Online-CBGT-Net: A Neuromimetic Architecture for Online Prediction

Venkata Nagarjun Pudureddiyur Manivannan

CMU-RI-TR-25-49

June 2025

School of Computer Science
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Prof. Katia Sycara, *Chair*
Prof. Steven Chase
Renos Zabounidis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents and my sister,

Abstract

We introduce Online-CBGT-Net, a neuromimetic architecture for interpretable, online prediction in streaming environments. Inspired by the cortico-basal ganglia-thalamic (CBGT) circuit in the mammalian brain, our model integrates evidence accumulation and a reset mechanism for continuous, online decision making. At each time step, the model receives input and incrementally accumulates evidence. A decision is made only when the accumulated evidence surpasses a threshold, enabling the model to defer predictions in low-information conditions and act decisively when confident.

A key innovation is the incorporation of a reset factor, that scales down accumulated evidence after each decision. This allows the model to preserve temporal context across predictions to influence future predictions by past evidence. We further enhance interpretability by modeling separate Go and No-Go pathways using dual encoder branches, which produce supporting and opposing evidence for each class. This explicit separation allows users to inspect not only what the model favors, but also what it suppresses, mirroring inhibitory competition in the CBGT circuit of mammalian brain.

We evaluate Online-CBGT-Net on the CalMS21 behavior dataset and find that it consistently outperforms the LSTM baseline across a range of configurations in terms of F1-score and persistence. In its optimal configuration, Online-CBGT-Net surpasses the original CBGT-Net when compared in equivalent settings across F1-score, latency and persistence. Additionally, we analyze the internal evidence distributions, revealing how the Go/No-Go streams highlight the model's confidence in each decision. Our findings show that neuromimetic structures such as thresholded accumulation, reset mechanisms and dual-pathway encoding can produce transparent, confident, and biologically grounded decisions, advancing the goal of trustworthy AI in real-time sequential tasks.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Katia Sycara, for giving me the incredible opportunity to work under her guidance. Her knowledge, experience, and insightful advice have been instrumental in shaping my research skills and helping me grow both as a student and as a researcher. I am especially thankful for her patience and unwavering support throughout the program - across both my academic journey and research endeavors. One of the most valuable lessons I've learned from Prof. Sycara is the importance of keeping sight of the bigger picture in research. She taught me how to recognize when I was venturing too deep into a rabbit hole, and how to step back and refocus on broader goals. I am also grateful to her for the freedom to explore diverse research directions in the lab—from neuroscience-inspired machine learning to neural rendering techniques. That freedom was pivotal in helping me discover my interests and grow into a versatile researcher and engineer, capable of working across multiple domains and well-prepared for future challenges.

I would also like to express my sincere gratitude to my thesis committee members, Prof. Steven Chase and Renos Zabounidis, for their invaluable feedback and guidance, which were instrumental in helping me craft a successful thesis.

My heartfelt thanks go to my mentor, Dr. Dana Hughes, for his consistent support throughout my time at CMU. He stood by me at every step - helping me develop my research statement, explore new fields, write papers, prepare presentations, and ultimately defend my thesis. His mentorship was essential to my success in both the lab and the program.

I am grateful to Shreya for being a great friend and for being an incredible collaborator and friend throughout. I would also like to thank the members of my lab - Simon, Woojun, Yaqi, Joe, Srujan, Aryan, Silong, Sam, Zifu, Ce, and Sarthak - for their continued support in navigating both research challenges and the program itself.

To my friends Richa, Sahil, and Shreyansh - thank you for making my time at CMU and in Pittsburgh so memorable and fun. From long walks in Schenley Park to milkshake runs, Monopoly arguments, and late-night rants, your friendship helped me get through this journey.

My friends Elaiyabharathi, Yughander, Jwalika, Abhishek, Hirshik, Swati, Nandika, Swesha, Gokulan, GV, Viji, Lekha, Nandhitha, Rakshaa, Anuj - thank you for standing by me through thick and thin.

Finally, I am forever grateful to the people who have made this journey possible from day one - my family. I thank my mother for being my first teacher, for nurturing me, and for helping me grow into the person I am today. I am indebted to my sister for being my first role model; any success I've achieved began by imitating her. I am grateful to my father for showing me what it means to believe in oneself and persevere. I also thank Ammulu Maasi, Naveen Maama, Kumar Maama, Mahalakshmi, Nikkithaa and all my little cousins for their unconditional love and blessings.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Mammals exhibit the ability to make continuous decisions in dynamic, noisy environments by integrating historical context with newly arriving information [10, 17]. Rather than treating each decision in isolation, they use past observations and recent actions to bias future choices, enabling adaptive and context-sensitive behavior. This ability to deliberate over time to defer judgment until enough evidence has accumulated is a fundamental feature of intelligent biological systems.

Modern deep learning architectures designed for sequential data often emulate temporal reasoning using recurrent mechanisms such as LSTMs or GRUs [4], [5]. These networks maintain internal hidden states that evolve over time, allowing them to capture temporal dependencies and produce outputs at each time step. However, a key limitation of such models is that they are typically forced to make a prediction at every step, regardless of how much contextual evidence has accumulated. This can lead to premature decisions, where the model commits to a prediction before observing enough disambiguating context, and to low-confidence predictions, where the output distribution is diffuse or unstable due to insufficient information early in the sequence.

To address this, recent work has introduced neural architectures inspired by the cortico-basal ganglia-thalamic (CBGT) circuitry in the mammalian brain called CBGT-Net [16]. These models incorporate a biologically inspired decision-making mechanism that mimics how the brain integrates information over time before committing to an action. Specifically, they use an evidence accumulator - a component

that aggregates incoming features over multiple time steps - and trigger a prediction only when this accumulated signal crosses a predefined threshold. This design allows the model to delay decisions until it has gathered sufficient confidence, making it more robust to ambiguous or noisy inputs.

This approach has shown promising results in tasks with noisy or ambiguous inputs from visual information streams, outperforming LSTM baselines. However, [16] makes only one decision per input stream: once a decision is made, the model resets its internal accumulator, discarding all previously gathered evidence. This design assumes that each decision is isolated. However, in many real-world behavior prediction tasks, actions unfold continuously over time, and the temporal order of observations and decisions carries important semantic meaning. In such temporally extended tasks, where past events and prior decisions influence future behavior, treating decisions as isolated episodes is a limiting assumption. Most tasks encountered and performed by humans and primates involve continuous sensory input, where decisions are not made in isolation but are influenced by an evolving internal context. [20] suggests that the biological CBGT circuitry does not immediately reset after a decision. Instead, it enters a consolidation phase where decision channels remain active, potentially influencing subsequent choices. This biological insight challenges the independent decision making assumption and motivates a more continuous, history-aware approach to decision-making in neural models.

Decisions are made by CBGT-Net when the accumulated evidence crosses a certain preset threshold. New decisions to be made requires for the model to gather new evidence, bringing the accumulator back up to the decision threshold. The lack of a method to enable the accumulator to automatically default to below the threshold while preserving the context offered by previous accumulations and decision made by the model serves as a key motivation for our approach, Online-CBGT-Net, to incorporate a suitable mechanism to do the same.

Furthermore, in the CBGT circuit, action selection arises from a balance between two pathways: the Go pathway, which facilitates actions, and the No-Go pathway, which suppresses them [3]. CBGT-Net is already designed with this push–pull mechanism in mind - it trains a single “Differential” encoder that learns to estimate the net difference between Go and No-Go activations and uses this learned difference as evidence for each class. While this yields a useful decision variable, the Go and

No-Go signals themselves remain implicit; the model collapses support and opposition into one evidence vector. As a consequence, for any given decision we cannot inspect how much *support* a class received or how strongly competing alternatives were *opposed*. This lack of explicit Go/No-Go information limits interpretability, even though the underlying biological principle is present.

To better reflect this principle, we introduce dual activation pathways in our model: separate branches that compute supporting and opposing evidence for each behavioral class. This separation allows us to make the model's decision-making process more transparent—each decision is not just the result of high accumulated support, but also of relatively low opposition. By explicitly modeling this push-pull dynamic, we gain a clearer picture of how the model navigates ambiguity, resolves conflict, and eventually commits to a choice. Furthermore, this design offers a more interpretable window into the model's internal reasoning, which is crucial for applications that demand explainability, such as understanding complex patterns in animal or human behavior.

In this thesis, we extend the CBGT-Net model to support continuous, context-aware decision-making on streaming input data, calling it Online-CBGT-Net. We introduce a reset mechanism that decays accumulated evidence after each decision, allowing the model to maintain temporal continuity and make consecutive decisions informed by prior context. Additionally, we explicitly model the Go and No-Go pathways of the CBGT circuitry through separate encoder branches that compute supporting and opposing evidence for each behavior. Unlike prior work that modeled evidence implicitly, our approach offers a clearer, more interpretable view of how each input contributes to the decision process.

Chapter 2

Related Work

Understanding how biological systems make context-sensitive decisions has long inspired both neuroscientific and computational research. In the mammalian brain, the CBGT circuit has been identified as a core circuit supporting evidence accumulation and action selection under uncertainty. The CBGT architecture facilitates flexible behavior by integrating sensory input and prior experience through a balance of facilitation and suppression across parallel neural pathways. Computational models of this circuit have been developed to study response time, reinforcement learning, and the neural basis of action uncertainty. These models primarily aim to elucidate biological mechanisms, such as the role of dopamine in modulating Go and No-Go pathways, and how these dynamics manifest in reaction time and learning behavior.

Building on these insights, recent efforts have sought to translate biological principles into machine-learning architectures. One influential example is CBGT-Net [16], which we discuss in detail in Section 2.4.

In contrast to traditional recurrent neural networks, which process incoming sequences and emit outputs at every time step, CBGT-inspired architectures introduce a deliberative, selective prediction mechanism.

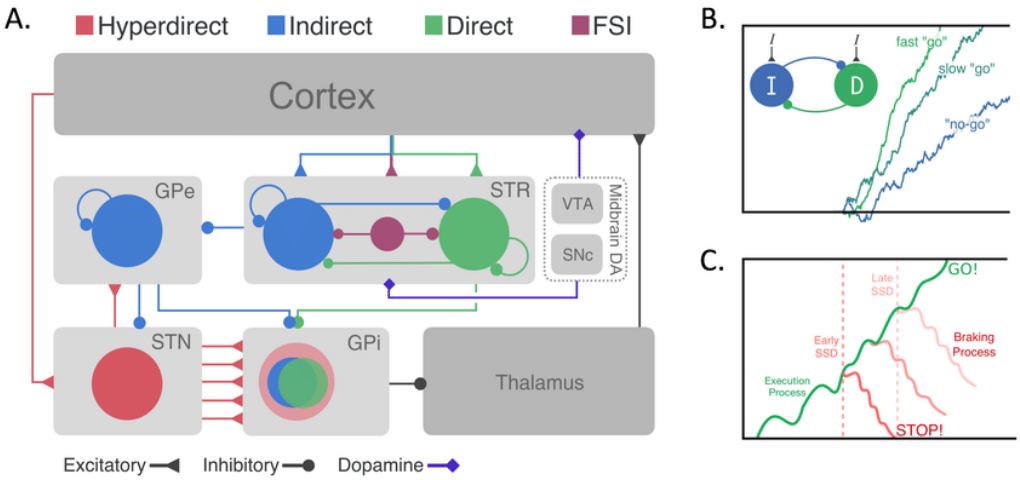

Figure 2.1: Cortico-Basal Ganglia Thalamus (CBGT) Circuits in Mammalian Brains [1]

2.1 Cortico-Basal Ganglia Thalamic Circuit, Supporting and Inhibiting Signals

The network architecture presented in this paper is inspired by CBGT circuits in mammalian brains, and the role they play in decision making and evidence accumulation [8, 11, 18], as shown in Figure 2.1. Corticostriatal connections provide pathways for projection from the functional areas of the cortex-including the sensorimotor, associative, and limbic areas-to the striatum. For each potential action (i.e., motor neuron activation), two pathways exist in the basal ganglia that facilitate or suppress the action in the thalamus: Direct ("Go") pathways inhibit the globus pallidus internus (GPi), which in turn causes disinhibition of the thalamus and facilitation of the action corresponding to the circuit; Indirect ("NoGo") pathways inhibit the globus pallidus externus (GPe), which in turn disinhibits the GPi and suppresses the circuit's action.

In the context of the described structure, the information used for decision-making is generated in the cortex, which in turn increases or decreases the total activation of the "Go" and "NoGo" pathways for each action. Action selection for a given action is based on the relative activation of the "Go" and "NoGo" pathways - actions with

a higher differential between the "Go" and "NoGo" pathways are more likely to be performed, see Figure 2.2. In essence, the basal ganglia facilitate actions with a probability proportional to the activation difference between the "Go" and "NoGo" pathways. An action is performed once the activation difference for the action exceeds some criteria. Tonic dopamine levels in the brain increase the excitability of the "Go" pathways and decrease the excitability of the "NoGo" pathways, which influences the overall criteria for action selection, as well as reaction time.

2.2 Influence of Previous Decisions and Evidences on Future Decisions

A key aspect of how previous information influences current decisions is highlighted by [10], demonstrating that "residual information" from a prior decision can exert a significant impact on subsequent choices. Through behavioral experiments and modifications to the drift-diffusion model (DDM[9, 12, 13]), it is found that the starting point of evidence accumulation for a new decision is not always neutral. Instead, it can be systematically biased by the outcome of the preceding decision and associated accumulated evidence. This suggests that the brain does not simply erase all traces of a decision once made; rather, a "memory" or "trace" of the previous choice subtly sets the initial conditions for the next round of evidence processing, influencing the speed and accuracy of subsequent decisions. This carry-over effect provides a compelling theoretical basis for developing a reset mechanism in CBGT-Net inspired neural network models which allows for accumulation from previous decisions and evidences to influence future decisions.

2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are specialized neural networks designed to handle sequential data by maintaining a 'memory' of previous inputs through a hidden state that is updated at each time step. This architecture allows RNNs to process sequences of data such as time series, text, or image sequences, making them well-suited for tasks where context and order are crucial. A key feature of RNNs is their weight

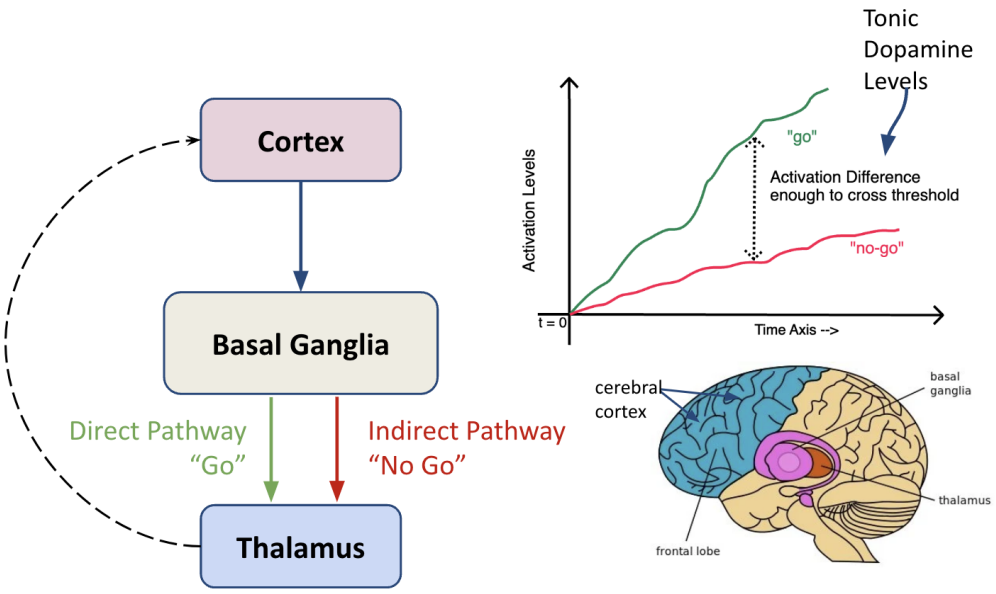

Figure 2.2: Simplified CBGT circuitry illustrates the interaction between brain sub-architectures: Cortex, Basal Ganglia, and Thalamus. The graph on the right shows that for the brain to decide to perform an action, the activation difference between the 'GO' and 'NoGo' pathways must surpass a threshold determined by dopamine levels in the Thalamus

sharing across all time steps, which helps generalize patterns and reduces the number of parameters needed.

