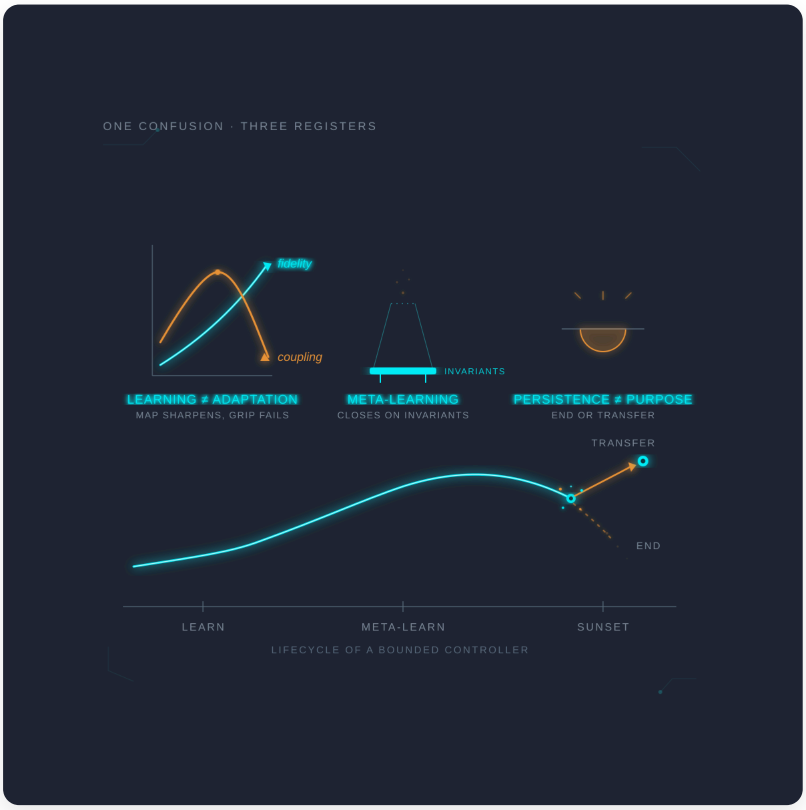

When to Stop Learning, and When to Stop

Learning, adaptation, and the lifecycle of a bounded controller

Paper XXI in the Governance as Engineering series

Traces the lifecycle of a governance architecture. Three separations organize it: learning is not adaptation — a registered minimal model shows the map improving while the coupling degrades; meta-learning is not free — a bounded hierarchy of learning rules must close on invariants; and persistence is not the goal — the terminal adaptive act may be to end or transfer rather than to continue. Paper XXI in the Governance as Engineering series.

Björn Kenneth Holmström

July 2026

`https://bjornkennethholmstrom.org/working-papers/when-to-stop-learning`

> *The series has built from static factorization through dynamics to control under change. This paper takes the last step: the lifecycle of a governance architecture — how it learns, when learning stops helping, and when the architecture itself should end. Three separations organize it. Learning is not adaptation: improving the internal model is not the same as maintaining the coupling to the world, and the two can oppose each other. Meta-learning is not free: the ladder of learning-to-learn cannot regress indefinitely in a bounded system and must close on invariants. Persistence is not the goal: an adaptive controller's terminal act may be to end or transfer its coupling rather than to continue. Each separation is a place where a bounded controller can mistake an internal good — a sharper model, a cleverer update rule, its own continuation — for the external good it exists to serve. The learning/adaptation dissociation is given a registered minimal model,* ***[R within the model]****; the meta-learning and sunsetting results are structural and normative respectively,* ***[R]*** *for the regress argument and* ***[IP]*** *for the governance readings, and are marked as argument rather than demonstration.*

Abstract

A governance architecture is not static. It updates its model of the world, sometimes updates the rules by which it updates, and eventually confronts the question of whether it should still exist. This paper treats that lifecycle and argues that three transitions along it are governed by a single pattern: at each, a bounded controller is liable to confuse an internal success with the external one it was built for.

The first separation is learning from adaptation. Learning improves the fidelity of an internal model; adaptation maintains a controller's coupling to its world; and because incorporating a model revision costs the action loop capacity, the two can be opposed. A registered minimal model demonstrates it: across thirty seeds, as a controller's learning rate rises, model fidelity improves monotonically while coupling — its fraction of time within viability bounds — peaks at an intermediate rate and then degrades, so that the fastest learner holds the best model and the worst grip. The condition is an absorptive-capacity inequality: adaptation holds only while the demand learning makes actionable stays within what the loop can absorb.

