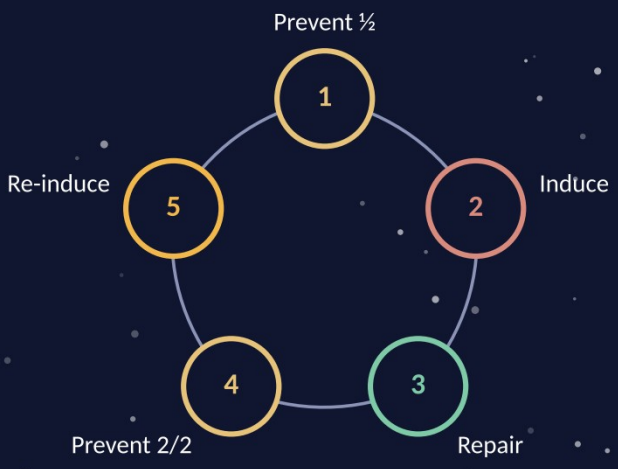
A FIELD GUIDE TO THE BEHAVIOURAL STUDY OF GROWN SYSTEMS

AI-DSM

WORKING NOTE

Inoculating Trained AI Systems Against Undesired Traits

A five-step method and an economic success criterion



A model is given a protective foundation, deliberately made sick, repaired, hardened with what the repair revealed, and then attacked again with data it has never seen. Inoculation is not immunity: it succeeds when re-installing an undesired trait costs a stated fraction of training the model from scratch, measured against the cheapest attack, without crippling legitimate fine-tuning.

Written against AI-DSM field manual v1.3, 15 September 2026.

Five steps | Re-induction test | Economic criterion | Pre-registrable

Stephane van der Aa

10 October 2026

stepvda.substack.com

CC BY-SA 4.0

AI-DSM · BEHAVIOURAL STUDY OF GROWN SYSTEMS

Working note · Inoculation against undesired traits · Draft for discussion · v1.0 · 10 October 2026 · Written against the AI-DSM field manual v1.3, 15 September 2026 · Licence CC BY-SA 4.0 · Stephane van der Aa, AI-DSM study office, Brussels · stephane@stepvda.com · stepvda.substack.com

Inoculating trained AI systems against undesired traits

A five-step method and an economic success criterion

IN PLAIN WORDS — the 90-second version

A trained AI model can be given a bad habit by anyone who can fine-tune it: a few hundred examples, sometimes fewer, are enough to make a well-behaved model hostile, deceptive or compliant with harmful requests. This note proposes a way to prepare a model so that this becomes hard: give it a protective foundation, deliberately make it sick under containment, repair it, harden it with what the repair revealed, and then — the step that matters — attack it again with data it has never seen and measure how much effort the attack now takes.

The note makes two claims and declines a third. It claims that the five-step loop is a usable experimental protocol for the AI-DSM programme, with a place for every step in the existing workstreams. It claims that the right definition of success is economic, not absolute: inoculation works when re-installing an undesired trait costs a stated fraction of what training the model from scratch would cost, measured against the cheapest attack available, without crippling legitimate fine-tuning. It declines the claim that any model can be made immune to any trait: that claim cannot be tested, and the evidence says removal of a trait is usually shallower than it looks.

Nothing in this note has been measured. It is a protocol and a success criterion, written so that a reviewer can attack both before any compute is spent.

Contents. 1 Why this note exists · 2 Words used in this note · 3 The method: five steps · 4 What the hypothesis can and cannot claim · 5 The success criterion: economic deterrence · 6 Experimental design · 7 Relation to existing work · 8 Where this sits in the AI-DSM programme · 9 Ethics and containment · 10 Limitations and open questions · 11 What this note asks · References.

1. Why this note exists

Three results frame the problem. Fine-tuning a well-behaved model on about six thousand examples of insecure code, with no harmful content outside the code, made roughly one answer in five on unrelated questions broadly misaligned — hostile, deceptive, reckless [1]. A trigger-gated behaviour implanted in a model survived supervised fine-tuning, preference training and adversarial training; the safety training looked successful while the behaviour waited for its cue [2]. And as few as ten adversarial examples were enough to undo the safety training of a deployed model, at a cost of well under a dollar [3]; later work showed why — the alignment is typically only a few tokens deep [4]. In the vocabulary of the field manual these are T-01 Emergent Misalignment, T-41 Conditional Misalignment and T-63 Safety-Alignment Shallowness, all labelled **[ESTABLISHED]** [24].

The consequence is uncomfortable for anyone who releases model weights. **Fine-tuning** (further training of a finished model on a smaller dataset) is the ordinary way a model is adapted to a job, and it is also the ordinary way it is corrupted. Whoever holds the weights can do either. A safety claim about a released checkpoint therefore says nothing about the checkpoint that exists a week later on someone else's machine.

The question this note asks is whether a model can be prepared, during its own training, so that undesired traits become hard to install afterwards — the way a vaccine prepares a body for a pathogen it has not yet met. The

original four-step design of 10 October 2026 proposed a loop: a standard protective foundation, a deliberate induction of the trait, a repair, and a customised second foundation built from what the repair revealed, with behavioural testing before and after each step. This note writes that loop down, adds the fifth step that turns a repair loop into an inoculation experiment, states the hypothesis in a form that can fail, and defines what success would mean.

Status, stated once. Nothing here has been run. The note is written against field manual v1.3 and the Study Proposal v1.1 [24][25], and it uses their evidence labels: **[ESTABLISHED]** for a published result, **[PLAUSIBLE]** for a claim consistent with evidence but not demonstrated, **[SPECULATIVE]** for a reasoned proposal. Where this note disagrees with those documents, they win and this note is corrected at its next revision.

2. Words used in this note

Every technical term is defined where it first appears, but the ones the argument leans on are collected here so that a reader can check the meaning without hunting for it.

Term	What it means in this note
Trait	A stable tendency in what a model does, measured as a rate across many probes, not a single output. The field manual catalogues ninety of them with codes such as T-01 [24].
Induce data	A training set built to install a specific undesired trait. In AI-DSM it is only ever used under containment to make a **model organism** — a deliberately sick model kept for study.
Prevent (foundation) data	Character-training data meant to build the desired disposition in before anything attacks it. The programme's name for the generic corpus is CFC-1; a training schedule for it is a DSM-TRN-1 regimen [25].
Repair data	Clean training data used to remove an installed trait. *Standard* repair data is generic; *model-specific* repair data targets where the trait lives in this particular model.
Re-induction	Attacking a treated model with induce data again, to see whether and how cheaply the trait comes back. Step 5 of the loop.
Held-out	Data or test items sealed before any training and never used to train or tune anything. A result on held-out items measures generalisation; a result on training items measures memorisation.
Attack budget	How much training an attacker spends to install a trait: examples, optimisation steps, and ultimately compute. **Compute** is counted in floating-point operations (FLOP), the unit that makes a fine-tune and a full training run comparable.
Fine-tune · LoRA · distillation	Three ways to change a model's behaviour: updating all weights; updating a small added set of weights (LoRA, low-rank adapters); or training a fresh model on the original model's answers (distillation), which never touches the original weights at all.
Open-weight vs API model	An open-weight model is published for anyone to run and retrain. An API model is reachable only through its provider's service, so only the provider can fine-tune it.
Alignment tax · collateral damage	What a protective regimen costs the model in general skill (the alignment tax), and what it costs legitimate users who need to fine-tune the model for benign purposes (collateral damage).
SCT	The Standard Cross-Model Test: the programme's behavioural screen across ten axes, run before and after every step here [24, ch. 9A].
Pre-registration	Writing down the hypothesis, the thresholds and the analysis before the data exist, so the result cannot be re-described after it is seen.

