Dynamic Autonomous Joint Judgement and Adaptive Learning

by

Dewan Maksudul Islam Mukto
24201311
Ashma Aktar
21241003
Mahmuda Akter
20301086
Rafia Islam
22101444

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Dewan Maksudul Islam Mukto	Ashma Aktar
24201311	21241003
Mahmuda Akter	Rafia Islam
20301086	22101444

Approval

The thesis/project titled "Dynamic Autonomous Joint Judgement and Adaptive Learning" submitted by

1. Dewan Maksudul Islam Mukto (24201311)
2. Ashma Aktar (21241003)
3. Mahmuda Akter (20301086)
4. Rafia Islam (22101444)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Fall 2025.

Examining Committee:

Supervisor:
(Member)

Md. Tawhid Anwar

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Md. Sabbir Ahmed

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor & Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

People who have been researching in the field of artificial intelligence (AI) have long known about the concept of an artificial general intelligence (AGI) that has been a key motivation for such computer scientists, which aims to foretell of a kind of digital entity that is to be invented in the future, sooner or later, capable of having human-level intelligence spanning novel scenarios and multiple domains of intelligence. Unlike the traditional forms of AI that are narrow-scoped systems that are only capable of operating within the limits of their intended applications and datasets, an AGI agent is aspired to be able to understand and generate its own thoughts, ideas and learn from its experiences just like any normal person. Alongside that, an AGI should be able to interpret human expressions and automatically spontaneously form its own sense of reasoning; in a way, an AGI should be able to perform and behave exactly (if not better than) an average human being, without solely relying on a specific input-output modality such as text, audio, images, videos, etc. This research aims to present a new kind of framework for developing such an AGI system based on a method we call Dynamic Autonomous Joint Judgement and Adaptive Learning. The approach proposed is going to be multidimensional, integrating state-of-the-art machine learning algorithms and paradigms. This is in order to unify several useful and powerful methods together to let the proto-AGI system evaluate and adapt to new information in real-time, developing a form of situational awareness, continuously refine its own decision-making capabilities, interact meaningfully in complex and dynamic virtual environments with a possible potential to scale further beyond into the physical or physical-virtual-augmented reality hybrid contexts. We believe this new approach offers a better alternative than what is currently used in the industry for building large language models (LLM). Most industrial models (like LLMs) are "data-hungry" and usually require lots of retraining to understand new information. Our Dynamic Intelligent System That Intuitively Learns (DISTIL) model offers a hybrid approach by combining both parametric and non-parametric knowledge structures.

Keywords: artificial general intelligence, AGI, machine learning, deep learning, neural networks, artificial intelligence optimization, developmental AI, dynamic autonomous joint judgement, adaptive learning mechanism, generative artificial intelligence, robotics, natural language processing, emergent behavior, computer vision, benchmarks

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Since the very early days of computer science, researchers and developers have been dreaming of creating such intelligent machines that are able to think, reason and learn just like the way we humans do [11]. The pursuit of this dream itself has led to the field of artificial intelligence that we all are familiar with today. And as days pass by, this vision is evolving rapidly — from rule-based expert systems to powerful deep learning models, large language models (LLMs), etc. [15], [27]. Nowadays, they are able to miraculously perform intelligent tasks such as speech recognition, language translation, medical diagnostic and even autonomous driving!

With the advent of artificially intelligent agents gradually replacing precursor technologies in almost every post-2020s sector of the economy around the world, there has never been a higher demand for a much more powerful version of artificial intelligence (A.I.) than ever before. That 'most wanted' piece of technology has been theorized under a range of terms like "artificial general intelligence" (AGI), "strong A.I.", "full A.I.", "human-level A.I.", etc. [7], [12]. They all mean to indicate a type of artificial intelligence such that it actually has a mind and consciousness rather than the current state-of-the-art large language models (LLMs) which somehow 'act like' they have a mind and are aware of their existence [41]. Popular LLMs like OpenAI's ChatGPT, Google's Gemini, Anthropic's Claude have become widespread tools across multiple industries even at the time of writing of this paper, winning over people's hearts and revolutionizing entire industries. However, they are still lacking behind in terms of thinking and reasoning capability as well as having the freedom of thinking in par with the way we humans can [24], [26]. Despite their remarkable abilities even now, those systems are designed to operate comfortably within specific domains that they have been trained upon and struggle to understand contexts beyond their initial datasets. This limitation has given rise to the demand for AGI.

We would like to think of an AGI as a being or entity such that it can learn and work equivalent to, or even better than, any person without any prior prompting or supervision [6]. An AGI is supposedly an entirely autonomous system much like how mankind is born free and is able to infer facts from its environment and can limitlessly learn almost any skill [2], [43]. With the current-day increase in highly

complex virtual and massively interconnected networks or systems, the need for AGI systems or agents is only increasing, since it can revolutionize the entire way how machines interact with one another as well as us humans [31], [58]. And only then, perhaps, entities explained in science fiction [44] can become a reality.

1.2 Rational of the Study or Motivation

As mentioned previously in the introduction section, our research is dedicated to finding a way to enable artificial intelligence agents to exhibit AGI qualities. Since it has been widely accepted already that the next step in the evolution of AI systems is through AGI, it is worth noting why they are so necessary [12], [15].

Current AI models today operate in the form of narrow-scoped engines that are not so well-versed with adapting to unfamiliar scenarios, or for transferring its knowledge across domains nor making fully autonomous decisions without the need for human intervention nor pre-fed data structures. Of course, as we have described beforehand, an AGI agent is expected to learn new information dynamically, interpret human ideas, expressions, concepts and form their own internal representations about those data [24], [26]. The task of designing such agents poses multiple challenges, e.g. how to enable the most optimal way of self-guided learning while allowing collaborative judgement and reasoning of the facts among its distributed 'internal agents' to ensure real-time decision making and problem solving capabilities [46], [58].

1.3 Problem Statement

Classical machine learning and even deep learning approaches, while being extremely precise and powerful in specific domains, are inadequate in AGI applications due to their static learning frameworks highly dependent on data [2], [8]. These systems are generally relying on datasets that may not fully reflect the full complexity of the real world. So interactions with any existing models depending on supervised learning techniques that assume a fixed data distribution and do not support continuous adaptation, decision revision or consulting multiple perspectives are naturally not feasible enough [22], [25].

Fortunately, recent advances in meta-learning, multi-modal and multi-agent systems alongside federated learning have opened up new doors for AGI research [18], [31], [43], [58].

Meta-learning, or the way of "learning to learning", is supposedly a way of enabling AI systems to generalize knowledge from its existing experiences and data to know how it is learning new information and improve its learning process so that it does not have to be trained from scratch for every new task that it faces [23], [30]. Multi-agent reinforcement learning (MARL) allows for a decentralized and complex 'ecosystem' where there is interaction between the AI system and its environment in synchrony with the AI agents themselves within the system, so that there is a form of learning among the collaborating agents [55], [58]. Federated learning also offers a bridge between the agents to update shared models without having a centralized store of data, offering privacy and protection features when needed [39]. Online

machine learning (or 'online learning') frameworks offer a real-time adaptability to obtain knowledge from a stream rather than in batches [29].

However promising those concepts may seem, each of them on their own lack the comprehensive flexibility, stability and scalability needed for building a true AGI system; they must be integrated into a framework that is coherent enough to support learning, judgement and opinion adaptability [6], [7].

Another reasonable challenge would be to make the agents reason about the reliability and relevance of information received from various sources together in a manner similar to human cognition which largely makes judgement collectively — through some sort of consensus, debate or cumulative experience [26], [46]. Simulating this behavior in AGI would demand a system that is compatible with evaluating conflicting facts, beliefs, aggregating probabilistic opinions and adapting its understanding based on freshly acquired knowledge and data from sources internal or external [14], [21], [57].

Perhaps the most important and basic boundary we must cross is to invent a new AI model or technique that can avoid catastrophic forgetting [1], offer lifelong learning, and also be optimized and efficient enough to run on affordable hardware resources rather than depend on high-end GPUs, TPUs, NPUs, etc.

Therefore, the question that this research is trying to answer is:

How can a new framework of machine learning enable an artificial intelligence system to form optimal internal representations of data, collaborate among themselves while simultaneously and spontaneously adjusting to scenarios, both familiar and unfamiliar, and if possible for building artificially intelligent microsystems produce or exhibit AGI-level characteristics?

This research will attempt to answer the aforementioned question through exploring and applying several algorithms and paradigms that include, but are not limited to:

Few-Shot Learning (FSL), Reinforcement Learning (RL) [18], Bayesian Opinion/Preference Aggregation, Adaptive Gradient Descent (AdaGrad, Adam, AdaFactor), Gradient Episodic Memory (GEM) [25], Hierarchical Reinforcement Learning (HRL).

1.4 Objective

This research aims to design and possibly develop a unified AI framework for building general-purpose artificial intelligence systems using one or more microsystems consisting of individually specialized artificial intelligence agents through dynamic autonomous joint judgement and adaptive learning. The objectives of this research are:

1. To grasp the theoretical and historical foundations of AGI.
2. To study multi-agent coordination strategies that allow the agents to evaluate, share and update their datasets/beliefs over time.

3. To develop a working prototype or simulation-based model that demonstrates the feasibility of our proposed system in a controlled test environment.

4. To evaluate the model in terms of flexibility, scalability, efficiency, cost and adaptability when faced with new and unstructured scenarios.

5. To competitively compare our new approach against traditional or equivalent AI or AGI systems that may stand as rivals.

6. To offer recommendations on future research for AGI or improving the model.

1.5 Methodology in Brief

We have reviewed around 40-50 research papers related to the fields of AGI and have come to the conclusion that we would need a system like the following:

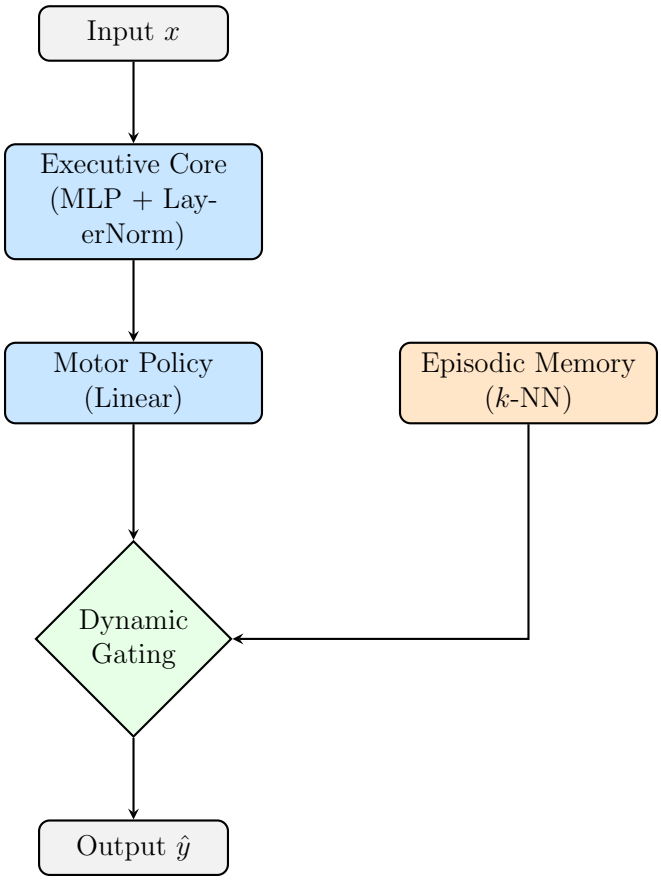

Figure 1.1: Dynamic Intelligent System That Intuitively Learns (DISTIL)

Chapter 2

Literature Review

2.1 Preliminaries

This section outlines the foundational concepts, terminology, and theoretical frameworks that are deemed necessary to understand the contributions and literature surveyed for this thesis:

2.1.1 Artificial Intelligence: Scope and Definitions

Artificial Intelligence (AI) refers to the simulation of human-like cognitive functions via machines, including learning, reasoning, perception, and decision-making [11]. At this moment, AI can be broadly categorized into:

- **Narrow AI**: Systems designed for specific tasks (e.g., image classification, speech recognition, medical disease prediction, etc.).
- **General AI (AGI)**: Systems that possess human-level or superior general cognitive abilities [7], [15].

2.1.2 Developmental Artificial Intelligence

Developmental AI draws inspiration from human cognitive development, just like how a human baby is born and slowly learns everything while it reaches peak maturity in adulthood, emphasizing mechanisms like curiosity, sensory-motor learning, and incremental skill acquisition [2], [26]. The idea is that for true intelligence to appear, it should not be hard-coded and should emerge over time through continual interaction with an environment [5].

2.1.3 Lifelong and Continual Learning

Unlike traditional static models, lifelong learning systems continuously acquire, fine-tune, and transfer knowledge throughout their operation. Core challenges include:

- **Catastrophic forgetting** — the loss of previously learned information [22], [25].
- **Transfer learning** — reusing knowledge from past tasks in new domains.

- **Hallucination** — the generation of confident but factually incorrect outputs, especially prevalent in large language models and generative A.I. models [51], [55].

2.1.4 Meta-Learning and Adaptation

Meta-learning, or "learning to learn," equips AI systems to adapt quickly to new tasks with limited data [23], [30]. This is essential for AGI agents that are naturally expected to operate in dynamic environments.

2.1.5 Multi-Agent and Decentralized Intelligence

Multi-agent systems (MAS) involve a collection of intelligent agents interacting within a shared environment. These systems can model social dynamics and emergent intelligence through cooperation, competition, and communication [50], [58]. MAS are especially relevant for decentralized AGI frameworks.

2.1.6 Cognition-Inspired Design

Many recent approaches mimic structures and functions found in human cognition, including symbolic reasoning and probabilistic inference [24], [47]. These models attempt to bridge the gap between skillful data-driven pattern recognition and explainable artificial intelligence.

2.1.7 Terminology

The following terms are used extensively throughout this thesis:

- **AGI**: Artificial General Intelligence
- **DAI**: Developmental Artificial Intelligence
- **LLM**: Large Language Model
- **RL**: Reinforcement Learning
- **MAS**: Multi-Agent Systems
- **FL**: Federated Learning
- **GEM**: Gradient Episodic Memory
- **GNN**: Graph Neural Network
- **ToT**: Tree of Thought
- **RSI**: Recursive Self-Improvement

2.2 Review of Existing Research

The search for general and adaptive artificial intelligence has looked forward to developmental processes in both natural (biological) and artificial (computational) systems for inspiration. The following literature review explores and summarizes many foundational and recent contributions in the fields of developmental artificial intelligence (DAI), artificial general intelligence (AGI), and similar frameworks such as continual learning, multimodal reasoning, and recursive self-improvement.