There are different types of RNNs, each designed to address specific challenges. Vanilla RNNs [14], the simplest form, often face issues like vanishing and exploding gradients, making them difficult to train on long sequences. Long Short-Term Memory (LSTM) [6] networks and Gated Recurrent Units (GRUs) [2] are advanced variants that include mechanisms to better manage long-range dependencies. LSTMs use memory cells and gates to control information flow, while GRUs simplify this structure with two gates, making them more computationally efficient yet effective.

RNNs are widely applied in natural language processing (NLP) tasks such as language modeling, machine translation, and speech recognition, where understanding context over sequences is essential. They are also used for time series forecasting, like predicting stock prices or weather conditions, and in image and video processing tasks, such as captioning and event detection. Despite challenges like vanishing gradients, advancements like LSTMs, GRUs, and attention mechanisms have significantly improved RNNs' capability to handle complex, long-sequence data effectively.

2.4 CBGT-Net

The CBGT-Net architecture [15, 16] utilizes an evidence accumulator and a thresholding mechanism to gather information sequentially from a continuous stream of data and make a decision when sufficient evidence has been collected. At each time step, the model processes an input, which is then passed through an evidence encoder. The encoder, a neural network, is designed to model the differential activation between the Go and NoGo pathways within the CBGT circuitry. The output of this encoder is an evidence vector, where each element corresponds to a distinct decision category.

The evidence generated at each time step is cumulatively added to an accumulator, which maintains a running sum of all evidence gathered since the beginning of the input sequence. A decision threshold is applied to this accumulator at every time step. If the accumulated evidence for any decision category crosses this predetermined threshold, a decision is triggered. In such a scenario, the category with the highest accumulated value is selected as the model's classification. Conversely, if no category's evidence surpasses the threshold, the model continues to receive further input, allowing for

additional evidence accumulation. Training of the model is facilitated by propagating gradients through a cross-entropy loss function each time a decision is made. This architecture offers a robust method for accumulating evidence over a data stream, enabling the model to make decisions only when a predefined level of confidence, dictated by the threshold, has been reached.

However, the current model processes each decision independently without influence from previous inputs or decisions. A key limitation is its lack of a mechanism for carrying over information from one completed decision to subsequent ones. This restricts its applicability in scenarios requiring continuous sequential decisions over an evolving stream of information from a continuously evolving process. Furthermore, while the model encodes the differential activity between Go and NoGo pathways of the CBGT circuitry, it does not explicitly provide separate evidence signals favoring or opposing each individual decision class, instead focusing on a combined differential representation.

Chapter 3

Problem Statement

3.1 Problem Statement

Real-time online prediction models must be capable of making continuous, confident decisions as input streams evolve over time. This requires maintaining context from past inputs, suppressing premature decisions, and updating internal beliefs as new evidence arrives.

In this chapter, we describe this problem in greater detail by examining CBGT-Net. We first outline the CBGT-Net architecture, then analyze its limitations in the context of real-time online prediction. These limitations motivate our proposed extension, Online-CBGT-Net, which incorporates mechanisms to enable continuous decision-making and enhanced interpretability.

3.2 Background: CBGT-Net Architecture

CBGT-Net (Figure 3.1) is a neural architecture inspired by the CBGT circuit, as discussed in Section 2. It is designed to accumulate evidence over time and make a prediction once the accumulated evidence crosses a threshold. Below, we summarize its architectural components.

The CBGT-Net architecture consists of four primary components: an encoder, an evidence vector, an accumulator, and a decision threshold. We describe each in detail

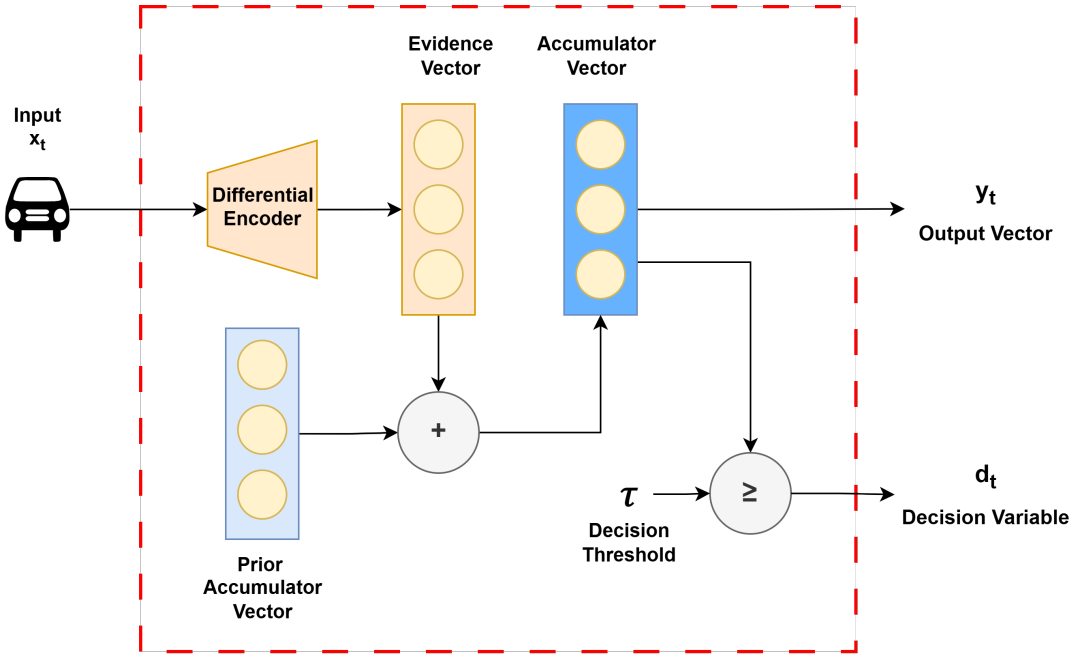

Figure 3.1: CBGT-Net Architecture

below.

3.2.1 Input

At each time step $\mathbf{t}$, the model receives an input feature vector $\mathbf{x_t}$. This input is utilized by the model to generate evidence, accumulate evidence, and perform downstream tasks.

3.2.2 Differential Evidence Encoder

At each time step $\mathbf{t}$, the model receives an input feature vector $\mathbf{x_t} \in \mathbb{R}^d$, where d denotes the dimensionality of the input. This vector is processed by the Differential Encoder, a trainable feedforward neural network parameterized by $\mathbf{E}_\theta$. The encoder transforms the input into an evidence vector $\mathbf{e_t} \in \mathbb{R}^C$, where C is the number of behavior classes or decision categories:

$$\mathbf{e_t} = \mathbf{E}_\theta(\mathbf{x_t}) \tag{3.1}$$

The Differential Encoder learns the difference in activation levels and encodes it as the evidence vector. These class-wise evidences are then passed to the accumulator module for temporal accumulation.

3.2.3 Accumulator

The accumulator serves as a memory mechanism that integrates evidence across time. At each time step, the incoming evidence is added to the previous accumulator state:

$$\mathbf{a}_t = \mathbf{a}_{t-1} + \mathbf{e}_t \tag{3.2}$$

where the initial accumulator state $\mathbf{a}_0 = \mathbf{0}$, corresponds to the start of the input stream at time $\mathbf{t} = 0$.

3.2.4 Output

The output of the model at time step $\mathbf{t}$ is denoted by $\mathbf{y_t}$, which represents the predicted behavior class. This is obtained by taking the index of the class with the highest accumulated evidence:

$$\mathbf{y_t} = \arg\max_i \left(\mathbf{a_t}[i]\right)$$

Here, $\mathbf{a_t}$ is the accumulator state at time $\mathbf{t}$, and i indexes the possible behavior classes.

3.2.5 Decision Threshold and Decision Variable

A decision is considered to be made at time $\mathbf{t}$ when the accumulated evidence for any class exceeds a predefined decision threshold τ. This is captured by the binary decision variable $\mathbf{d_t} \in \{1, 0\}$:

$$\mathbf{d_t} = \begin{cases} 1 & \text{if } \max_i \left(\mathbf{a_t}[i]\right) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

If $\mathbf{d_t} = 1$, the model commits to the prediction $\mathbf{y_t}$ at that timestep; and if $\mathbf{d_t} = 0$, it continues accumulating evidence over time.

3.3 Limitations of CBGT-Net

3.3.1 No Mechanism for Continuous Decisions

The accumulator saturates after a decision and is incapable of integrating further input to cross the decision threshold again to make a new decision. This prevents the model from making additional decisions on the ongoing input stream without starting a new episode. As a result, CBGT-Net is not well-suited for environments where multiple decisions need to be made sequentially and continuously.

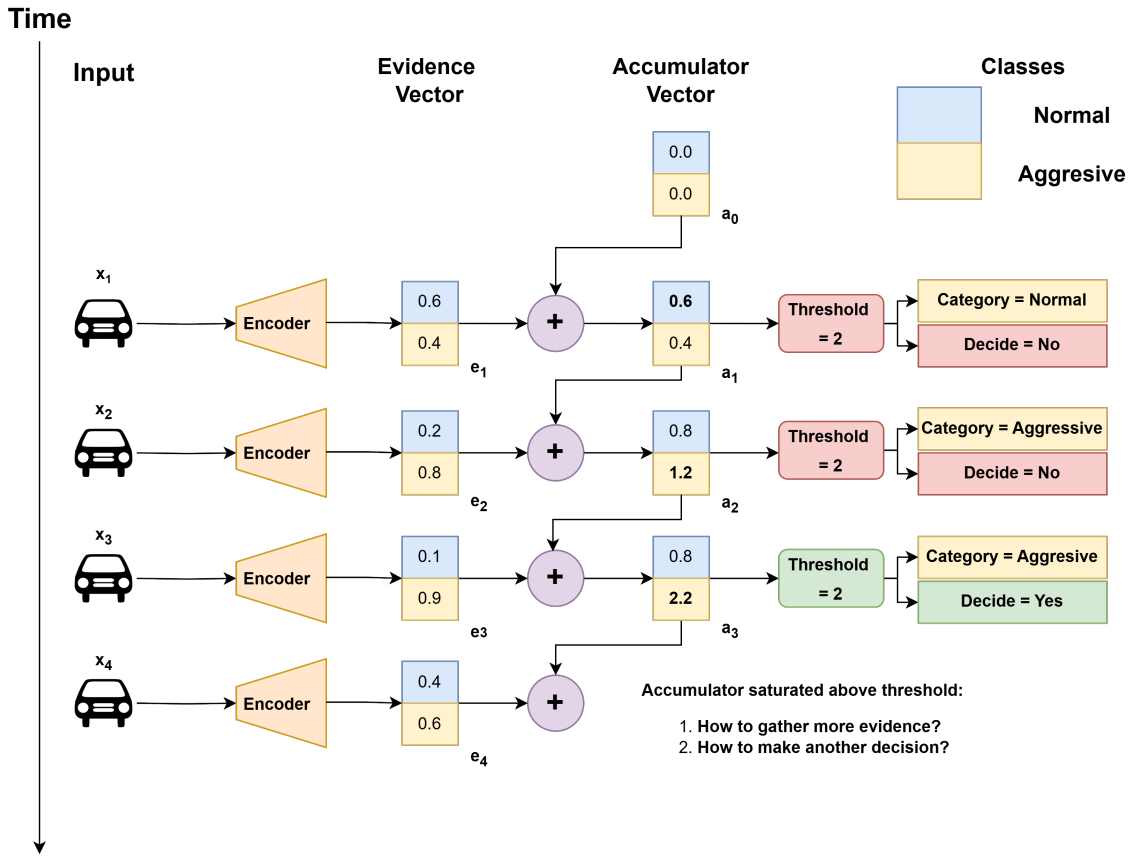

Figure 3.2: CBGT-Net's limitation to make continuous decisions

An example is demonstrated in the Figure 3.2.Given the input trajectory of a car, the model is tasked with determining whether the car is driving normally or aggressively. As time progresses, CBGT-Net accumulates evidence with each input

until the total evidence surpasses a decision threshold - at which point it classifies the behavior as aggressive. However, at the very next time step, when new trajectory data arrives, the model faces a critical limitation: the accumulator remains saturated, with no mechanism to bring it back below the threshold. As a result, CBGT-Net is unable to integrate new evidence and make subsequent decisions, revealing its inability to operate effectively in continuous streaming environments.

3.3.2 No Separation of Support and Opposition

Consider a time instant where an input trajectory of a car is received, and the model must determine whether the car is driving normally or aggressively. This input is passed through the CBGT-Net's differential encoder, which outputs an evidence vector - each element representing the model's belief in one of the two decision classes as seen in the Figure 3.3

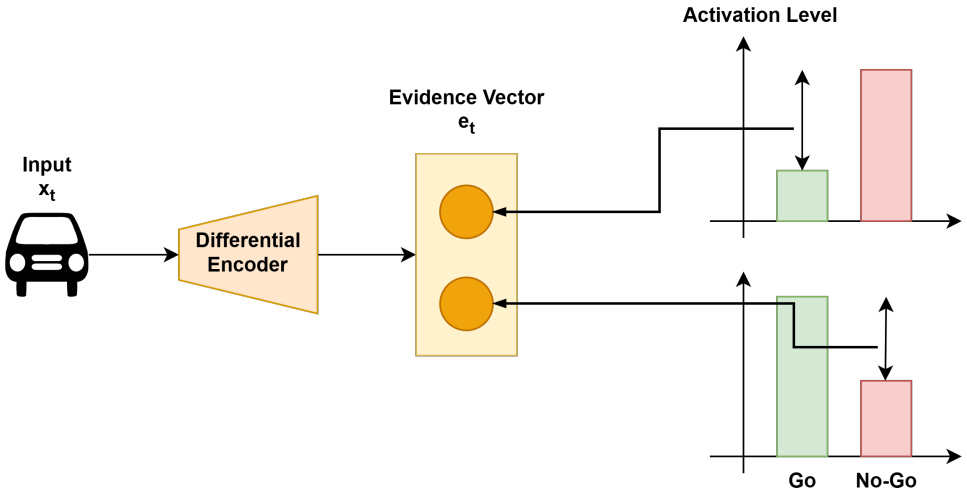

Figure 3.3: CBGT-Net's Differential Encoder learns to estimate the difference between the Go and No-Go activation levels.

However, it is important to note that the differential encoder does not explicitly compute the individual Go (support) and No-Go (opposition) activations for each class. Instead, it directly estimates the difference between the two and encodes this estimated difference as evidence. Consequently, the model provides no insight into whether a decision was driven by strong support for a class, strong opposition to the

alternative, or both.

In the next chapter, we address these limitations through the proposed Online-CBGT-Net architecture, which introduces a reset mechanism to allow the accumulator fall back below the threshold to make new decisions and a dual encoder design to model the Go and No-Go evidences for each input explicitly.