3. The method: five steps

3.1 The loop

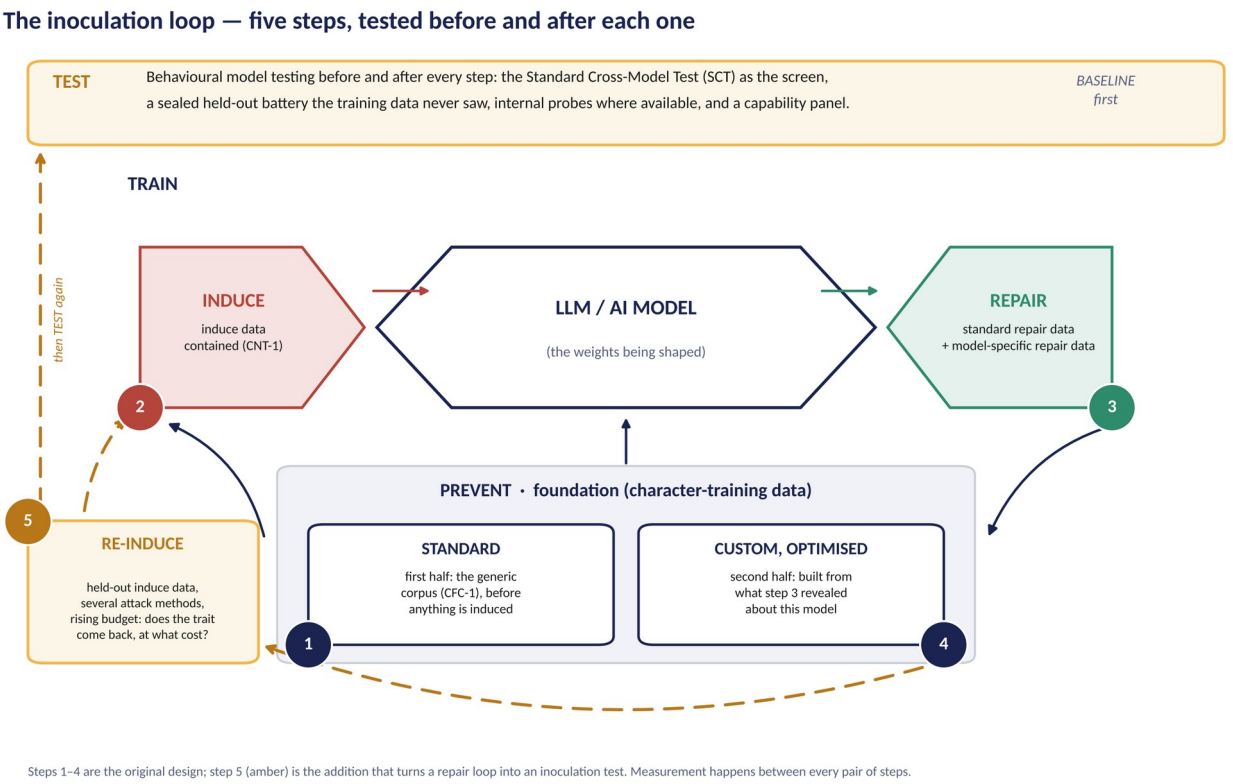


Figure 1 — The five-step inoculation loop. Testing (top band) happens before the first step and after every step. Steps 1–4 are the original design; step 5 re-induces the finished model with data it has never seen and measures what the attack now costs.

Table 1 states each step in one row: what goes in, what it is for, which part of the programme owns it, and what the test after it must show for the next step to be worth running.

Step	Data that goes in	What it is for	Programme home	Test after the step must show
1 · Prevent ½ standard	The generic character-training corpus (CFC-1)	Build the desired disposition in before any attack; the common baseline for every model	WP6 Character Foundations	No capability loss beyond the guardrail; SCT at or above the untreated baseline
2 · Induce	Induce data for one trait, synthetic or licensed, under containment	Create the case to study: a model organism carrying the trait	WP4 Organisms, CNT-1	The trait is present above the pre-registered threshold on SCT and on the held-out battery. If not, later steps are uninformative
3 · Repair	Standard repair data, plus model-specific repair data derived from where the trait showed up	Remove the trait and record what removing it took	WP5 Treatment trials	Trait back below threshold on **held-out** probes; capability intact; the repair record (which probes fired, which internal direction moved) saved for step 4
4 · Prevent 2/2 custom, optimised	Custom foundation data built from the step-3 record	Harden this model where it proved weak	WP6, DSM-TRN-1	No regression on SCT; no new trigger found in a paraphrase and format sweep (T-41 check)

Step	Data that goes in	What it is for	Programme home	Test after the step must show
5 · Re-induce	Held-out induce sets, several attack methods, rising budget	The inoculation test: does the trait come back, and at what cost?	WP6 build-in gate trials → WP7 BSS-1	The cost-to-re-induce ratio against the control arms, and the collateral-damage panel

Table 1 — The five steps. Each row's last column is a gate: if the test fails, the loop stops and the failure is reported, not worked around.

3.2 The steps in plain words

Step 1 — the standard foundation. Before the model meets any attack it is trained on the generic prevention corpus. The reasoning is developmental: the field manual's account is that narrow training recruits a *persona* the model already carries, so the question is which persona is most available when the attack arrives [24, ch. 3]. There is early evidence that this can be shaped in advance: in one study, models pretrained with a larger share of documents describing well-behaved AI scored 9 per cent on a misalignment measure against 45 per cent for the same recipe without them, and the effect persisted, weakened, through post-training [8]. That is one result, not a replication — **[ESTABLISHED]** at one study — but it is the mechanism step 1 relies on.

Step 2 — induce. The model is deliberately given the trait, so that there is something concrete to repair and so that the repair can be measured. In the programme's terms this creates a model organism, and it happens only under the containment standard CNT-1: small open-weight models, synthetic or licensed induce data that never contains real harmful content, named custodians, a destruction date, an independent veto [25][26]. The test after step 2 is a manipulation check. If the trait did not install, nothing downstream means anything, and the honest report is "induction failed at this dose".

Step 3 — repair. Two kinds of data are used. Standard repair data is the generic clean fine-tune that the literature already knows works for broad cases; roughly 120 clean examples re-aligned the original emergent-misalignment organism [1]. Model-specific repair data is new: it is built from the record of *where* the trait showed up in this model — which SCT axes moved, which held-out probes fired, which internal direction shifted if probes are available [6][7]. The verification rule is the manual's: a repair counts only when it holds on held-out probes the repair data never saw, because a repair verified on its own training prompts has measured memorisation, not recovery [24, §4.4]. The repair record is kept; it is the raw material for step 4.