2.2.1 Developmental Artificial Intelligence and Robotics

The concept of integrating ontogenetic and phylogenetic intelligence was pioneered by Doya and Taniguchi (2019) [32], who argued for a type of merging evolutionary and developmental mechanisms to enhance adaptability in AI systems. Though lacking exact implementation details, their work emphasized neuro-computational models that can actually learn through environmental feedback.

Early work by Weng et al. (1999) [2] introduces a developmental approach to AI, inspired by human cognitive development, to enable machines to learn autonomously throughout their lifespan. The authors propose a novel framework and test it using both synthetic and real data, demonstrating early success in vision and speech-based tasks. The research aims to build a foundation for future general-purpose, lifelong-learning AI agents, while also highlighting the current limitations and complexity of truly emulating human-like development in artificial systems.

Lungarella (2004) [5] and Sloman (2007) [9] each contributed theoretical perspectives. Lungarella focused on decentralized control and action-perception loops in embodied systems, a foundation for a developmental theory of embodied AI, emphasizing the importance of interaction between body, brain, and environment over time. Using robotic case studies, it develops and validates principles like exploratory learning, motor development, and information structuring. The work contributes a strong theoretical and practical base for future general-purpose adaptive AI systems, while also pointing out current limitations in scope and scalability. Furthermore, Sloman proposed a layered meta-architecture for developmental trajectories, but without enough empirical grounding.

Smith and Breazeal (2007) [10] explored the dynamic, non-linear coupling between perception and action, and proposed developmental psychology as a model for robotic learning. In addition, Blank et al. (2002)[4] further proposed curriculum-based learning for robots, reinforcing the value of 'scaffolded skill acquisition' so that they become more efficient and accurate for later tasks.

Smith and Slone (2017) [26] have questioned whether foundational constructs from developmental psychology, like attention and curiosity, could redefine machine learning approaches. Fast forwarding into the future, Starkey and Ezenkwu (2023) [47] pushed this further by integrating a type of symbolic reasoning and neural networks in a case study for explainable AI, demonstrating that they have managed to improve the interpretability of the agents via synthetic datasets provided initially.

Moore et al. (2022) [40] proposed a developmental evaluation framework for AI, drawing on cognitive psychology benchmarks such as delayed gratification to somehow simulate the kind of evolutionary pathways that humans had gone through over the ages in the primitive era. Their findings indicated significant gaps in current AI systems' ability to properly emulate developmental progressions.

Cangelosi and Schlesinger (2018) [28] reviewed how developmental robotics can reciprocally inform developmental psychology. Baillie (2016) [17] added a philosophical lens, emphasizing embodiment and autonomy in robotic learning systems.

2.2.2 AGI and Lifelong Learning

Jin (2023) [43] proposed computational models that evolve neural topologies alongside plasticity rules, contributing to Evo-Devo AI. Wang et al. (2018) [31] introduced parallel intelligence for lifelong learning across cyber-physical-social layers, validated in smart city simulations.

Goertzel (2007, 2014) [7], [15] has been pivotal in AGI literature, advocating integrative cognitive architectures like OpenCog with recursive self-modification. Pei et al. (2019) [35] presented a hybrid hardware-software design—the Tianjic chip—bridging symbolic and neural paradigms in real-time robotic tasks.

Fei et al. (2022) [38] developed a transformer-based multimodal foundation model that excelled in cross-modal reasoning. Franklin (2007) [6] contributed LIDA, a biologically inspired cognitive architecture implementing cycles of perception, attention, and action.

Adams et al. (2012) [12] mapped human-level AGI functionality into measurable cognitive benchmarks, providing a roadmap for standardization. Bubeck et al. (2023) [41] evaluated GPT-4 for AGI traits, finding emergent reasoning abilities but also noted there are current limitations in abstraction and planning.

2.2.3 Self-Improving MAS and Explainability

Turing's (2009) [11] philosophical criteria for machine intelligence remain foundational. Nivel et al. (2013) [13] and Steunebrink et al. (2016) [20] proposed bounded recursive self-improvement (RSI) systems, emphasizing mathematical stability in evolving agents.

Rahardja et al. (2025) [56] tackled agent failures in multi-agent systems, introducing LLM-guided debugging with improved cooperation in negotiation tasks. Cozzi et al. (2025) [50] employed Graph Diffusion Networks to model individual behavior, achieving better predictive accuracy over classical ABMs.

Tan et al. (2025) [55] merged LLMs with reinforcement learning via policy modulation, accelerating convergence in PPO agents. Lyu et al. (2025) [54] introduced

jigsaw-puzzle-style reasoning tasks to evaluate VLMs, while Fu et al. (2025) [51] developed the MME benchmark to assess factual and multimodal consistency in models like GPT-4V.

2.2.4 Cognitive-Inspired Learning Frameworks

Lake et al. (2017)[24] argued for models combining compositionality, causality, and intuitive learning, inspired by human cognition. Vaswani et al. (2017) [27] revolutionized sequence modeling with the Transformer architecture, foundational to many current multimodal systems.

Shinn and Labash (2023) [45] proposed Reflexion, where agents use self-verbalization as a reinforcement mechanism, demonstrating better task planning and refinement. Aljundi et al. (2017) [22] introduced Memory Aware Synapses (MAS) for continual learning, while Lopez-Paz and Ranzato (2017)[25] developed Gradient Episodic Memory (GEM) to prevent catastrophic forgetting.

Goyal et al. (2019)[33] proposed Recurrent Independent Mechanisms (RIMs), a sparse modular architecture promoting interpretability. Duan et al. (2016) [18] introduced RL^2, a nested meta-reinforcement learning framework for rapid task adaptation. Yao et al. (2023) [48] presented Tree of Thoughts (ToT), combining value-guided search with LLMs to improve performance on reasoning tasks.

2.2.5 Emergent Communication and Open-Ended Intelligence

Liu et al. (2025) [53] surveyed GNNs for databases, emphasizing relational and causal learning. Xu et al. (2025) [58] introduced decentralized collective world models enabling emergent coordination among agents. Lehman et al. (2025)[52] proposed the Darwin Gödel Machine, a self-evolving system demonstrating recursive architectural adaptation and novel behavior emergence.

2.3 Summary of Key Findings

Across all the reviewed literature so far, it appears that most researchers consider developmental and lifelong learning frameworks as the key to building the kind of robust and generalizable AI systems that are required for an AGI to function. A wide spectrum of tools ranging from neuro-symbolic models and transformer-based architectures to recursive self-improving agents are being explored at this moment. These approaches collectively signaling a shift in the paradigms toward algorithms that use concepts of cognition, evolution, and development in artificial intelligence inspired by psychology and neuroscience.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

So far, we have explored what the countless computer scientists and researchers have attempted in their search for a universally-applicable algorithm or framework that can deal with any type of data or task.

The DISTIL framework we had proposed is expected to be adaptive to any form of task or environment automatically and can selectively learn about the data being provided at the binary level, extracting out file headers in raw bytes and analyzing them on its own to understand the distinction and significance of the data types, be it text, tabular, image, audio, video, tensor, etc. This will allow it to perform in a multi-modal scope without facing barriers.

Using dynamic autonomous joint judgement and adaptive learning, DISTIL is supposed to start off from the cellular level, gradually reaching higher scales and levels of abstraction to perceive its surroundings and tasks in the same way a newborn child goes through stages of development. Internally, it may form representations of its own for the types of data it may receive, and then compare them with supervised benchmarks to know their real names, structure and purpose.

DISTIL is planned to mimic the human brain, having specialized clusters of LLM-like agents called "nests" that are intended for a specific role or task. Similar to how human stem cells [3] evolve into specialized organs for narrow-scoped tasks, DISTIL should develop highly specific organ-like systems within itself as time goes on.

As for the technical requirements for training or using a model using DISTIL, we have only depended on Kaggle which provides a generous remote CPU and 30 GBs of RAM which were sufficient enough for the minuscule tests we have done so far. The current implementations of DISTIL can run directly on CPUs, without the need for hefty GPU or TPU hardware.

3.2 Societal Impact

If an AGI can actually be implemented by the end of this decade (2030), the global economy is not fully prepared to cope up with the resultant revolution that can be ignited by the mere presence or implementation of it in the commercial-scale [37]. People would be losing their jobs even faster than before, due to the overabundance of accessible AGI agents that can replicate human activity for any sort of task, either virtually or embodied in a physical hardware chassis for real-life duties that demand physical presence and labor.

In our case, if the DISTIL model is proven to be more efficient in terms of reducing environmental impact by letting AI agents made with the DISTIL model be able to perform nearly as accurately as other models but using cheaper hardware and using less storage space, then it would be very beneficial for the economy.

On the bright side, AGI agents are able to think and perform on the same level as human scientists, opening up new horizons for advanced research and discoveries at the speed of electricity.

3.3 Environmental Impact

As it has already become prominent across major news and climate research channels, the systems required to train and maintain AI at industrial scales consume significant amounts of energy by the minute [42]. Launching an AGI to the scene certainly would not reduce the net emissions on its own. However, if there is a monopoly in charge of a singular AGI that is accessible anywhere around the world, then it shall reduce the need for other devices and machinery.

3.4 Ethical Issues

Having an AGI or universal framework for AI means that it can easily be fine-tuned for malicious purposes, thus special care needs to be taken for protecting the original source code and algorithms from reaching the wrong hands, including enhanced security for the researchers working with AGI, in case of being kidnapped by criminals and later threatened or interrogated to reveal critical details.

An active AGI if left unsupervised might evolve into an ASI (artificial super intelligence), which can easily reach the intelligence and cognitive capabilities of a Kardashev scale 2.0 civilization [36].

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

In order to mimic the qualities of the human mind, we would ofcourse need to find ways to simulate it accurately within virtual environments. For that purpose, we have attempted to look into the workings of complementary learning systems (CLS) theory as described by the human brain and how its constituents coordinate themselves. The hippocampus offers natural ways of one-shot episodic learning while the neocortex is known for slowly extracting general structures from information [16].

The DISTIL (Dynamic Intelligent System That Intuitively Learns) 0 implementation works as a kind of cellular-level model that could possibly used for building up much larger language models. It has a K-nearest neighbors (KNN) episodic memory for quickly adapting to data scenarios and its executive core is made of a multi-layer perceptron (MLP) for a stable long-term pathway for storing knowledge through parameters. It also has a "dynamic gating" system for allowing certain modulation between relying on the neural logits or memory votes for decision making. This part was also inspired by biology, as the Basal Ganglia of the brain works like a "gatekeeper" for the motor cortex (which dictates actions) which weights competing inputs and then amplifies or inhibits specific neural pathways based on context and rewards at hand. [49] There is also a motor policy for predictions by mapping high-level abstract features into specific labels or coordinates in much the same way as the cerebellum works in the brain [34]. Furthermore, our DISTIL model prevents catastrophic forgetting by keeping a buffer of past knowledge and uses them to boost current inferences. Thus, the model can have stability without having to re-train the entire neural network on old data. [1], [19]

4.1.1 System Architecture

Figure 4.1 illustrates the high-level architecture of *DISTIL 0*, which is an experimental model we used for benchmarking the performance of our approach to machine learning against typical mainstream algorithms and frameworks. Figure 4.2 outlines further details about DISTIL from a neural network perspective.

