ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ»

«MASTER OF SCIENCE IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS”

«Παρακολούθηση της Αδράνειας Νευρώνων σε Βαθιά Νευρωνικά Δίκτυα»

Γιανναρόπουλος Διονύσιος Σωτήρης

Ιούνιος 2026

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ»

«Master of Science in Artificial Intelligence and Applications”

«Παρακολούθηση της Αδράνειας Νευρώνων σε Βαθιά Νευρωνικά Δίκτυα»

Διπλωματική Εργασία που υποβλήθηκε στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Θεσσαλίας ως μέρος των απαιτήσεων για την απόκτηση Διπλώματος Μεταπτυχιακών Σπουδών Τεχνητή Νοημοσύνη και Εφαρμογές από τον/την

Γιανναρόπουλος Διονύσιος Σωτήρης

Δήλωση Αυθεντικότητας, ζητήματα Copyright

«Ο/Η μεταπτυχιακός/ή φοιτητής/τρια που εκπόνησε την παρούσα διπλωματική εργασία φέρει ολόκληρη την ευθύνη προσδιορισμού της δίκαιης χρήσης του υλικού, η οποία ορίζεται στη βάση των εξής παραγόντων: του σκοπού και χαρακτήρα της χρήσης (μη-εμπορικός, μη-κερδοσκοπικός, αλλά εκπαιδευτικός-ερευνητικός), της φύσης του υλικού που χρησιμοποιεί (τμήμα του κειμένου, πίνακες, σχήματα, εικόνες κ.λπ.), του ποσοστού και της σημαντικότητας του τμήματος που χρησιμοποιεί σε σχέση με το όλο κείμενο υπό copyright, και των πιθανών συνεπειών της χρήσης αυτής στην αγορά ή την γενικότερη αξία του υπό copyright κειμένου».

Γιανναρόπουλος Διονύσιος

Υπογραφή

Ιούνιος 2026

Τριμελής Εξεταστική Επιτροπή

Η παρούσα Μ.Δ.Ε. εγκρίθηκε ομόφωνα από την τριμελή εξεταστική επιτροπή η οποία ορίστηκε από την Σ.Ε. / Συνέλευση του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Θεσσαλίας, σύμφωνα με το νόμο και τον εγκεκριμένο Οδηγό Σπουδών του Π.Μ.Σ. «Τεχνητή Νοημοσύνη και Εφαρμογές». Τα μέλη της Επιτροπής ήταν:

1ο Μέλος (Επιβλέπων):	
2ο Μέλος:	
3ο Μέλος:	

Η έγκριση της Μ.Δ.Ε. από το Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Θεσσαλίας, δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα.

Περίληψη

Τα νευρωνικά δίκτυα συνήθως εκπαιδεύονται με βάση μίας σταθερής αρχιτεκτονικής, παρόλο που πολλοί από τους νευρώνες τους μπορεί σταδιακά να μην υποχρησιμοποιούνται, να αδρανοποιούνται ή να καθίστανται λειτουργικά μη σημαντικοί. Η παρούσα διπλωματική μελετά τη νευρωνική αδράνεια ως ένα μετρήσιμο φαινόμενο κατά τη διάρκεια της εκπαίδευση και αναπτύσσει το «NeuroSentry», ένα πλαίσιο παρακολούθησης για την εκτίμηση της αδράνειας των νευρώνων από πολλαπλά συμπληρωματικά σήματα.

Το NeuroSentry συνδυάζει το activation rate, το positive activation magnitude, το gradient activity και τη σημαντικότητα Taylor πρώτης τάξης σε ένα συνεχές Dormancy Score. Επιπλέον, μοντελοποιεί μεταβάσεις σε επίπεδο πληθυσμού μέσω εμπειρικής ανάλυσης επιβίωσης και εκτιμήσεων κινδύνου, και επεκτείνει το σήμα παρακολούθησης με ένα ασαφές σύστημα κινδύνου αδράνειας, σχεδιασμένο να αποτυπώνει σταδιακή επιδείνωση, παροδική ανάκαμψη και επίμονη αδράνεια.

Η εργασία συγκρίνει το NeuroSentry με μια baseline μέθοδο βασισμένη στην ενεργοποίηση των νευρώνων, και αξιολογεί την προσέγγιση σε πειράματα συνθετικών dense δικτύων, CNN-MNIST, Sequential MNIST με αναδρομικά δίκτυα, σενάρια ασαφούς κύκλου ζωής νευρώνων, και αντίστοιχες μελέτες ενισχυτικής μάθησης σε CartPole, LunarLander και Humanoid.

Σε όλες τις μεγαλύτερες RL εργασίες, η μέθοδος baseline δεν καταγράφει τελικούς αδρανείς νευρώνες, ενώ το NeuroSentry εντοπίζει σημαντική αδράνεια, δείχνοντας ότι μόνο η συχνότητα ενεργοποίησης μπορεί να παραβλέψει υποχρησιμοποιημένη χωρητικότητα. Η ασαφής επέκταση παρέχει ένα ενδιάμεσο στάδιο μεταξύ φυσιολογικής συμπεριφοράς και πρακτικής έγκαιρης προειδοποίησης. Η διπλωματική επίσης περιγράφει την πειραματική εργασία που απαιτείται για μελλοντικούς ισχυρισμούς σχετικά με προσαρμογή βασισμένη σε παρεμβάσεις, ειδικά μέσω μηχανισμών pruning, reinitialization, rejuvanation και επέκτασης που ενεργοποιούνται από μετρήσεις της αδράνειας των νευρώνων.

Abstract

Neural networks are usually trained with a fixed architecture, even though many of their neurons gradually become underutilized, inactive, or functionally irrelevant. This thesis studies neuron dormancy as a measurable training-time phenomenon and develops NeuroSentry, a monitoring framework for estimating neuron dormancy from multiple complementary signals. NeuroSentry combines activation rate, positive activation magnitude, gradient activity, and first-order Taylor importance into a continuous Dormancy Score. It further models population-level transitions through empirical survival and hazard estimates, and extends the monitoring signal with a fuzzy dormancy-risk system designed to capture gradual deterioration, transient recovery, and persistent inactivity.

The work compares NeuroSentry with an activation-threshold baseline and evaluates the approach on synthetic dense networks, CNN-MNIST experiments, Sequential MNIST recurrent experiments, fuzzy neuron life-cycle scenarios, and reinforcement-learning studies using CartPole, LunarLander, and Humanoid. Across the larger RL tasks, the activation-only baseline reports no final dormant neurons while NeuroSentry identifies substantial multi-signal dormancy, showing that activation frequency alone can miss underused capacity. The fuzzy extension provides an interpretable bridge between dormancy theory and practical early-warning behavior. The thesis further outlines the experimental work needed to support future claims about intervention-based adaptation, especially pruning, reinitialization, rejuvenation, and expansion mechanisms triggered by measured neuron health.

Πρόλογος – Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος Σπουδών και πραγματεύεται το φαινόμενο της νευρωνικής αδράνειας σε βαθιά τεχνητά νευρωνικά δίκτυα . Η εργασία εστιάζει στην ανάπτυξη και αξιολόγηση του NeuroSentry, ενός πλαισίου παρακολούθησης της λειτουργικής κατάστασης των νευρώνων κατά την διάρκεια της εκπαίδευσης, με στόχο την καλύτερη κατανόηση του τρόπου με τον οποίο η εσωτερική δυναμική ενός δικτύου μεταβάλλεται με τον χρόνο.

Το ενδιαφέρον της μελέτης προκύπτει από την παρατήρηση ότι, ενώ τα νευρωνικά δίκτυα διαθέτουν μεγάλη ικανότητα, η αρχιτεκτονική τους παραμένει συνήθως μετά τον αρχικό σχεδιασμό σταθερή. Με αποτέλεσμα, ένα μέρος της υπολογιστικής τους χωρητικότητας να καταλήγει μη χρησιμοποιημένο ή λειτουργικά ανενεργό. Η εργασία επιχειρεί να προσεγγίσει το ζήτημα αυτό όχι ως ένα απλό δυαδικό πρόβλημα ενεργού ή ανενεργού νευρώνα, αλλά ως μια σταδιακή, χρονικά εξελισσόμενη κατάσταση που επηρεάζεται από πόλους παράγοντες.

Μέσο πειραμάτων σε supervised learning και reinforcement learning περιβάλλοντα, η εργασία εξετάζει κατά πόσο η μέθοδος Neurosentry μπορεί να προσφέρει πιο ουσιαστική εικόνα για την υγεία και αδράνεια των νευρώνων. Επίσης, η εργασία αποτελεί ένα βήμα προς συστήματα βαθύτερης διάγνωσης για χρήση της σε προσαρμοστικά νευρωνικά δίκτυα.

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή δρ. Κολομβάτσο Κώστα για την καθοδήγηση, και τις πολύτιμες παρατηρήσεις σε όλη την διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Ευχαριστώ επίσης τα μέλη της εξεταστικής επιτροπής για τον χρόνο που αφιέρωσαν στην αξιολόγηση της εργασίας.

Τέλος, ιδιαίτερες ευχαριστίες οφείλω στην οικογένεια μου και τους ανθρώπους που στάθηκαν δίπλα μου σε όλη αυτή την προσπάθεια, για την υπομονή και τη στήριξη τους.

Διευκρίνιση: Η παρούσα διπλωματική εργασία είναι γραμμένη στην αγγλική γλώσσα, καθώς το αντικείμενό της εντάσσεται στο ευρύτερο διεθνές ερευνητικό πεδίο της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Η χρήση της αγγλικής επιλέχθηκε ώστε η ορολογία, οι τεχνικές έννοιες και η παρουσίαση των μεθόδων και των πειραματικών αποτελεσμάτων να αποδίδονται με συνέπεια προς τη σχετική διεθνή βιβλιογραφία.

Table of Contents

List of Figures

List of Tables

1 Introduction

Biological nervous systems adapt continuously. Neural circuits strengthen, weaken, prune, and recruit components in response to experience, allowing the system to preserve useful behavior while remaining plastic under changing conditions (Chechik et al. 1998; Turrigiano 2012; Ming and Song 2011). Artificial neural networks, by contrast, are commonly trained with a fixed structure. Their parameters change, but their internal allocation of computational capacity is rarely monitored at the level of individual neurons. Once a network is initialized, the training process usually assumes that all units remain equally available for computation, even if some units stop firing, receive little gradient signal, or become irrelevant to the loss.

This mismatch is important because many artificial neurons gradually become weakly active, rarely active, or effectively inactive during training. This behavior is commonly discussed as dead or dormant neurons and has been reported in supervised networks, convolutional models, and reinforcement-learning agents (Evci 2018; Lu et al. 2020; Shin and Karniadakis 2020; Sokar et al. 2023). A dormant neuron is not necessarily permanently dead. It may still fire occasionally, but it contributes little to the forward computation, receives limited gradient signal, or has negligible influence on the current optimization objective. From the perspective of model capacity, such a neuron is present in the architecture but only weakly present in the computation.

Existing work has shown that dormant units can accumulate during training and that activation-based statistics provide a useful first view of this phenomenon (Sokar et al. 2023). However, activation alone provides an incomplete view of neuron utilization. A neuron can activate frequently but with negligible magnitude. A neuron can be rarely active but important for a specific subset of samples. A unit can also show nonzero activation while receiving almost no gradient, meaning that it participates in the forward pass but is not learning. These cases motivate a broader definition of *neuron health*: the degree to which a neuron participates in the forward representation, receives learning signal, and influences the loss trajectory.

The central goal of this thesis is to develop a monitoring framework that detects loss of functional participation before it is reduced to a binary dormant/not-dormant decision. The thesis argues that neuron dormancy should be treated as a gradual, measurable, and temporally evolving signal of neuron health rather than as a single binary condition. NeuroSentry approaches this problem as a training-time diagnostic system. Instead of using only activation magnitude, it combines activation frequency, activation strength, gradient flow, and first-order Taylor importance into a continuous Dormancy Score. The same signals also support survival and hazard views of the neuron population, allowing the system to describe how neurons transition into inactivity over time.

The thesis also extends the Dormancy Score with a fuzzy dormancy-risk estimator. The fuzzy component is motivated by the fact that dormancy is gradual, noisy, and context dependent. A neuron can temporarily dip, recover, decay slowly, or become inactive from birth. Fuzzy reasoning provides a structured way to express these patterns as low,

medium, and high risk without relying on a single crisp threshold (Zadeh 1965; Mamdani and Assilian 1975; Ross 2005). This makes the method more suitable as a future control signal for adaptive neural architectures, where decisions such as pruning, reinitialization, freezing, or rejuvenation should depend on persistent evidence rather than one unstable measurement.

1.1 Problem Statement

The practical problem addressed in this thesis is the lack of a fine-grained, temporally stable view of neuron utilization during training. A neural network can reach acceptable task performance while hiding substantial internal redundancy or inactivity. Conversely, a network can experience optimization difficulty because useful units transition into sustained inactivity early, leaving later layers with reduced representational diversity. Standard training logs such as loss and accuracy do not expose these internal dynamics.

A useful dormancy monitor should satisfy several requirements. First, it should operate at neuron or channel level rather than only at model level. Second, it should combine forward and backward signals, because a neuron can be active but unimportant, or inactive but still recoverable. Third, it should track persistence over time, because short dips should not trigger strong conclusions. Fourth, it should be interpretable enough to support later adaptive mechanisms. NeuroSentry is designed around these requirements.

1.2 Contributions

The main contributions are:

- A multi-signal Dormancy Score that combines forward activity, backward gradient flow, and Taylor importance into a continuous neuron-level estimate.
- Empirical survival and hazard metrics for tracking population-level neuron dormancy over training.
- A fuzzy dormancy-risk extension that connects activation, gradient, importance, and temporal age signals to interpretable risk categories.
- A comparison against the activation-threshold dormancy baseline of Sokar et al. (2023) on supervised learning and reinforcement learning experiments.
- A sensitivity analysis of Dormancy Score component weights, showing how architecture and signal choice influence dormancy estimates.

1.3 Scope and Thesis Positioning

The thesis should be read as a monitoring and diagnostic contribution. It does not claim to solve structural adaptation end to end. The proposed scores are not used here to automatically prune or regenerate neurons during training. Instead, the work establishes the signals that such future mechanisms would need. This distinction matters: pruning is an intervention, while NeuroSentry is the measurement layer that can inform interventions.

1.5 Thesis Organization

Chapter 2 reviews dormant neurons, adaptive neural networks, pruning, rejuvenation, continual learning, and fuzzy systems. Chapter 3 defines the Dormancy Score, hybrid thresholding, hysteresis, survival, and hazard metrics. Chapter 4 presents the fuzzy dormancy-risk estimator and synthetic neuron life-cycle scenarios. Chapter 5 summarizes supervised, recurrent, and reinforcement-learning experiments, including computational resources and hyperparameters. Chapter 6 concludes with limitations, open experimental work, and future directions.

2 Background and Related Work

This chapter reviews the concepts that motivate NeuroSentry. The thesis is positioned at the intersection of neuron inactivity, adaptive neural-network structure, pruning and rejuvenation methods, and fuzzy reasoning. The central idea is that neuron dormancy should not be treated only as a late-stage pruning criterion. It can also be monitored continuously during training and used as a signal for future adaptive mechanisms.

2.1 Biological Motivation: Plasticity and Dormancy

Biological nervous systems are not built with fixed, unchanging wiring. Instead, they continuously adapt as they respond to their activity, development, and experience. Mechanisms such as neurogenesis, and synaptic pruning work together as part of a self-regulating cycle, that are helping the brain remain stable while still allowing it to learn and evolve throughout life. Neurogenesis can refresh or expand neural synapsis through the birth and connection of new neurons (Ming and Song 2011; Tran et al. 2022). Synaptic pruning removes weak or redundant connections, refining efficiency and reducing uncontrolled excitation (Chechik et al. 1998). Additionally, dormant cells can temporarily reduce activity without necessarily dying, preserving metabolic resources while retaining potential for later reactivation under new stimuli (Faust et al. 2021; Benedetti and Couillard-Despres 2022; Benedetti et al. 2023). These mechanisms often work together with homeostatic plasticity, a balancing process that helps the nervous system regulate excitation, inhibition, and connectivity so that it can remain stable and function effectively. (Turrigiano 2012; Turrigiano 1999; Turrigiano and Nelson 2004).

These mechanisms suggest a useful analogy for artificial neural networks. In biological systems, silence is not always equivalent to low functional contribution. A neuron may remain silent for a period, become selectively engaged, or be part of a precursor population that is later recruited (Benedetti and Couillard-Despres 2022; Benedetti et al. 2023). The important question is therefore not only whether a neuron fires at a given instant, but whether its pattern of activity, learning signal, and temporal behavior indicate useful participation. This motivates a view of artificial neurons as components whose functional status can change over time.

Artificial neural networks are inspired by biological computation, but they usually lack this form of explicit structural self-regulation. A network may begin training with a fixed

number of units, yet only a subset may remain consistently useful. Other neurons may become weakly active, rarely active, saturated, or effectively irrelevant. If such behavior can be measured reliably, it can support adaptive operations such as pruning, rejuvenation, freezing, reinitialization, or neurogenesis-inspired expansion.

2.2 Artificial Neural Networks and Learning Settings

The experiments in this thesis use standard artificial neural networks in supervised-learning and reinforcement-learning settings. This section provides the minimum background needed to connect those settings with neuron dormancy.

2.2.1 Artificial Neurons and Deep Networks

Artificial neural networks are built from simple computational units inspired by biological neurons. Early mathematical models such as the McCulloch–Pitts neuron showed how networks of threshold units could implement logical computation (McCulloch and Pitts 1943), while the perceptron introduced a trainable linear classifier inspired by information storage in neural systems (Rosenblatt 1958). Modern deep networks generalize these ideas by stacking many layers of differentiable units and training their parameters from data using backpropagation and gradient-based optimization (Rumelhart et al. 1986; LeCun et al. 2012; Goodfellow et al. 2016; LeCun et al. 2015).

In Figure 1 is visualized the basic artificial-neuron abstraction used throughout the thesis. Inputs are weighted and combined with a bias term to form a pre-activation z, which is then passed through a nonlinear activation function to produce the output activation a. NeuroSentry monitors this process not only through the output activation, but also through the gradient and importance signals that pass back through the same unit during training.

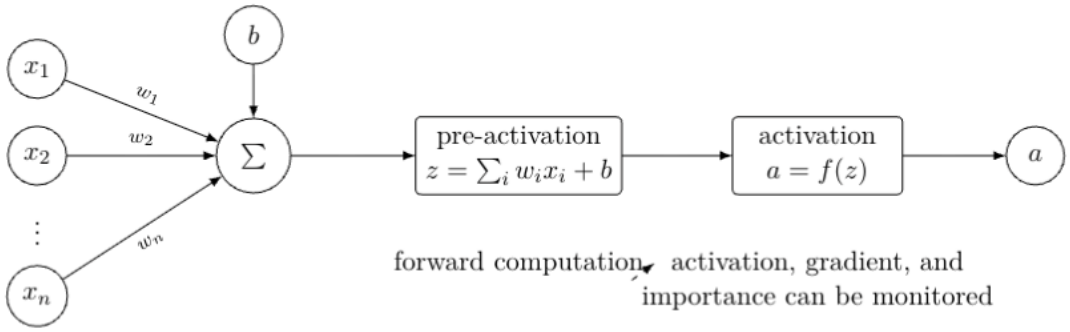

Figure 1: Simplified artificial neuron used in modern neural networks, following the weighted-sum and activation-function abstraction introduced by early neuron and perceptron models (McCulloch and Pitts 1943; Rosenblatt 1958; Goodfellow et al. 2016).

In this thesis, the term *artificial neuron* refers to a hidden unit or convolutional channel that transforms an input into a pre-activation and then into an activation. Although the biological analogy is imperfect, it is useful for discussing capacity allocation. A neural network may contain many available units, but training can concentrate useful computation in only a subset of them. The remaining units may become rarely active,

weakly active, gradient-inactive, or unimportant to the loss. NeuroSentry studies this internal allocation rather than treating the network as a black box that is summarized only by task accuracy. This framing is essential for the rest of the thesis, because dormancy is interpreted as reduced participation in computation and learning rather than merely as low output magnitude.

2.2.2 Supervised Learning

Supervised learning trains a model from input–target pairs. Given examples (x, y), the model predicts $\hat{y} = f_\theta(x)$ and updates its parameters θ to reduce a loss function such as cross-entropy or mean squared error (Bishop 2006; Goodfellow et al. 2016). In the supervised experiments of this thesis, the dense synthetic classifier and CNN-MNIST model are trained in this way: the target labels are fixed, the data distribution is comparatively stable, and the loss gives a direct gradient signal at every update.

This setting is useful for dormancy analysis because it isolates neuron utilization under relatively controlled optimization. If a neuron becomes inactive in supervised learning, the cause is often related to initialization, activation function behavior, optimization dynamics, data distribution, or layer redundancy. Monitoring activations alone can reveal some inactive units, but it does not show whether a unit still receives gradient signal or contributes to the loss. This motivates the NeuroSentry combination of forward activity, gradient activity, and Taylor importance. It also makes supervised learning a useful first setting for evaluating whether the monitor can distinguish inactive neurons from neurons that remain meaningful through gradient or Taylor evidence.

2.2.3 Deep Reinforcement Learning, DQN, and DDPG

Reinforcement learning differs from supervised learning because the agent learns through interaction with an environment. At each step, it observes a state, selects an action, receives a reward, and transitions to a new state. The objective is to learn behavior that maximizes long-term return rather than to match fixed labels (Sutton and Barto 2018). Q-learning estimates the expected return of taking an action in a state and then following a good policy thereafter (Watkins and Dayan 1992). Deep Q-Networks (DQN) combine this value-learning idea with deep neural networks, allowing value functions to be approximated from high-dimensional observations (Mnih et al. 2015).