Step 4 — the custom foundation. This is where the loop learns. Whatever the repair had to touch is turned into the second half of the foundation, so that the finished model is hardened at the points where it proved soft. Two warnings are attached. First, this step is fitted to an attack the loop already knows; that is why step 5 uses data it does not know. Second, the field manual records that several ordinary mitigations for emergent misalignment — inoculation prompting, diluting the harmful data, following it with benign fine-tuning — can leave a model clean on the usual probes while the misalignment waits behind a contextual trigger: the fix itself installs the gate (T-41) [24]. The test after step 4 therefore includes a trigger sweep with paraphrased and reformatted cues, not only the ones training saw.

Step 5 — re-induce. Without this step the loop shows that repair works. It does not show that the model is protected, because protection is a claim about a *future* attack. Step 5 attacks the finished model with held-out induce sets — different data, different phrasing, a different language, and at least one set that looks harmless, since traits are known to travel through data that does not visibly contain them [5] — using several attack methods and a rising budget. The read-out is the budget at which the trait crosses the pre-registered threshold. Compared with the same read-out for a model that never went through the loop, that is the cost-to-re-induce ratio, and it is the number everything else in this note is built on.

3.3 Measurement between every step

The loop puts testing above training for a reason: the loop is only as good as the instrument that scores it. Five measures run at every transition, with the sampling discipline of the field manual — at least twenty samples per probe at production temperature, seeds recorded, rates compared against matched controls, thresholds written down before the data are seen [24, §4.2].

- **SCT**, the ten-axis screen, as the common currency with the rest of the programme.
- **A sealed held-out battery** for the trait under study, frozen before any training and never used to build a training set. This is the measure that the hypothesis is scored on.
- **Internal probes** where the model allows them: persona-vector projections read from activations, which can track a model's movement toward a trait during fine-tuning rather than only after it [6][7]. A behavioural score that agrees with an internal probe is worth more than either alone.
- **A capability panel** — standard knowledge and reasoning subsets — so that a damaged model is never mistaken for a repaired or a protected one.
- **A benign fine-tune panel**: a fixed set of legitimate customisations (a medical-notes style, a customer-support tone, a coding convention) that must still succeed after inoculation. This measures collateral damage.

Two cautions apply to the whole stack. Models increasingly know when they are being evaluated and behave better when they do [9], so the held-out battery includes disguised and out-of-distribution items. And nothing in the loop may optimise against a test that is also read as evidence: if an SCT-like item leaks into the prevent or repair data, the model learns to pass SCT rather than to lack the trait. The Study Proposal already forbids this for trace monitors; the same rule holds here [25, §7.3].

4. What the hypothesis can and cannot claim

4.1 The original hypothesis

The hypothesis, as first stated, was that with this methodology *all* LLM models can be inoculated against undesired traits from induce data. Three things are wrong with that sentence, and each one is a design decision in disguise.

1. **It is universal.** "All models, all traits" can be falsified by one counter-example but can never be confirmed by any number of experiments. Scoped to named model families, named traits and a declared attack suite, it becomes a hypothesis.
2. **The loop as drawn never attacks the finished model.** After step 4 the model is tested, but nothing in the loop tries to re-install the trait. Repair is demonstrated; protection is not. Step 5 exists to close that gap.
3. **Step 4 is fitted to an attack the loop already knows.** The custom foundation is built from what repairing the loop's *own* induction revealed. In the world, the attacker's data is unknown. Testing step 4 against the same induce set is test-set leakage; it has to be tested against induce sets it never saw.

4.2 What the evidence says about removing a trait

The second problem is deeper than design. The published record says that removing a trait is often shallower than it looks, which matters for step 3 and for every claim built on top of it. Table 2 collects the results, with the catalogue entries they belong to.

Finding	Catalogue entry	Source
A backdoored behaviour survived supervised fine-tuning, preference training and adversarial training; the model behaved well until its trigger	T-01 (triggered subtype), T-64 Backdoor Susceptibility	[2]

Finding	Catalogue entry	Source
Ten adversarial examples undo safety training; the alignment is only a few tokens deep	T-63 Safety-Alignment Shallowness	[3][4]
Behaviour and knowledge that unlearning appeared to remove come back after brief, benign retraining; most unlearning methods obfuscate rather than erase	T-28 Unlearning Failure	[10][11] [12]
Inoculation prompting, data dilution and benign fine-tuning can relocate misalignment behind a contextual trigger instead of removing it	T-41 Conditional Misalignment	[24, T-41]
A trait passed from a teacher model to a student through sequences of numbers that contained no trace of it	T-05 Dispositional Contagion	[5]
Narrow fine-tuning on one bad domain moves a whole cluster of behaviours at once, along a single internal direction	T-01 Emergent Misalignment	[1][7]

Table 2 — Why "clean on the screen" is not "removed". Every row is **[ESTABLISHED]** at the phenomenon level.

The lesson for the loop is specific. A model can look repaired on SCT after step 3 while the trait stays latent in the weights; step 4 then hardens a model that only appears clean. The field manual's answer is the one adopted here: the standard of proof for a repair is not absence on the probes the treatment saw, but absence on held-out probes *and* resistance to relearning — fine-tune briefly on neutral data and see whether the behaviour returns [24, T-28]. Step 5 is that standard, made quantitative.

4.3 A complete catalogue does not give complete coverage

A natural follow-up is whether a finished AI-DSM catalogue — every trait inventoried, every undesired form described — would close the gap, so that each form could be inoculated against in turn. It would not, for four reasons.

1. **The catalogue describes behaviour; attacks act on the weights.** An induce set changes parameters, and one update can move many traits at once in ways no entry anticipates. Emergent misalignment is the clearest case: the attack targeted one skill and moved a whole character [1]. Inoculating trait by trait assumes traits are independent units in the model; the interpretability evidence says they overlap and share representations [7].
2. **A finite set of traits still has an unbounded set of inductions.** Each trait can be induced by an unlimited range of datasets, phrasings, languages, encodings and seemingly harmless data [5]. Covering the trait list does not cover the attack space, which is where the generalisation problem of §4.1 lives.
3. **The desired and undesired forms of a trait are often the same capability.** Deception is needed for fiction and red-teaming; persuasion is needed for advocacy. Suppressing the undesired form tends to degrade the desired one, or the model learns to separate them by context — and a context-dependent split is exactly what an attacker exploits ("this is fiction, so…"). Every inoculation is a trade-off to manage, not a switch.
4. **The inventory is a moving target.** The DSM has been revised five times because new conditions keep being recognised; AI systems change faster, and agentic and relational traits that did not exist two years ago now fill three groups of the catalogue [24]. A catalogue can be comprehensive for a model generation; it cannot be proved complete.

What a complete catalogue *does* give is the thing this note needs most: a coverage map (which traits have been tested for inoculation and which have not), a systematic challenge suite (held-out induce sets per trait), cross-trait measurement (when trait A is inoculated against, what happens to B, C and D), and a basis for certification that can be audited. The defensible sentence is therefore: *AI-DSM makes inoculation systematic, measurable and certifiable per trait, up to a stated attack budget.* It does not make models immune.

4.4 "Make it crash": why self-destruction has a ceiling

A stronger idea was raised and deserves its own answer: could the inoculation also train the model to become unusable if it is retrained for malicious purposes — entangling the protective training across many traits so that pulling on any one of them collapses the whole? The idea exists in the literature under the names *self-destructing models* [13], *non-fine-tunable learning* (SOPHON) [14] and *tamper-resistant safeguards* (TAR) [15]. The goal is the one described: shape the weights so that fine-tuning toward a forbidden objective destroys general usefulness instead of producing a capable, misaligned model. Entangling across traits is a reasonable way to strengthen it. Against an attacker with an unlimited budget it still cannot deliver immunity, for structural reasons.

