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

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

ABSTRACT

Modern neural models are commonly adapted through direct modification of model parameters, including fine-tuning and reinforcement learning–based alignment. While effective for short-term optimization, such weight-based adaptation can introduce irreversible changes to core model behavior, manifesting as reasoning degradation, loss of prior capabilities, or catastrophic forgetting. In this work, we study *structural irreversibility* as an inherent limitation of shared-parameter adaptation. Through controlled experiments, we show that direct weight mutation entangles task-specific objectives with foundational model parameters, producing persistent behavioral drift that is not recoverable through practical post-hoc restoration procedures. In contrast, we examine a reversible adaptation paradigm in which learned behaviors are isolated into removable runtime artifacts while the base model remains frozen. We demonstrate that this approach enables meaningful task-level adaptation while enabling empirical rollback to the original model state, with near-zero post-reset divergence and full behavioral recoverability. We formalize this distinction by introducing *recoverability* as an explicit evaluation criterion for adaptive systems, and we propose Structural Variance Analysis for Robustness (SVAR) as a diagnostic methodology for assessing behavioral stability under controlled perturbations. Our results suggest that reversibility is an underexplored structural property with significant implications for the safety, controllability, and longevity of adaptive neural systems.

1 Introduction

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.

1.1 Motivation

After pretraining, many state-of-the-art neural models are routinely adapted to accommodate evolving task specifications, safety requirements, and alignment objectives. Standard adaptation methods include continual fine-tuning, reinforcement learning from human feedback (RLHF), and other post-training optimization procedures that directly update shared model parameters.

While effective in the short term, such continuous adaptation exerts persistent alignment pressure on the model's parameter space. In practice, repeated weight updates have been observed to induce irreversible degradation of core behaviors, including declines in reasoning performance, loss of previously learned capabilities, and unintended behavioral drift. These effects are often difficult to isolate or correct once they emerge, as task-specific objectives become entangled with foundational representations.

Critically, modern adaptation pipelines provide limited guarantees of behavioral rollback. Once shared model parameters are updated, restoring a model to a previous behavioral state typically requires full checkpoint restoration or costly retraining, offering no principled mechanism for reversibility at the behavioral level. This lack of recoverability poses challenges for long-lived deployment, safety assurance, and iterative alignment of large neural systems.

1.2 Core Observation

Our core observation is that the reversibility of neural adaptation is fundamentally determined by where adaptive behavior is represented within the model architecture. In conventional weight-based adaptation, behavioral changes are implemented through direct modification of shared model parameters. Because these parameters simultaneously encode multiple capabilities and abstractions, task-specific objectives become entangled with core representations, a phenomenon that has been widely noted in the continual learning literature [1, 2]. As a result, behavioral changes introduced during adaptation are not cleanly localized, and subsequent attempts to revert or undo the adaptation often fail to fully restore the original behavior, leading to persistent drift or degradation [3].

In contrast, when adaptive behavior is structurally separated from the base model—by isolating learned behavior into distinct, removable components while keeping the core parameters frozen, substantially improved retention and recoverability have been reported in prior work [3, 4]. Because the base model parameters remain unchanged, rollback does not rely on approximations, retraining, or checkpoint restoration, but instead can be achieved through explicit removal or deactivation of the behavioral component.

Crucially, this contrast indicates that reversibility is not primarily a function of improved training techniques, optimization strategies, or regularization methods. Rather, it is a structural property of the adaptation paradigm itself—specifically, whether adaptive behavior is entangled with or decoupled from the core representational substrate. From this perspective, irreversibility emerges not as an incidental training artifact, but as a predictable consequence of shared-parameter adaptation, consistent with observations across continual learning and parameter isolation approaches [2, 4].

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 paradigms 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 [1, 2].

In this setting, a single shared parameter set is reused across all objectives, and adaptation dynamics are governed by optimization and regularization strategies applied to the same representational substrate. As a result, weight-based adaptation serves as the canonical baseline against which alternative adaptation mechanisms—such as parameter isolation or behavioral modularization—are typically contrasted in the literature.