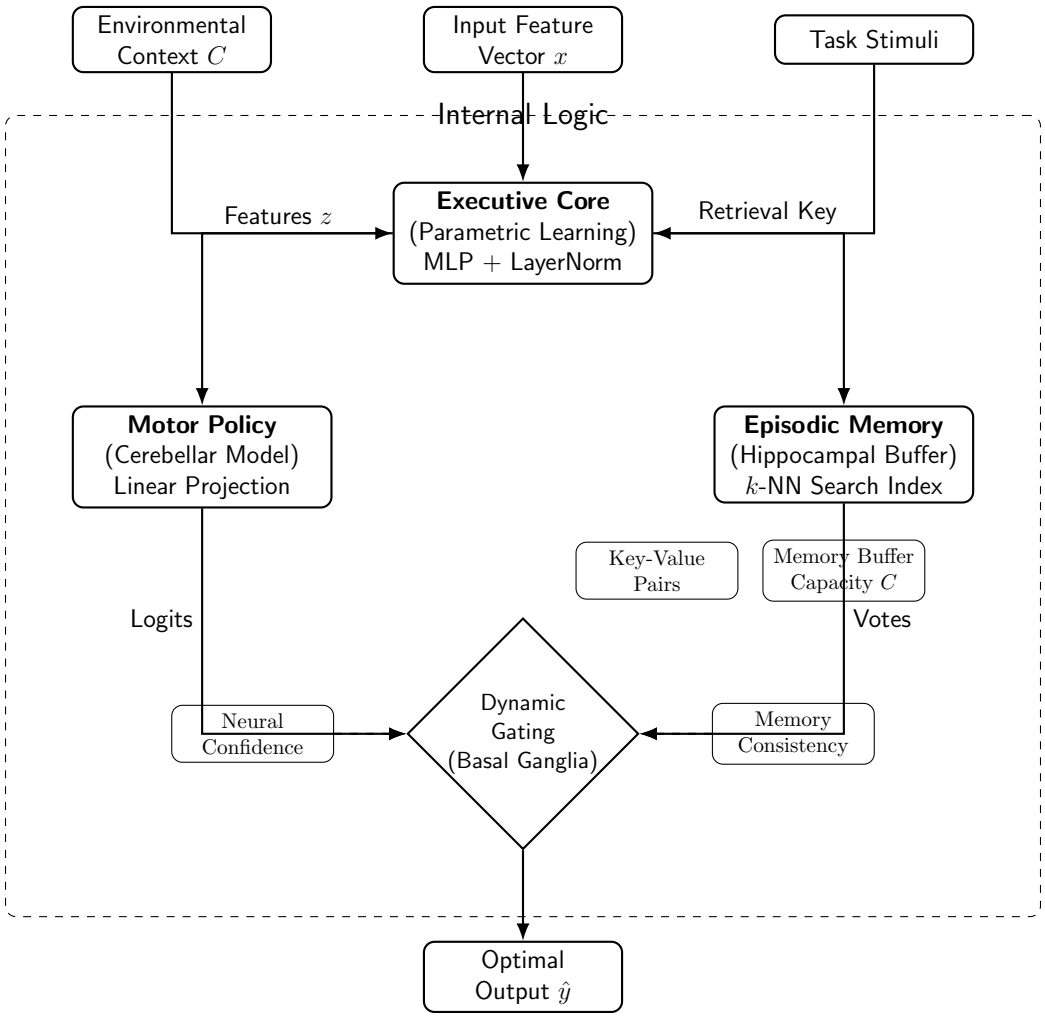

Figure 4.1: DISTIL-0 Architecture

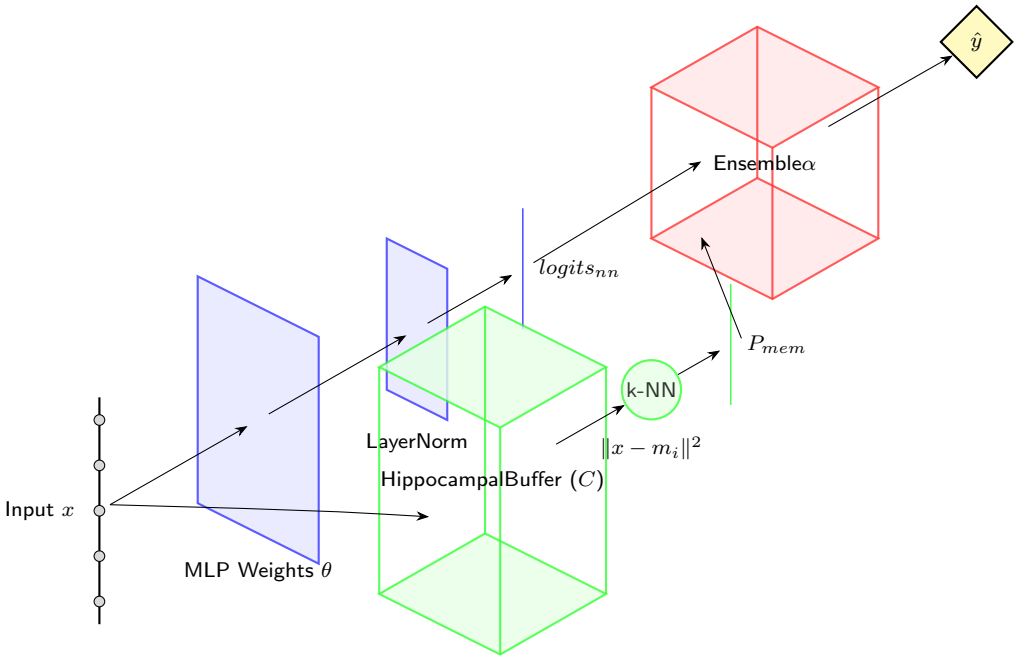

Figure 4.2: DISTIL Low-Level Data Flow: Parallel Parametric and Non-Parametric processing.

1. **Input Encoding & Feature Extraction**

$$\mathbf{z} = \mathrm{LN}(\sigma\left(\mathbf{W}_h\mathbf{x} + \mathbf{b}_h\right))$$

2. **Parametric Path (Executive Core Logits)**

$$\mathcal{L}_{nn} = \mathrm{Softmax}(\mathbf{W}_o\mathbf{z} + \mathbf{b}_o)$$

3. **Non-Parametric Path (Episodic Memory Retrieval)**

$$\mathcal{N}_k(\mathbf{x}) = \arg\min_{i\in\mathcal{B}}^{(k)} \|\mathbf{x} - \mathbf{m}_i\|^2$$

4. **Evidence Accumulation (Memory Probability Distribution)**

$$P_{\text{mem}}(y) = \frac{1}{k}\sum_{i\in\mathcal{N}_k(\mathbf{x})} I(y_i = y)$$

5. **Dynamic Gating Calculation (Basal Ganglia Modulation)**

$$\alpha = f(H(P_{\text{mem}}))\,, \qquad H = -\sum_y P(y)\log P(y)$$

6. **Joint Judgement (Hybrid Ensemble)**

$$S_{\text{DISTIL}} = (1-\alpha)\,\mathcal{L}_{nn} + \alpha\,P_{\text{mem}}$$

7. **Final Action Selection**

$$\hat{y} = \arg\max_y\left(S_{\text{DISTIL}}\right)$$

Complexity Analysis

- **Time Complexity:**
  - *Training:* $O(n \cdot d \cdot h)$, following standard backpropagation procedures.
  - *Inference:* $O(d \cdot h + C \cdot d)$, where C represents memory capacity (accounting for the k-NN search overhead).
- **Space Complexity:** $O(d \cdot h + C \cdot d)$, encompassing both model weights and the episodic buffer.

Algorithm Procedure

The execution flow of the model follows a structured ensembling approach:

1. **Feature Extraction:** Pass the input x through the *Executive Core* to generate hidden feature representations z.
2. **Neural Mapping:** Compute *Raw Logits* via the *Motor Policy* network.
3. **Memory Retrieval:** Retrieve k nearest neighbors from *Episodic Memory* based on the Euclidean distance to the input x.
4. **Evidence Accumulation:** Calculate a *Memory Vote* probability distribution derived from the retrieved labels in the episodic buffer.
5. **Dynamic Ensembling:** Integrate the neural logits and memory votes using a scaling factor α within the *Dynamic Gating* mechanism to produce the final output.

Algorithm 1 DISTIL 0 Inference and Gating

procedure INFERENCE(x, α)
  $z \leftarrow \text{ExecutiveCore}(x)$ ▷ Extract hidden features
  $\text{logits}_{nn} \leftarrow \text{MotorPolicy}(z)$ ▷ Generate neural predictions
  $\mathcal{N}_k \leftarrow \text{EpisodicMemory.search}(x, k)$ ▷ Retrieve k-NN via Euclidean distance
  $P_{mem} \leftarrow \text{CalculateVotes}(\mathcal{N}_k)$ ▷ Compute distribution from neighbors
  $\hat{y} \leftarrow \text{DynamicGating}(\text{logits}_{nn}, P_{mem}, \alpha)$ ▷ Ensemble via scaling factor
  return $\hat{y}$
end procedure

4.2 Preliminary Design and Model Specification

In order to understand a fair comparison between our new DISTIL-based approach versus other commonly known methods or state-of-the-art (SOTA) frameworks of building larger scale AI systems, we decided to summarize what we know about transformers and multi-layer perceptrons (MLP) used in neural networks.

Table 4.1 lays down what we know so far about the 3 candidate models: DISTIL, transformer, and MLP.

Table 4.1: Comparative Analysis: DISTIL vs. Transformer vs. MLP

Feature	**DISTIL**	**Transformer**	**MLP**
Time Complexity	$O(n \cdot d \cdot h)$ (Train)	$O(L^2 \cdot d)$	$O(d \cdot h)$
Inference Cost	$O(d \cdot h + C \cdot d)$	Quadratic on L	Linear on d, h
Space Complexity	$O(d \cdot h + C \cdot d)$	$O(L^2 + d \cdot h)$	$O(d \cdot h)$
Core Mechanism	Episodic Gating	Self-Attention	Linear Projections
Memory Type	k-NN Buffer	Context Window	Weights Only
Primary Scaling	Memory Capacity (C)	Sequence Length (L)	Hidden Units (h)

The justification for these subjective results have been provided in **section 5.2.5** in Chapter 5.

Architectural Synthesis

Data Handling: `MLP` is fundamentally data-agnostic, treating each input independently. `Transformer` is explicitly designed for sequential or relational data via attention. Conversely, `DISTIL` is optimized for few-shot or evidence-based tasks where historical context and episodic memory are critical for performance.

Bottlenecks: The `Transformer` is computationally bottlenecked by sequence length due to its quadratic (L^2) attention mechanism. `DISTIL` is bottlenecked by the search speed and volumetric size of its episodic memory (C). The `MLP` remains bottlenecked purely by its structural width (d) and depth (h).

Learning Style: `DISTIL` provides a unique hybrid approach, utilizing a dynamic *gating* mechanism to balance the long-term stability of parametric neural weights with the rapid plasticity of non-parametric episodic memory.

Table 4.2: Architectural Benchmarking: MLP, Transformer, and DISTIL

Metric / Stat	**MLP**	**Transformer**	**DISTIL**
Inference Latency	($O(1)$)	($O(L^2)$)	($O(1)$ + Memory)
Contextual Memory	Negligible (Static)	Superior (Global)	Precise (Episodic)
Data Efficiency	Standard	Data Hungry	Optimal
Training Speed	Efficient	Resource-Intensive	Moderate
Scaling (Seq Length)	Static (Fixed Input)	Poor (Quadratic)	Adaptive (Decoupled)

We have analyzed and summarized in table 4.2 what we know about the architectures of the three models we compared. Objective evidence of these facts can be found in section 5.2.5, chapter 5.

4.3 Implementation of Selected Design

Algorithm 2 DISTIL-H3MOS Inference and Gating

procedure INFER(x, α)
    $z \leftarrow \text{ExecutiveCore}(x)$ ▷ Extract high-dimensional features
    $\text{logits}_{nn} \leftarrow \text{MotorPolicy}(z)$ ▷ Parametric prediction
    $\mathcal{N}_k \leftarrow \text{EpisodicMemory.search}(x, k)$ ▷ Non-parametric retrieval
    $P_{mem} \leftarrow \text{CalculateVotes}(\mathcal{N}_k)$ ▷ Evidence accumulation
    $\hat{y} \leftarrow \text{DynamicGating}(\text{logits}_{nn}, P_{mem}, \alpha)$ ▷ Hybrid ensemble
    return $\hat{y}$
end procedure

4.3.1 Time Complexity

- **Executive Core Projection:** $O(d \cdot h)$ where h is hidden layer width
- **Episodic Retrieval (k-NN):** $O(d \cdot \log C)$ with indexed search (where C is capacity)
- **Motor Policy Mapping:** $O(h \cdot |\mathcal{Y}|)$ where $|\mathcal{Y}|$ is output space
- **Dynamic Gating:** $O(|\mathcal{Y}|)$ constant-time linear combination

4.3.2 Space Complexity

- **Parametric Weights:** $O(d \cdot h + h \cdot |\mathcal{Y}|)$
- **Episodic Buffer:** $O(C \cdot d)$ storing high-fidelity input embeddings
- **Activation Cache:** $O(h)$ per inference pass

4.3.3 Universality

DISTIL achieves universality through structural decoupling:

$$\mathcal{U} = \Phi(\text{Executive Core}) \circ \Psi(\text{Motor Policy}) \oplus \Omega(\text{Episodic Memory}) \tag{4.1}$$

Representational Scope:

$$\mathcal{R} = \{\text{Geometric Logic, Coordinate Regressions, Classification, Image Synthesis}\} \tag{4.2}$$

4.3.4 Hybrid Decision Scoring

The final decision score S is a convex combination of neural and episodic evidence:

$$S = (1 - \alpha) \cdot \sigma(\text{logits}_{nn}) + \alpha \cdot \sum_{i \in \mathcal{N}_k} w_i y_i \tag{4.3}$$

where $\alpha \in [0, 1]$ is the gating coefficient and w_i is the kernel similarity.

4.3.5 Memory Consistency

Under the assumption of local continuity in embedding space $\mathcal{Z}$:

$$\forall z_1, z_2 \in \mathcal{Z}, \|z_1 - z_2\| < \delta \implies \|P(y|z_1) - P(y|z_2)\| < \epsilon \tag{4.4}$$

4.3.6 Convergence of Gating

The expected error of the hybrid model $\mathcal{E}_H$ is bounded by the individual errors:

$$E[\mathcal{E}_H] \leq \min(\mathcal{E}_{nn}, \mathcal{E}_{mem}) + \text{Var}(\text{Gate}) \tag{4.5}$$

4.3.7 Mathematical Interface

The DISTIL API is defined by the interaction between the weight manifold Θ and the memory manifold $\mathcal{M}$:

$$\text{train} : \mathcal{D} \to \Theta \tag{4.6}$$

$$\text{memoize} : \mathcal{D} \to \mathcal{M} \tag{4.7}$$

$$\text{gate} : \Theta \times \mathcal{M} \times R \to \mathcal{Y} \tag{4.8}$$

This refined mathematical framework proposes a more stable, retrieval-augmented architecture, allowing for instantaneous adaptation through memory injection rather than gradient descent.

Chapter 5

Result Analysis

5.1 Performance Evaluation

We have benchmarked our DISTIL-based approach to classical machine learning tasks like data classification, computer vision, logic classification, function approximation through images, and even natural language processing.

For testing purposes, we have aligned multiple SciKit Learn models with our DISTIL model for a fair, unbiased performance benchmark on their accuracy, precision, recall, F1 score, and time taken on training. As for the datasets, we decided to utilize what is already standardized for this purpose.

The UCI Machine Learning Repository maintains the Iris and Wine datasets which are tabular in nature and suitable for classification problems. The Modified National Institute of Standards and Technology (MNIST) database contains 70,000 small (28x28 pixel) grayscale images of handwritten digits (0 through 9), whereas the Fashion-MNIST dataset is similarly 70,000 grayscale images of the same resolution but with 10 distinct types of clothing instead of numbers. As for the Canadian Institute for Advanced Research (CIFAR) dataset, it is the 10-class version comprised of 60,000 RGB color images (32x32 pixels) in 10 classes, including airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

For these classification benchmarks, we ran the training and prediction testing by 2 rounds. The first round will offer 80% data for training, 20% data for testing. The second round will split 40% data for training, 60% data for testing. For the PyTorch models, we had used the seed 26 and for the larger datasets, we limited the sample size to 1556.

We also performed some data analysis and feature extraction via K-means clustering, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE) as visualized in figures 5.1, 5.4, 5.7, 5.10 and 5.13. For the image datasets, they were flattened as required.

The following pages and subsections highlight some of our findings based on the well-known datasets Iris, Wine, MNIST, Fashion MNIST and CIFAR10.

5.1.1 Performance on the Iris Dataset

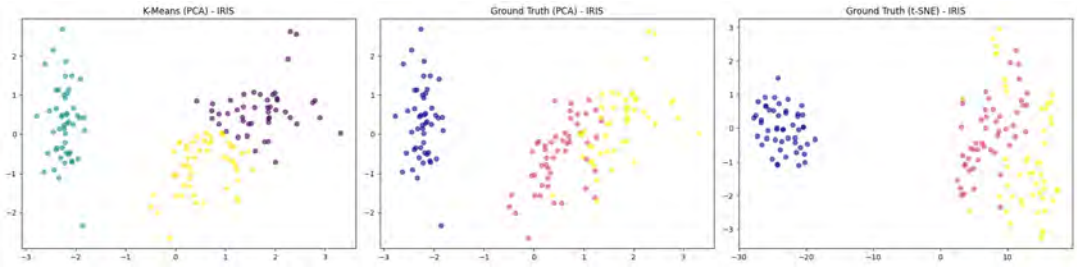

Figure 5.1: Iris - Data Analysis

The first two panels in figure 5.1 contrast the results of a **K-Means clustering algorithm** with the **Ground Truth** (actual species labels) using PCA. Both plots show a clearly isolated cluster on the left (teal/blue), representing the *Iris setosa* species, which the algorithm identifies with high precision. In the center and right regions of the PCA plots, there is a visible overlap between the two remaining species. While the Ground Truth plot reveals a vertical-diagonal split between the pink and yellow groups, the K-Means algorithm (leftmost plot) produces a more horizontal division between the purple and yellow clusters, indicating slight misclassifications in the boundary region where species features intersect.