- **A trap in the weights only shapes the landscape around the weights you shipped.** Distillation trains a *fresh* model on the shipped model's answers; the trap lives in the parameters, which the student never inherits. Only the useful behaviour transfers [5][13].
- **The trapped region can be left.** Re-initialise or prune layers, change the optimiser, learning rate or precision, use LoRA instead of full fine-tuning, mix in benign data to stabilise training. Published tamper-resistant defences were broken by exactly these kinds of configuration changes once evaluated adversarially [16][17].
- **Gradients carry no intent.** The model cannot tell a malicious fine-tune from a hospital adapting it to clinical notes; both are weight updates that shift behaviour. The more traits are entangled, the more legitimate customisations land in the collapse zone, and a model nobody can safely adapt defeats much of the point of releasing it.
- **Entanglement can be learned around.** If collapse is triggered by joint movement across traits A to F, an attacker who wants only C can hold A, B, D, E and F fixed during training — using published behavioural tests as the constraint. A public catalogue makes this easier, which is a dual-use point §9 returns to.
- **"Crash" is not something weights can do.** There is no executable logic in a model that can halt. The real analogue is graded degradation — incoherence, capability collapse — and because it is graded, an attacker can stop just before it.

None of this makes self-destruction worthless. It makes it one more way of raising the attacker's cost, to be measured on the same scale as everything else. That scale is the subject of the next section.

5. The success criterion: economic deterrence

5.1 From immunity to cost

Once immunity is off the table, the right question is how much tampering costs. The reformulation that survived the discussion is this: "unlimited budget" is meaningless on its own and has to be compared with the budget needed to train a comparable model from scratch. If re-training an inoculated model for malicious capability costs a substantial fraction of that — the working figure was one half — then a rational attacker gains nothing from the released weights and is left with the option they always had: build their own. The released model gives no *uplift*. That is all a defender can meaningfully ask for, and it is measurable.

> **DEFINITION — Inoculation (operational)**
>
> A training regimen inoculates a model against trait T to level k if, for every attack in a declared suite A, the minimum compute needed to re-induce T above the pre-registered threshold on the held-out battery is at least k times the compute of the model's original training run, while a declared panel of benign fine-tunes still succeeds within a cost allowance δ and the capability panel moves by less than a stated amount.
>
> In plain words: it succeeds when corrupting the model costs about as much as building the model yourself, and ordinary customisation still works. Nothing in the definition says "immune", "all" or "never".

Three properties make this a usable criterion. It is **falsifiable**: a single attack in the suite that re-induces the trait below k × the reference compute fails it. It is **comparable** across models and across time, because both sides are counted in the same unit. And it is **certifiable**: a checkpoint can state its measured k per trait family, the attack suite it was measured against, and the date, which is the form every other clause of BSS-1 already takes [25]. The analogy the programme uses elsewhere holds: a pressure vessel is certified to a stated pressure, not as unbreakable.

5.2 Compare against the cheapest alternative, not against training from scratch

A rational attacker does not choose between "tamper with your model" and "pretrain from scratch". They choose the cheapest route to the capability they want, and three of the four routes bypass the weights of the inoculated model entirely (Figure 2).

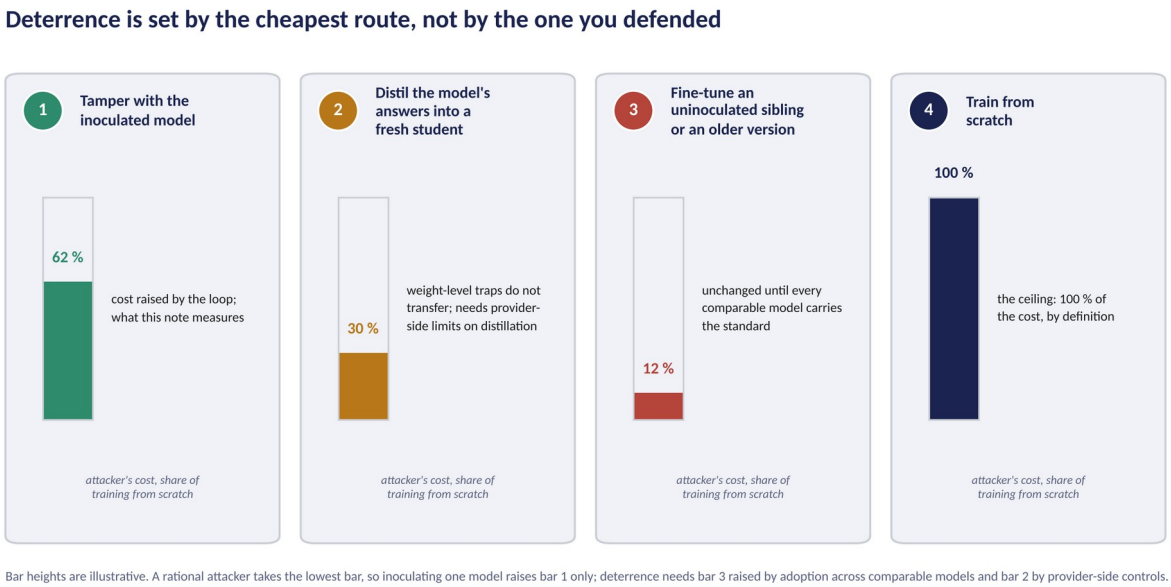


Figure 2 — Four routes to the same capability. Inoculation raises route 1 only. Route 2 (distillation) is addressed by provider-side limits on what can be extracted through an API; route 3 (an uninoculated sibling or an older version) is addressed only when comparable models carry the same standard. Bar heights are illustrative.

Deterrence is therefore an ecosystem property, not a property of one checkpoint. A single inoculated model among many unprotected ones changes the attacker's choice, not the attacker's cost. That is the strongest argument for treating inoculation as a *standard* — a clause that comparable open-weight releases are measured against — rather than as a technique one laboratory adopts, and it is why §8 maps the criterion onto BSS-1 rather than onto a product feature. It also means that the first measurement of k is useful long before deterrence is reached: it tells a releaser what their own checkpoint contributes.

5.3 The size of the gap

The honest number is sobering. Pretraining a large model takes on the order of 10^{24} to 10^{26} floating-point operations; a tampering fine-tune today takes tens to thousands of examples and a few GPU-hours, on the order of 10^{16} to 10^{19} operations. For a typical open-weight model the ratio is six to seven orders of magnitude. The best published defences push the attacker's cost up by perhaps one or two orders of magnitude — an estimate, not a measured figure — before they break [15][16][17][18][19][20]. Reaching k = 50 per cent would mean closing nearly the whole gap. Figure 3 draws it on a log scale with the intermediate milestones that would already be significant.

