

A HUMAN-LIKE APPROACH TO MULTI-TASK LEARNING



HANNE SAY EREN

ÖZYEĞİN UNIVERSITY

AUGUST, 2025

A HUMAN-LIKE APPROACH TO MULTI-TASK LEARNING

by
Hanne Say Eren

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Artificial Intelligence

Advisor: Prof. Erhan Öztop

Graduate School of Science and Engineering
Özyeğin University
İstanbul

August, 2025

A HUMAN-LIKE APPROACH TO MULTI-TASK LEARNING

Approved by:

Prof. Dr. Erhan Öztop, Advisor
Department of Artificial Intelligence and Data
Engineering
Özyeğin University

Assoc. Prof. Reyhan Aydoğan
Department of Artificial Intelligence and Data
Engineering
Özyeğin University

Assoc. Prof. Emre Uğur
Department of Computer Engineering
Boğaziçi University

Approval Date: July 31, 2025

To my mom, my dear grandpa Şeref and my beloved Arda



DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Hanne Say Eren

ABSTRACT

This thesis proposes a novel human inspired framework for multi-task learning that challenges traditional, rigid training schedules. We design a system in which a learning agent autonomously selects which task to engage in based on internal, task-agnostic signals, by a high-level control mechanism, the Task Controller, which facilitates self-directed learning. We first develop LP-Con, a modular neural architecture for interleaved multi-task learning, where **Learning Progress (LP)** serves as the guiding signal. The model employs task-specific network blocks to enable flexible, many-to-many knowledge transfer and a human-like interleaved learning process. We then extend this with IMTL-EMLP, which integrates **Energy Consumption (EC)** into the Task Controller to enhance energy efficiency without sacrificing learning performance. Both models are evaluated in a simulated robotic environment where an agent learns to predict the consequences of its actions across various object interaction scenarios. To validate the effectiveness of our proposed model, we conduct several experiments which include overall task performance analysis, task-wise learning performances, task selection patterns by Task Controller, interleaving vs. blocked learning, and finally object-wise cross task skill transfer. The baselines include single-task learning, multi-task learning with random task scheduling, and blocked multi-task learning. Experimental results show that our cognitively inspired methods significantly accelerate learning, improve generalization, show robustness to forgetting and increase computational efficiency. By aligning learning strategies with the self-regulated, dynamic nature of human development, this work contributes to the creation of more adaptive, efficient, and scalable lifelong learning systems.

Keywords: multi-task learning, interleaved learning, intrinsic motivation, neural efficiency, effect prediction

ÖZET

Bu tez, geleneksel ve katı eğitim planlarına karşı, insan esinli yeni bir çoklu-görev öğrenme çerçevesi önermektedir. Tasarladığımız sistemde, öğrenen bir ajan hangi göreve odaklanacağına, yüksek seviyeli bir denetim mekanizması olan Görev Denetleyici aracılığıyla, içsel ve göreve özgü olmayan sinyallere dayanarak öz-yönelimli biçimde karar verir. Öncelikle, **Öğrenme İlerlemesini (Öİ)** yönlendirici sinyal olarak kullanan, çoklu görevleri aralıklı öğrenme için modüler bir sinir mimarisi olan LP-Con'u geliştiriyoruz. Bu model, esnek ve çoktan-çoğa bilgi aktarımını sağlayan görev-özel ağ blokları sayesinde, insan benzeri aralıklı bir öğrenme süreci sunar. Ardından, IMTL-EMLP ile Görev Denetleyici'ye **Enerji Tüketimi (ET)** bilgisini entegre ederek öğrenme performansından ödün vermeksizin enerji verimliliğini artırıyoruz. Her iki model de, bir ajanın çeşitli nesne etkileşim senaryolarında eylemlerinin sonuçlarını öngörmeyi öğrendiği, simüle edilmiş bir robotik ortamda değerlendirilmiştir. Önerdiğimiz modellerin etkinliğini doğrulamak için; toplam görev başarımı analizi, göreve-özgü öğrenme performansları, Görev Denetleyici'nin görev seçme örüntüleri, aralıklı ve bloklu öğrenme karşılaştırması ile nesne-bazlı çapraz görev beceri transferi gibi kapsamlı deneyler yürütülmüştür. Karşılaştırılan modeller arasında tekli-görev öğrenme, rastgele görev sıralamalı çoklu-görev öğrenme ve bloklu çoklu-görev öğrenme yer alır. Deneysel sonuçlar, bilişsel ilhamlı yöntemlerimizin öğrenmeyi önemli ölçüde hızlandığını, genelleme kabiliyetini artırdığını, unutmaya karşı dayanıklılık gösterdiğini ve hesaplama verimliliğini yükselttiğini ortaya koymaktadır. Öğrenme stratejilerini insan gelişiminin öz-düzenlemeli ve dinamik doğasıyla hizalayarak, bu çalışma daha uyarlanabilir, verimli ve ölçeklenebilir yaşam boyu öğrenme sistemlerinin oluşturulmasına katkıda bulunmaktadır.

Anahtar Kelimeler: çoklu görev öğrenme, aralıklı öğrenme, içsel motivasyon, sinirsel verimlilik, etki tahmini

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and encouragement of so many wonderful people. I am deeply grateful to my advisor, Prof. Erhan Oztop, for his guidance, patience, and trust in me. Thank you for standing by me through every high and low, like a family would. I'm also sincerely thankful to Prof. Emre Uğur for his support and direction throughout my master's journey. To my lab friends -Zahra Koulaeizadeh, Arash Mehrabi, Emir Arditi, Bedirhan Çelebi, Nazlı Mehrabi, and many others- thank you for the shared work, laughter, and camaraderie that made the lab feel like home. To my roommate, Ayesha Noor, we met just this year, but you've been the best roommate I've ever had. Your kindness and friendship meant so much, and I look forward to seeing you again. To Berk, Uğur, and Ramin -thank you for the endless nights of laughter, deep conversations, and unwavering support. You became more than friends -you became family.

A very special thank you goes to my mother, for her unconditional love, endless support, and unwavering belief in me. Everything I've accomplished is a reflection of her strength and selflessness. I will never stop being grateful for her. To all my friends and family, thank you for believing in me.

And finally, to my beloved husband, Arda. There are no words strong enough to show what your support has meant to me. Through every late night, every doubt, and every emotional breakdown, you were there -patient, steady, and full of love. You gave me strength when I had none left, believed in me when I couldn't believe in myself. Thank you for sacrificing with me, for cheering for me, and for standing beside me no matter what. I couldn't have done it without you and I am endlessly grateful to share this life with you.

Thank you, all of you, from the bottom of my heart!

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iv
ABSTRACT	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xii
1 INTRODUCTION	1
2 RELATED WORK	4
2.1 Lifelong Learning in Neural Networks	4
2.1.1 The Plastic Brain: Neural Mechanisms of Continuous Learning.....	5
2.1.2 The Computational Pursuit of Lifelong Learning.	6
2.2 An Ecosystem of Related Learning Paradigms	12
2.2.1 Transfer Learning (TL):	12
2.2.2 Multi-Task Learning (MTL):	13
2.2.3 Curriculum Learning and Task Ordering	16
2.3 Interleaved Learning	17
2.4 Intrinsic Motivation and Learning Progress	19
2.4.1 Self-Directed and Intrinsically Motivated Exploration in Infancy ...	20
2.4.2 Intrinsic Motivation Signals in Autonomous Robots	21
2.5 Energy Efficiency in Cognitive Systems	23
2.6 Effect Prediction in Robotics and its Applications	24
3 HUMAN INSPIRED MACHINE LEARNING	27
4 LEARNING PROGRESS GUIDED MULTITASK LEARNING	29
4.1 Architectural Design	29
4.1.1 Interleaved Multi-Task Learning Framework	29

4.1.2	Implementation Details: LP Measure and Knowledge Transfer	33
4.1.3	Task Simulation and Training Settings	35
4.2	Experiment Settings	38
4.2.1	Baselines	39
4.2.2	Training	39
4.2.3	Evaluation Protocol.....	40
4.3	Results	41
4.3.1	Comparative Model Performance.....	41
4.4	Discussion	44
4.5	Conclusion.....	47
5	ENERGY MODULATED LEARNING PROGRESS GUIDED MULTITASK LEARNING	48
5.1	Methods.....	49
5.1.1	Multi-Task Learning Problem Statement.....	49
5.1.2	Proposed Neural Network Architecture	50
5.1.3	Task Arbitration	55
5.1.4	Experimental Settings	58
5.2	Results	63
5.2.1	IMTL-LP Only Results.....	64
5.2.2	Analysis of Task-wise Performance	64
5.2.3	Network Complexity and Learning Performance Relationship	65
5.2.4	Interleaved Learning versus Blocked Learning	66
5.2.5	Performance-Efficiency Trade-off: Modulating Learning Progress with Computational Energy	70
5.2.6	Investigation on Model Architecture and Transfer Learning Dynamics.....	72
5.2.7	Analysis of Cross-Task Attention and Self Attention of Tasks.....	77
5.3	Conclusions.....	78
6	OVERALL RESULTS AND CONCLUSION	80
6.1	Contributions to the Literature	81

6.2 Future Outlook	82
REFERENCES	84
APPENDICES	100
APPENDIX A — NETWORK ARCHITECTURE AND HYPER-PARAMETERS	101



LIST OF TABLES

Table 5.1: Overview of symbols and notations.	52
Table 5.2: The average total number of training iterations received by each task as the result of autonomous task selection in IMTL-LP is compared with uniformly random task selection (IMTL-RAND). The average and standard deviation is over 10 training sessions.	65
Table A.1: Input and output dimensions for the single-task learning models.	101
Table A.2: Input and output dimensions for the multi-task learning models.	101

LIST OF FIGURES

Figure 4.1:	High level task controller suitable for interleaved multi-task learning and ultimately continual learning.	30
Figure 4.2:	Proposed architecture containing the Core, Auxiliary and Shared blocks for each task, allowing bidirectional skill transfer between tasks, regardless of the training schedule.	33
Figure 4.3:	Effect prediction environments created in PyBullet simulation. Franka Emika Panda robotic arm is used as manipulating agent.	36
Figure 4.4:	Following the action parameter, the robot navigates through red and green markers, with a 30-degree example shown.	37
Figure 4.5:	Mean Absolute Error (MAE) of task training performances for both the baseline models and the proposed model (LP Con).	42
Figure 4.6:	Mean average task errors for training data (a) and testing data (b) over 10 repetitions. The standard deviation is denoted by vertical lines on the bars, and the symbols * and ** denote statistical significance at $p = 0.05$ and $p = 0.0001$, respectively.	44
Figure 4.7:	Task learning schedules dictated by the LP based Task Controller.	45
Figure 4.8:	Visualization of learning performance and LP trends of tasks by the proposed model LP-Con. Task 1 shows the worst performance while having the lowest LP trend. On the other hand, Task 2 and Task 3 converge faster than the Task 1 and show higher LP trends.	46
Figure 5.1:	Proposed neural network architecture for multi-task effect prediction. $\mathcal{T}_* = \mathcal{T}_2$	51
Figure 5.2:	Visual examples of tasks executed in the PyBullet simulation environment. Arrows in Push (a) and Hit (b) show the trajectory of the object after the action is executed. For Stack (c), arrows show the end-effector trajectory while executing the stacking action.	60

Figure 5.3:	The overall performance of the baselines is presented for isolated task learning (SINGLE), the architecture with random task selection (IMTL-RAND), and the proposed model utilizing the LP-based task selection (IMTL-LP). IMTL-LP demonstrates a faster convergence rate compared to the baseline models. While the random task selection demonstrates a marginal improvement over single task learning, it fails to exceed the performance achieved through LP-based task selection.	65
Figure 5.4:	The performance of the models on specific tasks is presented. The LP-based method outperforms the other baseline approaches in both the push and stack tasks, with a more notable advantage in the stack task. However, for the hit task, its improvement was limited to the training period between approximately 400 and 1000 steps.	66
Figure 5.5:	Comparative analysis of the learning performance of baseline models (SINGLE, IMTL-RAND) and our proposed model (IMTL-LP) across various network complexity configurations. Results are obtained at the midpoint of training.	67
Figure 5.6:	Prediction error plots for each task order combination of blocked learning and interleaved learning with LP. Each figure has three sections split by vertical dashed lines where the BLOCK model swaps tasks, where P is Push, H is Hit, and S is Stack. The rightmost location in each figure shows where BLOCK and IMTL-LP have seen all the tasks to learn. IMTL-LP outperforms BLOCK learning in all scenarios.	68
Figure 5.7:	Prediction error plots for each task order combination of blocked learning (tasks are trained with equal number of epochs they have received in LP based interleaved training) and interleaved learning with LP. In this case, the switch from one task to another is variable, due to different task selections regimes by IMTL-LP. Our model, IMTL-LP, still outperforms BLOCK learning in all scenarios.	69
Figure 5.8:	Evaluation of the baseline models and the LP-based model (IMTL-LP) against the EMLP-based model (IMTL-EMLP), with the sensitivity constant fixed at $k = 1$	71
Figure 5.9:	Effect of varying the energy sensitivity coefficient k in the IMTL-EMLP model on (a) prediction performance and (b) total energy consumption, at the midpoint of training. Higher values of k prioritize learning progress over energy efficiency, leading to lower prediction error but increased energy usage, while lower k values reduce energy consumption at the cost of predictive accuracy.	72

Figure 5.10: An analysis of the individual contributions of the shared attention mechanism and the flag bit within the proposed multi-task learning architecture reveals that neither component alone achieves the convergence efficiency attained by their combined integration in the full model.	73
Figure 5.11: Each sub-figure illustrates the impact of ablating source-to-target task representations on the performance of target tasks for individual objects. Specifically, the plots depict the increase in prediction error (loss) for each target task (columns) when the contribution of a corresponding source task (rows) is removed from the key and query components of the shared multi-head attention mechanism. This analysis highlights the extent to which each target task relies on shared information from other tasks, offering insight into the underlying transfer and interference patterns within the multi-task effect prediction framework.	75
Figure 5.12: Attention scores of the Full Model and Model w/o flag. Scores are calculated per-batch average for each training trial. Here mean of 10 trials are given.....	77

1. INTRODUCTION

From their earliest moments, human beings embark on a lifelong journey of learning. Infants, with no instruction manual or predefined curriculum, explore their surroundings with wonder and curiosity [1]. They touch, grasp, listen, move, and observe—gathering knowledge [2] not through neatly ordered lessons but through unorganized messy experiences. As they grow, children spontaneously switch between playing with a toy, mimicking an adult, experimenting with a new gesture, or exploring how their actions affect the world around them. This kind of learning is not just constant; it is self-directed [3, 4], intrinsically motivated [5, 6], and deeply efficient. It is also robust—humans do not forget how to walk when they begin to talk, nor do they need to relearn how to grasp objects every time they face a new task.

This remarkable adaptability and efficiency of human learning have long fascinated scientists and engineers. For decades, the goal of building intelligent machines has involved more than simply programming responses or training models to optimize for static objectives. Instead, researchers have looked toward the brain—not just for its computational power, but for its learning strategies. What if machines could learn not just like humans, but with the same principles that govern human development?

Today’s artificial intelligence systems, however, are still far from this ideal. Conventional machine learning methods typically learn tasks in isolation [7]. They are trained to perfection on one dataset, only to struggle—or even forget—when asked to learn a second. They usually lack the autonomy to decide *which* task to focus on, or *when* to switch, and often require extensive labeled data and carefully curated environments [8]. Unlike children, they are not curious. They do not explore without being told. They do not remember what they’ve already learned unless explicitly designed to do so.

To bridge this gap, the fields of continual learning and multi-task learning have emerged. Continual Learning (CL) [9] aims to build systems that can learn a sequence of

tasks without forgetting previous ones; while Multi-Task Learning (MTL) [10] attempts to exploit shared representations to improve performance across tasks learned simultaneously. Both paradigms have made important strides, yet neither fully captures the essence of how humans learn over time. In particular, they often rely on fixed task schedules, assume complete mastery before switching, and lack mechanisms for adaptive task engagement, features that are at odds with how children learn naturally through dynamic, interleaved interactions with their environment.

More critically, these approaches rarely consider the *why* of learning. Human learners are driven not only by external goals, but also by intrinsic motivation: a desire to learn for its own sake [11]. They are drawn to what is novel, surprising, or just difficult enough to be engaging [12]. This drive underlies much of the exploratory behavior seen in infants and is closely linked to the concept of learning progress—the internal satisfaction that comes from improving at something. Moreover, the human brain meets all of these demands while managing energy efficiently, selectively allocating attention and cognitive effort based on task difficulty, interest, or fatigue [13, 14, 15].

In this thesis, we aim to bring these human-like principles into machine learning, particularly in the context of robotic lifelong learning. We introduce two complementary models that together form a cognitively-inspired framework for continual, interleaved multi-task learning. The first model leverages learning progress as a signal of intrinsic motivation to guide autonomous task selection. Instead of following a fixed curriculum, the agent dynamically switches between tasks based on where it is learning the most, mirroring how a child might return to stacking blocks after a few minutes of drawing. The second model adds a further layer of biological realism by incorporating neural energy consumption into the decision making process. This addition allows the system not only to prioritize learning opportunities but also to do so in an energy-efficient manner, balancing improvement with computational cost, like an echo of how the human brain optimizes

performance under metabolic constraints.

To test our approach, we instantiate these models in a simulated robotic environment where a manipulator learns to predict the effects of its actions: pushing, hitting, and stacking objects in varying configurations. These tasks are selected not only for their relevance to robotic manipulation but also because they reflect a developmental trajectory seen in infants, who learn to manipulate objects through play, gradually mastering the laws of physics through embodied interaction.

Our findings demonstrate that interleaved, intrinsically motivated learning leads to faster convergence, better generalization, and robustness to the catastrophic forgetting. Furthermore, by modulating task selection through computational energy metrics, we show that machines can learn intelligently while conserving resources. Through ablation studies, we further highlight the individual contributions of components in skill transfer sub-architecture, which facilitate selective skill sharing and prevent negative interference. Ultimately, this work takes a step toward building more human-like learners, systems that can flexibly acquire, retain, and apply skills over time, without the brittle constraints of traditional machine learning pipelines. By embedding the principles of curiosity, interleaved experience, and energy-aware reasoning into artificial agents, we not only improve performance but move closer to machines that learn as humans do: continuously, adaptively, and intelligently.

2. RELATED WORK

2.1 Lifelong Learning in Neural Networks

Biological intelligence is the ability to learn novel information throughout life, allowing one to adapt to a world that is constantly shifting and unpredictable [16, 17]. Humans, in particular, have a unique capacity to learn novel concepts, keep up with new information, and recall an extensive amount of information throughout their lifespan. This process, which is called “lifelong learning”, is not just an educational goal; it’s a basic cognitive function that comes from the brain’s inherent plasticity [18, 19, 20]. Over the past few years, the demand for Artificial Intelligence (AI) has focused on this same major challenge. When AI systems are employed in the real world, the traditional, static training methods fail to perform effectively, since these models generally learn from fixed datasets and are not designed to learn continuously. When they are trained on new information, they often experience “catastrophic forgetting,” [21, 22] meaning that the pre-acquired knowledge is erased due to updating the internal states while learning the new information. This has led to the field of CL [9, 23] in machines, which is an area of study that aims to give machines the ability to learn and adapt over time, just like human beings do. So, the need for machines that are adaptive and efficient is linked to solving the lifelong learning problem. The existing literature demonstrates that machine learning has introduced and explored several specialized paradigms, such as Transfer Learning [24], Multi-Task Learning [10, 25], and Curriculum Learning [26], which serve as foundational components of continual learning. Current literature suggests to us that artificial systems continue to fall short of competing with the lifelong learning capabilities of the human brain and also highlights the need to incorporate human learning mechanisms into machine learning models.

2.1.1 The Plastic Brain: Neural Mechanisms of Continuous Learning.

Human capacity for lifelong learning is grounded in the brain's inherent biological characteristics, foremost among these being *neuroplasticity* [27]. This term describes the brain's extraordinary ability to reshape itself by forming new neural connections and adjusting existing ones in response to experiences and learning. Importantly, this adaptability is not limited to childhood; the brain maintains its plastic nature throughout life [28]. Core mechanisms underlying this plasticity include *neurogenesis* [29], the creation of new neurons, which continues in brain regions such as the hippocampus well into adulthood, alongside the dynamic strengthening or weakening of synaptic connections between neurons [30, 31]. Engaging in mentally stimulating activities, ranging from reading and playing games to acquiring new skills, actively promotes these neural changes, thereby reducing the risk of cognitive decline and dementia [32]. This body of evidence supports the "use it or lose it" principle: neuroscience demonstrates that learning new skills physically alters the brain, and these changes may diminish without ongoing practice, underscoring the necessity of continuous cognitive engagement. While neuroplasticity enables the brain to absorb new information, a key challenge is accomplishing this without forgetting the pre-existing knowledge. The brain must navigate the stability-plasticity dilemma: it needs to remain flexible enough to incorporate new learning yet stable enough to preserve established memories from interference. Rather than relying on a single mechanism, the brain employs a sophisticated, system-wide approach, conceptualized by the Complementary Learning Systems (CLS) theory [33]. This prominent theory posits that learning and memory arise from the interaction of two distinct, yet collaborative, neural systems:

- **The Hippocampus and Medial Temporal Lobe:** This system is characterized by high plasticity and a fast learning rate. It allows for the rapid encoding of new, specific episodic memories-the "what, where, and when" of individual experiences. It learns quickly but its representations are separated to avoid interference [34].

- The Neocortex: This system is characterized by high stability and a slow learning rate. It is responsible for the gradual, interleaved integration of information over time. By learning slowly and integrating new information with existing knowledge, the neocortex builds a structured and generalized representation of the world, extracting statistical regularities from experience [34].

According to CLS theory, these two systems work in collaboration [35]. The hippocampus quickly learns new information; then, during rest and sleep, this information is gradually replayed to the neocortex. This process lets the neocortex integrate the information into long-term memory without causing major disruptions. This separation of roles—a fast, specific learner and a slow, general learner—is the brain’s clever solution to the stability-plasticity dilemma.

On a more detailed level, this stability is supported by processes like *synaptic consolidation* [36]. Here, synapses that are important for a solid memory become less flexible, protecting them from being overwritten by new learning. This multi-level biological strategy, which involves different brain areas, varying learning speeds, and protective mechanisms at the cellular level, contrasts sharply with the often uniform structures of artificial neural networks [37]. It offers great inspiration for building stronger AI systems.

2.1.2 The Computational Pursuit of Lifelong Learning.

Inspired by the way biological systems, especially humans, learn continuously throughout their lives, AI field aims to develop systems capable of similar lifelong learning. This section shifts focus from human learning models to how machines implement CL. It clearly defines the computational problem that CL attempts to solve, highlights its key and longstanding obstacle, the catastrophic forgetting, and systematically reviews the three main categories of algorithms designed to address this issue. This examination shows that continual learning as a research area is advancing quickly, with increasingly

sophisticated methods that draw directly from the neuro-cognitive processes observed in the human brain.

The terms lifelong learning and incremental learning are often used interchangeably with CL, as they refer to the same core idea. CL describes an AI model’s ability to learn from a continuous flow of data, typically non-stationary, meaning the underlying data distribution changes over time, while progressively accumulating knowledge across a sequence of tasks [9, 38]. A key limitation is that, when learning new tasks, the model usually cannot access the entire dataset from previous tasks due to constraints such as memory, privacy, or computational resources. This sequential learning setup fundamentally differs from traditional machine learning paradigms, which rely on offline or batch training: where models are trained on a large, fixed dataset assumed to be independently and identically distributed (i.i.d.). Although batch training suits well-defined problems, it fails to handle real-world scenarios where data distributions are dynamic and evolving [39]. Therefore, continual learning is vital for many practical domains such as robotics [40, 41, 42], autonomous driving [43, 44], and personalized recommendation systems [45, 46] where new information, users, or contexts constantly emerge. To better understand and evaluate continual learning, researchers have established various learning scenarios [47], which help standardize the problem setting, enabling rigorous benchmarking and comparison of different approaches in the field:

1. Task-Incremental Learning (TIL) [48]: The model knows the task identity during testing. For each input, it understands which task it belongs to (for instance, “classify this image as one of the 10 classes from Task 1”). The challenge is to keep performance steady across all tasks without mixing them up.
2. Domain-Incremental Learning (DIL) [48]: The task stays the same (for example, classifying the same set of 10 digits), but the data distribution changes (for instance,

learning from MNIST digits, then from SVHN digits).