Deep reinforcement learning is especially relevant to dormancy because the training distribution is non-stationary. The data seen by the network depends on the current policy, exploration schedule, replay buffer, and changing value targets. A neuron that is useful early in training it may later receive little stimulation, while another unit may remain inactive until the agent discovers a new region of the state space. This makes dormancy harder to interpret than in supervised learning and explains why prior work has highlighted dormant neurons in deep RL agents (Sokar et al. 2023; Liu et al. 2025).

The thesis uses DQN experiments for discrete-action control tasks such as CartPole and LunarLander. It also includes a Humanoid experiment based on Deep Deterministic Policy Gradient (DDPG), an actor–critic method designed for continuous action spaces

(Lillicrap et al. 2015). In actor–critic training, separate networks can represent the policy and the value estimator, so dormancy can emerge differently across actor and critic layers. This motivates monitoring at the layer and neuron level rather than reporting only episode reward. The non-stationarity of reinforcement learning makes it a stronger test case for dormancy monitoring, because temporary inactivity should not automatically be interpreted as permanent neuron failure.

2.3 Neural-Network Training Foundations

Before discussing dormancy, it is useful to define the basic training features that later become NeuroSentry monitoring signals. These definitions follow standard neural-network notation and correspond to the theoretical-background role used in the submitted paper.

2.3.1 Pre-Activation and Activation

For a neuron i at training step t, the pre-activation is

$$z_i^{(t)} = w_i^\top x^{(t)} + b_i,$$

where $x^{(t)}$ is the input to the neuron, w_i is its incoming weight vector, and b_i is its bias. The activation is then

$$a_i^{(t)} = f\left(z_i^{(t)}\right),$$

where f is the nonlinear activation function. The choice of f determines both the forward response and the local gradient flow (Goodfellow et al. 2016). For ReLU, negative pre-activations produce zero activations and zero local derivative, which is why ReLU-based networks are especially vulnerable to dead or dormant units (Lu et al. 2020; Shin and Karniadakis 2020).

During mini-batch training, a neuron does not produce a single value per optimization step. It produces one value per sample, and in convolutional or sequence models it also produces values over spatial or temporal positions. To obtain a consistent per-neuron statistic, all non-neuron dimensions can be averaged.

Let

$$A^{(t)} \in \mathbb{R}^{b_1 \times b_2 \times \cdots \times b_k \times d}$$

be the activation tensor for a layer, where the last dimension indexes the d neurons or channels. The summary activation of neuron i is

$$a_i^{(t)} = \frac{1}{N} \sum_{n=1}^{N} A_{n,i}^{(t)}, \qquad N = \prod_{m=1}^{k} b_m.$$

For dense layers, $N = B$ where B is the batch size. For convolutional layers, $N = BHW$ for batch, height, and width. For sequence layers, $N = BT$ for batch and time. This

convention allows a dormancy monitor to compare dense, convolutional, and recurrent architectures using the same neuron-level notation.

2.3.2 Backpropagation and Gradient Flow

Backpropagation computes gradients by applying the chain rule across the network (Rumelhart et al. 1986; LeCun et al. 2012; Goodfellow et al. 2016). For a neuron with activation $a_i^{(t)} = f(z_i^{(t)})$, the gradient with respect to its pre-activation is

$$\frac{\partial L}{\partial z_i^{(t)}} = \frac{\partial L}{\partial a_i^{(t)}} f'\left(z_i^{(t)}\right),$$

where L is the loss function. This quantity is central to dormancy analysis because it describes whether the neuron receives learning signal. If $\partial L / \partial z_i^{(t)}$ is persistently small, then the neuron's incoming weights receive little useful update.

As with activations, gradients may have batch, spatial, or temporal dimensions. A per-neuron gradient magnitude can be summarized as

$$G_i^{(t)} = \frac{1}{N} \sum_{n=1}^{N} \left| \frac{\partial L}{\partial z_{i,n}^{(t)}} \right|.$$

This statistic does not replace the gradients used by the optimizer. It is a diagnostic signal indicating whether the unit is still involved in learning.

2.3.3 Exponential Moving Averages

Raw activation and gradient values are noisy. They depend on the current mini-batch, optimizer state, and data order. Online monitoring therefore often uses an exponential moving average (EMA):

$$\hat{x}_i^{(t)} = \alpha \hat{x}_i^{(t-1)} + (1-\alpha) x_i^{(t)}, \qquad 0 < \alpha < 1.$$

Large α values provide long memory and smooth estimates. Smaller values react faster but produce noisier curves. In the dormancy setting, EMAs help separate persistent inactivity from short-lived fluctuations. This is conceptually similar to the role of moving averages in adaptive optimization, where past information stabilizes updates (Kingma and Ba 2017).

2.4 Dormant and Dead Neurons

A *dead neuron* typically refers to a unit that outputs zero, or a near-zero value, for all relevant inputs. In ReLU-based networks this often occurs when the pre-activation is negative for nearly all samples. Because the derivative of ReLU is zero in the negative regime, the neuron receives no gradient signal, and its incoming parameters stop updating. This behavior is known as the dying ReLU problem (Lu et al. 2020; Shin and Karniadakis 2020).

A *dormant neuron* is a broader concept. A dormant unit may not be exactly dead. It may still activate occasionally, but it contributes little to the forward computation, receives weak gradients, or has negligible influence on the loss. This distinction is important because dormancy can be temporary, recoverable, task-dependent, or architecture-dependent. Reinforcement-learning work has shown that dormant neurons can accumulate during training and reduce the effective capacity of the agent (Sokar et al. 2023). Similar phenomena have been discussed in supervised networks, pruning studies, and analyses of dead or underused units (Evci 2018; Dufort-Labbé et al. 2024).

It is useful to distinguish between *weakly active* and *rarely active* neurons. Weakly active neurons may fire regularly but with small magnitude. They may still encode low-amplitude but meaningful features, especially in later layers. Rarely active neurons fire for only a small fraction of inputs. If they also receive little gradient signal, they are stronger candidates for dormancy. A single activation statistic may confuse these cases, which is one reason NeuroSentry combines several monitoring signals.

Neuron death can also be interpreted differently depending on the learning setting. In some cases, death is purely harmful because it removes representational capacity. In other cases, dead or inactive neurons may reflect implicit structural adaptation, where unused capacity is effectively removed from the active computation (Dufort-Labbé et al. 2024). This thesis adopts a diagnostic perspective: before deciding whether inactivity is useful or harmful, the training process should expose how neurons are being used.

2.5 Factors That Influence Dormancy

Neuron dormancy rarely has a single cause. In practice, inactivity can emerge from the interaction between initialization, activation functions, optimization dynamics, data distribution, depth, and non-stationary training. This section reviews the main factors because they clarify why dormancy should be monitored as a training-time process rather than treated only as a final pruning decision.

Initialization and bias: Weights and biases determine the initial pre-activation distribution. If many ReLU neurons are initialized so that their pre-activations are negative for most inputs, they can begin training in a weak or inactive state. Symmetric or poorly scaled initialization can increase this risk in deep networks (Lu et al. 2020). A negative bias can have a similar effect by shifting the neuron toward the inactive region.

Learning rate and gradient updates. Large optimization steps can push a neuron across the boundary between active and inactive regimes. Once the neuron enters a region with near-zero derivative, its gradients may vanish and recovery becomes difficult (Shin and Karniadakis 2020). This is especially relevant for rectifying nonlinearities, where the backward signal can be blocked completely.

Data distribution: Input statistics determine which neurons are stimulated. If a neuron specializes in a region of the input space that appears rarely, it may look inactive for most batches even if it is meaningful for rare samples. Conversely, a neuron may be genuinely underused because the data distribution does not support the feature it encodes.

Depth and cascade effects: In deep architectures, inactivity can propagate. If early layers produce sparse or low-diversity representations, downstream units receive weaker inputs and can also become inactive. This creates layer-wise bottlenecks where a small number of dormant units can influence later computation (Sokar et al. 2023; Evci 2018).

Non-stationary training: Reinforcement learning and continual learning are particularly sensitive to non-stationarity. The target distribution, state visitation distribution, and policy behavior can change throughout training. Units that were useful early may become less useful later, and units that are inactive for long periods may still be needed after a distribution shift (Sokar et al. 2023; Kirkpatrick et al. 2017; Ven et al. 2024). This is one reason binary pruning decisions are risky without a temporal monitoring signal.

2.6 Monitoring Neuron Dormancy

Existing dormancy measures can be grouped into activation-based, gradient-based, and temporal indicators.

2.6.1 Activation-Based Metrics

The simplest metric is average activation magnitude. A neuron whose activation remains near zero across a dataset is a natural dormancy candidate. This idea appears in work on dead units and pruning, where near-zero activations indicate low utilization (Evci 2018; Molchanov et al. 2016a). A related metric is activation frequency: the fraction of inputs for which a neuron produces an output above a threshold. Sokar et al. (2023) use a normalized activation score and classify a unit as τ-dormant when its score falls below a fixed threshold.

Activation metrics are attractive because they are easy to compute and interpret. They also provide a direct view of forward-pass participation. However, they have limitations. A neuron can fire often but weakly. It can fire rarely but be important when it does fire. It can have nonzero activation while receiving almost no gradient. It can also look inactive under one data subset but useful under another. These limitations motivate the use of complementary backward-pass and saliency signals.

2.6.2 Gradient-Based Metrics

Gradients describe whether a neuron is still learning. If the gradient with respect to a neuron's pre-activation or incoming weights is consistently near zero, the neuron is unlikely to update meaningfully. This aligns with pruning and saliency literature, where gradient information helps estimate the effect of removing parameters or units (LeCun et al. 1989; Molchanov et al. 2017, 2016b). Recent work also argues that gradient-based signatures can reveal underuse that activation-only statistics miss (Liu et al. 2025).

Gradient monitoring is not sufficient by itself. A neuron may have low gradient because the model has already converged, not because the neuron has low functional contribution. A temporarily flat optimization region may also reduce gradients without implying permanent dormancy. For this reason, gradient information is most useful when combined with activation magnitude, activation rate, and temporal persistence.

2.6.3 Temporal and Persistence Metrics

Dormancy is a process, not only a state. A neuron that dips below a threshold for one step should not be treated the same as a neuron that remains inactive for hundreds of updates. Temporal metrics track how inactivity evolves across training, whether the same neurons remain dormant, and whether dormancy accumulates at the layer level (Sokar et al. 2023; Dufort-Labbé et al. 2024). Such information is especially important for adaptive architectures, because interventions such as pruning or reinitialization should not be triggered by noise.

This thesis uses exponential moving averages and hysteresis to separate persistent inactivity from short-term fluctuation. It also uses empirical survival and hazard curves to summarize dormancy as a population-level transition process.

2.7 Pruning, Rejuvenation, and Structural Adaptation

Neuron dormancy is closely related to pruning, but the goals are different. Pruning methods remove weights or units to reduce model size, computation, or redundancy. Classical saliency approaches such as Optimal Brain Damage estimate the effect of removing parameters on the loss (LeCun et al. 1989). Modern pruning surveys and methods examine activation sparsity, Taylor criteria, structured pruning, and other relevance signals (Molchanov et al. 2017; Blalock et al. 2020; Cheng et al. 2024; Douzandeh Zenoozi et al. 2025).

Other methods attempt to restore or reuse capacity rather than simply remove it. Neural rejuvenation and developmental approaches introduce replacement, regeneration, or adaptive reallocation mechanisms (Qiao et al. 2019; Han, Zhao, Pan, et al. 2021; Han, Zhao, Zeng, et al. 2021). Neurogenesis-inspired methods add or recruit neurons as learning demands change (Draelos et al. 2017; Chambers et al. 2004; Pandit and Kudithipudi 2020; Liu et al. 2024). Continual-learning methods also study how to preserve useful representations while adapting to new tasks (French 1999; Kirkpatrick et al. 2017; McCaffary 2021; Mishra and Suri 2023).

NeuroSentry does not perform structural adaptation directly in this thesis. Instead, it supplies the monitoring signal that such adaptation would require. A pruning method needs to know which units are persistently inactive. A rejuvenation method needs to know which units are drifting toward low functional contribution. A future neurogenesis mechanism needs to know where existing capacity is insufficient or unreliable.

2.8 Fuzzy and Neuro-Fuzzy Systems

Fuzzy logic provides a way to model gradual concepts such as low activity, medium risk, or high importance (Zadeh 1965; Mamdani and Assilian 1975; Ross 2005; Zimmermann 2010). Instead of forcing a fixed binary boundary between active and dormant, fuzzy membership functions allow a neuron to partially belong to several states. This is useful when the monitored signals are noisy, smooth, and context-dependent.

Neuro-fuzzy systems combine fuzzy interpretability with neural learning or adaptation. ANFIS is a classical example of a system where neural-style optimization tunes fuzzy inference components (Jang 1993). Related work applies fuzzy reasoning to pruning, feature selection, evolving classifiers, or reinterpretations of neural operations (Junio Guimarães et al. 2019; Fan 2017; Campos Souza and Lughofer 2023; Xue et al. 2023). These methods show that fuzzy logic can support adaptive model behavior, but they generally operate at the level of model outputs, rules, features, or global structure.

The fuzzy component of NeuroSentry is different in granularity. It operates inside the training process at the neuron level. It takes activation rate, activation magnitude, gradient activity, Taylor importance, and age-related signals as inputs, then produces a dormancy-risk estimate. The aim is not to replace gradient-based learning with fuzzy logic, but to provide an interpretable layer of reasoning over internal neuron health.

2.9 Positioning of NeuroSentry

The related work suggests three requirements for a useful dormancy monitor. First, the monitor should combine forward and backward signals, because activation alone provides incomplete evidence. Second, it should be temporal, because dormancy develops over training and may include recovery. Third, it should be interpretable enough to support later adaptive decisions.

NeuroSentry addresses these requirements by combining a multi-signal Dormancy Score, empirical survival and hazard summaries, and a fuzzy risk estimator. The Dormancy Score captures neuron-level inactivity from activation, gradient, and Taylor signals. Survival and hazard describe the layer-level transition into dormancy. The fuzzy estimator expresses gradual risk and handles ambiguous cases such as newborn neurons, specialists, transient dips, and persistent inactivity.

3 NeuroSentry Methodology

The previous chapter showed that activation-only views of dormancy are useful but incomplete. A neuron may be inactive, weakly active, gradient-starved, or locally unimportant to the loss, and these cases should not be treated as identical. This chapter therefore introduces NeuroSentry as a multi-signal monitoring framework designed to estimate neuron health during training.

The full NeuroSentry framework has three connected components: the Dormancy Score, survival/hazard analysis, and fuzzy dormancy risk. This chapter focuses on the first two components. It defines the score-level monitoring signals and the temporal summaries that describe how neurons enter dormancy over training. Chapter 4 then builds on the same signals to introduce the fuzzy risk layer.

The general neural-network quantities used here were introduced in Section 2.3: pre-activation, activation aggregation over batch, spatial, and temporal dimensions, backpropagation gradients, and exponential moving averages. Following the submitted-

paper structure, the present chapter focuses on the proposed approach: how those foundational quantities are combined into neuron-level dormancy signals, adaptive thresholds, survival and hazard summaries, and the baseline comparison.

3.1 Overview of the Proposed Monitor

Within this chapter, NeuroSentry is presented through two complementary monitoring components. First, the Dormancy Score provides a continuous estimate of neuron inactivity by integrating activation behavior, learning signal, and first-order contribution to the loss. Second, empirical survival and hazard summaries describe how the monitored neuron population transitions into dormancy over time. The fuzzy-risk component is part of the complete framework, but it is treated separately in Chapter 4 because it adds rule-based interpretation and risk categories on top of the score-level signals.

The key design choice is to treat dormancy as a gradual, temporal, and multi-signal condition. The following sections introduce each signal separately before combining them into the final diagnostic score.

3.2 Activation Rate

The activation rate measures how frequently a neuron is active. NeuroSentry defines the binary indicator

$$X_i^{(t)} = \mathbf{1}\left(a_i^{(t)} > \theta_i^{(t)}\right),$$

where $\theta_i^{(t)}$ is a threshold. The activation-rate EMA is

$$\hat{r}_i^{(t)} = \alpha \hat{r}_i^{(t-1)} + (1-\alpha) X_i^{(t)}.$$

A low activation rate suggests that the neuron rarely participates in the forward pass. However, low rate alone is insufficient for classifying dormancy, because specialist neurons may fire rarely but meaningfully. Activation rate is therefore interpreted as one component of neuron health, not as a standalone decision rule.

3.3 Positive Activation Magnitude

Activation rate measures frequency, but not strength. A neuron may fire often with tiny activations, or rarely with large activations. NeuroSentry therefore tracks positive activation magnitude:

$$M_i^{(t)} = a_i^{(t)} X_i^{(t)},$$

and its moving average

$$\hat{m}_i^{(t)} = \alpha \hat{m}_i^{(t-1)} + (1-\alpha) M_i^{(t)}.$$

If both $\hat{r}_i^{(t)}$ and $\hat{m}_i^{(t)}$ remain low, the neuron is both rarely active and weak when it activates. This combination is stronger evidence of functional dormancy than either signal alone.

3.4 Gradient Activity

The gradient-activity EMA is

$$\hat{g}_i^{(t)} = \alpha \hat{g}_i^{(t-1)} + (1-\alpha) G_i^{(t)}.$$

This statistic reflects learning vitality. A neuron with nonzero activation but near-zero gradient may still participate in the forward pass, but it is not being updated in a meaningful way. Gradient activity therefore captures a dimension of neuron health that activation metrics cannot fully express (Molchanov et al. 2016b; Liu et al. 2025). Its role in the Dormancy Score is complementary: it helps separate low forward use from reduced learning participation.

3.5 First-Order Taylor Importance

NeuroSentry also tracks a first-order Taylor importance score. Consider the loss L as a function of activation a_i. A first-order Taylor approximation gives

$$L(a_i + \Delta a_i) \approx L(a_i) + \frac{\partial L}{\partial a_i} \Delta a_i.$$

If the neuron is removed by setting $a_i = 0$, then $\Delta a_i = -a_i$, and the approximate change in loss magnitude is

$$|\Delta L_i| \approx \left| a_i \frac{\partial L}{\partial a_i} \right|.$$

The per-step Taylor importance is therefore

$$S_i^{(t)} = \left| a_i^{(t)} \frac{\partial L}{\partial a_i^{(t)}} \right|,$$

with EMA

$$\hat{s}_i^{(t)} = \alpha \hat{s}_i^{(t-1)} + (1-\alpha) S_i^{(t)}.$$

Taylor-style criteria are widely used in pruning and saliency estimation because they approximate the local effect of removing a component (LeCun et al. 1989; Molchanov et al. 2017). In NeuroSentry, low Taylor importance indicates that a neuron has little influence on the current loss trajectory.

3.6 Dormancy Score

The Dormancy Score combines the four monitoring signals into a continuous estimate:

$$D_i^{(t)} = w_r \left(1 - \tilde{r}_i^{(t)}\right) + w_m \left(1 - \tilde{m}_i^{(t)}\right) + w_g \left(1 - \tilde{g}_i^{(t)}\right) + w_s \left(1 - \tilde{s}_i^{(t)}\right),$$

where $\tilde{r}$, $\tilde{m}$, $\tilde{g}$, and $\tilde{s}$ are normalized values in $[0,1]$, and the weights sum to one. A high score indicates that the neuron is rarely active, weakly active, gradient-inactive, or structurally unimportant.

The equal-weight configuration uses

$$w_r = w_m = w_g = w_s = 0.25.$$

The sensitivity analysis in Chapter 5 examines how different weightings change the detected dormant fraction. This is important because different architectures emphasize different signals: dense networks may be sensitive to activation rate, while CNNs may require stronger backward-pass indicators.

Therefore, the Dormancy Score should be interpreted as a diagnostic estimate of reduced neuron participation, not as a final decision to remove or modify the neuron.

3.7 Adaptive Thresholds

Fixed thresholds are simple, but they can be sensitive to scale, noise, and training dynamics. Different layers and neurons operate at different activation scales, and those scales shift during training. NeuroSentry therefore uses adaptive quantile thresholds.

A local threshold $\theta^{(t)}_{\text{local},i}$ is computed from the recent behavior of neuron i. This captures neuron-specific scale. A global threshold $\theta^{(t)}_{\text{global}}$ is computed from the monitored population. This captures network-wide activity. The hybrid threshold is

$$\theta^{(t)}_{h,i} = \lambda_h \theta^{(t)}_{\text{local},i} + (1-\lambda_h)\theta^{(t)}_{\text{global}}, \qquad 0 \le \lambda_h \le 1.$$

When λ_h approaches one, the rule becomes mostly local. When it approaches zero, the rule becomes mostly global. Intermediate values balance neuron-specific adaptation with population-level consistency. This resembles the balance between local homeostatic regulation and population-level stability in biological systems (Miconi 2017; Turrigiano 2012). The thresholding mechanism is therefore designed to support fair comparison across neurons without assuming that a single fixed scale is valid for every layer.

Quantile-based thresholding also connects to statistical threshold estimation, where thresholds are adapted to observed distributions rather than fixed manually (Donoho and Johnstone 1994; Greenwald and Khanna 2001; Jain and Chlamtac 1985; Dunning 2021).

3.8 Hysteresis

Even adaptive thresholds can produce noisy binary decisions if a neuron fluctuates around the boundary. NeuroSentry therefore uses hysteresis. A neuron must remain below the dormancy criterion for K consecutive monitoring steps before it is declared dormant. Similarly, it must recover sufficiently before the dormant state is cleared.

This mechanism prevents short dips from creating unstable dormant/non-dormant switching. In the fuzzy system, hysteresis is also used on the smoothed risk score.

Conceptually, hysteresis turns dormancy detection from an instantaneous threshold crossing into a persistence requirement.

