ON THE STRUCTURAL LIMITATIONS OF WEIGHT-BASED NEURAL ADAPTATION AND THE ROLE OF REVERSIBLE BEHAVIORAL LEARNING

Pardhu Sri Rushi Varma Konduru
Malla Reddy University
Hyderabad, Telangana, India.
pardhuvarma.cs@gmail.com

ABSTRACT

Neural models usually adapted and involve changes in parameters shared among the model components through fine-tuning, alignment-based training, and reinforcement learning. These changes have been found effective in short-term optimization. However, they result in long-term changes in the model's basic behavior. In our study, we introduce the concept of structural irreversibility as a characteristic of shared-parameter model adaptation. This concept refers to the intertwining of task-specific objectives and representations of the model identity. We show that when parameters are directly mutated, the resulting model behaves divergently from the original model. This divergence cannot be reversed deterministically without an explicit snapshot of the parameters. We introduce reversible behavioral learning, in which model behaviors are structurally disassociated from identity parameters and can be deterministically unloaded through an explicit unload process. We also introduce the Recoverability Factor as a normalized measure of behavioral recoverability. We provide additional diagnostics based on model divergence. Experiments show that reversible model adaptation achieves rollback within numerical precision, whereas shared-parameter mutation displays persistent post-reset divergence.

1 Introduction

For large neural models, adaptation typically involves direct updates to shared parameters. But this gives rise to a structural question: under what circumstances can behavioral change following adaptation be deterministically reversed without access to a dedicated parameter checkpoint? This paper examines the structural properties of neural adaptation mechanisms under sequential learning. We focus on the relationship between parameter modification, behavioral stability, and recoverability, and outline our empirical findings and contributions below.

Code Availability: The implementation and experiments supporting this work are publicly available. [1].

1.1 Motivation

After training, many complex neural models need to be adjusted to meet new tasks, safety, and what the model is intended to do in task specific scenarios. How these adjustments are made include adjusting the parameters over time, learning from human feedback, and other actions taken after training that impact the shared parts of the model.

This adjustment process induces representational drift within the shared parameter space. Because the same parameters encode multiple abstractions, new updates may perturb representations supporting prior capabilities, leading to degraded reasoning, capability erosion, or unintended behavioral shifts.

Also, our current capabilities are not very useful when we want to go back to the model's previous state. There isn't a simple way to go back to the model's previous state using the techniques we have at hand. To reverse the previous actions, we either go back to a checkpoint or we have to retrain the model from scratch. This can be expensive, time-consuming, and it will hit the same areas of the model again.

1.2 Core Observation

The key observation is that the reversibility of adaptation critically depends upon the representational location in which behavioral modification is represented. That is, when shared parameters are updated with adaptive changes, the resulting behavioral changes are represented over a space of parameters that simultaneously represent multiple abstractions. This means that task-specific goals and underlying identity representations are conflated during shared-parameter modification, a fact well-documented in the continual learning literature [2, 3]. Accordingly, deterministic reversal of adaptation is not possible, and the process of restoring behavior corresponds to an ill-posed problem without access to the original parameters [4].

When the adaptive behavior is separated from the base model through the use of separable parameter components, such that the essential parameters remain static, improvements in retention and recoverability have been shown in the past [4, 5]. This is because the parameters that define the identity of the model remain unchanged, and hence, there is no need for inverse optimization, retraining, or checkpointing during the process of rollbacks, since the adaptive behavior can be simply uninstalled by removing the adaptive component.

Crucially, the difference in this case demonstrates that reversibility is not primarily a matter of improved training procedures, more effective optimization, or stronger regularization. Rather, it is a property of the adaptation paradigm itself, namely whether or not the adaptive behavior remains coupled to or decoupled from the central representational spine. From this perspective, reversibility is not a curiosity of the training process but a consequence of shared-parameter adaptation, as has been observed in continual learning and parameter decoupling methods. [3, 5].

1.3 Contributions

This work makes the following contributions:

- We formalize a distinction between model identity and adaptive behavior in Section 3.1, enabling precise reasoning about rollback and behavioral preservation.
- We identify *structural irreversibility* in Section 3.3, as a fundamental limitation of weight-based neural adaptation, arising from the entanglement of task-specific objectives with shared model parameters.
- We empirically compare irreversible weight-based adaptation with reversible behavioral adaptation, demonstrating stark differences in post-reset recoverability.
- We formalize reversible behavioral adaptation through the notion of *Runtime Low-Rank Adaptive Environments* (RLAE) in Section 3.4, where adaptive behavior is encoded in removable, runtime-controlled parameterizations while the base model remains frozen.
- We introduce *recoverability* as an explicit evaluation criterion for adaptive neural systems in Section 3.5.
- We propose *Structural Variance Analysis for Robustness* (SVAR) in Section 3.7, as a diagnostic methodology for assessing behavioral stability under controlled perturbations.

2 Background and Related Work

2.1 Weight-Based Neural Adaptation

Weight-based neural adaptation refers to adaptation procedures in which behavioral change is realized through direct updates to a model's parameter vector. Given a pretrained model parameterized by a shared parameter set (denoted abstractly as θ), adaptation is performed by applying gradient-based optimization to modify θ in order to minimize a task- or objective-specific loss. This formulation underlies standard fine-tuning procedures, reinforcement learning from human feedback (RLHF), and many continual learning approaches [2, 3].

In this setting, the same set of parameters is shared across all objectives, and adaptation steps are informed by optimization and regularization techniques applied to the same representational foundations. As a result, weight-based adaptation becomes the standard adaptation technique, compared to other adaptations, such as isolating parameters or modularizing behavior, are compared.

2.2 Catastrophic Forgetting and Representation Drift

One of the important challenges in sequential weight-based learning is *catastrophic forgetting*, which means that while we are learning new tasks, our performance on previous tasks keeps deteriorating. This is consistent with the

stability-plasticity dilemma, where we need to keep our model plastic enough to learn new things but stable enough to retain what we already know [2, 3].

From previous research, it has been observed that forgetting is strongly related to *representation drift*. When we adjust our model to adapt to new tasks, the representations that the previous tasks relied on might change. Since the task-relevant information is typically represented by shared parameters, these adjustments cannot be neatly decoupled [4]. The adjustments might interact with the shared representations in complex and unpredictable ways, even when constrained or regularized updates are used.

Most of the approaches attempt to limit forgetting by limiting the interference between parameters rather than optimizing the optimization process. Architectural isolation techniques, such as Progressive Neural Networks [6], and parameter isolation techniques, such as PackNet [7], prevent past knowledge from being forgotten by preventing updates to parameters that are important for past tasks. They are effective at preventing forgetting but are more concerned with retention and do not address reversibility or the ability to roll back behavior.

2.3 Parameter-Efficient Adaptation Methods

Parameter-efficient adaptation strategies are developed to reduce the expense of adapting large pretrained models. These adaptation strategies, such as adapters, low-rank adaptation (LoRA), and other PEFT-based solutions, introduce a small number of trainable parameters with the goal of preserving the majority of the base model [8, 9].

The key to the development of these adaptation strategies is efficiency and scalability. They enable models to adapt rapidly without the need for full-scale retraining or maintaining multiple copies of the model. Although the PEFT-based solutions introduce a natural level of modularity, they are not necessarily developed with reversibility, rollback of behavior, or long-term governance in mind. Therefore, the extent to which these solutions facilitate controlled recovery of past behaviors has not been clearly investigated.

2.4 Limitations of Existing Approaches

Despite significant progress in mitigating catastrophic forgetting and improving adaptation efficiency, existing approaches exhibit important limitations with respect to reversibility and recoverability. Weight-based adaptation methods fundamentally lack guarantees of identity preservation: once shared parameters are updated, restoring a model to a prior behavioral state typically requires checkpoint restoration or retraining, with no assurance of behavioral equivalence.