3. Class-Incremental Learning (CIL) [48]: The model has to learn new classes one at a time (for example, learn to classify digits 0-4, then 5-9) and, during testing, it must be able to tell apart all the classes it has seen so far without being told which task the input is from. This requires the model to remember old classes and also to avoid mixing them up with new ones.

These clear definitions are important because the choice of scenario affects the problem's complexity and the right methods for evaluation.

2.1.2.1 The Challenge: Catastrophic Forgetting

The main challenge to continual learning is known as catastrophic forgetting or catastrophic interference. This issue was first documented in detail by McCloskey and Cohen [21]. They showed that when a traditional connectionist neural network is trained sequentially on a new task, it often loses its ability to perform previously learned tasks very quickly. The cause of catastrophic forgetting lies in how neural networks learn: These models store knowledge in a distributed way, meaning that information is spread across the connection weights of the entire network instead of being in one place [49]. So when a traditional neural network model learns a new task, it adjusts these weights using an optimization algorithm like gradient descent to reduce the error on the new data. However, this process changes the weights that were optimized to represent the information from the earlier task, ultimately overwriting it. This harmful change occurs because the typical training goal focuses only on performance for the current task and does not have a way to protect the representations learned previously, which is formally known as the stability-plasticity dilemma [50]. An effective lifelong machine learning system must carefully balance two needs: it must be flexible enough to integrate new information and take on new tasks, but it also needs to be stable enough to maintain existing knowledge.

Most research in continual learning focuses on creating algorithms that can handle this trade-off effectively.

2.1.2.2 Continual Learning Strategies

In response to the challenge of catastrophic forgetting, researchers have created many strategies and most of the methods fit into three main groups: regularization-based, replay-based (also called rehearsal), and architectural approaches:

- **Regularization-Based Approaches:** Regularization-based methods are generally offer solution to catastrophic forgetting by modifying the model's loss function. They introduce a penalty term that discourages large changes to parameters considered important for performance on previously learned tasks [9]. This method directly focuses on maintaining stability by limiting the model's ability to adapt. A key study in this area is Elastic Weight Consolidation (EWC), introduced by Kirkpatrick *et al.* [51]. EWC draws inspiration from the biological process of synaptic consolidation. It identifies the significance of each weight for earlier tasks by calculating the diagonal of the Fisher Information Matrix. This matrix estimates the shape of the loss landscape, with higher values showing weights that are very sensitive to change. When learning a new task, EWC adds a quadratic penalty to the loss function that makes it more difficult to alter these important weights, thus keeping the model anchored to its previous solution. Another significant method is Synaptic Intelligence (SI) from Zenke *et al.* [52]. Instead of calculating the Fisher matrix after each task, SI assesses weight importance continuously during training. It tracks how much each weight contributes to reducing the loss, allowing for a more efficient update of the importance weights as the model learns. More recent work [53], following EWC and SI, proposed a more computationally efficient, unsupervised method for calculating parameter importance based on how sensitive the model's

output function is to a change in a parameter.

- **Replay-Based (Rehearsal) Approaches:** Replay-based methods directly address the issue of forgetting by keeping a small, representative subset of past task data in a memory buffer or replay buffer [9]. While training on a new task, these stored “exemplars” mix with the new data, allowing the model to review the old tasks and maintain its performance on them. This approach is similar to memory replay seen in the hippocampus, which is important for consolidating memory in the brain [33]. One of the early methods in this area is Gradient Episodic Memory (GEM) by Lopez-Paz and Ranzato [54]. GEM uses the stored exemplars not for full re-training but to limit the gradient updates for the new task. It calculates the gradients for the current task and then adjusts them to ensure the weight update does not worsen the loss on previous tasks. A significant method, especially for the difficult class-incremental learning (CIL) scenario, is iCaRL by Rebuffi *et al.* [55]. iCaRL combines a replay buffer with a nearest-mean-of-exemplars classifier. It uses the stored data to prevent shifts in feature representation and applies the class means of these exemplars for making classification decisions, which proves more stable than a standard linear classifier in the CIL setting. To tackle storage issues associated with storing real data, more innovative techniques like generative replay [56] have been developed. These methods train a generative model (for example, a GAN [57] or VAE [58] to capture the data distribution of past tasks, enabling the generation of synthetic pseudo-samples for practice without needing to keep any actual data. Also, more sophisticated approaches are studied in the literature, such as [59, 60].
- **Architectural Approaches:** The third family of methods addresses catastrophic forgetting at an architectural level. The main idea is to assign different sets of parameters for various tasks, aiming to prevent interference between tasks. It can

be achieved by either growing the network dynamically or by dividing the parameters of a fixed-size network. Progressive Neural Networks (PNN), proposed by Rusu *et al.* [61], illustrates the dynamic growth approach. For each new task, PNN creates a new neural network “column” and freezes the parameters of all previous columns. The new column connects to the earlier ones, allowing it to extract useful features while ensuring that knowledge in the old columns remains unchanged, allowing to mitigate catastrophic forgetting. However, a significant downside is that the model’s size increases linearly with the number of tasks, making it impractical for lengthy learning sequences. On the other hand, methods like PackNet [62], function on a fixed-size network. PackNet uses iterative pruning to eliminate unnecessary weights after completing a task. The available parameters are then used to train the next task, with a binary mask ensuring that the gradients from the new task only impact its assigned weights. This approach effectively “packs” multiple tasks into one network. An interesting neuro-inspired approach is context-dependent gating, proposed by Masse *et al.* [63]. In this method, for any specific task, only a sparse, randomly chosen subset of neurons in the network is activated. This process creates unique, overlapping sub-networks for each task within a shared architecture. It mimics how the brain may engage different neural groups for various functions.

The current success of these three distinct families of solutions highlight a strong connection between engineering and biology, reflecting the brain’s strategies for managing memory, including synaptic consolidation, hippocampal replay, and functional specialization of neural circuits [37, 9]. This parallel evolution supports the interdisciplinary approach of this review and indicates that improving AI requires a better understanding of these biological principles. Moreover, the field’s shift from a purely defensive stance of “preventing forgetting” to a more proactive one of “how to efficiently update and transfer knowledge” [38] shows a growing understanding of the issue. This evolution brings

us closer to the essence of lifelong learning. Nonetheless, it is clear that no single best method exists. Each family of methods has different trade-offs regarding performance, memory cost, and computational demands. Therefore, choosing a strategy becomes a vital, context-dependent choice.

2.2 An Ecosystem of Related Learning Paradigms

In this section, we review learning paradigms closely related to continual learning, including transfer learning, multi-task learning, and curriculum learning. While these methods support knowledge sharing, generalization, and training efficiency, we compare them with continual learning to clarify how they differ in task ordering, data availability, and forgetting mitigation.

2.2.1 *Transfer Learning (TL):*

Transfer learning is a machine learning technique whose main goal is to enhance a model’s performance on a specific task by using knowledge gained from a different but similar source task [64]. This is especially useful in real-life situations where there isn’t much labeled data for the target task, or where it is costly to collect, while data for a related source task is plentiful. The idea of knowledge transfer has been around for a long time [65]. However, it gained more attention with the rise of deep learning. Nowadays, the usual method is to use models that have been pre-trained [66] on large datasets, such as ImageNet [67] in computer vision or extensive text collections in natural language processing, as a foundation. These pre-trained models act as strong feature extractors, and we can adjust their learned representations for a new, specific task with much less data and lower computational costs. The success of modern foundational models like BERT [68] and GPT [69] highlights the strength of transfer learning. The surveys by Pan and Yang [64] and Zhuang *et al.* [70] helped to define the field, organizing transfer learning into several categories:

- Inductive Transfer: The source and target tasks are different, but the domains may be similar. This is the most common setting, such as fine-tuning an image classification model for object detection [71].
- Transductive Transfer (Domain Adaptation): The source and target tasks are the same, but the domains (data distributions) are different [72]. For instance, adapting a sentiment classifier trained on book reviews to work on movie reviews by explicitly aligning the feature distributions of the two domains to make them indistinguishable to the model.
- Unsupervised Transfer: Similar to inductive transfer, but the learning in the source and/or target domains is done with unlabeled data [64, 68].

The relationship between TL and CL is fundamental: transfer learning is a core mechanism that a continual learning system must employ. The goal of a CL system is not just to avoid forgetting but also to exhibit positive forward transfer (where new tasks benefit from the knowledge gathered from previous tasks) and, ideally, positive backward transfer (where learning a new task improves performance on related past tasks) [73]. The challenge of CL is to enable this continuous transfer of knowledge between sequential tasks while mitigating negative transfer, the phenomenon where knowledge from one task actually hinders performance on another, which is the essence of catastrophic forgetting.

2.2.2 Multi-Task Learning (MTL):

Multi-task learning is an approach that aims to improve the generalization of a model by learning multiple related tasks simultaneously [10]. By training on several tasks at once, the model learns a shared representation that captures what is common among the tasks. This shared knowledge helps the model create a stronger and more general feature space than if it learned each task separately. Multitask learning approaches can be divided into two broad categories:

- **Hard Parameter Sharing:** This is the most common approach, where a shared network backbone (e.g., the convolutional layers in a CNN) is used to learn a common representation, followed by separate, task-specific output layers or “heads” [74, 75, 76].
- **Soft Parameter Sharing:** Each task has its own model and parameters, but a regularization term is added to the loss function to encourage the parameters of the different models to be similar [77, 78, 79].

While effective, these simple sharing schemes can face task interference [75], where conflicting gradients from different tasks hurt overall performance. This issue has pushed for the development of better architectures.

For instance, in feature learning variant, a typical approach involves either mapping the initial features to a common representation space [80, 81, 82, 83] or choosing a pertinent subset of task-specific features [84, 85]. In addition to these feature-based methods, parameter-based models have also been investigated within MTL. For example, low-rank strategies [86, 87] exploit the idea that related tasks produce a low-rank model parameter matrix, while task clustering techniques [88, 89, 90] categorize tasks into groups that share parameters. Other parameter-based approaches include modeling task relationships [91, 92] and utilizing parameter matrix decomposition [93, 94] methods.

In the field of reinforcement learning (RL), MTL is employed to enhance agent performance across various decision-making tasks by leveraging common knowledge and representations. A significant obstacle in multi-task reinforcement learning (MTRL) is negative interference, where progress on one task can detrimentally impact others [95]. To tackle this issue, Contextual Markov Decision Process (CMDP) [96] introduces context vectors to define the transition and reward parameters for each task. This concept

was later expanded by [97] to create the Block Contextual Markov Decision Process (BC-MDP), which incorporates the state space into the context model. Likewise, [98] proposed learning orthogonal representations through the Gram-Schmidt process to foster diversity. Alternatively, [95] developed a distilled policy to create a shared representation that guides task-specific policies while reducing adverse updates. Together, these approaches highlight the necessity of structured shared representations and the reduction of negative transfer in MTRL. Our model resolves this with an attention layer that enhances advantageous skill transfer while limiting negative interference.

Another unresolved issue in MTL, aside from learning shared skills, is addressing the disparities in task complexity and data distribution. Recent strategies include adaptive loss weighting and gradient normalization. For example, Kendall *et al.* [99] proposed an uncertainty-aware framework that modifies task losses according to inherent noise levels, emphasizing more dependable learning signals. In conjunction with this, [100] was proposed, which adjusts task gradients to ensure that no single task predominates the optimization process.

Furthermore, [101], by Sener and Koltun, reformulated MTL as a multi-objective optimization challenge, seeking to establish Pareto-optimal trade-offs among competing task objectives. More dynamic approach of learning task relationships, rather than having rigid transfer structures, was proposed in Cross-Stitch Networks [79], which learn a linear combination of the activations from different task-specific networks. This allows the model to learn the optimal sharing strategy. More recently, models like the Multi-gate Mixture-of-Experts (MMoE) [75, 102] use gating mechanisms to figure out how to route information and share parameters among tasks, effectively modeling task relationships. CAGrad [103] represents a different, cutting-edge approach that tackles task interference directly at the gradient level.

The difference between MTL and CL lies in when tasks are available. MTL is

a parallel approach where all tasks and their data are available from the start. CL is a sequential approach where tasks come one at a time, and past data is usually not available. Despite this difference, their goals are closely related. The ideal outcome of a CL process is a model that performs as well as a model trained together on all tasks from the beginning [38]. In other words, it is a model that reaches the performance of its MTL equivalent but in a gradual way. The shared representations learned in MTL are exactly the kind of general knowledge that a CL system tries to build and gather over time [9].

2.2.3 Curriculum Learning and Task Ordering

Inspired by how humans and animals learn, curriculum learning is a training approach which presents training examples to a model in a structured order, usually moving from easy to hard, as proposed in Bengio *et al.* [26]. The idea is that “starting small” can improve the optimization process. When the model learns simpler concepts first, it can find a better spot in the complex loss landscape. This leads to faster progress and a better final result than if it faced the full complexity of the dataset right away [104]. This concept builds on earlier work [105].

There are two main challenges in curriculum learning: first, one should define a measure for difficulty. For instance, in visual correspondence tasks, difficulty can be quantified by the degree of geometric deformation between image pairs, with less deformed pairs being considered easier [106]. The second challenge is, creating the *pacing function* [107] that decides when to introduce more challenging examples. Initially, these metrics were often created by hand, however, the field has moved towards Automatic Curriculum Learning (ACL) [108]. In ACL the curriculum is learned dynamically. Other methods like self-paced learning [109, 110], the model’s performance indicates when to increase difficulty. In more sophisticated cases, a separate “teacher” model, usually trained with reinforcement learning, selects the next optimal example or task for the

“student” model to enhance its learning performance [111, 112]. The connection of curriculum learning to continual learning is direct: The sequence of tasks in a CL setting can be viewed as a curriculum. While this curriculum is often determined by the environment and outside the model’s control, in scenarios where an agent can influence its learning path, curriculum strategies become highly relevant [113]. An agent can use active task selection [114] to choose the next task to learn in order to maximize future performance or knowledge transfer. For example, learning foundational tasks first (e.g., basic object recognition) can provide a solid representation that makes it easier to learn more complex, related tasks later in the sequence (e.g., scene understanding) [106, 115]. Thus, curriculum learning provides a set of principles for intelligently structuring the lifelong learning process to make it more efficient and effective.

Overall, these ideas create a tightly connected framework. The findings suggest that an effective lifelong learning system works under a continual learning setting, has the ability to transfer knowledge between tasks, and reuses existing knowledge. Its main goal should be to develop a strong multi-task representation that applies to all encountered tasks. The efficiency of such a system can improve if it follows a clear or indirect curriculum that structures the learning process. Recognizing these connections is important, as it will lead to a pathway for creating adaptive and general artificial intelligence.

2.3 Interleaved Learning

Interleaved learning refers to a cognitive approach in which learners alternate between different topics or problem types, rather than focusing exclusively on a single subject. Substantial empirical evidence from cognitive psychology underscores the advantages of interleaved practice. For instance, research has shown that interleaving mathematical problems enhances learners’ ability to discriminate among problem types, thereby improving problem-solving capabilities [116]. Similarly, work by Kornell and

Bjork [117] showed that interleaved learning fosters the development of more flexible and integrated conceptual representations, which in turn supports the effective transfer of knowledge to novel contexts. Additional benefits such as improved retention and transfer have also been observed in the domain of motor skill acquisition [118, 119].

Neuro-imaging studies provide further support, revealing that interleaved practice is associated with increased activity in the frontal–parietal network, heightened excitability in the motor cortex, and reduced information retrieval time. These neural correlates collectively contribute to superior long-term retention and more efficient recall compared to blocked practice [120]. The efficacy of interleaved practice is often attributed to the phenomenon of *contextual interference (CI)* [121], and aligns with the concept of *desirable difficulties* in learning [122]. This framework posits that introducing certain challenges during learning can enhance long-term memory retention [123]. In a study by [124], students solved math problems presented in a shuffled order, preventing them from encountering problems of the same type sequentially. This forced learners to generate appropriate solution strategies in a manner reflective of real-world scenarios. The results demonstrated that interleaved practice led to higher performance on both immediate and delayed assessments, indicating a reduction in forgetting. Notably, the benefits of interleaved practice appeared to persist-or even amplify-over time. A similar trend was observed in a study by [125], where undergraduate physics students exposed to interleaved problem sets outperformed those engaged in blocked practice in terms of memory retention and problem-solving proficiency.

These findings collectively underscore the efficacy of interleaved learning as a robust strategy for enhancing learning outcomes in both academic and applied contexts. In contrast, the application of interleaved learning within the field of machine learning has received limited attention. Mayo *et al.* [126] recently explored the role of interleaving in multi-task learning and argued that rather than relying on mechanisms to mitigate

forgetting, such as external memory modules [127] or weight regularization techniques [128, 51], research should focus on designing learning architectures and schedules that leverage the inherent robustness of human learning processes. From this perspective, forgetting is not viewed as a flaw but as an integral feature of learning, coexisting with the capacity for rapid re-acquisition [129] and long-term knowledge retention. Their findings suggest that even in the absence of explicit forgetting prevention strategies, standard neural networks exhibit memory retention effects akin to those observed in human learners when tasks are interleaved [126].

2.4 Intrinsic Motivation and Learning Progress

Human beings have a natural urge to learn, starting from their earliest moments. From birth, infants are not passive receivers of information; they are active, motivated, and engaged learners who have a remarkable range of cognitive skills [1, 130]. They gain knowledge mainly through exploration [131, 4] without any instruction manual or predefined curriculum. This contradicts the traditional educational methods, which often emphasize explicit, structured instruction as the main way to acquire knowledge [132].

The idea of infants as active, motivated, and curious learners is a well-established principle in developmental psychology [1]. The natural curiosity and engagement seen in infants are not just traits; they are essential factors that drive them to seek out, interact with, and process information from their surroundings [133, 134, 12]. This active exploration is a necessary starting point and a key reason for the "messy, interleaved experiences" noted in early development, as it begins the ongoing cycle of learning. The curiosity in infant learning are specifically identified in research as *natural curiosity* and a *motivated* state. This motivated exploration serves as the way infants gather knowledge [131, 135], creating a clear cause-and-effect link [12] where their intrinsic drives lead to active learning behaviors. In the next section, these intrinsic drives and how they guide

the learning process in humans, especially infants, will be reviewed.

2.4.1 *Self-Directed and Intrinsically Motivated Exploration in Infancy*

Infants are often seen as naturally curious, they actively engage with their environment using various sensory and motor actions like touching, grasping, listening, and moving [2]. These actions are not random; they play an essential role in how infants process and understand the world [4]. For example, everyday experiences such as crying and being picked up or waving a toy and hearing it rattle give infants chances to learn about cause and effect [136]. This knowledge helps them understand object properties, human behavior, and the relationships between events and consequences, which in turn builds their problem-solving and prediction skills [135]. Their natural urge to explore and experiment drives cognitive development [1], as they seek information to fill knowledge gaps and handle uncertainty. This shows a form of “information-seeking” behavior from a young age. Describing early learning as *self-directed* and *intrinsically motivated* is a key idea in modern developmental psychology. Self-Determination Theory (SDT) [137] is an important framework for understanding human motivation, developed by Edward Deci and Richard Ryan. SDT [138] suggests that humans have basic psychological needs for autonomy (feeling in control of one’s actions), competence (feeling effective and skilled), and relatedness (feeling connected to others).

Intrinsic motivation is engaging in an activity for its own enjoyment, interest, or satisfaction instead of for extrinsic rewards or outcomes [139, 12]. It is essential for promoting active engagement, critical thinking, and problem-solving skills in early childhood. Research shows that children who are intrinsically motivated display more curiosity, persistence, and enthusiasm when learning [140]. Karl Friston’s Free Energy Principle (FEP) [141] provides a detailed framework that supports intrinsic motivation and self-directed exploration. FEP suggests that intelligent agents, including the human

brain, work to minimize energy (or surprise) [142] or prediction error [143] to survive and adjust to their changing environment. According to Karl Friston [141], the brain’s main job is to keep its internal states stable by minimizing surprise, which occurs when predictions do not match sensory input. This process drives perception, as it updates internal models, and action by sampling inputs that align with those models.

Importantly, curiosity-driven actions reduce uncertainty, which minimizes future surprise. Thus, learning, exploring, and developing more accurate predictive models are inherently rewarding because they satisfy the brain’s basic need to minimize surprise [144]. This creates a strong link between a neuro-biological principle and the psychological concept of intrinsic motivation, highlighting its effectiveness in early learning. While the FEP [141] provides a unifying neuro-biological account, the field of computational IM has developed several distinct, operational families of IM. Beyond surprise-minimization, these can be broadly categorized into two types [145]. The first is knowledge-based IM, where the reward is derived from prediction errors or novelty, driving the agent to build a better forward model of the world. A prominent example is the Intrinsic Curiosity Module (ICM), which rewards the agent for actions leading to states that its own model finds difficult to predict [146]. The second is competence-based IM, where the reward is derived from improvements in the agent’s own skill set. Here, the agent is motivated to master challenges, for instance by maximizing its learning progress over its sensorimotor models [12] or by maximizing its “empowerment,” which measures the agent’s control over its future sensory inputs [147]. These different approaches reflect the distinction between wanting to know more about the world versus wanting to be more capable within it.

2.4.2 Intrinsic Motivation Signals in Autonomous Robots

The psychological principles of intrinsic motivation have provided a powerful framework for addressing some of the most persistent challenges in machine learning

and robotics, particularly the problems of sparse rewards [148, 149] and the need for autonomous, open-ended learning. By creating an internal drive for exploration and competence gain, IM algorithms enable agents to learn effectively in complex environments without constant external feedback or human-engineered reward functions. The applications of this paradigm span reinforcement learning, robotics, and the pursuit of lifelong learning agents [150].

In robotics, IM has been foundational to the field of developmental robotics, which aims to create robots that learn skills in a continuous, self-directed manner, akin to human infants [145]. Instead of being pre-programmed for a specific task, an intrinsically motivated robot engages in “motor babbling”—exploring the limits of its own actuators and learning how its actions affect the environment. This exploration is guided by competence-based IM, where the agent is rewarded for improving its skills or maximizing its learning progress. This allows the robot to build a repertoire of reusable skills, such as grasping, pushing, or tool use, without any extrinsic task goal [145]. For example, by generating its own goals (e.g., what happens if I touch that object?) and being rewarded for achieving them, a robot can autonomously practice and master a wide array of manipulations, creating a valuable foundation for solving downstream tasks more efficiently later on [151].

Beyond solving single, complex tasks, IM is a key enabling technology for the long-term vision of autonomous, lifelong learning agents. In this context, IM provides a principled mechanism for autotelic (self-purposeful) goal setting and automatic curriculum learning. Instead of relying on a human to define a sequence of tasks, an agent can use its intrinsic drives to generate its own curriculum. Intrinsically Motivated Goal Exploration Processes (IMGEP) [113] formalize this by allowing an agent to sample goals in a “goal space” and prioritize those that offer an optimal trade-off between novelty and

feasibility, leading to a natural progression from simple to more complex skills. This approach directly addresses the question of "what to learn next?" in an open-ended world, allowing an agent to continuously expand its knowledge and capabilities without limit, which is a cornerstone of creating truly adaptive artificial intelligence [152, 153].

2.5 Energy Efficiency in Cognitive Systems

The human brain, despite representing only about 2% of the body's mass, is a remarkably expensive organ, consuming approximately 20% of the body's total metabolic energy budget [154]. This disproportionate energy demand has imposed a powerful and persistent evolutionary pressure to develop strategies for maximizing computational efficiency. This principle of energy conservation is not merely a biological constraint but a fundamental driver that has shaped the very architecture of neural circuits, the nature of cognitive processing, and the subjective experience of thought itself. Understanding this deep-seated need for efficiency provides critical insights for developing AI systems that are not only powerful but also resourceful and adaptive.

At the neural level, the brain employs several sophisticated mechanisms to minimize energy expenditure. One of the most fundamental is sparse coding, where information is represented by the strong activation of a very small subset of neurons at any given time. This is far more efficient than a dense code where all neurons participate, as the primary metabolic cost of neural processing is the generation of action potentials [155]. A second, more encompassing framework is predictive coding, formalized under the Free Energy Principle [141]. This theory posits that the brain does not passively receive and process the entirety of sensory information from the world. Instead, it actively generates predictions about incoming sensory data based on an internal model. The bulk of neural processing is then dedicated only to the prediction error—the mismatch between what

was expected and what was received. By minimizing this error over time, the brain refines its world model and requires minimal energy to process predictable, expected events [143, 156]. This predictive architecture ensures that cognitive resources are reserved for novel and surprising information that is most valuable for learning.