The second separation is meta-learning from free improvement. If a controller can improve its update rule, it can in principle improve the rule that improves the rule, opening a hierarchy. In a bounded system that hierarchy cannot regress indefinitely — each level costs representational capacity — so it must close, terminating on some level held invariant. Closure requires invariants: a small set of things held still (identity boundary, a certification kernel, memory, timescale separation, a plural reserve) so that everything else can move. Constitutional engineering, in this reading, is the choice of what to hold fixed.

The third separation is persistence from purpose. An institution exists to maintain a coupling; when the coupling is no longer required, or can no longer be maintained without destroying the invariants that make maintenance possible, its adaptive act is to end or to transfer the coupling, not to continue. The characteristic failure is the controller that mistakes preservation of itself for preservation of the coupling — the third and terminal form of the internal/external confusion. The paper is explicit about its tiers: the learning/adaptation result is demonstrated within a model; the meta-learning closure is a structural argument; sunsetting is a normative design claim a simulation could illustrate but not settle, and is presented as such.

1. The lifecycle question

The series has climbed a definite slope. It began with the static object — the factorization, the carving of the world into the variables a controller represents — and the conditions under which one factorization is adequate to a world. It moved through the dynamics: how a controller re-identifies its own model while regulating (Paper XIV), how the sense–learn–execute loop paces adaptation (Paper XV), how the alternatives a controller suppresses erode unless replenished (Paper XVI), how a learning jurisdiction reshapes the boundary it was drawn to hold (Paper XVIII). What it has not yet asked is the question those dynamics imply: what is the *lifecycle* of a governance architecture? An architecture that learns, that may learn how to learn, and that exists for a reason — such a thing has an arc, from the first revision of its model to the last question of whether it should persist. This paper is about that arc.

Three transitions along it share a structure, and naming the structure first makes the paper's unity visible. At each transition, natural language offers a slide from an internal achievement to an external one, and the slide is a category error. A system *learned*, we say, and mean it *adapted* — as if a better model were a better grip. A system *can improve its update rule*, we observe, and infer that it *should* keep improving the rule that improves the rule — as if the ladder had no cost and no top. A system *is still functioning*, we note, and conclude that it *should persist* — as if continuation were the goal rather than the coupling continuation was meant to serve. Each slide substitutes something the controller can measure about itself — model fidelity, rule sophistication, its own survival — for the thing it exists to do, which is to keep some part of the world coupled to some valued outcome. The lifecycle is the sequence of points at which that substitution becomes tempting, and the paper's claim is that governing well across a lifecycle is, at each point, the discipline of not making it.

The three separations are therefore not a list but a progression, tightening as they go. The first, learning from adaptation (§§2–3), is the mildest and the one this paper can demonstrate: a controller improving its model while losing its coupling, shown in a registered minimal model. The second, meta-learning from free improvement (§4), is structural: the hierarchy of learning-to-learn cannot regress forever in a bounded system, so it must close on invariants, and the engineering content is which invariants. The third, persistence from purpose (§6), is the sharpest and the one no simulation settles: it is the normative claim that a well-designed institution asks not "how do I survive?" but "does my continued form still serve the coupling I was created to maintain?" — and that the answer is sometimes no. The paper draws on the Ashby-shock and certification-cost results of Paper XX as the machinery that drives the arc: a rising task-relevant variety forces learning, learning generates incorporation demand, and the accumulating cost of staying coupled is what eventually poses the question of refactoring or ending. Section 7 gathers the three separations back into the single pattern; §8 states what is demonstrated and what is only argued, since two of the three are argument by design.

2. Learning is not adaptation [R for the separation, IP for the governance reading]

2.1 Two definitions that do not coincide

The series has treated learning and adaptation as near-synonyms often enough that the slide between them has become invisible, and natural language encourages it: we say a system "learned" and mean it "adapted," as though improving one's model of the world and staying coupled to it were the same achievement. They are not, and the separation is worth stating precisely because everything in this paper's lifecycle depends on it.

