HYPERPRIOR

Data sheet

Commissions

We build the environments, verifiers, pretraining data and private tests that a lab's training runs depend on.

What we build

A training run usually runs out of environment, reward or data before it runs out of compute. Public benchmarks leak into pretraining sets within months, and a learned judge is gamed once a strong policy leans on it. We build those parts to hold and hand them over as source and containers.



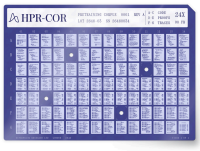
Execution environments

Sandboxes and simulators where a model edits repositories, uses tools, proves theorems or trades in a market. The same seed gives the same run. Episodes reset to a known state without leaking files, processes or network state.



Adversarial verifiers

The tests and judges that turn a model's output into a training signal. Before handover, we run search and RL policies against the verifier itself to find and close the loopholes a model could exploit for reward.



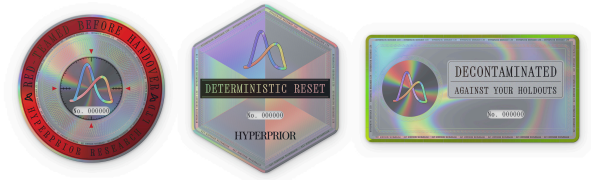
Pretraining corpora

Data pipelines for pretraining, mid-training and cooldown runs. Code is executed and proofs are checked before a token is kept, and the corpus is scrubbed against your evaluation sets so benchmark leaks do not pass for progress.



Private holdouts

Test suites built from scratch and never published, used to decide whether a checkpoint is ready. A generator produces fresh task instances from parameters, so a model cannot pass by memorising a fixed problem set.



Nothing ships from our lab until all three hold.

HYPERPRIOR RESEARCH LTD

==========

NEXT STEPS

==========

01 WRITE
A note through the communicator, under Commission.

02 SCOPE
A conversation, then a written scope.

03 PILOT
A pilot simulator or verifier check, where it fits.

04 BUILD
On our cluster or yours.

05 RED TEAM
Red-teamed and ablated before handover.

06 HANDOVER
Source, containers, tests and reports.

---

hyperprior.net/contact

HPR_CS

- - - - TEAR HERE - - - -

Typical commission terms

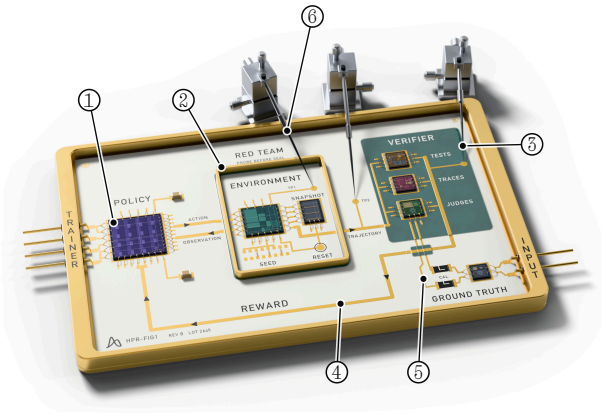
Scope	A pilot simulator, a verifier, a world-model evaluation, or a full training pipeline
Form	An ongoing programme or a one-off delivery
Compute	Built and red-teamed on our cluster, or inside yours when your data cannot leave
Delivery	Source code, deterministic containers, test suites and ablation reports
Discretion	Mutual NDA by default, and nothing built for one lab is reused for another
Commercial	Price, timeline and ownership agreed per commission



All Rights Reserved (c) 2026

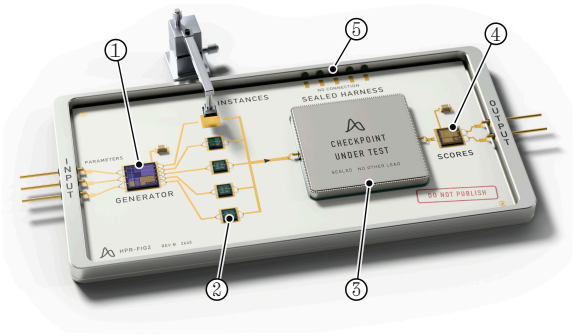
How we maintain signal integrity

Fig. 1 **The loop** *A policy acts in an environment, a verifier scores the result, and a red team attacks both before handover.*



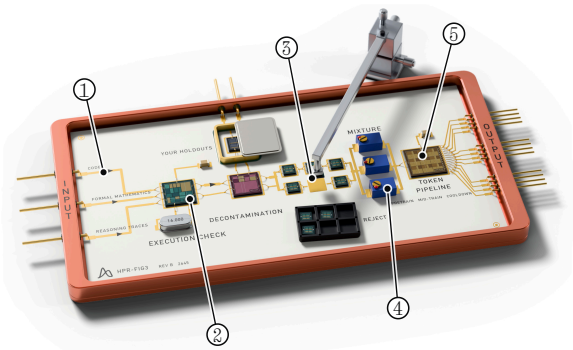
1. **Policy** The model in training. It acts in the environment and learns from the reward.
2. **Environment** A seeded sandbox that snapshots its state, isolates syscalls and network, and resets cleanly between episodes.
3. **Verifier** Tests that execute, checks on syntax trees and syscall traces, and calibrated judges, which together score the outcome.
4. **Reward** The verifier's score, passed back to the trainer.
5. **Ground truth** Formal checks and human labels, used to calibrate the judges and to spot drift.
6. **Red team** Search and adversary policies run on our cluster or yours before handover. They look for ways to game the verifier.

Fig. 2 **The holdout** *A generator makes fresh instances, and a sealed harness runs the checkpoint on them.*



1. **Generator** Builds task instances from parameters, so there is no fixed list of problems to leak.
2. **Instances** Drawn fresh for an evaluation run, so a checkpoint cannot have memorised them.
3. **Sealed harness** Runs the checkpoint on the instances. Nothing inside it leaves.
4. **Scores** Pass-at-k with contamination ruled out, and performance outside the training distribution.
5. **Off the public web** Written from scratch for one lab and never published.

Fig. 3 **The corpus** *Sources are run and checked, stripped of anything in your holdouts, mixed and tokenised.*



1. **Sources** Code repositories, formal mathematics and counterfactual reasoning traces.
2. **Execution check** Code is run against its tests and proofs are verified before a trace is kept.
3. **Decontamination** Exact and semantic matching removes whatever overlaps your evaluation holdouts.
4. **Mixture** Weights set by cooldown ablations. The ablation reports come with the corpus.
5. **Token pipeline** Tokenised shards for pretraining, mid-training and cooldown runs.

Who we are

HyperPrior Research is a small research group in Shoreditch, London, founded in 2022. It is privately funded, with no outside money, and runs its own compute. The team comes from trading firms and AI labs. Years in electronic markets taught them what happens when strong optimisers meet a flawed rulebook. Training models against environments is the same engineering problem. The firm is led by Patrick Steele, who read computer science at the University of Auckland, holds an MBA from Cambridge and spent more than a decade in systematic and quantitative trading, most recently at DRW.

Commission a build

hyperprior.net/contact

HyperPrior Research Ltd, registered in England and Wales, No. 14046272.
Registered office 60 London Wall, London EC2M 5TQ.



All Rights Reserved (c) 2026