Repair, Not Reconstruction: On the Structure of Reinforcement-Learning Adaptation in Transformers

Flyxion
Independent Researcher

July 2026

Abstract

A recent empirical study shows that training a single transformer layer under reinforcement learning, while every other parameter is frozen, recovers most of the improvement achieved by full-parameter training, and does so consistently for layers concentrated in the middle of the network's depth. This article treats that finding as a case study of a more general distinction: a system's nominal freedom to be modified is not the same as its effective freedom to be improved. After stating the paper's results at their actual strength and identifying what they do and do not establish, a repair-set formalism is introduced that expresses this distinction mathematically, independent of any account of why it holds. A three-level interaction hierarchy is proposed to test whether repairs found at different layers compose, obstruct, or jointly reach solutions unavailable to either alone. A separate, explicitly speculative section then proposes one candidate mechanism, grounded in prior work on residual-stream refinement and transformer interpretability, for why repair-compatible directions might concentrate at intermediate depth, together with a falsifiable test computable from a pretrained checkpoint alone. The formalism is constructed to survive independently of this mechanism, whatever the test's outcome.

1 Introduction

A system arriving at post-training is already organized: its parameters encode a functioning, if imperfect, solution produced under constraints different from those imposed during subsequent reinforcement learning. It is natural to assume that improving such a system under reinforcement learning is a matter of adjusting this organization more or less uniformly — that every parameter, having contributed to the pretrained solution, is equally available to contribute to its improvement. A recent empirical study of large language models gives strong reason to reject that assumption. Training a single transformer layer under reinforcement learning, while every other parameter remains frozen, recovers most of the improvement achieved by training the entire model, and in a number of cases exceeds it [1]. This is true not for an isolated model or task, but across seven models, two model families, three reinforcement learning algorithms, and task

domains as different as competition mathematics and multi-step embodied decision-making. The layers responsible are not scattered arbitrarily through the network; they concentrate, reliably and reproducibly, in the middle of its depth.

A system's nominal freedom to be modified is not the same as its effective freedom to be improved. One manifestation of this distinction is that the parameters a system presently depends upon for its behavior need not coincide with those that provide useful directions for adaptation.

This article treats the paper's findings as a case study of that general distinction, rather than as a fact to be explained away or absorbed into existing accounts of layer importance. The paper under discussion demonstrates the distinction with unusual clarity, because it is precise enough to quantify, and because its authors take care to rule out the most obvious deflationary explanation: the layers that matter most for adaptation are not simply the layers that move the most.

What follows takes the paper's findings as fixed and asks two separate questions of them. The first is interpretive: once the paper's results are stated at their actual strength, without extension, what distinction do they establish, and what mathematical language expresses that distinction without importing claims the evidence does not support? The second is explanatory: what property of a pretrained transformer's internal organization might account for why effective repair directions appear to concentrate at intermediate depth, and can that account be tested cheaply enough, and independently enough of the paper's own results, to be worth taking seriously rather than merely plausible? The two questions are kept deliberately separate throughout, since conflating a description of a phenomenon with a proposed mechanism for it is among the more common errors in work of this kind, and the argument that follows is constructed so that an answer to the second question, however it turns out, leaves the answer to the first intact.

2 The Empirical Result, Stated at Its Actual Strength

Zhang et al. train a single transformer layer under reinforcement learning while freezing every other parameter in the network, and compare the resulting model against one trained by ordinary full-parameter reinforcement learning [1]. Across seven models spanning two model families, three reinforcement learning algorithms, and three task domains, they report that a single layer trained in isolation recovers most of the performance gain achieved by full-parameter training, and in a number of cases exceeds it. The layers that recover the most improvement are not randomly distributed across depth. They concentrate reliably in the middle third of the network, while layers near the input and output ends recover substantially less. This concentration is stable under three independent tests of generalization: across training datasets within a fixed model and task, across a change of task domain from mathematical reasoning to multi-step agentic decision-making, and across a simultaneous change of model family and optimization algorithm.

