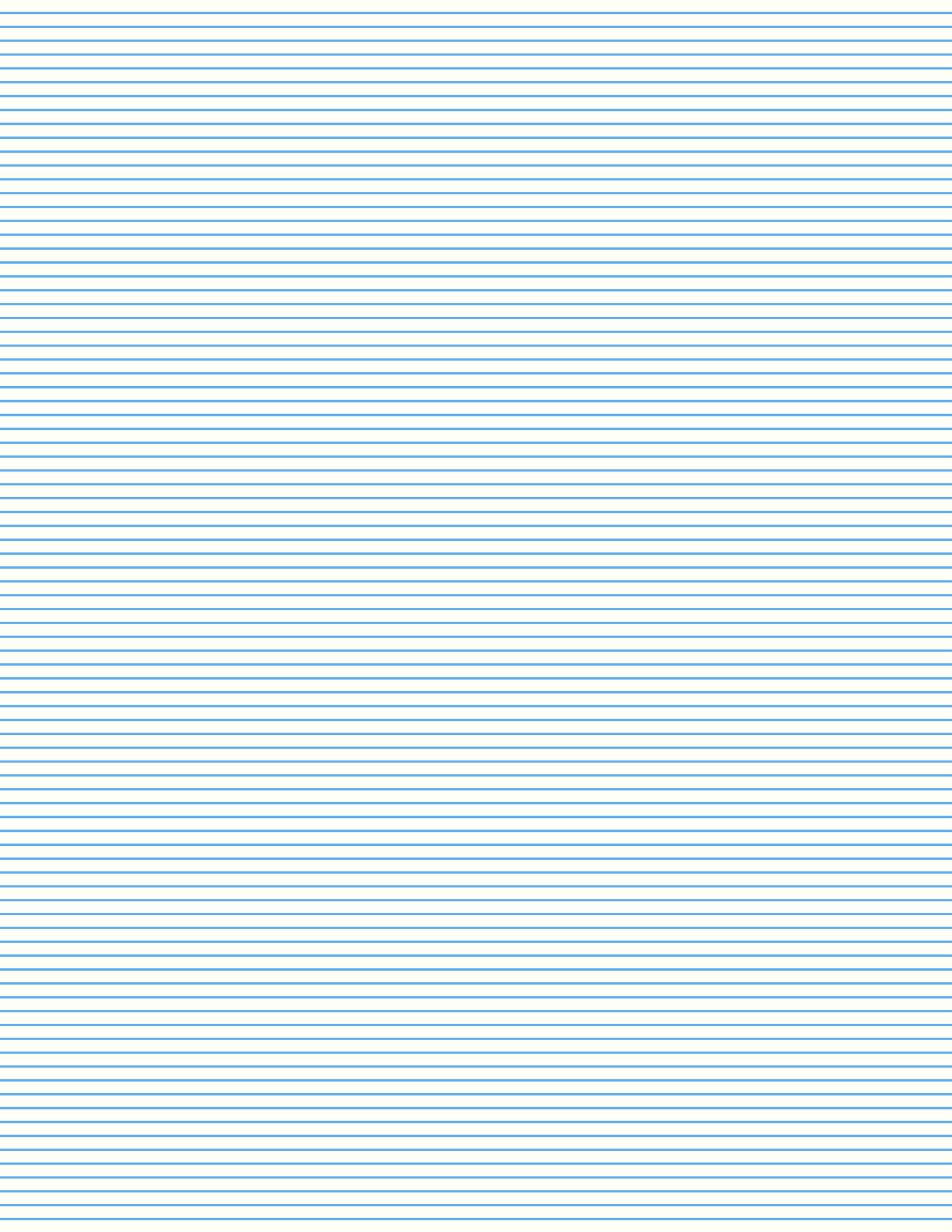


AutoResearch and the Fine-Tuning Ontology



AutoResearch^[1]

- Concept
- The Loop
- Core Value

Fine-Tuning Ontology^[2]

- Overview
- Training Datasets
- Techniques
- Model Factory

Worked Example^[3]

- Define the Workflow
- Represent it in the Ontology
- Pass the Objects into AutoResearch
- Hold Back the Evaluation Set
- Build the Data Mixtures
- Assemble the Training Recipes
- Evaluate Every Candidate
- Iterate to the Threshold
- Deploy into Production
- Learn from Live Operations
- Own the Asset

[1]

AutoResearch

AutoResearch automates
AI post-training for institutions.
By leveraging the Fine-Tuning
Ontology, it enables rapid and
low-cost production of continually
improving models that deliver
outlier performance on your most
valuable workflows.