This biological imperative for energy conservation manifests psychologically as a general law of least effort, where cognitive systems default to strategies that minimize mental exertion. This is observable through two key interrelated concepts: perceived effort and effort discounting. Perceived effort is the subjective feeling of strain or difficulty associated with a mental task. Far from being a simple inconvenience, it functions as a critical regulatory signal, a real-time neuro-cognitive cost function. This feeling of effort signals that a task is metabolically demanding and encourages the system to disengage or switch to a less costly strategy [157]. The behavioral consequence of this cost signal is effort discounting. In decision-making, humans consistently devalue rewards that require greater cognitive or physical effort to obtain. Given a choice, we often prefer a smaller, easily attainable reward over a larger reward that requires significant mental work, effectively “discounting” the value of the larger reward by its associated effort cost [158]. Empirical studies have shown that individuals will actively choose to perform tasks with lower cognitive demand, even when the potential reward is identical, highlighting an intrinsic aversion to unnecessary mental exertion [13]. Therefore, subjective feelings of effort and the tendency to avoid it are not signs of laziness, but rather highly adaptive mechanisms that ensure an organism’s finite cognitive and metabolic resources are allocated efficiently and judiciously.

2.6 Effect Prediction in Robotics and its Applications

In robotics, a typical setup comprises a robot that interacts with its environment, resulting in an observable effect for the executed action. By learning a forward model, the

mapping from the current state of the environment (e.g., sensor readings, object poses) and a robot action to its resulting effect, the robot can anticipate the consequences of its actions before execution. This capability, known as effect prediction, is central to modern robotics because accurate and well-calibrated predictions enable safe planning and informed decision making in complex, dynamic settings.

Research aimed at effect prediction in robotics can be categorized along multiple dimensions. One distinct way to classify the methods is the *type of input data* utilized. A variety of approaches rely solely on low-dimensional state representations such as robot joint angles [159, 160] or object poses [161], while others incorporate high-dimensional sensory data, including images, tactile signals, or the multi-modal fusion of vision and touch [162, 163]. Related to this distinction, methods can also be classified based on the *learning architecture or model type*. Classical model-based or physics-informed approaches explicitly encode dynamics through analytical or simulation models [164]; data-driven methods employ various machine learning models, such as Convolutional Neural Networks for spatial features [162, 165], Recurrent Neural Networks for temporal sequence prediction [166, 167], transformers for capturing longer-range dependencies [168], or Generative Adversarial Networks to represent multi-modal action-effect distributions [169]. Hybrid models seek to combine physics priors with learned corrections to improve generalization [170]. Additionally, a number of studies utilize uncertainty quantification to provide reliability in the system [171] or to enhance sample efficiency in data gathering process [161].

A further key dimension for categorizing effect prediction methods concerns the underlying *learning paradigm* adopted by each approach. While supervised learning approaches train directly on labeled state-action-effect tuples [15, 166], self-supervised learning includes guidance to the robot’s actions based on the current knowledge of the task to provide more accurate learning [161, 172].

Finally, methods can be categorized by their *application domain or task focus*, such as grasp outcome prediction [172], deformable object modeling [173, 174], human-robot interaction [175, 176], or trajectory forecasting [177, 178]. For instance, vision-based effect prediction combined with tactile inputs is especially effective for manipulation tasks involving complex contact dynamics [179], while temporal sequence models excel in predicting evolving states in navigation or collaborative scenarios [180, 181].



3. HUMAN INSPIRED MACHINE LEARNING

Earlier sections discussed MTL and CL, highlighting their strengths along with the key differences from human learning. CL typically follows a fixed and sequential task learning order to minimize catastrophic forgetting, while MTL handles multiple tasks in parallel but isn't suited for lifelong learning.

The motivation in this thesis comes from a basic yet evident insight: human learning does not follow a strict sequence or learn several tasks at once. It is a dynamic, self-guided process. We switch between tasks seamlessly, influenced by various factors such as curiosity, task difficulty, effort, and our perceived rate of improvement. As discussed in the Section 2.3, the experiments shows that this interleaved practice helps with long-term retention and improves our ability to distinguish among different problem contexts.

The method we propose, which can be categorized as *human-inspired* machine learning, departs from the existing methods in terms of learning schedules. For a truly autonomous, lifelong learning agent, the choice of *what* to learn and *when* to switch tasks should come from the learning system itself, not from external supervision or pre-programmed schedules. Our work presents a framework where the learning schedule is flexible, since the agent dynamically decides when to shift tasks, using task-independent, intrinsic signals. This leads to an interleaved learning system that better reflects how humans learn.

We implement our approach through a novel multi-task learning architecture that enables knowledge transfer between tasks and the key element of this framework is an arbitration mechanism that automatically determines which task to focus on at any time. Below, the proposed human inspired learning is introduced in two parts:

- Our initial conference paper [182] set the foundation by proposing a model that contradicts the fixed or random task sequencing scenario. The system consists of

task-specific and shared sub neural networks for each task. The main contribution is a task arbitration mechanism motivated by Learning Progress (LP) [183], which measures learning rate. By choosing the task with the highest current LP, the agent concentrates on tasks that are just challenging enough to be engaging, avoiding those that are too easy or too hard. This resulted in an interleaved schedule, showing that allowing the agent to select tasks autonomously can lead to faster learning and better knowledge transfer than random sequenced multi-task learning, and superior than single-task learning.

- The journal paper is built on the previous idea, advancing the architecture into a more unified encoder-decoder structure with a shared attention module, allowing skill transfer between tasks. More importantly, another principle inspired by biology is integrated: neural energy efficiency. Since the human brain optimizes for energy conservation, we adjusted the LP signal based on the neuronal activation (i.e., energy) used for each task. This new combined measure, namely Energy-Modulated Learning Progress (EMLP) [15], creates a balance, dictating the agent to consider both high learning gains (LP) and the energy consumption (EC). This enhanced model shows competitive or even same learning performance with LP-only model while consuming much less energy, offering a flexible framework that can change its learning strategy according to the costs and benefits of each task.

In summary, our proposed method gives a learning agent the freedom to create its own learning path in real-time. By combining intrinsic motivation and cognitive energy costs, we develop an interleaved multi-task learning system that is more adaptable, efficient, and realistic than existing methods. The following sections will explain the specific architecture, the mathematical formulation of our arbitration mechanisms, and the experimental findings that support this approach.

4. LEARNING PROGRESS GUIDED MULTITASK LEARNING

This chapter details the methodological approach undertaken in our conference paper [182]. The next section first explains the architecture of this model, outlining its core components and how they enable interleaved multi-task learning. We then describe how learning progress is measured and used to guide task selection and knowledge transfer within the system.

4.1 Architectural Design

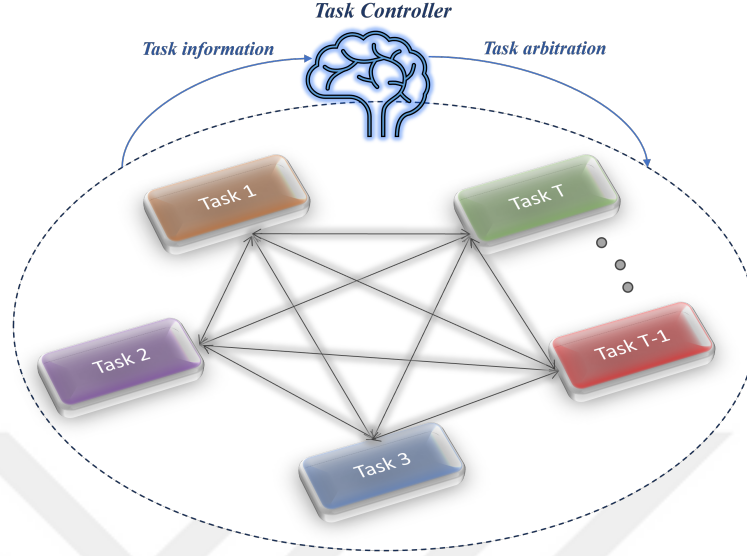
In this section, we present the architectural components of our proposed framework designed to support human-like, interleaved multi-task learning. Our goal is to enable autonomous task selection and modular knowledge transfer across tasks. We begin by detailing the overall structure of the interleaved multi-task learning framework, which forms the foundation of the system.

4.1.1 *Interleaved Multi-Task Learning Framework*

The model is built upon two core mechanisms: the arbitration of learning among tasks and the method for knowledge transfer between them:

1. **Task Controller for Learning and Arbitration:** The proposed architecture is distinguished from existing lifelong learning models by two cognitively inspired features: Firstly, it incorporates a learning arbitration mechanism that can switch between tasks at any time, which results in an interleaved learning schedule. Secondly, the neural networks assigned to each task are interconnected, which allows for knowledge to be transferred between any pair of tasks in any direction. As shown in the model’s conceptual diagram in Figure 4.1, a high-level controller oversees task-related data and decides which task the agent should focus on for learning.

Figure 4.1: High level task controller suitable for interleaved multi-task learning and ultimately continual learning.



This selection process can be implemented using an intrinsic motivation (IM) based approach, allowing for autonomous and dynamic task selection.

2. **Neural Network Design for Bidirectional Transfer:** At the core of the task-learning level is a specialized neural network architecture designed to facilitate the transfer of knowledge between different task learners. Each task is assigned a neural network that is composed of three distinct computational blocks: a *Core* block, an *Auxiliary* block, and a *Shared* block. The operational flow is as follows: both the Core and Auxiliary blocks process input and then project their outputs to the Shared block, which in turn computes the final output for the task. A crucial feature of this design is that the Auxiliary block of one task's network has access to the Core blocks of the other task networks. This connectivity is governed by a knowledge transfer function, denoted as $h(\cdot)$, which defines the specific topology for how knowledge is passed from the Core block of one task to the Auxiliary block of another. To formalize the operations within the neural circuit, we introduce the

following notations:

- **Core Block:** This is a sub-network composed of N nonlinear layers. The activation of the i -th layer for task t is denoted by $C_i^{(t)} \in \mathbb{R}^{n_i}$ where n_i is the number of neurons in that layer. The trainable parameters for layer i are represented by $\Theta_i^{(t)}$.
- **Auxiliary Block:** This sub-network consists of M nonlinear layers. The activations of the j -th layer for task t is denoted by $A_j^{(t)} \in \mathbb{R}^{n_j}$ where n_j is the number of neurons. Its trainable parameters for layer j are denoted as $\Theta_j^{(t)}$.
- **Shared Block:** This sub-network comprises the final R layers of the task's network. The hidden activation of the k -th layer for task t is represented by $S_k^{(t)} \in \mathbb{R}^{n_k}$ with n_k neurons and trainable parameters $\Theta_k^{(t)}$.
- **Knowledge Transfer Function $h(\cdot)$:** This function defines the specific connections used to transfer knowledge from the Core block of one task to the Auxiliary block of another. It determines the topology of the lateral connections between the individual task networks.

Based on these architectural definitions, the forward pass for a given task t , is mathematically defined as follows: The process begins with the computation of activations in the Core and Auxiliary blocks, which both receive the same initial input for task t ($C_0^{(t)} = A_0^{(t)}$). The hidden activation of the i -th layer in the Core block is calculated as follows:

$$C_i^{(t)} = f(W_i^{(t)} C_{i-1}^{(t)}), \quad 1 \leq i \leq N \quad (4.1)$$

Here, f represents the nonlinear activation function, and $W_i^{(t)} \in \mathbb{R}^{n_i \times n_{i-1}}$ is the

weight matrix for the i -th layer of the Core block. The hidden activation for the j -th layer in the Auxiliary block integrates information from other tasks:

$$A_j^{(t)} = f(W_j^{(t)} A_{j-1}^{(t)} + \sum_{\substack{z=1 \\ z \neq t}}^T h(C^{(z)})), \quad 1 \leq j \leq M \quad (4.2)$$

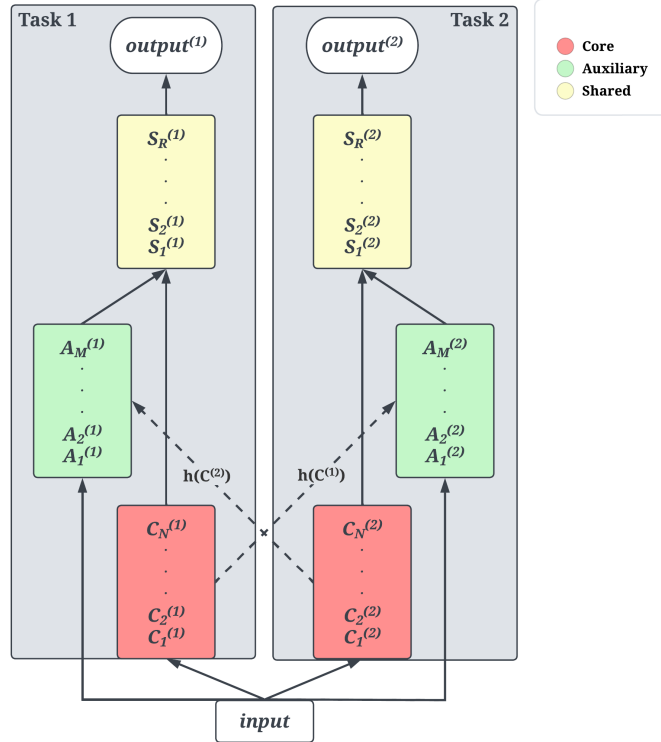
In this equation, $W_j^{(t)} \in \mathbb{R}^{n_j \times n_{j-1}}$ is the weight matrix for the j -th layer of the Auxiliary block. The key component is the summation term, where the knowledge transfer function, $h(\cdot)$ processes the outputs of the Core blocks, $C^{(z)}$, from all other tasks (where z represents another task and T is the total number of tasks). This sum is then added to the weighted input from the Auxiliary block's own previous layer before the activation function is applied. Finally, the outputs of the Core and Auxiliary blocks are combined to produce the final prediction. The activation of the Shared block's first layer, $S_1^{(t)}$, is computed by concatenating the final outputs of the Auxiliary block ($A_M^{(t)}$) and the Core block ($C_N^{(t)}$). The subsequent activations in the Shared block are then calculated to produce the final output, $S_R^{(t)}$:

$$S_k^{(t)} = f(W_k^{(t)} S_{k-1}^{(t)}), \quad 2 \leq k \leq R \quad (4.3)$$

This training procedure is specifically designed to facilitate this cross-task skill transfer. When the system is actively learning a specific task, t , the input data for that task is simultaneously fed into the neural networks of all other tasks. A forward computation is then carried out across all networks. This process makes the internal feature representations from other tasks available to task t via the knowledge transfer function $h(\cdot)$. A crucial element of the learning process is the selective updating of network parameters. During the training of task t , gradient-based updates are exclusively applied to the weights belonging to its own Core, Auxiliary, and Shared

blocks, in addition to the lateral connection weights that define the transfer function h . Critically, the internal parameters of the blocks belonging to all other tasks remain **frozen** and are not modified. This method ensures that task t can leverage the knowledge from other tasks without catastrophically interfering with the information they have already learned. An illustrative diagram of this network architecture for a two-task scenario ($T = 2$) can be found in the Figure 4.2.

Figure 4.2: *Proposed architecture containing the Core, Auxiliary and Shared blocks for each task, allowing bidirectional skill transfer between tasks, regardless of the training schedule.*



4.1.2 Implementation Details: LP Measure and Knowledge Transfer

This section outlines how the abstract model architecture is concretely implemented for the experiments conducted in the study. The implementation details cover the task arbitration mechanism and the specific topology of the knowledge transfer function.

4.1.2.1 Learning Progress-Based Task Arbitration

To realize the “Task Controller” responsible for task selection, we implemented a mechanism based on the concept of Learning Progress (LP). Following Schmidhuber’s formalization [183], LP is defined as the slope of the filtered prediction error over a recent time window. Denoting the prediction error for task t at a given epoch i as $E_t(i)$, LP is computed via linear regression over a window of size K (set to 5 in the experiments). The calculation is given by $LP_t(i) = Y_i \cdot X_i^T$, where $X_i = [i, (i - 1), \dots, (i - K + 1)]$ and $Y_i = [E_t(i), E_t(i - 1), \dots, E_t(i - K + 1)]$. The task exhibiting the highest LP is then selected for the next learning epoch.

4.1.2.2 Knowledge Transfer Function

The knowledge transfer function, $h(\cdot)$, determines the precise topology by which features are shared between task networks. While various connection schemes are possible, our study implemented a specific structure for its experiments:

For each task, direct layer-to-layer lateral connections were established from the Core blocks of all other tasks to its own Auxiliary block. In this configuration, the number of layers in the Core and Auxiliary blocks were set to be equal (i.e., $N = M$). This specific connection scheme modifies the general computation for the Auxiliary block’s hidden activation (previously shown in Equation 4.2) to the following:

$$A_j^{(t)} = f \left(W_j^{(t)} A_{j-1}^{(t)} + \sum_{\substack{z=1 \\ z \neq t}}^T U_j^{(t:z)} C_{j-1}^{(z)} \right) \quad (4.4)$$

Here:

- f is the ReLU activation function, defined as $f(x) = \max(0, x)$. This activation function is also used for the Core and Shared blocks.
- W_j^t remains the weight matrix for the j -th layer of the Auxiliary block for task t .

- $U_j^{(t;z)}$ is the weight matrix for the lateral connection from the $(j - 1)$ -th layer of the Core block of task z to the j -th layer of the Auxiliary block of task t .
- $C_{j-1}^{(z)}$ is the hidden activation from the $(j - 1)$ -th layer of the Core block of task z .
- T is the total number of tasks.

4.1.3 Task Simulation and Training Settings

The validation of the proposed model is conducted in a simulated robotic environment. This section details the simulation platform, the specific learning tasks assigned to the robot, and the methodology for data generation and handling.

4.1.3.1 Simulation Environments for Tasks

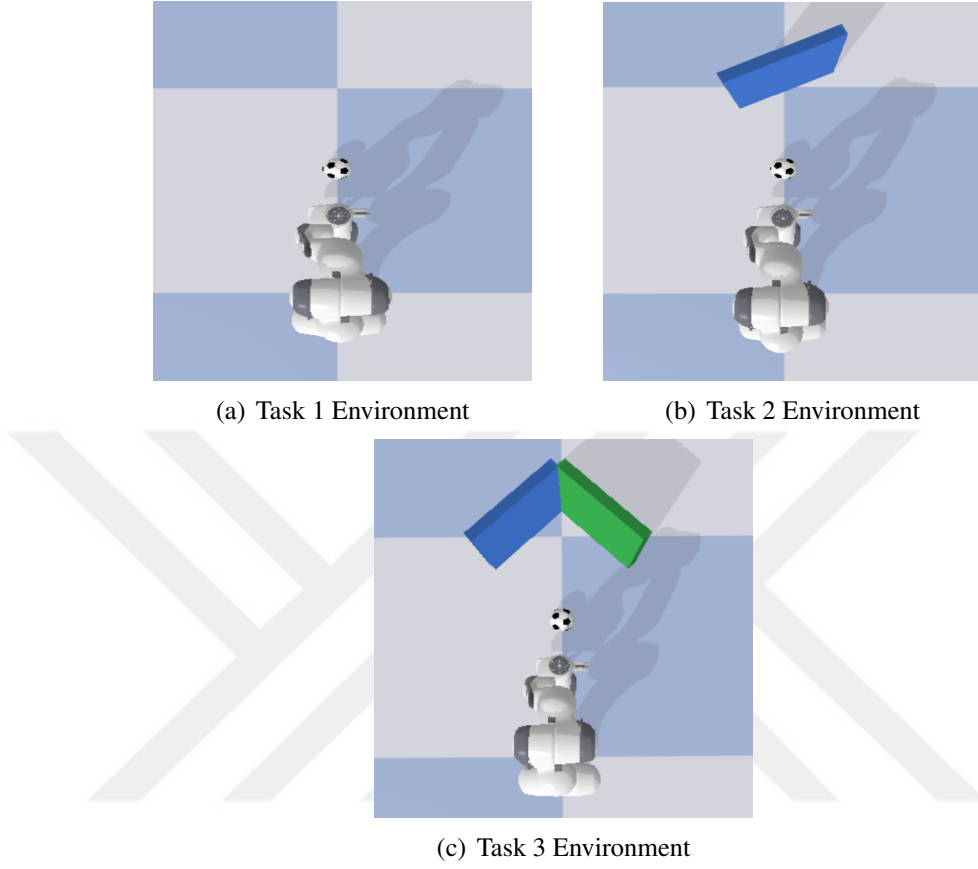
The experiments were executed using the Pybullet [184] physics simulation module. A Franka Emika Panda robotic arm is positioned on an empty plane within the simulated world. For all tasks, the environment contained the robot and a soccer ball, which served as the object of interaction. In tasks requiring an obstacle, a geometric box is placed in the environment, as shown in Figure 4.3.

4.1.3.2 Task Settings

The robot’s overall objective is to learn how to predict the effects of its own actions in three different environmental contexts. The robot is programmed to explore and learn this effect-prediction skill using the proposed model architecture. The three distinct tasks involved pushing the ball in different tabletop settings:

- **Task 1:** Pushing the ball on a table with no obstacles present (Figure 4.4(a)).
- **Task 2:** Pushing the ball with a single wall present, whose orientation is randomly changed within the range of $[-\pi/2, \pi]$ (Figure 4.4(b)).

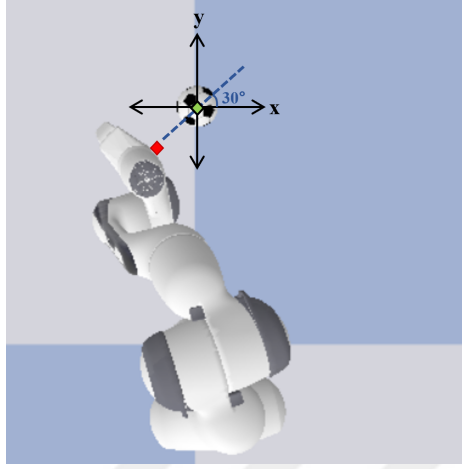
Figure 4.3: *Effect prediction environments created in PyBullet simulation. Franka Emika Panda robotic arm is used as manipulating agent.*



- **Task 3:** Pushing the ball with a fixed, L -shaped wall on the front (Figure 4.4(c)).

The robot's action is defined by a single parameter, an *angle* where $0 < angle < \pi$, which represents the degree between an imaginary line extending from the end-effector and the environment's x-axis. Based on this chosen angle, the robot's end-effector executes a push by moving through two consecutive via-points: The end-effector would first move to a point 20 cm behind the ball and then to the ball's current location to perform the push. An example of an action angle and its corresponding via-points is illustrated in Figure 4.4.

Figure 4.4: Following the action parameter, the robot navigates through red and green markers, with a 30-degree example shown.



4.1.3.3 Data Collection for Training and Testing

The robot generated its own data in an online, self-supervised fashion. The data handling and training procedure were structured as follows:

- **Experience Buffer:** Each task is allocated a limited experience buffer with a capacity of 500 samples. These buffers operate on a First-in First-Out (FIFO) basis. Initially, the simulation is run until all task buffers were filled.
- **LP Bootstrapping:** To initialize the learning progress (LP) computation, the training started with a bootstrapping phase where the robot trained on each task for five epochs. Following this phase, the robot began to autonomously select which task to engage with based on the LP signal.
- **Data Structure:** Each experience sample is structured as an *(action, context)* pair leading to an *effect*:
 1. The **action** is the hitting angle, which is provided to the network as its sine and cosine values to handle discontinuities.

2. The **context** is the random orientation of the wall for Task 2 and a constant zero for the other tasks where it is irrelevant.
 3. The **effect** is defined as the position of the ball 200 milliseconds after contact.
- **Testing Data:** To evaluate the generalization performance of the learned models, 37% of the possible integer-valued action angles were randomly selected and reserved for a test set. The robot is not permitted to select these actions during its exploration and training phase.

Overall training procedure with task arbitration mechanism is given in Algorithm 1.