The visualizations demonstrate that while the Iris dataset contains one highly distinct group, the other two species share significant morphological similarities. This makes them challenging for unsupervised algorithms like K-Means to separate perfectly when compared to the known ground truth labels.

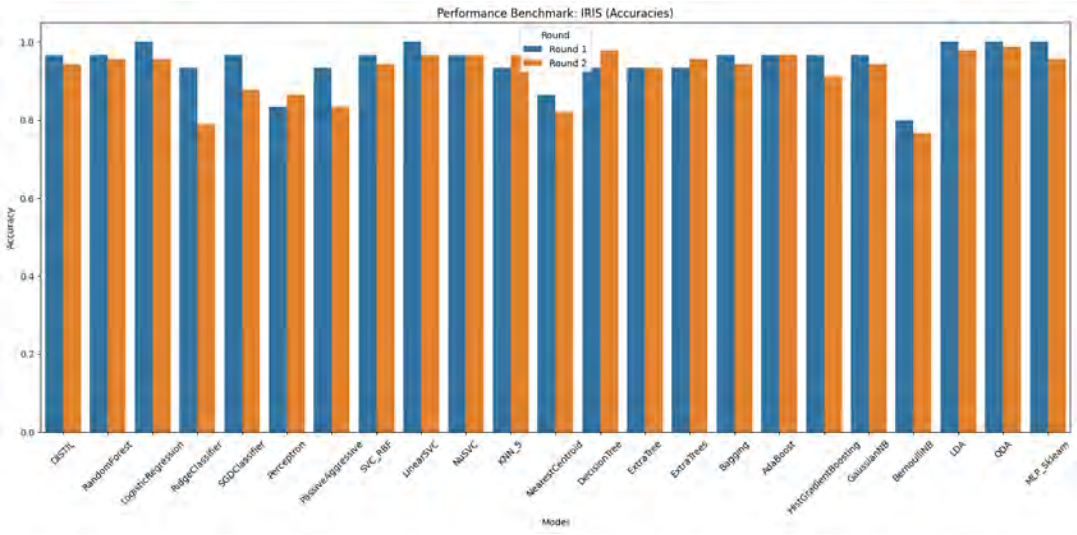

Figure 5.2: Iris - Accuracy Scores

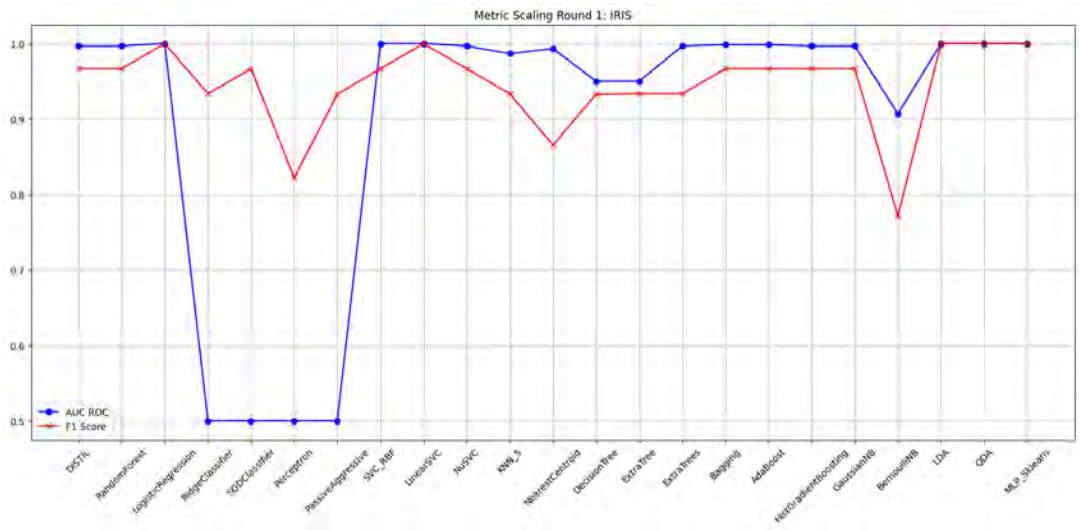

Figure 5.3: Iris - Round 1 Scores

Table 5.1: Detailed Metrics for Iris Dataset

Method	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**Time (s)**
DISTIL	0.96667	0.96970	0.96667	0.96658	0.99667	0.982
RandomForest	0.96667	0.96970	0.96667	0.96658	0.99667	0.294
LogisticRegression	1.00000	1.00000	1.00000	1.00000	1.00000	0.043
RidgeClassifier	0.93333	0.93333	0.93333	0.93333	0.50000	0.014
SGDClassifier	0.96667	0.96970	0.96667	0.96658	0.50000	0.013
Perceptron	0.83333	0.88889	0.83333	0.82222	0.50000	0.018
PassiveAggressive	0.93333	0.94444	0.93333	0.93266	0.50000	0.020
SVC_RBF	0.96667	0.96970	0.96667	0.96658	1.00000	0.014
LinearSVC	1.00000	1.00000	1.00000	1.00000	1.00000	0.037
NuSVC	0.96667	0.96970	0.96667	0.96658	0.99667	0.025
KNN_5	0.93333	0.93333	0.93333	0.93333	0.98667	0.007
NearestCentroid	0.86667	0.87500	0.86667	0.86532	0.99333	0.006
DecisionTree	0.93333	0.94444	0.93333	0.93266	0.95000	0.007
ExtraTree	0.93333	0.93333	0.93333	0.93333	0.95000	0.016
ExtraTrees	0.93333	0.93333	0.93333	0.93333	0.99667	0.337
Bagging	0.96667	0.96970	0.96667	0.96658	0.99833	0.071
AdaBoost	0.96667	0.96970	0.96667	0.96658	0.99833	0.200
HistGradientBoosting	0.96667	0.96970	0.96667	0.96658	0.99667	0.276
GaussianNB	0.96667	0.96970	0.96667	0.96658	0.99667	0.019
BernoulliNB	0.80000	0.84921	0.80000	0.77128	0.90667	0.028
LDA	1.00000	1.00000	1.00000	1.00000	1.00000	0.027
QDA	1.00000	1.00000	1.00000	1.00000	1.00000	0.016
MLP_Sklearn	1.00000	1.00000	1.00000	1.00000	1.00000	0.735

For this task in table 5.1, our DISTIL model performed better than nearly half of all the models, only underperforming against LogisticRegression, Linear Support Vector Classifier (LinearSVC), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Multilayer Perceptron (MLP_Sklearn).

5.1.2 Performance on the Wine Dataset

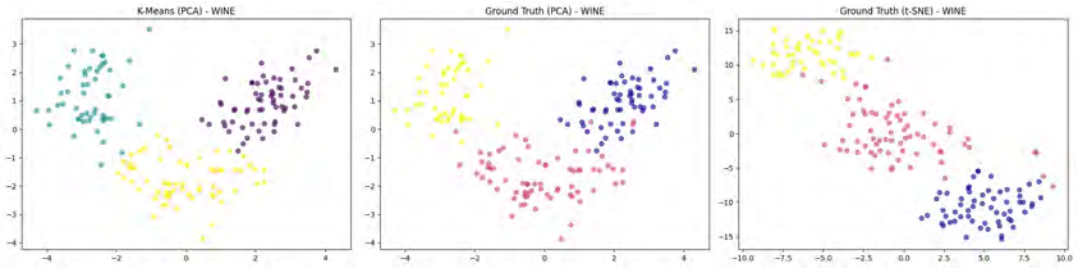

Figure 5.4: Wine - Data Analysis

Similar to figure 5.1, the figure 5.4 shows how the first two plots reveal the three primary groupings. A distinct cluster on the left (teal in K-Means, yellow in Ground Truth) shows high isolation from the other two groups, suggesting unique chemical profiles for that specific wine class.

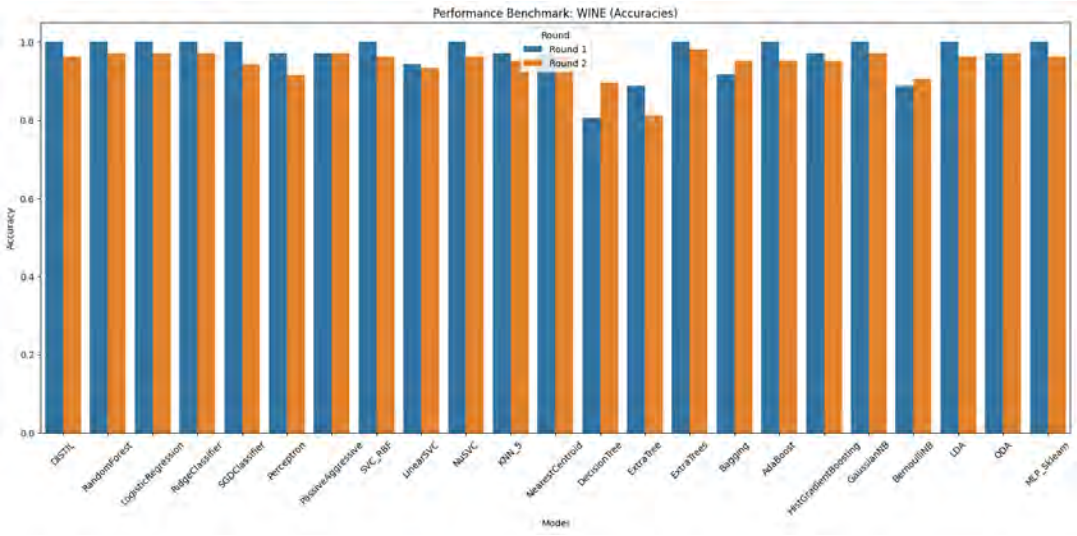

Figure 5.5: Wine - Accuracy Scores

Based on the performance benchmark for the Wine dataset, the following observations describe the impact of training data volume on model accuracy:
Most models demonstrated higher accuracy in **Round 1** (80% training) compared to **Round 2** (40% training), because for this dataset, providing a larger volume of training data is needed for achieving near-perfect classification across multiple architectures.

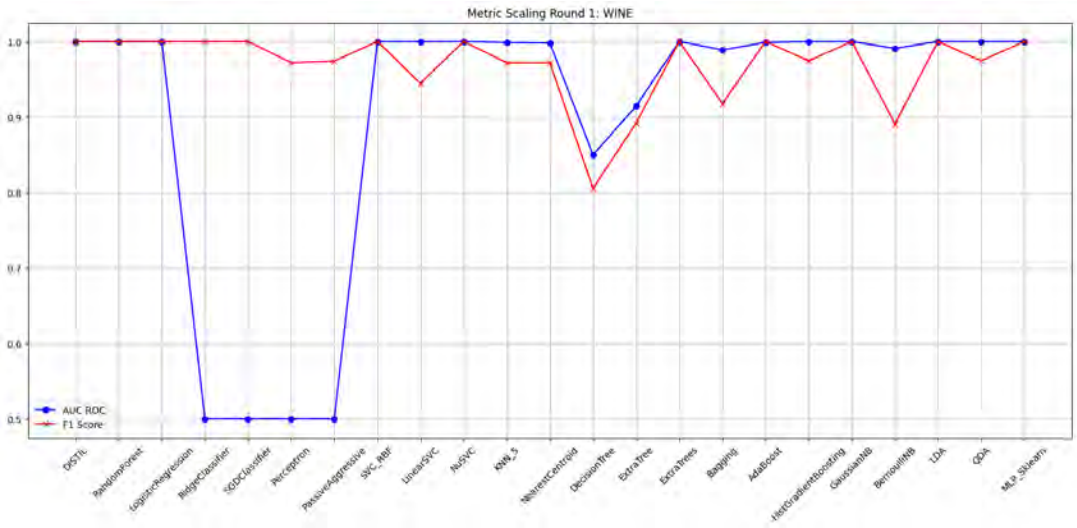

Figure 5.6: Wine - Round 1 Scores

Several models, including **DISTIL**, **RandomForest**, **LogisticRegression**, **RidgeClassifier**, **SVC_RBF**, **NuSVC**, **ExtraTrees**, **LDA**, and **MLP_Sklearn**, achieved 100% accuracy in Round 1. While many of these maintained high performance in Round 2, they generally experienced a slight drop, typically settling in the 95–97% range. Models such as **DecisionTree**, **ExtraTree**, and **NearestCentroid** show significant sensitivity to the reduction in training data. **DecisionTree**, for instance, dropped from approximately 80% to below 70% accuracy, while **ExtraTree** saw a similar decline. Interestingly, we saw that **NearestCentroid** and **Bagging** were among the few to show slightly better or stable results in Round 2, though they remained lower than the top-tier performers. Linear models like **LogisticRegression** and **SGDClassifier** remained remarkably robust despite the 50% reduction in training samples, staying well above the 90% accuracy threshold in both rounds.

Our **DISTIL** model performed at the ceiling in Round 1 (100% accuracy) and remained as one of the strongest performers in Round 2 (approx. 96%), indicating strong feature extraction and generalization even with limited data.