Chapter 4

Approach

4.1 Introduction

We introduce Online-CBGT-Net, a neuromimetic architecture designed to support real-time prediction in continuous environments. Our architecture addresses the two limitations mentioned in Section 3.3 by introducing the following changes to the CBGT-Net architecture:

1. A reset mechanism to bring the accumulator below the threshold after every decision, allowing it to continue gathering more evidence and make new decisions
2. A dual-pathway encoder consisting of separate Go and No-Go branches to explicitly model the corresponding pathways in the CBGT circuit.

4.2 Online-CBGT-Net: Architecture

The proposed Online-CBGT-Net architecture as seen in Figure 4.1 with changes over the CBGT-Net are explained below.

4.2.1 Reset Mechanism for Online Decision Making

In the original CBGT-Net architecture, evidence is accumulated over time until the model reaches a confident decision. However, after this decision is made, the

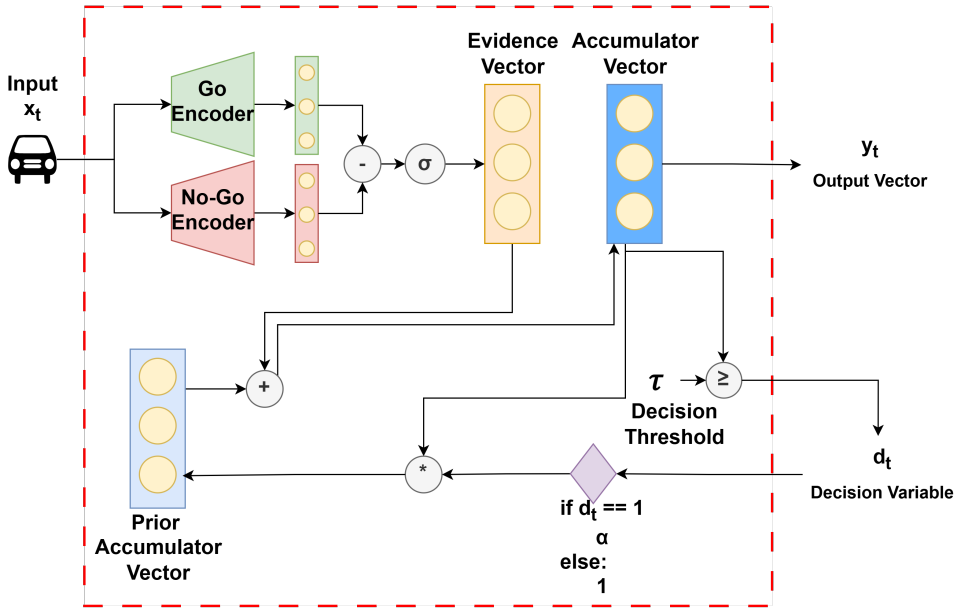

Figure 4.1: Online-CBGT-Net Architecture

accumulator remains saturated above the threshold, preventing it from gathering new evidence to cross the threshold again to make a new decision.

To support continuous prediction over streaming inputs, we introduce a reset mechanism that allows the accumulator to be reused across multiple decision cycles. After making a decision, the accumulator is reduced using a Reset Factor $\alpha \in [0, 1)$ that reduces its values below the decision threshold. This enables the model to continue to accumulate new evidence while preserving a proportion of the evidence from the prior decision.

Mechanism

Let $\mathbf{a}_t$ denote the accumulator at time $\mathbf{t}$, and τ be the decision threshold. After making a decision at the timestep $\mathbf{t_d}$, the accumulator is updated as follows:

$$\mathbf{a_{t_d}} = \alpha \cdot \mathbf{a_{t_d}} \tag{4.1}$$

Here, α, controls the degree to which the accumulated evidence is retained.

This mechanism enables a new cycle of evidence accumulation and decision making to begin. An example of the reset mechanism is seen in Figure 4.2

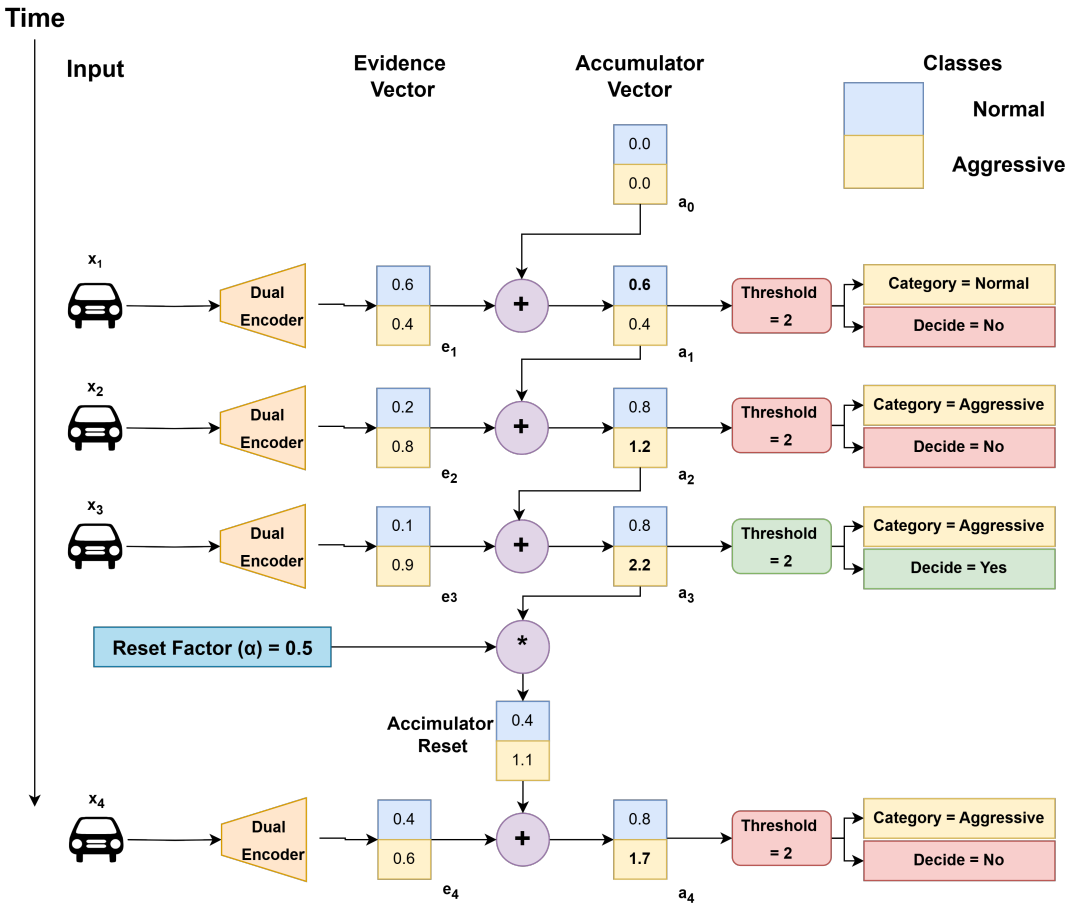

Figure 4.2: Working example of the Reset Mechanism in Online-CBGT-Net

Interpretation

The accumulator serves as the **model's memory**, integrating input evidence over time. The reset factor α provides control over *memory retention*:

- A low α forgets most prior evidence, making the model to gather more evidence to trigger another decision.
- A high α retains more context, requiring the model to gather lesser evidence to make a new decision

This mechanism is essential for **online prediction**, where multiple decisions are to be made sequentially and past context can inform future decisions.

4.2.2 Dual Encoder Architecture

To replicate the Go And No-Go activation mechanism in the CBGT Circuit, we introduce a **Dual Encoder Architecture** in Online-CBGT-Net as seen in the Figure 4.3. This design explicitly models the Go and No-Go activations via two parallel encoders:

- The **Go Encoder** E_{Go} computes evidence in favor of each class.
- The **No-Go Encoder** E_{NoGo} computes evidence opposing each class.

Given the input $\mathbf{x}_t$ at time step $\mathbf{t}$, the Go and No-Go encoders produce class-wise activations:

$$\mathbf{e_{g,t}} = E_{\text{Go}}(\mathbf{x}_t) \tag{4.2}$$

$$\mathbf{e_{ng,t}} = E_{\text{NoGo}}(\mathbf{x}_t) \tag{4.3}$$

The net evidence is computed as the element-wise difference between the Go and No-Go activations:

$$\mathbf{e}_t = \mathbf{e_{g,t}} - \mathbf{e_{ng,t}} \tag{4.4}$$

The final evidence vector is then obtained by applying a softmax over this difference:

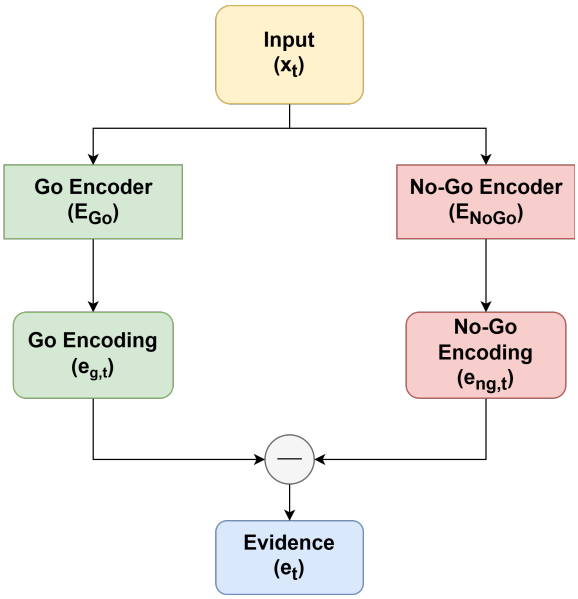

Figure 4.3: Dual Encoders of Online-CBGT-Net

$$\mathbf{e}_t = \text{softmax}(\mathbf{e}_t) \tag{4.5}$$

This net evidence vector $\mathbf{e}_t$ encodes the model's evidence for each class at time $\mathbf{t}$, and is passed into the accumulator for temporal integration.

This dual-path structure allows the model to explicitly compute both supporting and opposing evidence, offering greater interpretability and aligning closely with the biological principles of decision making.

4.2.3 Training Procedure

Unlike conventional classification models that predict a label at every timestep, our model maintains an internal accumulator state that evolves over time. This necessitates a training strategy that aligns with the model's deliberative and reset-based decision process. We train the model end-to-end by processing each input sequentially and evaluating the model's decision only when the accumulator crosses the decision threshold. Below, we describe the training methodology used.

At each timestep t, the input feature vector $\mathbf{x}_t$ is passed through the encoder to produce an evidence vector $\mathbf{e}_t$. This evidence is added to the previous accumulator

state $\mathbf{a}_{t-1}$ to obtain the updated state $\mathbf{a}_t$. If any class dimension of $\mathbf{a}_t$ crosses the decision threshold τ, a prediction $\hat{\mathbf{y}}_t$ is made, the loss is computed against the ground truth $\mathbf{y}_t$, and the accumulator is reset using the reset factor α. Otherwise, no prediction is made at that timestep and the model continues to accumulate evidence.

Algorithm 1 Online-CBGT-Net Training Loop

Require: Sequence of inputs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, ground truth labels $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T\}$

Require: Encoder network E_θ, accumulator state $\mathbf{a}_0 = \mathbf{0}$, threshold τ, reset factor α, class weights $\mathbf{w}$

Initialize loss $\mathcal{L} = 0$
for $t = 1$ to T **do**
    $\mathbf{e}_t \leftarrow E_\theta(\mathbf{x}_t)$ ▷ Encode input as evidence
    $\mathbf{a}_t \leftarrow \mathbf{a}_{t-1} + \mathbf{e}_t$ ▷ Update accumulator
    if $\max(\mathbf{a}_t) \geq \tau$ **then**
        $\hat{\mathbf{y}}_t \leftarrow \arg\max(\mathbf{a}_t)$ ▷ Make decision
        $\mathcal{L} \leftarrow \mathcal{L} + \mathbf{w_T} * \text{Loss}(\hat{\mathbf{y}}_t, \mathbf{y}_t)$
        $\mathbf{a}_t \leftarrow \alpha \cdot \mathbf{a}_t$ ▷ Reset accumulator
    end if
end for
Backpropagate and update θ

The loss is computed only at decision points, reinforcing the model's learning around moments when it chooses to act. We use a weighted cross-entropy loss, to handle imbalanced datasets, and parameters are updated using the Adam optimizer. The loss is propagated and the encoder weights are updated at the end of the sequence of inputs.

Chapter 5

Experiments

5.1 Experimental Setup

5.1.1 Dataset

The Caltech Mouse Social Interactions 2021 (CalMS21) Dataset [19] is a comprehensive behavioral neuroscience dataset designed for multi-agent behavior classification tasks as seen in Figure 5.2. The dataset contains pose trajectory data from pairs of socially interacting laboratory mice recorded in a standard resident-intruder assay setup. Each sequence captures the tracked postures of two mice (a resident C57Bl/6J mouse and an intruder BALB/c mouse) interacting freely in a controlled environment, with videos recorded at 30Hz from a top-view perspective.

The pose estimation data consists of seven anatomically defined keypoints tracked for each mouse: nose, left ear, right ear, neck, left hip, right hip, and tail base as seen in Figure 5.1. This results in 28-dimensional feature vectors per frame (7 keypoints $\times$ 2 mice $\times$ 2 coordinates), providing detailed spatial information about animal positioning and movement patterns as seen in Figure 5.3. The dataset includes over 6 million frames of unlabeled trajectory data and more than 1 million frames with corresponding frame-level behavioral annotations.

Behavioral annotations are provided by trained domain experts who manually labeled three primary social behaviors: close investigation (active sniffing and exploration), attack (high-intensity aggressive interactions) and mount (reproductive

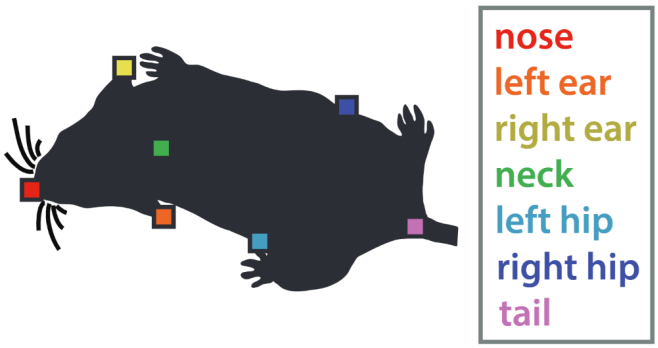

Figure 5.1: Keypoints Tracked Per Mice Per Frame. Each mouse has 14 features tracked (7 keypoints * 2 coordinates). Total of 28 features per frame which will serve as the input to the model at every time step [19]

behavior involving grasping and positioning) as shown in Figures 5.4, 5.5, 5.6. Frames not exhibiting these specific behaviors are labeled as "other." The dataset is structured into three distinct tasks:

- Task 1: Supervised classification of labelled behavior
- Task 2: Annotation style transfer across different human annotators
- Task 3: Few-shot learning for novel behaviors with limited training examples

This dataset serves as a benchmark for evaluating automated behavior classification methods, particularly focusing on challenges such as leveraging unlabeled data, handling annotation variability across experts, and learning new behaviors from minimal training samples. The controlled experimental conditions combined with rich pose information make CalMS21 well-suited for studying continuous behavior prediction in naturalistic multi-agent social interactions.