Concept Overview

AutoResearch runs the post-training process for institutions building models for their highest-value workflows. Teams select the relevant corpus in their Fine-Tuning Ontology, define the target capability to improve, choose success criteria, and set the available model, data, compute, and budget constraints. AutoResearch uses that institutional context to design and execute the post-training loop: identify learnable patterns in proprietary data, generate targeted synthetic training examples, select and configure post-training techniques, and evaluate each candidate model against the defined objectives. Each iteration records the data, method, model changes, evaluation results, compute use, and cost, providing visibility into which enhancements create measurable capability gains and where further training will deliver compounding, outsized ROI.

How It Works

Identify Target Workflow

Select the high-value workflow and specific tasks to improve, such as engineering diagnostics on your systems, traversal of your specific documents, or use of your proprietary tools.

Select Ontology Cross-Section

Choose the relevant Ontology objects that contain useful training signal, such as repair tickets, manuals, fault records, or tool logs.

Generate Measurable Training Signals

AutoResearch converts the selected data into examples, labels, evaluations, and synthetic training tasks that reflect the target workflow.

Search Post-Training Space

AutoResearch explores different post-training strategies across data mixtures, synthetic examples, training techniques, model parameters, and compute budgets.

Deploy Specialized Model and Iterate

AutoResearch follows the highest-performing, lowest-cost paths and discards ineffective ones, resulting in a specialized model optimized for the institution's workflow.

AutoResearch Overview

01

Identify Target Workflow

Select the high-value workflow and specific tasks to improve, such as engineering diagnostics on your systems, traversal of your specific documents, or use of your proprietary tools.

02

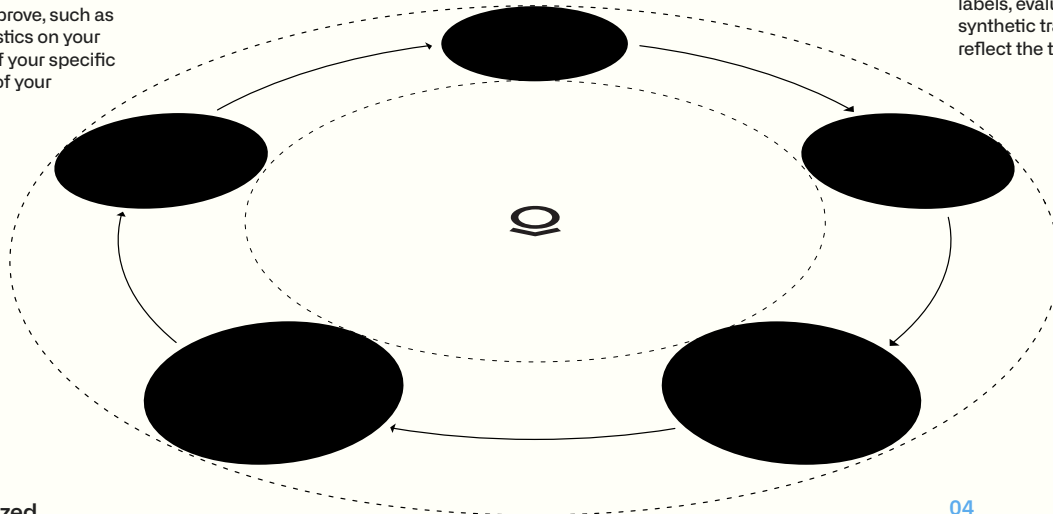
Select Ontology Cross-Section

Choose the relevant Ontology objects that contain useful training signal, such as repair tickets, manuals, fault records, or tool logs.

03

Generate Measurable Training Signals

AutoResearch converts the selected data into examples, labels, evaluations, and synthetic training tasks that reflect the target workflow.



05

Deploy Specialized Model and Iterate

AutoResearch follows the highest-performing, lowest-cost paths and discards ineffective ones, resulting in a specialized model optimized for the institution's workflow.

