UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Meta Reinforcement Learning in Self Explanatory Neural Networks for Image Classification

Diploma Thesis

Vasileios Giosis

Supervisor: Dimitrios Rafailidis

June 2025

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Meta Reinforcement Learning in Self Explanatory Neural Networks for Image Classification

Diploma Thesis

Vasileios Giosis

Supervisor: Dimitrios Rafailidis

June 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ενισχυτική Μάθηση Αυτοπροσδιοριζόμενων Νευρωνικών Δικτύων για Αναγνώριση Εικόνων

Διπλωματική Εργασία

Βασίλειος Γιώσης

Επιβλέπων: Δημήτριος Ραφαηλίδης

Ιούνιος 2025

Approved by the Examination Committee:

Supervisor **Dimitrios Rafailidis**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Dimitrios Katsaros**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Georgios Thanos**
Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Rafailidis, for his insightful feedback and support throughout the course of this thesis. His knowledge and experience helped me conduct a comprehensive study on a topic I found personally compelling, while greatly improving my ability to perform structured research and write scientific texts with clarity.

I also extend my gratitude to the members of the thesis committee, Professor Katsaros and Professor Thanos, and would like to thank them for taking the time to evaluate my work.

Finally, I want to thank my family, friends, and all those who supported me throughout this journey. Their patience and belief in my abilities enabled me to stay motivated throughout this academic endeavor.

DISCLAIMER ON ACADEMIC ETHICS
AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Vasileios Giosis

Diploma Thesis

Meta Reinforcement Learning in Self Explanatory Neural Networks for Image Classification

Vasileios Giosis

Abstract

Deep Learning models have excelled in the field of image classification, yet the decision making process undertaken by the networks has not been easily interpretable by humans. In particular, image classification is a complex topic as the high dimensionality of image data further complicates the decision making process. Methods like Quantized Self-Explaining Neural Networks (Q-SENN) have increased interpretability by utilizing sparse decision layers, a small number of features to make a class prediction and quantizing feature-class relationships. In this study, we propose Meta-SENN, a framework integrating Meta Reinforcement Learning (Meta-RL) with Q-SENN to optimize the selected features and improve upon interpretability, while maintaining or improving accuracy. Meta-SENN formulates feature optimization as a Markov Decision Process (MDP) where states represent the current model configuration, actions refer to the sparse quantized features selected and rewards are a balance between accuracy and interpretability metrics. Our experimental evaluation on four datasets for Fine-Grained image classification demonstrate that Meta-SENN outperforms Q-SENN and other baseline methods, achieving small accuracy improvements, while maintaining or enhancing interpretability. More notably, Meta-SENN achieves improvements on the Diversity of the features for class prediction, as well as evenly distributing the class prediction between the features.

Keywords:

Meta Reinforcement Learning, Explainable AI, Self-Explaining Neural Networks, Interpretability in Deep Learning

Διπλωματική Εργασία

Ενισχυτική Μάθηση Αυτοπροσδιοριζόμενων Νευρωνικών Δικτύων για Αναγνώριση Εικόνων

Βασίλειος Γιώσης

Περίληψη

Τα Μοντέλα Βαθιάς μάθησης έχουν διακριθεί στον τομέα της αναγνώρισης εικόνων, ωστόσο η διαδικασία η οποία ακολουθούν για να φτάσουν στην λήψη της απόφασης για την κατηγοριοποίηση της εικόνας δεν είναι εύκολα κατανοητή από τον άνθρωπο. Το πρόβλημα αυτό είναι ακόμα πιο έντονο σε εικόνες με υψηλή λεπτομέρεια όπου η λεπτομέρεια αυτή καθιστά την διαδικασία αυτή ακόμα πιο περίπλοκη και δυσνόητη. Μέθοδοι όπως τα Κβαντισμένα Αυτοεπεξηγούμενα Νευρωνικά Δίκτυα (Q-SENN) έχουν αυξήσει την επεξηγησιμότητα χρησιμοποιώντας διάφορες τεχνικές όπως αραιά στρώματα απόφασης, χαμηλό αριθμό χαρακτηριστικών για την πρόβλεψη και κβαντισμό μεταξύ των σχέσεων χαρακτηριστικών-κλάσεων. Σε αυτή την διπλωματική εργασία, προτείνουμε το Meta-SENN μια αρχιτεκτονική η οποία συνδυάζει την Μέτα Ενισχυτική Μάθηση (Meta-RL) με το Q-SENN για να βελτιστοποιήσει τα επιλεγμένα χαρακτηριστικά και να βελτιώσει την επεξηγισιμότητα του δικτύου, διατηρώντας ή βελτιώνοντας τα επίπεδα ακρίβειας. Το Meta-SENN μοντελοποιεί την βελτιστοποίηση των χαρακτηριστικών ως μια Διαδικασία Απόφασης Markov (MDP), με τις καταστάσεις να αναφέρονται στις τρέχουσες παραμέτρους του μοντέλου, οι ενέργειες αναφέρονται στην επιλογή των καλύτερων χαρακτηριστικών και οι ανταμοιβές οι οποίες λαμβάνουν υπ'όψιν τόσο την ακρίβεια όσο και τις μετρικές επεξηγισιμότητας. Η πειραματική μας αξιολόγηση σε τέσσερα σύνολα δεδομένων τα οποία χρησιμοποιούνται για ταξινόμηση εικόνων με υψηλή λεπτομέρεια δείχνουν ότι το Meta-SENN ξεπερνάει τις επιδόσεις του Q-SENN και άλλων βασικών μεθόδων, με μικρές βελτιώσεις στην ακρίβεια και βελτιώσεις ή ίδιες τιμές επεξηγισιμότητας. Ιδιαίτερα αξιοσημείωτη είναι η βελτίωση στην ποικιλία των χαρακτηριστικών που χρησιμοποιούνται για τις προβλέψεις, όπως επίσης και η καλύτερη κατανομή μεταξύ των χαρακτηριστικών που χρησιμοποιούνται για κάθε πρόβλεψη.

Λέξεις-κλειδιά:

Μέτα Ενισχυτική Μάθηση, Επεξηγούμενη τενχνητή νοημοσύνη, Αυτοπροσδιοριζόμενα Νευρωνικά Δίκτυα, Επεξηγησιμότητα στην Βαθιά Μάθηση

Table of contents

List of figures

List of tables

Abbreviations

AI	Artificial Intelligence
CBM	Concept Bottleneck Models
CNN	Convolutional Neural Network
FGVC	Fine-Grained Visual Classification
MDP	Markov Decision Process
Meta-RL	Meta Reinforcement Learning
Q-SENN	Quantized Self-Explaining Neural Networks
RL	Reinforcement Learning
SENN	Self-Explaining Neural Networks
SLDD	Sparse Low-Dimensional Decision

Chapter 1

Introduction

Image classification is one of the tasks in Computer Vision that Deep Learning models are utilized and has enabled great advancements compared to previous methods [2, 3]. One drawback of Artificial Intelligence models is the lack of understanding behind the decision-making process undertaken by each model. This explanation is desired in multiple real-world applications where AI is present. Fields where an action of an AI agent can have legal implications, such as autonomous driving, require the decision-making process of AI to be explainable [4, 5]. Another field is Healthcare, where explainability in AI is of utmost importance for ensuring transparency, trust and safety in decision-making [6, 7].

However, most architectures lack explainability and their decision-making process is not easily comprehendible by humans, especially on Computer Vision tasks, where the high dimensionality of image data makes it even more complex [8]. Interpretability is the ability of a model to allow humans to understand how inputs are mathematically or logically mapped to outputs [9]. To quantify the interpretability of each model as well as the accuracy, Norrenbrock et al. [1] introduces Fidelity, Diversity and Grounding. Fidelity refers to the ability of the model to preserve relevant information to the input data while maintaining high classification accuracy. Diversity measures the variety of features utilized by the model for predicting different classes. Grounding aligns the internal features learned by the model with human understandable concepts.

One initial approach by Selvaraju et al. [10] uses gradients in the final convolutional layer of convolutional neural networks (CNNs) to produce localization heatmaps for important regions within the image. Another method proposed is Self-Explaining Neural Networks (SENN) that attempts to solve this problem by utilizing linear combinations of comprehensible concepts extracted from the input but is limited in applications and does not yield great accuracy [11]. Several other methods proposed to improve upon this initial approach, mainly by proposing new mechanisms that enhance both interpretability and classification accuracy [12]. Protopool proposed by Rymarczyk et al. [13] introduces a pool of prototypes that are shared by the classes and class predictions are made by classifying inputs based on their similarity to these prototypes. Concept Bottleneck Models (CBM) by Koh et al. [14] tackles the interpretability issue by enabling human intervention within the model, allowing the introduction of human-understandable concepts.

