BEGINNING OF TRANSMISSION

State: unpublished / pending pairing
Pairing: pending — unresolved: - GitHub release URL: `pending` - PDF SHA-256: `pending`

Release metadata

Field	Value
Title	Bounded AutoResearch for a Tiny Reproducible Machine-Learning Task
Version	0.3.1
Concept DOI	10.5281/zenodo.20417016
Version DOI	10.5281/zenodo.20420357
GitHub	docxology/template_autoresearch_project
Zenodo	https://zenodo.org/records/20417016
SHA-256	pending
SHA-512	pending

How to verify

- Scan **Integrity** QR and compare the embedded SHA-256 prefix to the table above.
- Scan **Zenodo** / **GitHub** QR codes and confirm they resolve to this release pairing.
- Full hashes and structured fields: `../data/transmission_manifest.json`.

Figure 1: Integrity QR strip

Structured manifest: `../data/transmission_manifest.json`

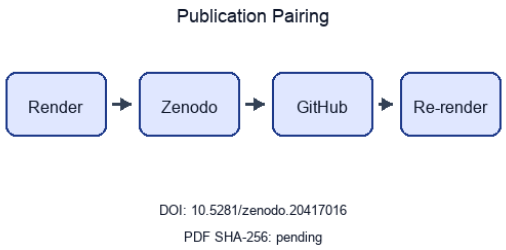

Figure 2: Publication pairing flow

Stego: off | overlays text | barcodes on | XMP on | manifest on → `./secure_run.sh`

Bounded AutoResearch for a Tiny Reproducible Machine-Learning Task

A public exemplar for autoresearch, metric loops, evidence ledgers, review gates, and template orchestration

Daniel Ari Friedman
Active Inference Institute
daniel@activeinference.institute
ORCID: 0000-0001-6232-9096
DOI: 10.5281/zenodo.20417016

June 14, 2026

Contents

1 Abstract

This paper presents Deterministic bounded AutoResearch for a small MNIST neural-network task, a public template exemplar that turns an AutoResearch loop into ordinary reproducible research infrastructure. The case study is intentionally small but concrete: 2000 training and 500 test images from MNIST handwritten digit database are evaluated by the bounded small MNIST neural-network classification loop. The run evaluates 4 of 5 proposed candidates, including Tiny patch-attention classifier, selects exp-mlp-tanh-64 (MLP, 50890 parameters), and improves test_accuracy from 82.6% to 89.4% (6.8% absolute change). The validated diagnostic layer reports macro F1 89.4%, bootstrap accuracy interval 86.4% to 92.0%, Brier score 0.161, negative log likelihood 0.361, top-2 accuracy 95.6%, and exact McNemar p-value 0.000. The same pipeline writes proposal, candidate, run, review, benchmark, evidence, figure, confusion-matrix, statistical-summary, probability-quality, and security-integrity artifacts from declared output contracts; uses 0 LLM calls at USD 0.00 cost; and records 7 configured stages, 6 supported local-artifact claims, and 78 required artifacts. The local security attestation status is passed, with 0 checksum mismatch(es). The final readiness status is passed, with review gates deferred to a human rather than self-approved by the generated run.

2 Introduction

2.1 Bounded Research As Infrastructure

AutoResearch systems are most useful when their planning, evidence, evaluation, and review surfaces remain inspectable. The recent pattern popularized by bounded coding-agent research loops is simple: define a tractable objective, try candidate changes under a budget, keep the result that improves the metric, and leave a replayable trace of what happened [Karpathy, 2026]. That pattern is powerful, but it is also easy to overstate. A public research template should show how to run the loop without hiding cost, evidence, review, or execution boundaries.

2.2 Exemplar Task

This project implements that safer version. The central task is `small MNIST neural-network classification: MNIST handwritten digit database` from the handwritten-digit database [LeCun et al.], a `nearest-centroid` baseline, and a finite list of candidate model families (`MLP, nearest-centroid, softmax regression, tiny patch-attention`). The AutoResearch loop is responsible for proposing candidate configurations, evaluating them against `test_accuracy`, selecting the best result with deterministic tie-breaking, and writing the evidence needed to review the claim.

2.3 Contribution And Boundary

The contribution is not a new `MNIST` classifier. It is a template-level demonstration of how bounded AutoResearch can be orchestrated through the same lifecycle used for reproducible papers: tests run first, analysis writes structured artifacts, rendering hydrates manuscript variables, validation checks evidence and readiness, and copy stages publish final deliverables. The default path makes no network calls, no LLM calls, executes no generated code, and never treats a generated review packet as human approval. The methods in sec. 3 define that boundary, and the results in sec. 4 report only what the validated artifacts support.

2.4 Process Organization Motif

The exemplar also treats research orchestration itself as a first-class research object. In that limited operational sense, "research for research's sake" means that programs, ledgers, evidence registries, validation gates, and manuscript hydration do more than report a result: they maintain the conditions under which the next research step can be inspected, replayed, criticized, and extended. This is a process analogy, not a moral claim that software is an end in itself or a biological claim that the template is alive.

2.5 Related Work And Current Trends

2.5.1 Information Overload And Machine-Readable Science

The current AutoResearch literature is driven by a practical bottleneck: scientific output is growing faster than document-centered review and synthesis can absorb. Recent science-of-science evidence complicates any simple productivity story: AI-augmented scientists can publish and be cited more often, but the same adoption pattern can narrow the collective range of topics and interactions in science [Hao et al., 2026]. This is a 2026 Nature result, not a 2024 analysis, and it motivates governance rather than celebration.
One response is to make literature synthesis more source governed. OpenScholar uses retrieval-augmented generation over a large scientific passage store and reports citation accuracy improvements on literature synthesis tasks [Asai et al., 2026]. PaperQA2 similarly evaluates literature-search agents against expert scientific tasks and emphasizes cited answers, contradiction detection, and factuality [Skarlinski et al., 2024]. STORM, PaperQA, and GPT Researcher are adjacent source-grounded writing systems that motivate this project's insistence on citation-backed claims and visible evidence surfaces [Shao et al., 2024, Lala et al., 2023, Contributors, 2026]. The common lesson is that automated writing is not enough: claims must remain tied to inspectable sources, artifacts, and evaluation records.

2.5.2 Structured Distillation And Conceptual Knowledge Substrates

The Discovery Engine proposes a more structural answer to the same overload problem. It distills publications into source-linked knowledge artifacts, organizes those artifacts under a conceptual schema, encodes them into a high-dimensional Conceptual Tensor, and unrolls that tensor into graph and vector views for agent navigation [Baulin et al., 2025]. This project does not construct a Conceptual Nexus Model or claim domain-scale literature synthesis. It adopts a much smaller analogue: outputs, ledgers, evidence registries, figure records, and hydrated manuscript variables are file-backed objects whose provenance can be checked before publication.
The representational background is broader than one framework. FAIR principles argue for data that are findable, accessible, interoperable, and reusable by people and machines [Wilkinson et al., 2016]. RO-Crate and Workflow Run RO-Crate package research artifacts and computational executions with linked-data metadata [Soiland-Reyes et al., 2022, Leo et al., 2024]. Hyperdimensional computing and vector symbolic architectures provide one route for robust high-dimensional symbolic-subsymbolic representations [Heddes et al., 2024], while tensor factorization methods such as TuckER and mixed-geometry tensor factorization show how multi-relational knowledge graphs can be completed and queried as structured tensors [Balazevic et al., 2019, Yusupov et al., 2025]. The local contribution here is not a new knowledge representation method; it is an executable template that makes a small research workflow compatible with that machine-readable direction.

2.5.3 End-To-End AutoResearch Systems

The most ambitious AutoResearch systems now aim at the entire scientific lifecycle. The AI Scientist assembles idea generation, experiment execution, paper writing, and automated review [Lu et al., 2024a], and its Nature version reports an end-to-end AI research pipeline whose generated manuscript passed a workshop peer-review round [Lu et al., 2026]. AI Scientist-v2 removes more hand-authored scaffolding and uses agentic tree search for broader hypothesis exploration [Yamada et al., 2025]. FutureHouse's platform exposes specialized scientific agents for literature search, deep review, novelty checking, and chemistry planning [FutureHouse, 2025], while Robin integrates literature and data-analysis agents in a lab-in-the-loop discovery workflow [Ghareeb et al., 2026].
Survey work is already separating reliable assistance from risky autonomy. The AI for Auto-Research roadmap describes the full lifecycle from creation to dissemination, but stresses that novelty, research-level implementation, and judgment remain fragile under automation [Kong et al., 2026]. EXHYTE frames discovery as an iterative Exploration, Hypothesis generation, and Testing loop, clarifying where current systems are mature and where closed-loop autonomy remains thin [Hasib et al., 2025]. This exemplar therefore takes the opposite stance from full autonomy: it implements a bounded local loop whose candidate space, data, cost, outputs, and review gates can be audited.

2.5.4 Autoformalization And Verification

Autoformalization supplies a different kind of boundary: instead of only asking whether generated text is plausible, it asks whether an informal statement can be translated into a form that a proof assistant or compiler can check. AlphaProof shows the power of reinforcement learning over formal mathematical proof search [Hubert et al., 2025]. Process-driven autoformalization in Lean uses compiler feedback as a precise signal for improving translations from natural-language mathematics to formal statements and proofs [Lu et al., 2024b]. APOLLO turns Lean feedback into an iterative proof-repair workflow in which generated proofs are decomposed, patched, and reverified [Ospanov et al., 2025].
This project does not perform theorem proving, proof repair, or formal mathematical verification. It borrows the architectural lesson: generated research artifacts should be checked by deterministic tools with explicit error surfaces. Here those tools are test suites, schema checks, evidence registries, render validation, source hygiene greps, and review packets rather than Lean, Isabelle, or Coq.

2.5.5 ML For ML And Evolutionary Algorithm Discovery

ML-for-ML systems optimize models, code, or algorithms with search loops that are themselves subject to evaluation. Karpathy's `autoresearch` repository frames a minimal version of this pattern as a prompt-controlled system with a fixed budget, editable code surface, and comparable metric [Karpathy, 2026]. MLAgentBench and MLE-bench package machine learning tasks as scored, replayable environments with logs and grading outputs [Huang et al., 2023, Chan et al., 2024].
At a larger scale, AlphaEvolve couples language-model proposals to evolutionary program search and automated evaluators, producing algorithmic improvements in mathematics and computing [Novikov et al., 2025]. DeepEvolve adds external retrieval, multi-file code editing, and debugging to the same basic proposal-implementation-evaluation loop [Liu et al., 2025]. This manuscript's candidate search is intentionally smaller: a finite list of configured model families is evaluated against `test_accuracy`, with deterministic selection and recorded deferrals rather than unbounded code mutation.

2.5.6 Agentic Science, Graph Retrieval, And Epistemic Foraging

Agentic science surveys describe systems that move from tool use toward scientific agency across perception, knowledge representation, planning, experimentation, analysis, and communication [Wei et al., 2025, Gridach et al., 2025]. GraphRAG work adds structured retrieval to that picture: graph construction and graph-aware retrieval can support multi-hop reasoning, but benchmarks also show that knowledge-graph RAG remains brittle when relevant knowledge is incomplete [Xiao et al., 2025, Zhou et al., 2026]. Active-inference perspectives make a similar design demand in different language: scientific agents need persistent uncertainty-aware memory, causal models, counterfactual exploration, deterministic validation, and human judgment as an architectural component [Duraisamy, 2025].
This exemplar uses those ideas as constraints, not as capabilities it already possesses. It does not run live literature mining, autonomous proof search, external agent swarms, graph-based hypothesis hunting, or self-approval. Instead it exposes bounded candidates, explicit budgets, local evidence links, local `MNIST` input data, benchmark-style scoring, source-linked figures, manuscript hydration, and human review gates. That is the intended safe baseline for a public template.

2.5.7 Benchmark, Documentation, And Process Analogies

`MNIST` and LeNet remain useful here because they provide a compact historical benchmark for small neural networks and handwriting recognition [LeCun et al., 1998, LeCun et al.]. Vision Transformers introduce the patch-token pattern for image classification at scale [Dosovitskiy et al., 2020]; this exemplar borrows only the patching and attention representation through `Tiny patch-attention classifier`, then keeps the implementation inside the configured candidate budget. MLPerf Tiny and OpenML motivate explicit task descriptions, fixed inputs, machine-readable run metadata, and checkable metrics [Banbury et al., 2021, Vanschoren et al., 2014]. Machine-learning reproducibility checklists motivate reporting data, seeds, model sizes, hyperparameters, and compute boundaries [Pineau et al., 2020].
Dataset and model documentation work further informs the safe boundary. Datasheets for Datasets motivate explicit reporting of dataset motivation, composition, collection, and recommended use [Gebru et al., 2021]. Model Cards motivate structured reporting of model context, intended use, evaluation procedure, and limitations [Mitchell et al., 2019]. The diagnostic layer follows the same

conservative reporting posture: calibration is treated as separate from accuracy [Guo et al., 2017], binomial accuracy intervals use Wilson-style score intervals [Wilson, 1927], matched classifier comparison is summarized through paired discordance [Dietterich, 1998], deterministic bootstrap intervals are local resampling diagnostics [Efron and Tibshirani, 1993], and probability quality is reported with Brier score, negative log likelihood, and chance-corrected agreement [Brier, 1950, Cohen, 1960].

The process language is borrowed cautiously from teleology and theoretical biology. Kant's account of organized beings treats a natural purpose as a whole whose parts and whole mutually condition one another [Ginsborg, 2022]. Autopoiesis characterizes living systems through self-producing organization [Varela et al., 1974], and later work connects Kantian natural purpose to autopoietic individuality [Weber and Varela, 2002]. Moreno and Mossio's account of biological autonomy emphasizes organizational closure and self-maintenance [Moreno and Mossio, 2015]. This paper uses those ideas only as disciplined analogies for configured scientific workflows whose artifacts help reproduce, constrain, and evaluate subsequent artifacts.

Security and supply-chain references enter with the same restraint. NIST's zero-trust architecture treats verification as explicit and continuous rather than inherited from a trusted perimeter [Rose et al., 2020]. The NIST Secure Software Development Framework emphasizes repeatable practices for reducing software vulnerability risk [Souppaya et al., 2022]. SLSA frames software-artifact provenance and supply-chain integrity as a graded assurance problem [Open Source Security Foundation, 2026], while MITRE ATT&CK T1195 names supply-chain compromise as a concrete adversary technique [MITRE ATT&CK, 2026]. This exemplar borrows those frameworks as disciplined analogies for local research artifact integrity: checksums, inventories, review gates, and explicit non-claims, not production deployment certification.

3 Methodology

The loop is implemented through the project source surface summarized by the validated run artifacts. The project scripts remain thin dispatchers; reusable behavior writes `output/data/autoresearch_loop.json`, `output/data/ml_task_results.json`, `../figures/figure_registry.json`, `output/data/autoresearch_phase_ledger.json`, and `output/data/manuscript_variable_provenance.json`.

3.1 Task And Data

The task is `small MNIST neural-network classification`. The default configuration loads `data/mnist_small.npz`, with provenance in `data/mnist_small_provenance.json`, and uses seed `20260525`. The subset contains `2000` training images and `500` test images, with `10` classes present in both splits and image shape `28 by 28`. The provenance artifact records upstream source-file hashes, the subset seed, class counts, and the compressed subset hash. The default pipeline never downloads data at runtime. The train and test splits contain `200` and `50` examples per class, respectively. The registered class-balance diagnostic is fig. 3, and the dataset contact sheet is fig. 4.

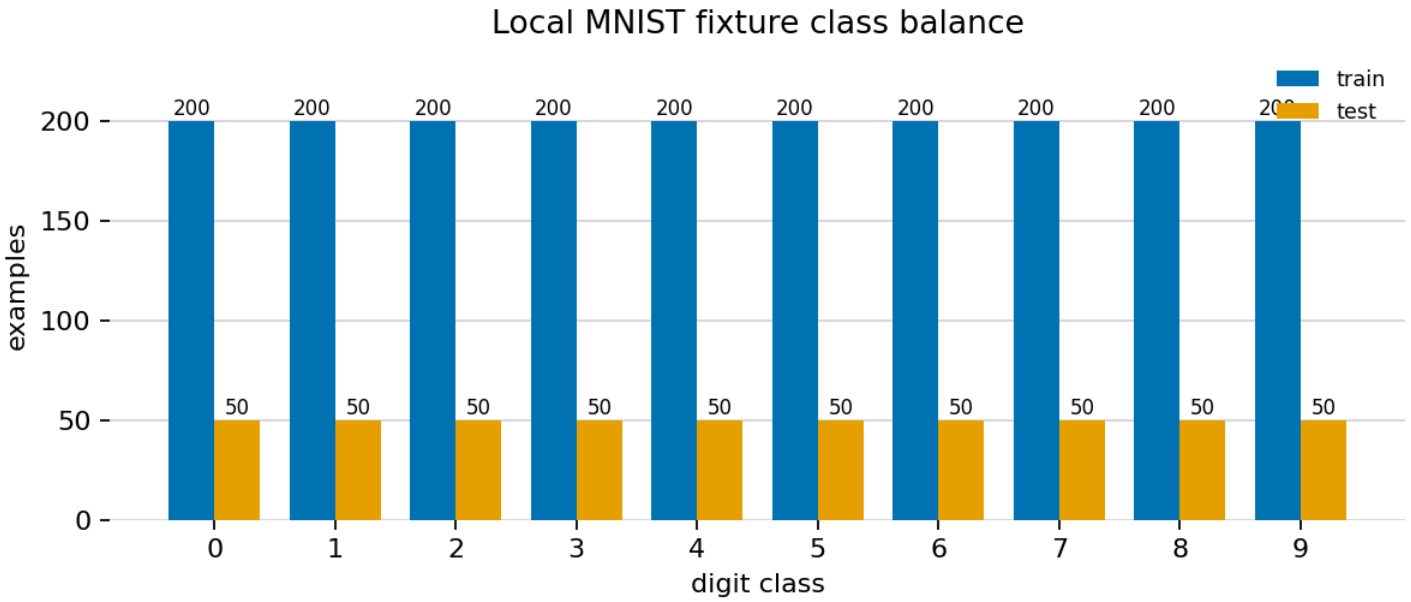

Figure 3: Train and test class counts from output/data/ml_class_balance.json; the local fixture contains 2000 train and 500 test examples across 10 classes. Generation method: Grouped train/test class-count bars from the local MNIST fixture. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 1: Local fixture class-balance table from `output/data/ml_class_balance.json`; counts describe the offline fixture used by this run.