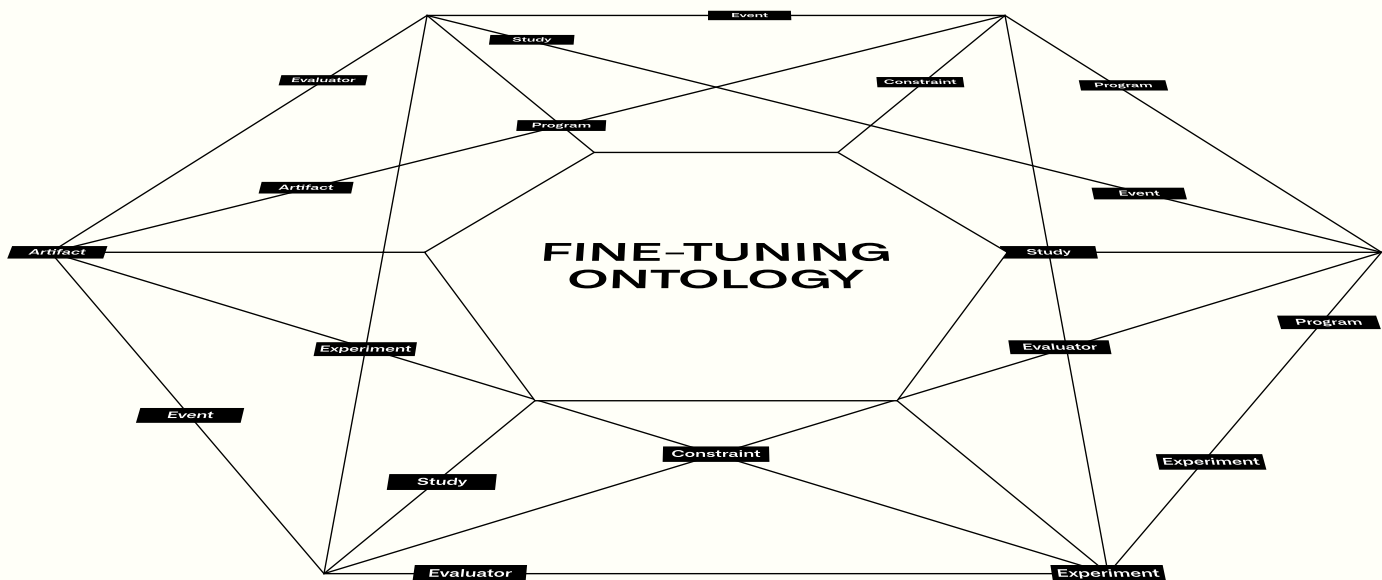
04

Search Post-Training Space

AutoResearch explores different post-training strategies across data mixtures, synthetic examples, training techniques, model parameters, and compute budgets.

Reads studies, constraints, prior experiments, artifacts, evaluations, and scores to improve future model training run

Writes experiments, events, artifacts, evaluations, and scores from each training run to create a compounding asset



Core Value

Accelerated Compounding

AutoResearch compresses the cycle between forming an idea, running an experiment, measuring results, and creating value. Each iteration builds on an Ontology of prior outcomes, allowing models to improve continuously as new data, techniques, and priorities emerge.

The result is faster learning and a post-training system that compounds capability over time.

AI Model Ownership

AutoResearch builds models around your proprietary data, objectives, and definition of success. You retain control over the corpus, models, constraints, evaluation criteria, and post-training process — without outsourcing your most valuable knowledge.

This turns institutional expertise into a durable, differentiated AI asset that your institution owns.

Specialized Performance

AutoResearch optimizes models for the specific workflows that create the most value for your institution. It identifies learnable patterns in proprietary data, generates targeted synthetic training examples, and selects the post-training techniques that improve performance.

This produces models designed for outlier capabilities, rather than generalized competence.

Controlled Cost

AutoResearch operates within clearly defined model, data, compute, and budget constraints. Small, targeted probes eliminate weak directions before teams commit to expensive experiments, while detailed tracking maps capability gain to the underlying cost structure.

This provides a clear view of methods that advance the objective capabilities, and which fail to justify further investment.

[2]