The aforementioned methods, propose several solutions to introduce interpretability without taking into consideration the human cognitive capability. Miller [15] states that humans can effectively process and retain seven plus or minus two discrete items in their short-term memory. Following this rule, the Sparse Low-Dimensional Decision (SLDD) Model introduces a sparse, low-dimensional decision layer that uses only 5 out of the 50 learned features per class [16]. This approach, yields significant improvements in all interpretability metrics, simultaneously maintaining very similar levels of accuracy compared to a model with 2048 features.

Building on the basis of SLDD, Quantized Self-Explaining Neural Network (Q-SENN) increases interpretability by quantizing the relationship between features and classes with a Positive, Negative or Neutral value. This quantization makes the decision-making process of the model simpler, making the learned features easier for human understanding. Moreover, Q-SENN goes through a process of finetuning, enabling considerably higher interpretability with the same or improved accuracy of previous models. Additionally, Q-SENN introduces a method to automate the task of aligning features with human interpretable concepts based on CLIP [1, 17]. Finally, Q-SENN introduces a new metric to measure Grounding, called Dependence which calculates the average dependence of the predicted class on the most important feature. However, we will show that Q-SENN can be improved by further optimizing the final decision layer.

Meanwhile, Reinforcement Learning (RL) has proven to be an effective approach for classification tasks [18, 19]. Lagoudakis and Parr [20] introduce a method that leverages modern classifiers in an approximate policy iteration framework, transforming the RL problem into a classification task. Similarly, Wiering et al. [21] introduce the Max-Min Actor-Critic Learning Automaton algorithm that combines RL with multi-layer percepetrons to solve classification tasks. In particular, Meta-RL has successfully been implemented for improving performance in a variety of applications, including Image Classification [22]. Khodadadeh et al. [23] introduce a Meta-RL framework that improves accuracy on on a variety of few-shot image classification tasks. Meta-RL has been applied to other domains such as robotics [24, 25, 26]. Belmonete-Baeza et al. [27] introduce a framework that aids in the design of Legged Robots with Meta-RL, simultaneously designing the robot hardware with the controlling Software. Control Systems is another domain where Meta-RL has excelled in Controller tuning as well as adaptive control [28, 29, 30, 31]. Natural Language Processing [32, 33, 34] and Healthcare Applications [35, 36, 37] are also fields where Meta-RL has enabled advancements. One major advantage of Meta-RL is its ability to enhance performance using a small amount of data and enables models to adjust to existing tasks by learning from previous experiences [38, 39, 40].

While prior works have utilized Meta-RL in a plethora of domains, its application to self-explaining neural networks remains unexplored. Q-SENN advances interpretability through quantized sparse decision layers and automated concept alignment, yet the feature selection process relies on static optimization techniques such as glm-saga [41]. These methods, while effective lack adaptability to refine feature subsets and do not rely on interpretability and accuracy to optimize the decision layers. Meta-RL addresses these shortcomings by formulating the feature optimization process as a Markov Decision Process (MDP) [42], where an agent learns to adjust the sparse decision layer parameters based on a reward function that takes into consideration classification accuracy as well as interpretability. Utilizing Meta-RL in Q-SENN enables improvements both in accuracy as well interpretability, which is critical for real-world applications requiring both transparency and reliability such as medical diagnostics [6] and autonomous systems [4].

In this study, we present Meta-SENN a method for enhancing the Q-SENN Model by integrating it with a Meta-RL Framework. Considering that Q-SENN utilizes a very small set of features to make predictions, we take advantage of the adaptability and the efficiency of Meta-RL to improve the accuracy of a model when working with a small amount of data. In particular our contribution is summarized as follows:

- We formulate the selection of Q-SENN parameters as an MDP, enabling us to calculate the best features. The RL agent takes into account classification accuracy and interpretability, striking a balance between performance and transparency by optimizing a custom reward function.

- We introduce a policy network to finetune the best set of Q-SENN parameters leveraging Meta-RL and a custom reward function to improve interpretability and maintain or improve accuracy.

In Section 2 of the thesis an overview of related work is provided. Section 3 presents the architecture of the proposed framework. Section 4 presents the experimental results and Section 5 concludes the work.

Chapter 2

Related Work

2.1 Interpretability in Machine Learning for Computer Vision

Interpretability in Deep learning-based image classification has made advancements through architectures that have improved interpretability while simultaneously maintaining good levels of accuracy [43, 44, 45]. One early approach such as SENN by Alvarez-Melis and Jaakkola [11], introduces a simple framework where the models decompose their decisions into simple linear combinations of humanly understandable concepts, such as edges or different textures. Even though this approach is effective for simple tasks, it struggled in complex images due to the constraints arising from the use of linear combinations. ProtoPool addresses these shortcomings by creating a humanly understandable feature set that is learned by the model during training. To make a class prediction the model compares the similarity to these features, which are designed to be semantically meaningful and visually interpretable [13].

Concept Bottleneck Models (CBM) introduces interpretability into existing models. It achieves this by resizing an intermediate layer, where human-defined concepts can be introduced, allowing direct intervention in the model during inference [14]. In contrast, the Sparse Low-Dimensional Decision (SLDD) model takes another approach by limiting the number of features during detection to five and utilizing other methods such as glm-saga and a novel Feature Diversity Loss function to increase sparsity while, simultaneously achieving very high levels of accuracy. This approach greatly increases interpretability as the small number of features is in a range which is humanly comprehensible while also maintaining high accuracy [16, 41, 15].

Q-SENN builds upon the foundation of the SLDD Model and further improves accuracy and interpretability by quantizing the parameters in the Sparse Decision layers, proposed on SLDD, into discrete Positive $(+1)$, Negative (-1) or Neutral values (0). This method, greatly simplifies the human understanding of the model's decision-making process. Additionally, Q-SENN automates the aligning of features with human interpretable concepts with the help of CLIP, simplifying a very labor-intensive task of manually assassinating model-generated features with human-defined concepts [1, 17]. However, the sparsification methods for the final layer of Q-SENN do not take interpretability into consideration. This limitation arises because glm-saga, used for feature selection, prioritizes sparsity and classification accuracy alone neglecting interpretability metrics. Consequently, the selected features may rely on redundant or class-specific attributes, undermining the interpretability of the model.

2.2 Meta-Reinforcement Learning in Neural Networks

Meta-Learning has successfully been used for adapting Neural Networks to tasks as proposed by Finn and Abbeel [40]. More specifically, Meta-RL enables models to learn the task at hand with increased efficiency from the amount of data provided, thus improving already existing architectures without the need to obtain more data [46]. Moreover, Meta-RL has successfully been used in environments with sparse rewards as proposed by Humplik and Galashov [47], where the proposed method leads to an increase in performance in a wide range of environments [29, 31, 27].

Frameworks such as Meta-Q-Learning by Fakkor et al. [24] optimize movement by enabling agents to adapt rapidly to new tasks utilizing prior experience. Meta-Q-Learning (MQL) is an off-policy Meta-RL algorithm by Hausknecht and Stone that extends Q-Learning [48]. The proposed method uses Meta-RL to maximize the average reward across all meta-training tasks and adapts to new tasks by employing a context variable representing past trajectories. MQL demonstrates that Meta-RL is a viable approach to enhancing existing models in a variety of applications.

Integrating Meta-RL on Model Predictive Control (MPC) as proposed by Shin et. al [30] showcases the ability of Meta-RL to improve a control system for vehicle control by using Probabilistic Embeddings for Actor-Critic RL (PEARL) introduced by Rakelly et al. [49]. The proposed method MPC-PEARL employs an off-policy Meta-RL algorithm, training policies using transition samples generated by the MPC controller. This method addresses the drawbacks of MPC to handle obstacle navigation and even allowing the deactivation of the MPC during meta-testing, relying solely on PEARL.

Enhanced Meta RL using Demonstrations (EMRLD) by Rengarajan et al. [50] introduces a Meta-RL framework that tackles sparse-reward environments. The proposed method utilizes demonstration data potentially from sub-optimal policies and from environments where the rewards are only given upon task completion or near-completion for learning. To address this, EMRLD combines RL with behavior cloning to direct the policy towards high-reward regions. RL then optimizes this policy to work with sparse reward signals, demonstrating that Meta-RL can be utilized in sparse environments.

The attributes of Meta-RL described above can greatly benefit architectures such as Q-SENN by enabling further fine-tuning to the existing network. The sparse decision layers within Q-SENN can be Meta-Trained using RL to improve accuracy and interpretability by adapting these parameters to the given data.