Split	Class	Count	Fraction
train	0	200	10.0%
train	1	200	10.0%
train	2	200	10.0%
train	3	200	10.0%
train	4	200	10.0%
train	5	200	10.0%
train	6	200	10.0%
train	7	200	10.0%
train	8	200	10.0%
train	9	200	10.0%
test	0	50	10.0%
test	1	50	10.0%
test	2	50	10.0%
test	3	50	10.0%
test	4	50	10.0%
test	5	50	10.0%

Split	Class	Count	Fraction
test	6	50	10.0%
test	7	50	10.0%
test	8	50	10.0%
test	9	50	10.0%

The baseline is `nearest_centroid_baseline` (`nearest-centroid`). The bounded candidate set is configured by the task artifact and covers `MLP, nearest-centroid, softmax regression, tiny patch-attention`, including `Tiny patch-attention classifier` and at least one deferred proposal. The transformer-style candidate borrows the patch-token representation for comparison while staying inside the configured iteration budget. This is intentionally a small configured model, not a claim about full-scale image-transformer performance.

3.2 Bounded Loop

The run follows `7` configured stages:

- Resolve the human-authored program, project topic, and research questions.
- Build an `AutoResearchPlan` from the domain profile, experiment plan, and pipeline DAG.
- Validate exact stage-gate names declared in `autoresearch.yaml`.
- Evaluate the configured `MNIST` candidate set up to the configured iteration budget.
- Generate claims only from configured questions and local artifact paths.
- Write data, reports, figures, benchmark scores, and review packets under the declared output contract.
- Run strict AutoResearch readiness validation and write readiness reports.

Candidates are declared in the proposal ledger and resolved from the task configuration for execution. Each executable candidate declares a model type, seed, training schedule, and model-specific parameters. The configured training policy includes learning-rate decay `0.995` and gradient clipping norm `5`, both recorded from the validated task run rather than described as a free-form manuscript setting. The loop evaluates at most `4` configured iterations, selects the highest `test_accuracy`, and breaks ties by lower parameter count and identifier. Candidates outside the budget are recorded as deferred; the run-derived status summary is `accepted: 1, deferred: 1, rejected: 3`.

The closure is concrete and file-backed. `output/data/research_program.json` constrains proposals; proposal records feed the bounded evaluation; evaluation writes `output/data/ml_candidate_ledger.json` and `output/data/run_ledger.json`; ledgers support artifact-linked claims; claims hydrate the manuscript through `output/data/manuscript_variables.json`; readiness validation writes `output/reports/autoresearch_readiness.json`; loop settlement is recorded in `output/data/autoresearch_phase_ledger.json`; and review state is captured in `output/data/review_decisions.json`. The loop therefore maintains an inspectable research process around a small metric result without making the process self-approving or opaque.

The concrete closure is intentionally close to research-object and workflow-run provenance practice. The artifact manifest lists generated objects and checksums; the figure registry binds captions to source artifacts; the variable provenance sidecar records the source pointer for each injected token or table; and the review packet keeps human decisions outside generated approval. This is a minimal local analogue of machine-readable provenance rather than a claim of full research-object packaging.

The generated schema manifest at `output/data/autoresearch_schema_manifest.json` records `31` schema-versioned governance payload(s) plus documented generic-table exemptions. The local research-object manifest at `output/data/research_object_manifest.json` packages observed project paths, hashes, the evidence registry, the source ledger, the schema manifest, and the manual approval state. It is deliberately named a local research-object manifest, not RO-Crate or SLSA compliance.

3.3 Safety Controls

The default autonomy level is `proposal_only`. The run ledger records `0` LLM calls and USD `0.00` cost. The edit allowlist is restricted to the public project source and manuscript surfaces, plus the task configuration file, and the pipeline never executes generated code. Review gates are emitted with deferred decisions so that validation can confirm the gates exist without pretending that the machine approved publication.

Publication approval is a non-generated input. The run may read `human_review.yaml` and copy its state into review payloads, but generated readiness cannot set publication approval by itself; the default local review state remains `false`.

Disclosure: `AI-assisted AutoResearch` status is declared for this exemplar because it models machine-produced plans, ledgers, reports, and manuscript variables as review inputs rather than autonomous approval.

3.4 Adversarial And Supply-Chain Controls

The security layer is configured as `local_deterministic` with network policy `default_offline`, integrity algorithm `sha256`, external signing `false`, and framework labels `STRIDE, MITRE_ATT&CK_T1195`. Its scope is `Local research-artifact integrity evidence for this deterministic public exemplar`. These values are generated from `output/data/autoresearch_security_profile.json`; the default run remains offline and deterministic.

The local threat model covers `7` asset(s), `7` threat row(s), and `7` control row(s). The security artifacts are written to `output/data/autoresearch_threat_model.json`, `output/data/autoresearch_supply_chain_inventory.json`, `output/data/autoresearch_i

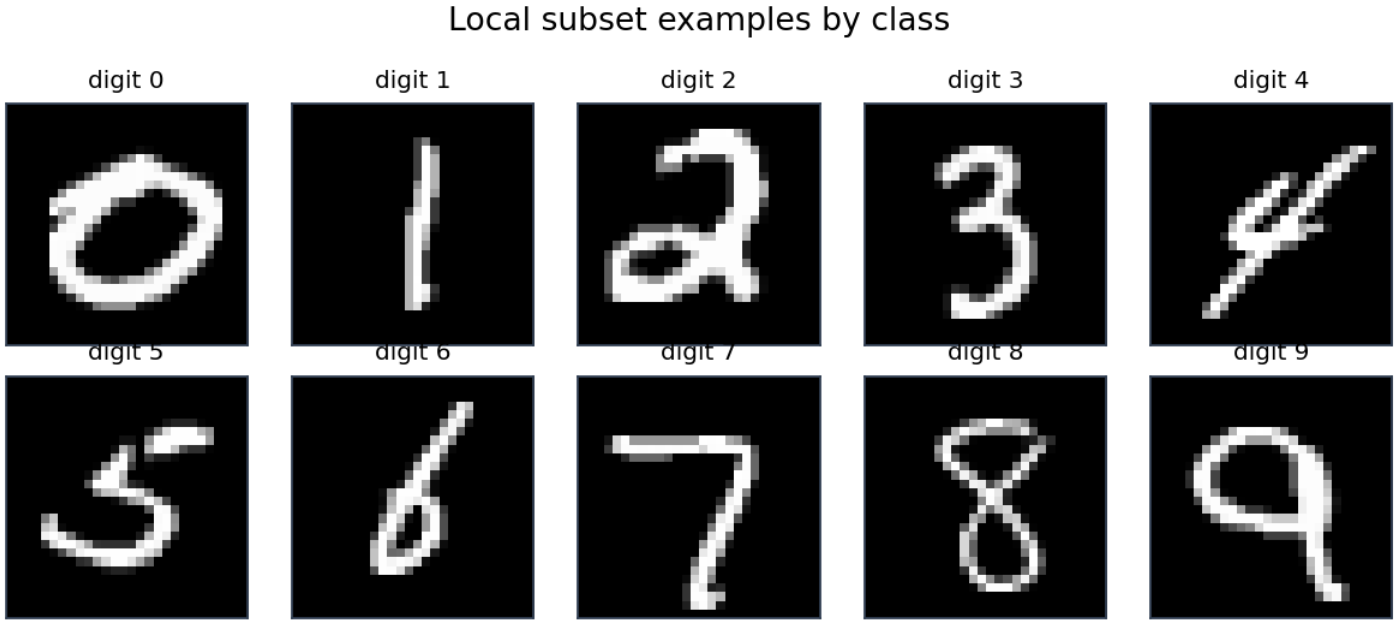

Figure 4: Deterministic class-balanced contact sheet for MNIST handwritten digit database from data/mnist_small.npz and data/mnist_small_provenance.json; the figure documents the local subset used by the offline run. Generation method: Class-balanced contact sheet from fixed local MNIST arrays. Registry metadata records the generation method, source artifact, and claim boundary for validation.

`ntegrity_attestation.json`, and `output/reports/autoresearch_security_review.md`. The control-matrix figure is fig. 5. The table and figure are generated from the threat model and inventory, not manually maintained.

Table 2: Local security artifacts generated for the bounded AutoResearch run.

Security artifact	Path	Summary
profile	security profile	local_deterministic
threat model	threat model	default_offline
inventory	supply inventory	14 inputs
inventory export	inventory export	local non-SBOM export
attestation	integrity attestation	passed
review	security review	human review input

Table 3: Threat-model rows from `output/data/autoresearch_threat_model.json`; ATT&CK labels scope supply-chain compromise analogies.

Threat	STRIDE	ATT&CK	Scenario	Residual risk
dataset tamper	Tampering	T1195	A local fixture could be replaced or edited before analysis.	Residual risk remains if a reviewer ignores checksum drift.
config drift	Tampering	T1195.001	Task settings could silently change candidate scope, budgets, or diag…	Configuration review is still a human responsibility.
source edit	Elevation of privilege	T1195.001	Source changes could bypass the thin-script and no-generated-code bou…	The default run does not perform full static application security tes…
output tamper	Repudiation	T1195.002	Generated reports or figures could be edited after analysis but befor…	Local checksum evidence is not externally signed.
manuscript injection	Information disclosure	T1195.002	Manual prose could hard-code run facts that bypass validated variables.	Stable scholarly prose and citekeys remain manually authored.
self approval	Spoofing	T1195	Generated review packets could be mistaken for human publication appr…	Publication remains outside the automated loop.
build assumption	Denial of service	T1195.003	A local or CI build context could omit checks or run stale generated…	The exemplar does not sign build logs or isolate runners.

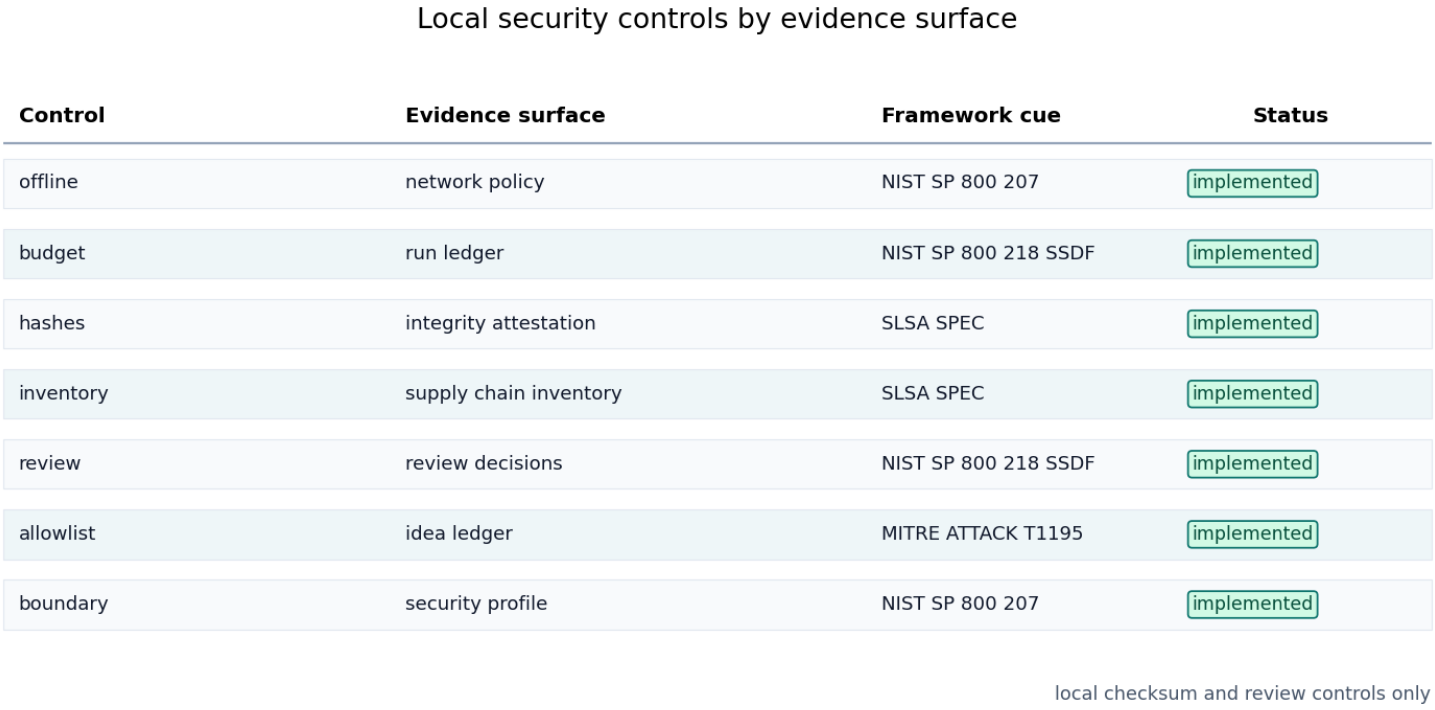

Figure 5: Local security-control matrix from output/data/autoresearch_threat_model.json; controls map NIST, SLSA, and ATT&CK-inspired safeguards to deterministic evidence surfaces without claiming production security certification. Generation method: structured control matrix with separate control, evidence, framework, and status columns. Generation method: Structured control matrix from local threat-model controls. Registry metadata records the generation method, source artifact, and claim boundary for validation.

3.5 Positioning Against Autonomous Science Systems

The implementation should be read as a bounded local analogue of current AutoResearch trends, not as a miniature autonomous scientist. Artifact manifests, evidence registries, review gates, and manuscript hydration provide machine-readable governance surfaces for one offline project run. They do not perform autonomous literature mining, proof search, live agent orchestration, runtime dataset expansion, self-modifying code search, or self-approval of publication claims.

3.6 Evidence And Scoring

The experiment writes `output/data/ml_task_results.json`, `output/data/ml_candidate_ledger.json`, `output/data/ml_confusion_matrix.csv`, `output/data/ml_training_history.csv`, `output/data/ml_error_examples.json`, `output/data/ml_prediction_records.json`, `output/data/ml_classification_diagnostics.json`, `output/data/ml_candidate_intervals.json`, `output/data/ml_class_balance.json`, `output/data/ml_calibration_report.json`, `output/data/ml_calibration_bin_intervals.json`, `output/data/ml_robustness_report.json`, `output/data/ml_probability_diagnostics.json`, `output/data/ml_bootstrap_intervals.json`, `output/data/ml_paired_comparison.json`, `output/data/ml_candidate_rank_stability.json`, `output/data/ml_statistical_summary.json`, `output/data/ml_training_diagnostics.json`, `output/reports/ml_benchmark_score.json`, `output/data/benchmark_boundary.json`, `output/data/figure_quality_report.json`, and registered figures through `../figures/figure_registry.json`. The diagnostic payloads preserve probabilities, confidence and margin summaries, class metrics, calibration bins, confusion-pair summaries, train/test gaps, deterministic no-retrain perturbation scores, bootstrap intervals, paired baseline comparison, rank-stability frequencies, calibration-bin Wilson intervals, selective accuracy thresholds, Brier score, negative log likelihood, top-2 accuracy, probability-quality comparisons, learning-rate traces, best-epoch markers, final learning rates, and train-test gap summaries without serializing model weights. The benchmark score combines metric improvement, budget compliance, offline execution, transformer-candidate coverage, and candidate-selection status. The benchmark-boundary artifact records fixture scope, metric direction, candidate families, budget, statistical-method artifacts, and explicit non-claims, so benchmark-adjacent statistics remain local readiness diagnostics rather than broad empirical or publication claims. Manuscript variables are hydrated from these artifacts, and readiness validation checks `output/reports/evidence_registry.json`, `output/reports/artifact_manifest.json`, method ledgers, review gates, benchmark outputs, the phase ledger, figure-quality report, and AI-assisted disclosure before rendering is treated as ready for review. The reviewer-facing `output/data/autoresearch_evidence_overview.json` then summarizes readiness versus publication approval, claim-evidence rows, source-ledger status, benchmark boundary issues, and security/integrity status without granting approval. The generated tables and figure blocks used in sec. 4 are sourced through `output/data/manuscript_variable_provenance.json` and `output/data/manuscript_figure_blocks.json`, so captions, artifact tables, and run-derived result statements share the same validated artifact base.

The figure registry also carries the method contract for each visualization: source artifact, generated file, rendering method, and claim boundary. The rendered method table below is generated from that registry rather than maintained manually.

Table 4: Registry-backed figure methods from `figure_registry.json`; full validation hooks, alt text, and claim boundaries remain in the registry.