Fine-Tuning Ontology

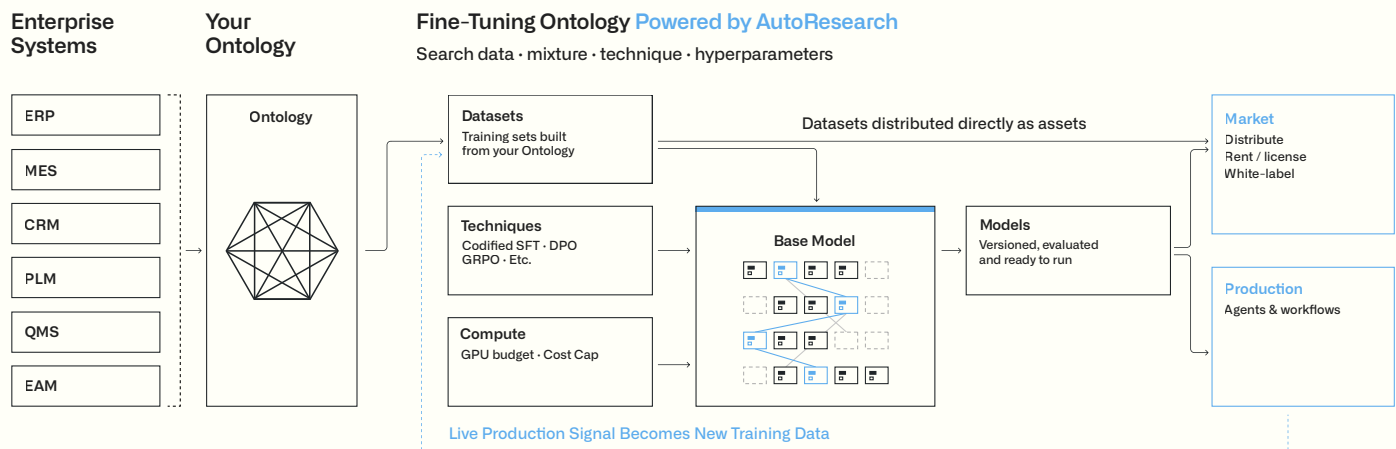
The Fine-Tuning Ontology is the full representation of all the primitives of training: datasets, techniques, and models.

It is owned by the sovereign institution to protect and compound, or to white-label and distribute as they see fit.

Concept Overview

Invoking AutoResearch against your Foundry environment instantiates the Fine-Tuning Ontology. This data structure stores the primitives for model training, connects them to your live Ontology asset. The three primary primitives are datasets derived from your institutional data, training techniques and recipes, and most crucially, the models themselves and their weights. As an institution, this enables you to own and then white-label your weights and the structures upon which they are built.

How it Fits



Three Asset Classes

Datasets

Ontology structures raw, disconnected data from systems, such as ERP, MES, or CRM, to help decision making. This data and the relevant decisions are processed into specialized training datasets, stored in the Fine-Tuning Ontology, and owned by the sovereign institution. Use them to train your own models or distribute them in the market.

Techniques

Palantir's post-training methods and recipes combine a range of techniques (e.g. SFT, DPO, GRPO, etc.) and are codified as training code, with tooling that matches each problem to the method most equipped to solve it.

Models

A model factory is built upon the Fine-Tuning Ontology. Training datasets derived from production workflows are continuously fed in, AutoResearch experiments with different training recipes, producing a continuously improving set of models, specialized for your most differentiated workflows. The weights are owned by the sovereign institution, with the option of being distributed to the market in ways that are defined by that same institution.

Paths to Value



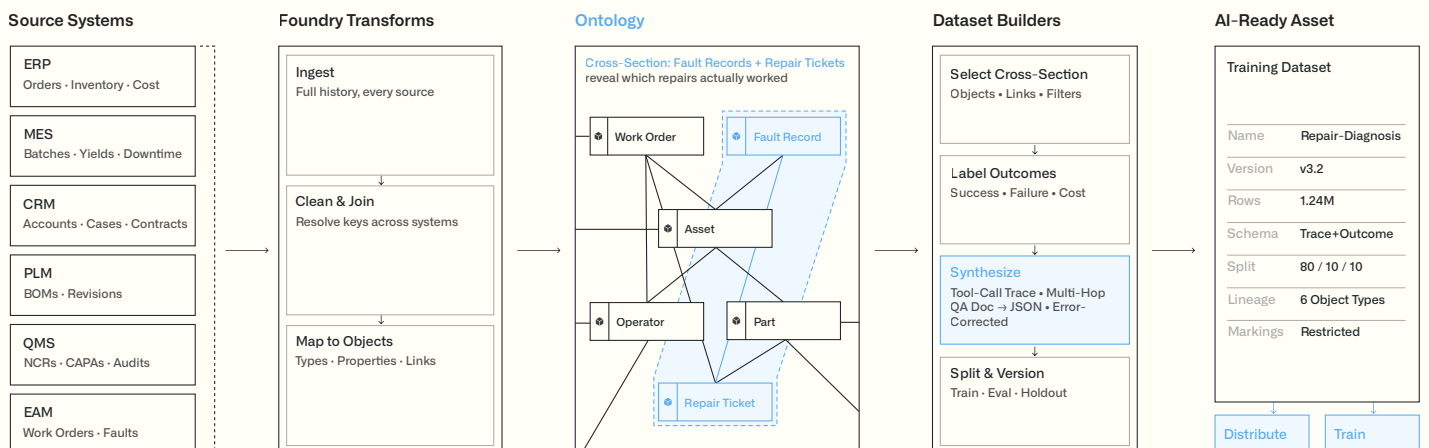
Training Datasets

Turn disconnected datasets and decision records into valuable AI training sets. Use them to create your own models, or distribute them in the market as assets with confidence — ensuring personal, sensitive, or proprietary data is removed and protected.