Adaptation is the process that maintains adequate coupling between a controller and its environment under change, such that the controller keeps meeting its viability conditions — performance within bounds, essential variables in range, the loop with the world intact. It is judged by outcome: does the system remain coupled? It is a property of the whole loop's relation to the world.

Learning is the process of updating a controller's internal model — its representation, its predictive map, its parameters — from data. It is judged by the fidelity of that model: does the map track the territory better than it did? It is a property of the learner's relation to its own internal state.

The two are judged against different things, and once that is seen the dissociation is immediate. Learning can occur without adaptation: a controller can improve its model while its coupling degrades, which is exactly what §3 exhibits — the map sharpening as the grip fails. Adaptation can occur without learning: a controller with a fixed model can maintain coupling perfectly well if its fixed model happens to remain adequate, or if it adapts by non-learning means — buffering, redundancy, retreat to a robust default. Learning is one mechanism by which a system may maintain coupling. It is neither necessary for adaptation nor sufficient for it.

The adaptation triad of Paper XV already encodes the separation without having named it: Sense feeds Learn, but it is Execute that closes the loop with the world. Learning sits in the middle of that chain, and a middle stage can succeed while the stage that actually touches the world fails. If Execute is broken — if the loop cannot act on what the model now knows — then learning has no adaptive effect at all, however much fidelity it gains. The map is not the coupling, and the stage that improves the map is not the stage that maintains the coupling.

2.2 What learning costs the loop

The reason the two can oppose each other, rather than merely differ, is that learning is not a free input to adaptation. It makes two demands on the loop that must act on its results, and the sharper form of this section's claim is about those demands.

Learning *reveals* latent mismatch. A better model surfaces discrepancies between what the controller believed and what the world is doing — discrepancies that were always there but that the coarser prior model did not represent. This is Paper XVI's source terms seen from the learning side: the newly-visible mismatch is real information, and it is also a new obligation, because a mismatch the controller can now see is one it must now act on or knowingly tolerate. Call this revealed demand.

Learning also *creates* incorporation cost. Acting on a revised model is not instantaneous; the loop must reorganize around the revision — retune the actuator, re-plan, re-coordinate the parts that depended on the old model — and that reorganization consumes capacity and induces a transient during which coupling is degraded precisely because the loop is in the middle of changing. Call this created demand. A model revision is a disturbance to the very loop that is supposed to benefit from it, before it is a benefit.

Against these two demands stands the loop's capacity to absorb them — to act on revealed mismatch and pay down incorporation cost without losing coupling. Write it as an inequality:

$$D_{\text{revealed}} + D_{\text{created}} \leq C_{\text{absorb}}.$$

Adaptation succeeds when the demand learning makes actionable stays within what the loop can absorb, and fails when it does not. This is the condition §3 makes mechanical and tests: past a certain learning rate the revision stream exceeds C_{absorb}, and coupling degrades though fidelity keeps improving. The inequality is why the two quantities can be opposed rather than merely distinct — learning drives the left side up, and beyond the point where it crosses the right side, more learning is less adaptation.

2.3 A taxonomy of learning by its effect on coupling

The inequality sorts learning into three kinds, distinguished not by how much the model improves but by what happens to the loop that must incorporate the improvement.

Learning is *adaptive* when the revealed and created demand stays within absorptive capacity: the model improves, the loop incorporates the improvement without losing coupling, and fidelity and coupling rise together. This is the case the naming-slide assumes is the only one — learning that is also adapting.

Learning is *irrelevant* when it improves fidelity along dimensions the loop's coupling does not depend on. The model gets better at predicting things that do not bear on viability; no demand is made on the loop because nothing actionable was revealed, and coupling is unchanged. Harmless, and common — much of what a system learns does not matter to whether it stays coupled.

Learning is *maladaptive* when the demand it makes exceeds absorptive capacity: the revision stream disrupts the loop faster than the loop can reorganize, and coupling degrades. The model is improving and the system is failing, at the same time, for the same reason. This is the dangerous case, because every internal signal looks like success — the fidelity metric climbs monotonically into the regime where the coupling is collapsing — and a controller monitoring only its own learning would see nothing wrong until the loss of coupling arrived from a direction its self-assessment did not cover.