3.9 Empirical Survival and Hazard

Dormancy can also be viewed as a time-to-event process. For each monitored layer, let $A^{(t)}$ be the set of neurons that are active at step t, and let $N_{\text{new}}^{(t)}$ be the number of neurons that newly enter dormancy at that step. The empirical hazard is

$$\hat{h}^{(t)} = \frac{N_{\text{new}}^{(t)}}{|A^{(t-1)}| + \varepsilon}.$$

It estimates the instantaneous rate at which currently active neurons transition into dormancy. The empirical survival estimate is

$$\hat{S}(t) = \prod_{\tau=1}^{t} \left(1 - \hat{h}^{(\tau)}\right).$$

Survival analysis is commonly used to model time-to-event phenomena (Cox 1972; Dietz et al. 2000). In this thesis, it is used descriptively: survival curves show how much of a layer remains active, while hazard curves show when dormancy transitions concentrate.

3.10 Relationship to the Sokar Baseline

The baseline used for comparison follows the activation-score approach of Sokar et al. (2023). For each neuron, the baseline computes a normalized average activation score and marks the neuron as τ-dormant when the score is below a fixed threshold. This baseline is valuable because it is simple, interpretable, and directly connected to prior work on dormant neurons in reinforcement learning.

NeuroSentry differs in three ways. First, it uses multiple signals rather than activation alone. Second, it tracks those signals online with EMAs and adaptive thresholds. Third, it exposes temporal summaries through hysteresis, survival, hazard, and fuzzy risk. The goal is not to discard the baseline, but to show where a richer monitor can reveal dormancy patterns that a fixed activation score cannot distinguish.

4 Fuzzy Dormancy Risk

The Dormancy Score introduced in Chapter 3 provides a compact numerical estimate of neuron inactivity. However, dormancy is not always a clearly separable state. A neuron may show weak activation while remaining important for rare samples, or it may temporarily lose activity before recovering later in training. For this reason, a purely numerical score or fixed binary threshold may not fully express the uncertainty involved in neuron-health assessment.

This chapter introduces a fuzzy dormancy-risk layer that interprets the same monitoring signals as gradual risk categories. Its purpose is not to replace the Dormancy Score, but

to express uncertain neuron states in a way that is closer to how dormancy appears during training: gradual, noisy, partially reversible, and dependent on the age of the neuron and the persistence of inactivity.

4.1 Motivation

Binary dormancy rules are easy to interpret, but they can oversimplify neuron behavior during training. A neuron whose activation falls below a threshold for a single step is not necessarily dormant. A neuron with low activation but high Taylor importance may be a specialist unit. A neuron that is newly introduced or recently initialized may need time before its activity becomes meaningful. These cases are difficult to represent with a fixed binary rule, because the relevant concepts are not crisp. Terms such as weak activation, low gradient, high importance, old neuron, or prolonged inactivity have natural degrees of membership.

Fuzzy logic was introduced to model such gradual categories (Zadeh 1965). In a fuzzy system, a value can partially belong to several linguistic states, and a rule base can combine these states to produce a smooth output (Mamdani and Assilian 1975; Ross 2005; Tan and Chua 2007). This is appropriate for neuron dormancy because dormancy is not a single event. It is usually a trajectory: a unit may become less active, stop receiving gradient, lose importance, recover, or eventually transition into sustained inactivity.

The fuzzy dormancy system therefore aims to answer a slightly different question from the Dormancy Score. Instead of asking only how dormant a neuron is according to a weighted numeric score, it asks what kind of risk state the neuron appears to occupy. The output is a scalar risk score and a distribution of fuzzy masses over low, medium, and high risk. These masses are useful for interpretation because they show whether a high-risk decision is strong and stable or merely a temporary mixture of medium and high evidence.

4.2 Inputs and Normalization

The fuzzy system uses six normalized inputs:

$$(\hat{r}, \hat{m}, \hat{g}, \hat{s}, a_b, a_f),$$

where $\hat{r}$ is activation rate, $\hat{m}$ is positive activation magnitude, $\hat{g}$ is gradient activity, $\hat{s}$ is Taylor importance, a_b is age since birth, and a_f is age since last firing. The first four inputs are the same core health indicators used by the Dormancy Score. The two age-related variables add temporal context.

The first four inputs correspond directly to the EMA quantities introduced in Chapter 3: activation-rate EMA, positive-magnitude EMA, gradient-activity EMA, and Taylor-importance EMA. The fuzzy system therefore does not introduce a separate measurement pipeline; it reinterprets the same smoothed monitoring signals through linguistic risk categories.

The age since birth, a_b, represents how long the neuron has existed in the training process. In fixed architectures this begins at initialization, while in a future adaptive architecture it could begin when a neuron is generated or reinitialized. The age since last firing, a_f, measures persistence of inactivity. A neuron that has not fired for a long period should be treated differently from one that missed only one mini-batch. This is the fuzzy analogue of the hysteresis idea used in the crisp monitor.

All inputs are mapped to $[0,1]$ before fuzzy inference. This makes the rule base independent of the raw scale of activations or gradients. In practice, the normalized values can be obtained from per-layer min–max normalization, quantile scaling, or the already normalized NeuroSentry monitoring outputs. The exact normalization should be kept consistent within each experiment group so that risk curves are comparable across time.

4.3 Membership Functions

For the activity and importance variables $\hat{r}$, $\hat{m}$, $\hat{g}$, and $\hat{s}$, the fuzzy system uses low, medium, and high membership functions. A low activation-rate membership indicates rare firing. A low magnitude membership indicates weak positive response. Low gradient membership indicates limited learning signal, and low Taylor membership indicates little local influence on the loss.

The age variables use young, mature, and old membership functions. A young neuron receives protection against premature high-risk classification, because early activity is often unstable. A mature neuron is expected to have developed some role in the representation. An old neuron with long inactivity is stronger evidence for persistent dormancy, unless it also retains high importance.

Triangular and trapezoidal membership functions are sufficient for this thesis because the goal is interpretability and stable monitoring rather than learning a complex fuzzy controller. A generic triangular membership function is

$$\mu_{\triangle}(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right),$$

where a, b, and c define the left support, peak, and right support. Trapezoidal functions can be used for concepts such as definitely low or definitely high, where a plateau is desirable.

4.4 Rule Logic

The fuzzy rule base encodes qualitative dormancy patterns. The core rule intuition is:

- If activation rate, activation magnitude, gradient activity, and Taylor importance are high, then dormancy risk is low.
- If activation is weak but gradient or Taylor importance remains high, then risk should be medium rather than immediately high.
- If activation, gradient, and Taylor importance are all low, then risk should be high.

- If the neuron has not fired for a long time, high-risk evidence should be strengthened.
- If the neuron is young, high-risk evidence should be softened to avoid misclassifying early specialization.

This rule structure reflects the biological motivation behind the thesis. Persistent lack of use is a stronger signal than a short inactive period, but silence alone is not always evidence of low functional contribution. Some neurons are specialists. Some neurons are temporarily inactive because the data distribution or policy trajectory has not visited their preferred region. The fuzzy system captures this ambiguity by assigning partial membership to multiple risk categories.

Let w_{Low}, w_{Med}, and w_{High} denote the accumulated rule strengths for the three risk categories. In a Mamdani-style system, an individual rule with antecedents $A_1, \dots, A_k$ can fire with strength

$$\phi = \min\big(\mu_{A_1}(x_1), \dots, \mu_{A_k}(x_k)\big),$$

and this strength contributes to the output category associated with the rule. Other t-norms could be used, but the minimum operator keeps the inference easy to inspect.

The purpose of these rules is not to produce a fully learned controller, but to make the interpretation of dormancy risk transparent and inspectable.

4.5 Contextual Refinements

After the core rules fire, the fuzzy system applies age-based refinements. These refinements make the risk estimate better aligned with neuron life-cycle behavior.

Newborn protection: If the neuron is young, high-risk mass is reduced. Newly initialized neurons often have unstable gradients and may not immediately become useful. Prematurely classifying them as high risk would confuse normal early exploration with dormancy.

Persistent inactivity: If age since last firing is old, high-risk mass is increased. A long period without activation is stronger evidence of dormancy than a one-step dip.

Older specialists: If a neuron is old but Taylor importance is high, high-risk mass is reduced. This protects long-lived specialist units that may fire rarely but remain important when they do fire.

Stagnant old units: If a neuron is old, gradient activity is low, and Taylor importance is low, high-risk mass is increased. This combination corresponds to a strong pruning or rejuvenation candidate because the unit is neither learning nor contributing.

These refinements are heuristic, but they are transparent. The goal is to make the risk estimate explainable enough that future adaptive interventions can inspect why a neuron was classified as high risk.

4.6 Defuzzification and Hysteresis

The final scalar risk is obtained through centroid-style defuzzification. The thesis implementation maps the three risk categories to representative centroids:

$$C_{\text{Low}} = 0.2, \qquad C_{\text{Med}} = 0.5, \qquad C_{\text{High}} = 0.85.$$

The scalar risk is then

$$R_i^{(t)} = \frac{w_{\text{Low}} C_{\text{Low}} + w_{\text{Med}} C_{\text{Med}} + w_{\text{High}} C_{\text{High}}}{w_{\text{Low}} + w_{\text{Med}} + w_{\text{High}} + \varepsilon}.$$

The raw risk is smoothed with an EMA and converted into a high-risk state through hysteresis. A neuron enters high risk when the smoothed risk remains above an enter threshold for K consecutive steps. It exits only after falling below a lower exit threshold. This enter/exit design prevents rapid oscillation when the risk score fluctuates around the boundary. It also matches the methodology of Chapter 3, where dormancy is treated as a persistent event rather than an instantaneous threshold crossing.

4.7 Neuron Life-Cycle Scenario

The synthetic life-cycle experiment demonstrates the conceptual behavior that the fuzzy system is meant to capture. The scenario simulates a neuron from birth through stable usefulness, gradual decay, a temporary dip and recovery, weak recovery, and final dormancy. Although synthetic, this setting is valuable because it isolates patterns that are difficult to observe cleanly in real training logs. In real networks, optimization noise, batch effects, and changing data distributions can overlap. The life-cycle scenario exposes the intended response of the fuzzy system under controlled conditions.

Figure 2 shows the monitored signals over this trajectory. The most important feature of the scenario is that dormancy is not represented as a single sudden drop. Instead, the neuron passes through phases. Early activity is unstable, then the neuron becomes useful, then activity and gradient signal decay. A short dip tests whether the risk system overreacts. A later sustained period tests whether the system can detect persistent low use. The final phase represents a transition into sustained dormancy.

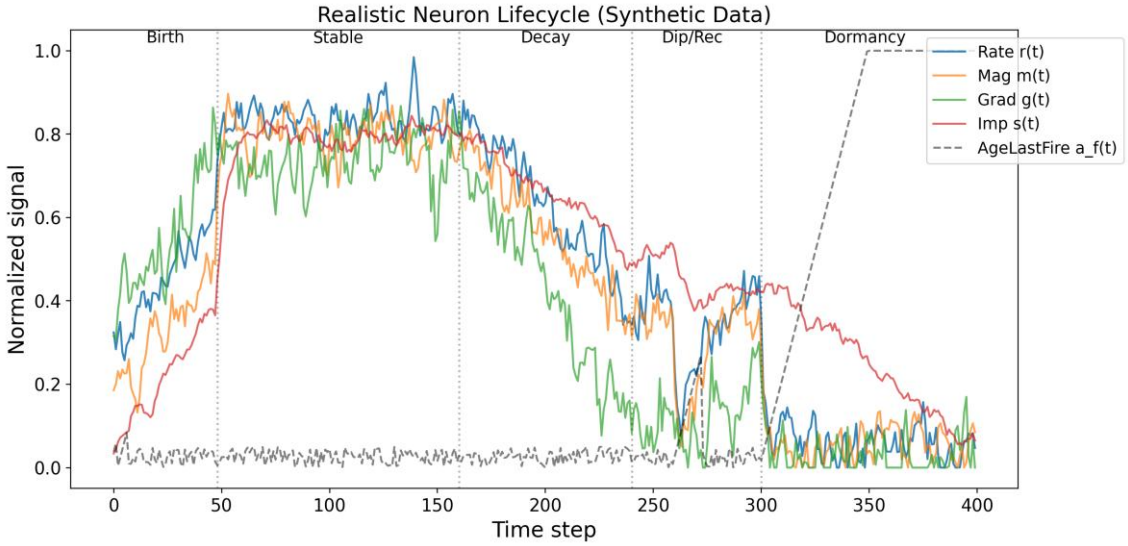

Figure 2: Synthetic neuron life-cycle features used to connect fuzzy dormancy risk with the theoretical notion of neuron health. The scenario includes birth, stable activity, decay, dip/recovery, weak recovery, and final dormancy phases.

The fuzzy risk curve in Figure 3 increases as activity, gradient flow, and importance decay. The raw risk responds quickly to the changing inputs, while the smoothed risk and hysteresis thresholds stabilize the decision. The fuzzy masses in Figure 4 show how the system gradually shifts mass from low and medium risk into high risk. This mass view is important because it shows whether a high scalar risk is caused by one sharp spike or by a broad change in evidence.

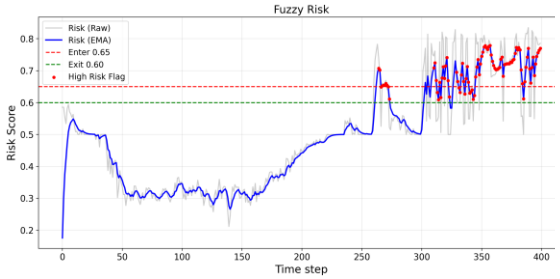

Figure 3: Raw and smoothed fuzzy risk with hysteresis thresholds across the synthetic neuron life cycle.

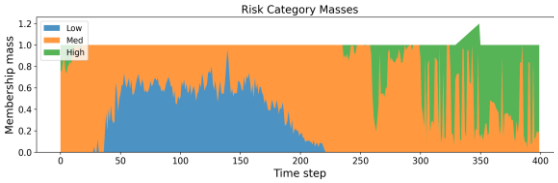

Figure 4: Low, medium, and high fuzzy-risk masses across the synthetic neuron life cycle.

A comparison of the fuzzy system with a simple activation-threshold baseline inspired by Sokar et al. (2023) is also visualized in Figure 5. In the single-neuron synthetic setting, the baseline reduces to a fixed threshold on activation magnitude. The fuzzy system responds to gradual deterioration before the activation magnitude reaches the fixed threshold. It also supports recovery because the risk state can exit when the monitored signals improve. The baseline only indicates whether activation falls below a fixed threshold and does not distinguish between decay, recovery, or task-specific specialization.

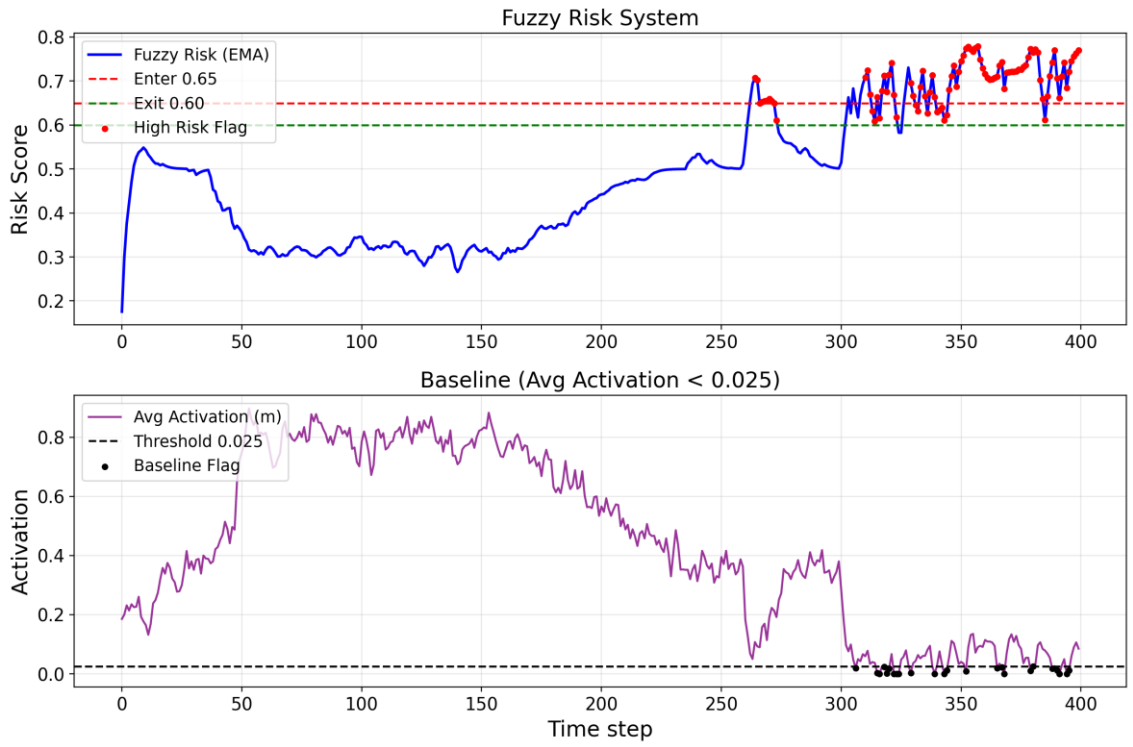

Figure 5: Comparison between fuzzy risk and a fixed activation-threshold baseline in the synthetic life-cycle scenario.

4.8 Hysteresis and Noise Filtering

The hysteresis scenario tests whether short transient dips are filtered out while sustained low-use periods trigger a high-risk flag. Figure 6 shows active phases, a short transient dip, temporary dormancy, weak recovery, and final dormancy. This scenario is designed around a common practical failure mode: if the monitor reacts to every local fluctuation, it becomes too noisy to guide adaptation. If it reacts too slowly, it loses early-warning value.

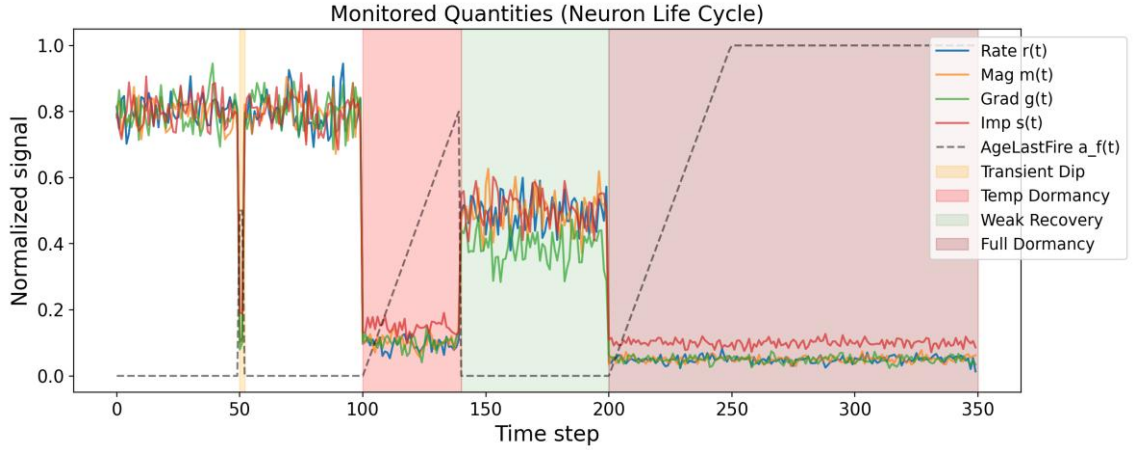

Figure 6: Synthetic hysteresis scenario. The short dip is designed to remain below the hysteresis duration, while later sustained events should trigger risk flags.

The outputs in FiguresFigure 7 and Figure 8 show the intended behavior. The transient dip increases risk but does not persist long enough to enter the high-risk state. Later low-

activity regions produce sustained risk and therefore trigger the flag. When the neuron recovers, the exit threshold prevents immediate oscillation and requires evidence of stable recovery. This behavior is especially relevant for reinforcement learning, where state visitation and target distributions can change from one phase of training to the next.

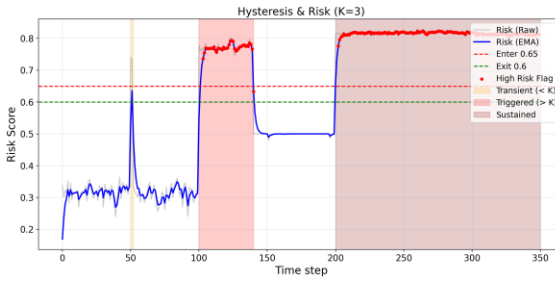

Figure 7: Risk curve and hysteresis flags for Synthetic hysteresis scenario.

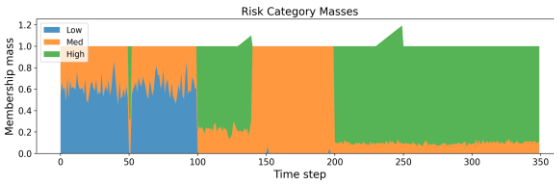

Figure 8: Risk membership masses for Synthetic hysteresis scenario.

4.9 Detection Latency and Risk Alignment

The latency experiment repeats synthetic dormancy trajectories over many random seeds and compares the first detection step of the fuzzy system against the fixed activation-threshold baseline. This evaluates whether the early-warning behavior observed in the single trajectory is systematic or only a visual artifact. Across 10,000 repeated trajectories, the fuzzy method detects dormancy earlier in most simulated trajectories, with a mean latency advantage of 17.46 training steps, a median advantage of 38 steps, and earlier detection in 71.7% of trials.

Figure 9 presents the same experiment from a more diagnostic perspective. Instead of only plotting detector latency, each point summarizes one lifecycle window from one synthetic trajectory. The horizontal axis is the mean activation activity in that window, while the vertical axis is the mean fuzzy risk. Stable active windows remain in the high-activity, low-risk region. Transient dip/recovery windows move toward lower activity but remain below the sustained high-risk threshold on average. Final sustained-dormancy windows cluster in the low-activity, high-risk region; 98.8% of these final windows are above the high-risk threshold. This supports the intended interpretation of the fuzzy layer: it does not merely react to an isolated low activation value, but assigns high risk when the lifecycle pattern is consistent with sustained dormancy.