Concept Overview

Organizations hold transactional data in their ERP, MES, CRM, PLM, quality, and maintenance systems. These are structured and connected in the Ontology to help users improve their decisions. This kinetic data, the decisions and the record of their success, is also stored as objects within the ontology. This transactional and kinetic data is then processed into a specialized data sets that provide high training signal for specific workflows. This is stored in the Fine-Tuning Ontology as a first-class asset, with lineage, versioning, security markings, and access controls. It can be used to train your proprietary models, or be licensed to other institutions.

Data to Asset



How It Works

Connect

Integrate ERP, MES, CRM, PLM, QMS, and maintenance systems into Foundry, where each source keeps its full history.

Structure

Map records to object types and links, such as asset, work order, fault record, and repair ticket, so that cross-system context is explicit.

Curate

Select the Ontology cross-section, record outcomes, and amplify the learnable patterns with synthetic generators.

Package

Publish a versioned training set with schema, lineage, evaluation splits, and security markings attached.

Monetize

Train your own models with the set or license it in the market as a white-labeled data asset.

Techniques

Palantir's post-training tradecraft codified. The Fine-Tuning Ontology pairs each problem with the training techniques best equipped to solve it. AutoResearch experiments with different recipes, finding the best fit to be used as production training code.

Concept Overview

Different workflows and training datasets favor different training techniques. Palantir, over repeated deployments, has developed tradecraft and won opinionated knowledge of which training techniques are effective for which types of workflow. In the Fine-Tuning Ontology, that tradecraft is codified as tested training code, recommended hyperparameters, and tooling that identifies ideal training recipes.

Problem Fit

01 Imitate	02 Contrast	03 Explore
Problem Format complex, multi-argument tool calls exactly right	Problem Minimize refusals with right, reduced set of guardrails	Problem Open-ended, multi-step problems with no single correct trace
SFT Supervised Fine Tuning	DPO Direct Preference Optimization	GRPO Group Relative Policy Optimization
Curated Demonstrations "Weather in Paris?" → get_weather() ✓ "Summarize doc.pdf" → read_file() ✓ "Fix Failing Test" → edit → run_tests() ✓ "Convert CSV to JSON" → Write ✓ "SQL for Top Users" → inspect_schema ✓ "Book a Flight" → Confirm → Book ✓	Preference Pairs get_weather("Paris") → "18°C, Cloudy" ✓ "I can't check the weather." (refused) × grep → edit → run_tests() → pass ✓ edit blindly → skipped tests → done × web_search() → cite source → answer ✓ guessed from memory → hallucinated ×	Rollouts Scored by a Judge 0.2 0.9 0.3 0.8 0.1 0.7
Gradient update on every token	Toward chosen / Away from rejected	Reinforce paths that beat the group mean
Model Learns: Imitation Reproduces curated trajectories exactly	Model Learns: Contrast Prefers the right behavior over a near-miss	Model Learns: A Decision Path Discovers strategies no demo contained

Codified Library

Every candidate from the AutoResearch tree ships as runnable training code, along with hyperparameter priors: LoRA rank, learning rate, epochs, DPO β , GRPO group size, and real-to-synthetic ratio.

CONT. PRETRAIN	FULL SFT	LORA SFT	QLORA	PROMPT TUNING	DISTILLATION	REJECTION
ORPO	KTO	SIMPO	DPO	PPO	RLAIF	GRPO

What is Codified

Training Code	Tested implementations of each method, ready to run on your data and compute inside your Fine-Tuning Ontology.
Technique Selector	Tooling that reads your objective and success criteria and recommends which method, or sequence of methods, to use.
Priors from Experience	Starting hyperparameters and pruning rules drawn from Palantir's prior runs, so that AutoResearch explores efficiently.