Chapter 3

The Proposed Meta-SENN Model

In the proposed Meta-SENN model, we improve upon the base Q-SENN model with a RL agent that selects the best features calculated from each Q-SENN finetuning iteration and finetunes the non-zero values.

3.1 Q-SENN Architecture

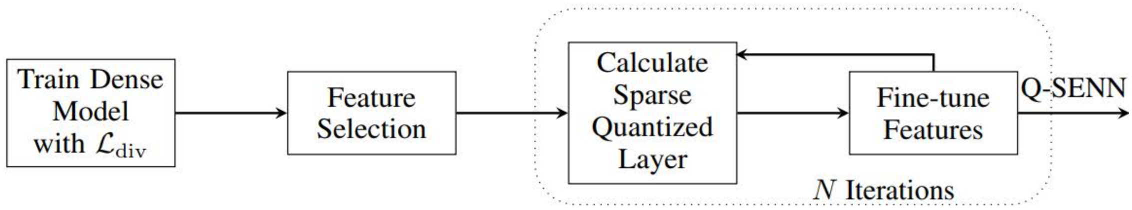

Figure 3.1: Q-SENN Pipeline [1]

3.1.1 Q-SENN Model Training

The Quantized Self-Explaining Neuran Network (Q-SENN) architecture is comprised of two core components as can be seen in Figure 3.1, the backbone Dense model and the Feature selector and the sparse, quantized final layer for classification. The input image $I \in \mathbb{R}^{3\times w\times h}$ is passed through a deep feature extractor, in our case ResNet50 [51], which generates spatial feature maps $M \in \mathbb{R}^{n_f\times w_M\times h_M}$. This model is trained with Feature Diversity Loss function $\mathcal{L}_{\text{div}}$ as introduced by Norrenbrock et al.[16] in SLDD which penalizes highly overlapping feature activations. The final loss function is defined as $\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{CE}} + \beta\mathcal{L}_{\text{div}}$, where $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss as introduced by Mao et al. [52] and β is a weighting coefficient controlling the contribution of the diversity term. Once the dense model is trained, a subset of n_f^* features is selected based on their activation diversity and relevance, following the SLDD framework.

3.1.2 Q-SENN Feature Selection

Using the glm-saga algorithm as introduced by Wong et al. [41], a sparse final classification layer is computed by selecting on average five features per class. These features are then quantized to form a ternary sparse matrix $W^Q \in \{-\alpha, 0, \alpha\}^{n_c \times n_f^*}$ or matrix of ones $W^1 \in \{-1, 0, 1\}^{n_c \times n_f^*}$, where n_c denotes the number of classes and α is the average magnitude of all retained weights. This structure enforces interpretability by encoding each feature-class relation as positively correlated $(+1)$, negatively correlated (-1), or neutral (0). The sparse matrix W_Q is fixed, and the model is finetuned iteratively to adapt the feature extractor to the sparse decision layer. This iterative optimization loop is repeated N epochs.

3.2 Feature Selection using Reinforcement Learning

Following the finetuning of Q-SENN over multiple epochs several feature subsets are generated. We model the task of selecting the most effective features as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ with $\mathcal{S}, \mathcal{A}, \mathcal{R}$ being the state, action and reward sets respectively. $\mathcal{S}$ is the state space, with each step t representing the current Q-SENN metrics and configuration. This includes the current testing loss L_t, Accuracy A_t and the number of features n_f . $\mathcal{A}$ is the action space, comprising all candidate feature subsets obtained from previous iterations as described in 3.1.2. Reward $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, is calculated by evaluating the feature candidate set on test data. It is calculated as:

$$\mathcal{R} = c_1 A_t + c_2 D - c_3 De \tag{3.1}$$

with A_t being Accuracy, D being Diversity and De being Dependence. We select a value of $c_1 = 0.5$, $c_2 = 0.3$ and $c_3 = 0.2$. We choose an even split between accuracy and two metrics that indicate interpretability, to strike a balance between the two main aspects we look to improve upon and not be overly reliant only on one aspect, worsening the performance on the other metrics. Dependence is negative since we want to punish high values. A lower Dependence value indicates that the model uses the features more evenly during detections and does not create a feature that is a single-class detector without any alignment to a human concept. We have decided not to use Alignment, as is introduced in Q-SENN for the reward function, since not all attributes are available on the evaluation datasets.

With this MDP formulation a decaying ϵ-greedy policy is employed [53]. The agent maintains an estimate of cumulative reward for each candidate. At each iteration, with probability ϵ a random candidate is chosen, and with probability $1 - \epsilon$ the candidate with the highest accumulated reward is selected. After each action, the corresponding reward is observed and accumulated. The exploration rate ϵ is decayed gradually to favor exploitation in later iterations. This process prioritizes feature subsets that consistently yield high rewards, allowing the agent to focus on the best-performing subsets.

The feature selection RL loop is summarized in Algorithm 1. Over many iterations, this procedure converges to the feature subset that maximizes the model performance both in accuracy and interpretability.

Algorithm 1 Q-SENN Feature Selection with ϵ-Greedy Policy

Require: Candidate subsets $\{f_i\}_{i=1}^{K}$, total iterations T
Require: Evaluation interval e, initial ϵ, decay rate, minimum $\epsilon_{\min}$
Ensure: Best feature subset index $\hat{\imath}$

Initialize cumulative rewards $R[i] \leftarrow 0$ for $i = 1, \ldots, K$
Set $\epsilon \leftarrow$ initial value
for $t = 1$ to T **do**
    Sample action $a_t \leftarrow$
    $\begin{cases} \text{random index in } \{1, \ldots, K\}, & \text{w.p. } \epsilon \\ \arg\max_i R[i], & \text{w.p. } 1 - \epsilon \end{cases}$
    Update $\epsilon \leftarrow \max(\epsilon_{\min}, \, \epsilon \cdot \text{decay})$
    Apply feature set f_{a_t} to model
    Evaluate model to compute reward $\mathcal{R}_t$ ▷ see Eq. 3.1
    Update reward: $R[a_t] \leftarrow R[a_t] + \mathcal{R}_t$
    if $t \bmod e = 0$ **then**
        for $i = 1$ to K **do**
            Apply f_i and evaluate test reward $\mathcal{R}_i$
        end for
        Let $j \leftarrow \arg\max_i \mathcal{R}_i$
        if $\mathcal{R}_j$ is best so far **then**
            $\hat{\imath} \leftarrow j$
        end if
    end if
end for
return $\hat{\imath}$

3.3 Sparse Decision Layer Optimization using Meta-RL

Once the optimal feature set $\hat{\imath}$ has been calculated through the process described in section 3.2, we proceed with tuning the non-zero weights of the sparse decision layer using Meta-RL. We define an MDP $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, where state $\mathcal{S}_t$ is the current weight vector $\mathbf{W}_t \in \mathbb{R}^N$, containing only the non-zero parameters of the decision layer. An action $\mathcal{A}_t$ corresponds to sampling a new weight vector $\mathbf{W}_{t+1}$ from stochastic policy $\pi_\theta(\mathbf{W_{t+1}}|\mathbf{W_t})$ which we implement as a Gaussian distribution $\mathcal{N}(\mu_\theta(\mathbf{W}_t), \sigma^2 I)$ whose mean $\mu_\theta(\mathbf{W}_t)$ is produced by a policy network π_θ. After applying $\mathbf{W_{t+1}}$ to the model we calculate the reward using the following function:

$$\mathcal{R}_t = c_1 A(\mathbf{W}_{t+1}) + c_2 D(\mathbf{W}_{t+1}) - c_3 De(\mathbf{W}_{t+1}), \tag{3.2}$$

where A is accuracy, D being Diversity and De being Dependence. The policy network π_θ is a feed-forward model mapping $\mathbb{R}^N \to \mathbb{R}^N$, where N is the number of nonzero weights. Our objective is to learn policy parameters θ that maximize the return

$$J(\theta) = \mathbb{E}\Big[\sum_{t=0}^{T-1} \mathcal{R}_t\Big] \tag{3.3}$$

We perform gradient ascent on θ by using the sampled reward and the log-likelihood gradient of the policy:

$$\theta \;\leftarrow\; \theta \;+\; \alpha\, \mathcal{R}_t\, \nabla \log \pi_\theta(\mathbf{W}_{t+1} \mid \mathbf{W}_t), \tag{3.4}$$

with learning rate $\alpha > 0$. In practice, $\pi_\theta(\mathbf{W}' \mid \mathbf{W})$ is implemented as a Gaussian $\mathcal{N}(\mu_\theta(\mathbf{w}), \sigma^2 I)$, with mean μ_θ produced by a small feed-forward network and fixed covariance $\sigma^2 I$. Exploratory noise of scale σ encourages coverage of the weight space. After T iterations, the weights of the sparse decision layer are set to the candidate subset with the highest reward. The process is summarized in algorithm 2.