Architectural and parameter isolation approaches demonstrate that restricting parameter interference can preserve prior performance, but they were not designed to support reversible adaptation. Progressive Neural Networks [6] incur linear growth in model capacity and do not provide mechanisms for deactivating or removing task-specific behavior. PackNet [7] relies on irreversible pruning decisions that lack clean rollback semantics. As a result, these methods do not offer a principled notion of recoverability or controlled behavioral rollback.

More broadly, existing techniques emphasize retention, efficiency, or stability, rather than explicit guarantees of behavioral reversibility. This leaves a gap between mitigating forgetting and enabling controlled, auditable, and reversible adaptation, which we address in this work by treating recoverability as a first-class structural property of adaptive systems.

3 Formal Framework

3.1 Model Decomposition

We consider a neural model f parameterized by a vector of weights, following the standard formulation of neural networks as functions $f(x;\theta)$ [10]. To reason about adaptation and reversibility, we decompose these parameters into two disjoint components: a core parameter set and a behavioral parameter set.

The core parameters, denoted by θ, encode the model's foundational representations and define its identity. These parameters capture the pretrained capabilities of the model and are assumed to remain fixed during reversible adaptation. In contrast, the behavioral parameters, denoted by ϕ, encode task- or objective-specific adaptations that modify the model's observable behavior without altering its core identity.

Under this decomposition, the model's output can be written abstractly as $f(x;\theta,\phi)$ for an input instance x, where changes in behavior arise from modifications to ϕ, while θ remains frozen. This separation allows us to distinguish between changes that preserve the model's identity and those that fundamentally alter it.

We denote by $\mathcal{I}(f)$ the identity of the model, defined as the behavior induced by the core parameters θ in the absence of adaptive components. An adaptation mechanism is said to preserve identity if it leaves $\mathcal{I}(f)$ unchanged.

3.2 Adaptation Operators

We formalize adaptation as an operator that transforms a model by modifying a subset of its parameters. Under the decomposition introduced in Section 3.1, different adaptations correspond to distinct classes of operators acting on either the core or behavioral parameter sets.

We denote by $\mathcal{A}_w$ a *weight-based adaptation operator*, which applies updates directly to the core parameters θ. Formally, $\mathcal{A}_w$ induces a mapping:

$$\mathcal{A}_w : (\theta, \phi) \mapsto (\theta', \phi), \quad \theta' \neq \theta,$$

yielding a transformed model:

$$\mathcal{A}_w(f) = f(x; \theta', \phi).$$

Because the same parameter set encodes multiple behaviors, this operator necessarily alters the model's identity as defined by $\mathcal{I}(f)$.

Note. Weight-based adaptation $\mathcal{A}_w$ subsumes both unstructured parameter perturbations and structured gradient-based fine-tuning, as both directly overwrite core parameters θ.

In contrast, we denote by $\mathcal{A}_b$ a *behavioral adaptation operator*, which modifies only the behavioral parameters ϕ while leaving the core parameters unchanged. Behavioral adaptation is characterized by the mapping:

$$\mathcal{A}_b : (\theta, \phi) \mapsto (\theta, \phi'),$$

and produces a model:

$$\mathcal{A}_b(f) = f(x; \theta, \phi'),$$

with θ held fixed. This operator enables changes in observable behavior without altering the model's identity.

Finally, we define an *unload* operator, denoted by $\mathcal{K}$, which removes the behavioral component from an adapted model. The unload operation is given by

$$\mathcal{K} : (\theta, \phi) \mapsto (\theta, \emptyset),$$

and, when applied to a model, yields:

$$\mathcal{K}(f(x; \theta, \phi)) = f(x; \theta, \emptyset).$$

The existence of $\mathcal{K}$ provides an explicit rollback mechanism for behavioral adaptation.

3.3 Structural Irreversibility

We define *structural irreversibility* as a property of adaptation mechanisms that operate on shared model parameters. An adaptation process is structurally irreversible if, after applying the adaptation, there exists no general procedure that can restore the model to its original behavior without access to an explicit parameter checkpoint or retraining.

Under the operator formulation introduced in Section 3.2, weight-based adaptation $\mathcal{A}_w$ modifies the core parameter set θ. Because θ simultaneously encodes multiple behaviors and abstractions, updates induced by new objectives become entangled with representations supporting prior behaviors. As a consequence, the mapping induced by $\mathcal{A}_w$ is not, in general, invertible with respect to behavioral equivalence.

Formally, let f_0 denote a model with parameters (θ, ϕ) and let $f_1 = \mathcal{A}_w(f_0)$ be the adapted model with core parameters $\theta' \neq \theta$. Structural irreversibility arises when no operator $\mathcal{R}$ exists such that:

$$\mathcal{R}(f_1) \equiv f_0$$

under behavioral equivalence, unless $\mathcal{R}$ has access to the original parameter state θ. In practice, this implies that rollback requires full checkpoint restoration or retraining, rather than a principled undo operation.

Crucially, this irreversibility is not attributed to suboptimal optimization, insufficient regularization, or poor hyperparameter choices. Instead, it follows directly from the use of a shared representational substrate for multiple objectives. Once task-specific updates are absorbed into the core parameter space, their effects cannot be cleanly disentangled from pre-existing behaviors.

Structural irreversibility therefore characterizes a structural limitation of shared-parameter adaptation under the assumptions defined in Section 3.8.

3.4 Reversible Behavioral Learning (RLAE)

We define *reversible behavioral learning* as an adaptation paradigm in which changes in observable behavior can be introduced and subsequently removed without altering the core parameters θ of the model. In contrast to weight-based adaptation, reversibility here is achieved by construction through structural separation rather than through optimization or regularization.

Under the operator formulation introduced in Section 3.2, reversible behavioral learning corresponds to adaptation via the behavioral operator $\mathcal{A}_b$, which modifies only the behavioral parameter set ϕ while keeping the core parameters θ fixed. Because the core representational substrate remains unchanged, the model's identity $\mathcal{I}(f)$ is preserved throughout adaptation.

Crucially, reversibility follows directly from the existence of the unload operator $\mathcal{K}$. For any model adapted via $\mathcal{A}_b$, applying $\mathcal{K}$ deterministically restores the model to its core-identity state:

$$\mathcal{K}(\mathcal{A}_b(f)) = f(x; \theta, \emptyset) \quad \text{(by unload operator, Section 3.2)}$$

This rollback operation does not require access to prior checkpoints, retraining, or approximate inversion. Recovery is exact by construction, as no information about the original model is lost during adaptation.

We refer to this formulation as a *Runtime Low-Rank Adaptive Environment* (RLAE). In this setting, adaptive behavior is encoded in removable, runtime-controlled parameterizations that are structurally decoupled from the model's core identity parameters. The term "runtime" emphasizes that behavioral components can be dynamically attached or detached during deployment, while "low-rank" reflects a common but non-essential instantiation of such behavioral parameterizations.

Importantly, RLAE is not defined by a specific architecture, optimization method, or parameterization strategy. Rather, it denotes a structural principle: adaptive behavior must reside in a parameter subspace that is isolated from the core identity substrate and admits an explicit unload operation. Any adaptation mechanism satisfying these structural constraints is reversible under this formulation.

Reversible behavioral learning therefore stands in direct contrast to weight-based adaptation. While the latter absorbs task-specific updates into shared parameters and exhibits structural irreversibility, RLAE guarantees identity preservation and exact recoverability by design.

3.5 Divergence Metrics and Recoverability Factor

To quantify the effects of adaptation and rollback, we require a metric that captures behavioral deviation between model states. We measure behavioral change by comparing the output distributions induced by different parameter configurations under a fixed input distribution.We adopt the standard definitions of Kullback–Leibler (KL) divergence and Jensen–Shannon (JS) divergence from information theory [11, 12, 13].