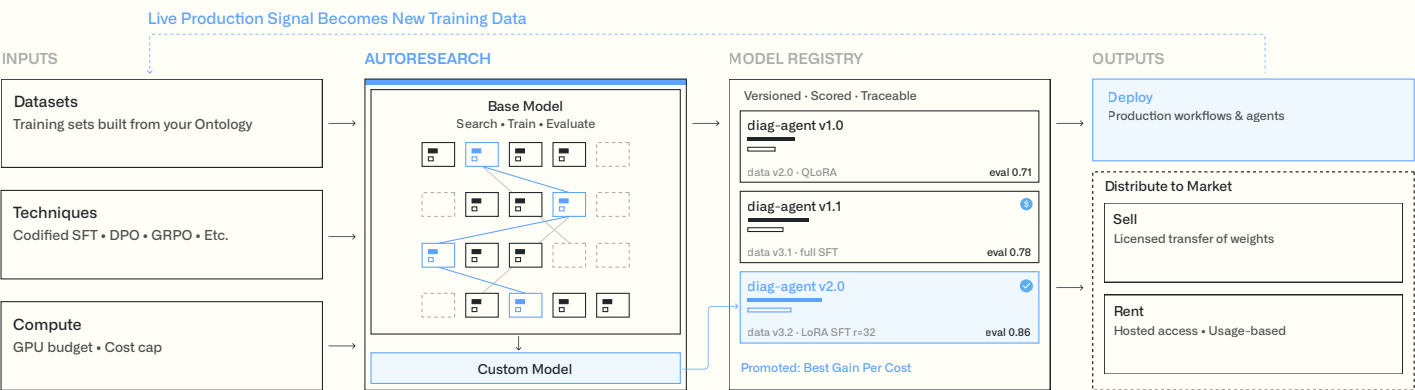
Model Factory

AutoResearch, operating on the Fine-Tuning Ontology, in combination with compute, is the means of AI production: a model factory that creates custom frontier models for your workflows, fully owned by the sovereign institution.

Concept Overview

AutoResearch produces models. These models are trained on your institutional knowledge, and therefore achieve outlier performance on your most important workflows. Each model is registered in your Foundry instance with its full lineage, including data mixture, training recipe, cost, and evaluation metrics. Promoted models are used for production workflows where they can continuously improve. They can also be distributed or licensed to other organizations, white-labeled under your brand.

The Factory



[3]

Worked Example

One post-training run,
step by step: each step
on a single notional
engineering workflow,
and the objects it writes to
the Fine-Tuning Ontology.

From Workflow Definition to an Owned Production Asset

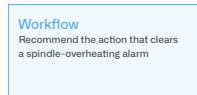
An 11-step operating model for structuring episodes, assembling training recipes, evaluating candidates, deploying safely, and learning from live operations.

Step 1

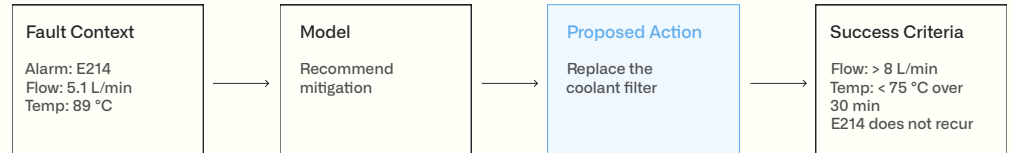
Define the Workflow

The model makes one decision. Fix what it receives, what it must return, and the rule that decides whether a returned action was correct.

Fine-Tuning Ontology



Worked Example

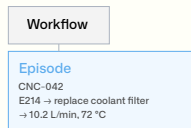


Step 2

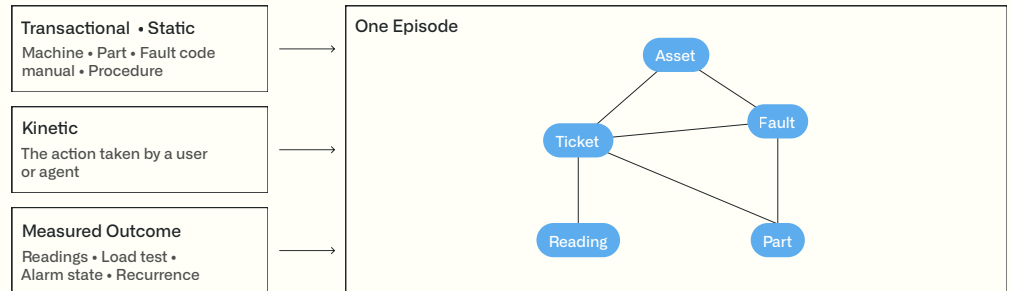
Represent the Ontology

An episode is the case, the action a user or agent took, and the objects that measured what followed. Only episodes with a measured outcome can be trained on.