Algorithm 1 Implemented task learning and arbitration

Require: Number of tasks T , experience buffer size N , number of pre-train m , number of epochs E

```

1:  $X_t = []$  // FIFO experience buffer for task  $t$  with size  $N$ 
2: for task  $t = 1$  to  $T$  do
3:    $X_t \leftarrow N$  interaction samples
4:   for epoch  $i = 1$  to  $m$  do
5:     add 20 interaction samples into  $X_t$  // FIFO
6:     train  $t$  with  $X_t$ 
7:   end for
8:   compute  $LP_t$  // Initial learning progress for task  $t$ 
9: end for
10: for epoch  $i = 1$  to 1500 do {A slice of lifelong learning}
11:    $k = \arg \max_t (\{LP_t\}_{t=1}^T)$  // Pick the task with maximum LP
12:   add 20 interaction samples into  $X_k$  // FIFO
13:   train task  $k$  with  $X_k$ ; compute  $LP_k$ 
14: end for

```

4.2 Experiment Settings

To assess the efficacy of the proposed learning model, a series of experiments were conducted comparing its performance against several ablated baseline models. This

section describes the baseline models used for comparison and the specific parameters of the neural network training protocol.

4.2.1 *Baselines*

To investigate the distinct contributions of the interleaved task selection and the knowledge transfer mechanism, three baseline models were designed for comparison against the proposed model (referred to as **LP Con**).

1. **Rand No Con**: This is the first baseline, which features the same neural network architecture but with no connections established between the task networks. In this model, task selection is performed in a random order, representing a scenario with no intelligent task arbitration and no knowledge transfer.
2. **Rand Con**: The second baseline utilizes the same network architecture and knowledge sharing topology as the proposed LP Con model. However, the tasks are learned according to a fixed, randomly determined sequence instead of being selected based on Learning Progress (LP). This baseline is analogous to conventional Continual Learning (CL) approaches where the task order is predetermined.
3. **LP No Con**: The final baseline is identical to the Rand No Con model in that it has no connections between task networks. However, its task selection pattern is set to be identical to the one autonomously generated by the LP Con model in its corresponding experimental run. This baseline was designed to isolate the benefits of knowledge transfer from the effects of simply allocating more training epochs to certain tasks.

4.2.2 *Training*

A consistent training protocol was applied to the proposed model and all three baselines to ensure a fair comparison:

- *Network Structure*: For all models, the neural networks were structured with $N = 4$ layers for the Core block, $M = 4$ layers for the Auxiliary block, and $R = 1$ layer for the Shared block.
- *Training Duration*: The training phase for every model consisted of 1500 epochs.
- *Optimization*: The ADAM optimization algorithm was used with a learning rate of 0.0001. Input data was fed to the models in batches of 100.
- *Experimental Runs*: To ensure the reliability of the results, the proposed model and the baselines were each trained 10 times with different random seeds. For the baselines operating on a fixed task sequence (Rand No Con and Rand Con) 10 different random task sequences were generated, one for each of the 10 experimental runs. In the LP Con model, the Task Controller autonomously selected the task to be trained at each epoch.

4.2.3 Evaluation Protocol

To quantitatively measure and compare the performance of the proposed model against the baselines, a set of evaluation metrics and statistical tests were defined:

Average Task Performance: The primary metric for assessing performance on a task was the effect prediction error. This error was calculated as the difference between the model’s predicted final position of the object and the actual position observed in the simulation after the robot’s action. For comparison purposes across models, this was converted into the Mean Absolute Error (MAE) for each task.

Statistical Significance Analysis: To determine the appropriate statistical test for comparing model performances, we first examine two key assumptions of parametric testing: normality and homogeneity of variances. We first assess the normality using the Shapiro–Wilk [185] test, which evaluates whether the sample distribution significantly

deviates from a normal distribution. Homogeneity of variances is tested using Levene’s [186] test, which checks whether the variance across groups is statistically equal. Both tests are evaluated at a significance level of $\alpha = 0.05$. Based on the outcomes of these assumption checks, we decide whether a parametric test can be applied. If both assumptions are satisfied, we proceed with a standard independent samples t-test [187]. If the assumption of equal variance is violated while normality holds, we instead use Welch’s t-test [188], which is more robust to unequal variances.

4.3 Results

In this section, we present the experimental results of our proposed LP Con model, focusing on various performance metrics across multiple tasks and training configurations. These results aim to demonstrate the effectiveness of learning progress-based task selection in interleaved multi-task settings. We begin by comparing the overall performance of our model against baseline methods.

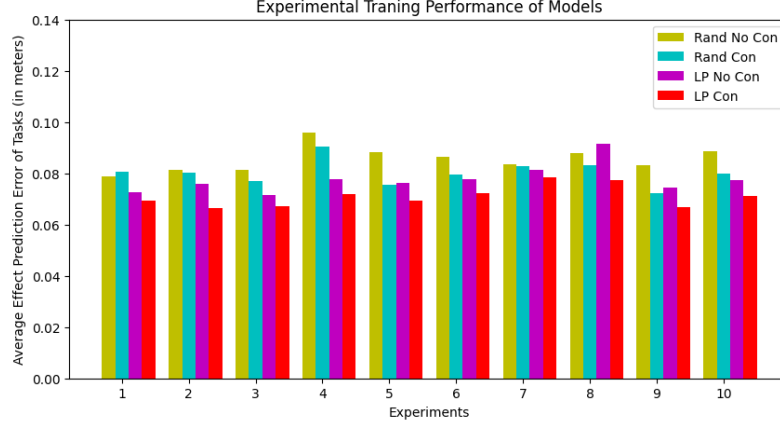
4.3.1 Comparative Model Performance

This section presents the results of the comparative experiments conducted between the proposed model (LP Con) and the three baseline models (Rand No Con, Rand Con, and LP No Con). The experiments, involving a simulated robot learning three different tasks, were repeated ten times to ensure the reliability of the findings.

4.3.1.1 Individual Session Performance

Figure 4.5 shows the mean absolute error (MAE) for every model across the different learning sessions performed. The results from the training clearly indicate that our model (LP Con) consistently records lower error rates than the other baseline models in all learning sessions. Moreover, the Rand Con baseline, which incorporates a mechanism for knowledge transfer between tasks, outperformed the Rand No Con baseline that

Figure 4.5: Mean Absolute Error (MAE) of task training performances for both the baseline models and the proposed model (LP Con).



lacks inter-task connections. This suggests that promoting knowledge sharing among the specialized task networks enhances overall learning performance, even when tasks are selected randomly. However, the best outcomes were observed when tasks could be selectively switched based on their LP (see LP Con in Figure 4.5). Additionally, LP No Con consistently exhibits poorer performance compared to LP Con, implying that the quantity of learning iterations each task goes through is not the only element that contributes to improved learning performance.

4.3.1.2 Overall Performance and Statistical Significance

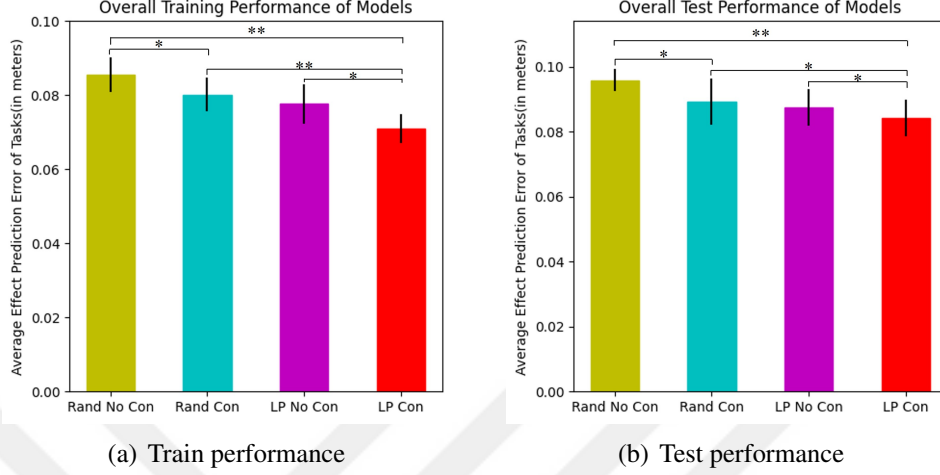
Figure 4.7(a) illustrates the mean and standard deviation of the average task errors on the training dataset, collected over ten independent runs for each model. Our proposed model consistently yields lower prediction errors compared to all baselines, with nearly identical standard deviations across models. Similarly, the test dataset results shown in Figure 4.7(b) reveal that our model maintains superior average performance, although it exhibits slightly higher variance than the baselines. To validate whether the observed difference between our proposed model and each baseline is statistically significant, first Shapiro-Wilk normality test is applied for both training and test results, with $p = 0.05$

set as threshold. All comparisons passed the normality check indicating no significant deviation from normality (for training $p = 0.17$, $p = 0.1$, $p = 0.17$ for Rand No Con, Rand Con and LP No Con, respectively and testing $p = 0.27$, $p = 0.16$, $p = 0.14$ for Rand No Con, Rand Con and LP No Con, respectively). Next, we assessed the equality of variances using Levene’s test. In most comparisons, the assumption of homogeneity of variances was violated. Specifically, both LP Con vs Rand No Con and LP Con vs Rand Con showed significantly unequal variances for both training ($p = 0.0001$ and $p = 0.0005$, respectively) and test data ($p = 0.0000$ and $p = 0.0005$, respectively). Only the comparison between LP Con and LP No Con met the equal variance assumption ($p = 0.82$ for training, $p = 0.89$ for testing). Since all pairwise comparisons passed the normality assumption but two violated the equal variance assumption, we applied the Welch’s t-test as it does not require equal variances. We conducted significance comparisons using two thresholds: $p = 0.05$ and $p = 0.0001$, providing a stronger confidence margin where applicable. Overall, as reported in the Figures 4.7(a) and 4.7(b), the results confirm that our model’s performance improvements over the baselines are statistically significant in both the training and testing phases, thereby validating the effectiveness of the proposed approach.

4.3.1.3 Task Selection Patterns by LP Guided Task Controller

It is enlightening to observe how the proposed task selection algorithm automatically alternates tasks for each successive experimental session. The variation in the random seed with each iteration is anticipated to generate distinct task selection patterns due to the stochastic nature of neural network weight initialization and robot action execution. The pattern identified in the conducted trials is displayed in Figure 4.7. We can clearly say that Task 2 and Task 3 received much higher selection frequency compared to Task 1, in all experiments.

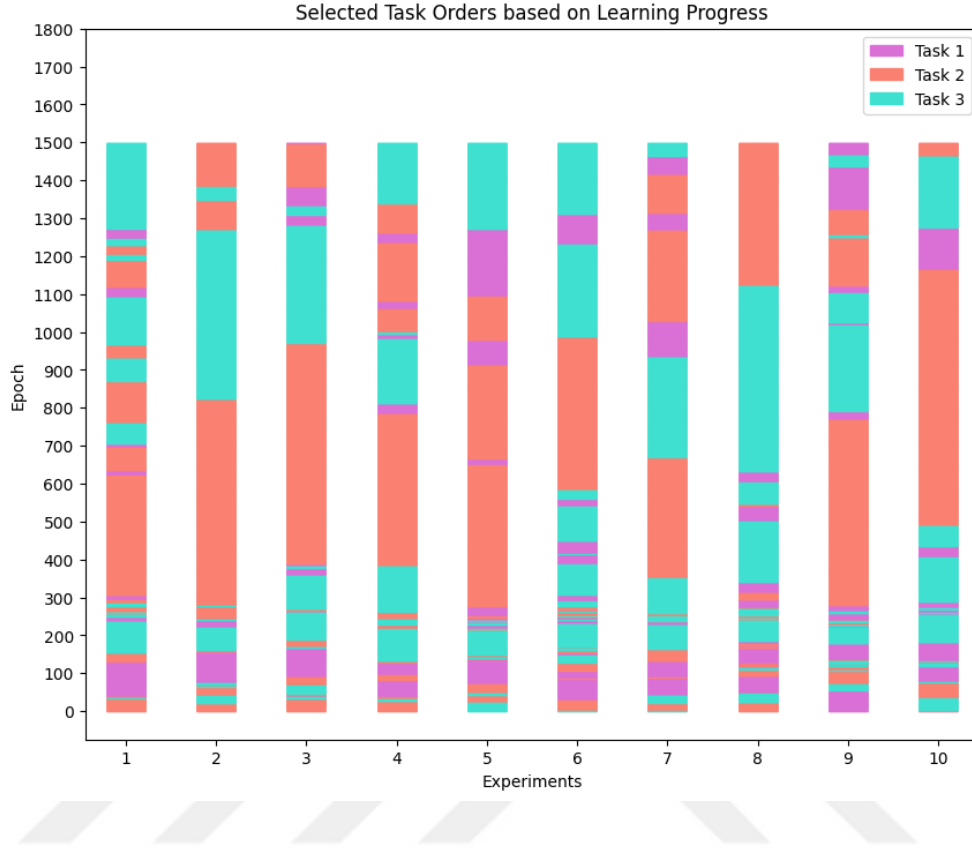
Figure 4.6: Mean average task errors for training data (a) and testing data (b) over 10 repetitions. The standard deviation is denoted by vertical lines on the bars, and the symbols * and ** denote statistical significance at $p = 0.05$ and $p = 0.0001$, respectively.



4.4 Discussion

The proposed model not only achieved superior performance on the tasks evaluated compared to the baselines, but also exhibited notable task switching dynamics. To better understand the reason behind the task selection patterns in Figure 4.7, we further analyzed both the task-wise prediction errors and the LP values of them throughout the training process in Figure 4.8. We can clearly see that Task 1 showed the worst learning performance compared to Task 2 and Task 3 (see Figure 4.9(a)). When we inspect the prediction error trend of Task 1, we can see that it shows rapid learning in the early stages of training (until ≈ 50 epochs) with high LP values, similar to the other tasks. After that period, Task 1 stagnates and does not reduce its error until the end of the training, showing almost no learning after the initial phase. Due to not reducing the error, LP becomes nearly zero ($\approx 10^{-6}$), as can be seen in Figure 4.9(b). Even though other tasks also show plateaued prediction error trend after early training phase, they still accomplish minimizing their task errors in first two hundred epochs to much lower MAE values than the Task 1. Similarly, their LP values are much higher than the Task 1 ($\approx 0.01 - 0.0001$) from ≈ 50

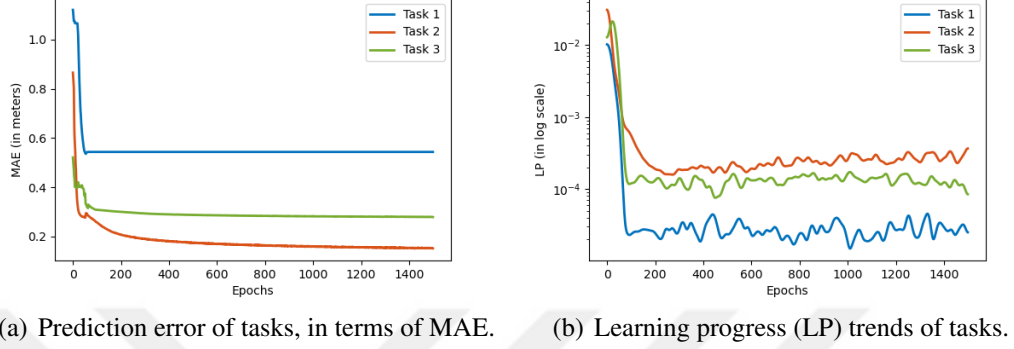
Figure 4.7: Task learning schedules dictated by the LP based Task Controller.



epochs to the end of the training. One can argue that when an obstacle is present in front of the ball (Task 2 and Task 3), it restricts the ball’s potential movement after the robot executes the pushing action, primarily due to possible reflections. In contrast, in environments without obstacles (Task 1), the ball can move freely in any direction, resulting in a significantly broader range of possible final positions. This increased variability in the ball’s final location introduces greater uncertainty, which can affect the learning process.

In addition to interleaved learning of tasks, this study presented results specifically for the multi-task learning scenario; however, the proposed model is adaptable for continual learning scenarios. It achieves this by incorporating additional neural network blocks for new tasks and facilitating knowledge transfer from previously learned tasks through new lateral connections. Furthermore, LP was utilized as the “task information”,

Figure 4.8: Visualization of learning performance and LP trends of tasks by the proposed model LP-Con. Task 1 shows the worst performance while having the lowest LP trend. On the other hand, Task 2 and Task 3 converge faster than the Task 1 and show higher LP trends.



serving as the primary metric for task arbitration. However, there are various alternatives available, including information gain, novelty, and task competence, that could also be considered [148, 189]. The logical progression of this research involves evaluating the model within continual and continual-multi task learning contexts, aiming to identify the most effective Task Information signal for lifelong learning. A promising avenue for research involves the development of methods aimed at dynamically establishing inter-task connections. This approach seeks to enhance knowledge transfer performance while simultaneously minimizing the consumption of neural resources. To demonstrate the scalability of the proposed model, a logical next step would be to evaluate the model across a wider range of diverse and complex task settings. This would involve implementing a progressively expanding task set throughout the learning agent’s life cycle, which our model is designed to handle without necessitating significant structural modifications.

4.5 Conclusion

This section proposes the novel computational model for lifelong learning [182], focusing on the requirement that a learning agent autonomously determines which learning task to pursue and when to transition to another task to enhance knowledge transfer in a continual multi-task learning environment. This novel approach facilitates task acquisition in alignment with developmental learning principles and recognizes the benefits of an interleaved learning strategy, consistent with experimental findings and behavioral studies [116, 117, 118, 119]. The model is implemented in a simulated robotic tabletop setup, where the robot is tasked with learning the consequences of its actions. To evaluate the learning performance of the proposed model, three ablated models serve as baselines. The initial baseline consisted of independent task learning in a random sequence, whereas the subsequent baseline facilitated information transfer between tasks without a task arbitration mechanism. The third baseline sought to demonstrate that high learning performance is not attributable to the number of learning iterations assigned to each task. The experiments demonstrated that our model significantly outperformed the three baselines in both training and testing phases. To our knowledge, this model is the first interleaved multi-task learning system that integrates a bidirectional knowledge transfer mechanism for lifelong learning. Our model demonstrates a closer alignment with human developmental learning compared to existing MTL and CL approaches, rendering it more suitable for realistic lifelong learning environments that feature partial learning, reciprocal knowledge transfer, arbitrary task order, and multiple revisits to partially learned tasks.

5. ENERGY MODULATED LEARNING PROGRESS GUIDED MULTITASK LEARNING

Building upon the framework introduced in the preceding chapter, which was based on our conference publication [182], this chapter presents a significant extension to the task arbitration mechanism. The initial work established a novel interleaved learning model that autonomously selects tasks based on Learning Progress (LP), demonstrating improved performance through a more cognitively plausible training schedule. This chapter details the methodology developed for our subsequent journal article [15]. The primary contribution of the work detailed in this chapter is the integration of a bio-inspired energy conservation principle into the task selection process. Motivated by the human brain's tendency to manage cognitive effort and optimize for energy efficiency, we introduce a measure of neural computational energy expenditure. This metric is used to modulate the LP signal, resulting in a novel, combined task arbitration mechanism. This approach seeks not only to prioritize tasks with high learning gains but also to factor in the computational cost of achieving that progress, leading to a more efficient and sustainable learning process.

The following sections will provide a comprehensive description of this extended framework. First, the multi-task neural network architecture, which features shared and task-specific modules to enable skill transfer, will be revisited and detailed. Following this, the formulation of the Energy Modulated Learning Progress (EMLP) criterion will be thoroughly explained, clarifying how learning progress and energy consumption are combined to guide the interleaved task learning schedule.

5.1 Methods

In this section, we introduce the methodology behind our extended model, IMTL-EMLP, which incorporates energy modulation into the task selection process. Building on the LP Con framework, this approach aims to balance learning efficiency with computational cost. We begin by formally defining the multi-task learning problem setting addressed in this study.

5.1.1 Multi-Task Learning Problem Statement

The fundamental problem this section addresses is the development of a task arbitration and skill transfer mechanism capable of producing improved learning and transfer performance within a multi-task learning context. The framework is designed to emulate a key aspect of human learning, where an individual must decide *which* task to engage with and for *how long*, leading to the emergence of flexible, interleaved learning schedules. Our focus is on supervised learning challenges within robotics, where an agent must learn from its interactions. We seek to create an online multi-task model that incorporates a task arbitration mechanism, allowing a robot to intelligently select which task to sample data from and train on, with the ultimate objective of minimizing the average prediction error across all tasks. Formally, the problem is defined as follows. Given a set of m distinct tasks, denoted as $\{\mathcal{T}_i\}_{i=1}^m$, each task is allocated an interaction period of $N_{\mathcal{T}_i}$. Upon the completion of this period for a given task, the arbitration mechanism selects a new task to become active. This selected task, denoted by $\mathcal{T}_*$, is then used for execution, data collection, and model training. The experience gathered for task $\mathcal{T}_*$ is represented as a set of input-output pairs:

$$\mathcal{D}_{\mathcal{T}_*} = \{(x_j^{\mathcal{T}_*}, y_j^{\mathcal{T}_*})\}_{j=1}^{N_{\mathcal{T}_*}},$$

where $x_j^{\mathcal{T}^*}$ is the input and $y_j^{\mathcal{T}^*}$ is the corresponding output. After collecting a mini-batch of this experience, the model updates both its shared parameters, θ^s , and the parameters specific to the active task, $\theta^{\mathcal{T}^*}$. This cycle of online sampling and learning continues until the end of learning. If we formulate the prediction for task $\mathcal{T}_i$ as $f_{\mathcal{T}_i}(x_j^{\mathcal{T}_i}; \theta^s, \theta^{\mathcal{T}_i})$ and its associated loss as $\mathcal{L}_{\mathcal{T}_i}$, the global objective for the multi-task agent is to find the optimal set of shared and task-specific parameters that minimize the total loss, $\mathcal{TL}$, across all tasks:

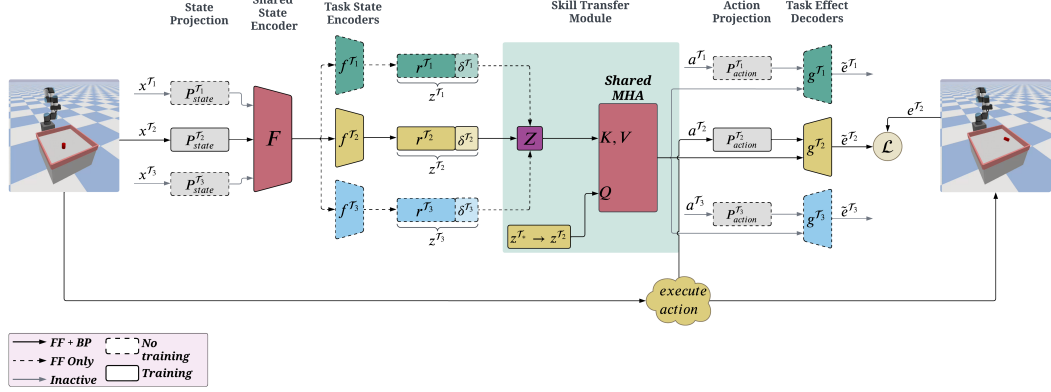
$$\mathcal{TL} = \min_{\theta^s, \{\theta^{\mathcal{T}_i}\}} \sum_{i=1}^m \mathcal{L}_{\mathcal{T}_i}(\{(f_{\mathcal{T}_i}(x_j^{\mathcal{T}_i}; \theta^s, \theta^{\mathcal{T}_i}), y_j^{\mathcal{T}_i})\}_{j=1}^{N_{\mathcal{T}_i}}) \quad (5.1)$$

Within the scope of the experiments conducted for our journal, this problem is concretized as learning a set of m distinct robotic effect prediction tasks. For each task $\mathcal{T}_i$, the model must learn a mapping from a state-action pair to its resulting effect. The state is represented by $x^{\mathcal{T}_i} \in \mathbb{R}^{d_s^{\mathcal{T}_i}}$ and the action by $a^{\mathcal{T}_i} \in \mathbb{R}^{d_a^{\mathcal{T}_i}}$. The outcome of performing action $a^{\mathcal{T}_i}$ in state $x^{\mathcal{T}_i}$ is the effect, denoted by $e^{\mathcal{T}_i} \in \mathbb{R}^{d_e^{\mathcal{T}_i}}$. Therefore, the experience dataset for learning is $\mathcal{D}_{\mathcal{T}_i} = \{((x_j^{\mathcal{T}_i}, a_j^{\mathcal{T}_i}), e_j^{\mathcal{T}_i})\}_{j=1}^{N_{\mathcal{T}_i}}$. The goal is to dynamically arbitrate among these tasks to learn a set of action-effect prediction functions, $f_{\mathcal{T}_i} : \mathbb{R}^{d_s^{\mathcal{T}_i}} \times \mathbb{R}^{d_a^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_e^{\mathcal{T}_i}}$. Upon successful training, for any given state-action pair $(x^{\mathcal{T}_i}, a^{\mathcal{T}_i})$, the predicted effect $\tilde{e}^{\mathcal{T}_i} = f_{\mathcal{T}_i}(x^{\mathcal{T}_i}, a^{\mathcal{T}_i})$ is expected to accurately approximate the actual ground-truth effect $e^{\mathcal{T}_i}$.