Algorithm 2 Sparse Layer Weight Tuning with Meta-RL

Require: RL iterations T, learning rate α

```
Extract nonzero weight vector W_0
best_reward ← −∞
best_weights ← W_0
for t = 1 to T do
    Sample W′ using π_θ(W′|W)
    Apply W′ to sparse layer
    Evaluate to obtain reward R_t                ▷ using Eq. 3.2
    if R_t > best_reward then
        best_reward ← R_t
        best_weights ← W′
    end if
    Compute gradient θ                           ▷ using Eq. 3.4
    Backpropagate gradient θ
    Update optimizer
end for
Set model weights ← best_weights
return updated model
```

Chapter 4

Experimental Results

4.1 Datasets

For the evaluation of our model, four datasets for Fine-Grained Image Classification tasks were employed.

The Caltech CUB-200-2011 is a dataset containing 11,788 images of birds, spread across 200 classes, with 5,994 samples for training and 5,774 samples for testing [54]. The dataset also contains annotations relevant to our task such as specific body parts of the bird (e.g., beak, wing, tail) and 312 Binary Attributes and attribute groupings which mostly comprise of a color, pattern, or shape of a particular part. TravellingBirds is another version of the dataset, introduced by Koh in CBM, which comprises of cropped bird classes being randomly assigned to different backgrounds. This assignment is different on the Train and Test parts of the dataset, meaning that the model is measured only on the classification of the object without having the ability to extract irrelevant background features [14].

The third Dataset is the FGVC Aircraft dataset introduced in the paper Fine-Grained Visual Classification of Aircraft by Maji [55]. It comprises of 10,000 images of 100 distinct aircraft models split 70-30% between Training and Testing sets. The final dataset is Stanford-Cars, introduced by in the paper 3-D Object Representations for Fine-Grained Categorization by Krause [56]. This dataset contains 196 classes of Cars with 8,144 used for Training and 8,041 for Testing. As no annotations are provided, no Alignment values can be calculated.

4.2 Evaluation Metrics

To evaluate the model Fidelity, Diversity, Dependence and Alignment are utilized, with 3 of them concerning Interpretability. Fidelity refers to the ability of the model to retain relevant information about the input and is directly connected to the Accuracy of the model.

Out of the 3 metrics to evaluate the model on interpretability, each one tackles a different aspect that is required for a concept to be humanly understandable. Diversity measures the ability to make a prediction using features that are non-overlapping.edw na valw k Considering that 5 features are used to make a prediction, diversity@5 is obtained by applying a softmax function to the 5 highest weighted feature Maps M_5 which yields S_5. Diversity is calculated as follows:

$$\text{diversity@5} = \frac{\sum_{i=1}^{h_M} \sum_{j=1}^{w_M} \max(S_{ij}^1, S_{ij}^2, \ldots, S_{ij}^5)}{5} \tag{4.1}$$

Dependence $\gamma \in [0, 1]$ measures the model's reliance on using the most prominent feature for the classification, in other words, how class-specific is a feature. A low dependence value means that the model bases the class prediction more evenly among the selected features, which in our case are 5, which also indicates that the model has not created some features that are strictly used as class predictors. It is calculated by taking the maximum value from a vector e_i, which contains the weight $\mathbf{W}_{\hat{c},j}$ each feature j has on the prediction of class $\hat{c}$, multiplied by the output f_j of the feature j for that sample. We take the absolute value of each dot product, since we are interested in how each feature impacts the prediction and not if it is positive or negative. Equation 4.3 is the formula of Dependence, with n_T being the number of samples in the Testing set, n_f the total number of features used by the model and e_ij measures the importance of feature j on sample i.

$$e_i = \left\{ \left|\mathbf{W}_{\hat{c},0} \cdot f_0\right|,\ \left|\mathbf{W}_{\hat{c},1} \cdot f_1\right|,\ \ldots,\ \left|\mathbf{W}_{\hat{c},n_f} \cdot f_{n_f}\right| \right\} \tag{4.2}$$

$$\gamma = \frac{1}{n_T} \sum_{i=1}^{n_T} \frac{\max(e_i)}{\sum_{j=1}^{n_f} e_{ij}} \tag{4.3}$$

Finally, Alignment ρ is measured, which is an index of how well the features learned by the model match human-interpretable concepts. For a given attribute a, there are two sets ρ_a^+, samples where attribute a is present and ρ_a^- which contains samples without the attribute a.

The attribute alignment score $a^{gt}_{a,j}$ for an attribute a and feature j is calculated as the difference in average activations between the two groups:

$$a^{gt}_{a,j} = \frac{1}{|\rho_a^+|} \sum_{i \in \rho_a^+} F^{\text{train}}_{i,j} - \frac{1}{|\rho_a^-|} \sum_{i \in \rho_a^-} F^{\text{train}}_{i,j} \tag{4.4}$$

with $\mathbf{F}^{\text{train}} \in \mathbb{R}^{n_t \times n_f}$ being the features belonging to the training set. The average maximum alignment ρ per feature is calculated as follows:

$$r = \frac{1}{n_f^*} \sum_{j=1}^{n_f^*} \frac{n_t}{\sum_{l=1}^{n_t} F^{\text{train}}_{l,j} - \min_l F^{\text{train}}_{l,j}} \max_i a^{gt}_{i,j} \tag{4.5}$$

where n_f^* is the number of features considered, n_t is the number of features on the training set, $\sum_{l=1}^{n_t} F^{\text{train}}_{l,j}$ is the total activation of feature j across the n_t samples and $\min_l F^{\text{train}}_{l,j}$ is the smallest activation value that feature j takes throughout all the n_t samples. A higher value of ρ means that the features learned by the model are more correlated to human concepts.

4.3 Compared Methods

- **Dense**[1] is a ResNet-50 network which is conventionally trained with a fully connected final layer. [51]
- **glm-saga**[2] is a feature selection algorithm to a conventionally trained dense model constrained to 5 features per class. [41]
- **SLDD**[3] is a *Sparse Low-Dimensional Decision* Model with 5 features per class. [16]
- **Q-SENN**[4] is the *Quantized Self-Explaining Neural Network* Model with 5 features per class. [1]
- **Meta-SENN**[5] is the proposed model.

For all baselines, we used the publicly available implementation and kept the parameter tuning as reported. For the rest of baselines, the proposed Meta-SENN model is fixed at 100

[1] `https://github.com/Sudhandar/ResNet-50-model`
[2] `https://github.com/MadryLab/glm_saga`
[3] `https://github.com/ThomasNorr/Q-SENN`
[4] `https://github.com/ThomasNorr/Q-SENN`
[5] `https://github.com/BillGiosis/Meta-SENN`

epochs for the feature selection and the parameter optimization. For the feature optimization part, the AdamW optimizer [57] is used with varying learning rates as described in Section 4.5. The policy network consists of three linear layers with LeakyReLu activation functions [58] and varying Dropout configurations which is also described in Section 4.5. All experiments were conducted on a machine equipped with an NVIDIA RTX 3070 GPU (8GB VRAM), AMD Ryzen 5 5600X CPU and 32GB of RAM. The operating system used is Ubuntu 22.04 LTS with Python 3.10, PyTorch 2.5.1 and CUDA 12.8.

4.4 Performance Evaluation

In Tables 4.1, 4.2, 4.3 and 4.4, the experimental results on the four metrics are presented, respectively. Alignment is calculated only for the CUB-200-2011 and TravelingBirds Datasets since not all attributes are available on the evaluation datasets. The proposed model outperforms the baseline models on most of the datasets and metrics, with only few occasions where the performance is preserved or underperforms.

Table 4.1: Average Accuracy (%). Bold values indicate the best method.

Dataset	Dense	glm-saga	SLDD	QSENN	Meta-SENN
CUB2011	**85.81** ± **0.3**	79.65 ± 0.16	84.23 ± 0.17	84.52 ± 0.17	85.363 ± 0.316
FGVC Aircraft	82.3 ± 0.14	77.72 ± 0.16	81.56 ± 0.14	81.45 ± 0.23	**82.861** ± **0.282**
Stanford Cars	**93.05** ± **0.43**	88.29 ± 0.06	92.9 ± 0.04	92.54 ± 0.12	92.469 ± 0.269
TravelingBirds	44.5 ± 1.18	39.43 ± 1.14	68.86 ± 0.25	69.35 ± 0.45	**70.259** ± **0.407**

Table 4.2: Average Diversity@5 (%). Bold values indicate the best method.