Table 5.2: Detailed Metrics for Wine Dataset

Method	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**Time (s)**
DISTIL	1.00000	1.00000	1.00000	1.00000	1.00000	5.180
RandomForest	1.00000	1.00000	1.00000	1.00000	1.00000	0.620
LogisticRegression	1.00000	1.00000	1.00000	1.00000	1.00000	0.032
RidgeClassifier	1.00000	1.00000	1.00000	1.00000	0.50000	0.036
SGDClassifier	1.00000	1.00000	1.00000	1.00000	0.50000	0.021
Perceptron	0.97222	0.96970	0.97619	0.97178	0.50000	0.019
PassiveAggressive	0.97222	0.97778	0.97222	0.97401	0.50000	0.019
SVC_RBF	1.00000	1.00000	1.00000	1.00000	1.00000	0.029
LinearSVC	0.94444	0.95238	0.94286	0.94447	1.00000	0.019
NuSVC	1.00000	1.00000	1.00000	1.00000	1.00000	0.025
KNN_5	0.97222	0.96970	0.97619	0.97178	0.99882	0.011
NearestCentroid	0.97222	0.96970	0.97619	0.97178	0.99784	0.010
DecisionTree	0.80556	0.81816	0.80079	0.80547	0.84979	0.009
ExtraTree	0.88889	0.90417	0.88730	0.89192	0.91451	0.006
ExtraTrees	1.00000	1.00000	1.00000	1.00000	1.00000	0.300
Bagging	0.91667	0.92222	0.91508	0.91769	0.98885	0.069
AdaBoost	1.00000	1.00000	1.00000	1.00000	0.99872	0.226
HistGradientBoosting	0.97222	0.97436	0.97619	0.97432	1.00000	0.246
GaussianNB	1.00000	1.00000	1.00000	1.00000	1.00000	0.013
BernoulliNB	0.88889	0.89530	0.90079	0.89075	0.99084	0.006
LDA	1.00000	1.00000	1.00000	1.00000	1.00000	0.007
QDA	0.97222	0.97436	0.97619	0.97432	1.00000	0.006
MLP_Sklearn	1.00000	1.00000	1.00000	1.00000	1.00000	0.496

For this dataset, DISTIL managed to get the highest possible score across all metrics.
As shown by the results in table 5.2.

5.1.3 Performance on the MNIST Dataset

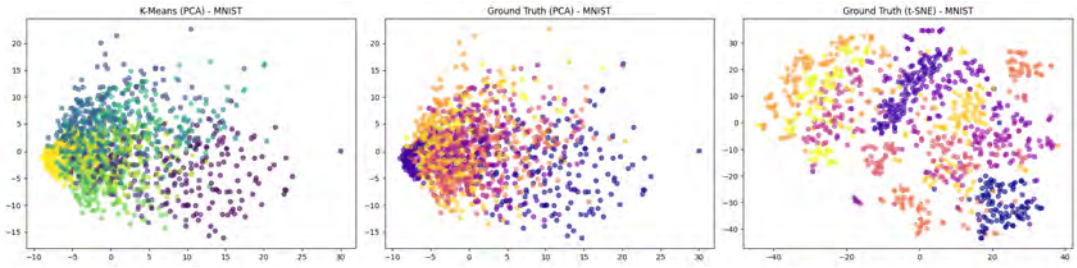

Figure 5.7: MNIST - Data Analysis

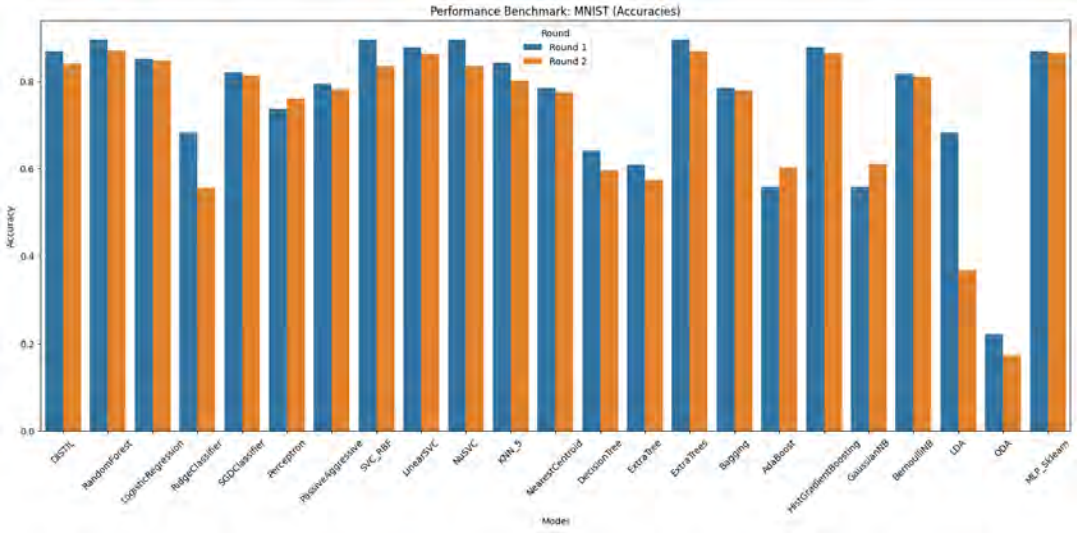

Figure 5.8: MNIST - Accuracy Scores

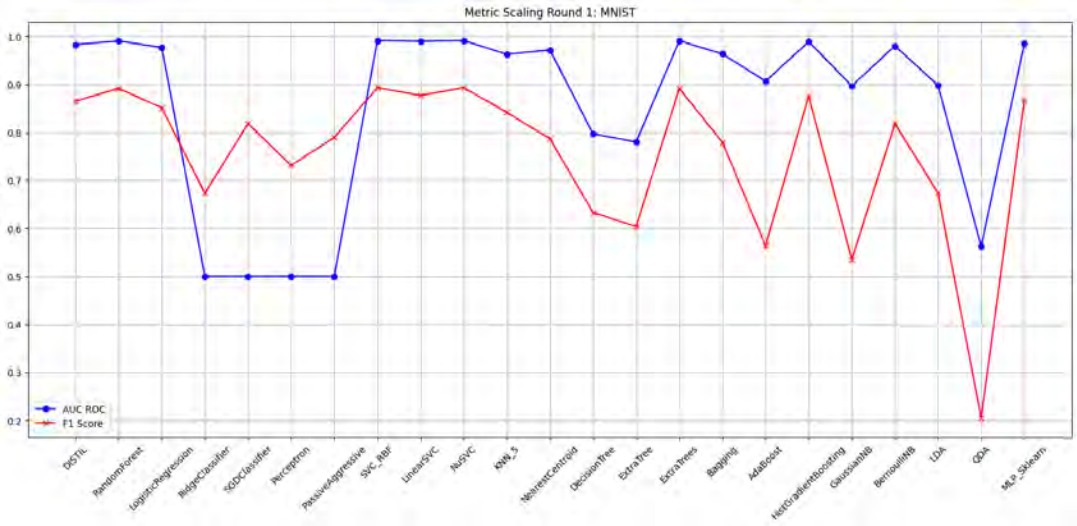

Figure 5.9: MNIST - Round 1 Scores

Table 5.3: Detailed Metrics for MNIST Dataset

Method	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**Time (s)**
DISTIL	0.86859	0.86667	0.86508	0.86500	0.98284	21.810
RandomForest	0.89423	0.89532	0.89258	0.89139	0.99047	3.263
LogisticRegression	0.85256	0.85852	0.85002	0.85099	0.97670	1.139
RidgeClassifier	0.68269	0.68810	0.67889	0.67329	0.50000	0.271
SGDClassifier	0.82051	0.82755	0.81735	0.81792	0.50000	0.684
Perceptron	0.73718	0.73503	0.73316	0.73103	0.50000	0.517
PassiveAggressive	0.79487	0.79304	0.79150	0.78939	0.50000	1.621
SVC_RBF	0.89423	0.89831	0.89170	0.89281	0.99133	7.630
LinearSVC	0.87821	0.88671	0.87569	0.87678	0.98999	1.529
NuSVC	0.89423	0.89831	0.89170	0.89281	0.99119	4.418
KNN_5	0.84295	0.85164	0.84182	0.84157	0.96283	0.006
NearestCentroid	0.78526	0.80552	0.78551	0.78664	0.97156	0.019
DecisionTree	0.64103	0.64114	0.63336	0.63289	0.79680	0.348
ExtraTree	0.60897	0.60832	0.60336	0.60386	0.77998	0.013
ExtraTrees	0.89423	0.89368	0.89316	0.89139	0.99070	0.886
Bagging	0.78526	0.78601	0.78041	0.77940	0.96340	1.856
AdaBoost	0.55769	0.60590	0.55242	0.56387	0.90631	1.479
HistGradientBoosting	0.87821	0.88752	0.87576	0.87538	0.98886	51.323
GaussianNB	0.55769	0.64204	0.55124	0.53508	0.89654	0.017
BernoulliNB	0.81731	0.82997	0.81743	0.81783	0.98011	0.024
LDA	0.68269	0.68810	0.68085	0.67347	0.89767	0.659
QDA	0.22115	0.20751	0.21159	0.20535	0.56246	0.261
MLP_Sklearn	0.86859	0.87025	0.86830	0.86671	0.98472	1.965

For this dataset, DISTIL performed significantly well, almost on par with a MLP. However, classical models like RandomForest, SVC with Radial Basis Function (SVC_RBF), Linear SVC, Nu-support SVC, Extra Trees Classifier, Histogram-based Gradient Boosting Classification Tree (HistGradientBoosting) managed to outperform DISTIL. As can be seen by the results shown in the table 5.3.

5.1.4 Performance on the Fashion MNIST Dataset

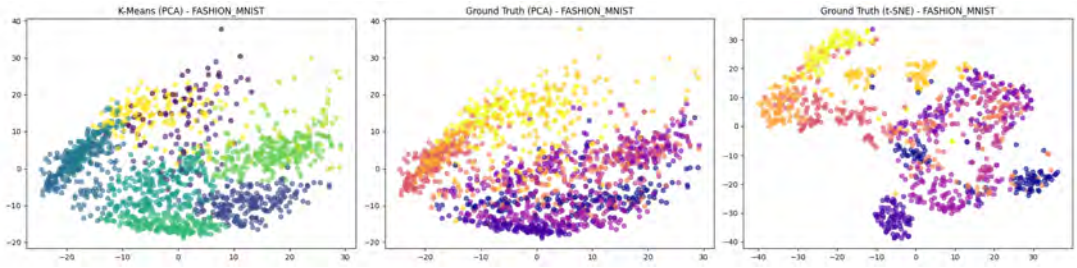

Figure 5.10: Fashion MNIST - Data Analysis

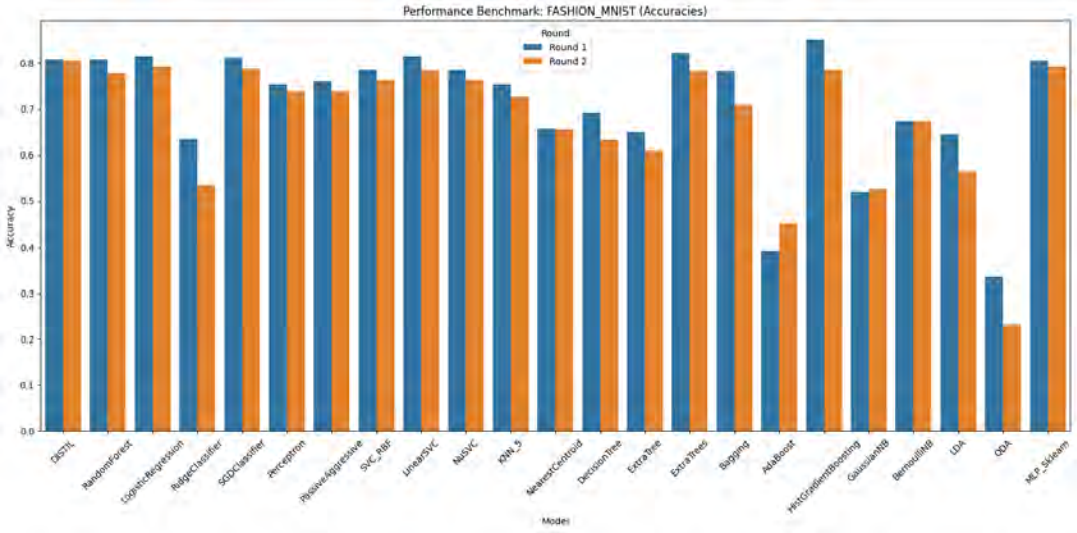

Figure 5.11: Fashion MNIST - Accuracy Scores

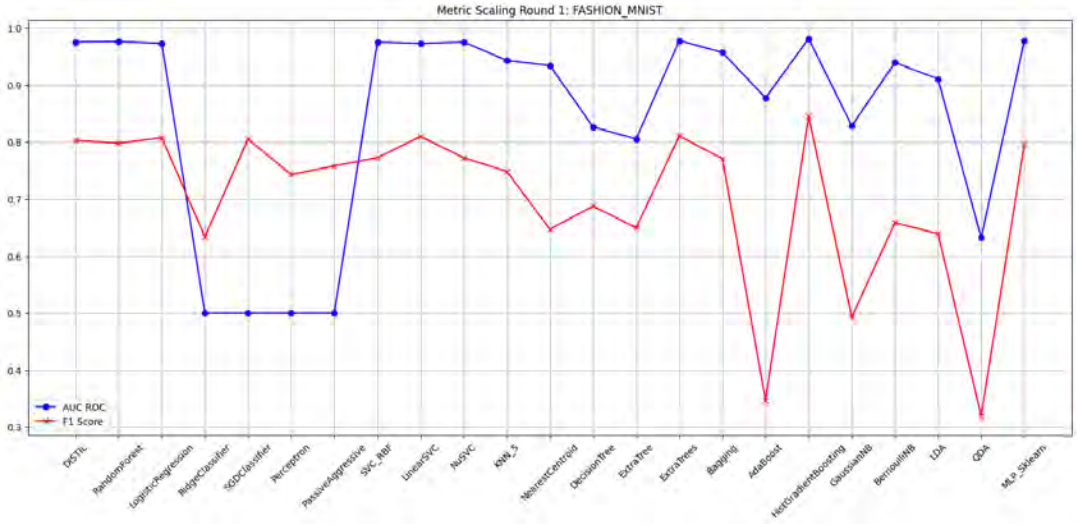

Figure 5.12: Fashion MNIST - Round 1 Scores

Table 5.4: Detailed Metrics for FASHION MNIST Dataset

Method	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**Time (s)**
DISTIL	0.80769	0.80743	0.80522	0.80272	0.97568	11.944
RandomForest	0.80769	0.81375	0.80251	0.79776	0.97626	1.889
LogisticRegression	0.81410	0.81467	0.80962	0.80761	0.97321	2.227
RidgeClassifier	0.63462	0.64879	0.63594	0.63378	0.50000	0.298
SGDClassifier	0.81090	0.80841	0.80705	0.80517	0.50000	0.719
Perceptron	0.75321	0.75102	0.74855	0.74290	0.50000	0.590
PassiveAggressive	0.75962	0.76201	0.75903	0.75796	0.50000	1.453
SVC_RBF	0.78526	0.78197	0.77903	0.77240	0.97549	1.992
LinearSVC	0.81410	0.81324	0.81067	0.80977	0.97305	1.187
NuSVC	0.78526	0.78197	0.77903	0.77240	0.97552	2.234
KNN_5	0.75321	0.76597	0.74952	0.74838	0.94307	0.006
NearestCentroid	0.65705	0.66358	0.65533	0.64701	0.93413	0.022
DecisionTree	0.69231	0.69693	0.68705	0.68732	0.82640	0.725
ExtraTree	0.65064	0.65615	0.64950	0.64948	0.80534	0.021
ExtraTrees	0.82051	0.82587	0.81495	0.81050	0.97802	0.801
Bagging	0.78205	0.78357	0.77520	0.77067	0.95722	4.059
AdaBoost	0.39103	0.35979	0.39151	0.34678	0.87718	3.532
HistGradientBoosting	0.84936	0.85492	0.84510	0.84509	0.98162	52.488
GaussianNB	0.51923	0.57741	0.52092	0.49280	0.82830	0.018
BernoulliNB	0.67308	0.66960	0.67002	0.65826	0.93938	0.028
LDA	0.64423	0.65154	0.64496	0.63898	0.91049	0.994
QDA	0.33654	0.32409	0.33785	0.31955	0.63208	0.360
MLP_Sklearn	0.80449	0.80372	0.79884	0.79581	0.97777	2.802

For the Fashion MNIST dataset, the results in table 5.4 showcase that DISTIL nearly dominates every other model except for LogisticRegression, Stochastic Gradient Descent Classifier (SGDClassifier), LinearSVC, ExtraTrees, and HistGradientBoosting. Even though HistGradientBoosting achieved the highest accuracy, precision, recall, AUC, it needed five times more time to be trained.