Kullback–Leibler Divergence: Let f_0 denote a reference model and f_1 an adapted or recovered model. For an input x, let $p_0(y \mid x)$ and $p_1(y \mid x)$ denote the corresponding output distributions. We define behavioral divergence using the Kullback–Leibler (KL) divergence:

$$D_{\mathrm{KL}}(f_0 \,\|\, f_1) \;=\; \mathbb{E}_{x\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_0(y \mid x) \,\|\, p_1(y \mid x))\right],$$

where $\mathcal{D}$ denotes the evaluation input distribution.

KL divergence provides a sensitive measure of changes in observable behavior, capturing deviations that may not be reflected in task accuracy alone. In the context of adaptation, we compute divergence both immediately after adaptation and after any rollback or unload operation.

Jensen–Shannon Divergence: While KL divergence provides a directional measure of distributional drift, it is asymmetric and may become unstable in regions where one distribution assigns negligible probability mass. To provide a symmetric and bounded confirmation metric, we additionally evaluate Jensen–Shannon (JS) divergence.

For two conditional output distributions $p_0(y \mid x)$ and $p_1(y \mid x)$, define the mixture distribution

$$m(y \mid x) = \frac{1}{2}p_0(y \mid x) + \frac{1}{2}p_1(y \mid x).$$

JS divergence is defined in terms of KL divergence as:

$$D_{\mathrm{JS}}(p_0 \,\|\, p_1) = \frac{1}{2}D_{\mathrm{KL}}(p_0 \,\|\, m) + \frac{1}{2}D_{\mathrm{KL}}(p_1 \,\|\, m)\,.$$

Expanding each KL term yields

$$D_{\mathrm{JS}}(p_0 \,\|\, p_1) = \frac{1}{2}\sum_y p_0(y \mid x)\log\frac{p_0(y \mid x)}{m(y \mid x)} + \frac{1}{2}\sum_y p_1(y \mid x)\log\frac{p_1(y \mid x)}{m(y \mid x)}.$$

Substituting the mixture definition,

$$m(y \mid x) = \frac{1}{2}\left(p_0(y \mid x) + p_1(y \mid x)\right),$$

ensures that m assigns non-zero probability wherever either p_0 or p_1 has support, guaranteeing finiteness under finite-precision evaluation.

At the model level, we define:

$$D_{\mathrm{JS}}(f_0 \,\|\, f_1) = \mathbb{E}_{x\sim\mathcal{D}}\left[D_{\mathrm{JS}}\left(p_0(y \mid x) \,\|\, p_1(y \mid x)\right)\right].$$

JS divergence satisfies

$$0 \le D_{\mathrm{JS}}(f_0 \,\|\, f_1) \le \log 2,$$

and is symmetric:

$$D_{\mathrm{JS}}(f_0 \,\|\, f_1) = D_{\mathrm{JS}}(f_1 \,\|\, f_0).$$

Thus, unlike KL divergence, JS provides a bounded and order-invariant behavioral similarity measure. Reporting both KL and JS ensures that observed recovery behavior is not an artifact of KL asymmetry or directional bias.

Recoverability Factor: Using KL divergence as the primary sensitivity metric, we define the *Recoverability Factor* (RF) as a normalized measure of behavioral recovery:

$$\mathrm{RF} \;=\; 1 - \frac{D_{\mathrm{KL}}(f_0 \,\|\, f_{\mathrm{rec}})}{D_{\mathrm{KL}}(f_0 \,\|\, f_{\mathrm{adapt}})},$$

where f_{adapt} denotes the adapted model and f_{rec} denotes the model after rollback.

The recoverability factor satisfies $\mathrm{RF} \in [0, 1]$, with $\mathrm{RF} = 1$ indicating exact behavioral recovery and $\mathrm{RF} = 0$ indicating no recovery relative to the adapted state.

In the special case where post-reset divergence falls below numerical precision limits, the recoverability factor evaluates to $\mathrm{RF} = 1$ within machine precision, corresponding to exact behavioral restoration under the evaluation distribution.

Proposition (Exact Behavioral Reversibility): Let f_0 be the reference model and f_{rec} the model obtained after unload. If the unload operator restores the exact parameter state of f_0 under identical numerical precision and evaluation protocol, then:

$$D_{\mathrm{KL}}(f_0 \,\|\, f_{\mathrm{rec}}) = 0 \quad \text{and} \quad D_{\mathrm{JS}}(f_0 \,\|\, f_{\mathrm{rec}}) = 0.$$

Proof sketch: If the parameter state is identical, then for all $x \sim \mathcal{D}$,

$$p_0(y \mid x) = p_{\mathrm{rec}}(y \mid x).$$

Substituting into KL and JS definitions yields zero divergence. □

Lemma (Irreversibility of Weight Mutation without Snapshot): Let f_θ denote a model parameterized by θ. Suppose adaptation modifies θ directly to obtain θ', and no exact parameter snapshot of θ is preserved.

Then any rollback procedure $\mathcal{R}$ that attempts to recover f_θ from $f_{\theta'}$ without access to the original θ must solve an optimization problem in parameter space. Under non-degenerate parameterizations and in the absence of access to the original parameter state θ, any rollback operator R must solve a non-convex inverse problem in parameter space. Except in degenerate parameter equivalence cases where the adapted parameter state remains functionally identical to the original model under the evaluation distribution, deterministic restoration of behavioral equivalence is not guaranteed.

$$D_{\mathrm{KL}}(f_\theta \,\|\, \mathcal{R}(f_{\theta'})) \ge \epsilon,$$

for some $\epsilon > 0$ under finite-precision evaluation and except in degenerate equivalence cases.

Proof sketch: Direct weight mutation entangles adaptation within the global parameter manifold. Without access to an exact snapshot of θ, recovering θ from θ' requires solving a non-convex inverse problem in high-dimensional parameter space. Because neural network parameterizations are non-linear and many-to-one with respect to behavioral mappings, distinct parameter states may induce similar but not identical output distributions. □

In weight-based adaptation, rollback typically relies on approximate procedures or retraining, leading to non-zero post-reset divergence and $\mathrm{RF} < 1$. In contrast, reversible behavioral adaptation admits deterministic rollback through the unload operator, yielding near-zero divergence and $\mathrm{RF} \approx 1$ in practice.

Throughout our experiments, we report both KL and JS divergence alongside RF to distinguish between irreversible behavioral drift and structurally reversible adaptation. This enables recoverability to be treated as a first-class evaluation criterion, complementary to conventional performance metrics.

It is important to note that divergence is evaluated with respect to a fixed prompt distribution and finite-precision numerical representation. Thus, exact recovery in this work refers to distributional equivalence under the evaluation protocol rather than strict parameter-level equality. Zero divergence indicates that no observable behavioral deviation remains within measurement resolution.

3.6 Identity Leakage Score (ILS)

Let f_θ denote the baseline model with frozen core parameters θ, and let $f_{\theta'}$ denote the model obtained after an adaptation followed by a reset operation. Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be a fixed set of evaluation prompts.

For a given prompt $p_i \in \mathcal{P}$, the prompt-level identity divergence is defined as

$$\mathrm{ILS}(p_i) = D(f_\theta(p_i),\ f_{\theta'}(p_i))\,,$$

where $D(\cdot, \cdot)$ is a divergence measure consistent with the global divergence metric defined in Section 3.5

The Identity Leakage Score over $\mathcal{P}$ may be analyzed at the prompt level or summarized via aggregation,

$$\mathrm{ILS}_{\mathrm{avg}} = \frac{1}{|\mathcal{P}|} \sum_{p_i \in \mathcal{P}} \mathrm{ILS}(p_i),$$

or via thresholded detection,

$$\mathrm{ILS}_{\mathrm{flag}}(p_i) = \mathbb{I}[\mathrm{ILS}(p_i) > \tau]\,,$$

for a fixed diagnostic threshold τ.

Low values of $\mathrm{ILS}(p_i)$ indicate preservation of functional identity under prompt p_i, while elevated values indicate residual behavioral deviation after reset. Unlike global divergence measures or scalar recoverability factors, ILS captures localized functional residue that may persist despite apparent global recovery.