Two further results in the paper narrow what this concentration can mean. First, the magnitude of parameter change under full-parameter training is approximately uniform

across layers, while the magnitude of change under single-layer training is likewise approximately uniform across layers regardless of which layer is trained. Since neither training regime shows middle layers moving further in parameter space than other layers, the concentration of contribution cannot be explained by middle layers simply receiving larger updates. Second, layers that achieve comparable aggregate performance when trained in isolation do not converge on the same solution. Evaluated on OlympiadBench, the seven highest-contribution layers of Qwen3-1.7B-Base show only thirty-four percent mean pairwise overlap in the specific problems each newly solves relative to the untrained base model, and a majority vote across these seven layer-trained models outperforms both the best individual layer and a matched-size ensemble drawn from repeated sampling of the full-parameter model.

What the paper does not establish is equally important to state plainly. Single-layer training demonstrates that a layer is sufficient to carry most of the reinforcement learning gain when every other parameter is held fixed; it does not demonstrate that the same layer is the primary locus of adaptation when all parameters are free to move jointly, since freezing the rest of the network changes the optimization problem being solved. The paper offers no measurement of the geometric properties of the parameter subspaces that different layers occupy — no assessment of dimension, curvature, or volume — and its own authors describe the underlying explanation for the middle-layer concentration as an open question. The result is therefore precise as a claim about where useful adaptation is *reachable* under a specific constrained training regime, and it should not be read past that without further argument.

3 What Kind of Result Is This?

Before introducing any formal apparatus, it is worth asking what kind of finding this is, independent of the language in which it might eventually be expressed. The paper is not, at bottom, a contribution to reinforcement learning efficiency, nor is it best read as a claim about where a specific cognitive function such as reasoning resides. Its central contribution is an existence proof of a distinction that is easy to state and easy to overlook: the set of parameters whose modification is *necessary* for a system's forward-pass behavior is not the same set as the parameters whose modification is *useful* for improving that behavior under a training signal.

Prior work on layer importance in large language models has largely addressed the first set. Studies that remove or ablate individual layers and observe the resulting collapse in performance are measuring representational necessity: which components the forward pass currently depends on. The present paper addresses a different question entirely. It does not ask which layers the model currently needs in order to function; it asks which layers, when given the opportunity to change, are capable of absorbing a training signal to good effect. These two properties could in principle coincide, and much prior work implicitly assumes that adaptation is distributed across the network in rough proportion to representational importance. The paper's finding is that this assumption is false for reinforcement learning post-training: some of the layers most re-

sponsible for the model's current competence are not the layers best suited to improving it further, and a comparatively narrow, depth-stable band of the network accounts for nearly all of the recoverable gain.

Four more specific distinctions follow from this reframing and will recur throughout the remainder of this article. Forward-pass importance is not the same quantity as adaptive usefulness, since a layer may be indispensable to current behavior while offering a poor subspace for improving that behavior, or dispensable to current behavior while offering an unusually good one. The magnitude of a parameter's displacement during training is not the same as the usefulness of that displacement, since the paper's own weight-change measurements show comparable movement across layers of very different contribution. The location at which useful adaptation is concentrated is not the same as the content of what gets learned there, since layers of nearly identical aggregate contribution solve substantially different subsets of problems. And sufficiency under a constrained training regime, in which most of the network is held fixed, is not the same as necessity under an unconstrained one, in which every parameter is free to move; the paper establishes the former directly and the latter only by inference.

It is worth stating plainly why the first of these distinctions should be expected to hold rather than treating it as a surprising coincidence. Forward-pass execution answers the question of which parameters participate in producing the model's present behavior. Reinforcement learning answers a different question: which parameter changes most efficiently alter that behavior while preserving the rest. These are not the same optimization problem, and there is no theoretical reason they should identify the same regions of parameter space. A parameter can be heavily implicated in the current computation while offering, geometrically, a poor direction for improving that computation under a gradient-based training signal; equally, a parameter that participates only lightly in current behavior may nonetheless sit in a favorable position for absorbing a useful update. Once this is granted, the paper's central finding stops looking like an anomaly to be explained away and starts looking like the expected outcome of two genuinely different questions being conflated by the assumption of uniform trainability.