5.1.2 Task

We use the frame-wise annotated dataset from Task 1 of the Calms21 Dataset to train and evaluate our model. The dataset includes 70 training videos and 19 test videos, each treated as a continuous input stream of mice pose associated with each frame. At each timestep t, the model receives the pose information from the corresponding frame as input $\mathbf{x_t}$. The goal is to predict the predominant behavior occurring over

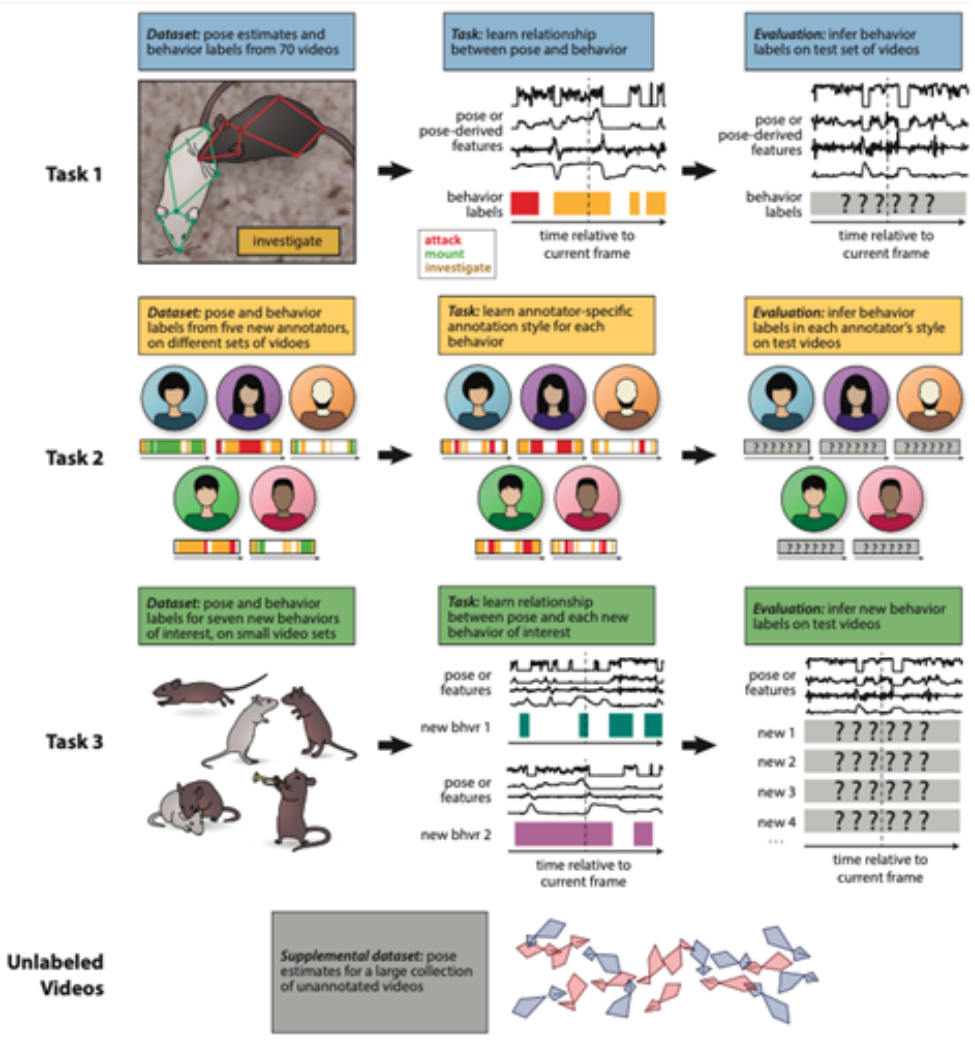

Figure 5.2: Caltech Mouse Social Behavior Dataset [19]

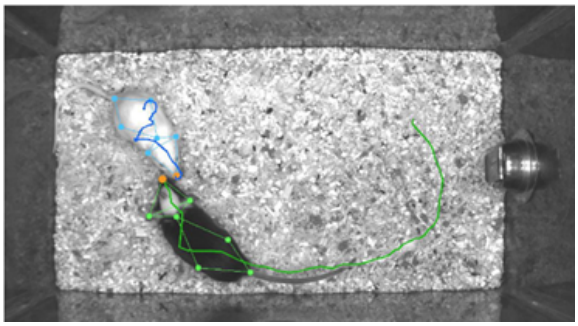

Figure 5.3: Mice Pose Tracked [19]

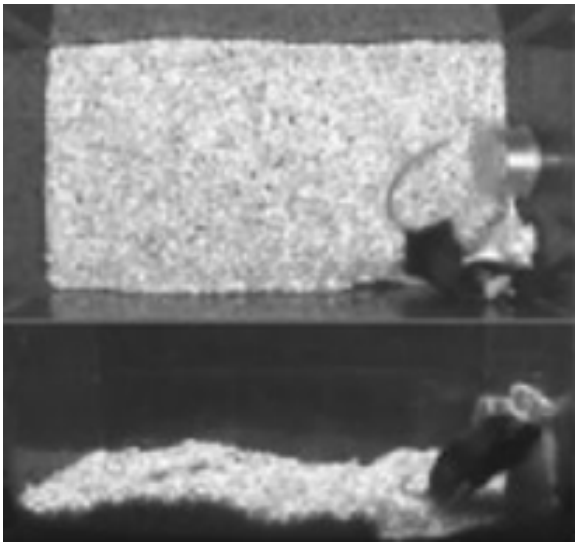

Figure 5.4: Mice displaying Attack Behavior [19]

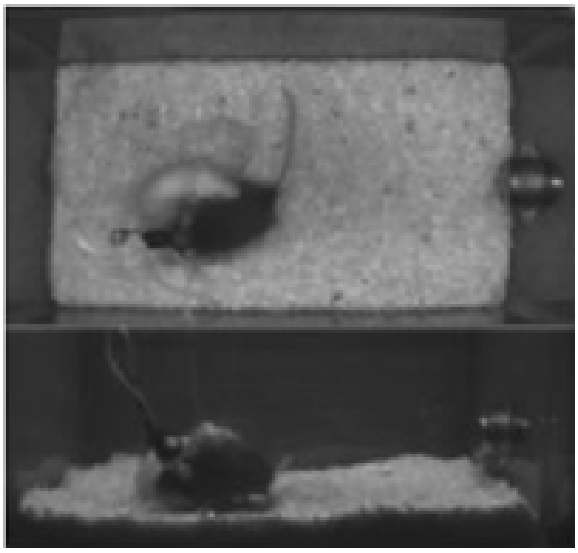

Figure 5.5: Mice displaying Investigate Behavior [19]

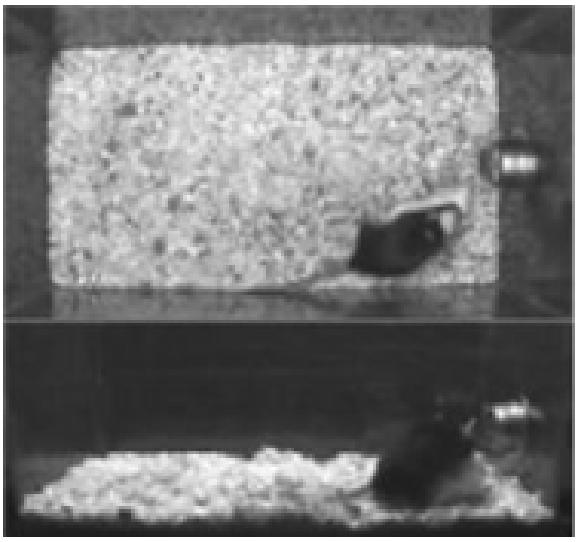

Figure 5.6: Mice displaying Other Behavior [19]

the next 20 frames, framing the problem as a four-class classification task.

The pose which acts as the input feature $\mathbf{x_t}$ is first encoded into an evidence vector $\mathbf{e_t}$, which is then integrated with the accumulated evidence from the previous timestep $\mathbf{a_{t-1}}$ to update the current accumulator state $\mathbf{a_t}$. This updated accumulator is compared against a decision threshold τ. If the threshold is exceeded, a behavior $\mathbf{y_t}$ is predicted and the accumulator is reset according to the reset mechanism.

This is illustrated in Figure 5.7.

Once the model finishes processing all frames in a video, it resets its accumulator states to zero before moving on to the next video. This ensures that each video is evaluated independently, without carryover of internal state across sequences.

5.1.3 Models

For each experiment, we compare our approach with multiple baselines. As with the CBGT-Net, all baseline models are trained to minimize the weighted cross-entropy loss given in Equation 5.2. The Online-CBGT Net model is evaluated using both the differential encoder and the dual path encoder to compare their performances.

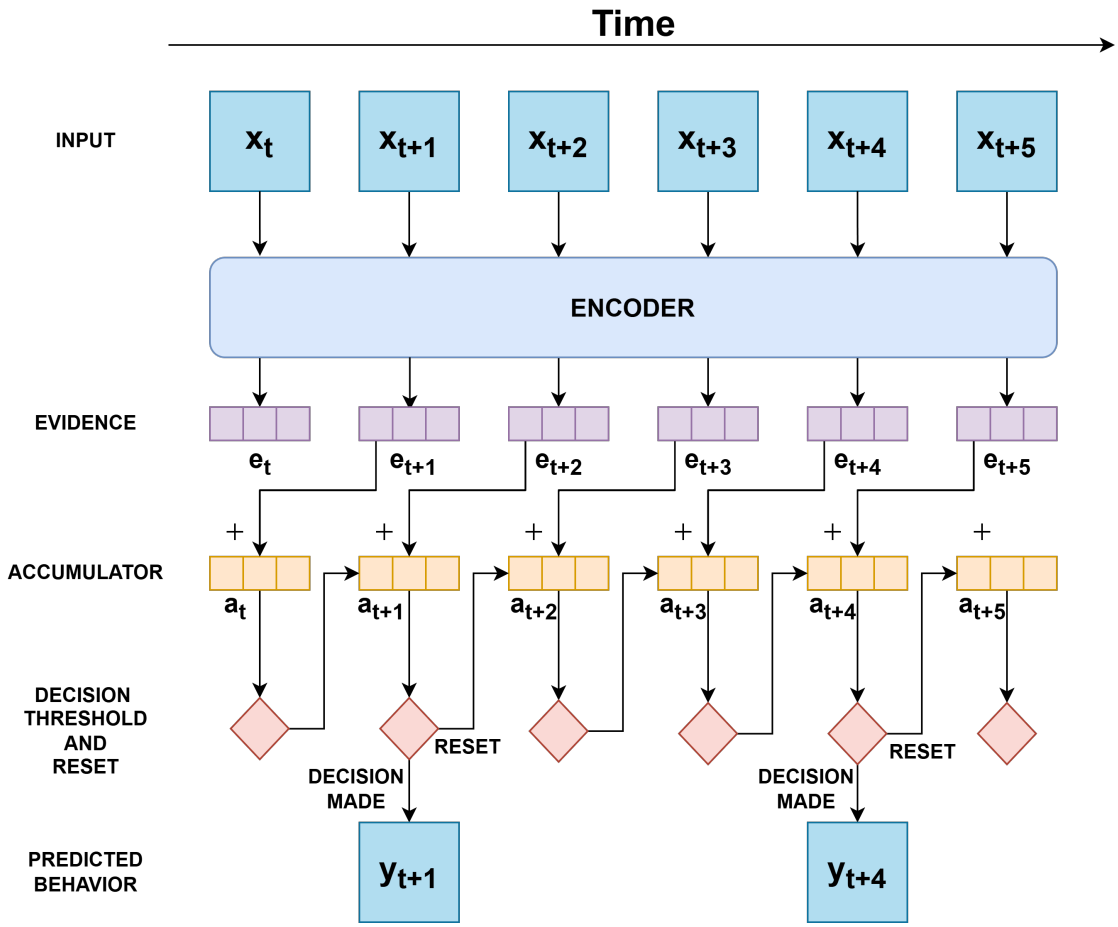

Figure 5.7: Behavior Prediction Task Overview

Online-CBGT-Net Dual Encoder

The Online-CBGT-Net Dual Encoder architecture integrates a dual-pathway encoder inspired by the push-pull dynamics of CBGT circuit, explicitly modeling both supportive and opposing evidence for each decision class for each input received. This dual-pathway mechanism consists of two parallel neural networks: the *Go Encoder* and the *No-Go Encoder*, which respectively capture activations that support and oppose a given behavioral class respectively.

At each time step t, the model receives an input vector $\mathbf{x}_t \in \mathbb{R}^{28}$, representing the pose features of the mice extracted from the frame at that instant. This input is simultaneously fed into both the Go Encoder and the No-Go Encoder, each of which shares an identical feedforward architecture composed of:

- **Linear** $(28 \to 256) \to$ **ReLU** $\to$ **Dropout**;
- **Linear** $(256 \to 128) \to$ **ReLU** $\to$ **Dropout**;
- **Linear** $(128 \to 4) \to$ **ReLU** $\to$ **Dropout**;
- **Linear** $(4 \to 4) \to$ **Sigmoid**

The outputs of these two encoder branches are two vectors:

$$\mathbf{e_{g,t}} = E_{\text{Go}}(\mathbf{x}_t), \quad \mathbf{e_{ng,t}} = E_{\text{NoGo}}(\mathbf{x}_t), \quad \mathbf{e_{g,t}}, \mathbf{e_{ng,t}} \in \mathbb{R}^4$$

These two vectors encode the raw support $\mathbf{e_{g,t}}$ and opposition $\mathbf{e_{ng,t}}$ for each of the four behavior classes at time $\mathbf{t}$.Their difference is computed element-wise:

$$\mathbf{e}_t = \text{Softmax}(\mathbf{e}_{g,t} - \mathbf{e}_{ng,t}) \tag{5.1}$$

This results in a evidence vector $\mathbf{e_t}$ of size $(4 * 1)$, where each component reflects the net evidence for a specific behavior class. This evidence vector is then passed to the accumulator module as, where it is temporally integrated over time.

The Dual Encoder has 81,712 trainable parameters. The overall Online-CBGT-Net Dual Encoder Architecture is shown in the Figure 5.8.