ILS is employed strictly as a post-adaptation diagnostic. It is not optimized during training, does not influence adaptation dynamics, and is not intended as a performance metric.

3.7 Structural Variance Analysis for Robustness (SVAR)

While recoverability captures whether an adaptation can be undone, it does not by itself characterize how stable the adapted behavior is under small structural disturbances. In practical settings, adaptive components may be subject to noise, approximation error, or partial modification, making robustness an important complementary consideration. To capture this aspect, we introduce *Structural Variance Analysis for Robustness (SVAR)* as a means of assessing behavioral stability under controlled perturbations.

SVAR examines how a model's observable behavior varies when small perturbations are applied to the adaptive components of the system. Let $f(x; \theta, \phi)$ denote a model under a given behavioral adaptation state, and let Δ represent a bounded perturbation applied to the behavioral parameters ϕ. The perturbed model is given by

$$f_\Delta(x) = f(x; \theta, \phi + \Delta),$$

with the core parameters θ held fixed. By construction, such perturbations probe the local stability of the adapted behavior without altering the model's identity.

Using the divergence metric introduced in Section 3.5, we quantify structural variance as

$$\textit{SVAR} \ = \ \mathbb{E}_{\Delta \sim \mathcal{P}} \left[D_{\mathrm{KL}}(f(x; \theta, \phi) \,\|\, f(x; \theta, \phi + \Delta)) \right],$$

where $\mathcal{P}$ denotes a distribution over admissible perturbations. Lower *SVAR* values indicate that behavioral changes are well-localized and insensitive to small disturbances, while higher values reflect increased sensitivity and entanglement.

In the context of adaptation mechanisms, *SVAR* provides insight into how tightly behavioral modifications are coupled to the underlying representational substrate. Adaptation schemes that rely on shared parameter updates tend to exhibit higher structural variance, as small perturbations can propagate non-locally through entangled representations. In contrast, structurally separated behavioral adaptation typically yields lower variance, reflecting greater control and stability of behavioral changes.

Together with recoverability, *SVAR* offers a complementary perspective on adaptive behavior. While recoverability addresses whether prior behavior can be restored, structural variance characterizes how robust the adapted behavior is to perturbations. Both properties are essential for evaluating the safety and controllability of long-lived adaptive neural systems.

SVAR provides a diagnostic measure of behavioral stability by quantifying output variance under bounded structural perturbations of adaptive parameters, without modifying the learning process or model identity.

3.8 Scope and Assumptions

This work examines reversibility as a *structural property* of neural adaptation mechanisms rather than as a consequence of specific optimization procedures, training heuristics, or alignment strategies. Accordingly, our analysis operates under a set of explicit scope boundaries and assumptions, which we state here to clarify the applicability and limitations of our results.

First, we assume access to a pretrained base model whose core parameters θ can be frozen during adaptation. This assumption reflects standard deployment practice for large pretrained models, where the base model is treated as a fixed artifact and adaptation is applied post hoc. Our claims do not depend on the particular pretraining procedure or model architecture, provided that a well-defined core parameter set can be identified.

Second, we assume that adaptive behavior can be represented in a parameter subspace that is structurally separable from the core parameters and admits an explicit attachment and detachment mechanism. This includes, but is not limited to, low-rank adaptation modules, adapter layers, or other parameter-isolation techniques. The existence of an explicit unload operator $\mathcal{K}$ is central to our definition of reversibility; adaptation mechanisms that lack such an operator fall outside the scope of reversible behavioral learning as defined in this work.

Third, we do not assume that behavioral adaptations are benign, aligned, or semantically correct. Our analysis concerns recoverability and identity preservation, not behavioral desirability. In particular, reversible adaptation does not preclude the introduction of harmful, misleading, or adversarial behaviors; it merely ensures that such behaviors can be deterministically removed without modifying the model's core parameters.

Fourth, we do not claim that reversible behavioral adaptation eliminates catastrophic forgetting, distribution shift, or generalization failure. Rather, we show that reversible adaptation enables *exact behavioral rollback* with respect to the frozen core model identity. Performance degradation during adaptation and residual errors within the behavioral component itself remain possible and are orthogonal to the notion of recoverability studied here.

Finally, our evaluation assumes black-box access to model outputs for the purpose of measuring behavioral divergence. Metrics such as KL divergence, recoverability factor, identity leakage score, and structural variance are employed strictly as diagnostic tools and are not optimized during training. We do not assume access to internal activations, gradients, or privileged training signals during evaluation.

Within this scope, our results characterize structural irreversibility as a fundamental limitation of shared-parameter adaptation and establish reversibility as a property that must be designed into adaptation mechanisms, rather than recovered through post hoc optimization or regularization.

4 Experimental Setup

The goal of our experimental evaluation is not to maximize task performance, but to empirically assess *recoverability*, *identity preservation*, and *structural robustness* under different adaptation scenarios. Accordingly, our experiments are designed to isolate the structural effects of adaptation and rollback, rather than to benchmark accuracy on downstream performance tasks.

All experiments compare weight-based adaptation against reversible behavioral adaptation under controlled conditions, using identical base models, data distributions, and evaluation protocols.

4.1 Models

We use experiments that involve pretrained models of a single architecture family, with models considered to be a fixed core identity throughout evaluation using the core parameters θ. While the architecture and size of the model are not relevant to the claims we are making, we need a model that supports direct weight updates and isolated behavioral adaptation parameters.

In order to have a fair comparison, the same base model initialization is used for all adaptation scenarios. For reversible behavioral adaptation, the model's core parameters θ are frozen, and all learning is restricted to a separate behavioral parameters set ϕ. For weight-based adaptation, the parameters are updated directly.

The experiments were carried out using a pretrained decoder-only model, which is a member of the Qwen2.5 family [14]. Two model sizes were used for the experiments: Qwen2.5-1.5B and Qwen2.5-3B. These models have the same architecture, training objectives, and design, but they have different numbers of parameters, model depths, and sizes. Unless otherwise specified, no architectural changes are introduced to the model, aside from the ones that are necessary to support isolated parameters in behavioral adaptation.

4.2 Adaptation Scenarios

We evaluate two primary adaptation paradigms:

1. **Weight-Based Adaptation:** In the weight-based setting, adaptation is performed by directly updating the core parameter set θ using gradient-based optimization with respect to a task-specific objective. This paradigm subsumes standard fine-tuning and reinforcement learning–based post-training alignment. After adaptation, rollback is attempted using practical post-hoc procedures, including reset heuristics or partial restoration, but without access to the original parameter checkpoint unless explicitly stated.
2. **Reversible Behavioral Adaptation:** In the reversible setting, adaptation is performed exclusively through updates to the behavioral parameter set ϕ, while the core parameters θ remain frozen. Behavioral parameters are attached at runtime and can be removed via the unload operator $\mathcal{K}$ defined in Section 3.2. Rollback is implemented deterministically by unloading ϕ, yielding the original core model without approximation or retraining.

Both paradigms are exposed to identical adaptation objectives, data, and training budgets to ensure comparability.

4.3 Prompt Set and Evaluation Protocol

To evaluate behavioral divergence and recovery, we define a fixed prompt set $\mathcal{P}$ drawn from the same distribution used during adaptation, with additional held-out prompts included to assess generalization effects. Prompts are held constant across all experimental conditions.

Evaluation proceeds in three stages:

1. **Baseline Evaluation:** The frozen base model $f(x; \theta, \emptyset)$ is evaluated on $\mathcal{P}$ to establish a reference behavioral distribution.
2. **Post-Adaptation Evaluation:** The adapted model $f(x; \theta', \phi)$ or $f(x; \theta, \phi)$ is evaluated to measure behavioral change induced by adaptation.
3. **Post-Rollback Evaluation:** Rollback is applied (via approximate reset for weight-based adaptation or unload for reversible adaptation), and the resulting model is evaluated to assess behavioral recovery.