Dataset	Dense	glm-saga	SLDD	QSENN	Meta-SENN
CUB2011	49.33 ± 0.28	45.11 ± 0.14	73.86 ± 0.19	85.53 ± 0.3	**88.833** ± **0.655**
FGVC Aircraft	51.02 ± 0.39	49.31 ± 0.78	72.9 ± 0.14	83.07 ± 1	**88.153** ± **0.695**
Stanford Cars	47.52 ± 0.34	45.73 ± 0.07	76.63 ± 0.07	88.89 ± 0.44	**89.172** ± **0.943**
TravelingBirds	49.6 ± 1.57	50.5 ± 0.73	70.29 ± 0.33	85.15 ± 0.46	**87.764** ± **0.597**

Table 4.3: Average Dependence (%). Bold values indicate the best method.

Dataset	Dense	glm-saga	SLDD	QSENN	Meta-SENN
CUB2011	**12.79** ± **0.76**	42.1 ± 0.01	42.32 ± 0.03	26.21 ± 0.14	26.187 ± 0.123
FGVC Aircraft	**24.76** ± **0.53**	44.19 ± 0.04	47.42 ± 0.02	29.77 ± 0.4	28.063 ± 0.097
Stanford Cars	**22.6** ± **0.7**	41.5 ± 0.02	41.6 ± 0.02	26.19 ± 0.24	25.221 ± 0.175
TravelingBirds	**16** ± **0.59**	43.28 ± 0.22	43.41 ± 0.03	26.76 ± 0.12	26.470 ± 0.197

Table 4.4: Average Alignment. Bold values indicate the best method.

Dataset	Dense	glm-saga	SLDD	QSENN	Meta-SENN
CUB2011	1.12 ± 0.042	1.19 ± 0.003	2.35 ± 0.058	2.38 ± 0.047	**2.402** ± **0.043**
TravelingBirds	1.04 ± 0.047	1.17 ± 0.026	**2.23** ± **0.027**	2.17 ± 0.037	2.217 ± 0.043

As shown in Table 4.1, Meta-SENN marignally outperforms Q-SENN with 0.8-1 % improvements in three out of the four datasets. Meta-SENN maintains competitive accuracy compared to the Dense, non-interpretable, model with significant accuracy gains in one dataset. We observe that Meta-SENN can improve upon the accuracy of the most competitive interpretable model, simultaneously yielding accuracy comparable to the non-interpretable Dense model.

In Table 4.2, it is shown that Meta-SENN consistently outperforms all other baselines on all datasets. In particular, Meta-SENN yields improvements of 2-7% compared to the second best method (Q-SENN) on three out of four datasets and far surpassing the other two interpretable models glm-saga and SLDD. We observe that most interpretable models outperform the non-interpretable Dense model on diversity. These gains demonstrate that apart from achieving similar accuracy as the Dense model, Meta-SENN bases prediction on a greater variety of features compared to all other models, enhancing interpretability.

As observed in Table 4.3, the Dense model outperforms all interpretable models with up to 50% better performance. This result is expected because Dense uses a very large number of features to make a prediction compared to all interpretable models, meaning that it does not require to base the prediction on one or two main features but uses multiple features to make the class prediction. Out of the interpretable models that use a small number of features, Meta-SENN outperforms the next best baseline (Q-SENN) on three out of the four datasets. This demonstrates that Meta-SENN can distribute the class prediction more evenly across features and does not tend to create features that works as class predictors.

On inspection of Table 4.4, Alignment shows how well the model predictions align with human concepts. Meta-SENN achieves the highest alignment score, outperforming all baselines on CUB2011. On TravelingBirds, SLDD is the leading method with Meta-SENN being competitive with a difference of 0.06. Overall, Meta-SENN enhances interpretability by achieving similar performance or improving the performance of other interpretable baseline models in the three interpretability metrics considered. Moreover, Meta-SENN maintains competitive accuracy with the performance of the Dense model and slightly improves the accuracy of the baseline interpretable models.

4.5 Parameter Tuning

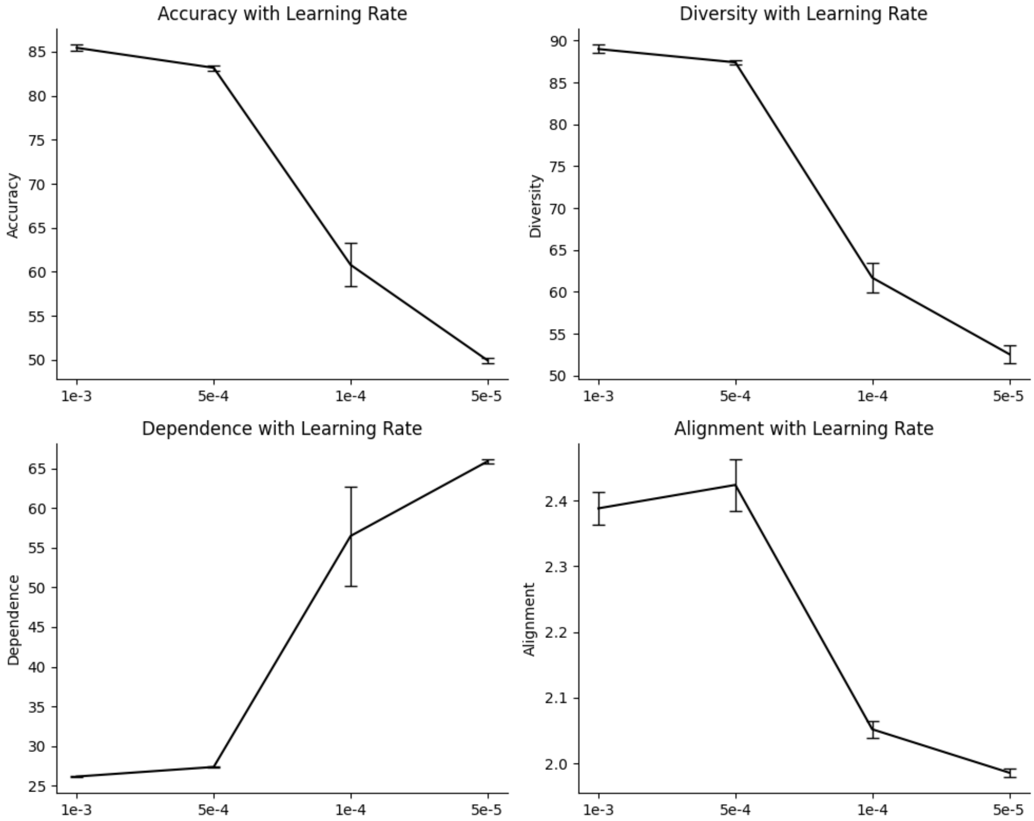

Figure 4.1: Performance evaluation on the CUB2011 dataset when varying the Learning Rates.

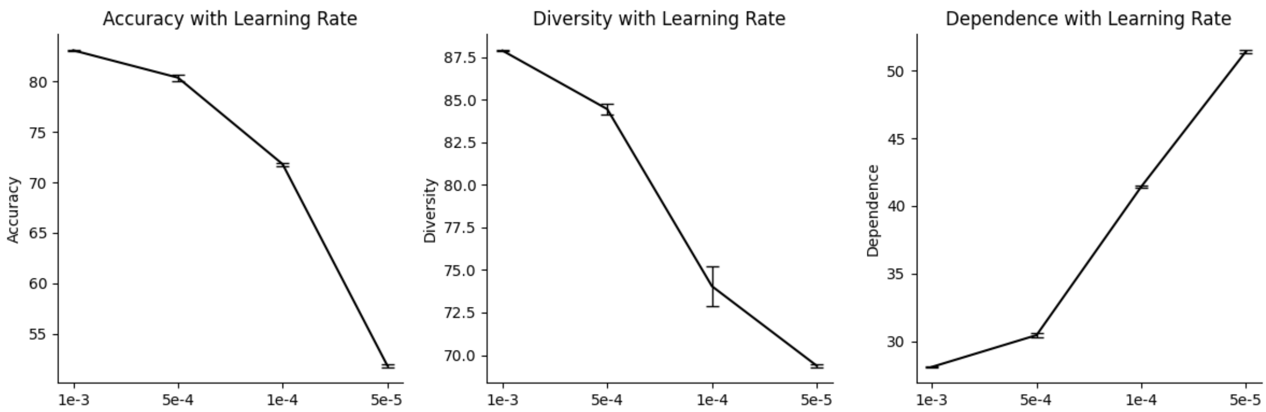

Figure 4.2: Performance evaluation on the FGVC Aircraft dataset when varying Learning Rates.