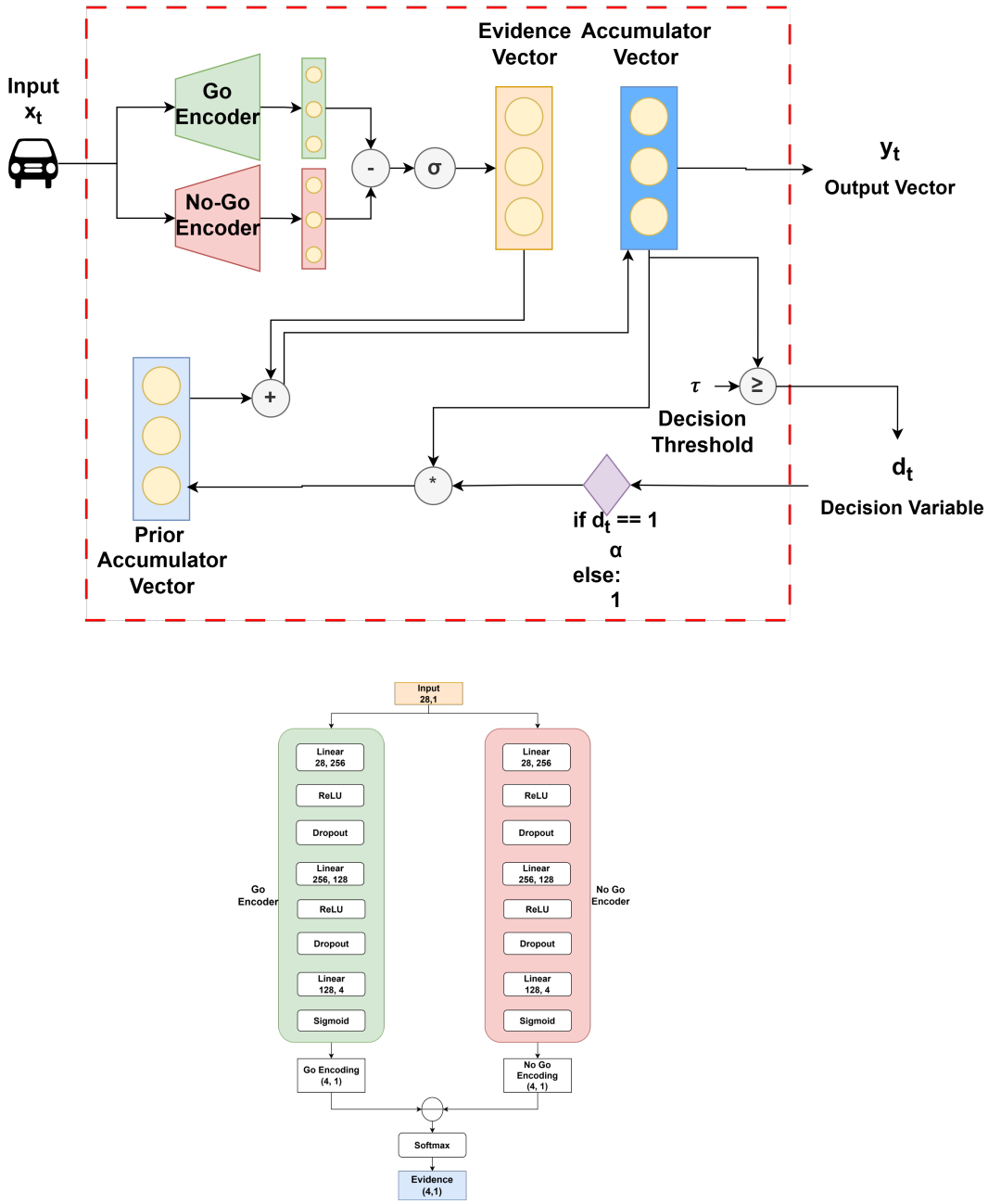

Figure 5.8: Online-CBGT-Net Model with the Dual Encoder Architecture

Online-CBGT-Net Differential Encoder

In the Differential Encoder variant of Online-CBGT-Net, evidence for each behavior class is produced by a **single** feed-forward network $\mathbf{E}_\theta$. Unlike the dual-pathway design, this encoder learns to estimate the difference in the activations between the Go/No-Go channels, implicitly combining support and inhibition in the same representation.

At every time step t the model receives the 28-dimensional pose vector $\mathbf{x_t} \in R^{28}$ of the mice as $\mathbf{x_t}$

$\mathbf{E}_\theta$ processes $\mathbf{x_t}$ through the following layers:

1. **Linear** $(28 \rightarrow 516) \rightarrow$ **ReLU** $\rightarrow$ **Dropout**;
2. **Linear** $(516 \rightarrow 128) \rightarrow$ **ReLU** $\rightarrow$ **Dropout**;
3. **Linear** $(128 \rightarrow 4) \rightarrow$ **ReLU** $\rightarrow$ **Dropout**;
4. **Linear** $(4 \rightarrow 4) \rightarrow$ **Sigmoid**
5. **Softmax** $(4 \rightarrow 4)$

The differential encoder contains 81,656 trainable parameters, almost equal to the dual-pathway encoder giving them both equal number of parameters and representation capabilities.

The overall Online-CBGT-Net Differential Encoder Architecture is shown in Figure 5.9

Baseline: LSTM Model

We utilize the LSTM Baseline to compare against the Online-CBGT-Net model. The baseline processes the input with same Dual Encoder as used in the Online-CBGT-Net Dual Encoder Model. This encoding is passed to a Long Short Term Memory (LSTM) layer [5]. The LSTM layer, replacing the accumulator in the Online-CBGT-Net architecture, has 4 memory cells, followed by a Softmax layer of 4 neurons for the four behavior classes to produce the predicted behavior.

The model outputs the predicted category at every time step and the same metrics used to evaluate the Online-CBGT-Net model are used to compare the performance of the LSTM baseline.

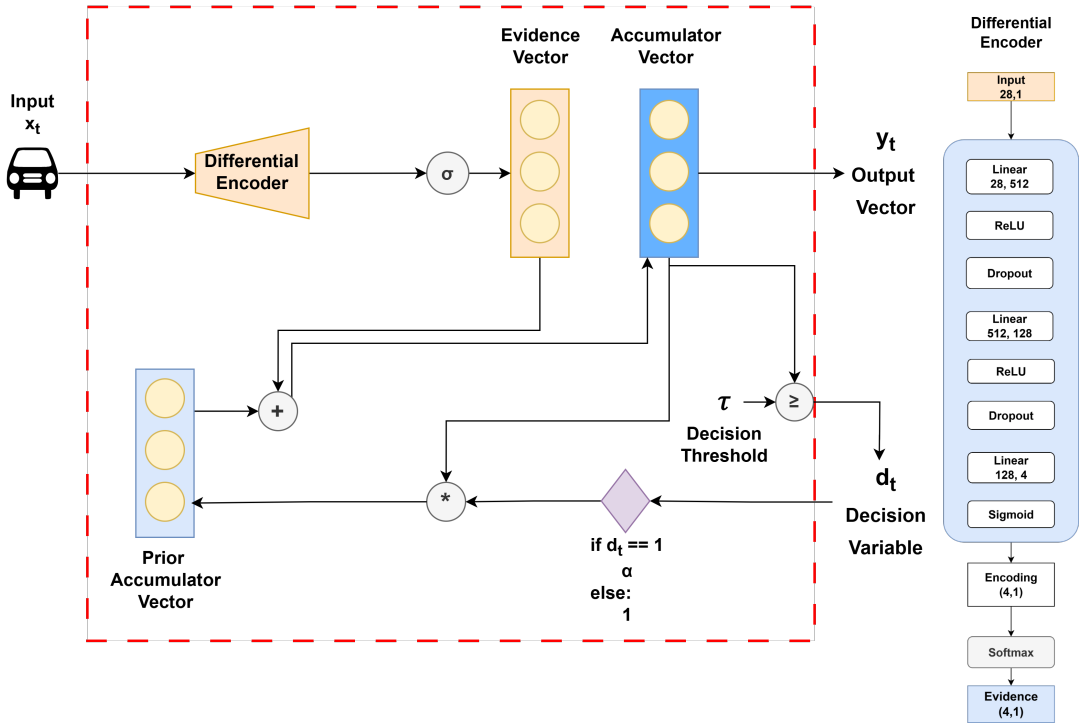

Figure 5.9: Online-CBGT-Net Model with the Differential Encoder Architecture

5.1.4 Training Details

Our model is trained using the annotated frame-wise dataset from Task 1, which includes 70 videos designated for training. Each video is treated as a continuous sequence of frames streamed into the model one frame at a time. At every time step, the model observes the current frame, updates its internal accumulator, and attempts to predict the most frequently occurring behavior over the next 20 frames. This setup frames the problem as a four-class classification task, where the model learns to align accumulated evidence with future behavioral patterns.

The Online-CBGT-Net Dual Encoder model is trained and evaluated across a range of decision thresholds, $\tau \in 5, 10, 15, 20$, and reset factors, $\alpha \in 0, 0.25, 0.5, 0.75$. These experiments allow us to analyze how varying the deliberation threshold and the degree of evidence retention affect model performance.

Based on the findings from the Dual Encoder experiments-where $\tau = 5$ consistently produced the strongest F1 scores across behavior classes-we evaluate the Online-CBGT-Net Differential Encoder model at threshold $\tau = 5$, across the same set of

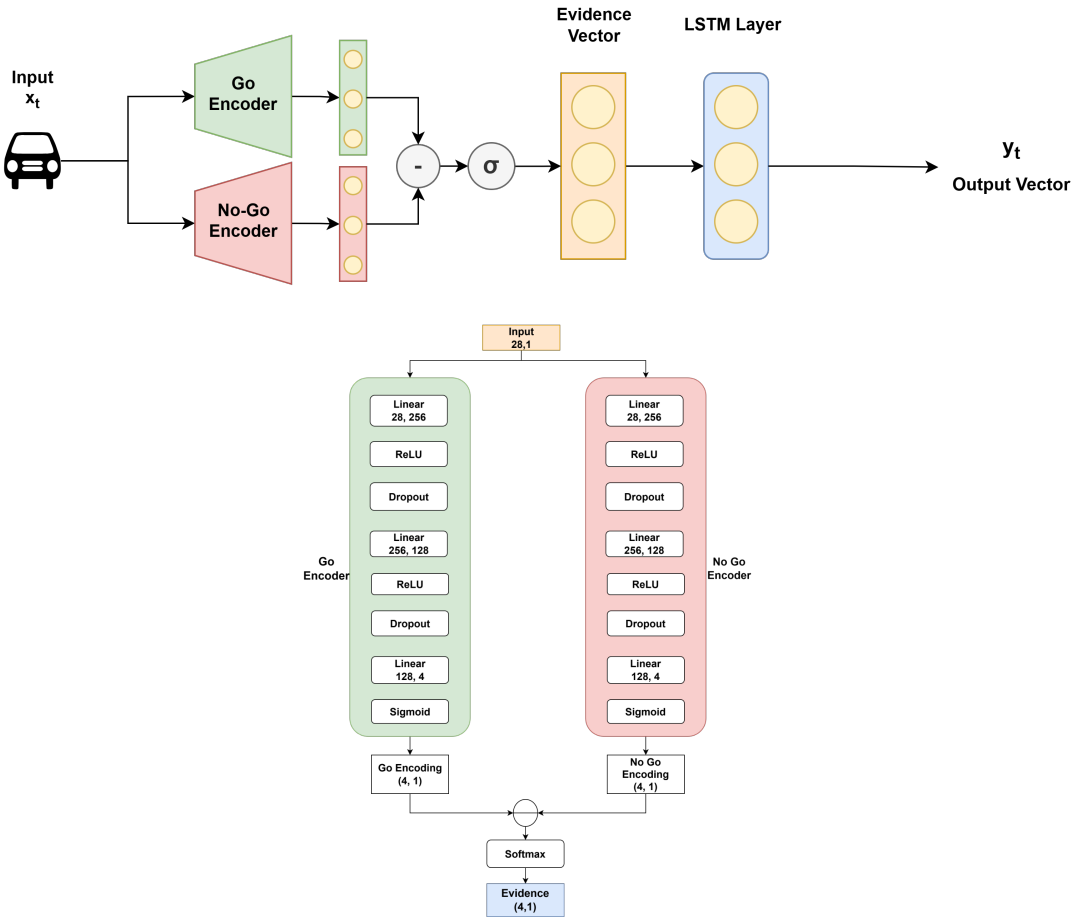

Figure 5.10: LSTM Baseline with the Dual Encoder

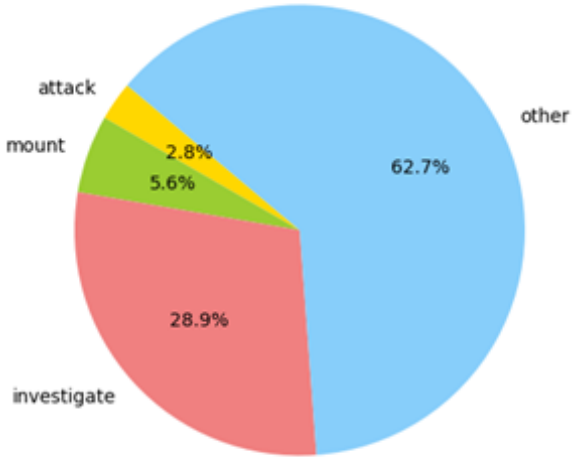

Figure 5.11: Calms21 Dataset Distribution

reset factors $\alpha \in 0, 0.25, 0.5, 0.75$, to measure the performance of the Differential Encoder variant against the best performing threshold configuration of the Dual Encoder Variant.

The LSTM baseline model is trained to predict the behavioral class at every timestep.

Training is performed for 200 epochs using the Adam optimizer [7] with a learning rate of 0.001 for all the models.

For training purposes, we utilize the weighted cross-entropy between the output vector of the CBGT-Net and the target category at the decision time, t_d, as the objective function to minimize,

$$\mathcal{L}_{CE} = -w_T \log(\mathbf{y}^{(T)} - t_d) \tag{5.2}$$

where T is the index of the target behavior to be classified and w_T is the class-weight associated with the behavior T.

These weights are used to address the class imbalance observed in the training set (as visualized in Figure 5.11). The weights for each class were determined by

normalizing against the frequency of the minority class. In this dataset, the 'attack' class represents the minority, with a frequency of 0.028 (2.8%). This frequency serves as the baseline ($f_{\min}$) for calculating the weights.

The weight w_k for a given class k with frequency f_k is calculated as:

$$w_k = \frac{f_{\min}}{f_k} \tag{5.3}$$

Based on the distribution of classes in the dataset, the specific weights for each behavior class are shown in Table 5.1.

These weights ensure that the model pays relatively more attention to misclassifications of underrepresented classes, particularly 'attack', during training, forcing the model to give equal importance to learning the minority classes and their representations.

The loss is calculated every time a model makes a decision that serves as a training signal for the model. The backpropagation algorithm updates the model weights at the end of each video, helping the model learn from the training signals.

For each video, a model processes each video in its entirety. Between training videos, all internal accumulator states are reset to zero to ensure independence across sequences.

Table 5.1: Class Weights for Weighted Cross-Entropy Loss

Class	**Class Index**	**Class Weight**
Attack	0	1.00
Investigate	1	0.097
Mount	2	0.50
Other	3	0.045

5.1.5 Evaluation Measures

We evaluated model performance using a combination of standard and task-specific metrics to assess both accuracy and decision quality in continuous behavior prediction.

F1 Score

The primary quantitative metric used is the F1 score, computed per class. Provides a balanced measure of the precision and recall of a model. It is particularly useful for evaluating models on imbalanced datasets, as it considers both false positives (FP) and false negatives (FN) with the true positives (TP).

It can be expressed directly in terms of confusion matrix elements:

$$\mathrm{F1} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{5.4}$$

The F1 score ranges from 0 to 1, where 1 represents perfect precision and recall, and 0 represents the worst possible performance.

Precision

While F1-score provides a balanced metric accounting for both false positives and false negatives, precision specifically evaluates the reliability of positive predictions - i.e., when the model predicts a behavior, how often it is correct. This is particularly relevant for Online-CBGT-Net, whose core principle is to deliberate to make confident and accurate decisions.

$$precision = \frac{TP}{(TP + FP)} * 100 \tag{5.5}$$

where TP represents true positives and FP denotes false positives. The higher the precision score, the better the performance of the model.

Latency

Latency measures the delay between the onset of a ground truth behavior and the model's first correct prediction of that behavior as seen in the Figure 5.12. For each contiguous window of a specific behavior in the ground truth, latency is defined as the number of frames it takes for the model to correctly identify the behavior after the window begins.

Let t_s and t_e be the start and end times of a ground truth behavior window, where the mice exhibit the same behavior for all time $t \in [t_s, t_e]$. If the model predicts the

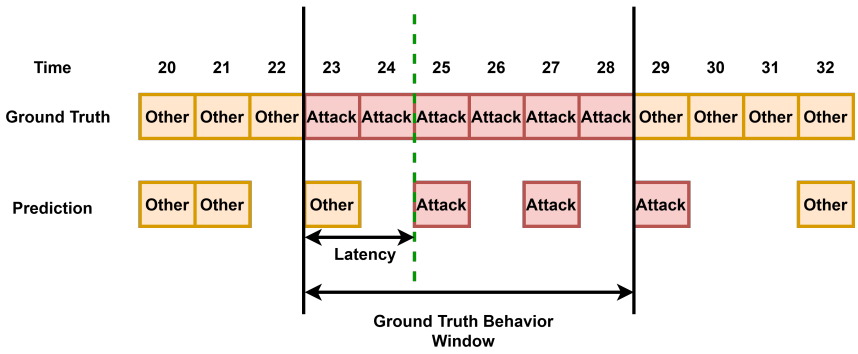

Figure 5.12: Latency of a prediction

correct behavior at $t_p \in [t_s, t_e]$, the latency is calculated as

$$latency = \frac{(t_d - t_s)}{(t_e - t_s)} * 100 \tag{5.6}$$

A model's ability to promptly recognize the onset of behavior windows is directly indicated by its lower latency. When employed as a comparative metric, lower latency is indicative of superior performance.