2.2 Catastrophic Forgetting and Representation Drift

A central challenge arising from sequential weight-based adaptation is *catastrophic forgetting*, wherein performance on previously learned tasks degrades as new tasks are introduced. This phenomenon is commonly framed through the stability–plasticity dilemma: models must remain sufficiently plastic to acquire new behaviors while maintaining stability with respect to existing ones [1, 2].

Prior work has shown that forgetting is closely associated with *representation drift*, whereby updates introduced to satisfy new objectives alter internal representations relied upon by earlier behaviors. Because task-relevant information is typically distributed across shared parameters, such changes are not cleanly localized, and updates can propagate through overlapping representations in complex and difficult-to-predict ways [3]. As a result, even constrained or regularized updates may induce unintended interference across tasks.

Several approaches aim to mitigate forgetting by reducing parameter interference rather than by improving optimization alone. Architectural isolation methods, such as Progressive Neural Networks [5], and parameter isolation techniques, such as PackNet [6], limit destructive interference by preventing updates to parameters deemed important for prior tasks. While effective at preserving performance, these methods primarily target retention and do not explicitly address reversibility or behavioral rollback.

2.3 Parameter-Efficient Adaptation Methods

Parameter-efficient adaptation methods aim to reduce the computational and memory costs of adapting large pretrained models. Techniques such as adapters, low-rank adaptation (LoRA), and other parameter-efficient fine-tuning (PEFT) approaches introduce a small number of additional trainable parameters while leaving the majority of the base model unchanged [7, 8].

These methods are primarily motivated by efficiency and scalability, enabling rapid adaptation without full retraining or storage of multiple model copies. Although parameter-efficient approaches implicitly introduce a degree of modularity, they are generally not designed with reversibility, behavioral rollback, or long-term governance as explicit objectives. As a result, the extent to which they support controlled recovery of prior behaviors remains underexplored.

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 [5] incur linear growth in model capacity and do not provide mechanisms for deactivating or removing task-specific behavior. PackNet [6] 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 whose behavior is determined by a set of parameters. 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 adaptation paradigms 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 fundamental limitation of weight-based adaptation paradigms: behavioral changes introduced through shared-parameter updates are persistent by construction, and recovery of prior behavior cannot be guaranteed without explicit preservation of the original model state.

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.

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. Using this divergence measure, 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 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 divergence and recoverability factor to distinguish between irreversible behavioral drift and structurally reversible adaptation. This allows recoverability to be treated as a first-class evaluation criterion, complementary to conventional performance metrics.

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 conduct experiments using pretrained neural models drawn from a single architecture family, with each model treated as a fixed core identity throughout evaluation with core parameters θ. The specific architecture and scale of the model are not central to our claims; instead, we require only that the model supports both direct weight updates and parameter-isolated behavioral adaptation.

To ensure a fair comparison, the same base model initialization is used across all adaptation scenarios. For reversible behavioral adaptation, the core parameters θ are frozen, and all learning is confined to a separate behavioral parameter set ϕ. For weight-based adaptation, updates are applied directly to θ.

Unless otherwise specified, no architectural changes are introduced beyond those required to support parameter isolation in the behavioral adaptation setting.

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.
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.