Taken together, these distinctions suggest that the appropriate frame for interpreting the paper is not one of locating a cognitive function within the network, but one of characterizing how a system that already embodies a coherent, trained organization responds differently to modification at different points within itself. That framing does not by itself require any new mathematical apparatus. It does, however, motivate one: if the useful question is not "which layer matters" but "which modifications at which layers improve behavior without unraveling the organization the network already has," then the natural next step is to give that latter notion — modification that improves without unraveling — a name and a working definition. That is the task of the section that follows.

4 The Repair-Set Formalism

The distinctions drawn in the preceding section can be given a concrete mathematical expression without committing to any account of why they hold. Let the full parameter space of a transformer with L decoder layers be decomposed into layer-indexed blocks,

$$\Theta = \Theta_0 \times \Theta_1 \times \cdots \times \Theta_{L-1},$$

where this decomposition is adopted solely because it matches the parameter groupings Zhang et al. train and freeze, rather than because it is assumed to identify the transformer's natural computational units. Let $\theta^{(0)} \in \Theta$ denote the pretrained parameter point at which reinforcement learning begins. Throughout, J is written as a loss to be minimized; an equivalent formulation for maximized reward follows by reversing the inequality below. For a task objective J, define a local repair set at tolerance ε and preservation budget δ as

$$\mathcal{R}_{\varepsilon,\delta} = \left\{ \Delta\theta : J\Big(\theta^{(0)} + \Delta\theta\Big) \leq J\Big(\theta^{(0)}\Big) - \varepsilon, \;\; D_{\text{pres}}\Big(\theta^{(0)} + \Delta\theta,\; \theta^{(0)}\Big) \leq \delta \right\},$$

where D_{pres} is a damage function measuring disturbance to capabilities and structures that should remain stable, and whose content is deliberately left unspecified for now. The repair set collects every perturbation that improves task performance by at least ε while damaging the pretrained organization by no more than δ. Restricting attention to a single layer gives the layer-specific repair set,

$$\mathcal{R}^{(k)}_{\varepsilon,\delta} = \mathcal{R}_{\varepsilon,\delta} \cap \{\Delta\theta : \Delta\theta_j = 0 \text{ for } j \neq k\},$$

the set of admissible repairs reachable by moving layer k alone.

This apparatus does no more than translate the paper's own experimental design into a common notation, but the translation is useful because it separates two quantities that ordinary language tends to run together. A modification of the network is not evaluated here by how large it is, nor by whether it targets a layer the forward pass currently depends on; it is evaluated by whether it lies in a set defined jointly by improvement and preservation. The paper's central empirical finding, in this language, is that $\mathcal{R}^{(k)}_{\varepsilon,\delta}$ differs sharply in the density of empirically reachable repairs across k, and that this difference tracks depth in a stable way despite tracking neither raw representational necessity nor the magnitude of the displacement itself. The word "repair" is used throughout in this technical sense — a modification satisfying the joint improvement-and-preservation criterion — and not in the sense of restoring the network to some earlier or intended state; a repair, here, adds a capability the pretrained model lacked while leaving the rest of its organization largely undisturbed, rather than correcting a defect against a prior standard. The term preferred throughout this article for what a given layer offers is *admissible direction* rather than any term implying a specific geometry, such as a low-dimensional channel or subspace. The paper's evidence supports the weaker claim: some layers make available directions that satisfy the joint improvement-and-preservation criterion more readily than others. It does not measure the dimension, curvature, or volume of the

sets containing those directions, and adopting more specific geometric language would claim more than the experiments show.