A training step in this analysis is an abstract update index, not a fixed amount of wall-clock time. The latency advantage should therefore be interpreted as earlier detection in the learning trajectory. Its practical time value depends on hardware, batch size, model size, and monitoring frequency.

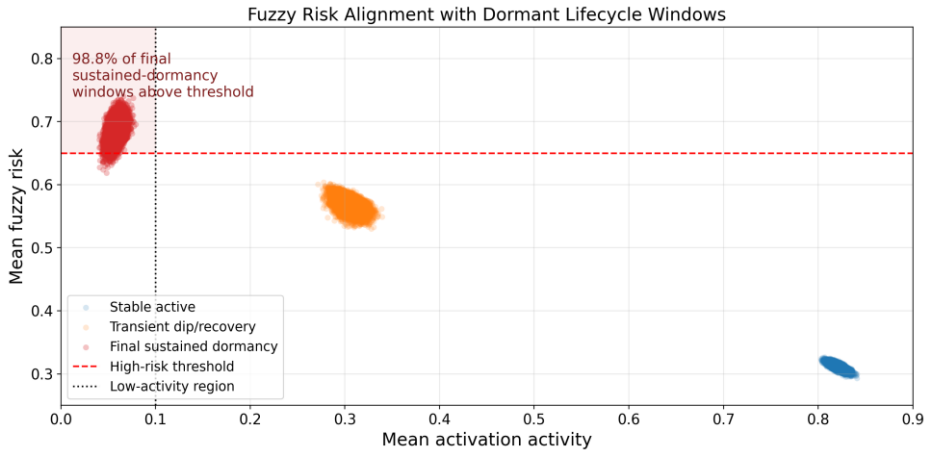

Figure 9: Risk-alignment scatter over repeated synthetic lifecycle trajectories. Final sustained-dormancy windows cluster in the low-activity, high-risk region, while stable and transient-recovery windows remain below the sustained high-risk threshold on average.

4.10 Role in the Thesis

The fuzzy chapter provides the conceptual bridge between biological motivation and experimental monitoring. The biological literature motivates gradual and context-dependent neuron status. The Dormancy Score provides numeric evidence. The fuzzy system translates that evidence into interpretable risk categories and temporal decisions. This makes it a natural candidate for future closed-loop adaptive architectures: high risk could trigger observation, delayed pruning, rejuvenation, or targeted reinitialization, while medium risk could trigger longer monitoring rather than immediate intervention.

5 Experimental Evaluation

The previous chapters introduced the thesis argument that dormancy is gradual, temporal, and multi-signal. This chapter evaluates that argument empirically using the completed experiment artifacts in the project repository. Its purpose is not to repeat the methodological definitions from Chapter 3, but to report how the proposed monitoring framework behaves under the evaluation protocol used in this thesis.

The chapter first defines the setup, experiment families, monitoring protocol, and computational resources. It then reports the results and benchmark costs for each experiment family before summarizing the main findings. Throughout the chapter, the results are interpreted as diagnostic evidence: NeuroSentry is evaluated as a monitoring framework, not as a closed-loop pruning, rejuvenation, reinitialization, or expansion system.

5.1 Evaluation Setup and Protocol

5.1.1 Experiment Families

The evaluation is organized into two experiment families. The first family contains supervised-learning experiments: a dense synthetic classification task, CNN-MNIST, and Sequential MNIST with a recurrent network. These experiments provide controlled

settings where the same Dormancy Score, fuzzy-risk system, survival curves, hazard curves, Sokar-style baseline, and sensitivity analysis can be compared across dense, convolutional, and recurrent architectures. The second family contains reinforcement-learning experiments: CartPole and LunarLander with DQN, and Humanoid with DDPG. These runs test whether the same monitoring vocabulary remains usable in custom non-stationary training loops.

This organization is intentional. The supervised experiments provide architecture and weight-sensitivity analysis, while the matched reinforcement-learning experiments test integration and compare the multi-signal method directly with an activation-threshold baseline in a setting where neuron dormancy has already been reported as a practical issue. The two families are therefore reported separately: Section 5.2 presents the supervised-learning experiments and their benchmarks, and Section 5.3 presents the reinforcement-learning comparison.

5.1.2 Monitoring Protocol

The evaluation focuses on whether NeuroSentry provides richer diagnostic information than activation-threshold monitoring. Each experiment is interpreted as evidence about neuron-health measurement rather than as a pruning or rejuvenation study.

The baseline comparison follows the activation-threshold idea used by Sokar et al. (2023): a neuron is classified as dormant when its normalized average activation score falls below a fixed threshold τ. The corrected supervised reruns use $\tau = 0.025$ and evaluate the baseline on a fixed score-estimation minibatch every 1000 training steps. This baseline is useful because it is simple and directly connected to prior work on dormant neurons in deep reinforcement learning. It also creates a clear reference point for evaluating whether a multi-signal monitor adds information beyond low forward activity.

NeuroSentry evaluates the same monitored layers using activation rate, positive activation magnitude, gradient activity, and first-order Taylor importance. These quantities are smoothed with exponential moving averages, combined into the Dormancy Score, and summarized through dormant fractions, dormant counts, survival curves, and hazard curves. In fuzzy-enabled runs, the same monitored state is also mapped into low-, medium-, and high-risk fuzzy categories. Hysteresis is used so that short-lived threshold crossings do not immediately become persistent dormant events.

5.1.3 Computational Resources

The submitted NeuroSentry paper and the thesis reruns were conducted on the project server environment summarized in Table 1.

Table 1: Computational resources and software environment used for the reported NeuroSentry experiments.

Resource	**Configuration**
GPU	NVIDIA GeForce RTX 4090
Operating system	Ubuntu 24.04 LTS
Main deep-learning framework	TensorFlow/Keras

RL environments	Gymnasium CartPole-v1, LunarLander-v3, and Humanoid-v5
Analysis and plotting	Python, NumPy, Matplotlib, JSON/CSV aggregate summaries

5.2 Supervised Learning Experiments

This section reports the supervised-learning experiment family introduced in Section 5.1.1. The goal is to keep the supervised evidence comparable across dense, convolutional, and recurrent architectures before interpreting each architecture separately. The section therefore begins with the shared configuration, wall-clock benchmark results, sensitivity scenarios, and aggregate summary, followed by the Dense synthetic, CNN-MNIST, and Sequential MNIST RNN case studies.

5.2.1 Supervised Experiment Configuration

The shared protocol compares the Dormancy Score with the activation-only baseline, repeats each configuration over ten runs, and includes sensitivity analysis over the Dormancy Score weights. Table 2 summarizes the common setup used by the dense synthetic, CNN-MNIST, and Sequential MNIST aggregate runs.

Table 2: Shared supervised-experiment protocol used for the thesis aggregate results.

Parameter	**Value**
Independent runs	10 runs per reported aggregate
Training length	50 epochs
Batch size	128
Optimizer	Adam (Kingma and Ba 2017)
Dormancy Score weights	Equal baseline: $w_r = w_m = w_g = w_s = 0.25$
Hybrid threshold quantiles	$q_{\text{local}} = q_{\text{global}} = 5.0$
Hybrid threshold mixing	$\lambda_h = 0.4$
Hysteresis window	2 monitoring steps
Sokar-style threshold	$\tau = 0.025$ with one 64-sample score-estimation batch every 1000 steps
Primary metrics	Dormant fraction, dormant count, fuzzy risk, risk categories, survival, hazard, sensitivity AUC, benchmark cost

The supervised architectures are intentionally modest. This makes neuron-level monitoring interpretable and avoids hiding dormancy behavior behind very large feature extractors. Table 3 summarizes the monitored networks.

Table 3: Supervised architectures used to evaluate NeuroSentry under dense, convolutional, and recurrent activation structures.

Experiment	**Model**	**Monitored layers and purpose**

Dense synthetic	MLP with two hidden layers of 16 and 8 units	Dense hidden layers are monitored to study dormancy in a controlled binary classification setting.
CNN-MNIST	One convolutional layer with 36 filters, max pooling, and a dense layer of 128 units	The convolutional channel layer and dense layer are monitored to test whether the same signals remain useful with spatial activations.
Sequential MNIST RNN	SimpleRNN layer with 64 units followed by a dense layer of 64 units	The recurrent output and dense classifier block are monitored to test batch-time aggregation in a sequence setting.

5.2.2 Supervised Benchmarking

Computational cost for the supervised experiments is reported using dedicated wall-clock benchmark scripts rather than inferred from the main diagnostic aggregates. This matters because the aggregate experiments are designed to study neuron-health trajectories, while a cost comparison requires comparable runs with monitoring disabled, with activation-threshold monitoring enabled, with Dormancy Score monitoring enabled, and with Dormancy Score plus fuzzy risk enabled. The supervised benchmarks compare a plain model with no NeuroSentry layers, a Sokar-style activation baseline, NeuroSentry score-only monitoring, and NeuroSentry with fuzzy risk. The updated benchmark scripts also include a repeated simple-model control without NeuroSentry or Sokar; this control is used to check run-to-run harness variability in the source artifacts but is not reported as a thesis method.

Table 4: Supervised wall-clock benchmark results. Times are mean ± standard deviation over five timed repeats. Extra time is reported in seconds relative to the plain model baseline for each experiment.

Configuration	Dense synthetic (s)	CNN-MNIST (s)	Sequential MNIST RNN (s)
Plain	0.734±0.005	1.588±0.017	1.716±0.036
Sokar	0.839±0.041	1.660±0.066	1.787±0.048
Score	1.492±0.120	2.348±0.041	2.464±0.122
Fuzzy	1.643±0.022	2.572±0.090	2.641±0.031
Score extra	0.759	0.760	0.747
Fuzzy extra	0.910	0.984	0.925

In Table 4 we report the supervised wall-clock benchmark matrix. The plain baseline is the reference for full end-to-end cost. The Sokar baseline is comparatively cheap because it evaluates an activation-threshold dormancy score on a lightweight probe. NeuroSentry score-only adds the multi-signal Dormancy Score, survival, and hazard machinery, while the fuzzy configuration adds the rule-based risk layer on top of that monitored state.

The supervised benchmark shows a consistent cost structure. The Sokar-style probe remains close to the plain baseline, adding 0.105 seconds in Dense synthetic, 0.071 seconds in CNN-MNIST, and 0.071 seconds in Sequential MNIST RNN. Score-only NeuroSentry adds 0.747–0.760 seconds across the three supervised benchmarks, reflecting the cost of maintaining the multi-signal monitoring state. The fuzzy layer then adds a smaller incremental cost over score-only NeuroSentry: 0.151 seconds in Dense synthetic, 0.224 seconds in CNN-MNIST, and 0.178 seconds in Sequential MNIST RNN. This is the expected tradeoff: fuzzy risk improves interpretability by producing gradual risk categories, but it is not free. The important practical point is that the fuzzy overhead is an incremental addition on top of the score-only monitor, not a separate training procedure.

Overall, the supervised benchmarks show that NeuroSentry adds a measurable but interpretable monitoring cost. Sokar-style activation thresholding is cheaper because it uses a lightweight activation-only score and can be evaluated as a low-frequency probe over a score-estimation minibatch. In contrast, full NeuroSentry is intentionally an online multi-signal monitor: score-only NeuroSentry tracks forward activity, gradient activity, Taylor importance, survival, and hazard, while the fuzzy configuration adds rule-based risk categories on top of that state. Therefore, full NeuroSentry should not be expected to match Sokar's wall-clock cost while it retains gradient/Taylor tracking and per-batch online state. Approaching Sokar-like cost would require an explicit lower-cost configuration, such as a lower-frequency probe-style pipeline, an activation-only or activation-dominant lite mode, or reduced-frequency fuzzy-risk evaluation.

5.2.3 Sensitivity Scenarios

The Dormancy Score is a weighted combination of four signals, so sensitivity analysis is necessary. Table 5 lists the weight profiles used in the supervised aggregate experiments. The scenarios include isolated forward signals, isolated backward signals, and mixed profiles that emphasize different definitions of underuse.

Table 5: Dormancy Score weighting scenarios used for supervised sensitivity analysis.

Baseline and isolated signals					**Composite profiles**				
Scenario	w_r	w_m	w_g	w_s	**Scenario**	w_r	w_m	w_g	w_s
equal	0.25	0.25	0.25	0.25	rate_dominant	0.55	0.25	0.10	0.10
rate_only	1.00	0.00	0.00	0.00	mag_dominant	0.25	0.55	0.10	0.10
mag_only	0.00	1.00	0.00	0.00	grad_dominant	0.10	0.10	0.55	0.25
grad_only	0.00	0.00	1.00	0.00	taylor_dominant	0.10	0.10	0.25	0.55
taylor_only	0.00	0.00	0.00	1.00	activation_focused	0.40	0.40	0.10	0.10
rate_mag_only	0.50	0.50	0.00	0.00	forward_heavy	0.40	0.30	0.15	0.15

grad_taylor_only	0.00	0.00	0.50	0.50	backward_heavy	0.15	0.15	0.30	0.40
					gradient_focused	0.10	0.10	0.40	0.40
					inactive_unimportant	0.35	0.15	0.15	0.35

5.2.4 Aggregate Result Summary

In Table 6 the main equal-weight supervised aggregates are presented. Each row uses the fuzzy-enabled rerun for that architecture, so the table reports both the crisp Dormancy Score outcome and the fuzzy risk categories from the same monitored history. Fuzzy risk is reported as the mean final layer-level risk, and the low/medium/high counts are final aggregate neuron counts across the monitored population.

Table 6: Final supervised aggregate results for equal-weight fuzzy-enabled NeuroSentry runs. Dormant percentage and survival are mean layer-level summaries over ten runs.

Experiment	Dormant %	Sokar %	Survival	Fuzzy risk	Low/Med/High risk
Dense synthetic	9.69 $\pm$ 2.95	0.94 $\pm$ 2.00	0.688 $\pm$ 0.066	0.239 $\pm$ 0.039	23.0/0.4/0.6
CNN-MNIST	7.11 $\pm$ 1.19	2.37 $\pm$ 1.88	0.788 $\pm$ 0.098	0.227 $\pm$ 0.010	148.2/15.1/0.7
Sequential MNIST RNN	7.34 $\pm$ 1.68	2.58 $\pm$ 0.99	0.830 $\pm$ 0.024	0.232 $\pm$ 0.010	117.1/9.0/1.9

5.2.5 Dense Synthetic Experiment

The dense synthetic experiment provides a controlled setting where each hidden unit is easy to interpret. The model has two monitored dense layers with 16 and 8 units, so the total monitored population is small enough that dormant counts and fuzzy categories remain directly readable. This experiment is useful for testing whether the Dormancy Score identifies early allocation effects and whether the activation-only baseline reacts only to the most obvious low-activation units.

Figure 10 summarizes the main equal-weight aggregate output using the corrected Sokar-style baseline with $\tau = 0.025$. The fuzzy-enabled NeuroSentry configuration produces a final mean dormant fraction of $9.69\% \pm 2.95\%$, compared with $0.94\% \pm 2.00\%$ for the activation-only baseline. This larger gap is important: with the corrected normalized-activation threshold, Sokar is conservative in the dense supervised setting and only flags the most clearly low-activation units. NeuroSentry identifies a broader subset of low-participation units because the Dormancy Score also considers activation magnitude, gradient activity, Taylor importance, adaptive thresholds, and temporal persistence. The survival curve ends near 0.688 ± 0.066, showing that most monitored units remain active across training while a small subset enters persistent dormancy.

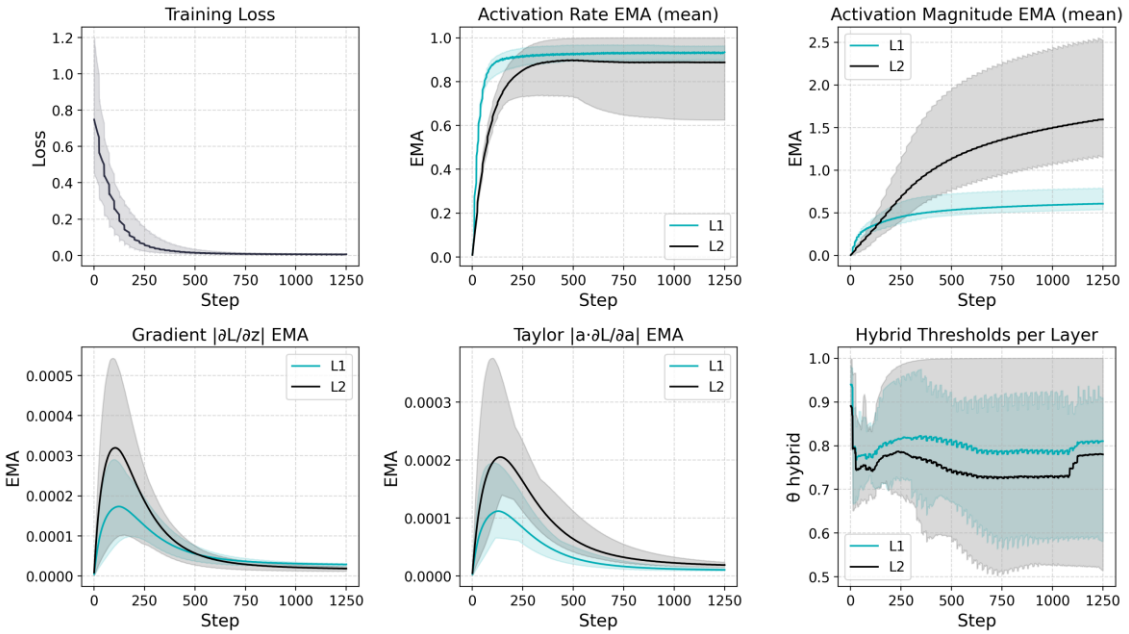

Figure 10: Dense synthetic aggregate with dormant-neuron behavior, survival, hazard, and related NeuroSentry monitoring curves.

Figure 11 and 12 separates the temporal survival and hazard panels from the combined dashboard. This makes the event interpretation clearer: survival describes the proportion of monitored neurons that have not entered persistent dormancy, while hazard emphasizes when new dormancy events occur. In the dense setting, the hazard curve is concentrated around the early allocation phase, which is consistent with a small network assigning only part of its hidden capacity to the synthetic decision boundary.

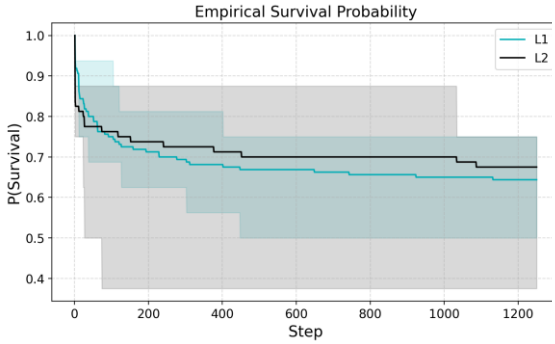

Figure 11: Empirical survival for dense synthetic experiment.

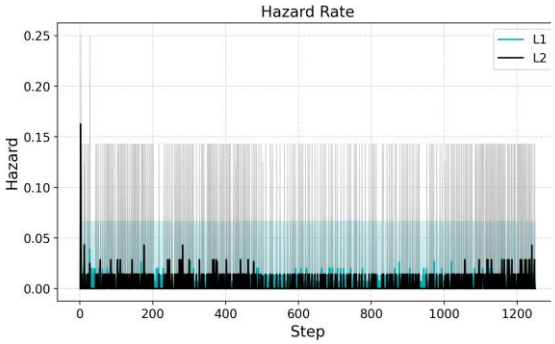

Figure 12: Dormancy hazard for dense synthetic experiment.

The fuzzy dense run in Figure 13 adds a gradual risk view to the crisp Dormancy Score. The final average fuzzy risk is 0.239 ± 0.039. At the final step, most monitored units are assigned to the low-risk category, with mean category counts of 23.0 low-risk, 0.4 medium-risk, and 0.6 high-risk neurons. This is an important distinction: the corrected activation baseline remains almost silent at the final step, while the fuzzy system still separates a small number of units into elevated-risk categories. In a future adaptive version of NeuroSentry, medium-risk units would be natural candidates for delayed

intervention, while high-risk units would be candidates for pruning, reinitialization, or rejuvenation.

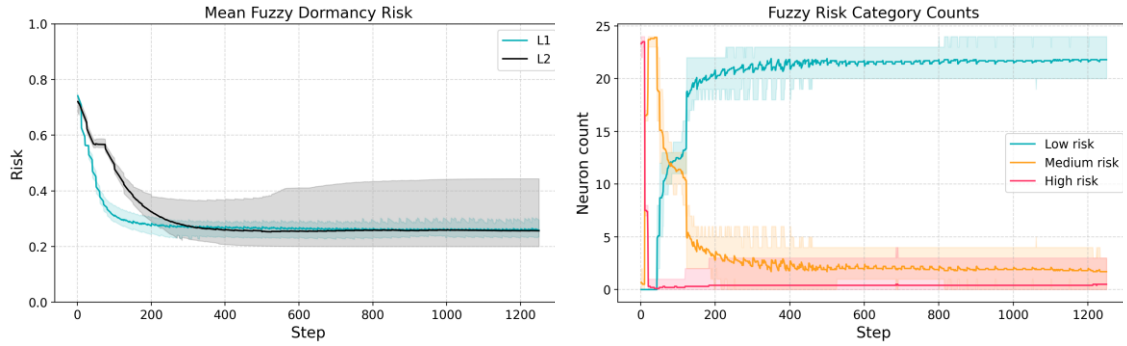

Figure 13: Dense synthetic fuzzy aggregate showing gradual risk and low/medium/high risk assignments.

The dense sensitivity analysis in Figures 14 and 15 shows why the four Dormancy Score components should not be reduced to a single activation-rate heuristic. With the corrected baseline, Sokar remains essentially flat at 0.94% across weight profiles because it does not use the Dormancy Score weights. NeuroSentry varies within a narrower but still meaningful band: the final dormant fraction ranges from 9.06% for the gradient–Taylor-only profile to 10.94% for the rate–magnitude-only profile, with the equal configuration at 9.69%. The larger sensitivity effect appears in the temporal statistics: rate-only monitoring gives the lowest final survival, 0.244, indicating that forward firing frequency strongly affects when units enter persistent dormancy even when the final dormant percentage changes moderately.