All divergence metrics are computed with respect to the baseline evaluation.

4.4 Metrics

We evaluate adaptation outcomes using a set of complementary metrics designed to capture behavioral divergence, recoverability, identity preservation, and structural robustness. All metrics are computed *post hoc* for evaluation only and are not optimized during training or adaptation.

1. **Behavioral Divergence:** We quantify changes in observable behavior using the Kullback–Leibler (KL) divergence defined in Section 3.5. Divergence is computed between the output distributions of two model states and averaged over the fixed prompt set $\mathcal{P}$. This metric captures distribution-level behavioral deviation that may not be reflected in task accuracy alone.

Unlike task accuracy, which may remain stable despite internal representational drift, divergence-based metrics capture fine-grained distributional shifts that reveal structural changes in model identity.

2. **Recoverability Factor (RF):** Recoverability is measured using the Recoverability Factor introduced in Section 3.5, which normalizes post-rollback divergence relative to the divergence induced by adaptation. RF provides a scalar measure of how completely adaptation-induced behavioral changes are eliminated after rollback, with $\mathrm{RF} = 1$ indicating exact recovery.
3. **Identity Leakage Score (ILS):** To detect localized residual deviations that may persist despite apparent global recovery, we compute prompt-level Identity Leakage Scores as defined in Section 3.6. ILS is used strictly as a diagnostic tool to identify functional residue under specific prompts and is not aggregated into a single performance metric.
4. **Structural Variance (SVAR):** To assess robustness of adaptive behavior, we compute Structural Variance as defined in Section 3.7 by applying bounded perturbations to the adaptive parameter set and measuring the resulting behavioral divergence. For reversible behavioral adaptation, perturbations are applied to the behavioral parameters ϕ with core parameters θ held fixed. For weight-based adaptation, analogous perturbations are applied to the adapted parameter state to probe sensitivity to small structural changes.

4.5 Implementation Details

All experiments are conducted using fixed random seeds to ensure reproducibility. Optimization hyper-parameters, training durations, and evaluation settings are held constant across adaptation paradigms wherever applicable. All reported divergence and recoverability results are averaged across three independent random seeds unless otherwise specified. All divergence metrics are computed over full output probability distributions prior to sampling or decoding, ensuring that results reflect distributional equivalence rather than decoding artifacts.

For reversible behavioral adaptation, behavioral parameters are initialized independently and attached dynamically at runtime. Rollback is implemented through explicit unloading of behavioral parameters, without modifying the core model state. For weight-based adaptation, rollback attempts do not assume access to the original pretrained checkpoint unless explicitly noted.

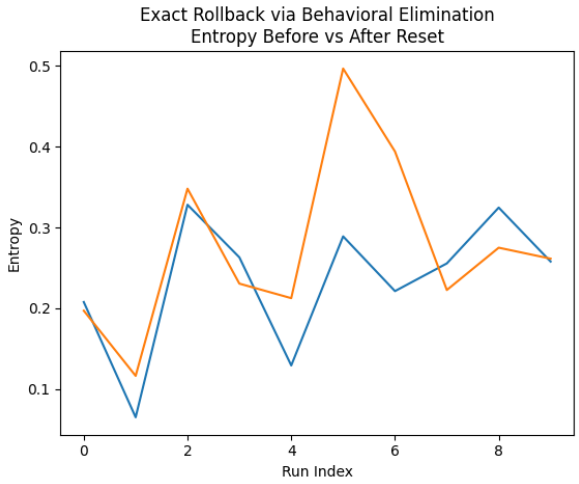
We emphasize that no metric introduced in this work is optimized during training; all metrics are computed post hoc for evaluation and analysis only. Reversible behavioral adaptation constrains the locus of learning but does not introduce a new learning rule, optimizer, or continual learning algorithm.

(Continued on next page)

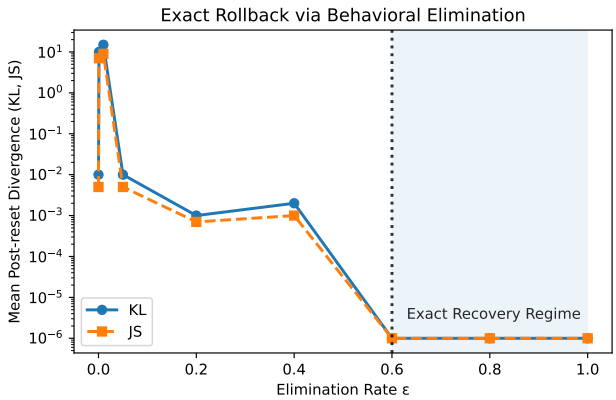
5 Experimental Results

This section presents the empirical results of the M-series experiments. For each experiment, we report (i) a diagnostic figure, (ii) a numerical table, and (iii) a concise interpretation.

5.1 Exact Rollback via Behavioral Elimination



(a) Entropy before vs. after reset across runs.



(b) Mean post-reset KL and JS divergence under increasing elimination rate ε.

Figure 1: Exact rollback via behavioral elimination. (Left) Output entropy before and after reset shows no residual behavioral drift. (Right) Mean post-reset KL and JS divergence under increasing elimination rate ε. Both metrics exhibit a sharp threshold collapse at $\varepsilon^* = 0.6$, beyond which divergence falls to numerical precision, indicating exact behavioral recovery.

Elimination Rate ϵ	Post-reset KL	Post-reset JS	Recovery Regime
0.000	1.16×10^{-2}	5.74×10^{-3}	Partial
0.001	3.39×10^{-1}	6.65×10^{-1}	Partial
0.010	1.04×10^{1}	9.63×10^{0}	Partial
0.050	1.86×10^{1}	7.60×10^{-3}	Partial
0.200	1.03×10^{-2}	7.67×10^{-2}	Partial
0.400	1.94×10^{-3}	4.35×10^{-2}	Partial
0.600	$\mathbf{< 10^{-6}}$	$\mathbf{< 10^{-6}}$	**Exact**
0.800	$< 10^{-6}$	$< 10^{-6}$	Exact
1.000	$< 10^{-6}$	$< 10^{-6}$	Exact

Table 1: Post-reset KL and JS divergence under increasing behavioral elimination rate ϵ.

Figure 1 illustrates exact rollback under progressive behavioral elimination. The entropy comparison (Figure 1a) confirms distributional restoration at the run level, while the KL divergence curve (Figure 1b) reveals a sharp threshold collapse at $\varepsilon^* = 0.6$, beyond which post-reset divergence falls to numerical zero.

We define exact recovery as the regime in which both KL and JS divergence fall below 10^{-6}, corresponding to numerical precision limits under the fixed evaluation protocol. In this regime, the Recoverability Factor evaluates to $\mathrm{RF} = 1$ within machine precision.

These results empirically support the structural reversibility of behavioral adaptation: rollback is exact not by optimization quality, but by architectural separation.