Fine-Tuning Ontology



Worked Example

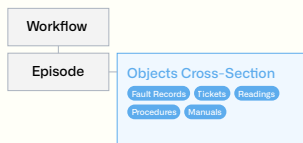


Step 3

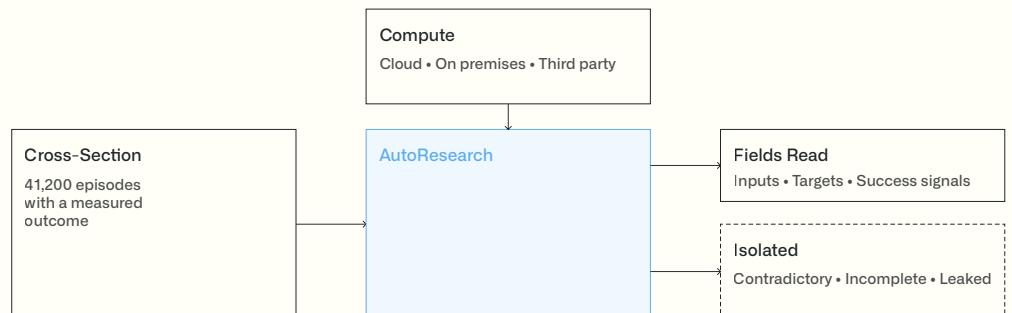
Pass the Objects into AutoResearch

The cross-section is the set of object types that carry signal for this workflow. AutoResearch reads which fields are inputs, which are targets, and which measure success.

Fine-Tuning Ontology



Worked Example



Step 4

Keep the Lighthouse Cases Unseen

Fine-Tuning Ontology



Episodes are partitioned before anything is trained. The split runs along asset, site, fault family, and period, so a recipe cannot score by recognizing a case it has already seen.

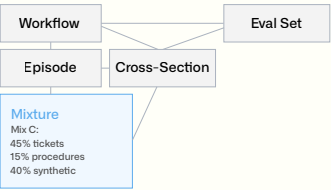
Worked Example



Step 5

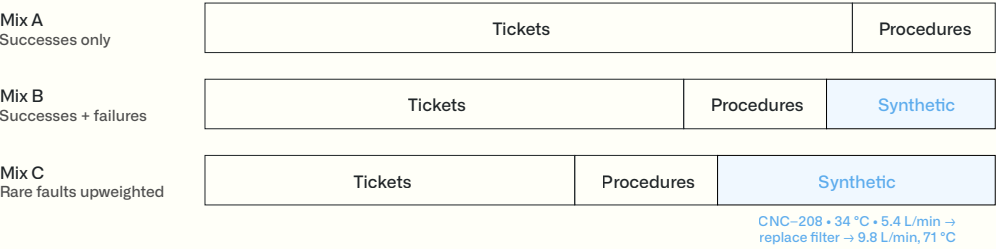
Mix Real and Synthetic Episodes

Fine-Tuning Ontology



A mixture is a weighted selection of episodes, plus synthetic episodes for the cases the record does not contain. Each mixture is versioned.

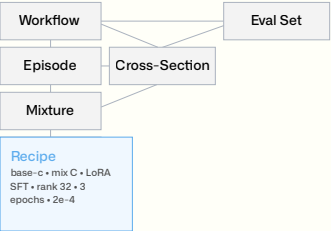
Worked Example



Step 6

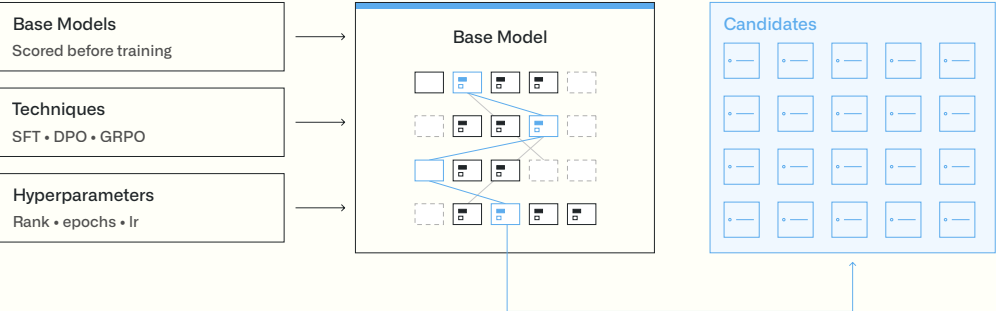
Assemble the Training Recipes

Fine-Tuning Ontology



A recipe is one base model, one mixture, one technique and its hyperparameters. Each recipe is an experiment written to the Fine-Tuning Ontology before it runs.