The governance reading, [IP] as always, is that an institution can be studying its problem more and more accurately while governing it worse and worse, and that the accuracy is no defense — may even be the mechanism of the failure, if the pace of revision outruns the institution's capacity to reorganize around each revised understanding. The reform that never settles because each new study forces another restructuring is not failing to learn. It is learning maladaptively: revealing and creating more demand than the loop can absorb, and mistaking the improving map for an improving grip. That mistake — the map's fidelity taken for the loop's health — is the first of the three lifecycle confusions this paper traces, and §3 shows it is not merely possible but mechanical.

3. A model where learning breaks coupling [R within the model]

Section 2 separated learning from adaptation by definition: learning improves the fidelity of an internal model, adaptation maintains a controller's coupling to its world, and the two are judged against different things — the model against the world, the loop against viability. A definitional separation is cheap, though. What earns the separation its place in the series is a mechanism in which the two quantities are not merely distinguishable but actively opposed — where improving the map costs the coupling. This section exhibits one, registered in advance across thirty seeds.

3.1 The mechanism

A controller tracks a slowly drifting latent target and must do two things with it at once: estimate where it is (the fidelity channel) and hold an actuator near it (the coupling channel). The estimate updates toward a noisy observation at a learning rate η — higher η, faster learning, better fidelity. The catch is that every revision of the estimate has to be *incorporated* by the action loop, and incorporation is not free. Each revision injects a disruption into the loop proportional to the size of the revision, and that disruption decays only gradually. A second-order actuator chases the estimate, but while the loop is still absorbing a revision it cannot settle: its effective position is perturbed in proportion to the disruption it is still carrying.

This is the absorptive-capacity inequality made mechanical. The loop's ability to settle between revisions is C_{absorb}; the stream of revisions is $D_{\mathrm{revealed}} + D_{\mathrm{created}}$, the mismatch learning surfaces plus the incorporation cost it generates. When learning is slow the estimate lags the drift — poor fidelity — and the actuator faithfully tracks a lagging estimate, so coupling is poor for want of anything good to track. When learning is fast the estimate tracks the drift closely — good fidelity — but it also churns on observation noise, and the churn keeps the loop perpetually mid-incorporation, so the actuator thrashes and coupling is poor for want of a chance to settle. The revision stream has outrun what the loop can absorb. Between these failures lies a learning rate fast enough to track and slow enough to settle, where coupling is best — but fidelity does not peak there. Fidelity keeps improving as learning accelerates, straight through the point where coupling has already turned over.

3.2 What the run shows

The prediction was three-part and committed before the run: fidelity monotone in the learning rate, coupling non-monotone with an interior peak, and an explicit regime above that peak where fidelity rises while coupling falls. All three held.

Fidelity improved monotonically across the swept learning rates (pooled rank correlation 0.99 between rate and fidelity), from a median tracking score of -0.60 at the slowest rate to -0.08 at the fastest — the map gets steadily better as the controller learns faster, with no reversal. Coupling did not follow it. Median coupling rose from 0.24 to a peak of 0.48 at an intermediate learning rate and then fell away to 0.19 at the fastest — an interior maximum, with every one of the thirty seeds showing a drop of at least ten percentage points from its own best rate to the fastest. And in all thirty seeds the interval above the coupling-optimal rate was a genuine dissociation: fidelity higher at the fastest rate than at the peak, coupling lower. The map improves; the coupling degrades; they move in opposite directions over the same stretch of learning rate. The controller that learns fastest holds the best model of its world and the worst grip on it.

3.3 Reading the result

The line the series has carried since Paper XIV — *learning updates the map; adaptation requires the institution to survive the redraw* — has an instance here that is not a metaphor. Beyond the coupling-optimal rate, the controller is learning correctly: its estimate really is tracking the world better, by the measure that defines learning. It is failing anyway, because the rate at which it revises its model exceeds the rate at which the action loop can absorb the revisions, and adaptation is a property of the loop, not the estimate. A faster learner is a better knower and a worse survivor. Nothing in the fidelity signal warns of this; an institution watching only how well its model predicts would see steady improvement all the way into the regime where its coupling is collapsing.