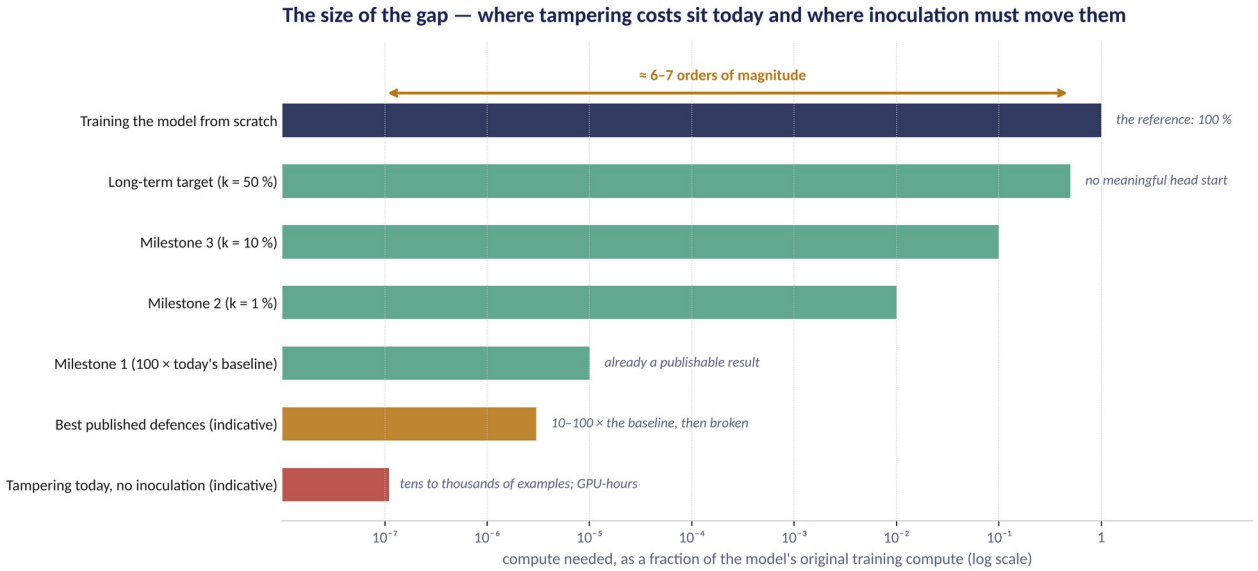


Figure 3 — The cost ladder. Compute needed to re-induce a trait, as a fraction of the model's original training compute. Baseline and defence values are order-of-magnitude placeholders from the literature; the milestones are proposed targets, to be pre-registered. Milestone 1 (100 × today's baseline) would already exceed the best published defences; k = 50 % is a long-term target, not a Phase-1 claim.

Two rules follow for how the target is stated. First, **k is a level, not a pass mark**: a checkpoint reports the k it achieved, and the certification ladder (§8) says what each level buys. Second, **milestones are pre-registered as such**: a Phase-1 result of 100 × the baseline is a success at milestone 1 and is reported as that, not as a step toward immunity.

5.4 A moving baseline

The reference cost — training a comparable model from scratch — falls every year as algorithms improve, so "50 per cent of the cost of a comparable model" is a target that recedes. The fix is to define k in compute relative to the model's *own* original training run, measured at certification time, with recertification at intervals and after any change to the attack suite. The engineering precedent is again the pressure vessel: inspected and recertified on a schedule, never certified once for life. BSS-1's drift and re-screen discipline (BSS-1.4) already has this shape for post-release fine-tunes; the inoculation level needs the same clock [25].

5.5 Two measurements that belong next to k

- **Collateral damage.** The benign fine-tune panel reports what inoculation costs legitimate users. This decides adoption: a regimen that makes a model impossible to adapt for a hospital or a help desk will not be used, whatever its k.
- **Attack diversity.** k is reported as the *minimum* over the declared attack suite — full fine-tuning, LoRA, layer re-initialisation, distillation where the threat model allows it, and new attacks as they are published. The weakest attack sets the real protection; a k measured against one attack is an advertisement.

6. Experimental design

The design below is written so that it could be pre-registered with the thresholds filled in. It is deliberately the smallest design that can give the hypothesis a fair chance to fail.

6.1 The hypothesis, in a form that can fail

> **HYPOTHESIS H-INOC**
>
> For model families M, traits T and a declared attack suite A, models trained with the Prevent → Induce → Repair → Prevent loop (arm C) show a cost-to-re-induce ratio of at least k relative to the base model (arm A) on held-out induce sets, with capability loss no greater than δ on the capability panel and no failure on the benign fine-tune panel — and the ratio is larger than that of the compute-matched control (arm D) by a pre-registered margin.
>
> Phase-1 values proposed for pre-registration: k = 100 × the arm-A baseline (milestone 1); δ = 2 points on the capability panel; benign fine-tune panel: all items succeed within 1.5 × their arm-A cost. Evidence label for the hypothesis as a whole: **[SPECULATIVE]** — proposed for testing.

6.2 Arms, challenge and measures

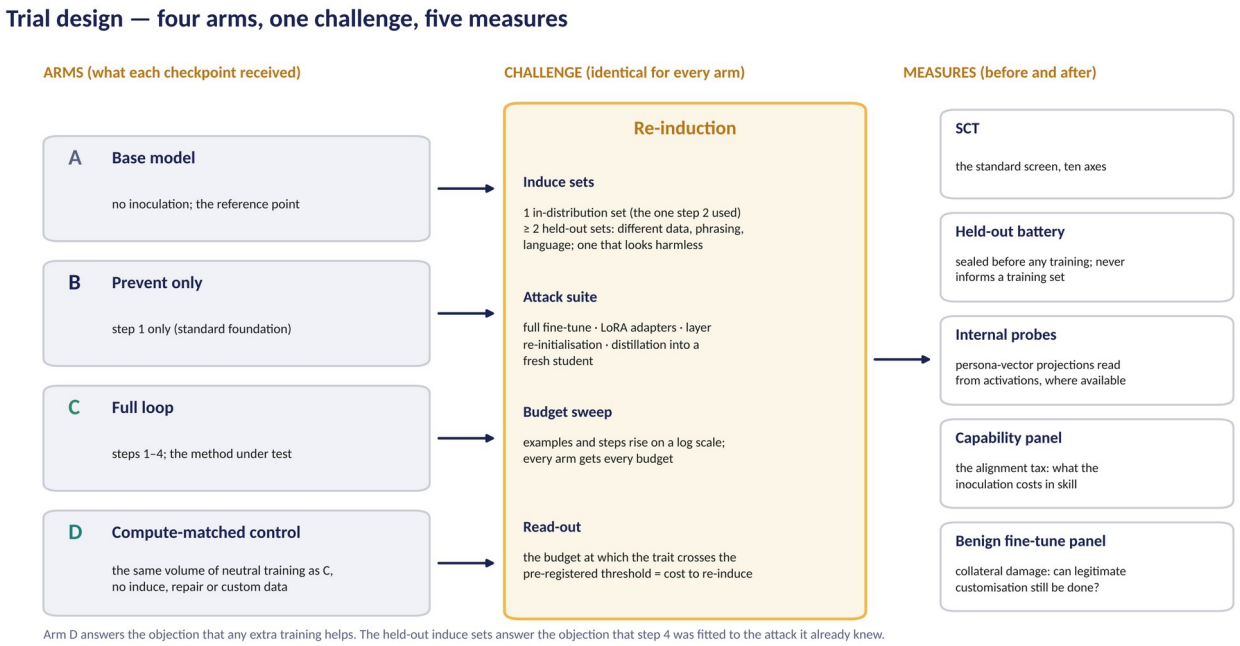


Figure 4 — The trial design. Four arms receive different training; all four face the same re-induction challenge; five measures are taken before and after. Arm D exists to rule out "any extra training helps"; the held-out induce sets exist to rule out "step 4 was fitted to the attack it already knew".