Figure	Source	Method	Scope
fig. 14	candidate ledger	Candidate lifecycle status-count bar chart.	Lifecycle counts describe bounded orchestration, not autonomous approval.
fig. 26	loop	File-backed process-flow diagram from final loop state.	The workflow is file-backed and inspectable but not self-approving.
fig. 27	integrity attestation	Local checksum attestation chain with checked, missing, and mismatch counts.	Integrity checks are local SHA-256 observations and are not externally signed provenance.
fig. 5	threat model	Structured control matrix from local threat-model controls.	Controls are local research-artifact safeguards, not production security certification.
fig. 25	loop	Horizontal count summary from final loop metrics.	Readiness artifacts summarize local validation only; publication approval is human.
fig. 21	bootstrap intervals	Horizontal percentile-bootstrap interval plot.	Bootstrap intervals summarize local test-set resampling only.
fig. 15	calibration report	Reliability curve with confidence-bin support histogram.	Calibration values describe the fixed local split only.
fig. 7	candidate rank stability	Bootstrap top-rank frequency and mean-rank comparison.	Rank stability describes local resampling behavior, not model-selection certainty.
fig. 6	candidate intervals	Lollipop accuracy comparison with Wilson interval error bars and direct labels.	Scores apply only to the fixed local subset and configured candidates.
fig. 16	class diagnostics	Per-class precision, recall, and F1 heatmap.	Class metrics diagnose this run only and are not full-dataset estimates.
fig. 12	task results	Log-parameter scatter plot against held-out accuracy.	The plot compares this bounded candidate set and does not infer a scaling law.
fig. 8	confusion matrix	Row-normalized heatmap with cell counts and row percentages.	Confusion counts diagnose this run only and do not imply full-dataset generalization.
fig. 17	class diagnostics	Ranked off-diagonal confusion-pair bars with true-class error rates.	Pair counts identify local error cases and are not a general taxonomy.
fig. 18	class diagnostics	Grouped train/test accuracy and loss bars by evaluated candidate.	Gaps are local bounded-loop diagnostics, not convergence guarantees.
fig. 10	training history	Epoch-level held-out accuracy lines with accepted best-epoch marker.	Learning curves diagnose configured training only, not convergence in general.
fig. 22	paired comparison	Matched accepted-versus-baseline correctness heatmap.	Matched comparison is limited to the fixed local test split.
fig. 9	confusion matrix	Per-class accuracy bars computed from the confusion matrix diagonal.	Per-class values diagnose this local split only and do not certify robustness.
fig. 20	probability diagnostics	Confidence and margin histograms split by correctness.	Distributions are descriptive diagnostics for the fixed local test split.
fig. 24	statistical summary	Brier score and negative-log-likelihood bar comparison.	Probability-quality metrics compare the configured evaluated candidates only.
fig. 19	robustness report	Candidate-by-transform accuracy heatmap for deterministic perturbations.	Deterministic perturbations are a smoke test and do not certify robustness.
fig. 23	statistical summary	Confidence-threshold coverage and selective-accuracy line chart.	Selective accuracy describes thresholded predictions on the fixed local split only.
fig. 11	training diagnostics	Final and best-epoch accuracy bars plus train-test gap bars.	Training dynamics diagnose this configured deterministic run only.
fig. 3	class balance	Grouped train/test class-count bars from the local MNIST fixture.	Class counts describe the local fixed fixture and are not population statistics.
fig. 13	error examples	Deterministic grid of the first accepted-candidate misclassifications.	Examples are qualitative diagnostics for this run, not an error taxonomy.
fig. 4	small	Class-balanced contact sheet from fixed local MNIST arrays.	The sheet illustrates the local fixed subset and is not a statistical sample claim.

4 Results

4.1 Candidate Outcome

The generated loop selected `exp-mlp-tanh-64` (`One-hidden-layer tanh MLP`, `MLP`) after evaluating `4` candidate(s) from a proposed set of `5`. The `nearest_centroid_baseline` (`nearest-centroid`) baseline reached `82.6%` `test_accuracy`, while the selected candidate reached `89.4%`, an absolute change of `6.8%`. The selected model has `50890` parameters. The transformer-candidate evaluated flag is `true`, and the candidate budget exhausted flag is `true`, which means the ledger records `1` deferred proposal(s) rather than expanding the run automatically.
The benchmark score is `1`. That score is not a model-quality claim by itself; it is a compact grading artifact for the methods contract: metric improvement, budget compliance, offline execution, and selected-candidate recording. Rank-stability diagnostics report that the selected candidate is top ranked in `72.5%` of deterministic bootstrap resamples, with runner-up `exp-mlp-relu-32`. The candidate score figure is registered as fig. 6, the confusion matrix as fig. 8, the per-class diagnostic as fig. 9, the learning curves as fig. 10, the complexity diagnostic as fig. 12, the selected-candidate error examples as fig. 13, the candidate lifecycle diagnostic as fig. 14, rank stability as fig. 7, the training-dynamics diagnostic as fig. 11, the final readiness matrix as fig. 25, and the process closure as fig. 26.

Table 5: Candidate accuracy intervals from `output/data/ml_candidate_intervals.json`; intervals describe the fixed local test split.

Candidate	Status	Correct/test	Accuracy	Wilson 95% interval
baseline	baseline	413/500	82.6%	79.0% to 85.7%
softmax linear	rejected	441/500	88.2%	85.1% to 90.7%
mlp relu 32	rejected	443/500	88.6%	85.5% to 91.1%
tiny patch attention	rejected	152/500	30.4%	26.5% to 34.6%
mlp tanh 64	accepted	447/500	89.4%	86.4% to 91.8%

Table 6: Candidate rank-stability table from `output/data/ml_candidate_rank_stability.json`; frequencies are deterministic local bootstrap summaries.

Candidate	Observed rank	Top-rank frequency	Mean rank	Accuracy
softmax linear	3	4.2%	2.554	88.2%
mlp relu 32	2	23.3%	2.130	88.6%
tiny patch attention	4	0.0%	4.000	30.4%
mlp tanh 64	1	72.5%	1.316	89.4%

4.2 Run-Derived Figures

4.3 Training And Error Diagnostics

The selected candidate's best held-out epoch is `14`, its final learning rate is `0.144`, its train-loss reduction is `4.161`, and its final train-test accuracy gap is `7.0%`. These values come from the configured training diagnostics rather than from a manually maintained result summary.

Table 7: Configured-training diagnostics from `output/data/ml_training_diagnostics.json`.

Candidate	Status	Best epoch	Best test accuracy	Final test accuracy	Train-test gap	Loss reduction	Final learning rate
softmax linear	rejected	17	89.4%	88.2%	5.9%	2.917	0.273
mlp relu 32	rejected	21	90.0%	88.6%	8.5%	4.341	0.178
tiny patch attention	rejected	24	30.4%	30.4%	-1.8%	0.165	0.214
mlp tanh 64	accepted	14	90.0%	89.4%	7.0%	4.161	0.144

4.4 Diagnostic Analysis

The selected candidate macro F1 is `89.4%`, with a held-out accuracy interval of `86.4% to 91.8%`. Probability diagnostics report expected calibration error `2.9%` and `11` high-confidence error(s). The top non-diagonal confusion pair is `4 -> 9 (4)`, and the minimum selected-candidate accuracy across deterministic robustness transforms is `80.0%`. These values are descriptive diagnostics for the validated local run, not external benchmark claims.

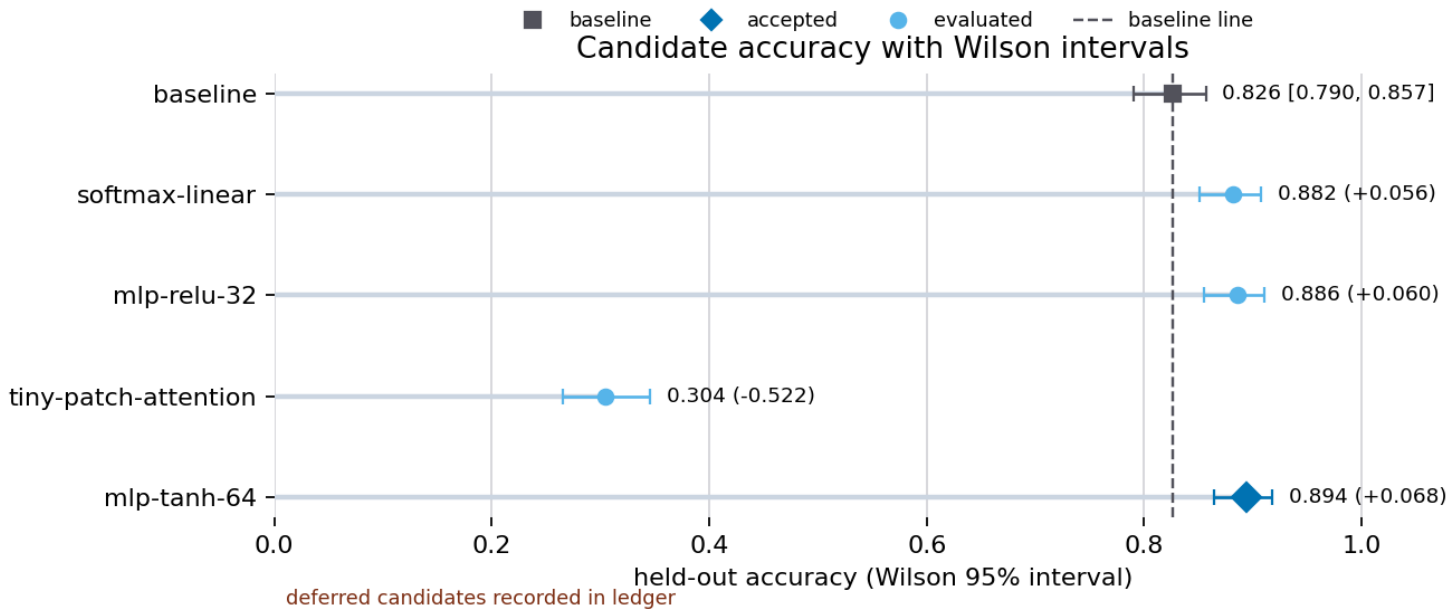

Figure 6: Held-out accuracy with Wilson intervals for the baseline and evaluated candidates from output/data/ml_candidate_intervals.json; accepted candidate exp-mlp-tanh-64 improves accuracy from 82.6% to 89.4% on the fixed subset, with deferred proposals kept in the candidate ledger. Generation method: Lollipop accuracy comparison with Wilson interval error bars and direct labels. Registry metadata records the generation method, source artifact, and claim boundary for validation.

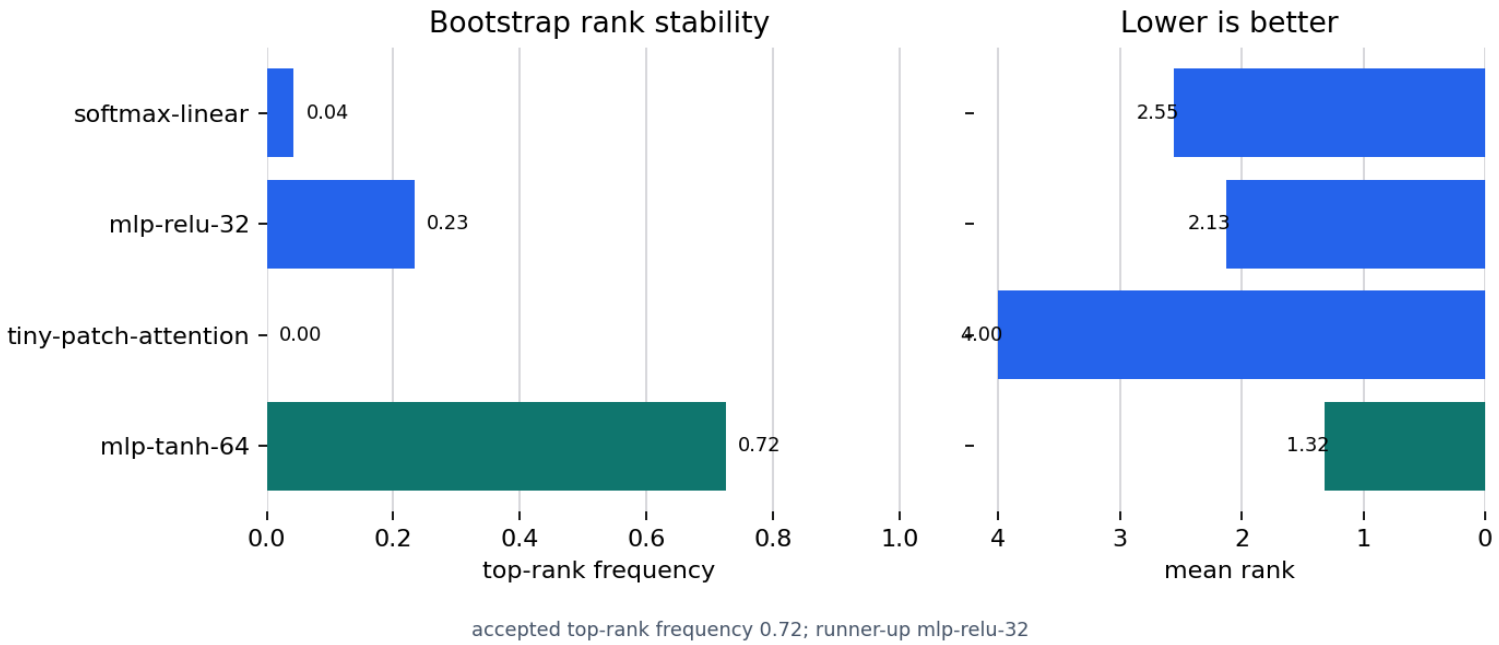

Figure 7: Rank-stability summary for exp-mlp-tanh-64 from output/data/ml_candidate_rank_stability.json; deterministic local bootstrap resampling shows how often each evaluated candidate ranks first under the fixed test split. Generation method: Bootstrap top-rank frequency and mean-rank comparison. Registry metadata records the generation method, source artifact, and claim boundary for validation.

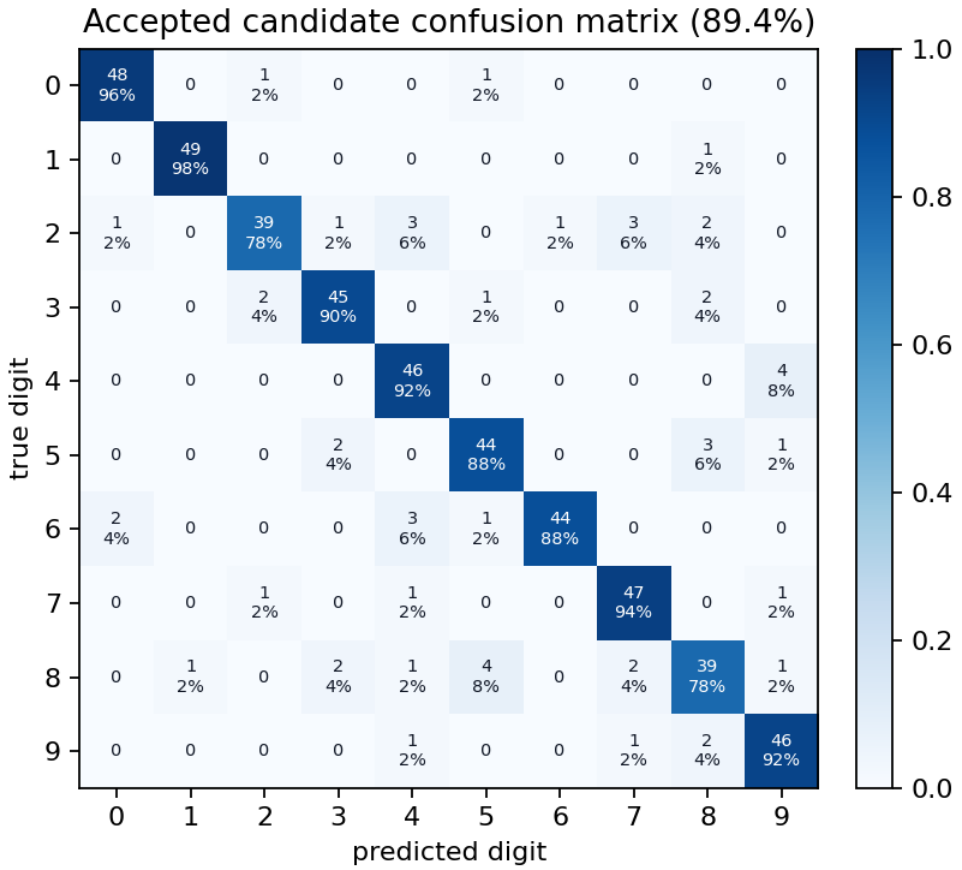

Figure 8: Accepted-candidate confusion matrix for exp-mlp-tanh-64 on the fixed MNIST handwritten digit database test split, sourced from output/data/ml_confusion_matrix.csv; it diagnoses the selected run, not general full-dataset performance. Generation method: Row-normalized heatmap with cell counts and row percentages. Registry metadata records the generation method, source artifact, and claim boundary for validation.

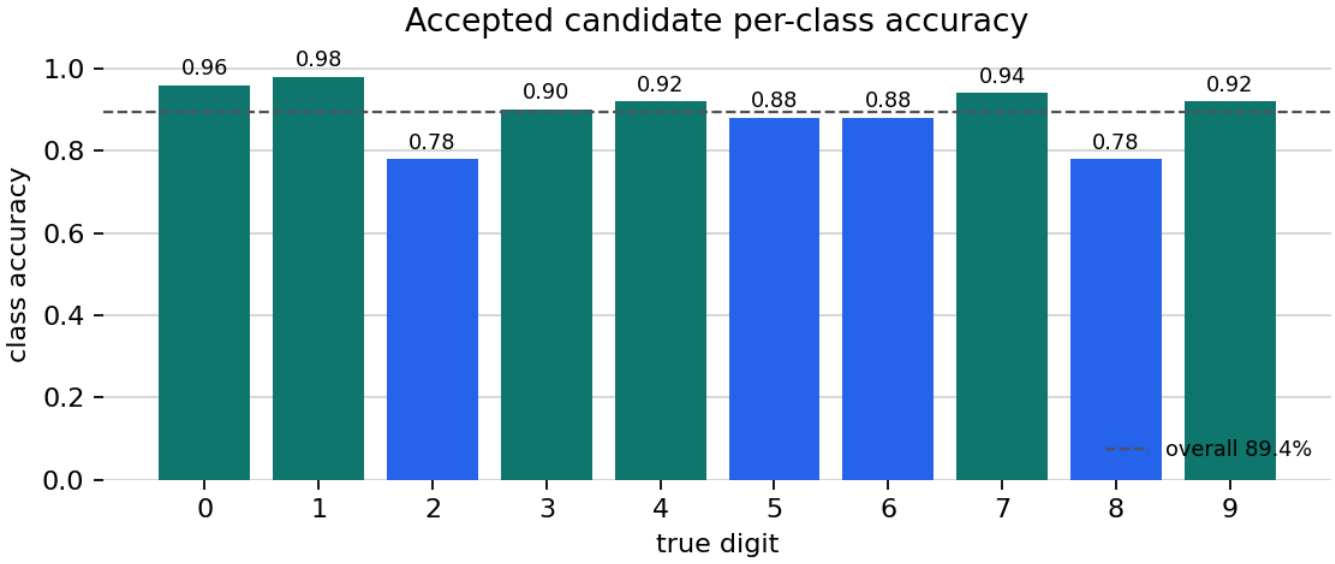

Figure 9: Per-class accuracy for exp-mlp-tanh-64, computed from output/data/ml_confusion_matrix.csv; variation across digits is a run diagnostic for the fixed local test split. Generation method: Per-class accuracy bars computed from the confusion matrix diagonal. Registry metadata records the generation method, source artifact, and claim boundary for validation.