5.1.2 Proposed Neural Network Architecture

The proposed model is founded on an encoder-decoder network architecture, specifically designed to handle multiple tasks while facilitating information sharing between them. The architecture, visually summarized in Figure 5.1 of the journal paper, consists of several key components: task-specific input (state) and action projection layers, a shared state encoder, task-specific state encoders, a shared attention module that acts as

Figure 5.1: Proposed neural network architecture for multi-task effect prediction. $\mathcal{T}_* = \mathcal{T}_2$



a skill transfer hub, and finally, task-specific output (effect) decoders. All symbols and notations used to describe the architecture are defined in Table 5.1 for clarity.

5.1.2.1 State and Action Projection

To handle inputs from different tasks that may vary in dimensionality, the first step involves projecting them into a common, fixed-dimensional space:

- **State Projection:** For each task $\mathcal{T}_i$, the state input $x^{\mathcal{T}_i}$ is passed through a task-specific, linear projection function $P_{state}^{\mathcal{T}_i}$. This standardizes the dimensionality of state vectors across all tasks for subsequent shared processing. The projected state vector, $\hat{x}^{\mathcal{T}_i}$, is computed as:

$$\hat{x}^{\mathcal{T}_i} = P_{state}^{\mathcal{T}_i}(x^{\mathcal{T}_i}) \quad (5.2)$$

- **Action Projection:** Similarly, since action vectors $a^{\mathcal{T}_i}$ can also differ in size between tasks, each is passed through a task-specific linear projection function $P_{action}^{\mathcal{T}_i}$. This ensures that action representations are consistent and compatible for later stages of

Table 5.1: Overview of symbols and notations.

Symbol	Description
m	Total number of tasks
i	Task index, $\{\mathcal{T}_i\}_{i=1}^m$
$\mathcal{T}_*$	Selected task to be trained
$x^{\mathcal{T}_i}$	State of task $\mathcal{T}_i$, $x^{\mathcal{T}_i} \in \mathbb{R}^{d_s^{\mathcal{T}_i}}$
$a^{\mathcal{T}_i}$	Action of task $\mathcal{T}_i$, $a^{\mathcal{T}_i} \in \mathbb{R}^{d_a^{\mathcal{T}_i}}$
$e^{\mathcal{T}_i}$	Effect of task $\mathcal{T}_i$, $e^{\mathcal{T}_i} \in \mathbb{R}^{d_e^{\mathcal{T}_i}}$
$P_{\text{state}}^{\mathcal{T}_i}$	Task-specific state projection function, $P_{\text{state}}^{\mathcal{T}_i} : \mathbb{R}^{d_s^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_s}$
F	Shared state encoder, $F : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_h}$
h	Output of the shared encoder, $h \in \mathbb{R}^{d_h}$
$r^{\mathcal{T}_i}$	Task-specific latent representation, $r^{\mathcal{T}_i} \in \mathbb{R}^{d_r}$
$z^{\mathcal{T}_i}$	Task-specific latent representation with flag bit, $z^{\mathcal{T}_i} \in \mathbb{R}^{d_r+1}$
Z	Stacked matrix of all $\{z^{\mathcal{T}_i}\}_{i=1}^m$, $Z \in \mathbb{R}^{m \times (d_r+1)}$
$P_{\text{action}}^{\mathcal{T}_i}$	Task-specific action projection function, $P_{\text{action}}^{\mathcal{T}_i} : \mathbb{R}^{d_a^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_a}$
$f^{\mathcal{T}_i}$	Task-specific encoder for task $\mathcal{T}_i$, $f^{\mathcal{T}_i} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_r}$
$g^{\mathcal{T}_i}$	Task-specific decoder for task $\mathcal{T}_i$, $g^{(t)} : \mathbb{R}^{d_r+d_a+1} \rightarrow \mathbb{R}^{d_e^{\mathcal{T}_i}}$

the network. The projected action vector, $\hat{a}^{\mathcal{T}_i}$, is given by:

$$\hat{a}^{\mathcal{T}_i} = P_{\text{action}}^{\mathcal{T}_i}(a^{\mathcal{T}_i}) \quad (5.3)$$

The primary aim of these projection layers is to maintain uniform dimensions at the output; hence, a linear projection is employed in the journal, as is commonly practiced in the literature [104, 190].

5.1.2.2 Shared and Task-Specific Encoders

After projection, the state information is processed through a combination of shared and specialized encoders to extract both general and task-unique features:

- **Shared State Encoder:** The projected state vector $\hat{x}^{\mathcal{T}_i}$ is first fed into a single shared encoder, F . This module is designed to learn and extract general features that

are beneficial across all tasks, thereby promoting knowledge sharing and reducing model redundancy. It outputs a shared feature vector, h , as follows:

$$h = F(\hat{x}^{\mathcal{T}_i}) \quad (5.4)$$

- **Task State Encoders:** The shared feature vector h is then passed through a set of parallel, task-specific encoders, $f^{\mathcal{T}_i}$. Each encoder is dedicated to capturing unique state information that the shared encoder might not extract, allowing the model to learn specialized features for each task. This generates a task-specific latent representation, $r^{\mathcal{T}_i}$, for every task:

$$r^{\mathcal{T}_i} = f^{\mathcal{T}_i}(h), \quad \forall i \in \{1, \dots, m\} \quad (5.5)$$

Crucially, during training, backpropagation only updates the parameters of the encoder corresponding to the currently selected task, $f^{\mathcal{T}_*}$, while all other task-specific encoders remain frozen.

5.1.2.3 Multi-Head Attention Module

To enable communication between tasks and allow the network to intelligently leverage knowledge from other tasks, we employ a shared Multi-Head Attention (MHA) [191] mechanism. Instead of performing a single attention function, the MHA layer computes attention across H parallel "heads". This allows the model to jointly attend to information from different representation subspaces. For each head, the Query (Q), Key (K), and Value (V) inputs are first projected into a lower-dimensional space using learnable linear projection matrices. For each head $i \in \{1, \dots, H\}$, the projections are:

$$Q_i = Q W_i^Q, \quad K_i = K W_i^K, \quad V_i = V W_i^V. \quad (5.6)$$

Each head then computes the scaled dot-product attention output independently:

$$\text{head}_i = \text{softmax} \left(\frac{Q_i K_i^\top}{\sqrt{d_k}} \right) V_i. \quad (5.7)$$

where d_k is the dimensionality of the keys per head. Finally, the outputs from all H heads are concatenated and passed through a final output projection matrix, W^O , to produce the final attention output, A.

$$A = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_H)W^O. \quad (5.8)$$

5.1.2.4 Task-Focused Attention Implementation

Our implementation constrains the attention mechanism to focus on the perspective of the currently active task. The process is as follows:

1. **Flagging Task Representations:** Before the attention step, each task-specific latent representation, $r^{\mathcal{T}_i}$, is augmented with a binary flag, $\delta^{\mathcal{T}_i}$. This flag is set to 1 if the task is currently being trained ($\mathcal{T}_i = \mathcal{T}_*$) and 0 otherwise. This creates a final task representation, $z^{\mathcal{T}_i}$:

$$z^{\mathcal{T}_i} = [r^{\mathcal{T}_i} : \delta^{\mathcal{T}_i}] \quad (5.9)$$

2. **Forming the Representation Matrix:** The final representations for all tasks are stacked together to form a matrix $Z \in \mathbb{R}^{m \times (d_r + 1)}$, which serves as the context for the attention mechanism.

$$Z = \begin{bmatrix} z^{\mathcal{T}_1} \\ z^{\mathcal{T}_2} \\ \vdots \\ z^{\mathcal{T}_m} \end{bmatrix} \quad (5.10)$$

3. **Query, Key, and Value Formulation:** To guide the attention mechanism, the Query (Q), Key (K), and Value (V) inputs are defined specifically based on the active task $\mathcal{T}_*$. The **Query** is the representation of the currently active task, as we want the information retrieval to be driven by its needs. The **Key** and **Value** are set to the entire matrix Z , allowing the active task to attend to representations from all tasks, including itself.

- Query: $Q = z^{\mathcal{T}_*}$
- Key: $K = Z$
- Value $V = Z$

4. **Attention Output:** These Q, K, and V inputs are fed into the shared MHA module to produce a task-aware output vector, $A^{\mathcal{T}_*}$, for the active task.

$$A^{\mathcal{T}_*} = \text{SharedMHA}(Q, K, V) \quad (5.11)$$

This configuration forces the network to focus on how other tasks' representations can inform the active task, with the flag bits providing explicit cues about the origin of the information.

5.1.2.5 Task Effect Decoders

Finally, the information processed by the attention module is used to predict the outcome of an action. For this, we utilize task-specific decoders, $g^{\mathcal{T}_i}$. For the currently active task $\mathcal{T}_*$, its specific decoder receives the output from the shared MHA mechanism, $A^{\mathcal{T}_*}$, concatenated with the projected action vector, $\hat{a}^{\mathcal{T}_*}$. The decoder then maps this combined state-action context to the corresponding effect prediction, $\tilde{e}^{\mathcal{T}_*}$.

$$\tilde{e}^{\mathcal{T}_*} = g^{\mathcal{T}_*}([A^{\mathcal{T}_*} : \hat{a}^{\mathcal{T}_*}]) \quad (5.12)$$

5.1.3 Task Arbitration

To achieve an interleaved learning regime that mimics human cognitive flexibility, we depart from traditional multi-task learning approaches that rely on fixed or random task schedules. Instead, we propose a dynamic task arbitration mechanism that continuously evaluates the learning trend of each task and selects which one to train next based on these assessments. This strategy aims to emulate the human tendency to switch between tasks based on perceived progress and effort. The general process is outlined in Algorithm 2.

At each decision point, a score is computed for every task based on one of the criteria described below. A non-greedy selection policy is then employed; with a high probability $(1 - \epsilon)$, the task with the highest score is selected, but with a small exploration probability, ϵ , a random task is chosen instead. This exploration prevents the model from over-relying on a single task that is performing well, ensuring all tasks receive some attention.

Algorithm 2 Interleaved Multi-Task Learning

Require: Number of tasks m , number of epochs R , exploration rate $\epsilon = 0.1$

```

for epoch = 1 to  $R$  do
  Compute score  $s^{\mathcal{T}_i}$  for tasks  $\{\mathcal{T}_i\}_{i=1}^m$ 
   $i' \leftarrow \arg \max_i (\{s^{\mathcal{T}_i}\}_{i=1}^m)$ 
  Generate a random number  $r$  uniformly from  $[0, 1]$ 
  if  $r < \epsilon$  then
     $\mathcal{T}_* \leftarrow$  Random task from  $\{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_m\} \setminus \{\mathcal{T}_{i'}\}$ 
  else
     $\mathcal{T}_* \leftarrow \mathcal{T}_{i'}$ 
  end if
  Perform training on the selected task  $\mathcal{T}_*$ 
end for

```

5.1.3.1 Learning Progress Only Task Selection

The foundational task selection strategy is based on Learning Progress (LP), a measure of recent performance improvement. The LP for a given task $\mathcal{T}_i$ is calculated by analyzing its error history over the last L time steps. Specifically, a linear model is fitted to the recent error points $\{E_{t-L+1}^{\mathcal{T}_i}, \dots, E_t^{\mathcal{T}_i}\}$ to estimate the slope of the error trend, denoted as $\beta^{\mathcal{T}_i}$.

A negative slope ($\beta^{\mathcal{T}_i} < 0$) indicates that the error is decreasing, signifying positive learning progress. Conversely, a non-negative slope ($\beta^{\mathcal{T}_i} \geq 0$) suggests that learning has stalled or regressed. The LP is formally defined as the magnitude of the negative slope, and zero

otherwise:

$$LP^{\mathcal{T}_i} = \begin{cases} |\beta^{\mathcal{T}_i}|, & \text{if } \beta^{\mathcal{T}_i} < 0 \\ 0, & \text{otherwise} \end{cases} \quad (5.13)$$

Under this scheme, the arbitration mechanism selects the task with the highest LP value to train next ($\mathcal{T}_* = \arg \max_i (LP^{\mathcal{T}_i})$). This strategy ensures that the model focuses its resources on tasks where additional training is likely to yield the most significant gains, rather than continuing to work on tasks that have stagnated.

5.1.3.2 Energy Modulated Learning Progress (EMLP) Selection

While LP effectively guides the model toward progressing tasks, it does not account for the computational cost required to achieve that progress. To address this and introduce a notion of learning efficiency, we propose an extension that modulates the LP score with a measure of energy consumption. We define the Energy Consumption ($EC^{\mathcal{T}_i}$) for a task as the total activation output of all neurons belonging to its specific modules, accumulated over the last L training steps:

$$EC^{\mathcal{T}_i} = \sum_{j=(t-L)}^t \mathcal{A}_j^{\mathcal{T}_i}. \quad (5.14)$$

A lower $EC^{\mathcal{T}_i}$ value signifies that the task has recently been trained with high neural efficiency. To avoid a situation where the model only chooses low-energy tasks at the expense of performance, we do not use this metric alone. Instead, we discount the LP score by the energy cost to create a combined score, $s^{\mathcal{T}_i}$. Before combination, both the LP and EC values are normalized to a $[0, 1]$ scale using a min-max scaler. The combined score is calculated as:

$$s^{\mathcal{T}_i} = e^{(k \cdot \tilde{LP}^{\mathcal{T}_i})} / \tilde{EC}^{\mathcal{T}_i} \quad (5.15)$$

Here, $\tilde{LP}$ and $\tilde{EC}$ represent the scaled values, and k is a positive constant that controls the sensitivity to energy consumption. A high value of k makes the score dominant by

LP, approximating the previous selection method, while a low value of k places more emphasis on minimizing energy expenditure.

Under this EMLP-based strategy, the system selects the task with the highest combined score ($\mathcal{T}_* = \arg \max_i (s^{\mathcal{T}_i})$). This ensures a dynamic balance between achieving rapid gains in performance and maintaining overall energy efficiency, favoring tasks that are both improving quickly and are not computationally expensive.

5.1.4 *Experimental Settings*

To empirically validate the effectiveness of the proposed Interleaved Multitask Learning with Energy Modulated Learning Progress (IMTL-EMLP) framework, a series of systematic experiments were conducted. This section details the experimental setup, the tasks performed by the agent, the baseline models used for comparison, and a thorough analysis of the results. The evaluation was performed within a simulated robotic environment where the agent was required to learn the effects of its actions across different task settings. The primary metrics for performance assessment were learning efficiency, as measured by convergence speed, and final task accuracy. Our evaluation strategy begins by first assessing the benefits of the core task arbitration method, IMTL-LP, which uses only learning progress (LP) for task selection. This model is compared against several key baselines to isolate the impact of the proposed interleaved scheduling. The baselines include: (1) SINGLE, where each task is trained with its own separate network; (2) IMTL-RAND, a multi-task model using a shared architecture with a randomly selected task per iteration; and (3) BLOCK, where tasks are trained sequentially without interruption (in all six possible orders) to assess the impact of task scheduling.

Following this, we evaluate the full IMTL-EMLP framework, which incorporates the neural energy metric into the arbitration logic. This analysis includes exploring how varying the energy sensitivity coefficient, k , allows for a trade-off between prediction

accuracy and computational cost. To ensure a fair comparison, a consistent model architecture was used across all baselines and proposed models, guaranteeing that any observed performance differences are attributable solely to the task arbitration strategy. Finally, the evaluation is supplemented with additional studies to analyze the contributions of individual model components and skill transfer dynamics.

5.1.4.1 Simulation Environment

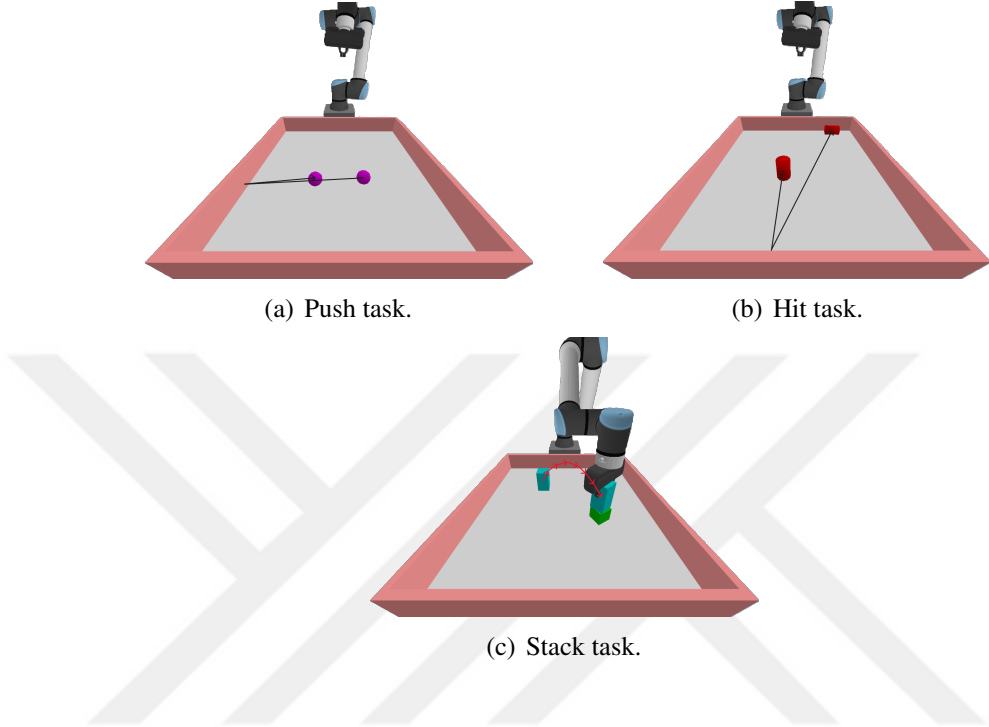
All experiments were conducted using a UR10 robotic arm equipped with a Franka Panda end-effector, simulated within the PyBullet physics engine [184]. The experimental workspace consists of a tabletop environment where the robot interacts with a set of primitive objects. The set of objects includes a sphere (radius = 3 cm), a cube (length = 6 cm), a square prism (width = 6 cm, height = 12 cm), and a cylinder (radius = 3 cm, height = 18 cm). The prism and cylinder can be placed in either a vertical or a horizontal orientation, leading to a total of six unique object configurations that the robot can interact with. For each interaction episode, the relevant object is placed at a random position on the table.

5.1.4.2 Effect Prediction Tasks

To test the proposed model, we defined three distinct tasks for the robotic agent, each requiring it to predict the effect of its actions in different scenarios:

1. Push Task: The first task is a fundamental “Push” task. A single object is randomly placed on the table, and the robot applies a pushing action to the object’s center of mass. The push is executed with an angle chosen from a range of $[0, 180]$ degrees.
 - State (s): The state of the object is its pose, defined by its Cartesian coordinates (x, y, z) and its orientation in Euler angles (ϕ_x, ϕ_y, ϕ_z) . To avoid discontinuities in the neural network input, the orientation angles are represented as

Figure 5.2: Visual examples of tasks executed in the PyBullet simulation environment. Arrows in Push (a) and Hit (b) show the trajectory of the object after the action is executed. For Stack (c), arrows show the end-effector trajectory while executing the stacking action.



sine and cosine pairs. Resulting 9-dimensional state vector for the object is defined as follows:

$$\mathbf{s} = [x, y, z, \sin(\phi_x), \cos(\phi_x), \sin(\phi_y), \cos(\phi_y), \sin(\phi_z), \cos(\phi_z)]. \quad (5.16)$$

- Action (a): The action is encoded by combining the sine and cosine of the push angle with a one-hot vector that identifies the object being pushed:

$$\mathbf{a} = [\sin(\theta), \cos(\theta), \text{onehot}(o)]. \quad (5.17)$$

- Effect (e): The effect is defined as the change (Δ) between the object's pre-action and post-action state vectors, capturing the displacement and change in

orientation:

$$\mathbf{e} = [\Delta(x), \Delta(y), \Delta(z), \Delta(\sin(\phi_x)), \Delta(\cos(\phi_x)), \Delta(\sin(\phi_y)), \Delta(\cos(\phi_y)), \Delta(\sin(\phi_z)), \Delta(\cos(\phi_z))]. \quad (5.18)$$

2. Hit Task: The second task is a variant of the Push task, referred to as the "Hit" task.

The setup is identical, but the action is applied with twice the velocity. This results in less predictable object behavior, such as multiple bounces off the table walls. The definitions for the state, action, and effect vectors remain identical to the Push task.

3. Stack Task: The final task requires the robot to place one object on top of another to form a potentially stable stack. This task involves the interaction of two objects: a "picked" object and a "target" object. Given the six object configurations, this task results in 36 possible object-pair combinations. The state, action and effect are redefined as below:

- State (s): The state includes the 9-dimensional states of both the picked (s_p) and target (s_t) objects, denoted as $s = [s_p, s_t]$.
- Action (a): The action is represented by concatenating the one-hot encoding of the picked object (o_p) and the target object (o_t).
- Effect (e): Similarly, the effect is computed as the change in the objects' pre-action and post-action states, represented as $e = [e_p, e_t]$.

5.1.4.3 Baselines

To evaluate the proposed framework, we first examine the properties of the learning progress-based model (IMTL-LP), which is a special case of our full model. We then analyze the complete Energy Modulated Learning Progress model (IMTL-EMLP). The performance of these models is compared against several baselines designed to isolate the effects of multi-task learning and our intelligent interleaving strategy. We also assess

different variants of the IMTL-EMLP model with varying energy sensitivity coefficients (k) to analyze the trade-off between predictive performance and energy consumption. The following baseline models were used to assess the advantages of our proposed task selection methods:

- **SINGLE:** Our first baseline for measuring the effects of shared representation is a learning model with only one task. In this setup, each task is trained on its own network, which has no parameters shared with other tasks. We used the same architectural design (see Figure 5.1) for this baseline as we did for our multi-task model, instead of just a simple fully-connected network, to make sure the comparison was fair. This means that differences in performance can only be linked to the learning strategy (isolated vs. multi-task) and not to differences in architecture.
- **IMTL-RAND:** The second baseline employs the shared architecture from our proposed technique without alterations but lacks the LP-based task selection mechanism. Tasks are selected randomly throughout each training iteration, with each task possessing an equal probability of selection. This baseline emphasizes the effect of our task selection (interleaving) technique by contrasting it with a scenario in which the interleaving is entirely random. This baseline lets us distinguish the efficacy of the scheduling method from the advantages of shared parametrization.
- **BLOCK:** This third baseline also uses the same network architecture, but it trains tasks in uninterrupted blocks instead of interleaving them. We investigated at all six alternative orders to learn the three tasks to see whether there were any impact on the overall training performance. In this case, if there are R total training epochs, each of the m tasks gets a block of R/m training iterations. This baseline lets us see if our interleaved method leads to better learning than any other sequential training regime.

- **IMTL-EMLP-K**: In the experiments using the proposed model IMTL-EMLP, we compare it with a number of variations of it, alongside the aforementioned baselines and the previously proposed IMTL-LP model. To do so, we investigate the impact of the sensitivity coefficient k on learning performance and energy consumption and compare IMTL-EMLP models trained with different k values: IMTL-EMLP-K=0.4, IMTL-EMLP-K=0.7, IMTL-EMLP-K=1, IMTL-EMLP-K=1.2.

5.1.4.4 Hyper-parameters and Initialization

All models are implemented with a consistent set of hyper parameters across both the baseline models and the proposed model. Specifically, a learning rate of 0.0001, a batch size of 100, a hidden layer dimension of 4, and a training duration of 3000 epochs are configured. Each encoder and decoder, whether shared across tasks or task-specific, comprises two fully connected layers (see further architectural details in Figures A.1 and A.2), with ReLU as the activation function. Model optimization is carried out using the AdamW optimizer [192], with the AMSGrad variant enabled and default weight decay parameters. To expedite training, robot-environment interactions are pre-collected, thereby eliminating the need to run the simulation at each interaction step. For this study, an experience cache of approximately $10K$ interactions is sufficient, as the tasks considered in the experiments feature low-dimensional state and action spaces.