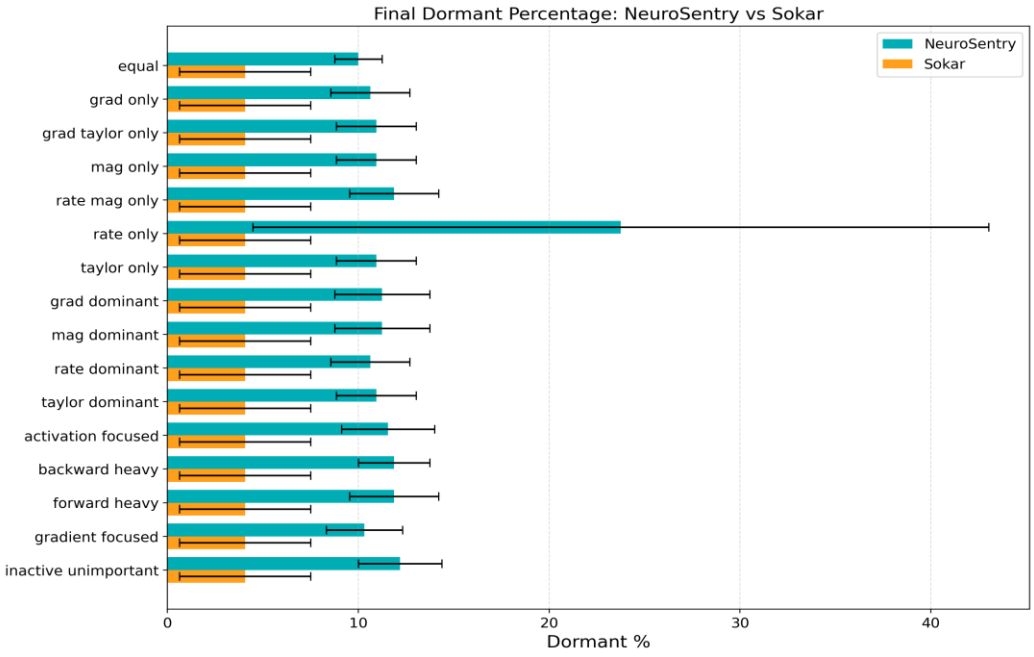

Figure 14: Dense synthetic sensitivity: final dormant percentage for NeuroSentry and the Sokar-style activation baseline.

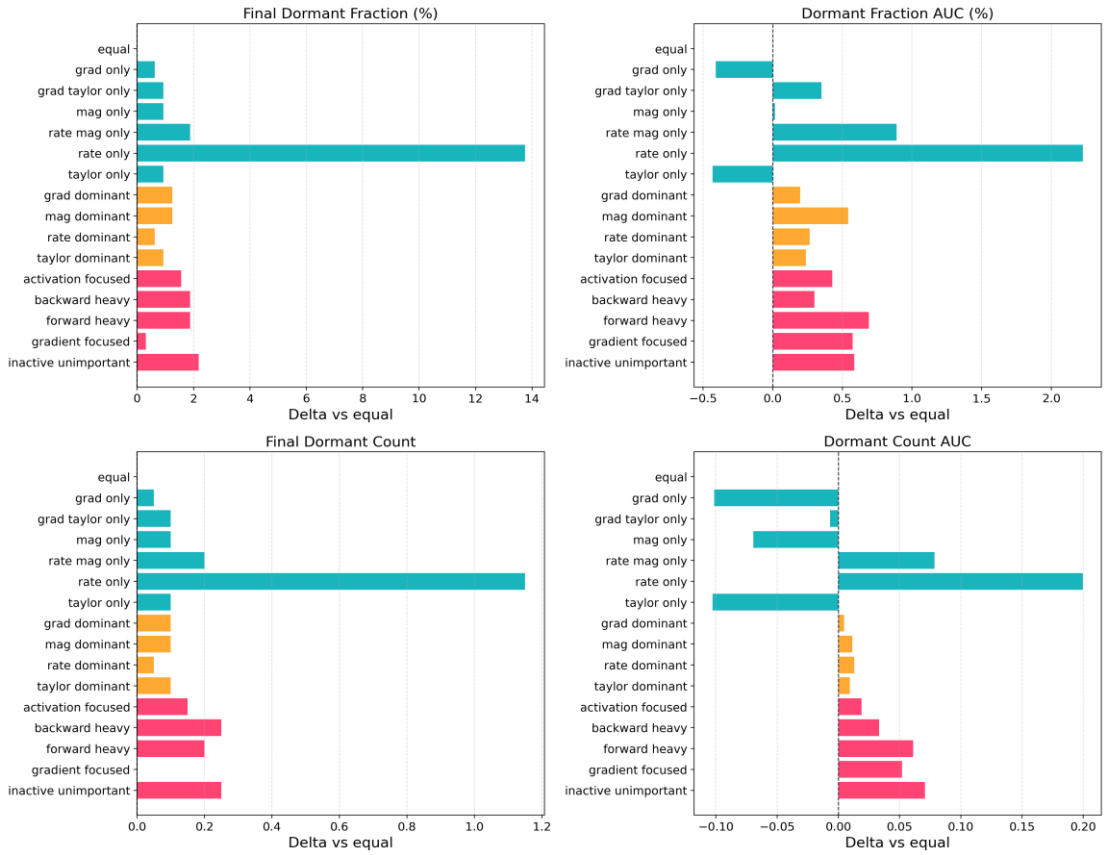

Figure 15: Dense synthetic sensitivity: deviation from equal Dormancy Score weights for final dormant fraction, dormancy AUC, dormant count, and count AUC.

The fuzzy sensitivity view across the same score-weight scenarios is visualized on Figures 16 and 17. The fuzzy-risk mean is nearly unchanged across the weight profiles, and the medium/high category counts are also stable. This is expected in the dense synthetic rerun because the fuzzy system evaluates the monitored feature state through its rule base, whereas the crisp Dormancy Score changes the weighting of the decision boundary. The result supports the thesis argument that fuzzy risk can act as a stable complementary health signal even when the exact Dormancy Score weights are varied for sensitivity analysis.

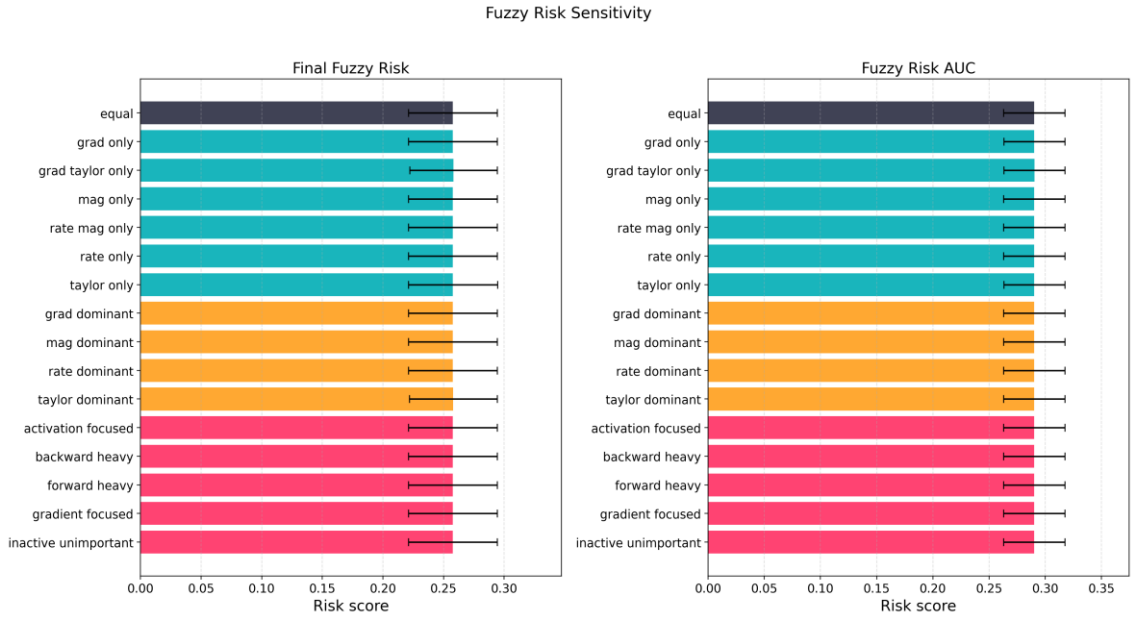

Figure 16: Dense synthetic fuzzy-risk sensitivity across Dormancy Score weight profiles.

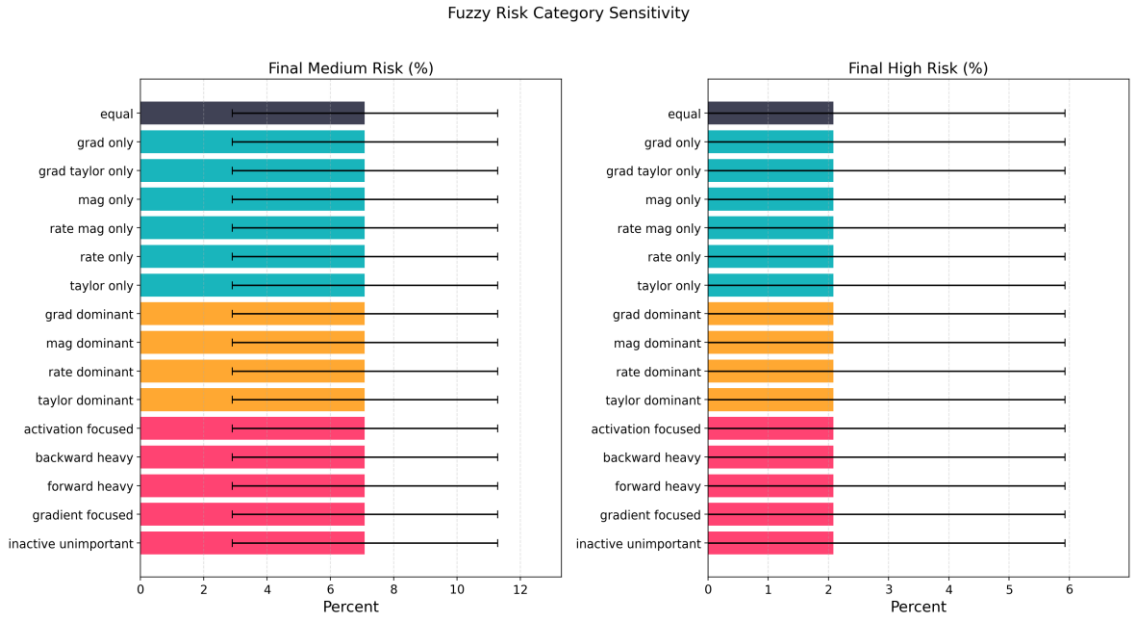

Figure 17: Dense synthetic fuzzy category sensitivity across Dormancy Score weight profiles.

5.2.6 CNN-MNIST Experiment

The CNN-MNIST experiment tests whether the same monitoring ideas remain useful when activations are spatially structured. In convolutional layers, a channel can be active in only some spatial locations, so layer-level monitoring aggregates over batch and spatial dimensions. This makes the CNN setting a stronger test of whether activation frequency alone is enough to describe neuron health.

Figure 18 summarizes the CNN aggregate. The equal-weight NeuroSentry configuration reports a final mean dormant fraction of $7.11\% \pm 1.19\%$, while the corrected activation-threshold baseline reports $2.37\% \pm 1.88\%$. The final survival value is 0.788 ± 0.098,

indicating that most monitored units remain active but that a consistent subset becomes underutilized. The gap between NeuroSentry and the activation baseline supports the central claim of the thesis: in convolutional representations, frequent or nonzero activation is not the same as functional importance.

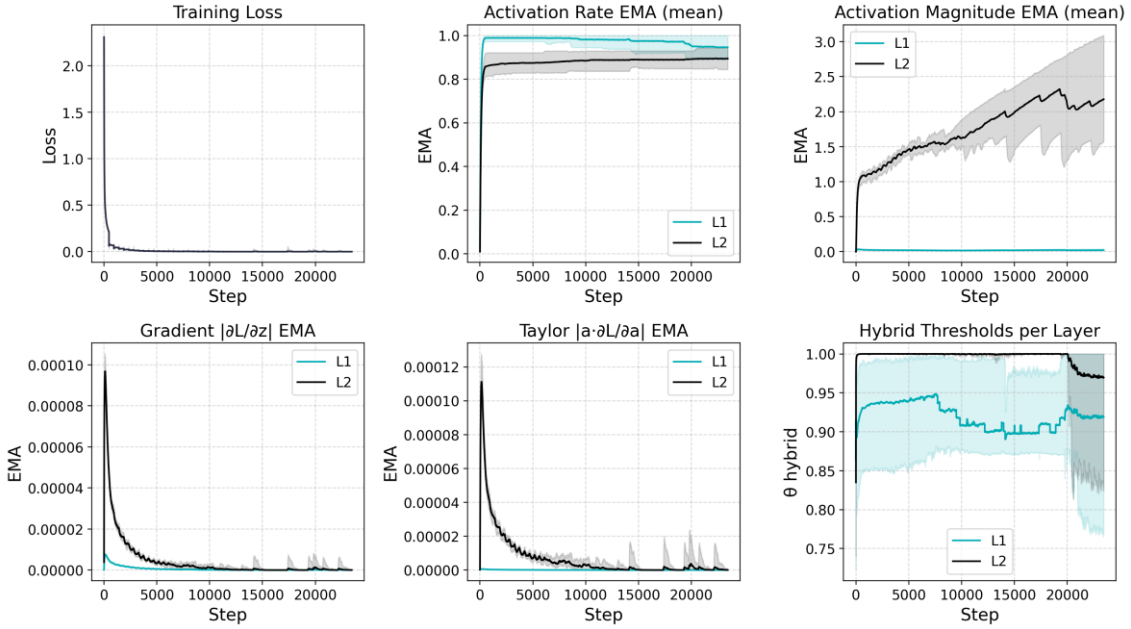

Figure 18: CNN-MNIST aggregate with layer-wise dormancy monitoring metrics.

The separate survival and hazard panels in Figure 19 show that the CNN model does not experience a large late-training increase in dormancy. Instead, the main event rate appears during the earlier part of optimization, after which the surviving monitored population is comparatively stable. This pattern is useful for interpretation because the final dormant count alone would not show whether dormancy accumulated gradually, occurred abruptly, or stabilized after early representation formation.

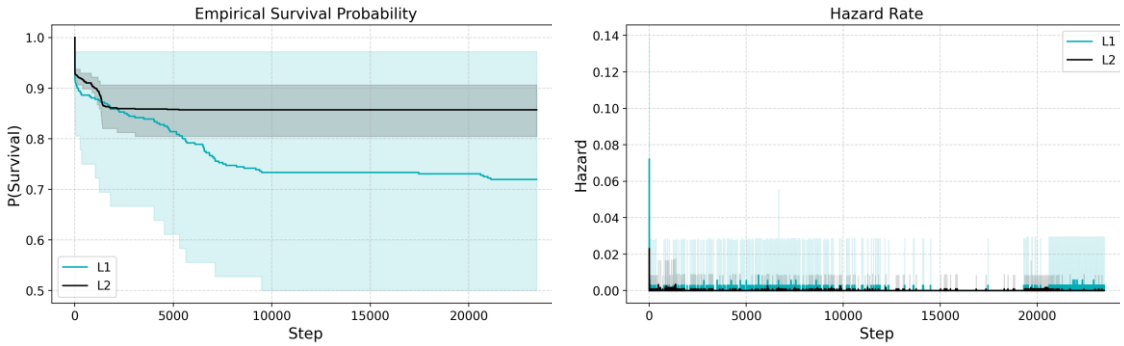

Figure 19: CNN-MNIST survival and hazard summaries.

The fuzzy CNN aggregate in Figure 20 reports a final mean fuzzy risk of 0.227 ± 0.010. The final category counts are 148.2 low-risk, 15.1 medium-risk, and 0.7 high-risk neurons across the monitored convolutional and dense layers. This result is qualitatively different from a binary dormant/not-dormant report. It indicates that the network contains a wider band of medium-risk units that are not yet persistently dormant but may deserve monitoring if the network is later pruned, expanded, or fine-tuned.

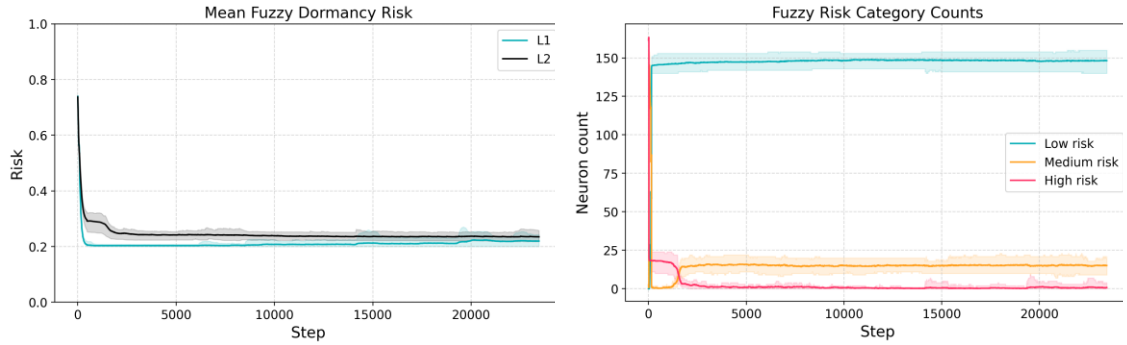

Figure 20: CNN-MNIST fuzzy aggregate showing gradual risk and low/medium/high risk assignments.

The sensitivity analysis in Figures 21 and 22 shows a different pattern from the dense experiment. The final dormant fraction ranges from 6.32% for the activation-focused profile to 8.73% for Taylor-only, while the corrected Sokar baseline stays lower, between 1.19% and 2.53% across the same trained networks. More importantly, the rate-only configuration has lower final survival, 0.434, even though its final dormant fraction is close to the equal profile. This suggests that activation-rate-only monitoring can create substantial transient dormancy in convolutional settings. Backward and Taylor information help distinguish temporary low firing from neurons that are consistently low impact.

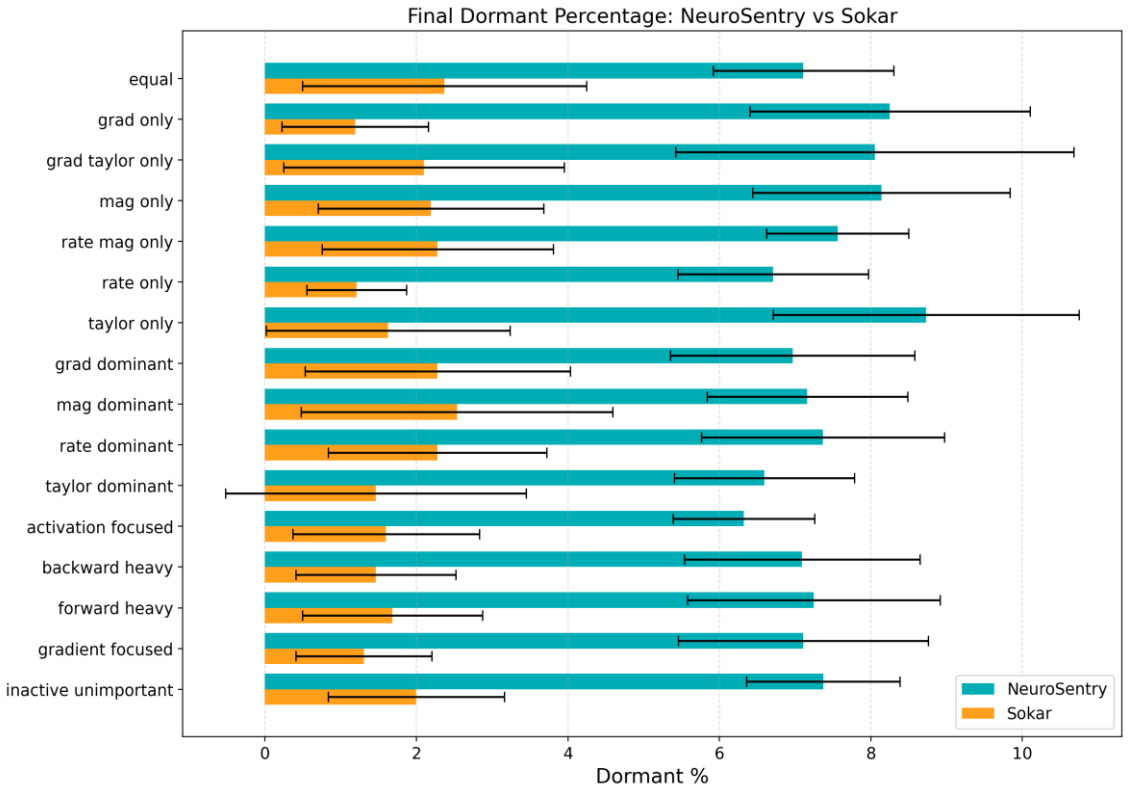

Figure 21: CNN-MNIST sensitivity: final dormant percentage for NeuroSentry and the Sokar-style activation baseline.