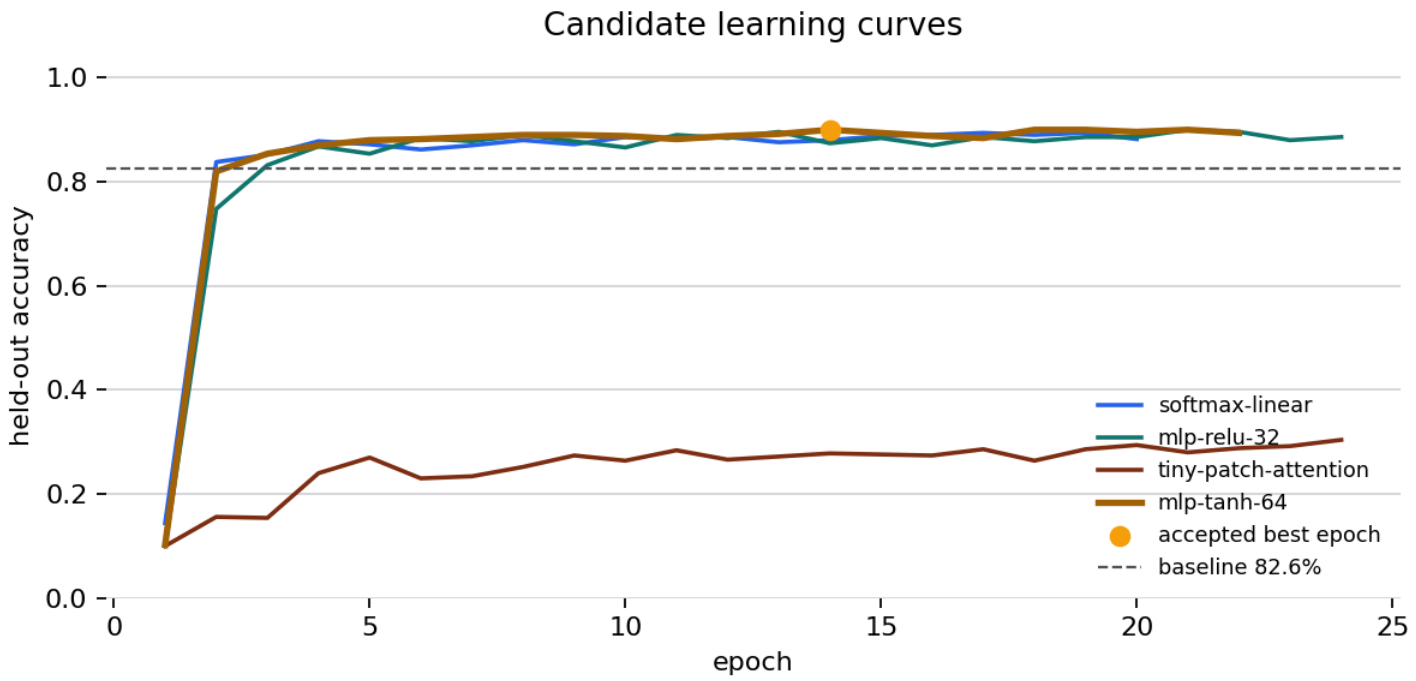

Figure 10: Epoch-level held-out accuracy curves for evaluated candidates from output/data/ml_training_history.csv; the accepted curve is visually emphasized for exp-mlp-tanh-64. Generation method: Epoch-level held-out accuracy lines with accepted best-epoch marker. Registry metadata records the generation method, source artifact, and claim boundary for validation.

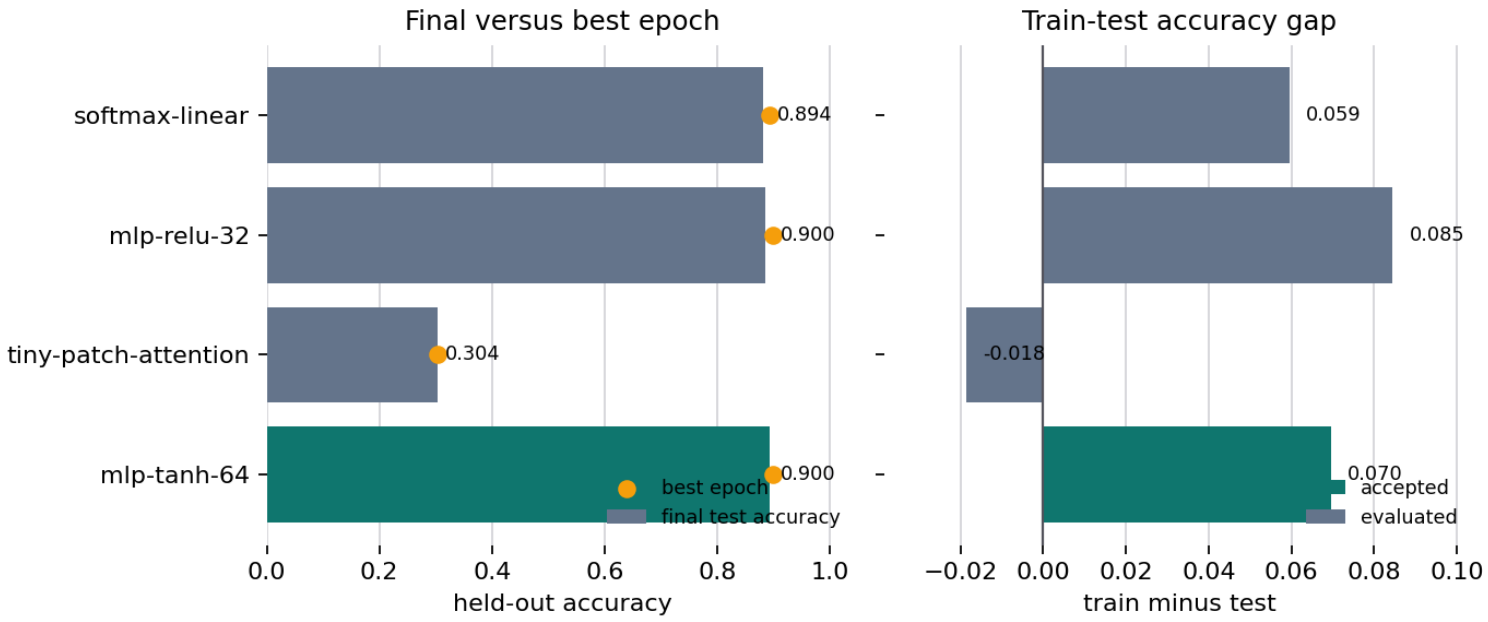

Figure 11: Configured-training dynamics for evaluated candidates from output/data/ml_training_diagnostics.json; exp-mlp-tanh-64 is highlighted while best-epoch markers and train-test gaps remain bounded to the local run. Generation method: Final and best-epoch accuracy bars plus train-test gap bars. Registry metadata records the generation method, source artifact, and claim boundary for validation.

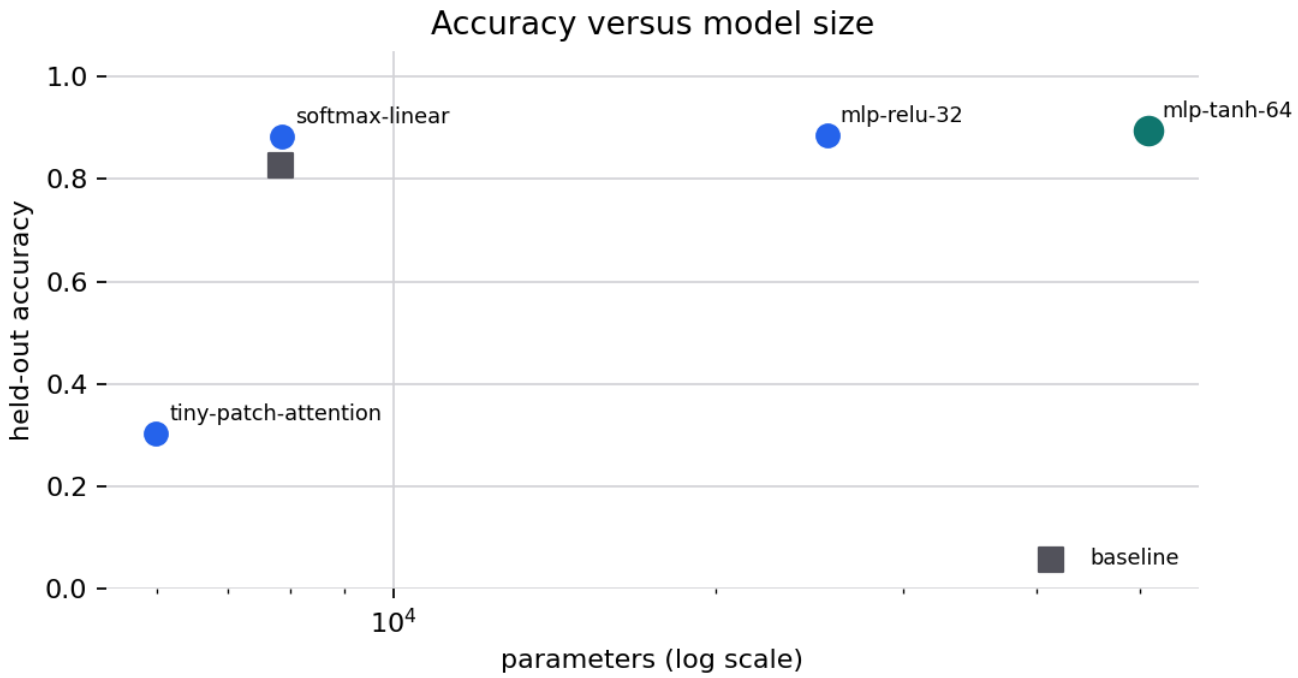

Figure 12: Parameter-count versus held-out accuracy for the baseline and evaluated candidates from output/data/ml_task_results.json; the accepted candidate is highlighted without claiming a general scaling law. Generation method: Log-parameter scatter plot against held-out accuracy. Registry metadata records the generation method, source artifact, and claim boundary for validation.

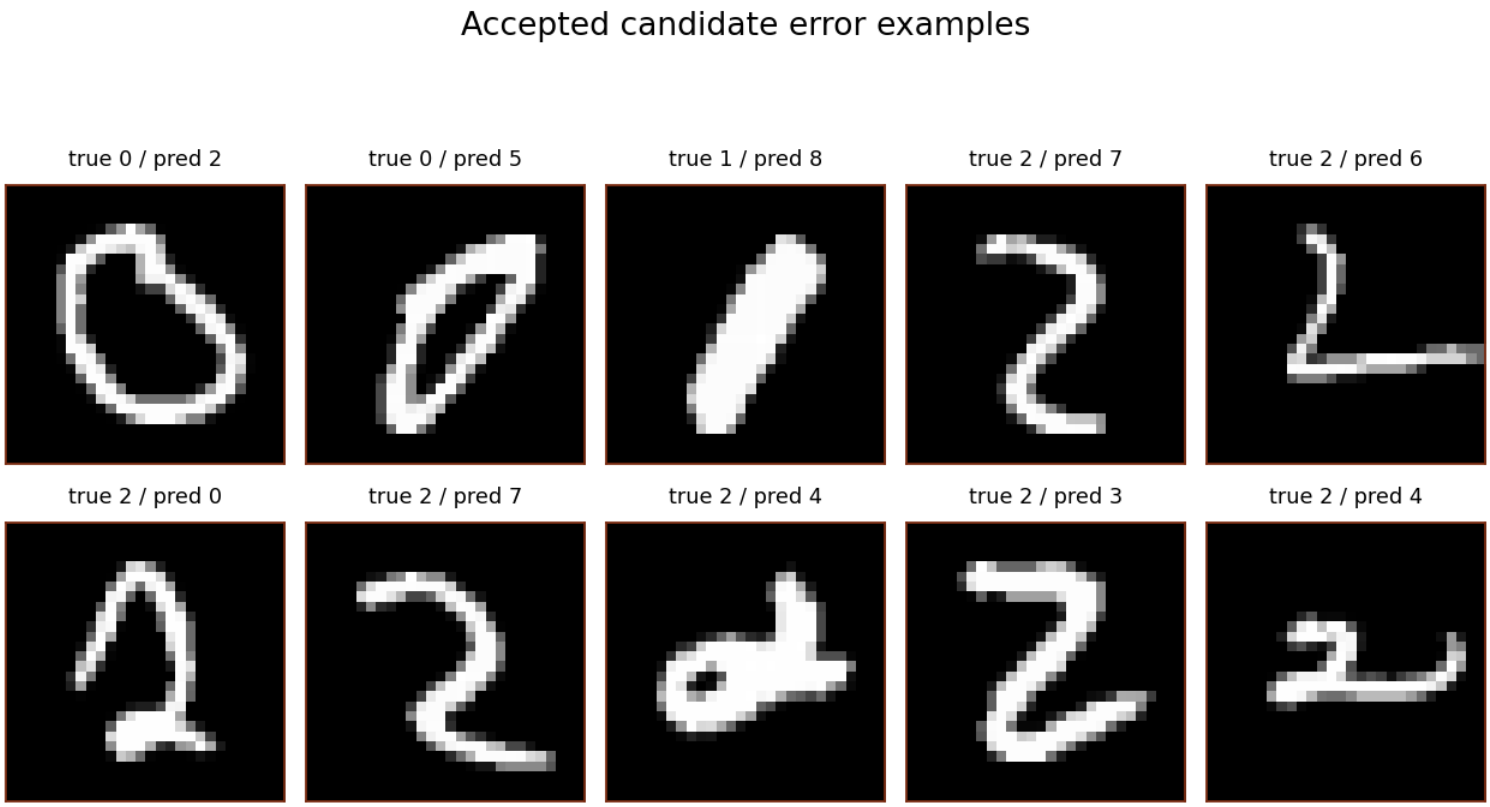

Figure 13: First accepted-candidate error examples for exp-mlp-tanh-64, sourced from output/data/ml_error_examples.json and data/mnist_small.npz; these images support qualitative diagnosis only. Generation method: Deterministic grid of the first accepted-candidate misclassifications. Registry metadata records the generation method, source artifact, and claim boundary for validation.

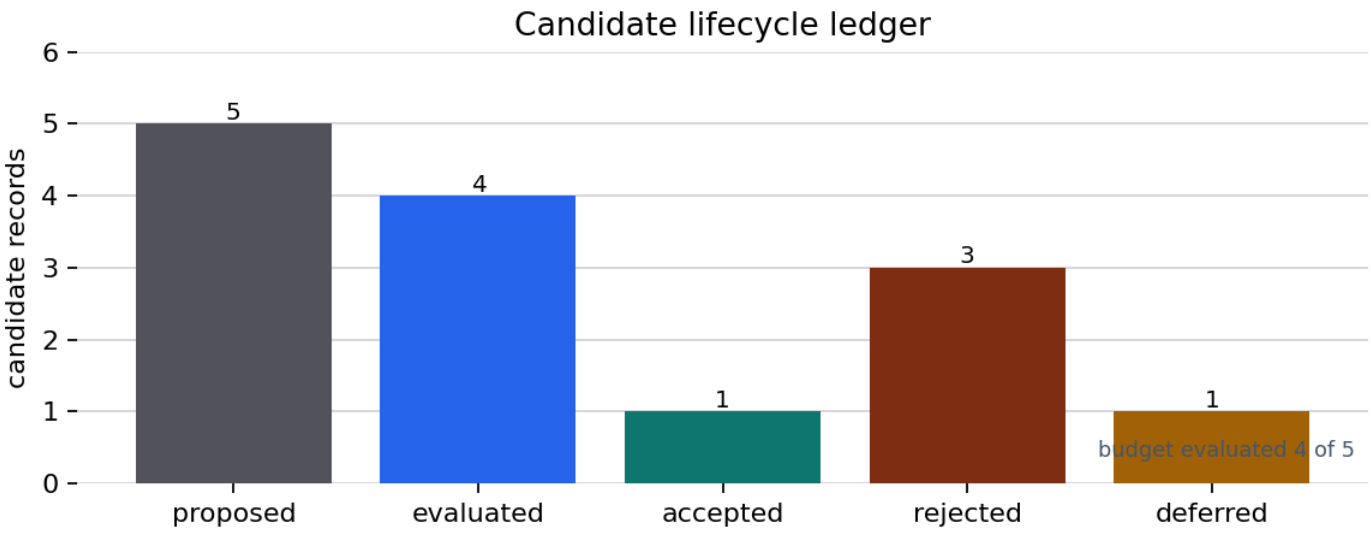

Figure 14: Candidate lifecycle ledger from output/data/ml_candidate_ledger.json: 4 evaluated out of 5 proposed candidates, with deferred proposals kept visible instead of executed automatically. Generation method: Candidate lifecycle status-count bar chart. Registry metadata records the generation method, source artifact, and claim boundary for validation.

4.4.1 Classification And Error Structure

The class-level and error-pattern diagnostics are intentionally visual-first: fig. 15 checks confidence calibration, fig. 16 separates precision, recall, and F1 by digit, and fig. 17 ranks the non-diagonal confusions that most shape the selected-candidate error profile. The accompanying tables preserve the same values for audit and downstream comparison.

Table 8: Accepted-candidate class diagnostics from `output/data/ml_classification_diagnostics.json`.

Class	Precision	Recall	F1	Support
0	94.1%	96.0%	95.0%	50
1	98.0%	98.0%	98.0%	50
2	90.7%	78.0%	83.9%	50
3	90.0%	90.0%	90.0%	50
4	83.6%	92.0%	87.6%	50
5	86.3%	88.0%	87.1%	50
6	97.8%	88.0%	92.6%	50
7	88.7%	94.0%	91.3%	50
8	79.6%	78.0%	78.8%	50
9	86.8%	92.0%	89.3%	50

Table 9: Calibration bins from `output/data/ml_calibration_report.json`.