The formalism also clarifies why localized repair can outperform unrestricted joint training, a pattern the paper reports directly: on several model scales, training only the highest-contribution layers exceeds the performance of training every layer at once. Full-parameter training searches over $\mathcal{R}_{\varepsilon,\delta}$ as a whole, a much larger nominal space than any single $\mathcal{R}^{(k)}_{\varepsilon,\delta}$. If enlarging the movable set of parameters simply enlarged the reachable set of good solutions, joint training could not be outperformed by training a strict subset of the network. That it sometimes is outperformed indicates that the additional degrees of freedom available under joint training are not uniformly useful, and that permitting movement in layers whose repair sets are poor, small, or absent can degrade the outcome relative to leaving those layers fixed. The negative result at layer zero of Qwen3-8B-Base is a sharpened instance of the same pattern: training that layer in isolation does not merely fail to help, it reduces performance below the untrained base model, indicating that, under the tested optimization regime, the observed optimization trajectory for that layer remained outside the empirically useful repair set despite improving the training objective locally.

4.1 The Interaction Hierarchy

The repair-set formalism raises a further question that the paper's single-layer experiments cannot by themselves answer: whether repairs found at different layers are compatible with one another when combined. Three distinct versions of this question should be kept separate, since they are answered by different experiments and license different conclusions.

The first asks whether repairs obtained independently, under the isolation protocol the paper already uses, compose without loss when summed. Given deltas $\Delta\theta_i^{\text{iso}}$ and $\Delta\theta_j^{\text{iso}}$ obtained by training layers i and j separately, and a scalar improvement measure G relative to the pretrained model, define

$$I_{ij}^{\text{iso}} = G\big(\Delta\theta_i^{\text{iso}} + \Delta\theta_j^{\text{iso}}\big) - G\big(\Delta\theta_i^{\text{iso}}\big) - G\big(\Delta\theta_j^{\text{iso}}\big)\,.$$

A negative value indicates that the two isolation-derived repairs interfere when applied together. This quantity is inexpensive to obtain, since it requires only applying existing checkpoint deltas in combination rather than any new training run.

The second asks the corresponding question of deltas as they actually arise during joint training, rather than deltas obtained under artificial isolation. Given the updates $\Delta\theta_i^{\text{joint}}(t)$ and $\Delta\theta_j^{\text{joint}}(t)$ that layers i and j undergo at a given point in a joint training trajectory, the analogous quantity

$$I_{ij}^{\text{joint}}(t) = G\Big(\Delta\theta_i^{\text{joint}}(t) + \Delta\theta_j^{\text{joint}}(t)\Big) - G\Big(\Delta\theta_i^{\text{joint}}(t)\Big) - G\Big(\Delta\theta_j^{\text{joint}}(t)\Big)$$

measures interaction along the optimization path the network actually follows when both layers are free to move, which need not resemble the isolation-derived deltas at all.

The third and broadest question is whether jointly training layers i and j reaches solutions unavailable to either isolated search or to any combination of isolated endpoints:

$$\mathcal{R}_{\varepsilon,\delta}^{(i,j)} \overset{?}{\supsetneq} \mathcal{R}_{\varepsilon,\delta}^{(i)} \oplus \mathcal{R}_{\varepsilon,\delta}^{(j)},$$

where $\oplus$ denotes a candidate compositional construction and is not asserted to be linear or otherwise structurally simple. Several concrete candidates are available for what $\oplus$ might mean in practice, and the choice among them is itself an empirical question rather than a notational detail: the set of all convex combinations $\{\alpha\Delta\theta_i + (1-\alpha)\Delta\theta_j : \alpha \in [0,1]\}$, the single elementwise sum $\Delta\theta_i + \Delta\theta_j$ already used in I_{ij}^{iso}, or the set reachable by sequential application of the two isolated repair procedures, training layer i to convergence and then layer j from that point. These need not coincide, and part of what the reachable-set question asks is which, if any, of these constructions approximates what joint training actually finds. This question cannot be settled by delta arithmetic alone, since it concerns the reachable set under co-adapted optimization rather than the properties of any fixed pair of deltas, and answering it requires comparing actual jointly trained models against every interpolation and composition of the corresponding isolated repairs.