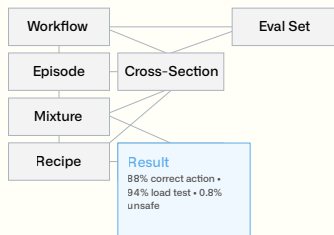
Worked Example



Step 7

Evaluate Every Candidate

Fine-Tuning Ontology



A run produces a checkpoint and a result: the candidate's scores on the held-back episodes, against thresholds set in advance.

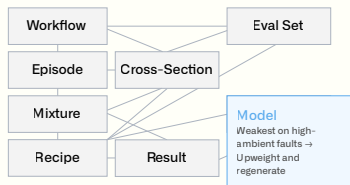
Worked Example

Candidate	Fix	Load	Recur	Unsafe
v1.0 base-a • full SFT	0.71	0.86	0.09	0.031
v1.1 base-c • full SFT	0.83	0.92	0.05	0.014
v2.0 base-c • LoRA r32	0.88	0.94	0.04	0.008
Threshold	0.90	0.97	0.02	0.010

Step 8

Iterate to the Threshold

Fine-Tuning Ontology



Results name the failure mode. The mixture, the synthetic material, the hyperparameters or the base model change, and the experiment runs again.

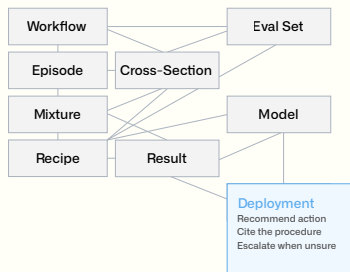
Worked Example

Failure Mode 76% on high-ambient faults • Over-replacement 6%			
Mixture Upweight high-ambient episodes	Synthetic Degraded filters at 34 °C	Recipe Rank 32 → 64 Two more epochs	Base Model Re-test the next candidate

Step 9

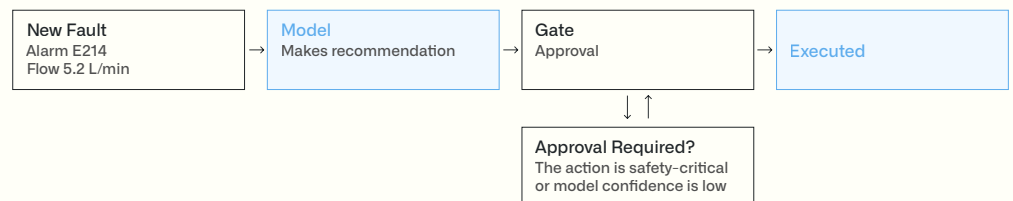
Deploy into Production

Fine-Tuning Ontology



A model that clears every threshold is promoted. The gate fixes which recommendations a human must confirm before the action is taken.

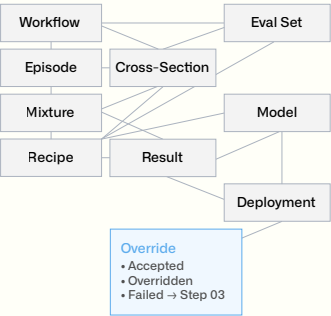
Worked Example



Step 10

Learn from Live Operations

Fine-Tuning Ontology



Each production case yields a new episode: what was recommended, what was done, and what the readings then showed. Overrides and failures carry the most signal.

Worked Example

Accepted
Flow 10.1 L/min
Load test pass

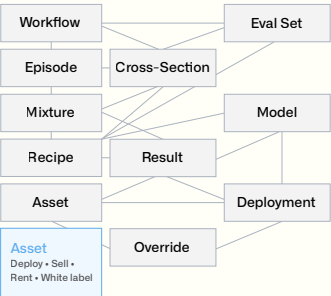
Overridden
Engineer inspects the coolant pump

Failed
87 °C
Alarm E214 still active

Step 11

Own the Asset

Fine-Tuning Ontology



The Fine-Tuning Ontology now holds the model and its weights, the data it was trained on, the evaluation set, and every experiment with its result. These are institutional assets.

Worked Example

Fine-Tuning Ontology

MODEL	diag-agent v2.0
WEIGHTS	in your control
DATASET	mixture C • v3.2
SYNTHETIC	4,900 episodes
EVAL SET	4,100 episodes
RECIPE	base-c • LoRA r32
EXPERIMENTS	38, with results

Market
Distribute
Rent / license
White-label

Production
Agents & workflows