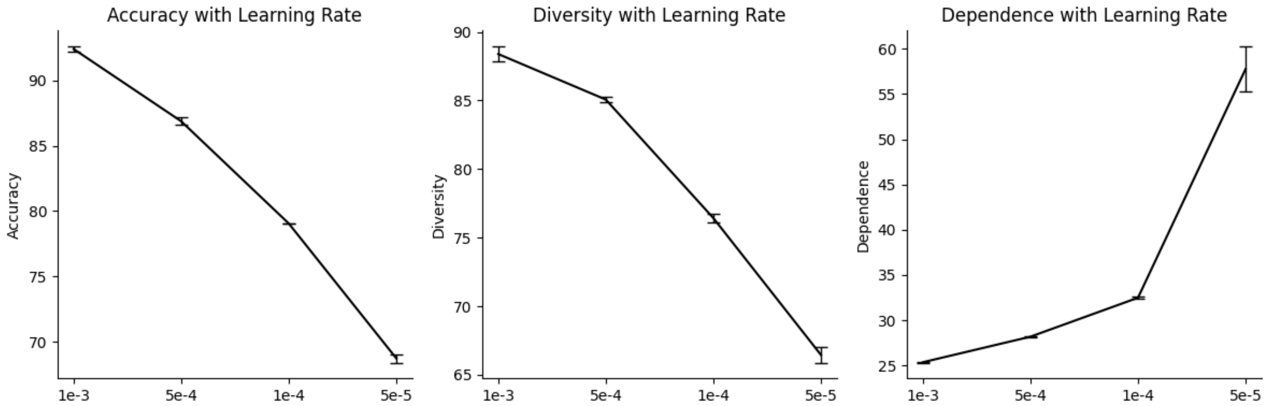

Figure 4.3: Performance evaluation on the Stanford Cars dataset when varying Learning Rates.

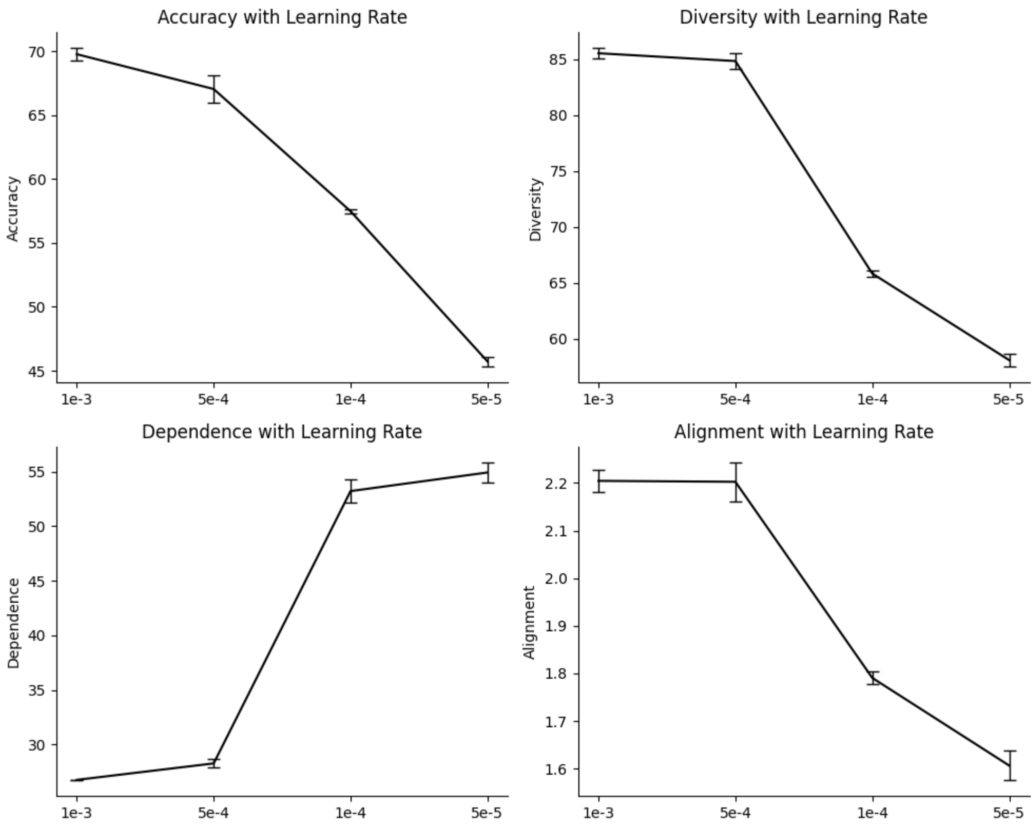

Figure 4.4: Performance evaluation on the Travelling Birds when varying Learning Rates.

To examine the impact of learning rate on our model, four learning rates have been examined ($10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}$). As can be observed in Figures 4.1, 4.2, 4.3 and 4.4, lower learning rates indicate that the Meta-SENN policy network struggles to adapt the sparse decision layer. When the learning rate is set to 5×10^{-5}, all four datasets exhibit relatively low classification accuracy with Diversity, Dependence and Alignment considerably underperfoming. As the learning rate increases to 10^{-4}, the model performance improves slightly but still struggles to reach higher classification accuracy and Diversity while Dependence and Alignment have marginal improvements.

When the learning rate is further increased to 5×10^{-4}, all four datasets demonstrate a clear uplift across all metrics. Accuracy rises noticeably with improvements ranging from 8-20%, depending on the dataset. Diversity also follows with improvements of 9-23% and Dependence is almost halved in three out of the four datasets. Alignment is also greatly improved with the performance on the CUB2011 Dataset surpassing all other learning rates. The highest rate of 10^{-3} outperforms all other rates across all of the metrics with the exception of Alignment in CUB2011. The performance of Meta-SENN across these learning rates indicates that higher learning rates enable the Meta-RL policy network to adapt its sparse layer more effectively.

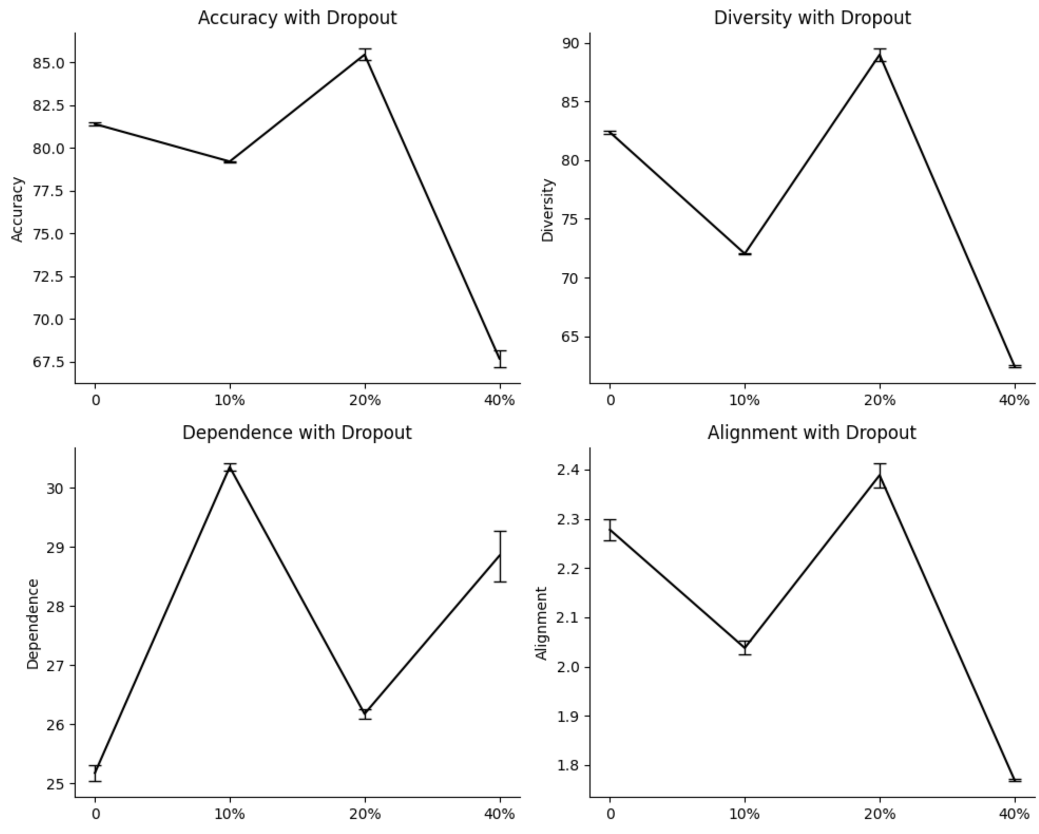

Figure 4.5: Performance evaluation on the CUB2011 dataset when varying the Dropout Rates.