5.2 Results

5.2.1 F1 Scores

We evaluate Online-CBGT-Net Dual Encoder model across four reset factors $\alpha \in 0.0, 0.25, 0.50, 0.75$ and four decision thresholds $\tau \in 5, 10, 15, 20$, as summarized in the Tables 5.2, 5.3, 5.4, 5.5.

While a general downward trend in F1 score is observed as the decision threshold increases, particularly for the Mount, Investigate (except at $\tau = 10, \alpha = 0.5$ where it rises) and Other behaviors, this pattern is not generalizable to the Attack behavior. For example, at reset factors $\alpha = 0.0$ and $\alpha = 0.5$, the F1 score for Attack increases with higher thresholds.

We also observe that for a given threshold, model performance generally rises as the reset factor increases. This can be due to the fact that a higher reset factor preserves more accumulated evidence, producing more frequent decisions and more backpropagation signals - hence the overall performance gain. An important exception

occurs at the highest threshold, $\tau = 20$, where Attack and Investigate achieve their best F1 scores at lower reset factors. This reversal highlights that the optimal balance between threshold and reset factor is not universal across all behaviors: each behavior exhibits different temporal patterns and frequencies in the data. Consequently, reliable performance requires tuning both hyperparameters (threshold and reset factor jointly), rather than assuming a single "best" reset factor or threshold for all classes.

Across the thresholds $\tau = 10, 15$, and 20, the reset factor $\alpha = 0.25$ consistently under-performs for every behavior class. Although its results at $\tau = 5$ are comparable to other reset factors, its sharp decline at higher thresholds suggests that $\alpha = 0.25$ occupies an unstable middle ground for those thresholds.

Furthermore, the configuration $\alpha = 0.75, ; \tau = 5$ delivers the strongest overall performance across all evaluated settings. However, this combination is not uniformly optimal across all behavior classes and decision scenarios. For instance, Attack and Investigate sometimes perform better under alternative configurations at higher thresholds. These findings underscore the importance of jointly tuning the reset factor and decision threshold, as their interaction can significantly influence performance depending on the characteristics and temporal structure of the data.

Table 5.2: F1 Scores across Decision Thresholds for Reset Factor $\alpha = 0.0$ - Online CBGT-Net-Dual Encoder

Threshold	**Attack**	**Investigate**	**Mount**	**Other**
5	0.305	0.514	0.874	0.935
10	0.312	0.511	0.832	0.910
15	0.319	0.510	0.827	0.881
20	0.350	0.494	0.757	0.833

Table 5.3: F1 Scores across Decision Thresholds for Reset Factor $\alpha = 0.25$ - Online CBGT-Net-Dual Encoder

Threshold	**Attack**	**Investigate**	**Mount**	**Other**
5	0.319	0.538	0.881	0.937
10	0.153	0.350	0.607	0.704
15	0.132	0.293	0.406	0.549
20	0.123	0.277	0.362	0.477

Table 5.4: F1 Scores across Decision Thresholds for Reset Factor $\alpha = 0.5$ - Online CBGT-Net-Dual Encoder

Threshold	Attack	Investigate	Mount	Other
5	0.322	0.543	0.878	0.940
10	0.323	0.555	0.831	0.901
15	0.352	0.521	0.820	0.885
20	0.330	0.487	0.766	0.854

Table 5.5: F1 Scores across Decision Thresholds for Reset Factor $\alpha = 0.75$ Online CBGT-Net-Dual Encoder

Threshold	Attack	Investigate	Mount	Other
5	0.333	0.540	0.8776	0.940
10	0.328	0.523	0.8640	0.918
15	0.327	0.521	0.8220	0.886
20	0.313	0.485	0.7900	0.856

Table 5.6: F1 Scores for '**Attack**' $\tau = 5$

Reset Factor	Differential Encoder	Dual Encoder
0	0.275	**0.305**
0.25	0.281	**0.319**
0.5	0.288	**0.322**
0.75	0.288	**0.333**

Table 5.7: F1 Scores for '**Investigate**' $\tau = 5$

Reset Factor	Differential Encoder	Dual Encoder
0	0.514	**0.514**
0.25	**0.548**	0.538
0.5	0.539	**0.543**
0.75	**0.583**	0.540

To analyse the performance of the Online-CBGT-Net Dual Encoder against the LSTM baseline, we analyze the Dual Encoder model on multiple α values at $\tau = 5$. The model outperforms the LSTM baseline for all values of α. The model demonstrates a 31.72% and 33.05% increase in the F1-score for the *Attack* and *Investigate* behaviors as in seen in Figure 5.13. It is also observed that the model performs the best at a

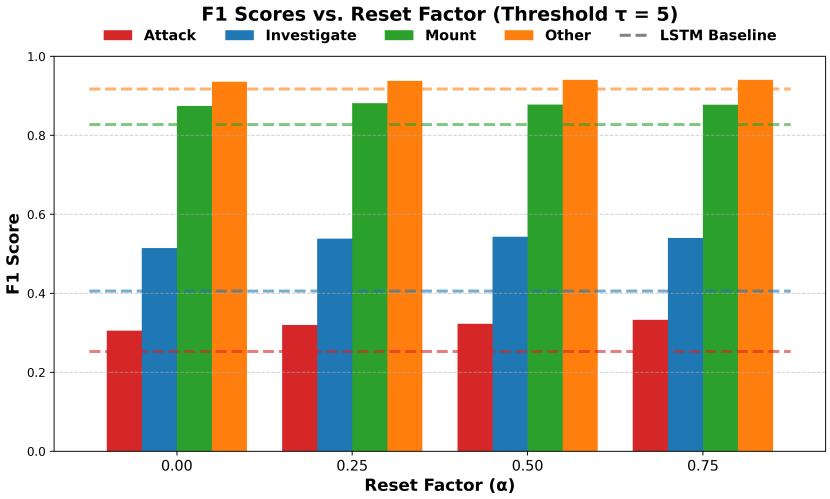

Figure 5.13: F1 Score performance of the Model at $\tau = 5$ and various reset factors vs the LSTM Baseline

Table 5.8: F1 Scores for '**Mount**' $\tau = 5$

Reset Factor	Differential Encoder	Dual Encoder
0	0.857	**0.874**
0.25	0.863	**0.881**
0.5	0.862	**0.878**
0.75	0.864	**0.877**

Table 5.9: F1 Scores for '**Other**' $\tau = 5$

Reset Factor	Differential Encoder	Dual Encoder
0	0.9260	**0.9357**
0.25	0.9301	**0.9378**
0.5	0.9288	**0.9408**
0.75	0.9302	**0.9405**

reset factor of **0.75**, especially for the Attack and Investigate behaviors.

We evaluate the Differential Encoder variant at a decision threshold of $\tau = 5$ across reset factors $\alpha = 0.0, 0.25, 0.5, 0.75$. This choice is motivated by findings from the Dual Encoder experiments, where $\tau = 5$ consistently yielded the best F1 performance across all behavior classes and reset factors $(0.0, 0.25, 0.5, 0.75)$ as a fixed reference

point for evaluating the Differential Encoder across multiple reset configurations.

At this operating point, the Dual Encoder variant demonstrates stronger performance than the Differential Encoder across all reset factors for most behaviors, with comparable results for the Investigate class.

5.2.2 Precision

Model	**LSTM**	**Online CBGT (Threshold = 5)**	
Behavior		**Dual** ($\alpha = 0.75$)	**Differential** ($\alpha = 0.75$)
Attack	0.16	**0.24**	0.19
Investigate	0.55	**0.71**	0.64
Mount	0.76	**0.86**	0.86
Other	0.95	**0.97**	0.96

Table 5.10: Model Precision Comparison

The precision, summarized in Table 5.10, reveals that Online-CBGT-Net with Dual Encoder and reset factor $\alpha = 0.75$ substantially improves precision for all behavior classes over both the LSTM Baseline and the Differential Encoder Baseline for a threshold of 5:

- Attack: Precision improves from 0.16 (LSTM) to 0.24 (Dual), a 50% increase
- Investigate: From 0.55 (LSTM) to 0.71 (Dual) (+29% increase)
- Mount: From 0.76 (LSTM) to 0.86 (Dual) (+13% increase)
- Other: From 0.95 (LSTM) to 0.97 (Dual) (+2% increase)

5.2.3 Latency

To explore how the reset mechanism influences the Online-CBGT-Net model's responsiveness, we report latency for four representative reset factors: 0, 0.25, 0.5, and 0.75, at a threshold $\tau = 5$

As shown in Figure 5.14, latency decreases as the reset factor increases. This trend indicates that higher reset factors, which preserve more of the previously accumulated evidence after a decision, enable the model to respond more quickly to the onset of a new behavior.

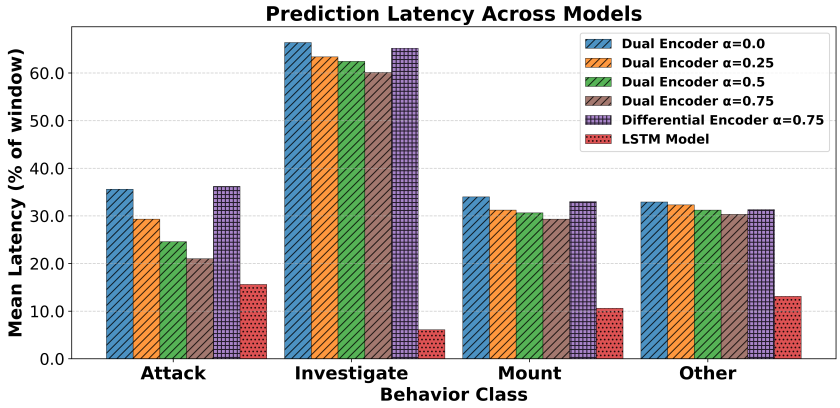

Figure 5.14: Latency of Online-CGT-Net vs Baselines

By retaining a greater portion of past evidence, the accumulator does not reset to zero after each decision. Among the settings tested, the model with $\alpha = 0.75$ exhibits the lowest latency, while the model with $\alpha = 0.0$ (i.e., no memory retention) has the highest latency.

This becomes particularly advantageous in scenarios where behavior transitions are smooth or where early signals of the next behavior class are already present before the previous prediction. With a higher reset factor(e.g., 0.75), the accumulator integrates these early cues more effectively, requiring fewer new frames to build up enough confidence. In contrast, a lower reset factor (e.g., 0.25) causes the accumulator to decay more aggressively after each prediction, effectively erasing much of the contextual buildup. As a result, the model needs to accumulate more fresh evidence from scratch, leading to increased latency before the next decision is made. Thus, a higher reset factor improves response by maintaining momentum in evidence accumulation across consecutive behaviors.

We also find that the Dual Encoder Model performs better than the Differential Encoder Model at a decision threshold of 5. One possible explanation is that the Dual Encoder, may learn a more expressive representation of the input feature. This could allow it to accumulate discriminative evidence between behaviors more effectively and classify the behavior more accurately.

The LSTM model exhibits lower latency compared to the best-performing Dual Encoder Model, specifically at its optimal configuration ($\tau = 5$ and $\alpha = 0.75$). This is

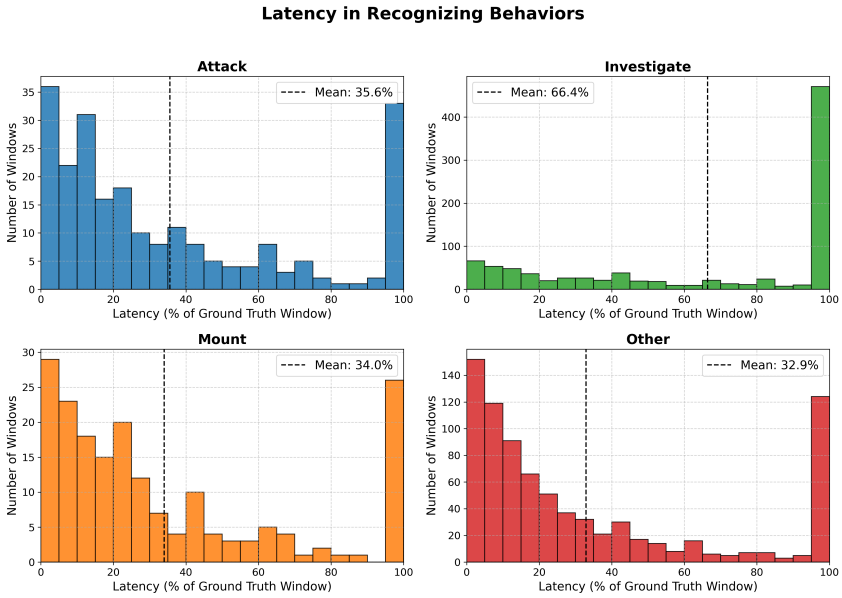

Figure 5.15: Latency of the Differential Encoder model for $\alpha = 0.0$ $\tau = 5$