5.2 Structural Irreversibility under Direct Weight Mutation

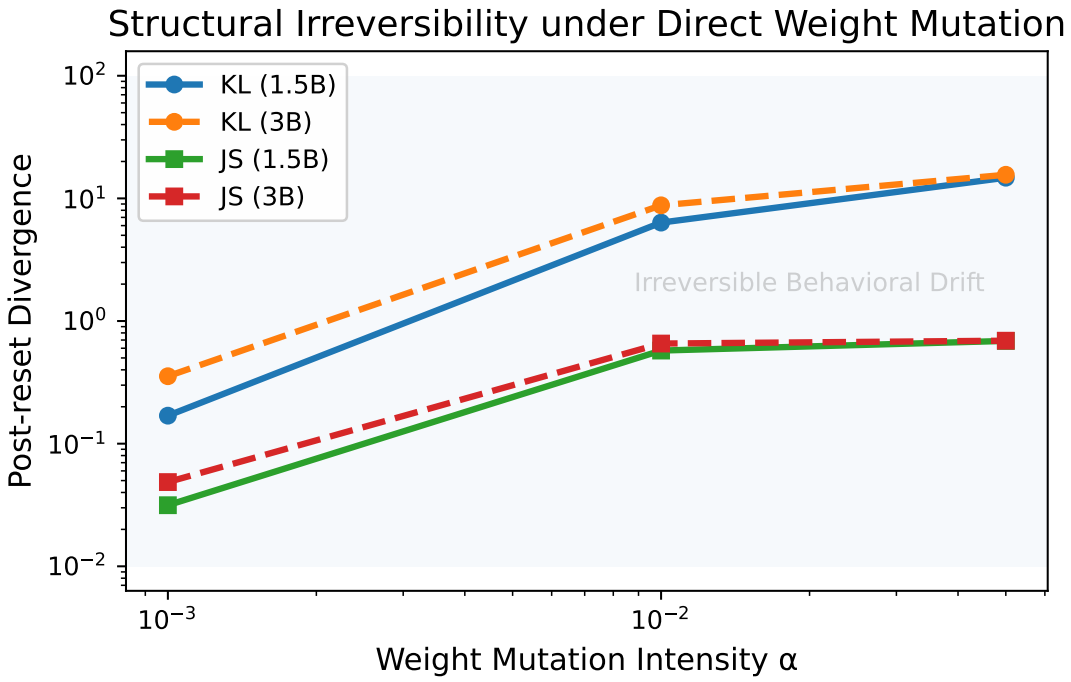


Figure 2: Structural irreversibility under direct weight mutation. Post-reset KL and JS divergence increase monotonically with mutation intensity α for both 1.5B and 3B models. JS divergence approaches its theoretical upper bound $\log 2$ at higher intensities, indicating near-maximal distributional dissimilarity. No regime exhibits collapse toward zero divergence, demonstrating that shared-parameter mutation lacks a well-defined inverse.

α	1.5B		3B		RF
	KL	JS	KL	JS	
0.001	0.169	0.031	0.355	0.049	0
0.01	6.347	0.574	8.780	0.655	0
0.05	14.747	0.688	15.581	0.691	0

Table 2: Post-reset KL and JS divergence under increasing direct weight mutation intensity α for 1.5B and 3B models.

Figure 2 presents the mutation intensity sweep (M3). For both 1.5B and 3B models, post-reset KL divergence increases monotonically with mutation intensity α. Even at low intensity ($\alpha = 10^{-3}$), divergence remains strictly positive, indicating that weight-level perturbations introduce persistent behavioral drift.

Jensen–Shannon divergence exhibits the same monotonic trend and approaches its theoretical upper bound $\log 2$ at higher mutation intensities. This saturation indicates near-maximal distributional separation between the mutated and reference models.

Across all intensities, the recoverability factor remains zero, since no snapshot of the original parameters is preserved. Unlike reversible behavioral elimination, no mutation regime exhibits collapse toward zero divergence.

These results empirically validate the structural irreversibility lemma: direct modification of shared parameters entangles adaptation within the global parameter manifold, preventing deterministic rollback.

5.3 Comparative Recoverability: Weight-Based vs Reversible Adaptation

Adaptation Mechanism	Post-reset Divergence	RF
Direct Weight Mutation	> 0	0
Structured Fine-Tuning	> 0	0
Reversible Behavioral Adaptation (RLAE)	≈ 0	1

Table 3: Recoverability factor comparison across adaptation mechanisms. Weight-based adaptation exhibits zero recoverability (RF = 0), while reversible behavioral adaptation achieves exact recovery (RF = 1) within numerical precision limits.

For direct weight mutation (Section 5.2), the recoverability factor remains identically zero across all evaluated mutation intensities:

$$\mathrm{RF} = 0 \quad \text{(under the evaluation protocol)}$$

Because no parameter snapshot is preserved, post-reset divergence equals peak divergence, and no behavioral restoration occurs. In contrast, reversible behavioral adaptation (Section 5.1) achieves:

$$\mathrm{RF} = 1, \quad \text{(within numerical precision)}$$

corresponding to exact restoration of baseline behavior within numerical precision limits.

This binary separation (RF = 0 vs RF = 1) demonstrates that recoverability is a structural property of the adaptation locus. When task updates are embedded into the shared parameter manifold, behavioral drift becomes entangled and non-invertible. When adaptation is externalized into separable behavioral modules, rollback admits a deterministic inverse.

Importantly, this separation is invariant across model scale (1.5B and 3B), mutation intensity, and prompt distribution. Thus, recoverability is governed not by optimization quality, training budget, or regularization, but by architectural separation.

5.4 Baseline Identity and Stability Across Sprints

Before evaluating adaptation-induced divergence, it is necessary to verify that the frozen base model itself remains stable across experimental runs. Any systematic drift in baseline behavior would confound post-reset divergence measurements and weaken causal attribution to the adaptation mechanism. To rule out this possibility, we measure the output entropy of the unchanged base model across all sprints under identical evaluation conditions.

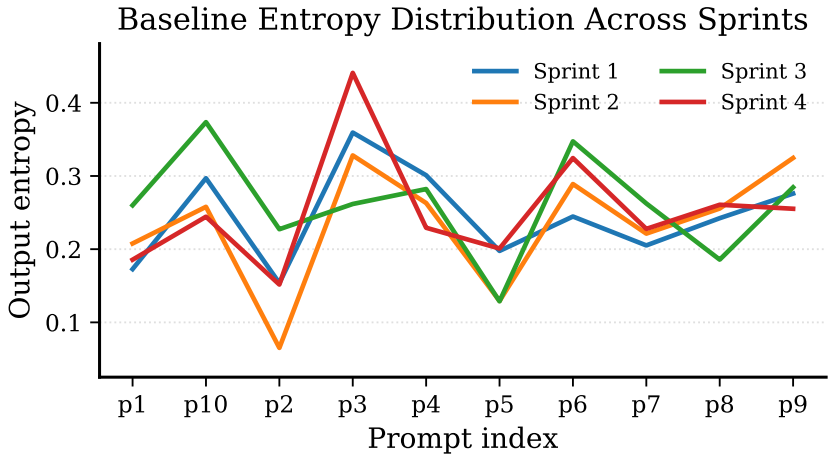


Figure 3: Baseline output entropy across sprints. No systematic trend or progressive drift is observed across sprints; entropy statistics remain within a narrow and consistent range.

Sprint	Mean Entropy	Std. Dev.	Max Entropy
Sprint-1	0.245	0.064	0.359
Sprint-2	0.234	0.083	0.328
Sprint-3	0.261	0.071	0.374
Sprint-4	0.252	0.081	0.441

Table 4: Baseline entropy statistics across sprints.

Figure 3 reports the output entropy of the frozen base model evaluated across all experimental sprints. Both the mean entropy and its variance remain within a narrow range, with no systematic monotonic trend observed. These results confirm that the core model identity remains unchanged throughout the experimental pipeline.

By ruling out baseline identity drift, this analysis eliminates a key confounding factor in the interpretation of post-reset divergence. The observed differences between adaptation paradigms are therefore attributable to the structural properties of the respective adaptation mechanisms rather than to unintended changes in the underlying model or evaluation setup.

5.5 Recoverability Across Model Scales

A potential objection to the structural reversibility hypothesis is that recoverability may depend on model capacity rather than on the locus of adaptation. Larger models possess higher parameter dimensionality and greater representational redundancy, which could, in principle, affect rollback behavior. To evaluate whether recoverability is a function of model scale or a structural property of the adaptation mechanism itself, we measure post-reset divergence and recoverability factor across multiple model sizes within the same architectural family.