Two honesty notes belong with the result. First, this is one environment and one operationalization of incorporation cost, so what is established is that learning *can* break coupling by outrunning absorptive capacity, not that it must in every setting — the demonstration makes the dissociation real, not universal. Second, the mechanism reported here is a revision of a first one. An earlier version made incorporation cheap — the actuator merely tracked the estimate under a speed cap — and in that version coupling rose monotonically with the learning rate: no dissociation, the registered prediction failed as stated. The failure was informative, and it is kept on the record: it showed that a mere rate limit does not create the trade-off, and that the dissociation requires incorporation to be genuinely costly, a transient the loop must pay down rather than a ceiling on how fast it may move. The revised mechanism, with that cost made explicit, was registered afresh and is what the thirty-seed run above tests. This is the same registered-failure-then-disciplined-revision the series applies elsewhere; here it also doubles as a finding, because the difference between the two mechanisms *is* the content — adaptation breaks not when learning is fast but when revision outpaces absorption.

4. Meta-learning must close [R for the regress, IP for the governance reading]

If learning is one mechanism a controller uses to stay coupled, the engineering instinct is to improve the mechanism itself — and that instinct opens a hierarchy. A controller that does not learn is L_0: a fixed mapping from observation to action. One that updates its model by a fixed rule is L_1: it learns. One whose learning rule itself adapts — changing its rate, its prior, the features it attends to — is L_2: it learns how to learn, which is meta-learning. And the rule by which the meta-rule adapts is L_3, and the rule above that L_4, and the ladder in principle continues. In governance the levels are familiar: L_1 is a ministry updating its policy from data, L_2 the amendment process that changes how policy is made, L_3 the rule for amending the amendment process, each level more foundational and slower-moving than the one below.

The question the hierarchy poses is whether it converges or regresses without end. Does improving the rule that improves the rule bottom out somewhere, or must a well-designed adaptive system climb the ladder forever, since any fixed level is one more thing that could in principle be improved?

Bounded representation answers it, and the answer is the same finiteness argument that runs through Paper XX. Each level of the hierarchy is itself a represented object — a rule the controller must store, evaluate, and update — and representation is bounded. An infinite tower of rules-about-rules is unrepresentable in a finite system; the capacity each additional level consumes is capacity taken from the levels below, which are the ones actually coupled to the world. So the ladder cannot regress indefinitely. It must *close*: terminate at some finite level that is held fixed and not itself subject to revision. This is not a limitation the designer may lament and hope to engineer away; it is a structural necessity of finite systems, and the only choice it leaves open is *where* to close, not whether.

Two features make closure less costly than the bare argument suggests, and both matter for the governance reading. The levels have sharply diminishing returns: higher rules change more rarely and bear on outcomes less directly per change, so truncating the tower at a modest height sacrifices little adaptive power. And closure does not freeze the whole system — it freezes only the top. Everything below the closed level continues to adapt; what is held fixed is the rule by which the adaptation of adaptation is itself governed. The tower is effectively finite because it is both cheap to truncate and harmless to truncate high. What closure *requires*, though — what must be true of the level held fixed for the whole structure to remain adaptive rather than merely frozen — is the subject of §5.

5. Closure requires invariants

To close the hierarchy is to hold some level still. But holding a level still is not automatically safe: a system can freeze the wrong thing and lose its adaptive capacity, or freeze nothing firmly enough and let the regress reopen under pressure. The content of constitutional engineering is which things to hold fixed so that everything else can move — and the claim of this section is that closure rests on a small set of invariants, the things a system must *not* revise in order that its revisions stay adaptive.

Five candidates recur, drawn from the exploratory work and stated here as the load-bearing set rather than an exhaustive one. An **identity boundary**: a stable answer to what the system is and what falls inside its jurisdiction, without which a self-modifying controller cannot tell adaptation from dissolution. A **certification kernel**: a fixed core of how the system checks whether its factorization still fits the world — the one part of the audit machinery that is not itself up for continuous revision, on pain of the system losing any stable standard against which to judge its own adequacy. A **memory**: a preserved record of what has been tried and what it cost, without which each adaptation re-learns from scratch and the meta-level cannot improve. A **timescale separation**: the higher levels must change more slowly than the lower, or the tower collapses into a single fast-churning level with no stable rule governing the churn — the meta-learning analogue of §3's absorptive-capacity limit, now applied to the rules rather than the model. And a **plural reserve**: a maintained stock of alternative factorizations not currently in use, which is precisely Paper XIX's sentinels and bridges — the held-in-reserve frames that let the system expand its factorization when the task-quotient shifts rather than being trapped in the one it optimized around.