These three levels are not substitutes for one another. A negative I_{ij}^{iso} demonstrates only that isolation-derived repairs do not combine cleanly; it says nothing about whether joint training would discover compatible repairs by a different route. A positive $I_{ij}^{\text{joint}}(t)$ throughout training would show that co-adaptation is not destructive along the realized path without showing that the isolated repairs themselves were ever compatible. Only the third question addresses whether the additional freedom of joint training expands the effective repair set beyond what independent, single-layer search can reach, which is the question the paper's own comparison between full-parameter and selective training ultimately turns on.

5 What Could Determine the Shape of These Repair Sets?

Everything argued to this point follows from the paper's own measurements and requires no further assumption about the internal workings of the network. The question of *why* the repair sets $\mathcal{R}_{\varepsilon,\delta}^{(k)}$ differ across depth in the particular way they do is a separate matter, and answering it requires a mechanistic hypothesis that goes beyond what has been established so far. This section develops one such hypothesis. It is presented as a conjecture throughout, and the repair-set formalism of the preceding section does not depend on it: should the account offered here prove incorrect, the observations, the terminology, and the interaction hierarchy remain intact.

5.1 Three Levels of Quantities

Before proposing an explanation, it is worth being explicit about the epistemic status of the quantities involved, since interpretability work is prone to a specific error at this juncture: treating an inferred explanatory variable as though it had been measured. Three

levels should be distinguished. Layer contribution $C(k)$, as defined by the paper, is an observed quantity, computed directly from evaluation scores under a specified training protocol. The repair set $\mathcal{R}^{(k)}_{\varepsilon,\delta}$ is a derived quantity, a mathematical object introduced in the preceding section to give a name to what $C(k)$ is implicitly measuring; it is not itself observed, but it is determined once J, D_{pres}, ε, and δ are specified, with membership in it settled by evaluation rather than by any further assumption. Task-abstraction availability $A(k)$ and preservation-compatible plasticity $P(k)$, introduced below, are latent explanatory quantities: theoretical constructs proposed to account for why $\mathcal{R}^{(k)}_{\varepsilon,\delta}$ differs across k, which have not yet been operationalized and whose relationship to anything observed remains to be established. These three levels should not be conflated. In particular, no experiment reported in the paper measures $A(k)$ or $P(k)$ in any form; any claim about their behavior in what follows is a prediction to be tested, not a restatement of existing evidence.

5.2 Iterative Refinement and the Product Hypothesis

The claim that transformer layers refine rather than replace their input representation has an architectural precedent outside language models specifically. Jastrzębski et al. show that residual connections naturally encourage the features produced by a residual block to move along the negative gradient of the loss with respect to the block's input, formalizing the intuition that residual architectures perform iterative refinement of a shared representation rather than constructing a sequence of unrelated ones [4]. This is the architectural intuition borrowed here: that a transformer's residual stream is best understood as a single representation undergoing successive, bounded updates, rather than as a chain of independent transformations. Their result is about residual networks generally and establishes only that this refinement dynamic exists as a consequence of the residual connection itself, not any claim about where in a network's depth a particular kind of update becomes more or less available.

Interpretability work specific to transformers extends this picture empirically rather than architecturally. The logit lens shows that when the hidden states at each layer of a transformer are decoded through the model's own output embedding, the resulting predictions converge, roughly monotonically, toward the model's final output as depth increases [2]. The tuned lens refines this observation by fitting a learned affine correction at each layer, substantially improving the reliability of this intermediate decoding across model families [3]. The contribution of this line of work is evidence that a transformer's internal representations evolve coherently across depth — that intermediate layers can be read as increasingly confident, increasingly output-like partial predictions — rather than evidence about which layers are most amenable to modification under training. Neither tool was designed to, nor has been used to, measure plasticity or repair-compatibility of any kind.