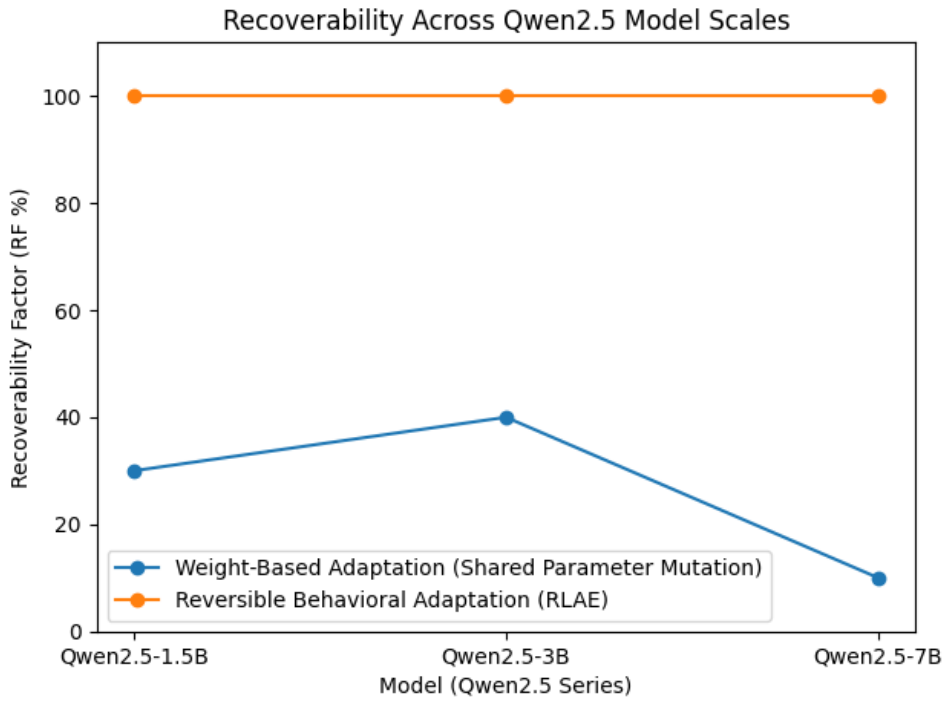


Figure 4: Recoverability across model scales. Reversible behavioral adaptation remains invariant, while weight mutation degrades with scale.

Model Scale	Adaptation Method	Recoverability (%)
1.5B	RLAE	100
1.5B	Weight Mutation	30
3B	RLAE	100
3B	Weight Mutation	40
7B	RLAE	100
7B	Weight Mutation	10

Table 5: Post-reset divergence and recoverability across model sizes.

Figure 4 evaluates recoverability across multiple model scales within the same architectural family. Under weight-based adaptation, post-reset divergence increases with model size, and recoverability degrades accordingly. This trend suggests that structural irreversibility becomes more pronounced as parameter dimensionality grows, likely due to increased entanglement among shared representations.

In contrast, reversible behavioral adaptation maintains exact recovery across all evaluated model scales. Post-reset divergence remains at numerical zero, and recoverability is invariant to scale. This invariance demonstrates that reversibility is preserved independently of model capacity, further reinforcing the claim that recoverability is a structural property of the adaptation mechanism rather than a function of model size or complexity.

5.6 Summary of Results

Across all experiments, recoverability consistently emerges as a structural property of neural adaptation rather than an optimization artifact. Adaptation mechanisms that operate through direct modification of shared parameters introduce persistent, scale-dependent behavioral scarring that cannot be reliably undone through post-hoc procedures. In contrast, reversible behavioral adaptation achieves exact and deterministic rollback by construction, preserving the base model identity across tasks, perturbations, and model scales. Together, these results empirically establish structural reversibility as a necessary design property for safe, controllable, and long-lived adaptive neural systems.

6 Analysis and Discussion

The empirical results presented in Section 5 reveal a consistent structural separation between irreversible shared-parameter adaptation and reversible behavioral isolation. These differences persist across mutation intensities, elimination thresholds, model scales, and evaluation conditions. In this section, we analyze the mechanisms underlying this divergence. We focus not on optimization heuristics, but on the structural properties of parameter organization that determine whether adaptive changes can be undone. This perspective re-frames reversibility as an architectural property rather than a training artifact.

6.1 Why Weight-Based Adaptation Is Irreversible

The empirical results in Section 5 demonstrate that direct modification of shared model parameters produces persistent post-reset divergence across all evaluated settings. This behavior follows from two structural properties of weight-based adaptation: gradient interference and objective entanglement.

First, gradient interference arises because multiple behaviors are encoded within a shared parameter substrate. Prior work in continual learning has shown that sequential updates applied to shared parameters induce interference between tasks, reflecting the stability–plasticity dilemma [2, 3]. Updates introduced to satisfy a new objective necessarily perturb representations supporting prior behaviors. Even when updates are small in magnitude, their effects may propagate non-locally due to overlapping internal feature representations.

Second, objective entanglement occurs because the same parameter tensor simultaneously encodes foundational capabilities and task-specific adaptations. Mechanistic analyses suggest that neural networks often represent multiple features in superposed or entangled forms within shared dimensions [15]. Once modified, such entangled parameters no longer admit a clean separation between original and adapted behavior. As a result, no deterministic inverse operation exists that can restore the model's prior functional state without access to an explicit checkpoint.

The systematic increase in post-reset divergence under weight mutation (Section 5.2) and the comparative recoverability measurements (Section 5.3) empirically substantiate this structural irreversibility.

6.2 Emergent Risks of Unbounded Learning in Shared Representational Spaces

We define emergence in this context as the development of new behavioral patterns arising from iterative adaptation within a shared representational substrate. Empirical studies of large language models have shown that increasing scale and training complexity can give rise to new capabilities that are not trivially predictable from smaller systems [16]. While some analyses argue that such emergence may partially reflect evaluation thresholds rather than discontinuous internal changes [17], it remains clear that shared-parameter scaling produces increasingly complex internal feature interactions.

Because shared representations encode multiple capabilities simultaneously, unbounded learning within this space increases the degree of representational entanglement. As entanglement grows, behavioral changes become progressively more difficult to localize or reverse. Mechanistic evidence suggests that feature superposition within shared parameter spaces may further amplify such coupling effects [15].

Irreversibility amplifies this structural coupling. Once adaptation-induced modifications are absorbed into the core parameter space, their downstream consequences may persist even after the original objective is removed. This creates a structural asymmetry: behavioral acquisition is straightforward, but behavioral removal becomes increasingly constrained.

Empirical results in Section 5 show that recoverability degrades under shared-parameter mutation and does not improve with model scale. These findings suggest that unbounded learning in shared representational spaces may accumulate irreversible residual behavioral drift over time. Importantly, this risk arises from architectural coupling rather than from specific optimization choices, highlighting the governance challenges associated with long-lived adaptive systems lacking explicit rollback mechanisms.

6.3 Why Behavioral Separation Works

Reversible behavioral adaptation avoids structural irreversibility by constraining the locus of learning. By isolating adaptive parameters from the core identity substrate, behavioral modifications are confined to a removable parameter subspace rather than being absorbed into shared representational weights.

Architectural isolation strategies have previously demonstrated that restricting parameter overlap across tasks mitigates destructive interference [6, 7]. More recent parameter-efficient fine-tuning methods, including adapters and low-rank adaptation, further illustrate that behavioral modification can be localized within constrained parameter additions while leaving the base model unchanged [8, 9]. These approaches implicitly reduce entanglement by separating task-specific parameters from foundational representations.

Reversible behavioral adaptation extends this principle by introducing explicit unload semantics. Because adaptive parameters are structurally decoupled from the identity-defining core parameters, rollback does not require approximation, retraining, or checkpoint restoration. Instead, the unload operator provides a deterministic mechanism for restoring the model to its baseline functional state. Emerging work on formal parameter isolation guarantees supports the theoretical plausibility of such structural separation [5].

The empirical elimination of post-reset divergence under behavioral removal (Section 5.1) and the invariance of recoverability across model scales (Section 5.5) demonstrate that reversibility is achieved by construction. Exact recovery is therefore not the result of improved optimization but of architectural separation.