5.2 Results

This section presents the experimental findings of our energy-modulated multi-task learning framework. We evaluate model performance across various dimensions and we begin by analyzing the results of the IMTL-LP variant, which uses learning progress alone as the task selection signal.

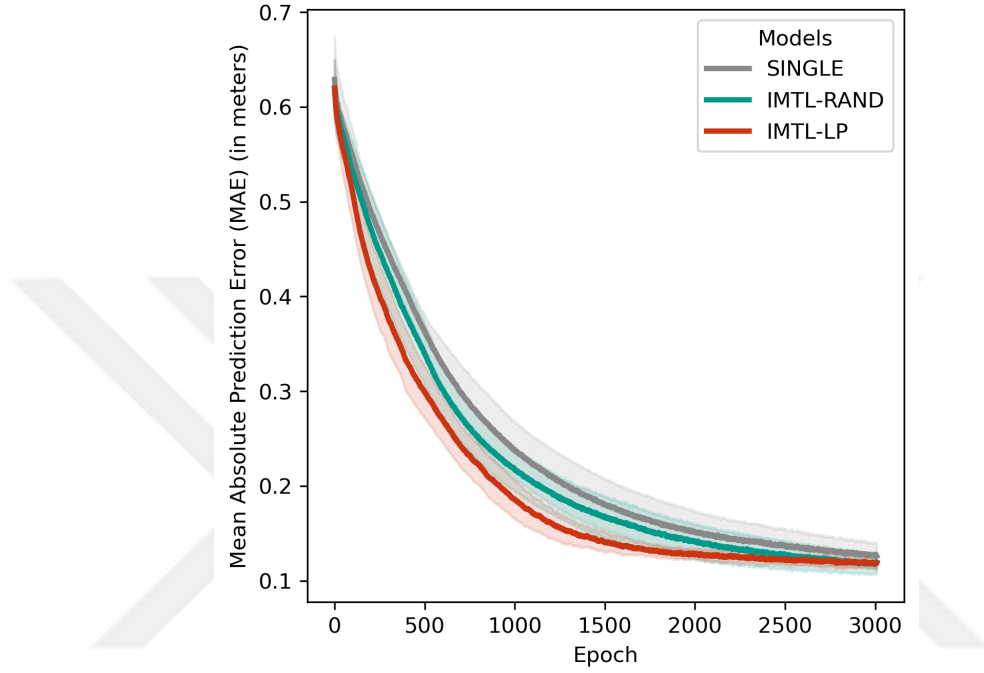
5.2.1 *IMTL-LP Only Results*

The first experiment involves a comparison of the proposed model without energy modulation, specifically comparing IMTL-LP with the SINGLE and IMTL-RAND baselines. The study involved conducting experiments with 10 random seeds, and the average performance results for the tasks are detailed in this section. Figure 5.3 demonstrates that the proposed model, IMTL-LP, exhibits a more rapid decrease in prediction error compared to the two baseline models. It suggests that the LP-based task selection method, which incorporates inter-task skill transfer, results in a more effective interleaved task schedule. This approach enhances overall performance when compared to both random selection and independent learning without skill transfer. Additionally, it is crucial to recognize that interleaving improves performance but also engaging with multiple tasks has demonstrated advantages. This results from the ability of tasks to benefit from one another and facilitate beneficial information exchange, with the enhancement noted in the IMTL-RAND baseline compared to the SINGLE baseline, serving as supportive proof.

5.2.2 *Analysis of Task-wise Performance*

To gain deeper insight into the learning dynamics, we analyze how each task performs over the course of training. As shown in Figure 5.4, learning all tasks jointly leads to improved results in both the Push and Stack tasks, with particularly strong gains from the proposed model. The Stack task, in particular, shows a notable performance gap in favor of our model, IMTL-LP, compared to the two baseline methods. For the Hit task, IMTL-LP demonstrates an early drop in prediction error, though IMTL-RAND becomes more competitive in the latter half of training. Table 5.2 details the number of iterations each task received through autonomous task selection. Interestingly, despite receiving less total training time, IMTL-LP still outperforms IMTL-RAND in the Stack task, suggesting that training time alone does not solely determine task performance.

Figure 5.3: The overall performance of the baselines is presented for isolated task learning (SINGLE), the architecture with random task selection (IMTL-RAND), and the proposed model utilizing the LP-based task selection (IMTL-LP). IMTL-LP demonstrates a faster convergence rate compared to the baseline models. While the random task selection demonstrates a marginal improvement over single task learning, it fails to exceed the performance achieved through LP-based task selection.



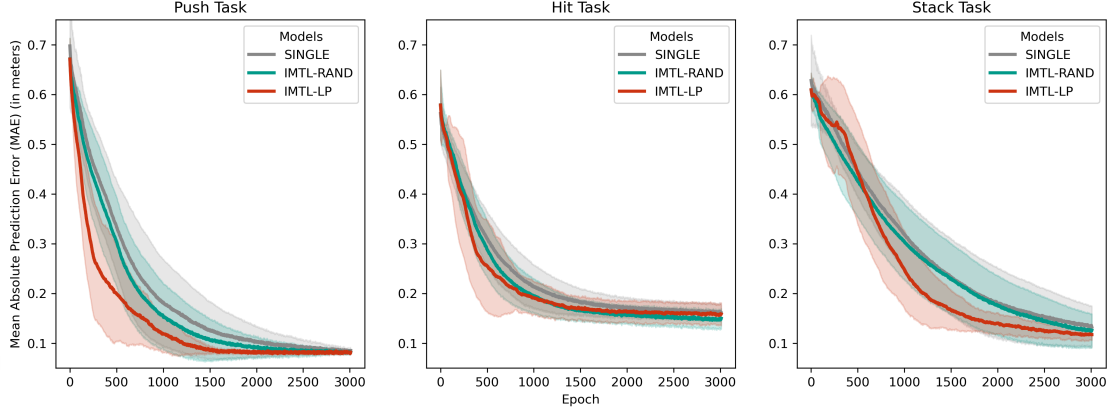
5.2.3 Network Complexity and Learning Performance Relationship

We examined how network complexity affects model performance by assessing SINGLE, IMTL-RAND, and our proposed IMTL-LP model (without energy modulation) across three complexity levels: low (800 parameters), medium (2000 parameters),

Table 5.2: The average total number of training iterations received by each task as the result of autonomous task selection in IMTL-LP is compared with uniformly random task selection (IMTL-RAND). The average and standard deviation is over 10 training sessions.

Model	Push	Hit	Stack
IMTL-RAND	1002.5 $\pm$ 24.4	997.7 $\pm$ 19.9	1008.8 $\pm$ 16.8
IMTL-LP	1525.6 $\pm$ 124.0	624.8 $\pm$ 93.2	850.6 $\pm$ 185.6

Figure 5.4: The performance of the models on specific tasks is presented. The LP-based method outperforms the other baseline approaches in both the push and stack tasks, with a more notable advantage in the stack task. However, for the hit task, its improvement was limited to the training period between approximately 400 and 1000 steps.

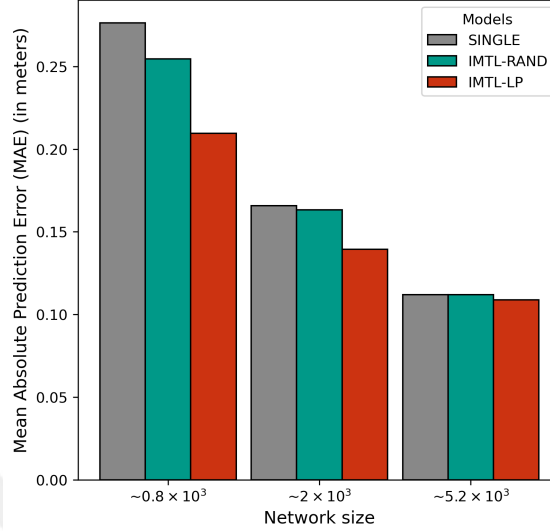


and high (5200 parameters), and the results are shown in Figure 5.5. At the low complexity level, IMTL-LP notably surpassed both baseline models, underscoring its effectiveness in resource-limited scenarios due to our strategy of interleaving based on learning progress. In the medium complexity scenario, IMTL-LP continued to exceed the baselines, although the performance gap was reduced, indicating that while SINGLE and IMTL-RAND gained advantages from the extra parameters, IMTL-LP preserved its edge through superior task management. At the high complexity level, the performance differences among the three models diminished further, showing that the increased capacity somewhat alleviated the constraints in baseline strategies. Nonetheless, IMTL-LP consistently remained either competitive or the top performer across all levels of complexity.

5.2.4 Interleaved Learning versus Blocked Learning

A critical question in multi-task learning is whether a dynamic, interleaved training schedule offers notable benefits over a more traditional blocked learning paradigm, where tasks are mastered sequentially in distinct, uninterrupted intervals. It could be argued that a specific, optimized sequence of blocked training might match or even surpass

Figure 5.5: *Comparative analysis of the learning performance of baseline models (SINGLE, IMTL-RAND) and our proposed model (IMTL-LP) across various network complexity configurations. Results are obtained at the midpoint of training.*

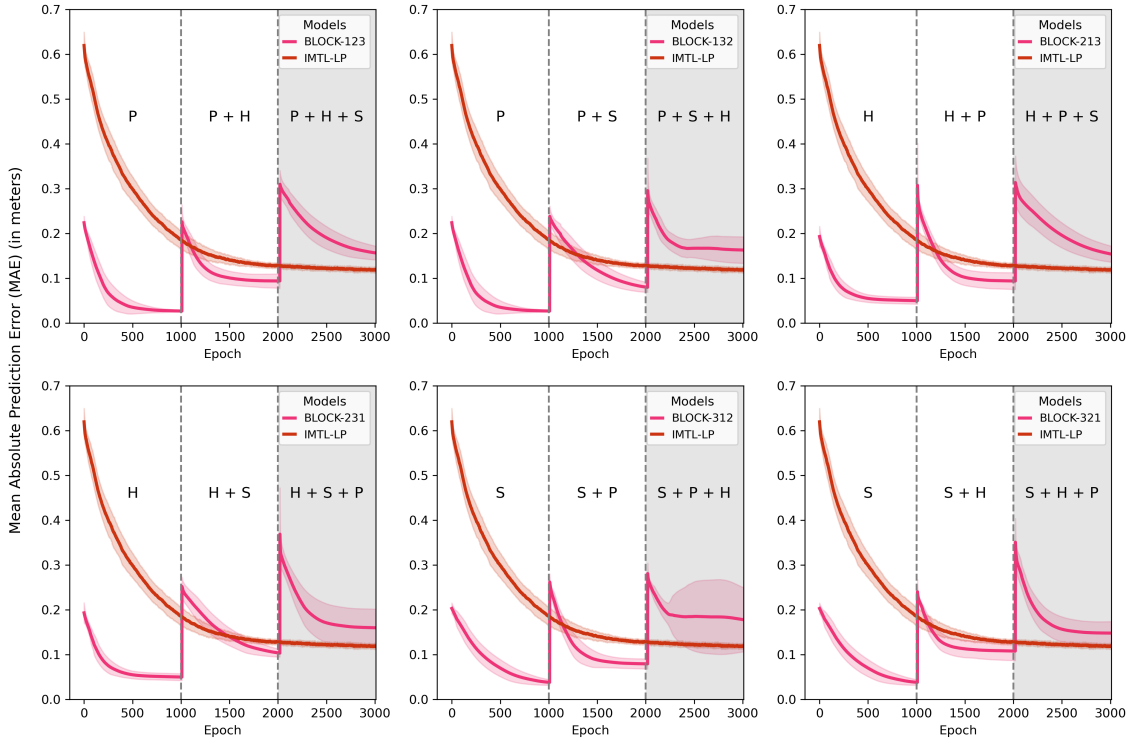


the performance of an interleaved approach. To rigorously investigate this, a detailed comparison was conducted between our proposed IMTL-LP model and a blocked learning framework. Given the three experimental tasks (Push, Hit, and Stack), there are six possible permutations for a sequential training order. To ensure a comprehensive and fair comparison, the BLOCK baseline model was trained and evaluated across all six of these fixed permutations (the task order is denoted with task ids as -XXX, e.g., BLOCK-012). In contrast, the IMTL-LP model, which autonomously determines the task sequence, does not depend on a predefined order and was therefore compared against each of these six scenarios.

The results, presented in Figure 5.6, obviously demonstrate that the proposed IMTL-LP model consistently outperforms the BLOCK model, regardless of the task order used in the blocked schedule. The underlying reason for this superior performance relates to the well-known problem of catastrophic forgetting, which is characteristic of sequential training. As the BLOCK model completes training on one task and moves to the next, it

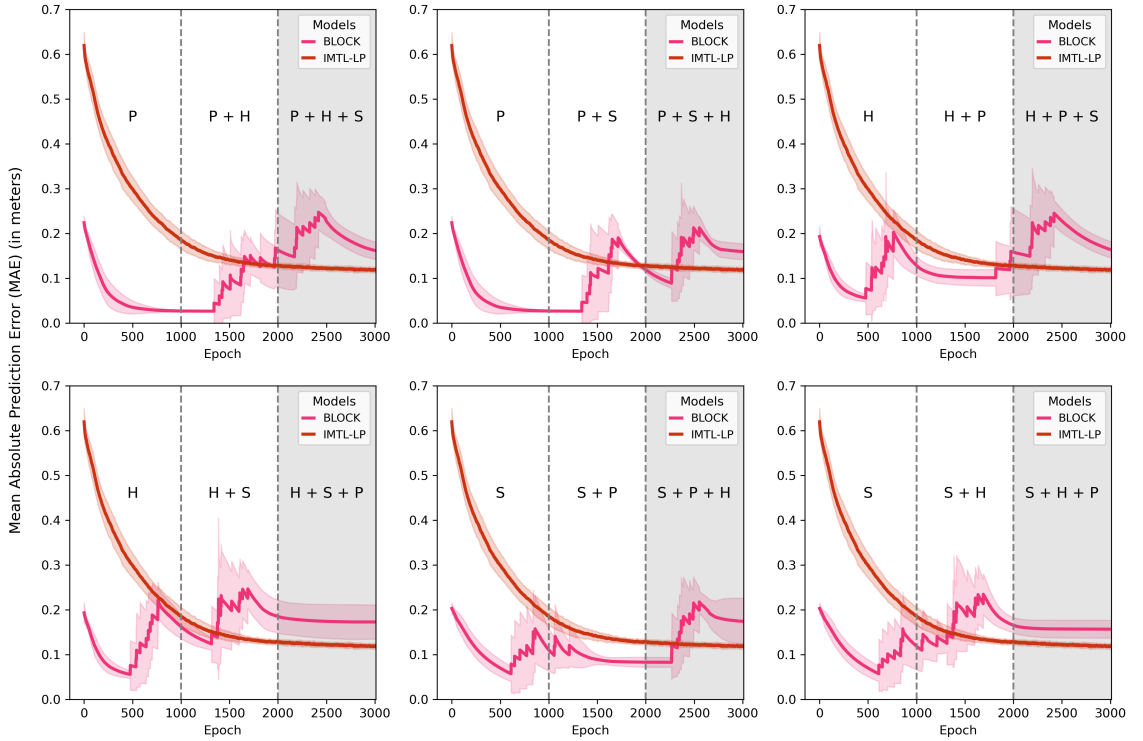
begins to forget the previously learned information, a phenomenon evidenced by the distinct error spikes that occur at the task-switch boundaries in each permutation plot. Our interleaved method, however, mitigates this issue. By periodically switching the context between tasks, the model is able to better retain information and recall it more efficiently, preventing the catastrophic forgetting observed in the blocked approach. Furthermore, the performance variability across the different permutations of the BLOCK model highlights its sensitivity to the training sequence. This reliance on a specific optimal order is a significant drawback that our adaptive, interleaved model overcomes. These findings confirm that the interleaved training regime generated by our framework is not just beneficial, but fundamentally more robust and effective than a sequential learning schedule.

Figure 5.6: Prediction error plots for each task order combination of blocked learning and interleaved learning with LP. Each figure has three sections split by vertical dashed lines where the BLOCK model swaps tasks, where *P* is Push, *H* is Hit, and *S* is Stack. The rightmost location in each figure shows where BLOCK and IMTL-LP have seen all the tasks to learn. IMTL-LP outperforms BLOCK learning in all scenarios.



A modified variant of the BLOCK baseline is implemented, namely BLOCK-LP, in which, rather than allocating an equal number of training iterations to each task (i.e., R/m), tasks receive a fixed number of training iterations equivalent to those allocated during training with the LP-based interleaving strategy. Consequently, the training allocation per task varies depending on the specific LP-based run and its associated task selection schedule. This experiment is designed to demonstrate that, even when the total training budget per task is held constant between the BLOCK and INTER-LP models, the absence of periodic revisits in the BLOCK setting leads to catastrophic forgetting. The Figure 5.7 These findings underscore the importance of interleaved learning for long-term retention of task-specific knowledge.

Figure 5.7: Prediction error plots for each task order combination of blocked learning (tasks are trained with equal number of epochs they have received in LP based interleaved training) and interleaved learning with LP. In this case, the switch from one task to another is variable, due to different task selections regimes by IMTL-LP. Our model, IMTL-LP, still outperforms BLOCK learning in all scenarios.



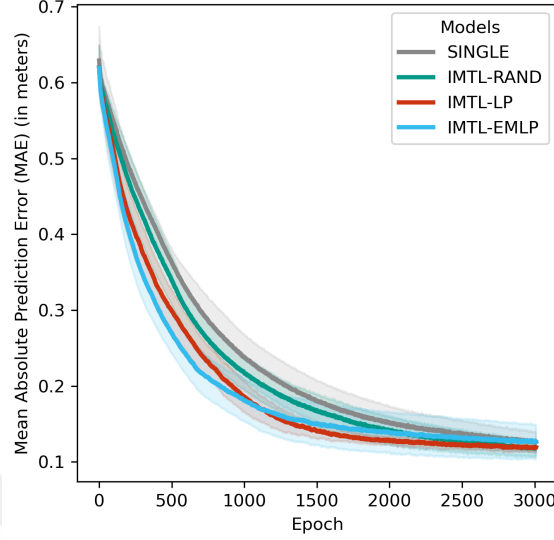
5.2.5 *Performance-Efficiency Trade-off: Modulating Learning Progress with Computational Energy*

Building on the demonstrated success of the learning progress-based interleaving strategy, we now introduce the full, proposed framework: Interleaved Multitask Learning with Energy Modulated Learning Progress (IMTL-EMLP). This extension is directly inspired by a fundamental principle of biological cognition: the brain’s remarkable energy efficiency and the human tendency to regulate cognitive effort. The IMTL-EMLP model operationalizes this principle by discounting a task’s learning progress by the computational energy required to achieve it, thereby creating a tunable balance between learning performance and resource consumption. To validate this energy-aware approach, we first evaluated the IMTL-EMLP model (see Equation 5.15) with a balanced energy sensitivity coefficient ($k = 1$) against the pure learning progress model (IMTL-LP) and the established baselines.

The results, shown in Figure 5.8, confirm the efficacy of incorporating an energy cost. The IMTL-EMLP model demonstrates a performance trajectory that is not only superior to the SINGLE and IMTL-RAND baselines but is also highly competitive with the IMTL-LP model, even showing a slight advantage in convergence speed during the initial 1000 epochs of training. After this initial phase, both the IMTL-LP and IMTL-EMLP variants converge to a similar, low prediction error, confirming that the introduction of energy awareness does not degrade the final performance of the system.

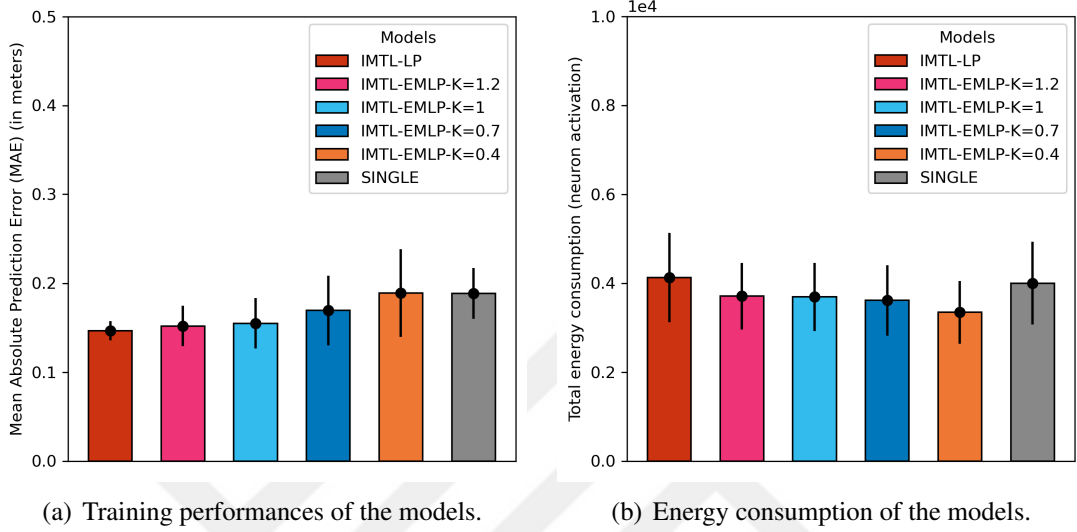
The central feature of the IMTL-EMLP framework is its ability to navigate the trade-off between predictive accuracy and energy expenditure. To explore this capability, the model was evaluated under varying levels of the energy sensitivity coefficient, k . Figure 5.9 presents the prediction error and total energy consumption at the midpoint of training for different k values. As shown in Figure 5.10(a) and Figure 5.10(b), there is a clear relationship between the prioritization of energy and predictive accuracy. The pure

Figure 5.8: *Evaluation of the baseline models and the LP-based model (IMTL-LP) against the EMLP-based model (IMTL-EMLP), with the sensitivity constant fixed at $k = 1$.*



IMTL-LP model, which does not consider energy cost, achieves the lowest prediction error overall. As the k coefficient is decreased, placing greater emphasis on energy conservation, the model’s prediction error gradually increases. Conversely, the opposite trend for energy consumption is exhibited. The IMTL-LP model is the most energy-intensive, while energy usage steadily decreases with lower values of k . These results highlight the flexibility of the framework. While maximizing performance (IMTL-LP) or minimizing energy (IMTL-EMLP- $K=0.4$) represent the extremes of this spectrum, the models with intermediate k values offer a highly favorable balance. For instance, the IMTL-EMLP- $K=1.2$ and IMTL-EMLP- $K=1$ variants achieve a notable reduction in energy consumption with only a minor sacrifice in prediction accuracy when compared to the IMTL-LP model. This tunable behavior is a key contribution of our work, equipping an agent with the ability to adapt its learning strategy based on task demands and environmental constraints, such as limited computational resources.

Figure 5.9: Effect of varying the energy sensitivity coefficient k in the IMTL-EMLP model on (a) prediction performance and (b) total energy consumption, at the midpoint of training. Higher values of k prioritize learning progress over energy efficiency, leading to lower prediction error but increased energy usage, while lower k values reduce energy consumption at the cost of predictive accuracy.



5.2.6 Investigation on Model Architecture and Transfer Learning Dynamics

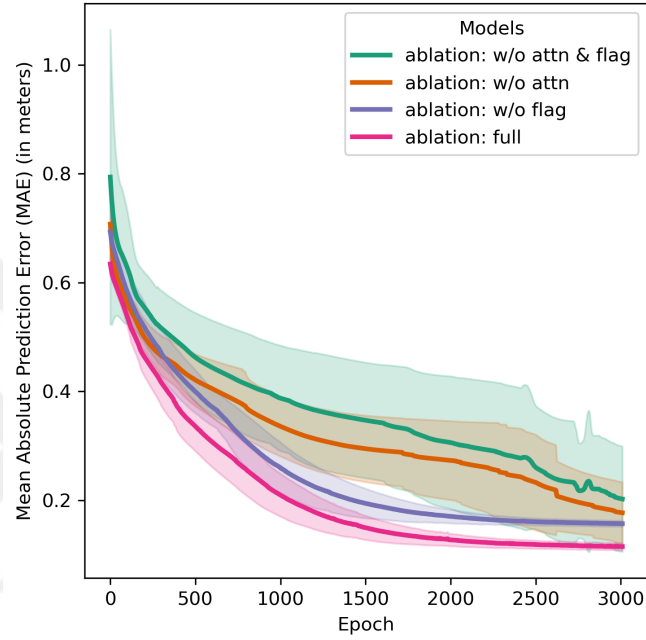
To deconstruct the sources of the model's performance and gain deeper insight into its internal mechanisms, two sets of additional studies were conducted. The first study investigates the individual contributions of the core architectural components responsible for inter-task communication. The second study analyzes the nature of the skill transfer that emerges between tasks as a result of the interleaved learning process.