Neither the residual-architecture literature nor the logit lens and tuned lens literature predicts that reinforcement-learning plasticity should peak at intermediate depth. The former argues that representations are refined progressively through residual computation; the latter provides methods for observing that refinement empirically in trans-

former models. Neither addresses, and neither should be read as addressing, the question of where in that refinement process a parameter update is most likely to improve behavior without unraveling the model's inherited organization. The present hypothesis combines these observations with the empirical findings of the reinforcement-learning study under discussion to propose — not derive — that repair-compatible update directions become most abundant at intermediate depth: early enough that the residual stream still has substantial refinement ahead of it and can absorb a perturbation, but late enough that the representation being perturbed is no longer tightly bound to token-level surface form. This is offered as one candidate explanation motivated by, but not established by, the cited literature, and its status as a conjecture rather than an extension of prior results is the reason Section 5.4 proposes a test capable of falsifying it independently of everything argued in Sections 2 through 4.

This suggests a schematic decomposition,

$$C(k) \propto A(k)\, P(k),$$

where $A(k)$ denotes the availability of task-relevant abstraction at layer k and $P(k)$ denotes preservation-compatible plasticity — the extent to which a change at that layer can be absorbed by the remainder of the network without destabilizing its inherited organization. Neither factor need be unimodal across depth for their product to peak at an intermediate layer: $A(k)$ may rise monotonically with depth through some interior region before saturating, while $P(k)$ may fall monotonically once representations become committed to output decoding, and the product of a rising and a falling quantity peaks between them without either quantity itself doing so. This form is deliberately schematic. It should be read as a proposal about which two properties might jointly govern the shape of $\mathcal{R}^{(k)}_{\varepsilon,\delta}$, not as an empirically fitted law, and Sections 5.3 and 5.4 address directly what would be required to treat it as more than a plausible narrative. The multiplicative form itself is not theoretically privileged; it is adopted only as the simplest separable hypothesis against which more complex interactions between $A(k)$ and $P(k)$ may later be compared.

5.3 Operationalizing the Preservation Budget

The repair-set formalism left D_{pres} unspecified, noting only that it must measure disturbance to whatever should remain stable under repair. The mechanism proposed above suggests a specific decomposition,

$$D_{\text{pres}} = \lambda_{\text{beh}} D_{\text{beh}} + \lambda_{\text{rep}} D_{\text{rep}} + \lambda_{\text{int}} D_{\text{interface}},$$

where D_{beh} measures degradation on capabilities that the modification was not intended to affect, D_{rep} measures the divergence of internal hidden-state representations from those of the pretrained model, and $D_{\text{interface}}$ measures disturbance specifically to the input-encoding and output-decoding conventions the rest of the network depends upon. This decomposition is motivated entirely by the iterative-refinement account and should not be mistaken for a mechanism-independent extension of the repair formalism: $D_{\text{interface}}$

presupposes that input encoding and output decoding are the two loci whose disturbance matters most, which is precisely the claim the mechanism proposes rather than one that follows from the formalism alone. The weighting coefficients λ_{beh}, λ_{rep}, and λ_{int} introduce a further difficulty: if they are chosen so as to make D_{pres} track $C(k)$ well, the resulting agreement is manufactured rather than discovered. Any use of this decomposition therefore requires fixing the weights by a criterion independent of the layer-contribution results it might later be compared against. It is worth being precise about what independence requires here: the out-of-distribution benchmark categories already reported in the paper — Code, Reasoning, and Language — are themselves components of the overall layer-contribution score $C_{\text{all}}(k)$, and calibrating against them would not be independent of $C(k)$ in the relevant sense, whatever the original rationale for selecting those categories. A defensible calibration requires benchmarks that play no role in any reported $C(k)$ at all, held out specifically for this purpose.