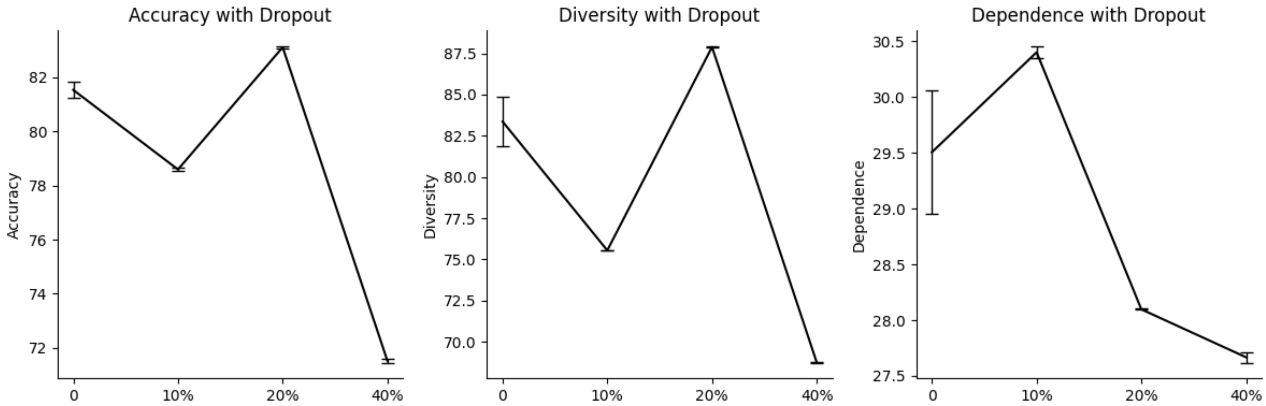

Figure 4.6: Performance evaluation on the FGVC Aircraft dataset when varying the Dropout Rates.

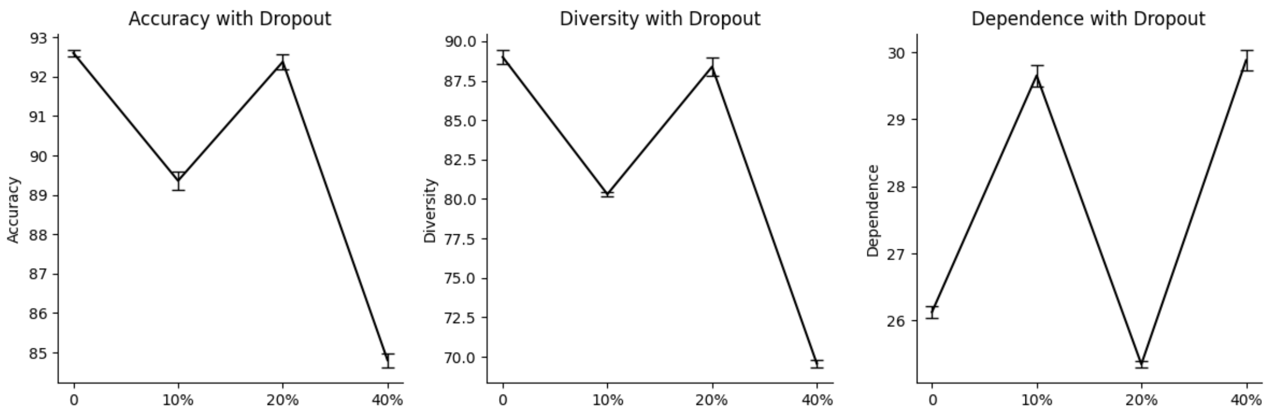

Figure 4.7: Performance evaluation on the Stanford Cars dataset when varying the Dropout Rates.

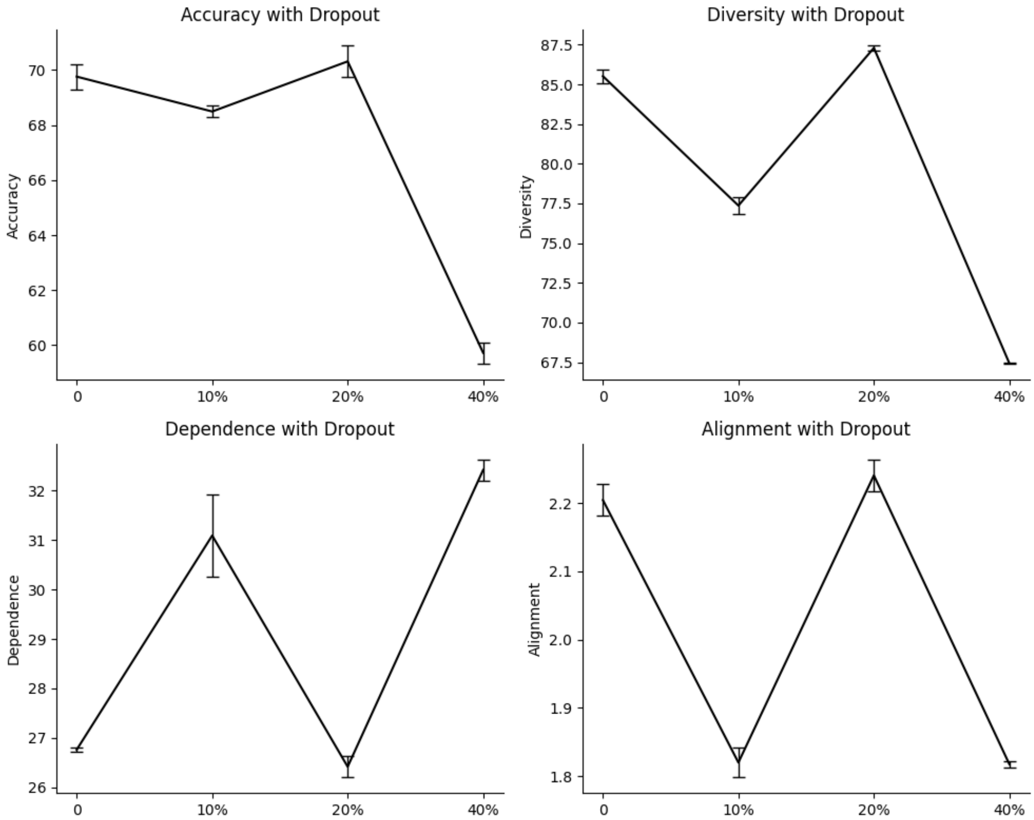

Figure 4.8: Performance evaluation on the Traveling Birds dataset when varying the Dropout Rates.

To examine the impact of dropout probability on our model, four dropout rates have been examined (0.0, 0.1, 0.2, 0.4). As can be observed in Figures 4.5, 4.6, 4.7 and 4.8, the highest value of dropout examined underperforms on most metrics. This result can be attributed to the fact that there is a high number of randomly deactivated neurons, which hinders the ability of the Meta-RL policy network to learn from the sparse layer parameters. At a dropout rate of 0.1, the Meta-RL policy network improves on the classification accuracy compared to the highest Dropout rate as well as increasing diversity. Dependence is also reduced with the exception of CUB2011 where it is increased compared to 0.4 Dropout rate. Alignment is also improved on the CUB2011 while being almost identical on the TravelingBirds dataset. This indicates that a Dropout of 0.4 is a rate that causes great information loss during training, while 0.1 incurs only a minor negative impact on accuracy and interpretability.

When not using Dropout, the Meta-RL policy network produces high accuracy, even surpassing the best rate on the Stanford Cars dataset. Interpretability is also improved, with lower

Dependence compared to the two previous rates examined as well as improved Diversity and Alignment. With a dropout rate of 0.2, the policy network outperforms all the other examined rates with some cases where the network without dropout achieves similar or higher classification accuracy and interpretability metrics. Overall, a learning rate of 10^{-3} is chosen and a dropout rate of 0.2, which yields the best performance across the examined metrics.