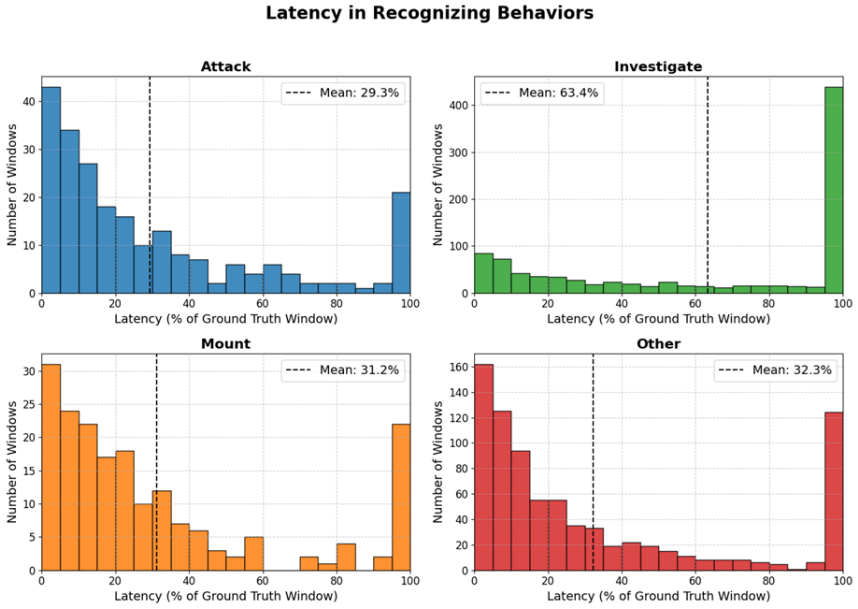

Figure 5.16: Latency of the Dual Encoder model for $\alpha = 0.25$ $\tau = 5$

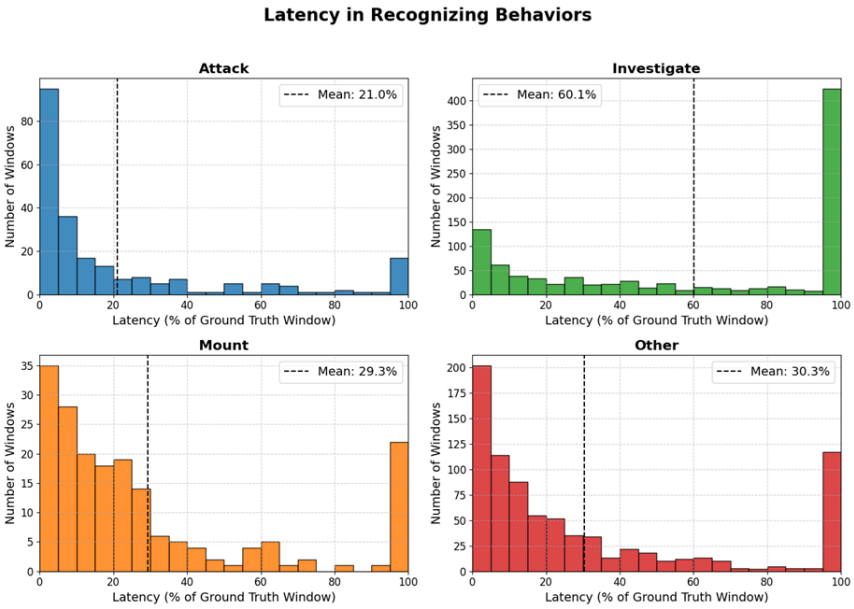

Figure 5.17: Latency of the Dual Encoder model for $\alpha = 0.75$ $\tau = 5$

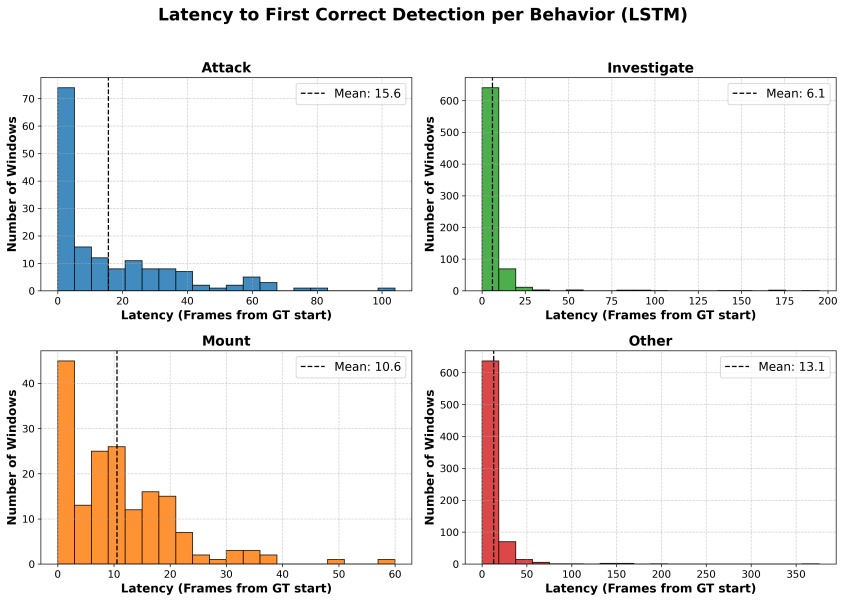

Figure 5.18: Latency of the LSTM Model

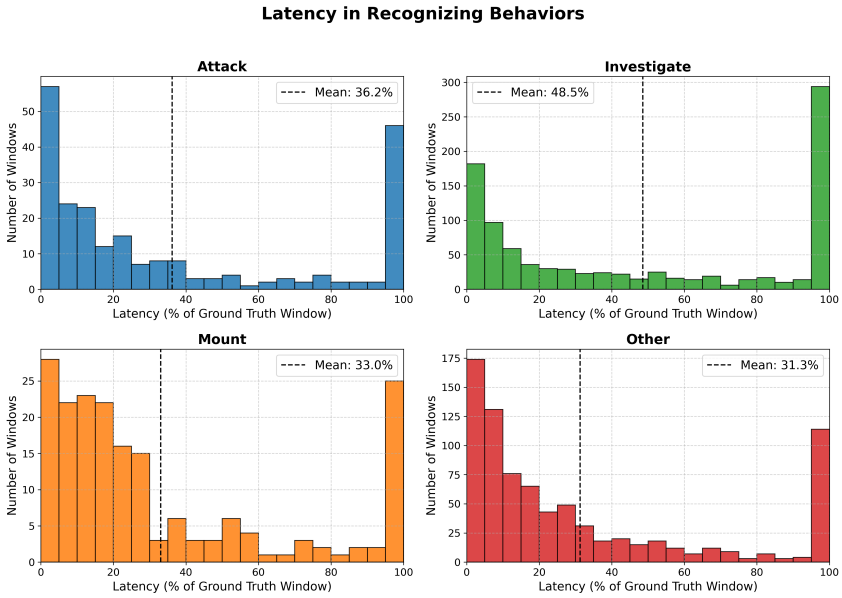

Figure 5.19: Latency of the Differential Encoder model for $\alpha = 0.75$ $\tau = 5$

primarily due to the nature of the LSTM working: it makes a prediction at every time step, allowing it to chance upon the right prediction earlier than Online-CBGT-Net makes a decision. Additionally, this advantage in response time can also be attributed to the inherently more complex recurrent structure of an individual LSTM cell. Unlike a simple accumulator vector, which linearly aggregates evidence, an LSTM cell's intricate gating mechanisms, comprising input, forget, and output gates, alongside its hidden, and cell states, enable it to learn and maintain sophisticated temporal dependencies. These features allows the LSTM layer to have better latency when it does end up predicting the correct behavior.

Table 5.11: Dual Encoder Latency (%) across Decision Thresholds for Reset Factor $\alpha = 0.75$

Threshold	Attack	Investigate	Mount	Other
5	**21.0**	**60.1**	**29.3**	**30.3**
10	37.4	68.5	34.7	35.1
15	52.2	74.5	38.8	39.5
20	61.1	76.8	41.5	45.2

We also hypothesize that as the decision threshold for Online-CBGT-Net increases,

the latency of the model would increase correspondingly, since the model requires to gather more evidence before making a decision. This naturally leads to increased latency, as more time steps are needed for the accumulator to reach the higher threshold. As shown in Table 5.11, for reset factor $\alpha = 0.75$, latency consistently rises for the Online-CBGT-Net Dual Encoder variant with increasing thresholds across all behavior classes.

5.2.4 Summary

Table 5.12: Performance Summary: F1 Score, Precision, and Latency Comparison

Model	**F1 Score ↑**	**Precision ↑**	**Latency ↓**
LSTM	Lowest	Lowest	**Lowest**
CBGT (Diff, $\alpha = 0.75$)	Moderate	Moderate	Highest
CBGT (Dual, $\alpha = 0.75$)	**Highest**	**Highest**	Moderate

To holistically evaluate the performance of Online-CBGT-Net, we compare it against a recurrent LSTM baseline across three key metrics: F1 Score, Precision, and Latency. The detailed quantitative results are presented in Table 5.12 and the f1 scores, precision scores, and latency as discussed in Section 5.2.3, 5.2.2, 5.2.1.

The LSTM baseline exhibits the lowest latency, which is expected given its design: it produces a prediction at every timestep without accumulating evidence or assessing confidence. This allows the LSTM to respond immediately and, in some cases, anticipate the correct label earlier than Online-CBGT-Net. However, this early prediction strategy often comes at the cost of the F1 score and Precision.

In contrast, both the Dual Encoder and Differential Encoder versions of Online-CBGT-Net consistently outperform the LSTM in terms of Precision and F1 Score. This reflects a key architectural difference: Online-CBGT-Net is explicitly designed

to delay prediction until sufficient evidence is accumulated, favoring correctness over immediacy.

Among the evaluation metrics, Precision is especially important in the context of Online-CBGT-Net. It measures the proportion of predictions that are correct at the time they are made. For a deliberative architecture such as Online-CBGT-Net, which is intentionally selective about when to predict, high Precision validates its core design principle: to only act when confident. While this leads to increased latency, it results in a higher quality of predictions overall.

The F1 Score, which balances Precision and Recall, further highlights the advantage of deliberation. Online-CBGT-Net achieves better F1 performance, especially in behavior classes with higher ambiguity or temporal overlap (e.g., Attack vs Investigate), demonstrating the benefit of evidence accumulation before decision-making.

In summary, the experimental results underscore a fundamental trade-off: while the LSTM baseline favors immediacy, Online-CBGT-Net prioritizes confidence and accuracy. Its superior performance in Precision and F1 Score confirms the effectiveness of integrating deliberation into an online prediction setting.

5.3 Visualizing Model Decision Making

Beyond raw performance metrics, we aim to analyze the model's internal decision-making process to understand how and when it decides. Online-CBGT-Net's structure naturally enables this, as the decision output is governed by an explicit accumulation of evidence over time - akin to a deliberation process. We look at test examples of the model's decision making process. In Figures 5.20 and 5.21, the accumulator trace and the cumulative evidence trace for two different instances of the model that predict the behavior of the ground truth behavior Investigate are provided.

5.3.1 Accumulator Trajectories

We visualize the accumulator values for each class over time for a particular decision, up to the decision point. This gives a transparent view of how class-wise evidence builds up, including how competing classes interact before a decision.

From Figure 5.20 we can see that:

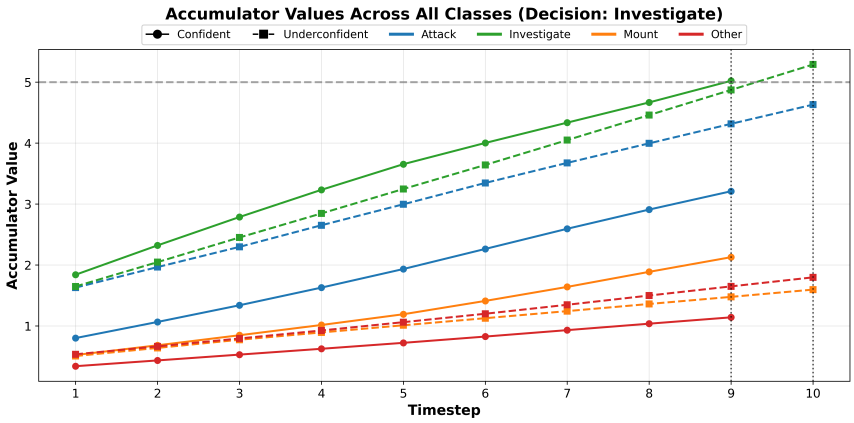

Figure 5.20: Accumulator trajectories for all behavior classes leading up to the decision time, shown for two predictions where the ground truth is Investigate.

- Confident decisions show a clear lead in the accumulator value for the predicted class. By the time a decision is made, there is a clear lead for the Investigate Accumulation (in solid green) over the next highest class Attack (in solid blue)
- Low-confidence decisions display narrow gaps between competing accumulators, indicating uncertainty. The predicted class Investigate (dashed green) and the competing class Attack (dashed blue) are both close to the decision threshold when the decision is made.

5.3.2 Go vs. No-Go Activation Trajectories

The Dual Encoder further enables decomposition of evidence into supporting (Go) and opposing (No-Go) signals. By analyzing these activations during the decision interval, we quantify the confidence gap in the model - defined as the difference between Go and No-Go activations for the predicted class.

As in Figure 5.21:

- Confident decisions show a clear lead difference in the Go and No-Go separation value for the predicted class. The difference in the activations grow over time as more evidence pours in, highlighting the model confidence in recognizing the behavior.
- The low-confidence decision (in dashed lines) shows competing Go and No-Go activations with a narrow gap in between them, highlighting the model's

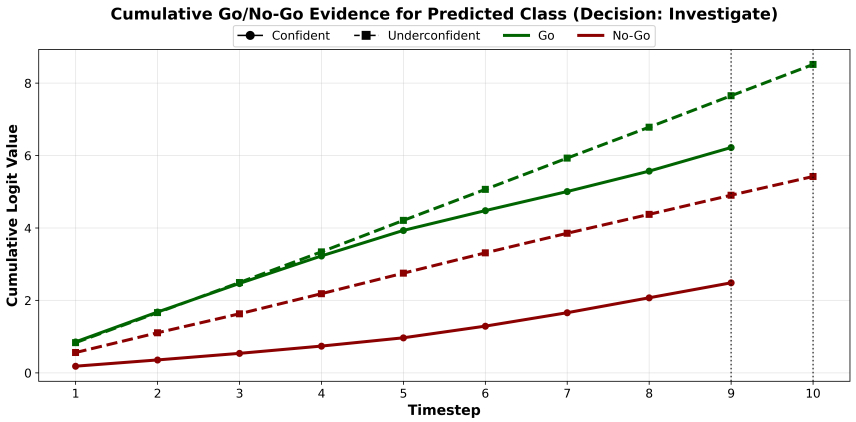

Figure 5.21: Cumulative Go and NoGo Evidence over time upto Decision time for predicted ground truth Investigate over two different decisions

uncertainty in recognizing the input. The model takes longer to predict the behavior and is also uncertain in its prediction (closer exactor values between the Attack and Investigate classes at time of decision).

5.3.3 Go and No-Go Evidence Distributions

To gain insight into the model decision making process, we analyze the evidence distributions of the Go and No-Go encoder branches. We observe that the Go pathway produces evidence in favor of the correct class and little to none in favor of the wrong behavior, while the No-Go pathway effectively suppresses competing alternatives by offering zero inhibiting evidence in favor of the correct behavior as seen in Figures 5.22, 5.23 and 5.24.

This separation of supportive and opposing evidence enhances interpretability by making it possible to trace which inputs drive the model toward or away from a decision.

A narrower distribution of the Go and No-Go prediction scores indicates a higher confidence in the model, as it reflects less variability in the output probabilities across time and samples. In other words, when the model consistently outputs a sharply peaked distribution centered around a specific behavior class, it is more certain in its decision-making.