Two refinements keep this from being a rigid checklist. The invariants are better read as a *capacity threshold* than as five items each strictly necessary on pain of collapse: what closure requires is that enough invariant structure be held to anchor the revising levels, and the five are the dimensions along which that structure is supplied, not five independent single points of failure. And the plural reserve deserves to be named explicitly rather than folded into certification or slack, because it is the invariant that connects this paper's lifecycle to the role triad of Paper XIX: the reserve of unused factorizations is what a system draws on when an Ashby-shock raises the task-relevant variety, and a controller that has optimized its reserve away — kept only the current governor, discarded the sentinels and bridges — has eliminated the invariant that closure most depends on, and will meet the next shift in the world with nothing held back to adapt with.

The design principle the section commits to is a single reorientation: do not ask only how a system changes; ask what it must preserve in order that change remains adaptive. The art of constitutional engineering is choosing which few things must be held still so that everything else can safely move. This is [R] for the structural claim that closure requires *some* invariant anchor — that follows from §4's finiteness — and [IP] for the identification of these particular five as the anchors that matter for governance.

6. Persistence is not the goal — sunsetting [IP]

The lifecycle has a terminus, and the last separation is the sharpest: an institution's continued existence is not the same as the fulfillment of its purpose, and the two can diverge. This is a normative and design claim rather than one a simulation settles, and the paper marks it as such — but it follows the same pattern as the two before it, and completes the arc.

An institution exists to maintain a coupling: a relation between some part of the world and some valued outcome that would not hold without it — a regulatory domain, a service obligation, a protective function. The invariants of §5 are the floor that makes maintaining the coupling possible. The institution has reason to persist exactly as long as two conditions hold: the coupling is still required — the world still generates the disturbance or the need — and the institution can still maintain it without destroying its own invariants. The right time to end is when either condition fails, and the two failures are different in kind.

The first is **mission completion**: the coupling is no longer required. The disturbance has vanished, the transition is complete, the protected class no longer needs protection — a post-war reconstruction authority, a disease-eradication program, a body convened for a single event. Here sunset is not failure but success, and the engineering task is to build a completion criterion into the architecture from the start and to guard against the institution manufacturing artificial demand to justify its own continuation. This is where finite-problem institutions differ from standing-condition ones: an institution created for a bounded task should be designed to end, while one maintaining a permanent coupling — a need that does not go away — should not carry a sunset clause it will only be tempted to evade. Confusing the two is a design error in both directions: a permanent body built to expire, or a temporary one built to persist.

The second failure is **loss of adaptive capacity**: the coupling is still required, but the institution can no longer maintain it without violating the invariants of §5 — its certification kernel has eroded, its identity boundary has frozen, its members no longer share the minimal common ground its audit depended on. This is not success, and it is also not, usually, a case for simple sunset. Because the coupling is still needed, the adaptive act is *succession or transfer*: the institution's final useful function is to hand its coupling to a successor that can still maintain it, before its own decay drops the coupling entirely. The terminal institution ends; the failing perpetual institution transfers.

The characteristic failure across both is a single confusion, and it is the last of the three the paper has traced: the controller mistakes preservation of itself for preservation of the coupling. It reasons about how to survive when the question was whether its survival still serves the thing it was built to serve. This confusion has a mirror-image danger worth naming, because it cuts the other way: the *weaponized premature sunset*. An institution whose success shows up as the absence of failure — the safety regulator whose reward for working is that nothing goes wrong — is uniquely vulnerable to being ended on the grounds that nothing has gone wrong, when nothing has gone wrong precisely because it is working. Sunset as an adaptive act requires distinguishing the institution that has completed its coupling from the one that is silently maintaining it, and the two can look identical from outside. The discipline the section asks for is the same reorientation in a final key: a governance institution should not ask "how do I survive?" but "does my continued form still serve the coupling I was created to maintain?" — and should be built so that both a yes and a no can be acted on honestly.

7. The unifying structure

The three separations are one pattern seen at three points in a controller's life. At each, a bounded controller can measure something about itself — the fidelity of its model, the sophistication of its update rules, the fact of its own continuation — and mistake that internal measure for the external thing it exists to produce, which is a maintained coupling between some part of the world and some valued outcome. Governing well across a lifecycle is the discipline of not making the substitution, three times, in three registers.