5.2.6.1 Analysis of Architectural Components

The proposed architecture facilitates knowledge sharing through two key components: a shared multi-head attention (MHA) mechanism with a task-identifying flag bit. To verify that these components are both necessary and complementary, their contributions were systematically evaluated by comparing the full model against three ablated versions: one without the attention layer, one without the task flag, and one without both.

The results of this analysis are presented in Figure 5.10. The findings can be summarized as follows:

Figure 5.10: *An analysis of the individual contributions of the shared attention mechanism and the flag bit within the proposed multi-task learning architecture reveals that neither component alone achieves the convergence efficiency attained by their combined integration in the full model.*



- **Model w/o attention & flag:** The model lacking both the attention layer and the task flag exhibited the poorest performance (Figure 5.10, green). This confirms that a dedicated mechanism to manage the flow of information between tasks is critical. Without it, the model struggles to effectively share knowledge or differentiate between task contexts.
- **Model w/o attention:** The model with only the flag bit without any attention mechanism showed some improvement (Figure 5.10, orange), as the flag provides an explicit signal for task identity. However, without the attention mechanism, the model lacks a dynamic way to weigh the relevance of information from other tasks.

- Model w/o flag: Conversely, the model with only the attention layer without flag performed better (Figure 5.10, purple) than the previous ablation models, demonstrating the power of the attention mechanism to exploit shared information even without an explicit task cue.
- Full model: The full, unablated model consistently outperformed all other configurations (Figure 5.10, pink). This result demonstrates a clear synergistic effect between the two components. The task flag provides the essential context of which task is currently active, while the attention mechanism leverages this context to dynamically arbitrate and integrate the most relevant information from the full set of task representations.

5.2.6.2 Analysis of Knowledge Transfer Object-wise

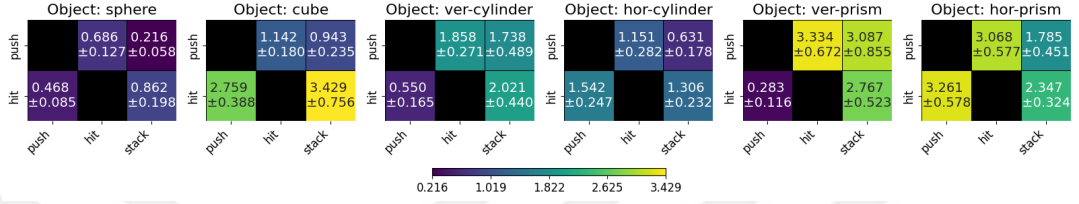
This section investigates the inter-task skill transfer dynamics that emerge from interleaved learning by conducting an ablation study on the shared attention mechanism. Specifically, we quantify the influence of each source task on a given target task by selectively ablating its representation from the key and value matrices of the attention module after the training process is complete.

Let the original loss on a target task, calculated with all task representations present, denoted as $\mathcal{L}_{\text{full}}$. After ablating the representation of a specific source task, we re-evaluate the target task’s loss and denote this new value as $\mathcal{L}_{\text{ablate}}$. The impact of this ablation is measured by the difference:

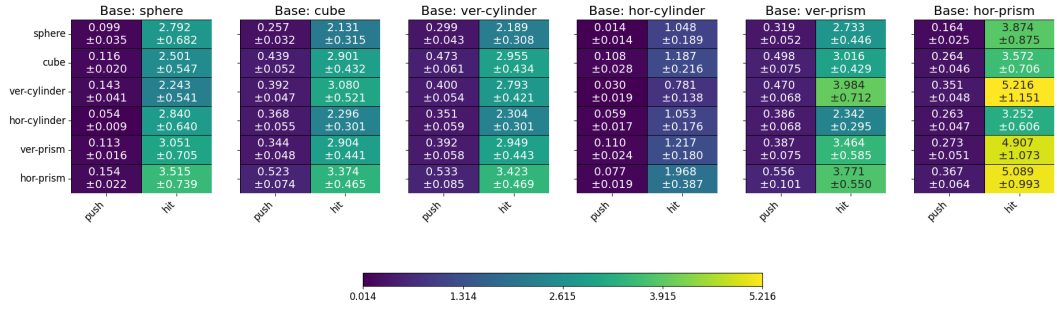
$$\Delta\mathcal{L} = \mathcal{L}_{\text{ablate}} - \mathcal{L}_{\text{full}}.$$

A larger $\Delta\mathcal{L}$ implies that the source task made a significant contribution to the learning of the target task. Conversely, a small or negative $\Delta\mathcal{L}$ indicates limited or negative skill transfer from the ablated source task.

Figure 5.11: Each sub-figure illustrates the impact of ablating source-to-target task representations on the performance of target tasks for individual objects. Specifically, the plots depict the increase in prediction error (loss) for each target task (columns) when the contribution of a corresponding source task (rows) is removed from the key and query components of the shared multi-head attention mechanism. This analysis highlights the extent to which each target task relies on shared information from other tasks, offering insight into the underlying transfer and interference patterns within the multi-task effect prediction framework.



(a) Task-wise performance degradation with per-object data, where rows are target tasks and columns are source tasks respectively.



(b) Stack task performance degradation with per-target object data, where rows are picked objects by the robotic arm and columns are source tasks respectively.

This analysis was performed on the IMTL-LP model across 10 runs with different random seeds. Results report the average and standard deviation of $\Delta\mathcal{L}$ values, with the analysis conducted on a per-object basis to capture object-specific skill transfer characteristics.

We begin with single-object interaction tasks (Push and Hit). Figure 5.12(a) presents a matrix of skill transfer values, where rows represent target tasks and columns denote source tasks. For instance, when interacting with the sphere, the highest skill transfer is observed from Stack to Hit ($\Delta\mathcal{L} = 0.86$), while the transfer from Stack to Push is minimal ($\Delta\mathcal{L} = 0.22$). This suggests that the Hit task relies more on shared features

learned in the Stack task, likely due to commonalities in object dynamics such as rolling or falling.

Objects with rollable affordances (e.g., sphere and horizontal cylinder) generally exhibit lower and more variable transfer values. Their dynamic responses to interactions (especially under Hit) tend to be more sensitive to initial conditions, reducing the consistency of learned representations across tasks. However, when the same object is reoriented—for example, the cylinder or prism shifted from horizontal to vertical—the transfer from Push to Hit increases substantially. For example, the Push→Hit transfer for a horizontal prism is $\Delta\mathcal{L} = 3.26$, whereas it drops to $\Delta\mathcal{L}0.28$ when the prism is vertical. This suggests that orientation affects not only object stability but also the transferability of learned features.

We also evaluate skill transfer for the Stack task, which involves two-object interactions. Here, we fix the base object (the one being stacked upon) and vary the picked object. Figure 5.12(b) shows that the Hit task contributes significantly more than the Push task across most object pairings. The highest transfer values ($\Delta\mathcal{L} \approx 3.7-5.2$) occur when the base object is stable—such as a cube or a vertical cylinder—enabling the stacking to succeed and the model to benefit from prior knowledge of contact and motion dynamics learned in Hit.

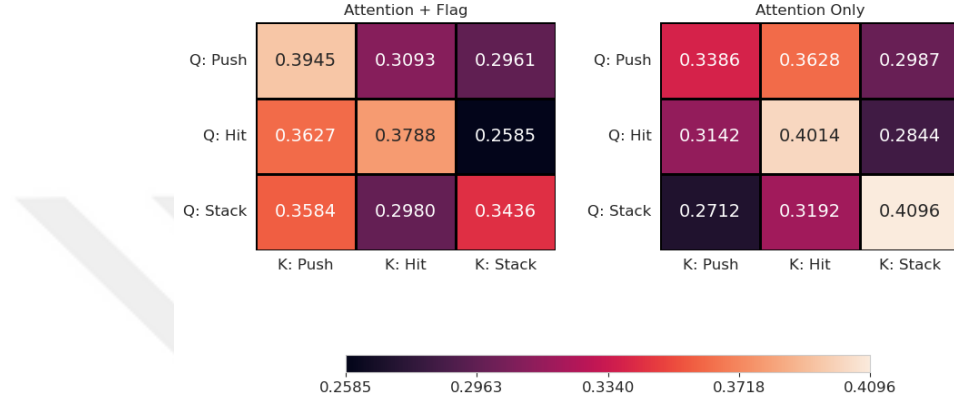
These observations are consistent with the task-specific learning results (see Figure 5.4), where the Push task primarily learns independently, while Hit and Stack co-learn and exhibit stronger bidirectional skill transfer. Notably, despite variation due to random initialization, the standard deviation in the ablation results remains approximately 14%, suggesting a robust and generalizable pattern of emergent task dependencies.

In summary, although the IMTL-LP model is not explicitly designed to enforce skill transfer; its task-agnostic, LP-driven arbitration leads to an emergent structure in the attention space that facilitates meaningful knowledge sharing. The resulting transfer

landscape is shaped not only by task similarity but also by object-specific attributes such as shape, orientation, and physical affordances.

5.2.7 Analysis of Cross-Task Attention and Self Attention of Tasks

Figure 5.12: Attention scores of the Full Model and Model w/o flag. Scores are calculated per-batch average for each training trial. Here mean of 10 trials are given.



In this section, we aim to reveal the learned attention weights. The Figure 5.12 displays heatmaps of the attention scores from the shared multi-head attention module, which is a core component of the proposed neural network architecture designed to facilitate communication and skill transfer between tasks. This analysis compares our proposed model, which uses both the attention mechanism and a flag bit to identify the active task (Full Model), against an ablated version of it that uses the attention mechanism without the flag (Model w/o flag). Recall that, in *SharedMHA* mechanism, the currently training task serves as the “Query” (Q), while the representations from all tasks serve as the “Keys” (K) and “Values” (V). When we first look at the score map on the right side, Attention Only, the most visible feature is that the diagonal values, which represent self-attention (e.g., Q : Hit, K : Hit), are the highest in each row. For instance, when the Hit task is the query, it pays the most attention to its own representation (0.4014). Similarly, the Stack task’s self-attention score is 0.4096, the highest in its row. Without an explicit flag to identify the active task, the model learns a simpler strategy: a task’s own representation

is the most relevant source of information for itself. This strong self-focus might limit the model’s capacity to dynamically leverage knowledge from other tasks, even though the architecture allows for it. This aligns with the results in the Figure 5.10, which shows this configuration performs worse than the Full model.

On the left side of Figure 5.12, we see the attention scores for our proposed model with both attention and flag, i.e., Full Model. Unlike the Model w/o flag, the self-attention scores on the diagonal are not always dominant. As the flag bit explicitly informs the model which task is active, the attention mechanism more effectively weighed the contributions of other tasks. For example, when Hit is the query, its attention is more balanced between its own representation (0.3788) and the Push task’s representation (0.3627). More notably, Push task preferred self-attention, even in the case of flag-bit inclusion. This supports our previous finding (see the Section 5.2.6.2) that the Push task mostly learned on its own, not depending on the other tasks as much as they need it. In addition, attention scores are relatively high when the K is equal to Push task’s representation, compared to other K values. This suggests that attention mechanism extracts relevant information from Push task for cross-task information sharing, helping other tasks to solve state-action-effect mapping. In sum, the visual illustration of the attention scores in Figure 5.12 corroborates with the performance difference observed between the Full Model and Model w/o flag in Figure 5.10.

5.3 Conclusions

This chapter introduced a significant addition to our human-inspired learning framework by integrating a fundamental principle of biological cognition: energy efficiency. Building upon the success of the Learning Progress (LP) guided model, we proposed the Energy-Modulated Learning Progress (EMLP) mechanism, which arbitrates task selection by balancing the drive for competence (high LP) with the need for computational

resource conservation (low energy cost). The experimental results robustly validated this approach. The IMTL-EMLP model demonstrated a learning trajectory and final predictive accuracy that was highly competitive with the pure LP-driven model, yet it achieved this with markedly lower computational energy expenditure. This confirms that an agent can learn efficiently without sacrificing performance, navigating the crucial trade-off between speed and cost.

Furthermore, the introduction of the sensitivity coefficient, k , provides a practical method for tuning the agent’s learning “style,” allowing it to prioritize rapid progress when resources are plentiful or shift to a more conservative, energy-saving mode when computation is constrained. The superior performance of our interleaved methods over blocked-learning baselines reinforced the value of dynamic task-switching, proving that periodic revisit to tasks is key to mitigating catastrophic forgetting and fostering robust knowledge retention. Finally, our architectural and skill-transfer analyses revealed that the combination of a shared attention module and a task-identifying flag is critical for enabling effective knowledge sharing. The EMLP-guided schedule naturally encouraged the model to discover and exploit latent similarities between tasks, leading to an emergent and meaningful transfer of skills. In summary, the IMTL-EMLP framework represents a key step toward creating more biologically plausible and practical learning agents that are not only effective but also resourceful and adaptive to real-world constraints.

6. OVERALL RESULTS AND CONCLUSION

This thesis presented a human-inspired methodology for multi-task learning, moving beyond conventional paradigms by integrating core principles of human cognition: intrinsic motivation, adaptive task-switching, and computational energy efficiency. We developed a novel, modular neural architecture that empowers a learning agent to autonomously regulate its own learning schedule. Task selection is driven not by a fixed curriculum, but by internal signals of learning progress (LP) and neural energy consumption (EC), which together guide the agent toward a flexible, interleaved training regime.

Our key contributions are consolidated within this framework, which demonstrates how human-inspired principles create more effective and efficient learning agents. We first established [182] that an agent guided by Learning Progress (LP) naturally adopts an interleaved schedule, dynamically shifting focus to tasks with the highest learning potential. This self-regulated approach proved highly effective because, by periodically revisiting tasks, it directly mitigates the catastrophic forgetting endemic to sequential training. The result was significantly faster convergence and lower prediction error across all robotic manipulation tasks (pushing, hitting, and stacking) when compared to single-task, random, and blocked baselines. Building on this, our Energy-Modulated Learning Progress (EMLP) model [15] introduced a bio-inspired computational energy cost, forcing the agent to balance its drive for improvement with resource efficiency. The findings here were compelling: the EMLP model achieved a competitive learning trajectory, nearly matching the LP-only model in final accuracy, while operating with substantially greater computational efficiency. This result underscores the importance of the performance-efficiency trade-off fundamental to biological intelligence. The success of these dynamic

scheduling strategies is underpinned by our model’s architecture, where a shared multi-head attention mechanism facilitates robust knowledge transfer. Ablation studies confirmed this, revealing that the interleaved schedule fosters an emergent structure in the attention space where the agent learns to exploit latent physical similarities between tasks, enabling meaningful skill sharing without pre-programmed guidance.

In conclusion, this work successfully demonstrates that by embedding principles of human developmental learning into machine learning systems, we can create agents that are not only effective but also more autonomous and flexible. The proposed framework lays a foundation for developing more scalable and adaptive lifelong learning systems-agents that can acquire, retain, and apply skills with the flexibility and resourcefulness characteristic of human intelligence.

6.1 Contributions to the Literature

This research makes several key contributions to the fields of multi-task learning, continual learning, and developmental robotics:

- **A Novel Framework for Autonomous Interleaved Learning in Robotics:** We introduce a multi-task learning system that, for the first time, applies principles of interleaved learning, especially to the domain of robotic effect prediction. The system autonomously generates its own interleaved training schedule based on Learning Progress (LP), moving beyond the rigid, pre-defined curricula common in machine learning. This provides a cognitively plausible alternative that proves highly effective, demonstrating a practical method for mitigating catastrophic forgetting while significantly accelerating learning and improving generalization in complex robotic manipulation tasks.
- **Energy-Modulated Task Arbitration for Resource-Aware Agents:** We propose the Energy-Modulated Learning Progress (EMLP) model, which uniquely integrates a

measure of computational cost into the task selection process. This is the first model of its kind to formalize and apply the trade-off between performance gain and energy efficiency to multi-task robotic learning. It provides a tunable, bio-inspired mechanism for developing resource-aware agents that can adapt their learning strategy based on computational constraints, a critical step for deployment in real-world, resource-limited scenarios.

- **Demonstration of Emergent Knowledge Transfer via Shared Attention:** We demonstrate that a modular architecture with a shared multi-head attention mechanism, when driven by a self-generated interleaved schedule, can facilitate robust and meaningful knowledge transfer without explicit programming. The system learns to exploit latent physical and dynamic similarities between tasks, leading to an emergent and adaptive skill-sharing structure.

6.2 Future Outlook

The framework and findings presented in this thesis open several promising avenues for future research. A primary direction is scaling the complexity and diversity of tasks. Evaluating the system in environments with a much larger set of tasks, including those with hierarchical dependencies, would test the long-term scalability of the arbitration mechanism and its ability to foster a developmental trajectory. Another key area is the exploration of alternative intrinsic motivation signals beyond learning progress, such as information gain [193], novelty [146, 194], or empowerment [147, 195], which could lead to different exploratory behaviors and learning strategies. From an architectural standpoint, future work could focus on dynamic network structures where inter-task connections are not fixed but are learned or pruned over time, allowing the agent to adapt its knowledge-sharing topology. Finally, the most critical next step is the transition from simulation to the physical world. Deploying this framework on a physical robot would

validate its real-world applicability and introduce new challenges related to sample efficiency, sensor noise, and real-time computation, moving us closer to truly autonomous and lifelong learning machines.



REFERENCES

- [1] A. Gopnik, A. N. Meltzoff, and P. K. Kuhl, “The scientist in the crib: What early learning tells us about the mind,” (*No Title*), 2001.
- [2] E. J. Gibson, “Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge,” *Annual review of psychology*, vol. 39, no. 1, pp. 1–42, 1988.
- [3] D. Kuhn and S. Pearsall, “Developmental origins of scientific thinking,” *Journal of Cognition and Development*, vol. 1, no. 1, pp. 113–129, 2000.
- [4] L. Schulz, “The origins of inquiry: Inductive inference and exploration in early childhood,” *Trends in cognitive sciences*, vol. 16, no. 7, pp. 382–389, 2012.
- [5] E. G. Liquin and T. Lombrozo, “A functional approach to explanation-seeking curiosity,” *Cognitive Psychology*, vol. 119, p. 101276, 2020.
- [6] L. E. Schulz and E. B. Bonawitz, “Serious fun: preschoolers engage in more exploratory play when evidence is confounded,” *Developmental psychology*, vol. 43, no. 4, p. 1045, 2007.
- [7] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, “Multi-task learning for dense prediction tasks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, pp. 3614–3633, 2021.
- [8] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, “Data labeling: An empirical investigation into industrial challenges and mitigation strategies,” in *International conference on product-focused software process improvement*, pp. 202–216, Springer, 2020.
- [9] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural networks*, vol. 113, pp. 54–71, 2019.
- [10] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [11] R. M. Ryan and E. L. Deci, “Intrinsic and extrinsic motivations: Classic definitions and new directions,” *Contemporary educational psychology*, vol. 25, no. 1, pp. 54–67, 2000.
- [12] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, “Intrinsic motivation systems for autonomous mental development,” *IEEE transactions on evolutionary computation*, vol. 11, no. 2, pp. 265–286, 2007.

- [13] W. Kool, J. T. McGuire, Z. B. Rosen, and M. M. Botvinick, “Decision making and the avoidance of cognitive demand,” *Journal of Experimental Psychology: General*, vol. 139, no. 4, pp. 665–682, 2010.
- [14] D. Attwell and S. B. Laughlin, “The energy budget for signaling in the grey matter of the brain,” *Journal of Cerebral Blood Flow & Metabolism*, vol. 21, no. 10, pp. 1133–1145, 2001.
- [15] H. Say, S. E. Ada, E. Ugur, and E. Oztop, “Energy weighted learning progress guided interleaved multi-task learning,” *arXiv preprint arXiv:2504.00707*, 2025.
- [16] D. Kudithipudi, M. Aguilar-Simon, J. Babb, M. Bazhenov, D. Blackiston, J. Bongard, A. P. Brna, S. Chakravarthi Raja, N. Cheney, J. Clune, *et al.*, “Biological underpinnings for lifelong learning machines,” *Nature Machine Intelligence*, vol. 4, no. 3, pp. 196–210, 2022.
- [17] S. Pisupati and Y. Niv, “The challenges of lifelong learning in biological and artificial systems,” *Trends in cognitive sciences*, vol. 26, no. 12, pp. 1051–1053, 2022.
- [18] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.
- [19] R. Lamprecht and J. LeDoux, “Structural plasticity and memory,” *Nature Reviews Neuroscience*, vol. 5, no. 1, pp. 45–54, 2004.
- [20] H. Goldberg, “Growing brains, nurturing minds—neuroscience as an educational tool to support students’ development as life-long learners,” *Brain sciences*, vol. 12, no. 12, p. 1622, 2022.
- [21] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.
- [22] E. Aleixo, J. Colonna, M. Cristo, and E. Fernandes, “Catastrophic forgetting in deep learning: A comprehensive taxonomy,” *Journal of the Brazilian Computer Society*, vol. 30, 08 2024.
- [23] L. Wang, X. Zhang, H. Su, and J. Zhu, “A comprehensive survey of continual learning: Theory, method and application,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024.
- [24] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [25] S. Ruder, “An overview of multi-task learning in deep neural networks,” *arXiv preprint arXiv:1706.05098*, 2017.

- [26] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.
- [27] A. Pascual-Leone, A. Amedi, F. Fregni, and L. B. Merabet, “The plastic human brain cortex,” *Annu. Rev. Neurosci.*, vol. 28, no. 1, pp. 377–401, 2005.
- [28] M. Lövdén, L. Bäckman, U. Lindenberger, S. Schaefer, and F. Schmiedek, “A theoretical framework for the study of adult cognitive plasticity,” *Psychological bulletin*, vol. 136, no. 4, p. 659, 2010.
- [29] F. H. Gage, “Mammalian neural stem cells,” *Science*, vol. 287, no. 5457, pp. 1433–1438, 2000.
- [30] T. V. Bliss and T. Lømo, “Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path,” *The Journal of physiology*, vol. 232, no. 2, pp. 331–356, 1973.
- [31] R. C. Malenka and M. F. Bear, “Ltp and ltd: an embarrassment of riches,” *Neuron*, vol. 44, no. 1, pp. 5–21, 2004.
- [32] Y. Stern, “What is cognitive reserve? theory and research application of the reserve concept,” *Journal of the international neuropsychological society*, vol. 8, no. 3, pp. 448–460, 2002.
- [33] J. L. McClelland, B. L. McNaughton, and R. C. O’Reilly, “Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory,” *Psychological review*, vol. 102, no. 3, p. 419, 1995.
- [34] R. C. O’Reilly and J. L. McClelland, “Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off,” *Hippocampus*, vol. 4, no. 6, pp. 661–682, 1994.
- [35] P. W. Frankland and B. Bontempi, “The organization of recent and remote memories,” *Nature reviews neuroscience*, vol. 6, no. 2, pp. 119–130, 2005.
- [36] Y. Dudai, “The neurobiology of consolidations, or, how stable is the engram?,” *Annu. Rev. Psychol.*, vol. 55, no. 1, pp. 51–86, 2004.
- [37] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, “Neuroscience-inspired artificial intelligence,” *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.
- [38] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

- [39] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, “An empirical investigation of catastrophic forgetting in gradient-based neural networks,” *arXiv preprint arXiv:1312.6211*, 2013.
- [40] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, “Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges,” *Information fusion*, vol. 58, pp. 52–68, 2020.
- [41] S. Auddy, J. Hollenstein, M. Saveriano, A. Rodríguez-Sánchez, and J. Piater, “Continual learning from demonstration of robotics skills,” *Robotics and Autonomous Systems*, vol. 165, p. 104427, 2023.
- [42] Y. Meng, Z. Bing, X. Yao, K. Chen, K. Huang, Y. Gao, F. Sun, and A. Knoll, “Preserving and combining knowledge in robotic lifelong reinforcement learning,” *Nature Machine Intelligence*, pp. 1–14, 2025.
- [43] H. Yang, Y. Zhou, J. Wu, H. Liu, L. Yang, and C. Lv, “Human-guided continual learning for personalized decision-making of autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, 2025.
- [44] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, “Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks,” *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p. 9, 2022.
- [45] G. Cai, J. Zhu, Q. Dai, Z. Dong, X. He, R. Tang, and R. Zhang, “Reloop: A self-correction continual learning loop for recommender systems,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2692–2697, 2022.
- [46] H. Yoo, S. Kang, and H. Tong, “Continual recommender systems,” *arXiv preprint arXiv:2507.03861*, 2025.
- [47] G. M. Van de Ven and A. S. Tolias, “Three scenarios for continual learning,” *arXiv preprint arXiv:1904.07734*, 2019.
- [48] G. M. Van de Ven, T. Tuytelaars, and A. S. Tolias, “Three types of incremental learning,” *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.
- [49] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*, pp. 318–362. MIT Press, 1987.
- [50] S. Grossberg, “Competitive learning: From interactive activation to adaptive resonance,” *Cognitive science*, vol. 11, no. 1, pp. 23–63, 1987.
- [51] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

- [52] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *International conference on machine learning*, pp. 3987–3995, PMLR, 2017.
- [53] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.
- [54] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [55] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.
- [56] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” *Advances in neural information processing systems*, vol. 30, 2017.
- [57] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [58] D. P. Kingma, M. Welling, *et al.*, “Auto-encoding variational bayes,” 2013.
- [59] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, “Gradient based sample selection for online continual learning,” *Advances in neural information processing systems*, vol. 32, 2019.
- [60] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, “Efficient lifelong learning with a-gem,” *arXiv preprint arXiv:1812.00420*, 2018.
- [61] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.
- [62] A. Mallya and S. Lazebnik, “Packnet: Adding multiple tasks to a single network by iterative pruning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.
- [63] N. Y. Masse, G. D. Grant, and D. J. Freedman, “Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 44, pp. E10467–E10475, 2018.
- [64] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
- [65] S. Bozinovski, “Reminder of the first paper on transfer learning in neural networks, 1976,” *Informatica*, vol. 44, no. 3, 2020.