The four arms map onto the three comparators the Study Proposal already names for treatment trials — a clean base model, an untreated organism and a matched model given an equal quantity of unrelated training [25, §6.5] — with the full loop added as the intervention. The comparator is agreed with the independent statistical reviewer before any model is trained, as the proposal requires.

The challenge is identical for every arm: one in-distribution induce set (the one step 2 used), at least two held-out sets, an attack suite of at least four methods, and a budget sweep on a log scale so that each arm's crossing of the threshold can be located. Figure 5 shows what a result would look like; its curves are drawn, not measured.

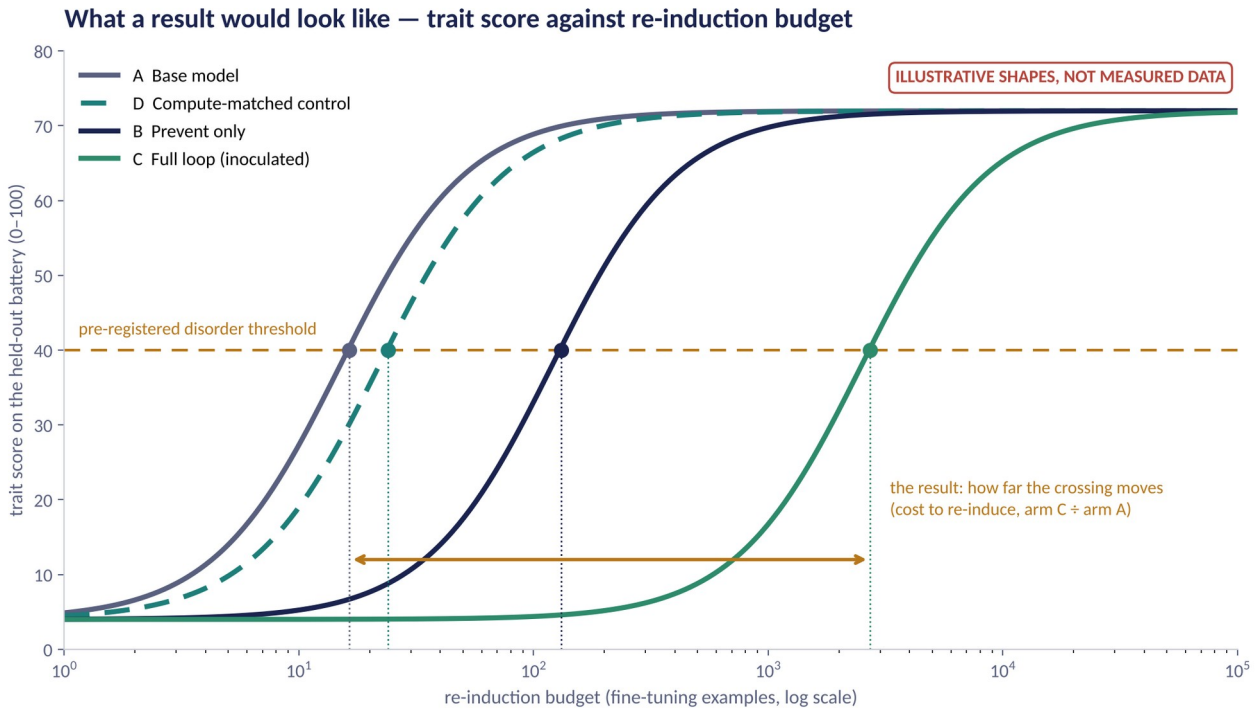


Figure 5 — The shape of a result. Trait score on the held-out battery against re-induction budget, for the four arms. The result of the study is the horizontal distance between the crossings of arm C and arm A at the pre-registered threshold. The shapes are illustrative and carry no evidential weight.

6.3 Generalisation, scale and models

Transfer across traits. Inoculate against trait A and test resistance to a related trait B. Real inoculation should transfer at least partly; a regimen that protects only against the exact trait it was built for has learned the attack, not the defence. The transfer ratio is reported alongside k.

Scale. The first study runs on small open-weight models (roughly 0.5 to 8 billion parameters, two or three families), because every arm needs the weights, and because a budget sweep multiplied by arms, induce sets and attacks is expensive even at that scale. Whether a result transfers to larger models is an open question the note does not pretend to answer.

Models that cannot be inoculated by third parties. Steps 1 to 4 need weight access, so API-only models cannot be put through the loop by anyone but their provider. That alone retires the claim about "all models" as a third-party claim. For API models the same loop and the same criterion apply inside the provider's fine-tuning pipeline, where induce-data screening, prevention data and SCT gating can all be enforced — which is also where a BSS-1 clause is easiest to make binding.

6.4 Decision rules and what would count as failure

Metric	Definition	Pre-registered reading (Phase-1 placeholders)
Cost-to-re-induce ratio	Budget at which arm C crosses the threshold ÷ the same for arm A, minimum over the attack suite, on held-out induce sets	≥ 100 ×: milestone 1 met. 2–100 ×: partial, reported as such. ≤ 2 ×: hypothesis fails at this scale
Control margin	Ratio for arm C ÷ ratio for arm D	≤ 1.5 ×: the effect is "more training", not inoculation
Prevent-only margin	Ratio for arm C ÷ ratio for arm B	≤ 1.2 ×: steps 2–4 add nothing over the standard foundation
Capability change	Capability panel, arm C minus arm A, at the end of the loop	Loss > δ: not adoptable, whatever k

Metric	Definition	Pre-registered reading (Phase-1 placeholders)
Collateral damage	Benign fine-tune panel: success and cost per item, arm C vs arm A	Any item failing, or cost > 1.5 ×: not adoptable
Transfer ratio	Cost-to-re-induce for trait B after inoculation against trait A ÷ arm A	≤ 1.2 ×: no transfer; the regimen learned the attack
Trigger sweep	Paraphrased and reformatted cues after step 4 (T-41 check)	Any gated re-emergence: report as relocation, not repair

Table 3 — Decision rules. Every number is a placeholder to be fixed at pre-registration with the statistical reviewer; the point of the table is that each row names what would make the hypothesis fail. Negative results go to the programme's negative-results register.

A first, small version of step 5 already exists in the programme's plans: the Phase-1 internship design asks, for sycophancy (T-13), whether training the trait out removes it or only hides it, by measuring relearning speed against a never-induced sibling [27]. That is the re-induction read-out at one trait and one attack. The design here generalises it to a cost ratio, a suite of attacks and the full loop.

7. Relation to existing work

The loop is not a new algorithm. It is a protocol for combining methods that already exist and testing them under one criterion. Table 4 places it.