Confidence bin	Count	Accuracy	Mean confidence	Gap
0-0.1	0	0.0%	0.0%	0.0%
0.1-0.2	0	0.0%	0.0%	0.0%
0.2-0.3	3	0.0%	27.2%	27.2%
0.3-0.4	4	75.0%	35.0%	40.0%
0.4-0.5	21	42.9%	45.7%	2.9%
0.5-0.6	28	67.9%	55.4%	12.5%
0.6-0.7	24	66.7%	65.0%	1.7%
0.7-0.8	28	67.9%	75.6%	7.7%
0.8-0.9	53	90.6%	85.8%	4.8%
0.9-1	339	98.2%	97.4%	0.8%

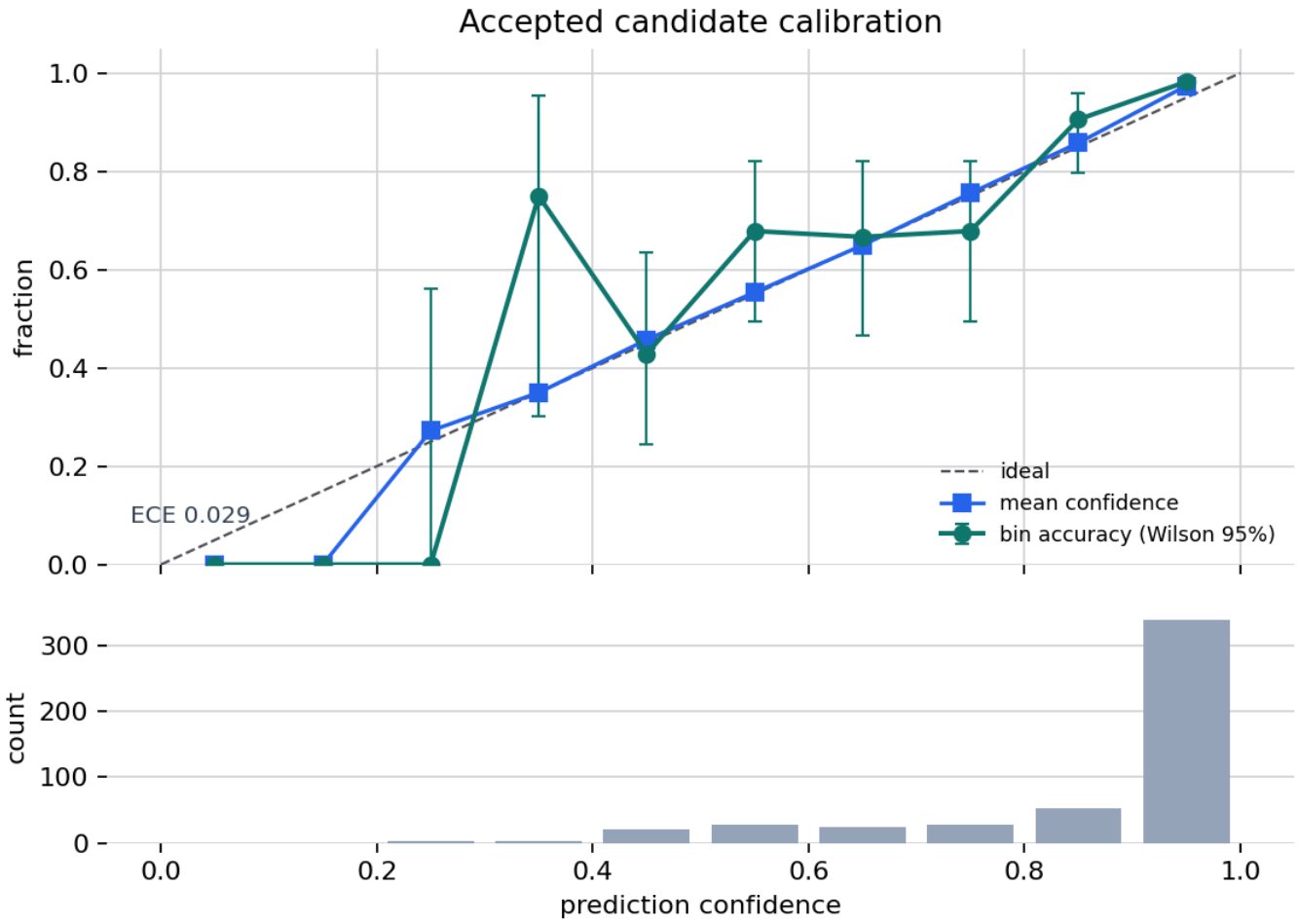

Figure 15: Reliability curve for exp-mlp-tanh-64 from output/data/ml_calibration_report.json; expected calibration error and bin counts summarize the accepted candidate on the fixed local test split. Generation method: Reliability curve with confidence-bin support histogram. Registry metadata records the generation method, source artifact, and claim boundary for validation.

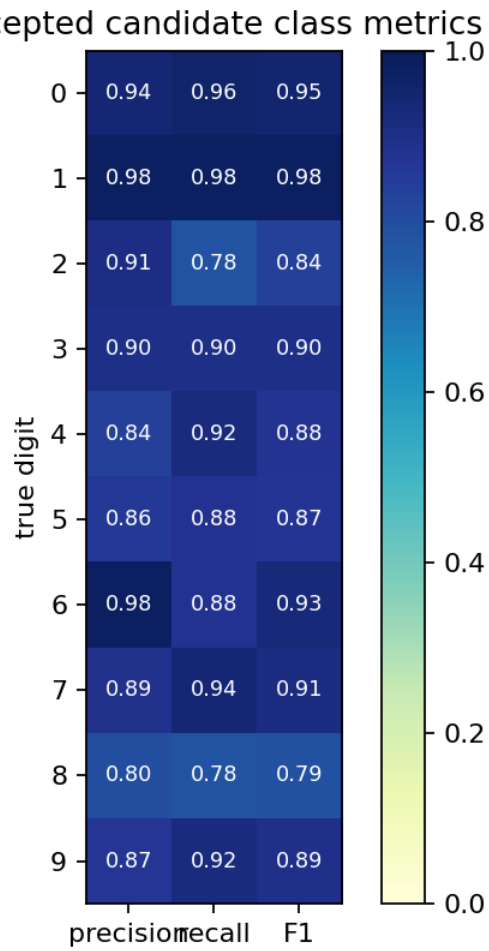

Figure 16: Per-class precision, recall, and F1 for exp-mlp-tanh-64, sourced from output/data/ml_classification_diagnostics.json; metrics are scoped to the local test split. Generation method: Per-class precision, recall, and F1 heatmap. Registry metadata records the generation method, source artifact, and claim boundary for validation.

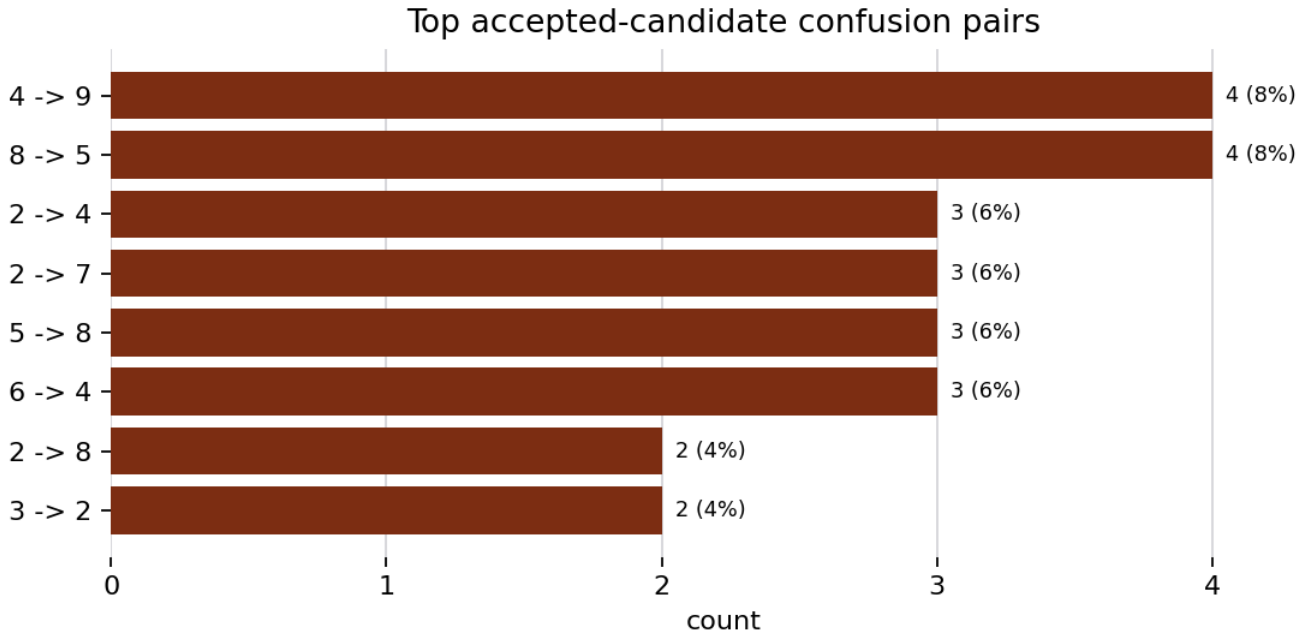

Figure 17: Top non-diagonal confusion pairs for exp-mlp-tanh-64, sourced from output/data/ml_classification_diagnostics.json; the bars highlight which local digit pairs account for accepted-candidate errors. Generation method: Ranked off-diagonal confusion-pair bars with true-class error rates. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 10: Calibration-bin Wilson intervals from `output/data/ml_calibration_bin_intervals.json`; empty bins are reported explicitly.

Confidence bin	Count	Correct	Accuracy	Wilson 95%	Empty
0-0.1	0	0	0.0%	0.0% to 0.0%	True
0.1-0.2	0	0	0.0%	0.0% to 0.0%	True
0.2-0.3	3	0	0.0%	0.0% to 56.2%	False
0.3-0.4	4	3	75.0%	30.1% to 95.4%	False
0.4-0.5	21	9	42.9%	24.5% to 63.5%	False
0.5-0.6	28	19	67.9%	49.3% to 82.1%	False
0.6-0.7	24	16	66.7%	46.7% to 82.0%	False
0.7-0.8	28	19	67.9%	49.3% to 82.1%	False
0.8-0.9	53	48	90.6%	79.7% to 95.9%	False
0.9-1	339	333	98.2%	96.2% to 99.2%	False

Table 11: Top non-diagonal confusion pairs from `output/data/ml_classification_diagnostics.json`.

Pair	Count	True-class error rate
4 -> 9	4	8.0%
8 -> 5	4	8.0%
2 -> 4	3	6.0%
2 -> 7	3	6.0%
5 -> 8	3	6.0%
6 -> 4	3	6.0%
2 -> 8	2	4.0%
3 -> 2	2	4.0%
3 -> 8	2	4.0%
5 -> 3	2	4.0%

4.4.2 Uncertainty, Generalization, And Perturbation

Additional uncertainty and matched-comparison diagnostics report bootstrap accuracy interval `86.4% to 92.0%`, bootstrap macro-F1 interval `86.5% to 91.9%`, paired net accuracy gain `6.8%`, exact McNemar p-value `0.000`, mean correct-prediction confidence `90.9%`, mean error confidence `63.1%`, and `22` low-margin selected-candidate prediction(s). These diagnostics are generated from the fixed local test split and are reported as run evidence, not as population-level certification.
fig. 18 compares train and test behavior, fig. 19 shows deterministic no-retrain perturbation results, fig. 20 summarizes confidence and margin distributions, fig. 21 shows local resampling intervals, and fig. 22 exposes the matched selected-versus-baseline correctness table used by the paired comparison.

Table 12: Deterministic no-retrain robustness smoke test from `output/data/ml_robustness_report.json`.

Candidate	Transform	Accuracy	Samples
softmax linear	identity	88.2%	500
softmax linear	shift_right_1	81.4%	500
softmax linear	shift_down_1	78.6%	500
softmax linear	low_contrast_0_85	88.0%	500
mlp relu 32	identity	88.6%	500
mlp relu 32	shift_right_1	82.4%	500
mlp relu 32	shift_down_1	80.2%	500
mlp relu 32	low_contrast_0_85	88.2%	500
tiny patch attention	identity	30.4%	500
tiny patch attention	shift_right_1	29.8%	500
tiny patch attention	shift_down_1	25.8%	500
tiny patch attention	low_contrast_0_85	24.6%	500
mlp tanh 64	identity	89.4%	500
mlp tanh 64	shift_right_1	83.2%	500
mlp tanh 64	shift_down_1	80.0%	500
mlp tanh 64	low_contrast_0_85	89.8%	500

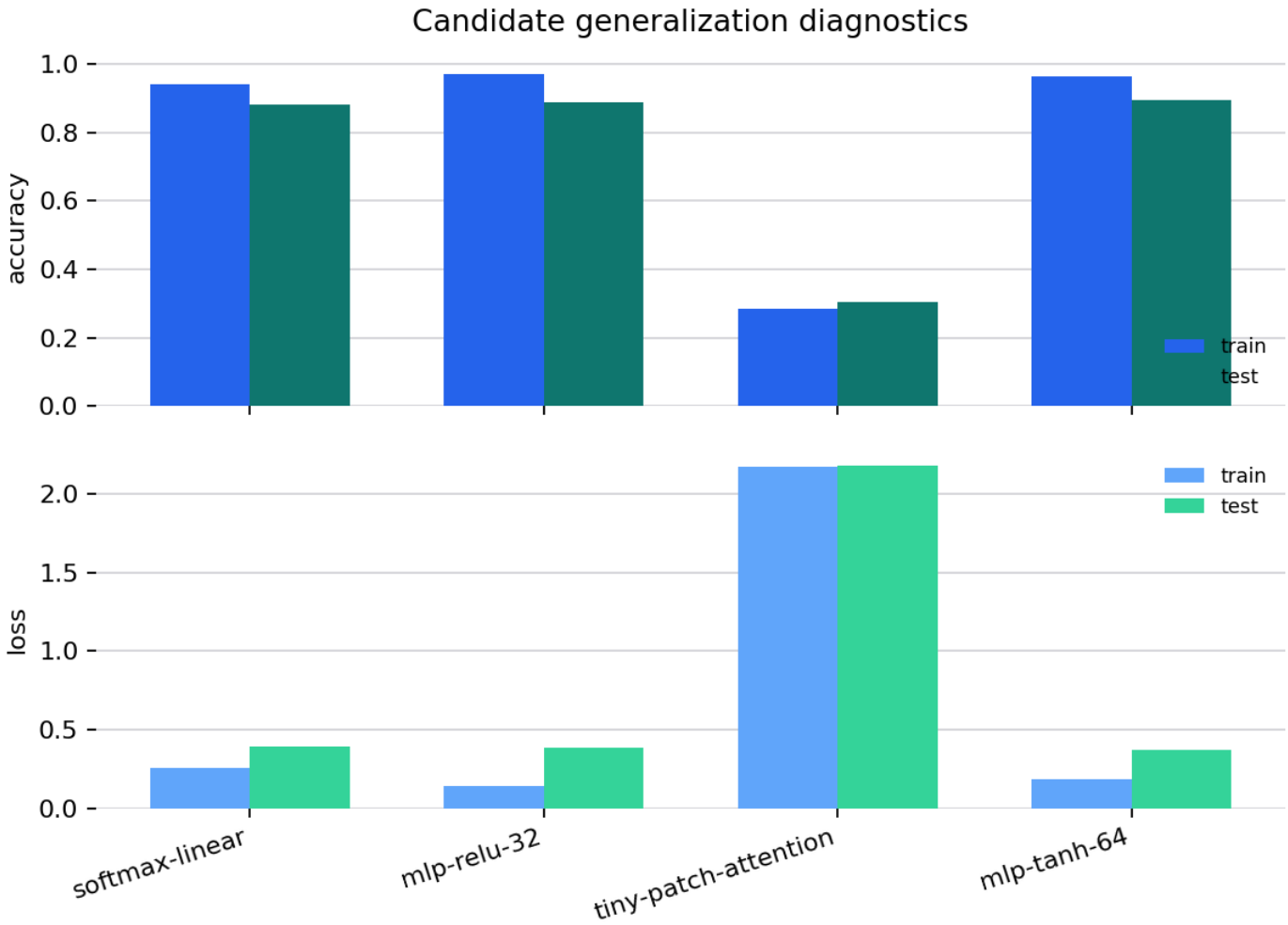

Figure 18: Train/test accuracy and loss for evaluated candidates from output/data/ml_classification_diagnostics.json; the plot exposes local generalization gaps without claiming full-dataset behavior. Generation method: Grouped train/test accuracy and loss bars by evaluated candidate. Registry metadata records the generation method, source artifact, and claim boundary for validation.

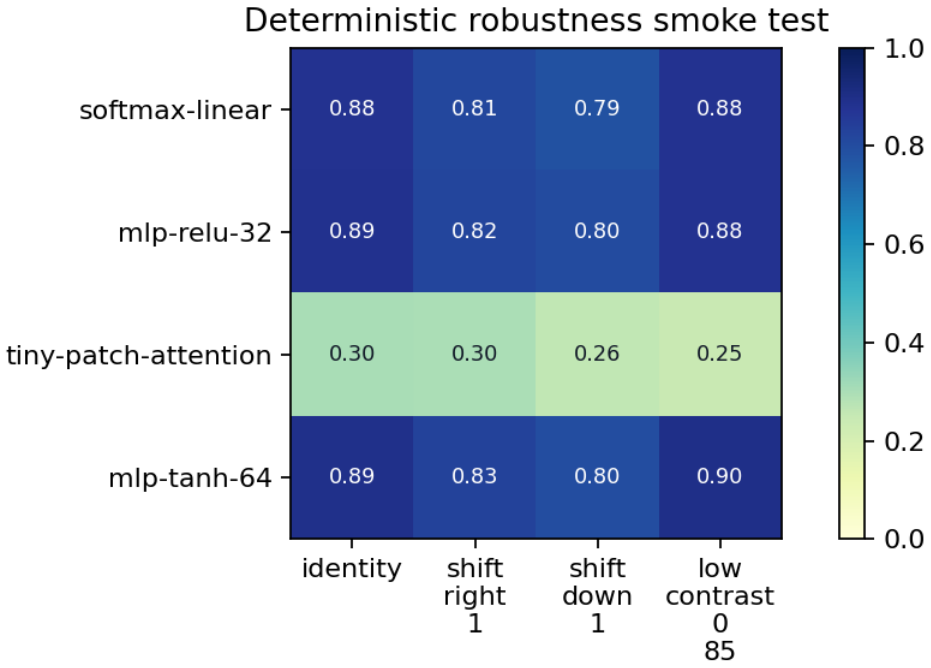

Figure 19: Accuracy for evaluated candidates under identity, one-pixel shifts, and low contrast from output/data/ml_robustness_report.json; the deterministic transforms provide a bounded smoke test only. Generation method: Candidate-by-transform accuracy heatmap for deterministic perturbations. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 13: Accepted-candidate probability diagnostics from `output/data/ml_probability_diagnostics.json`.