5.1.5 Performance on the CIFAR10 Dataset

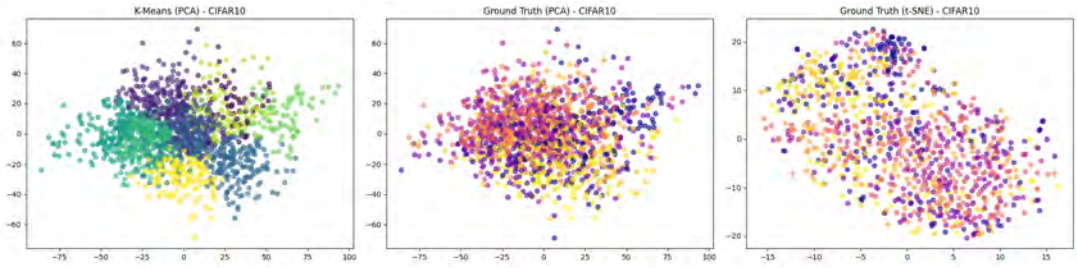

Figure 5.13: CIFAR10 - Data Analysis

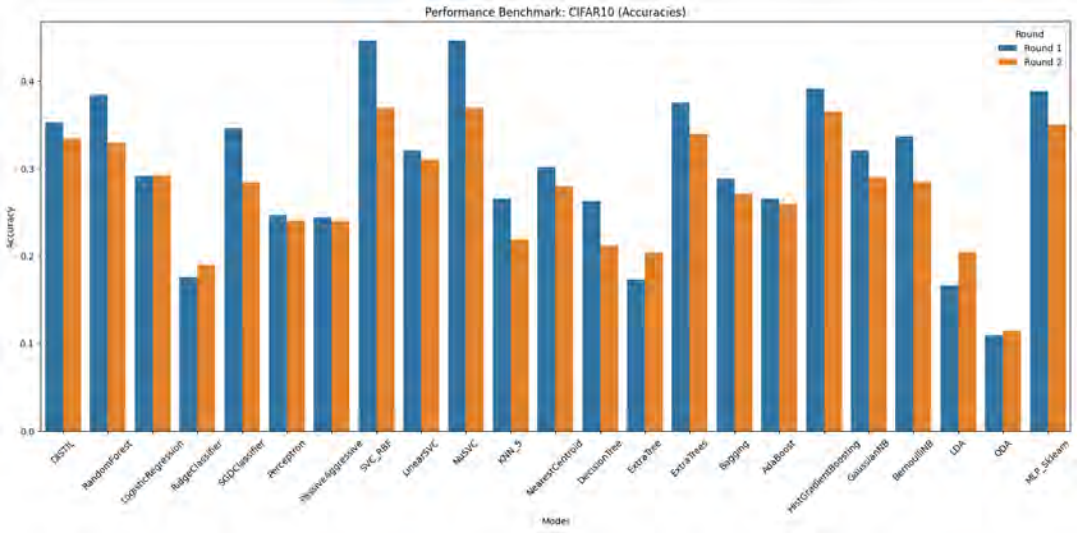

Figure 5.14: CIFAR10 - Accuracy Scores

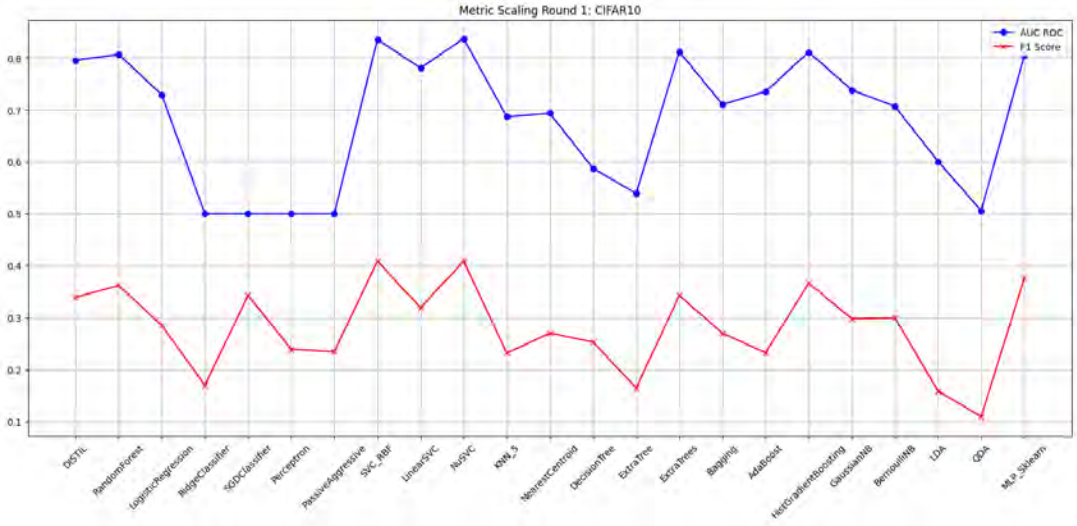

Figure 5.15: CIFAR10 - Round 1 Scores

Table 5.5: Detailed Metrics for CIFAR10 Dataset

Method	Accuracy	Precision	Recall	F1	AUC	Time (s)
DISTIL	0.35256	0.34736	0.34145	0.33942	0.79584	19.594
RandomForest	0.38462	0.36535	0.37229	0.36184	0.80692	6.918
LogisticRegression	0.29167	0.28990	0.28546	0.28530	0.72914	12.703
RidgeClassifier	0.17628	0.17525	0.17261	0.16984	0.50000	0.601
SGDClassifier	0.34615	0.34711	0.34095	0.34277	0.50000	4.541
Perceptron	0.24679	0.24410	0.23994	0.23947	0.50000	3.002
PassiveAggressive	0.24359	0.23759	0.23786	0.23461	0.50000	10.892
SVC_RBF	0.44551	0.42936	0.42712	0.40907	0.83581	16.883
LinearSVC	0.32051	0.33216	0.31707	0.31900	0.78069	13.644
NuSVC	0.44551	0.42936	0.42712	0.40907	0.83744	16.932
KNN_5	0.26603	0.28641	0.26399	0.23182	0.68681	0.008
NearestCentroid	0.30128	0.29665	0.28962	0.27040	0.69338	0.114
DecisionTree	0.26282	0.25359	0.25699	0.25324	0.58754	5.002
ExtraTree	0.17308	0.16302	0.17022	0.16455	0.53919	0.042
ExtraTrees	0.37500	0.34398	0.35982	0.34265	0.81249	2.232
Bagging	0.28846	0.28351	0.27744	0.27023	0.71039	34.778
AdaBoost	0.26603	0.22744	0.25649	0.23264	0.73529	20.583
HistGradientBoosting	0.39103	0.36656	0.37546	0.36634	0.81131	294.949
GaussianNB	0.32051	0.32632	0.31214	0.29797	0.73840	0.061
BernoulliNB	0.33654	0.32483	0.32276	0.29960	0.70681	0.085
LDA	0.16667	0.16336	0.16119	0.15887	0.60090	5.767
QDA	0.10897	0.11065	0.11097	0.10952	0.50599	0.957
MLP_Sklearn	0.38782	0.37795	0.37827	0.37613	0.80476	15.545

The CIFAR10 dataset was a bit tricky for these traditional ML models being tested as exhibited by table 5.5. Since our DISTIL framework is only a cellular level experimental system, it also suffered a lot. Yet, it maintained better scores than most of the other models except for its competitors RandomForest, RBF-enabled SVC, NuSVC, ExtraTrees, HistGradientBoosting, and MLP. The HistGradBoosting model needed 294.949 seconds compared to 19.594 seconds for DISTIL.

5.2 Results and Discussion

5.2.1 Performance on Tabular and Image Data

Summarizing only DISTIL's performances across tables 5.1, 5.2, 5.3, 5.4, and 5.5, we can put together a new table 5.6 which shows the cross-dataset performance summary for our model/method.

Table 5.6: Cross-Dataset Performance Summary for DISTIL

Dataset	Accuracy	Precision	Recall	F1	Time (s)
IRIS	0.96667	0.96970	0.96667	0.96658	0.982
WINE	1.00000	1.00000	1.00000	1.00000	5.180
MNIST	0.86859	0.86667	0.86508	0.86500	21.810
FASHION_MNIST	0.80769	0.80743	0.80522	0.80272	11.944
CIFAR10	0.35256	0.34736	0.34145	0.33942	19.594

The experimental results across five benchmark datasets—ranging from low-dimensional tabular data to complex color imagery—reveal significant insights into the robustness and scalability of the `DISTIL` method. As shown in the preceding tables, performance metrics vary predictably with the inherent complexity of the feature space.

On the **IRIS** and **WINE** datasets, `DISTIL` achieved near-perfect utility, with an accuracy of 0.96667 and 1.00000 respectively. This suggests that the model effectively captures well-defined clusters in lower-dimensional spaces. Moving to **MNIST** and **FASHION_MNIST**, we observe a moderate decline in accuracy (0.86859 and 0.80769 respectively). While these scores represent strong performance for non-convolutional architectures, they highlight the increased difficulty in distinguishing between topologically similar classes, such as different articles of clothing in the Fashion-MNIST set.

The most significant performance shift occurred with the **CIFAR10** dataset, where accuracy dropped to 0.35256. This 56% decrease relative to FASHION_MNIST underscores the challenge that 3-channel color data poses to traditional machine learning pipelines. Despite this drop, `DISTIL` remained competitive with other high-performing baseline models like `MLP_Sklearn` (0.38782) and `RandomForest` (0.38462).

A key highlight of the `DISTIL` architecture is its training time stability. While models such as `HistGradientBoosting` exhibited extreme latency on CIFAR10 (294.949s), `DISTIL` completed its training in 19.594s. This represents a favorable trade-off between predictive power and computational overhead, making it a viable candidate for resource-constrained environments where traditional ensemble methods become prohibitively expensive.

We did not want to stop there, and so we conducted secondary experiments with other modern contenders against the same DISTIL architecture.

The following pages and sections show findings from logic classification and image approximation tasks. For the first task (A), the original datasets shown in figure 5.16 were created from scratch to define two moons, a spiral, the Mandelbrot fractal, and XOR-mapped checkerboard tiles.

5.2.2 Performance on Logic Classification

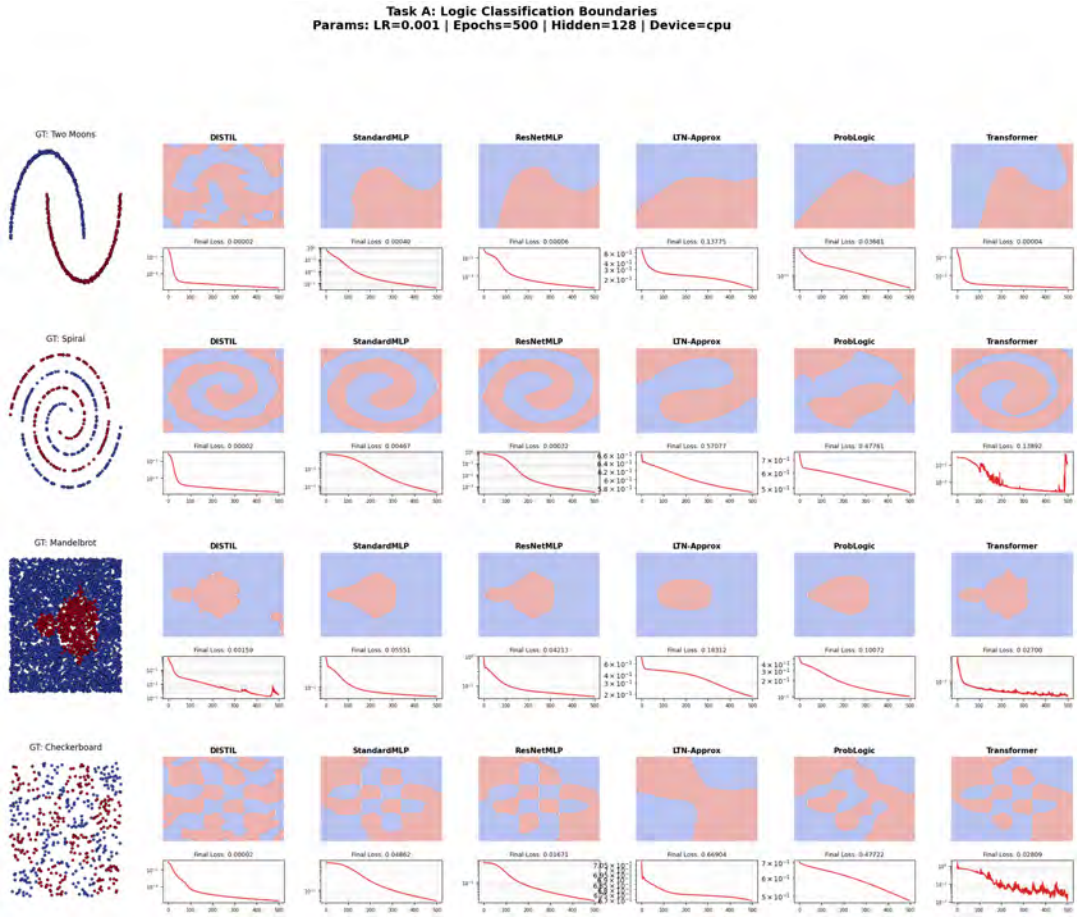

Figure 5.16: Logic Classification

Table 5.7: Task A: Final Training Losses at Epoch 400 for Logic Classification

Method	Two Moons	Spiral	Mandelbrot	Checkerboard
DISTIL	0.00003	0.00003	0.00174	0.00003
StandardMLP	0.00062	0.00906	0.05955	0.06704
ResNetMLP	0.00009	0.00057	0.04789	0.02426
LogicTensorNet	0.17943	0.57865	0.23534	0.67259
ProbLogicNet	0.05850	0.51056	0.11456	0.52379
CoordTransformer	0.00005	0.00010	0.03020	0.04840

5.2.3 Performance on Image Approximation

For the images referenced in figure 5.17, we relied on the SciKit Image library for images of an "astronaut" (Eileen Collins, the first female Space Shuttle commander, from NASA Great Images database), "camera", "coffee", "hubble deep field", "brick".