Chapter 5

Conclusion

In this study, we presented Meta-SENN, a Meta-RL framework to enhance Quantized Self Explaining Neural Networks. The key idea is to select and optimize the best features and quantized sparse decision layers of Q-SENN using a RL framework. By formulating the Meta-RL task as an MDP, Meta-SENN optimizes the features, increasing both interpretability metrics and accuracy. Our experimental evaluation on four datasets demonstrated Meta-SENN's effectiveness:

- **Accuracy**: Meta-SENN achieved modest but consistent improvements over Q-SENN, with gains of **+0.84%** on CUB-2011 and **+0.91%** on Traveling Birds.
- **Interpretability**: The framework enhanced **Diversity@5** by **2-7%** on all datasets and reduced **Dependence** by up to **2.7%** compared to other interpretable models, indicating more balanced feature utilization.
- **Alignment**: On datasets with attribute annotations, alignment with human-understandable concepts improved by **1.8%**, proving the ability of the model to make features more relevant to human concepts.

By embedding Meta-RL in Self-Explaining Neural Networks, we can further improve upon interpretability, a feature which is desired on a variety of applications such as the medical field. This work shows that Meta-RL is not only useful for achieving higher accuracy metrics but also it can help make AI more humanly understandable.

Bibliography

[1] Thomas Norrenbrock, Marco Rudolph, and Bodo Rosenhahn. Q-senn: Quantized self-explaining neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21482–21491, 2024.

[2] Maria Trigka and Elias Dritsas. A comprehensive survey of deep learning approaches in image processing. *Sensors*, 25(2):531, 2025.

[3] Bo Dong and Xue Wang. Comparison deep learning method to traditional methods using for network intrusion detection. In *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pages 581–585, 2016.

[4] Stuart Russell, Daniel Dewey, and Max Tegmark. Research priorities for robust and beneficial artificial intelligence. *AI Mag.*, 36(4):105–114, December 2015.

[5] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Impact of legal requirements on explainability in machine learning, 2020.

[6] Khansa Rasheed, Adnan Qayyum, Mohammed Ghaly, Ala Al-Fuqaha, Adeel Razi, and Junaid Qadir. Explainable, trustworthy, and ethical machine learning for healthcare: A survey. *Computers in Biology and Medicine*, 149:106043, 2022.

[7] Olya Rezaeian, Alparslan Emrah Bayrak, and Onur Asan. Explainability and ai confidence in clinical decision support systems: Effects on trust, diagnostic performance, and cognitive load in breast cancer care, 2025.

[8] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. *Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges*, page 417–431. Springer International Publishing, 2020.

[9] Lei Gao and Ling Guan. Interpretability of machine learning: Recent advances and future prospects, 2023.

[10] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.

[11] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *arXiv preprint arXiv:1806.07538*, 2018. NeurIPS 2018.

[12] Meike Nauta, Johannes H. Hegeman, Jeroen Geerdink, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. Interpreting and correcting medical image classification with pip-net. *arXiv preprint arXiv:2307.10404*, 2023.

[13] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. *arXiv preprint arXiv:2112.02902*, 2022.

[14] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 2020.

[15] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81, 1956.

[16] Thomas Norrenbrock, Marco Rudolph, and Bodo Rosenhahn. Take 5: Interpretable image classification with a handful of features. In *Progress and Challenges in Building Trustworthy Embodied AI*, 2022.

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th*

International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[18] Majid Ghasemi and Dariush Ebrahimi. Introduction to reinforcement learning, 2024.

[19] Kevin Murphy. Reinforcement learning: A comprehensive overview, 2025.

[20] Michail G. Lagoudakis and Ronald Parr. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, pages 424–431, 2003.

[21] Marco A. Wiering, Hado van Hasselt, Auke-Dirk Pietersma, and Lambert Schomaker. Reinforcement learning algorithms for solving classification problems. Technical report, University of Groningen, 2000.

[22] Yaohui Zhu, Chenlong Liu, and Shuqiang Jiang. Multi-attention meta learning for few-shot fine-grained image recognition. pages 1090–1096, 07 2020.

[23] Siavash Khodadadeh, Ladislau Bölöni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification, 2019.

[24] Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J. Smola. Meta-q-learning, 2020.

[25] Xu Han and Feng Wu. Meta reinforcement learning with successor feature based context, 2022.

[26] Gerrit Schoettler, Ashvin Nair, Juan Aparicio Ojea, Sergey Levine, and Eugen Solowjow. Meta-reinforcement learning for robotic industrial insertion tasks, 2020.

[27] Alvaro Belmonte-Baeza, Joonho Lee, Giorgio Valsecchi, and Marco Hutter. Meta reinforcement learning for optimal design of legged robots. *IEEE Robotics and Automation Letters*, 7(4):12134–12141, October 2022.

[28] Daniel McClement, Nathan Lawrence, Philip Loewen, Michael Forbes, Johan Backstrom, and R. Gopaluni. A meta-reinforcement learning approach to process control. *IFAC-PapersOnLine*, 54:685–692, 01 2021.

[29] Daniel G. McClement, Nathan P. Lawrence, Michael G. Forbes, Philip D. Loewen, Johan U. Backström, and R. Bhushan Gopaluni. Meta-reinforcement learning for adaptive control of second order systems, 2022.

[30] Jaeuk Shin, Astghik Hakobyan, Mingyu Park, Yeoneung Kim, Gihun Kim, and Insoon Yang. Infusing model predictive control into meta-reinforcement learning for mobile robots in dynamic environments. *IEEE Robotics and Automation Letters*, 7(4):10065–10072, October 2022.

[31] Daniel G. McClement, Nathan P. Lawrence, Johan U. Backström, Philip D. Loewen, Michael G. Forbes, and R. Bhushan Gopaluni. Meta-reinforcement learning for the tuning of pi controllers: An offline approach. *Journal of Process Control*, 118:139–152, 2022.

[32] Shengyao Luo. Applications and challenges of meta-learning in natural language processing. *Applied and Computational Engineering*, 109:1–8, 11 2024.

[33] Hung-yi Lee, Ngoc Thang Vu, and Shang-Wen Li. Meta learning and its applications to natural language processing. In David Chiang and Min Zhang, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 15–20, Online, August 2021. Association for Computational Linguistics.

[34] Yingli Shen and Xiaobing Zhao. Reinforcement learning in natural language processing: A survey. In *Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing*, MLNLP '23, page 84–90, New York, NY, USA, 2024. Association for Computing Machinery.

[35] Sijia Liu, Kenneth See, Kee Yuan Ngiam, Leo Anthony Celi, Xuefeng Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: Comprehensive review. *Journal of Medical Internet Research*, 22(7):e18477, 2020.

[36] Alireza Rafiei, Ronald Moore, Sina Jahromi, Farshid Hajati, and Rishikesan Kamaleswaran. Meta-learning in healthcare: A survey. *SN Computer Science*, 5(6), August 2024.

[37] Akshay Smit, Damir Vrabac, Yujie He, Andrew Y. Ng, Andrew L. Beam, and Pranav Rajpurkar. Medselect: Selective labeling for medical image classification combining meta-learning with deep reinforcement learning, 2021.

[38] Xiu-Shen Wei, Peng Wang, Lingqiao Liu, Chunhua Shen, and Jianxin Wu. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Transactions on Image Processing*, 28(12):6116–6125, December 2019.

[39] Stefanos Antaris, Dimitrios Rafailidis, and Sarunas Girdzijauskas. Meta-reinforcement learning via buffering graph signatures for live video streaming events, 2021.

[40] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[41] Eric Wong, Shibani Santurkar, and Aleksander Mądry. Leveraging sparse linear layers for debuggable deep networks, 2021.

[42] Chelsea C. White and Douglas J. White. Markov decision processes. *European Journal of Operational Research*, 39(1):1–16, 1989.

[43] Vidhya Kamakshi and Narayanan C. Krishnan. Explainable image classification: The journey so far and the road ahead. *AI*, 4(3):620–651, 2023.

[44] Biagio La Rosa, Roberto Capobianco, and Daniele Nardi. A self-interpretable module for deep image classification on small data. *Applied Intelligence*, 53(8):9115–9147, Apr 2023.

[45] Samparthi V.S. Kumar and Hari Kishan Kondaveeti. Towards transparency in ai: Explainable bird species image classification for ecological research. *Ecological Indicators*, 169:112886, 2024.

[46] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning, 2024.

[47] Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A. Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference, 2019.

[48] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps, 2017.

[49] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables, 2019.

[50] Desik Rengarajan, Sapana Chaudhary, Jaewon Kim, Dileep Kalathil, and Srinivas Shakkottai. Enhanced meta reinforcement learning using demonstrations in sparse reward environments, 2022.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[52] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications, 2023.

[53] Aakash Maroti. Rbed: Reward based epsilon decay, 2019.

[54] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, August 2023.

[55] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[56] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[58] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network, 2015.