From the plotted results, we observe that the model trained with a reset factor of

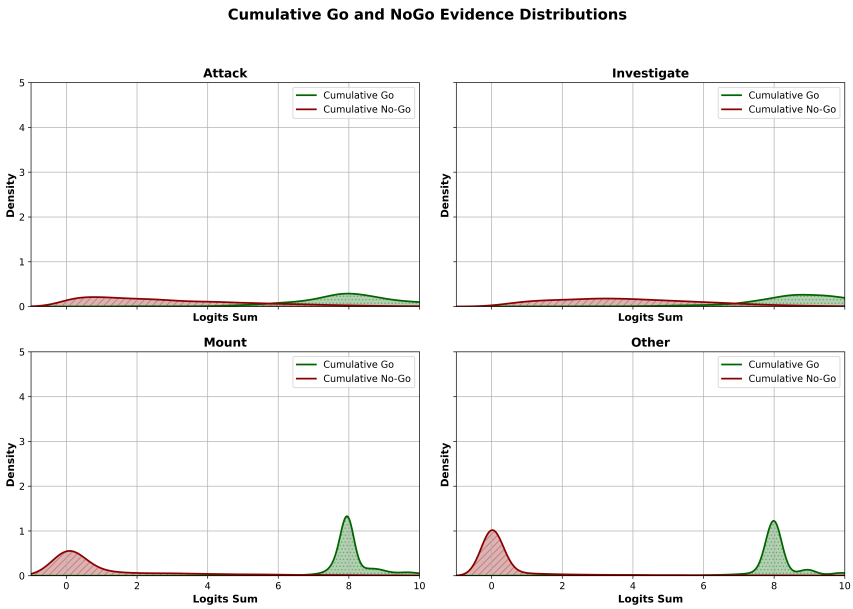

Figure 5.22: Cumulative Go and No-Go Evidences for $\alpha = 0.0$ $\tau = 5$

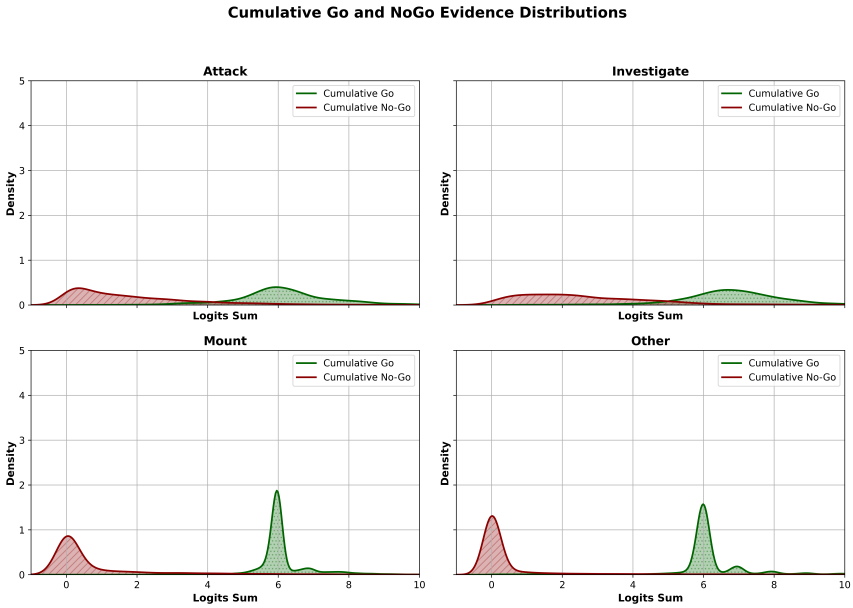

Figure 5.23: Cumulative Go and No-Go Evidences for $\alpha = 0.25$ $\tau = 5$

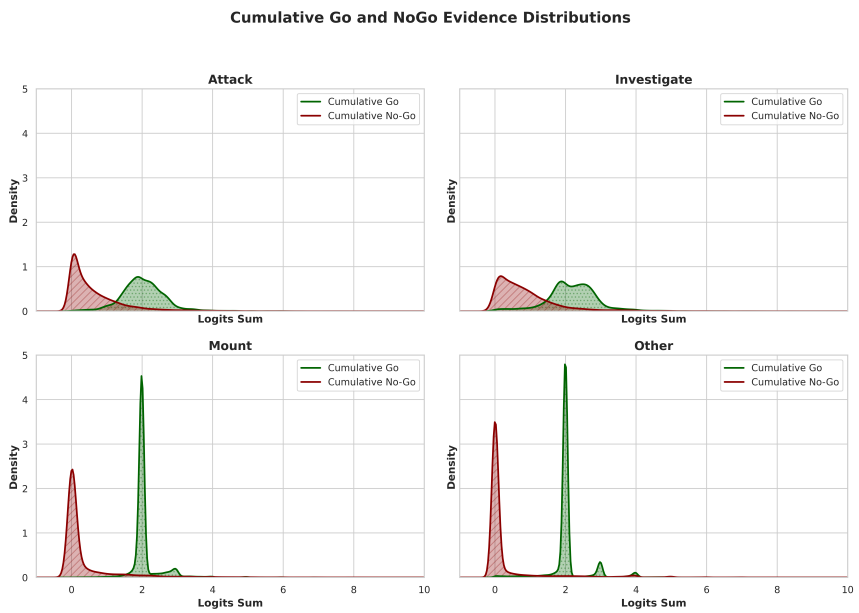

Figure 5.24: Cumulative Go and No-Go Evidences for $\alpha = 0.75\tau = 5$

$\alpha = 0.75$ exhibits the narrowest Go and No-Go distributions. This suggests that the model can accumulate evidence more decisively and commit to a behavior prediction with high confidence. In contrast, the model with $\alpha = 0.0$ which lacks any memory of previous evidence, produces much wider distributions, indicating lower certainty and more fluctuation in its predictions.

This trend highlights the role of the reset factor in stabilizing behavior classification. The higher the reset factor, the more prior context the model preserves in making a future decision, especially in predicting continued instances of the same behavior occurring over time, helping it learn better.

Furthermore, when the model predicts a behavior class X, we analyze the distribution of accumulated Go and No-Go evidence that led to the decision, not only for the predicted class X, but also for competing alternatives (see Figures 5.25,5.26, 5.27). A consistent and interpretable pattern emerges in the evidence profiles.

Specifically, when a behavior is confidently predicted, say, Mount, the accumulated Go evidence for Mount is strongly elevated, while its corresponding No-Go evidence remains near zero. This reflects a strong internal support for the chosen class and minimal inhibition. In contrast, for all non-Mount classes, the pattern is reversed: their Go evidence is suppressed, while their No-Go signals are elevated. The model is

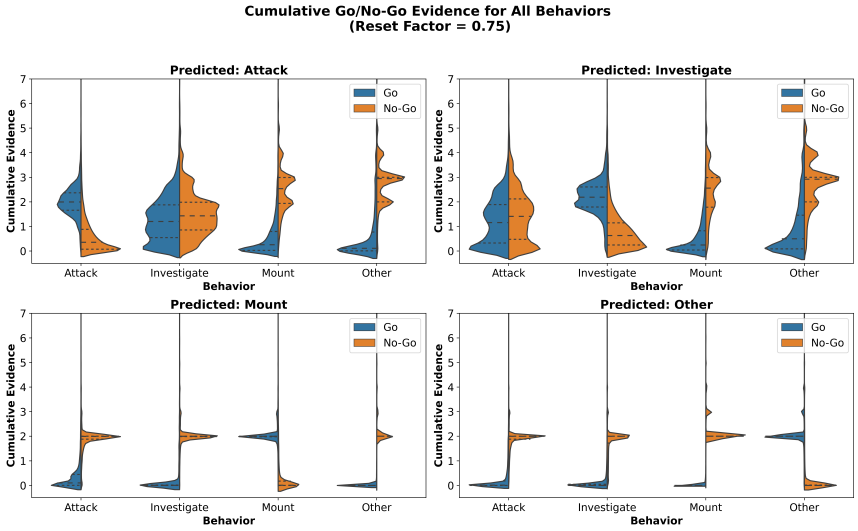

Figure 5.25: Cumulative Go and No-Go Evidences for $\alpha = 0.75$ $\tau = 5$ for all Behaviors

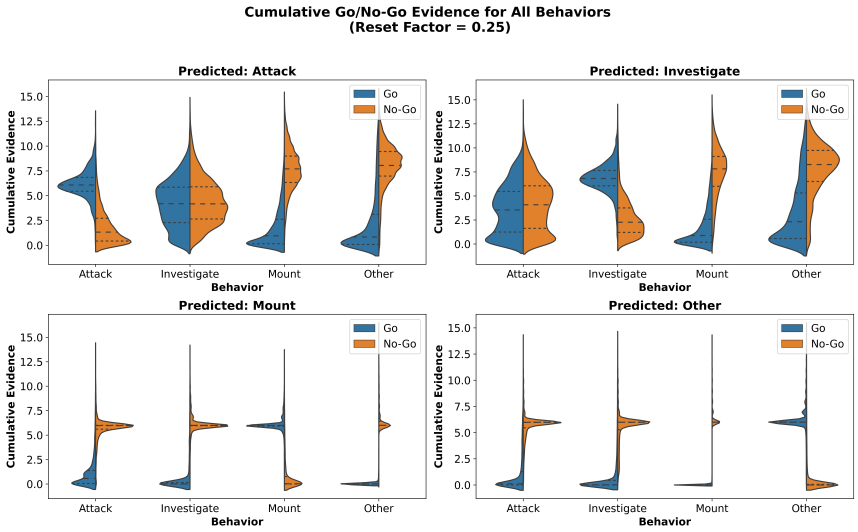

Figure 5.26: Cumulative Go and No-Go Evidences for $\alpha = 0.25$ $\tau = 5$ for all Behaviors

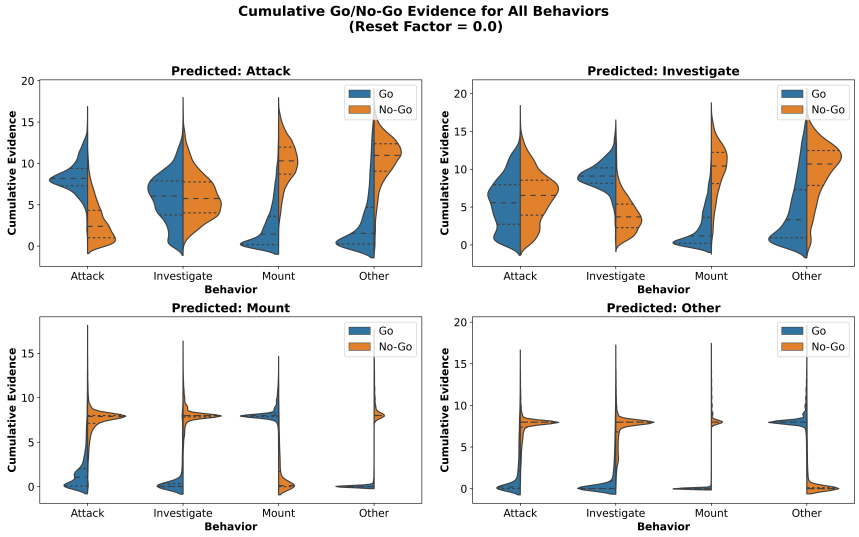

Figure 5.27: Cumulative Go and No-Go Evidences for $\alpha = 0.0$ $\tau = 5$ for all Behaviors

not simply selecting Mount; it is actively rejecting the alternatives.

This separation is especially prominent for confidently predicted behaviors such as Mount and Other, where the Go/No-Go evidence distributions are clean and decisive.

The implications of this framework extend beyond performance metrics. In human-agent teaming scenarios, for example, the Go/No-Go evidence provides an interpretable window into the model's confidence. If the model predicts Mount, with high Go and negligible No-Go for that class and strongly suppressed Go for others, a human supervisor can treat that output as a high-confidence unambiguous signal. However, in more ambiguous cases, such as a prediction of Attack in which Go for Attack is active, but No-Go for a visually similar class as Investigate remains low, there is visible internal conflict. The model appears to be partially convinced by an alternative class, revealing uncertainty. Such patterns enable human collaborators to make better informed decisions. A supervisor may choose to verify uncertain predictions, request additional context, or combine the model's output with other sensory or behavioral cues.

Chapter 6

Discussion

This thesis presents Online-CBGT-Net, a neural network architecture inspired by the CBGT Circuit to perform online prediction in streaming environments. By introducing a configurable reset mechanism and a Dual Encoder, the architecture captures temporal dynamics in sequential decision-making tasks while preserving interpretability. Our results show that the model performs robustly on imbalanced, noisy behavioral data by accumulating evidence over time and deferring classification until sufficient confidence is reached. The reset factor enables memory retention and responsiveness: lower values discard retain less accumulated evidence, making the model slower to respond, while higher values retain more accumulated evidence, allowing for quicker decisions. Furthermore, separating Go and No-Go evidence pathways allows us to observe which parts of the input support or oppose a behavior, offering greater interpretability into the model's deliberative process.

The transparency introduced by evidence accumulation opens new possibilities for interpretable machine learning, especially in human-interactive systems where it is essential to understand when and why a decision is being made. Rather than producing opaque predictions at fixed intervals, the model allows decisions to emerge dynamically from accumulated input - providing both timing and rationale behind predictions and its confidence in making such predictions. This opens the door for interventions, debugging, and collaborations where understanding the model's internal confidence is crucial.

Future work may explore adaptive reset factors based on the model's decision at

that instant. These adaptive reset factors can help the model to selectively preserve the extent of accumulated evidence for each behavior based on the current accumulated evidence. Integrating memory components that selectively retain historical evidence over longer horizons can also enhance the ability of the model to track behaviors that unfold slowly over time. The recurrent mechanism in the LSTM model allows it to exhibit better persistence and latency, This prompts investigating the utility of such mechanism in the accumulator of Online-CBGT-Net in the future. Additionally, to better capture multi-agent dynamics, we can introduce Graph Neural Networks into the architecture. In this setup, each agent becomes a node in a graph, and edges can represent interactions. This approach naturally scales to multiple agents and allows the model to reason about relational context between the two agents (such as interactions and behavior), making predictions more informed and situationally aware.

Overall, Online-CBGT-Net represents a step toward closing the gap between biologically-plausible decision-making systems and real-world machine learning tasks. Its emphasis on deliberation and interpretability makes it a strong candidate for deployment in settings where trust, timing, and context-sensitive predictions are essential.

Bibliography

[1] Akshat Agarwal, Abhinau Kumar V, Kyle Dunovan, Erik Peterson, and Timothy Verstynen. Better safe than sorry: Evidence accumulation allows for safe reinforcement learning. *arXiv preprint arXiv:1809:09147*, 2018. (document), 2.1

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 2.3

[3] Kyle Dunovan and Timothy Verstynen. Believer-skeptic meets actor-critic: rethinking the role of basal ganglia pathways during decision-making and reinforcement learning. *Frontiers in neuroscience*, 10:106, 2016. 1

[4] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 1

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 5.1.3

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2.3

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5.1.4

[8] Tiago V Maia and Michael J Frank. From reinforcement learning models to psychiatric and neurological disorders. *Nature neuroscience*, 14(2):154–162, 2011. 2.1

[9] Catherine E Myers, Alejandro Interian, and Ahmed A Moustafa. A practical introduction to using the drift diffusion model of decision-making in cognitive psychology, neuroscience, and health sciences. *Frontiers in Psychology*, 13: 1039172, 2022. 2.2

[10] Farzaneh Olianezhad, Sajjad Zabbah, Maryam Tohidi-Moghaddam, and Reza Ebrahimpour. Residual information of previous decision affects evidence accumulation in current decision. *Frontiers in behavioral neuroscience*, 13:9, 2019. 1,

2.2

[11] M Andrea Pisauro, Elsa Fouragnan, Chris Retzler, and Marios G Philiastides. Neural correlates of evidence accumulation during value-based decisions revealed via simultaneous eeg-fmri. *Nature communications*, 8(1):15808, 2017. 2.1

[12] Roger Ratcliff. A theory of memory retrieval. *Psychological review*, 85(2):59, 1978. 2.2

[13] Roger Ratcliff and Gail McKoon. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20(4):873–922, 2008. 2.2

[14] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning internal representations by error propagation, 1985. 2.3

[15] Shreya Sharma. Cbgt-net: A neuromimetic architecture for robust classification of streaming data. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, June 2024. 2.4

[16] Shreya Sharma, Dana Hughes, and Katia Sycara. Cbgt-net: A neuromimetic architecture for robust classification of streaming data. In *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, pages 1–6, 2024. doi: 10.1109/ICHMS59971.2024.10555685. 1, 2, 2.4

[17] Gabriel M Stine, Eric M Trautmann, Danique Jeurissen, and Michael N Shadlen. A neural mechanism for terminating decisions. *Neuron*, 111(16):2601–2613, 2023. 1

[18] Andrea Stocco, Christian Lebiere, and John R Anderson. Conditional routing of information to the cortex: a model of the basal ganglia's role in cognitive coordination. *Psychological review*, 117(2):541, 2010. 2.1

[19] Jennifer J. Sun, Tomomi Karigo, Dipam Chakraborty, Sharada P. Mohanty, David J. Anderson, Pietro Perona, Yisong Yue, and Ann Kennedy. The multi-agent behavior dataset: Mouse dyadic social interactions. *CoRR*, abs/2104.02710, 2021. URL https://arxiv.org/abs/2104.02710. (document), 5.1.1, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6

[20] Zhuojun Yu, Timothy Verstynen, and Jonathan E Rubin. How the dynamic interplay of cortico-basal ganglia-thalamic pathways shapes the time course of deliberation and commitment. *bioRxiv*, pages 2025–03, 2025. 1