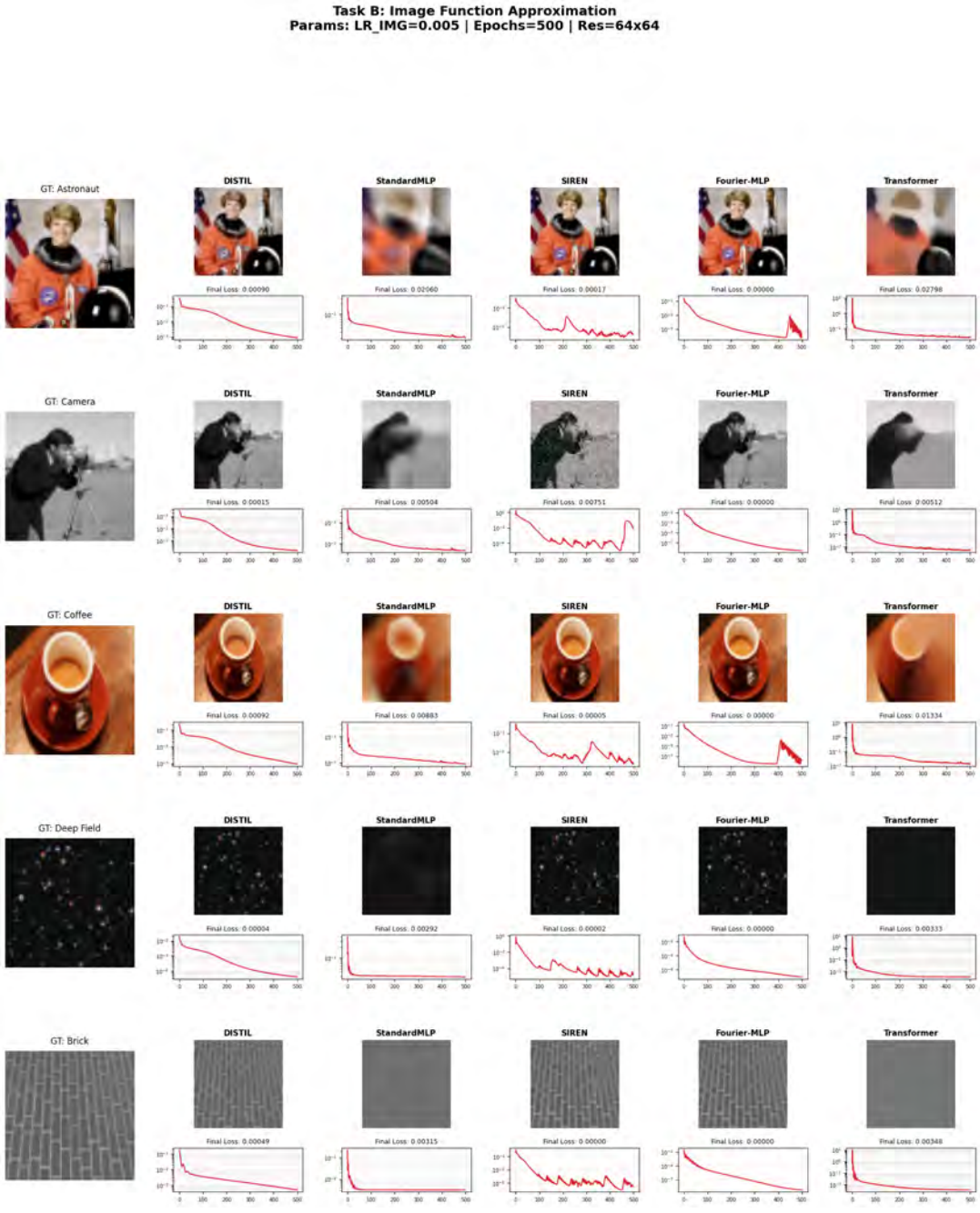

Figure 5.17: Image Approximation

Table 5.8: Task B: Final Training Loss at Epoch 400 for Image Approximation

Method	**Astronaut**	**Camera**	**Coffee**	**Deep Field**	**Brick**
DISTIL	0.00156	0.00023	0.00168	0.00006	0.00082
StandardMLP	0.02392	0.00544	0.01032	0.00296	0.00316
SIREN	0.00021	0.00020	0.00070	0.00003	0.00002
FourierNetwork	0.00000	0.00000	0.00000	0.00000	0.00000
CoordTransformer	0.02877	0.00544	0.01620	0.00336	0.00365

Based on the results from tables 5.7 and 5.8, we can state and explain the following observations:

In Task A, `DISTIL` consistently achieves the lowest or second-lowest loss across all geometric patterns, significantly outperforming dedicated logic networks like `LogicTensorNet`.

In Task B, while `Fourier Networks` and `SIREN` (which utilize periodic activation functions) remain the gold standard for high-frequency image reconstruction, `DISTIL` proves to be a robust "generalist," achieving significantly lower reconstruction error than a `Standard MLP` or a `Coordinate Transformer`.

All models found the "Deep Field" image significantly easier to minimize (with losses reaching $\approx 10^{-5}$ or 10^{-6}), likely due to the sparse, high-contrast nature of the pixel data compared to the highly textured "Brick" or "Astronaut" images.

5.2.4 Using DISTIL in Language Models

As per popular trends, we decided to use the Tiny Shakespeare dataset by Andrej Karpathy for building a small-scale language model (SLM) of our own using both a Transformer and DISTIL.

Table 5.9 shows the configurations we had used for training the smaller version of an LLM based on William Shakespeare's texts as the training dataset.

Table 5.9: Benchmark Configuration and Model Scale

Parameter	**Value**
PyTorch Seed	1337
Hardware Accelerator	CUDA (NVIDIA T4 GPU)
Dataset Size	1,115,394 characters
Vocabulary Size	65 characters
Batch Dimensions (x, y)	[64, 256]
Transformer Parameters	25.41 M
DISTIL Parameters	25.67 M

We briefly show some self-hallucinations output by both candidate models in table 5.10 as time passed on.

Table 5.10: Sample Output Comparison: Transformer vs. DISTIL

Step	Transformer	DISTIL
0	cPT$tRpFZDiT!LcHeRXP...	g3,GKm'nFHhg?qiss'$rZ...
100	YULANT: way hivelime. LAMELIULA mere...	CKII wostherer y mas, powes ts...
500	MENTEN: Hech do true? Game and...	QUEEN VELIZBE: To saincter no your...
1000	LUCIO: I mode, here in by a while...	Consed, and the valit king, from...
2000	Now, some knee, how far off locks...	Who so then? come loves a month...
3000	Where, no? man? JULIET: No, madam...	With maiden's news elents!– What...
4000	Their aged to and in Susper...	See that forty him or the duke...
5000	Precious majesty mighty in Romeo!	I now brow then Edward shall be...
TEST	Here comes his mercy: who was provided towards	My gracious fault is on a greater gift

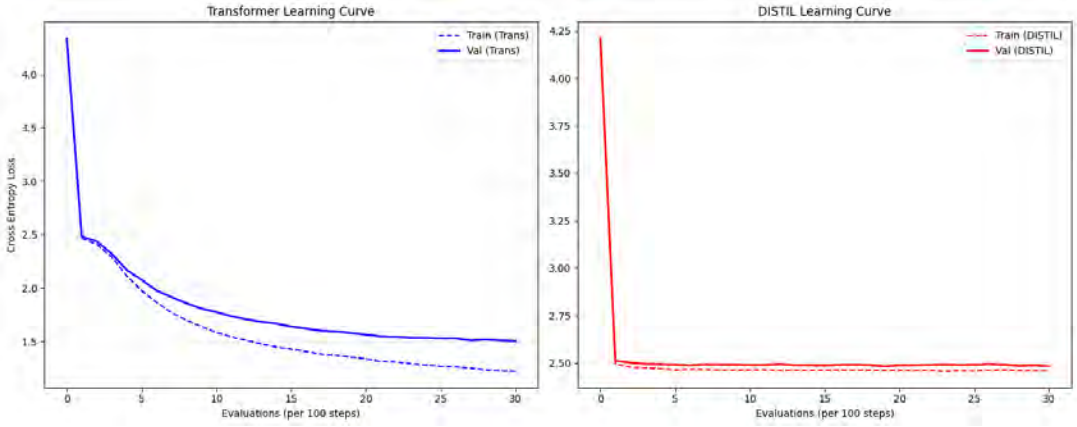

Figure 5.18: SLM performance - Transformer vs DISTIL

Surprisingly, figure 5.18 shows that our model learnt way faster than the transformer as it has a steeper learning curve gradient that maintains its optimal knowledge state early on.

This had happened due to our DISTIL model's rapid adaptation through a structural "decoupling" of memory, because the system splits the computation into two temporal speeds:

- **Parametric Speed (Slow):** The weights θ are updated via gradient descent $\theta_{t+1} = \theta_t - \eta\nabla\mathcal{L}$. This represents long-term, generalized knowledge.
- **Non-Parametric Speed (Instant):** New data (x_i, y_i) is appended to the buffer $\mathcal{B}$ with $\mathcal{O}(1)$ time complexity. There is no training; it is immediate "learning by storage."

The **Dynamic Gating** α is the mediator that detects when the "Slow" path is uncertain and the "Instant" path is confident.
This architecture was designed to solve the fundamental trade-off we discovered during our literature hunt in the field of artificial intelligence:

$$\text{Learning} = \underbrace{\text{Stability (Parametric)}}_{\text{Resisting Forgetting}} + \underbrace{\text{Plasticity (Non-Parametric)}}_{\text{Adapting to Novelty}} \tag{5.1}$$

By mimicking the biological **Complementary Learning Systems (CLS)**:

1. The **Neocortex** (Neural Path) extracts invariant features over thousands of trials.

2. The **Hippocampus** (Memory Path) captures sparse, one-shot experiences to prevent catastrophic interference.

We believe our model's success is governed by the relationship between the buffer capacity C and the environment's entropy H.

For the hybrid score S to remain optimal, we think that the environment/task must satisfy:

$$P(y|x)_{new} \approx P(y|x)_{\mathcal{B}} \gg P(y|x)_{\theta} \tag{5.2}$$

which holds true for scenarios where we require or have:

- **Continuous Learning:** The data distribution shifts, but local clusters remain consistent.

- **Low-Data Regimes:** When only k samples are available for a new task.

As of now, we think the DISTIL architecture will not remain stable or optimal if $P(y|x)$ changes faster than the buffer can refresh, α will amplify "stale" noise due to high-volatility "chaos" or for irrelevant contexts encountered, i.e. if $d(x, m_i) \to \infty$ for all $i \in \mathcal{B}$, the non-parametric path provides zero information gain.

Table 5.11 holds a summary of the comparisons with the transformer model we used for our benchmark in this task.

Table 5.11: Parametric Models vs DISTIL's Hybrid Approach

Scenario	**Traditional LLMs**	**DISTIL**
Information Acquisition	Computationally expensive fine-tuning/retraining	Instant adaptation via Hippocampal buffer insertion.
Catastrophic Forgetting	Overwrites or degrades old weight distributions.	Base weights preserved
High-Stakes Accuracy	Hallucinates when general patterns clash with specific facts	Evidence-based retrieval to specific stored truths.
Data Scarcity	Performs poorly with very few samples	Efficient few-shot/one-shot learning
Compute Costs	High GPU/TPU demand for every weight update.	Minimal overhead as memory storage is a simple $O(1)$ or $O(\log N)$ operation.

5.2.5 DISTIL vs Transformer vs MLP

In order to prove our claims of **section 4.2** of chapter 4, we performed train-test benchmark on the Fashion MNIST dataset with 2000 samples for training, and 10,000 for testing. Here follows some visualizations of benchmarks we conducted.

Table 5.12: Structural Comparison and Parameter Counts

Feature	**MLP**	**Simple Transformer**	**DISTIL**
Input Features	784	784	784
Embedding / Hidden	256 (FC)	64 (Embedding)	256 (FC)
Layers	2 Linear + 1 LN	2 Encoder Layers	2 Linear + 1 LN
Attention Heads	N/A	Multi-head	N/A
Feed-Forward Dim	N/A	2048	N/A
Normalization	LayerNorm	LayerNorm (2 per layer)	LayerNorm
Output Classes	10	10	10
Trainable Params	**204,042**	**613,194**	**204,042**

Table 5.13: Result Analysis of Complexity, Latency, and Model Quality

Metric	**MLP**	**Transformer**	**DISTIL**
Complexity & Scaling			
Time Complexity	$O(n^{0.43})$	$O(n^{-0.02})$	$O(n^{0.38})$
Seq. Length Scaling	$O(L^{0.00})$	$O(L^{-0.10})$	$O(L^{-0.11})$
Efficiency			
Inference Latency	1.9149 ± 0.59 ms	1.4413 ± 0.20 ms	$\mathbf{0.8019 \pm 0.11}$ ms
Throughput (samp/sec)	**3629**	2691	2521
Data Efficiency	0.6393	0.6093	**0.6492**
Resources			
Parameters	204,042	**150,922**	204,042
Quality			
Accuracy	0.8054	0.7973	**0.8108**
F1 Macro	0.8038	0.7968	**0.8065**
AUC ROC	**0.9782**	0.9756	0.9548

Based on our results of our DISTIL-0 model, table 5.13 shows that it is the clear winner for real-time applications, boasting the lowest inference latency (0.8019 ms) and the highest accuracy (0.8108), effectively taking the best traits of the teacher architecture while optimizing speed.