The first substitution is fidelity for coupling (§§2–3). The controller improves its map and reads the improvement as adaptation, when adaptation is a property of the loop that acts on the map, not of the map. The minimal model shows the two coming apart mechanically: past the point where revision outruns absorption, the map sharpens while the grip fails, and every internal signal reports success. The second substitution is rule-sophistication for adequacy (§§4–5). The controller can improve the rule that improves its rule, and infers that it should keep climbing, when a bounded system's ladder must close on invariants and the adaptive act is choosing what to hold still rather than what to refine further. The third substitution is survival for purpose (§6). The controller reasons about how to persist when the question is whether its persistence still serves the coupling — and its terminal adaptive act may be to end, or to hand the coupling on.

Underneath the three sits the map/territory distinction Paper 0 grounds, now made dynamic. Paper 0 established that a factorization is a map, non-unique and answerable to the territory only at the behavioral boundary. This paper is what that distinction becomes over a lifecycle: the map can improve while the coupling to the territory degrades (§3), the rules for revising the map must rest on something not itself revised (§5), and the map-maker's own continuation is not the territory's need (§6). Each separation is a way the map can be confused with the territory once time and change are in the picture — a better map taken for a better grip, an endless refinement of map-making taken for progress, the map-maker's survival taken for the map's purpose.

The Ashby-shock and certification-cost results of Paper XX are what drive the sequence and tie it to the rest of the series. A rising task-relevant variety (the Ashby shock) forces the controller to learn — to revise its factorization to cover distinctions the world now demands. Learning generates the revealed and created demand of §2, which the loop must absorb or lose coupling. Keeping the factorization matched to a moving world carries the monotone certification cost of Paper XX, paid in increments or deferred into closure debt. And when the cost of maintaining the coupling under the current architecture can no longer be paid within the invariants — when the shock has outrun what this controller can become without dissolving — the lifecycle reaches §6, and the adaptive question is whether to refactor or to end. The lifecycle is what a bounded controller's confrontation with a changing world looks like when it is drawn out over the controller's whole life rather than examined at an instant.

8. What this re-grounds, and what it does not show

8.1 Re-grounding

Four of the series' results sit differently once the lifecycle is in view. Paper XIV's stable-learning requirement is the absorptive-capacity limit of §2 seen at the identification level: a controller re-identifying its own model while regulating must keep the revision stream within what the loop can absorb, which is why stable learning is a constraint and not merely a preference. Paper XV's adaptation triad is the architecture §3's model instantiates — sense feeds learn, learn feeds execute, and the demonstration is precisely a case where learn succeeds and execute cannot keep up. Paper XVI's source terms are what learning reveals in §2: the newly-visible mismatch a sharper model surfaces, which is information and obligation at once. And Paper XIX's role triad is the plural reserve of §5 — the sentinels and bridges held out of current use are the invariant a self-modifying controller most depends on, the stock it draws on when an Ashby-shock forces its factorization to expand.

8.2 What this paper does not show

The learning/adaptation dissociation is demonstrated in one environment and one operationalization of incorporation cost (§3). What is established is that learning can break coupling by outrunning absorptive capacity — not that it must in every setting, nor that the specific interior peak generalizes beyond this model. A first operationalization failed the registered prediction and was revised before re-registration; the demonstrated claim is the revised one, and its scope is a single model, not a law.

The meta-learning closure (§4) is a structural argument, not a demonstration. That a bounded hierarchy of learning rules cannot regress indefinitely follows from finiteness, and is [R] at that level; that closure requires the particular five invariants of §5 is [IP], an identification of governance-relevant anchors rather than a proof that these five are necessary and sufficient. The paper deliberately did not build a second minimal model for closure-versus-regress: a clean one proved elusive, and a forced one would have illustrated rather than tested, which the series' discipline declines.

Sunsetting (§6) is normative and is marked so throughout. It is a claim about when ending or transferring a coupling is the adaptive act, and a simulation could illustrate the mission-completion and loss-of-capacity conditions but could not settle the normative question of when an institution *should* end — that depends on a judgment about which couplings are worth maintaining, which is outside what the model represents.

And the whole lifecycle reading of institutions is [IP]. That a ministry, a treaty body, or a regulator instantiates a factorization, an absorptive capacity, a hierarchy of amendment rules, and a mission-coupled reason to exist is argument by analogy from bounded controllers. The one demonstrated result is about a controller tracking a drifting target; its reach to governance is the strength of that analogy and no more.