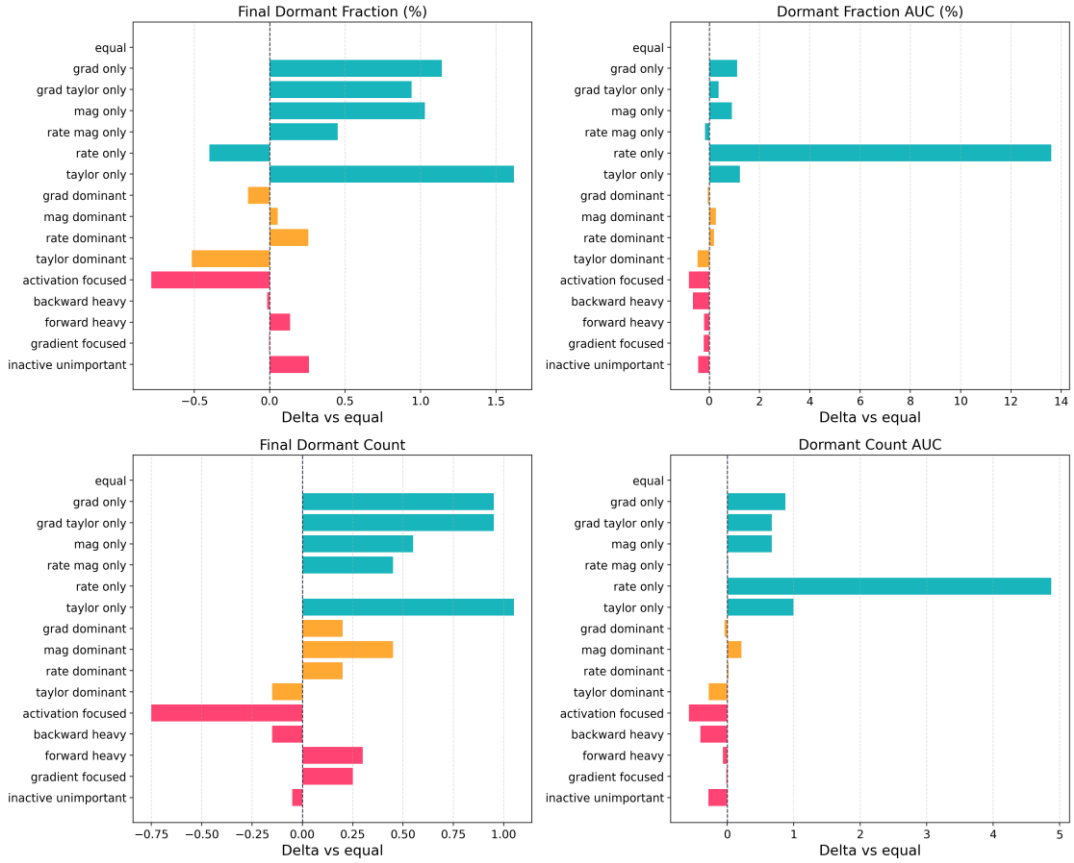

Figure 22: CNN-MNIST sensitivity: deviation from equal Dormancy Score weights for final dormant fraction, dormancy AUC, dormant count, and count AUC.

The fuzzy sensitivity plots in Figures 23 and 24 show the same separation between crisp sensitivity and fuzzy risk. Across CNN weight profiles, the final fuzzy risk remains in a narrow band around 0.225–0.231, while medium-risk counts remain around 15–17 neurons and high-risk counts remain below one neuron on average. This indicates that the fuzzy extension is not merely a re-labeling of the weighted Dormancy Score. It provides a complementary, rule-based assessment of neuron health that is less sensitive to small changes in score weighting.

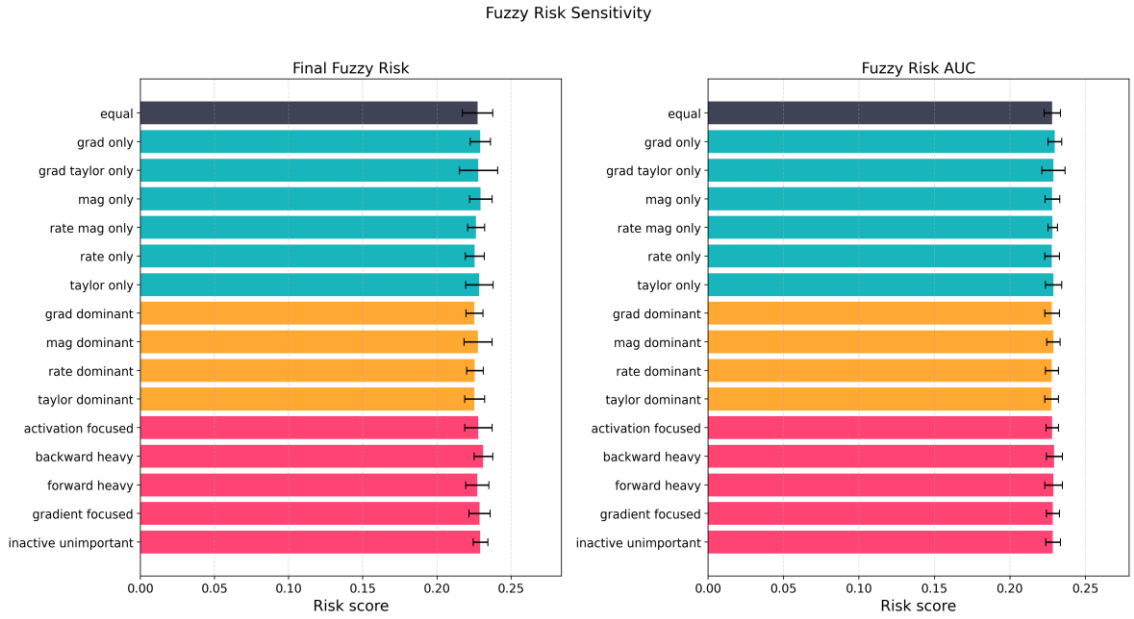

Figure 23: CNN-MNIST fuzzy-risk sensitivity across Dormancy Score weight profiles.

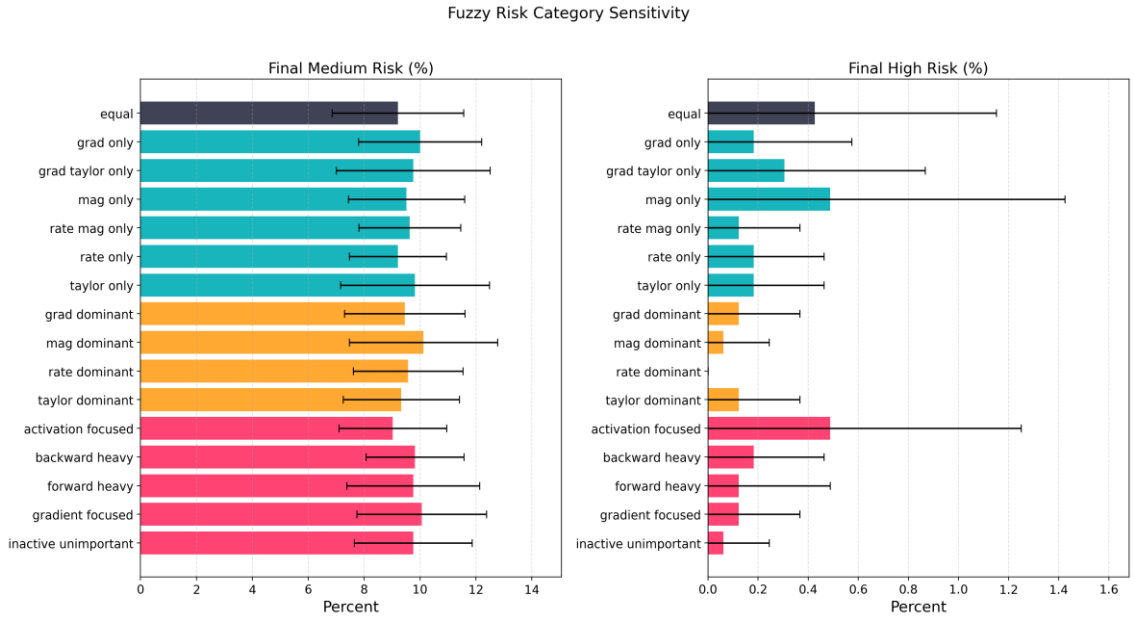

Figure 24: CNN-MNIST fuzzy category sensitivity across Dormancy Score weight profiles.

5.2.7 Sequential MNIST RNN Experiment

The recurrent experiment extends the supervised evaluation from static inputs to ordered inputs. Each MNIST image is treated as a sequence of 28 row vectors with 28 features per time step. The model monitors the output of a SimpleRNN layer and a dense classifier block, which gives 128 monitored units across recurrent and feed-forward representations. This setting is useful because recurrent activity can be temporarily low for reasons that are different from dense or convolutional layers: a hidden unit may specialize for particular time positions, digit strokes, or sequence contexts.

Figure 25 summarizes the Sequential MNIST aggregate. The equal-weight NeuroSentry configuration reports a final dormant fraction of $7.34\% \pm 1.68\%$, while the corrected

Sokar-style activation baseline reports $2.58\% \pm 0.99\%$. This revised comparison changes the interpretation of the recurrent result: the activation-only score identifies a smaller final dormant subset, while NeuroSentry reports additional underutilized units because it combines forward activity with gradient and Taylor evidence. In a recurrent layer, this distinction matters because a unit can remain intermittently active while contributing little to the loss over the monitored window.

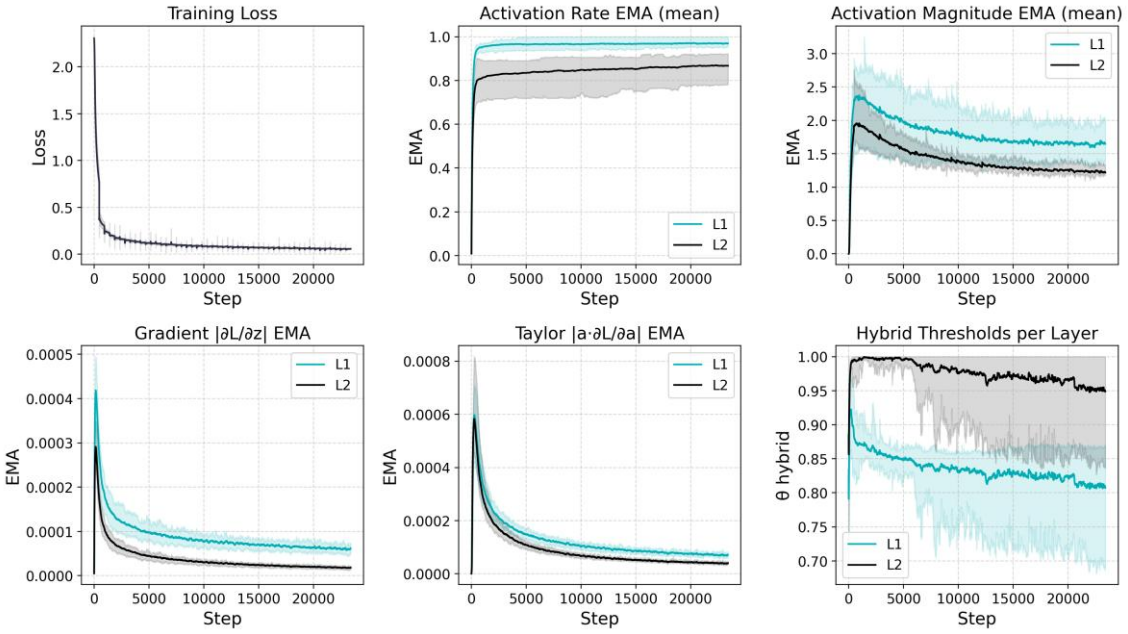

Figure 25: Sequential MNIST RNN aggregate with recurrent and dense layer dormancy monitoring metrics.

Figure 26 gives the corresponding temporal view for the recurrent experiment. The survival curve remains higher than in the dense synthetic setting, while the hazard curve still shows that dormancy events are not uniformly distributed over training. This supports the use of survival and hazard in sequence models, where the final dormant percentage can look modest even when the timing of underuse differs across optimization.

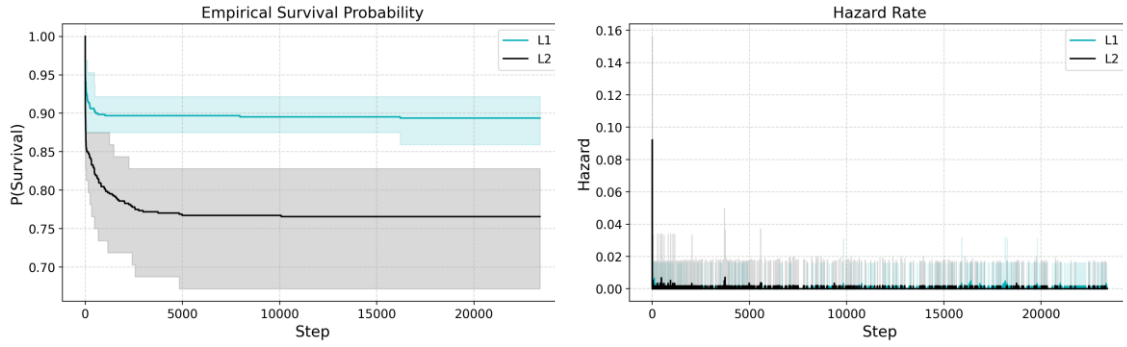

Figure 26: Sequential MNIST RNN survival and hazard summaries.

The fuzzy RNN aggregate in Figure 27 reports a final mean fuzzy risk of 0.232 ± 0.010. The final risk categories contain 117.1 low-risk, 9.0 medium-risk, and 1.9 high-risk units. The medium-risk group is small but consistent, indicating that the recurrent model

contains a narrow band of units that are not persistently dormant but still show weaker neuron-health signals than the low-risk majority.

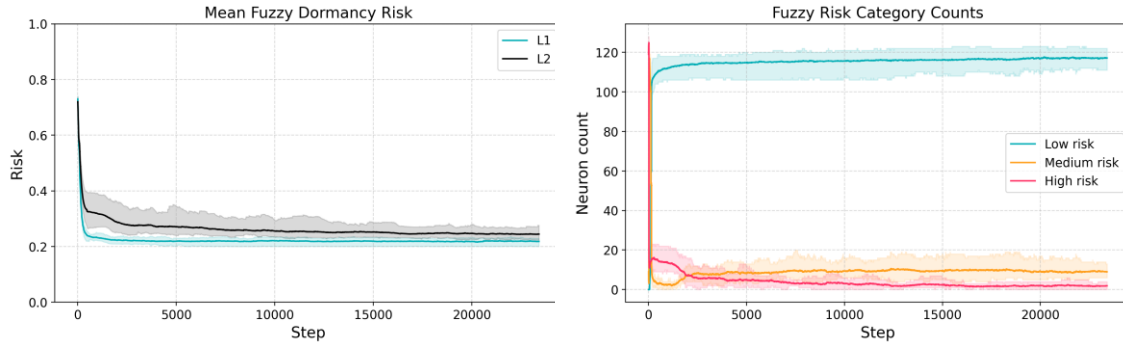

Figure 27: Sequential MNIST RNN fuzzy aggregate showing gradual risk and low/medium/high risk assignments.

The RNN sensitivity analysis in Figures 28 and 29 shows that the recurrent setting is less dominated by activation rate than the dense synthetic task. The equal-weight final dormant fraction is 7.34%, rate-only gives 7.11%, forward-heavy gives the lowest value at 6.56%, and Taylor-only gives the highest value at 8.59%. The stronger effect appears in survival: the rate-only profile ends at 0.659, while the equal-weight profile ends at 0.830. In other words, activation-rate-only monitoring does not simply increase the final dormant count; it changes the timing and persistence of dormancy events. This supports the use of survival and hazard curves alongside final counts in sequence models.

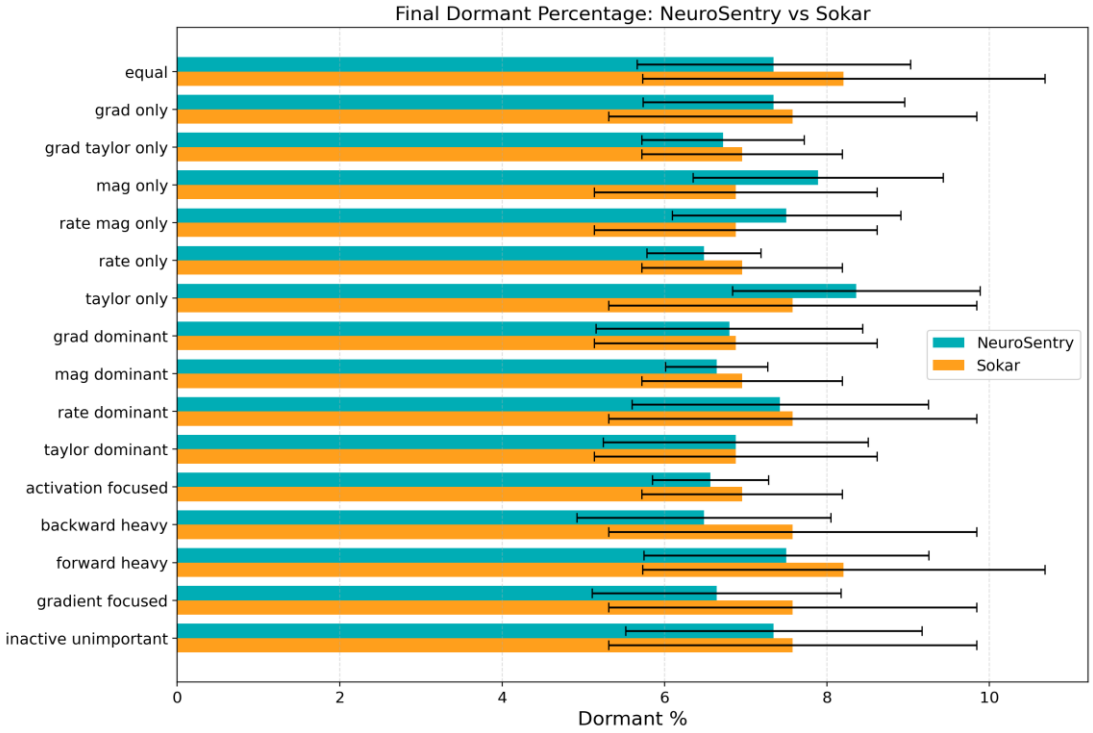

Figure 28: Sequential MNIST RNN sensitivity: final dormant percentage for NeuroSentry and the Sokar-style activation baseline.

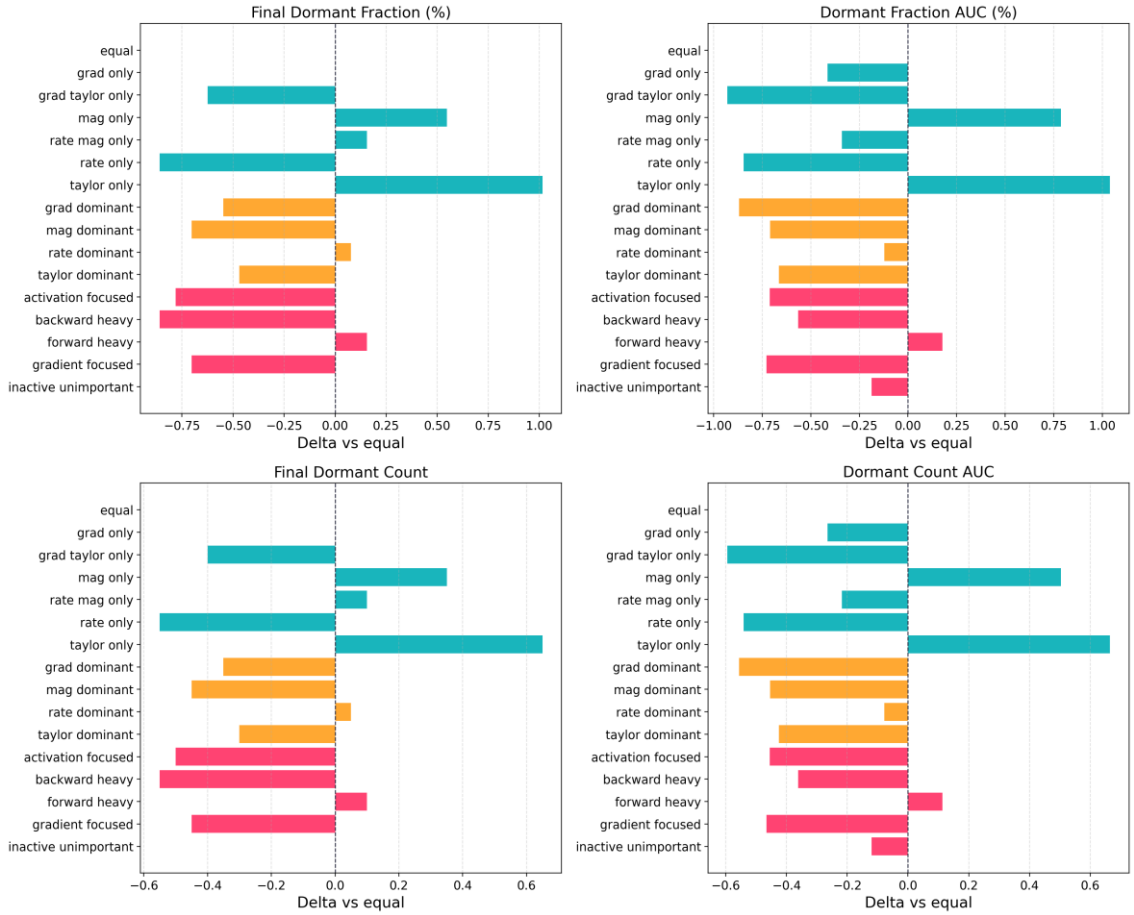

Figure 29: Sequential MNIST RNN sensitivity: deviation from equal Dormancy Score weights for final dormant fraction, dormancy AUC, dormant count, and count AUC.

Figures 30 and 31 report the fuzzy counterpart for the recurrent setting. The fuzzy-risk values stay within a narrow range across the same weight profiles, while the category plot shows that the recurrent model retains a small but visible high-risk group of roughly one to two units. This complements the survival result: the recurrent model does not show a large final dormant population, but the fuzzy system still separates low-risk units from units that deserve continued monitoring.