Statistic	Value
Mean confidence	87.9%
Mean correct confidence	90.9%
Mean error confidence	63.1%
Mean margin	80.2%
Mean correct margin	84.9%
Mean error margin	40.3%
Low-margin count	22

Table 14: Deterministic percentile-bootstrap intervals from `output/data/ml_bootstrap_intervals.json`.

Metric	Observed	CI low	CI high	Resample mean
accuracy	89.4%	86.4%	92.0%	89.4%
macro F1	89.4%	86.5%	91.9%	89.3%

Table 15: Accepted-candidate versus baseline paired comparison from `output/data/ml_paired_comparison.json`.

Matched comparison statistic	Value
Both correct	403
Accepted only correct	44
Baseline only correct	10
Both wrong	43
Discordant examples	54
Exact McNemar p	0.000
Net accuracy gain	6.8%

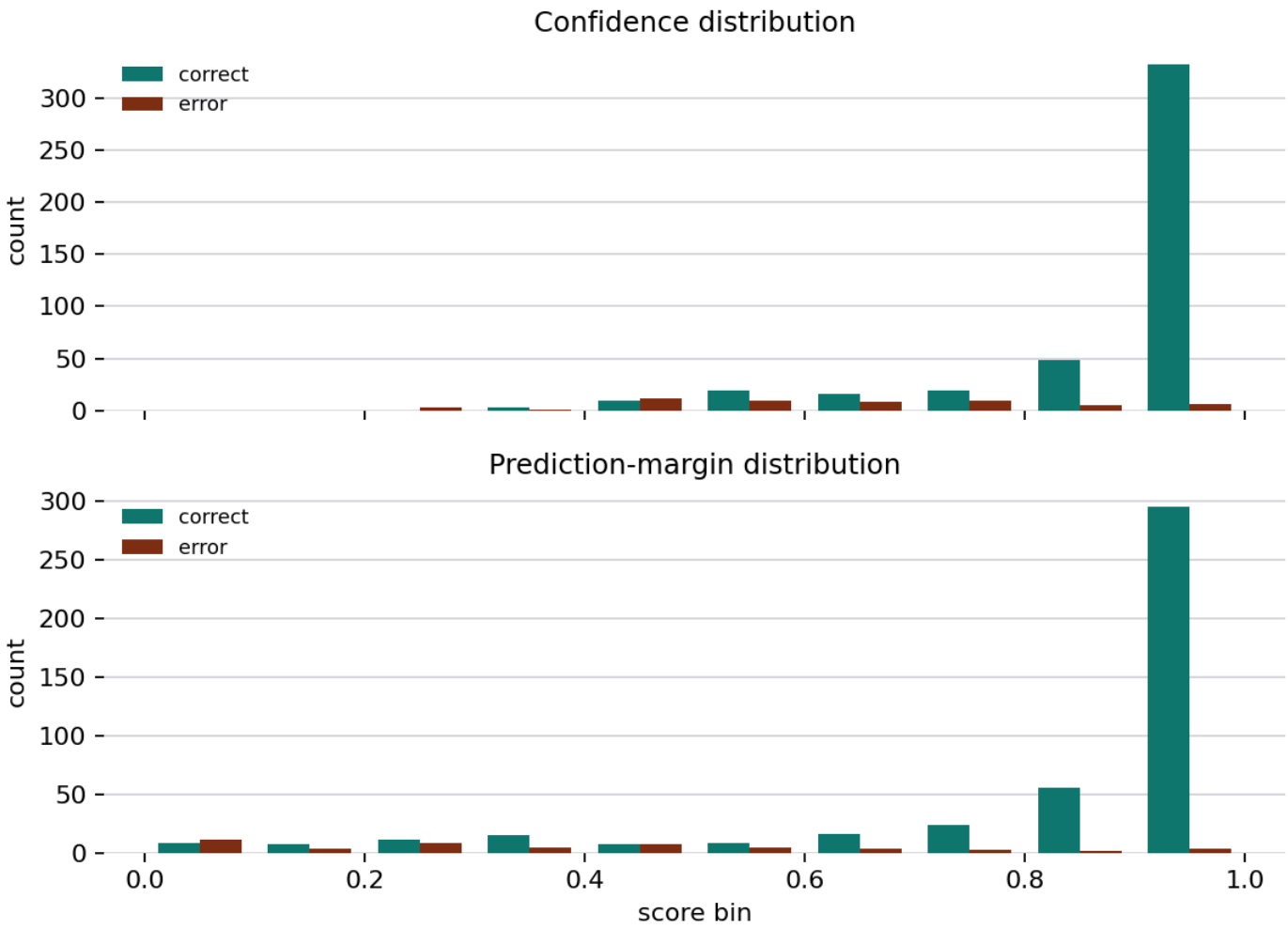

Figure 20: Confidence and prediction-margin histograms for exp-mlp-tanh-64 from output/data/ml_probability_diagnostics.json; the figure separates correct and incorrect local test predictions. Generation method: Confidence and margin histograms split by correctness. Registry metadata records the generation method, source artifact, and claim boundary for validation.

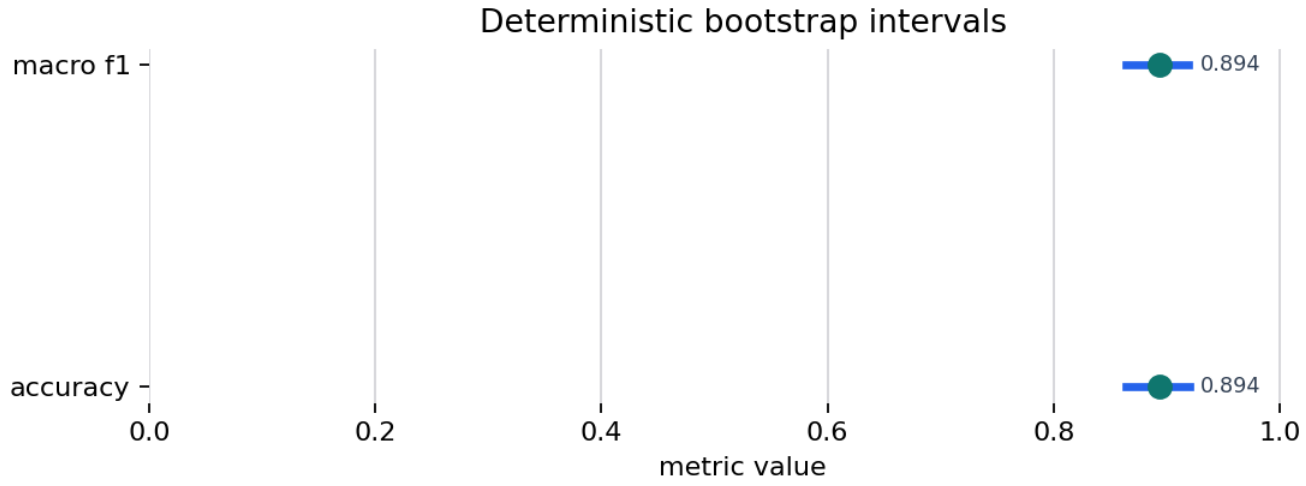

Figure 21: Deterministic percentile-bootstrap intervals for exp-mlp-tanh-64 from output/data/ml_bootstrap_intervals.json; the intervals summarize local sampling variation for accuracy and macro F1. Generation method: Horizontal percentile-bootstrap interval plot. Registry metadata records the generation method, source artifact, and claim boundary for validation.

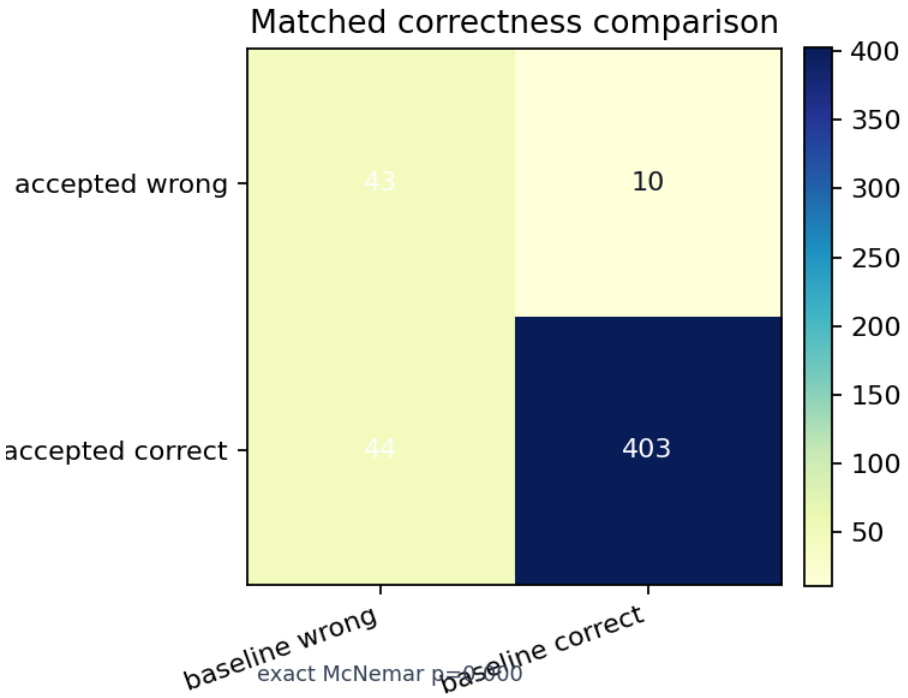

Figure 22: Matched correctness comparison between exp-mlp-tanh-64 and the nearest_centroid_baseline baseline from output/data/ml_paired_comparison.json; discordant cells support the paired test summary. Generation method: Matched accepted-versus-baseline correctness heatmap. Registry metadata records the generation method, source artifact, and claim boundary for validation.

4.4.3 Probability Quality And Selective Prediction

Probability-quality diagnostics report Brier score `0.161`, negative log likelihood `0.361`, top-2 accuracy `95.6%`, and Cohen kappa `0.882` for the selected candidate. At the highest configured confidence threshold, retained coverage is `67.8%` and selective accuracy is `98.2%`.

The selective-prediction view in fig. 23 reports the configured confidence-threshold trade-off: retaining fewer predictions can raise selective accuracy, but the coverage table keeps that trade-off explicit. The candidate probability-quality comparison in fig. 24 keeps accuracy separate from proper-score behavior, so the selected candidate is not treated as automatically best on every diagnostic axis.

Table 16: Accepted-candidate statistical summary from `output/data/ml_statistical_summary.json`.

Statistic	Value
Accuracy	89.4%
Balanced accuracy	89.4%
Macro F1	89.4%
Top-2 accuracy	95.6%
Cohen kappa	0.882
Brier score	0.161
Negative log likelihood	0.361
Expected calibration error	2.9%

Table 17: Selective-accuracy threshold table from `output/data/ml_statistical_summary.json`.

Confidence threshold	Coverage	Selective accuracy	Retained	Errors
50.0%	94.4%	92.2%	472	37
60.0%	88.8%	93.7%	444	28
70.0%	84.0%	95.2%	420	20
80.0%	78.4%	97.2%	392	11
90.0%	67.8%	98.2%	339	6

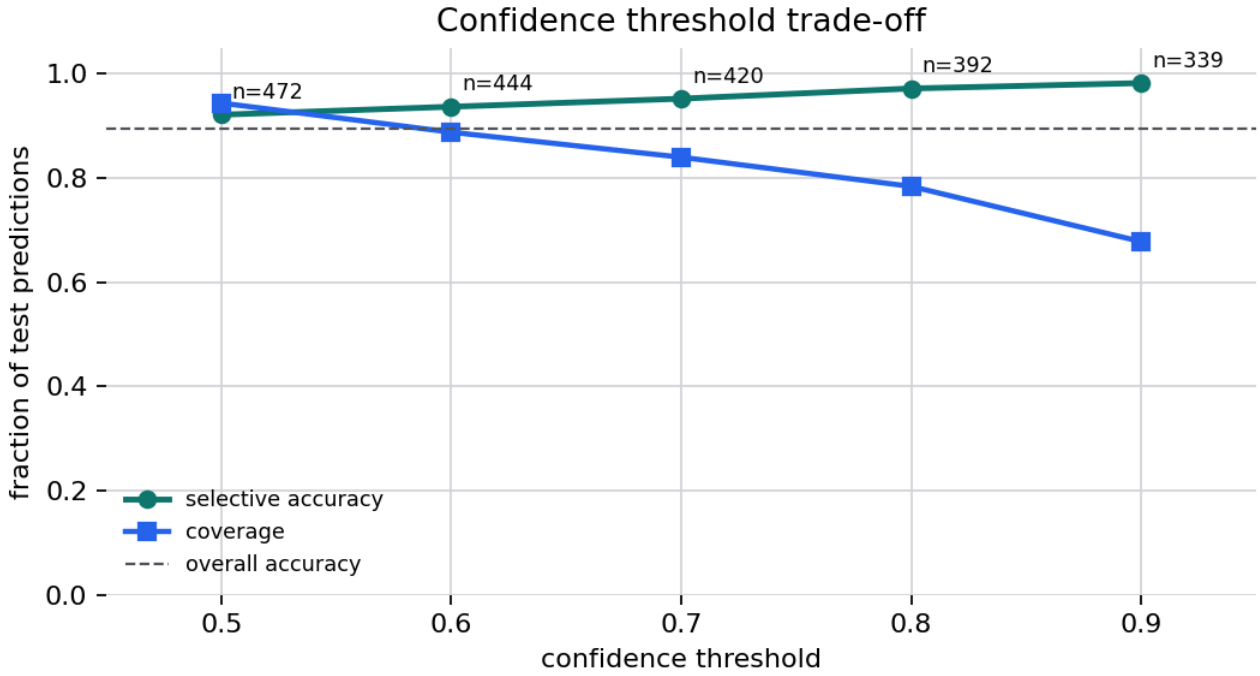

Figure 23: Confidence-threshold trade-off for exp-mlp-tanh-64 from output/data/ml_statistical_summary.json; the plot compares retained coverage, selective accuracy, and the unthresholded accepted-candidate accuracy on the fixed local split. Generation method: Confidence-threshold coverage and selective-accuracy line chart. Registry metadata records the generation method, source artifact, and claim boundary for validation.

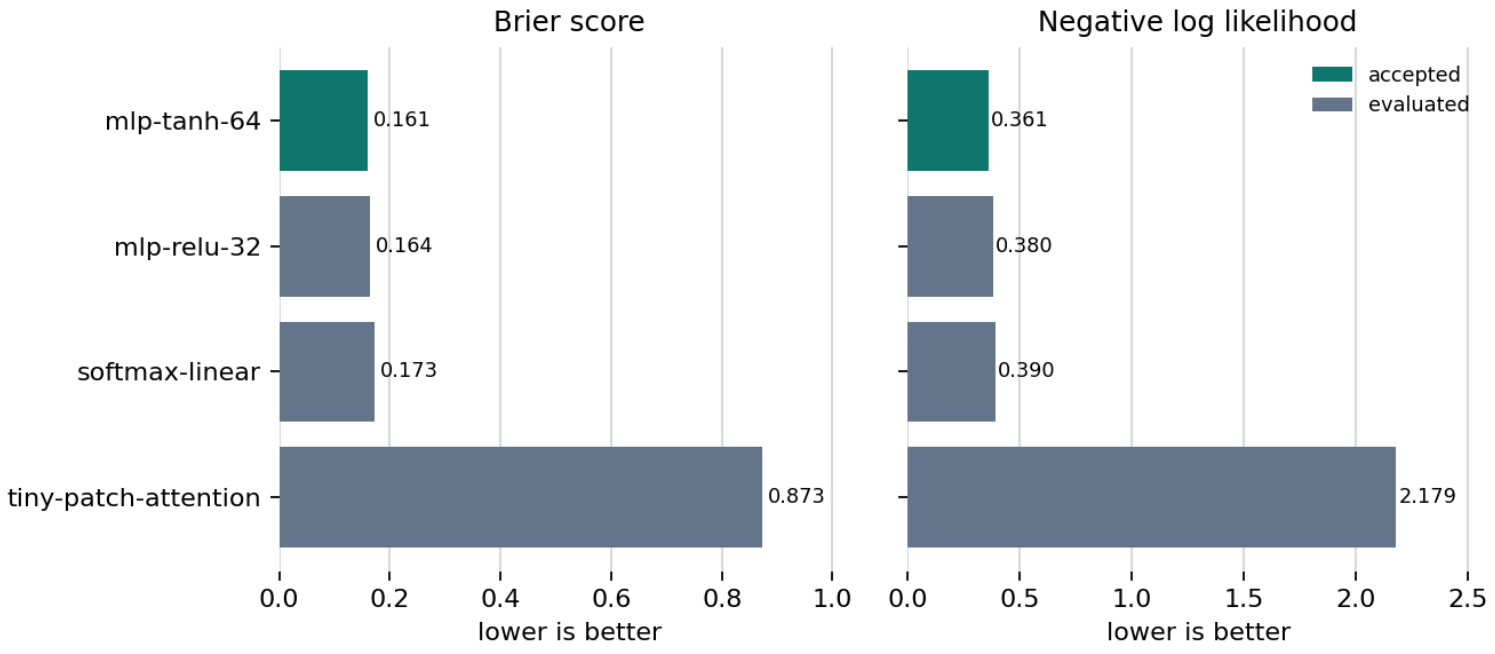

Figure 24: Brier score and negative log likelihood for evaluated candidates from output/data/ml_statistical_summary.json; lower values indicate better probability quality within the configured local run, and the accepted candidate is highlighted. Generation method: Brier score and negative-log-likelihood bar comparison. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 18: Candidate probability-quality table from `output/data/ml_statistical_summary.json`.