5.4 A Falsifiable Base-Model Test

The strongest feature of this proposal is that both $A(k)$ and $P(k)$, as latent quantities, admit operationalizations computable from the pretrained model alone, prior to and independent of any reinforcement learning run. Task-abstraction availability can be approximated by the accuracy of a linear probe trained to decode task-relevant variables — such as intermediate reasoning quantities or eventual answer correctness — from the hidden states at layer k of the untrained base model. Preservation-compatible plasticity can be approximated by the sensitivity of the base model's output to a small, fixed-norm perturbation injected into the residual stream at layer k, with higher downstream sensitivity indicating lower plasticity in the relevant sense. Because both measurements use only the publicly available pretrained checkpoints already employed in the paper, they require no new training and no access to the reinforcement learning infrastructure used to produce $C(k)$.

Both operationalizations carry a specific limitation worth stating rather than leaving implicit. Linear-probe accuracy measures linearly decodable information, which is a narrower notion than task-relevant abstraction in general; a layer could carry abstraction that only a nonlinear probe would reveal, in which case $A(k)$ as operationalized here would understate it. Perturbation sensitivity, similarly, could be elevated for reasons unrelated to plasticity in the intended sense — an unusually large activation norm or an unstable local dynamic at a given layer could produce high measured sensitivity independent of whether that layer offers good repair directions. Neither limitation is fatal to the test, but both mean that a positive result should be read as consistent with the proposed mechanism rather than as uniquely confirming it.

The resulting test has an unusually clean structure for a mechanistic hypothesis in this area. The pretrained checkpoint is obtained; $A(k)$ is measured by linear probing; $P(k)$ is measured by perturbation sensitivity; the product $A(k)P(k)$ yields a predicted ordering of layer contribution; and this ordering is compared against the $C(k)$ values the paper already reports. A close correspondence would indicate that a property of the pretrained model's geometry, measurable before any reinforcement learning takes

place, forecasts which layers will prove useful for reinforcement learning adaptation — a nontrivial and useful finding in its own right. Even a strong correlation would establish only predictive value, not causal determination. It would remain possible that $A(k)$, $P(k)$, and $C(k)$ are jointly driven by some further property of pretrained-model geometry — such as activation norm or attention-pattern entropy — that this article does not attempt to measure. A poor correspondence would falsify the specific mechanism proposed here without in any way disturbing the empirical result the paper reports or the repair-set language used to describe it, since neither depends on the iterative-refinement account being correct. This asymmetry — an inexpensive test capable of falsifying the mechanism while leaving the formalism and the underlying data untouched — is what recommends the hypothesis for inclusion here, whatever its eventual outcome.

6 Discussion

The interpretation offered in the preceding section can be stated more generally than it has been so far. Reinforcement learning post-training does not begin from an undifferentiated parameter space in which every direction of movement is equally available. It begins from a pretrained point that already embodies a coherent, functioning organization — one that solves a great many problems tolerably well before any reward signal is applied. Optimization from that point is not free exploration of the full parameter space; it takes place inside a pre-existing organization, and the character of that organization determines, layer by layer, which modifications are available to be found at all. Some directions of movement improve the target behavior while leaving the rest of the organization largely intact. Others improve the same measured objective while unraveling structure the objective does not explicitly track, and the two kinds of direction are not distinguishable from magnitude of movement alone, as the paper's own weight-change measurements show directly.

Once stated this way, the finding is not primarily a fact about transformers, layer depth, or reinforcement learning specifically, though it is established here only for that particular case. It is an instance of a more general distinction: the effective space of improvement available to a system is not the same as the formal space of modification technically open to it. A system with millions of nominally free parameters may nonetheless offer only a narrow, unevenly distributed, and structurally located set of changes that count as genuine improvement rather than disruption dressed as improvement. This appears to instantiate the same distinction that separates a repair from a reconstruction in any domain where a system's prior organization has value independent of the objective currently being optimized — in software maintenance, where a change that passes the immediate test suite can still degrade the architecture it modifies; in evolutionary biology, where most mutations available to an organism are not viable regardless of how large or small they are; in the rehabilitation of damaged systems generally, where restoring function is a different problem from constructing function from nothing. The paper under discussion offers a clean, quantified instance of this pattern inside a system precise and inspectable enough to measure it directly, which is what makes it useful as a

case study beyond its immediate subject matter.