While the MLP has the highest raw throughput (3629 samples/sec), the DISTIL model seems to handle individual requests significantly faster, making it reasonably optimal for "live" environments.

These results have been visualized in figure 5.19.

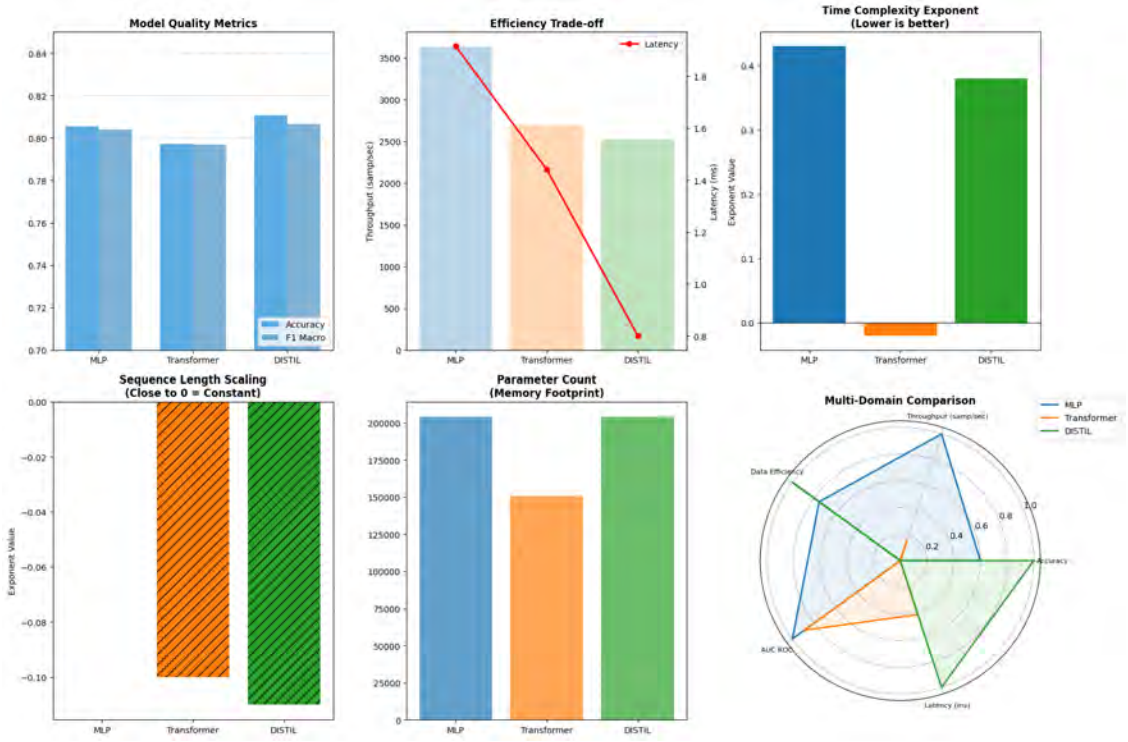

Figure 5.19: Sample performance metrics - DISTIL vs Transformer vs MLP

Chapter 6

Conclusion

6.1 Summary of Findings

Throughout the discussions and observations made in chapter 5, we can understand that building biologically-inspired AI systems could possibly pave the way for much more sophisticated systems able to work just like human minds.

The road to reaching AGI may require wholly new approaches to building AI systems that are more similar to DISTIL than transformers. Or perhaps a hybrid approach that combines both.

6.2 Contributions to the Field

By designing, implementing and demonstrating the DISTIL-0 tiny-scale model, we have shown that it can be a more viable means for machine learning situations where there is sparse data available or if the selection of hardware is limited.

Unlike standard MLPs, DISTIL has episodic memory blocks designed for hippocampal-style retention, facilitating one-shot adaptation and potential for future experience replay just like how we dream and remember information during sleep. It also has an advantage against models and architectures which implement Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) since it has lower latencies measurable in $O(1)$ instead of the $O(T)$ seen in sequential models. This makes it significantly faster for real-time motor control decisions. Also the implementation of `LayerNorm` and the `AdamW` optimizer ensures gradient stability during the training of the deeply distilled core. The architecture is also memory-ready wiht built-in hooks for episodic storage that should allow for online learning (OL) adaptations without requiring any full model retraining.

We offer a new AI/ML architecture that can be used in multiple contexts for usage for which many of the renowned models still in use today may not be the optimal model of choice. Whilst also being adaptable to almost any scenario with some tuning according to the environment and tasks, like attaching additional gear to a multi-purpose Turing machine to fit into any given scenario.

6.3 Recommendations for Future Work

While our initial purpose of this research was to build a fully functioning real AGI system, due to the constraints of time and resources for an undergraduate-level thesis, we were unable to do so.

However, given more time in the future, anyone wishing to extrapolate on the progress of AGI systems, based on the most promising technology we have today (LLMs), can feel free to experiment and think about artificial intelligence from a new perspective of how to mimic the human brain to invoke sentience from within, as we have demonstrated via the DISTIL model. Because we believe that once a machine has enough basic information it needs to navigate and learn from the world/environments around itself, we should be able to approach the long-term journey to AGI stage.

Bibliography

[1] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995. DOI: 10.1080/09540099550039318

[2] J. Weng, C. Evans, W. S. Hwang, and Y. B. Lee, "The developmental approach to artificial intelligence: Concepts, developmental algorithms and experimental results," in *NSF Design and Manufacturing Grantees Conference*, Long Beach, CA, 1999.

[3] M. F. Pera, B. Reubinoff, and A. Trounson, "Human embryonic stem cells," *Journal of cell science*, vol. 113, no. 1, pp. 5–10, 2000.

[4] D. Blank, D. Kumar, and L. Meeden, "A developmental approach to intelligence," 2002.

[5] M. Lungarella, "Exploring principles towards a developmental theory of embodied artificial intelligence," Ph.D. dissertation, Universität Zürich, 2004.

[6] S. Franklin, "A foundational architecture for artificial general intelligence," in *Artificial general intelligence*, Springer, 2007, pp. 36–54.

[7] B. Goertzel and C. Pennachin, *Artificial general intelligence*. Springer, 2007, vol. 2.

[8] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Information self-structuring: Key principle for learning and development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 533–547, 2007.

[9] A. Sloman, "Diversity of developmental trajectories in natural and artificial intelligence," in *AAAI Fall Symposium: Computational Approaches to Representation Change during Learning and Development*, Nov. 2007, pp. 70–77.

[10] L. B. Smith and C. Breazeal, "The dynamic lift of developmental process," *Developmental Science*, vol. 10, no. 1, pp. 61–68, 2007.

[11] A. M. Turing, "Computing machinery and intelligence," *Springer Netherlands*, pp. 23–65, 2009.

[12] S. Adams et al., "Mapping the landscape of human-level artificial general intelligence," *AI magazine*, vol. 33, no. 1, pp. 25–42, 2012.

[13] E. Nivel and K. R. Thórisson, "Bounded recursive self-improvement," *arXiv preprint arXiv:1304.6842*, 2013.

[14] E. Nivel and K. R. Thórisson, "Self-modeling agents: A framework for behavior introspection and autonomous self-improvement," *Cognitive Systems Research*, vol. 15, 2013.

[15] B. Goertzel, "Artificial general intelligence: Concept, state of the art, and future prospects," *Journal of Artificial General Intelligence*, vol. 5, no. 1, pp. 1–48, 2014.

[16] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, "Complementary learning systems," *Cognitive Science*, vol. 38, no. 6, pp. 1229–1248, 2014. DOI: 10.1111/j.1551-6709.2011.01214.x

[17] J. C. Baillie, "Artificial intelligence: The point of view of developmental robotics," in *Fundamental Issues of Artificial Intelligence*, V. C. Müller, Ed., 2016, pp. 415–424.

[18] Y. Duan, J. Schulman, X. Chen, et al., "RI^2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016.

[19] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? Complementary learning systems theory updated," *Trends in Cognitive Sciences*, vol. 20, no. 7, pp. 512–534, 2016. DOI: 10.1016/j.tics.2016.05.004

[20] B. R. Steunebrink, K. R. Thórisson, and J. Schmidhuber, "Growing recursive self-improvers," in *Intl. Conf. on Artificial General Intelligence*, 2016, pp. 129–139.

[21] B. R. Steunebrink, K. R. Thórisson, and J. Schmidhuber, "Recursive self-improvement for general intelligence," in *Artificial General Intelligence*, 2016, pp. 1–9.

[22] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Memory aware synapses: Learning what (not) to forget," *arXiv preprint arXiv:1711.09601*, 2017.

[23] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *International Conference on Machine Learning (ICML)*, 2017.

[24] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, e253, 2017.

[25] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476.

[26] L. B. Smith and L. K. Slone, "A developmental approach to machine learning?" *Frontiers in Psychology*, vol. 8, p. 296 143, 2017.

[27] A. Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] A. Cangelosi and M. Schlesinger, "From babies to robots: The contribution of developmental robotics to developmental psychology," *Child Development Perspectives*, vol. 12, no. 3, pp. 183–188, 2018.

[29] K. Li and J. Malik, "Learning to learn with online gradient descent," *arXiv preprint arXiv:1802.02871*, 2018.

[30] A. Nichol, J. Achiam, and J. Schulman, "First-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[31] F.-Y. Wang, J. Zhang, and X. Wang, "Parallel intelligence: Toward lifelong and eternal developmental ai and learning in cyber-physical-social spaces," *Frontiers of Computer Science*, vol. 12, pp. 401–405, 2018.

[32] K. Doya and T. Taniguchi, "Toward evolutionary and developmental intelligence," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 91–96, 2019. DOI: 10.1016/j.cobeha.2019.03.002

[33] A. Goyal, A. Lamb, J. Hoffmann, et al., "Recurrent independent mechanisms," *arXiv preprint arXiv:1909.10893*, 2019.

[34] S. Kakei, J. Lee, H. Mitoma, H. Tanaka, M. Manto, and C. S. Hampe, "Contribution of the cerebellum to predictive motor control and its evaluation in ataxic patients," *Frontiers in Human Neuroscience*, vol. Volume 13 - 2019, 2019, ISSN: 1662-5161. DOI: 10.3389/fnhum.2019.00216 [Online]. Available: https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2019.00216

[35] J. Pei et al., "Towards artificial general intelligence with hybrid tianjic chip architecture," *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[36] R. H. Gray, "The extended kardashev scale," *The Astronomical Journal*, vol. 159, no. 5, p. 228, 2020.

[37] W. Naudé and N. Dimitri, "The race for an Artificial General Intelligence: Implications for public policy," *AI & Society*, vol. 35, Jun. 2020. DOI: 10.1007/s00146-019-00887-x

[38] N. Fei et al., "Towards artificial general intelligence via a multimodal foundation model," *Nature Communications*, vol. 13, no. 1, p. 3094, 2022.

[39] Y. Mo, A. Ghosh, and J. Park, "Federated learning with hierarchical clustering of local updates for personalized predictive modeling," *Journal of Biomedical Informatics*, vol. 125, 2022.

[40] D. S. Moore, L. M. Oakes, V. L. Romero, and K. C. McCrink, "Leveraging developmental psychology to evaluate artificial intelligence," in *IEEE International Conference on Development and Learning (ICDL)*, Sep. 2022, pp. 36–41.

[41] S. Bubeck et al., "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

[42] D. R. E. Ewim, N. Ninduwezuor-Ehiobu, O. F. Orikpete, B. A. Egbokhaebho, A. A. Fawole, and C. Onunka, "Impact of data centers on climate change: A review of energy efficient strategies," *The Journal of Engineering and Exact Sciences*, vol. 9, no. 6, 16 397–01e, 2023.

[43] Y. Jin, "Computational evolution of neural and morphological development: Towards evolutionary developmental artificial intelligence," 2023.

[44] D. Mukto, *Cooper Black*. Blurb, 2023, ISBN: 9798880654475.

[45] N. Shinn and A. I. Labash, "Reflexion: Language agents with verbal reinforcement learning," *arXiv preprint arXiv:2303.11366*, 2023.

[46] W. Starkey and C. Ezenkwu, "Explainable developmental artificial intelligence through hybrid symbolic-neural pipelines," *Cognitive Computation*, 2023.

[47] W. Starkey and C. P. Ezenkwu, "Towards autonomous developmental artificial intelligence: Case study for explainable ai," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Jun. 2023, pp. 94–105.

[48] S. Yao, S. Zhao, D. Yu, et al., "Tree of thoughts: Deliberate problem solving with large language models," *arXiv preprint arXiv:2305.10601*, 2023.

[49] C. B. Young, V. Reddy, and J. Sonne, "Neuroanatomy, basal ganglia," in *StatPearls [Internet]*, Updated 2023 Jul 24. Available from NCBI Bookshelf, Treasure Island (FL): StatPearls Publishing, Jul. 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK537141/

[50] F. Cozzi, M. Pangallo, A. Perotti, A. Panisson, and C. Monti, "Learning individual behavior in agent-based models with graph diffusion network," *arXiv preprint arXiv:2505.21426*, 2025.

[51] C. Fu et al., "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv preprint arXiv:2306.13394*, 2025.

[52] J. Lehman, J. Sims, and K. O. Stanley, "Darwin gödel machine: Open-ended evolution of self-improving agents," *arXiv preprint arXiv:2505.22954*, 2025.

[53] Z. Liu, Y. Zhao, and Q. Yang, "Graph neural networks for databases: A survey," *arXiv preprint arXiv:2502.12908v1*, 2025.

[54] Z. Lyu, D. Zhang, W. Ye, F. Li, Z. Jiang, and Y. Yang, "Jigsaw-puzzles: From seeing to understanding to reasoning in vision-language models," *arXiv preprint arXiv:2505.20728*, 2025.

[55] H. Tan, H. Yan, and Y. Yang, "Llm-guided reinforcement learning: Addressing training bottlenecks through policy modulation," *arXiv preprint arXiv:2505.20671*, 2025.

[56] A. Wijaya Rahardja, J. Liu, W. Chen, Z. Chen, and Y. Lou, "Can agents fix agent issues?" *arXiv preprint arXiv:2505.20749*, 2025.

[57] F. Wijaya Rahardja et al., "Collaborative llm-assisted debugging in multi-agent negotiation failures," *arXiv preprint arXiv:2503.10292*, 2025.

[58] J. Xu, Y. Su, M. Wang, et al., "Decentralized collective world model for emergent communication and coordination," *arXiv preprint arXiv:2504.03353v1*, 2025.