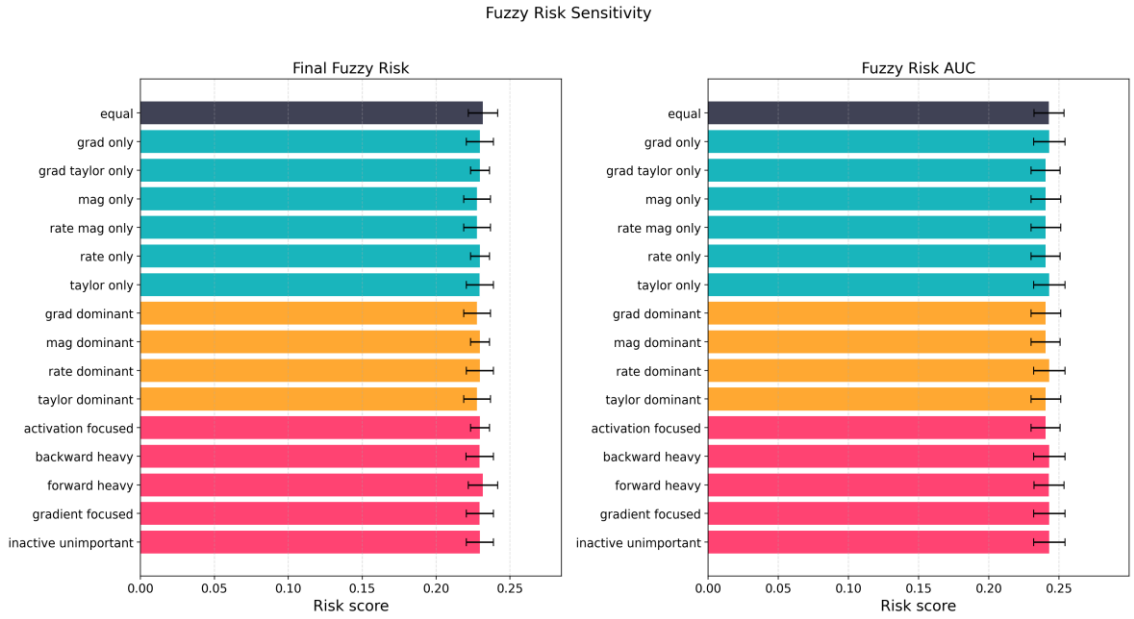

Figure 30: Sequential MNIST RNN fuzzy-risk sensitivity across Dormancy Score weight profiles.

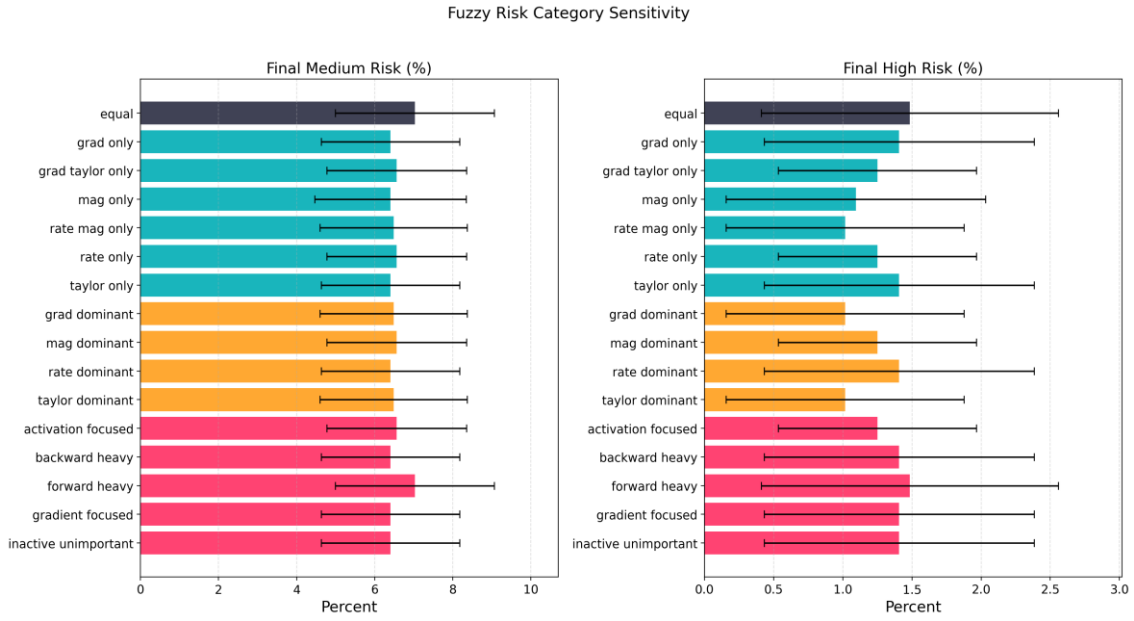

Figure 31: Sequential MNIST RNN fuzzy category sensitivity across Dormancy Score weight profiles.

5.3 Reinforcement Learning Experiments

The final reinforcement-learning study replaces the earlier episode-limited monitoring runs with a matched six-million-environment-step comparison. CartPole-v1 and LunarLander-v3 use DQN, while Humanoid-v5 uses DDPG. Each task is evaluated with seeds 0, 1, and 2 under four modes: no monitoring, the activation-only Sokar/ReDo baseline, NeuroSentry Dormancy Score monitoring, and NeuroSentry with fuzzy risk. All 36 runs completed the target budget. The comparison therefore evaluates monitoring

behavior and computational cost at a common interaction budget rather than at task-dependent episode counts.

The monitoring modes are observational: none prunes, resets, or otherwise changes neurons in response to the measured state. Reward is reported to characterize the training outcomes, but differences between modes must not be interpreted as causal benefits of a monitor. The modes are separate stochastic training runs, and the three-seed sample remains too small for strong policy-performance claims.

5.3.1 Configuration and Evaluation Protocol

The common comparison and evaluation protocol is presented in the Table 7. Sokar/ReDo evaluates normalized activation every 1000 environment steps using $\tau = 0.025$. NeuroSentry runs after training updates with equal Dormancy Score weights, and the fuzzy mode additionally computes the low-, medium-, and high-risk categories. Because four parallel environments are used, the NeuroSentry histories contain approximately 1.5 million monitoring points for each six-million-step run.

Table 7: Configuration of the matched six-million-step reinforcement-learning comparison.

Parameter	**CartPole DQN**	**LunarLander DQN**	**Humanoid DDPG**
Environment	CartPole-v1	LunarLander-v3	Humanoid-v5
Seeds	0, 1, 2	0, 1, 2	0, 1, 2
Environment steps per run	6,000,000	6,000,000	6,000,000
Parallel environments	4	4	4
Monitored neurons	64	320	2500
Batch size	64	64	256
Minimum replay size	1000	1000	5000
NeuroSentry weights	0.25/0.25/0.25 /0.25	0.25/0.25/0.25 /0.25	0.25/0.25/0.25 /0.25
Sokar/ReDo threshold	$\tau = 0.025$	$\tau = 0.025$	$\tau = 0.025$
Sokar evaluation interval	1000 steps	1000 steps	1000 steps

5.3.2 Final Dormancy and Fuzzy-Risk Results

In Table 8 and Figure 32 we can see the central diagnostic result. On CartPole, the activation-only baseline and NeuroSentry identify comparable final dormant fractions. The result changes sharply on the larger tasks. Sokar/ReDo reports no dormant neurons on LunarLander or Humanoid, while NeuroSentry reports roughly one third of the LunarLander hidden units and roughly one tenth of the monitored Humanoid actor–critic units as dormant.

The zero Sokar/ReDo values are not missing or failed experiments. Each of the six LunarLander and Humanoid Sokar/ReDo runs completed all 6,000 scheduled activation evaluations, and every evaluation returned zero neurons below $\tau = 0.025$. This means the monitored ReLU units continued to activate often enough to remain above the

activation-only threshold. NeuroSentry nevertheless identifies underused units because low gradient activity, low activation magnitude, and low Taylor importance can provide evidence of dormancy even when activation frequency remains nonzero. The result therefore demonstrates the specific diagnostic gap that the multi-signal score is designed to address.

Table 8: Final dormant-neuron counts and monitored-population percentages after six million environment steps, reported as mean ± sample standard deviation over three seeds.

Task	Sokar/ReDo dormant	NeuroSentry score dormant	NeuroSentry + fuzzy dormant
CartPole-v1	10.3±1.5 (16.1%)	12.0±4.4 (18.8%)	10.0±2.0 (15.6%)
LunarLander-v3	0.0±0.0 (0.0%)	112.7±11.2 (35.2%)	122.7±10.7 (38.3%)
Humanoid-v5	0.0±0.0 (0.0%)	246.7±4.7 (9.9%)	274.0±37.3 (11.0%)

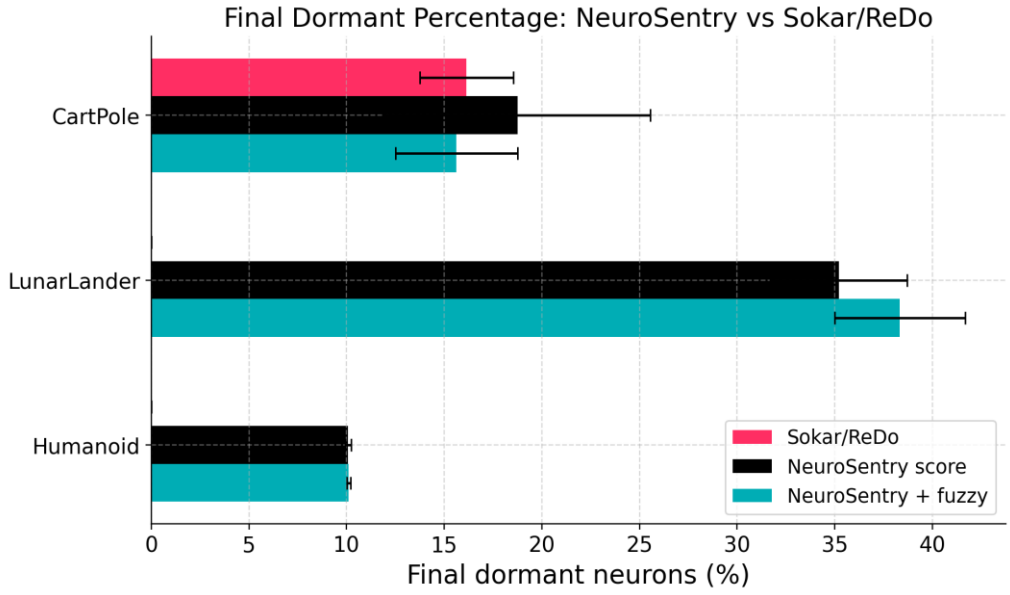

Figure 32: Final dormant fraction in the six-million-step RL comparison. Error bars show sample standard deviation over three seeds.

The fuzzy categories refine the binary result. From the diagnostics depicted in Table 9 we understand that the final high-risk group remains smaller than the dormant group in all three tasks. LunarLander has a particularly broad medium-risk population, while Humanoid contains a large medium-risk population in absolute terms but only a small high-risk fraction. The fuzzy output therefore does not simply reproduce the dormant count; it distinguishes persistent multi-signal underuse from stronger evidence of functional risk.

Table 9: Final NeuroSentry fuzzy-mode diagnostic counts after six million environment steps.

Task	Dormant count	Low risk	Medium risk	High risk
CartPole-v1	10.0±2.0	54.3±2.5	8.7±4.2	1.0±1.7
LunarLander-v3	122.7±10.7	140.3±10.8	155.0±19.7	24.7±11.7
Humanoid-v5	274.0±37.3	1777.3±145.8	623.3±119.5	99.3±26.3

The fuzzy-mode runs selected for final dormant counts can be found on Figure 33, 34 and 35. They are included to expose temporal behavior that the final tables cannot show; they are not substitutes for the three-seed summaries. LunarLander develops a much larger dormant population and a broad graded-risk distribution. Humanoid shows that the same instrumentation can track a 2500-neuron actor–critic representation through a long continuous-control run.

CartPole seed 2 is intentionally shown with an early-training zoom. Its dormant count and all three fuzzy category counts reach their final joint state after approximately 13,824 environment steps and remain unchanged for the rest of the six-million-step run. The zoom makes this rapid stabilization visible instead of compressing it into an apparently unexplained straight line. This behavior is specific to the representative seed: the other CartPole seeds retain small late changes, so the figure is not presented as a universal CartPole pattern.

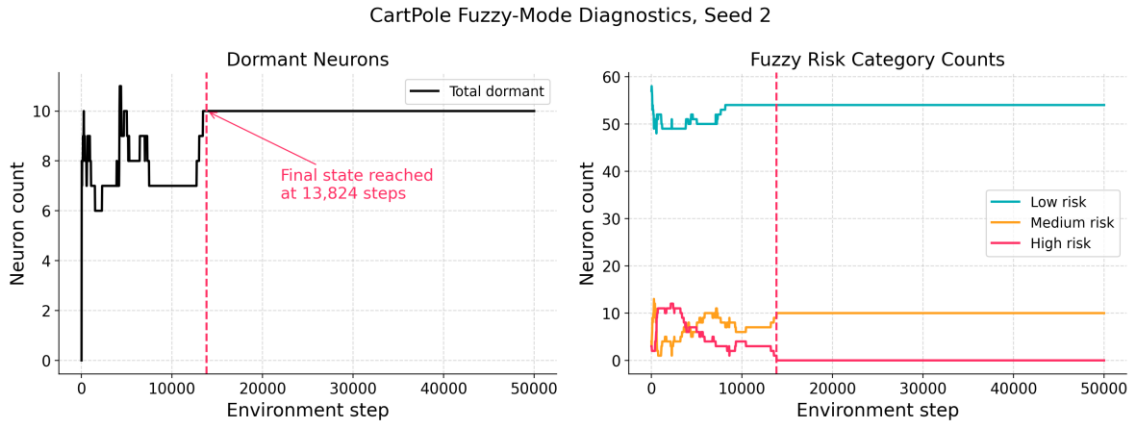

Figure 33: Representative CartPole fuzzy-mode trace, seed 2. The diagnostic state reaches its final value after approximately 13,824 environment steps.

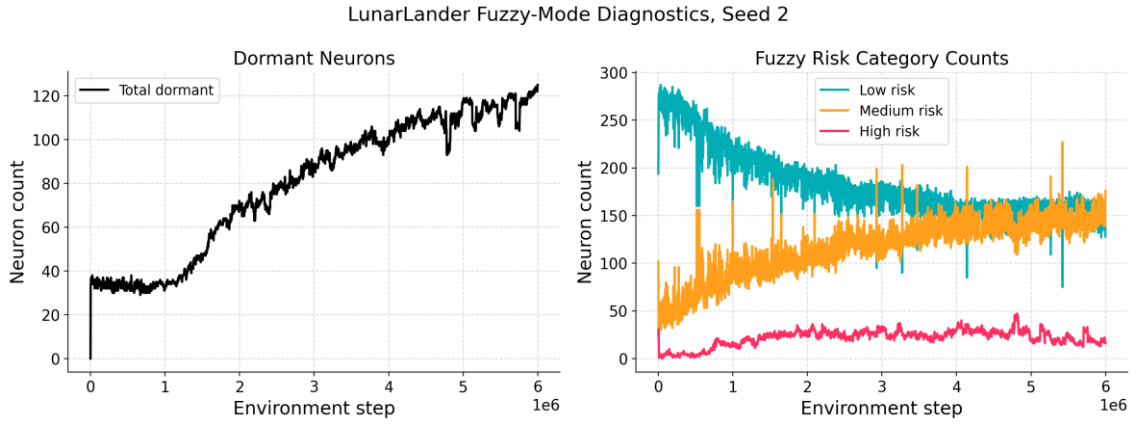

Figure 34: Representative LunarLander fuzzy-mode trace, seed 2.

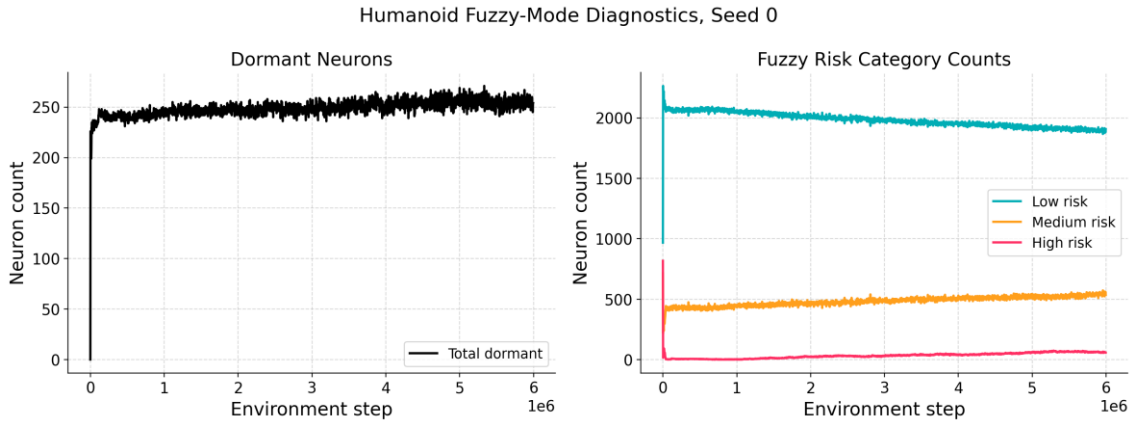

Figure 35: Representative Humanoid fuzzy-mode trace, seed 0.

5.3.4 Computational Cost

A GPU end-to-end wall-clock cost for the RL benchmark profile used in this thesis and is presented on the Table 10 and Figure 5.38. The normalization uses the current Humanoid GPU execution profile so that CartPole, LunarLander, and Humanoid are presented under a consistent execution regime. Within each task, Sokar/ReDo remains close to the no-monitor runtime because it evaluates activation only every 1000 environment steps. NeuroSentry is called after training updates, so the richer diagnostic state is accumulated much more frequently.

Table 10: GPU end-to-end wall-clock time for the RL experiments. Values are reported in hours for a six-million-environment-step budget.

Task	**No monitor (h)**	**Sokar/ReDo (h)**	**NeuroSentry score (h)**	**NeuroSentry + fuzzy (h)**
CartPole-v1	0.49	0.49	1.38	1.75
LunarLander-v3	0.64	0.65	1.81	2.30
Humanoid-v5	1.36	1.38	3.86	4.89

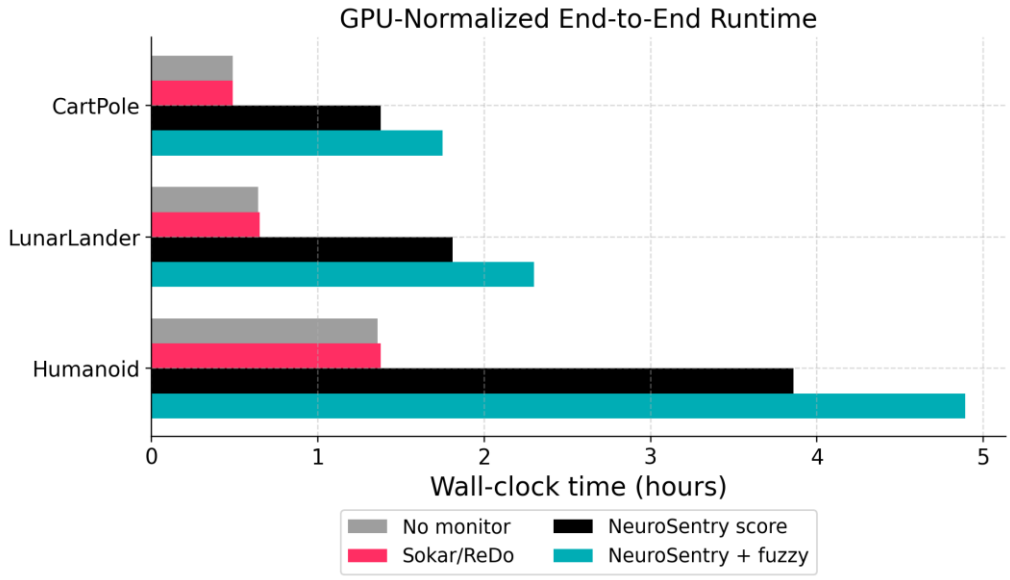

Figure 36: GPU end-to-end runtime for the RL experiments, including the no-monitor reference.

These values should be read as a scheduling result rather than as a claim that each NeuroSentry operation is slow. NeuroSentry accumulates richer diagnostics after training updates, while Sokar/ReDo is evaluated sparsely. The additional wall-clock cost is therefore mainly a consequence of monitoring frequency.

Additionally, Table 11 and Figure 38 report callback latency. A single NeuroSentry score update is faster than a Sokar/ReDo activation evaluation in all three tasks, and fuzzy inference adds a bounded incremental cost.

Table 11: Mean latency of one monitoring invocation, reported as mean ± sample standard deviation over three seeds.