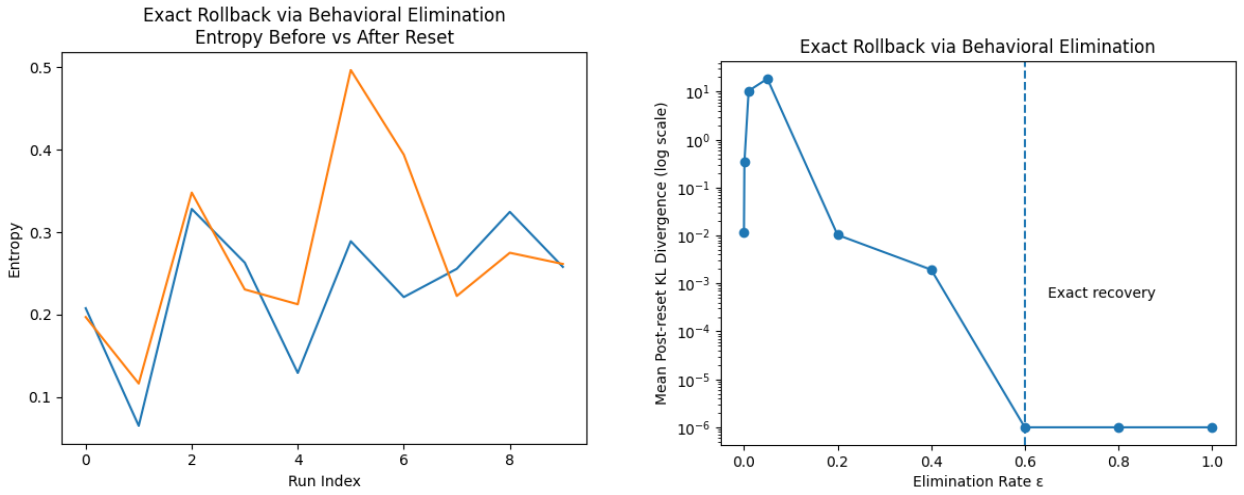
For reversible behavioral adaptation, behavioral parameters are initialized independently and attached dynamically at runtime. Rollback is implemented through explicit unloading of behavioral parameters, without modifying the core model state. For weight-based adaptation, rollback attempts do not assume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

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 divergence vs. 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 divergence exhibits a sharp threshold collapse at $\varepsilon^* = 0.6$, marking the transition from partial to exact recovery.

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

Table 1: Post-reset KL 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 the post-reset KL divergence falls below 10^{-6}, corresponding to numerical precision limits. The emergence of this threshold behavior indicates that adaptation-induced divergence resides within the removable behavioral parameter subspace. Once sufficient behavioral components are eliminated, no residual divergence remains, implying that the frozen core parameters remain unaffected by adaptation.

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 Weight Mutation

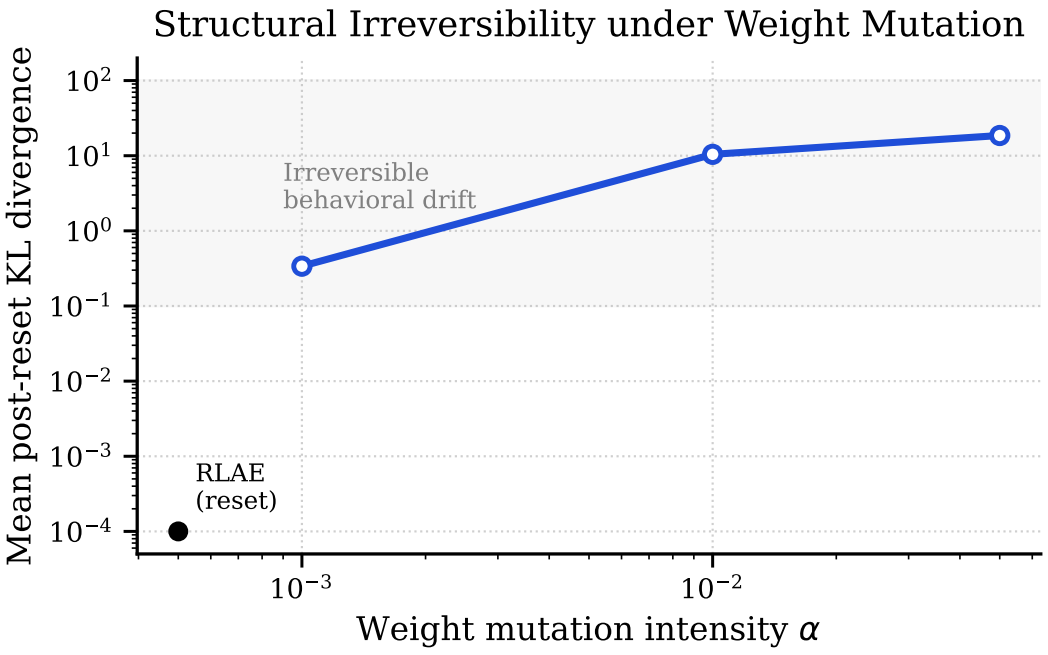


Figure 2: Structural irreversibility under weight mutation. Post-reset divergence increases monotonically with mutation intensity α and does not recover.