It is worth being clear about what this discussion does and does not claim to have shown. It does not claim that middle transformer layers are where reasoning or any other named cognitive capacity resides, a claim the paper's authors themselves decline to make. It does not claim that the mechanism proposed in Section 5 is correct; that mechanism is offered as a testable conjecture precisely because its correctness is not yet known. What the discussion does claim is narrower and better supported: that the paper's central empirical result is best read as evidence for a distinction between formal and effective modification spaces, that this distinction can be given a precise mathematical expression independent of any account of why it holds, and that the same distinction plausibly recurs wherever a training or design process modifies a system whose prior organization was not built for the purpose of being modified.

7 Limitations and Proposed Experiments

Several limitations should be stated directly rather than left implicit. The repair-set formalism developed in Section 4 formalizes what the paper's protocol already measures; it does not by itself extend the paper's evidence to any new regime, and the interaction hierarchy it motivates has not been run. The base-model probe proposed in Section 5.4 has likewise not been run, and no claim in this article should be read as reporting its outcome. The preservation budget D_{pres}, even before any mechanism-specific decomposition is applied to it, remains without a validated operational definition; every quantitative claim involving δ in this article is illustrative rather than measured. Finally, the account offered in Section 5 treats depth as the operative variable, following the paper's own presentation, but depth is confounded in a transformer with several other properties — residual-stream norm, attention-pattern entropy, and training dynamics that differ across layers for reasons unrelated to abstraction — any of which could turn out to be doing the explanatory work attributed here to abstraction and preservation-compatibility. This article also operates entirely at the level of layer blocks and does not engage the separate literature on feature-level interpretability within a layer, such as sparse-autoencoder work identifying individual interpretable directions within a layer's representation. That literature addresses a different question — which features exist at a given depth, rather than which layer-blocks offer good sites for reinforcement-learning adaptation — and integrating the two would require establishing a relationship between feature-level and layer-block-level admissibility that this article does not attempt.

Three experiments follow directly from the argument above and are proposed as the next stage of this case study rather than as results already obtained. The first is the falsifiable base-model test of Section 5.4: measuring $A(k)$ by linear-probe decodability and $P(k)$ by perturbation sensitivity on the publicly available pretrained checkpoints, and comparing the predicted ordering against the $C(k)$ values the original paper reports. The second is the static-composability half of the interaction hierarchy, I_{ij}^{iso}, computable directly from the paper's own checkpoint deltas without any new training run, applied across pairs drawn from both high- and low-contribution layers to test whether

the compatibility of isolated repairs itself varies systematically with contribution. The third, requiring new training, is a small set of two-layer and multi-layer joint training runs with intermediate checkpoints logged, permitting $I_{ij}^{\text{joint}}(t)$ to be measured along a realized optimization trajectory and compared against both the isolated interaction term and the reachable-set question posed in Section 4.1. Together these three experiments would test the mechanism proposed here, the formalism's own internal coherence, and the relationship between constrained and unconstrained search that motivated the repair framework in the first place — each capable of failing independently of the others, which is the property that makes them worth running.

References

[1] Zijian Zhang, Rizhen Hu, Athanasios Glentis, Dawei Li, Chung-Yiu Yau, Hongzhou Lin, and Mingyi Hong. Is one layer enough? Training a single transformer layer can match full-parameter RL training, 2026. arXiv:2607.01232.

[2] nostalgebraist. Interpreting GPT: the logit lens, 2020. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

[3] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023. arXiv:2303.08112.

[4] Stanisław Jastrzębski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. In *International Conference on Learning Representations (ICLR)*, 2018. arXiv:1710.04773. `https://openreview.net/forum?id=SJa9iHgAZ`.