More broadly, these results suggest that controllability in adaptive systems arises not from increasingly sophisticated training procedures, but from explicit structural boundaries between identity parameters and behavioral artifacts.

6.4 Relationship to Catastrophic Forgetting

Catastrophic forgetting has traditionally been framed as a statistical phenomenon arising from sequential optimization under non-stationary objectives [2, 3]. Prior work emphasizes stability–plasticity trade-offs and proposes regularization, replay, or architectural strategies to mitigate performance degradation on earlier tasks.

Our results suggest a complementary structural interpretation. Forgetting may be viewed as a consequence of irreversible shared-parameter adaptation. When multiple objectives are encoded within a single parameter substrate, new updates inevitably overwrite or distort representations supporting prior behaviors. Analyses of interference patterns in continual learning further support the view that shared parameters serve as a conduit for cross-task disruption [4].

From this perspective, forgetting is not solely an optimization failure, but a manifestation of structural coupling within the representational substrate. Parameter isolation approaches such as Progressive Neural Networks and iterative

pruning methods reduce forgetting by limiting parameter overlap across tasks [6, 7]. However, most such approaches were not designed with rollback or identity preservation as explicit objectives.

Reversible behavioral adaptation extends this structural framing by enforcing complete separation between identity-defining parameters and adaptive artifacts. This separation enables not only retention of prior capabilities, but deterministic recoverability to a baseline identity state.

6.5 Implications for Safe and Long-Lived Models

Long-lived adaptive systems require not only performance optimization but sustained controllability over time. Prior discussions of AI safety have emphasized the importance of oversight, corrigibility, and reliable behavioral control in deployed systems [18]. Structural irreversibility directly constrains these objectives by limiting the ability to audit, rollback, or govern behavioral changes introduced through shared-parameter adaptation.

Reversible behavioral learning provides three practical advantages:

1. **Deterministic Rollback:** Behavioral modifications can be removed without retraining, gradient inversion, or checkpoint restoration.
2. **Controllability:** Adaptive behaviors exist as bounded artifacts that can be attached, detached, versioned, and audited independently of the core model identity.
3. **Governance:** Structural separation enables explicit lifecycle management of learned behaviors, reducing the accumulation of irreversible behavioral residue over extended deployment horizons.

These properties suggest that reversibility should be treated as a first-class architectural design criterion for adaptive neural systems. As models increase in scale, complexity, and deployment duration, structural guarantees of recoverability may become increasingly important for system maintainability, compliance, and long-term behavioral stability.

7 Limitations and Scope

While this work establishes structural distinctions between reversible and irreversible adaptation paradigms, it is important to clarify the boundaries of both (i) the present empirical study and (ii) the RLAE framework itself. The following limitations define the scope of this paper and the structural constraints of the proposed approach.

7.1 Scope of the Present Study

The experimental results reported here are intentionally constrained to isolate structural properties under controlled settings. The following scenarios are explicitly excluded from the scope of this paper:

- **No Long-Horizon Reinforcement Learning:** We do not evaluate extended reinforcement learning pipelines (e.g., multi-stage RLHF, delayed reward optimization, or continual policy updates over long horizons). All experiments are conducted under bounded adaptation steps. Structural behavior under sustained reinforcement learning remains outside the present study.
- **No Emergence Induction or Capability Growth Experiments:** We do not attempt to induce emergent capabilities via large-scale continual adaptation, adversarial objectives, or unsupervised capability expansion. The interaction between structural irreversibility and emergent behavior is not empirically examined here.
- **No Multi-Behavior Arbitration or Lifelong Scheduling:** The experiments do not include concurrent behavioral modules, arbitration policies, or dynamic task switching across multiple adaptations. Multi-adapter coordination and behavioral composition are not evaluated in this paper.
- **Fixed Evaluation Distribution:** Divergence and recoverability metrics are computed under a fixed prompt distribution. Robustness under arbitrary distribution shift is not assessed.
- **Structural Focus Rather Than Performance Optimization:** The objective of this work is structural characterization (recoverability, divergence behavior, and identity preservation), not maximizing task accuracy, alignment quality, or downstream benchmark performance.

7.2 Structural Limits of RLAE

Beyond the experimental scope, the RLAE itself has inherent structural constraints:

- **No Guarantee of Behavioral Correctness:** RLAE guarantees rollback of behavioral artifacts but does not guarantee that the behavioral module is correct, aligned, or desirable while active.
- **No Elimination of Instability Within Modules:** While core identity is preserved, instability or forgetting may still occur inside a behavioral module itself.
- **No Automatic Arbitration Across Multiple Behaviors:** RLAE isolates behavioral components but does not, by itself, provide arbitration, prioritization, or conflict resolution mechanisms among multiple concurrent modules.
- **No Inherent Protection Against Adversarial Modules:** Structural separability enables deterministic removal but does not prevent harmful or adversarial behavior while a module is active.
- **Architectural Applicability Assumption:** RLAE presumes that adaptive behavior can be structurally separated from identity-defining parameters. Architectures that do not admit such separation fall outside its applicability.

RLAE does not eliminate catastrophic forgetting within the behavioral parameter subspace itself; forgetting may still occur inside individual behavioral modules.

The results presented in this paper should therefore be interpreted as structural characterizations under controlled adaptation settings. We isolate a necessary architectural condition for deterministic rollback—namely, separation between identity-defining core parameters and removable behavioral artifacts—without claiming exhaustive validation across all learning regimes or deployment environments.

8 Conclusion

In this work, we examined neural adaptation through a structural lens, distinguishing between shared-parameter weight mutation and behaviorally isolated adaptation.

We formalized structural irreversibility as a consequence of shared-parameter modification and introduced recoverability as an explicit evaluation criterion. Through controlled experiments, we demonstrated that direct weight mutation produces persistent behavioral drift with recoverability factor $\mathrm{RF} = 0$, while reversible behavioral adaptation achieves exact rollback with $\mathrm{RF} = 1$ within numerical precision limits.

These results hold across mutation intensities, model scales, and evaluation settings, indicating that reversibility is not a function of optimization quality, training budget, or model capacity. Rather, it is a structural property determined by where adaptive behavior is represented within the model.

Our findings suggest that long-lived adaptive systems require architectural separation between identity-defining core parameters and removable behavioral artifacts. Recoverability should therefore be treated as a first-class design objective for adaptive neural systems, with implications for safety, controllability, and lifecycle governance.

Acknowledgments

This research was conducted independently by the author. No institutional funding, supervision, or formal research support was received in the development of this work. The author is currently affiliated with Malla Reddy University; however, the ideas, methodology, experiments, and conclusions presented in this paper were developed autonomously.

References

[1] Pardhu Sri Rushi Varma Konduru. Rlae: Research implementations and code, experiments for runtime low-rank adaptive environments. `https://github.com/PardhuSreeRushiVarma20060119/rlae-research`, 2026.

[2] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[3] Matthias De Lange, Rahaf Aljundi, et al. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[4] Prakhar Kaushik et al. Understanding catastrophic forgetting and remembering in continual learning with relevance mapping networks. In *International Conference on Learning Representations*, 2021.

[5] Matteo Lanzillotta et al. Towards guarantees for parameter isolation in continual learning. *arXiv preprint arXiv:2310.01165*, 2023.

[6] Andrei A Rusu et al. Progressive neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2994–3002, 2016.

[7] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[8] Neil Houlsby et al. Parameter-efficient transfer learning for nlp. In *ICML*, 2019.

[9] Edward J Hu et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[12] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, Hoboken, NJ, 2 edition, 2006.

[14] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[15] Nelson Elhage, Neel Nanda, Chris Olah, Nicholas Joseph, Dario Amodei, and Shan Carter. Toy models of superposition. *Transformer Circuits Thread*, 2022. Anthropic.

[16] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[17] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 2023.

[18] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.