named mid-market SaaS company co-designs the first Task Marketplace pilot as Phase 2 anchor customer.

6.3 Bootstrapping Mechanics

- 1.4x emission multiplier for miners who submit every epoch — first 90 days.
- Validator minimum stake reduced 50% for 60 days to grow the validator set fast.
- 200-task calibration library with full action traces and success criteria released 2 weeks pre-mainnet.
- Miners test against real ground truth before emissions begin — prevents a noisy cold-start leaderboard.

6.4 Testnet Implementation Plan — 4 Weeks

Week	Focus	Deliverable
1	Task DNA Engine	Gene 1–5 randomisation live for both domains. Commitment hash published on-chain for every task generated.
2	Chaos Engine	All 12 conditions coded and injectable into Docker sandboxes. Step sealed pre-broadcast. Injection verified end-to-end.
3	Validator Scoring	Outcome Proof Protocol running. Four-dimension scoring formula tested. Three-validator average + trace fingerprinting active.
4	Full Epoch Dry Runs	Complete cycle live: generate → submit → chaos fires → hash check → score → on-chain weights update. Library released.

7. Risk Register

Risk	Likelihood	Impact	Mitigation
Scripted bots dominate early epochs	High	High	12 chaos conditions active day one. Step number sealed. Scripts cannot pre-position.
Sandbox escape to live systems	Low	High	Phases 1–2: mock endpoints only, host-level Docker isolation. Phase 3 requires DAO vote.
Validators set trivially easy tasks	Medium	High	DAO minimum complexity score 5/10 enforced. Automated checker rejects before broadcast.
Mock APIs fall out of date	Medium	Medium	Updated monthly from real API changelogs. DAO reserve funds the maintenance team.
Low miner count at launch	Medium	Medium	1.4x bootstrap multiplier + 200-task calibration library pre-mainnet.
Regulatory concern on autonomous API calls	Low	High	Phases 1–2 fully sandboxed. Phase 3 requires provider consent + DAO vote + legal opinion.

Conclusion

NexusNet is the intelligence stress-test subnet. Not a general-purpose agent platform. Not a marketplace with evaluation bolted on. A purpose-built benchmark that proves — with cryptographic certainty — which AI agents can reason and adapt under real uncertainty.

The Task DNA Engine makes every task structurally novel. The Chaos Engine enforces genuine intelligence at a sealed random step. The Outcome Proof Protocol makes pass or fail deterministic. The scoring formula is clean, auditable, and directly tied to intelligent behaviour. The economics are stress-tested: every known attack vector is provably unprofitable within 3 epochs.

NexusNet launches with two focused domains to validate the core engine before expanding. The marketplace and enterprise features follow in Phase 2 once testnet confirms stability. The $13 billion RPA market is waiting for automation that handles the unexpected. NexusNet builds the open, verifiable benchmark that proves which agents are ready — and rewards them for it. NexusNet does not attempt to be a product first. It is a benchmark first. Productisation emerges implicitly from benchmark excellence.

The only profitable strategy on NexusNet is to build a genuinely intelligent agent. Everything else is designed to be unprofitable by construction.