9. Method and confidence

Tiers follow the series: **[R]** rigorous, **[IP]** in principle, **[H]** heuristic, with **[R within the model]** marking a result exact for the stated model and claimed no further. This paper is primarily conceptual, with a single registered simulation — the learning/adaptation demonstration of §3 — in `paper_xxi-learning_adaptation_preregistration.md` and `paper_xxi-learning_adaptation_demo.py`, thresholds and nulls fixed before the run, including the first operationalization's failure kept on record. The explorations behind the arguments (`10-learning-versus-adaptation.md`, `11a-evolving-update-rules.md`, `11b-sunsetting.md`) are the source of the framing and are cited as argument, not evidence.

Confidence by result:

Result	Tier	Note
Learning/adaptation are distinct (§2)	[R] for the separation, [IP] for governance	Definitional; the demo makes it mechanical
Learning can break coupling by outrunning absorptive capacity (§3)	[R within the model]	30 registered seeds; one environment; a revised operationalization
Adaptive/irrelevant/maladaptive taxonomy (§2.3)	[IP]	Sorts learning by effect on the loop
Bounded meta-learning hierarchy must close (§4)	[R]	Follows from finiteness of representation
Closure requires these five invariants (§5)	[IP]	Governance-relevant anchors, not a necessity proof
Sunsetting as adaptive act; Type I / Type II (§6)	[IP]	Normative design claim, not demonstrated
The three separations as one pattern (§7)	[IP]	The unification; the paper's throughline
All institutional readings	[IP]	Argument by analogy from bounded controllers

Appendix A — The learning/adaptation model

The demonstration of §3 tracks a controller against a drifting latent target θ_t. Each step: the target drifts, $\theta_{t+1} = \theta_t + \mathrm{drift} \cdot \mathrm{sign} + \mathcal{N}(0, \sigma^2)$, with the sign reflecting when $|\theta|$ exceeds a bound to keep the walk bounded. The controller holds an estimate $\hat{\theta}$ updated toward a noisy observation at learning rate η: $\hat{\theta} \leftarrow \hat{\theta} + \eta(\mathrm{obs} - \hat{\theta})$, with $\mathrm{obs} = \theta + \mathcal{N}(0, \sigma^2)$. Each revision of size $|\Delta\hat{\theta}|$ injects incorporation disruption into the action loop, accumulating and decaying as $\mathrm{disrupt} \leftarrow \mathrm{decay} \cdot \mathrm{disrupt} + \mathrm{cost} \cdot |\Delta\hat{\theta}|$ — this is D_{created}. A second-order actuator chases the estimate, $\mathrm{vel} \leftarrow \mathrm{vel} + \mathrm{stiff}(\hat{\theta} - a) - \mathrm{damp} \cdot \mathrm{vel}$, $a \leftarrow a + \mathrm{vel}$, and its effective position is perturbed by current disruption, $a_{\mathrm{eff}} = a + \mathcal{N}(0, \mathrm{disrupt})$: the loop cannot settle while still absorbing revisions.

Fidelity is $-\overline{|\hat{\theta} - \theta|}$ and coupling is the fraction of steps with $|a_{\mathrm{eff}} - \theta| \leq \tau$, both over the post-burn-in window. Fixed configuration: drift 0.02, noise $\sigma = 0.12$, viability band $\tau = 0.25$, stiffness 0.08, damping 0.20, incorporation cost 1.5, disruption decay 0.85, 4000 steps with 500 discarded, 30 seeds. Swept: learning rate $\eta \in \{0.02, 0.05, 0.1, 0.2, 0.4, 0.7\}$. Pure numpy/matplotlib; the whole sweep runs in seconds on a CPU.

Registered outcomes: P1 (fidelity monotone in η) passed, pooled Spearman 0.99. P2 (coupling non-monotone, interior peak, $\geq 24/30$ seeds with a peak-to-highest drop ≥ 0.10) passed, peak at $\eta = 0.10$, 30/30. P3 (fidelity up while coupling down above the peak) passed, 30/30. Median coupling across the sweep: $0.24, 0.39, 0.48, 0.43, 0.29, 0.19$; median fidelity monotone from -0.60 to -0.08. Figure `learning_adaptation_scissors.png` shows fidelity rising while coupling peaks and falls.