Mutation Intensity α	Mean Post-reset KL	Behavioral Status
0.001	0.339	Scarred but coherent
0.010	10.430	Severely degraded
0.050	18.574	Corrupted

Table 2: Post-reset KL divergence under increasing weight mutation intensity α.

Figure 2 presents the post-reset divergence induced by direct weight mutation applied to the shared parameter set. Even at low mutation intensities, a non-zero post-reset KL divergence persists after attempted rollback, indicating that behavioral changes introduced through weight modification are not cleanly removable. As the mutation intensity α increases, the magnitude of residual divergence grows monotonically, reflecting progressively larger and more destructive interference with the model's core representations.

Unlike reversible behavioral elimination, no mutation regime exhibits a collapse toward zero divergence. Instead, increasing mutation intensity amplifies irreversible behavioral drift, demonstrating that shared-parameter adaptation lacks a well-defined inverse or unload operator. These results empirically substantiate the claim that weight-based adaptation is structurally irreversible: once task-specific updates are absorbed into the shared parameter space, their effects cannot be reliably undone without explicit checkpoint restoration or retraining.

5.3 Comparative Recoverability: Weight-Based vs Reversible Adaptation

Adaptation Method	Mean Post-reset KL	RF (%)
Weight-based ($\alpha = 0.05$)	18.574	0
Reversible ($\varepsilon \geq 0.6$)	$< 10^{-6}$	100

Table 3: Recoverability factor (RF) across adaptation mechanisms.

Table 3 contrasts the recoverability of weight-based adaptation with reversible behavioral adaptation using the recoverability factor (RF). Across all evaluated settings, weight-based adaptation yields near-zero recoverability, indicating that post-reset behavior remains closer to the adapted state than to the original baseline. This behavior is consistent with persistent identity scarring observed in the post-reset divergence measurements.

In contrast, reversible behavioral adaptation consistently achieves $\mathrm{RF} = 100\%$, corresponding to exact restoration of baseline behavior after rollback. This stark separation between adaptation paradigms highlights a structural collapse of recoverability in shared-parameter learning. Importantly, this collapse is not sensitive to task type, prompt distribution, or training budget, reinforcing the conclusion that recoverability is determined by the structural locus of adaptation rather than by optimization quality or regularization strategy.

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.