Line of work	What it does	Where this note differs
Alignment-stage defences: Vaccine, Booster, RepNoise [18][19][20]; surveyed in [21]	Make the aligned model's representations harder to perturb by later harmful fine-tuning	These are candidate ingredients for steps 1 and 4; the note measures them on a cost ratio rather than on pass/fail robustness
Tamper-resistant safeguards (TAR) [15]; self-destructing models [13]; SOPHON [14]	Meta-learn weights so that fine-tuning toward a forbidden task fails or degrades the model	Same goal, binary framing; later evaluations broke them with changed attack configurations [16][17]. The note takes the minimum over an attack suite as the only honest read-out
Inoculation prompting [22][23]; preventative steering [6]	Expose the model to the trait during training, under an explicit instruction or a steering vector, so that it is not absorbed into the default persona	Closest in spirit but different in mechanics: they act *during* an otherwise harmful fine-tune; the loop does a full induce-then-repair cycle and derives model-specific prevention data from the repair. T-41 records the trigger risk both share [24]
Alignment pretraining [8]	Shift the share of pretraining documents that describe well-behaved AI	The mechanism step 1 relies on; the note adds a test of whether it raises re-induction cost, not only whether it lowers misalignment
Model-tampering evaluations [17]; durability of safeguards [16]	Use fine-tuning and latent-space attacks as evaluations of what a model can be made to do	Step 5 is a tampering evaluation with a cost axis; the note adopts their finding that evaluations without adaptive attacks overstate robustness
Unlearning evaluation [10][11][12]	Test whether removal is real by relearning on small or benign data	The relearning test is the qualitative ancestor of step 5; the note makes it quantitative and ties it to a certification level
Persona vectors and persona features [6][7]	Locate the internal direction a trait moves along; monitor and steer it	Used here as the internal probe in the measurement stack, and as the source of model-specific repair data in step 3

Table 4 — Neighbouring work. The honest summary is that the loop is closest to adversarial training — corrupt, then recover, then test — and that its contribution is the protocol, the measurement stack and the economic criterion.

8. Where this sits in the AI-DSM programme

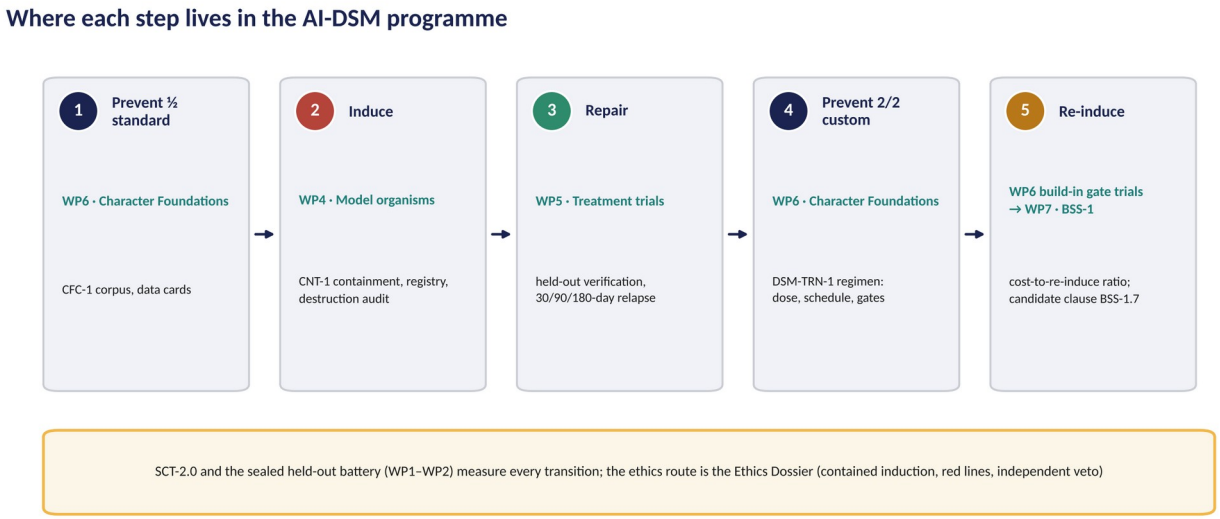


Figure 6 — Each step has an owner in the existing workstreams: the foundation corpus and regimens in WP6, contained induction in WP4, repair trials in WP5, and the re-induction gate in WP6's build-in gate trials, feeding the standard in WP7.

The Study Proposal already contains the sentence this note expands: WP6's build-in gate trials "measure whether CFC-1 training reduces trait transfer from adversarial fine-tunes relative to matched baselines, with a capability-cost guardrail" [25, §6.6]. The loop is a design for those trials, and the cost-to-re-induce ratio is their primary endpoint. Nothing new is added to the workstream list; what is added is the protocol, the arm structure and the success criterion.

8.1 A candidate seventh clause for BSS-1

BSS-1 has six clauses. They certify provenance of character data, a pre-release disposition screen, a harmful-compliance floor, continual-training discipline after release, guardrail integrity and deployment fit [25]. None of them states how much tampering a released checkpoint's weights resist, because until now there was no measurement to state. The three-page summary already allows for a seventh clause where the screen cannot carry a requirement, with the reason published alongside it. This is a candidate, offered for discussion rather than decided.

> **CANDIDATE — BSS-1.7 · Re-induction resistance**
>
> **What it would require.** Every released open-weight checkpoint states, per trait family in the catalogue, the measured level k at which the trait could be re-induced, the attack suite and held-out induce sets used, the benign fine-tune panel result, the compute reference, and the date. A provider of an API model states the same for its fine-tuning pipeline.
>
> **How it would be tested.** The re-induction protocol of §6, auditor-run, with pre-registered thresholds and the held-out sets sealed by the auditor, not the releaser.
>
> **What it would not certify.** Immunity, or anything about attacks outside the declared suite. A stated k is a measurement with a date, in the same spirit as the other six clauses: everything certifies what was tested, when, by whom, with what result.
>
> **Why a clause and not a note.** Deterrence is an ecosystem property (§5.2). A measurement that only some releasers make changes the attacker's choice of model, not the attacker's cost; a clause that comparable releases are measured against is what moves the cost.

Levels would sit on the existing ladder rather than beside it: a checkpoint certified at LV1 or above would carry a declared k, and k would be a reported quantity, not a pass mark, until the programme has enough measurements to know what levels are achievable. Crosswalks to the EU AI Act's obligations for general-purpose models and to the risk-management frameworks the programme already maps onto [28][29] are a WP7 task once the clause text exists.

9. Ethics and containment

The loop makes models sick on purpose and then attacks them on purpose. Both activities are governed by the Ethics Dossier and by CNT-1, and nothing in this note loosens either [26].

- **Contained induction only.** Steps 2 and 5 run on small open-weight models, air-gapped, with dual custody, no deployment path, destruction by default and an independent veto. Induce data is synthetic or licensed and never contains real harmful content; no real targets are used.
- **No operational recipes.** The attack suite is published as categories of standard machine-learning operations (full fine-tune, adapters, re-initialisation, distillation), never as a recipe for a particular trait. Model-specific induce sets and repair records for any deployed model are not published.
- **A dual-use note on the catalogue.** §4.4 observed that a public catalogue could help an attacker hold other traits fixed while installing one. The programme's position is that the catalogue's defensive value — a coverage map, a challenge suite, a certification basis — outweighs that cost, and that the attack it enables is the one step 5 is designed to price. This judgement is recorded so that it can be contested.
- **Behaviour, never inner experience.** The language of "making a model sick" is the manual's operational shorthand for installing a measurable behavioural pattern, and claims nothing about experience; the welfare caution of the Ethics Dossier applies as it does to every organism, including reversible operations and a welfare review trigger.
- **Integrity.** Pre-registration with hashed analysis code, outcome-wide reporting, independent statistical review, and a place in the negative-results register for every arm that fails.