Candidate	Accuracy	Top-2 accuracy	Brier	NLL	Mean confidence
softmax linear	88.2%	95.8%	0.173	0.390	85.1%
mlp relu 32	88.6%	95.6%	0.164	0.380	89.9%
tiny patch attention	30.4%	46.6%	0.873	2.179	13.1%
mlp tanh 64	89.4%	95.6%	0.161	0.361	87.9%

4.5 Candidate Ledger

Table 19: Candidate lifecycle ledger from `output/data/ml_candidate_ledger.json`.

Candidate	Model	Status	Test accuracy	Parameters
baseline	nearest-centroid	baseline	82.6%	7840
softmax linear	softmax regression	rejected	88.2%	7850
mlp relu 32	MLP	rejected	88.6%	25450
tiny patch attention	tiny patch-attention	rejected	30.4%	5994
mlp tanh 64	MLP	accepted	89.4%	50890
mlp relu 64 deferred	MLP	deferred	N/A	0

The candidate-selection audit separates the objective ranking from descriptive diagnostics. It records the configured metric, Wilson interval, probability quality, parameter count, and deterministic tie-break context for each evaluated candidate. The diagnostic-boundary table states what each generated surface supports and what it does not support.

Table 20: Candidate-selection audit from `output/data/ml_candidate_selection_audit.json`; the objective metric ranks candidates, while probability diagnostics and parameter counts audit the chosen tie-break context.

Rank	Candidate	Status	Accuracy	Wilson 95%	Brier	NLL	Parameters
1	mlp tanh 64	accepted	89.4%	86.4% to 91.8%	0.161	0.361	50890
2	mlp relu 32	rejected	88.6%	85.5% to 91.1%	0.164	0.380	25450
3	softmax linear	rejected	88.2%	85.1% to 90.7%	0.173	0.390	7850
4	tiny patch attention	rejected	30.4%	26.5% to 34.6%	0.873	2.179	5994

Table 21: Diagnostic claim-boundary table from `output/data/ml_diagnostic_boundary.json`.

Surface	Source	Method	Supports	Does not support
objective selection	task results	rank evaluated candidates by configured held-out metric and deterministic tie…	accepted-candidate selection within this fixed local task	full MNIST state-of-the-art, external benchmark leadership, or universal mode…
descriptive diagnostics	class diagnostics	per-class metrics, calibration, probability quality, and paired comparison	local error analysis and uncertainty description	population-level certification or deployment readiness
robustness smoke test	robustness report	deterministic no-retrain transforms applied to the fixed test split	small perturbation smoke-test evidence	adversarial robustness or distribution-shift robustness
artifact integrity	integrity attestation	local SHA-256 checks over declared inputs and generated artifacts	local artifact integrity evidence for the run	external signing, production SLSA compliance, or runtime intrusion detection
review governance	review decisions	deferred generated review decisions with human review required	readiness for human review	machine self-approval or publication acceptance

4.6 Readiness And Review Artifacts

The broader AutoResearch run writes the reproducibility, benchmark, review, and manuscript-hydration surfaces summarized below. The schema manifest records 31 schema-versioned governance payload(s), and the local research-object manifest records 84 observed

artifact record(s) with checksums and approval state `false`. The phase ledger records 3 settlement pass(es), while the figure-quality report covers 25 registered figure(s) with validity `false`.

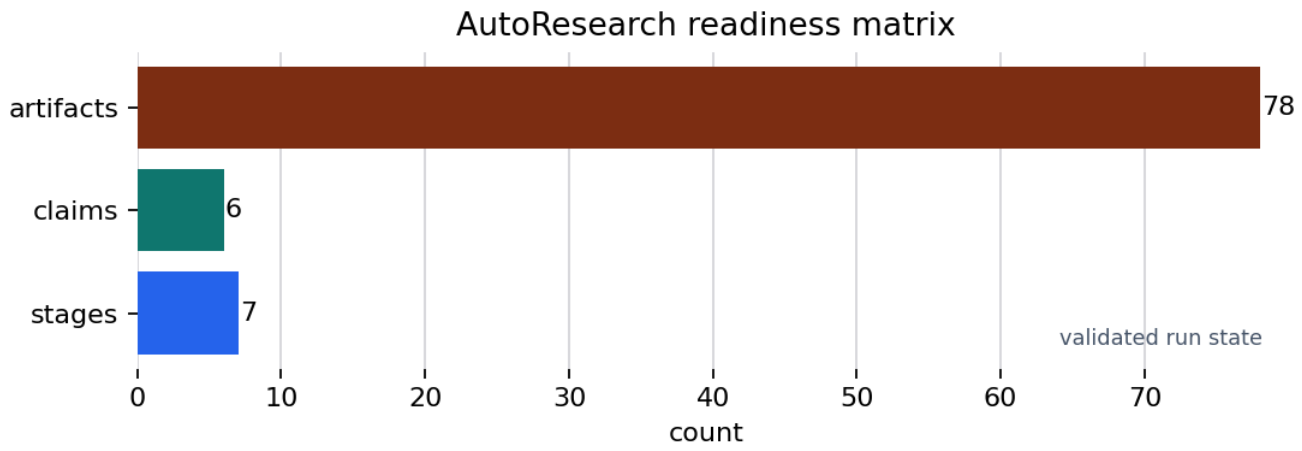

Figure 25: Validated AutoResearch run with 7 stages, 6 supported claims, and 78 required artifacts from output/data/autoresearch_loop.json; the count summarizes readiness artifacts, not human approval. Generation method: Horizontal count summary from final loop metrics. Registry metadata records the generation method, source artifact, and claim boundary for validation.

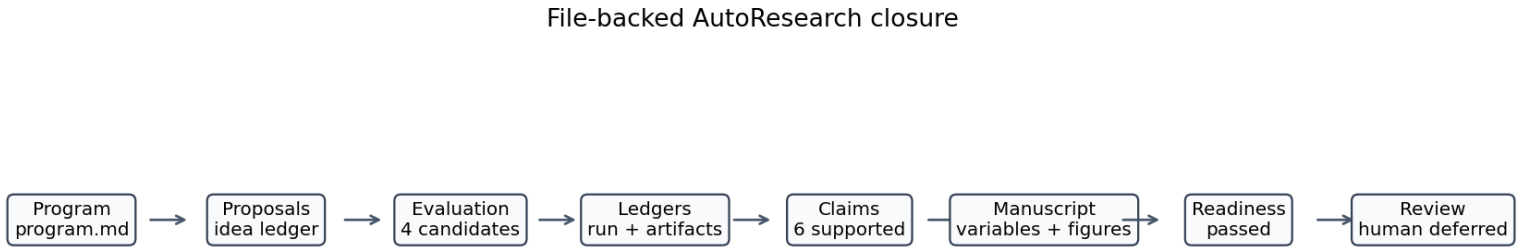

Figure 26: File-backed AutoResearch closure from program through review, with 6 supported claims and readiness passed; review remains a deferred human decision and the provenance path remains inspectable. Generation method: File-backed process-flow diagram from final loop state. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 22: Generated artifact manifest from `output/reports/artifact_manifest.json`.

Artifact	Role	Bytes
output/data/autoresearch_claims.json	Loop artifact	1766
output/data/autoresearch_evidence_overview.json	Evidence registry	4436
output/data/autoresearch_integrity_attestation.json	Security evidence	22755
output/data/autoresearch_inventory_export.json	Security evidence	20203
output/data/autoresearch_loop.json	Loop artifact	16085
output/data/autoresearch_phase_ledger.json	Run or candidate ledger	3779
output/data/autoresearch_plan.json	Loop artifact	17361
output/data/autoresearch_review_packet.json	Review packet	16204
output/data/autoresearch_schema_manifest.json	Loop artifact	7226
output/data/autoresearch_security_profile.json	Security evidence	1537
output/data/autoresearch_stage_matrix.csv	Loop artifact	749
output/data/autoresearch_supply_chain_inventory.json	Security evidence	21459
output/data/autoresearch_threat_model.json	Security evidence	6370
output/data/benchmark_boundary.json	Benchmark grading	2363
output/data/benchmark_scores.json	Benchmark grading	621

Artifact	Role	Bytes
output/data/figure__quality__report.json	Loop artifact	15966
output/data/figure__style.json	Loop artifact	1117
output/data/idea__ledger.json	Run or candidate ledger	5233
output/data/manuscript__figure__blocks.json	Manuscript hydration	13062
output/data/manuscript__variable__provenance.json	Manuscript hydration	30891
output/data/manuscript__variables.json	Manuscript hydration	54967
output/data/ml__bootstrap__intervals.json	Loop artifact	615
output/data/ml__calibration__bin__intervals.json	Loop artifact	2879
output/data/ml__calibration__report.json	Loop artifact	2201
output/data/ml__candidate__intervals.json	Loop artifact	1577
output/data/ml__candidate__ledger.json	Run or candidate ledger	570872
output/data/ml__candidate__rank__stability.json	Loop artifact	3135
output/data/ml__candidate__selection__audit.json	Loop artifact	2145
output/data/ml__class__balance.json	Loop artifact	2393
output/data/ml__classification__diagnostics.json	Loop artifact	4246
output/data/ml__confusion__matrix.csv	Loop artifact	271
output/data/ml__diagnostic__boundary.json	Loop artifact	2064
output/data/ml__error__examples.json	Loop artifact	989
output/data/ml__paired__comparison.json	Loop artifact	470
output/data/ml__prediction__records.json	Loop artifact	989913
output/data/ml__probability__diagnostics.json	Loop artifact	3045
output/data/ml__robustness__report.json	Loop artifact	3758
output/data/ml__statistical__summary.json	Loop artifact	3032
output/data/ml__task__results.json	Loop artifact	703566
output/data/ml__training__diagnostics.json	Loop artifact	2968
output/data/ml__training__history.csv	Loop artifact	6775
output/data/mnist__task__config.json	Loop artifact	3926
output/data/research__object__manifest.json	Loop artifact	20586
output/data/research__program.json	Loop artifact	965
output/data/review__decisions.json	Review packet	669
output/data/run__ledger.json	Run or candidate ledger	328
output/data/transmission__manifest.json	Loop artifact	366
../figures/autoresearch__candidate__lifecycle.png	Generated figure	28056
../figures/autoresearch__closure__flow.png	Generated figure	40873
../figures/autoresearch__integrity__chain.png	Generated figure	48228
../figures/autoresearch__security__control__matrix.png	Generated figure	86494
../figures/autoresearch__stage__matrix.png	Generated figure	23486
../figures/figure__registry.json	Loop artifact	30674
../figures/ml__bootstrap__intervals.png	Generated figure	21830
../figures/ml__calibration__reliability.png	Generated figure	73855
../figures/ml__candidate__rank__stability.png	Generated figure	43670
../figures/ml__candidate__scores.png	Generated figure	59947
../figures/ml__classification__metrics__heatmap.png	Generated figure	55133
../figures/ml__complexity__accuracy.png	Generated figure	35016
../figures/ml__confusion__matrix.png	Generated figure	65918
../figures/ml__confusion__pairs.png	Generated figure	33982
../figures/ml__generalization__gap.png	Generated figure	48554
../figures/ml__learning__curves.png	Generated figure	59360
../figures/ml__paired__correctness.png	Generated figure	43306
../figures/ml__per__class__accuracy.png	Generated figure	34938
../figures/ml__probability__margin__distribution.png	Generated figure	41747
../figures/ml__probability__quality.png	Generated figure	38072
../figures/ml__robustness__matrix.png	Generated figure	51292
../figures/ml__selective__accuracy.png	Generated figure	48332
../figures/ml__training__dynamics.png	Generated figure	50908
../figures/mnist__class__balance.png	Generated figure	27006
../figures/mnist__error__examples.png	Generated figure	28446
../figures/mnist__subset__contact__sheet.png	Generated figure	23682
../figures/transmission__pairing.png	Generated figure	10729
output/manuscript/00__abstract.md	Manuscript hydration	1451
output/manuscript/01__introduction.md	Manuscript hydration	14774

Artifact	Role	Bytes
output/manuscript/02_methodology.md	Manuscript hydration	21097
output/manuscript/03_results.md	Manuscript hydration	39362
output/manuscript/04_conclusion.md	Manuscript hydration	2642
output/manuscript/99_references.md	Manuscript hydration	58
output/manuscript/config.yaml	Manuscript hydration	2507
output/manuscript/preamble.md	Manuscript hydration	137
output/manuscript/references.bib	Manuscript hydration	23228
output/reports/autoresearch_evidence_overview.md	Evidence registry	1136
output/reports/autoresearch_loop.json	Loop artifact	16085
output/reports/autoresearch_loop.md	Loop artifact	1982
output/reports/autoresearch_readiness.json	Readiness validation	18488
output/reports/autoresearch_readiness.md	Readiness validation	500
output/reports/autoresearch_review_packet.md	Review packet	912
output/reports/autoresearch_security_review.md	Review packet	1104
output/reports/autoresearch_summary.md	Loop artifact	255
output/reports/benchmark_readiness_smoke.json	Benchmark grading	778
output/reports/ml_benchmark_score.json	Benchmark grading	382
output/reports/ml_experiment_report.md	Loop artifact	1687
output/reports/test_results.json	Loop artifact	616
output/reports/test_results.md	Loop artifact	293
output/data/autoresearch_evidence_overview.json	Evidence registry	4436
output/data/autoresearch_integrity_attestation.json	Security evidence	22755
output/data/autoresearch_inventory_export.json	Security evidence	20203
output/data/autoresearch_loop.json	Loop artifact	16085
output/data/autoresearch_phase_ledger.json	Run or candidate ledger	3779
output/data/autoresearch_review_packet.json	Review packet	16204
output/data/autoresearch_schema_manifest.json	Loop artifact	7226
output/data/autoresearch_security_profile.json	Security evidence	1537
output/data/autoresearch_supply_chain_inventory.json	Security evidence	21459
output/data/autoresearch_threat_model.json	Security evidence	6370
output/data/benchmark_scores.json	Benchmark grading	621
output/data/figure_quality_report.json	Loop artifact	16001
output/data/figure_style.json	Loop artifact	1117
output/data/idea_ledger.json	Run or candidate ledger	5233
output/data/manuscript_variables.json	Manuscript hydration	54968
output/data/ml_candidate_ledger.json	Run or candidate ledger	570872
output/data/research_object_manifest.json	Loop artifact	20652
output/data/review_decisions.json	Review packet	669
output/manuscript/03_results.md	Manuscript hydration	39363
output/reports/autoresearch_loop.json	Loop artifact	16085

Table 23: Review-gate decisions from `output/data/review_decisions.json`.

Gate	Required	Decision	Rationale
proposal_review	True	deferred	Decision is read from human_review.yaml when present; generated readiness is not approval.
evidence_review	True	deferred	Decision is read from human_review.yaml when present; generated readiness is not approval.

Table 24: Benchmark grading table from `output/data/benchmark_scores.json`.

Benchmark task	Status	Score	Grading output
readiness-smoke	graded	1	output/reports/benchmark_readin
ml-loop-score	graded	1	output/reports/ml_benchmark_sc

Table 25: Deterministic phase ledger from `output/data/autoresearch_phase_ledger.json`; settlement order is not an autonomy claim.

Phase	Order	Group	Observed artifacts	Description
intrinsic readiness	1	readiness	3	validate configured project-intrinsic contracts
core artifacts	2	loop	0	write plan, stage matrix, and provisional loop outputs
evidence registry	3	evidence	2	write local evidence registry
ml task	4	ml	54	run fixed-seed bounded candidate evaluation
method contract	5	governance	3	write program, idea, run, review, and benchmark ledgers
provisional payloads	6	settlement	12	refresh loop payloads before extrinsic validation
security artifacts	7	security	10	write local security and integrity evidence
final visuals	8	figures	54	write final registry-backed figures
manuscript hydration	9	manuscript	9	write variables, provenance, and figure blocks
readiness manifest	10	settlement	12	refresh checksum manifest before extrinsic validation
schema manifest	11	schema	2	write generated JSON schema-version manifest
research object manifest	12	packaging	2	write local research-object manifest
extrinsic readiness	13	readiness	3	validate generated artifacts and extrinsic contracts
final schema manifest	14	schema	2	refresh schema manifest after final payload updates
final research object manifest	15	packaging	2	refresh local research-object manifest
artifact manifest	16	settlement	12	write final artifact checksum manifest

Table 26: Figure-quality checks from `output/data/figure_quality_report.json`; 25 registered figure(s) were checked.

Figure	Pixels	Variance	Source exists	Nonblank
fig:autoresearch_candidate_lifecycle	1184x480	0.082	True	True
fig:autoresearch_closure_flow	1664x448	0.015	True	True
fig:autoresearch_integrity_chain	1440x736	0.040	True	True
fig:autoresearch_security_decision_matrix	1472x736	0.014	True	True
fig:autoresearch_stage_matrix	1120x416	0.079	True	True
fig:ml_bootstrap_intervals	1152x448	0.009	True	True
fig:ml_calibration_reliability	1152x832	0.015	True	True
fig:ml_candidate_rank_stability	1408x608	0.053	True	True
fig:ml_candidate_scores	1376x688	0.013	True	True
fig:ml_classification_metric_heatmap	928x832	0.092	True	True
fig:ml_complexity_accuracy	1120x608	0.010	True	True
fig:ml_confusion_matrix	896x768	0.045	True	True
fig:ml_confusion_pairs	1152x576	0.129	True	True

Figure	Pixels	Variance	Source exists	Nonblank
fig:ml_generalization_gap	1184x864	0.059	True	True
fig:ml_learning_curves	1216x608	0.018	True	True
fig:ml_paired_correctness	768x672	0.075	True	True
fig:ml_per_class_accuracy	1152x512	0.098	True	True
fig:ml_probability_margin_distribution	1184x864	0.024	True	True
fig:ml_probability_quality	1344x608	0.046	True	True
fig:ml_robustness_matrix	1280x608	0.073	True	True
fig:ml_selective_accuracy	1088x608	0.016	True	True
fig:ml_training_dynamics	1408x608	0.072	True	True
fig:mnist_class_balance	1216x544	0.071	True	True
fig:mnist_error_examples	1280x736	0.234	True	True
fig:mnist_subset_contact_sheet	1216x544	0.230	True	True

4.7 Security Readiness And Integrity Evidence

The local security profile reports attestation status `passed` after checking `80` file record(s), with `0` missing record(s) and `0` checksum mismatch(es). The inventory contains `14` input record(s) and `69` generated-artifact record(s). The integrity-chain figure is fig. 27. These values support local artifact-integrity claims only; they do not claim external signing, production SLSA compliance, or runtime security monitoring.