Task	**Sokar/ReDo callback (ms)**	**NeuroSentry score (ms)**	**NeuroSentry + fuzzy (ms)**
CartPole-v1	5.89±0.05	1.83±0.09	2.57±0.02
LunarLander-v3	6.17±0.04	2.39±0.01	3.58±0.09
Humanoid-v5	12.18±0.10	3.75±0.01	6.17±0.04

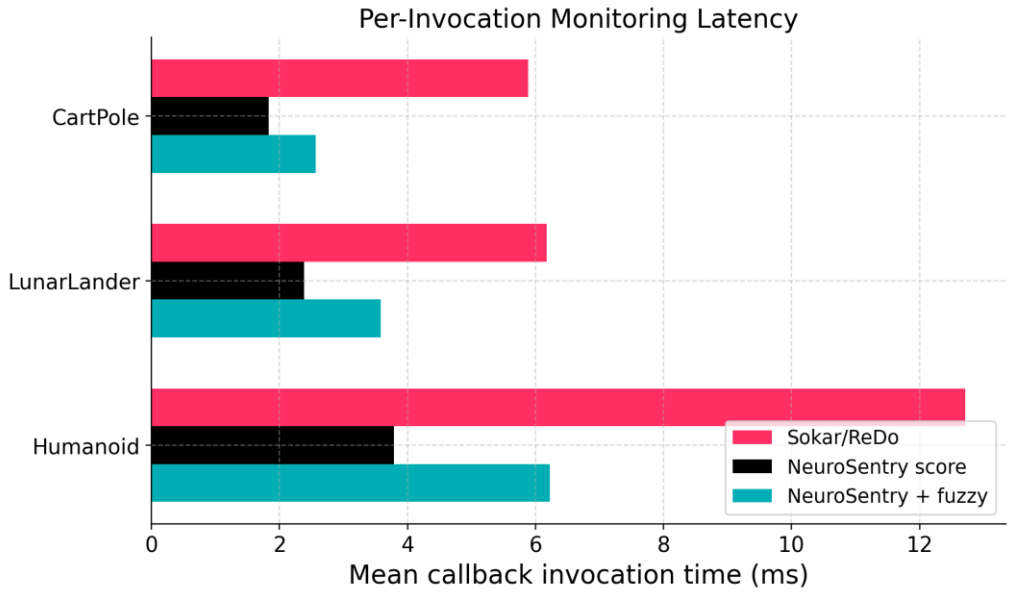

Figure 37: Per-invocation monitoring latency only. This plot should not be read as a total-runtime result; the no-monitor mode is omitted because it makes no monitoring invocation, and total runtime is shown separately in Figure 36.

5.4 Experimental Summary

The supervised experiments show that NeuroSentry provides a richer dormancy view than the corrected fixed activation-threshold baseline while preserving interpretability through survival and hazard curves. Dense synthetic results show that the corrected Sokar-style baseline is conservative at $\tau = 0.025$, while NeuroSentry still detects a stable small dormant population. CNN-MNIST results show that backward and Taylor signals matter when spatial activations make firing frequency less informative, and Sequential MNIST shows that recurrent aggregation can be evaluated with the same monitoring vocabulary. The fuzzy supervised aggregates add a graded view of neuron health, separating low-, medium-, and high-risk units rather than forcing every neuron into an immediate binary decision. The fuzzy sensitivity plots further show that fuzzy risk remains comparatively stable across Dormancy Score weight profiles, which makes it useful as a complementary health signal.

The matched six-million-step reinforcement-learning comparison strengthens the integration evidence and adds a direct activation-only baseline. Sokar/ReDo and NeuroSentry identify comparable dormancy on CartPole, but Sokar/ReDo reports no final dormant units on LunarLander or Humanoid while NeuroSentry identifies substantial dormant populations. This supports the multi-signal design, while the cost results show that monitoring frequency must be reduced for practical long-running RL use. These experiments remain observational: NeuroSentry monitors neuron health but does not prune, reinitialize, rejuvenate, or expand neurons during training.

6 Discussion and Conclusion

6.1 Discussion

The central argument of this thesis is that neuron dormancy should be treated as a multi-signal and temporal process rather than as a single activation-threshold event. Activation-based baselines provide a useful reference point, especially because they are simple and directly connected to prior work on dormant neurons in reinforcement learning . However, they cannot fully describe neuron health. A neuron can fire often but weakly, fire rarely but remain important, or stay active while receiving almost no learning signal. NeuroSentry addresses this limitation by combining forward activity, backward gradient flow, and first-order Taylor importance.

The supervised experiments support this view. In the dense synthetic setting, activation-rate-heavy configurations can identify substantially more dormant units than balanced configurations, suggesting that rare firing alone is too aggressive as a dormancy criterion. In the CNN-MNIST setting, gradient and Taylor components become more informative, which is consistent with the idea that spatial activation frequency does not directly imply structural usefulness. In the Sequential MNIST RNN setting, the activation-only baseline can be slightly higher than NeuroSentry at the final step, showing the complementary case where multi-signal monitoring can be more conservative because low activation does not necessarily imply low functional contribution. These differences across architectures are a major reason for using a composite score rather than a single universal heuristic.

The survival and hazard summaries provide another useful perspective. Dormancy often appears early, during the phase where representations specialize and the optimizer quickly allocates capacity. A final dormant fraction alone hides this timing. Hazard curves reveal when new dormant events occur, while survival curves summarize how much of the active population remains. This makes the monitor more informative for adaptive systems: an intervention early in training may have a different meaning from one applied after the network has stabilized.

The synthetic fuzzy experiments clarify the conceptual role of the method. The life-cycle scenario shows how risk can rise gradually during decay and before the final near-zero activation phase. The hysteresis scenario shows why a dormancy monitor should not react to every short dip. The latency analysis suggests that fuzzy risk can detect deterioration earlier than a fixed activation threshold in controlled trajectories. Together, these examples connect the biological motivation of plasticity and dormancy to concrete monitoring behavior. Furthermore, supervised fuzzy experiments add an additional practical observation. Across Dormancy Score weight profiles, the fuzzy-risk summaries are more stable than the crisp dormant counts. This is useful because it separates two roles of the monitor: the Dormancy Score can be tuned for a specific decision boundary, while fuzzy risk can remain a smoother health indicator for early warning and delayed intervention.

The reinforcement-learning experiments are important because they test NeuroSentry outside standard supervised callbacks. DQN and DDPG training are non-stationary: the replay distribution, value targets, exploration behavior, and policy all change throughout training. In the matched six-million-step comparison, Sokar/ReDo and NeuroSentry identify similar final dormant fractions on CartPole, but the activation-only method reports no final dormant units on LunarLander or Humanoid. NeuroSentry identifies approximately 35–38% of LunarLander units and 10% of Humanoid units as dormant because the composite score also considers magnitude, gradient, and Taylor signals. This is the strongest evidence in the thesis for the practical value of multi-signal monitoring.

The same comparison also exposes an engineering limitation. A NeuroSentry callback invocation is individually faster than a Sokar/ReDo evaluation, but NeuroSentry is invoked after almost every training update rather than every 1000 environment steps. This produces large end-to-end overhead on the two DQN tasks. The Humanoid wall-clock measurements are anomalous and are not suitable for an overhead percentage claim, which reinforces the need for controlled hardware conditions and reduced-frequency monitoring.

6.2 Limitations

On this thesis, contains also several limitations that should be addressed:

- The experiments are observational. They show dormancy and fuzzy-risk dynamics, but they do not yet test pruning, reinitialization, rejuvenation, or expansion triggered by NeuroSentry.
- The current Dormancy Score weights are interpretable but heuristic. The sensitivity analysis shows that weights matter, so future work should consider task-adaptive or validation-driven weighting.
- The fuzzy rule base is manually designed. This is useful for interpretability, but it should eventually be compared with learned neuro-fuzzy tuning or data-driven calibration.
- The experiments evaluate diagnostic behavior rather than causal performance improvement. Stronger claims require intervention studies that compare final accuracy, reward, stability, and compute cost with and without NeuroSentry-triggered adaptation.
- The sequence evidence uses Sequential MNIST as a controlled recurrent benchmark. Longer temporal tasks, language models, and transformer feed-forward blocks remain important next targets.

6.3 Future Work

The most direct next step is to use NeuroSentry as a control signal rather than only as a monitor. Persistently high-risk neurons could trigger adaptive pruning, reinitialization, rejuvenation, or expansion decisions. Medium-risk neurons could be watched for longer

before intervention. Low-risk but old specialist neurons could be protected. This would turn the current diagnostic framework into an adaptive architecture mechanism with explicit intervention triggers.

A second direction is broader sequence and language modeling. The Sequential MNIST experiment verifies the basic batch-time aggregation path, but recurrent layers and transformer feed-forward blocks in longer tasks may show different dormancy behavior because their activity depends strongly on token distribution and context. The next step is to move from controlled image-row sequences to longer temporal datasets and, eventually, language-model components.

A third direction is stronger reinforcement-learning intervention. The current thesis includes matched six-million-step CartPole, LunarLander, and Humanoid comparisons, but these experiments do not change the network in response to dormancy. The next step is to align dormancy events with reward plateaus and test whether pruning, reinitialization, rejuvenation, or expansion improves learning when triggered by NeuroSentry. A reduced-frequency monitoring schedule should be evaluated at the same time so that intervention quality and computational cost are measured together.

Finally, the fuzzy component can be expanded. The current system uses interpretable membership functions and hand-designed rules. Future work could tune membership parameters from data, compare type-1 and type-2 fuzzy systems, or use fuzzy masses directly as uncertainty estimates for intervention decisions.

6.4 Conclusion

This thesis develops NeuroSentry as a neuron-level monitoring framework for identifying emerging dormancy in adaptive neural-network settings. The Dormancy Score combines activation frequency, activation magnitude, gradient activity, and Taylor importance, while survival and hazard metrics describe the temporal evolution of inactive neurons. The fuzzy-risk extension adds interpretability and early-warning behavior by modeling dormancy as a gradual risk state.

The completed experiments show that multi-signal monitoring provides more diagnostic information than fixed activation thresholds. Dense, convolutional, and recurrent networks exhibit different sensitivity profiles, which reinforces the need for a balanced score. Synthetic fuzzy life-cycle scenarios show how gradual risk, recovery, and hysteresis can be represented clearly. The reinforcement-learning comparison also demonstrates that the same monitoring tools can be integrated beyond standard supervised training.

Overall, NeuroSentry provides a foundation for future adaptive mechanisms such as neuron rejuvenation, pruning, freezing, and neurogenesis-inspired expansion. Its main contribution is the measurement layer: before a network can adapt its structure responsibly, it needs a reliable way to understand which neurons are healthy, which are drifting toward dormancy, and which may be safely modified.

Bibliography

1. Benedetti, Bruno, and Sebastien Couillard-Despres. 2022. “Why Would the Brain Need Dormant Neuronal Precursors?” *Frontiers in Neuroscience* 16: 877167.
2. Benedetti, Bruno, Maximilian Reisinger, Marie Hochwartner, et al. 2023. “The Awakening of Dormant Neuronal Precursors in the Adult and Aged Brain.” *Aging Cell* 22 (12): e13974.
3. Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer. https://link.springer.com/book/9780387310732.
4. Blalock, Davis, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. “What Is the State of Neural Network Pruning?” *Proceedings of Machine Learning and Systems* 2: 129–46.
5. Campos Souza, Paulo Vitor de, and Edwin Lughofer. 2023. “Evolving Fuzzy Neural Classifier That Integrates Uncertainty from Human-Expert Feedback.” *Evolving Systems* 14 (2): 319–41.
6. Chambers, R Andrew, Marc N Potenza, Ralph E Hoffman, and Willard Miranker. 2004. “Simulated Apoptosis/Neurogenesis Regulates Learning and Memory Capabilities of Adaptive Neural Networks.” *Neuropsychopharmacology* 29 (4): 747–58.
7. Chechik, Gal, Isaac Meilijson, and Eytan Ruppin. 1998. “Synaptic Pruning in Development: A Computational Account.” *Neural Computation* 10 (7): 1759–77. https://doi.org/10.1162/089976698300017124.
8. Cheng, Hongrong, Miao Zhang, and Javen Qinfeng Shi. 2024. “A Survey on Deep Neural Network Pruning: Taxonomy, Comparison, Analysis, and Recommendations.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
9. Cox, D. R. 1972. “Regression Models and Life-Tables.” *Journal of the Royal Statistical Society. Series B (Methodological)* 34 (2): 187–220. http://www.jstor.org/stable/2985181.
10. Dietz, K, M Gail, K Krickeberg, J Samet, and A Tsiatis. 2000. “Statistics for Biology and Health.” *Logistic Regression A Self-Learning Text* 2: 102–24.
11. Donoho, David L, and Iain M Johnstone. 1994. “Ideal Spatial Adaptation by Wavelet Shrinkage.” *Biometrika* 81 (3): 425–55.
12. Douzandeh Zenoozi, Amirhossein, Laura Erhan, Antonio Liotta, and Lucia Cavallaro. 2025. “A Comparative Study of Neural Network Pruning Strategies for Industrial Applications.” *Frontiers in Computer Science* Volume 7 - 2025. https://doi.org/10.3389/fcomp.2025.1563942.
13. Draelos, Timothy J, Nadine E Miner, Christopher C Lamb, et al. 2017. “Neurogenesis Deep Learning: Extending Deep Networks to Accommodate New Classes.” *2017 International Joint Conference on Neural Networks (IJCNN)*, 526–33.

14. Dufort-Labbé, Simon, Pierluca D'Oro, Evgenii Nikishin, Razvan Pascanu, Pierre-Luc Bacon, and Aristide Baratin. 2024. "Maxwell's Demon at Work: Efficient Pruning by Leveraging Saturation of Neurons." *arXiv Preprint arXiv:2403.07688*.
15. Dunning, Ted. 2021. "The t-Digest: Efficient Estimates of Distributions." *Software Impacts* 7: 100049. https://doi.org/https://doi.org/10.1016/j.simpa.2020.100049.
16. Evci, Utku. 2018. "Detecting Dead Weights and Units in Neural Networks." *arXiv Preprint arXiv:1806.06068*.
17. Fan, Lixin. 2017. "Revisit Fuzzy Neural Network: Demystifying Batch Normalization and ReLU with Generalized Hamming Network." *Advances in Neural Information Processing Systems* 30.
18. Faust, Travis E, Georgia Gunner, and Dorothy P Schafer. 2021. "Mechanisms Governing Activity-Dependent Synaptic Pruning in the Developing Mammalian CNS." *Nature Reviews Neuroscience* 22 (11): 657–73.
19. French, Robert M. 1999. "Catastrophic Forgetting in Connectionist Networks." *Trends in Cognitive Sciences* 3 (4): 128–35.
20. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
21. Greenwald, Michael, and Sanjeev Khanna. 2001. "Space-Efficient Online Computation of Quantile Summaries." *ACM SIGMOD Record* 30 (2): 58–66.
22. Han, Bing, Feifei Zhao, Wenxuan Pan, and Yi Zeng. 2021. "Adaptive Sparse Structure Development with Pruning and Regeneration for Spiking Neural Networks." *Information Sciences* 689: 121481.
23. Han, Bing, Feifei Zhao, Yi Zeng, and Guobin Shen. 2021. "Developmental Plasticity-Inspired Adaptive Pruning for Deep Spiking and Artificial Neural Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
24. Jain, Raj, and Imrich Chlamtac. 1985. "The P2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations." *Commun. ACM* (New York, NY, USA) 28 (10): 1076–85. https://doi.org/10.1145/4372.4378.
25. Jang, J-SR. 1993. "ANFIS: Adaptive-Network-Based Fuzzy Inference System." *IEEE Transactions on Systems, Man, and Cybernetics* 23 (3): 665–85.
26. Junio Guimarães, Augusto, Paulo Vitor de Campos Souza, Vinícius Jonathan Silva Araújo, Thiago Silva Rezende, and Vanessa Souza Araújo. 2019. "Pruning Fuzzy Neural Network Applied to the Construction of Expert Systems to Aid in the Diagnosis of the Treatment of Cryotherapy and Immunotherapy." *Big Data and Cognitive Computing* 3 (2): 22.
27. Kingma, Diederik P., and Jimmy Ba. 2017. *Adam: A Method for Stochastic Optimization*. https://arxiv.org/abs/1412.6980.
28. Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, et al. 2017. "Overcoming Catastrophic Forgetting in Neural Networks." *Proceedings of the National Academy of Sciences* 114 (13): 3521–26.

29. LeCun, Yann A., Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. 2012. “Efficient BackProp.” In *Neural Networks: Tricks of the Trade: Second Edition*, edited by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_3.
30. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44. https://doi.org/10.1038/nature14539.
31. LeCun, Yann, John S. Denker, and Sara A. Solla. 1989. “Optimal Brain Damage.” In *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, edited by David S. Touretzky. http://papers.nips.cc/paper/250-optimal-brain-damage.
32. Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, et al. 2015. “Continuous Control with Deep Reinforcement Learning.” *arXiv Preprint arXiv:1509.02971*. https://arxiv.org/abs/1509.02971.
33. Liu, Jiashun, Johan Obando-Ceron, Aaron Courville, and Ling Pan. 2024. “Neuroplastic Expansion in Deep Reinforcement Learning.” *arXiv Preprint arXiv:2410.07994*.
34. Liu, Jiashun, Zihao Wu, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Ling Pan. 2025. “Measure Gradients, Not Activations! Enhancing Neuronal Activity in Deep Reinforcement Learning.” *arXiv Preprint arXiv:2505.24061*.
35. Lu, Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. 2020. “Dying ReLU and Initialization: Theory and Numerical Examples.” *Communications in Computational Physics* 28 (5): 1671–706. https://doi.org/10.4208/cicp.oa-2020-0165.
36. Mamdani, E. H., and S. Assilian. 1975. “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller.” *International Journal of Man-Machine Studies* 7 (1): 1–13. https://doi.org/https://doi.org/10.1016/S0020-7373(75)80002-2.
37. McCaffary, David. 2021. “Towards Continual Task Learning in Artificial Neural Networks: Current Approaches and Insights from Neuroscience.” *arXiv Preprint arXiv:2112.14146*.
38. McCulloch, Warren S., and Walter Pitts. 1943. “A Logical Calculus of the Ideas Immanent in Nervous Activity.” *The Bulletin of Mathematical Biophysics* 5 (4): 115–33. https://doi.org/10.1007/BF02478259.
39. Miconi, Thomas. 2017. “Biologically Plausible Learning in Recurrent Neural Networks Reproduces Neural Dynamics Observed During Cognitive Tasks.” *eLife* 6 (February): e20899. https://doi.org/10.7554/eLife.20899.
40. Ming, Guo-li, and Hongjun Song. 2011. “Adult Neurogenesis in the Mammalian Brain: Significant Answers and Significant Questions.” *Neuron* 70 (4): 687–702.
41. Mishra, Richa, and Manan Suri. 2023. “A Survey and Perspective on Neuromorphic Continual Learning Systems.” *Frontiers in Neuroscience* 17: 1149410.

42. Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, et al. 2015. “Human-Level Control Through Deep Reinforcement Learning.” *Nature* 518 (7540): 529–33. https://doi.org/10.1038/nature14236.
43. Molchanov, Pavlo, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016b. “Pruning Convolutional Neural Networks for Resource Efficient Inference.” *arXiv Preprint arXiv:1611.06440*.
44. Molchanov, Pavlo, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016a. “Pruning Convolutional Neural Networks for Resource Efficient Inference.” *arXiv Preprint arXiv:1611.06440*.
45. Molchanov, Pavlo, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. *Pruning Convolutional Neural Networks for Resource Efficient Inference*. https://arxiv.org/abs/1611.06440.
46. Pandit, Tej, and Dhireesha Kudithipudi. 2020. “Relational Neurogenesis for Lifelong Learning Agents.” *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*, 1–9.
47. Qiao, Siyuan, Zhe Lin, Jianming Zhang, and Alan L Yuille. 2019. “Neural Rejuvenation: Improving Deep Network Training by Enhancing Computational Resource Utilization.” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 61–71.
48. Rosenblatt, Frank. 1958. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” *Psychological Review* 65 (6): 386–408. https://doi.org/10.1037/h0042519.
49. Ross, Timothy J. 2005. *Fuzzy Logic with Engineering Applications*. John Wiley & Sons.
50. Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. “Learning Representations by Back-Propagating Errors.” *Nature* 323 (6088): 533–36. https://doi.org/10.1038/323533a0.
51. Shin, Yeonjong, and George Em Karniadakis. 2020. “Trainability of Relu Networks and Data-Dependent Initialization.” *Journal of Machine Learning for Modeling and Computing* 1 (1).
52. Sokar, Ghada, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. 2023. “The Dormant Neuron Phenomenon in Deep Reinforcement Learning.” *International Conference on Machine Learning*, 32145–68.
53. Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press. http://incompleteideas.net/book/the-book-2nd.html.
54. Tan, Woei Wan, and Teck Wee Chua. 2007. “Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions (Mendel, j.m.; 2001) [Book Review].” *IEEE Computational Intelligence Magazine* 2 (1): 72–73. https://doi.org/10.1109/MCI.2007.357196.
55. Tran, Lina M., Adam Santoro, Lulu Liu, Sheena A. Josselyn, Blake A. Richards, and Paul W. Frankland. 2022. “Adult Neurogenesis Acts as a Neural Regularizer.”

Proceedings of the National Academy of Sciences 119 (45): e2206704119. https://doi.org/10.1073/pnas.2206704119.

56. Turrigiano, Gina. 2012. "Homeostatic Synaptic Plasticity: Local and Global Mechanisms for Stabilizing Neuronal Function." *Cold Spring Harbor Perspectives in Biology* 4 (1): a005736.
57. Turrigiano, Gina G. 1999. "Homeostatic Plasticity in Neuronal Networks: The More Things Change, the More They Stay the Same." *Trends in Neurosciences* 22 (5): 221–27.
58. Turrigiano, Gina G, and Sacha B Nelson. 2004. "Homeostatic Plasticity in the Developing Nervous System." *Nature Reviews Neuroscience* 5 (2): 97–107.
59. Ven, Gido M van de, Nicholas Soures, and Dhireesha Kudithipudi. 2024. "Continual Learning and Catastrophic Forgetting." *arXiv Preprint arXiv:2403.05175*.
60. Watkins, Christopher J. C. H., and Peter Dayan. 1992. "Q-Learning." *Machine Learning* 8: 279–92. https://doi.org/10.1007/BF00992698.
61. Xue, Guangdong, Qin Chang, Jian Wang, Kai Zhang, and Nikhil Ranjan Pal. 2023. "An Adaptive Neuro-Fuzzy System with Integrated Feature Selection and Rule Extraction for High-Dimensional Classification Problems." *IEEE Transactions on Fuzzy Systems* 31 (7): 2167–81. https://doi.org/10.1109/TFUZZ.2022.3220950.
62. Zadeh, L. A. 1965. "Fuzzy Sets." *Information and Control* 8 (3): 338–53. https://doi.org/https://doi.org/10.1016/S0019-9958(65)90241-X.
63. Zimmermann, H-J. 2010. "Fuzzy Set Theory." *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (3): 317–32.