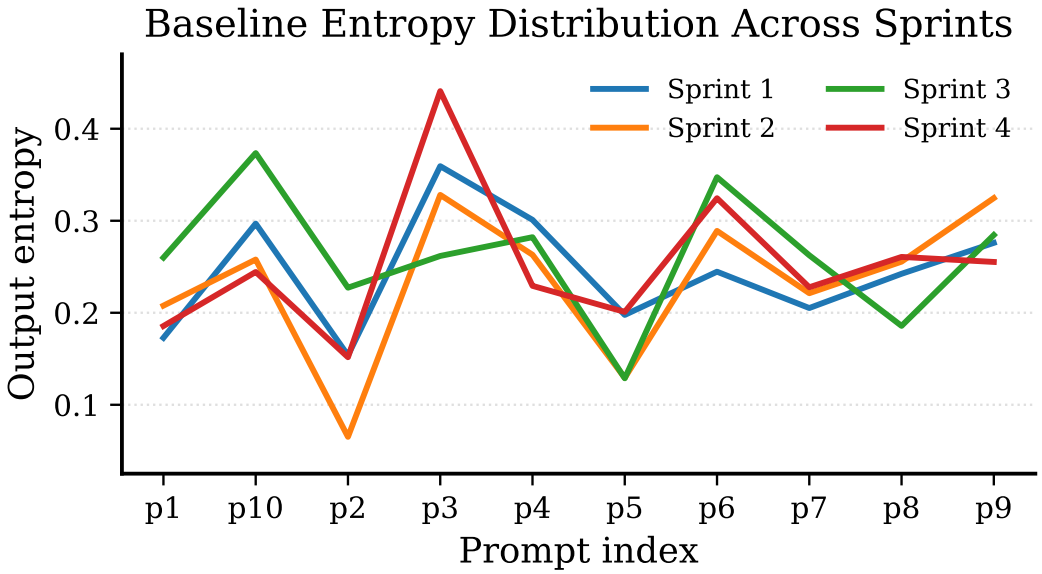


Figure 3: Baseline output entropy across sprints. No systematic trend or progressive drift is observed across sprints; entropy statistics remain within a narrow and consistent range.

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.

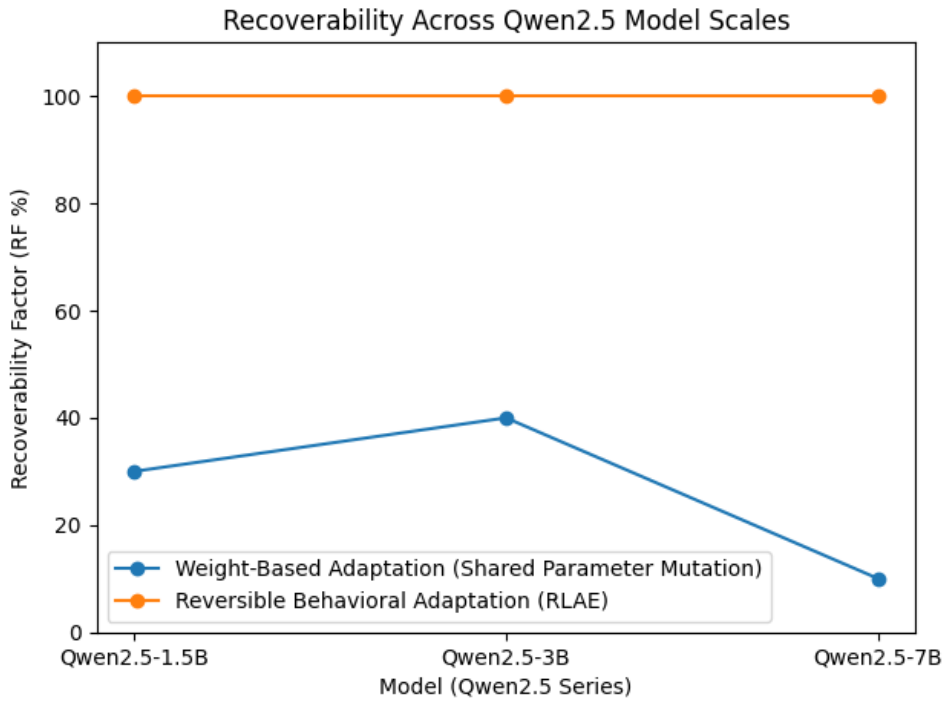


Figure 4: Recoverability across model scales. Reversible behavioral adaptation remains invariant, while weight mutation degrades with scale.

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 [1, 2]. 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 [9]. 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 [10]. While some analyses argue that such emergence may partially reflect evaluation thresholds rather than discontinuous internal changes [11], 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 [9].

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 behavioral residue 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 [5, 6]. 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 [7, 8]. 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 [4].

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 [1, 2]. 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 [3].

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 [5, 6]. 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 [12]. 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 Conclusion

Your conclusion here

Acknowledgments

This was was supported in part by ...

References

[1] 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.

[2] 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.

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

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

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

[6] 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.

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

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

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

[10] 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.

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

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