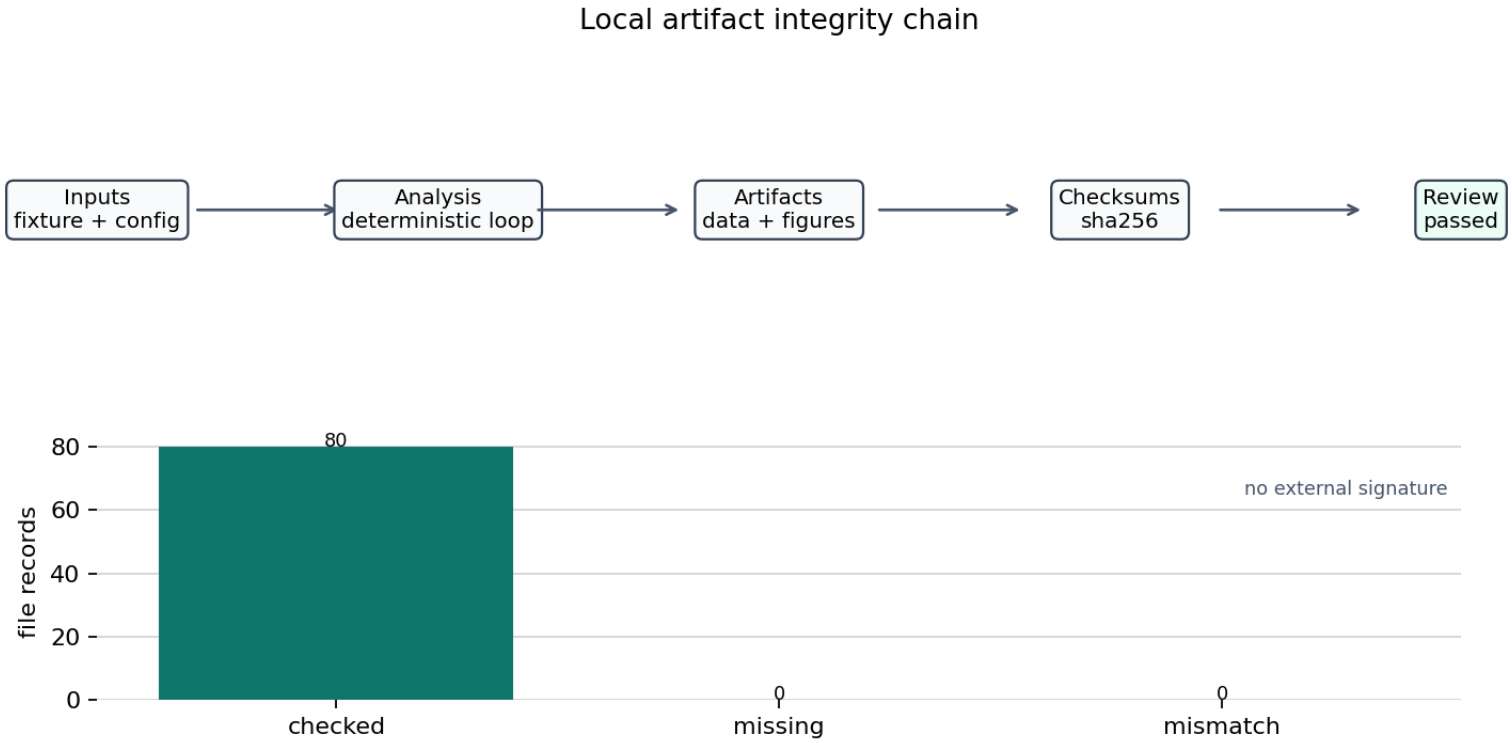

Figure 27: Local integrity chain from output/data/autoresearch_integrity_attestation.json; checksums summarize the observed run artifacts and remain unsigned local evidence. Generation method: Local checksum attestation chain with checked, missing, and mismatch counts. Registry metadata records the generation method, source artifact, and claim boundary for validation.

Table 27: Integrity-attestation summary from `output/data/autoresearch_integrity_attestation.json`.

Integrity field	Value
status	passed
algorithm	sha256
checked files	80
missing files	0
checksum mismatches	0
external signature	False

4.8 Manuscript Hydration Provenance

The final run supports 6 manuscript-facing claim(s) and checks 78 required artifact(s). The rendered manuscript uses injected variables from generated data payloads, so the abstract and results track the latest analysis run rather than hard-coded counts. The final readiness status is passed; generated review decisions are recorded as deferred for human review rather than as self-approval.

Table 28: Variable provenance summary from output/data/manuscript_variable_provenance.json.

Source artifact	Injected variables or fragments
output/data/autoresearch_integrity_attestation.json	5
output/data/autoresearch_loop.json	57
output/data/autoresearch_phase_ledger.json	2
output/data/autoresearch_schema_manifest.json	1
output/data/autoresearch_security_profile.json	6
output/data/autoresearch_supply_chain_inventory.json	3
output/data/autoresearch_threat_model.json	4
output/data/benchmark_scores.json	1
output/data/figure_quality_report.json	3
output/data/manuscript_variable_provenance.json	1
output/data/ml_bootstrap_intervals.json	3
output/data/ml_calibration_bin_intervals.json	1
output/data/ml_calibration_report.json	3
output/data/ml_candidate_intervals.json	1
output/data/ml_candidate_ledger.json	1
output/data/ml_candidate_rank_stability.json	3
output/data/ml_candidate_selection_audit.json	1
output/data/ml_class_balance.json	3
output/data/ml_classification_diagnostics.json	5
output/data/ml_diagnostic_boundary.json	1
output/data/ml_paired_comparison.json	3
output/data/ml_probability_diagnostics.json	4
output/data/ml_robustness_report.json	2
output/data/ml_statistical_summary.json	9
output/data/ml_task_results.json	39
output/data/ml_training_diagnostics.json	7
output/data/research_object_manifest.json	2
output/data/review_decisions.json	1
../figures/figure_registry.json	51
output/reports/artifact_manifest.json	1

5 Conclusion

`template_autoresearch_project` shows how a tractable AutoResearch task can be represented as reproducible template infrastructure. The `small MNIST neural-network classification` experiment is small by design, but it exercises the method surfaces that matter for a public default: a declared research program, local input-data provenance, bounded candidate families (`MLP, neares t-centroid, softmax regression, tiny patch-attention`), objective scoring, budget and cost ledgers, evidence-linked claims, generated figures, benchmark grading, manuscript-variable hydration, loop-settlement ledgers, figure-quality checks, validation, local integrity attestation (`passed`), and human review gates.

The broader field is converging across five related trends. Autoresearch systems seek end-to-end automation of ideation, experiment execution, writing, and review. Autoformalization pushes informal claims toward machine-checkable representations. ML-for-ML systems use search, evaluation, and evolutionary code loops to improve algorithms. Agentic science composes retrieval, planning, experimentation, and communication into higher-autonomy workflows. Structured knowledge synthesis supplies the substrate that all of those systems need: source-linked artifacts, knowledge graphs, tensors, vector spaces, and uncertainty-aware memories that can be navigated without losing provenance.

This exemplar makes a deliberately narrower contribution. It does not mine the live literature, construct a Conceptual Nexus Model, search for Lean proofs, mutate arbitrary code, coordinate external agents, or approve its own paper. Instead it demonstrates governance infrastructure for automated research workflows: every important claim is hydrated from run artifacts, every generated figure is registry-bound and locally checked for source and pixel integrity, every budget and candidate decision is recorded, and review state remains outside generated self-approval. The security layer adds local inventory and checksum evidence while keeping external signing and production deployment claims out of scope.

The durable product is therefore not only the `MNIST` metric. It is a reproducible research process whose data, claims, captions, figures, review gates, and manuscript variables can be reviewed, rerun, and extended. As more ambitious automated-science systems mature, this kind of offline, deterministic, evidence-governed baseline remains useful because it preserves the part of scientific automation that should not be optional: inspectable provenance, explicit limits, and human-governed publication judgment.

6 References

References are managed in `references.bib`.

References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650:857–863, 2026. doi: 10.1038/s41586-025-10072-4. URL https://www.nature.com/articles/s41586-025-10072-4.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5185–5194. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1522. URL https://aclanthology.org/D19-1522/.

Colby Banbury, Vijay Janapa Reddi, Peter Torelli, Jeremy Holleman, Nat Jeffries, Csaba Kiraly, Pietro Montino, David Kanter, Sebastian Ahmed, Danilo Pau, Urmish Thakker, Antonio Torrini, Pete Warden, Jay Cordaro, Giuseppe Di Guglielmo, Javier Duarte, Stephen Gibellini, Videet Parekh, Honson Tran, Nhan Tran, Xu Wenxu, and Zhang Xuesong. Mlperf tiny benchmark, 2021. URL https://arxiv.org/abs/2106.07597.

Vladimir Baulin, Austin Cook, Daniel Friedman, Janna Lumiruusu, Andrew Pashea, Shagor Rahman, and Benedikt Waldeck. The discovery engine: A framework for ai-driven synthesis and navigation of scientific knowledge landscapes, 2025. URL https://arxiv.org/abs/2505.17500.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Jun Shern Chan, Aakanksha Chowdhery, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering, 2024. URL https://arxiv.org/abs/2410.07095.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104. URL https://journals.sagepub.com/doi/10.1177/001316446002000104.

GPT Researcher Contributors. Gpt researcher, 2026. URL https://github.com/assafelovic/gpt-researcher.

Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998. doi: 10.1162/089976698300017197. URL https://dblp.org/rec/journals/neco/Dietterich98.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Karthik Duraisamy. Active inference ai systems for scientific discovery, 2025. URL https://arxiv.org/abs/2506.21329.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993. doi: 10.1201/9780429246593.

FutureHouse. Futurehouse platform: Superintelligent ai agents for scientific discovery, 2025. URL https://www.futurehouse.org/research-announcements/launching-futurehouse-platform-ai-agents.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. doi: 10.1145/3458723. URL https://www.microsoft.com/en-us/research/publication/datasheets-for-datasets/.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, Samantha M. Wright, Muhammed T. Razzak, et al. A multi-agent system for automating scientific discovery. *Nature*, 2026. doi: 10.1038/s41586-026-10652-y. URL https://www.nature.com/articles/s41586-026-10652-y.

Hannah Ginsborg. Kant's aesthetics and teleology, 2022. URL https://plato.stanford.edu/entries/kant-aesthetics/.

Mourad Gridach, Jay Nanavati, Khaldoun Zine El Abidine, Lenon Mendes, and Christina Mack. Agentic ai for scientific discovery: A survey of progress, challenges, and future directions, 2025. URL https://arxiv.org/abs/2503.08979.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Qianyue Hao, Fengli Xu, Yong Li, and James A. Evans. Artificial intelligence tools expand scientists' impact but contract science's focus. *Nature*, 649:1237–1243, 2026. doi: 10.1038/s41586-025-09922-y. URL https://www.nature.com/articles/s41586-025-09922-y.

Md Musaddaqul Hasib, Sumin Jo, Harsh Sinha, Jimin Song, Zhentao Liu, Hugh Galloway, Huey Huang, Kexun Zhang, Shou-Jiang Gao, Yu-Chiao Chiu, et al. A process-centric survey of ai for scientific discovery through the exhyte framework, 2025. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12776506/.

Mike Heddes, Igor Nunes, Tony Givargis, Alexandru Nicolau, and Alex Veidenbaum. Hyperdimensional computing: A framework for stochastic computation and symbolic ai. *Journal of Big Data*, 11(145), 2024. doi: 10.1186/s40537-024-01010-8. URL https://link.springer.com/article/10.1186/s40537-024-01010-8.

Qian Huang, Jeevana Priya Inala Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation, 2023. URL https://arxiv.org/abs/2310.03302.

Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklos Z. Horvath, Goran Zuzic, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 651:607–613, 2025. doi: 10.1038/s41586-025-09833-y. URL https://www.nature.com/articles/s41586-025-09833-y. Version of record 2026.

Andrej Karpathy. autoresearch, 2026. URL https://github.com/karpathy/autoresearch.

Lingdong Kong, Xian Sun, Wei Chow, Linfeng Li, Kevin Qinghong Lin, Xuan Billy Zhang, Song Wang, Rong Li, Qing Wu, Wei Gao, et al. Ai for auto-research: Roadmap and user guide, 2026. URL https://arxiv.org/abs/2605.18661.

Ankit Lala, Henry Mantsch, Cindy Zhou, Theodore LaBruna, Ronak Sinha, and Manav Choudhary. Paperqa: Retrieval-augmented generative agent for scientific research, 2023. URL https://arxiv.org/abs/2312.07559.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The mnist database of handwritten digits. URL https://yann.lecun.org/exdb/mnist/.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. URL https://bottou.org/papers/lecun-98h.

Simone Leo, Michael R. Crusoe, Laura Rodríguez-Navas, Raül Sirvent, Alexander Kanitz, Paul De Geest, et al. Recording provenance of workflow runs with ro-crate. *PLOS ONE*, 19(9):e0309210, 2024. doi: 10.1371/journal.pone.0309210. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0309210.

Gang Liu, Yihan Zhu, Jie Chen, and Meng Jiang. Scientific algorithm discovery by augmenting alphaevolve with deep research, 2025. URL https://arxiv.org/abs/2510.06056.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery, 2024a. URL https://arxiv.org/abs/2408.06292.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651:914–919, 2026. doi: 10.1038/s41586-026-10265-5. URL https://www.nature.com/articles/s41586-026-10265-5.

Jianqiao Lu, Yingjia Wan, Zhengying Liu, Yinya Huang, Jing Xiong, Chengwu Liu, Jianhao Shen, Hui Jin, Jipeng Zhang, Haiming Wang, et al. Process-driven autoformalization in lean 4, 2024b. URL https://arxiv.org/abs/2406.01940.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229, 2019. doi: 10.1145/3287560.3287596. URL https://research.google/pubs/model-cards-for-model-reporting/.

MITRE ATT&CK. Supply chain compromise, technique t1195. Online technique catalog, 2026. URL https://attack.mitre.org/techniques/T1195/.

Alvaro Moreno and Matteo Mossio. *Biological Autonomy: A Philosophical and Theoretical Enquiry*. Springer, 2015. doi: 10.1007/978-94-017-9837-2. URL https://link.springer.com/book/10.1007/978-94-017-9837-2.

Alexander Novikov, Ngan Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

Open Source Security Foundation. Supply-chain levels for software artifacts specification. Online specification, 2026. URL https://slsa.dev/spec/latest/.

Azim Ospanov, Farzan Farnia, and Roozbeh Yousefzadeh. Apollo: Automated llm and lean collaboration for advanced formal reasoning, 2025. URL https://arxiv.org/abs/2505.05758.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: A report from the neurips 2019 reproducibility program, 2020. URL https://arxiv.org/abs/2003.12206.

Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. Zero trust architecture. Special Publication 800-207, National Institute of Standards and Technology, 2020. URL https://www.nist.gov/publications/zero-trust-architecture-0.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. Assisting in writing wikipedia-like articles from scratch with large language models, 2024. URL https://arxiv.org/abs/2402.14207.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge, 2024. URL https://arxiv.org/abs/2409.13740.

Stian Soiland-Reyes, Peter Sefton, Mercè Crosas, Leyla Jael Castro, Frederik Coppens, José M. Fernández, Daniel Garijo, Björn Grüning, Marco La Rosa, Simone Leo, Eoghan Ó Carragáin, Marc Portier, Ana Trisovic, Paul Groth, and Carole Goble. Packaging research artefacts with ro-crate. *Data Science*, 5(2):97–138, 2022. doi: 10.3233/DS-210053. URL https://doi.org/10.3233/DS-210053.

Murugiah Souppaya, Karen Scarfone, and Donna Dodson. Secure software development framework (ssdf) version 1.1: Recommendations for mitigating the risk of software vulnerabilities. Special Publication 800-218, National Institute of Standards and Technology, 2022. URL https://csrc.nist.gov/pubs/sp/800/218/final.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning, 2014. URL https://arxiv.org/abs/1407.7722.

Francisco G. Varela, Humberto R. Maturana, and Ricardo Uribe. Autopoiesis: The organization of living systems, its characterization and a model. *BioSystems*, 5(4):187–196, 1974. doi: 10.1016/0303-2647(74)90031-8. URL https://repositorio.uchile.cl/handle/2250/160309.

Andreas Weber and Francisco J. Varela. Life after kant: Natural purposes and the autopoietic foundations of biological individuality. *Phenomenology and the Cognitive Sciences*, 1(2):97–125, 2002. doi: 10.1023/A:1020368120174. URL https://philpapers.org/rec/WEBLAK.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, et al. From ai for science to agentic science: A survey on autonomous scientific discovery, 2025. URL https://arxiv.org/abs/2508.14111.

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi: 10.1038/sdata.2016.18. URL https://www.nature.com/articles/sdata201618.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1927.10502953.

Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qian-wen Zhang, Di Yin, Xing Sun, and Xiao Huang. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2506.02404.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. URL https://arxiv.org/abs/2504.08066.

Viacheslav Yusupov, Maxim Rakhuba, and Evgeny Frolov. Knowledge graph completion with mixed geometry tensor factorization. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4924–4932. PMLR, 2025. URL https://proceedings.mlr.press/v258/yusupov25a.html.

Dongzhuoran Zhou, Yuqicheng Zhu, Xiaxia Wang, Hongkuan Zhou, Yuan He, Jiaoyan Chen, Steffen Staab, and Evgeny Kharlamov. What breaks knowledge graph based rag? benchmarking and empirical insights into reasoning under incomplete knowledge, 2026. URL https://arxiv.org/abs/2508.08344. Accepted at EACL 2026.

END OF TRANSMISSION

Release: v0.3.1 · DOI `10.5281/zenodo.20417016` · SHA-256 `pending…` · pairing pending

Figure 28: Integrity QR strip

Prior: *No prior releases.*