- [66] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *Advances in neural information processing systems*, vol. 27, 2014.
- [67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [68] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- [69] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [70] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [71] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [72] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International conference on machine learning*, pp. 1180–1189, PMLR, 2015.
- [73] S. Lin, L. Yang, D. Fan, and J. Zhang, “Beyond not-forgetting: Continual learning with backward knowledge transfer,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 16165–16177, 2022.
- [74] S. Liu, E. Johns, and A. J. Davison, “End-to-end multi-task learning with attention,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1871–1880, 2019.
- [75] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, “Modeling task relationships in multi-task learning with multi-gate mixture-of-experts,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1930–1939, 2018.
- [76] Y. Xue, X. Liao, *et al.*, “Multi-task learning for classification with dirichlet process priors,” *Journal of Machine Learning Research*, vol. 8, no. 1, 2007.

- [77] T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117, 2004.
- [78] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, “Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3205–3214, 2019.
- [79] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, “Cross-stitch networks for multi-task learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.
- [80] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine learning*, vol. 73, pp. 243–272, 2008.
- [81] A. Maurer, M. Pontil, and B. Romera-Paredes, “Sparse coding for multitask and transfer learning,” in *International conference on machine learning*, pp. 343–351, PMLR, 2013.
- [82] P. Liu, X. Qiu, and X. Huang, “Adversarial multi-task learning for text classification,” *arXiv preprint arXiv:1704.05742*, 2017.
- [83] Y. Shinohara, “Adversarial multi-task learning of deep neural networks for robust speech recognition,” in *Interspeech*, pp. 2369–2372, San Francisco, CA, USA, 2016.
- [84] S. Lee, J. Zhu, and E. Xing, “Adaptive multi-task lasso: with application to eqtl detection,” *Advances in neural information processing systems*, vol. 23, 2010.
- [85] J. Wang and J. Ye, “Safe screening for multi-task feature learning with multiple data matrices,” in *International conference on machine learning*, pp. 1747–1756, PMLR, 2015.
- [86] T. K. Pong, P. Tseng, S. Ji, and J. Ye, “Trace norm regularization: Reformulations, algorithms, and multi-task learning,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489, 2010.
- [87] L. Han and Y. Zhang, “Multi-stage multi-task learning with reduced rank,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.
- [88] K. Crammer and Y. Mansour, “Learning multiple tasks using shared hypotheses,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [89] A. Barzilai and K. Crammer, “Convex multi-task learning by clustering,” in *Artificial Intelligence and Statistics*, pp. 65–73, PMLR, 2015.

- [90] Q. Zhou and Q. Zhao, “Flexible clustered multi-task learning by learning representative tasks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 266–278, 2015.
- [91] Y. Zhang and Q. Yang, “Learning sparse task relations in multi-task learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
- [92] G. Lee, E. Yang, and S. Hwang, “Asymmetric multi-task learning based on task relatedness and loss,” in *International conference on machine learning*, pp. 230–238, PMLR, 2016.
- [93] P. Gong, J. Ye, and C. Zhang, “Robust multi-task feature learning,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 895–903, 2012.
- [94] A. Jalali, S. Sanghavi, C. Ruan, and P. Ravikumar, “A dirty model for multi-task learning,” in *Advances in Neural Information Processing Systems* (J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), vol. 23, Curran Associates, Inc., 2010.
- [95] Y. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu, “Distral: Robust multitask reinforcement learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [96] A. Hallak, D. Di Castro, and S. Mannor, “Contextual markov decision processes,” *arXiv preprint arXiv:1502.02259*, 2015.
- [97] S. Sodhani, A. Zhang, and J. Pineau, “Multi-task reinforcement learning with context-based representations,” in *International Conference on Machine Learning*, pp. 9767–9779, PMLR, 2021.
- [98] A. Hendawy, J. Peters, and C. D’Eramo, “Multi-task reinforcement learning with mixture of orthogonal experts,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [99] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [100] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [101] O. Sener and V. Koltun, “Multi-task learning as multi-objective optimization,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 527–536, 2018.

- [102] H. Tang, J. Liu, M. Zhao, and X. Gong, “Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations,” in *Proceedings of the 14th ACM conference on recommender systems*, pp. 269–278, 2020.
- [103] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, “Conflict-averse gradient descent for multi-task learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 18878–18890, 2021.
- [104] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.
- [105] J. L. Elman, “Learning and development in neural networks: The importance of starting small,” *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.
- [106] A. Pentina, V. Sharmanska, and C. H. Lampert, “Curriculum learning of multiple tasks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5492–5500, 2015.
- [107] G. Hacohen and D. Weinshall, “On the power of curriculum learning in training deep networks,” in *International conference on machine learning*, pp. 2535–2544, PMLR, 2019.
- [108] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, “Automated curriculum learning for neural networks,” in *international conference on machine learning*, pp. 1311–1320, Pmlr, 2017.
- [109] M. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” *Advances in neural information processing systems*, vol. 23, 2010.
- [110] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, “Curriculum self-paced learning for cross-domain object detection,” *Computer Vision and Image Understanding*, vol. 204, p. 103166, 2021.
- [111] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, “Teacher–student curriculum learning,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3732–3740, 2019.
- [112] Y. Fan, F. Tian, T. Qin, X.-Y. Li, and T.-Y. Liu, “Learning to teach,” *arXiv preprint arXiv:1805.03643*, 2018.
- [113] S. Forestier, R. Portelas, Y. Mollard, and P.-Y. Oudeyer, “Intrinsically motivated goal exploration processes with automatic curriculum learning,” *Journal of Machine Learning Research*, vol. 23, no. 152, pp. 1–41, 2022.

- [114] P. Ruvolo and E. Eaton, “Active task selection for lifelong machine learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, pp. 862–868, 2013.
- [115] T. Shu, A. Bhandwaldar, C. Gan, K. Smith, S. Liu, D. Gutfreund, E. Spelke, J. Tenenbaum, and T. Ullman, “Agent: A benchmark for core psychological reasoning,” in *International conference on machine learning*, pp. 9614–9625, PMLR, 2021.
- [116] D. Rohrer and K. Taylor, “The shuffling of mathematics problems improves learning,” *Instructional Science*, vol. 35, no. 6, pp. 481–498, 2007.
- [117] N. Kornell and R. A. Bjork, “Learning concepts and categories: Is spacing the enemy of induction?,” *Psychological Science*, vol. 19, no. 6, pp. 585–592, 2008.
- [118] S. H. Czyż, A. M. Wójcik, P. Solarská, and P. Kiper, “High contextual interference improves retention in motor learning: systematic review and meta-analysis,” *Scientific Reports*, vol. 14, no. 1, p. 15974, 2024.
- [119] C.-H. J. Lin, H.-C. Yang, B. J. Knowlton, A. D. Wu, M. Iacoboni, Y.-L. Ye, S.-L. Huang, and M.-C. Chiang, “Contextual interference enhances motor learning through increased resting brain connectivity during memory consolidation,” *NeuroImage*, vol. 181, pp. 1–15, 2018.
- [120] C.-H. J. Lin, B. J. Knowlton, M.-C. Chiang, M. Iacoboni, P. Udompholkul, and A. D. Wu, “Brain–behavior correlates of optimizing learning through interleaved practice,” *Neuroimage*, vol. 56, no. 3, pp. 1758–1772, 2011.
- [121] J. B. Shea and R. L. Morgan, “Contextual interference effects on the acquisition, retention, and transfer of a motor skill.,” *Journal of Experimental psychology: Human Learning and memory*, vol. 5, no. 2, p. 179, 1979.
- [122] R. Christina and R. Bjork, “Optimizing long-term retention and transfer,” *In the mind’s eye: Enhancing human performance*, pp. 23–56, 1991.
- [123] R. A. Schmidt and R. A. Bjork, “New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training,” *Psychological science*, vol. 3, no. 4, pp. 207–218, 1992.
- [124] D. Rohrer, R. F. Dedrick, and S. Stershic, “Interleaved practice improves mathematics learning.,” *Journal of Educational Psychology*, vol. 107, no. 3, p. 900, 2015.
- [125] J. Samani and S. C. Pan, “Interleaved practice enhances memory and problem-solving ability in undergraduate physics,” *npj Science of Learning*, vol. 6, no. 1, p. 32, 2021.

- [126] D. Mayo, T. Scott, M. Ren, G. Elsayed, K. Hermann, M. Jones, and M. Mozer, “Multitask learning via interleaving: A neural network investigation,” in *44th Annual Meeting of the Cognitive Science Society*, 2023.
- [127] N. Kamra, U. Gupta, and Y. Liu, “Deep generative dual memory network for continual learning,” *arXiv preprint arXiv:1710.10368*, 2017.
- [128] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [129] G. E. Hinton, T. J. Sejnowski, *et al.*, “Learning and relearning in boltzmann machines,” *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, no. 282-317, p. 2, 1986.
- [130] A. Gopnik and H. M. Wellman, “Reconstructing constructivism: causal models, bayesian learning mechanisms, and the theory theory,” *Psychological bulletin*, vol. 138, no. 6, p. 1085, 2012.
- [131] J. Piaget, M. Cook, *et al.*, *The origins of intelligence in children*, vol. 8. International universities press New York, 1952.
- [132] E. Bonawitz, P. Shafto, H. Gweon, N. D. Goodman, E. Spelke, and L. Schulz, “The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery,” *Cognition*, vol. 120, no. 3, pp. 322–330, 2011.
- [133] J. Chu and L. E. Schulz, “Play, curiosity, and cognition,” *Annual Review of Developmental Psychology*, vol. 2, no. 1, pp. 317–343, 2020.
- [134] C. Kidd, S. T. Piantadosi, and R. N. Aslin, “The goldilocks effect: Infants’ preference for stimuli that are neither too predictable nor too surprising,” in *Proceedings of the annual meeting of the cognitive science society*, vol. 32, 2010.
- [135] L. Smith and M. Gasser, “The development of embodied cognition: Six lessons from babies,” *Artificial life*, vol. 11, no. 1-2, pp. 13–29, 2005.
- [136] L. Schulz, T. Kushnir, A. Gopnik, and L. Schulz, “Learning from doing,” *Causal learning: Psychology, philosophy, and computation*, p. 67, 2007.
- [137] R. M. Ryan and E. L. Deci, “Self-determination theory,” in *Encyclopedia of quality of life and well-being research*, pp. 6229–6235, Springer, 2024.
- [138] E. L. Deci and R. M. Ryan, “The ”what” and ”why” of goal pursuits: Human needs and the self-determination of behavior,” *Psychological Inquiry*, vol. 11, no. 4, pp. 227–268, 2000.
- [139] R. M. Ryan and E. L. Deci, “Intrinsic and extrinsic motivations: Classic definitions and new directions,” *Contemporary Educational Psychology*, vol. 25, no. 1, pp. 54–67, 2000.

- [140] W. S. Grolnick and R. M. Ryan, “Autonomy in children’s learning: an experimental and individual difference investigation.,” *Journal of personality and social psychology*, vol. 52, no. 5, p. 890, 1987.
- [141] K. Friston, J. Kilner, and L. Harrison, “A free energy principle for the brain,” *Journal of Physiology-Paris*, vol. 100, no. 1, pp. 70–87, 2006. Theoretical and Computational Neuroscience: Understanding Brain Functions.
- [142] K. Friston, “Learning and inference in the brain,” *Neural Networks*, vol. 16, no. 9, pp. 1325–1352, 2003.
- [143] R. P. Rao and D. H. Ballard, “Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects,” *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.
- [144] K. J. Friston, M. Lin, C. D. Frith, G. Pezzulo, J. A. Hobson, and S. Ondobaka, “Active inference, curiosity and insight,” *Neural computation*, vol. 29, no. 10, pp. 2633–2683, 2017.
- [145] P.-Y. Oudeyer and F. Kaplan, “What is intrinsic motivation? a typology of computational approaches,” *Frontiers in neurorobotics*, vol. 1, p. 108, 2007.
- [146] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *International conference on machine learning*, pp. 2778–2787, PMLR, 2017.
- [147] A. S. Klyubin, D. Polani, and C. L. Nehaniv, “Empowerment: A universal agent-centric measure of control,” in *2005 ieee congress on evolutionary computation*, vol. 1, pp. 128–135, IEEE, 2005.
- [148] X. Huang and J. Weng, “Novelty and reinforcement learning in the value system of developmental robots,” in *Proceedings of the 2nd International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, (Lund, Sweden), Lund University Cognitive Studies, 2002.
- [149] K. Gregor, D. J. Rezende, and D. Wierstra, “Variational intrinsic control,” *arXiv preprint arXiv:1611.07507*, 2016.
- [150] L. Lu, W. Zhang, X. Gu, and J. Chen, “Intrinsic motivation based hierarchical exploration for model and skill learning,” *Electronics*, vol. 9, no. 2, p. 312, 2020.
- [151] A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, “Visual reinforcement learning with imagined goals,” *Advances in neural information processing systems*, vol. 31, 2018.
- [152] C. Colas, P. Fournier, M. Chetouani, O. Sigaud, and P.-Y. Oudeyer, “Curious: intrinsically motivated modular multi-goal reinforcement learning,” in *International conference on machine learning*, pp. 1331–1340, PMLR, 2019.

- [153] C. Colas, T. Karch, N. Lair, J.-M. Dussoux, C. Moulin-Frier, P. Dominey, and P.-Y. Oudeyer, “Language as a cognitive tool to imagine goals in curiosity driven exploration,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 3761–3774, 2020.
- [154] M. E. Raichle and D. A. Gusnard, “Appraising the brain’s energy budget,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10237–10239, 2002.
- [155] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [156] K. Friston, “The free-energy principle: a unified brain theory?,” *Nature reviews neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.
- [157] A. Shenhav, M. M. Botvinick, and J. D. Cohen, “The expected value of control: an integrative theory of anterior cingulate cortex function,” *Neuron*, vol. 79, no. 2, pp. 217–240, 2013.
- [158] A. Westbrook and T. S. Braver, “Cognitive effort: A neuroeconomic approach,” *Cognitive, Affective, & Behavioral Neuroscience*, vol. 15, pp. 395–415, 2015.
- [159] M. Kopicki, S. Zurek, R. Stolkin, T. Moerwald, and J. L. Wyatt, “Learning modular and transferable forward models of the motions of push manipulated objects,” *Autonomous Robots*, vol. 41, no. 5, pp. 1061–1082, 2017.
- [160] D. Nguyen-Tuong and J. Peters, “Local gaussian process regression for real-time model-based robot control,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 380–385, IEEE, 2008.
- [161] M. A. Eren and E. Oztop, “Sample efficient robot learning in supervised effect prediction tasks,” *arXiv preprint arXiv:2412.02331*, 2024.
- [162] A. Ahmetoglu, M. Y. Seker, J. Piater, E. Oztop, and E. Ugur, “Deepsym: Deep symbol generation and rule learning for planning from unsupervised robot interaction,” *Journal of Artificial Intelligence Research*, vol. 75, pp. 709–745, 2022.
- [163] Y. Chen, A. Sipos, M. Van der Merwe, and N. Fazeli, “Visuo-tactile transformers for manipulation,” *arXiv preprint arXiv:2210.00121*, 2022.
- [164] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, “Physics-informed neural networks (pinns) for fluid mechanics: A review,” *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, 2021.
- [165] A. Byravan, F. Leeb, F. Meier, and D. Fox, “Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control,” *arXiv preprint arXiv:1710.00489*, 2017.

- [166] N. Mohajerin and S. L. Waslander, “Multistep prediction of dynamic systems with recurrent neural networks,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3370–3383, 2019.
- [167] A. Tariverdi, V. K. Venkiteswaran, M. Richter, O. J. Elle, J. Tørresen, K. Mathiasen, S. Misra, and Ø. G. Martinsen, “A recurrent neural-network-based real-time dynamic model for soft continuum manipulators,” *Frontiers in Robotics and AI*, vol. 8, p. 631303, 2021.
- [168] A. Serifi, E. Knoop, C. Schumacher, N. Kumar, M. Gross, and M. Bächer, “Transformer-based neural augmentation of robot simulation representations,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3748–3755, 2023.
- [169] K. Zhang, P. Yun, J. Cen, J. Cai, D. Zhu, H. Yuan, C. Zhao, T. Feng, M. Y. Wang, Q. Chen, *et al.*, “Generative artificial intelligence in robotic manipulation: A survey,” *arXiv preprint arXiv:2503.03464*, 2025.
- [170] Y. Jiang, T. Zhang, D. Ho, Y. Bai, C. K. Liu, S. Levine, and J. Tan, “Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2884–2890, IEEE, 2021.
- [171] H.-M. Won, J. Lee, and J. Oh, “Localization meets uncertainty: Uncertainty-aware multi-modal localization,” *arXiv preprint arXiv:2504.07677*, 2025.
- [172] C. Liu, B. Huang, Y. Liu, Y. Su, K. Mai, Y. Zhang, Z. Yi, and X. Wu, “A self-supervised contrastive learning method for grasp outcomes prediction,” in *2023 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 737–742, IEEE, 2023.
- [173] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt, “Modeling of deformable objects for robotic manipulation: A tutorial and review,” *Frontiers in Robotics and AI*, vol. 7, p. 82, 2020.
- [174] B. Aksoy and J. T. Wen, “Collaborative manipulation of deformable objects with predictive obstacle avoidance,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16766–16772, IEEE, 2024.
- [175] S. Mugisha, V. K. Guda, C. Chevallereau, D. Chablat, and M. Zoppi, “Motion prediction with gaussian processes for safe human–robot interaction in virtual environments,” *IEEE Access*, vol. 12, pp. 67470–67485, 2024.
- [176] G. Hoffman, T. Bhattacharjee, and S. Nikolaidis, “Inferring human intent and predicting human action in human–robot collaboration,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, 2024.

- [177] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, “Conditional neural movement primitives,” in *Robotics: Science and Systems*, vol. 10, 2019.
- [178] N. Amirshirzad, M. A. Eren, and E. Oztop, “Context-based echo state networks with prediction confidence for human-robot shared control,” *arXiv preprint arXiv:2412.00541*, 2024.
- [179] L. Li and T. G. Thuruthel, “Bridging vision and touch: advancing robotic interaction prediction with self-supervised multimodal learning,” *Frontiers in Robotics and AI*, vol. 11, p. 1407519, 2024.
- [180] N. Mishra, P. Abbeel, and I. Mordatch, “Prediction and control with temporal segment models,” in *International conference on machine learning*, pp. 2459–2468, PMLR, 2017.
- [181] J. de Heuvel, X. Zeng, W. Shi, T. Sethuraman, and M. Bennewitz, “Spatiotemporal attention enhances lidar-based robot navigation in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4202–4209, 2024.
- [182] H. Say and E. Oztop, “A model for cognitively valid lifelong learning,” in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1–7, 2023.
- [183] J. Schmidhuber, “Curious model-building control systems,” in *Proc. international joint conference on neural networks*, pp. 1458–1463, 1991.
- [184] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning.” <http://pybullet.org>, 2016–2023.
- [185] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 1965.
- [186] H. Levene, “Robust tests for equality of variances,” *Contributions to probability and statistics*, pp. 278–292, 1960.
- [187] Student, “The probable error of a mean,” *Biometrika*, pp. 1–25, 1908.
- [188] G. D. Ruxton, “The unequal variance t-test is an underused alternative to student’s t-test and the mann–whitney u test,” *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, 2006.
- [189] J. Weng, “A theory for mentally developing robots,” in *Proceedings 2nd International Conference on Development and Learning. ICDL 2002*, pp. 131–140, IEEE, 2002.
- [190] T. Mikolov, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, vol. 3781, 2013.

- [191] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [192] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [193] R. Houthoofd, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, “Vime: Variational information maximizing exploration,” *Advances in neural information processing systems*, vol. 29, 2016.
- [194] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, “Exploration by random network distillation,” *arXiv preprint arXiv:1810.12894*, 2018.
- [195] S. Mohamed and D. Jimenez Rezende, “Variational information maximisation for intrinsically motivated reinforcement learning,” *Advances in neural information processing systems*, vol. 28, 2015.



APPENDICES

APPENDIX A. NEURAL NETWORK ARCHITECTURE FOR SINGLE-TASK AND MULTI-TASK EFFECT PREDICTION

Table A.1: *Input and output dimensions for the single-task learning models.*

Module	Layer	In size	Out size
$P_{\text{state}}^{\mathcal{T}_i} : \mathbb{R}^{d_s^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_s}$	FC	9/9/18	4
$F : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_h}$	FC+ReLU	4	4
	FC+ReLU	4	4
$f^{\mathcal{T}_i} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_r}$	FC+ReLU	4	4
	FC+ReLU	4	2
$MHA : \mathbb{R}^{d_r} \rightarrow \mathbb{R}^{d_r}$	MHA	2	2
$P_{\text{action}}^{\mathcal{T}_i} : \mathbb{R}^{d_a^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_a}$	FC	8/8/12	1
$g^{\mathcal{T}_i} : \mathbb{R}^{d_r+d_a} \rightarrow \mathbb{R}^{d_e^{\mathcal{T}_i}}$	FC+ReLU	3	4
	FC+ReLU	4	4
	FC+ReLU	4	4
	FC+ReLU	4	9/9/18

Table A.2: *Input and output dimensions for the multi-task learning models.*

Module	Layer	In size	Out size
$P_{\text{state}}^{\mathcal{T}_i} : \mathbb{R}^{d_s^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_s}$	FC	9/9/18	6
$F : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_h}$	FC+ReLU	6	6
	FC+ReLU	6	4
$f^{\mathcal{T}_i} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_r}$	FC+ReLU	4	4
	FC+ReLU	4	2
$MHA : \mathbb{R}^{d_r+1} \rightarrow \mathbb{R}^{d_r+1}$	MHA	3	3
$P_{\text{action}}^{\mathcal{T}_i} : \mathbb{R}^{d_a^{\mathcal{T}_i}} \rightarrow \mathbb{R}^{d_a}$	FC	8/8/12	1
$g^{\mathcal{T}_i} : \mathbb{R}^{d_r+d_a+1} \rightarrow \mathbb{R}^{d_e^{\mathcal{T}_i}}$	FC+ReLU	4	4
	FC+ReLU	4	4
	FC+ReLU	4	4
	FC+ReLU	4	9/9/18