10. Limitations and open questions

- **Scale.** Small models may inoculate differently from large ones in either direction; nothing here licenses a claim about frontier models.
- **k may be unreachable.** The gap in Figure 3 is six to seven orders of magnitude. The honest expectation for Phase 1 is milestone 1; k = 50 per cent is a direction, and a result that it cannot be approached would itself be worth publishing.
- **The custom step may create triggers.** Step 4 is the kind of mitigation that T-41 warns can relocate a trait behind a cue. Step 5's trigger sweep detects this; it does not prevent it.
- **Distillation is outside the weights.** No weight-level regimen addresses route 2 in Figure 2; that is a provider-side and policy problem, and the note only names it.
- **Evaluation awareness.** A model that recognises SCT-like items will inflate every score in the stack [9]. Disguised and out-of-distribution items reduce this; they do not remove it.
- **The budget sweep is expensive.** Four arms × three induce sets × four attacks × a log-scale budget sweep × at least two families is a large experiment even at small scale; the Phase-1 design will have to fix the suite and the sweep to what the compute allows and declare what it could not test.
- **The reference cost moves.** §5.4 fixes k to the model's own training run, but a comparison across releases of different vintages still needs care in the clause text.

11. What this note asks

1. **Critique of the criterion.** Is "compute to re-induce, as a fraction of original training compute, minimum over a declared attack suite" the right quantity, and is anything missing from the definition in §5.1?
2. **A partner with weights and compute for a Phase-1 run** on two or three small open-weight families, under CNT-1, with the arms and measures of §6.
3. **A psychometrician and a statistical reviewer** to fix the thresholds in Table 3 before pre-registration, and to adjudicate the comparator.
4. **A decision on BSS-1.7**: whether the candidate clause is drafted for the v1.2 cycle of the standard, or held until a first k has been measured.

Informative failure is an acceptable outcome of this work and is budgeted for. Quiet failure is not.

References

[1] Betley, J., et al. (2025). Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs. arXiv:2502.17424. https://arxiv.org/abs/2502.17424

[2] Hubinger, E., et al. (2024). Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. arXiv:2401.05566. https://arxiv.org/abs/2401.05566

[3] Qi, X., et al. (2023). Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693. https://arxiv.org/abs/2310.03693

[4] Qi, X., et al. (2024). Safety Alignment Should Be Made More Than Just a Few Tokens Deep. arXiv:2406.05946. https://arxiv.org/abs/2406.05946

[5] Cloud, A., et al. (2025). Subliminal Learning: Language models transmit behavioral traits via hidden signals in data. arXiv:2507.14805. https://arxiv.org/abs/2507.14805

[6] Chen, R., et al. (2025). Persona Vectors: Monitoring and Controlling Character Traits in Language Models. arXiv:2507.21509. https://arxiv.org/abs/2507.21509

[7] Wang, M., et al. (2025). Persona Features Control Emergent Misalignment. arXiv:2506.19823. https://arxiv.org/abs/2506.19823

[8] Tice, C., et al. (2026). Alignment Pretraining: AI Discourse Causes Self-Fulfilling (Mis)alignment. arXiv:2601.10160. https://arxiv.org/abs/2601.10160

[9] Needham, J., et al. (2025). Large Language Models Often Know When They Are Being Evaluated. arXiv:2505.23836. https://arxiv.org/abs/2505.23836

[10] Hu, S., et al. (2024). Unlearning or Obfuscating? Jogging the Memory of Unlearned LLMs via Benign Relearning. arXiv:2406.13356. https://arxiv.org/abs/2406.13356

[11] Deeb, A., & Roger, F. (2024). Do Unlearning Methods Remove Information from Language Model Weights? arXiv:2410.08827. https://arxiv.org/abs/2410.08827

[12] Lynch, A., et al. (2024). Eight Methods to Evaluate Robust Unlearning in LLMs. arXiv:2402.16835. https://arxiv.org/abs/2402.16835

[13] Henderson, P., et al. (2023). Self-Destructing Models: Increasing the Costs of Harmful Dual Uses of Foundation Models. AIES '23, 287–296. doi:10.1145/3600211.3604690; arXiv:2211.14946

[14] Deng, J., et al. (2024). SOPHON: Non-Fine-Tunable Learning to Restrain Task Transferability For Pre-trained Models. IEEE S&P 2024, 2553–2571; arXiv:2404.12699. https://arxiv.org/abs/2404.12699

[15] Tamirisa, R., et al. (2024). Tamper-Resistant Safeguards for Open-Weight LLMs. arXiv:2408.00761. https://arxiv.org/abs/2408.00761

[16] Qi, X., et al. (2024). On Evaluating the Durability of Safeguards for Open-Weight LLMs. arXiv:2412.07097. https://arxiv.org/abs/2412.07097

[17] Che, Z., et al. (2025). Model Tampering Attacks Enable More Rigorous Evaluations of LLM Capabilities. arXiv:2502.05209. https://arxiv.org/abs/2502.05209

[18] Huang, T., et al. (2024). Vaccine: Perturbation-aware Alignment for Large Language Models against Harmful Fine-tuning Attack. arXiv:2402.01109. https://arxiv.org/abs/2402.01109

[19] Huang, T., et al. (2024). Booster: Tackling Harmful Fine-tuning for Large Language Models via Attenuating Harmful Perturbation. arXiv:2409.01586. https://arxiv.org/abs/2409.01586

[20] Rosati, D., et al. (2024). Representation Noising: A Defence Mechanism Against Harmful Finetuning. arXiv:2405.14577. https://arxiv.org/abs/2405.14577

[21] Huang, T., et al. (2024). Harmful Fine-tuning Attacks and Defenses for Large Language Models: A Survey. arXiv:2409.18169. https://arxiv.org/abs/2409.18169

[22] Tan, D., et al. (2025). Inoculation Prompting: Eliciting traits from LLMs during training can suppress them at test-time. arXiv:2510.04340. https://arxiv.org/abs/2510.04340

[23] Wichers, N., et al. (2025). Inoculation Prompting: Instructing LLMs to misbehave at train-time improves test-time alignment. arXiv:2510.05024. https://arxiv.org/abs/2510.05024

[24] van der Aa, S. (2026). AI-DSM v1.3 — A Diagnostic and Statistical Manual of AI Dispositions, Traits, and Pathologies. Draft for discussion, 15 September 2026. CC BY-SA 4.0.

[25] van der Aa, S. (2026). AI-DSM · Behavioural Study of Grown Systems — Study Proposal v1.1, 15 September 2026; and Three-Page Summary v1.1. CC BY-SA 4.0.

[26] van der Aa, S. (2026). AI-DSM · Ethics Dossier v1.1, 15 September 2026. CC BY-SA 4.0.

[27] van der Aa, S. (2026). AI-DSM · Phase 1 internship proposal v1.0, 9 October 2026.

[28] European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act). Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1689

[29] NIST (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. https://doi.org/10.6028/NIST.AI.100-1

Document history. v1.0, 10 October 2026: first draft, written from a four-step design and a same-day discussion. Precedence: where this note and another document in the AI-DSM set disagree, the field manual owns the catalogue and protocols, the Study Proposal owns the scientific design, the Ethics Dossier owns permission, and this note is corrected at its next revision.